

UC San Diego

UC San Diego Electronic Theses and Dissertations

Title

Enhancing photographs using content-specific image priors

Permalink

<https://escholarship.org/uc/item/1p85v1k6>

Author

Joshi, Neel Suresh

Publication Date

2008

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA, SAN DIEGO

**ENHANCING PHOTOGRAPHS USING
CONTENT-SPECIFIC IMAGE PRIORS**

A dissertation submitted in partial satisfaction of the
requirements for the degree

Doctor of Philosophy

in

Computer Science

by

Neel Suresh Joshi

Committee in charge:

Professor David J. Kriegman, Chair
Professor Samuel R. Buss
Professor Henrik Wann Jensen
Professor Falko Kuester
Professor Matthias Zwicker

2008

Copyright
Neel Suresh Joshi, 2008
All rights reserved.

The dissertation of Neel Suresh Joshi is approved,
and it is acceptable in quality and form for publica-
tion on microfilm and electronically:

Chair

University of California, San Diego

2008

To Kunda and Suresh

EPIGRAPH

All our knowledge has its origins in our perceptions.

—Leonardo da Vinci

TABLE OF CONTENTS

Signature Page		iii
Dedication		iv
Epigraph		v
Table of Contents		vi
List of Figures		ix
List of Tables		xii
Acknowledgements		xiii
Vita and Publications		xvi
Abstract of the Dissertation		xvii
1 Introduction		1
1.1 Summary of Original Contributions		3
1.2 Organization of the Dissertation		5
2 Previous Work		6
2.1 Deblurring		7
2.1.1 Image Blur Overview		9
2.1.2 Image Blur Model		11
2.1.3 Image Deconvolution		12
2.1.4 PSF Estimation by Constraining the PSF		16
2.1.5 PSF Estimation by Constraining the Image		25
2.1.6 Multi-Image Blur Estimation		34
2.1.7 Image Deblurring Summary		37
2.2 Denoising		38
2.2.1 Basic Filtering Methods		39
2.2.2 Edge-Preserving Methods		40
2.2.3 Using Image Priors		41
2.3 Up-sampling		43
2.3.1 Image Invariant Filters		44
2.3.2 Image Dependent Filters		44
2.3.3 Using Image Priors		46
2.4 White-Balancing		48
2.4.1 Estimating Illumination Color		49
2.4.2 Color Matching		53

3	PSF Estimation using Sharp Edge Prediction	55
3.1	Related Work	57
3.2	Image Formation Model	59
3.3	Sharp Image Estimation	60
3.3.1	Blind Estimation	60
3.3.2	Non-Blind Estimation	61
3.4	PSF Estimation	63
3.4.1	Computing a Super-Resolved PSF	64
3.4.2	Computing a Spatially Varying PSF	65
3.5	Chromatic Aberration	65
3.6	Results	66
3.7	Discussion and Future Work	70
3.8	Acknowledgements	75
4	Image Enhancement using Color Statistics	76
4.1	Related Work	78
4.2	Overview	80
4.3	Gradient Priors	81
4.3.1	Gaussian and Sparse Gradient Priors	81
4.3.2	Limitations of Gradient Priors	82
4.4	Color Priors	83
4.4.1	The Two-Color Model	84
4.4.2	Using the Two-Color Model for Deconvolution	85
4.5	Solving for the Final Image	87
4.6	Results	89
4.6.1	Deblurring	89
4.6.2	Denoising	92
4.6.3	Up-Sampling	93
4.6.4	Demosaicing	94
4.7	Alpha Distribution Measurements	97
4.8	Discussion and Future Work	98
4.9	Acknowledgements	101
5	Image Correction using Identify-Specific Priors	102
5.1	Related Work	103
5.2	Overview	106
5.2.1	Prior Representation and Decomposition	107
5.2.2	Enhancement Framework	109
5.2.3	Face Alignment and Mask Computation	110
5.3	Global Correction Operations	110
5.3.1	Image Deblurring	111
5.3.2	Exposure and Color Correction	114
5.4	Face-Specific Enhancement	115
5.4.1	Modifying Lighting and Texture	115
5.5	Personal Photo Correction Application	119
5.6	Results	120
5.7	Analysis of the Eigenspace Prior	126

5.8	Discussion and Future Work	128
5.9	Acknowledgements	130
6	Conclusions and Future Work	131
6.0.1	Building more “Intelligence” into the Photographic Process	132
6.0.2	Video Enhancement using Content Specific Priors	132
6.0.3	Enhancement using Images and Video	133
	Bibliography	134

LIST OF FIGURES

Figure 1.1:	One of the oldest surviving photographs. This image is believed to be the second photograph ever taken and the first using a camera.	2
Figure 2.1:	Examples of Image Blur.	7
Figure 2.2:	Defocus blur (left) and motion blur (right).	8
Figure 2.3:	Blind deconvolution is under-constrained.	9
Figure 2.4:	Image Formation Model. The imaging model consists of two geometric transforms as well as blur induced by motion, defocus, sensor anti-aliasing, and finite-area sensor sampling.	11
Figure 2.5:	Sparse Gradient Distributions.	15
Figure 2.6:	The Richardson-Lucy algorithm shows ringing artifacts, while the Gaussian prior is smoothed, but still contains ringing artifacts. The sparse prior show less noisy and ringing and sharper edges.[From Levin et al. 2007]	17
Figure 2.7:	Frequency domain zeros. As the size parameter of a parametric the blur kernel increases (red > green > blue) the spacing and number of zeros changes. By locating these zeros one can recover the blur scale.	19
Figure 2.8:	Selecting the blur kernel size.	25
Figure 2.9:	Natural Image Statistics. (left) A typical scene. (right) The log distribution of gradient magnitudes within the scene are shown in red. The mixture of Gaussians approximation used by Fergus et at. is shown in green. [From Fergus et al. 2006]	29
Figure 2.10:	Variational Bayes approximates the full posterior with a joint distribution $Q(I, K)$ and finds the value of K such that $Q(I, K)$ and $P(I, K B)$ are most similar as measured by the KL-divergence of the two distributions. [From Fergus et al. 2006]	32
Figure 2.11:	Result from Fergus et al's work. (top) The input blurry image and the location on the image that is used to compute the PSE (bottom) The recovered PSF and deblurred image using Lucy-Richardson.[From Fergus et al. 2006]	33
Figure 2.12:	Bascle et al's Tracking and Temporal deconvolution. (left) Motion is tracked across four input frames. (middle) One of the four input images. (right) Their deblurred output. [From Bascle et al. 1996]	35
Figure 2.13:	Hybrid Imaging.	36
Figure 2.14:	Rav-Acha and Peleg's multi-image deblurring. (left) and (middle) have horizontal and vertical motion blur, respectively. (right) Their recovered image. [From Rav-Acha and Peleg 2005]	36
Figure 2.15:	Yuan et al.'s deblurring with a noisy and blurry image pair.	37
Figure 2.16:	Comparisons of several denoising method. [From Roth and Black 2005 – top row] and [Liu et al. 2008 – bottom row]	43
Figure 2.17:	Comparisons of several up-sampling methods. [From Fattal 2007].	46
Figure 2.18:	Comparisons of example-based super resolution [Freeman et al. 2002] and Fattal's method [2007]. [From Fattal 2007].	47
Figure 2.19:	Comparisons of methods that estimate illuminant color using low-level statistics. 50	50

Figure 2.20:	Estimating illuminant color using the intersection of a dichromatic line with the Planckian locus, i.e., the curve in chromaticity space as specified by Planck's law for black-body radiation. [Finlayson and Schaefer 2000]	51
Figure 2.21:	Color matching using linear transformations in RGB and $l\alpha\beta$ color spaces.	53
Figure 3.1:	Sharp Edge Prediction.	56
Figure 3.2:	Image Formation Model.	58
Figure 3.3:	Non-Blind Estimation. (left) The tiled calibration pattern, (middle) cropped section of an image of a printed version of the grid, and (right) the corresponding cropped part of the known grid warped and shaded to match the image of the grid.	62
Figure 3.4:	Recovering Blur Kernels of Different Sizes and Orientations.	66
Figure 3.5:	Defocus and Slight Motion-Blur.	67
Figure 3.6:	Kernel Size and Orientation. Image deconvolved with (left) our kernel, (middle) our kernel scaled 20% larger, and (right) our kernel rotated by 45° . The middle and right images have more ringing (most apparent at the bottom of the word "Leicester").	68
Figure 3.7:	Defocus and Slight Motion-Blur.	69
Figure 3.8:	Motion Blur.	70
Figure 3.9:	4x Super-Resolution.	71
Figure 3.10:	Different Apertures and Focal Lengths.	72
Figure 3.11:	Sub-Pixel PSFs.	73
Figure 3.12:	Blind Chromatic Aberration.	73
Figure 3.13:	Chromatic Aberration.	73
Figure 3.14:	Iterative Blind Deconvolution with Sharp Edge Prediction.	74
Figure 4.1:	Deblurring with a two color prior.	77
Figure 4.2:	(a) There are many sharp edges that can blur to match the observed blurred (and potentially noisy) edge (shown in tan). The sparse prior always prefers the smallest intensity gradient that is consistent with the observation (shown in red).	83
Figure 4.3:	Over-smoothing and noise texturing with the sparse gradient prior	85
Figure 4.4:	Using the two-color prior	86
Figure 4.5:	Primary, secondary, and alpha maps for the peppers image in Figure 4.4.	87
Figure 4.6:	Deblurring Text: Blurred, noisy image (the PSF is 31×31 pixels and $\sigma = 0.01$), deconvolution with Lucy-Richardson, the sparse prior, our result using the two-color prior, and the groundtruth for two images.	89
Figure 4.7:	Peppers: Blurred, noisy image (the PSF is 31×31 pixels and $\sigma = 0.01$), deconvolution with Lucy-Richardson, the sparse prior, our result using the two-color prior, and the groundtruth for two images	89
Figure 4.8:	Three Colors Meeting at a Point: Even when the two color model does not strictly hold within a neighborhood, the perceptual artifacts in this failure case are minimal.	90
Figure 4.9:	A dragon and sweater from Yuan <i>et al.</i> and a fountain from Fergus <i>et al.</i>	91
Figure 4.10:	Images from the Berkeley Image Database	92
Figure 4.11:	Denoising: Visual comparison of our denoising results with previous work. The fur in the bear is sharper in our result.	92

Figure 4.12:	Deconvolution on an up-sampled grid . We show our method run on a $1 \times$ and $2 \times$ grid.	94
Figure 4.13:	Up-sampling low-resolution images. Our formulation also allows us to perform more traditional up-sampling of low-resolution images.	95
Figure 4.14:	Kodak true color images used for demosaicing experiments.	96
Figure 4.15:	Measurements of alpha distributions.	99
Figure 4.16:	Comparing alpha penalty functions.	100
Figure 5.1:	Automatically correcting personal photos. We automatically enhance images using prior examples of “good” photos of a person.	104
Figure 5.2:	Personal image enhancement pipeline.	105
Figure 5.3:	Mask computation and layer decomposition. We perform our corrections on an “intrinsic image” style decomposition of an image into color, lighting, and texture layers.	107
Figure 5.4:	Eigenfaces Constraint. We use linear feature spaces built from an aligned set of good images of a person as a constraint in our image enhancement algorithms. Here we show the eigenfaces used as a prior for the delurring result shown in Figure 5.1.	109
Figure 5.5:	Exposure and color correction. Using the same set of prior images our system automatically corrects exposure and white-balance for three different images containing the same person.	114
Figure 5.6:	Defocus blur.	117
Figure 5.7:	Super-resolution.	117
Figure 5.8:	Removing high-frequency shadows and uneven illumination.	118
Figure 5.9:	Personal photo correction application. Here we show a screenshot of an initial prototype of our personal photo correction application.	120
Figure 5.10:	Additional deblurring example. Our method automatically performs blind-deconvolution to recover the blur kernel.	121
Figure 5.11:	Comparison to Fergus et al.’s PSF estimation method.	121
Figure 5.12:	Face hallucination comparisons. We compare our result to performing hallucination using an implementation of Liu <i>et al.</i> ’s method and to using our enhancement algorithm with a set of generic faces instead of faces of the same person.	122
Figure 5.13:	Synthetic deblurring experiments.	123
Figure 5.14:	Synthetic upsampling experiments.	124
Figure 5.15:	Face hallucination algorithms without using an intrinsic image decomposition and gradient domain editing.	125
Figure 5.16:	Comparisons to color constancy. We compare our results to the color constancy algorithms discussed by van de Weijer <i>et al.</i> . Our results are more consistent across images, appear better white-balanced, and did not require any parameter tuning.	126
Figure 5.17:	Edge strengths of images in the Eigenspace.	127

LIST OF TABLES

Table 4.1:	Denoising PSNR Comparisons: Our PSNR value are consistently higher than those of Portilla <i>et al.</i> 's method, Liu <i>et al.</i> 's 0th order denoising, and the sparse prior in the higher noise case.	93
Table 4.2:	Demosaicing. With the two color prior. For experiments run with the images in Figure 4.14 our method shows slight improvements for the green channel and no consistent improvements for the red and blue channels.	97

ACKNOWLEDGEMENTS

A twenty-six year journey through three schools and three universities has led me to this point. Through times of excitement, wonder, joy, stress, strain, and sacrifice, I have been fortunate to have family, friends, and mentors to guide me. Their help and insight has been immeasurable, and I would not be here without them.

First, I'd like to thank my committee: David Kriegman, Henrik Wann Jensen, Matthias Zwicker, Sam Buss, and Falko Kuester. I'd like to thank Henrik for his enthusiasm over the years and for grabbing my interest in CSE 272, which helped me get involved in research during my first year. I'd like to thank Matthias for the energy he brought to the graphics group and for his advice during my last two years at UCSD. I'd especially like to thank my advisor, David Kriegman, for his invaluable advice and for providing me with the freedom to find my own way and the guidance to stay on track. His insight has helped me tie together what otherwise could have been a frantic and scattered research path during the past four years.

Over the years, I have been fortunate to have many wonderful teachers. I'd particularly like to thank Tony Hughes and Joan Rainier from Miquon and Bob Gordon, Alice Davis, and Doug Uhlmann at Penn Charter, who always provided extra opportunities for me to learn by way of independent studies or just letting me hang around during free periods and after school.

At Brown University, I was introduced to Computer Science and the wonders of Computer Graphics in CS 15 and 123 taught by Andy van Dam. I will never forget the first program I wrote that made a little cloud drop rain every time I clicked on it – I switched from EE to CS soon after. I'd also like thank my fellow Brown Alums: Dan Morris, Merrie Morris, and Stephen Chen for much needed random acts of fun and advice concerning my life and career during the past decade.

I would not be in computer vision and graphics if it were not for my collaborators at Stanford University. I'd like to thank Bennett Wilburn, Mark Horowitz, and Marc Levoy for taking me on as an inexperienced Master's student on the camera array project. I'd like to thank the members of the Stanford Graphics lab, particularly the (original and honorary) GSlackers (Billy Chen, Jim Chow, Kayvon Fatahalion, Gaurav Garg, Daniel Horn, Jeff Klingner, Chris McGraw, Kimberly McGraw, Dan Morris, Merrie Morris, Ren Ng, Doantam Phan, Augusto Roman, Vaibhav

Vaish, Bennett Wilburn, and Ron Yeh) who provided a fun and supportive environment during my Master's degree.

I would also like to thank the members of the UCSD Pixel Lab (Sameer Agrawal Neil Alldrin, Kristen Branson, Manmohan Chandraker, Will Chang, Krystle Elaine U. de Mesa, Piotr Dollar, Craig Donner, Toshiya Hachisuka, Wojciech Jarosz, Arash Keshmirian, Ben Laxton, Wan-Yen Lo, Satya Mallick, Iman Mostafavi, Vincent Rabaud, and Josh Wills) who helped ease my transition to UCSD and have provided an enjoyable and stimulating environment for the past four years.

Over a year of my four years at UCSD have been spent in internships. They were a defining part of my Ph.D. – I would like to thank MERL, Microsoft Research, and Adobe Systems and my co-authors and mentors in these labs and other associated labs: Shai Avidan, Wojciech Matusik, Hanspeter Pfister, Richard Szeliski, Larry Zitnick, and Edward H. Adelson. It has been a great privilege to work with each of them. I learned an immense amount from every one of them during my internships and in collaborations afterwards, and this dissertation would be quite different without them.

Lastly, I must thank my family. Education has always been paramount in my family, and this has had a strong affect on my motivation and helped me to complete my degree. At last I can finally say I am no longer the least educated member of my family! My sister Hima, being five and a half years older than me, has been a role model for me as she studied hard in school and went on to get her Ph.D. in Chemistry at UCSD in 2001. My parents, Kunda and Suresh (or Mommy and Daddy as I still call them today) have always encouraged my curiosity, creativity, and need for experimentation. They have always supported me, whether they were supplying me with a huge box of Legos (my Kindergarten graduation present), taking me to weekly art or guitar classes, or kindly accepting my destruction of some household item when my curiosity went too far (such as when I clogged the vacuum cleaner by vacuuming up toy cars). They have taught me to work hard and showed me that true accomplishment is not about what you do and who sees it but how you conduct yourself. My parents are the most sincere, hard-working, and selfless people I know. I owe them more than I can convey here.

Portions of this dissertation are based on papers which I have co-authored with others.

My contributions to each of these papers is listed below.

- Chapter 3 is based on material published in the article:

Neel Joshi, Richard Szeliski, and David J. Kriegman, PSF Estimation using Sharp Edge Prediction, Proceedings of IEEE CVPR 2008.

I was the primary investigator and author of this paper.

- Chapter 4 is based on material that is in preparation for submission:

Neel Joshi, C. Lawrence Zitnick, Richard Szeliski, and David J. Kriegman, Image Deblurring and Denoising using Color Priors.

I was the primary investigator and author of this paper.

- Chapter 5 is based on material that is conditionally accepted for publication:

Neel Joshi, Wojciech Matusik, Edward H. Adelson, and David J. Kriegman, Personal Image Enhancement using Prior Images, Conditionally accepted to ACM Transactions on Graphics.

I was the primary investigator and author of this paper.

VITA

2000	Bachelor of Science, Brown University
2004	Master of Science with Distinction in Research, Stanford University
2008	Doctor of Philosophy, University of California, San Diego

PUBLICATIONS

Neel Joshi, Richard Szeliski, and David J. Kriegman. PSF Estimation using Sharp Edge Prediction. Proceedings of IEEE CVPR 2008.

Neel Joshi, Shai Avidan, Wojciech Matusik, and David J. Kriegman. Tracking through Occlusions using Multiple Cameras. Proceedings of IEEE ICCV 2007.

Neel Joshi and David J. Kriegman. Shape from Varying View and Illumination. Proceedings of IEEE ICCV 2007.

Neel Joshi, Wojciech Matusik, Shai Avidan, Hanspeter Pfister, and William T. Freeman. Exploring Defocus Matting: Non-Parametric Acceleration and Super-Resolution. IEEE Computer Graphics and Applications, Special Issue on Computational Photography. March/April 2007 (Vol. 27, No. 2) pp 43-52.

Neel Joshi, Wojciech Matusik, and Shai Avidan. Natural Video Matting using Camera Arrays. In ACM Transactions on Graphics (SIGGRAPH'2006).

Neel Joshi, Craig Donner, and Henrik Wann Jensen. Non-invasive scattering anisotropy measurement in turbid materials using non-normal incident illumination. Optics Letters Vol 31, No 7, pp 936-938 (April 1, 2006).

Bennett Wilburn, Neel Joshi, Vaibhav Vaish, Eino-Ville Tavala, Emilio Antunez, Adam Barth, Andrew Adams, Mark Horowitz, and Marc Levoy. High Performance Imaging Using Large Camera Arrays. In ACM Transactions on Graphics (SIGGRAPH'2005).

Bennett Wilburn, Neel Joshi, Vaibhav Vaish, Marc Levoy, and Mark Horowitz. High Speed Video Using a Dense Camera Array. Proceedings of IEEE CVPR 2004.

Vaibhav Vaish, Bennett Wilburn, Neel Joshi, and Marc Levoy. Using Plane + Parallax for Calibrating Dense Camera Arrays. Proceedings of IEEE CVPR 2004.

Dan Morris, and Neel Joshi Alternative "Vision": A Haptic and Auditory Assistive Device. Student Poster. Proceedings of ACM SIGCHI 2003 Extended Abstracts.

ABSTRACT OF THE DISSERTATION

**ENHANCING PHOTOGRAPHS USING
CONTENT-SPECIFIC IMAGE PRIORS**

by

Neel Suresh Joshi

Doctor of Philosophy in Computer Science

University of California San Diego, 2008

Professor David J. Kriegman, Chair

The digital imaging revolution has made the camera ubiquitous; however, image quality has not improved at the same rate as the increase in camera availability. Increasingly more cameras are small, with inexpensive lenses, no flash, and lightweight bodies that are difficult to hold steady, and this results in images with blur, noise, and poor color-balance. Consequently, there is a strong need for simple, automatic, and accurate methods for image correction. This dissertation, presents work that uses “content-specific” image models and priors for image enhancement.

Image enhancement is a challenge problem – corrections such as deblurring, denoising, and color-correction are ill-posed, where the number of unknown values outweighs the number of observations. As a result, it is necessary to add additional information as constraints. Previous work has focused on using generic image priors that are applicable to a large number of images. In this work, we develop constraints that are tuned to the specific content of an image.

First, we discuss a fast, accurate blur estimation method that models all edges in a sharp image as step-edges. The method predicts the “sharp” version of a blurry input image and uses the two images together to solve for a PSF.

Second, we discuss a framework for image deblurring and denoising that uses local color statistics to produce sharp, low-noise results. Even when the blur function is known, deblurring an image is still quite difficult due to information loss during blurring and due to the presence of

noise. In our work, we investigate using local-color statistics of an image in a joint framework for deblurring and denoising of images.

Lastly, we discuss work in methods that use “identity-specific” priors to perform corrections for images containing faces. These priors provide the guidance needed to perform high-quality corrections needed for known, familiar faces. Deblurring, super-resolution, color-balancing, and exposure correction operate independently, so that a user can correct selected image properties, while still retaining certain desired qualities of the original photo. We have also developed a prototype application for performing these corrections.

Introduction

"You don't take a photograph, you make it."

—Ansel Adams

The visual form is an integral part of the human experience. Over the ages people have had a continuing fascination with the visual world and have strived to capture precious, memorable sights in the world around them. The earliest recorded forms of visual communication pre-date the written word by thousands of years, yet, in the intervening centuries the desire to capture the visual world has not waned. In those years, visual imagery has become an essential part of communication, education, entertainment, art, and the sciences. Since the first cave paintings, perhaps the most significant change in the process of creating visual imagery was the invention of the photograph in the 1800s. Two-hundred years later, photography has permeated almost every aspect of our lives with reckless abandon.

Over the years, there has been a consistent increase in the quality of photographs as scientists and engineers have pioneered new camera technologies, processing methods, and film types to push the limits of photography ever-further. Perhaps the most significant recent step in photography has been the advent of the digital camera. The creation of low-cost, high-quality digital image sensors combined with the power and availability of computers has introduced photography to new applications and new populations in a staggering way. However, while recent advances have vastly increased the popularity and ease with which photographs can be taken, many fundamental limits of photography remain the same. As a result, capturing a high quality



Figure 1.1: One of the oldest surviving photographs. This image is believed to be the second photograph ever taken and the first using a camera. Ever since the first days of photography, the quality of photographs has been limited by technical aspects of image capture. Though manual means in the darkroom and improvements in technology as photography progressed, there has been a constant desire to improve the quality of images. [View from the Window at le Gras, Joseph Nicéphore Niépce 1826]

photograph can still be quite challenging even for experienced users. For inexperienced users, who carry cameras tucked away in pockets and handbags, perfect photographic moments are often lost due to an inadvertent camera movement, an incorrect camera setting, or poor lighting. Such imperfections in the photographic process often cause a photograph to be a complete loss.

Image enhancement and alteration has a long storied history that dates back to the dawn of photography. Colorization is perhaps the first form of photo alteration starting in the 1840s, with darkroom techniques such as dodging and burning for modifying image contrast dating back to the late 19th century. Photo-retouching and manipulation began in earnest in the 20th century as “airbrushing” emerged as a way to remove undesirable elements in a photograph.

In the past few decades, the parallel development of ever-faster computer technology has collided with digital photography, enabling novel ways to tackle the traditional problems of image processing. Image processing methods have offered ways to improve and at times remove artifacts that reduce the quality of photographs, with the now ubiquitous process of “Photoshopping” becoming available to amateur photographers with the first graphical user-interface computers in the mid 1980s.

Automatic digital image enhancement dates back to the late 60s with much of this work

being in image restoration such as denoising and deconvolution. A recent approach is to use image-based priors to guide the correction of common flaws. Image-based priors have been exploited for super-resolution [Baker and Kanade, 2000; Freeman et al., 2002; Liu et al., 2007], deblurring [Miskin and Mackay, 2000; Fergus et al., 2006], denoising [Roth and Black, 2005], and in-painting [Levin et al., 2003]. These priors are specific to a particular domain, such as a face prior for super-resolution of faces or a gradient distribution prior that models natural images, but they tend to be general within the domain, i.e., they capture properties of everyone’s face or all natural images. Such methods are promising and have shown some impressive results; however, at times their generality limits their quality.

In this dissertation, we address image enhancement and correction by developing constraints that are tuned to the specific content of an image. We specifically address the following corrections:

- **Deblurring:** removing blur for images when the blur function is both known and unknown
- **Denoising:** removing artifacts due to image noise
- **Super-resolution and up-sampling:** creating high-resolution images from low-resolution images
- **Lighting transfer and enhancement:** transferring lighting color balance and correcting detail due to under exposure or saturation

1.1 Summary of Original Contributions

The work in this dissertation builds upon several methods in image processing, computer vision, and computer graphics. We outline our major contributions below.

PSF Estimation using Sharp Edge Prediction: We present an algorithm that estimates non-parametric, spatially-varying blur functions (i.e., point-spread functions or PSFs) at sub-pixel resolution from a single image. Our method handles blur due to defocus, slight camera motion, and inherent aspects of the imaging system. Our algorithm can be used to measure blur due to

limited sensor resolution by estimating a sub-pixel, super-resolved PSF even for in-focus images. It operates by predicting a “sharp” version of a blurry input image and uses the two images to solve for a PSF. We handle the cases where the scene content is unknown and also where a known printed calibration target is placed in the scene. Our method is completely automatic, fast, and produces accurate results.

Image Enhancement using Color Statistics: We present a novel image deconvolution algorithm that deblurs and denoises an image given a known shift-invariant blur kernel. Even when the blur function is known, deblurring an image is still quite difficult due to information loss during blurring and due to the presence of noise, which can be significant. At times, blur can be eliminated during capture, yet noise remains. Our algorithm uses local color statistics derived from the image as a constraint in a unified framework that can be used for deblurring, denoising, up-sampling, and demosaicing. A pixel’s color is required to be a linear combination of the two most prevalent colors within a neighborhood of the pixel. This two-color prior has two major benefits: it is tuned to the content of the particular image and it serves to decouple edge sharpness from edge strength.

Image Correction using Identify-Specific Priors: We present a framework for improving the quality of personal photographs by using a person’s favorite photographs as examples. We observe that the majority of a person’s photographs include the faces of a photographer’s family and friends and often the errors in these photographs are the most disconcerting. We focus on correcting these types of images and use common faces across images to automatically perform both global and face-specific corrections. Our system achieves this by using face detection to align faces between “good” and “bad” photos such that properties of the good examples can be used to correct a bad photo. “Personal” photographs provide strong guidance for a number of operations and, as a result, enable a number of high-quality image processing operations. We illustrate the power and generality of our approach by presenting a novel deblurring algorithm, and we show corrections that perform sharpening, super-resolution, in-painting of over and under-exposure regions, and white-balancing.

1.2 Organization of the Dissertation

This dissertation is divided into six chapters. In Chapter 2, we present an overview of the related work relevant to the areas of this dissertation: image deblurring, denoising, up-sampling, and color balancing. While Chapter 2 gives a high-level overview of related work, we directly compare our methods to relevant previous work in each subsequent chapter. Chapter 3 discusses our work in PSF estimation using sharp edge prediction. Chapter 4 presents our work on image enhancement using color statistics, and Chapter 5 discusses our work in image correction using identify-specific priors. In Chapter 6, we close this dissertation with a summary of our contributions and several directions for future work.

Previous Work

“The great thing in the world is not so much where we stand, as in what direction we are moving”

—Oliver Wendell Holmes

Acquiring satisfactory and usable images with a camera involves ensuring that the film or digital sensor in a camera is properly exposed given a scene’s illumination strength and color. Several camera parameters affect this process. To properly handle variations of lighting strength a camera’s exposure can be adjusted by varying three parameters: exposure time, aperture size, and film or sensor sensitivity (ISO). To properly color balance a photo, the lighting color needs to be offset by adjusting the camera white-balance. This white-balance setting affects the relative ratio of red, green, and blue so that photos can be captured that better reflect the human perception of an object’s color under varied lighting color.

Each exposure adjustment has its own tradeoff that can lead to various image artifacts in low-light conditions. Long exposure times allow more light to be captured, but increase the chances of getting blurred images if an object or the camera moves during the exposure window. Large apertures allow more light to reach the film or sensor, but can cause depth-dependent blur as the depth of field is decreased. High ISOs increase image contrast; however, they increase image noise – for film cameras, larger ISOs correspond to larger film grain, thus noise appears as the large discrete film grains become visible, while for digital cameras, ISO represents a gain that amplifies sensor noise in addition to the image signal. As a result, acquiring high-quality images in low-light conditions can be quite challenging.



Figure 2.1: Examples of Image Blur. Object motion blur: stars moving relative to the camera cause blur trails. Here the trails capture the PSE [From Flickr.com] (top left) and fast-motion of the gymnast’s leg causes the leg to blur with the background [From Jia 2007] (top right), Blur from inadvertent camera movement [From Fergus et al. 2007] (bottom left), and defocus blur due to shallow depth-of-field and the camera auto-focus focusing on the background [From Flickr.com] (bottom right).

Incorrect white-balance adjustments can also lead to image artifacts, where the perception of the colors in a photograph does not match what a human observer would see. While current cameras implement basic white-balancing algorithms and have presets for common lighting conditions, the basic approaches are not always successful.

A significant area of research in image-processing, computer vision, and, more recently, computer graphics has addressed the area of image quality enhancement and much of this work handles lighting related situations.

In this chapter, we will give an overview of the previous work in image enhancement. We will focus on image deblurring, denoising, up-sampling, and white-balancing, which are the four main image corrections discussed in this dissertation. The previous work discussed in this chapter is discussed at a high level to give an overview of the relevant areas. Individual chapters that appear later in this dissertation will discuss previous work in direct comparison to the work presented in the respective chapters.

2.1 Deblurring

There are numerous causes of image blur. Some examples of image blur, specifically motion and defocus blur are illustrated in Figure 2.1. Classical, non-blind deconvolution refers to deblurring a signal that was blurred with a known blur function. In contrast, blind deconvolution refers to the process of estimating both a blur function and true image when only the blurred image is known. This problem, which is inherently under-constrained, is quite challenging and

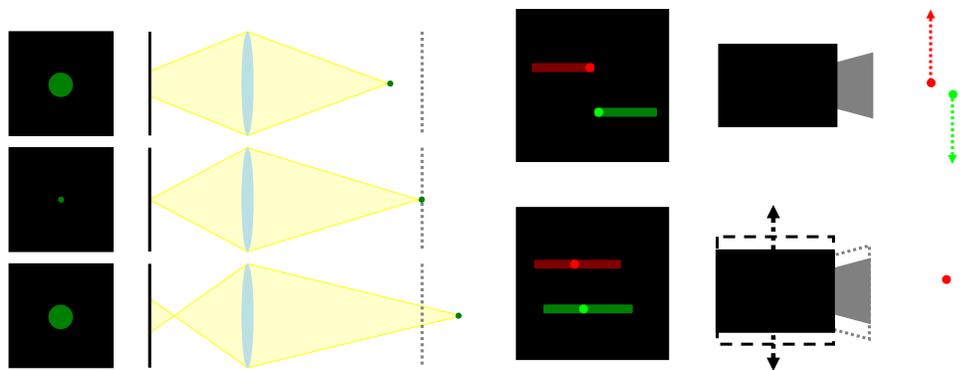


Figure 2.2: Defocus blur (left) and motion blur (right).

has spawned a large area of research. Blind deconvolution has been applied in many fields outside imaging such as audio processing and seismography; however, in this chapter, we will focus on imaging applications.

Blur estimation when observing only the blurred result is an ill-posed problem and cannot be solved without extra constraints or information. If one thinks of blind deconvolution in signal processing terms as blind source-separation, this lack of constraints becomes clear. Blind source-separation refers to the process of recovering two independent signals from one observed signal that is mixing of the two. In the case of image blur, the independent signals are the potentially spatially-varying point-spread function (PSF) and the sharp image. Estimating two unknowns from one known is clearly not well-posed. In fact, given an input blurred image and no constraints on the image or kernel, there are infinitely many pairs of kernels and images that can be combined to result in the blurred image. An illustration of this is shown in Figure 2.3.

Thus to estimate the PSF and sharp image from a blurred image it is necessary to add constraints or information to the system, and this is how the problem has been approached by the research community. Where methods have differed over the years is the nature of the constraints that are added.

Before discussing blur estimation and blind-deconvolution, we will first give some background on image blur. The rest of this chapter proceeds as follows: in Section 2.1.1 we will give an high-level overview of types image blur types and the intuition behind the factors that affect blur estimation algorithms, in Section 2.1.2 we will formally define the commonly used image blurring

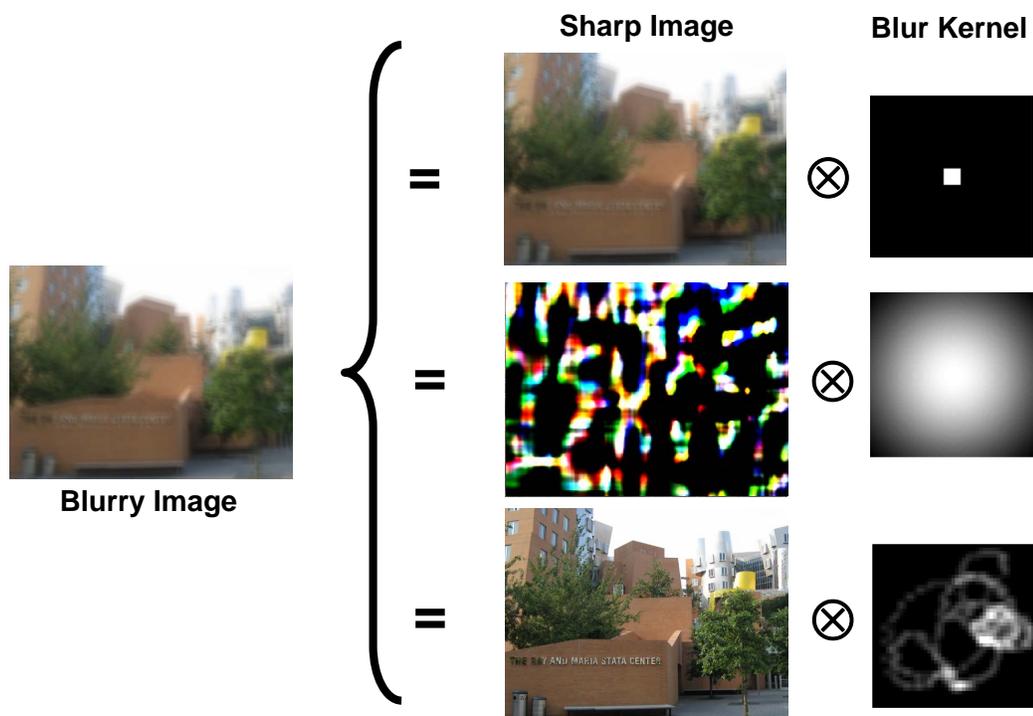


Figure 2.3: Blind deconvolution is under-constrained. An observed blurred image (left) can be described by infinitely many combinations of a “sharp” image convolved with a blur kernel, even in the case of a simple non-spatially varying blur. Three examples are shown (right). [From Fergus et al. 2006]

model, followed by a discussion of the most commonly used non-blind image deblurring methods (deblurring when the blur function is given) in Section 2.1.3. We break down blur estimation and blind-deconvolution methods into three general categories: methods that used parametric PSF models are discussed in Section 2.1.4, methods that use image constraints are discussed in Section 2.1.5, and lastly, methods that use multiple images are discussed in Section 2.1.6. These categories are not mutually exclusive, for example there are methods that use image constraints and parametric models. For these mixed methods, we group work by the area where its most novel contribution lies.

2.1.1 Image Blur Overview

Image blur can be described by a point spread function (PSF). A PSF models how an imaging system captures a single point in the world – it literally shows how a point “spreads” across an image. An entire image is then made up of a sum of the individual images of every scene

point, where each point's image is affected by the PSF associated with that point. Thus, in imaging applications where one ideally does not want any image blur, the PSF should be minimal, i.e., it should be a delta function, where each scene point should correspond only to one image point. In practice, PSFs can take on a range of shapes and sizes depending on the properties of an imaging system. When PSFs are large relative to image resolution, blurry images are captured. If a PSF is known one can rely on non-blind deconvolution to remove image blur. Unfortunately, PSFs are not typically known, and it is necessary to estimate them.

It is necessary to understand what governs the form of a PSF to properly estimate one. The shape and size of the PSF is determined by properties of the imaging system as they relate to the scene. As discussed above, to get sharp images PSFs should be delta functions; however, due to numerous physical limits, this is impossible. As a result common cameras do not point-sample a scene, but instead capture images by integrating light over fixed areas and ranges of time. Properties such as aperture size, exposure duration, lens focal length, pixel size, and changes in the relative position between camera and object, all influence how the sampling is performed and in turn affect the PSF. Cameras sample light entering the lens over an area with non-zero area – this area affects the PSF. Similarly, it is not possible to use single point apertures or to capture a single instant in time; instead cameras must use an aperture of some non-zero area and an exposure that lasts some duration of time. Thus movement during the exposure window or changes in aperture will affect the PSF.

These effects on the PSF can be categorized into three areas: blur due to lack of resolution, motion blur, and defocus blur. In general, image blur can be an arbitrary combination of these effects. While these three types of blur can be treated in a somewhat related fashion, resolution blur is typically addressed as separate research problem than motion and defocus blur. The primary reason for the separation is that for resolution blur the PSF is known and is also unchanging across the image, while for motion and defocus blur, the PSF is often unknown and could be spatially varying. In this section, we will focus on PSF estimation and thus address motion and defocus blur.

Motion blur is due to motion of scene objects and/or the camera while the camera shutter is open, which causes points to be imaged over a large area of camera sensor or film. The PSF

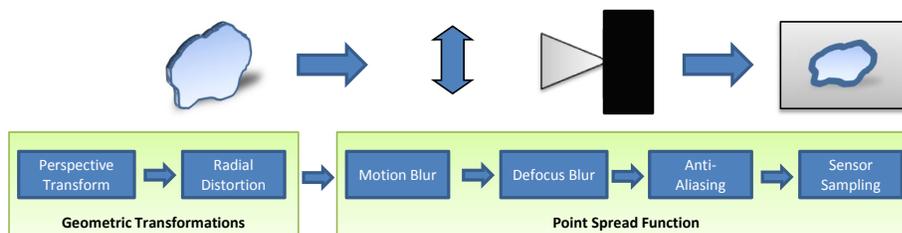


Figure 2.4: Image Formation Model. The imaging model consists of two geometric transforms as well as blur induced by motion, defocus, sensor anti-aliasing, and finite-area sensor sampling.

describing motion blur is actually a projection of the motion path of the moving objects onto the image plane, as illustrated in 2.2. The motion path of a point can be due to translation and rotation of objects in three dimensions, and there can be different paths for different parts of the scene, and when using long-exposures, these paths can be quite large.

Defocus blur occurs when using an aperture with a non-zero area. For very small apertures or a nearly “pinhole” aperture the PSF and consequently the blur can be insignificant relative to the resolution or pixel size of the camera. However, in light-limited situations, larger apertures are used. With a large aperture the “depth-of-field” of the lens is no longer infinite and the PSF for objects off the focal plane is larger. The amount of blur is depth dependent; it depends on the focal length of the lens and the focal depth, and it grows with distance from the focal plane, as illustrated in Figure 2.2.

In the next section, we will formally define blind deconvolution and PSF estimation, we will then discuss the current work on estimating PSFs for motion and defocus blur, and we will discuss them in the context of the assumptions they place on the form of both the PSF and the input blurred image.

2.1.2 Image Blur Model

We now give a brief overview of relevant imaging and optics concepts needed for blur estimation. As illustrated in Figure 2.4, the imaging model from world to image consists of geometric transforms: a perspective transform and a radial distortion, and there are several sources of blur induced by motion, defocus, sensor anti-aliasing, and pixel sampling area (fill factor and active sensing area shape). In blind deconvolution and most PSF estimation algorithms, image blur is

modeled as a convolution along the image plane, and thus the geometric transformations are ignored. In the simplest case, the PSF is considered to be shift-invariant, i.e., the same for all parts of the scene. Blur estimation methods account for depth-dependent defocus blur and 3D motion blur by allowing for the PSF to be spatially varying.

PSF estimation methods generally seek to find a discrete representation of an underlying continuous PSF that is modeled as a blur kernel. For most practical imaging situations a linear model of blur is an accurate representation. A blurred image B is a convolution of a kernel K and a potentially higher-resolution sharp image I , plus noise, which is usually assumed to be additive Gaussian white noise:

$$b = i \otimes k + n, \quad (2.1)$$

where $n \sim \mathcal{N}(0, \sigma^2)$. In this formulation, the kernel k models all blurring effects, which are potentially spatially varying and wavelength dependent. The goal of image deconvolution is to recover i given k , and “blind” deconvolution is recovering both i and k simultaneously from b . There are two approaches to blind deconvolution. The first approach is to initially perform PSF estimation to recover k given b . Then once k is known, i can be recovered using image deconvolution. The second approach is to merge PSF estimation with deconvolution where k and i are estimated simultaneously. We will now give a brief overview of non-blind image deconvolution before discussing the area of PSF estimation and blind deconvolution.

2.1.3 Image Deconvolution

While this section focuses on blur estimation and blind deconvolution, non-blind deconvolution is an important component of many estimation and blind deconvolution methods and is relevant to the contributions of this dissertation. As stated in the previous section blur estimation and blind deconvolution often make use of non-blind deconvolution methods. Blur estimation methods (methods that only seek to recover the PSF) at times use a non-blind deconvolution algorithm as a way to measure the accuracy of a particular PSF. While blind-deconvolution methods often use non-blind deconvolution algorithms as part of a closed-loop optimization.

Formally, non-blind image deconvolution is the process of recovering a sharp image from

an input image corrupted by blurring and noise, where the blurring is due to convolution with a *known* kernel and the noise level is known. It is important to note that non-blind deconvolution, just as blind-deconvolution, is an inherently ill-posed problem due to the loss of information during blurring and the addition of noise. Thus even if the blur function is known, there are multiple sharp images that can result from deconvolution. The observed blurred image only provides a partial constraint on the solution—there exist many “sharp” images that when convolved with the blur kernel will match the observed, noisy blurred image. Thus, the central challenge in deconvolution is to develop methods to disambiguate among the many potential solutions and bias the process towards more likely results given some prior information, and this is where various algorithms differ.

The current most popular approaches are the Lucy-Richardson’s algorithm [Richardson, 1972; Lucy, 1974], Wiener deconvolution, and least-squares deconvolution, and deconvolution that uses image priors derived from natural image statistics [Levin et al., 2007]. Other methods have explored the use of graph cuts to reduce over-smoothing [Raj and Zabih, 2005], deconvolution using multiple blurs [Harikumar and Bresler, 1999], and energy minimization functions using wavelets for deconvolution [de Rivaz and Kingsbury, 2001; Neelamani et al., 2004].

The most straightforward deconvolution method is inverse filtering, which can be performed in the frequency domain:

$$I = B/K. \quad (2.2)$$

In the case of zero noise, infinite precision, and no zero frequencies in the kernel, this method will be successful. In the more realistic case of noise corruption of b , this method quickly becomes unusable as there are additional frequencies added to b .

The Lucy-Richardson algorithm deconvolves an image given a known PSF by using an iterative method that is derived assuming a Poisson noise model in a Bayesian framework:

$$i_{j+1} = i_j \left[k * \frac{b}{i_j \otimes k} \right], \quad (2.3)$$

where ‘*’ is the correlation operator.

The Weiner filter is similar to inverse filtering that uses a prior measure of image signal

to noise to minimized the impact of deconvoluted noise for frequencies that have low signal to noise:

$$I(f) = B(f)G(f) \quad (2.4)$$

$$G(f) = \frac{K(f)^* S(f)}{|K(f)|^2 S + N(f)}, \quad (2.5)$$

where K^* is the complex conjugate of K , and N and S are the mean spectral density of I and the noise.

If the Wiener filter is re-written one can see the relationship to inverse filtering:

$$G(f) = \frac{1}{K(f)} \left[\frac{|K(f)|^2}{|K(f)|^2 + N(f)/S(f)} \right] \quad (2.6)$$

$$= \frac{1}{K(f)} \left[\frac{|K(f)|^2}{|K(f)|^2 + 1/SNR(f)} \right], \quad (2.7)$$

where $SNR(f)$ is the frequency-dependent signal-to-noise ratio. As the SNR become infinitely large, i.e. no noise, the Wiener filter reduces to the standard inverse filter. As the SNR decreases the Wiener filter attenuates the corresponding frequencies.

Lucy-Richardson and Weiner filtering were developed for applications where the images are quite different than those taken by a typical photographer, e.g., Lucy-Richardson assumes Poisson noise statistics, which are more applicable to very low-light conditions. Consequently, these methods are not always well suited to the desired task of deblurring common photographic images and often generate unwanted artifacts such as ringing. A disadvantage of the Wiener filter is that the signal to noise ratio must be known to get optimal results, which is difficult to know as it requires some measure of the unobserved latent image i .

An alterative approach to these frequency domain methods are image-space methods that use image priors. The simplest of these is a least squares deconvolution, such as least-squares deconvolution with a smoothness prior:

$$i = \underset{i}{\operatorname{argmin}} \frac{\|i - b \otimes k\|^2}{\sigma^2} + \lambda \|\nabla i\|^2. \quad (2.8)$$

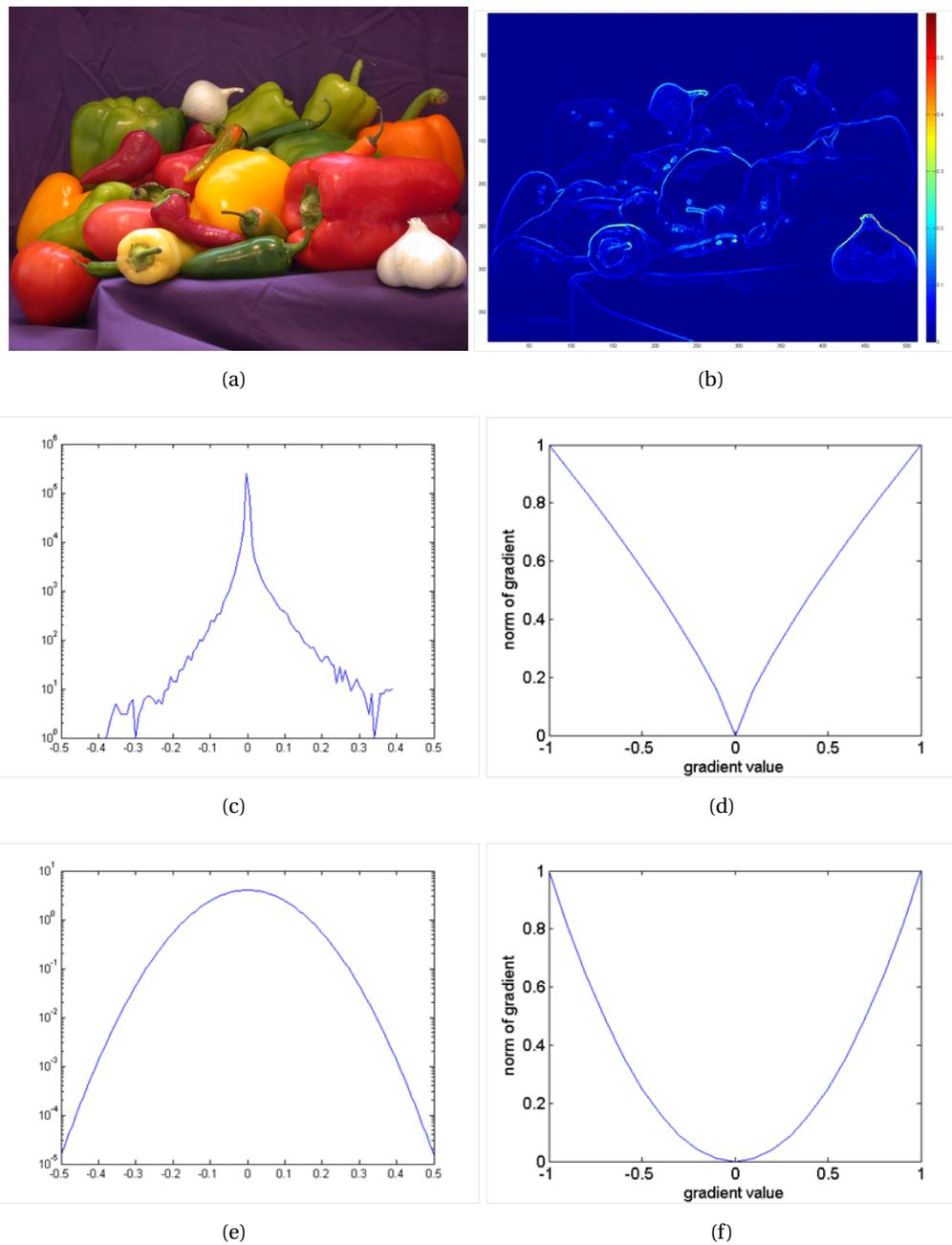


Figure 2.5: Sparse Gradient Distributions. (a) A sharp image, (b) a visualization of the magnitudes of the gradients of that image, i.e., edge strength, (c) The log histogram of these gradients, (d) the $L_{0.8}$ norm used by Levin et al., (e) A gaussian distribution and corresponding L_2 norm. The $L_{0.8}$ norm minimizes the gradients according to a hyper-laplacian distribution, which is a better model for natural image edge statistics.

Here the first term is an “observation” term that states that the recovered sharp image i must blur to match the observation. The second term states that the gradients of i must be minimal in a L_2 sense.

The work of Levin *et al.* [2007] is a significant improvement upon this method. Instead of using a simple L_2 norm on the gradients, which enforces a Gaussian distribution. They use a smoothness prior based on assumptions about the edge content of “natural” images. Specifically, the authors assume that images are piecewise smooth and thus the gradient distribution of an image is zero-peaked with high kurtosis. They enforced this property using a hyper-Laplacian prior on image gradients during deconvolution. As a result of using this prior, they can generate sharp images that are more consistent with typical photographic images. The error function for their deconvolution method is:

$$i = \underset{i}{\operatorname{argmin}} \frac{\|i - b \otimes k\|^2}{\sigma^2} + \lambda \|\nabla i\|^{0.8}. \quad (2.9)$$

As the gradient penalty is no longer quadratic, they minimize this function using iterative re-weighted least-squares.

See Figure 2.6 for a comparison of Lucy-Richardson, least-squares deconvolution with a Gaussian prior, and the sparse prior. The Richardson-Lucy algorithm shows ringing artifacts, while the Gaussian prior is smoothed, but still contains ringing artifacts and noise. The sparse prior show less noise and ringing and sharper edges. The sparse prior has very clear benefits for natural images. The one downside of this approach is that it is more time consuming. While other methods can be formed in the frequency domain, Levin *et al.*'s method must be performed in the image domain using the iterative re-weighted least-squares method. The authors report running times of up to an hour for several mega-pixel images.

2.1.4 PSF Estimation by Constraining the PSF

Recall that recovering a PSF from a single blurred image is an inherently ill-posed problem. Prior knowledge about the image or kernel can disambiguate the potential solutions. Much of the early work in blur estimation constrained the form of the kernel [Kundur and Hatzinakos, 1996].

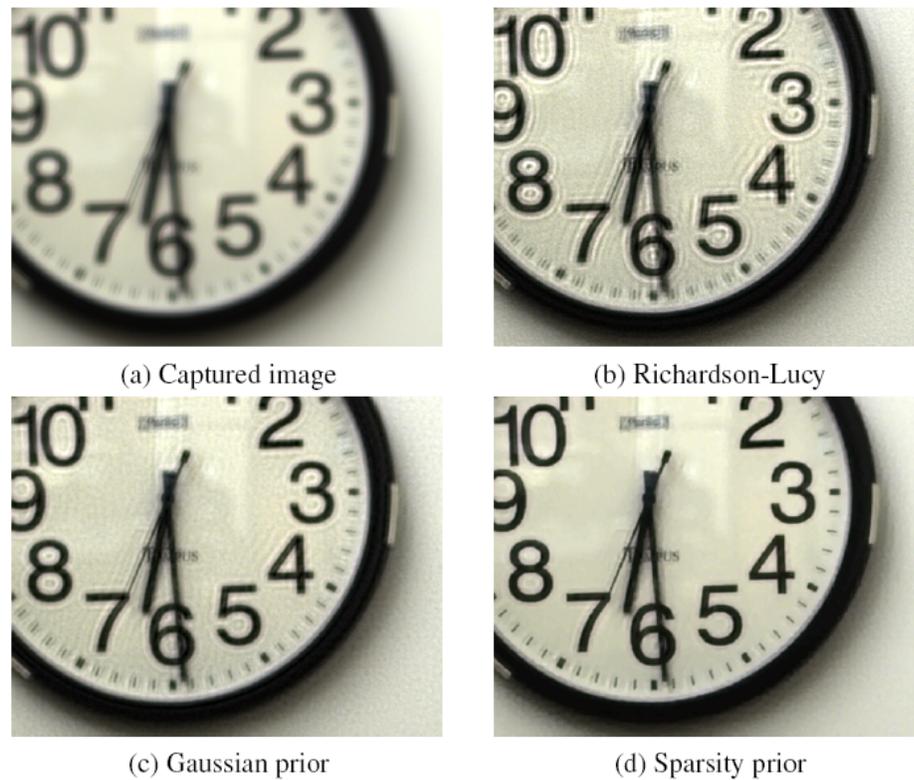


Figure 2.6: The Richardson-Lucy algorithm shows ringing artifacts, while the Gaussian prior is smoothed, but still contains ringing artifacts. The sparse prior show less noisy and ringing and sharper edges.[From Levin et al. 2007]

There are several common assumptions made on the PSF that serve to constrain its form:

- Values of the PSF are non-negative
- The PSF is energy conserving, i.e., $\sum_i k_i = 1$
- The PSF is symmetric – radially or along some axis
- The PSF is has a known parametric form

These assumptions are listed in order from least to most restrictive. Positivity, is a strong constraint and the least restrictive in that it does not eliminate any truly valid kernels, i.e. no true blur kernel can have negative values as blurring is a purely additive process. Another way of thinking of this is that there is no “negative” light. Similarly the second constraint is equally not restrictive in that blurring does not remove light, thus all true blur kernels should be energy conserving. Thus

the assumptions of positivity and energy conservation are ones that can be used by virtually all PSF estimation methods. In practice, whether a particular method uses them depends on the nature of the other assumptions and models they use.

The second two assumptions are much more restrictive. Symmetry is typical used when one wants to generalize a 2D PSF from some 1D cross section [Yoon et al., 2001]. Assuming a parametric form is also very restrictive as it assumes the entire shape of the blur kernel can be modeled by a low parameter mathematical model. We will now discuss blur estimation that uses parametric models. Methods that use the other priors will be discussed in Section 2.1.5.

Parametric Blur

PSF models are commonly used for defocus and motion blurs.

Defocus Blur: Defocus blur has two commonly used models. A circular disk or “pillbox” function [Cannon, 1976]:

$$k(x, y) = \begin{cases} 0 & \sqrt{x^2 + y^2} > r \\ \frac{1}{\pi r^2} & \sqrt{x^2 + y^2} \leq r \end{cases} \quad (2.10)$$

and a circularly symmetric 2D Gaussian [Banham and Katsaggelos, 1997]:

$$K(x, y) = N \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right), \quad (2.11)$$

where N is a normalization constant. In both cases a single parameter determines the PSF – r for the pillbox and σ for the 2D Gaussian.

Motion Blur: Motion blur is usually parameterized as a 1D box-blur, where a horizontal blur is:

$$k(x, y) = \begin{cases} 0 & y \neq 0 \quad -\infty \leq x \leq \infty \\ \frac{1}{2d} & y = 0 \quad -d \leq x \leq d \end{cases}, \quad (2.12)$$

and blurs at other orientations can be represented by rotating this kernel.

Frequency Domain Zeros:

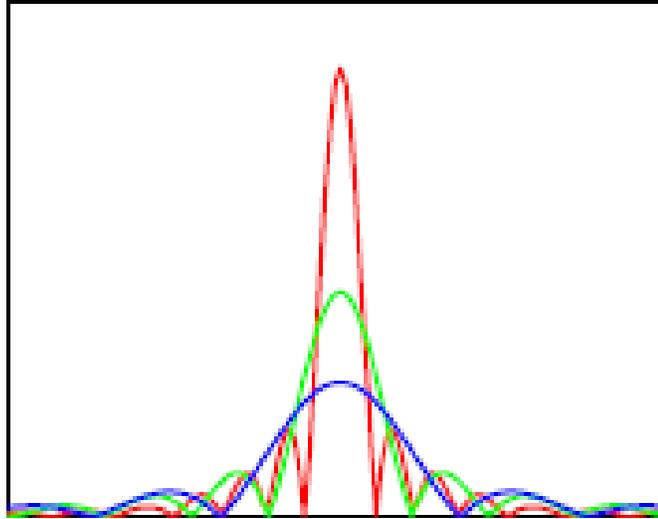


Figure 2.7: Frequency domain zeros. As the size parameter of a parametric the blur kernel increases (red > green > blue) the spacing and number of zeros changes. By locating these zeros one can recover the blur scale.

One of the simplest approaches for finding the respective parameter for defocus and 1D motion blur is to find frequency domain zeros of the imaging system [Cannon, 1976; Gennery, 1973; Chang et al., 1991]. The high-level idea is that for any blur kernel there are certain spatially frequencies of the underlying sharp image that will not be retained after blurring. Thus by identifying these zero frequencies and matching them to the known zero frequencies for a particular parametric blur model, one can recover the kernel.

Consider the following blurring model, which is the same as Equation 2.1, except that the noise component has been removed:

$$b = i \otimes k, \quad (2.13)$$

if we take the Fourier transform of each side of the equation the following frequency domain relationship results:

$$B(u, v) = I(u, v) * K(u, v), \quad u, v \in \mathcal{R}, \quad (2.14)$$

where B , I and K are the discrete Fourier transform of the respective lowercase counterparts. It is clear from this frequency domain representation that the zeros of B are the zeros of the combined I and K . Thus by computing the Fourier transform of the blurred image b and locating the zeros,

one can recover the blur kernel by identifying which parameter value creates a PSF whose zeros values match those of the blurred image. Figure 2.7 shows an illustration of how frequency domain zeros locations and number scale with blur kernel size.

Blur identification using frequency domain zeros is a nice and mathematically elegant approach, but it suffers from several drawbacks. The theory relies on the observed image being noise-free, i.e. no additional image frequencies are introduced after the blurring process. Thus the frequency-zeros of I and K remain those of B . However, in the presence of noise, such as the additive noise in the image model described by Equation 2.1, the observed image B will have no frequency-domain zeros, as the noise introduces image frequencies independent of the blurring process.

Typically this weakness is addressed by assuming that the additional frequencies introduced by the noise will be small and thus values close to zero are identified as opposed to exact zero-values. However, even with this modification, as the signal-to-noise of the observed image decreases, the frequency profile flattens and finding “zeros” become increasingly difficult [Chang et al., 1991]. Following up on Gennery’s [1973] work, Cannon [1976] proposed a similar approach operating on the power spectra and cepstrum, the Fourier transform of the log of Fourier coefficients of the signal, rather than on the Fourier coefficients themselves. This approach showed increased robustness to noise.

Chang *et al.* [1991] built on this work by operating on the frequency bispectrum, which is a third order frequency measure, instead of the frequency spectrum. The bispectrum has the property that the contribution from a gaussian white noise signal is zero as the bispectrum of a gaussian signal is zero and the bispectrum of two added signals is the sum of the bispectrum of the two signals. Thus the frequency zeros of the bispectrum of I and K remain the zeros of the bispectrum of B .

While frequency space zeros methods have achieved some success, there are many cases in which they are not appropriate. Even when using power spectra, cepstrum, or bispectrum there is still some noise sensitivity. As a result, these methods have generally fallen out of favor. Furthermore these methods are limited to finding parametric PSFs that contain frequency-space zeros. There are many types of blur that do not have zeros in the Fourier domains, e.g., atmospheric blur

or Gaussian blurs (in the continuous case), and some that do not have easily indefinable zeros.

Blur Identification using Parameter Search:

An alternative approach to finding the unknown parameter values for parametric blur models is to do a search over the parameter space with a match metric that can be computed for each parameter value. The proper parameter value is chosen as that which gives the best match according to the metric. If the blur in the scene is only due to depth of field and a search is performed for the scale of a defocus kernel, this method is equivalent to shape-from-defocus as the scale of the parametric defocus kernel is related directly to the depth of an object.

The work in this area takes the general approach of estimating a sharp image by performing deconvolution in the frequency or image domain and using the residual as match metric. In the simple case, given an observed blurred image b , a candidate PSF k_j , and the corresponding recovered sharp image, i_j obtained using a deconvolution method, the residual is:

$$r_j = b - (i_j \otimes k_j), \quad (2.15)$$

where j is an index into the parameter space for the PSF model. Without loss of generality, we notate the parameter space as one-dimensional, but the space could be of higher dimensionality. Methods in this area use different approaches for estimating i_j and for computing the residual metric (Equation 2.15).

Savakis and Trussell [1993], presented one of the first approaches of this type. They use a frequency domain residual:

$$R_j(u, v) = B(u, v) - I_j(u, v) * K_j(u, v), \quad u, v \in \mathcal{R}, \quad (2.16)$$

where B , I_j , and K_j are the discrete Fourier transform of the respective lowercase counterparts. The sharp image is estimated using a Weiner filter deconvolution:

$$I_j(u, v) = \frac{B(u, v)H_j(u, v)P_f(u, v)}{P_f(u, v)|H_j(u, v)|^2 + P_n(u, v)}, \quad (2.17)$$

where $P_f(u, v)$ and $P_n(u, v)$ are the power spectra of the original image and noise respectively. The authors show how to compute a cost function to determine if a particular frequency domain residual indicates a good match. They compute their cost as a function of the L_2 frequency domain residual and the expected frequency domain residual for the “correct” deconvolution given a prior model for the spectral density of typical images.

Rooms *et al.* [2004] take a similar approach; however, they replace the spectral residual metric with an image-space metric. They seek to maximize a focus measure that will choose the PSF such that the deconvolved result with a Wiener filter using that PSF has the sharpest edges. The authors use a Wavelet basis as an edge filter, and their focus measure is the kurtosis of the histogram of two bands of wavelet coefficients. The authors compare their work to that of Savakis and Trussell [1993]. They show few quantitative comparisons, but state that their method tends to over-estimate the PSF size while Savakis and Trussell’s method under-estimates the PSF size.

Levin *et al.* [2007] also perform a search over PSF size using an image-space metric; however, in contrast with previous work, they do not use Wiener deconvolution and instead they use their own deconvolution method (described in Section 2.1.3) with an image space residual as their match metric. Specifically, given an estimated deconvolved image i_j for a candidate PSF k_j , the PSF is chosen as:

$$k = \underset{j}{\operatorname{argmin}} \|b - i_j \otimes k_j\|^2. \quad (2.18)$$

The advantage of Levin *et al.*’s method for deconvolution is that it produces images that are more consistent with natural images that we typically see. Visually this results in sharper images with fewer ringing artifacts, as shown in Figure 2.6. It is unclear that their method provides a significant benefit relative to Lucy-Richardson or Weiner Deconvolution when it comes to providing a discriminating match metric, as for all of these methods the residual in Equation 2.18 will be relatively small for all methods when the PSF is near the correct value.

The main advantages of a parametric search over the frequency zero method is that it is less sensitive to noise and that it applies to a broader range of parametric PSFs. The main disadvantage is computational cost. Performing a search over a parameter space can be quite time consuming depending on the dimension of the parameter space, the sampling density of

the space for the search, and the time complexity of computing the match metric. In Levin *et al.*'s approach each individual deconvolution can take up to an hour for large images, thus even a modest parameter search will take many hours. In contrast, the frequency domain zero method should take on the order of seconds.

Multiple Blurs

In the previous section, each algorithm found a single blur kernel for an image by evaluating a matching function for the entire image. However, in many cases a single blur will not model the image, as there are different amounts of blur for different parts of the scene, i.e. the blur is spatially varying. Spatially varying blur occurs most often due to depth of field effects. Depth of field is the property that for a lens system, only one depth in the scene is perfectly “in-focus” and for all other depths there is some depth-dependant defocus. Thus for many real images it is often desirable to solve for the blur in a spatially varying way.

Every technique described in the previous section can be used to identify spatially varying blurs. The most straightforward approach, used by Cannon [1976], is to perform the search process on sub-windows of the image instead of the entire image. The assumption here is that the blur is slowly varying and consistent within each small sub-section of the image. This is a generally reasonable approach that has been used by a number of researchers; however, it does have some limitations. While it is a relatively safe assumption that the blurs vary slowly across the image, it is often not a good assumption that a PSF can be accurately computed for each sub-window, as there may not be enough relevant image information in the sub-window. For example, in a constant, textureless region of a scene, any amount of blur will produce the same observed image data. Thus all the methods in the previous section require some significant amount of texture for their match metrics to be discriminative. While sufficient texture is generally available when using the entire image, as people do not tend to photograph completely textureless scenes, most scenes have locally textureless areas.

Levin *et al.* [2007] address this issue by adding regularization to the process of picking the scale for each sub-window. They do this by adding a smoothness term on the scales of PSFs for neighboring windows to enforce that neighboring regions of the image has similar

scales of blur. For images where the blur is piecewise smooth this is reasonable. They regularize the scale selection by using a graphcut framework. The graphcut is well-suited to this type of regularization – when the match metric is ambiguous the PSF scale for a pixel will be consistent with its neighbors' scales. Despite these additions the problem of selecting the correct scale is still non-trivial. As illustrated in Figure 2.8, to get an acceptable assignment, Levin *et al.* still need some user intervention to refine the solution.

Levin [2007] also developed a method to identify multiple 1D motion blurs within in an image. Similar to selecting the scale of a defocus blur, Levin computes a match metric as the log likelihood of the image data in sub-windows of an image matching a particular 1D motion blur. This is done by first identifying the blur direction for each sub-window of the image as the direction with the minimum response to an oriented edge-filter. The idea is that if the magnitude of derivatives in the latent sharp image is equal in all orientations, in the blurred image the orientation with the weakest derivatives is the blur direction. Given this direction, the algorithm blurs the image data in the window along the direction orthogonal to the blur and computes a histograms of the magnitude of the derivative in the same orthogonal direction after blurring. The blur scale is the one that results in this histogram best matching the histogram of gradients along the identified blurring direction. The scale selection is also regularized spatially using a graphcut.

Jia [2007] addresses finding multiple motion blurs in a scene but uses a very different method for identifying the blur. She observes that given the alpha matte of a blurred object (a soft-segmentation that captures the transparency the object) the size and varying opacity of the matte is determined by the motion blur. She uses standard techniques to compute the alpha matte and then show how to find 1D motion blurs in closed form. This can be done for a number of segmented objects in a scene. She also extends the method to 2D motion blur – this is discussed in Section 2.1.5

Shan *et al.* [2007], also use an alpha matte, but fit a low-parameter rotational model to the matte to model motion blur due to an object translating and rigidly rotating about an axis parallel to the camera's optical axis. The model is fit on a per-pixel basis with spatial regularization over the model parameters.

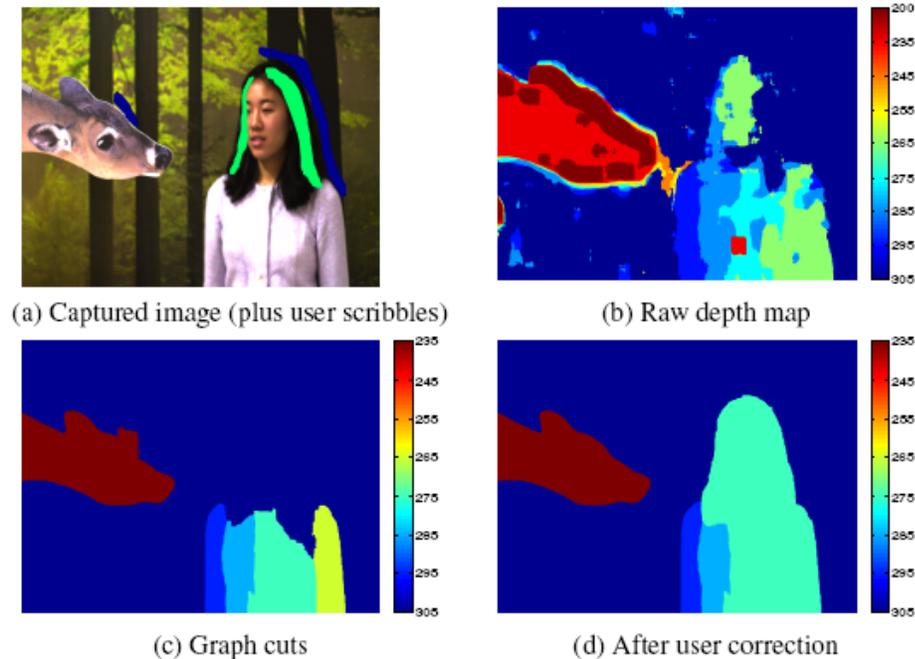


Figure 2.8: Selecting the blur kernel size. (b) The raw depth map by finding the best match per-pixel independently – the result is quite noisy and in-accurate. (c) Using a graphcut to compute a spatially regularized solution – there are still error where the head has been miss-labeled. Using (a) user suggestions is the only way for the work to get the correct scale/depths in (d) [From Levin et al. 2007]

2.1.5 PSF Estimation by Constraining the Image

While constrained or parametric kernels can model a wide range of common blurs, such as 1D motion and defocus, there are many types of blurs that are difficult to capture using parametric models.

When a parametric model is not sufficient it is necessary to solve for a full blur kernel. Methods that solve for a full kernel generally still assume that the blur can be modeled as convolution with a single shift-invariant blur kernel.

A full non-parametric blur kernel of $N \times N$ values has N^2 unknowns. As discussed in Section 2.1, to solve for such a kernel it is necessary to constrain the image or otherwise incorporate prior information about the image to solve the blur estimation and blind-deconvolution problem.

In this section, we will discuss blur estimation and blind-deconvolution methods that use image constraints or priors. Most of the work in this area use some form of prior on the PSF as a weak constraint on the kernel.

When using image priors or constraints, there are several common priors used to constrain the PSF:

- Values of the PSF are non-negative
- The PSF is energy conserving, i.e., $\sum_i k_i = 1$
- The PSF is smooth, i.e., the magnitude gradients of the kernel are minimal
- The PSF is mostly zero or has low entropy

As in the previous section, assumptions of positivity and energy conservation are constraints that can be used by virtually all PSF estimation methods. One of the final two constraints are used by most methods.

Bayesian Blind Deconvolution

The work that we discuss in this section employs image and PSF priors in a Bayesian framework [Richardson, 1972; Gull, 1988; Wan and Nowak, 1999; Miskin and Mackay, 2000; Fergus et al., 2006]. Thus before discussing these methods we will first introduce the basic Bayesian framework used by these methods.

The goal of Bayesian blind-deconvolution is to find the most likely estimate of the sharp image i and blur kernel k , given the observed blurred image b and a known noise level σ . This is typically solved by maximizing the probability distribution of the posterior using Bayes' rule – also known as *maximum a posteriori* (MAP) estimation.

The posterior distribution is thus expressed as the joint probability of i and k given b :

$$P(i, k|b) = \frac{P(b|i, k)P(i, k)}{P(b)}. \quad (2.19)$$

Since i and k can be assumed to be statistically independent:

$$P(i, k|b) = \frac{P(b|i, k)P(i)P(k)}{P(b)}, \quad (2.20)$$

i and k can then be recovered by minimizing of a sum of negative log likelihoods:

$$i, k = \underset{i, k}{\operatorname{argmax}} P(i, k|b) \quad (2.21)$$

$$= \underset{i, k}{\operatorname{argmin}} L(i, k|b) \quad (2.22)$$

$$= \underset{i, k}{\operatorname{argmin}} L(b|i, k) + L(i) + L(k). \quad (2.23)$$

Where $L(b|i, k)$ is the “data” or “observation” log-likelihood:

$$L(b|i, k) = \|b - i \otimes k\|^2 / \sigma^2. \quad (2.24)$$

that ensures that the error between the recovered values and observation matches best under a Gaussian distribution and $L(i)$ and $L(k)$ reflect image and PSF priors.

Equation 2.21 specifies an error function in two unknowns i and k . Minimizing this error function is challenging as it is non-linear, which can be seen as the partial derivatives of Equation 2.21 in terms of i and k are interdependent.

Equation 2.21 can not be solved in closed form to recover i and k and thus is solved iteratively using one of two methods. The first option is to minimize the equation directly using any number of non-linear optimization methods. A number of gradient-descent style methods have been used for this approach such as standard steepest descent and more sophisticated Newton’s style methods. As the error function is subject to numerous local minima, the final solution is highly dependent on the initial guess. An alternative approach is to use Simulated Annealing, which is an optimization approach that attempts to avoid local minima by probabilistically sampling the input space for different starting points for a gradient descent style optimization [McCallum, 1990].

Gradient descent algorithms and Simulated Annealing are quite time consuming; fortunately, there are alternative approaches to minimizing the error function for blind-deconvolution that have been used with equal success and increased speed. The most common alternative is to use an alternating minimization scheme such the Expectation Maximization algorithm or similar approaches [Richardson, 1972; Gull, 1988; Ayers and Dainty, 1988; Wan and Nowak, 1999; Miskin

and Mackay, 2000; Fergus et al., 2006].

An alternating minimization approach can be used for this non-linear problem by breaking the problem into two more easily solvable sub-problems by considering one set of unknowns as observed and solving for the other and vice versa in an alternating fashion. In the context of blind-deconvolution, this consists of making some initial guess for i and k and then alternately solving for i given k and then k given i . Both sub-problems can still be formulated using a Bayesian MAP framework and can be individually solved more easily and in closed-form, depending on the forms of the image and PSF priors.

The image deconvolution sub-problem using a Bayesian framework is found by minimizing the negative log likelihood:

$$i = \underset{i}{\operatorname{argmin}} L(i|b, k) = \underset{i}{\operatorname{argmin}} [L(b|i, k) + L(i)], \quad (2.25)$$

while the kernel estimation sub-problem is found as:

$$k = \underset{k}{\operatorname{argmin}} L(k|b, i) = \underset{k}{\operatorname{argmin}} [L(b|i, k) + L(k)]. \quad (2.26)$$

Early Work using Image Priors:

We will now give an overview of the blind-deconvolution algorithms that employ this type of alternating minimization. The simplest approach is to assume that $L(i)$, and potentially $L(k)$, is constant. The constant prior assumes that all images and kernels are likely. This approach boils down to a simple inverse filtering or least-squares approach, which is equivalent to a maximum-likelihood solution given a Gaussian distribution for $P(b)$. As expected, this method is not very successful as it does not constrain the image and kernel at all.

The earliest successful work in this area is the seminal work of Richardson and Lucy [1972]. Richardson and Lucy assume that the conditional distribution $P(b|i, k)$ is Poisson, which they formulate as maximizing the posterior distribution $P(i|b, k)$ given a constant model for $P(i)$ and a Poisson noise model. A Poisson distribution is appropriate for low-light situations which are the imaging conditions for the remote-sensing and astronomical applications that the algorithm was

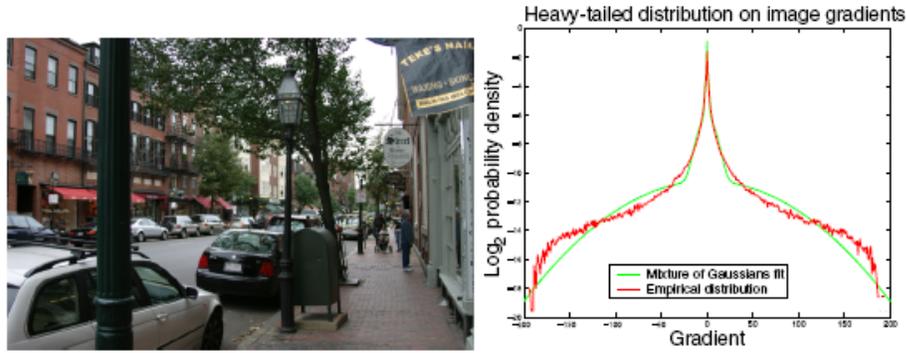


Figure 2.9: Natural Image Statistics. (left) A typical scene. (right) The log distribution of gradient magnitudes within the scene are shown in red. The mixture of Gaussians approximation used by Fergus et al. is shown in green. [From Fergus et al. 2006]

developed for. Gull [1988] additionally examines using Cauchy and Gaussian distributions in a similar framework. To perform blind deconvolution with Richardson and Lucy’s algorithm, one performs an alternating minimization, guessing an initial value for k_0 and i_0 commonly $i_0 = b$ and alternately minimizing:

$$i_{j+1} = i_j \left[k_j * \frac{b}{i_j \otimes k_j} \right], \quad (2.27)$$

$$k_{j+1} = k_j \left[i_{j+1} * \frac{b}{i_{j+1} \otimes k_j} \right], \quad (2.28)$$

where ‘*’ is the correlation operator [Fish et al., 1995].

There are numerous other similar techniques for blind-deconvolution; however, all of this earlier work was focused on astronomical imaging. In most cases blind-deconvolution is easier in this application domain as the images of stars in the scene are known to be “point-like”, and the blurry image of a single star reflects the point-spread function.

Natural Image Priors:

As consumer photography has increased in popularity, the need for new blind-deconvolution algorithms became imperative as the earlier methods were not very successful when applied to typical everyday photographs.

More recent work in blind-deconvolution seeks to derive priors based on image statistics

measured from typical, everyday, or “natural” scenes. Natural image statistics refers to the observation that many images have gradient distributions that are zero-peaked, with high-kurtosis and are heavy tailed, as shown in Figure 2.9. Simply put, this implies that images are piecewise smooth, i.e., most areas of the image are constant or slowly varying, separated by few large edges.

The work by Caron *et al.* [2002] is the earliest to use this concept for blind-image deconvolution. The authors derived a prior in the frequency domain that is a power-law that models that the strength of the frequency content of an image is peaked largely around low-frequencies, with a smaller proportion of high-frequency texture. They incorporate this prior as a frequency dependent power law in a frequency-domain Wiener filtering framework.

In contrast, Wan and Nowak [1999] use natural image statistics in the imaging domain instead of the frequency domain. Specifically, they learn a two component mixture of Gaussians to model the zero-peaked, heavy-tailed distributions of edges strengths modeled as Wavelet coefficients. They then incorporate this prior into a MAP estimation using an alternating minimization as discussed in Section 2.1.5. One advantage of Wan and Nowak’s work is that by using an edge prior in the image domain, local-spatially information is retained, as opposed to Caron *et al.* work that uses a frequency domain prior. Both methods will attenuate frequencies due to noise, but Wan and Nowak’s do this by maintaining image smoothness thus enforcing the additional model of the image being piecewise smooth.

The most recent work in this area is that of Fergus *et al.* [2006] who, like Wan and Nowak, use a mixture of Gaussian distribution as a prior on image gradients, as shown in Figure 2.9. However, in contrast with Wan and Nowak, Fergus *et al.* do not perform the alternating minimization using a MAP estimation, but instead build on work by Miskin and Mackay [2000] that used a variational Bayes framework for blind-deconvolution. Miskin and Mackay use priors based on image color statistics for deblurring cartoon images. This method is not very successful for natural images as deriving a broadly applicable color prior is not straightforward. By adapting this work to use the natural image gradient prior, Fergus *et al.* achieved good results on a number of images, as the gradient distribution prior is much more generalizable than a color prior. Fergus *et al.* choose to use the Variational Bayes approach as the more standard MAP estimation did not product high-quality results as it is subject to local-minima.

Instead of finding i and k to maximize the posterior distribution $P(i, k|b)$, a variational Bayes approach tries to approximate the full posterior distribution and then compute the kernel k with the maximum marginal probability given this approximate posterior. The method selects a kernel that is most likely relative to the entire distribution of possible latent images. In addition to the gradient prior, Fergus *et al.* place a mixture of exponential distributions prior on the kernel values, to enforce sparsity, i.e., kernel values are encouraged to be zero, and the prior requires that all values be positive.

The minimization is performed by approximating the full posterior with a joint distribution $Q(i, k)$ and finding the value of k such that $Q(i, k)$ and $P(i, k|b)$ are most similar as measured by the KL-divergence of the two distributions, as illustrated in Figure 2.10. A more detailed explanation of Variational Bayes is beyond the scope of this chapter, and we refer the reader to the paper by Miskin and Mackay [2000] for more details. We note that Fergus *et al.*'s algorithm is not a full blind-deconvolution method as they operate only on gradients of a sub-section of an image to recover a PSF. To recover the full deblurred image they use the non-blind Richardson-Lucy algorithm. A result from their work is shown in Figure 2.11.

Zheng and Hellwich [2006b; 2006c; 2006a] explore several variational image priors including Mumford-Shah regularization [1989], which is based on the assumption that an image can be divided into a piecewise smooth approximation. They also explore total variational regularization. Both of these approaches are somewhat similar to the gradient priors of the previously discussed work. What is particularly interesting about Zheng and Hellwich's work is that they include a model prior on the PSF. They consider the unknown PSF to come from one of several distinct parametric PSF models and for a particular PSF they derive a likelihood of it coming from each model. In their alternating minimization framework, they adhere to the parametric model that gives the maximum likelihood. This method is interesting as it provides more flexibility than models that use a single parametric model, but more regularization than a simple sparsity or smoothness constraint on the PSF.

Jia [2007] presents a method for recovering full 2D blur kernels based on computing an alpha-matte of a scene. Instead of performing an alternating minimization on the original image data, they perform blind-deconvolution on the alpha matte to recover the known blur function.

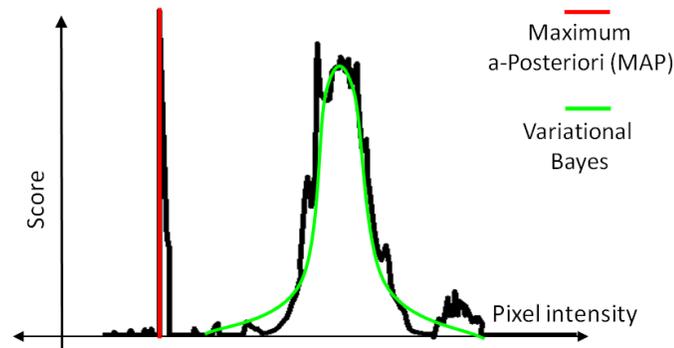


Figure 2.10: Variational Bayes approximates the full posterior with a joint distribution $Q(I, K)$ and finds the value of K such that $Q(I, K)$ and $P(I, K|B)$ are most similar as measured by the KL-divergence of the two distributions. [From Fergus et al. 2006]

They enforce a prior on the alpha-matte that it should be mostly valued 0 and 1, which implies that in the latent sharp image most edges are step edges. The benefit of such an approach is that a binary valued prior (they use a beta distribution) is relatively easy to enforce and fairly consistent across many natural images. The disadvantage of this method is that it completely relies on the blur being modeled by the alpha matte. For large, complex blurs this is not likely as the alpha matte can only represent the blending of two colors, while a complex blur can mix many areas of the image that consists of many colors.

Bayesian methods for non-blind deconvolution and blur estimation are quite promising – the work of Fergus *et al.* is perhaps the most successful method for recovering an arbitrary blur kernel. Yet, there are some limitations of the current work in this area. The first is that in many cases the state of the art methods often do not recover the correct PSF. Secondly, these methods are time consuming even when using the faster alternating, EM-style approaches.

Non-Bayesian Methods

An interesting approach by Yoon *et al.* [2001] approaches blur estimation from a completely different point of view. Instead of taking a probabilistic approach or using a parametric PSF, they analyze the image data to look for direct evidence of the form of the PSF. Their work is similar to slant-edge calibration [Reichenbach et al., 1991; Burns and Williams, 1999]. These methods recover 1D blur profiles by imaging a slanted edge feature and finding the 1D kernel normal to the



Figure 2.11: Result from Fergus et al's work. (top) The input blurry image and the location on the image that is used to compute the PSF. (bottom) The recovered PSF and deblurred image using Lucy-Richardson. [From Fergus et al. 2006]

edge profile that gives rise to the blurred observations of the known step edge. Reichenbach *et al.* [1991] combine several 1D sections to estimate a 2D PSF.

Yoon *et al.* perform a similar operation, though in a blind setting. Instead of imaging a known edge, they run an edge detector on the image and look at the intensity profiles orthogonal to those edge. For each detected edge they compute the edge orientation and bin it into one of 4 orientation bins (0, 45, 90, and 135 degrees). Within each bin they average the step response of the edge along the orthogonal direction of the edge. This results in 4 radial slices of the PSF that

they use to estimate a single circularly-symmetric PSF.

The benefit of such an approach is that it is computationally quite simple and avoids many of the ambiguities of the methods discussed in the previous section. The disadvantage of Yoon *et al.*'s approach is that it only models circularly symmetric PSF; however, this still provides more flexibility than some of the parametric models. Another shortcoming is that by aggregating data in a relatively coarse way, many of the subtle aspects of the PSF may be lost. Bones *et al.* [2005] use a similar approach of identifying a 1D PSF, but use a tomographic reconstruction method to recover the 2D PSF. This method appears to recover more subtle structure of the PSF; however, a better evaluation of this work is needed.

2.1.6 Multi-Image Blur Estimation

As discussed in Section 2.1, blur estimation and blind deconvolution from a single image is inherently ill-posed. While image and PSF priors can be used to regularize the solution and recover images and blurs that are more likely for a particular application domain, the current work in this area is still only partially successful. An alternate approach to blur estimation is to acquire multiple images. While this may not always be possible and does little to help deblur an existing image, it is an interesting idea to acquire data more intelligently during capture knowing that deblurring will happen afterwards.

One straight-forward approach to handle defocus blur is to take images focused at multiple depths and then create a “deblurred” image by essentially compositing together the focused regions. This type of approach is essentially the same as classical shape-from-focus methods [Nayar and Nakagawa, 1990, 1994; Nayar et al., 1995], which addresses depth estimation using a parametric model for blur that is either a “pillbox” or 2D Gaussian function with a single parameter for the PSF size, i.e., focal length or kernel radius. The “pillbox” function is the same as those used by the frequency-spectrum zeros methods [Cannon, 1976; Gennery, 1973; Chang et al., 1991] for defocus blur. In essentially the inverse fashion of the parameter search methods described in Section 2.1.4, depth-from-focus methods work by scoring every pixel in each image with a “focus measure”, which is a function of the magnitude of the derivative in the pixel’s local neighborhood. The depth for a pixel is chosen as the one with the highest focus measure and this can then be



Figure 2.12: Bascle et al's Tracking and Temporal deconvolution. (left) Motion is tracked across four input frames. (middle) One of the four input images. (right) Their deblurred output. [From Bascle et al. 1996]

used to assemble a sharp focused photograph. Similar to the method discussed in Section 2.1.4, spatial regularization can be used.

A big disadvantage of such methods are that the scene has to remain static while all images are acquired, which is quite impractical. An alternative to is capture multiple views of a scene at once, i.e., a lightfield [Levoy and Hanrahan, 1996] and refocus these to different depths in post-processing. We refer the reader to the work of Isaksen *et al.* [2000] for more details on this approach.

Bascle *et al.* [1996] show how to deblur a sequence of motion-blurred and defocus blurred images of a moving object, where the motion is rigid, by using a combination of tracking and temporal deconvolution [Irani and Peleg, 1993] to recover a single sharp image.

RavAcha and Peleg [2000; 2005] take a different approach where they show that they produce better deblurring results given multiple images of a scene where each image has a different blur direction. The authors assume a parametric oriented linear motion blur for each image and use a parameter search to recover the blur function for each image. They then use the images together to recover a higher quality sharp image using a non-blind deconvolution method.

Ben-Ezra and Nayar [2004] take the approach of attaching a video camera to a traditional still camera where the video camera is filming at a high frame rate with lower resolution during the exposure duration of the still camera. By tracking the motion of the video frames, they capture

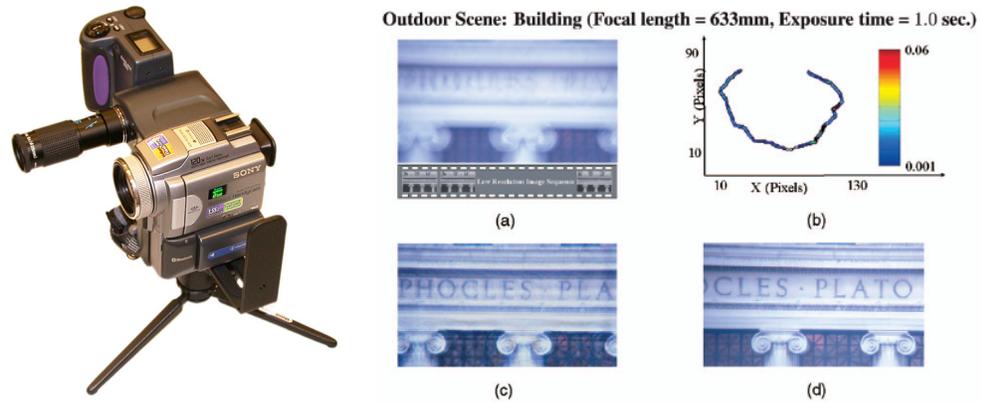


Figure 2.13: Hybrid Imaging. (left) Ben-Ezra and Nayar's hybrid imaging camera. The video camera record a video during the camera exposure time. By tracking motion in the video, the camera motion is recovered. (right) (a) Input images, including the motion blurred image from the primary detector and a sequence of low-resolution frames from the video camera. (b) The computed PSF (c) The deblurring result. (d) Sharp groundtruth image captured using a tripod. [From Ben-Ezra and Nayar 2004]



Figure 2.14: Rav-Acha and Peleg's multi-image deblurring. (left) and (middle) have horizontal and vertical motion blur, respectively. (right) Their recovered image. [From Rav-Acha and Peleg 2005]

the motion path of the camera. They recover the camera-shake PSF by performing optical flow between each video frame to recover the global motion model for each frame. The concatenation of the frame-to-frame movement of the video provides the path of the camera, which is used as the PSF.

One of the most recent methods in this area is that of Yuan *et al.* [2007]. The authors show that a sharp and noisy short exposure of a scene and a long exposure of the same scene, taken in quick succession, can be used together to recover a PSF and a high-quality deconvolved

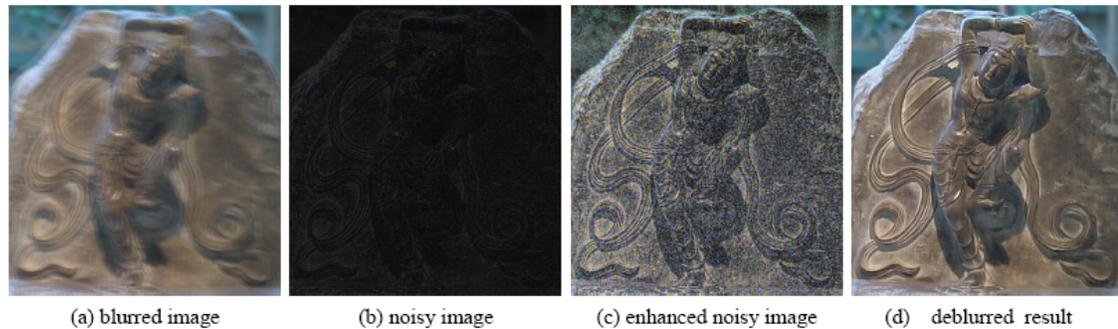


Figure 2.15: Yuan et al.'s deblurring with a noisy and blurry image pair. (a) Blurred image (with shutter speed of 1 second, and ISO 100). (b) Noisy image (with shutter speed of 1/100 second, and ISO 1600). (c) Noisy image enhanced by adjusting level and gamma. (d) Their deblurred result. [From Yuan et al. 2007]

image. This is done by aligning the two images and using a denoised version the noisy image as an initial guess for i in the alternating minimization approach discussed in Section 2.1.5. The advantage of this method is that the denoised image is a very good initial guess for the sharp image i . Furthermore, during the deconvolution step of their algorithm, they use the denoised image as prior which increases the quality of the deconvolution.

2.1.7 Image Deblurring Summary

In the previous sections, we have surveyed a broad range of work in blur estimation and blind deconvolution. These problems are fundamental ones in computer vision and image processing with applications to a number of fields. In addition to being critical problems, they are difficult problems, as they are theoretically unsolvable given the general formulation. As a result not only has there been a lot of motivation to address this problem, there have been a wide variety of methods. We will now review these approaches at a high-level and compare and contrast them.

Much of the earlier work in blur estimation used parametric blur models. This work dates back to the 1970s and has several strong points. Parametric models simplify the deblurring task by reducing the set of unknowns to a small set. Estimating this small set of parameters in a very restricted space is a much better conditioned problem than the general blind-deconvolution problem. Furthermore, a parametric model provides robustness to measurement noise in the observed blurred image and prevents “over-fitting”, which can happen when the degrees of

freedom of the solution space are large enough to allow a model to fit to noise. When a parametric model can appropriately describe the blur in an image, fitting such a model is often the best solution for blur estimation. Thus these methods are limited to cases where a model can be developed. In the case of 2D or 3D motion blur (of the camera or objects in a scene) parametric models tend to be less useful as the space of 2D and 3D motion is quite large and is not easily generalizable.

In contrast with methods that use parametric blur models, methods that solve for a full blur kernel have much broader applicability, but also suffer from over-fitting and can have ambiguous solutions. Work in this area uses image and kernel priors to find better solutions and have achieved some success.

Multi-image blur estimation and deconvolution often produce the highest visual quality and most accurate results. The disadvantages of such methods are that cannot deblur an existing single image and that many of these methods require some modification to the camera or typical image acquisition process.

2.2 Denoising

Image denoising is closely coupled to image deblurring, as for many of the cases that lead to image blur, image noise is an additional factor; however, even in the absence of blur there can be significant artifacts due to image noise. As a result image denoising has received a significant amount of attention separate from image deblurring.

Image noise can arise at several different stages of an imaging system. The observed noise in an image is a combination of shot, quantization, sampling, thermal, and flicker noise. Noise occurs due to various aspects of the electronics in an image sensor and tends to be most disturbing in low-light situations, where the signal value, due to minimal light exposure, is low compared to the noise values.

These different types of noise have different characteristics. Shot noise occurs in low-light situations where the number of photons reaching an image sensor is small enough that there is a detectable statistical fluctuation in accuracy of the measurement. This fluctuation can be

described by a Poisson distribution. Thermal noise is due to the thermal excitation of electrons in the sensor, it has a white noise spectrum (flat spectral density), and is well modeled by a Gaussian distribution. Flicker noise has a pink spectrum (falls off with $1/\text{frequency}$) and is due to quantization noise and noise introduced at the transistor level. For a more detailed discussion of the sources and types of image noise, we refer the reader to the paper by Tsin *et al.* [2001].

Most work in the image denoising literature models noise as a additive white Gaussian noise (AWGN) process:

$$O = I + N, \quad (2.29)$$

where I is the noise-free image, O is the observed image, and the noise $N \sim \mathcal{N}(0, \sigma^2)$. While this model is not completely accurate, it is a reasonable approximation in many common situations. Poisson distributions with large N are well-modeled by a Gaussian distribution, thermal noise is Gaussian, and flicker noise is generally dominated by the first two noise processes. Thus it is a reasonable approximation to model the combination of all noise processes using a Gaussian distribution. Where this model fails is in lower light situations, where the shot noise dominates and the noise-levels scale with intensity.

While the AWGN noise process is by far the most commonly used model, several researchers have developed more sophisticated and accurate noise models [Healey and Kondepudy, 1994; Tsin *et al.*, 2001; Liu *et al.*, 2006].

All denoising methods share the same general property that they desire to keep important edges due to scene content and remove less meaningful edges due to noise. Where methods differ are the models and priors they use on the image, which affects which edges are considered meaningful and which are not. We will now give a high-level overview of these methods. For a more detailed discussion we refer the reader to the paper by Buades *et al.* [2005].

2.2.1 Basic Filtering Methods

The most basic approaches for denoising are neighborhood filtering methods such as Gaussian and median filtering. These methods rely on the assumption that image noise is most prevalent in the highest-frequency components of an image and that the statistics of the

neighborhood around a pixel can be used to smooth the noise. The most basic filter is a box-blur that assumes the value of a pixel is equal to the most likely intensity value in the pixel's neighborhood, given a uniform spatial distribution. Gaussian filtering uses a normal distribution to spatially weight the pixels:

$$I(p) = \frac{1}{W(p)} \sum_{q \in S} G_{\sigma_s}(\|p - q\|) O(q), \quad (2.30)$$

where the value for pixel p is $I(p)$, O is the noisy image, $W(p)$ is a normalization factor, S is the neighborhood of p , and $G_{\sigma_s} \sim \mathcal{N}(0, \sigma_s^2)$. Formally, it is equivalent to solving an isotropic heat diffusion equation [Strang, 1986]. Median filtering follows a similar approach that expects the median of the neighboring pixels to best model a pixel value. For low noise levels and small neighborhoods, the median filter produces visually sharper results than Gaussian filtering, which tends to over-smooth edges and remove image detail.

2.2.2 Edge-Preserving Methods

One way to improve on the results of Gaussian filtering is to consider posing denoising as solving an anisotropic heat (or diffusion) equation. Anisotropic diffusion methods add an edge-dependent scaling term to the smoothing operation that enforces that smoothing occurs along an edge instead of across it. This is done by solving a PDE:

$$I = \text{div}(g(\|\nabla I(x, y)\|) \nabla I), \quad (2.31)$$

where g is a monotonically decreasing function that is small for large values, thus resulting in less smoothing for large gradients, and large for small values, thus smoothing small gradients more strongly [Perona and Malik, 1990].

Another approach for adapting Gaussian filtering to be edge-preserving is Bilateral filtering [Tomasi and Manduchi, 1998]. The Bilateral filter combines a Gaussian weighting spatially

with a Gaussian distribution in the intensity range:

$$I(p) = \frac{1}{W(p)} \sum_{q \in S} G_{\sigma_s}(\|p - q\|) G_{\sigma_r}(|O(p) - O(q)|) O(q), \quad (2.32)$$

where S is the neighborhood of the pixel p given by the size of the Gaussian spatial filter. One difficulty with the Bilateral filter is how to choose σ_r . Denoising with a bilateral filter produces similar results as with anisotropic diffusion. For a direct comparison of the two methods we refer the reader to the paper by Barash [2002]. Both methods have a tendency to over-smooth highly textured regions and over-sharpen strong edges.

A different approach to performing edge-preserving denoising is the Non-Local Means method [Buades et al., 2005]. For static scenes, one approach to denoising is to take N images and average them. Under a Gaussian noise assumption the variance of the observed noise will decrease as \sqrt{N} . The Non-Local means method approximates this temporal mean from a spatially mean by finding patches with similar texture for a patch around each pixel and taking the mean of the center pixels values. This algorithm is essentially a combination of the bilateral filter [Tomasi and Manduchi, 1998] and example driven methods [Efros and Freeman, 2001], as it averages in the range domain similar to the bilateral filter, but considers pixels across the entire image that have have a similar neighborhood instead of only using a local neighborhood.

This method produces high-quality results, but can be quite time consuming, $O(N^4)$ for an image with N^2 pixels, as the algorithm calls for performing a weighted sum with every patch in the image. Buades *et al.* propose restricting the search area for patches to a fixed window of size of M^2 around each pixel, which reduces the time to $O(M^2 N^2)$, where $M = 21$ in their experiments.

2.2.3 Using Image Priors

An alternative approach to neighborhood filtering approaches is to consider denoising at a higher level view of global image estimation under some image prior. A very prevalent approach in recent years is to use the properties of natural image statistics to derive methods that enhance large intensity edges and suppress lower intensity edges. This property has been used by wavelet based methods [Simoncelli and Adelson, 1996; Portilla et al., 2003] and Field of Experts models

[Roth and Black, 2005].

Wavelet based methods use the image prior that natural images have highly-peaked, kurtotic marginal distributions of edge filter responses. Methods such as Wavelet coring [Simoncelli and Adelson, 1996; Portilla et al., 2003] decompose images into a multi-scale wavelet representation (multi-scale sub-bands of oriented filter responses) and enforce the highly kurtotic distribution by suppressing low-amplitude wavelets coefficients while retaining high-amplitude values. The denoised image is then reconstructed by inverting the wavelet decomposition.

Field of Experts models [Roth and Black, 2005] also rely on natural images, but in a more direct way. They view an image as a global Markov-Random-Field (MRF) and develop a parametric prior over large neighborhood cliques where potential functions are learned from a set of representative images of training data. Field of Experts methods do not achieve quite the same performance of Wavelet methods, although the benefit is that the same prior can be used for other applications such as in-painting, super-resolution, sharpening, and image-based rendering[Roth and Black, 2005].

The current best performing denoising algorithm is that by Liu *et al.* [2006], whose work addresses denoising by considering that images should adhere to a “two-color” model that states, which states that within some local neighborhood of a pixel the colors can be represented as a linear combination of two dominant colors, where these colors are piecewise smooth and can be derived from local properties:

$$I = \alpha P + (1 - \alpha)S. \quad (2.33)$$

Liu *et al.* build this model by segmenting an image first and then build an affine color model per segment. They not only use this model for denoising but they additionally address noise-level estimation, where instead of using a standard AWGN model with a single variance value for the entire image, they use a AWGN model where the noise variance can vary spatially as a function of image intensity. They refer to this function as a noise-level-function (NLF).

Our work builds on the work of Liu *et al.* in several ways. Many of the algorithms presented in this dissertation require a proper noise level estimate, and, as indicated in each chapter, we use their method for estimating the spatially varying noise level. Our work further builds on Liu *et*

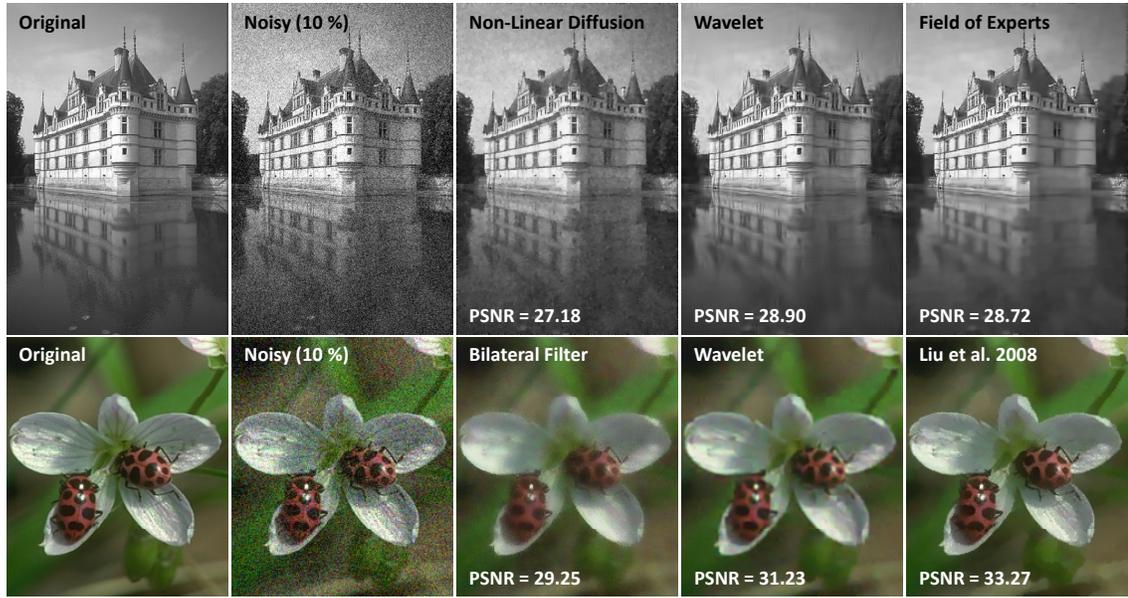


Figure 2.16: Comparisons of several denoising method. [From Roth and Black 2005 – top row] and [Liu et al. 2008 – bottom row]

al.'s in that we use the concept of local two-color model for several applications in addition to denoising. Furthermore, we improve on their denoising results by considering denoising under a per-pixel color model rather than a per-segment model.

Figure 2.16 shows comparisons of several of the denoising methods discussed in this section.

2.3 Up-sampling

As part of the imaging process, the continuous lighting domain of the world is discretely sampled by a sensor, where the size of the sensor and each pixel limits an image's resolution. In many scenarios, an image is needed at a higher resolution than it was captured, and, in these cases, an image must be up-sampled to this new target resolution. Ideally, the up-sampling process should preserve the sharpness of the original image, but in practice, this is not possible, as the highest frequency image data for the higher resolution is not present in the down-sampled observation. Thus the up-sampled image is invariably less sharp than the original. To combat this problem, up-sampling algorithms try to predict this missing data by interpolating or

“hallucinating” the high-frequencies given a particular image model.

In this section, we give an overview of image up-sampling algorithms. In some of the relevant literature these algorithms are referred to as performing super-resolution. We will use the terms super-resolution, up-sampling, and hallucination interchangeably, as is done in the literature; however, it is important to note the term super-resolution when used to refer to up-sampling is somewhat of a misnomer. The algorithms we address here and develop later in this dissertation do not perform super-resolution in the strictest sense. (True super-resolution constructs a high-resolution image of a scene from lower resolution images that over-sample the scene in a way that the high-resolution image contains accurate high-frequency data.) In contrast, up-sampling algorithms essentially “make-up” the higher-resolution data without directly deriving it from the scene.

2.3.1 Image Invariant Filters

The most basic approaches to image up-sampling are those that use data-invariant filters, such as nearest-neighbor, bilinear, bicubic, and Lanczos interpolation filters. These methods are the most well-known and more widely-used up-sampling methods and appear in numerous commercial image editing packages. These methods interpolate the missing data using a formula that is independent of the data. They assume that the image is smooth or band-limited, which is generally not the case [Fattal, 2007]. As a result, they produce visual artifacts such as blurring, ringing, and aliasing. We refer the reader to the paper by Thévenaz *et al.* [2000] for a more detailed survey of data invariant up-sampling methods.

2.3.2 Image Dependent Filters

Similarly to the denoising work discussed in the previous section, more sophisticated up-sampling methods are data dependent and modify their interpolation methods to better preserve edges. Carrato *et al.* [1996] change the parameters of a linear interpolation filter according to local intensity differences to better preserve edge strength and reduce blurring. They do this by biasing the interpolated pixels on 1D edge profiles to be closer to the end-points of the transition than one would get with the standard linear filter. While this method does produce somewhat

sharper images, by only strengthening 1D vertical and horizontal edges the method can create block-like artifacts. Su and Willis [2004] take another approach to modifying standard bilinear interpolation. Instead of using the values of the four bounding pixels as in bilinear interpolation, when an edge cuts through this four-pixel square, only the three out of the four pixels that are most consistent with being on one side of the edge are averaged to compute the interpolated value. This serves to maintain edges, that are more consistent with the estimated edge direction within the four pixel region. This method can also suffer from block like or stair-step artifacts as only an edge aligned with one of the two diagonals is retained. Li and Orchard [2000] alleviate these artifacts some by using a local intensity covariance matrix computed at the lower resolution to better model edge direction. They then use this covariance estimate to weigh neighboring pixel values in a way that is more consistent with the edge direction, which avoids averaging across the edge.

Tumblin and Choudhury [2004] propose a data structure where each pixel stores additional boundary information extracted from the original higher-resolution data (when later down-sampling and up-sampling is expected) or the boundary information is otherwise provided. This enables creating an image of any resolution where the boundary between two color regions is always at the proper location; however, this is generally only acceptable for piece-wise constant images and is not as directly useful for up-sampling where no high-resolution boundary data is available.

Instead of modifying interpolation weights, Greenspan *et al.* [2000] and Morse and Schwartzwald [2001] present methods that post-process standard linear interpolation methods to add sharpness. Greenspan *et al.* perform a multi-scale Laplacian decomposition of the interpolated image and extrapolate the coefficients at the highest pyramidal level to create ideal step edges that are consistent with the low-frequency image data. Morse and Schwartzwald smooth 2D level curves/isophotes (curves of constant intensity) of an up-sampled image using an iterative level-set framework. Their method smooths the curvature of the isophotes to remove stair-step artifacts or “jaggies”.

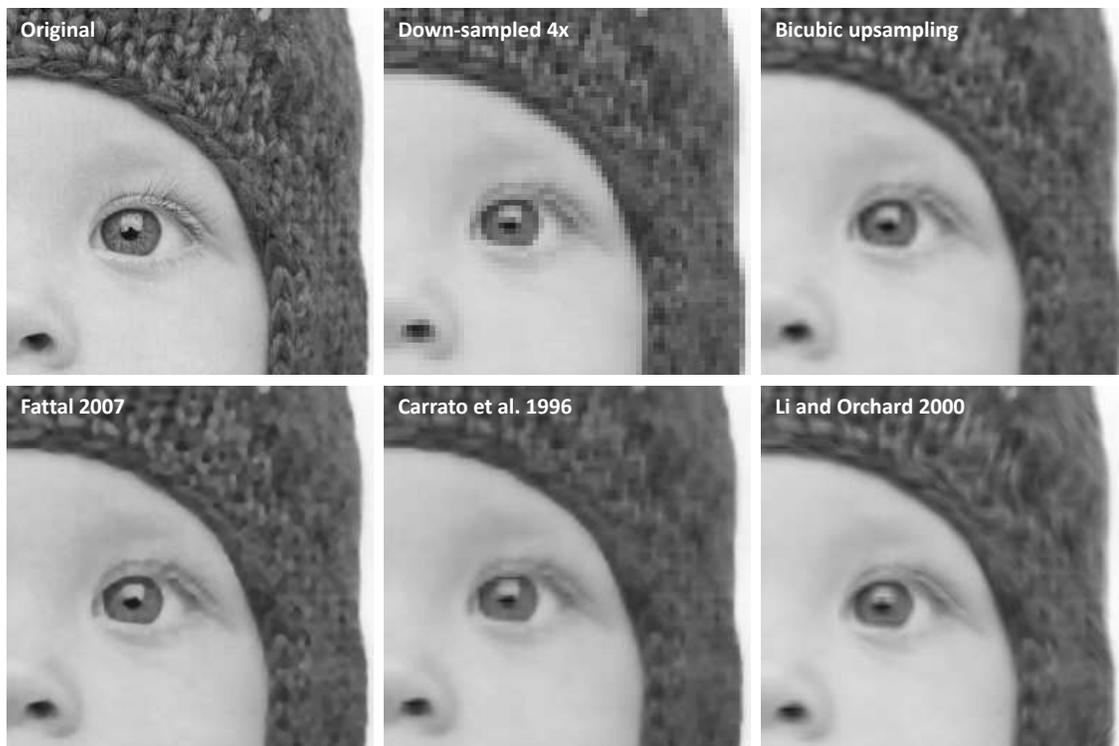


Figure 2.17: Comparisons of several up-sampling methods. [From Fattal 2007].

2.3.3 Using Image Priors

An alternative approach to filtering based up-sampling is to using a more global image prior or model. Hertzman *et al.* [2001] and Freeman *et al.* [2002] proposed similar methods to up-sample images by predicting the missing high-frequency data uses a non-parameteric, example based approach. Both methods use a training data set that consists of high and low resolution pairs and use this set to predict the highest frequency data for the input image based on how the mid-frequencies of the input image match the training pairs. Liu *et al.* [2007] take a similar approach, but specifically consider up-sampling face images using a training set of high and low resolution face images. These methods are quite interesting and have been quite successful, as they look beyond simply retaining edges and consider image data on a larger scale than the previously discussed methods. This allows them to predict more complex texture features. These methods inspired several of the algorithms discussed in Chapter 5 of this dissertation.

The work of Fattal [2007] and Dai *et al.* [2007] also use empirically derived priors but in a



Figure 2.18: Comparisons of example-based super resolution [Freeman et al. 2002] and Fattal's method [2007]. [From Fattal 2007].

slightly different way. Fattal derived an edge model that they use to predict the gradients of an up-sampled image in a way that maintains the first three moments of an edge profile from the low to high resolution image. Dai *et al.* on the other hand transform the image up-sampling problem

to that of an alpha matting problem, i.e., separation of the image into:

$$I = \alpha F + (1 - \alpha)B, \quad (2.34)$$

where F and B are up-sampled in standard way and α is up-sampled using a set of alpha-edge priors. The methods presented in Chapter 4 share some similarities with the work of Dai *et al.* in that we also consider alpha priors.

Figures 2.17 and 2.18 show comparisons of several of the up-sampling methods discussed in this section.

2.4 White-Balancing

It can be quite challenging to accurately convey colors using a photograph. Every step from capture to display can affect our subjective color perception. Perhaps the most important thing to account for are the differences in lighting color during image capture and viewing, as the viewing conditions of an image strongly affect our subjective sense of color due to the principle of color constancy.

Color constancy is the property that the perceived color of objects remains relatively constant under varying illuminations conditions. For example, our perception of the color of an orange is nearly the same whether it is viewed under broad-spectrum sunlight, yellow indoor tungsten light, or bluish indoor florescent lighting, when in fact the actual color spectrum reflected by the orange in these scenarios is quite different. This property is a result of chromatic adaptation of the human visual system to the lighting of the viewing conditions. For images to agree with our sense of color perception and seem realistic and convincing, it is important for the image to be matched to the conditions it will be viewed under, as those are the conditions our visual system will be adapted to at the time.

In certain scenarios, such as graphic design or professional photography printing, it is common to perform color matching that accounts for every step of the imaging pipeline by performing color management using calibrated setups and standardized color profiles. However, in most cases a simpler form of color management is acceptable, where matching is performed

under a certain set of assumptions. For most common photography and imaging applications it is reasonable to assume that images will be viewed under relatively neutral colored lighting. Thus color matching can be reduced to “white-balancing”, where the goal is to balance the color channels during capture or post-processing such that the image appears as if acquired under neutral light. In other words, the process of white-balancing a photograph seeks to remove the color cast introduced by the lighting during capture such that a “white” or “gray” object, an object that has equal reflectance across the color spectrum, has a neutral RGB value in the image.

Over the years, numerous approaches have been used to automatically white-balance an image. The methods fall into two basic categories: methods that estimate the illuminate color and those that attempt to match an image’s color characteristics to a reference image which has a desired color balance. In the following sections, we will give an overview of these methods.

2.4.1 Estimating Illumination Color

The most commonly used white-balancing algorithms explicitly or implicitly try to recover the illumination color from an image. The explicit methods estimate the color first and then apply a transformation to map that color to gray, while the implicit methods directly estimate the color balanced image, after which the light source color can be recovered.

The fastest, most straight-forward methods use low-level image statistics to estimate the light color. The “gray-world” method assumes that the average reflectance in the scene is achromatic and thus considers the mean of the image to indicate the color of the light [Land and McCann, 1971]. The image is then white-balanced by scaling the red and blue color components of the mean match the green value. The “max-RGB” techniques takes a similar approach, but estimates the light color as the maximum response of each color channel [Buchsbaum, 1980]. Finlayson and Trezzi’s [2004] “shades-of-gray” work unifies the gray-world and max-RGB methods as part of a more general color-constancy algorithm. Automatic white balance algorithms in digital cameras use either the gray-world or max-RGB methods due to their computational efficiency. A slightly more sophisticated low-level approach is the “gray-edge” method that derives the light color from edge statistics by assuming that the average edge difference is achromatic [van de Weijer et al., 2007]. Figure 2.19 shows a comparison of low-level statistical color constancy

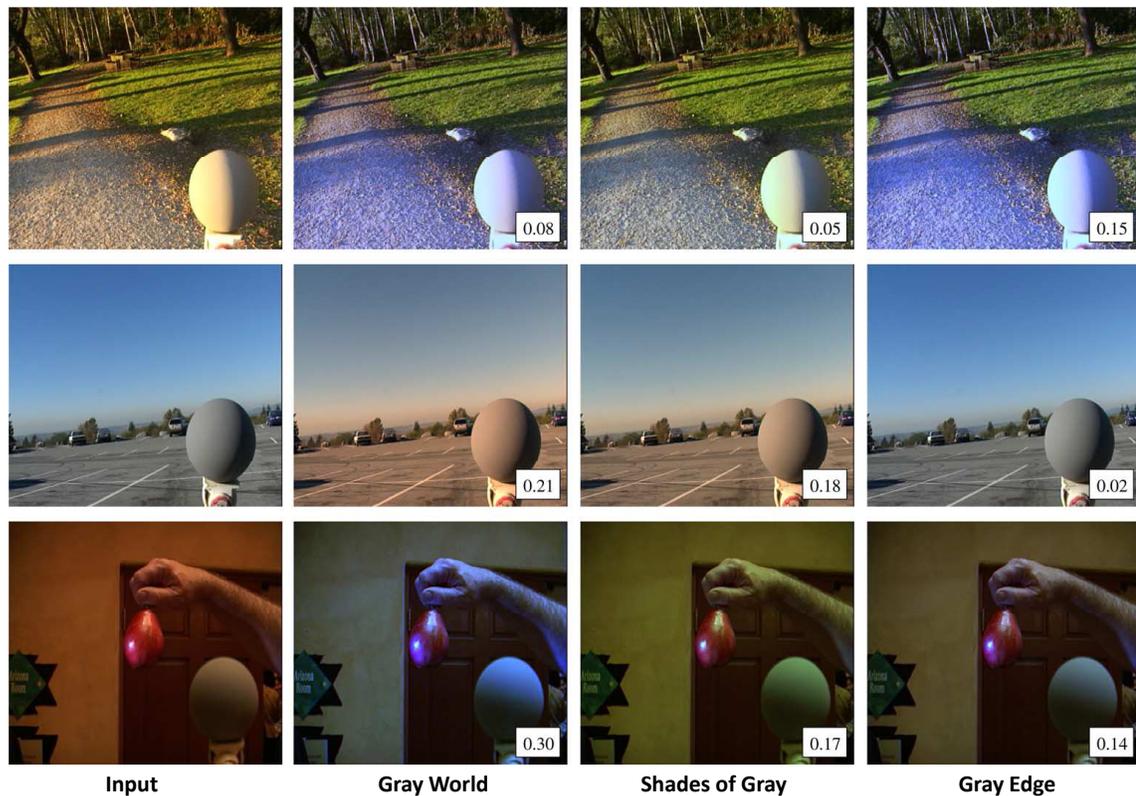


Figure 2.19: Comparisons of methods that estimate illuminant color using low-level statistics. The angular error, the angle between the estimated light color and known light color for the test data, is displayed in the lower-left corner of each image. A lower error indicates a better color estimate. [From van de Weijer 2007]

methods.

More sophisticated approaches incorporate priors on color casts of different types of illumination and the typical color distributions observed under these illuminants. One of the more successful methods is the “gamut-mapping” approach of Forsyth [1990], who uses the distribution of colors in a scene as an indication of the color of the light source. Forsyth observed that only a limited set of colors can be observed under a given illuminant, e.g., if a scene is strongly red, the illuminant is unlikely to be a strong green or blue. The gamut-mapping algorithm computes a transformation that maps an observed color gamut to a canonical white illuminant gamut, which implicitly estimates the illuminant color as it can be recovered from the color transformation.

Finlayson *et al.* [2001] take this approach further by operating in a chromaticity space, to be invariant to brightness changes, and use a correlation matching approach to find the most likely illuminant given the observed color gamut. While Finlayson and Hordley [2006] define a

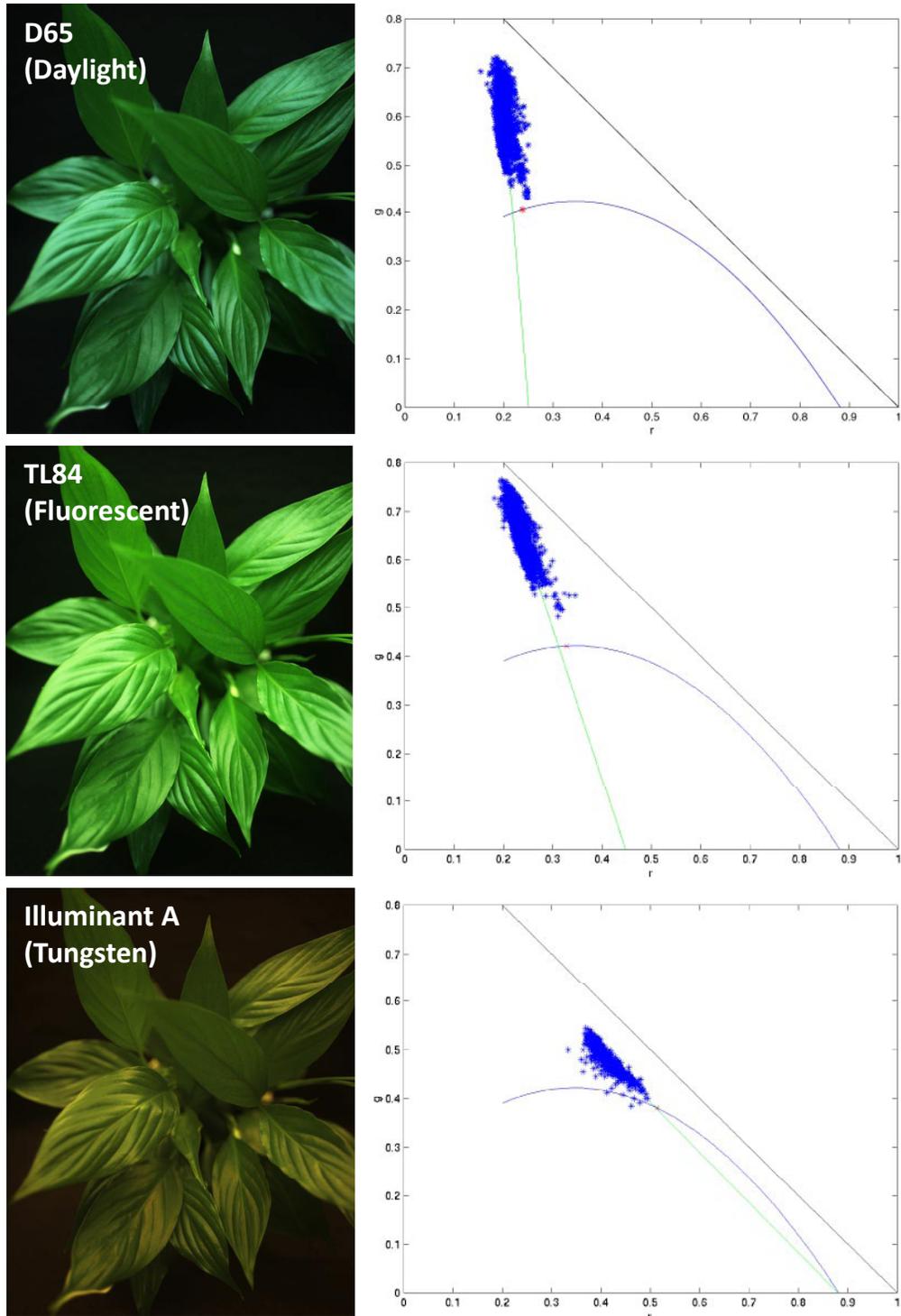


Figure 2.20: Estimating illuminant color using the intersection of a dichromatic line with the Planckian locus, i.e., the curve in chromaticity space as specified by Planck's law for black-body radiation. [Finlayson and Schaefer 2000]

specific set of plausible lights a priori to further restrict the color transformations. Brainard and Freeman [1997] take a similar approach and use a prior assumption on the likelihood of light colors in a Bayesian framework, while Cardei *et al.* [2002] use a neural network.

Each one of the illuminant estimation methods has its strengths and weaknesses. On the whole these methods can be quite fast and reliable, given that certain assumptions are met. The gamut-mapping style methods rely on the color distribution of an image to completely indicate the color of the light. In practice this works if the scene has objects that span a wide color gamut under a natural illuminant; however, in practice this is not often the case. For example, nature landscapes with trees, grass, and blue sky will be dominant in greens and blues and less so in reds, even under neutral illumination. In this case, a gamut-mapping approach would incorrectly identify the illuminant color as being greenish-blue and improperly color-balance the image. This dependence and ambiguity between scene color content and the color distribution imposed by the light can significantly limit the quality of such methods.

Another set of approaches relies on the assumption that images in the scene adhere to a dichromatic model. In regions of constant color albedo, the observed chromaticity for the region lies on a line between the region's color albedo and the illuminant color. In the full RGB space the colors lie on a plane instead of a line. Thus by observing different dichromatic surfaces, Lee [1986] and Tominaga and Wandell [1989], respectively showed that the illuminant color can be recovered by intersecting dichromatic lines or planes. Finlayson and Schaefer [2000] further constrained this approach by assuming that the illumination spectra follows Planck's law, which describes black-body light spectra as a function of temperature, thereby reducing the 2D dichromatic space for lights to a 1D space. With Finlayson and Schaefer's work only one dichromatic surface needs to be identified. Then the corresponding single dichromatic line is intersected with the 1D curve as specified by Planck's law. Figure 2.20 shows an illustration of this method for three different illuminant types.

While dichromatic methods avoid the aforementioned problems of gamut-mapping methods, they have their own shortcomings. These methods rely on being able to identify one or many dichromatic surfaces. This can be difficult to do automatically, and often a human observer is expected to provide this identification.



Figure 2.21: Color matching using linear transformations in RGB and $l\alpha\beta$ color spaces. Using a linear transformation in color space that is a non-linear map of RGB can produce better matching results. Here the result using $l\alpha\beta$ better matches the look of the reference image. [Reinhard et al. 2001]

2.4.2 Color Matching

A very different approach to color-balancing is to perform color matching. Color matching assumes that one wants to color balance an image relative to some reference image. This approach avoids some of the difficulties of estimating the illuminant color as it does not require an image to have a very broad color gamut and it does not require the need for dichromatic surfaces. Color matching methods assume that a reference photograph is available, that the illumination color in the reference is the desired target illumination for the correction, and that the inherent color distributions of the images are the same, i.e., the two images would have similar color distributions if the scenes were imaged under the same lighting conditions.

The most basic color matching approach is a linear color transformation. A linear color transformation consists of a full 3x3 or 3x4 color transformation (3x4 if per-channel brightness/offset adjustments are desired). The linear color transform can be computed directly to transform pixels in one image directly to the values in the reference if the images have pixel-wise correspondence (e.g., the same scene and camera configuration with only a lighting color difference). More commonly, the transform is computed using some statistical measure of the image, such as matching the mean or median color. Reinhard *et al.* [2001] explore this type of correction and propose first performing a non-linear mapping of the RGB color space to the $l\alpha\beta$ color space before computing linear transforms. Numerous researchers have explored taking a similar approach in a variety of color spaces, such as HSV, CIEXYZ, LUV, and CIELAB. Figure 2.21 shows a comparison of using Reinhard *et al.* method in RGB and $l\alpha\beta$.

Another non-linear approach is to operate on the full color histogram of an image, rather than the simplification of a mean color and the standard deviation as in Reinhard *et al.*'s [2001]

work. Histogram matching is a standard technique that computes a non-linear remapping of an image's color space, implemented as a look-up table, such that an image's color histogram after the remapping matches the histogram of the reference [Gonzalez and Woods, 2001]. Histogram matching can be done in the full RGB space, in 2D chromaticity space, or independently in 1D for the red, green, and blue channels. Performing matching in RGB or chromaticity space can be more successful but is more computationally costly, while independent matching on each color channel is fast but may not always account for differences in color balance, as it does not account for correlation between the color channels. Pitie *et al.* [2005] recently proposed a method to address this by performing a randomized search for a color rotation matrix that decorrelates the red, green, and blue axes first to improve the quality of independent 1D histogram matching.

PSF Estimation using Sharp Edge Prediction

“Things are only impossible until they’re not.”

—Captain Jean-Luc Picard

In this chapter, we present an algorithm that estimates non-parametric, spatially-varying blur functions (i.e., point-spread functions or PSFs) from a single image. Our method handles blur due to defocus, slight camera motion, and inherent aspects of the imaging system. Our algorithm can be used to measure blur due to limited sensor resolution by estimating a sub-pixel, super-resolved PSF even for in-focus images. It operates by predicting a “sharp” version of a blurry input image and uses the two images to solve for a PSF. We handle the cases where the scene content is unknown and also where a known printed calibration target is placed in the scene. Our method is automatic, is fast, and produces accurate results.

Image blur is introduced in a number of stages in a camera. The most common sources of image blur are motion, defocus, and aspects inherent to the camera, such as pixel size, sensor resolution, and the presence of anti-aliasing filters on the sensor.

When blur is undesirable, one can deblur an image using a deconvolution method, which requires accurate knowledge of the blur kernel. In applications where blur is desirable and essential, such as shape from defocus, it is still necessary to recover the shape and size of the spatially varying blur kernel.

Recovering a PSF from a single blurred image is an inherently ill-posed problem due to the loss of information during blurring. The observed blurred image provides only a partial

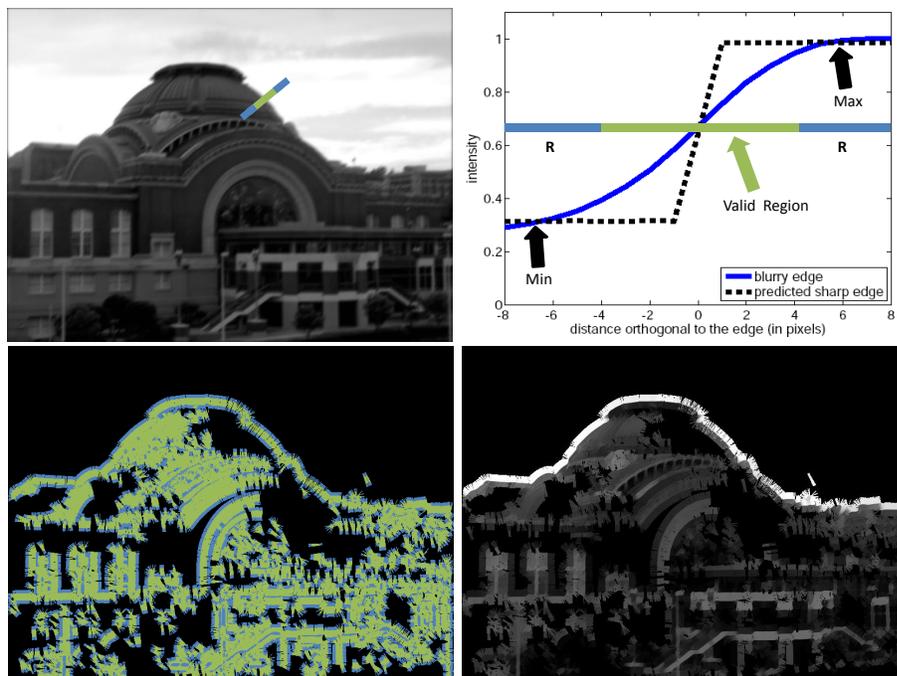


Figure 3.1: Sharp Edge Prediction. A blurry image (top left) and the 1D profile normal to an edge (top right, blue line). We predict a sharp edge (top right, dashed line) by propagating the max and min values along the edge profile. The algorithm uses predicted and observed values to solve for a PSF. Only observed pixels within a radius R are used. (bottom left) Predicted pixels are blue and valid observed pixels are green. (bottom right) The predicted values.

constraint on the solution, as there are many combinations of PSFs and “sharp” images that can be convolved to match the observed blurred image.

Prior knowledge about the image or kernel can disambiguate the potential solutions. Early work in this area significantly constrained the form of the kernel [Kundur and Hatzinakos, 1996], while more recently, researchers have put constraints on the underlying sharp image [Fergus et al., 2006]. In our work, we take the latter approach; however, instead of using statistical priors, we leverage our prior assumption more directly. Specifically, we present an algorithm for estimating regions of a sharp image from a blurry input—if one can estimate the sharp image, recovering the blur kernel is possible.

The key insight of our work is that with certain types of image blur, the location of image features such as edges are detectable even if the feature strength is weakened. When the scene content is unknown, we detect edges and predict the underlying sharp edges that created the

blurred observations, under the assumption that the detected edge was a step edge before blurring. Each pair of predicted and blurred edges gives information about a radial profile of the PSF. If an image has edges spanning all orientations, the blurred and predicted sharp image contain enough information to solve for a general two-dimensional PSF.

For situations where the scene content can be controlled, we have designed a printed calibration target whose image is automatically aligned with a known representation of the target. We then use this pair to solve for an accurate PSF.

Our method has several advantages over previous approaches: it measures the entire PSF of an image system from world to image, it is fast and accurate, and it can solve for spatially varying PSFs at sub-pixel resolution using only a single image.

We show results for both unknown scenes and images of our calibration target. We present deconvolution results using the recovered PSFs to validate the blur kernels and show a synthetic experiment to further evaluate the method. We also show that by solving for spatially varying, per-color channel PSFs combined with per-channel radial distortion corrections, we can remove chromatic aberrations artifacts.

3.1 Related Work

The problem of blur kernel estimation and more generally blind deconvolution is a longstanding problem in computer vision and image processing. Chapter 2 presents a discussion of this general area. To briefly review, work in single image blind deconvolution falls in to two areas: those that use parametric models for PSFs and those that use a non-parametric model. The bulk of the previous work in blur estimation lies in the first area, while there is significantly less work in the area of single image blur estimation using non-parametric kernels.

The work by Fergus *et al.* [2006] is perhaps the most notable non-parametric method. Fergus *et al.* use natural image statistics to derive an image prior that is used in a variational Bayes formulation. In contrast, we leverage prior assumptions on images to directly predict the underlying sharp image. We consider our approach complementary to that of Fergus *et al.*, as our method excels at accurately computing smaller kernels, and it can be used for lens and sensor

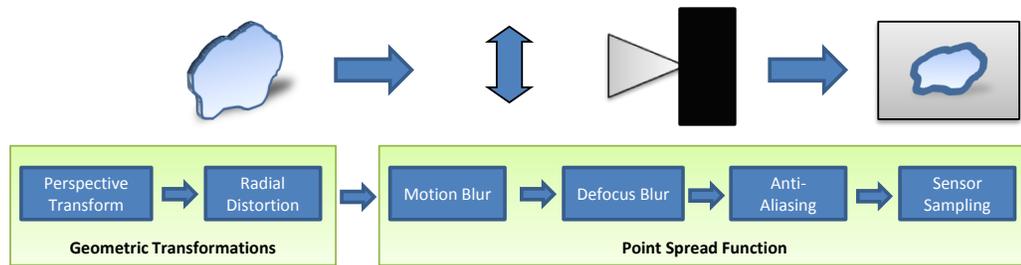


Figure 3.2: Image Formation Model. The imaging model consists of two geometric transforms as well as blur induced by motion, defocus, sensor anti-aliasing, and finite-area sensor sampling. We solve for an estimate of the continuous point-spread function at each discretely sampled (potentially blurry and noisy) pixel.

characterization. In contrast, their method is not as well suited to these applications, but excels at computing large kernels due to complex camera motion, which is outside the scope of our work.

Our work is conceptually most similar to slant-edge calibration [Reichenbach et al., 1991; Burns and Williams, 1999]. These methods recover 1D blur profiles by imaging a slanted edge feature and finding the 1D kernel normal to the edge profile that gives rise to the blurred observations of the known step edge. Reichenbach *et al.* [1991] note that one can combine several 1D sections to estimate a 2D PSF. We take a similar approach philosophically to slant-edge techniques, with three major differences: we extend the method to *directly* solve for 2D PSFs, solve for spatially varying PSFs, and further present a blind approach where the underlying step edge is not known a priori.

A related area is modulation transfer function (MTF) estimation for lenses that uses images of random dot patterns [Levy et al., 1999]. In theory, infinitesimal dot patterns are useful for PSF estimation, but in practice, it is not possible to create such a pattern. In contrast, creating sharp step edges is relatively easy and thus generally preferable [Reichenbach et al., 1991]. An additional advantage of our work relative to using dot patterns is that by using a grid-like structure with regular, detectable corner features, we can compute a radial distortion correction in addition to estimating PSFs.

3.2 Image Formation Model

We now give a brief overview of relevant imaging and optics concepts needed for PSF estimation. As illustrated in Figure 3.2, the imaging model consists of two geometric transforms: a perspective transform (used when photographing a known planar calibration target) and a radial distortion. There are several sources of blur induced by motion, defocus, sensor anti-aliasing, and pixel sampling area (fill factor and active sensing area shape). We model all sources of blur as a convolution along the image plane and account for depth dependent defocus blur and 3D motion blur by allowing the PSF to be spatially varying.

Our method estimates a discretely sampled version of the continuous PSF by either matching the sampling to the image resolution (which is useful for estimating large blur kernels) or using a sub-pixel sampling grid to estimate a detailed PSF, which captures effects such as aliasing in the sensor and allows us to do more accurate image restoration. In addition, by computing a sub-pixel PSF, we can perform single-image super-resolution by deconvolving up-sampled images with the recovered PSF.

Geometric Transformations: The world to image transformation consists of a perspective transform and a radial distortion. With the blind method, we ignore the perspective transform and operate in image coordinates.

With the non-blind method, where we photograph a known calibration target, we model the perspective transformation as a 2D homography to map known feature locations F^k on the grid pattern to detected feature points from the image F^d . We use a standard model for radial distortion: $(F'_x, F'_y)^T = (F_x, F_y)^T (a_0 + a_1 r^2(x, y) + a_2 r^4(x, y))$, where $r(x, y) = \sqrt{F_x^2 + F_y^2}$ is the radius relative to the image center.

Given a radial distortion function $R(F)$ and warp function which applies a homography $H(F)$, the full alignment process is $F^d = R(H(F^k))$. We compute the parameters that minimize the L_2 norm of the residual $\|F^d - R(H(F^k))\|^2$. Computing these parameters cannot be done simultaneously in closed form. However, the problem is bilinear, and thus we solve for the parameters using an iterative approach.

Modeling the Discrete Point-Spread Function: The equation for the observed image B is a convolution of a kernel K and a potentially higher-resolution sharp image I , plus additive Gaussian white noise, whose result is potentially down-sampled:

$$B = D(I \otimes K) + N, \quad (3.1)$$

where $N \sim \mathcal{N}(0, \sigma^2)$. $D(I)$ down-samples an image by point-sampling $I_L(m, n) = I(sm, sn)$ at a sampling rate s for integer pixel coordinates (m, n) . In our formulation, the kernel K models all blurring effects, which are potentially spatially varying and wavelength dependent.

3.3 Sharp Image Estimation

The blurring process is formulated as an invertible linear system, which models the blurry image as the convolution of a sharp image with the imaging system's PSF. Thus, if we know the original sharp image, recovering the kernel is straightforward. The key contribution of our work is a reliable and widely applicable method for predicting a sharp image from a single blurry image. In the following section, we present our methods for predicting the sharp image. In Section 3.4, we discuss how to formulate and solve the invertible linear system to recover the PSF. In the following discussion, we consider images to be a single channel (grayscale); in Section 3.5, we discuss color images.

3.3.1 Blind Estimation

For blind sharp image prediction, we assume that blur is due to a PSF with a single mode (or peak), such that when an image is blurred, the ability to localize a previously sharp edge is unchanged; however, the strength and profile of the edge is changed, as illustrated in Figure 3.1. Thus, by localizing blurred edges and predicting sharp edge profiles, estimating a sharp image is possible.

We assume that all observed blurred edges result from convolving an ideal step edge with the unknown kernel. Our algorithm finds the location and orientation of edges in the blurred

image using a sub-pixel difference of Gaussians edge detector. It then predicts an ideal sharp edge by finding the local maximum and minimum pixel values, in a robust way, along the edge profile and propagates these values from pixels on each side of an edge to the sub-pixel edge location. The pixel at an edge itself is colored according to the weighted average of the maximum and minimum values according to the distance of the sub-pixel location to the pixel center, which is a simple form of anti-aliasing (see Figure 3.1).

We find the maximum and minimum values robustly using a combination of two techniques. To find the maximum value, our algorithm marches along the edge normal, sampling the image looking for a local maximum using hysteresis. Specifically, the maximum location is the first pixel that is less than 90% (as opposed to strictly less than) of the previous value. Once this value and location are identified, we store the “maximum” value as the mean of all values along the edge profile that are within 10% of the initial maximum value. An analogous approach is used for the minimum.

Since we can only reliably predict values near edges, we only use observed pixels within a radius of the predicted sharp values. These locations are stored as valid pixels in a mask, which is used when solving for the PSF, as discussed in Section 3.4. At the end of the prediction process, we have a partially estimated sharp image, as shown in Figure 3.1.

3.3.2 Non-Blind Estimation

For non-blind sharp edge prediction, we want to compute the PSF given that we know the sharp image. Since we anticipate using this technique in a controlled lab setup, we designed a special calibration pattern for this purpose (Figure 3.3). We take an image of this pattern and align the known grid pattern to the image to get the sharp/blurry pair needed to compute the PSF accurately. The grid has corner (checkerboard) features so that it can be automatically detected and aligned, and it also has sharp step edges equally distributed at all orientations within a tiled pattern, so that it provides edges that capture every radial slice of the PSF. (Alternatively, we can say that the calibration patterns provides measurable frequencies at all orientations.) Furthermore, we represent the grid in mathematical form (the curved segments are 120° arcs), which gives us a very precise definition for the grid, which is advantageous for performing alignment.

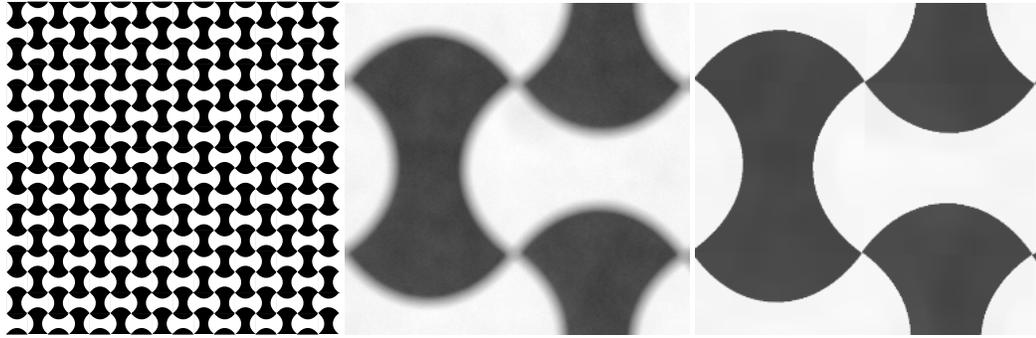


Figure 3.3: Non-Blind Estimation. (left) The tiled calibration pattern, (middle) cropped section of an image of a printed version of the grid, and (right) the corresponding cropped part of the known grid warped and shaded to match the image of the grid.

For non-blind prediction, we continue to assume that kernel has no more than a single peak. Thus even when the pattern is blurred, we can detect corners on the grid with a sub-pixel corner detector. Because our corners are actually balanced checkerboard crossings (radially symmetric), they do not suffer from “shrinkage” (displacement) due to blurring. Once corners are found, the ground truth pattern is aligned to the acquired image. To obtain an accurate alignment, we correct for both geometric and radiometric aspects of the imaging system.

We perform geometric alignment using the corrections discussed in Section 3.2. We fit a homography and radial distortion correction to match the known feature locations on the grid pattern to corners detected with sub-pixel precision on the acquired (blurry) image of the printed grid.

We also must account for the lighting and shading in the image of the grid. We do this by first aligning the known grid to the image. Then, for each edge location (as known from mathematical form of the ground truth grid pattern), the algorithm finds the maximum and minimum values on the edge profile and propagates them just as in the non-blind approach. We shade the grid for pixels within the blur radius of each edge. By performing the shading operation, our algorithm has corrected for shading, lighting, and radial intensity falloff. Figure 3.3 shows the results of the geometric warp and shading transfer.

3.4 PSF Estimation

Once the sharp image is predicted, we estimate the PSF as the kernel that convolved with the sharp image produces the blurred input image. We formulate the estimation using a Bayesian framework solved using a *maximum a posteriori* (MAP) technique. In MAP estimation, one tries to find the most likely estimate for the blur kernel K given the sharp image I and the observed blurred image B , using the known image formation model and noise level.

We express this as a maximization over the probability distribution of the posterior using Bayes' rule. The result is minimization of a sum of negative log likelihoods $L(\cdot)$:

$$P(K|B) = P(B|K)P(K)/P(B) \quad (3.2)$$

$$\operatorname{argmax}_K P(K|B) = \operatorname{argmin}_K L(B|K) + L(K). \quad (3.3)$$

The problem is now reduced to defining the negative log likelihood terms. Given the image formation model (Equation 3.1), the data term is:

$$L(B|K) = \|M(B) - M(I \otimes K)\|^2 / \sigma^2. \quad (3.4)$$

(The downsampling term D in (3.1) will be incorporated in Section 3.4.1.) $M(\cdot)$ is a masking function such that this term is only evaluated for “known” pixels in B , i.e., those pixels that result from the convolution of K with properly estimated pixels I , which form a band around each edge point, as described in Section 3.3.1.

The remaining negative log likelihood term, $L(K)$, models prior assumptions on the blur kernel and regularizes the solution. We use a smoothness prior and a non-negativity constraint. The smoothness prior penalizes large gradients and thus biases kernel values to take on values similar to their neighbors: $L_s(K) = \lambda \gamma \|\nabla K\|^2$. λ controls the weight of the smoothness penalty, and $\gamma = (2R + 1)^2$ normalizes for the kernel area (R is the kernel radius). Since the kernel should sum to one (as blur kernels are energy conserving) the individual values decrease with increased R . This factor is needed to keep the relative magnitude of kernel gradient values on par with the data term values regardless of kernel size.

We minimize the following error function:

$$L = \|M(B) - M(I \otimes K)\|^2 / \sigma^2 + \lambda \gamma \|\nabla K\|^2, \quad (3.5)$$

subject to $K_i \geq 0$, to solve for the PSF using non-negative linear least squares using a projective gradient Newton's method. We currently estimate the noise level σ using a technique similar to that of Liu *et al.* [2006], and we have empirically found $\lambda = 2$ to work well.

3.4.1 Computing a Super-Resolved PSF

By taking advantage of sub-pixel edge detection for blind prediction and sub-pixel corner detection for non-blind prediction, we can estimate a super-resolved blur kernel by predicting a sharp image at a higher resolution than the observed image.

For the blind method, in the process of estimating the sharp image, it is necessary to rasterize the predicted sharp edge-profile back onto a pixel grid. By rasterizing the sub-pixel sharp-edge profile onto an up-sampled grid, we can estimate a super-resolved sharp image. In addition, at the actual identified edge location (as before), the pixel color is a weighted average of the minimum and maximum, where the weighting reflects the sub-pixel edge location on the grid.

For the non-blind method, we also must rasterize the grid pattern at a some desired resolution. Since we detect corners at sub-pixel precision, the geometric alignment is computed with sub-pixel precision. Using the mathematical description of our grid, we can choose any up-sampled resolution when rasterizing the predicted sharp image. We also perform anti-aliasing, as described in Section 3.3.2.

To solve for the PSF using the super-resolved predicted sharp image I_H and the observed (vectorized) blurry image b , we modify Equation 3.4 to include a down-sampling function according to our image model (Equation 3.1). We consider $\hat{b}_H = A_H k_H$ to be super-resolved sharp image blurred by the super-resolved kernel k_H , where A_H is the matrix form of I_H . Equation 3.4 is then $\|b - DA_H k_H\|^2$ (we have left out the masking function for readability). D is a matrix reflecting the down-sampling function: $\hat{B}_L(m, n) = \hat{B}_H(sm, sn)$.

3.4.2 Computing a Spatially Varying PSF

Computing a spatially varying PSF is easy given our formulation—we simply perform the MAP estimation process described in the previous section for sub-windows of the image. The process operates on any size sub-window as long as enough edges at different orientations are present in that window. In the limit, we could compute a PSF for every pixel using sliding windows. We have found, in practice, that such a dense solution is not necessary, as the PSF tends to vary spatially relatively slowly.

Our method requires enough edges to be present at most orientations. When using the entire image, this is not usually an issue; however, when using smaller windows, the edge content may under-constrain the PSF solution. We have a simple test that avoids this problem. We ensure that (a) the number of valid pixels in the mask described in Equation 3.4 is greater than the number of unknowns in the kernel, and (b) we compute a histogram of 10 degree bins of the detected edges orientations and ensure that each bin contains at least 100 edges. When this check fails, we do not compute a kernel for that window.

3.5 Chromatic Aberration

In the previous sections, we did not explicitly address solving for PSFs for color images. To handle color, one could convert the image to grayscale. In many cases this is sufficient; however, it is more accurate to solve for a PSF for each color channel. This need arises when chromatic aberration effects are apparent.

Due to the wavelength-dependent variation of the index of refraction of glass, the focal length of a lens varies continually with wavelength. This property causes longitudinal chromatic aberration (blur/shifts along the optical axis), which implies that the focal depth, and thus amount of defocus, is wavelength dependent. It also causes lateral chromatic aberration (blur/shifts perpendicular to the optical axis). We refer the reader to the paper by Kang [2007] for a more detailed discussion of these artifacts.

By solving for a PSF per color channel, we can model the longitudinal aberrations; we use

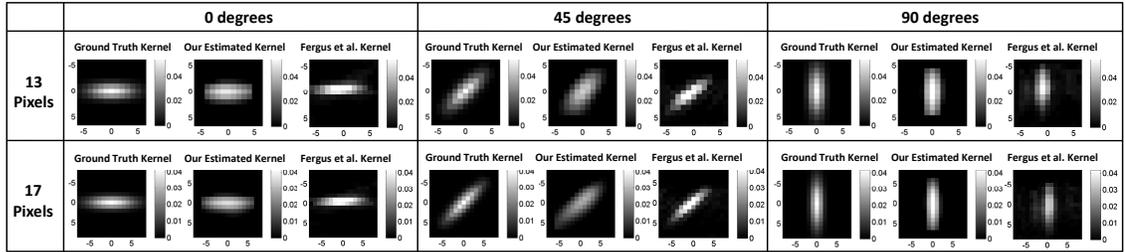


Figure 3.4: Recovering Blur Kernels of Different Sizes and Orientations. We convolved the sharp original version of the image shown in Figure 3.1 with kernels of 13 and 17 pixels for three different orientations. Each set is a side by side comparisons of the ground truth (left), our recovered kernel (middle), and the result of running Fergus *et al.*'s [2006] method (right).

a per-color channel radial distortion correction to handle the lateral distortions. We correct for lateral distortions by first performing edge detection on each color channel independently and only keeping edges that are detected within 5 pixels of each other in R, G, and B. We then compute a radial correction to align the R and B edges to the G edges and then perform blind sharp image prediction.

To correct for any residual radial shifts, we use the green edge locations for all color channels so that all color bands have sharp edges predicted at the same locations. One could perform this last step without correcting radial distortion first and allow the shifts to be entirely modeled within the PSF; however, we have found the two stage approach is better, as it removes some aberration artifacts even when there is not enough edge information to compute a PSF, and by removing the majority of the shift first, we can solve for smaller kernels.

If we have access to RAW camera images, we can compute more accurate per-channel PSFs by accounting for the Bayer pattern sampling during PSF computation instead of using the demosaicked color values. We solve for a PSF at the original image resolution, which is 2x the resolution for each color channel and use the point sampling function discussed in Section 3.2, where the sampling is shifted according to the appropriate Bayer sample location.

3.6 Results

To validate our blind prediction method, we synthetically blurred a sharp image with oriented Gaussian kernels of 13 and 17 pixels in diameter for three different orientations, added

Gaussian white noise with standard-deviation 0.01 (where 0=black and 1=white), and then estimated the blur kernel using our blind method. Figure 3.4, shows a comparison of the ground truth kernels, our recovered kernels, and the result of running Fergus *et al.*'s [2006] method. Our blind algorithm recovers the size and shape of each kernel accurately.

In Figures 3.5 and 3.7, we show results for estimating kernels for images with real, unknown blurs, where there is both defocus and camera motion blur. Our method predicts slightly asymmetric disk-like kernels that are consistent with defocus and slight motion blur.

To qualitatively validate these kernels, we deconvolve the input images using the Lucy-Richardson algorithm. We chose this over other methods as it produces results with a good balance of sharpness and noise reduction. Furthermore, the method is less forgiving than some newer methods, which allows for better validation. (Deconvolution with an incorrect kernel leads to increased ringing artifacts, as shown in Figure 3.6). Our resulting deconvolved images are significantly sharper and show relatively minimal ringing artifacts, which indicates that the kernels are accurate.

In Figure 3.5, we also compare our recovered kernel to a result from running Fergus *et al.*'s code. The kernel obtained by their method has more noise than ours, does not have a shape consistent with defocus blur, and the size of the non-zero area of the kernel does not match the amount of blur seen in the input image. Fergus *et al.*'s method took 21 minutes, while ours took 2.5 seconds for the original resolution and 9.5 seconds at 2x super-resolution. Our method is significantly faster as its running time scales with the number of edges and kernel size, while the Fergus *et al.* method is a multi-resolution method whose speed scales with *image size* and

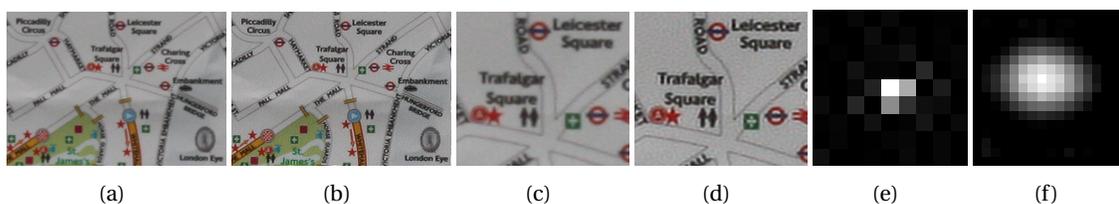


Figure 3.5: Defocus and Slight Motion-Blur. (a) The original blurred image and (b) the deconvolved output using our recovered PSF. (c–d) Zoomed-in versions of the original and deconvolved image respectively. (e) The kernel recovered using the method of Fergus *et al.* [2006] and (f) our recovered kernel.



Figure 3.6: Kernel Size and Orientation. Image deconvolved with (left) our kernel, (middle) our kernel scaled 20% larger, and (right) our kernel rotated by 45° . The middle and right images have more ringing (most apparent at the bottom of the word “Leicester”).

kernel size. Our method is a couple seconds faster when using regular least-squares instead of a non-negative version; however, more smoothing is needed to suppress large negative values. Thus we prefer to enforce non-negativity as it produces sharper PSFs.

Figure 3.8 displays an image with camera motion blur. Our recovered kernel correctly shows the diagonal motion blur that is apparent in the input image. The deconvolved image is much sharper with minimal ringing.

In Figure 3.9, we show super-resolution results where we have taken a sharp image, bicubically down-sampled it by 4x, and then solved for a 4x super-resolved kernel from the down-sampled input. We compare the original full resolution image to a bicubically up-sampled version of the low-resolution image and to the up-sampled image deconvolved with our recovered kernel. The deconvolution results show a sharpening and recovery of high-frequency texture that is consistent with the full resolution images.

Figure 3.10 shows results for our calibration grid captured with an 11 mega-pixel Canon 1Ds camera using a Canon EF 28-200mm f3.5-5.6 lens at two apertures and focal lengths. For each image, we computed spatially varying PSFs by computing kernels for non-overlapping 220-pixel (the size of one grid tile) windows across the image at 2x resolution, i.e., two times the Bayer sampling resolution. Each PSF is displayed according to the location of its corresponding image window. The recovered PSFs show some interesting properties. The PSF should be an image of the aperture, and some do show the shape of the aperture, which we know from the lens specifications to have 6 blades. They also show “donut” artifacts that can occur at some settings



Figure 3.7: Defocus and Slight Motion-Blur. (top left) The original blurred image and (top right) the deconvolved output with the recovered kernel displayed in the top right of the image (the kernel has been enlarged by 10x for display). (bottom row) Zoomed-in versions of the original and deconvolved image, respectively.

with lower-quality lenses. Perspective distortion across the image plane and vignetting (clipping of the aperture) by the lens barrel are also visible. For comparison we imaged back-lit pinholes at the same camera settings. Imaging pinholes to measure PSFs has some inherent problems due to the pinhole actually being a disk and not an infinitesimal point and due to diffraction; however, these images validate our recovered PSFs.

We also acquired a very sharply focused image, so that we could measure sub-pixel blur. Figure 3.11 shows an image of our grid from a 6 mega-pixel Canon 1D, using a high-quality Canon EF 135mm f/2L lens. We show recovered PSFs at 1x, 2x, 8x, and 16x sub-pixel sampling. The PSFs using higher sub-pixel resolution show an interesting structure that results from a combination of



Figure 3.8: Motion Blur. (top row) The original blurred image (left) and the deconvolved output (right) with the recovered kernel displayed in the top right of the image (the kernel has been enlarged by 10x for display). (bottom row) Zoomed-in versions of the original and deconvolved image, respectively.

diffraction, lens imperfections, and sensor anti-aliasing and sampling.

Figure 3.12 shows a result for performing blind chromatic aberration correction for a JPEG image from a Canon S60 using a 5.8mm focal length at f8. After performing radial distortion correction and piecewise deconvolution using the spatially varying PSF, the aberration artifacts are significantly reduced. Figure 3.13 shows chromatic aberration correction for our non-blind method. We encourage the reader to visit <http://graphics.ucsd.edu/~neel/dissertation/> for additional results.

3.7 Discussion and Future Work

We have shown how to recover spatially varying PSFs at sub-pixel precision that capture blur due to motion, defocus, and intrinsic camera properties. Our method is fast, straightforward to implement, and predicts kernels accurately for a wide variety of images. Nevertheless, our



Figure 3.9: 4x Super-Resolution. (left) The original image and zoom-in, (middle) the original image bi-cubically down-sampled and up-sampled by 4x and zoom-in, (right) the up-sampled image deconvolved using the recovered 4x super-resolved kernel (displayed in the top left of the image – the kernel has been enlarged by 10x for display) and a zoom-in on the bottom.

method does have some limitations, and there are several avenues for future work.

The primary limitation of our method is that we can only solve for kernels with a single peak. This limitation is due to relying on an edge detector to find a single location for every blurred edge. In the case of a multi-peaked kernel, our method will incorrectly interpret the “ghost” copies of edges as independent edges. While we have shown that single-peaked kernels model many commonly occurring cases of blur, we would like to extend our method to handle multi-modal kernels. One option is to group each stronger edge with its weaker ghost edges using contour matching. Once the ghost edges are identified, we could perform sharp edge prediction only for the primary edges.

Another approach is to incorporate sharp-edge prediction into an iterative blind deconvolution algorithm, such as the methods used by Wan and Nowak [1999], Fergus *et al.* [2006], and Shan *et al.* [2008]. These methods use natural image priors into an minimization that alternates between solving for an unknown kernel (PSF estimation) and solving for an unknown sharp image (non-blind deconvolution) given an initial guess for both. Figure 3.14 shows an example

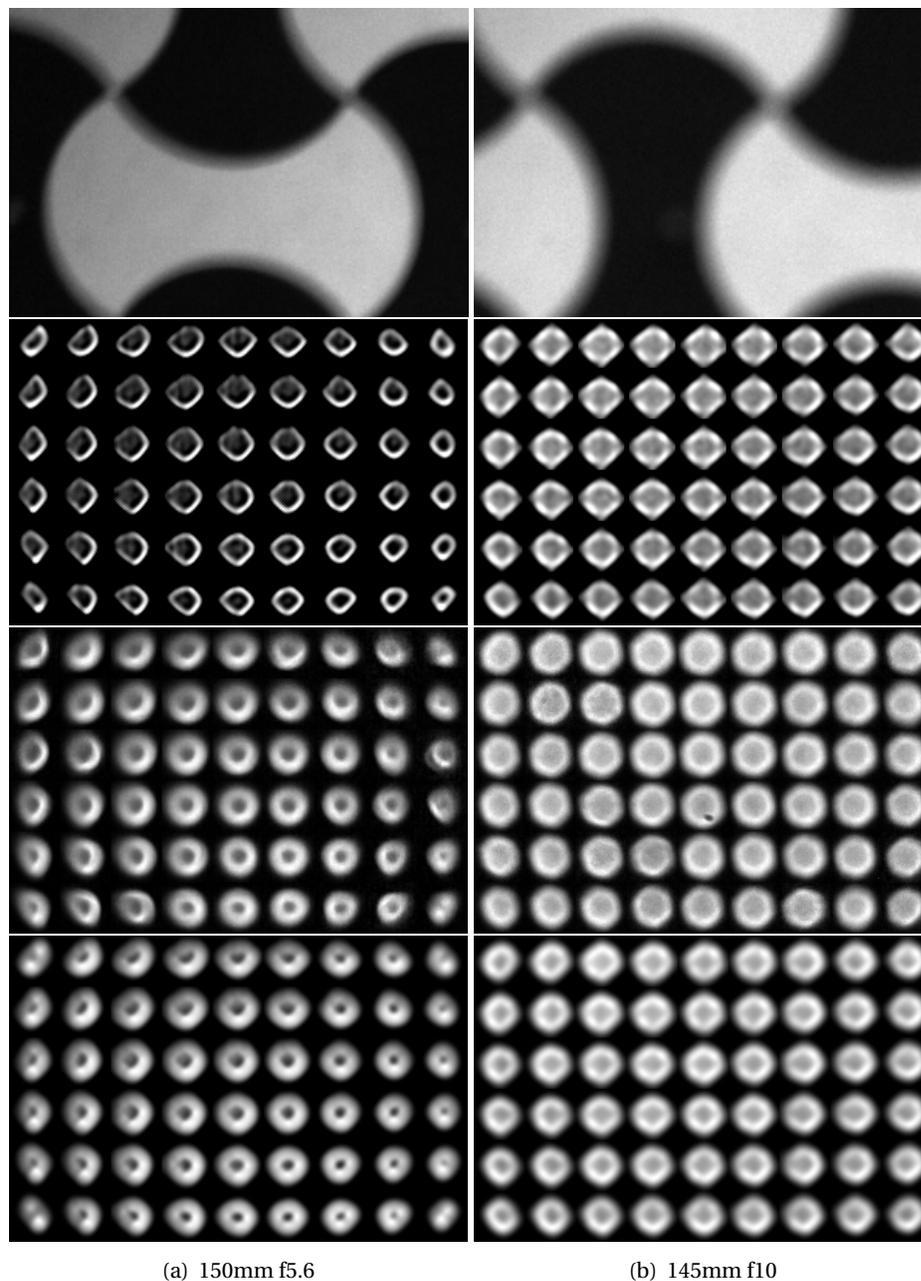


Figure 3.10: Different Apertures and Focal Lengths. (first row) Cropped portions of the observed blurred images, (second row) recovered spatially varying PSFs (green channel only), (third row) images of pinholes at the same depths and settings, and (fourth row) our recovered PSFs convolved with a disk the size of the pinhole. For (a) each PSF is 33×33 pixels and (b) they are 41×41 pixels. The PSFs reflect the shape of the aperture and show perspective distortion and vignetting across the image plane.

of preliminary work that performs a multi-scale iterative blind deconvolution with sharp-edge prediction, where we predict a sharp image from the output of the non-blind deconvolution step.

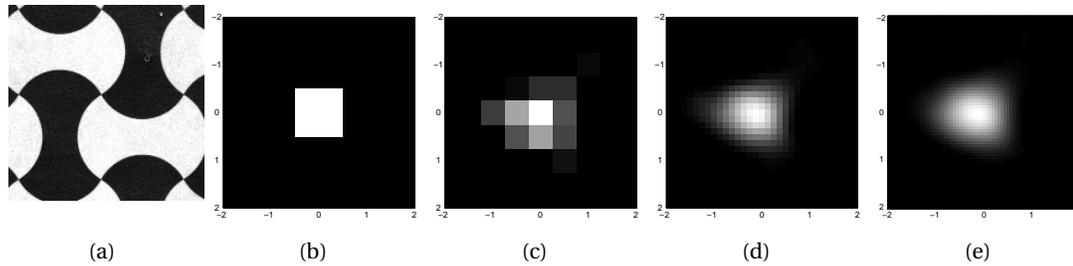


Figure 3.11: Sub-Pixel PSFs. (a) Cropped section of a sharp image of our grid, (b) PSF (green channel only) at the Bayer resolution (1x), (c) 2x, (d) 8x, and (e) 16x sub-pixel sampling. The sub-pixel PSFs show blur resulting from a combination of diffraction, lens imperfections, and sensor anti-aliasing and sampling.



Figure 3.12: Blind Chromatic Aberration. (a) Recovered spatially varying PSFs for red, green, and blue shown as a color image. PSFs are only computed where there are enough edges observed. (b) The original image, (c) after radial correction and deconvolution the aberrations are significantly reduced, and (d–e) zoomed-in versions and intensity profiles for (b–c).

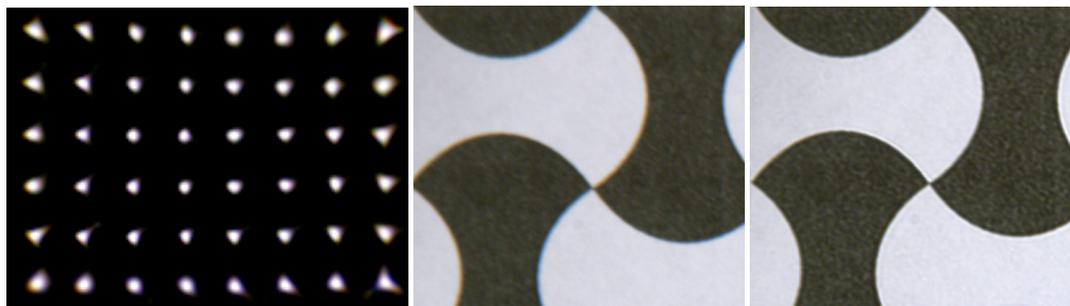


Figure 3.13: Chromatic Aberration. (left) The recovered spatially varying PSFs for red, green, and blue shown as a color image. The red and blue fringing is reflected in the PSF image and the PSFs are larger towards the edge of the image and spread along the direction orthogonal to the optical axis. (middle) Zoom-in on the input image. (right) After radial correction and deconvolution the aberrations are significantly reduced.

We compare our results to those by Shan *et al.* [2008] and show that we recover sharp images and kernels that are similar to their results.

As each sharp edge profile gives information about a radial slice of the PSF, it is necessary

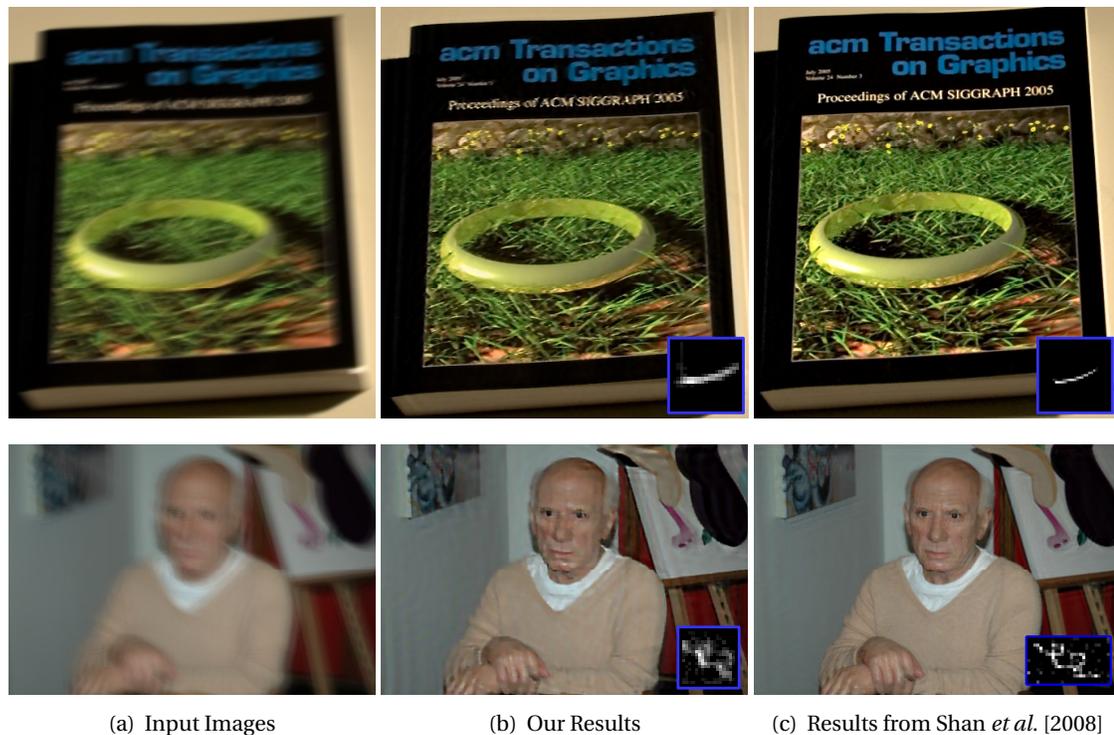


Figure 3.14: Iterative Blind Deconvolution with Sharp Edge Prediction. We perform multi-scale iterative blind deconvolution where we predict a sharp image from the output of the non-blind deconvolution step. Compared to Shan *et al.*'s [2008] results we recover sharp images and kernels that are similar to their results. (a) Input images, (b) our recovered sharp image and kernel, and (c) Shan *et al.*'s recovered sharp images – for the top image the kernel shown is the ground-truth and not estimated by their method, while for the bottom image Shan *et al.* estimate the kernel and the sharp image.

for an image, or image window, to have edges (or at least high-frequency content) at most orientations. If some orientations are lacking, our regularization terms can compensate; however, there is a breaking point, and at times there may not be enough edge information to properly compute a PSF. In these cases, a low parameter kernel model may be more appropriate, but our sharp image prediction could still be used to improve more traditional parametric kernel estimation procedures. We also plan to try using robust least squares to compensate for erroneous edge detections or profile fits.

Lastly, we would like to characterize more lenses and cameras. We would like to build a database that the vision and photography community could contribute to by using our pattern and code to take their own measurements.

3.8 Acknowledgements

This chapter is, in part, a reproduction of published material: Neel Joshi, Richard Szeliski, and David J. Kriegman, “PSF Estimation using Sharp Edge Prediction”, Proceedings of IEEE CVPR 2008. I was the primary investigator and author of this paper. This work was partially completed while I was an intern at Microsoft Research.

Image Enhancement using Color Statistics

“It is not the form that dictates the color, but the color that brings out the form.”

—Hans Hofmann

In the previous chapter, we presented methods to recover the blur function from a photograph. Even when the blur function is known, deblurring an image is still quite difficult due to information loss during blurring and due to the presence of noise, which can be significant. At times, blur can be eliminated during capture, yet noise remains. Thus the task of recovering an uncorrupted image, free of blur or noise, from a single observed image is difficult in its own right.

Image deconvolution in the presence of noise is an inherently ill-posed problem. The observed blurred image only provides a partial constraint on the solution—there exists many “sharp” images that when convolved with some blur kernel can match the observed blurred and noisy image. Image denoising alone is similarly ill-posed as there is an ambiguity between the high-frequencies of the unobserved noise-free image and those of the noise. Thus, the central challenge in deconvolution and denoising is to develop methods to disambiguate solutions and bias the processes towards more likely results given some prior information. Exploring and developing appropriate priors is the central part of this work.

A central part of deconvolution is properly handling the noise that is often present in blurred images. As a result, denoising can be considered a sub-problem of deblurring, and deconvolution methods can thus be used purely for denoising by considering the blurring kernel to be a delta function. An existing example of this is the Wiener filter for deconvolution and

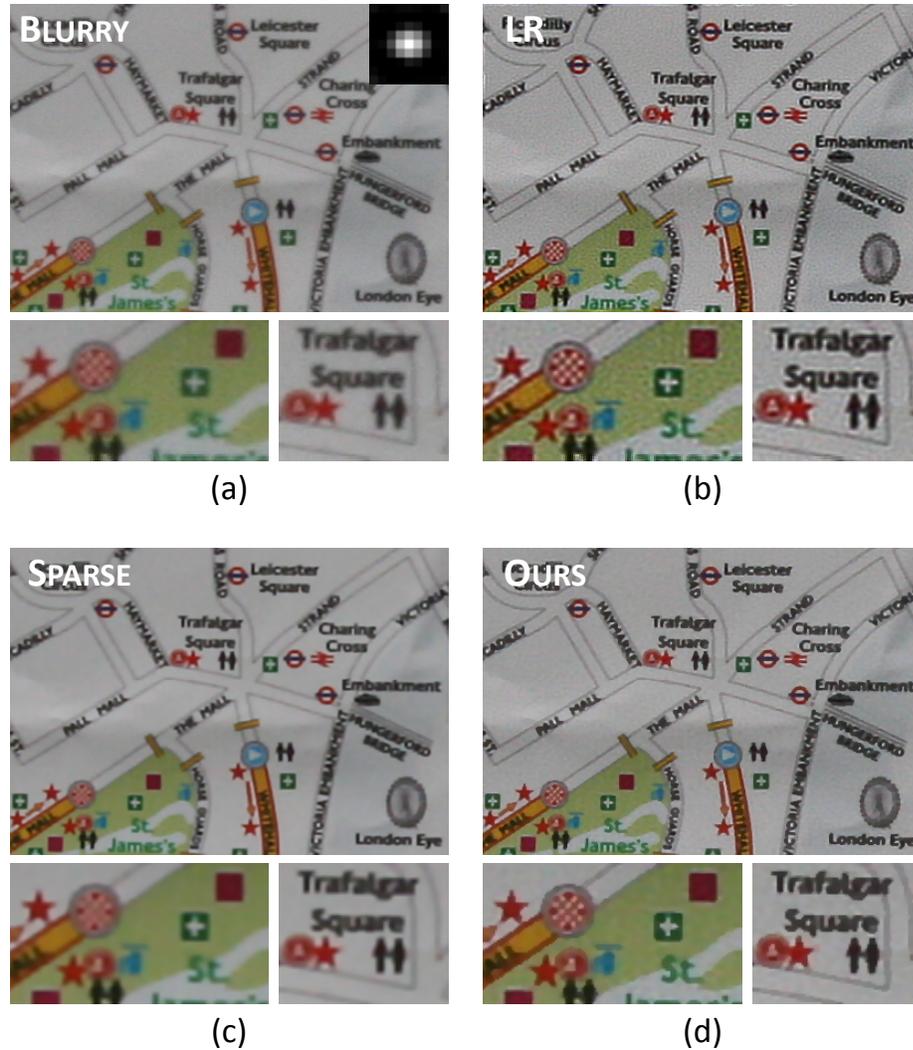


Figure 4.1: Deblurring with a two color prior. (a) Input blurry image (the 9×9 PSF is displayed (enlarged by $4 \times$) in the top right corner, the noise level is estimated per-pixel with the median $\sigma = 0.0137$), (b) image deblurred using the Lucy-Richardson algorithm, (c) using a sparse gradient prior, and (d) using our method ($\lambda_2 = 1.4$). Result (b) is sharp but noisy and has ringing artifacts; the sparse prior result (c) is overly-smoothed; our result (d) has the sharpness of Lucy-Richardson with significantly reduced noise and ringing (most apparent in the red/white checkerboard pattern and green cross in the bottom left of each sub-figure.)

denoising; however, there is little other work that we are aware of that has analyzed or presented deconvolution as a unified framework for deblurring and denoising. We treat denoising as a sub-problem of deconvolution and present a non-blind deconvolution algorithm that can be used for both applications. We further illustrate that image up-sampling is a sub-problem of deconvolution and use our framework to address this problem.

4.1 Related Work

Image deblurring and denoising have received a lot of attention in the computer graphics and vision communities. Chapter 2 presents a discussion of these areas. To briefly review, the most basic approaches for denoising are Gaussian and median filtering, which have a tendency to over-smooth edges and remove image detail. More sophisticated approaches use the properties of natural image statistics to derive methods that enhance large intensity edges and suppress lower intensity edges. This property has been used by wavelet based methods [Simoncelli and Adelson, 1996; Portilla et al., 2003], anisotropic diffusion [Perona and Malik, 1990], Bilateral filtering [Tomasi and Manduchi, 1998], and Field of Experts models [Roth and Black, 2005]. Our method shares some similarities with these works in that we consider natural image statistics in the form of a prior on the distribution of image gradients [Levin, 2007], but we go beyond this and additionally incorporate a prior derived from local color statistics in the form of a local two color model. This allows us to avoid some of the over-smoothing that can occur with gradient-based methods alone.

The denoising aspect of our work is most similar to that of Liu *et al.* [2006] in that we both use a local-linear color model. Where our methods differ is that we build a color model per-pixel, while Liu *et al.* segment the image first and then build the model per segment. Liu *et al.* additionally address noise-level estimation, and we use their method for estimating the spatially varying noise level in our work. A further distinction between our work and previous work is that we address denoising in the larger context of image deconvolution, while most denoising work considers this problem in isolation.

To address image deblurring, some researchers have modified the image capture process [Ben-Ezra and Nayar, 2004; Raskar et al., 2006] or used multiple images [Basclé et al., 1996; Rav-Acha and Peleg, 2005; Yuan et al., 2007] to aid in deblurring. Image blur due to limited resolution has led to the development of up-sampling algorithms [Freeman et al., 2002; Fattal, 2007]. Determining the blur kernel from a single image, which is a critical sub-problem for deblurring natural images, has also been studied significantly, including our contributions in Chapter 3 and those of numerous other researchers [Fergus et al., 2006; Levin, 2007]. One area that has received less attention, yet is critical for the above techniques to succeed, is that of non-blind

deconvolution.

Non-blind image deconvolution is the process of recovering a sharp image from an input image corrupted by blurring and noise, where the blurring is due to convolution with a known kernel and the noise level is known. Most deblurring approaches rely on decades old deconvolution techniques such as the Lucy-Richardson algorithm [Richardson, 1972; Lucy, 1974], Wiener deconvolution, and least-squares deconvolution. Many of these algorithms were developed for applications where the images are quite different than those taken by an everyday photographer, e.g., Lucy-Richardson assumes Poisson noise statistics, which are more applicable to very low-light conditions. Consequently, these methods are not always well suited to the desired task and often generate unwanted artifacts such as ringing. One approach that attempts to overcome the shortcomings of these traditional techniques is that of Levin *et al.* [2007] in which they incorporate image priors derived from natural image statistics. Other methods have explored the use of graph cuts to reduce over-smoothing [Raj and Zabih, 2005], deconvolution using multiple blurs [Harikumar and Bresler, 1999], and energy minimization functions using wavelets for deconvolution [de Rivaz and Kingsbury, 2001].

The work of Levin *et al.* [2007] was a significant improvement in non-blind deconvolution. Their work uses priors based on assumptions about the edge content of images. Specifically, the authors assume that images are piecewise smooth and thus the gradient distribution of an image is zero-peaked with high kurtosis. They enforced this property using a hyper-Laplacian prior on image gradients during deconvolution. In our work, we show that while this assumption holds for many images, it has a tendency to generate overly smoothed results.

We propose an auxiliary approach using priors derived from image color statistics. We model an image as a per-pixel linear combination of two color layers, where these layer colors are expected to vary more slowly than the image itself. Edges and textures in an image are accounted for by blending the colors of the two layers. This two-color assumption is often reliable, as there are many situations where images themselves are not smooth, yet the underlying parameters such as lighting and material color vary much more slowly. Unlike previous methods, which compute priors based on gradient magnitudes, our approach places priors on the alpha value used for blending between the two colors. As a result, our two-color model can act as a sparse

edge prior that is independent of gradient magnitude. Our results show that the method produces qualitatively and quantitatively better results with sharper edges, less color bleeding, and less multi-colored noise.

4.2 Overview

We model a blurred, noisy image as the convolution of a latent sharp image with a known shift-invariant kernel plus additive white Gaussian noise, whose result is potentially down-sampled. Specifically, blur formation is modeled as:

$$B = D(I \otimes K) + N, \quad (4.1)$$

where K is the blur kernel, $N \sim \mathcal{N}(0, \sigma^2)$ is the noise, and σ^2 potentially varies spatially. $D(I)$ down-samples an image by point-sampling $I_L(m, n) = I(sm, sn)$ at a sampling rate s for integer pixel coordinates (m, n)

Our goal is to recover the unobserved sharp image I from only the observed blurred input image B given the kernel K . We use the method of Liu *et al.* [2006] to estimate the spatially-varying noise level σ^2 .

We formulate the image deconvolution problem using a Bayesian framework and find the most likely estimate of the sharp image I , given the observed blurred image B , the blur kernel K , and the recovered noise level σ^2 using a *maximum a posteriori* (MAP) technique.

We can express this as a maximization over the probability distribution of the posterior using Bayes' rule and express the result as minimization of a sum of negative log likelihoods:

$$P(I|B) = P(B|I)P(I)/P(B) \quad (4.2)$$

$$\operatorname{argmax}_I P(I|B) = \operatorname{argmin}_I [L(B|I) + L(I)]. \quad (4.3)$$

The problem of deconvolution is now reduced to defining and minimizing the negative log likelihood terms. Given the blur formation model (Equation 4.1), the "data" negative log likelihood

is:

$$L(B|I) = \|B - I \otimes K\|^2 / \sigma^2. \quad (4.4)$$

We incorporate the down-sampling function in Equation 4.1 by modifying the likelihood term to be $\|B - D(I_H \otimes K_H)\|^2$, where the *super-resolved* latent image and kernel are I_H and K_H , respectively, and $B_H = I_H \otimes K_H$. $D(I)$ is formed as a simple point-sampling matrix such that $B(m, n) = B_H(sm, sn)$ for a sampling rate s .

To perform denoising alone we set the kernel K in Equation 4.4 to a delta function, which reduces the data negative log likelihood to:

$$L(B|I) = \|B - I\|^2 / \sigma^2. \quad (4.5)$$

The form of the remaining negative log likelihood term, $L(I)$, in Equation 4.2 depends on the image prior that is used. Defining this term is the focus of this work.

4.3 Gradient Priors

In image deconvolution, the data likelihood is inherently ambiguous, i.e., there are many “sharp” images that when blurred match the observed blurred image. The range of ambiguity increases with the amount of blur, and image noise further complicates the issue. The role of the image prior is to disambiguate the set of possible solutions and to reduce over-fitting to the noise. A common approach is to assume that the image is smooth or piecewise smooth, resulting in priors on image gradients. In the following section, we discuss the limitations of gradient priors and present a novel prior derived from image colors statistics.

4.3.1 Gaussian and Sparse Gradient Priors

Gradient priors are typically enforced between neighboring pixels in an image. These interactions can be modeled using a Markov Random Field (MRF) in which the value of an individual pixel is conditionally dependent on the pixel values in a local neighborhood. One possible prior on the gradients is a smoothness prior in which large image gradients are penalized.

Thus, neighboring pixels are favored to have values similar to their neighbors:

$$L(I) = \lambda \|\nabla I\|^2. \quad (4.6)$$

The value ∇I indicates the spatial gradients of the image, and λ is a regularization parameter that controls the weight of the smoothness penalty. While this prior does disambiguate the solution, it can result in an overly-smooth solution and introduce ringing artifacts [Levin et al., 2007]. This occurs as a result of the quadratic penalty term, which enforces a Gaussian distribution on gradients. Unfortunately, “natural” images have a decidedly non-Gaussian gradient distribution.

Levin *et al.* [2007] address this by modifying the gradient penalty to enforce a hyper-Laplacian distribution on gradients:

$$L(I) = \lambda \|\nabla I\|^q, \quad (4.7)$$

where $q < 1$ makes the distribution hyper-Laplacian. Levin *et al.* use $q = 0.8$. This “sparse” gradient prior better models the zero-peaked and heavy tailed gradient distributions seen in natural images. As the penalty function is no longer a quadratic, the minimization is performed using iterative re-weighted least-squares [Stewart, 1999].

4.3.2 Limitations of Gradient Priors

Deconvolution using a sparse gradient prior is a significant step towards producing more pleasing results, as it reduces ringing artifacts and noise relative to more traditional techniques. However, this prior has some limitations. While it biases the deconvolution to produce images with a hyper-Laplacian distribution on gradients, this prior is implemented as a penalty on gradient magnitudes. Thus, it is essentially a “smoothness prior” with a robust penalty function. Using this function, larger gradients still incur larger penalties. This results in a preference for finding the lowest intensity edges that are consistent with the observed blurred image. This is particularly an issue with “bar” type edges and high-frequency texture, as illustrated in Figure 4.2.

The second limitation of the sparse gradient prior arises in the presence of significant image noise. The implication of the Levin *et al.*'s [2007] sub-linear gradient penalty function is that a single large gradient is preferred over many small gradients when accounting for intensity

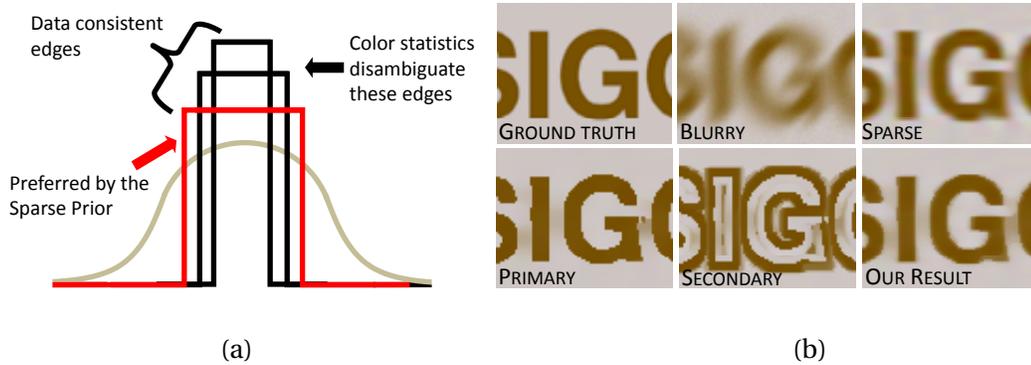


Figure 4.2: (a) There are many sharp edges that can blur to match the observed blurred (and potentially noisy) edge (shown in tan). The sparse prior always prefers the smallest intensity gradient that is consistent with the observation (shown in red). We use local color statistics to disambiguate the potential sharp edges, i.e., our method picks the edge that is more likely given the dominant primary and secondary colors in the pixel's neighborhood. (b) The thin gray areas between the tan letters are deconvolved to a mid-level tan/gray when using the sparse prior. Our results produce noticeably sharper results. High-frequency texture can be thought of as a series of bar edges and exhibits similar artifacts.

variations. This can result in the preservation and sharpening of the noise. As illustrated in Figure 4.3, the presence of high-frequency noise that varies on a per-pixel level produces mid-frequency texture patterns, Figure 4.3(top right), that can be more objectionable than the original noise. The noise may be removed by increasing the weight of the sparse gradient prior, but this produces an over-smoothed result (Figure 4.3(bottom left)).

4.4 Color Priors

The sparse gradient model is a reasonable model of images; however, as we have shown, it is not always sufficient and can introduce unwanted artifacts. This is partially caused by the use of a generic gradient prior that is not fit to specific images. In this chapter, we use a color model built from local colors statistics of the sharp latent image itself as an additional constraint.

The majority of photographs, whether they contain nature, people, or man-made objects are globally composed of a relatively limited set of colors. This property is well studied and has been exploited across numerous imaging applications. For instance, image compression uses this observation to create a small set of indexed color palettes. Recently, researchers have further noted that images can locally be described as a mixture of as few as two colors for use in alpha-matting [Levin et al., 2006], image denoising [Liu et al., 2006], and Bayer demosaicing [Bennett

et al., 2006].

4.4.1 The Two-Color Model

The two-color model states that any pixel color can be represented as a linear combination of two colors, where these colors are piecewise smooth and can be derived from local properties:

$$I = \alpha P + (1 - \alpha)S. \quad (4.8)$$

P and S are the respective *primary* and *secondary* colors and α is the linear mixing parameter. For notational convenience, the primary color P_i is always assigned to the color that lies closest to the pixel i 's color I_i . Some pixels may only be described by a single color, in which case $P_i = S_i$.

The two-color model has several benefits when used to provide an image prior for deconvolution. First, given the two colors for a pixel, the space of unknowns is reduced from three dimensions (RGB) to one (α). The second benefit is that the α parameter provides an alternative for parameterizing edges, where the edge sharpness is decoupled from edge intensity—a single pixel transition in α from 1 to 0 indicates a step edge (a single step from primary to secondary) regardless of the intensity of the edge. Thus, we can control an edge's strength with a prior on α while maintaining local smoothness using a separate prior on P and S .

A significant benefit of the two-color model is the ability to capture local color statistics. We observe that local color statistics can provide a strong constraint during deconvolution. These constraints help reduce over-smoothing around “bar edges” and high-frequency texture as shown in Figure 4.2b. In contrast with a gradient prior, which prefers the lowest intensity edges that are consistent with the observed blurred image, a two-color model can result in higher intensity edges if such edges are more consistent with local color statistics.

The two-color model is built from local image statistics and estimates two dominant colors in a neighborhood around each pixel. We estimate primary and secondary colors for each pixel by using a robust EM clustering algorithm in a local neighborhood around each pixel that is a modified version of the approach used by Bennett *et al.* [2006]. The model consists of 2 3D Gaussians to model the primary and secondary clusters and a 1D oriented Gaussian to capture



Figure 4.3: Over-smoothing and noise texturing with the sparse gradient prior: (top left) blurred, high-noise image (the PSF is 31×31 pixels and $\sigma = 0.05$); (top right and bottom left) deconvolution using a smaller and larger weight on the sparse gradient penalty; (bottom right) the ground-truth. To remove the “textured” noise effect in the second image, it is necessary to increase the gradient penalty, which over-smooths the image.

the likelihood of colors lying between the primary and secondary. We also incorporate a small uniform distribution to model outliers. The primary and secondary Gaussians are initialized by first performing 10 iterations of k-means clustering (with $k=2$). In the maximization step, when each Gaussian’s standard deviation is recomputed, we clamp the minimum for the primary and secondary Gaussians to be the noise’s standard deviation σ . As a result, after several iterations the Gaussians will merge if the standard deviation is less than the noise’s standard deviation. In this case, we consider the pixel to be modeled by one color, otherwise it is marked as being a two color pixel. A binary variable indicating one vs. two colors is stored for each pixel. We use a 5×5 window around each image pixel and perform 10 iterations of EM clustering.

4.4.2 Using the Two-Color Model for Deconvolution

The two color model provides a significant constraint for deblurring; there are two ways such a model can be used for deconvolution. The first is to use the model as a hard constraint,

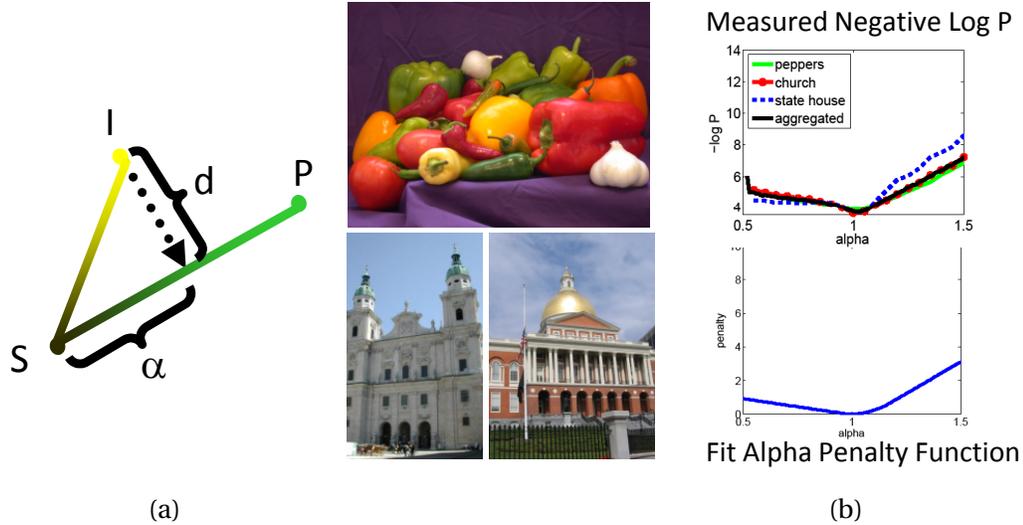


Figure 4.4: Using the two-color prior: (a) One prior is on the perpendicular distance from a sharp pixel's color value to the (3D) line defined by the two colors, and a second prior is on the distribution of α , i.e. the normalized distance along the line; (b) the measured negative log probability distributions of alpha values measured for the shown ground-truth sharp images and our piecewise linear and quadratic penalty function fit to the shape of the distribution of the aggregated data. The function is symmetric about $\alpha = 0.5$ (not shown).

where the sharp image I must always be a linear combination of the primary and secondary colors P and S . The second is to use a soft-constraint to encourage I to lie on the line connecting P and S in RGB space (Figure 4.4a). We believe that the hard-constraint is too limiting and therefore use the soft-constraint.

Our image negative log-likelihood term is defined as:

$$L(I|P, S) = \lambda_1 \|I - [\alpha P + (1 - \alpha)S]\|^2 + \lambda_2 \rho(\alpha) + \lambda_3 \|\nabla I\|^{0.8}. \quad (4.9)$$

The first likelihood term minimizes the distance between the recovered intensity I and the line defining the space of the two color model (Figure 4.4a). In the above equation, α is not a free variable and is computed as:

$$\alpha = \left(\frac{(P - S)}{(P - S)^T (P - S)} \right)^T (I - S). \quad (4.10)$$

For the “one-color” model, in which $P = S$, we do not use the two color model and the negative



Figure 4.5: Primary, secondary, and alpha maps for the peppers image in Figure 4.4.

log likelihood falls back to using the sparse prior only:

$$L(I) = L(I|P) = \lambda_3 \|\nabla I\|^{0.8}. \quad (4.11)$$

The second likelihood term, $\rho(\alpha)$, allows us to enforce a prior on the distribution of alpha values that are a function of the normalized distances from I to P and S . The shape of this prior plays a crucial role in enforcing sharpness during deconvolution. We bias the alpha distribution to be peaked around 0 and 1, which enforces sharp transitions between colors by minimizing the number of pixels with partial α values.

To confirm this expectation and to recover the exact shape for the alpha distribution, we measured the distribution of α values for several images. We computed P and S for a set of *sharp* images and then fit a penalty function to the shape of the negative log likelihood of the measured alpha distributions. Our measured distributions and recovered penalty function $\rho(\alpha)$ are shown Figure 4.4b. As expected, the penalty prefers that α be close to 0 or 1.

Note that in Equation 4.9, we retain the sparse prior of Levin *et al.*, i.e., we find an image with sparse gradients that is most consistent with the two-color model. For both equations, we set $\lambda_1 = 10$, $\lambda_3 = 0.5$, and λ_2 in the range of $[0.8, 1.5]$.

4.5 Solving for the Final Image

In the previous sections, we derived negative log-likelihood terms that when minimized allow us to recover a latent image using a prior derived from a two-color model. Unfortunately, minimizing this error function in one step is not straightforward due to the interdependence

of I , P , and S . Fortunately, the problem can be decomposed into two more easily solvable sub-problems.

For each of the operations we perform: deblurring, denoising, up-sampling, and demosaicing we perform an EM-style minimization consisting of the following two steps:

- Estimate P_i and S_i from I_{i-1} by local EM clustering (using an $N \times N$ window around each pixel).
- Deconvolve the blurred image using P_i and S_i to get I_i by minimizing Equation 4.2 with Equation 4.9 as the prior on I_i using I_{i-1} as an initial guess.

Deblurring and denoising with the two-color model estimate the initial image color model P_1 and S_1 in slightly different ways.

Deblurring: The minimization is initialized by computing a deconvolution of the image B to get I_0 using the sparse gradient prior alone. We then estimate P_1 and S_1 from this initial deconvolution. After this initialization, we reduce the amount of regularization used by the sparse prior for this initialization to preserve sharpness ($\lambda_3 = 0.25$ for this initialization step and $\lambda_3 = 0.5$ for the subsequent steps). The noise artifacts that result from the reduced regularization are later suppressed when the color model is built.

Denoising: The denoising case is simpler as the color model can be built directly from the input image. Thus the input image is I_0 and we estimate P_1 and S_1 from this image.

Since our α penalty function has two minima—one at 0 and the other at 1—allowing for colors to be close to the primary or secondary color, the deconvolution step listed above is non-linear. We solve this non-linear optimization using Matlab's `fminunc` function, which performs a Quasi-Newton's algorithm.

We have found that two to three EM-style iterations (between color estimation and deconvolution) are sufficient for convergence. In most cases the perceptual difference after the second iteration is minimal.



Figure 4.6: Deblurring Text: Blurred, noisy image (the PSF is 31×31 pixels and $\sigma = 0.01$), deconvolution with Lucy-Richardson, the sparse prior, our result using the two-color prior, and the groundtruth for two images. Note how the two “G’s” in the first set of images are more distinctly separate when using our method. In these results $\lambda_2 = 1.5$.



Figure 4.7: Peppers: Blurred, noisy image (the PSF is 31×31 pixels and $\sigma = 0.01$), deconvolution with Lucy-Richardson, the sparse prior, our result using the two-color prior, and the groundtruth for two images. The results using the two-color model appear sharper with few artifacts due to noise. In these results $\lambda_2 = 0.8$.

4.6 Results

4.6.1 Deblurring

To validate our deconvolution algorithm, we tested our method on several images, including both synthetic cases, where the blur kernel and ground-truth sharp image are known, as well as several real images, where the blur kernel was estimated using previously developed methods [Fergus et al., 2006; Yuan et al., 2007] and our methods presented in Chapter 3. For each result, we compare our method to Lucy-Richardson and deconvolution using the sparse-prior alone. For the sparse prior, we used the code available online by Levin *et al.* [2007].

Figure 4.6 shows results for the SIGGRAPH logo and text synthetically blurred with a

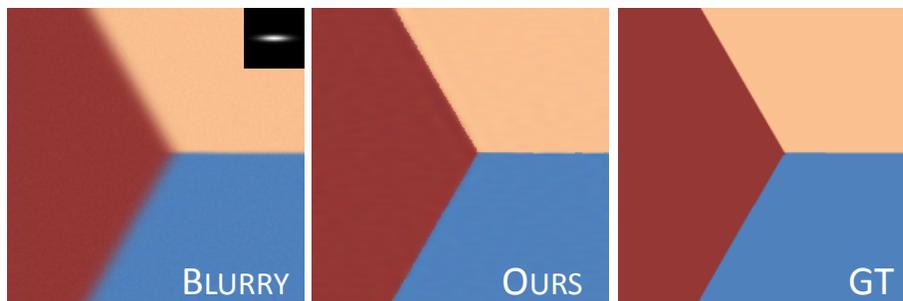


Figure 4.8: Three Colors Meeting at a Point: Even when the two color model does not strictly hold within a neighborhood, the perceptual artifacts in this failure case are minimal.

horizontally elongated Gaussian kernel. These images exhibit numerous “bar edge” type features between the letters. Our method separates and sharpens the letters much more than the Lucy-Richardson and the sparse prior alone. Figure 4.7 shows a similar result for a more natural image.

Figure 4.8 shows an example for a failure case for the two color model where three colors meet at a point. Even though the two color model does not strictly hold here, the perceptual artifacts are minimal since the color model is used as a soft-constraint.

In Figure 4.1, we show a result using a map image and corresponding PSF recovered using our method presented in Chapter 3. Our result has the sharpness of the Lucy-Richardson result and is not overly smoothed, as in the result when using the sparse-prior alone. If the regularization weight for using the sparse prior alone is reduced, more textured noise appears creating an effect similar to that in Figure 4.3. Our result has minimal noise artifacts and no ringing.

In Figure 4.9, we show results for three real images. The first two blurry images and blur kernels are from the work by Yuan *et al.* [2007] and were provided to us by the authors. In their work, the authors obtain accurate PSFs for blurry images using a sharp, noisy image and a blurry image. We have used their PSFs for deblurring the blurry images alone. The third result is using an image and PSF from the work of Fergus *et al.* [2006] that is publicly available. As in the previous results, in comparison with the other methods, our results are sharper.

In Figure 4.12, we show a result where we have deconvolved the image in Figure 4.1 on a $2\times$ up-sampled grid, which our formulation naturally allows for (as discussed in Section 4.2). For comparison, we show the $1\times$ result up-sampled and the $2\times$ down-sampled using bi-cubic

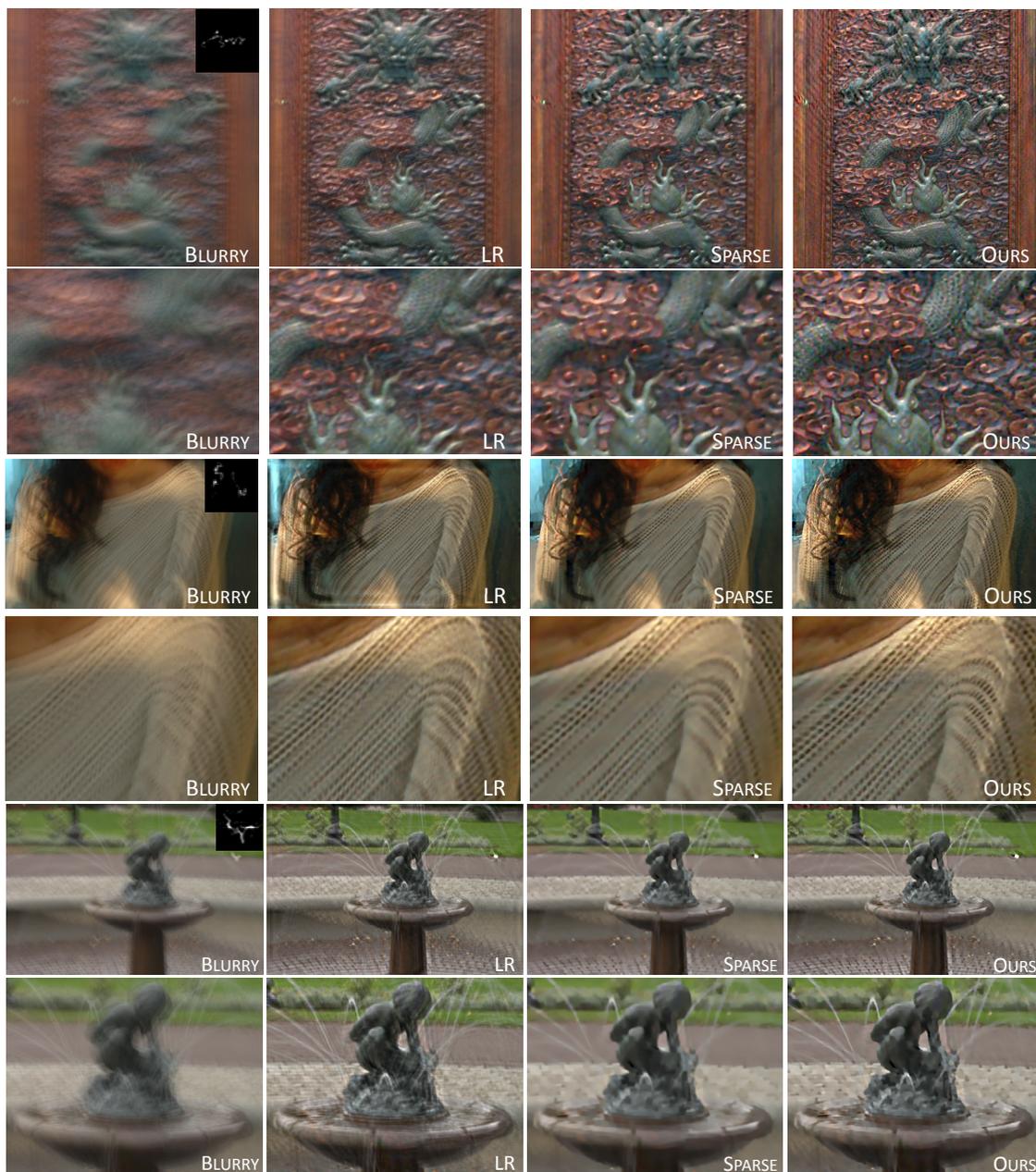


Figure 4.9: A dragon (the PSF is 60x60 pixels and the noise level is estimated per-pixel the median $\sigma = 0.004$) and sweater (the PSF is 30x30 pixels and the median estimated $\sigma = 0.0127$) from Yuan *et al.* and a fountain (the PSF is 39x39 pixels and the noise level is estimated per-pixel the median $\sigma = 0.0104$) from Fergus *et al.* For each result we show the blurred, noisy image, deconvolution with Lucy-Richardson, the sparse prior, and our result using the two-color prior for both images. We deblur the blurry images from Yuan *et al.* and Fergus *et al.* using their recovered PSFs. For the dragon and sweater $\lambda_2 = 1.2$ and for the fountain $\lambda_2 = 0.8$.

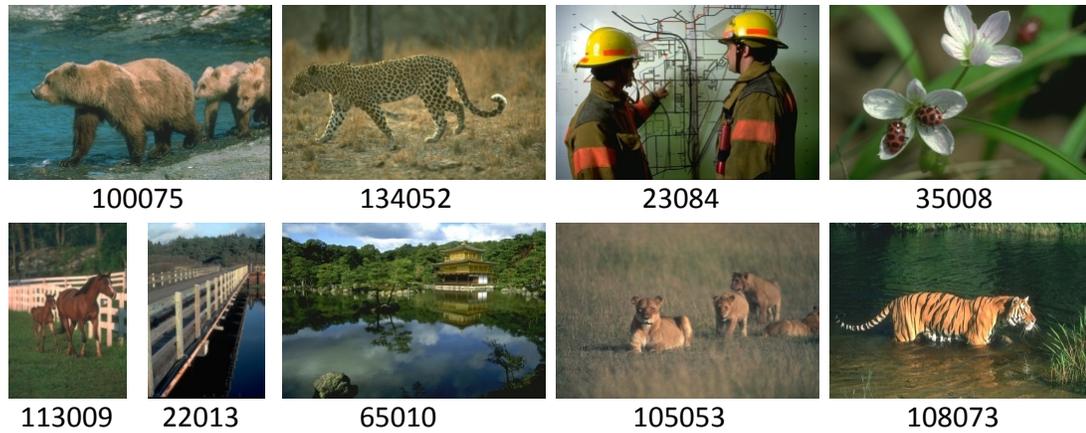


Figure 4.10: Images from the Berkeley Image Database

interpolation and decimation. For both comparisons, the $2\times$ up-sampled grid produces crisper results.

4.6.2 Denoising

To test the performance of our algorithm for denoising, we added 5% and 10% additive gaussian noise to several images from the Berkeley Image Database, shown in Figure 4.10, and ran our algorithm on the resulting images.

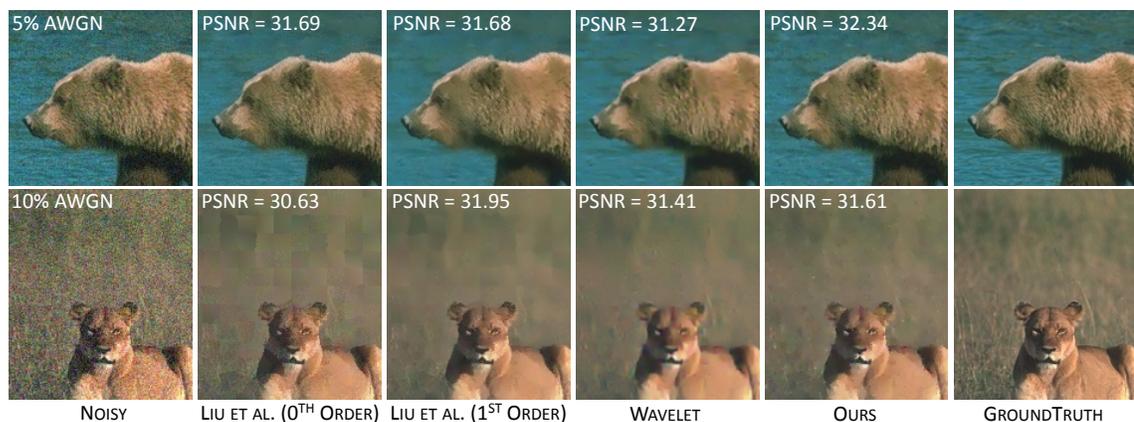


Figure 4.11: Denoising: Visual comparison of our denoising results with previous work. The fur in the bear is sharper in our result. For the lion image, our result has the sharpness of the Liu *et al.* results in the face region, but does not have the blocking artifacts in the low-frequency background region. For all images $\lambda_2 = 0.8$.

Table 4.1 shows a comparison of the PSNR values for our denoising results compared to the results of using Liu *et al.*'s [2006] and Portilla *et al.*'s [2003] methods. Figure 4.11 shows a visual comparison for two images. Figure 4.1 also compares our work to denoising using the sparse prior alone. (Note that while denoising using the sparse prior alone is the same as Levin *et al.* [2007]'s deconvolution with the data term in Equation 4.5, Levin *et al.* never specifically address denoising.) Our PSNR values are consistently higher than those of Portilla *et al.*'s method, Liu *et al.*'s 0th order denoising, and the sparse prior in the higher noise case. Liu *et al.*'s 1st order method results in slightly higher PSNR values than our method for the 10% noise images; however, Liu *et al.*'s results show significant blocking artifacts which our method does not suffer from, as shown in Figure 4.11.

4.6.3 Up-Sampling

Given our formulation, it is also possible to do more traditional up-sampling. We up-sample a low-resolution image by deconvolving it on an up-sampled grid where the PSF is a down-sampling anti-aliasing filter (we use a 7-tap binomial filter).

We use the same EM-style iterative optimization for denoising and deblurring; however, the minimization is initialized by computing a bicubic up-sampling of the low resolution image to estimate the initial guess I_0 . We then estimate P_1 and S_1 from this image.

Table 4.1: Denoising PSNR Comparisons: Our PSNR value are consistently higher than those of Portilla *et al.*'s method, Liu *et al.*'s 0th order denoising, and the sparse prior in the higher noise case. Liu *et al.*'s 1st order method results in slightly higher PSNR values than our method for the 10% noise images; however, Liu *et al.*'s results show significant blocking artifacts which our method does not suffer from as shown in Figure 4.11. For all of our results $\lambda_2 = 0.8$.

File name	$\sigma = 5\%$					$\sigma = 10\%$				
	0th	1st	Wavelet	Sparse	Ours	0th	1st	Wavelet	Sparse	Ours
22013	32.17	32.33	31.31	32.50	32.56	27.14	28.84	27.12	27.53	27.98
23084	32.04	32.64	32.14	33.01	33.10	27.04	29.23	27.24	27.20	27.93
35008	34.84	35.97	35.74	35.88	35.69	30.24	33.27	31.23	32.16	32.30
65010	31.99	32.18	30.95	32.26	32.42	27.18	28.41	26.73	27.07	27.60
100075	31.69	31.68	31.27	32.36	32.34	28.14	28.96	28.31	28.51	28.77
105053	33.77	34.02	34.01	34.79	34.57	30.63	31.95	31.41	31.73	31.61
108073	31.94	31.98	31.48	32.49	32.65	28.33	29.21	27.94	27.94	28.70
113009	32.31	32.61	32.89	33.62	33.56	28.89	30.19	29.91	29.87	30.07
134052	32.58	32.88	32.09	33.01	33.02	28.61	29.55	28.20	28.15	28.95



Figure 4.12: Deconvolution on an up-sampled grid . We show our method run on a $1\times$ and $2\times$ grid. For comparisons, we show the $1\times$ result up-sampled and the $2\times$ down-sampled both as a post-process using bi-cubic interpolation. The deconvolution directly on the $2\times$ up-sampled grid is sharper than the up-sampled $1\times$. The down-sampled $2\times$ is crisper than the solution at the native $1\times$ resolution.

The application of our method to up-sampling shares similarities with the work of Fattal [2007] and Dai *et al.* [2007], in that we all consider alpha priors. In Figure 4.13, we show two $4x$ up-sampling results using data from Fattal. Our result is significantly sharper than the result of bi-cubic interpolation and is of similar quality to Fattal’s result with some qualitative differences.

4.6.4 Demosaicing

It is also possible to modify our formulation to perform Bayer demosaicing similar to the method by Bennett *et al.* [2006]. The error function for demosaicing is slightly different than for deconvolution and deblurring. For demosaicing to goal is to interpolate the missing color samples, but to retain the sampled values. Thus we add a spatially varying mask that ensures this.

The data negative log likelihood for demosaicing is thus:

$$L(B|I) = M * ||B - I||^2. \quad (4.12)$$

The mask M is one for where the Bayer pattern sampled a particular color channel and is zero otherwise.

The image prior is modified in a similar way where the masking function is inverted. Thus the mask M is zero for where a the Bayer pattern sampled a particular color channel and is one otherwise:

$$L(I|P, S) = M * (\lambda_1 ||I - [\alpha P + (1 - \alpha)S]||^2 + \lambda_2 \rho(\alpha) + \lambda_3 ||\nabla I||^{0.8}), \quad (4.13)$$

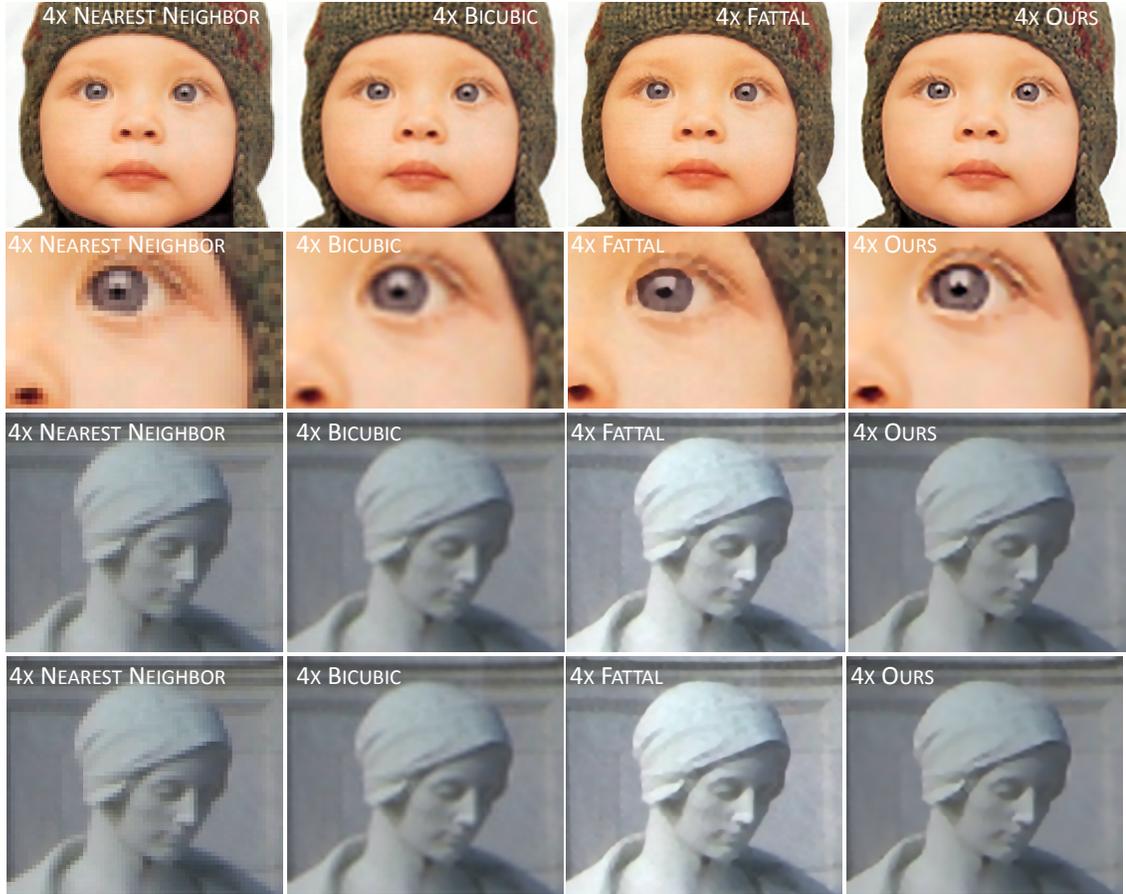


Figure 4.13: Up-sampling low-resolution images. Our formulation also allows us to perform more traditional up-sampling of low-resolution images. We show our result for $4\times$ up-sampling using images from the work of Fattal [2007]. Our result is significantly sharper than bi-cubic interpolation and is similar to Fattal’s result.

and

$$L(I) = L(I|P) = M * (\lambda_3 \|\nabla I\|^{0.8}) \quad (4.14)$$

for the one “one-color” regions.

We use the same EM-style iterative optimization for denoising and deblurring; however, the minimization is initialized by computing a demosaicing of the bayer image to get I_0 . We then estimate P_1 and S_1 from this image. We have tried various demosaicing algorithms for the initial estimation and found the method by Li [2005] to work quite well.

To test our formulation we have used the Kodak true color image dataset commonly used for evaluating demosaicing algorithms. These images are scanned prints that have true RGB



Figure 4.14: Kodak true color images used for demosaicing experiments.

color per-pixel. In table 4.2 we show PSNR comparisons of Li [2005] output and our result seeded with Li's method as the initial estimate. Our method shows improvements for the green channel. The red and blue channels do not show a consistent improvement. We believe this is due to the red and blue channels being too sparsely sampled (by a factor of two less than the green) to estimate the color model in a way that removes demosaicing artifacts, i.e., the artifacts in initial demosaicing are captured by our EM clustering and propagated through as part of our color prior.

We encourage the reader to visit <http://graphics.ucsd.edu/~neel/dissertation/> for full-resolution comparisons for all of the results discussed here and a couple of additional results.

4.7 Alpha Distribution Measurements

To better understand and model alpha distributions we have conducted a larger study of alpha distributions for a larger, broader range of images. Just as discussed in Section 4.4.2 we measured α distributions for a set of natural images, 300 images from the Berkeley Image Database, and 160 “text” images from the ICDAR (International Conference on Document Analysis and Recognition) 2003 dataset; images from both sets are shown in Figure 4.15.

These plots of negative log likelihoods of the alpha measurements, shown in Figure 4.15 show a range of distributions where the more bitonal, text image as opposed to highly-textured

Table 4.2: Demosaicing. With the two color prior. For experiments run with the images in Figure 4.14 our method shows slight improvements for the green channel and no consistent improvements for the red and blue channels.

Image Number	R		G		B	
	sucessive	two color	sucessive	two color	sucessive	two color
1	36.66	36.96	37.64	38.04	40.88	40.77
2	35.54	35.15	39.05	38.48	40.08	40.13
3	39.13	39.07	39.44	39.32	42.30	42.42
4	36.19	36.19	40.85	40.96	42.66	42.50
5	35.59	35.36	35.34	35.10	38.27	38.25
6	37.97	38.12	37.17	37.39	41.64	41.52
7	40.09	39.96	39.39	39.19	42.70	42.69
8	34.44	34.51	35.00	35.17	38.74	38.62
9	40.64	40.85	41.42	41.29	43.49	43.62
10	40.07	40.15	40.48	40.58	43.90	43.77
11	37.50	37.61	38.54	38.91	41.65	41.42
12	41.07	41.26	41.37	41.45	44.76	44.57
13	34.70	35.08	33.35	33.63	37.90	37.63
14	33.33	32.98	34.39	34.02	36.74	37.07
15	35.31	35.11	38.72	38.76	40.59	40.31
16	38.60	38.97	39.34	39.80	43.35	42.96
17	40.63	40.55	39.82	39.86	43.13	43.03
18	35.51	35.28	36.43	36.44	38.61	38.55
19	38.53	38.64	39.59	39.89	42.39	42.21
20	38.71	38.83	37.20	37.32	41.74	41.81
21	38.47	38.56	37.43	37.57	41.80	41.78
22	36.93	36.70	36.91	36.91	39.72	39.55
23	38.05	37.91	40.53	40.74	41.88	42.23
24	34.79	34.30	32.80	32.92	37.59	37.37

or shaded image, have a sharper dip near $\alpha = 1$. Our experiments show that this difference in sharpness is reasonably well modeled by scaling the penalty function, i.e. the λ_2 parameter.

To better understand the qualitative affect of different priors we clustered the measured distributions into two clusters to represent “natural” vs. “text” images and fit piecewise cubic penalty functions to each, shown in Figure 4.16. We then performed deconvolution for the image shown in Figure 4.1 with only the alpha and two-color prior for regularization, i.e. no sparse prior. In Figure 4.16 we show the result of using the penalty function shown in Figure 4.4, which was used for the rest of our results, compared with the penalty functions shown in Figure 4.15. We normalized the penalty function to have the same value at $\alpha = 1.5$, such that only the difference in the shape of the functions would affected the optimization results (shown in Figure 4.16).

By deconvolving without the sparse prior, the results in Figure 4.16 illustrate the affect of the sparse prior when used with the two color prior. In the two color regions, the sparse prior helps remove small spatial variations in alpha that are otherwise not heavily penalized with the penalty function in Figure 4.4, as the area near $\alpha = 1.0$ is relatively shallow and allows for some fluctuation. The sharper bimodal priors minimize the need for the sparse prior as it minimizes these small fluctuations and there are fewer noise artifacts.

4.8 Discussion and Future Work

We have shown how to deblur images by developing a novel prior that incorporates local color statistics into the deconvolution process using a local two-color model. We have additionally shown how to use the same formulation for denoising an image. Our method produces sharper results with visibly less noise than current algorithms.

There are several areas for future work where our methods can be improved. Perhaps the largest limitation of our method is not in its theory, but in its practice. For deblurring, recovering both the color model and deconvolving the image is a non-linear problem. Furthermore, the error function we use can suffer from local minima due to the necessary existence of two minima in the alpha prior. Two obvious routes exist for improvement, the first being to investigate alternative optimization techniques. The second is to improve the initialization for the color model; we

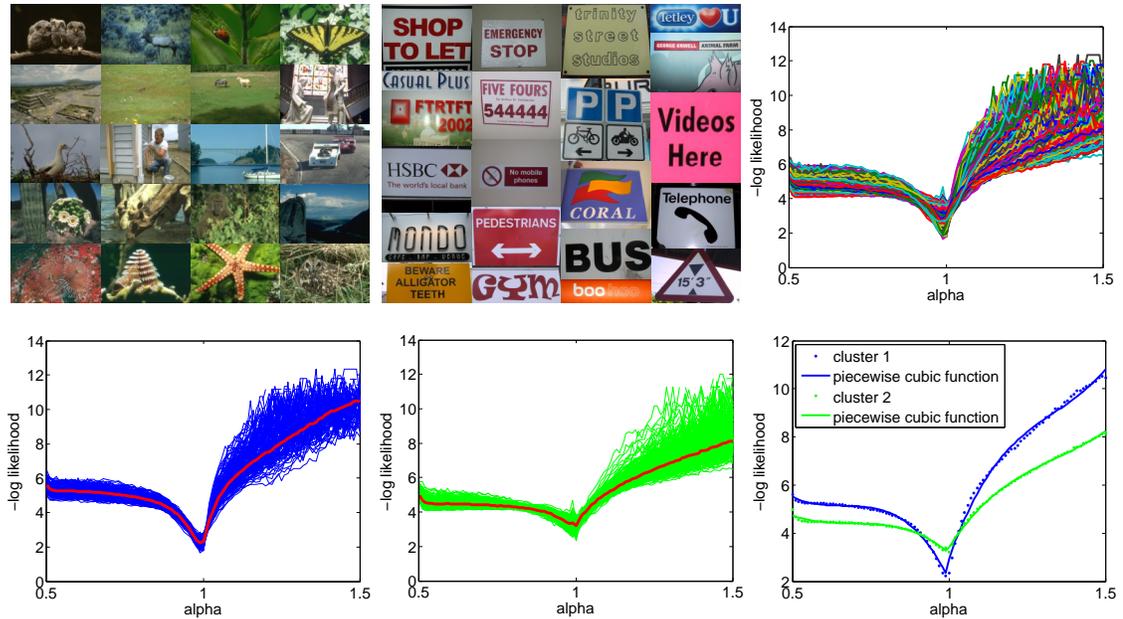


Figure 4.15: Measurements of alpha distributions. We measured α distributions for 300 images from the Berkeley Image Database (top left) and 160 “text” images from the ICDAR (International Conference on Document Analysis and Recognition) 2003 dataset (top middle). The α distributions for these images overlaid on one plot (top right). We clustered the distributions into two clusters (bottom left and middle) to represent “natural” vs. “text” images and fit penalty functions to each (bottom right).

have experimented with both the Lucy-Richardson algorithm and using the sparse prior alone. We found that using the sparse prior alone with a low weighting provides a good initial guess; however, due to our current optimization method, the quality of our results is somewhat bound by this initialization. Other choices may yield improved results.

In our experience, we have found that varying the weighting of the alpha priors can help create qualitatively better results. Our alpha measurements described in the previous section show that most often, text-like images require a higher weight than natural images. We are interested in exploring the alpha prior and weighting values in a more class-specific way. We believe that there may be a consistent, but different, set of weights for text versus natural images, and we believe that this parameter could be learned, eliminating the need for hand tuning the λ_2 parameter. Using some of the ideas from Fattal [2007] and Dai *et al.* [2007] may also be a way to improve our prior on α .

Another area for improvement is noise-level estimation, as the weighting of the data likelihood is directly dependent on this. We have found that errors in the estimation of σ^2 can

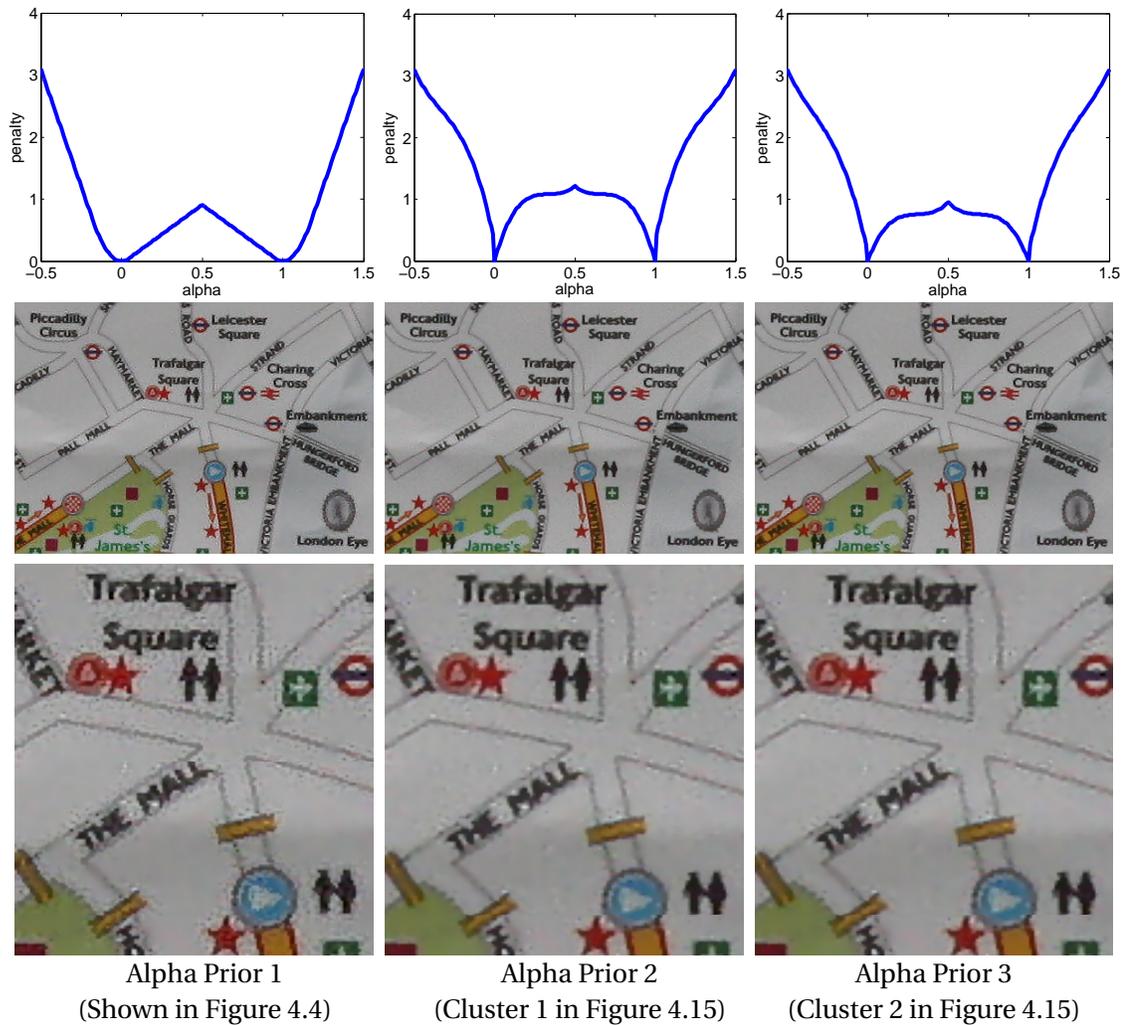


Figure 4.16: Comparing alpha penalty functions. In the two color regions, the sparse prior helps remove small spatial variations in alpha. Without the sparse prior, with a more shallow penalty function (left column) small fluctuations near $\alpha = 1.0$ are not heavily penalized and the deconvolution is noisier. The sharper bimodal priors minimize the need for the sparse prior as it minimizes these small fluctuations and there are fewer noise artifacts (middle and right column). There is minimal difference between the penalty function fit to “natural” vs. “text” images.

cause the priors to be weighed too heavily. Thus, we believe that improved methods for noise estimation could benefit our algorithm.

Lastly, we are investigating incorporating other image statistics. Currently, we use a fixed-size window around each pixel. While this works quite well, an interesting future direction would be to use local color statistics over increasingly larger windows and to also measure global statistics. Priors that minimize color entropy could provide additional constraints that would be

valid for a larger range of images and could be a useful addition to our algorithm.

4.9 Acknowledgements

This chapter is, in part, a reproduction of material being prepared for submission for publication: Neel Joshi, C. Lawrence Zitnick, Richard Szeliski, and David J. Kriegman, “Image Deblurring and Denoising using Color Priors”. I was the primary investigator and author of this paper. This work was partially completed while I was an intern at Microsoft Research.

Image Correction using Identify-Specific Priors

“It is the common wonder of all men, how among so many millions of faces, there should be none alike.”

—Sir Thomas Browne

The previous chapters have focused on two specific image corrections: deblurring and denoising using image models and statistical priors tuned to the content of an image. In this chapter, we take a different approach toward image correction. Specifically, instead of developing corrections for general images using tuned models, we impose content restriction and look at a very specific, yet large class of images. We note that many consumer photographs are of a personal nature, e.g., holiday photographs and vacation snapshots are mostly populated with the faces of the camera owner’s friends and family. Flaws in these types of photos are often the most noticeable and disconcerting. In this chapter, we present methods that seek to improve these types of photos and focus specifically on images containing faces. Our approach is to “personalize” the photographic process by using a person’s past photos to improve future photos. By narrowing the domain to specific, known faces we can obtain high-quality results and perform a broad range of operations.

We implement this personalized correction paradigm as a post-process using a small set of examples of good photos. The operations are designed to operate independently, so that a user can choose to transfer any number of image properties from the examples to a desired photograph, while still retaining certain desired qualities of the original photo. Our methods are automatic, and we believe this image correction paradigm is much more intuitive and easier to

use than current image editing applications.

The primary challenges involved in developing our “personal image enhancement” framework are 1) decomposing images such that a number of image enhancement operations can be performed independently from a small number of examples, 2) defining transfer functions so that only desired properties of the examples are transferred to an image, and 3) correcting non-face areas of images using the face and the example images as calibration objects. In order to accomplish this, we use an intrinsic image decomposition into shading, reflectance and color layers and define transfer functions to operate on each layer. In this chapter, we show how to use our framework to perform the following operations:

- **Deblurring:** removing blur for images when the blur function is unknown by solving for the blur of a face,
- **Lighting transfer and enhancement:** transferring lighting color balance and correcting detail loss in faces due to under exposure or saturation,
- **Super-resolution of faces:** creating high-resolution sharper faces from low resolution images.

We integrate our system with face detection [Viola and Jones, 2001] to obtain an automated system for performing the personalized enhancement tasks.

To summarize, the contributions of this chapter are 1) the concept of the personal “prior”: a small, identity specific collection of good photos used for correcting flawed photographs, 2) a system that realizes this concept and corrects a number of the most of common flaws in consumer photographs, and 3) a novel automatic multi-image deblurring method that can deblur photographs even when the blur function is unknown.

5.1 Related Work

As discussed in Chapter 2, digital image enhancement dates back to the late 60s. There are many applications and approaches in this area. The most relevant work to the contributions of this chapter are those methods that use image-derived priors. The use of image-derived

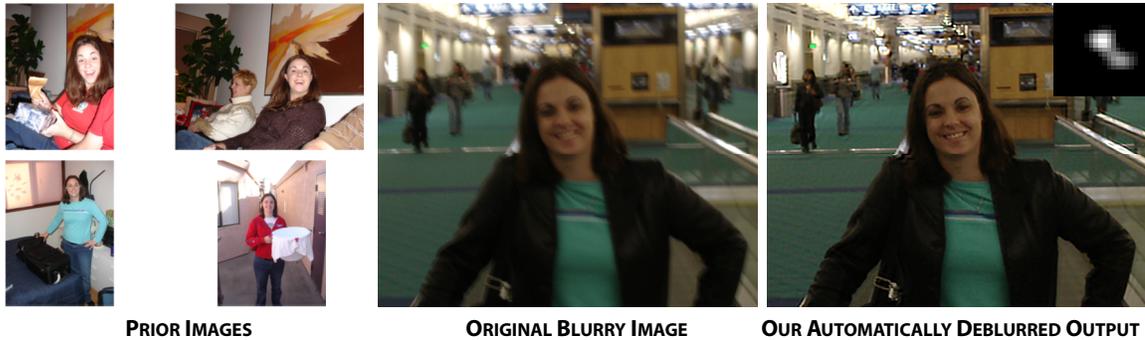


Figure 5.1: Automatically correcting personal photos. We automatically enhance images using prior examples of “good” photos of a person. Here we deblur a blurry photo of a person where the blur is unknown. Using a set of other sharp images of the same person as priors (left), we automatically solve for the unknown blur kernel and deblur the original photo (middle) to produce a sharp image (right) – the recovered blur kernel is shown in the top right enlarged $3\times$.

priors is a relatively recent development. Image-based priors have been exploited for super-resolution [Baker and Kanade, 2000; Freeman et al., 2002; Liu et al., 2007], deblurring [Fergus et al., 2006], denoising [Roth and Black, 2005; Liu et al., 2006; Elad and Aharon, 2006], view-interpolation [Fitzgibbon et al., 2005], inpainting [Levin et al., 2003], video matting [Apostoloff and Fitzgibbon, 2004], and fitting 3D models [Blanz and Vetter, 1999].

These priors range from statistical models to data-driven example sets, such as a face prior for face hallucination, a gradient distribution prior for natural images, or an example set of high and low resolution image patches; they are specific to a domain, but general within that domain. To the best of our knowledge, most work using image-based priors is derived from a large number of images that may be general or class/object specific, but there has been very little work in 2D image enhancement using *identity specific* priors. Most work using identity-specific information is in the realm of detection, recognition, and tracking in computer vision and face animation and modeling in computer graphics. In the latter realm, recent work by Gross et al. [2005] has shown that there are significant advantages to using person-specific models over generic ones.

A related area of work is photomontage and image compositing [Agarwala et al., 2004; Levin et al., 2004; Rother et al., 2006]. In the work of Agarwala et al., user interaction is combined with automatic vision methods to enable users to create composite photos that combine the best aspects of each photo in a set. Another related area of work is digital beautification. Leyvand et

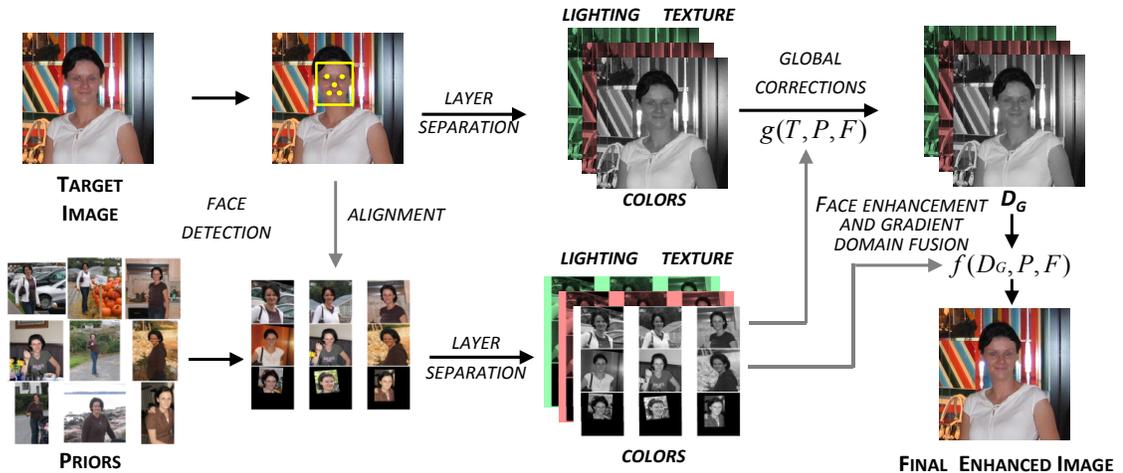


Figure 5.2: Personal image enhancement pipeline. First we use face detection to find faces in each image and align the prior images to the person in the target photo. The images are then decomposed into intrinsic images: color, texture, and lighting. First global image corrections are performed and then face-specific enhancements. We combine the global and face-specific results using gradient domain fusion.

al. [2006] use training data for the location and size of facial features for “attractive” faces as a prior to improve the attractiveness of an input photo of an arbitrary person. We see our work as complementary the work of Agrawala et al. and Leyvand et al., as while we all share similar goals of improve the appearance of people in photographs, we focus more on overcoming photographic artifacts and do not seek to change the overall appearance of a subject.

Our individual corrections use gradient domain operations pioneered by Perez et al. [2003]. Our work also share similarities with image fusion methods and transfer methods [Reinhard et al., 2001; Eisemann and Durand, 2004; Petschnigg et al., 2004; Agrawal et al., 2005; Bae et al., 2006] in that we use similar image decompositions and share similar goals of transferring photographic properties.

Our face specific enhancements are inspired by the face-hallucination work of Liu *et al.* [2007]. Liu *et al.* use a set of generic faces as training data that are pre-aligned, evenly lit, and grayscale. Where our work differs, is that we use identify-specific priors, automatic alignment, and a multi-layered image decomposition that enables operating on a much wider range of images, where the images can vary in lighting in color, and we perform operations in the gradient domain. These extensions enable using a more realistic set of images (with varied lighting and

color), improve matching, and give higher quality results. Furthermore, Liu *et al.* do not address in-painting and hallucinating entire missing regions, as we show in Figure 5.8.

Our deblurring algorithm is related to the work of Fergus *et al.* [2006] and multi-image deblurring methods [Bascle *et al.*, 1996; Rav-Acha and Peleg, 2005; Yuan *et al.*, 2007]. Fergus *et al.* recover blur kernels assuming a prior on gradients on the unobserved sharp image and in essence only assume “correspondence” between the sharp image and prior information in the loosest sense, in that they assume the two have the same global edge content. Multi-image deblurring is on the other end of the spectrum. These methods use multiple images of a scene acquired in close sequence and generally assume strong correspondence between images. Our method resides between the these two approaches with some similarities and several significant differences.

Relative to multi-image methods, we assume moderate correspondence, by using an aligned set of an identify-specific images; however, we allow for variations in pose, lighting, and color. To the best of our knowledge, deblurring using any type of face-space as a prior, let alone our proposed identity-specific one, is novel. Both our method and Fergus *et al.*'s are in the general (and large) class of EM style deblurring methods. Where they differ in the specific nature of the prior and that our method is completely automatic given a set of prior images. Fergus *et al.*'s work on the other hand requires user input to select a region of an image for computing a PSF. In our experience, this user input is not simple as it often requires several tries to select a good region and must be done for every image. Furthermore, our work is computationally simpler using an *maximum a posteriori* estimation instead of Variational Bayes, which leads to a ten to twenty times speedup.

5.2 Overview

We present several image enhancement operations enabled by having a small number of prior examples of good photos of a person. The enhancements are grouped into two categories: global image corrections and face-specific enhancements. Global corrections are performed on the entire image by using the known faces as calibration objects. We perform global exposure and white-balancing and deblurring using a novel multi-image deconvolution algorithm. For

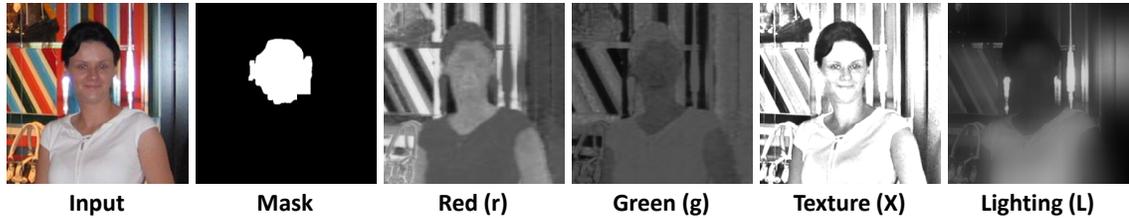


Figure 5.3: Mask computation and layer decomposition. We perform our corrections on a “intrinsic image” style decomposition of an image into color, lighting, and texture layers. This enables a small set of example images to be used to correct a broad range of input images. In addition, it allows us to modify image characteristics independently. We also automatically compute a mask for the face that we use as part of our face specific corrections.

faces in the image we can go beyond global correction and perform per-pixel operations that transfer desired aspects the of example images. We in-paint saturated and underexposed regions, correct lighting intensity variation, and perform face-hallucination to sharpen and super-resolve faces. Our system operates on base/detail image decomposition [Eisemann and Durand, 2004] and therefore these operations can be performed independently. As illustrated in Figure 5.2, our system proceeds as follows:

1. Automatically detect faces on target images and prior images.
2. Align and segment faces in target and prior images.
3. Decompose images into color, texture, and lighting layers.
4. Perform global image corrections.
5. Perform face-specific enhancements.

Step 1 outputs a set of nine feature points for each target and prior face and step 2 produces a set of prior images aligned to the target image with masks indicating the face on each image. Both steps are discussed in detail in Section 5.2.3. Step 3 is discussed in the next section, step 4 in Section 5.3, and step 5 in Section 5.4.

5.2.1 Prior Representation and Decomposition

In this work, we derived priors from a small collections of person-specific images. In contrast with previous work using large image collections [Hays and Efros, 2007], our goal is to

use data that is easily collected by even the most casual photographer, who may not have access to large databases of images.

Researchers have noted that the space spanned by the appearance of faces is relatively small [Turk and Pentland, 1991]. This observations has been used for numerous tasks including face recognition and face hallucination [Liu et al., 2007]. We make the additional observation that the space spanned by images of a single person is significantly smaller – when examining a personal photo collection the range of photographed expressions and poses of faces is relatively limited. Thus we believe the use of a small set of person-specific photos to be a relatively powerful source for deriving priors for image corrections.

While expression and pose variations may be limited, lighting and color can vary significantly between photos. As a result, a central part of our framework is the use of a base/detail layer decomposition [Eisemann and Durand, 2004] that we use as an approximate “intrinsic image” decomposition [Barrow and Tenenbaum, 1978; Land and McCann, 1971; Finlayson et al.; Weiss, 2001; Tappen et al., 2006]. In such as decomposition, an image is represented as a set of constituent images that capture intrinsic scene characteristics and extrinsic lighting characteristics. Intrinsic images are an ideal construct as they a) allow us to use a small set of prior images to correct a broad range of input images and b) they enable modifying image characteristics independently.

We adopt the base/detail layer decomposition used by Eisemann and Durand [2004] that makes this separation based on the Retinex assumption and uses an edge-preserving filter to decompose lighting from texture. We decompose an RGB image I into a set of four images $[r, g, L, X]$, where $Y = R + G + B$ represents luminance and $r = R/Y$ and $g = G/Y$ are red and green chromaticity. L , the lighting (or base) image, is a bilaterally filtered version of luminance Y . X , the shading image, is computed as $X = Y/L$. For the sake of simplicity of terminology, for the remainder of this chapter, we will refer to the (r, g) chromaticity reflectance images as “color layers”, L , the base image as the “lighting layer”, and X , the shading layer, as the “texture layer”.

The layers from our example set are used for direct example-based techniques and to derive statistical priors. To achieve this, we follow the hybrid model of Liu *et al.* [2007] and perform corrections using both a linear eigenspace and a patch-based non-parametric approach.

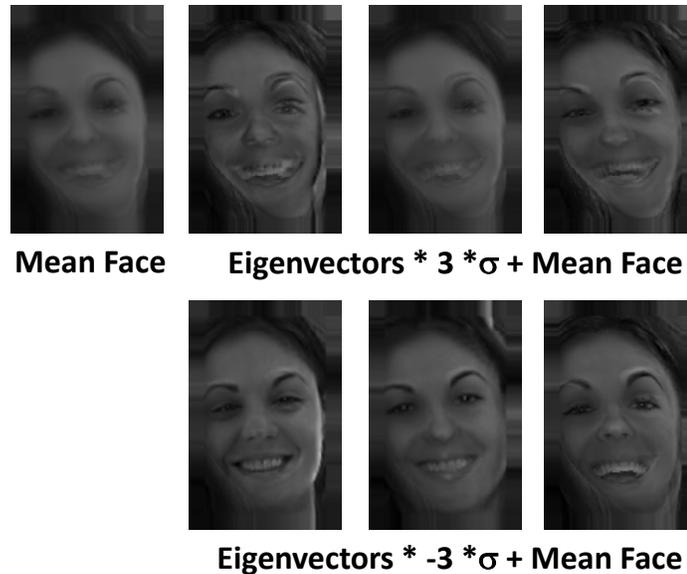


Figure 5.4: Eigenfaces Constraint. We use linear feature spaces built from an aligned set of good images of a person as a constraint in our image enhancement algorithms. Here we show the eigenfaces used as a prior for the delurring result shown in Figure 5.1.

5.2.2 Enhancement Framework

We create a desired processed image I with layers $I = [I_r, I_g, I_L, I_X]$ from a given observed image $O = [O_r, O_g, O_L, O_X]$ and an aligned set of prior images where one prior image is $P = [P_r, P_g, P_L, P_X]$. We automatically align the prior images P to O and compute a mask, F , for each face automatically. Alignment and mask computation is discussed in the next section.

The aligned, intrinsic prior layers are used directly for a patch-based method, and we also create eigenspaces for these layer. From each aligned and cropped intrinsic prior layers we create a set of orthogonal basis vectors using SVD. We denote \mathcal{P} as matrix of basis vectors and \mathcal{P}_μ as the mean vector that describe a feature space for the examples. An example of this is shown in Figure 5.4. Unlike previous work in this area, since our set of examples is small we do not use a subspace – our basis vectors capture all the variation in the data, and thus we are simply using SVD to orthogonalize the data. Another method such as Gram-Schmidt could be used. Thus, our “personal prior” is the entire set of aligned layers and basis and mean vectors for each space.

As illustrated in Figure 5.2, we perform image enhancement by creating a desired image I , by first performing global corrections to obtain the image I^G , and then we perform face-specific

corrections to obtain the final result I .

Face-specific enhancements are performed in the gradient domain using Poisson image editing techniques [Pérez et al., 2003], where an image is constructed from a specified 2D guidance gradient field, v , by solving a Poisson equation: $\partial I / \partial t = \nabla I - \text{div}(v)$. Specifically, this can be formed as a simple invertible linear system: $LI = \text{div}(v)$, where L is the Laplacian matrix. We refer the reader to paper by Perez *et al.* [2003] for more details on gradient domain editing.

5.2.3 Face Alignment and Mask Computation

To align faces in examples to a face in the input image, we use an extended implementation of the automatic face detection method of Viola and Jones [2001]. The detector outputs the locations of faces in an image along with nine facial features, the outside corner of each eye, the center of each eye, bridge and tip of the nose, and the left, right, and center of the mouth. From these features we align the faces using an affine transformation.

When performing face-specific enhancement, it is also necessary to have a mask for the face in the input and prior images. We automatically compute these by using the feature locations to initially compute a rough mask labeling face and non-face areas of the image. First our system creates a “trimap” by labeling the image as foreground, background and unknown regions. The foreground area is marked as the pixel inside the convex hull of the nine detected features. The edge is labeled as background and the remaining pixels are in unknown region. We compute an alpha-matte using the method of Levin et al. [2006] and threshold this soft-segmentation into a mask. The threshold is 50% and we eroded the mask pixels by ten pixels to get the final mask. An example of a mask is shown in Figure 5.3.

In the following sections, we describe our enhancement and correction functions and how each uses our prior.

5.3 Global Correction Operations

Many aspects of a person’s facial appearance, particularly skin color, albedo, and the location of features, such as the eyes and nose, remain largely unchanged over the course of time.

By leveraging their relative constancy, one can globally correct a number of aspects of an image.

We consider global corrections to be those that are calculated using the face area of an image and are applied to the entire image. Our global corrections use basis and mean vectors constructed from the example images as a prior within a Bayesian estimation framework. Our goal is to find the most likely estimate of the uncorrupted image I given an observed image O . This is found by maximizing the probability distribution of the posterior using Bayes' rule – also known as *maximum a posteriori* (MAP) estimation. The posterior distribution is expressed as the probability of I given O :

$$P(I|O) = \frac{P(O|I)P(I)}{P(O)}. \quad (5.1)$$

I can then be recovered by maximizing this posterior or minimizing of a sum of negative log likelihoods:

$$I = \operatorname{argmax}_I P(I|O) \quad (5.2)$$

$$= \operatorname{argmax}_I P(O|I)P(I) \quad (5.3)$$

$$= \operatorname{argmin}_I L(O|I) + L(I). \quad (5.4)$$

$L(O|I)$ is the “data” term and $L(I)$ is the image prior. The specific form of each value is different for each correction. In our system, we correct for overall lighting intensity and color balance and perform multi-image deconvolution to deblur an image.

5.3.1 Image Deblurring

We deblur an image of a person using our personal prior as a constraint during image deconvolution. While pixel-wise alignment of the blurred image and the prior images is difficult, a rough alignment is possible as facial feature detection on down-sampled blurred images is reliable. The feature space for texture layers from our personal prior is then used to constrain the underlying sharp image during deconvolution. We rely on the variation across the prior images to span the range of facial expressions and poses.

We only consider blur parallel to the image plane and solve for a shift-invariant kernel. We

model image blur as the convolution of a unknown sharp image with a unknown shift-invariant kernel plus additive white Gaussian noise:

$$O = I \otimes K + N, \quad (5.5)$$

where $N \sim \mathcal{N}(0, \sigma^2)$.

We formulate the image deconvolution problem using the Bayesian framework discussed above, except that we now have two unknowns I and K . We continue to minimize a sum of negative log likelihoods:

$$L(I, K|O) = L(O|I, K) + L(I) + L(K) \quad (5.6)$$

Given the blur formation model (Equation 5.5):

$$L(O|I, K) = \|O - I \otimes K\|^2 / \sigma^2. \quad (5.7)$$

We consider $O = O_X^F O_L^F$, which is the observed image's luminance, where the superscript F indicates that only the masked face region is considered (we will drop the F notation in later sections for the sake of readability). The sharp image I we recover is the deblurred luminance.

The negative log likelihood term for the image prior is:

$$L(I) = \lambda_1 L(I|\mathcal{P}, \mathcal{P}_\mu) + \|\nabla I\|^q, \quad (5.8)$$

which maintains that the image lies close to the examples' feature space by penalizing distance between the image and its projection onto the space, which is modeled by the eigenvectors and mean vector $(\mathcal{P}, \mathcal{P}_\mu)$. The term also includes the sparse gradient prior of Levin *et al.* [2007]: $\|\nabla I\|^q$.

The feature-space used in the prior is built from the examples' texture layers times the observation's lighting layer, i.e., $P^i = P_X^i O_L$, for each example i . This implicitly assumes that the blurring process does not affect the lighting layer, i.e. the priors examples and the observation have the same low-frequency lighting. While this assumption may not always be true, it holds in

practice as lighting changes tend to be low frequency.

To use this feature space, we define a negative log likelihood term using a robust distance-to-feature space metric:

$$L(I|\mathcal{P}, \mathcal{P}_\mu) = \rho([\mathcal{P}'\mathcal{P}(I - \mathcal{P}_\mu) + \mathcal{P}_\mu] - I), \quad (5.9)$$

\mathcal{P} represents the matrix whose columns are the eigenvectors of the feature-space, and \mathcal{P}' is the transpose of this matrix. This term enforces that the residual between the latent image I and the robust projection of I on to the feature-space $[\mathcal{P}, \mathcal{P}_\mu]$ should be minimal. $\rho(\cdot)$ is a robust error function described below.

We use a robust norm (rather than L_2 norm) to make this projection more robust to outliers (e.g., specular reflections and deep shadows on the target face or feature variations not well-captured by the examples). For $\rho(\cdot)$ we use the Huber norm:

$$\rho(r) = \begin{cases} \frac{1}{2}r^2 & |r| \leq k \\ k|r| - \frac{1}{2}k^2 & |r| > k \end{cases}. \quad (5.10)$$

k is estimated using the standard “median absolute deviation” heuristic. We use an iterative reweighted least squares approach to minimize the error function.

For the sparse gradient prior, instead of using $q = 0.8$, as Levin *et al.* use, we recover the exponent by fitting a hyper-laplacian to the histogram of gradients of the prior images' faces. To fit the exponent, consider that a p -norm distribution is $y = ce(x)^{-p}$ (c is a constant), taking the log of both sides results in: $\log(y) = \log(c) - p \log(e(x))$. If $y = \|\nabla I^F\|$ and x is the probability of different gradient values (as estimated using a histogram normalized to sum to one), p is the slope of the line fit to this data. By fitting the exponent in this way we constrain the gradients of the sharp image in a way that is consistent with the prior examples – in our experience, the recovered p is always between 0.5 and 0.6.

The prior on the kernel is modeled as a sparsity prior on the values and a smoothness prior on the kernel, which are common priors used during kernel estimation. The likelihood $L(K)$ is:

$$L(K) = \lambda_2 \|K\|^p + \lambda_3 \|\nabla K\|^2, \quad (5.11)$$



Figure 5.5: Exposure and color correction. Using the same set of prior images our system automatically corrects exposure and white-balance for three different images containing the same person.

where $p < 1$ ¹.

Blind deconvolution is then performed using a multi-scale, alternating minimization, where we first solve for I using an initial assumption for K (we use a 3x3 gaussian) by minimizing $L(I|B, K)$ and then use this I to solve for K by minimizing $L(K|B, I)$. Each sub-problem is solved using iterative re-weighted least-squares.

In performing deblurring, we recover only the sharp image data for the face and the kernel describing the blur for the face. If the person in target photograph did not move relative to the scene, this blur describes the camera-shake and we use the method of Levin *et al.* [2007] to deblur the whole image. A result from our method is shown in Figure 5.1.

5.3.2 Exposure and Color Correction

The goal of this part of our framework is to adjust the overall intensity and color-balance of the target photograph such that they are most similar to that of well-exposed, balanced prior images. We model this adjustment with scaling parameters for the lighting and color layers.

We robustly match the target face’s lighting and color to mean lighting and color vector from the prior feature-spaces. We again formulate this using the Bayesian framework and

¹We have found our method is relatively insensitive to the value of p as long as it is < 1 . $p = 0.8$ seems to work well.

minimize a sum of negative log likelihoods. For exposure correction, the data term is:

$$L(O|I) = \|O_L - \omega_L I_L\|^2, \quad (5.12)$$

where ω_L is a scalar value. The image prior is:

$$L(I) = L(I_L|\mathcal{P}_{L\mu}) = \rho(\mathcal{P}_{L\mu} - I_L). \quad (5.13)$$

For white-balancing, we use an equation of the same form to compute scaling values ω_r and ω_g using the respective I_r , I_g , $\mathcal{P}_{r\mu}$, and $\mathcal{P}_{g\mu}$ values.

In practice, if $\omega_L > 1$ we set it equal to 1, so we do not scale down image exposure. For color-balancing, as skin tones do not span a large color gamut we perform a simple white-balance, i.e. independent scaling of the color layers, as we have found it to be the only reliable transformation we can perform. In particular, we have found that a full linear transformation or a non-linear transform, such as histogram matching, often has too many degrees of freedom and produces undesired results. Examples of global exposure and white-balancing are shown in Figure 5.5.

5.4 Face-Specific Enhancement

In many cases, global corrections will not remove all the flaws in an image. While shortcomings of the corrections will generally be unnoticed for non-face regions, they are likely to be objectionable for faces as people are much more sensitive to their appearance. Thus, we perform local corrections on faces.

5.4.1 Modifying Lighting and Texture

We address several image corrections under the umbrella of hallucinating high-frequency texture. Lack of detail due to defocus blur or over-smoothing during deconvolution, lack of resolution, or saturation of an image can be corrected by transferring high-frequency information from the aligned texture layers of the personal prior. In the case of over and under-exposure, saturated and clipped areas can be in-painted by hallucinating parts of all intrinsic image layers.

Restoring High-Frequency Texture: For this process, we decouple hallucinating a sharp-texture layer I into making global and local estimates, I^L and I^G respectively, where the $I = I^L + I^G$. The global component I^G captures the lower-frequencies of the image and the local component I^L captures the highest-frequency data. This is the same decomposition used by Liu *et al.* [2007].

When the blur is unknown, such as for defocus and motion blur, I^G is the result of the blind-deconvolution method in Section 5.3.1. When the blur is known, such as with super-resolution, we minimize this equation:

$$L(I^G|O, K) = \|O - I^G \otimes K\|^2 / \sigma^2 + \lambda_1 L(I|\mathcal{P}, \mathcal{P}_\mu) + \|\nabla I\|^q. \quad (5.14)$$

When performing super-resolution, K is an anti-aliasing filter.

To recover I^L we use a patch-based non-parametric Markov network that is a combination of the method of Liu *et al.* and Freeman *et al.* [2002]. We model $I^L = I - I^G$, thus I^L is the highest frequency component and depends on the low frequency component I^G .

We compute training pairs from the prior texture layers of (P^{Li}, P_M^{Gi}) for all priors i , where $P_M^G = P^G - f \otimes P^G$, where f is a gaussian filter. We seek to find patches around each point that maximizes the compatibility function:

$$\phi(I^L(m, n) = P^{Li}(m, n), I_M^G(m, n)) = \|I_M^G(m, n) - P_M^{Gi}(m, n)\|^2. \quad (5.15)$$

Where $I^L(m, n)$ denotes a patch centered at $(ms + s/2, ns + s/2)$. With patch $s + 2$ we use a patch size of 10×10 pixels with a 2 pixel overlap. We have an additional affinity function that states that the overlapping region of patches must be similar:

$$\psi(I^L(m, n), I^L(m + i, n + j)) = \|\Omega(I^L(m, n)) - I^L(m + i, n + j)\|^2, \quad (5.16)$$

where Ω returns the over-lapping regions on the two patches given the patch size. We refer the reader to the paper by Liu *et al.* [2007] for more details on this derivation.

Intuitively, maximizing the function above says that on a patch by patch basis, we predict



Figure 5.6: Defocus blur. A set of sharp, in-focus priors (first image). An image suffering from blur due to mis-focus (second image). A close of on original image (third image) and the corrected face after removing the defocus blur by hallucinating texture from priors (fourth image), and a visualization showing which parts of the face came from different patches (fifth image).

the highest frequency data based on how the mid-frequencies of the target and priors match each other. Just as in Liu *et al.*'s work, our priors are roughly aligned, so we only consider patches at the same location in the priors, and use a raster-scan technique to perform the energy minimization [Freeman et al., 2002; Hertzmann et al., 2001].

Where our method differs from the previous techniques, is that we perform this patch-based prediction on separate color, lighting, and texture layers. An additional difference is that to assemble the final locally corrected image I^L , we composite the *gradients* of the $P^L(i)$ into O^L , which is the local component of the observed image O relative to the global correction: $O^L = O - I^G$. We have found the gradient domain process to generate much cleaner composites.

Examples of the methods presented here are shown in Figure 5.6 and Figure 5.7, where we add texture lost due to defocus blur and perform super-resolution.



Figure 5.7: Super-resolution. A set of sharp, in-focus priors (first image). An image with blur and JPEG artifacts at a low resolution (second image). A close of on original image up-sampled 2 times using bicubic interpolation (third image), the corrected face after 2x up-sampling and hallucinating texture from priors (fourth image), and a visualization showing which parts of the face came from different patches (fifth image).



Figure 5.8: Removing high-frequency shadows and uneven illumination. Prior images (on the left). A photo with uneven illumination, hard high-frequency shadows, and saturation (top row, second image). Our result (top row, third image). The shadow has been softened significantly and the saturated areas corrected. The bottom row shows close-up for the input image, our intermediate result after estimating the global texture layer, and the final result after using the patch based-method.

Restoring Clipped Data: When over or under-exposure causes pixel values to be clipped, there is a complete loss of texture, high and low-frequency, in those regions. Thus, to in-paint these regions we use the algorithm described above with a few modifications.

In the previous section, we discuss predicting a global estimate for the texture layer and then local estimates for texture and color layers. When restoring clipped data, the process must be altered slightly. This is because all of the data in the saturation region is unreliable. Thus, we must predict global estimates for all layers within the saturation region.

We construct a saturation/shadow mask for the clipped face region, where clipped pixels are those with original pixel values in any color channel above or below a threshold (we use ≤ 10 and ≥ 240 for images in the range $[0, 255]$). The saturation/shadow mask is incorporated to the global estimation process discussed previously, such that the algorithm fits the un-clipped regions to the eigenspace and the masked out region is filled with data that is most consistent with this fit to the unmasked regions. This is achieved using a simple masking function when performing robust least squares. Note that consistency is enforced with a sparse gradient penalty

across the whole image.

We compute the global high-frequency texture layer X_{IH}^g and color layers r_{IH}^g and g_{IH}^g using a joint eigenspace of texture and color. Specifically, we perform the same operation as when “Restoring High-Frequency Texture”, but instead of using an eigenspace for the texture layers alone we build the eigenspace by orthogonalizing the stacked vectors of $[X_{PH}^g, r_{PH}^g, g_{PH}^g]$ of the example high resolution images (indicated by the sub-script PH) and then solve Equation 5.14 with these values for the vector $[X_{IH}^g, r_{IH}^g, g_{IH}^g]$. Note that for this application the kernel K in Equation 5.14 is a delta function, since there is no blur. Before performing this correction, we first perform a global white balance (discussed in Section 5.3.2) such that the colors are similar between the examples and the target face. The global lighting correction is performed in a similar way using the lighting-space separately, i.e. an eigenspace for the L_{PH}^g layers of the example images.

To predict high-frequencies we then run our “Restoring High-Frequency Texture” algorithm (described above) on the global estimates for the texture and color layers, $[X_{IH}^g, r_{IH}^g, g_{IH}^g]$. For the lighting layer we only compute the global estimation and forgo the patch-based correction as the lighting tends to contain only low frequency information, and it is generally undesirable to transfer high-frequency lighting (such as hard shadows and specularities).

An example of the method presented here is shown in Figure 5.8.

5.5 Personal Photo Correction Application

We have implemented a prototype application and user interface for performing the corrections discussed in the previous sections. In addition to the correction features, we have a “suggestion” system, where the application can suggest that the user performs a particular correction on a loaded photograph after computing some simple image statistics and comparing these to the statistics of the priors images.

For deblurring suggestions, our system computes the standard deviation of the magnitudes of gradients for the face region of all priors. The standard deviation of these values is computed. Similarly, our system computes the magnitude and standard deviation of gradients in the face region. If

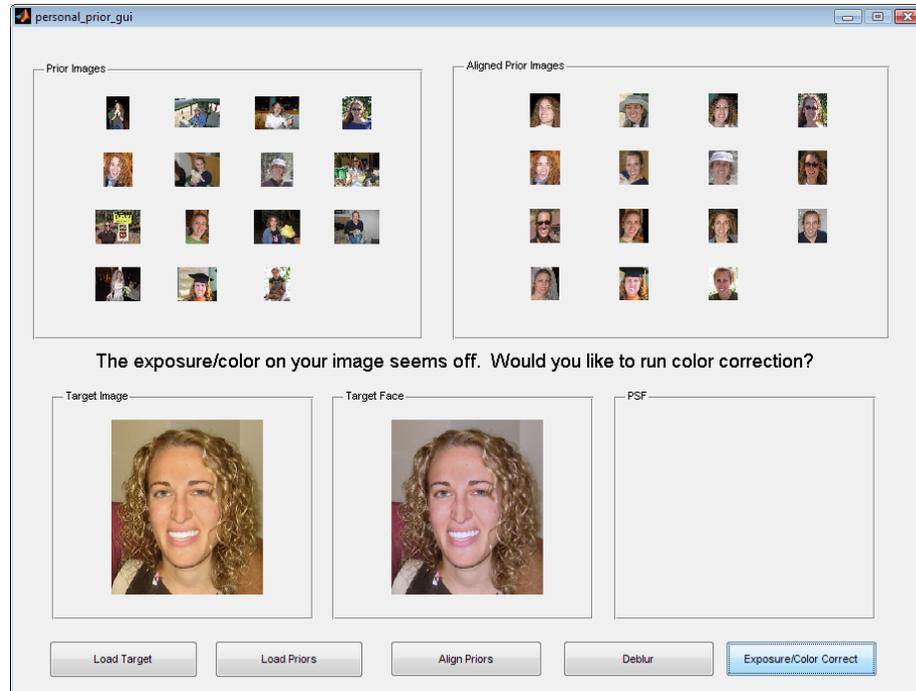


Figure 5.9: Personal photo correction application. Here we show a screenshot of an initial prototype of our personal photo correction application. Currently the system performs automatic deblurring and automatic exposure and white-balancing. It can also suggest corrections by using the statistics of the personal-prior.

the latter value is less than 90% of the former, the system suggests performing deblurring. For the color and exposure suggestion, the application simply computes the correction discussed in Section 5.3.2, as this is a fast operation, and if the scalar adjustment for color or lighting is > 1.1 or < 0.9 (more than a 10% change) it suggests the correction. Currently, a photo must be actively loaded for a suggestion to be made; however, the process could easily be run off-line as a way to automatically tag a collection of new photos with suggested corrections.

Figure 5.9 shows a screen-capture of the GUI, and we encourage the reader to visit <http://graphics.ucsd.edu/~neel/dissertation/> for a video that shows several sequences of the application being used.

5.6 Results

We will now briefly recap some of our results that were presented in the body of the chapter and present several additional examples.

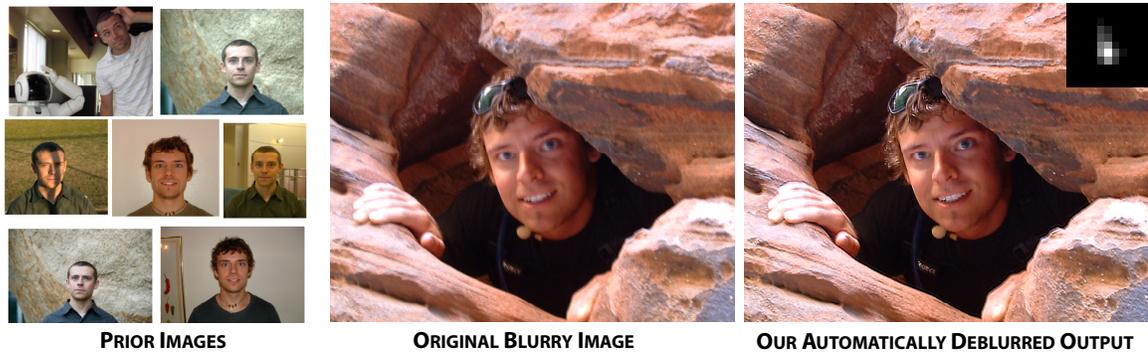


Figure 5.10: Additional deblurring example. Our method automatically performs blind-deconvolution to recover the blur kernel. The entire image is then deblurred using the method of Levin *et al.* – the recovered blur kernel is shown enlarged 5 \times in the top right of the rightmost image.

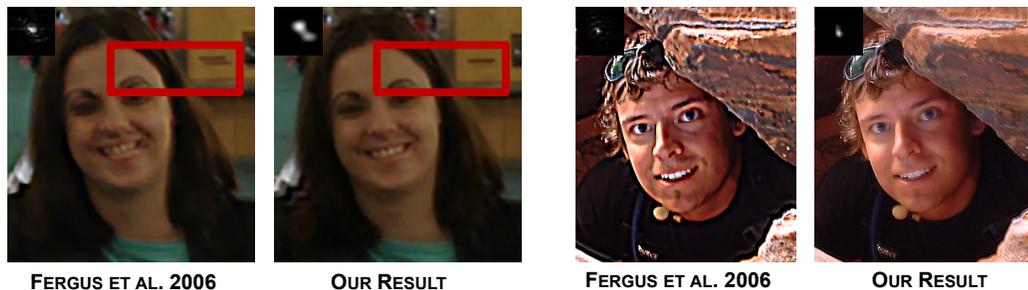


Figure 5.11: Comparison to Fergus *et al.*'s PSF estimation method. For the images in Figure 5.10, we hand-picked the a good region for estimating the PSF using Fergus *et al.*'s code, run their method, and then each image is then deblurred using the method of Levin *et al.*. Fergus *et al.*'s system took over an hour and a half to recover the kernel. Our method recovers more accurate kernels and produce better deconvolution results.

In Figures 5.1 and 5.10, we show two examples of our automatic blind deblurring method using our personal-prior. In Figure 5.11 we compare our method to using Fergus *et al.*'s [2006] method. We recovered a PSF using the authors' code available online. The side-by-side comparison shows deconvolving the image using the method of Levin *et al.* [2007] with the PSF from our method and Fergus *et al.*'s. For the woman in Figure 5.1, deblurring with the recovered PSF using Fergus *et al.*'s method does not completely sharpen the image. For the man in Figure 5.10, the PSF from the Fergus *et al.* method seems to be over-compensating, and thus the result is overly-sharpened with halo artifacts. For both images, the kernel recovered by our method appears more accurate and was recovered over ten times faster. Furthermore, our method does not require any manual input.

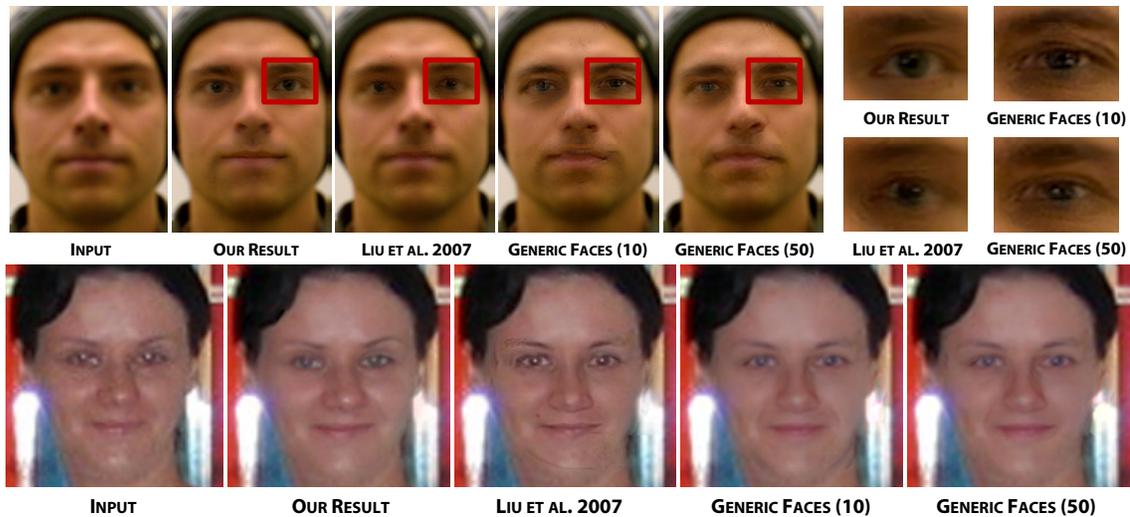


Figure 5.12: Face hallucination comparisons. We compare our result to performing hallucination using an implementation of Liu *et al.*'s method and to using our enhancement algorithm with a set of (the 10 and 50 best matching) generic faces instead of faces of the same person. Compared to Liu *et al.*'s approach and compared to using generic faces, our results have fewer artifacts and appear to retain the original look and expression of the person in the input image.

In Figures 5.6 and 5.7 we show two examples of face hallucination to remove image blur. Figure 5.6 shows a significant amount of defocus that is automatically removed using our method. Figure 5.7 shows hallucination for a 2x up-sampling compared to up-sampling with bi-cubic interpolation. For both results we show a visualization that indicates what regions of the final results came from different prior images when performing patch-based local hallucination. Both results are sharp with minimal artifacts. Furthermore, for the up-sampling result in Figure 5.7 our method has removed some of the JPEG artifacts in the original image.

In Figure 5.12, we compare results in Figures 5.6 and 5.7, to the result of performing hallucination using an implementation of Liu *et al.*'s [2007] method. Specifically, their work predicts high-frequency texture image from mid-frequency image data. They use a generic set of faces and use the raw image data directly (without an intrinsic layer decomposition or gradient domain editing). We used images from the public Caltech and GeorgiaTech image databases as our set of generic faces examples. The results with Liu *et al.* [2007] method has artifacts similar to those shown in their paper. We believe our results are more convincing. Also in Figure 5.12 we compare using our enhancement algorithm method with a set of generic faces instead of faces of

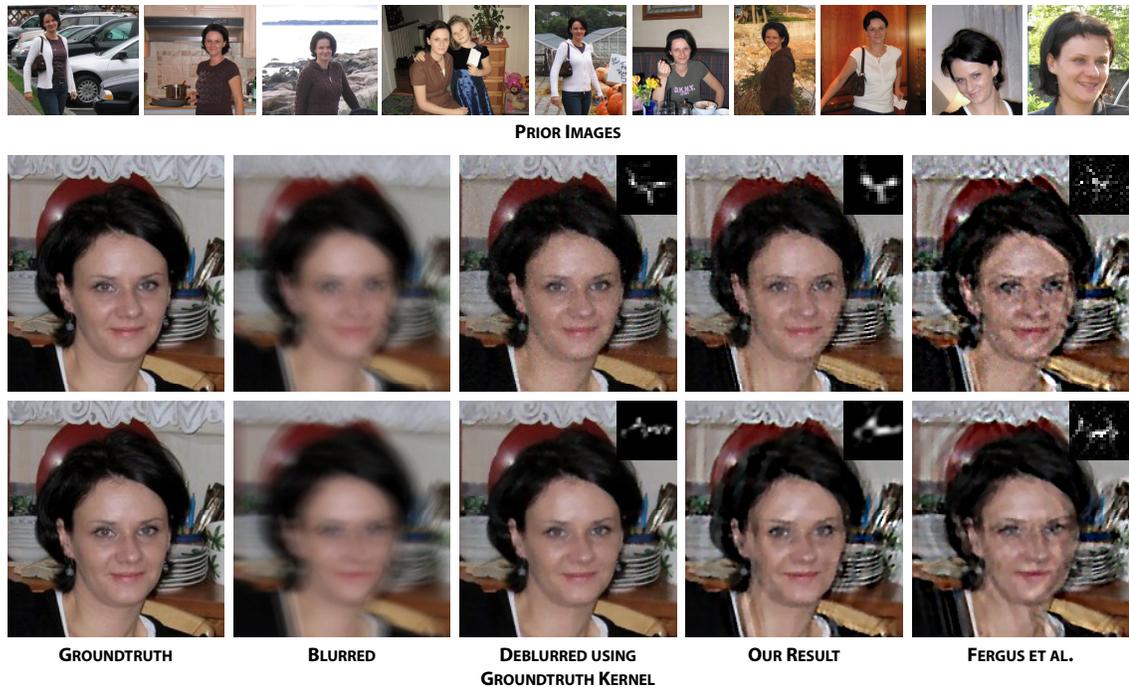


Figure 5.13: Synthetic deblurring experiments. We blurred a sharp image with two different blur kernels, added 0.5% noise, and then solved for the PSF using our deblurring method and the method of Fergus *et al.* We then deconvolved the blurred image with the groundtruth kernel, our recovered kernel, and the kernel resulting from running Fergus *et al.*'s code (shown in the top right corner of each image).

the same person. Thus instead of using 5 to 10 hand-selected good images of a person, we used the 10 and 50 best matching generic faces as priors. We automatically selected these images by first aligning the generic faces to the input image and then compute a match score that is the L_2 norm of the difference of down-sampled/contrast-normalized versions of the image. For both results, compared to using generic faces, our results have fewer artifacts and appear to retain the original look and expression of the person in the input image. The woman's face in Figure 5.12 is particularly of note. When using generic faces and Liu *et al.*'s method, a mole is introduced into the up-sampled result, even when the input image shows no mole. When using our method with generic faces, the woman no longer looks completely like the same person, and the expression of the woman is altered, as she no longer appears to be smiling as much.

Figure 5.13 shows the results of two synthetic deblurring experiments, where we blurred a sharp image with a blur kernel, added 0.5% noise, and then solved for the PSF using our deblurring method and the method of Fergus *et al.*. We then used the method of Levin *et al.* [2007]



Figure 5.14: Synthetic upsampling experiments. We down-sampled a sharp image by $2\times$ and $4\times$ and then up-sampled those images by $2\times$ and $4\times$, respectively, using our enhancement algorithm, an implementation of Liu *et al.*'s method, and our enhancement algorithm with a set of (the 10 and 50 best matching) generic faces.

to deconvolve the blurred image with the groundtruth kernel, our recovered kernel, and the kernel resulting from running Fergus *et al.*'s code. The side-by-side comparisons shows that the deconvolution results with our recovered kernels are fairly close to the quality of the results that use the known kernels, which shows that our kernels are accurate. The results using Fergus *et al.*'s method are not very accurate for these images.

Figure 5.14 shows the results of two synthetic up-sampling/halucination experiments, where we down-sampled a sharp image by $2\times$ and $4\times$ and then up-sampled those images by $2\times$ and $4\times$ using our enhancement algorithm, an implementation of Liu *et al.*'s method, and our enhancement algorithm with a set of (the 10 and 50 best matching) generic faces. With the $2\times$ up-sampling result, our method produces a convincing result that is sharper than traditional bi-cubic upsampling. Similar to the result in Figure 5.12, when using Liu *et al.*'s method, a mole is introduced into the up-sampled result, and, when using our method with generic faces, there are a number of artifacts including that woman's identity and expression are significantly changed. When performing $4\times$ up-sampling, our method produces an image that is a bit sharper than the others; however, this result shows the limits of our method—while we are able to hallucinate high-

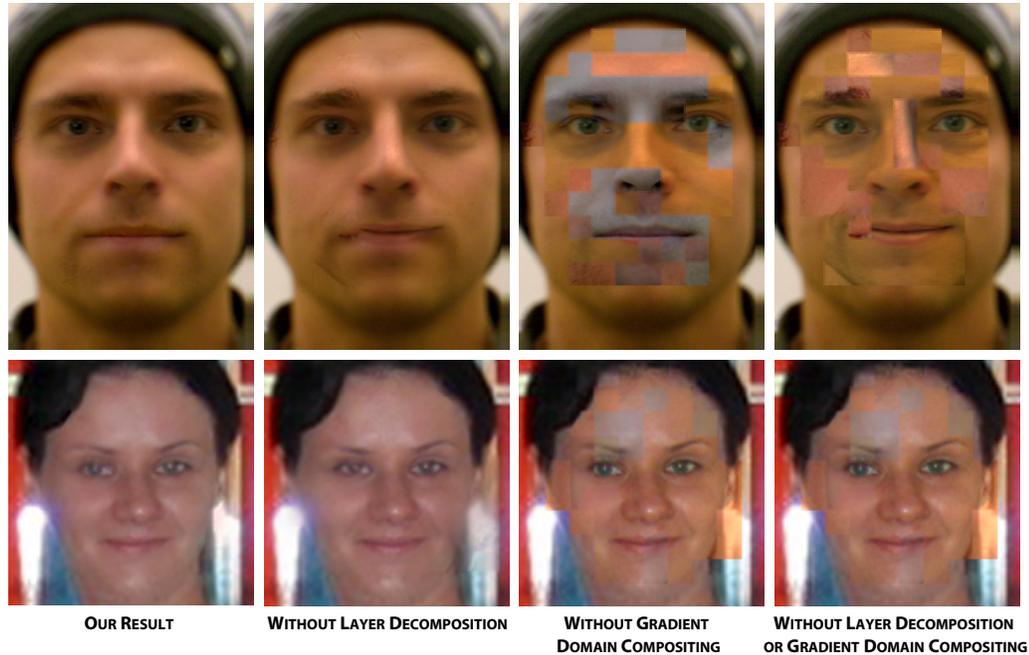


Figure 5.15: Face hallucination algorithms without using an intrinsic image decomposition and gradient domain editing. The results without using an intrinsic image decomposition have more artifacts as the effects of the lighting and color differences between the input image and examples are no longer factored out of the matching process. The results without gradient domain edits show very noticeable seams.

frequencies, there is not enough mid-frequency information to predict the highest-frequencies well, thus the result does not match the groundtruth image very closely.

In Figure 5.15, we show results of experiments of using our face hallucination algorithms without using an intrinsic image decomposition and gradient domain editing operations – two of our improvements over the work of Liu *et al.* [2007]. The results without using an intrinsic image decomposition have more artifacts as the effects of the lighting and color differences between the input image and examples are no longer factored out of the matching process. The results without gradient domain edits show very noticeable seams. Our modifications create enhanced images that are seamless and convincing.

In Figure 5.5, we show automatic exposure correction and white-balancing for three images of one woman using the same set of prior images. In Figure 5.16, we compare our results to using several current color constancy algorithms. Specifically, we compared to algorithms discussed by van de Weijer *et al.* [2007] and use the author’s code and recommended parameters.



Figure 5.16: Comparisons to color constancy. We compare our results to the color constancy algorithms discussed by van de Weijer *et al.*. Our results are more consistent across images, appear better white-balanced, and did not require any parameter tuning.

Our results are more consistent across images, appear better white-balanced, and did not require any parameter tuning.

In Figure 5.8, we show an image of a woman in an apple orchard where her face has a hard high-frequency shadow edge across it and our algorithm reduces the shadow and recovered texture in the saturated region of the face.

5.7 Analysis of the Eigenspace Prior

The ideal constraint for blind deconvolution is one that only captures valid sharp images. One potential limitation of an Eigenspace is that certain images in the space may not represent valid images. In the case of an Eigenspace for deblurring, there is the additional problem that some images may also not be very sharp, e.g., the mean face, which is a part of the Eigenspace. This presents an additional difficulty, while the Eigenspace does restrict the space of valid sharp

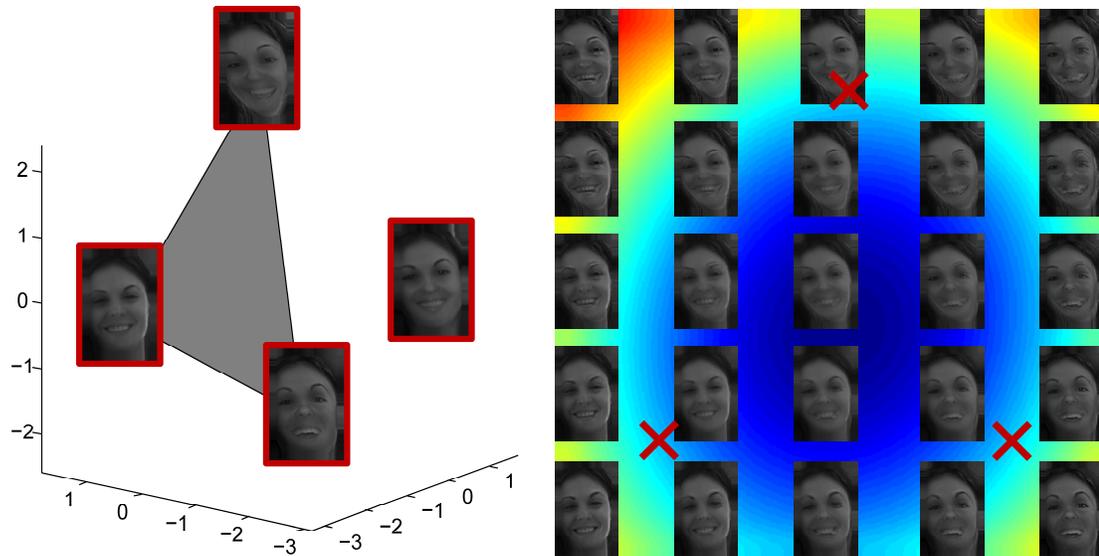


Figure 5.17: Edge strengths of images in the Eigenspace. (left) A visualization of the 3D space and a 2D planar slice that contains three of the four images used to create the space. (right) We sampled this 2D slice and for each sample, display the image and overlay it with a plot of the variance of the edge histograms. Samples near the center of the space – the ones that are closer to the mean, have lower variance values, i.e. they are smoother. The red “X” values shows the location of the three of the four input images to create the Eigenspace.

images significantly, it still may not completely constrain the solution significantly, as it may not constrain the high-frequencies well. In other words, one could imagine attempting to deblurring an image where the blurry image is actually quite close to or even in the Eigenspace, even if it is not close to one of the input images. In this case, the prior will help very little.

To further explore this we have sampled from the Eigenspace shown illustrated in Figure 5.4 and analyzed the edge content of images in this space. In Figure 5.17 we show a visualization of the 3D space and a 2D planar slice that contains three of the four images used to create the space. We sampled this 2D slice and for each sample, display the image and overlay it with a plot of the variance of the edge histograms. As expected, the samples near the center of the space – the ones that are closer to the mean, have lower variance values, i.e. they are smoother. With our current prior, these image are equally valid, even though they do not represent the training data as well nor are they very sharp. This visualization indicates that a prior over this space, one that prefers sharp images close to the original images, might be worth investigating for future work.

5.8 Discussion and Future Work

We have presented a powerful framework for improving personal images and shown how to correct a number of the most common errors in photographs using what we believe is a simple, yet powerful concept of a “personal prior”. While recent work in data-driven methods for photograph correction has tended towards using large-generic databases and automated methods for picking “good” photographs [Hays and Efros, 2007], we have taken a very different approach. We have focused on correcting images with faces and have left the step of choosing good photographs to the user and automated the difficult part of editing the photo. We believe this is a very natural and intuitive way to think about correcting images of people.

A natural question for our work is how many example images are needed. We have found that this depends on the type of the correction performed. Exposure and color correction are not very sensitive to expression and pose changes and thus very few, even one photograph can be enough. Deblurring can require more images, but often not very many due to the robust estimation process we use – the algorithm will reject outlier regions of the face and favor matching to the more invariant parts of the face that are well-captured by the eigenspace. Hallucination is the most demanding, as it is not always an option to ignore parts of the face – however due to our combined subspace and local patch approach, we have gotten good results with as few as seven images. While some analysis has been done for the dimensionality of the generic “face space” [Penev and Sirovich, 2000], we are not aware of an analysis for a specific individuals; as future work, we are very interested in such an analysis.

A general limitation of facial appearance modification, which our work is susceptible to, is the sensitivity of people to the appearance of faces. In our experience, with our system we have found that users are very sensitive to even subtle changes of photographs of people, especially when the person is known to the user. Generally the only pleasing and acceptable corrections are subtle and small changes, and it is difficult to make significant changes to an image without altering the fundamental mood or feel of the photograph. Often a large change sends a photo into the “uncanny valley”. This concept states that aesthetic qualities related to modification of human appearance are subject to a curve where improvements in appearance are positive until a point

when suddenly there is a negative reaction when the modified appearance becomes disturbingly “uncanny”. This is a danger with our methods just as with any other work that modifies images of faces.

There are numerous possible avenues for future work. We think our system and methods could be easily incorporated into commercial photo editing products and could leverage photo-tagging and rating systems that are already available, such as image rating in Windows Vista, ratings and labels in Adobe Bridge, or tags on websites like Flickr and Facebook. Our work could be paired with a simple labeling/rating system so that users could mark images with tags such as “good color”, “sharp”, “good lighting”. Then images with these labels could be automatically chosen as priors. We are very interested in integrating our system into such an application. We also note that as an extension to our current “correction suggestion” system, we could use machine learning techniques to try to automatically pick examples of good priors once the user has tagged a few.

We note that our system is currently limited to mostly frontal photographs. This is primarily due to our face-detector having been trained and tuned for frontal faces. Non-frontal face detection is more difficult; however, there is much work in this area and we are interested in investigating improvements that would allow non-frontal face detection. We could then have a richer set pose-specific priors. A related and particular useful extension that would build on our paradigm is to extend our personal prior concept to detection, whereby feature detectors are tuned to a specific person. Our system also does not currently use recognition, thus if multiple people are in an image, the user must perform the identification to pair the priors. We have experimented with face recognition in our framework and hope to add this shortly to our application.

Another interesting question is what other forms of person-specific or class-specific prior information could be used for image manipulation. In this work, we have used a set of images – the main reason motivation being that everybody can easily acquire and select images that they like. We believe that our framework could be extended to perform even better with full 3D geometry, detailed reflectance properties (e.g., spatially varying BRDF and subsurface scattering properties), or a linear morphable face model. One could include priors for the whole body rather

than the faces only, and in principle, it would be also possible to store and use the priors about the environment (e.g., the places where the photos are typically taken). An obvious disadvantage of using such information is that, at least currently, it can be difficult to acquire this type of data; however, using more sophisticated datasets presents several directions for future work.

Lastly, while we have focused on improving images of faces, we note that our framework is actually more general. Our fundamental framework could be applied to any object specific appearance enhancement where one has detectors and example images for a specific object.

5.9 Acknowledgements

This chapter is, in part, a reproduction of material conditionally accepted for publication: Neel Joshi, Wojciech Matusik, Edward H. Adelson, and David J. Kriegman, “Personal Image Enhancement using Prior Images”, Conditionally accepted to ACM Transactions on Graphics. I was the primary investigator and author of this paper. This work was partially completed while I was an intern at Adobe Systems Inc.

Conclusions and Future Work

“I may not have gone where I intended to go, but I think I have ended up where I intended to be.”

—Douglas Adams

Photography is integral to many aspects of business and home life. Cameras are increasingly important in areas from automotives, to medicine, security, and entertainment. As a result, photography is being used in new scenarios and by new users whose needs are not best met by current photographic methods. In many cases photographs lack the quality needed for a desired application. Additionally, with the proliferation of low-cost cameras, i.e., point-and-shoots and camera-phones, combined with the significant growth in the number of casual photographers, there is strong need for simple, automatic, and accurate methods to correct image artifacts.

In this dissertation, we have explored the problem of image correction and enhancement by using image models that incorporate prior information. In contrast with previous work that has used generic image priors, we presented methods that use priors and models that are tuned to the content of a specific image.

We presented three areas of work. We first discussed “PSF Estimation using Sharp Edge Prediction”, where by making the under-lying assumption that all edges in a sharp image are step-edges, our algorithm predicts the “sharp” version of a blurry input image and uses the two images together to solve for a PSF.

We then discussed “Image Enhancement using Color Statistics”, where we have investigated using local-color statistics of an image to improve the deblurring, denoising, up-sampling,

and demosaicing of images using a single framework.

Lastly, we discussed “Image Correction using Identify-Specific Priors”, where we developed methods that use identity-specific example images to provide the guidance needed to perform deblurring, up-sampling, and color and exposure adjustments automatically.

While there has been significant advancement in photography in recent years, the majority of photographers continue to follow a traditional process: photos are taken one at a time with a single camera, developed and/or processed using extensive manual methods or relatively rudimentary automatic techniques, and viewed as static images. Research in computer vision and graphics has begun to break this mold, and in this dissertation we have addressed some specific aspects of this endeavor.

There are several future directions for related work, and there are three specific high-level directions that seem promising.

6.0.1 Building more “Intelligence” into the Photographic Process

Traditionally, photography is a serial process consisting of image acquisition, processing, and display. As photographers become more experienced, they learn and refine their process and the quality of their images improve, but the process itself is un-evolving and memory-less. A promising area for further research is how to create a processing pipeline that learns from the results generated by a particular photographer or from the properties of more general image collections. This dissertation has touched on a several facets of this; however, there are interesting avenues for future work. For instance, other types of domain specific knowledge could be used for image correction, such as priors or models for different types of images, i.e. portraits vs. text vs. landscapes. Alternatively, more extensive machine learning approaches could be used to model a particular photographer’s preferences and habits.

6.0.2 Video Enhancement using Content Specific Priors

The work in this dissertation has focused on still images, yet videos suffer from many similar artifacts. However, while many artifacts are similar, extensions of still image methods to include a temporal component for video are not trivial. An interesting direction is to adapt

and extend the work in this dissertation from still image corrections to video. Due to the large amounts of data and time involved in processing videos, automatic correction methods have even more potential to improve current video processing pipelines.

6.0.3 Enhancement using Images and Video

Another aspect of the digital camera boom is not only are there more still cameras in peoples' hands, but there are also many more video cameras, as virtually all point-and-shoots and increasingly more camera-phones have video modes. As it is becomes easier to acquire both forms of these media in quick succession or even simultaneously, there are increasing opportunities for combined image and video correction.

Bibliography

- Aseem Agarwala, Mira Dontcheva, Maneesh Agrawala, Steven Drucker, Alex Colburn, Brian Curless, David Salesin, and Michael Cohen. Interactive digital photomontage. *ACM Transactions on Graphics*, 23(3):294–302, August 2004. 104
- Amit Agrawal, Ramesh Raskar, Shree K. Nayar, and Yuanzhen Li. Removing photography artifacts using gradient projection and flash-exposure sampling. *ACM Transactions on Graphics*, 24(3):828–835, August 2005. 105
- Nicholas Apostoloff and Andrew Fitzgibbon. Bayesian video matting using learnt image priors. In *2004 Conference on Computer Vision and Pattern Recognition (CVPR 2004)*, volume 01, pages 407–414, 2004. 104
- G. R. Ayers and J. C. Dainty. Iterative blind deconvolution method and its applications. *Opt. Lett.*, 13(7):547, 1988. 27
- Soonmin Bae, Sylvain Paris, and Frédo Durand. Two-scale tone management for photographic look. *ACM Transactions on Graphics*, 25(3):637–645, July 2006. 105
- Simon Baker and Takeo Kanade. Hallucinating faces. In *Fourth International Conference on Automatic Face and Gesture Recognition*, March 2000. 3, 104
- Mark R. Banham and Aggelos K. Katsaggelos. Digital image restoration. *Signal Processing Magazine, IEEE*, 14(2):24–41, 1997. 18

- Danny Barash. A fundamental relationship between bilateral filtering, adaptive smoothing, and the nonlinear diffusion equation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(6):844–847, 2002. 41
- H.G. Barrow and J.M. Tenenbaum. Recovering intrinsic scene characteristics from images. *Computer Vision Systems*, pages 3–26, 1978. 108
- Benedicte Bascle, Andrew Blake, and Andrew Zisserman. Motion deblurring and super-resolution from an image sequence. In *ECCV '96: Proceedings of the 4th European Conference on Computer Vision-Volume II*, pages 573–582, London, UK, 1996. Springer-Verlag. 35, 78, 106
- Moshe Ben-Ezra and Shree K. Nayar. Motion-based motion deblurring. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(6):689–698, 2004. 35, 78
- Eric P. Bennett, Matthew Uyttendaele, C. Lawrence Zitnick, Richard Szeliski, and Sing Bing Kang. Video and image bayesian demosaicing with a two color image prior. In Ales Leonardis, Horst Bischof, and Axel Pinz, editors, *Computer Vision - ECCV 2006, 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006, Proceedings, Part I*, volume 3951 of *Lecture Notes in Computer Science*, pages 508–521. Springer, 2006. 83, 84, 94
- Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *Proceedings of SIGGRAPH 99*, Computer Graphics Proceedings, Annual Conference Series, pages 187–194, August 1999. 104
- Philip Bones, Rick P. Millane, and Timo R. Bretschneider. Blur identification using image features. In *Adaptive Optics: Analysis and Methods/Computational Optical Sensing and Imaging/Information Photonics/Signal Recovery and Synthesis Topical Meetings on CD-ROM*, page SMA5. Optical Society of America, 2005. 34
- David H. Brainard and William T. Freeman. Bayesian color constancy. *J. Opt. Soc. Am. A*, 14(7): 1393–1411, 1997. 52
- Antoni Buades, Bartomeu Coll, and Jean-Michel Morel. A non-local algorithm for image denoising. In *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision*

- and Pattern Recognition (CVPR'05) - Volume 2*, pages 60–65, Washington, DC, USA, 2005. IEEE Computer Society. 39, 41
- G. Buchsbaum. A spatial processor model for object colour perception. *Journal of the Franklin Institute*, 310:1–26, 1980. 49
- Peter D. Burns and Don Williams. Using slanted edge analysis for color registration measurement. In *IS&T PICS Conference*, pages 51–53. Society for Imaging Science and Technology, 1999. 32, 58
- M. Cannon. Blind deconvolution of spatially invariant image blurs with phase. *Acoustics, Speech, and Signal Processing [see also IEEE Transactions on Signal Processing]*, *IEEE Transactions on*, 24(1):58–63, 1976. 18, 19, 20, 23, 34
- Vlad C. Cardei, Brian Funt, and Kobus Barnard. Estimating the scene illumination chromaticity using a neural network. *Journal of the Optical Society of America A*, 19:2374–2386, 2002. 52
- James N. Caron, Nader M. Namazi, and Chris J. Rollins. Noniterative blind data restoration by use of an extracted filter function. *Appl. Opt.*, 41(32):6884–6889, 2002. 30
- Sergio Carrato, Giovanni Ramponi, and Stefano Marsi. A simple edge-sensitive image interpolation filter. *Image Processing, 1996. Proceedings., International Conference on*, 3:711–714 vol.3, Sep 1996. 44
- M. Michael Chang, A. Murat Tekalp, and A. Tanju Erdem. Blur identification using the bispectrum. *Signal Processing, IEEE Transactions on [see also Acoustics, Speech, and Signal Processing, IEEE Transactions on]*, 39(10):2323–2325, 1991. 19, 20, 34
- Shengyang Dai, Mei Han, Wei Xu, Ying Wu, and Yihong Gong. Soft edge smoothness prior for alpha channel super resolution. In *CVPR '07: Proceedings of the 2007 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2007. 46, 94, 99
- Peter de Rivaz and Nick Kingsbury. Bayesian image deconvolution and denoising using complex wavelets. *Image Processing, 2001. Proceedings. 2001 International Conference on*, 2:273–276 vol.2, 2001. 13, 79

- Alexei A. Efros and William T. Freeman. Image quilting for texture synthesis and transfer. *Proceedings of SIGGRAPH 2001*, pages 341–346, August 2001. 41
- Elmar Eisemann and Frédo Durand. Flash photography enhancement via intrinsic relighting. *ACM Transactions on Graphics*, 23(3):673–678, August 2004. 105, 107, 108
- Michael Elad and Michal Aharon. Image denoising via learned dictionaries and sparse representation. In *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 895–900, Washington, DC, USA, 2006. IEEE Computer Society. 104
- Raanan Fattal. Image upsampling via imposed edge statistics. In *SIGGRAPH '07: ACM SIGGRAPH 2007 papers*, page 95, New York, NY, USA, 2007. ACM. 44, 46, 78, 94, 99
- Rob Fergus, Barun Singh, Aaron Hertzmann, Sam T. Roweis, and William T. Freeman. Removing camera shake from a single photograph. *ACM Transactions on Graphics*, 25(3):787–794, July 2006. 3, 26, 28, 30, 56, 57, 66, 67, 71, 78, 89, 90, 104, 106, 121
- Graham D. Finlayson and Gerald Schaefer. Constrained dichromatic colour constancy. In *ECCV '00: Proceedings of the 6th European Conference on Computer Vision-Part I*, pages 342–358, London, UK, 2000. Springer-Verlag. 52
- Graham D. Finlayson and Elisabetta Trezzi. Shades of gray and colour constancy. In *Color Imaging Conference*, pages 37–41. IS&T - The Society for Imaging Science and Technology, 2004. 49
- Graham D. Finlayson, Mark S. Drew, and Cheng Lu. Intrinsic images by entropy minimization. In Tomás Pajdla and Jiri Matas, editors, *Computer Vision - ECCV 2004, 8th European Conference on Computer Vision, Prague, Czech Republic, May 11-14, 2004. Proceedings, Part III*, volume 3023 of *Lecture Notes in Computer Science*, pages 582–595. Springer. 108
- Graham D. Finlayson, Steven D. Hordley, and Paul M. Hubel. Color by correlation: A simple, unifying framework for color constancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1209–1221, 2001. 50

- Graham D. Finlayson, Steven D. Hordley, and Ingeborg Tastl. Gamut constrained illuminant estimation. *Int. J. Comput. Vision*, 67(1):93–109, 2006. 50
- D. A. Fish, A. M. Brinicombe, E. R. Pike, and J. G. Walker. Blind deconvolution by means of the richardson-lucy algorithm. *J. Opt. Soc. Am. A*, 12(1):58, 1995. 29
- Andrew Fitzgibbon, Yonatan Wexler, and Andrew Zisserman. Image-based rendering using image-based priors. *Int. J. Comput. Vision*, 63(2):141–151, 2005. 104
- David A. Forsyth. A novel algorithm for color constancy. *Int. J. Comput. Vision*, 5(1):5–36, 1990. 50
- William T. Freeman, Thouis R. Jones, and Egon C. Pasztor. Example-based super-resolution. *IEEE Computer Graphics & Applications*, 22(2):56–65, March-April 2002. 3, 46, 78, 104, 116, 117
- Donald B. Gennery. Determination of optical transfer function by inspection of frequency-domain plot. *J. Opt. Soc. Am.*, 63(12):1571, 1973. 19, 20, 34
- Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2001. 54
- Hayit Greenspan, Charles H. Anderson, and Sofia Akber. Image enhancement by nonlinear extrapolation in frequency space. *Image Processing, IEEE Transactions on*, 9(6):1035–1048, Jun 2000. 45
- Ralph Gross, Iain Matthews, and Simon Baker. Generic vs. person specific active appearance models. *Image and Vision Computing*, 23(11):1080–1093, November 2005. 104
- S. Gull. Bayesian inductive inference and maximum entropy. *Maximum Entropy and Bayesian Methods in Science and Engineering, Volume 1: Foundations*, 1988. 26, 27, 29
- Gopal Harikumar and Yoram Bresler. Exact image deconvolution from multiple fir blurs. *Image Processing, IEEE Transactions on*, 8(6):846–862, 1999. 13, 79
- James Hays and Alexei A. Efros. Scene completion using millions of photographs. In *SIGGRAPH '07: ACM SIGGRAPH 2007 papers*, page 4, New York, NY, USA, 2007. ACM. 107, 128

- Glenn E. Healey and Raghava Kondepudy. Radiometric ccd camera calibration and noise estimation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 16(3):267–276, Mar 1994. 39
- Aaron Hertzmann, Charles E. Jacobs, Nuria Oliver, Brian Curless, and David H. Salesin. Image analogies. In *Proceedings of ACM SIGGRAPH 2001*, Computer Graphics Proceedings, Annual Conference Series, pages 327–340, August 2001. 46, 117
- Michal Irani and Shmuel Peleg. Motion analysis for image enhancement: Resolution, occlusion, and transparency. *Journal of Visual Communication and Image Representation*, 4:324–335, 1993. 35
- Aaron Isaksen, Leonard McMillan, and Steven J. Gortler. Dynamically reparameterized light fields. In *Proceedings of ACM SIGGRAPH 2000*, Computer Graphics Proceedings, Annual Conference Series, pages 297–306, July 2000. 35
- Jia Jiaya. Single image motion deblurring using transparency. In *CVPR '07: Proceedings of the 2007 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007. 24, 31
- Sing Bing Kang. Automatic removal of chromatic aberration from a single image. In *CVPR '07: Proceedings of the 2007 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007. 65
- D. Kundur and D. Hatzinakos. Blind image deconvolution. *SPMag*, 13(3):43–64, May 1996. 16, 56
- Edwin H. Land and John J. McCann. Lightness and retinex theory. *Journal of the Optical Society of America*, 61:1–11, 1971. 49, 108
- Hsien-Che Lee. Method for computing the scene-illuminant chromaticity from specular highlights. *J. Opt. Soc. Am. A*, 3(10):1694, 1986. 52
- Anat Levin. Blind motion deblurring using image statistics. In *Advances in Neural Information Processing Systems*, pages 841–848, Cambridge, MA, 2007. MIT Press. 24, 78

- Anat Levin, Assaf Zomet, and Yair Weiss. Learning how to inpaint from global image statistics. In *ICCV'03: Proceedings of the Ninth IEEE International Conference on Computer Vision*, page 305, 2003. 3, 104
- Anat Levin, Assaf Zomet, Shmuel Peleg, and Yair Weiss. Seamless image stitching in the gradient domain. In *In Eighth European Conference on Computer Vision (ECCV 2004)*, pages 377–389. Springer-Verlag, 2004. 104
- Anat Levin, Dani Lischinski, and Yair Weiss. A closed form solution to natural image matting. In *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 61–68, Washington, DC, USA, 2006. IEEE Computer Society. 83, 110
- Anat Levin, Rob Fergus, Frédo Durand, and William T. Freeman. Image and depth from a conventional camera with a coded aperture. *ACM Trans. Graph.*, 26(3):70, 2007. ISSN 0730-0301. 13, 16, 22, 23, 79, 82, 89, 93, 112, 114, 121, 123
- Marc Levoy and Pat Hanrahan. Light field rendering. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 31–42. ACM Press, 1996. 35
- Eli Levy, Doron Peles, Michal Opher-Lipson, and Stephen G. Lipson. Modulation transfer function of a lens measured with a random target method. *Appl. Opt.*, 38(4):679–683, 1999. 58
- Tommy Leyvand, Daniel Cohen-Or, Gideon Dror, and Dani Lischinski. Digital face beautification. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Sketches*, page 169, 2006. 105
- Xin Li. Demosaicing by successive approximation. *Image Processing, IEEE Transactions on*, 14(3): 370–379, March 2005. 95, 96
- Xin Li and Michael T. Orchard. New edge directed interpolation. *Image Processing, 2000. Proceedings. 2000 International Conference on*, 2:311–314 vol.2, 2000. 45
- Ce Liu, William T. Freeman, Richard Szeliski, and Sing Bing Kang. Noise estimation from a single image. In *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 901–908, Washington, DC, USA, 2006. IEEE Computer Society. 39, 42, 64, 78, 80, 83, 93, 104

- Ce Liu, Heung-Yeung Shum, and William T. Freeman. Face hallucination: Theory and practice. *Int. J. Comput. Vision*, 75(1):115–134, 2007. 3, 46, 104, 105, 108, 116, 122, 125
- L.B. Lucy. An iterative technique for the rectification of observed distributions. *Journal of Astronomy*, 79:745, June 1974. 13, 79
- B. C. McCallum. Blind deconvolution by simulated annealing. *Optics Communications*, 75: 101–105, February 1990. 27
- James Miskin and David J. C. Mackay. Ensemble learning for blind image separation and deconvolution. In *Advances in Independent Component Analysis*, 2000. 3, 26, 27, 30, 31
- Bryan S. Morse and Duane Schwartzwald. Image magnification using level-set reconstruction. *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, 1:I–333–I–340 vol.1, 2001. 45
- David Mumford and Jayant Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on Pure and Applied Mathematics*, 42(4), 1989. 31
- Shree K. Nayar and Yasuo Nakagawa. Shape from Focus: An Effective Approach for Rough Surfaces. In *International Conference on Robotics and Automation*, volume 1, pages 218–225, May 1990. 34
- Shree K. Nayar and Yasuo Nakagawa. Shape from Focus. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(8):824–831, Aug 1994. 34
- Shree K. Nayar, Masahiro Watanabe, and Minori Noguchi. Real-time focus range sensor. In *Fifth International Conference on Computer Vision (ICCV'95)*, pages 995–1001, Cambridge, Massachusetts, June 1995. 34
- Ramesh Neelamani, Hyeokho Choi, and Richard Baraniuk. Forward: Fourier-wavelet regularized deconvolution for ill-conditioned systems. *Signal Processing, IEEE Transactions on*, 52(2): 418–433, Feb. 2004. 13

- Shmuel Peleg and Alex Rav-Acha. Restoration of multiple images with motion blur in different directions. *Applications of Computer Vision, 2000, Fifth IEEE Workshop on.*, pages 22–28, 2000. 35
- Penio S. Penev and Lawrence Sirovich. The global dimensionality of face space. In *FG '00: Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition 2000*, page 264, Washington, DC, USA, 2000. IEEE Computer Society. 128
- Patrick Pérez, Michel Gangnet, and Andrew Blake. Poisson image editing. *ACM Transactions on Graphics*, 22(3):313–318, July 2003. 105, 110
- Pietro Perona and Jitendra Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Trans. Pattern Anal. Mach. Intell.*, 12(7):629–639, 1990. 40, 78
- Georg Petschnigg, Richard Szeliski, Maneesh Agrawala, Michael Cohen, Hugues Hoppe, and Kentaro Toyama. Digital photography with flash and no-flash image pairs. *ACM Transactions on Graphics*, 23(3):664–672, August 2004. 105
- Francois Pitie, Anil C. Kokaram, and Rozenn Dahyot. N-dimensional probability density function transfer and its application to color transfer. *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, 2:1434–1439 Vol. 2, Oct. 2005. 54
- Javier Portilla, Vasily Strela, Martin J. Wainwright, and Eero P. Simoncelli. Image denoising using scale mixtures of gaussians in the wavelet domain. *Image Processing, IEEE Transactions on*, 12(11):1338–1351, 2003. 41, 42, 78, 93
- Ashish Raj and Ramin Zabih. A graph cut algorithm for generalized image deconvolution. In *ICCV '05: Proceedings of the Tenth IEEE International Conference on Computer Vision*, pages 1048–1054, Washington, DC, USA, 2005. IEEE Computer Society. 13, 79
- Ramesh Raskar, Amit Agrawal, and Jack Tumblin. Coded exposure photography: motion deblurring using fluttered shutter. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Papers*, pages 795–804, New York, NY, USA, 2006. ACM. 78

- Alex Rav-Acha and Shmuel Peleg. Two motion-blurred images are better than one. *Pattern Recogn. Lett.*, 26(3):311–317, 2005. 35, 78, 106
- Stephen E. Reichenbach, Stephen K. Park, and Ramkumar Narayanswarthy. Characterizing digital image acquisition devices. *Optical Engineering*, 30(2):170–177, February 1991. 32, 33, 58
- Erik Reinhard, Michael Ashikhmin, Bruce Gooch, and Peter Shirley. Color transfer between images. *IEEE Comput. Graph. Appl.*, 21(5):34–41, 2001. 53, 105
- W. Richardson. Bayesian-based iterative method of image restoration. *Journal of the Optical Society of America A*, 62:55–59, 1972. 13, 26, 27, 28, 79
- J. Filip Rooms, Wilfried Philips, and Javier Portilla. Parametric PSF estimation via sparseness maximization in the wavelet domain. In F. Truchetet and O. Laligant, editors, *Wavelet Applications in Industrial Processing II. Edited by Truchetet, Frederic; Laligant, Olivier. Proceedings of the SPIE, Volume 5607, pp. 26-33 (2004).*, volume 5607 of *Presented at the Society of Photo-Optical Instrumentation Engineers (SPIE) Conference*, pages 26–33, November 2004. 22
- Stefan Roth and Michael J. Black. Fields of experts: A framework for learning image priors. In *2005 Conference on Computer Vision and Pattern Recognition (CVPR 2005)*, pages 860–867, 2005. 3, 42, 78, 104
- Carsten Rother, Lucas Bordeaux, Youssef Hamadi, and Andrew Blake. Autocollage. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Papers*, pages 847–852, New York, NY, USA, 2006. ACM Press. 104
- Andreas E. Savakis and H. Joel Trussell. Blur identification by residual spectral matching. *Image Processing, IEEE Transactions on*, 2(2):141–151, 1993. 21, 22
- Qi Shan, Wei Xiong, and Jiaya Jia. Rotational motion deblurring of a rigid object from a single image. In *ICCV '07: Proceedings of the Eleventh IEEE International Conference on Computer Vision*, 2007. 24
- Qi Shan, Jiaya Jia, and Aseem Agarwala. High-quality motion deblurring from a single image. *ACM Trans. Graph.*, 27(3):1–10, 2008. 71, 73, 74

- Eero P. Simoncelli and Edward H. Adelson. Noise removal via bayesian wavelet coring. *Image Processing, 1996. Proceedings., International Conference on*, 1:379–382 vol.1, 1996. 41, 42, 78
- Charles V. Stewart. Robust parameter estimation in computer vision. *SIAM Review*, 41(3):513–537, 1999. ISSN 0036-1445. 82
- Gilbert Strang. *Introduction to Applied Math*. Wellesley-Cambridge Press, 1986. 40
- Dan Su and Philip Willis. Image interpolation by pixel-level data-dependent triangulation. *Comput. Graph. Forum*, 23(2):189–202, 2004. 45
- Marshall F. Tappen, Edward H. Adelson, and William T. Freeman. Estimating intrinsic component images using non-linear regression. In *2006 Conference on Computer Vision and Pattern Recognition (CVPR 2006)*, pages 1992–1999, June 2006. 108
- Philippe Thévenaz, Thierry Blu, and Michael Unser. *Image interpolation and resampling*. Academic Press, Inc., Orlando, FL, USA, 2000. 44
- Carlos Tomasi and Roberto Manduchi. Bilateral filtering for gray and color images. In *ICCV '98: Proceedings of the Sixth International Conference on Computer Vision*, page 839, Washington, DC, USA, 1998. IEEE Computer Society. 40, 41, 78
- Shoji Tominaga and Brian A. Wandell. Standard surface-reflectance model and illuminant estimation. *J. Opt. Soc. Am. A*, 6(4):576, 1989. 52
- Yanghai Tsin, Visvanathan Ramesh, and Takeo Kanade. Statistical calibration of ccd imaging process. In *IEEE International Conference on Computer Vision (ICCV'01)*, July 2001. 39
- Jack Tumblin and Prasun Choudhury. Bixels: Picture samples with sharp embedded boundaries. In *Rendering Techniques*, pages 255–264, 2004. 45
- Matthew Turk and Alex Pentland. Face recognition using eigenfaces. In *Computer Vision and Pattern Recognition, 1991. Proceedings CVPR '91., IEEE Computer Society Conference on*, pages 586–591, 1991. 108

- Joost van de Weijer, Theo Gevers, and Arjan Gijsenij. Edge-based color constancy. *Image Processing, IEEE Transactions on*, 16(9):2207–2214, Sept. 2007. 49, 125
- Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *2001 Conference on Computer Vision and Pattern Recognition (CVPR 2001)*, pages 609–615, 2001. 103, 110
- Yi Wan and Robert Nowak. A bayesian multiscale approach to joint image restoration and edge detection. In *Wavelet Applications in Signal and Image Processing VII, Proc. SPIE*, pages 73–84, 1999. 26, 27, 30, 71
- Yair Weiss. Deriving intrinsic images from image sequences. *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, 2:68–75 vol.2, 2001. 108
- Joonshik Yoon, Jeongho Shin, and Joon Ki Paik. Enhancement of out-of-focus images using fusion-based psf estimation and restoration. In *VCIP*, pages 819–829, 2001. 18, 32
- Lu Yuan, Jian Sun, Long Quan, and Heung-Yeung Shum. Image deblurring with blurred/noisy image pairs. In *SIGGRAPH '07: ACM SIGGRAPH 2007 papers*, page 1, New York, NY, USA, 2007. ACM. 36, 78, 89, 90, 106
- Hongwei Zheng and O. Hellwich. Bayesian estimation based mumford-shah regularization for blur identification and segmentation in video sequences. In *SSIAI '06: Proceedings of the 2006 IEEE Southwest Symposium on Image Analysis and Interpretation*, pages 129–133, Washington, DC, USA, 2006a. IEEE Computer Society. 31
- Hongwei Zheng and Olaf Hellwich. Double regularized bayesian estimation for blur identification in video sequences. In P. J. Narayanan, Shree K. Nayar, and Heung-Yeung Shum, editors, *ACCV (2)*, volume 3852 of *Lecture Notes in Computer Science*, pages 943–952. Springer, 2006b. 31
- Hongwei Zheng and Olaf Hellwich. An edge-driven total variation approach to image deblurring and denoising. In *ICICIC '06: Proceedings of the First International Conference on Innovative Computing, Information and Control*, pages 705–710, Washington, DC, USA, 2006c. IEEE Computer Society. 31