

Multi-Channel Coded-Aperture Photography

by

Jongmin Baek

Submitted to the Department of Electrical Engineering and Computer
Science

in partial fulfillment of the requirements for the degree of

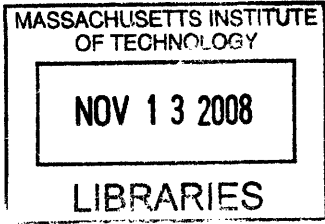
Master of Engineering in Computer Science and Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2008

© Massachusetts Institute of Technology 2008. All rights reserved.



Author
Department of Electrical Engineering and Computer Science
September 3, 2008

Certified by
Fredo Durand
Associate Professor
Thesis Supervisor

Accepted by
Arthur C. Smith
Chairman, Department Committee on Graduate Theses

ARCHIVES

Multi-Channel Coded-Aperture Photography

by

Jongmin Baek

Submitted to the Department of Electrical Engineering and Computer Science
on September 3, 2008, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Computer Science and Engineering

Abstract

This thesis describes the multi-channel coded-aperture photography, a modified camera system that can extract an all-focus image of the scene along with a depth estimate over the scene. The modification consists of inserting a set of patterned color filters into the aperture of the camera lens. This work generalizes the previous research on a single-channel coded aperture, by deploying distinct filters in the three primary color channels, in order to cope better with the effect of a Bayer filter and to exploit the correlation among the channels.

We derive the model and algorithms for the multi-channel coded aperture, comparing the simulated performance of the reconstruction algorithm against that of the original single-channel coded aperture. We also demonstrate a physical prototype, discussing the challenges arising from the use of multiple filters. We provide a comparison with the single-channel coded aperture in performance, and present results on several scenes of cluttered objects at various depths.

Thesis Supervisor: Fredo Durand

Title: Associate Professor

Acknowledgments

I would like to express deep gratitude to my thesis advisor, Fredo Durand, not only for his support and vision on this project for the past year, but also for introducing me to the exciting field of computational photography. It has been an amazing learning experience beyond the scope of the thesis, and I appreciate his insights, directions and discussions.

Also, I sincerely thank Anat Levin, whose experience with the original coded-aperture research was invaluable. Aside from laying the foundations and precursor to this thesis, she provided constant guidance, correspondence and meticulous attention to detail, without which this thesis would not have had its progress.

I further thank Ron Aiken for helping with CSAIL equipments, Bryt Bradley for handling administrative matters, and Jon Young at LEE filters for providing spectral data on the color filters.

Lastly, I thank my family for their support and prayers during my time at MIT, and all my friends and colleagues who have helped me thrive in perhaps the best four years of my life so far.

This work has been partly funded by a research assistantship from the Lincoln Laboratory.

THIS PAGE INTENTIONALLY LEFT BLANK

Contents

1	Introduction	17
1.1	Coded-Aperture Photography	18
1.2	Contributions of This Work	18
2	Related Work	19
2.1	Active versus Passive Methods	19
2.2	Single-Channel Coded-Aperture	21
2.2.1	Thin-Lens Model for Optics	21
2.2.2	Image and Depth Recovery	22
2.2.3	Sparse Prior	23
2.2.4	Filter Selection	24
2.2.5	Implementation	24
3	Mathematical Model	27
3.1	Notations	27
3.1.1	Signal Representation	27
3.1.2	Other Notations	28
3.2	Color Filter Array	29
3.3	Revised Thin-Lens Model for Optics	31
3.4	Prior for Multi-Channel Natural Images	32
3.4.1	Independent Prior	32
3.4.2	Independent Prior with a Change of Basis	32
3.4.3	Dependent Prior	33

4	Analysis of Model	35
4.1	Modelling the Pipeline	35
4.2	Reconstruction of Multi-Channel Image	36
4.2.1	Expected Reconstruction Error	37
4.2.2	Effect of Multiple-Channel Model	37
4.3	Depth Discrimination	41
4.3.1	Filter Selection Criterion	41
4.3.2	Depth Discriminateness of Multiple Filters	42
4.3.3	Summary	44
5	Numerical Techniques for Deconvolution	47
5.1	Newton’s Method	48
5.2	Iteratively Re-Weighted Least Squares (IRLS)	48
5.3	Gaussian Mixture Model (GMM)	50
5.4	Conjugate Gradient Method	51
5.5	Evaluation	51
5.5.1	Pixel Recovery in 1D Scenarios	51
5.5.2	Demosaicking in 2D Scenarios	54
5.5.3	Deblurring in 2D Scenarios	55
6	Physical Implementation	57
6.1	Filter Construction	57
6.1.1	Computing Attenuation	57
6.1.2	Filter Selection	59
6.1.3	Maintaining White Balance	60
6.1.4	Assembly	61
6.2	Calibration	63
7	Depthmap Generation	65
7.1	Classification with Deconvolution Error	65
7.1.1	Unnormalized Error	65
7.1.2	Normalized Error	66
7.1.3	Support Vector Machine (SVM)	66
7.2	Regularization	66

8	Experimental Results	69
8.1	Kernel Calibration	70
8.2	Robustness against Noise	72
8.2.1	Depth Classification in Presence of Noise	72
8.2.2	Quality of Deconvolution in Presence of Noise	72
8.3	Results on Physical Planar Scenes	75
8.3.1	Depth Classification	75
8.3.2	Quality of Deconvolution	76
8.4	Depthmap of Cluttered Scenes	78
8.5	Applications	80
8.6	System Performance	81
8.7	Discussion	81
9	Conclusion and Future Work	83
9.1	Successes	83
9.2	Limitations	84
9.3	Future Work	84
A	Useful Theorems	91
A.1	Parseval's Relation	91
A.2	Kullback-Liebler Distance	91
A.3	Conjugate Gradient Method	92
B	Derivation of Numerical Methods	95
B.1	Newton's Method	95
B.2	Iteratively Re-Weighted Least Squares	97
C	Calibration Setup	99
C.1	Physical Apparatus	99
C.2	Camera Settings	101
C.3	Registration Procedure	102
C.4	Solving for Kernels	104

D Test Scenes	105
D.1 Synthetic Scenes	105
D.2 Planar Scenes	106
D.3 Cluttered Scenes	108
D.3.1 Views	108
D.3.2 Depthmaps	111

List of Figures

2-1	An example of point cloud generated by a LIDAR (Light Detection and Ranging).	19
2-2	The thin-lens model.	21
3-1	The Bayer pattern as used in Canon EOS 10D Mk II.	29
4-1	Various models of the image-generation pipeline. Left: Naive process, compatible with single-channel model. Center: Joint process, compatible with multi-channel model. Right: Sequential process, compatible with single-channel model.	36
4-2	Expected reconstruction error from multi-channel deconvolution with a single filter, versus simulated single-channel deconvolution of pre-demosaicked image. Each point corresponds to a randomly sampled 15-by-15 symmetric binary pattern.	38
4-3	Expected reconstruction error from multi-channel deconvolution with two filters, versus the better result from the two individual filters. Each point corresponds to a pair of randomly sampled 15-by-15 symmetric binary pattern.	39
4-4	Average gain for the expected reconstruction error of the best-performing pair of filters, over that of the best-performing individual filter in an n -tuple. For each n , 100 sets were sampled.	40
4-5	Comparison of expected reconstruction error for individual filters and for pairs of filters. Each point corresponds to an n -tuple of filters, where the coordinates correspond to the score of best individual filter and that of the best pair of filters.	40

4-6	Minimum KL-distance estimate in the single-channel model, for both naive and sequential processes, with the best-fit line.	43
4-7	Average gain for the minimum pairwise KL divergence of the best-performing pair of filters, over that of the best-performing individual filter in an n -tuple. For each n , 100 sets were sampled.	43
4-8	Example of complementary filters. Note that g has its minimum score for scales (k_8, k_9) , while h has its minimum score for scales (k_3, k_4) , and the two filters complement each other in those two scales. As a result, the pair performs satisfactorily on both scales.	44
4-9	Comparison of minimum KL divergence score for individual filters against the case in which they are paired with a pinhole. Each point represents a 15-by-15, binary symmetric filter.	45
5-1	Application of Newton's Method on simple scenarios. Left: The second-order approximation is locally accurate. Right: However, Newton's Method iteratively ascends the gradient in the incorrect direction, because the objective function f is locally concave.	49
5-2	Application of IRLS on simple scenarios. Left: The second-order approximation is locally accurate. Right: IRLS approaches the local minimum in five iterations.	49
5-3	One-dimensional toy scenarios for evaluating numerical methods and multi-channel priors. Each scenario features a 1D image in two channels with one pixel missing in each channel.	52
5-4	Iterative reconstruction result on the scenarios in Figures 5.3(a) through 5.3(d), with GMM or IRLS, and various multi-channel priors. Each row corresponds to a particular scenario, and each column corresponds to a reconstruction method.	53
5-5	Demosaicking results for various two-dimensional images, with GMM or IRLS, and various multi-channel priors. Each row corresponds to a particular input image, and each column corresponds to a reconstruction method.	54
5-6	A two-dimensional image deblurred with multi-channel priors.	56

6-1	Spectral sensitivity of a Canon 10D CMOS sensor. Note the clear overlap between neighboring channels.	58
6-2	Ideal kernels for each channel. Each kernel is a 15-by-15 discrete two-dimensional signal, with binary values.	60
6-3	Spectral transmission characteristics of four color gels that are used in filter fabrication.	61
6-4	Assembly of the multi-channel aperture into the camera lens, with the expected kernels for each channel. Note that the filters overlap to a small extent.	62
6-5	Effective kernels observed from a pinhole light source (60cm away) at multiple distances. Left: regular aperture at f/4.0. Center: single-channel aperture. Right: multi-channel aperture. The last image was taken with longer exposure time to show the colors more clearly. . .	63
7-1	Effect of regularization on depthmap. Top: The captured scene. Bottom left: depthmap constructed from deconvolution error. Bottom right: regularized depthmap.	67
8-1	Effective kernels measured at distances between 2.10m and 3.20m, with 10cm increment. The displayed kernels have been scaled in intensity in order to show the pattern more clearly, and are arranged into a 6-by-2 block per channel. Left: kernels for the single-channel coded aperture. Right: kernels for the multi-channel coded aperture, in red, green, blue channels, respectively.	71
8-2	Depth classification accuracy on two synthetic datasets at various noise level. Left: results on SYNTHETIC1. Right: results on SYNTHETIC2.	72
8-3	Depth classification accuracy on two datasets for linear SVMs of varying parameter. Left: results on dataset BUILDING. Right: results on dataset POSTERS.	75
8-4	Depth classification accuracy on two datasets for normalized deconvolution error.	76
8-5	Depth classification accuracy for pixels of high local entropy. Left: results on dataset BUILDING. Right: results on dataset POSTERS. . . .	76

8-6	Depth and scene recovery on PRINTS.	78
8-7	Depth and scene recovery on SHOES.	79
8-8	Depth and scene recovery on KITCHEN.	79
8-9	Demonstration of simulated re-focusing using the sharp image and depthmap, generated from a single capture. Left: Scene refocused at 3.1m. Middle: Scene refocused at 2.6m. Right: Scene refocused at 2.1m.	80
C-1	Calibration pattern. The pattern consists of random binary noise in two scales.	100
C-2	Physical apparatus for calibration. The apparatus includes a horizontal rail with distance markers, a movable carriage constructed of plastic, a RAW-capable camera.	100
D-1	SYNTHETIC1 test scene.	105
D-2	SYNTHETIC2 test scene.	106
D-3	BUILDING test scene.	107
D-4	POSTERS test scene.	107
D-5	Views of PRINTS test scene.	108
D-6	Views of SHOES test scene.	109
D-7	Views of KITCHEN test scene.	110
D-8	Depthmaps from PRINTS test scene. Top: initial estimate. Bottom: regularized depthmap.	111
D-9	Depthmaps from SHOES test scene. Top: initial estimate. Bottom: regularized depthmap.	112
D-10	Depthmaps from KITCHEN test scene. Top: initial estimate. Bottom: regularized depthmap.	113

List of Tables

2.1	The post-processing algorithm for Levin et al’s single-channel coded-aperture camera.	25
5.1	Objective functions for the three multi-channel priors.	48
5.2	Expectation-Maximization for training the cluster weights and variances corresponding to each multi-channel prior. This algorithm needs to run only once for a given set of parameters.	50
5.3	Expectation-Maximization for iteratively estimating non-linear multi-channel priors.	51
7.1	Implementation of graphcut algorithm to regularize the initial depthmap.	67
8.1	Summary of test scenes prepared for single- and/or multi-channel coded aperture photography.	70
8.2	Results of deconvolution on SYNTHEIC1 at various noise level. . . .	73
8.3	Results of deconvolution on SYNTHEIC2 at various noise level. . . .	74
8.4	Results of deconvolution of planar datasets, at the correct depths. For clarity, we display zoomed-in portions of heavy texture for selected depths, for both single- and multi-channel deconvolution. The blue channel for multi-channel deconvolution is specifically shown, since red and green channels are sharp to begin with.	77
A.1	Algorithm for the Conjugate Gradient method.	93
C.1	Settings for Canon EOS-1D Mk II.	101

C.2	Flags for <code>dcraw</code> RAW conversion. The given settings generate a linear, Bayer-masked image as in the multi-channel model.	101
C.3	Registration algorithm for aligning images captured at varying depths.	103
C.4	Quadratic Programming for inferring the kernels from the blurred scene, along with the groundtruth.	104

Chapter 1

Introduction

Traditionally, a camera system records the three-dimensional world in a two-dimensional representation. Information is lost, inevitably, as the representation is more compact than the space it seeks to capture. The axis that the camera discards is the depth, the distance to the objects present in the scene. Furthermore, objects that are not “in focus” are blurred, indicating reduction in resolution. Granted, the latter effect is considered artistic, and is often necessary in providing perceptual cues as to which objects are important in the scene. Nonetheless, if the entire three-dimensional world is captured, the output of a conventional camera could be simulated synthetically. Not only could we replicate the effects of a conventional camera, but also an array of post-processing effects such as post-exposure focusing, automatic scene segmentation, alternate viewpoint rendering would be possible.

Indeed, as cameras become more advanced and ubiquitous, there has also been commensurate effort to gradually expand the range of information captured. To date, several systems or methods have been proposed that can simultaneously capture an all-focus image of the scene and extract a depth map. Unfortunately, these efforts are accompanied by certain tradeoffs. Many have relied on developing specialized optical hardware, which is expensive; others rely on pure post-processing algorithms such as matting or stereo vision, which are yet to be perfected and only yield coarse results.

More recent approaches rely on capturing a two-dimensional imagery of the scene while attempting to encode as much depth information as possible. Then, some

post-processing framework is employed to “recover” either the three-dimensional geometry of the scene, or the two-dimensional scene with a depth map. Nevertheless, the same tradeoff between cost and effectiveness has applied [15].

1.1 Coded-Aperture Photography

Coded-aperture photography is one such class of approach that seeks to minimize these tradeoffs. It is a simple modification to a conventional camera, achieved by introducing a patterned occluder within the aperture of the camera lens. The pattern controls how the scene is blurred, and is designed in a way that facilitates the extraction of both depth and original scene from the blurred output. A probabilistic model is employed to recover an estimate for the scene and the depth map. The version developed by Levin et al [14] demonstrates the viability of coded-aperture photography. However, it stipulates that the image output by the camera is a single-channel two-dimensional signal, which masks the presence of multiple (R,G,B) channels and also, perhaps more critically, the presence of the Bayer filter.

1.2 Contributions of This Work

In this work, we generalize the coded-aperture camera to discard several simplifying assumptions. Primarily, instead of relying on a single code to filter all light entering the camera, we seek to filter the three primary colors separately. We introduce the presence of a Bayer filter to more realistically model the imaging process, and derive the reconstruction algorithm for such multi-channel coded aperture. We study the correlations among multiple channels to enhance the performance of the camera, and discuss several numerical methods for recovering the scene. As before, the multi-channel coded-aperture system yields an all-focus image and a layered depth map. Lastly, we prototype a physical coded aperture and present results on several test scenes, comparing the performance to that of the single-channel coded aperture.

Chapter 2

Related Work

2.1 Active versus Passive Methods

Although the most basic imaging devices, such as a pinhole camera, were conceived as early as the 5th century B.C., extracting information from our surroundings and manipulating it have been an active area of research recently, with the advent of computational photography. There exists several optical systems and methods to understand the three-dimensional world around us, and they can largely be divided into *active* and *passive* techniques.

Active techniques utilize specialized illumination sources in certain spectrum, and involve specialized hardware systems to scan the scenery actively. Laser range finder[3] or LIDARs (Light Detection and Ranging) fall in this category, and they extract depth information in the form of a point cloud, which indicates the distance to objects at particular angles as the sensor spins to cover the scenery. The resulting point cloud can be visualized as in Figure 2-1. Active techniques have been successfully used in mapping terrains[12], cityscapes or other immediate surrounding. However, they are costly and do not yield imagery of the scene without

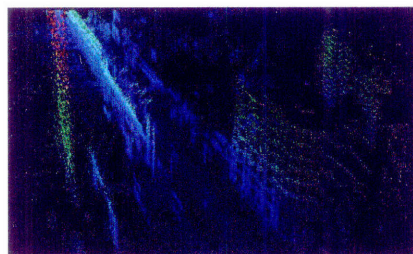


Figure 2-1: An example of point cloud generated by a LIDAR (Light Detection and Ranging).

additional hardware.

Passive techniques do not rely on sophisticated hardware, but instead perform mathematical analysis of the captured image. Among these, *monocular* methods operate on a single image signal captured at a given viewpoint. They tend to be purely algorithmic and implemented entirely on software. One example is matting[6][16], which seeks to find a very rudimentary background-foreground separation. Matting is robust and efficient, but the binary depth map is limited, and does not address blurs in the image. More recently, machine-learning methods that infer more substantial 3-D structure[23] using Markov Random Fields and supervised learning have been published, but similar limitations on coarseness and blurred images remain. The monocular method can be expanded to take multiple images from the same viewpoint at different focus settings, which allows computation of depth through analysis of the amount of *defocus*[5][8][10][21]. Because the defocus is a function of depth, the depth at each pixel can be inferred. While the method is robust and yields the desired information, namely an all-focus image and a depth map, it requires the user to take multiple photographs, which scales linearly to the resolution of the depth map.

The other major class of passive techniques is the set of *binocular* methods. These analyze images taken at two or more viewpoints to infer occlusion and depth. A detailed survey and analysis of these “stereo” algorithms are available[24]. In the same spirit, *Plenoptic* cameras use microlens arrays that collect light rays arriving from different directions[2][7][17][19], achieving the effect of multiple viewpoints, though at the cost of sacrificing spatial resolution. The number of simulated viewpoint corresponds to the factor of reduction in spatial resolution. In addition, the microlens arrays must be manufactured and installed in the camera.

Lastly, it has been possible to achieve illusion of multiple focus in a monocular method by introducing a non-conventional aperture[14][25]. These methods have the advantage that only a single photograph is required, and that they do not suffer reduction in the resolution of the photograph. In particular, we discuss [14] in considerable detail in Section 2.2, as it lays foundation to our contributions in subsequent chapters.

2.2 Single-Channel Coded-Aperture

The imaging process within a camera can be modeled in a simple framework that allows us to capture the effect of incorporating a coded aperture, when the goal of the camera is to recover both the scene itself and its depth. Levin et al[14] adopts the thin-lens model for the image generation within a camera, assuming that the camera captures a single two-dimensional image, for the sake of simplicity and elegance; in reality, modern cameras typically output signals in three channels—red, green, blue.

2.2.1 Thin-Lens Model for Optics

A typical digital SLR camera with an accompanying lens sports multiple glass elements with complex optical pathways, which are difficult to analyze altogether. Also, the lens demonstrates an array of particular phenomena, such as chromatic aberration, spherical aberration, diffraction. Rather than grappling with the full optical model, we deal with the thin-lens model for optics, which vastly simplifies the imaging process yet preserves the important parameters that affect the output. The

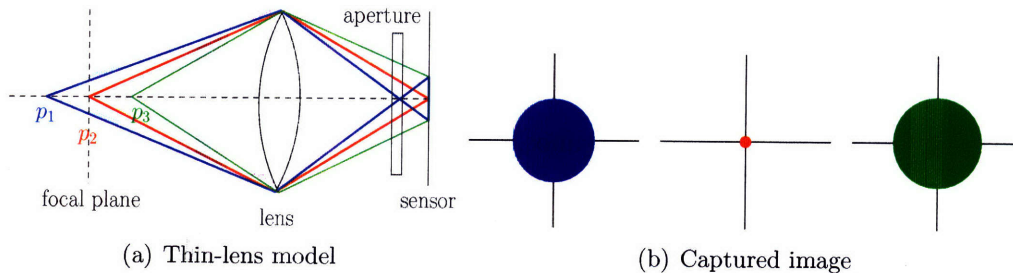


Figure 2-2: The thin-lens model.

thin-lens model is characterized by its focal distance, the object distance (depth), and the aperture size. It admits a two-dimensional signal in a single channel (say, grayscale). The light from the object enters the lens and converges toward the sensor, with the convergence being perfect when the object is at the focal distance. When the object distance differs, the light from the object fails to converge but instead strikes the sensor in a wider area that corresponds to the shape of aperture, which is the hole through which the light passes. The resulting images are blurred as shown in Figure 2-2, assuming a circular aperture.

The radius of the projection is in fact a function of the depth. Since a point light source projects onto a well-defined pattern, the projection can be modeled as a two-dimensional convolution. Let x be a 2D planar object at depth k , where the depth is chosen from a finite set K , and let f_k be the projection of a point light source at depth k , which we shall refer to as the *kernel*. Then, the resulting image y is,

$$y = (f_k * x) + \eta, \quad (2.1)$$

where $\eta \sim N(0, \sigma^2)$ is a two-dimensional signal of the same dimensions representing pixel-wise independently generated zero-mean Gaussian noise with variance σ^2 .

2.2.2 Image and Depth Recovery

Since the image generation is known, a simple probabilistic framework can be employed to recover the desired information. Given the output signal y and the depth k , we recover the input image x to be the maximum-likelihood estimate. Also, given the output signal y , we recover the depth k to be the maximum-likelihood estimate as well:

$$\hat{x} = \operatorname{argmax}_x P_k(x; y),$$

$$\hat{k} = \operatorname{argmax}_{k \in K} P_k(y).$$

Applying Bayes' Rule to the first and expanding the second over x , we obtain

$$\hat{x} = \operatorname{argmax}_x P_k(y|x)P(x), \quad (2.2)$$

$$\hat{k} = \operatorname{argmax}_{k \in K} \int P_k(y|x)P(x). \quad (2.3)$$

Equations (2.2) and (2.3) require us to compute $P_k(y|x)$ and $P(x)$. The first can be inferred from the image generation process in Equation (2.1):

$$P(y|x) \propto \exp - \frac{\|f_k * x - y\|^2}{2\sigma^2}. \quad (2.4)$$

What remains is $P(x)$, the prior on the input images. Levin et al selects the sparse prior for natural images, which is discussed below.

2.2.3 Sparse Prior

Natural images tend to follow a particular distribution; they are not simply white noise. They exhibit a statistical property that spatial derivatives on natural images are sparse[20]. In other words, we assume that x is drawn from the following zero-mean, heavy-tailed distribution:

$$P(x) \propto \prod_{i,j} \exp -\frac{\alpha}{2} [(x(i,j) - x(i+1,j))^\rho + (x(i,j) - x(i,j+1))^\rho], \quad (2.5)$$

where $0 < \rho < 1$ and α is set to match the correct variance observed from data. The assumption that the derivatives have heavy-tailed distribution is called the *sparse prior*. However, to make the analysis tractable, we also consider the Gaussian form in which $\rho = 2$:

$$P(x) \propto \prod_{i,j} \exp -\frac{\alpha}{2} [(x(i,j) - x(i+1,j))^2 + (x(i,j) - x(i,j+1))^2]. \quad (2.6)$$

These priors can be expressed in the frequency domain as well, via Parseval's Relation (See Equation (A.1) in appendix.) Let G^1 and G^2 be the convolution matrices in frequency domain corresponding to the two directional derivatives $\begin{bmatrix} 1 & -1 \end{bmatrix}$ and $\begin{bmatrix} 1 & -1 \end{bmatrix}^T$, and let uppercase characters represent the Fourier transforms of the lowercase variables, when applicable. Then,

$$P(X) \propto \exp -\frac{\alpha}{2n} X^T \Psi^{-1} X, \quad \text{where } \Psi = \text{diag} (\|G^1(v,w)\|^2 + \|G^2(v,w)\|^2)^{-1}. \quad (2.7)$$

Equation (2.4) can be similarly rewritten:

$$P(Y|X) \propto \exp -\frac{\|F_k \cdot X - Y\|^2}{2n\sigma^2}, \quad (2.8)$$

where F_k is the diagonal matrix with the Fourier transform of f_k along its diagonal.

Now it is possible to solve Equations (2.2),(2.3) entirely in Fourier domain. The fact that all probabilities are Gaussian vastly facilitates the ensuing optimizations, which are convex.

While the Gaussian prior does lead to a convex optimization with closed-form

solutions, it is necessary to directly solve the sparse prior, which is a more realistic model of natural images: while a Gaussian prior prefers to distribute derivatives equally over the image, leading to gradual edges, the sparse prior forces the derivative to fall over a smaller number of pixels. Thus, in order to solve Equation (2.5) when $\rho < 1$, Levin et al employs an iterative process called Iteratively Re-Weighted Least Squares (IRLS), which is a non-linear optimization technique. See Appendix B.2.

2.2.4 Filter Selection

Levin et al suggests how the filter f_k should be selected as to maximize the likelihood of accurate depth extraction. In essence, we would like the distributions of the output Y to differ as much as possible as the scale k varies. In other words, we want to maximize $D(P_{k_1}(Y), P_{k_2}(Y))$, where D is some measure of distance between two distributions. Letting D be the Kullback-Leibler (KL) divergence, we obtain

$$D_{KL}(P_{k_1}(Y), P_{k_2}(Y)) = \int P_{k_1}(\log P_{k_1}(Y) - \log P_{k_2}(Y))dy.$$

It can be shown that the above quantity equals

$$D_{KL}(P_{k_1}(Y), P_{k_2}(Y)) = -1 + \sum_{v,w} \left(\frac{\sigma_{k_1}(v,w)}{\sigma_{k_2}(v,w)} - \log \frac{\sigma_{k_1}(v,w)}{\sigma_{k_2}(v,w)} \right),$$

where $\sigma_{k_i}(v,w) = \sigma^2 + \frac{1}{\alpha} \|F_{k_i}(v,w)\|^2 \bar{\Psi}(v,w)^{-1}$.

Then, one may search a subset of all possible filters—Levin et al focused on binary filter that can be constructed on a single layer of cardboard—that minimizes $\max_{k_1, k_2} D_{KL}(P_{k_1}(Y), P_{k_2}(Y))$, meaning that the minimum distance between any two scales is maximized.

2.2.5 Implementation

Once a filter is selected, a patterned occluder is created based on the negative of the filter. The occluder is inserted into the camera lens, and the camera produces y as in the aforementioned model. Table 2.1 gives the full post-processing algorithm used by Levin at al, which then extracts the scene and depth estimate.

- 1: For each $k \in K$,
- 2: Compute \widehat{x}_k from y using IRLS and the sparse prior.
- 3: For each pixel location (i, j) ,
- 4: For each $k \in K$,
- 5: Take a 60-by-60 window and compute the mean deconvolution error at each scale, weighted by some learnt coefficients: $\lambda_k \|f_k * \widehat{x}_k - y\|^2$.
 This serves as an estimate of $P_k(y)$ near (i, j) .
- 6: Pick the depth k that minimizes the above deconvolution error. Set $\Theta(i, j) = k$.
- 7: Regularize the depth map Θ using a graph-cut algorithm[4].
- 8: Output \widehat{x} where $\widehat{x}(i, j) = \widehat{x}_{\Theta(i, j)}(i, j)$.

Table 2.1: The post-processing algorithm for Levin et al’s single-channel coded-aperture camera.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 3

Mathematical Model

Levin et al[14] introduced the mathematical theory for the coded aperture and demonstrated its viability in deblurring the observed image and inferring a depth map. We extend the model and analysis in order to account for the existence of multiple channels and the presence of Bayer filter, rederiving the necessary formulae. These modifications in fact generalizes the previous model, and provides important theoretical justification for the fabrication and implementation of multi-channel coded-aperture.

3.1 Notations

3.1.1 Signal Representation

A camera system is in essence a signal processing unit, where the input signal corresponding to the scenery undergoes certain transforms to generate an output signal that we observe. These images we deal with have their native representations as triples of discrete two-dimensional signals, each component corresponding to one of red, green, blue channels, in that order. We denote the input signal as $x = \{x^R, x^G, x^B\}$, and the output signal as $y = \{y^R, y^G, y^B\}$. We shall generally use R, G, B in superscript denote the respective channel as a component of a multi-channel entity, with C in superscript as a variable to index them. Furthermore, for each discrete two-dimensional signal in lower case, the upper-case version is reserved for its discrete Fourier transform in two dimensions (e.g. X^R).

Throughout our analysis, the signals must be treated as column vectors in order to facilitate linear-algebraic manipulations and analyses. Therefore, we denote by $M(:)$ the column vector formed by concatenating all the columns of M in order, where M is a two-dimensional signal. However, as we will shortly see, this conversion scheme is not very convenient in dealing with masks such as Bayer filter, so we address this issue by introducing another column-vector representation, denoted by \overline{M} . First, we presume that all two-dimensional signals have even widths and heights. This enables us to partition M into equally-sized 2-by-2 blocks M_1, M_2, M_3, M_4 , numbered from top to bottom, left to right:

$$M = \begin{bmatrix} M_1 & M_2 \\ M_3 & M_4 \end{bmatrix}, \quad (3.1)$$

where M_1, \dots, M_4 are matrices with dimensions that are half of those of M . Now we flatten M_1, \dots, M_4 into column vectors and concatenate them:

$$\overline{M} := \begin{bmatrix} M_1 \\ M_2 \\ M_3 \\ M_4 \end{bmatrix}.$$

When M is a triple of two-dimensional signals, having the form $M = \{M^R, M^G, M^B\}$, then we further stipulate that,

$$\overline{M} := \begin{bmatrix} \overline{M^R} \\ \overline{M^G} \\ \overline{M^B} \end{bmatrix}. \quad (3.2)$$

3.1.2 Other Notations

Other notations in this thesis closely adhere the well-known standard mathematical notations, with the possible exception of the following shorthands: \cdot^* denotes component-wise multiplication; ∂_1 and ∂_2 denote the partial derivative operator on discrete two-dimensional signals along the two dimensions, respectively (e.g. $\partial_1 x^R(i, j) = x^R(i, j) - x^R(i, j + 1)$.) Next, any single-variate function operating

on the real numbers or its subset, when applied to a vector or matrix, is interpreted to work component-wise, unless existing conventions dictate otherwise. Finally, $\text{diag}(\dots)$ is the diagonal matrix with the operand along the diagonal, in case the operand is a column vector or a comma-delimited sequence of square matrices; if the operand is a square matrix, it returns its diagonal. An alternative form $\text{diag}_i(\dots)$ with the operand as a function $f(i)$ is equivalent to $\text{diag}(f(1), f(2), \dots)$.

3.2 Color Filter Array

Most commercial digital cameras generate images in RGB format, composed of three discrete two-dimensional signals representing scene intensity in red, green and blue. However, most sensors deployed within the cameras do not record three values at each pixel location, with the notable exception of Foveon X3 sensors¹. In fact, the sensor records one value at each pixel location, which captures the intensity in either red, green, *or* blue light. The appropriate channel is determined by a preset pattern called the Bayer pattern, as in Figure 3-1. The missing values at each location are later interpolated from neighboring pixels, in a process known as *demosaicking*.

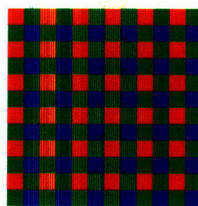


Figure 3-1: The Bayer pattern as used in Canon EOS 10D Mk II.

We can model the effect of Bayer filter as a component-wise multiplication ap-

¹See <http://www.foveon.com>.

plied to the full output image, where the matrices to be multiplied with are,

$$\omega^R = \begin{bmatrix} 1 & 0 & 1 & 0 & \dots \\ 0 & 0 & 0 & 0 & \\ 1 & 0 & 1 & 0 & \\ 0 & 0 & 0 & 0 & \\ \vdots & & & & \ddots \end{bmatrix}, \quad \omega^G = \begin{bmatrix} 0 & 1 & 0 & 1 & \dots \\ 1 & 0 & 1 & 0 & \\ 0 & 1 & 0 & 1 & \\ 1 & 0 & 1 & 0 & \\ \vdots & & & & \ddots \end{bmatrix}, \quad \omega^B = \begin{bmatrix} 0 & 0 & 0 & 0 & \dots \\ 0 & 1 & 0 & 1 & \\ 0 & 0 & 0 & 0 & \\ 0 & 1 & 0 & 1 & \\ \vdots & & & & \ddots \end{bmatrix}, \quad (3.3)$$

respectively, for red, green, blue channels. Since multiplication is equivalent to convolution in the dual domain, the effect of Bayer filter in the Fourier domain is to convolve with

$$\Omega^R = \frac{1}{4} \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}, \quad \Omega^G = \frac{1}{4} \begin{bmatrix} 2 & 0 \\ 0 & -2 \end{bmatrix}, \quad \Omega^B = \frac{1}{4} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}, \quad (3.4)$$

where $\Omega^R, \Omega^G, \Omega^B$ are matrices of the same dimensions with up to four nonzero entries found at the top-left corner of the four subsquares. Note that the column-vector representation becomes useful, as the convolution of a two-dimensional signal M with the above matrices can be rewritten as a multiplication:

$$\overline{\Omega^R * M} = \frac{1}{4} \begin{bmatrix} I_{n/4} & I_{n/4} & I_{n/4} & I_{n/4} \\ I_{n/4} & I_{n/4} & I_{n/4} & I_{n/4} \\ I_{n/4} & I_{n/4} & I_{n/4} & I_{n/4} \\ I_{n/4} & I_{n/4} & I_{n/4} & I_{n/4} \end{bmatrix} \overline{M},$$

$$\overline{\Omega^G * M} = \frac{1}{4} \begin{bmatrix} 2I_{n/4} & 0 & 0 & -2I_{n/4} \\ 0 & 2I_{n/4} & -2I_{n/4} & 0 \\ 0 & -2I_{n/4} & 2I_{n/4} & 0 \\ -2I_{n/4} & 0 & 0 & 2I_{n/4} \end{bmatrix} \overline{M},$$

$$\overline{\Omega^B * M} = \frac{1}{4} \begin{bmatrix} I_{n/4} & -I_{n/4} & -I_{n/4} & I_{n/4} \\ -I_{n/4} & I_{n/4} & I_{n/4} & -I_{n/4} \\ I_{n/4} & -I_{n/4} & -I_{n/4} & I_{n/4} \\ -I_{n/4} & I_{n/4} & I_{n/4} & -I_{n/4} \end{bmatrix} \overline{M}.$$

We denote these coefficient matrices by $\Omega_*^R, \Omega_*^G, \Omega_*^B$, respectively.

3.3 Revised Thin-Lens Model for Optics

The thin-lens model in Levin et al[14] adequately captures the necessary parameters even with multiple channels, since the channels arise simply because the model processes multiple wavelengths in parallel. However, our fully multi-channel model now affords us the ability to vary the kernels independently across the channels. We denote the three kernels at depth k by f_k^R, f_k^G, f_k^B , respectively. Equation (3.5) captures the image generation process using these kernels.

$$y = \omega^R * (f_k^R * x^R) + \omega^G * (f_k^G * x^G) + \omega^B * (f_k^B * x^B) + \eta. \quad (3.5)$$

Alternatively, in the Fourier domain,

$$\overline{Y} = \Omega_*^R (\overline{F_k^R} * \overline{X^R}) + \Omega_*^G (\overline{F_k^G} * \overline{X^G}) + \Omega_*^B (\overline{F_k^B} * \overline{X^B}) + \overline{\eta},$$

which we rewrite to

$$\overline{Y} = A_k \overline{X} + \overline{\eta}, \quad \text{where } A_k = \begin{bmatrix} \Omega_*^R & \Omega_*^G & \Omega_*^B \end{bmatrix} \text{diag} \left(\begin{bmatrix} \overline{F_k^R} \\ \overline{F_k^G} \\ \overline{F_k^B} \end{bmatrix} \right), \quad (3.6)$$

by treating the multi-channel signal as a column vector, as in Equation (3.2).

With this model, the original image may be recovered in the same fashion as in the single-channel case, utilizing maximum-likelihood with some intelligent prior.

3.4 Prior for Multi-Channel Natural Images

The sparse prior is critical to the deconvolution in the single-channel case, as it tends to generate images that are statistically likelier. A statistical model for multi-channel natural images could similarly be incorporated into the multi-channel deconvolution, in order to aid the process of deconvolution. In this thesis, we propose and test three priors for multi-channel natural images: independent prior, independent prior with change of basis, and dependent prior.

Unfortunately, none of these priors lend themselves to convenient conversion into Fourier domain. For the purpose of analyzing the priors and expected reconstruction errors in the subsequent chapters, we utilize the Gaussian prior on each channel independently. We also table the comparison among the priors presented in this section until the necessary numerical machinery for performing deconvolution with them is developed in Chapter 5.

3.4.1 Independent Prior

The prior for a single-channel case from Levin et al[14] can be extended to a multi-channel image simply by introducing the assumption that each channel is independently generated: $P(X) = P(X^R)P(X^G)P(X^B)$. In turn, the distribution of each channel follows the sparse prior, resulting in Equation (3.7)

$$P(X) \propto \prod_{C,i} \exp \left(-\frac{\alpha}{2n} \left\{ |g^1 \overline{x^C}(i)|^\rho + |g^2 \overline{x^C}(i)|^\rho \right\} \right), \quad (3.7)$$

where g^1, g^2 are appropriate square matrices that generate first-order spatial derivatives in either orientation. We term this distribution the *independent* prior. The independent prior simplifies the analysis and testing, and lends itself to easy comparison between an unconstrained multi-channel coded aperture and a single-channel coded aperture.

3.4.2 Independent Prior with a Change of Basis

In reality, the RGB channels are highly correlated, with edges often co-occurring at the same spatial location. Thus we seek alternative representations of RGB images

that can separate such dependencies. In fact, there does exist other color-space representations such as YUV and CYMK, which are simply linear changes of basis. They are each characterized by a p -by-3 matrix P , where each row corresponds to a channel in the color space, and that particular channel equals the linear sum of RGB channels, weighted by the entries in the row. Particularly, the YUV color space is meant to capture perceptually independent channels, satisfying the underlying assumption in the independent prior. Therefore, we can enforce the independent prior on these alternate color-space representations, thereby addressing the dependencies among RGB channels. Equation (3.8) details the resulting prior.

$$P(X) \propto \prod_{l=1}^p \prod_i \exp \left(-\frac{\alpha}{2n} \left\{ |g^1 \overline{x^{P_l}}(i)|^\rho + |g^2 \overline{x^{P_l}}(i)|^\rho \right\} \right),$$

where $x^{P_l} = P(l, 1)x^R + P(l, 2)x^G + P(l, 3)x^B$. (3.8)

This prior can also be considered to be an attempt to capture the dependencies between RGB channels through several linear combinations. In practice, P does not necessarily require full rank, as dependent rows serve as redundancy.

3.4.3 Dependent Prior

Lastly, we treat each spatial derivative as a three-dimensional column vector, where each component corresponds to the value in one of the three channels, and impose the sparse prior on the magnitude of the vector. Intuitively, this *dependent* prior penalizes spatially separating edges in multiple channels.

$$P(X) \propto \prod_{i,j} \exp \left(-\frac{\alpha}{2n} \left\{ (g^j \overline{x^R}(i))^2 + (g^j \overline{x^G}(i))^2 + (g^j \overline{x^B}(i))^2 \right\}^{\frac{\rho}{2}} \right). \quad (3.9)$$

The dependent prior could be combined with the change of basis, as with the independent prior, but its formula is omitted here for the sake of brevity.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 4

Analysis of Model

The mathematical model set forth in Chapter 3 admits a thorough analysis in terms of reconstruction, expected error and depth discrimination, using the Gaussian prior. Our main objective is to derive the formula for reconstruction, and the criteria for evaluating a set of filters $\{f_k^R, f_k^G, f_k^B\}$. In this chapter, we derive those formulae and compare the multi-channel model with its single-channel counterpart.

4.1 Modelling the Pipeline

In comparing the single-channel model with the multi-channel model, we adopt the following terminology to distinguish the possible differences in the imaging process. The *naive* process assumes that no Bayer pattern exists, and is compatible with the conventional single-channel model. The *joint* process assumes that the resulting signal has passed the Bayer filter, and is compatible with the multi-channel model. Finally, the *sequential* process assumes that the resulting signal has passed the Bayer filter, but has been demosaicked with the weighted nearest-neighbor algorithm, thereby rendering it compatible with the single-channel model. The motivation for the sequential model is to enable direct comparison between the single-channel and multi-channel deconvolution. While the original single-channel coded aperture assumes the naive process, the image has in fact undergone the sequential process. Therefore, a fair comparison would be between the joint process with multi-channel model, and the sequential process with single-channel model.

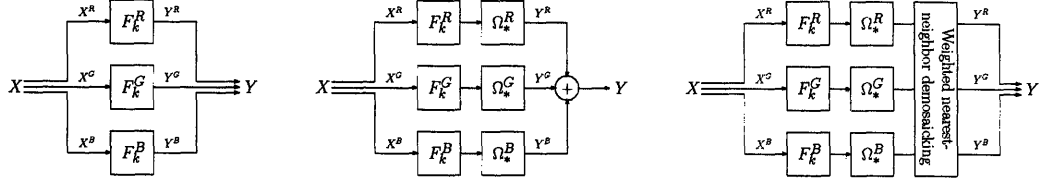


Figure 4-1: Various models of the image-generation pipeline. Left: Naive process, compatible with single-channel model. Center: Joint process, compatible with multi-channel model. Right: Sequential process, compatible with single-channel model.

4.2 Reconstruction of Multi-Channel Image

The reconstruction problem is to estimate the input image x given an observation y through the coded aperture $\{f_k^R, f_k^G, f_k^B\}$ at a fixed depth k . As with the single-channel case, the reconstruction problem admits a straightforward estimate through a simple probabilistic framework. From Equation (3.6), the conditional probability distribution for the output, given input, is

$$P(y|x) \propto \exp \left\{ - \frac{\|A_k \bar{X} - \bar{Y}\|^2}{2n\sigma^2} \right\},$$

while the prior on the input, assuming the independent Gaussian prior to render analysis feasible,

$$\begin{aligned} P(x) &\propto \exp \left\{ - \frac{\alpha}{2n} \left(\bar{X}^R T \Psi^{-1} \bar{X}^R + \bar{X}^G T \Psi^{-1} \bar{X}^G + \bar{X}^B T \Psi^{-1} \bar{X}^B \right) \right\} \\ &= \exp \left\{ - \frac{\alpha}{2n} \left(\bar{X}^T \Psi'^{-1} \bar{X} \right) \right\} \text{ where } \Psi' = \text{diag} (\Psi, \Psi, \Psi). \end{aligned}$$

Then, by Bayes' Rule, the joint probability distribution of the input and output is,

$$P(x, y) \propto \exp \left\{ - \frac{\|A_k \bar{X} - \bar{Y}\|^2}{2n\sigma^2} - \frac{\alpha}{2n} \left(\bar{X}^T \Psi'^{-1} \bar{X} \right) \right\}$$

The maximum-likelihood estimate of x follows:

$$\widehat{X}_k = \underset{X}{\text{argmax}} \exp - \frac{\|A_k \bar{X} - \bar{Y}\|^2}{2n\sigma^2} - \frac{\alpha}{2n} \bar{X}^T \Psi'^{-1} \bar{X} \quad (4.1)$$

$$= \operatorname{argmin}_X \|A_k \bar{X} - \bar{Y}\|^2 + \alpha \sigma^2 \bar{X}^T \Psi'^{-1} \bar{X} \quad (4.2)$$

$$= \operatorname{argmin}_X \bar{X}^T A_k^T A_k \bar{X} - 2 \bar{X}^T A_k^T \bar{Y} + \alpha \sigma^2 \bar{X}^T \Psi'^{-1} \bar{X} \quad (4.3)$$

$$= (A_k^T A_k + \alpha \sigma^2 \Psi'^{-1})^{-1} (A_k^T \bar{Y}). \quad (4.4)$$

Deconvolving at depth k has now been reduced to a quadratic minimization problem posed in Equation (4.3), which yields the solution in Equation (4.4).

4.2.1 Expected Reconstruction Error

The reconstruction of the input signal relies on a statistical estimate, which exhibits bias and variance. The quality of the deconvolution process can thus be measured by the squared error, namely $E_k = \frac{\|\widehat{X}_k - X\|^2}{n}$. From Equation (4.4), it follows immediately that

$$\frac{\|\widehat{X}_k - X\|^2}{n} = \frac{\|((A_k^T A_k + \alpha \sigma^2 \Psi'^{-1})^{-1} A_k^T A_k - I)X + ((A_k^T A_k + \alpha \sigma^2 \Psi'^{-1})^{-1} A_k^T \eta)\|^2}{n}.$$

Let $\Gamma_k = (A_k^T A_k + \alpha \sigma^2 \Psi'^{-1})^{-1}$. By the independence of the two expressions within the squared term, the expected squared error takes the form:

$$E \left[\frac{\|\widehat{X}_k - X\|^2}{n} \right] = E \left[\frac{\|(\Gamma_k A_k^T A_k - I)X\|^2}{n} \right] + E \left[\frac{\|\Gamma_k A_k^T \eta\|^2}{n} \right].$$

Incorporating the known priors for X and η , we obtain

$$E \left[\frac{\|\widehat{X}_k - X\|^2}{n} \right] = \alpha \sigma^4 \operatorname{diag} \left(\Psi'^{-1} \Gamma_k^T \Gamma_k \right) + \sigma^2 \sum \left((A_k \Gamma_k^T \Gamma_k A_k^T) .* \operatorname{diag} (\Omega_*^R, \Omega_*^G, \Omega_*^B) \right) \quad (4.5)$$

4.2.2 Effect of Multiple-Channel Model

The reconstruction error from Equation (4.5) cannot be directly compared to the error from the single-channel model, simply because they operate on images with different priors (and different numbers of filters), and also on different stages of the pipeline; the single-channel model assumes that the image has not passed through the Bayer filter. To enable a comparison, we isolate each difference and test it

independently using the two models.

Presence of Bayer Filter

To analyze the aspect of the new model that addresses the presence of Bayer filter, we consider a multi-channel image from the independent Gaussian prior, blurred using a single kernel f and masked with the Bayer pattern. For the multi-channel model, the expected reconstruction error is given in Equation (4.5), where $f_k^R = f_k^G = f_k^B = f$. Meanwhile, for the single-channel model, we adopt the sequential process with the same kernel.

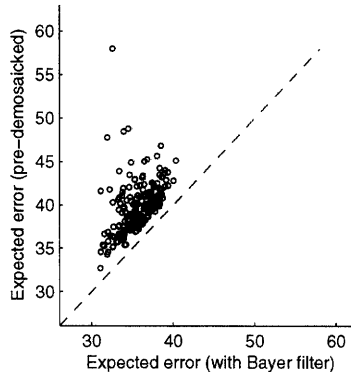


Figure 4-2: Expected reconstruction error from multi-channel deconvolution with a single filter, versus simulated single-channel deconvolution of pre-demosaicked image. Each point corresponds to a randomly sampled 15-by-15 symmetric binary pattern.

When f is sampled many times from 15-by-15 symmetric binary patterns, the joint deconvolution with Bayer filter consistently outperforms the sequential process in which the image is pre-demosaicked, as seen in Figure 4-2.

Effect of Multiple Filters

The effect of multiple filters can be isolated by comparing the full multi-channel model with a more constrained version in which $f_k^R = f_k^G = f_k^B$. However, because the independent Gaussian prior does not at all relate the channels, we impose that the three channels are in fact equal. Therefore, even with the Bayer filter, the output is fully observed in the sense that every blurred pixel is available. We mind

the readers that this assumption should exaggerate the benefit of the multi-channel model.

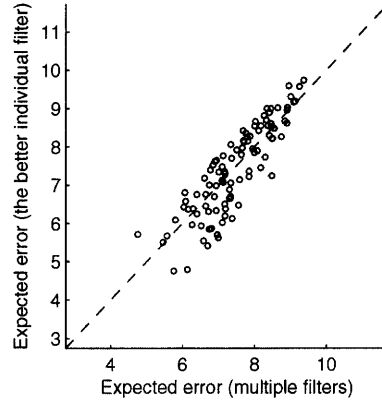


Figure 4-3: Expected reconstruction error from multi-channel deconvolution with two filters, versus the better result from the two individual filters. Each point corresponds to a pair of randomly sampled 15-by-15 symmetric binary pattern.

More formally, we sampled pairs of filters f_1, f_2 , and first compute the expected reconstruction error in the two cases where $f_k^R = f_k^G = f_k^B = f_1$ and $f_k^R = f_k^G = f_k^B = f_2$, respectively. These are equivalent to the single-channel case, since both the images and filters are identical across the channels. When the better of the two errors is compared to the expected reconstruction error in case $f_k^R = f_k^B = f_1$ and $f_k^G = f_2$ hold simultaneously, we see that combining filters improves performance some of the time—see Figure 4-3. However, because filters are matched randomly rather than ones that might be complementary, the observed rate of improvement is underestimated.

We extend this experiment to n -tuples of filters f_1, f_2, \dots, f_n , where we compare the performance of the best individual filter, and the performance of the best pair of filters. As n increases, the likelihood of finding a complementary filter, if such filter exists, scales as well. We find that when n is large enough, the best pair more consistently outperforms the best individual filter, as shown in the upward movement of the datapoints in Figure 4-5, as n increases. The average gain of the best pair over the best individual filter in the n -tuple is also shown in Figure 4-4.

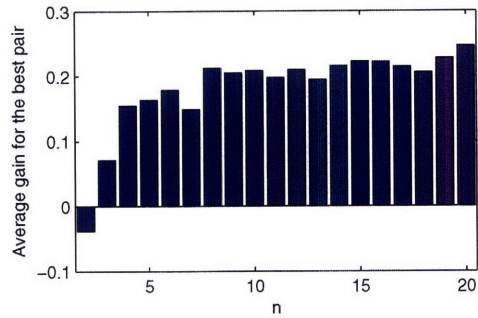


Figure 4-4: Average gain for the expected reconstruction error of the best-performing pair of filters, over that of the best-performing individual filter in an n -tuple. For each n , 100 sets were sampled.

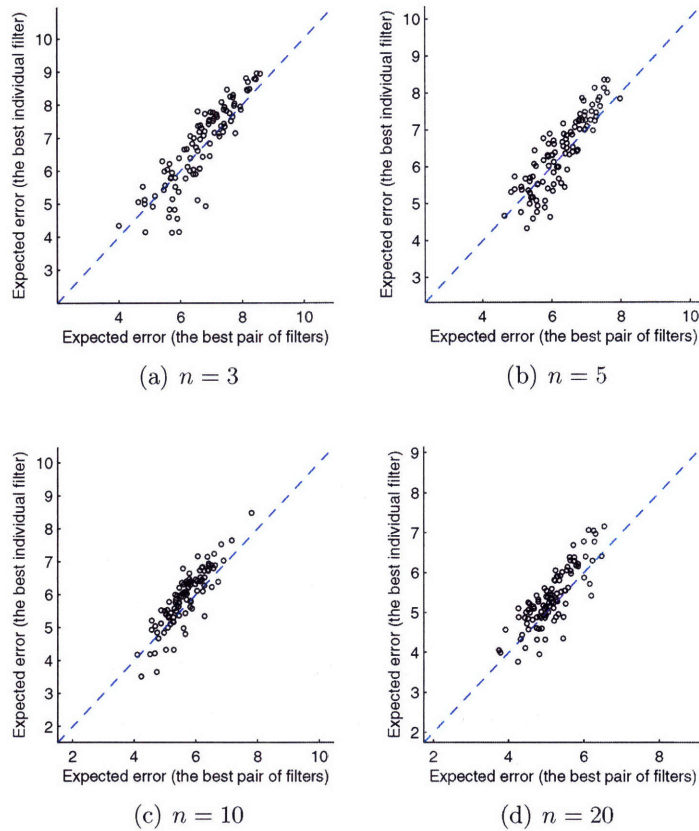


Figure 4-5: Comparison of expected reconstruction error for individual filters and for pairs of filters. Each point corresponds to an n -tuple of filters, where the coordinates correspond to the score of best individual filter and that of the best pair of filters.

4.3 Depth Discrimination

We pose the depth discrimination problem in the manner of Levin et al[14]: given a finite set K of possible depths and filters f_k^R, f_k^G, f_k^B for all $k \in K$, find the likeliest depth $\hat{k} \in K$ for an observed image y . Ideally, we seek the maximum-likelihood estimate

$$\hat{k} = \operatorname{argmax}_{k \in K} P_k(y).$$

In practice, $P_k(y)$ is difficult to solve analytically, as the non-linearity in the prior compounds the existing complexity. One can marginalize the distribution over x , and approximate it with $P_k(y|\hat{x}_k)$, which scales exponentially with the reconstruction error. Accounting for the coefficients, we may instead solve for

$$\hat{k} = \operatorname{argmin}_{k \in K} \lambda_k \|A_k \widehat{X}_k - \bar{Y}\|^2.$$

4.3.1 Filter Selection Criterion

In order for the maximum-likelihood estimate to be as accurate as possible, we seek to maximize the pairwise distances among the distributions $\{P_k(y) \mid k \in K\}$. We recall that $\bar{X} \sim N\left(0, \frac{n\Psi'}{\alpha}\right)$, which gives rise to

$$\begin{aligned} \bar{Y}_k = A_k \bar{X} + \eta &\sim N\left(0, \frac{nA_k \Psi' A_k^T}{\alpha}\right) * N(0, n\sigma^2 I) \\ &= N\left(0, n\left(\frac{A_k \Psi' A_k^T}{\alpha} + \sigma^2 I\right)\right). \end{aligned} \quad (4.6)$$

The classic measure of distances between distributions is the Kullback-Leibler distance[11] in Appendix A.2. Theorem A.2.1 tells us that the expected depth discriminativeness of a set of filters in the worst case is given by,

$$\min_{k_1, k_2 \in K} \frac{-\log \frac{|\Theta_{k_1}|}{|\Theta_{k_2}|} - n + \sum \Theta_{k_1} \cdot * \Theta_{k_2}^{-1}}{2},$$

where $\Theta_{k_i} = n\left(\frac{A_k \Psi' A_k^T}{\alpha} + \sigma^2 I\right)$.

4.3.2 Depth Discriminativeness of Multiple Filters

We wish to compare the depth discriminativeness of multi-channel coded aperture with that of its single-channel counterpart. Unlike the expected reconstruction error, however, depth discriminativeness is difficult to analytically compute. Therefore, the KL-divergence score can be used in lieu of the actual classification accuracy achieved by the sets of filters. We assume that the three channels are identical, and are observed through f^R, f^G, f^B . The appropriate kernels at particular depths can be estimated by downscaling the full-size kernels.

Presence of Bayer Filter

We repeat the analysis in Section 4.2.2 to isolate the effect of modelling the Bayer pattern on accurately measuring the KL-divergence scores of filters. In reality, there should not be any difference between the joint and sequential processes in terms of the posterior for the output, since the weighted nearest-neighbor demosaicking algorithm is reversible via applying the Bayer pattern again. It is worthwhile to note, however, that the single-channel model in Levin et al[14] can be empirically shown to overestimate the minimum KL-divergence of filters, as in Figure 4-6; the overestimation is due to the assumption of a naive process, whereas in practice the sequential process takes place. Of course, if the relation is monotonic, the overestimate can still serve fully as a comparative metric. However, as the figure indicates, the relation is noisy, and the best linear fit suffers from standard deviation of roughly 70, where the median score was slightly over 660.

Effects of Multiple Filters

The benefits arising from the use of multiple filters rather than a single filter can be similarly isolated by assuming that the channels are identical, and comparing the performance of individual filters where $f^R = f^G = f^B$ against the performance of pairs of filters where $f^R = f^B$.

There are two archetypical scenarios in which multiple filters should outperform single filters. First, because our metric considers the lowest KL-divergence score, filters that exhibit little depth discrimination between two particular scales can be paired with complementary filters that perform well on those scales. Figure 4-8

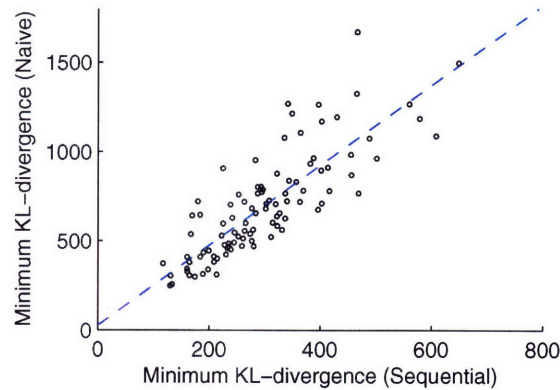


Figure 4-6: Minimum KL-distance estimate in the single-channel model, for both naive and sequential processes, with the best-fit line.

illustrates two such filters g, h and their KL-divergence scores for adjacent scales. To capture a more general trend, we randomly sampled n -tuples of symmetric binary filters, and compared the best individual filter in the tuple in terms of the minimum KL-divergence against the best pair, in which one filter masks R,B channels and the other G. Once n is large enough, the best pair outperforms the best individual filter, as shown in Figure 4-7.

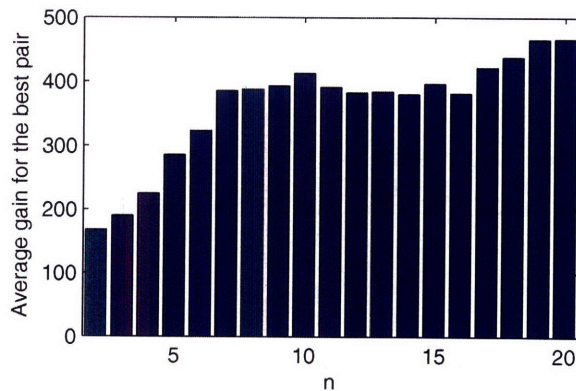
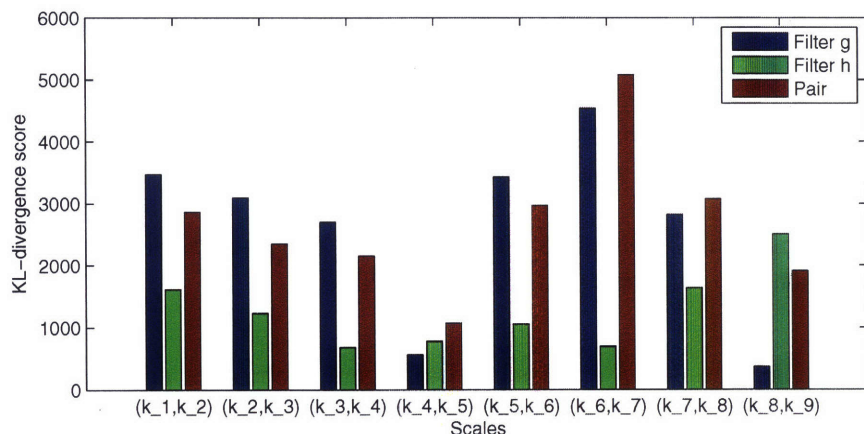
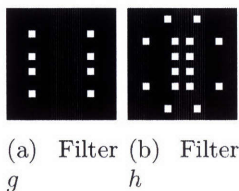


Figure 4-7: Average gain for the minimum pairwise KL divergence of the best-performing pair of filters, over that of the best-performing individual filter in an n -tuple. For each n , 100 sets were sampled.

The second scenario we consider is the strategic case in which one filter is a pinhole. While the pinhole by itself delivers no depth discrimination whatsoever, it



(c) KL-divergence scores for adjacent scales.

Figure 4-8: Example of complementary filters. Note that g has its minimum score for scales (k_8, k_9) , while h has its minimum score for scales (k_3, k_4) , and the two filters complement each other in those two scales. As a result, the pair performs satisfactorily on both scales.

aids in deconvolution and indirectly helps depth discrimination of its partner. We do note that the present assumption on the channels being equal does accentuate this property, and in practice, the channels are not identical. Figure 4-9 demonstrates decent improvement in depth discrimination when filters are paired with a pinhole.

4.3.3 Summary

We find that the multi-channel model offers better performance in terms of reconstruction error than does the single-channel model. Our analysis attributes this independently to both the inclusion of Bayer pattern and the use of multiple filters. Also, combining filters tends to raise the KL divergence score of the filters, giving rise to the notion of “complementary” filters.

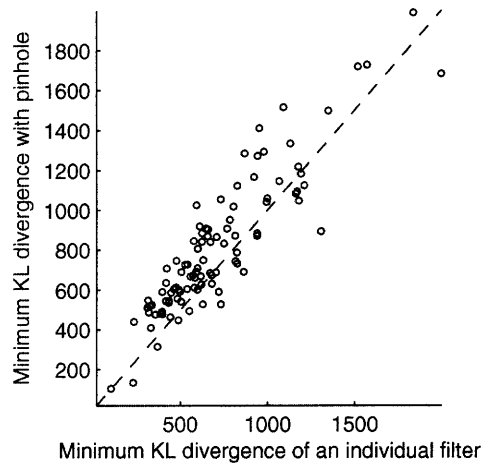


Figure 4-9: Comparison of minimum KL divergence score for individual filters against the case in which they are paired with a pinhole. Each point represents a 15-by-15, binary symmetric filter.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 5

Numerical Techniques for Deconvolution

The sparse priors introduced in Chapter 3 do not admit simple, analytic treatments in the Fourier domain because of its non-linearity when $\rho < 1$. Instead, the maximum-likelihood estimate of the input scene using the sparse prior must be accompanied by an adequate numerical approximation technique. This chapter explores and applies several numerical methods available.

The maximum-likelihood estimate solves

$$\widehat{x}_k = \operatorname{argmin}_x \frac{\|\omega^R \cdot * (f_k^R * x) + \omega^G \cdot * (f_k^G * x) + \omega^B \cdot * (f_k^B * x) - y\|^2}{2\sigma^2} + \log P(x). \quad (5.1)$$

We can express Equation (5.1) in a column-vector form:

$$\widehat{x}_k = \operatorname{argmin}_x \frac{\|D_k \bar{x} - \bar{y}\|^2}{2\sigma^2} + \frac{\alpha}{2} \sum_i q(g^i \bar{x}^R, g^i \bar{x}^G, g^i \bar{x}^B) \quad (5.2)$$

where D_k is the square matrix corresponding to the convolution and the application of Bayer filter, and $q(\dots)$ corresponds to the log of the prior given the derivatives in the three channels. Table 5.1 summarizes the form of the corresponding q for the three multi-channel priors. The methods discussed in this section solves Equation (5.2) for the given prior in an iterative fashion, taking x_0 as the initial solution and generating x_1, x_2, x_3, \dots until convergence or the maximum number of iterations is

reached.

Independent prior	$q(x, y, z) = \sum_j x_j ^\rho + y_j ^\rho + z_j ^\rho$
Independent prior with a change-of-basis matrix R	$q(x, y, z) = \sum_{l=1}^r \sum_j R(l, 1)x_j + R(l, 2)y_j + R(l, 3)z_j ^\rho$
Dependent prior	$q(x, y, z) = \sum_j x_j^2 + y_j^2 + z_j^2 ^{\rho/2}$

Table 5.1: Objective functions for the three multi-channel priors.

5.1 Newton's Method

Newton's Method is a standard numerical method for solving a non-linear equation via a second-order approximation. It evaluates the second-order Taylor expansion of the objective function centered at the current value of x_t , and solves the resulting quadratic optimization for the new value x_{t+1} . In closed form, the iterative step for Equation (5.2) is, assuming the independent prior,

$$\left(D_k^T D_k + \frac{\rho(\rho-1)}{2} \alpha \sigma^2 K \right) \bar{x}_{t+1} = D_k^T \bar{y} + \frac{\rho(\rho-2)}{2} \alpha \sigma^2 K \bar{x}_t, \quad (5.3)$$

where $K = \text{diag}_C \left(\sum_i (g^i)^T \text{diag} \left(|g_{j \rightarrow}^i \bar{x}_t^C|^{\rho-2} \right) g^i \right)$. The exact derivation of Equation (5.3) can be found in the appendix.

We find that Newton's Method fares poorly in practice because the underlying assumption of second-order approximation does not hold; the objective function f is locally concave due to the non-linear exponent $\rho < 1$, so gradient approach does not minimize the solution. Figure 5.2 demonstrates Newton's Method in action.

5.2 Iteratively Re-Weighted Least Squares (IRLS)

Iteratively Re-Weighted Least Square (IRLS) is another approximation technique that can be used to minimize the value of the objective function $\sum f(A_{i \rightarrow} x - B_i)$,

where x is a column vector and f is non-negative and maps zero to itself. Applying IRLS to the objective in Equation (5.2) for the independent prior, we obtain

$$\left(D_k^T D_k + \frac{\rho}{2} \alpha \sigma^2 K \right) \overline{x}_{t+1} = D_k^T \overline{y}, \quad (5.4)$$

where K is as before. See the appendix for the exact derivation and details on IRLS. Note that IRLS correctly converges to the local minimum for a simple scenario, as in Figure 5.2.

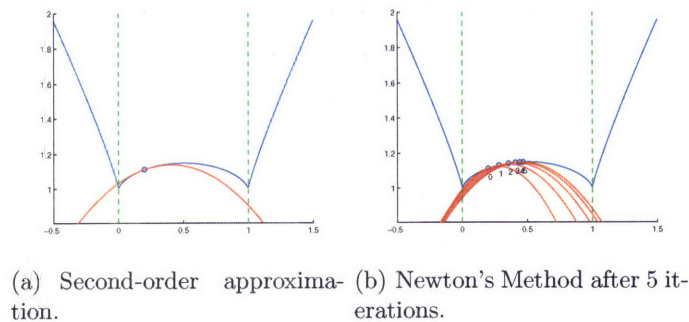


Figure 5-1: Application of Newton's Method on simple scenarios. Left: The second-order approximation is locally accurate. Right: However, Newton's Method iteratively ascends the gradient in the incorrect direction, because the objective function f is locally concave.

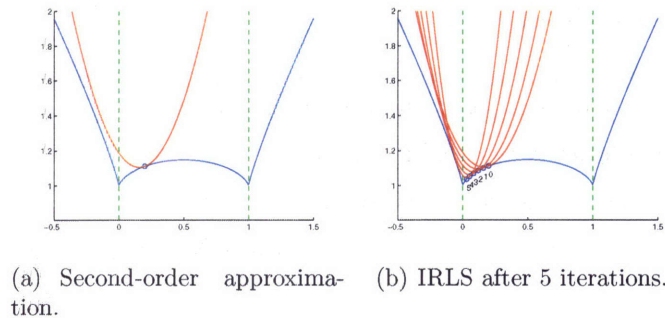


Figure 5-2: Application of IRLS on simple scenarios. Left: The second-order approximation is locally accurate. Right: IRLS approaches the local minimum in five iterations.

5.3 Gaussian Mixture Model (GMM)

The non-linear priors on the derivatives can be approximated roughly as a weighted sum of zero-mean multi-variate Gaussians. In other words,

$$f(x, y, z) \simeq \sum_{i=1}^c w_i N((x, y, z); (0, v_i)),$$

where w_i is the weight and v_i is the covariance matrix of the i -th Gaussian component, $i \in \{1, 2, \dots, c\}$. The weights and covariance matrices can be trained on the prior by treating it as an observed distribution and clustering the observations. Once the prior is expressed as a cluster of Gaussian components, the correct value of each derivative can be treated as an observation generated from the cluster. Solving for the image x then entails classifying each derivative in the image x into one of the c clusters, using Expectation-Maximization (EM), which is a standard optimization algorithm with well-documented behavior and convergence. Table 5.3 and 5.3 gives the exact algorithm.

- 1: Initialize weights w_1, w_2, \dots, w_c and covariance matrices v_1, v_2, \dots, v_c .
- 2: For each iteration,
- 3: Estimate: For each possible posterior observation¹ (x, y, z) , compute the $P_i(x, y, z)$, the probability that (x, y, z) is generated from the i -th cluster. This is equal to the normalized value of $w_i N((x, y, z); (0, v_i))$, with the constraint that $\sum_i P_i(x, y, z) = 1$.
- 4: Maximize: Recompute w_i, v_i to maximize the posterior, i.e. $w_i \propto \int P_i(x, y, z) f(x, y, z) dx dy dz$ with $\sum w_i = 1$, and v_i is the variance of (x, y, z) weighted by $P_i(x, y, z) f(x, y, z)$.

Table 5.2: Expectation-Maximization for training the cluster weights and variances corresponding to each multi-channel prior. This algorithm needs to run only once for a given set of parameters.

- 1: Let $x_0 = y$.
- 2: Compute the set of spatial derivatives of x_0 in each channel. Denote them by d_0^R, d_0^G, d_0^B , for the respective channel.
- 3: For each iteration $t = 1, 2, \dots$,
- 4: Estimate:
- 5: Compute x_t by solving the maximum-likelihood equation on x_{t-1} :

$$\left(A^T A + \frac{\rho}{2} \sigma^2 K\right) x_t = A^T y,$$

where $K = (G^1)^T \text{diag}(\sum_i P_i(d^C) v_i^{-2}) G^1$, with the entries of the diagonal matrix indexed by r .

- 6: Compute the spatial derivatives $d^R = \{d_1^R, d_2^R, \dots\}$, $d^G = \{d_1^G, d_2^G, \dots\}$, $d^B = \{d_1^B, d_2^B, \dots\}$ of $x_t = \{x_t^R, x_t^G, x_t^B\}$.
- 7: Maximize:
- 8: For each derivative d_i^C , update $P_i(d_i^C)$. This is the probability that the derivative is generated by the i -th component, which is

$$\frac{w_i N(d_i^C; (0, v_i^2))}{\sum_j w_j N(d_i^C; (0, v_j^2))}.$$

Table 5.3: Expectation-Maximization for iteratively estimating non-linear multi-channel priors.

5.4 Conjugate Gradient Method

All three numerical methods discussed thus far approximates the non-linear system as a locally linear system with a self-adjoint matrix as the coefficient, which can be solved directly via Gaussian elimination. Unfortunately, once the number of pixels in the image exceeds 10,000, the analytic solution becomes computationally expensive, so we instead employ the Conjugate Gradient method, given in Appendix A.3.

5.5 Evaluation

5.5.1 Pixel Recovery in 1D Scenarios

We can examine and evaluate the behaviors of the existing numerical methods combined with the multi-channel priors by executing them on small toy examples. Because they are general methods on recovering the input from blurred or partial im-

ages, we consider 1D images consisting of two channels (red and blue), with exactly one pixel missing each channel, in order to facilitate visualization. The scenarios are given in Figure 5.5.1.

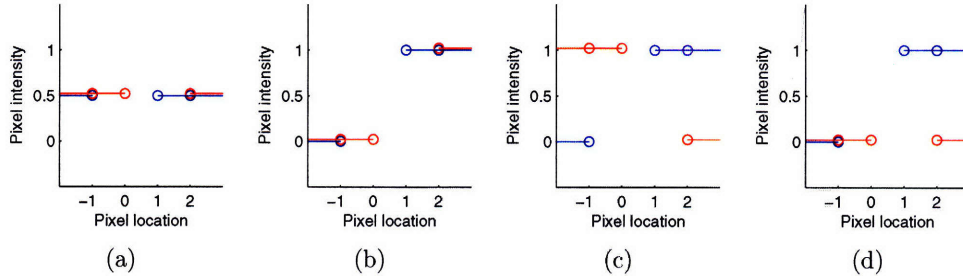


Figure 5-3: One-dimensional toy scenarios for evaluating numerical methods and multi-channel priors. Each scenario features a 1D image in two channels with one pixel missing in each channel.

For each scenario, we consider three priors defined previously, re-formulated to apply to 1D two-channel images, with R being a randomly generated 5-by-2

matrix $\begin{bmatrix} -0.4326 & 1.1909 \\ -1.6656 & 1.1892 \\ 0.1253 & -0.0376 \\ 0.2877 & 0.3273 \\ -1.1465 & 0.1746 \end{bmatrix}$, while varying the numerical method, between IRLS

and GMM. The appropriate numerical method is run for seven iterations, initialized randomly, and the progression is plotted on a 2D plane, each axis representing the value of one of the two missing pixels. The contour of the objective function is also mapped to better represent the direction of update, with blue being the lowest. Ideally, the prior should give local minima where likely, and the numerical methods should converge to those local minima. The ideal solution for each of the scenario is $(0.5, 0.5)$, $(0, 1)$, $(0, 0)$, and $\{(0, 0), (0, 1)\}$, respectively.

The results, available in Figure 5-4, demonstrate that IRLS and GMM are comparable in their convergence rate, while the independent prior with change of basis and the dependent prior seem to converge to more favorable local minima than does the independent prior on R,G,B channels. Especially in the second and third scenarios, the independent prior exhibits multiple local minima, including ones in which edges are misaligned.

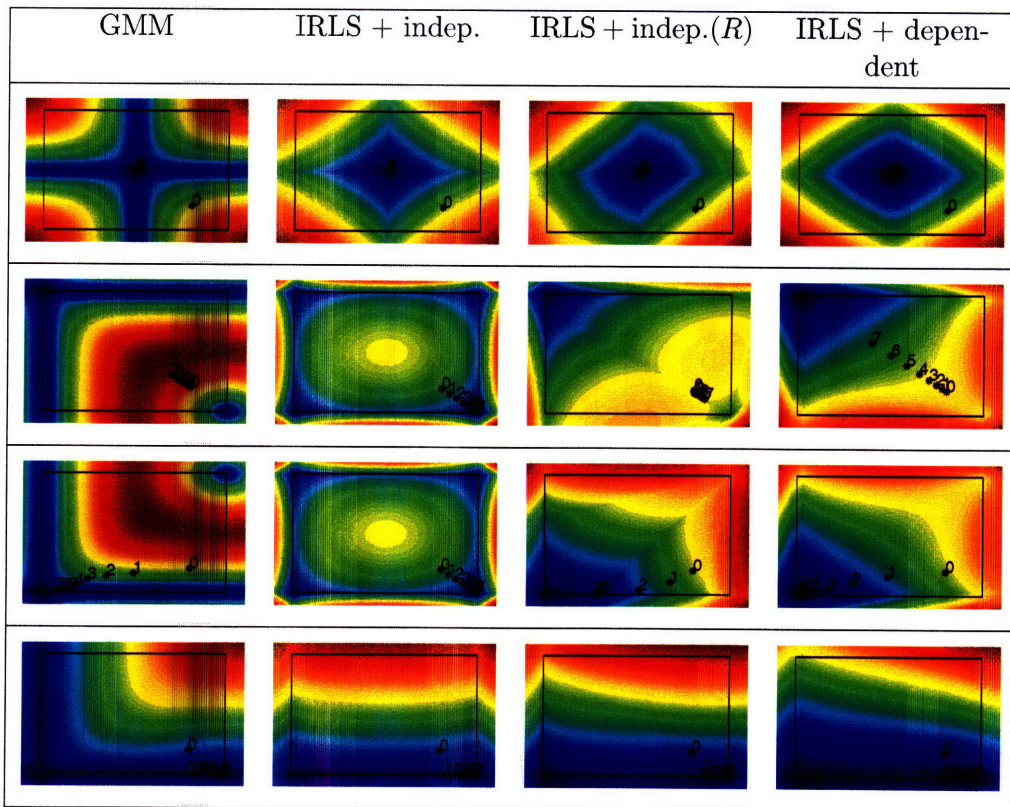


Figure 5-4: Iterative reconstruction result on the scenarios in Figures 5.3(a) through 5.3(d), with GMM or IRLS, and various multi-channel priors. Each row corresponds to a particular scenario, and each column corresponds to a reconstruction method.

5.5.2 Demosaicking in 2D Scenarios

The numerical methods are further compared on small two-dimensional images that have been masked by the Bayer pattern. In the multi-channel model, this setup is equivalent to having f^R, f^G, f^B all be pinhole. The masked image is deconvolved with the three priors, using either IRLS or GMM. Each numerical method is run for five iterations. See Figure 5-5 for the results. The results visually indicate that the dependent prior outperforms the independent prior in aligning edges. In case of the independent prior, misaligned edges can be discerned in form of color artifacts. For the change of basis, the YUV color space was used.

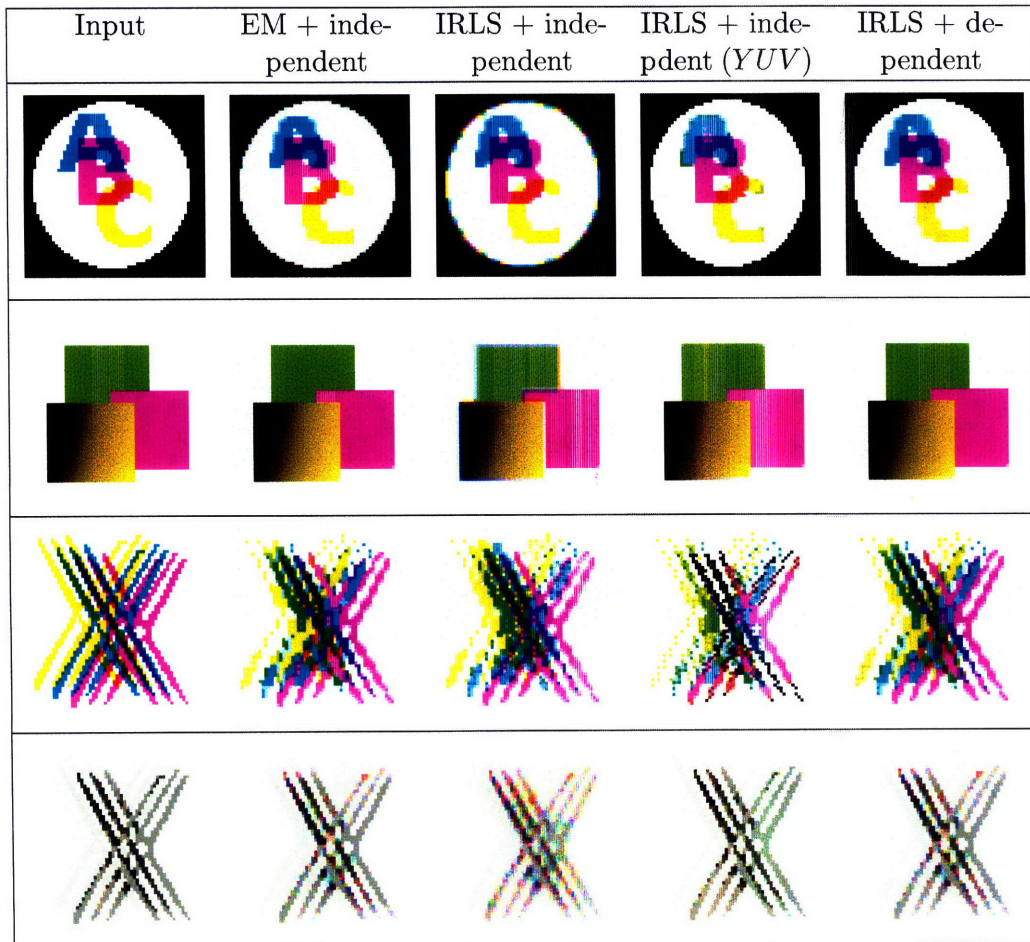


Figure 5-5: Demosaicking results for various two-dimensional images, with GMM or IRLS, and various multi-channel priors. Each row corresponds to a particular input image, and each column corresponds to a reconstruction method.

5.5.3 Deblurring in 2D Scenarios

We further compare the numerical methods on small two-dimensional images blurred by a single filter ($f^R = f^G = f^B$). See Figure 5-6 for the results. We observe that the YUV color space complements the numerical methods on monochromatic images than the RGB color space, and that dependent prior preserves sharp edges better than independent prior. In most of cases, however, the multi-channel priors produce satisfactory results visually.

While the groundtruth is available for these synthetic results, there is no single consistent way of quantifying the quality of the deconvolution results. Common metrics as the mean square error do not adequately capture it, because shifting edges produces no visible artifacts to the image, but it is penalized heavily by the mean square error. In fact, the mean square error prefers the Gaussian prior, which smooths out the elevations in image intensity. Finding an appropriate metric remains an open question.


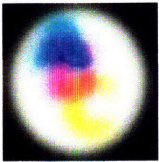





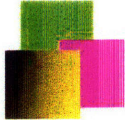

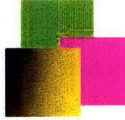


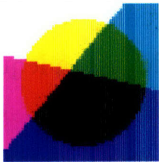











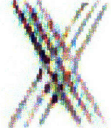

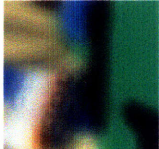



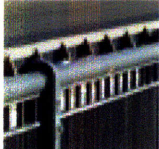

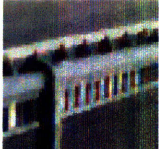
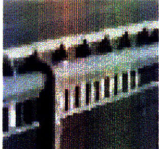
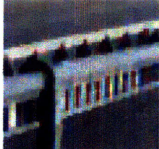

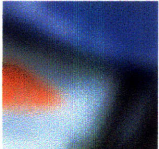
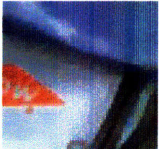
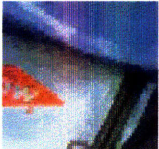
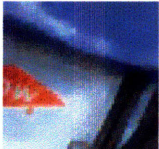
Input	Blurred	IRLS + independent	IRLS + independent (YUV)	IRLS + dependent
				
				
				
				
				
				
				
				

Figure 5-6: A two-dimensional image deblurred with multi-channel priors.

Chapter 6

Physical Implementation

Physical implementation of our system requires the fabrication of coded aperture in three primary channels to be incorporated into a camera lens. Subsequently, the kernels for the three channels are to be observed and calibrated, after which images taken with the lens can be fully reconstructed using the kernels. Rather than creating a lens or even an aperture from the scratch, we simply create an occluder that can selectively filter in different wavelengths, and insert it between the lens elements of a standard prime lens, as did Levin et al [14].

The core issue is the accurate construction of a set of filters that can ideally block the light rays in one or more primary colors entirely, while letting the other colors pass through unattenuated. In practice, this ideal filtering is unattainable, as the camera sensor itself does not separate lights in different channels perfectly. We discuss the selection and composition of material for achieving the desired attenuation as closely as possible mathematically, and the resulting spectral filtering we have selected.

6.1 Filter Construction

6.1.1 Computing Attenuation

Given the set of filters f^R, f^G, f^B , corresponding to the codes in three primary channels, we would like to construct an aperture with the property that the relative transmission obtained at location (x, y) is equal to $f^R(x, y)$ in case of red light,

$f^G(x, y)$ in case of green light, $f^B(x, y)$ in case of blue light. In reality, attenuation can be controlled as a function of wavelengths, and the wavelengths of what is perceived as red, green, blue in fact overlap. Figure 6-1 charts the spectral sensitivity of a Canon CMOS sensor, with the overlaps clearly discernible.

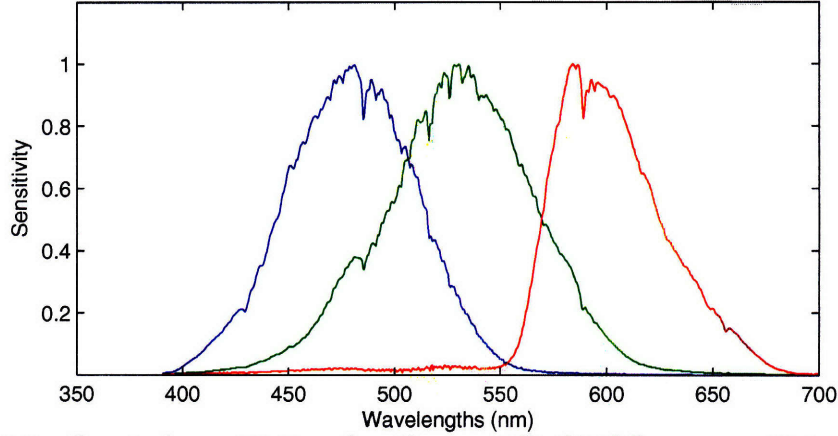


Figure 6-1: Spectral sensitivity of a Canon 10D CMOS sensor. Note the clear overlap between neighboring channels.

For instance, if the object has a particular spectrum such that it reflects only light at 500nm, it would register as equally green and blue. Since the attenuation depends on the wavelengths alone, it is impossible to prevent the camera sensor from registering the same value for both green and blue; whatever attenuation the filter achieves, both green and blue pixel values will be attenuated equally. Therefore, the actual reduction in red, green, blue pixel values will depend on the spectral characteristics of the object being lit. Secondly, the reduction also depends on the source of light, since different light sources have different spectrum.

More formally, let t be the spectral transmission characteristic of the filter as a function of wavelengths; l be the spectrum of the light source; m be the reflectance of the object; s^R, s^G, s^B be the spectral sensitivity of the sensor in the three channels. Then, the fraction of red light observable after the insertion of the filter is,

$$\frac{\int l(\lambda)m(\lambda)s^R(\lambda)t(\lambda)d\lambda}{\int l(\lambda)m(\lambda)t(\lambda)d\lambda}. \quad (6.1)$$

The reduction in green, blue pixel values can be computed similarly.

The quantity in Equation (6.1) is difficult to control or compute precisely, and is not purely a function of the filter material. Hence, we assume that both the source light and the object have the uniform spectrum. Most light sources, such as the Sun, in fact tend to have a smooth spectrum in visible wavelengths. Also, since the spectral characteristics are not available as continuous functions of wavelengths, we discretize them to make use of available data. Then, the resulting formulae for attenuation in the three channels are, represented as a triple,

$$\left\{ \frac{\sum_{\lambda} s^R(\lambda)t(\lambda)}{\sum_{\lambda} t(\lambda)}, \frac{\sum_{\lambda} s^G(\lambda)t(\lambda)}{\sum_{\lambda} t(\lambda)}, \frac{\sum_{\lambda} s^B(\lambda)t(\lambda)}{\sum_{\lambda} t(\lambda)} \right\}. \quad (6.2)$$

If f^R, f^G, f^B are unrestrained binary filters, at a given spatial location, the desired attenuation falls into one of $2^3 = 8$ types: $\{0, 0, 1\}$, $\{0, 1, 0\}$, $\{1, 0, 0\}$, $\{1, 1, 0\}$, $\{1, 0, 1\}$, $\{0, 1, 1\}$, $\{1, 1, 1\}$ and $\{0, 0, 0\}$. The last two triples are readily attained by air and any opaque layer, respectively, leaving six distinct spectral transmission characteristics to be found. In our implementation, however, we determined that two of the kernels, f^R and f^G , should be identical. This decision confers the benefit of reducing the number of required nontrivial materials into $2^2 - 2 = 2$, thereby reducing the complexity in fabrication.

6.1.2 Filter Selection

We followed the same guidelines as Levin et al[14] in restricting the space of possible filters, and limited our search to 15-by-15 binary patterns. Under other practical considerations in filter fabrication, the kernel for red and green channels was chosen to be pinhole. The kernel for blue channel was chosen from a large sampling of kernels that maximized the expected depth discrimination. See Figure 6-2.

Using pinhole filters opens up another possible benefit. In practice, full red and green channels are sufficient, even in presence of blurry blue channel, to generate eye-pleasing, perceptually sharp images, because the human eyes excel at reading contrasts in the luminosity channel, which is composed mainly of red and green. Finally, as previously discussed, sharp channels should aid in deconvolution of the blurred channels, and thereby improve the depth discrimination.

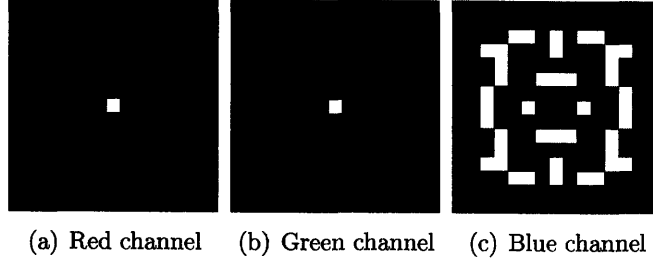


Figure 6-2: Ideal kernels for each channel. Each kernel is a 15-by-15 discrete two-dimensional signal, with binary values.

6.1.3 Maintaining White Balance

Another consideration is that the resulting image should have white balance that is as close to the original as possible. This is complicated by the fact that different kernels have different area, therefore admitting different amount of light under equal conditions. The problem is exacerbated especially if one of the filters is pinhole.

The ratio between the areas of two filters chosen for the implementation is 1/42. If the respective materials carry attenuation of $\{a_1, a_2, a_3\}$ and $\{b_1, b_2, b_3\}$ per unit area, the resulting pixel values will be proportional to $\{a_1+42b_1, a_2+42b_2, a_3+42b_3\}$. Therefore, the kernels with which the images are actually convolved are

$$\begin{aligned}\widehat{f^R} &= a_1 f^R + 42b_1 f^B, \\ \widehat{f^G} &= a_2 f^G + 42b_2 f^B, \\ \widehat{f^B} &= a_3 f^R + 42b_3 f^B.\end{aligned}$$

The effective kernels will be weighted sum of the ideal kernels, and to retain the distinctiveness of kernels and remain as close as possible to theoretical expectations, we must achieve the following balance:

$$a_1 \gg 42b_1, \quad a_2 \gg 42b_2, \quad a_3 \ll 42b_3, \quad a_1+42b_1 \simeq a_2+42b_2 \simeq a_3+42b_3. \quad (6.3)$$

Unfortunately, fabricating translucent material with the desired spectral characteristic is difficult. Instead, we sampled an existing set of color gels, available in sheets, with known spectral transmission characteristics¹, and chose colors that sat-

¹The spectral transmission characteristics were generously provided by LEE filters,

ified the condition in (6.3) in practice. Figure 6-3 shows the spectral transmission characteristics of the color gels used in the implementation. In practice, color gels were stacked to achieve the required gain in the ratio of transmission in different channels. See Figures 6.4(e) through 6.4(g) for estimation of the kernels in the three channels.

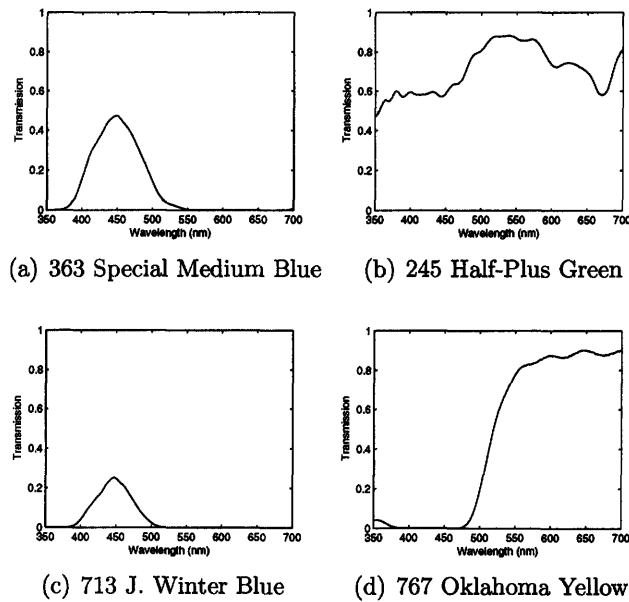


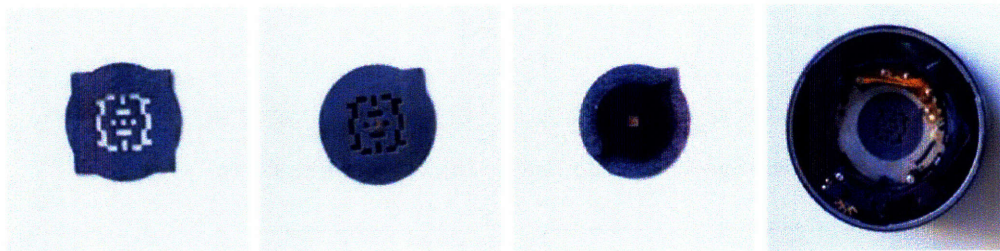
Figure 6-3: Spectral transmission characteristics of four color gels that are used in filter fabrication.

6.1.4 Assembly

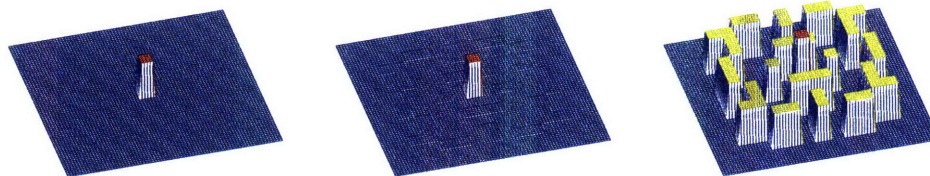
The final aperture assembly consisted of a vertical stack of materials with different spectral transmission characteristics. A negative of the union of the two kernels was cut out of black cardboard, and the stacked color gels for each kernel were taped on the cardboard to provide the desired attenuation at the correct spatial location, as in Figure 6-4. The aperture is then inserted into a standard Canon EF 50mm f/1.8 lens², and the lens is focused at 2.0m, after which the barrel is immobilized with tape.

<http://www.leefilters.com>.

²For instructions on disassembly of a Canon EF 50mm f/1.8 lens, see <http://www.ejarm.com/photo/ef5018iidis/>.



(a) The negative pattern in cardboard (b) Completed aperture (front) (c) Completed aperture (back) (d) Aperture inside the barrel of a Canon EF 50mm f/1.8 lens



(e) Expected kernel in red channel (f) Expected kernel in green channel (g) Expected kernel in blue channel

Figure 6-4: Assembly of the multi-channel aperture into the camera lens, with the expected kernels for each channel. Note that the filters overlap to a small extent.

The cost of the modified lens is nominal, and is dominated by the cost of the lens (under \$90.) The color filters and the black cardboard can be obtained at a fraction of a dollar per square feet.

6.2 Calibration

The coded aperture does not in practice achieve the correct blur with the ideal kernels in Figure 6-2, due to various distortions, diffractions, and reflections in the lens. The matter is further complicated by the fact that the color filters are not well-behaved and produce reflections, become dirty, shift in place. Figure 6-5 illustrates these distortions, especially the diffraction and scattering of light from the pinhole filters in red, green, channel. Thus, we measure the *effective* kernels instead, and carry out the subsequent computations with them. Well-controlled fabrication of the filters result in kernels that are similar to the ideal ones.

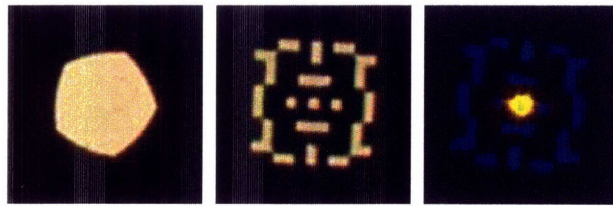


Figure 6-5: Effective kernels observed from a pinhole light source (60cm away) at multiple distances. Left: regular aperture at $f/4.0$. Center: single-channel aperture. Right: multi-channel aperture. The last image was taken with longer exposure time to show the colors more clearly.

Appendix C details the calibration process, including both the physical hardware for taking measurements and the algorithms for inferring the kernels.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 7

Depthmap Generation

The maximum-likelihood estimate of the depth of the image $\hat{k} = \operatorname{argmax}_k P_k(y)$ suffers from two crucial drawbacks. The first is that the expression cannot be easily evaluated, as it is the integral of the marginal distribution $P_k(y|x)$ over x , which is highly non-linear. The second is that the scene often includes multiple depths. In order to address these problems, the original single-code aperture photography estimates $P_k(y)$ in small local windows, assigning a depth to each pixel[14]. In place of $P_k(y)$, the deconvolution error $E_k = \|y - f_k * x\|^2$, weighted by some learnt coefficients, is used. A graph-cut algorithm then regularizes the resulting depthmap. In this chapter, we detail several approaches to determining the local depth, and the process of regularization, in the multi-channel case.

7.1 Classification with Deconvolution Error

7.1.1 Unnormalized Error

In single-channel coded aperture, it is necessary to weight the convolution error by coefficients λ_k , because the smaller scales consistently generate lower deconvolution error; in deconvolution process, they are penalized by the sparsity cost, but our depth estimate does not take the sparsity of the output into account. In multi-channel coded aperture with pinhole filters, however, smaller scale also generates large deconvolution error, because the blurred channel is deblurred to match the sharp channels. In certain cases, the unnormalized error $E_k = \|y - f_k * x\|^2$ is

sufficient to discriminate between depths. In other words,

$$\hat{k} = \operatorname{argmin}_k E_k.$$

7.1.2 Normalized Error

A more complex model involves normalization of the error terms by some constants: $\lambda_k E_k$. The constants are learned using logistic regression, maximizing the soft classification accuracy $\sum_{k,i} \frac{\exp \lambda_k E_k(i)}{\sum_l \exp \lambda_l E_l(i)}$, where $E_l(i)$ refers to the deconvolution error at depth l for the i -th datapoint. In this case,

$$\hat{k} = \operatorname{argmin}_k \lambda_k E_k.$$

7.1.3 Support Vector Machine (SVM)

The model with normalization can be further extended to a general k -dimensional linear space by employing linear SVMs. A two-class linear classifier is trained using $E(i) = \{E_1(i), E_2(i), E_3(i), \dots\}$ as datapoints to discriminate between every pair of depths, and a multi-class classifier is built with the two-class linear classifiers.

7.2 Regularization

The initial depthmap generated by estimating the likeliest depth at each pixel is noisy and rough, and must undergo a regularization process to produce reasonable results. One particular cause is that the ringing produced at objects at wrong depth tends to travel across interfaces between depths, hampering the deconvolution of objects at the correct depth. Regularization based on pixel similarity can repair this phenomenon.

More precisely, we iteratively apply Kolmogorov's implementation for binary mincut[4] on the depthmap for each depth, where the edges exist between adjacent pixels to capture the dissimilarity, and the synthetic source node connect to pixels of the correct depth, and the target node connect to the remaining pixels. See Table 7.1 for detail. Figure 7-1 shows the effect of regularization on a rough depthmap.

- 1: Iterate several times:
- 2: For each depth $k \in K$,
- 3: Construct graph $G = (V, E)$ where V is the set of pixels along with s, t . Here, we set $E = E_p \cup E_s \cup E_t$ where E_p is the edges between adjacent pixels with weights corresponding to the dissimilarity, E_s is the edges between s and pixels currently classified as depth k , and E_t is the edges between t and the remaining pixels.
- 4: Execute min-cut algorithm on G, s, t . For pixels still connected to s , relabel their depths as k .
- 5: For each pair of depth k_1, k_2 ,
- 6: Construct graph $G = (V, E)$ as in the outer loop, while restricting V to the set of pixels currently classified to k_1 or k_2 , plus s, t . The edge set E is constructed similarly, with E_s, E_t each containing edges between s or t and pixels currently classified to k_1 or k_2 , respectively.
- 7: Execute min-cut algorithm on G, s, t . For pixels still connected to s , relabel their depths as k_1 . Relabel the depths of remaining pixels in V as k_2 .
- 8: End iteration.

Table 7.1: Implementation of graphcut algorithm to regularize the initial depthmap.

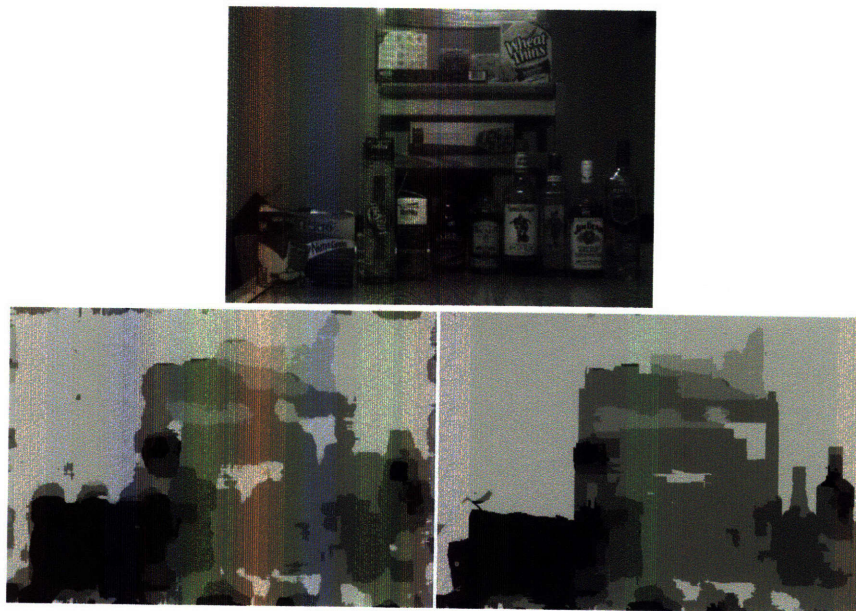


Figure 7-1: Effect of regularization on depthmap. Top: The captured scene. Bottom left: depthmap constructed from deconvolution error. Bottom right: regularized depthmap.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 8

Experimental Results

We evaluate the depth classification accuracy of single- and multi-channel coded aperture on planar scenes of known depths. Two synthetic scenes are blurred with the known kernels at 12 different depths, and two physical scenes are captured at 12 different depths by translating the position of the camera, using both single- and multi-channel coded aperture:

- SYNTHETIC1: A small digital image (aerial view of a building).
- SYNTHETIC2: A small digital image (busy street).
- BUILDING: A small planar, black-and-white printout.
- POSTERS: A collection of colorful magazine cutouts, with varying texture.

These planar scenes are primarily for evaluating the single- and multi-channel coded aperture in depth discrimination. The synthetic datasets are further used to test robustness against increasing level of noise in the blurred image.

Additionally, we demonstrate our system on the following staged scenes of cluttered objects, showing the deconvolution results and depth maps generated from the captured images:

- PRINTS: An arrangement of large planar, black-and-white printouts placed on a table at various depths.
- SHOES: An arrangement of small objects with varying amounts of texture.

- KITCHEN: An assortment of many small objects cluttered on a table, including translucent and thinly shaped objects.

Dataset	Width ¹	Height ¹	Synthetic	Planar
SYNTHETIC1	250px	372px	Y	Y
SYNTHETIC2	378px	250px	Y	Y
BUILDING	362px	330px	N	Y
POSTERS	868px	846px	N	Y
PRINTS	1183px	935px	N	N
SHOES	1629px	767px	N	N
KITCHEN	1407px	774px	N	N

Table 8.1: Summary of test scenes prepared for single- and/or multi-channel coded aperture photography.

8.1 Kernel Calibration

The Canon EF 50mm f/1.8 lens fitted with the multi-channel coded aperture was set to focus at 2.0m, and the effective kernels were measured at distances between 2.10m and 3.20m, with 10cm increment. The same calibration was performed also for single-channel coded aperture, using the code in the blue channel from the former. Figure 8-1 shows the kernels. The standard deviation for the convolution error on the calibration images ranged from $\sigma = 0.003$ to $\sigma = 0.004$. For deconvolution of physical test scenes, the value of $\sigma = 0.004$ was used.

¹If the scene is captured at multiple depths, the dimensions from the closest distance are shown.

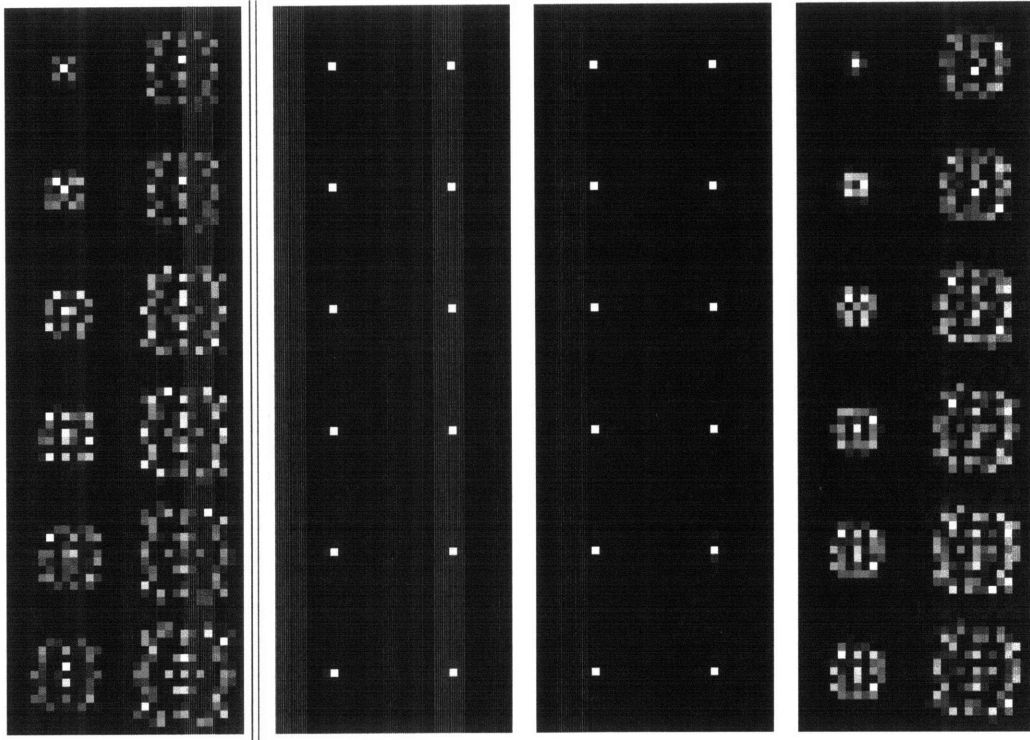


Figure 8-1: Effective kernels measured at distances between 2.10m and 3.20m, with 10cm increment. The displayed kernels have been scaled in intensity in order to show the pattern more clearly, and are arranged into a 6-by-2 block per channel. Left: kernels for the single-channel coded aperture. Right: kernels for the multi-channel coded aperture, in red, green, blue channels, respectively.

8.2 Robustness against Noise

The two synthetic scenes were blurred at each depth with either the single- or multi-channel coded aperture, and Gaussian noise of $\sigma = 0.004, 0.007, 0.010$ or 0.013 was added. For each model and noise level, the 24 resulting scenes ($= 2$ datasets $\times 12$ depths) were deblurred with all twelve kernels in the appropriate model.

8.2.1 Depth Classification in Presence of Noise

For each model and noise level, 4800 points were sampled from the deblurred images to train a classifier based on normalized deconvolution error. Figure 8-2 shows the classification accuracy when the classifiers were applied to the entire datasets: While

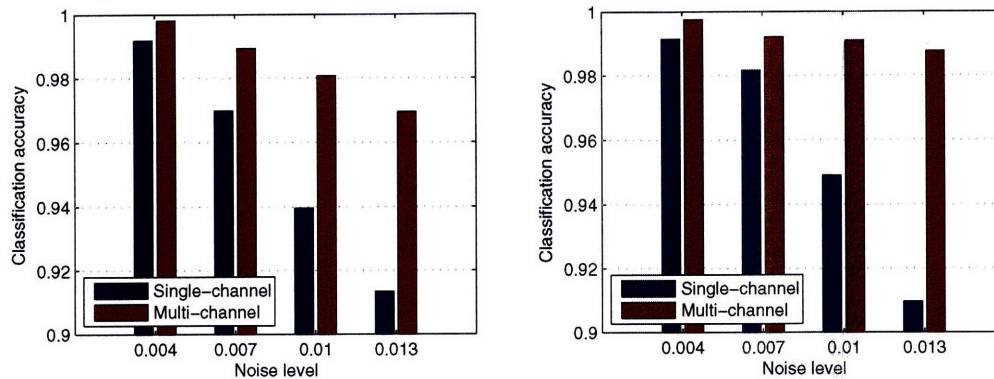


Figure 8-2: Depth classification accuracy on two synthetic datasets at various noise level. Left: results on SYNTHETIC1. Right: results on SYNTHETIC2.

both models perform satisfactorily, identifying the depth correctly at least 90% of the time, the multi-channel model remains robust to noise and outperforms the single-channel model on both datasets, at all noise level.

8.2.2 Quality of Deconvolution in Presence of Noise

Tables 8.2 and 8.3 show the deconvolution results on the synthetic datasets, at various noise level and depth. The results from the multi-channel model are sharper and clearer at all noise levels, thanks to the preservation of red and green channels.














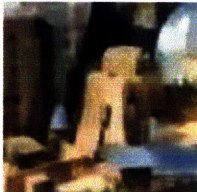
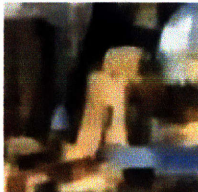


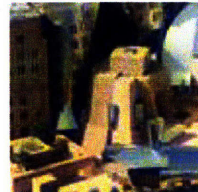
Depth	Single-channel model			Multi-channel model		
	$\sigma = 0.004$	$\sigma = 0.007$	$\sigma = 0.010$	$\sigma = 0.004$	$\sigma = 0.007$	$\sigma = 0.010$
2.20m						
2.70m						
3.20m						

Table 8.2: Results of deconvolution on SYNTHETIC1 at various noise level. 100-by-100 subsquare is shown.

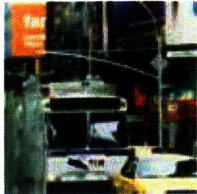










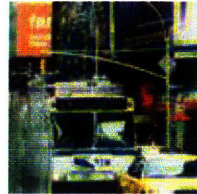

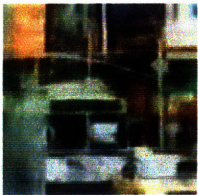
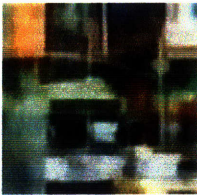
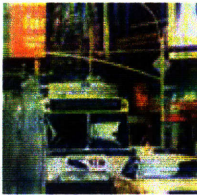


Depth	Single-channel model			Multi-channel model		
	$\sigma = 0.004$	$\sigma = 0.007$	$\sigma = 0.010$	$\sigma = 0.004$	$\sigma = 0.007$	$\sigma = 0.010$
2.20m						
2.70m						
3.20m						

Table 8.3: Results of deconvolution on SYNTHETIC2 at various noise level. 100-by-100 subsquare is shown.

8.3 Results on Physical Planar Scenes

8.3.1 Depth Classification

Each of the two physical datasets **BUILDING** and **POSTERS** consists of a single planar scene captured at 12 different depths. Figure 8-3 shows the result of training and testing SVMs on the two datasets. Figure 8-4 shows the same result for depth classification with normalized deconvolution error. The training set consisted of 900 regularly spaced points at each depth from both scenes, for a total of 21600 points (out of millions of possible vectors in each dataset.) Experimentally, SVMs outperform depth classification by minimum normalized deconvolution error; also, single-channel coded aperture eclipses multi-channel coded aperture in depth discrimination, which runs counter to the results from synthetic datasets. We further note that multi-channel coded aperture performs relatively better on the grayscale dataset **BUILDING** than on the full color dataset **POSTERS**, as the lost of depth information from red and green channel is less severely felt. The average error in depth estimate for **BUILDING** was 6.1cm for single-channel coded aperture and 8.7cm for multi-channel coded aperture, whereas **POSTERS** had 9.6cm and 13.5cm, respectively.

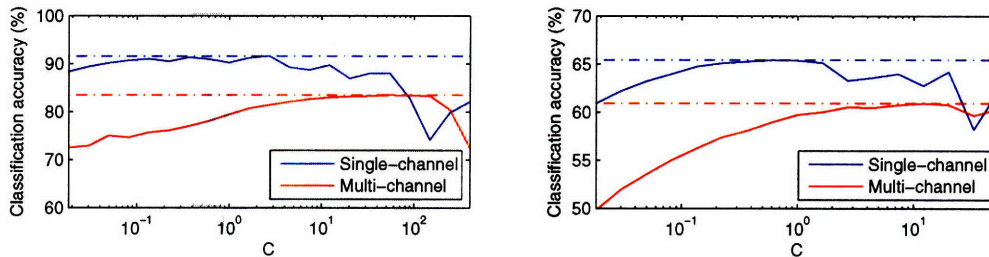


Figure 8-3: Depth classification accuracy on two datasets for linear SVMs of varying parameter. Left: results on dataset **BUILDING**. Right: results on dataset **POSTERS**.

The planar scenes include large textureless areas that provide no information on local depth, accounting for the high classification errors. Figure 8-5 graphs the depth classification accuracy when pixels with little local entropy, defined as the standard deviation in intensity inside a 60-by-60 window, are discarded from the test sets.

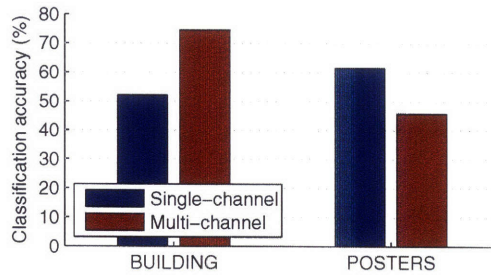


Figure 8-4: Depth classification accuracy on two datasets for normalized deconvolution error.

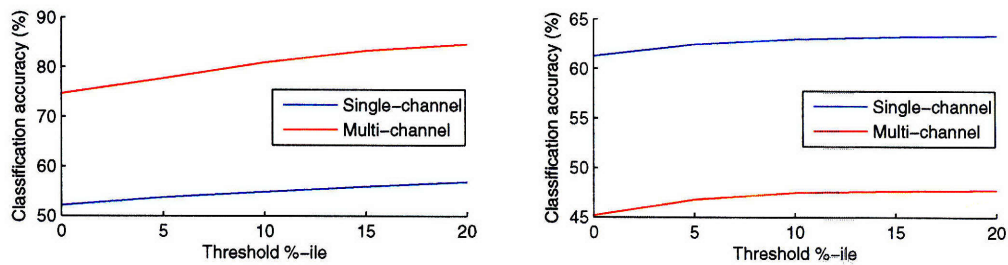


Figure 8-5: Depth classification accuracy for pixels of high local entropy. Left: results on dataset BUILDING. Right: results on dataset POSTERS.

8.3.2 Quality of Deconvolution

Table 8.4 displays the results of deconvolution on the planar datasets, assuming the correct depth. 100-by-100 windows at three distinct depths are shown. In deconvolution, the multi-channel coded aperture outperforms a typical non-pinhole code, as the red and green channels are preserved and can produce reasonable image even when the blue channel is blurred. We also note the reduction in contrast, due to the reflections from the color filters.


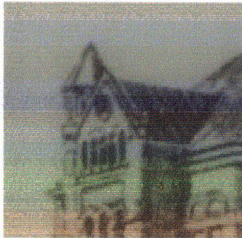
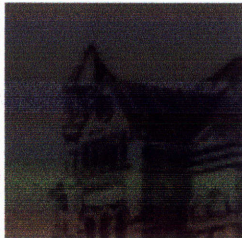


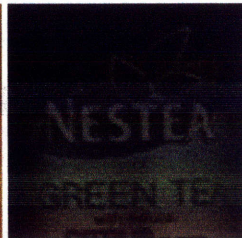

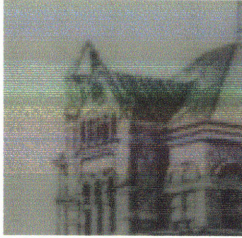





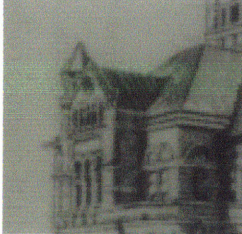




Depth	BUILDING			POSTERS		
	Single-channel	Multi-channel	Multi-channel (Blue)	Single-channel	Multi-channel	Multi-channel (Blue)
2.20m						
2.70m						
3.20m						

Table 8.4: Results of deconvolution of planar datasets, at the correct depths. For clarity, we display zoomed-in portions of heavy texture for selected depths, for both single- and multi-channel deconvolution. The blue channel for multi-channel deconvolution is specifically shown, since red and green channels are sharp to begin with.

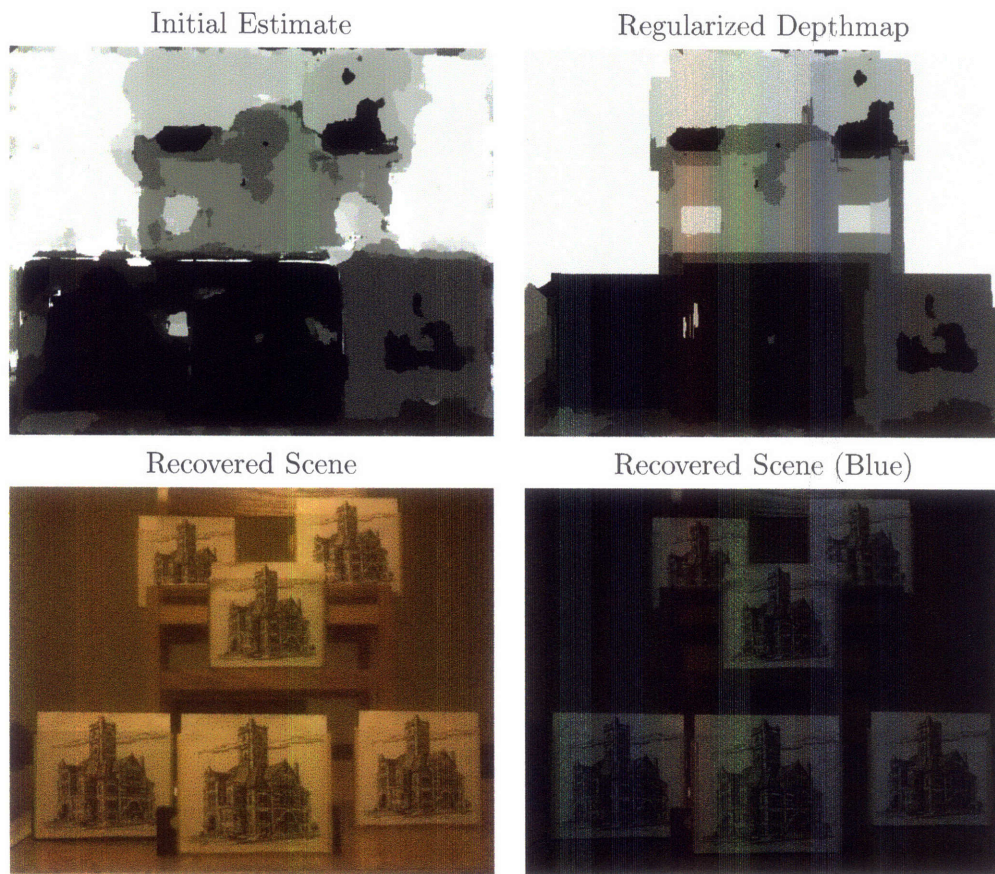


Figure 8-6: Depth and scene recovery on PRINTS.

8.4 Depthmap of Cluttered Scenes

Figures 8-6 through 8-8 show the depthmap generated from PRINTS, SHOES and KITCHEN, which are nonplanar scenes of cluttered objects, roughly in the order of complexity. Multi-channel coded aperture was used to capture a single image of the scene in each case. See Appendix D for more descriptions, views and larger-sized outputs. The results demonstrate that the multi-channel coded aperture successfully discriminates depths at the given granularity, even in presence of untextured, translucent, or reflective objects. We note that the datasets were captured with natural lighting, as opposed to the calibration that occurred indoor, and that minimal normalized deconvolution error was used to generate the initial depth estimate.

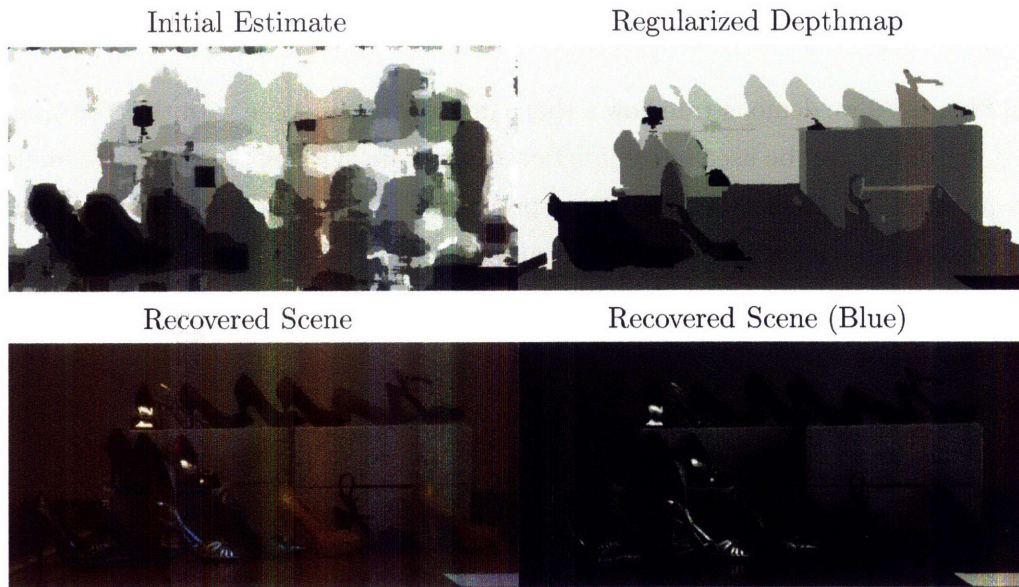


Figure 8-7: Depth and scene recovery on SHOES.

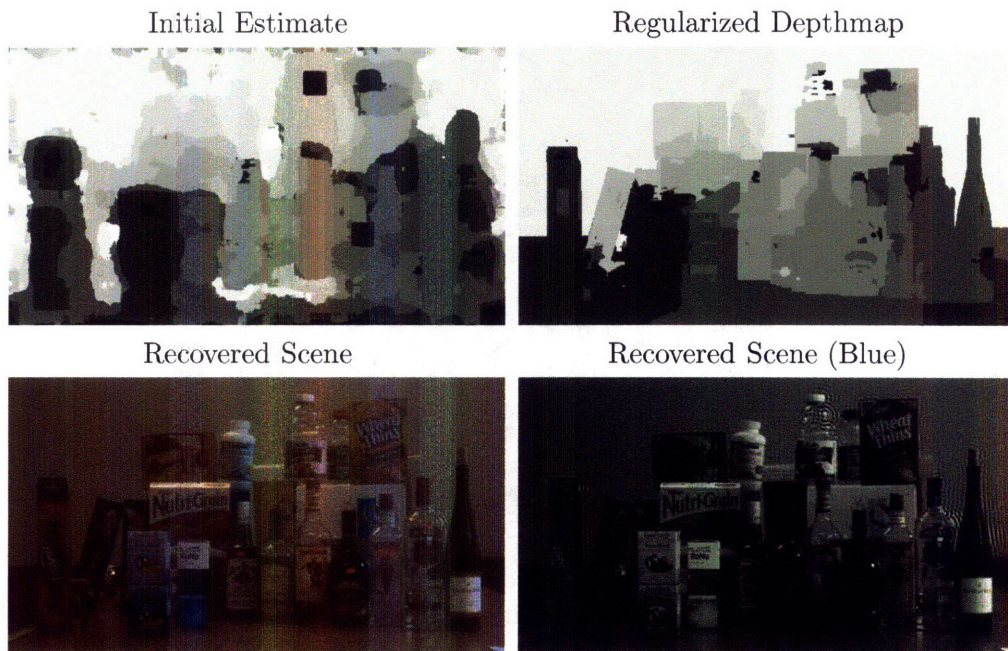


Figure 8-8: Depth and scene recovery on KITCHEN.

8.5 Applications

In Figure 8-9 we demonstrate how a sharp imagery of the scene with the corresponding depthmap can be used in conjunction to generate synthetic results that simulate re-focusing. In theory, the same post-processing algorithms discussed by Levin et al[14], such as viewpoint translation, can be applied.



Figure 8-9: Demonstration of simulated re-focusing using the sharp image and depthmap, generated from a single capture. Left: Scene refocused at 3.1m. Middle: Scene refocused at 2.6m. Right: Scene refocused at 2.1m.

8.6 System Performance

The multi-channel coded aperture incurs a one-time cost of calibrating the kernels, and learning the classifier for the deconvolution error. Per each image captured, deconvolution at each depth $k \in K$ is performed. The deconvolution process scales linearly with the granularity of depths, or the size of K , and also linearly with the number of pixels in the image. Each deconvolution process scales linearly with the number of iterations in IRLS and Conjugate Gradient method. Regularization takes negligible time compared to deconvolution.

With the current set of parameters (4 iterations of IRLS, 50 iterations of Conjugate Gradient method, 12 depths), it takes roughly 0.0124 seconds per pixel, or 1.4 hours per megapixel on a single 2.4GHz machine. In practice, we parallelized the process so that a 2-megapixel image could finish well within the hour.

8.7 Discussion

We have demonstrated the multi-channel coded aperture in practice and its reconstruction and depth discrimination capability. The result is a sharp imagery of scene, along with a regularized depthmap, which then can be combined for various post-processing algorithms.

There are several benefits throughout the depthmap-generation pipeline from using a pinhole filter. First, the complexity and uncertainty in aperture fabrication are reduced. Second, kernel calibration becomes easier since image registration can be performed accurately with the sharp channels. (See Appendix C.) Third, the resulting image is visually pleasant, even with the blue channel blurred. Fourth, the sharp channel aids in the deconvolution of the blurred channel, and ultimately in the depth discrimination. The synthetic datasets demonstrate the superiority of multi-channel model at various noise levels.

On the other hand, the pinhole filter requires additional exposure time to compensate for the small area. The attenuation from stacking color filters to achieve the necessary gain across channels also adds to the exposure time. In practice, the pinhole code requires a couple extra orders of magnitude in exposure time, almost hundred-fold. For a static scene, this amounts to about 5 seconds. A wiser choice

of the filter set could foreseeably overcome this limitation and achieve reasonable performance with subsecond exposure time. Secondly, the physical material for the color filter should be carefully chosen, as in practice the internal reflections within the lens due to color filters add noise and reduce dynamic range. The issue is evident when the results from physical scenes are compared to those from synthetic scenes. It remains to be seen how much of this can be attributed to the choice of physical material.

Chapter 9

Conclusion and Future Work

This chapter summarizes the performance of multi-channel coded aperture, along with its limitations and directions for future development.

9.1 Successes

In this thesis, the multi-channel coded-aperture photography, which captures the scene that is blurred with particular kernels in each channel, was demonstrated both in theory and in practice, generalizing the results of the single-channel coded-aperture photography. The accompanying mathematics and algorithms for recovering a sharp imagery of the scene along with a depthmap were also developed and presented.

The generalization onto multiple filters enables a more diverse choice of filters, and in particular, the use of the pinhole filter in conjunction with a nontrivial code could naturally generate sharp images, while the blurred channels could be reconstructed with help of the unblurred channels. The reconstructed images of the scenes indicate much improved resolution over their single-channel counterparts, thanks to the sharp red and green channels. On test scenes at distances ranging from 2.10m to 3.20m, a span of 1.10m, we could accurately extract depth with an average error around 0.1m *prior* to regularization. The system is decently robust to cluttered scenes, able to distinguish between items at adjacent depths.

9.2 Limitations

The version of multi-channel coded aperture suffers from the following limitations. First, the pinhole filter requires much higher exposure than regular lens or coded aperture with nontrivial filters. In practice, exposure larger than a second is impractical for dynamic scenes or portraits. Second, the color filter used in fabrication of the coded aperture creates internal reflections within the lens, adding a layer of noise and reducing dynamic range. Third, a more formal statistical study of correlation among color channels will be necessary to suggest a more appropriate prior, if possible. While intuitive and simplistic, the dependent prior may not capture exactly the co-occurrent sparsity in the channels. Fourth, the runtime of the system as a whole prohibits rapid development. Fifth, the range of depths distinguished by the current version (with Canon EF 50mm f/1.8 lens) is limited to distances between 2.10m and 3.20m, at 0.10m increment.

9.3 Future Work

The main focus of future work is bridging the gap between the results on synthetic datasets and physical datasets. To replicate the success of multi-channel coded aperture on synthetic datasets, we should aim to realize the model faithfully in implementation, which hopefully can be achieved by better selection of material and more precise control in the fabrication process. A study of spectral properties of materials is necessary to properly eliminate or model the reflections and noise from color gels.

At the same time, we would like to improve our overall methodology. First, a more comprehensive set of filter combinations would be considered in order to see if substituting the pinhole filter with a nontrivial filter would improve depth discrimination and exposure time without sacrificing image detail.

Second, our current estimate of the likeliest depth using deconvolution error could be examined more closely. An alternative approximation method may be possible, or a detailed study of the deconvolution error could suggest a better class of classifiers, or perhaps reveal non-linearity.

Next, a careful analysis of tradeoff between exposure time and reconstruction

error would help determine the ideal combination of filters. The criteria for filter selection should incorporate the exposure time or the complexity in physical fabrication, which could be modelled into the existing framework.

On the side of implementation, conversion to a moving aperture that scales may be possible to increase the range further, where the pre-computed kernels could be inferred and stored with each image taken by the camera itself as metadata.

Lastly, the deconvolution process could benefit from the regularized depthmap, as the assumption of locally planar scene breaks down at the interfaces between depths. A joint or iterative solution would improve both the reconstruction error and depth classification accuracy. A model for deconvolution of piecewise planar scene would be introduced.

THIS PAGE INTENTIONALLY LEFT BLANK

Bibliography

- [1] Adams, J., Parulski, K., and Spaulding K. *Color processing in digital cameras*. In *IEEE Micro* 18, No. 6 (1998), pp. 20-31.
- [2] Adelson, E. H., Wang. J. Y. A. *Single lens stereo with a plenoptic camera*. In *PAMI*, IEEE Trans. on Pattern Analysis and Machine Intelligence 14 (1999), pp. 99-106.
- [3] Axelsson, P. Processing of laser scanner data-algorithms and applications. In *ISPRS Journal of Photogrammetry and Remote Sensing* 54 (1999), pp. 138-147.
- [4] Boykov, Y. and Kolmogorov, V. *An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision*. In *PAMI*, IEEE Trans. on Pattern Analysis and Machine Intelligence 26 (2004), pp. 1124-1137.
- [5] Chaudhuri, S., Rajagopalan, A. *Depth from defocus: A real aperture imaging approach*. Springer-Verlag, New York (1999).
- [6] Chuang, Y., Curless, B., Salesin, D. H., Szeliski, R. *A Bayesian approach to digital matting*. In *CVPR*, IEEE Conf. on Computer Vision and Pattern Recognition (2001).
- [7] Georgiev, T. et al. *Spatio-angular resolution tradeoffs in integral photography*. In *Rendering Techniques 2006: 17th Eurographics Workshop on Rendering* (2006), pp. 263-272.
- [8] Grossmann, P. *Depth from focus*. In *Pattern Recognition Letters* 5, 1 (1987), pp. 63-69.
- [9] Gunturk, B. K. et al. *Demosaicking: color filter array interpolation in single chip digital cameras*. In *IEEE Signal Processing Magazine* (2005).
- [10] Hasinoff, S. W., Kutulakos, K. N. *Confocal stereo*. In *ECCV*, European Conf. on Computer Vision (2006), pp. 620-634.
- [11] Kullback, S. and Leibler, R. A. *On information and sufficiency*. In *Annals of Mathematical Statistics* 22 (1951), pp. 79-86.

- [12] Kraus, K., Pfeifer, N. *Determination of terrain models in wooded areas with airborne laser scanner data*. In *ISPRS Journal of Photogrammetry and Remote Sensing* 53 (1998), pp. 193203.
- [13] Levin, A., Fergus, R., Durand F., Freeman, W. *Deconvolution using natural image priors*. Supplementary document to [14].
- [14] Levin, A., Fergus, R., Durand F., Freeman, W. *Image and depth from a conventional camera with a coded aperture*. In *SIGGRAPH*, ACM Transactions on Graphics (2007).
- [15] Levin, A., Freeman. W., Durand, F. *Understanding camera trade-offs through a Bayesian analysis of light field projections*. In *ECCV*, European Conf. on Computer Vision (2008).
- [16] Levin, A., Rav-Acha, A., Lischinski, D. *Spectral matting*. In *CVPR*, IEEE Conf. on Computer Vision and Pattern Recognition (2007).
- [17] Levoy, M., Ng, R., Adama, A., Footer, M., Horowitz, M. *Light field microscopy*. In *SIGGRAPH*, ACM Transactions on Graphics (2006), pp. 924-934.
- [18] Meer, P. *Robust techniques for computer vision*. In *Emerging Topics in Computer Vision* (2004).
- [19] Ng, A., Levoy, M., Bredif, M., Duval, G., Horowitz, M., Hanrahan, P. *Light field photography with a handheld plenoptic camera*. Stanford University Computer Science Tech Report CSTR 2005-02.
- [20] Olshausen, B. A., Field, D. J. *Emergence of simple-cell receptive field properties by learning a sparse code for natural images*. *Nature* 381 (1996), pp. 607-609.
- [21] Pentland, A. P. *A new sense for depth of field*. In *PAMI*, IEEE Trans. on Pattern Analysis and Machine Intelligence 9 (1987), pp. 523-531.
- [22] Roth, S., and Black, M. *Fields of experts: A framework for learning image priors*. In *CVPR*, IEEE Conf. (2005), pp. 860-867.
- [23] Saxena, A., Sun, M., Ng, A. *Learning 3-D scene structure from a single still image*. In *ICCV*, IEEE Int'l Conf. on Computer Vision, workshop on 3D Representation for Recognition (2007).
- [24] Scharstein, D., Szeliski, R. *A taxonomy and evaluation of dense two-frame stereo correspondence algorithms*. In *IJCV*, Int'l Journal of Computer Vision 47 (2002), pp. 7-42.
- [25] Veeraraghaven, A., Raskar, R., Agrawal, A., Mohan, A., Tumblin, J. *Dappled photography: mask enhanced cameras for heterodyned light fields and coded aperture refocusing*. In *SIGGRAPH*, ACM Trans. on Graphics 2007.

- [26] Weiss, Y. and Freeman, W. *What makes a good model of natural images?* In *IEEE CVPR* (2007).
- [27] Wikipedia contributors, “Bayer filter,” Wikipedia, The Free Encyclopedia, http://en.wikipedia.org/w/index.php?title=Bayer_filter&oldid=201921114 (accessed May 11, 2008).

THIS PAGE INTENTIONALLY LEFT BLANK

Appendix A

Useful Theorems

A.1 Parseval's Relation

Parseval's Relation correlates the energy of a signal in both primal and Fourier domains. Let x be a discrete signal with n components, and let X be its Fourier transform. Then,

$$\|x\|^2 = \frac{\|X\|^2}{n}. \quad (\text{A.1})$$

A.2 Kullback-Liebler Distance

The Kullback-Liebler Distance measures the dissimilarity between two distributions[11].

$$D_{KL}(P, Q) = \int P(x) \log \frac{P(x)}{Q(x)} dx. \quad (\text{A.2})$$

One can show that the KL-divergence takes a particularly simple form when the two distributions are both multi-variate normal distributions.

Theorem A.2.1. *Let P, Q be two n -dimensional, zero-mean Gaussian distributions with covariance Ψ_P, Ψ_Q , respectively. Then,*

$$D_{KL}(P, Q) = \frac{-\log \frac{|\Psi_P|}{|\Psi_Q|} - 1 + \sum \Psi_P \cdot * \Psi_Q^{-1}}{2}.$$

Proof.

$$\begin{aligned}
D_{KL}(P, Q) &= \int P(x)(\log P(x) - \log Q(x))dx \\
&= E_{x \sim P} \left[\left(-\frac{n}{2} \log(2\pi) - \frac{\log |\Psi_P|}{2} - \frac{x^T \Psi_P^{-1} x}{2} \right) \right. \\
&\quad \left. - \left(-\frac{n}{2} \log(2\pi) - \frac{\log |\Psi_Q|}{2} - \frac{x^T \Psi_Q^{-1} x}{2} \right) \right] \\
&= -\frac{1}{2} \log \frac{|\Psi_P|}{|\Psi_Q|} + E_{x \sim P} \left[\sum -\frac{(xx^T) \cdot (\Psi_P^{-1} - \Psi_Q^{-1})}{2} \right] \\
&= \frac{-\log \frac{|\Psi_P|}{|\Psi_Q|} - \sum \Psi_P \cdot (\Psi_P^{-1} - \Psi_Q^{-1})}{2}. \\
&= \frac{-\log \frac{|\Psi_P|}{|\Psi_Q|} - n + \sum \Psi_P \cdot \Psi_Q^{-1}}{2}.
\end{aligned}$$

The last line follows from that the component-wise product of a symmetric matrix with its own inverse sums up to its dimension. The component-wise products can be arranged and collected to form the diagonal elements, which by definition are all 1's. \square

Corollary A.2.1. *If P, Q additionally have independent components, that is, Ψ_P, Ψ_Q are diagonal, then the KL-divergence is given by,*

$$D_{KL}(P, Q) = \frac{\sum_i \left(\frac{\Psi_P(i, i)}{\Psi_Q(i, i)} - \log \frac{\Psi_P(i, i)}{\Psi_Q(i, i)} - 1 \right)}{2}.$$

Proof. The proof follows from Theorem A.2.1 directly. \square

A.3 Conjugate Gradient Method

Conjugate Gradient Method gives an approximate solution to the equation of the form $\mathbf{A}x = \mathbf{b}$, where \mathbf{A} is self-adjoint. Table A.1 details the algorithm.

There are two caveats in deploying the Conjugate Gradient method to achieve multi-channel deconvolution. First, the Conjugate Gradient method is an approx-

- 1: Choose x_0 as the initial solution.
- 2: $r_0 \leftarrow \mathbf{b} - \mathbf{A}x_0$.
- 3: $p_0 \leftarrow r_0$.
- 4: For $t = 1, 2, 3, \dots$,
- 5: $\alpha_t \leftarrow \frac{\|r_{t-1}\|^2}{p_{t-1}^T \mathbf{A} p_{t-1}}$.
- 6: $r_t \leftarrow r_{t-1} - \alpha_t \mathbf{A} p_{t-1}$.
- 7: $\beta_t \leftarrow \frac{\|r_t\|^2}{\|r_{t-1}\|^2}$.
- 8: $x_t = x_{t-1} + \alpha_t p_{t-1}$.
- 9: $p_t = r_t + \beta_t p_{t-1}$.

Table A.1: Algorithm for the Conjugate Gradient method.

imation and sometimes fails to converge completely within reasonable time limit. Second, if the filters are significantly different in the three channels, the solutions in the three channels may converge at different rates. In practice, the Conjugate Gradient seems to prefer to optimize the channel with the simpler kernel. Therefore, when one or more filters are pinholes, we solve for those channels separately using the single-channel independent prior, before solving for the remaining channels using the multi-channel prior.

THIS PAGE INTENTIONALLY LEFT BLANK

Appendix B

Derivation of Numerical Methods

We derive the update rule for solving Equation (5.2) for both Newton's Method and Iteratively Re-Weighted Least Squares (IRLS), using the independent prior.

B.1 Newton's Method

Let $f(x)$ be the objective function we wish to optimize, that is,

$$f(x) = \frac{\|D_k \bar{x} - \bar{y}\|^2}{2\sigma^2} + \frac{\alpha}{2} \left(\sum_{i,j} \|g_{j \rightarrow}^i \bar{x}^R\|^\rho + \|g_{j \rightarrow}^i \bar{x}^G\|^\rho + \|g_{j \rightarrow}^i \bar{x}^B\|^\rho \right). \quad (\text{B.1})$$

The second-order Taylor expansion of $f(x)$ centered at x_t is,

$$f(x_t + \Delta x) \simeq f(x_t) + \nabla f(x_t) \Delta x + \frac{\Delta x^T H \Delta x}{2},$$

where H is the Hessian for $f(x_t)$. Taking its derivative and setting it to zero, we get

$$\nabla f(x_t) + \Delta x^T H = 0,$$

or alternatively,

$$\nabla f(x_t) + (\bar{x}_{t+1} - \bar{x}_t)^T H = 0. \quad (\text{B.2})$$

We now compute $\nabla f(x_t)$. Differentiating $f(x_t)$ with respect to the j -th component of x_t , we get

$$\sum_i 2(D_{k_i \rightarrow \bar{x}_t} - \bar{y}_i)A_{ij} + \rho\alpha\sigma^2 \sum_{i,l} (\|g_{l \rightarrow \bar{x}_t}^i\|^{\rho-1} \text{sign}(g_{l \rightarrow x_t}^i) g_{lj}^i).$$

Collecting this over all j , we obtain

$$\nabla f(x_t) = 2(D_k \bar{x}_t - \bar{y})^T D_k + \rho\alpha\sigma^2 \left[\sum_i \left(\left[|g_{l \rightarrow}^i|^{\rho-1} \text{sign}(g_{l \rightarrow x_t}^i) \right]_l g^i \right) \right]_C. \quad (\text{B.3})$$

The Hessian is obtained by differentiating $\nabla f(x_t)$ once more:

$$H = 2D_k^T D_k + \rho(\rho - 1)\alpha\sigma^2 K, \quad (\text{B.4})$$

where $K = \text{diag}_C \left(\sum_i (g^i)^T \text{diag}_l \left(|g_{l \rightarrow x_t}^i|^{\rho-2} \right) g^i \right)$. Then, plugging in (B.4) and (B.3) into (B.2) and rearranging the terms yield,

$$\begin{aligned} H^T \bar{x}_{t+1} &= H \bar{x}_t - \nabla f(x_t)^T \\ &\implies (2D_k^T D_k + \rho(\rho - 1)\alpha\sigma^2 K) \bar{x}_{t+1} = (2D_k^T D_k + \rho(\rho - 1)\alpha\sigma^2 K) \bar{x}_t \\ &\quad - 2D_k^T (D_k \bar{x}_t - \bar{y}) - \rho\alpha\sigma^2 \left[\sum_i \left(\left[|g_{l \rightarrow}^i|^{\rho-1} \text{sign}(g_{l \rightarrow x_t}^i) \right]_l g^i \right) \right]_C. \\ &\implies (2D_k^T D_k + \rho(\rho - 1)\alpha\sigma^2 K) \bar{x}_{t+1} = (2D_k^T D_k + \rho(\rho - 1)\alpha\sigma^2 K) \bar{x}_t \\ &\quad - 2D_k^T (D_k \bar{x}_t - \bar{y}) - \rho\alpha\sigma^2 \left[\sum_i \left(\left[|g_{l \rightarrow}^i|^{\rho-2} (g_{l \rightarrow x_t}^i) \right]_l g^i \right) \right]_C \\ &\implies (2D_k^T D_k + \rho(\rho - 1)\alpha\sigma^2 K) \bar{x}_{t+1} = (2D_k^T D_k + \rho(\rho - 1)\alpha\sigma^2 K) \bar{x}_t \\ &\quad - 2D_k^T (D_k \bar{x}_t - \bar{y}) - \rho\alpha\sigma^2 K \bar{x}_t \\ &\implies (2D_k^T D_k + \rho(\rho - 1)\alpha\sigma^2 K) \bar{x}_{t+1} = 2D_k^T \bar{y} + \rho(\rho - 2)\alpha\sigma^2 K \bar{x}_t \\ &\implies \left(D_k^T D_k + \frac{\rho(\rho - 1)}{2} \alpha\sigma^2 K \right) \bar{x}_{t+1} = D_k^T \bar{y} + \frac{\rho(\rho - 2)}{2} \alpha\sigma^2 K \bar{x}_t, \end{aligned}$$

as desired.

B.2 Iteratively Re-Weighted Least Squares

IRLS minimizes $\sum f(A_{i \rightarrow x} - B_i)$, where f is a function that crosses the origin, by iteratively posing the problem as a least-square optimization. In fact, if f is just the standard square function, IRLS simplifies to the standard least-square problem. Given x_t , the iterative step is to solve

$$A^T \text{diag}_i \left(\frac{f'(A_{i \rightarrow x} - B_i)}{A_{i \rightarrow x_t} - B_i} \right) (Ax_{t+1} - B) = 0,$$

for x_{t+1} . This ensures that, if convergent, x_∞ will be at a local optimum of the objective function.

The deconvolution with independent prior minimizes $\sum_{i,j,C} |g_{j \rightarrow x^C}^i|^\rho$, in addition to $(D_k \bar{x} - \bar{y})^2$. We can pose this problem as minimizing $\sum f(g_{j \rightarrow x^C}^i)$ where $f(x) = |x|^\rho$, with some linear parts. Then, IRLS iteratively solves

$$2D_k^T D_k (\bar{x} - \bar{y}) + \sum_{i,C} (g^i)^T \text{diag}_j \left(\frac{f'(g_{j \rightarrow x^C}^i)}{g_{j \rightarrow x^C}^i} \right) g^i x^C = 0,$$

where $f(x) = |x|^\rho$.

Evaluating the derivatives and simplifying the expressions yield,

$$2D_k^T D_k (\bar{x} - \bar{y}) + \sum_{i,C} (g^i)^T \text{diag}_j \left(\frac{|g_{j \rightarrow x^C}^i|^{\rho-1} \text{sign}(g_{j \rightarrow x^C}^i)}{g_{j \rightarrow x^C}^i} \right) g^i x^C = 0,$$

which is equivalent to Equation (5.4) after further cancellations in the main fraction.

THIS PAGE INTENTIONALLY LEFT BLANK

Appendix C

Calibration Setup

We present the physical apparatus and the accompanying parameters for calibrating the kernels of the multi-channel coded aperture, along with the algorithm to estimate the kernels.

C.1 Physical Apparatus

The physical setup consists of a horizontal rail with distance markers, a movable carriage that slides along the rail, a RAW-capable camera, and a calibration pattern. The calibration pattern, shown in Figure C.1, is a MATLAB-generated random binary noise, with two distinct resolutions to ensure that the sharp image taken at each depth will have resolution not too far from that of the camera. The pattern is pasted onto a wall and the rail is immobilized on the ground, with its principle axis orthogonal to the wall, as in Figure C.1. This setup allows accurate horizontal displacement of the camera relative to the pattern.

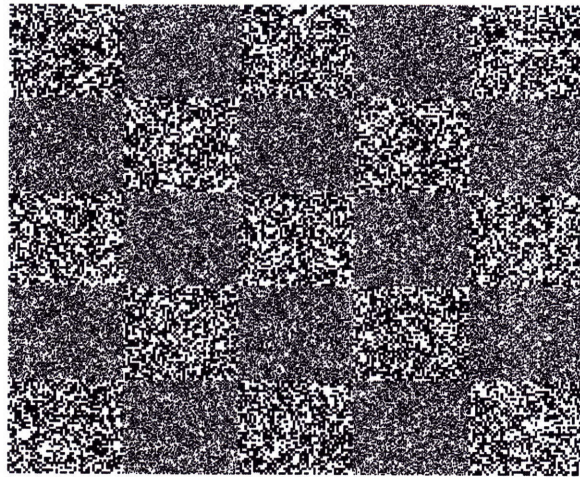


Figure C-1: Calibration pattern. The pattern consists of random binary noise in two scales.

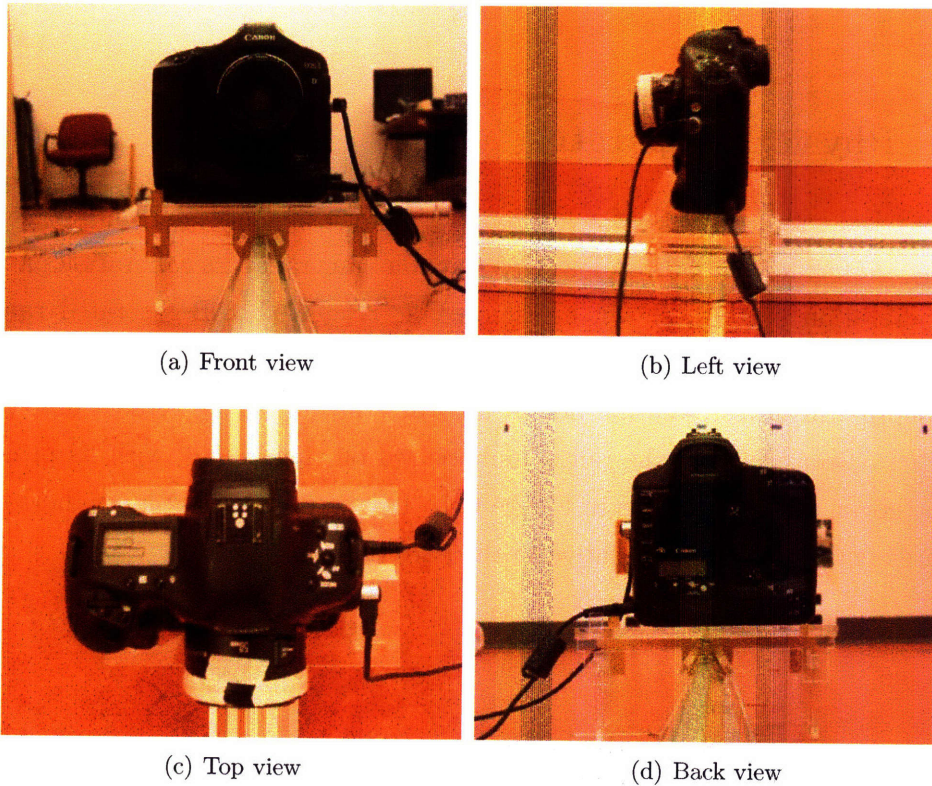


Figure C-2: Physical apparatus for calibration. The apparatus includes a horizontal rail with distance markers, a movable carriage constructed of plastic, a RAW-capable camera.

C.2 Camera Settings

All images obtained for calibration were taken with a Canon EOS-1D Mk II under a single set of parameters to produce consistent images. Table C.1 summarizes these parameters. Note that the native aperture size is kept at the maximal f/1.8, as the insertion of the coded aperture prevents the existing blades from turning. We estimate that the pinhole filter is closest to the smallest available aperture size, f/22. Under desirable lighting conditions, exposure lengths as short as 4 seconds can

Parameter	Value
Aperture size	f/1.8
ISO	100
Exposure	15 sec.
White balance	White fluorescent light
Format	RAW

Table C.1: Settings for Canon EOS-1D Mk II.

generate reasonable outputs. In calibration, high signal-to-noise ratio is desirable and the exposure was lengthened accordingly.

The output of the Canon EOS-1D Mk II is a proprietary format `.CR2`, which can be cast into MATLAB-readable portable pixmap (`.PPM`) via a popular conversion program `dcraw`, version 1.403. As provided, `dcraw` automatically demosaicks the RAW image, but the source code is available¹ and can be easily tweaked to yield the undemosaicked image. Table C.2 summarizes the flags invoked in the conversion.

Flag	Function
-4	Generates 16-bit linear output
-w	Uses the camera white balance
-h	Generates half-sized image

Table C.2: Flags for `dcraw` RAW conversion. The given settings generate a linear, Bayer-masked image as in the multi-channel model.

¹See <http://www.cybercom.net/~dcoffin/dcraw/>, courtesy of Dave Coffin.

C.3 Registration Procedure

Inferring the respective kernel at each depth requires not only the blurred image y_k taken with the coded aperture, but also the sharp groundtruth. The simplest approach to obtaining the groundtruth is to capture the same scene with an unmodified lens with the appropriate focal distance. However, the task of capturing the exactly same scene is infeasible: no two lenses are optically identical, and swapping the lens may perturb the camera setup. A more viable alternative is to capture the sharp image once at a close distance, denoted by x_* , and interpolate groundtruth at each subsequent depth.

Ideally, the images of the scene captured at various depths are related through a scaling operation. In practice, mechanical uncertainties and instability in the apparatus setup introduce translation, rotation, or perspective transform to some extent, resulting in deviations up to several pixels. Therefore, the relation between two groundtruth images is best modeled by a perspective transform.

The transform itself cannot be inferred from measuring the mechanical parameters, which are difficult to measure or maintain. Rather, we seek the transform that would best align the groundtruth x_* with the blurred image y_k , through the process of *registration*. Perceptually salient keypoints, such as corners of high-contrast rectangles, are picked from both images. The keypoint pairs do not correspond exactly, so we iteratively update the keypoints on the blurred image until the perspective transform that minimizes error is visually satisfactory. Once the correct keypoints are recovered, one can solve for the perspective transform that minimizes the sum of squared deviation, and apply the transform to x_* to obtain x_k . Table C.3 summarizes the algorithm.

We found experimentally that the intermediate-level loop needs to iterate about three times to achieve subpixel accuracy, as long as the initial estimates q_1, \dots, q_l are not too far off.

Another consideration is that because x_* is observed through a Bayer filter, we must first recover the full-channel version before applying the transformation T to generate groundtruths at other depths. In practice, because the calibration pattern is very noisy, using demosaicking to recover the full image is unreliable. Rather, x_* should be taken at much closer distance and be downsampled to bypass the

- 1: Select and fix keypoints p_1, \dots, p_l in x_* .
- 2: For each depth k ,
- 3: Visually select q_1, \dots, q_l in y_k that correspond to p_1, \dots, p_l . In other words, the patch centered at q_i in the blurred image y_k should contain roughly the same portion of the scene as the patch centered at p_i in the groundtruth x_* .
- 4: Iterate until alignment:
- 5: Compute a perspective transform T that minimizes $\sum_i \|T(p_i) - q_i\|^2$, using the MATLAB command `cp2tform`.
- 6: For each point q_i ,
- 7: Compute a translation L such that $L \circ T(p_i) = q_i$.
- 8: Overlay $L \circ T(x_0)$ on y_k .
- 9: Compute the necessary translation R so that the two images align visually. In other words, $L \circ T(x_*) \simeq R(y_k)$ near the keypoint q_i . Or, one can solve for R to minimize the squared error, if the two images have the same white balance.
- 10: Update the value of q_i to $R(q_i)$.
- 11: Compute a perspective transform T that minimizes $\sum_i \|T(p_i) - q_i\|^2$, using the MATLAB command `cp2tform`.
- 12: Set $x_k = T(x_*)$.

Table C.3: Registration algorithm for aligning images captured at varying depths.

reconstruction procedure.

C.4 Solving for Kernels

Once the groundtruth x_k and the blurred image y_k are available, the kernels f_k^R, f_k^G, f_k^B can be computed to minimize the sum of squared error $\|\omega^C * (y_k - f * x_k^C)\|^2$ for $C \in \{R, G, B\}$. Empirically, some regularization is required to prevent noise from excessively affecting the kernel estimate, so we employ the quadratic programming solver available in MATLAB. In other words, we minimize the sum of squared error while respecting the constraint that the kernels are component-wise nonnegative. See Table C.4 for the exact algorithm. Note that if the pixel values shift due to unwanted reflections in the lens, the model should be modified accordingly.

- 1: For each depth k , and each channel $C \in \{R, G, B\}$,
- 2: For each pixel $y_k(i)$,
- 3: If $\omega^C(i) = 1$,
- 4: Express the i -th component of $y_k - f * x_k^C$ as $B_i - A_i \cdot f(\cdot)$, where $B_i = y_k(i)$ and A_i is the row vector corresponding to the neighborhood of $x_k(i)$.
- 5: Else,
- 6: Set B_i to zero, and A_i to the all-zero row vector with the same number of elements as f .
- 7: Define A to be the vertical concatenation of $\{A_1, A_2, \dots\}$.
- 8: Define B to be the vertical concatenation of $\{B_1, B_2, \dots\}$.
- 9: Minimize $\|B - A \cdot f(\cdot)\|^2$ with the constraint that each component of $f(\cdot)$ is nonnegative. The applicable MATLAB command for this is `quadprog(A'*A, -A'*B, [], [], [], [], zeros(numel(f), 1), [], [])`.
- 10: Set $f_k^C = f$.

Table C.4: Quadratic Programming for inferring the kernels from the blurred scene, along with the groundtruth.

Appendix D

Test Scenes

D.1 Synthetic Scenes

The two synthetic test scenes **SYNTHETIC1** and **SYNTHETIC2** are shown here, at full resolution. The actual datasets consist of twelve images of the same scene blurred with the kernels at all twelve scales, with Gaussian noise overlaid.



Figure D-1: **SYNTHETIC1** test scene.



Figure D-2: SYNTHETIC2 test scene.

D.2 Planar Scenes

The two planar test scenes BUILDING and POSTERS are shown here, captured with a regular lens at 2.0m. The actual datasets consist of twelve images of the same scene taken with the single- and multi-channel coded aperture (focused at 2.0m) taken at distances ranging from 2.10m to 3.20m.



Figure D-3: BUILDING test scene.



Figure D-4: POSTERS test scene.

D.3 Cluttered Scenes

The three test scenes PRINTS, SHOES, KITCHEN are shown here, captured with a regular lens at 2.0m from the front, and also at an alternate angle to reveal the depths of the objects. The actual datasets consist of a single image taken with the multi-channel coded aperture. We also show the depthmaps from the previous chapters in higher resolution.

D.3.1 Views



Figure D-5: Views of PRINTS test scene.



Figure D-6: Views of SHOES test scene.



Figure D-7: Views of KITCHEN test scene.

D.3.2 Depthmaps

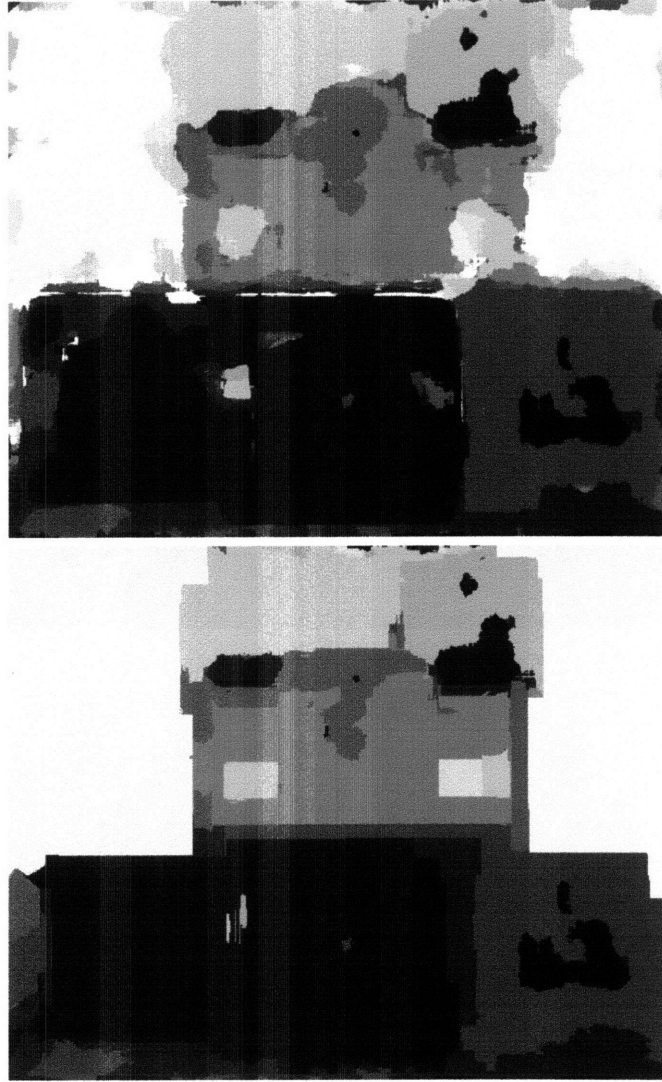


Figure D-8: Depthmaps from PRINTS test scene. Top: initial estimate. Bottom: regularized depthmap.



Figure D-9: Depthmaps from SHOES test scene. Top: initial estimate. Bottom: regularized depthmap.

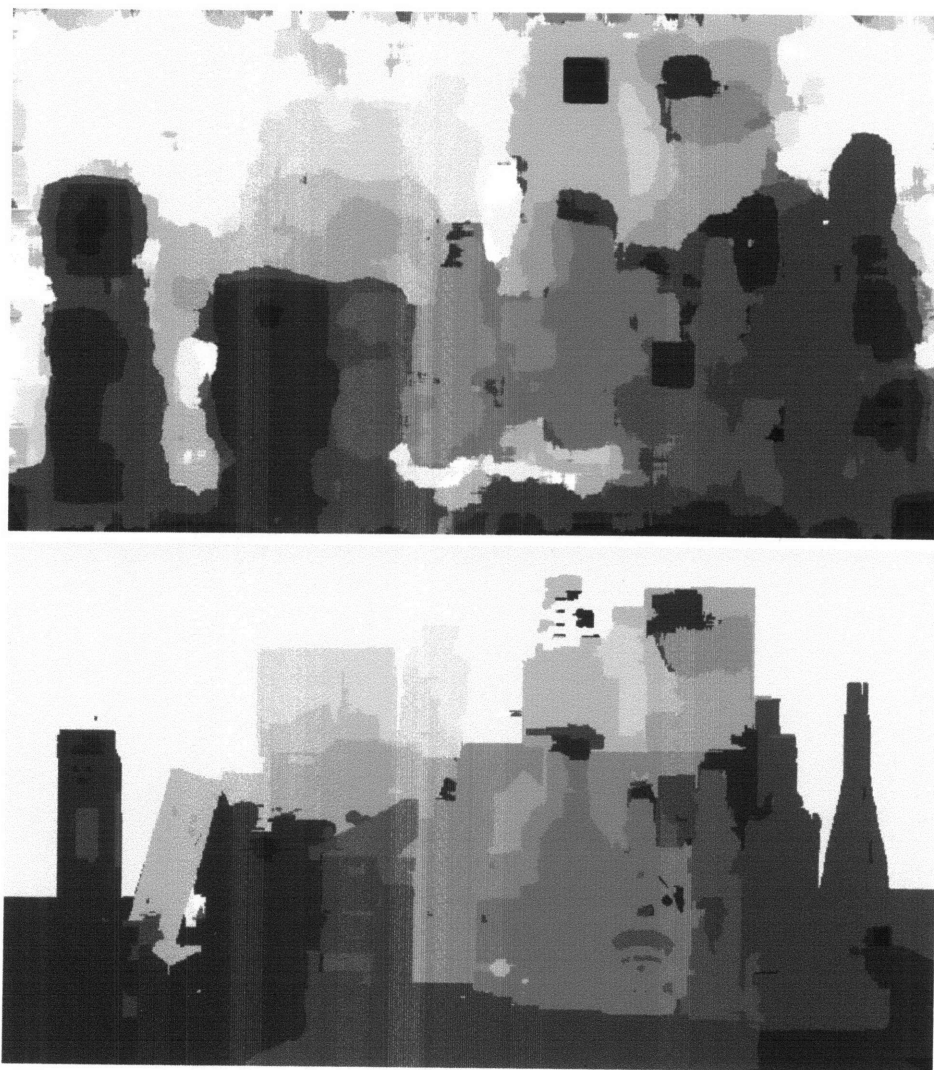


Figure D-10: Depthmaps from KITCHEN test scene. Top: initial estimate. Bottom: regularized depthmap.