

# **Structure-Aware Computational Imaging**

by

Felix Heide

MSc, University of Siegen, 2011

BSc, University of Siegen, 2009

A THESIS SUBMITTED IN PARTIAL FULFILLMENT  
OF THE REQUIREMENTS FOR THE DEGREE OF

**Doctor of Philosophy**

in

THE FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES  
(Computer Science)

The University of British Columbia  
(Vancouver)

November 2016

© Felix Heide, 2016



# Abstract

While traditional imaging systems *directly* measure scene properties, *computational* imaging systems add computation to the measurement process, allowing such systems to extract non-trivially encoded scene features. This dissertation demonstrates that exploiting structure in this process allows to even recover information that is usually considered to be completely lost. Relying on temporally and spatially convolutional structure, we extract two novel image modalities that were essentially “*invisible*” before: a new temporal dimension of light propagation, and a new per-pixel radial velocity dimension, both obtained using consumer Time-of-Flight cameras. These two novel types of images represent first steps toward the inversion of light transport. Specifically, we demonstrate that Non-Line-of-Sight imaging and imaging in scattering media can be made feasible with additional temporal information. Furthermore, structure-aware imaging also represents a completely new approach to traditional color image processing. We show that classical hand-crafted image processing pipelines can be replaced by a single optimization problem exploiting image structure. This approach does not only outperform the state-of-the-art for classical image processing problems, but enables completely new color camera designs. In particular, we demonstrate camera designs with radically simplified optical systems, as well as novel sensor designs. The computation for all imaging problems from this dissertation relies on Bayesian inference using large-scale proximal optimization methods. We present a mathematical framework and a corresponding domain-specific language to automate the development of efficient, structure-aware solvers, allowing to immediately apply the insights gained in this dissertation to new imaging problems.

# Preface

This dissertation is based on the following publications, which are all collaborative works. For each publication, the contributions of all the authors are listed.

*F. Heide, M. B. Hullin, J. Gregson, and W. Heidrich, Low-Budget Transient Imaging Using Photonic Mixer Devices, ACM Transactions on Graphics (SIGGRAPH), 2013, [1]:* This work is presented in Chapter 3. The initial idea was conceived by Prof. Heidrich, Prof. Hullin, and the author. Prof. Hullin contributed the image formation and the hardware prototype. The inverse problem and the optimization method for its solution were developed by the author. Prof. Heidrich wrote the initial draft of the manuscript. All the authors contributed equally to the writing of the final published article.

*M. O’Toole, F. Heide, L. Xiao, M. B. Hullin, W. Heidrich, and K. N. Kutulakos, Temporal Frequency Probing for 5D Transient Analysis of Global Light Transport, ACM Transactions on Graphics (SIGGRAPH), 2014, [2]:* We discuss this work at the end of Chapter 3. Dr. O’Toole and the author had the initial idea for this work. Prof. Kutulakos and Dr. O’Toole drew the connection to previous work using spatial light modulation. Chapter 3 gives an alternative derivation, which was developed by the author. Furthermore, the author contributed the transient inverse methods, while the 3D reconstructions were performed by Dr. O’Toole. Lei Xiao, Dr. O’Toole, and the author performed the experiments. The prototype camera was developed by Prof. Hullin. All the authors took part in writing the published manuscript.

*F. Heide, L. Xiao, W. Heidrich, and M. B. Hullin, Diffuse Mirrors: 3D Reconstruction from Diffuse Indirect Illumination Using Inexpensive Time-Of-Flight Sensors, Computer Vision and Pattern Recognition (CVPR), 2014, [3]:* Chapter 5 covers this work. The initial idea was conceived by the author. Prof. Hullin suggested the experimental setup and developed the imaging and light source hardware setup. Lei Xiao and the author captured the measurements. The image formation and inverse method were developed by the author. Prof. Heidrich and Prof. Hullin jointly coordinated this project. All the authors contributed to writing the published article.

*F. Heide, L. Xiao, A. Kolb, M. B. Hullin, and W. Heidrich, Imaging in Scattering Media Using Correlation Image Sensors and Sparse Convolutional Coding, Optics Express, 2014, [4]:* This work is presented in Chapter 4. The original idea was developed by Prof. Heidrich and the author. In particular, Prof. Heidrich proposed the experimental setup, while the author developed the convolutional model for transient images as well as the inverse methods. Lei Xiao and the author captured the measurements using an imager provided by Prof. Kolb and a prototype provided by Prof. Hullin. Prof. Heidrich, Lei Xiao, and the author wrote the article.

*F. Heide, W. Heidrich, and G. Wetzstein, Fast and Flexible Convolutional Sparse Coding, Computer Vision and Pattern Recognition (CVPR), 2015, [5]:* This publication is covered in Chapter 6. The initial idea as well as the concrete optimization methods were conceived by the author. Furthermore, the author performed the experimental validation of the presented approach and wrote the initial draft. Prof. Wetzstein wrote a significant portion of the published article that all the authors took part in writing.

*F. Heide, W. Heidrich, M. Hullin, and G. Wetzstein, Doppler Time-of-Flight Imaging, ACM Transactions on Graphics (SIGGRAPH), 2015, [6]:* The initial idea of this work, which is described in Chapter 7, was conceived by Prof. Heidrich and the author. The author developed the image formation model and the capture method based on Orthogonal Frequency-Division Multiplexing. After the first pilot experiments conducted in Prof. Heidrich's laboratory, all final experiments

were conducted in Prof. Wetzstein’s laboratory. Prof. Hullin contributed the camera prototype and many fruitful discussions before conducting the experiments. Prof. Heidrich and Prof. Wetzstein jointly coordinated this project. The published manuscript was written by all the authors.

*S. Shrestha, F. Heide, W. Heidrich, and G. Wetzstein, Computational Imaging with Multi-Camera Time-of-Flight Systems, ACM Transactions on Graphics (SIGGRAPH), 2016, [7]:* We discuss this publication at the end of Chapter 7. Prof. Wetzstein had the initial idea for this work. The author conceived the applications of the proposed system to interference canceling and dynamic Non-Line-of-Sight imaging. Shikhar Shrestha developed the hardware system and captured the experimental data together with the author. Prof. Wetzstein wrote the initial draft of the publication. All the authors took part in writing the final publication.

*F. Heide, M. Rouf, M. B. Hullin, B. Labitzke, W. Heidrich, and A. Kolb, High-Quality Computational Imaging through Simple Lenses, ACM Transactions on Graphics (TOG), 2013, [8]:* This work is discussed in Chapter 8. The initial idea was conceived by the author and Prof. Heidrich. The author proposed the inverse problem and developed the optimization methods as well as their implementations. Mushfiqur Rouf developed the extension to handle low-light regions. The author performed all measurements. The multispectral camera prototype discussed in Section A.11 was developed by Alex Gukov. Prof. Hullin contributed with fruitful discussions. The Point Spread Function estimation was adopted from previous work with Prof. Kolb and Dr. Labitzke. Prof. Heidrich, Prof. Hullin, and the author wrote the manuscript.

*F. Heide, Q. Fu, E. Peng, and W. Heidrich, Encoded Diffractive Optics for Full-Spectrum Computational Imaging, Nature Scientific Reports, 2016, [9]:* The author conceived the idea of this work. Parts of it are discussed in Chapter 8. Furthermore, he proposed the factorization and reconstruction method and implemented the corresponding algorithms. Dr. Fu designed the optical parameters and fabricated the lenses. Yifan Peng performed the Modulation Transfer Function analysis. Dr. Fu

and the author conducted the experiments. Prof. Heidrich coordinated the whole project. All the authors took part in writing the paper.

*F. Heide, M. Steinberger, Y. Tsai, M. Rouf, D. Pająk, D. Reddy, O. Gallo, J. Liu, W. Heidrich, K. Egiazarian, J. Kautz, and K. Pulli, FlexISP: A Flexible Camera Image Processing Framework, ACM Transactions on Graphics (SIGGRAPH Asia), 2014, [10]:* This publication is covered in Chapter 9. The author had the initial idea for this work. Furthermore, the inverse problem as well as the optimization methods were contributed by the author. While the initial implementations were also developed by the author, Yun-Ta Tsai and Prof. Steinberger contributed the efficient GPU implementations. Dr. Reddy and Mushfiqur Rouf provided the experimental results for the color array camera. Dr. Gallo, Dr. Pająk, and Jing Liu performed the interlaced High Dynamic Range experiments. Dr. Kautz performed the burst image denoising experiments. Prof. Egiazarian contributed with many insightful discussions and in the parameter selection. The published manuscript was written by all the authors.

*F. Heide, S. Diamond, M. Nießner, J. Ragan-Kelley, W. Heidrich, and G. Wetzstein, ProxImaL: Efficient Image Optimization using Proximal Algorithms, ACM Transactions on Graphics (SIGGRAPH), 2016, [11]:* This work is presented in Chapter 10. The original idea was conceived by the author. Furthermore, the author contributed the proposed mathematical framework. Steven Diamond and the author developed and implemented the mathematical compiler as well as all solver algorithms. The functions with Halide support were implemented by the author, with help from Dr. Nießner and Dr. Ragan-Kelley. Prof. Wetzstein wrote the initial manuscript draft. Steven Diamond, Prof. Wetzstein, Prof. Heidrich, and the author took part in writing the final publication.

# Table of Contents

<b>Abstract</b> . . . . .	<b>ii</b>
<b>Preface</b> . . . . .	<b>iii</b>
<b>Table of Contents</b> . . . . .	<b>vii</b>
<b>List of Tables</b> . . . . .	<b>xiv</b>
<b>List of Figures</b> . . . . .	<b>xv</b>
<b>Glossary</b> . . . . .	<b>xx</b>
<b>Acknowledgments</b> . . . . .	<b>xxii</b>
<b>1 Introduction</b> . . . . .	<b>1</b>
1.1 Dissertation Structure . . . . .	7
<b>2 Background and Related Work</b> . . . . .	<b>12</b>
2.1 Light . . . . .	12
2.2 The Propagation of Light . . . . .	14
2.2.1 Transmission, Reflection, and Refraction . . . . .	14
2.2.2 Global Light Transport . . . . .	17
2.2.3 Transient Global Light Transport . . . . .	19
2.3 Optics . . . . .	20
2.3.1 Diffraction . . . . .	21
2.3.2 Geometric Optics . . . . .	22

2.3.3	Aberrations . . . . .	24
2.4	Imaging Sensors . . . . .	28
2.4.1	Measuring Irradiance . . . . .	28
2.4.2	CCD and CMOS Sensors . . . . .	31
2.4.3	Sensor and Noise Model . . . . .	33
2.4.4	Color Imaging . . . . .	38
2.4.5	Modified Sensor Designs . . . . .	40
2.5	Time-of-Flight Imaging . . . . .	40
2.5.1	Optical Range Imaging . . . . .	40
2.5.2	Impulse Time-of-Flight Imaging . . . . .	42
2.5.3	Continuous Wave Time-of-Flight Imaging . . . . .	43
2.6	Working Principles of Correlation Image Sensors (CIS) . . . . .	46
2.6.1	Homodyne Measurement . . . . .	47
2.6.2	Heterodyne Measurement . . . . .	48
2.6.3	Analysis in the Spectral Domain . . . . .	49
2.6.4	Sensor and Noise Model . . . . .	51
2.6.5	Multi-Path Interference . . . . .	52
2.6.6	Dynamic Scenes . . . . .	54
2.7	Transient Imaging . . . . .	56
2.7.1	Impulse vs. Continuous Wave Transient Imaging . . . . .	57
2.7.2	Continuous Wave Transient Imaging . . . . .	58
2.8	Inverse Methods . . . . .	58
2.8.1	Inverse Problems . . . . .	59
2.8.2	Statistical Image Recovery . . . . .	59
2.8.3	Linear Models . . . . .	62
2.8.4	Optimization Methods . . . . .	63
<b>3</b>	<b>Transient Imaging . . . . .</b>	<b>66</b>
3.1	Introduction . . . . .	66
3.2	Transient CIS Image Formation . . . . .	67
3.2.1	Generalized CIS Model . . . . .	67
3.2.2	Single Reflection . . . . .	68
3.2.3	Global Illumination . . . . .	69

3.3	Transient Imaging using CIS . . . . .	71
3.3.1	Inverse Problem . . . . .	71
3.3.2	Optimization via Splitting . . . . .	74
3.3.3	Solving the $\mathbf{i}$ -subproblem . . . . .	75
3.4	Setup . . . . .	76
3.5	Results . . . . .	79
3.5.1	Synthetic Examples . . . . .	79
3.5.2	CIS Measurements . . . . .	80
3.6	Spatio-Temporal Modulation . . . . .	83
3.7	Discussion . . . . .	86
<b>4</b>	<b>Imaging in Scattering Media . . . . .</b>	<b>88</b>
4.1	Introduction . . . . .	88
4.2	Sparse Transient Mixture Model . . . . .	90
4.2.1	Are Transient Images Sparse ? . . . . .	90
4.2.2	Sparse Local Model for Transient Light Interaction . . . . .	92
4.3	Transient Convolutional Sparse Coding . . . . .	94
4.3.1	Reformulation in the Spectral Domain . . . . .	95
4.3.2	Synthetic Evaluation . . . . .	96
4.4	Results . . . . .	97
4.4.1	Setup . . . . .	97
4.4.2	Qualitative Results . . . . .	99
4.4.3	Quantitative Results . . . . .	99
4.5	Discussion . . . . .	101
<b>5</b>	<b>Diffuse Mirrors . . . . .</b>	<b>103</b>
5.1	Introduction . . . . .	103
5.2	Image Formation Model . . . . .	105
5.2.1	Assumptions . . . . .	105
5.2.2	Stationary Light Transport . . . . .	106
5.2.3	Transient Light Transport . . . . .	107
5.2.4	Discretization . . . . .	107
5.2.5	Transient CIS Image Formation . . . . .	108



5.3	Reconstruction from Diffuse Indirect Illumination . . . . .	108
5.3.1	Inverse Problem . . . . .	108
5.3.2	Optimization . . . . .	110
5.3.3	Enhancing Volume Sparsity . . . . .	112
5.4	Implementation and Parameter Selection . . . . .	112
5.5	Results . . . . .	113
5.5.1	Setup . . . . .	113
5.5.2	Qualitative Results . . . . .	114
5.5.3	Quantitative Results . . . . .	116
5.6	Discussion . . . . .	118
<b>6</b>	<b>Convolutional Sparse Coding . . . . .</b>	<b>119</b>
6.1	Introduction . . . . .	119
6.2	Mathematical Framework . . . . .	122
6.2.1	Efficient Splitting of the Objective . . . . .	123
6.2.2	Generalization of the Objective . . . . .	124
6.2.3	Biconvex Minimization via Coordinate Descent . . . . .	127
6.2.4	Implementation details . . . . .	127
6.3	Analysis . . . . .	128
6.3.1	Complexity Analysis . . . . .	128
6.3.2	Convergence . . . . .	129
6.4	Learning . . . . .	131
6.4.1	Boundary Conditions . . . . .	131
6.4.2	Learning from Sparse Data . . . . .	131
6.5	Reconstruction . . . . .	133
6.5.1	Validation of Reconstruction . . . . .	133
6.5.2	Non-Normalized Data . . . . .	133
6.5.3	Reconstruction with Known Basis . . . . .	135
6.6	Discussion . . . . .	137
<b>7</b>	<b>Doppler Velocity Imaging . . . . .</b>	<b>138</b>
7.1	Introduction . . . . .	138
7.2	Dynamic CIS Image Formation . . . . .	141

7.3	Doppler Velocity Imaging . . . . .	143
7.3.1	Velocity Imaging via Orthogonal Frequencies . . . . .	143
7.3.2	Simultaneous Range and Velocity . . . . .	147
7.4	Implementation . . . . .	148
7.5	Results . . . . .	153
7.6	Toward the 3D Velocity Field . . . . .	156
7.7	Multi-Camera System . . . . .	159
7.7.1	Multi-Source Interference Cancelling . . . . .	162
7.8	Discussion . . . . .	163
<b>8</b>	<b>Simple Lens Imaging . . . . .</b>	<b>166</b>
8.1	Introduction . . . . .	167
8.2	Image Formation Model . . . . .	170
8.3	Deconvolution Using Cross-Channel Correlation . . . . .	170
8.3.1	Cross-Channel Correlation . . . . .	170
8.3.2	Inverse Problem . . . . .	173
8.3.3	Optimization . . . . .	173
8.3.4	Low-Light Imaging . . . . .	175
8.4	Results . . . . .	177
8.5	Discussion . . . . .	184
<b>9</b>	<b>Flexible Camera Image Processing . . . . .</b>	<b>185</b>
9.1	Introduction . . . . .	186
9.2	Image Formation Model . . . . .	189
9.3	Flexible Image Reconstruction . . . . .	189
9.3.1	Inverse Problem . . . . .	189
9.3.2	Denoising Prior . . . . .	190
9.3.3	Optimization . . . . .	191
9.4	Applications . . . . .	193
9.5	Results . . . . .	195
9.6	Discussion . . . . .	205

<b>10 Proximal Image Optimization</b>	<b>207</b>
10.1 Introduction	208
10.2 Representing Image Optimization Problems	212
10.3 The ProxImaL Language	215
10.3.1 Proxable Functions	216
10.3.2 Linear Operators	217
10.4 Compiling Problems to Efficient Solvers	220
10.4.1 Rewriting Problems	220
10.4.2 Problem Splitting	223
10.4.3 Problem Scaling	223
10.5 Proximal Algorithms	224
10.5.1 Stopping Criteria	230
10.5.2 Convergence Properties	232
10.6 Analysis of Linear Systems	234
10.7 Implementation	236
10.8 Results	237
10.9 Discussion	244
<b>11 Conclusion</b>	<b>247</b>
<b>Bibliography</b>	<b>254</b>
<b>A Supporting Materials</b>	<b>284</b>
A.1 Detailed Spectral Analysis of the CIS Image Formation	284
A.2 Harmonic CIS Image Formation	284
A.3 Transient Imaging with Gaussian-Exponential Priors	286
A.4 Indirect Imaging of Specular Object, and with Ambient Light	291
A.5 Imaging in Scattering Media of Various Densities	296
A.6 Validation of Convolutional Sparse Coding	299
A.7 Non-Sinusoidal Doppler Time-of-Flight Imaging	299
A.8 Velocity Map Denoising	300
A.9 Residual and Scale-Space Deconvolution	301
A.10 PSF Estimation using Calibration Targets	303
A.11 Cross-Channel Deconvolution for Various Optical Systems	306

A.12 Natural Image Prior Design Choices . . . . .	311
A.13 Bayesian Color Arrays Imaging and Deblocking . . . . .	315
A.14 Poisson Data Fidelity . . . . .	321

# List of Tables

Table 6.1	Complexity of Convolutional Sparse Coding . . . . .	128
Table 9.1	Quantitative Deconvolution Results . . . . .	200
Table 9.2	FlexISP Performance . . . . .	204
Table 10.1	Core Primitives in the ProxImaL Language . . . . .	215
Table 10.2	Proxable Functions Provided by ProxImaL . . . . .	217
Table 10.3	Linear Operators Provided by ProxImaL . . . . .	218
Table 10.4	Logic for the <code>is_diag</code> and <code>is_gram_diag</code> Subroutines . . . . .	235
Table 10.5	Runtime for Linear Operators . . . . .	236
Table 10.6	Lines of Code Comparisons . . . . .	237
Table 10.7	Demosaicking Results . . . . .	238
Table 10.8	Quantitative Evaluation of Poisson Deconvolution . . . . .	243
Table A.1	Weighting Internal and External Priors . . . . .	312

# List of Figures

Figure 1.1	Thesis Overview . . . . .	4
Figure 2.1	Propagation of Light at an Interface . . . . .	16
Figure 2.2	Global Light Transport in a Diffuse Scene . . . . .	18
Figure 2.3	Time-resolved Global Light Transport . . . . .	19
Figure 2.4	Optical Systems . . . . .	20
Figure 2.5	Seidel Aberrations . . . . .	25
Figure 2.6	Point Spread Function (PSF)s of a Biconvex Lens . . . . .	26
Figure 2.7	CCD and CMOS APS Sensor Working Principle . . . . .	31
Figure 2.8	Sensor Model for Linear Digital Cameras . . . . .	34
Figure 2.9	Working Principle of a Photonic Mixer Device (PMD) Pixel . . . . .	45
Figure 2.10	CIS Time-of-Flight (TOF) Range Imaging . . . . .	46
Figure 2.11	Spectral Analysis of CIS TOF Imaging . . . . .	50
Figure 2.12	Multi-Path Interference . . . . .	53
Figure 2.13	Bayesian Inference . . . . .	60
Figure 3.1	Transient Imaging . . . . .	67
Figure 3.2	Transient CIS Image Formation Model . . . . .	69
Figure 3.3	Gaussian-Exponential Priors . . . . .	73
Figure 3.4	Setup for Transient Imaging . . . . .	76
Figure 3.5	Transient Measurement Matrix . . . . .	78
Figure 3.6	Simulated Transient Imaging Results . . . . .	79
Figure 3.7	Reconstructed Transient Profiles . . . . .	80
Figure 3.8	Test Scenes . . . . .	81

Figure 3.9	Transient Reconstructions for Complex Scenes . . . . .	82
Figure 3.10	Spatio-Temporal Modulation . . . . .	84
Figure 3.11	Spatio-Temporal Transient Imaging . . . . .	85
Figure 4.1	Imaging in Scattering Media . . . . .	89
Figure 4.2	Continuum of Indirect Paths in Scattering . . . . .	90
Figure 4.3	Sparsity of Transient Images . . . . .	92
Figure 4.4	Exponentially Modified Gaussians . . . . .	93
Figure 4.5	Synthetic Evaluation . . . . .	96
Figure 4.6	Setup . . . . .	98
Figure 4.7	Results for Scattering Milk Particles . . . . .	99
Figure 4.8	Results for Scattering Plaster Particles . . . . .	100
Figure 4.9	Error Analysis for Imaging in Scattering Media . . . . .	100
Figure 5.1	Diffuse Mirrors . . . . .	104
Figure 5.2	Schematic Overview . . . . .	105
Figure 5.3	Setup . . . . .	114
Figure 5.4	Albedo Reconstructions . . . . .	115
Figure 5.5	Simultaneous Albedo and Geometry Reconstructions . . . . .	115
Figure 5.6	Quantitative Evaluation . . . . .	116
Figure 6.1	Convolutional Sparse Coding . . . . .	120
Figure 6.2	Empirical Convergence . . . . .	129
Figure 6.3	Visualization of Filter Learning . . . . .	131
Figure 6.4	Learned Filters . . . . .	132
Figure 6.5	Boundary Extrapolation . . . . .	133
Figure 6.6	Learning From Sparse Observations . . . . .	134
Figure 6.7	Inpainting Results for Normalized Data . . . . .	135
Figure 6.8	Inpainting Non-Normalized Data . . . . .	135
Figure 6.9	Reconstructions Using Known Convolutional Basis . . . . .	136
Figure 7.1	Doppler Velocity Imaging . . . . .	139
Figure 7.2	Doppler Shift in CIS Imaging . . . . .	142
Figure 7.3	Velocity Imaging Using Orthogonal Frequencies . . . . .	144

Figure 7.4	Simulated and Measured Sensor Response . . . . .	146
Figure 7.5	Depth-Dependent Offset . . . . .	149
Figure 7.6	Experimental Verification for Varying Depth and Velocities . .	150
Figure 7.7	Measurement Accuracy . . . . .	151
Figure 7.8	Poisson Denoising . . . . .	153
Figure 7.9	Results for Large Outdoor Scene . . . . .	153
Figure 7.10	Results for Periodic Motion . . . . .	154
Figure 7.11	Results for Fast Motion . . . . .	154
Figure 7.12	Results for Textured Objects . . . . .	155
Figure 7.13	Results for Fast Gestures . . . . .	156
Figure 7.14	Results for Small Objects . . . . .	157
Figure 7.15	Optical Flow Failure Cases . . . . .	157
Figure 7.16	Towards 3D Flow . . . . .	158
Figure 7.17	Multi-Camera TOF System . . . . .	159
Figure 7.18	Multi-Camera Prototype Setup . . . . .	161
Figure 7.19	Instantaneous Doppler Velocity Imaging . . . . .	161
Figure 7.20	Multi-Device Interference Cancellation . . . . .	162
Figure 8.1	Simple Lens Imaging . . . . .	167
Figure 8.2	PSFs of Simple Lenses . . . . .	168
Figure 8.3	Cross-Channel Reconstruction . . . . .	171
Figure 8.4	Cross-Channel Prior . . . . .	172
Figure 8.5	Chromatic Artifacts in Low-Light Areas . . . . .	176
Figure 8.6	Results for a Plano-Convex Lens . . . . .	178
Figure 8.7	Additional Scenes Captured with a Plano-Convex Lens . . . .	179
Figure 8.8	Limitations . . . . .	180
Figure 8.9	Results for Commercial Lens Systems . . . . .	181
Figure 8.10	Results for an Ultra-Thin Diffractive Optical Element . . . . .	183
Figure 9.1	Flexible Camera Image Processing . . . . .	186
Figure 9.2	Replacing Traditional Camera Processing Pipelines . . . . .	187
Figure 9.3	Empirical Convergence . . . . .	196
Figure 9.4	Quantitative Demosaicking Results . . . . .	196



Figure 9.5	Qualitative Demosaicking Results . . . . .	197
Figure 9.6	Joint Demosaicking and Denoising Results . . . . .	198
Figure 9.7	Demosaicking Results for Real-World Captures . . . . .	198
Figure 9.8	Qualitative Deconvolution Results . . . . .	199
Figure 9.9	Interlaced HDR Reconstructions . . . . .	201
Figure 9.10	Burst Imaging Comparison . . . . .	203
Figure 10.1	ProxImaL . . . . .	208
Figure 10.2	Example DAG Representation . . . . .	216
Figure 10.3	DAG for Stacked Linear Operators . . . . .	219
Figure 10.4	ProxImaL Compiler Pipeline . . . . .	220
Figure 10.5	Absorbing Linear Operator . . . . .	221
Figure 10.6	Runtime Comparisons . . . . .	231
Figure 10.7	Interlaced HDR Results . . . . .	240
Figure 10.8	Burst Denoising Results . . . . .	241
Figure 10.9	Poisson Deconvolution Results . . . . .	242
Figure 10.10	Phase Retrieval Results . . . . .	245
Figure A.1	Illustration of Coordinate Descent . . . . .	287
Figure A.2	Transient Reconstructions for Diffuse Wall Scene . . . . .	290
Figure A.3	Transient Profiles for Diffuse Wall Scene . . . . .	290
Figure A.4	Non-Line-of-Sight (NLOS) Imaging of a Diffuse Foamboard . . . . .	291
Figure A.5	NLOS Imaging of two Whiteboards . . . . .	292
Figure A.6	NLOS Imaging of Mirrors . . . . .	292
Figure A.7	NLOS Imaging of Brushed Metal . . . . .	293
Figure A.8	NLOS Geometry Reconstruction with Ambient Light . . . . .	294
Figure A.10	Frame Averaging . . . . .	294
Figure A.9	NLOS Albedo Reconstruction with Ambient Light . . . . .	295
Figure A.11	Sparsity Analysis for NLOS Imaging . . . . .	295
Figure A.12	Additional Scattering Experiments Using Milk . . . . .	297
Figure A.13	Additional Scattering Experiments Using Plaster . . . . .	298
Figure A.14	Reconstruction Quality for Learned Filters . . . . .	299
Figure A.15	Synthetic Results for Velocity Map Denoising . . . . .	301

Figure A.16	Residual Scale-Space Deconvolution . . . . .	302
Figure A.17	PSF Calibration Target . . . . .	305
Figure A.18	Additional Results for Commercial Lens Systems . . . . .	306
Figure A.19	Deconvolution for Multispectral Cameras . . . . .	307
Figure A.20	Evaluation of the Cross-Channel Deconvolution . . . . .	308
Figure A.21	Additional Evaluation of the Cross-Channel Deconvolution . .	309
Figure A.22	Cross-Channel Deconvolution on Simple Lens Captures . . .	310
Figure A.23	Effects of Internal and External Priors . . . . .	313
Figure A.24	Comparison of Natural-Image Priors . . . . .	314
Figure A.25	Choice of Prior Weights . . . . .	316
Figure A.26	Cross-Channel Optical Flow . . . . .	317
Figure A.27	Color-Array Camera Results . . . . .	318
Figure A.28	JPEG Deblocking Results . . . . .	321

# Glossary

**CIS** Correlation Image Sensors

**LTI** Linear Time-Invariant

**PSF** Point Spread Function

**TOF** Time-of-Flight

**MPI** Multi-Path Interference

**NLOS** Non-Line-of-Sight

**CMOS** Complementary Metal Oxide Semiconductor

**APS** Active Pixel Sensor

**CCD** Charge-Coupled Device

**CSC** Convolutional Sparse Coding

**SPAD** Single-Photon Avalanche Diodes

**AMCW** Amplitude Modulated Continuous Wave

**PMD** Photonic Mixer Device

**HDR** High Dynamic Range

**SNR** Signal-to-Noise Ratio

**PDF** Probability Distribution Function

**MAP** Maximum A Posteriori

**MMSE** Minimum Mean Square Error

**TV** Total Variation

**ADMM** Alternating Direction Method of Multipliers

**CG** Conjugate Gradient

**D-TOF** Doppler Time-of-Flight

**OFDM** Orthogonal Frequency-Division Multiplexing

**DSL** Domain-Specific Language

# Acknowledgments

I am grateful to have had the opportunity to work with my academic advisor Prof. Wolfgang Heidrich. His guidance and support throughout my graduate studies went far beyond of what a doctoral student could wish for. Without his relentless encouragement, both in the laboratory and on the whiteboard, this dissertation would not exist. Since my very first visit in Vancouver, one year before making the decision to pursue a doctoral degree at the University of British Columbia, he has instilled in me a passion for the measurement, display, and inference using light, which has only grown since. I cannot imagine a better mentor and I now graduate full of new ideas that I am eager to explore.

Furthermore, I would like to thank the members of my doctoral supervisory committee Prof. Robert J. Woodham and Prof. Michiel van de Panne as well as the external examiner Prof. Yoav Y. Schechner. I sincerely appreciate the fruitful discussions on many of the ideas described in the following and the time spent on the thorough evaluation of my dissertation.

I feel lucky to have had terrific colleagues in Prof. Heidrich's laboratories. James Gregson, Lei Xiao, Gerwin Damberg, Shuochen Su, Qiang Fu, and Yifan Peng have made my doctoral studies a unique experience that I already miss.

I have been fortunate to collaborate with outstanding researchers over the last few years. Without their contribution it would not have been possible to explore the ideas from this dissertation as extensive as presented below. In particular, Prof. Gordon Wetzstein, Prof. Matthias Hullin, Prof. Ramesh Raskar and Dr. Douglas Lanman all inspired me to revisit problems which have been discarded as too hard in previous research. Furthermore, I thank Prof. Andreas Kolb, Dr. Matthew O'Toole and Prof. Kiriakos Kutulakos for their collaboration on some of the work discussed

in this thesis. Prof. Kolb has been a mentor of mine since I finished high school, and his guidance and encouragement during my undergraduate and graduate degree at the University of Siegen have laid the foundation for my doctoral studies. I thank my industry collaborators Dr. Paul Green, Dr. Kari Pulli, and Dr. Jan Kautz for taking on the challenge of realizing some of the technical ideas from this dissertation in commercial products.

I would also like to thank my wonderful friends back home in Germany and in North America. I am grateful to know that they are there to support me, especially in times of large workloads like those required for parts of this dissertation.

*Meinen Eltern, für ihre Unterstützung und das Interesse an der Wissenschaft, ist diese Doktorarbeit gewidmet. Ihnen danke ich von Herzen dafür, mir meine gesamte Ausbildung ermöglicht zu haben, und in gleichem Maße für das große, ermutigende Interesse an meiner Arbeit. Hätte ich nicht ihre vollste Unterstützung während des Studiums und der Promotion gehabt, gerade in der langen Zeit im Ausland, wäre die Vollendung dieser Arbeit nicht möglich gewesen.*

# Chapter 1

## Introduction

Through billions of years of evolution, pairs of single-lens eyes have emerged as the dominating sensor of the mammalian visual system. It is remarkable that complex tasks such as tracking, classification and localization can be performed robustly even when other sensory inputs are not present. This is enabled by the vast computational resources that are allocated for the visual system [12]. About half of the cerebral cortex of primates has been associated with visual tasks [13]. In particular, research suggests that the visual cortex represents natural scenes by local convolutional features [14], [15]. It is this compressive representation that enables the mammalian visual cortex to robustly solve complex high-level tasks.

Similar to the visual system, computational imaging systems attempt to decode scene features that are non-trivially encoded in the measurements by adding computation to the measurement process. During the past decade, computational imaging has made a huge impact in scientific applications [16], consumer imaging [17] and classification [18], robotic and industrial applications [19], microscopy [20] and health [21]. The key idea enabling most of these applications is to exploit structure in the unknowns, that is the scene information to be recovered, and the measurement itself.

In particular, convolutional structure in the measurement has the benefit that the latent information is encoded locally. The local encoding, due to the limited support of the convolutional kernel, preserves global context and therefore facilitates decoding the hidden information from the measurement. Furthermore, it leads to a linear



model that can be reasoned about using linear algebra, is computationally efficient, and Fourier analysis has proven to be an effective analysis tool for convolutional models.

An example of such a convolutional model is the spatial convolution that models aberrations in imaging optics. Since the end of the 19th century, imaging optics have been designed to minimize all existing optical aberrations. In 1841 Gauss first described the linear model of optics [22], and only a few years later, in 1857, Seidel introduced aberration theory [23], which describes non-linear deviations from Gauss's simplified linear model, that is aberrations. Since then optical design aims at eliminating these aberrations by introducing increasingly complex optical systems. Only recently, researchers have recognized that optical aberrations can not only be deliberately minimized but can be designed to carry additional scene information. One example is coded aperture imaging, where the optical system is designed to have depth-varying aberrations [24]. Another example is the recent advances in fluorescence microscopy, where depth-varying aberrations are designed using a diffractive optical element [20].

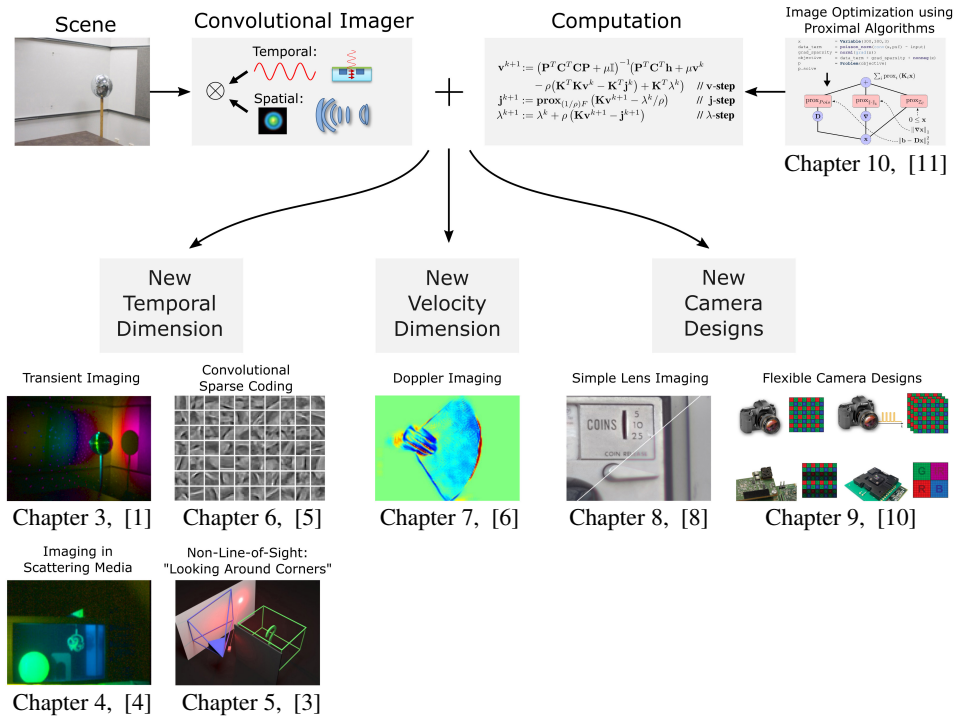
In addition to spatially convolutional measurements, spectrally convolutional or temporal convolutional measurements can also effectively encode scene information. An example of a temporally convolutional measurement can be found in nature, which has equipped a few species of animals with remarkable capabilities allowing them to survive in harsh environments. One of these evolutionary strategies is biological sonar. Animals using biological sonar create pulses of sound and listen to the reflections of the pulse. In other words, the Linear Time-Invariant (LTI) system defined by the scene is convolved with an impulse, and the temporal impulse response of the LTI system is measured. The delay between transmission and reception, that is the Time-of-Flight (TOF) of the pulses, is used to locate and classify objects. Notably, microchiropteran bats are therefore able to navigate in complete darkness, and toothed whales (including dolphins, porpoises, killer whales and sperm whales) rely on it in their underwater habitat, where scattering and absorption drastically limit the range of vision [25].

Inspired by biological sonar, optical TOF sensors measure the delay between light emitted into a scene and received by a co-located sensor. It was the invention of the laser in 1960 as well as accurate timing electronics that enabled a commercially

successful application of this principle in Lidar systems [26]. While delivering high accuracy and range, a major drawback is that distance is only measured to a single point. Hence scanning is required. Recently, Correlation Image Sensors (CIS) are revolutionizing the 3D depth imaging market by temporally convolving a scene with amplitude-modulated flood illumination, which no longer requires scanning. The sensors consist of an array of pixels that can shift charges generated by incoming photons into two buckets. By doing so, CIS convolve an incoming temporal signal with a reference signal. From the resulting convolution signals, the TOF can be estimated with high accuracy. As this range measurement technique is based on TOF, structured light or stereo matching techniques are outperformed by a significant margin due to the small baseline (that TOF-based systems allow) or due to lack of texture. CIS technology has been developed since the early 2000s [27]. Fast adoption of CIS-based range sensing has been accelerated by the fact that they can be fabricated in the Complementary Metal Oxide Semiconductor (CMOS) process. CMOS technology has been driven by the microprocessor industry and is therefore inexpensively available for mass-market production. Furthermore, additional logic for control, signal generation, or processing can be directly implemented on chip along with the sensor as in Microsoft’s Kinect sensor for Xbox One [28]. Recent CIS systems, however, do only very limited computation in post-processing [29].

The key argument of this thesis is to exploit structure, both in the unknown scene information and in the measurements, by adding computation to the measurement process. This is akin to the mammalian visual system, which performs this processing in the visual cortex. By adding the computation in the form of inverse problems derived from Bayesian inference, one can, in fact, overcome current limitations of imaging systems and “make the invisible visible.” We demonstrate this in this dissertation as visualized in Figure 1.1. Adding computation has allowed us to extract two entirely new image modalities, beyond traditional color and depth imaging. These new image dimensions enable a large variety of novel applications. For classical color imaging, this approach leads to new computational camera designs that significantly outperform traditional systems in many different scenarios.

As shown in Figure 1.1 on the left, the first additional dimension is an ultra-high-resolution time dimension. Exploiting temporal structure in CIS measurements, we can separate all light paths corresponding to different photon travel times along



**Figure 1.1:** Thesis overview and scope (key publications indicated next to the chapter labels): Adding computation to the measurement process can overcome limitations of imaging systems by exploiting structure in the measurements or unknowns. Adding computation to CIS allows extracting two new imaging dimensions. The first is an additional time dimension, which is introduced by transient imaging in Chapter 3. Two applications of transient imaging are demonstrated with imaging in scattering media Chapter 4, and non-line-of-sight imaging in Chapter 5. A model exploiting convolutional structure in the unknowns for both the temporal and spatial dimension is discussed in Chapter 6. The second dimension added to CIS imagers is velocity measured via the Doppler shift, which is presented in Chapter 7. Adding computation to color imagers enables new computational camera designs, that outperform the state-of-the-art. Imaging with radically simplified optics is demonstrated in Chapter 8. Novel computational sensor designs for challenging scenarios such as ultra-low-light imaging are shown in Chapter 9. An optimization framework unifying the computation for all of the methods above is discussed in Chapter 10.

this dimension. In other words, this means that such an image with additional time dimension is the impulse response of the light transport in the static scene. As light transport is a linear time invariant system that is fully described by its impulse response, this so-called transient image fully describes the light transport in the scene. We introduce a physically motivated model for transient images and reconstruction methods, relying on its convolutional structure in time. The ability to reconstruct transient images from a co-design of capture and computation has enabled new applications at the frontiers of vision. We demonstrate reconstruction of Non-Line-of-Sight (NLOS) objects from indirect reflections, thus turning diffuse walls into mirrors, and show imaging in strongly scattering media. In both cases, we exploit information that is treated as noise in regular range-measuring CIS systems. In this sense, our approach “makes the invisible visible.”

Shown in the center of Figure 1.1, the second additional dimension that we introduce is the velocity of moving objects. Recently, we were able to demonstrate that it is possible to capture the Doppler shift of sinusoidally modulated illumination using CIS with specifically designed modulation signals and low-light image reconstruction. Using the Doppler shift, one can compute the instantaneous axial velocity of an object. Measuring the modulated light reflected from the scene with multiple, differently correlating sensors enables 2D velocity capture along with usual the depth and amplitude capture. We imagine a variety of applications for this new imaging modality. In particular, traditionally challenging problems in vision such as segmentation, classification, and tracking could be drastically simplified by using this new image dimension.

Shown on the right of Figure 1.1, we demonstrate that adding computation to color imagers enables a variety of new computational camera designs that can drastically outperform classical color imaging systems. We demonstrate imaging with radically different optical designs that do not necessarily follow traditional design rules, such as minimizing optical aberrations. Instead, we exploit that aberrations can carry scene information embedded in the measurement. Recently, we have demonstrated that the specific structure of chromatic aberrations can be used for imaging with simple, cheap and very compact optics. In other words, rather than the classical approach of adding more elements to eliminate aberrations in increasingly complex lens designs, we simplify the optics up to a single element

and remove the aberrations in computation. Going a step further, this approach can even enable broadband imaging with ultra-thin diffractive optics. This opens up a space of very flexible, new optical designs, relying on thin refractive as well as diffractive elements.

Considering classical color image processing, also shown on the right of Figure 1.1, we demonstrate that adding computation in the form of Bayesian inference represents, in fact, a change of paradigm. Color image processing is traditionally organized as a pipeline model. Pipeline steps can accumulate error by never back-projecting on the measured observation. Replacing pipelines by formal optimization does not only yield a very flexible method without this drawback but also outperforms known methods in classical problems such as demosaicking and deconvolution. We demonstrate this on a variety of imaging systems using emerging array sensors, interleaved High Dynamic Range (HDR) and low-light burst imaging. Here, low-light burst imaging is another example where added computation can “make the invisible visible.”

Finally, this work covers the computational challenges for all of the methods shown in Figure 1.1. Expressing imaging tasks as optimization problems using Bayesian inference is a principled approach as the image-formation can often be well modeled. Hence, many state-of-the-art imaging methods follow this approach, like the ones in this dissertation. However, in practice, developing a given image formation model into a solver method can be time-consuming and error-prone, as there is a large span of choices in this process. Different combinations of statistical priors, various formulations of the same objective and different solver methods can lead to drastically varying reconstruction quality and computational efficiency. We introduce a generalized mathematical framework and a corresponding domain-specific language that allow quick exploration of this space. A novel mathematical compiler attempts to reformulate a problem automatically to find the most efficient problem translation. The key idea here is, again, to exploit structure in the problem. For the large variety of applications in this thesis, we demonstrate drastically increased accuracy, convergence, and computational efficiency compared to naive black-box optimization, in many cases multiple orders of magnitude in run-time.

This dissertation makes steps towards an “optimal camera” that does not attempt to capture the full plenoptic function by “brute-force” direct sampling using as many

sensors as possible, but rather as one that relies on structure in the scene and the image formation process to only recover the unknowns of interest using few, easily available measurement. The optics, sensor and reconstruction method of this camera are jointly optimized for a given application. This also means that there is not a single optimum. While solving such a higher-level camera optimization problem is extremely challenging due to its highly non-convex nature involving categorical and integer variables, parametrized architectures which allow the exploration of a subspace of the objective are feasible. This dissertation already shows samples of the search space that have been found by optimizing one variable (e.g. the algorithm parameters or optics) with all others fixed, essentially performing a first coordinate descent step. We also demonstrate first steps to extend this idea even further and consider the scene also as part of the camera. The proposed imaging system for NLOS objects can be interpreted as a method that essentially turns a diffuse surface into a large-scale image sensor. One could extend this system to be able to “unfold” light paths to a large depth while estimating the reflectance of each scene point on the fly. Starting from a direct bounce, reconstructing object points “around a corner” from a first bounce, and finally “around two corners” from a second bounce, and so forth. Ultimately every scene surface will act as a sensor. Such scene-adaptive imaging systems are yet another sample point of the rich high-level objective. Independent of this objective being minimized numerically or sampled by researchers, it becomes clear that the capabilities of known biological image systems driven by evolution can be outperformed by a significant margin. By capturing light propagating at the speed of light, this dissertation provides a first indicator for this margin.

## 1.1 Dissertation Structure

The remainder of this dissertation will be structured as follows.

*Chapter 2. Background and Related Work* This chapter explains the building blocks of the image formation models in this thesis and surveys previous related work. In particular, it describes the basic working principles of CIS, reviews imaging optics and explains recent color imaging sensors. Following, we review Bayesian

approaches for the inversion of the image formation from sensor measurements. Related traditional methods and work competing with any of the proposed approaches are discussed. Hence, this chapter lays the foundation for deriving and understanding the techniques in the chapters that follow.

*Chapter 3. Transient Imaging* This chapter demonstrates that transient images can be captured by adding computation to CIS measurements, covering our work [1] and [2]. We describe the capture sequence consisting of differently modulated images acquired with a CIS sensor, derives a model for local light/object interaction, and finally describes an optimization procedure to infer transient images given the measurements and model. The resulting method produces transient images at a cost several orders of magnitude below existing methods, speeds up the capture process by orders of magnitude, is robust to ambient illumination and allows for drastically increased scene sizes compared to previous methods. Furthermore, this chapter discusses how to combine transient imaging with spatial light modulators. Relying on the fact that light transport is separable in frequency, we show that all basic light transport techniques from conventional projector-to-camera systems can be applied per individual modulation frequency. The additional spatial light transport analysis helps to improve resolution in transient imaging.

*Chapter 4. Imaging in Scattering Media* We use the additional temporal dimension, which was introduced in the previous chapter, to enable imaging in scattering media. The core ideas from this chapter have been first presented in our work [4]. Imaging in volumetric scattering is an extremely hard task, since due to the scattering transport components are non-trivially mixed, leaving only the paths for ballistic photons undisturbed. Nevertheless, we demonstrate that by exploiting convolutional structure in transient images, we can separate mixed light paths. We introduce a new Convolutional Sparse Coding model for transient images, and we demonstrate that our system can analyze the light transport through scattering and turbid media.

*Chapter 5. Diffuse Mirrors* In this chapter, we use the additional temporal dimension of light to turn diffuse walls into mirrors, covering our publication [3].

Diffuse walls scatter back into all directions in contrast to mirrors that preserve directionality of the reflected light. However, in a diffuse reflection, the temporal dimension of light transport is left intact. This chapter describes an imaging system to recover NLOS geometry and albedo from second-order diffuse reflections, effectively turning walls into mirrors. An efficient optimization strategy is presented, linearizing the objective and exploiting sparsity of the unknowns.

*Chapter 6. Convolutional Sparse Coding* This chapter proposes an efficient optimization method for data fitting problems using Convolutional Sparse Coding models, described originally in our work [5]. The proposed technique is generalized to both reconstruction and learning of convolutional sparse codes. An optimization method is proposed that finds better minima than the state-of-the-art, has better empirical convergence and is therefore significantly faster. Due to its general formulation, the proposed approach can be applied to a large variety of problems (that do not have to be related to CIS). We demonstrate the application of feature learning for vision.

*Chapter 7. Doppler Velocity Imaging* This chapter shows that by adding computation to CIS measurements we can image instantaneous per-pixel velocity, covering our publications [6] and [7]. The proposed technique exploits the Doppler effect of objects in motion, which shifts the temporal frequency of the illumination before it reaches the camera. We describe how to design coded illumination and modulation frequencies that directly map velocities to measured pixel intensities. An effective reconstruction method handling the resulting Poisson noise and misalignment is introduced. Finally, we discuss a slight modification of our imaging system that allows for color, depth, and velocity information to be captured simultaneously.

*Chapter 8. Simple Lens Imaging* We demonstrate high-quality imaging with simple, ultra-compact optics, first proposed in our work [8] and [9]. Modern imaging optics are highly complex systems that aim at eliminating optical aberrations by introducing a large number of individual optical elements. We propose a radically different design approach using uncompensated, simple optics, which computation-



ally corrects for the severe aberrations introduced by the simple optics. In particular, a novel blind, self-calibrating deconvolution method is proposed that relies on the structure of the aberrations in different color channels of the latent image. We first demonstrate our approach using a simple refractive spherical lens and demonstrate first results indicating that it can even enable imaging using ultra-thin diffractive elements under broadband illumination.

*Chapter 9. Flexible Camera Image Processing* This chapter rethinks classical camera image processing as an optimization problem, covering our work [10]. Traditionally, camera image processing is organized as a pipeline of modules, each responsible for a particular part of the imaging task. This approach introduces a cumulative error as each step considers only the output of the previous step, not the original measurement. We replace the full pipeline with a single optimization problem using Bayesian inference. Observations are modeled using the image formation while priors are enforced for the latent image. We demonstrate that this approach is very flexible and applicable to a broad variety of computational camera designs ranging from traditional Bayer imaging to ultra-low-light burst imaging, in each case outperforming the state-of-the-art in terms of reconstruction quality.

*Chapter 10. Proximal Image Optimization* As demonstrated in all previous chapters, an increasingly large number of imaging tasks can be expressed as optimization problems using Bayesian inference, often outperforming the state-of-the-art. While designing the objective is often a very principled and elegant approach, there are many choices in the development of a solver, making this process time-consuming and error-prone. This chapter introduces a generalized mathematical representation for image optimization problems and a corresponding domain-specific language, published as [11], which allows for exploration of this vast array of choices. A mathematical compiler is proposed that identifies structure in the objective and based on this chooses the best way to translate a specific problem formulation into a solver implementation. We demonstrate that exploring the many choices in the solver development can lead to drastically increased accuracy and computational efficiency, often multiple orders of magnitude in reduced run-time.

*Chapter 11. Conclusion* This chapter describes avenues of future research that build toward the vision of an optimal imaging system, which not only matches naturally evolved visual systems found in nature but outperforms them for given tasks. The features of such an imaging system are outlined, major challenges on the way toward this vision are discussed, and the next immediate steps based on the work in this thesis explored.

## Chapter 2

# Background and Related Work

In this chapter, we describe the building blocks of the image formation in this thesis, and survey work related to the proposed techniques. Starting from first principles, we first describe established models of light in Section 2.1. Following, Section 2.2 discusses the propagation of light, and in particular introduces models for light transport in macroscopic scenes. Section 2.3 describes light transport in optical systems, while Section 2.4 reviews sensors that measure properties of the light transport. The concept of Correlation Image Sensors for range imaging is discussed in Section 2.5 and Section 2.6. This includes a discussion of CIS imaging in dynamic scenes and the Doppler effect. Having explained the properties of light transport and their measurement in various scenarios, Section 2.8 reviews inverse methods for the recovery of hidden scene features encoded in the measurements.

### 2.1 Light

Classical electrodynamics explains light as electromagnetic radiation [30]. In particular, it corresponds to the spectral band from 400 - 700 nm wavelength of the electromagnetic spectrum, which can be observed by the human eye [31]. This band lies between the ultra-violet (shorter wavelengths) and infrared (longer wavelengths). Described by Maxwell's equations [32], the fundamental laws of electromagnetic theory, various experiments verify that light behaves like an electromagnetic wave with synchronized oscillating magnetic and electric fields. These two fields are

oriented perpendicular to each other and the direction of propagation (and energy transfer). Hence, they form a transverse wave. Electric and magnetic fields do not require a medium to exist, and in vacuum an electromagnetic wave propagates at the speed of light  $c = 2.99792458 \cdot 10^8 m/s$ . Treating light as an electromagnetic wave allows modeling its propagation, including effects of diffraction, interference, scattering, transmission, reflection, and refraction. The approximation of geometric optics (i.e. ray optics) simplifies this propagation model. Geometric optics accurately models macroscopic propagation if the wavelength of light is small in comparison to the scene.

While treating light as an electromagnetic wave describes many of its properties, it is important to realize that light is in fact *not continuous* as a classical wave. This becomes evident in the photoelectric effect, describing that materials emit electrons for incident illumination only at a certain frequency of the light but not depending on its intensity. Einstein first explained that this effect is caused by light being composed of quanta, i.e. photons [33]. This quantum theory of light seems to be in conflict with the classical wave description. Modern quantum mechanics solves this conflict by accurately describing light as a dual wave-particle, which exhibits wave-like properties for propagating through space, and behaves like a particle during emission and absorption [34]. This means that propagation effects can be explained either by the wave or the particle nature of light. However, for a large number of photons, classical fields are produced *on average*. Hence, most macroscopic situations can be explained treating light as an electromagnetic wave [30].

In this dissertation, we describe imaging systems relying on both the wave and the particle behavior of light. All of these systems are *optical*, i.e. measure light. They can be used in a wide range of applications just like the human visual system mentioned in Chapter 1. This universality results from the fact that radiation with smaller wavelength (such as UV, X-ray or Gamma radiation) can be harmful to biological organisms due to the high photon energy. Radiation of lower wavelength, on the other hand, limits the resolution of a spatial sensor due to diffraction.

## 2.2 The Propagation of Light

### 2.2.1 Transmission, Reflection, and Refraction

The propagation of light in free space follows immediately from the models of light introduced in the previous section. Using the wave model, electromagnetic waves travel unaltered through the vacuum at the speed of light. The propagation direction is perpendicular to the electromagnetic wave front. This can be conveniently described using light rays, which are lines corresponding to the flow direction of radiant energy [30]. While the propagation in a vacuum can be described just by the properties of the wave itself, the propagation in homogeneous media and at interfaces between different media requires describing the interaction of light with the molecules in the media.

*Transmission* For molecules of a size much smaller than the wavelength of light, this interaction is described by Rayleigh Scattering [35]. This includes gases (e.g. present in air), transparent liquids (such as water) and clear solids (such as glass). In these media, every molecule acts as an oscillator that can be set to vibrate upon absorbing a photon. After the absorption, almost instantaneously, this oscillation causes the emission of another photon of the same frequency, that is elastic scattering occurs. Using the wave model of light, we can infer the forward propagation of light in homogeneous media. The randomly distributed molecules in the medium create electromagnetic waves in a scattering event. These waves interfere with each other, defining the new direction of propagation of light. For randomly spaced molecules, the waves will be in-phase in the forward direction and constructively interfere, while lateral phases will be randomly distributed and mostly destructively interfere. The denser the medium, the less lateral scattering occurs [30]. Hence, the propagation of light in a homogeneous dense medium (such as air and glass) is, in fact, the same forward propagation as in the vacuum (the lateral and back-scatter component can be neglected), and it can also be described by light rays.

While the propagation direction is the same as in free space, the apparent phase velocity of the transmitted light can change in a medium, although the photons all travel at the speed of light  $c$ . This is because the molecules in the

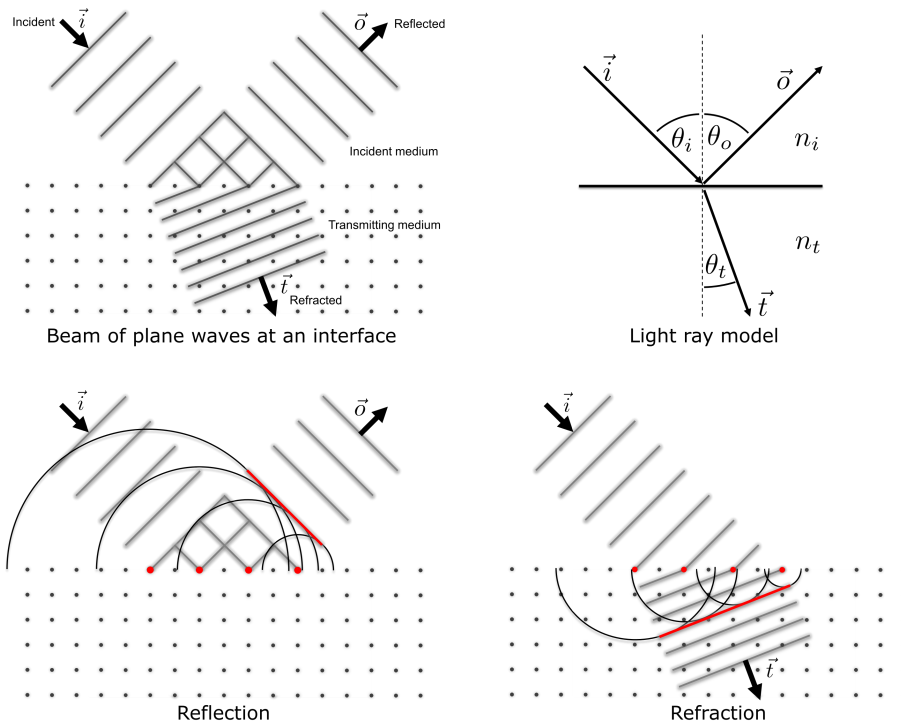
medium only oscillate perfectly in-phase at low frequencies. As the frequency of the electromagnetic radiation increases, a phase-lag in the scattered photons is introduced, hence reducing the phase velocity  $\nu$  of the apparent transmitted light [31]. This phase retardation by the molecules is macroscopically described by the medium's index of refraction

$$n = \frac{c}{\nu}. \quad \text{Index of Refraction (2.1)}$$

Per definition, it is  $n = 1$  in free-space and  $n > 1$  for other materials. See [30] for a table of  $n$  for common materials. Hence, intuitively, light appears to travel slower in most media, although its photons and the associated electric fields propagate at the speed of light.

*Reflection* Let us consider a discontinuity between two media with different refractive indices; see Figure 2.1. When a beam of light hits this interface, the photons in this beam scatter at the closely packed molecules across the interface. A portion of the light is back-scattered, and the remaining portion is transmitted. Hence, the beam of light is not sustained at the interface but is separated into a reflected and refracted beam [31]. The fact that two well-defined beams are created as the result of the scattering can be described using the wave model. A plane wave, incident on a flat surface at some angle, sweeps across the molecules on the surface, generating waves along this sweep. As the surface is locally flat and wave fronts are planar, these waves constructively interfere and form a well-defined reflected wave; see Figure 2.1. Note that, in principle, all molecules before and after the interface jointly create the reflected wave. However, the majority of the energy is reflected in a thin layer at the surface (about half of a wavelength in depth) [30]. Due to the planar geometry of the wave fronts and interface, the angle of the incident light beam  $\theta_i$  equals the angle of reflected light

$$\theta_o = \theta_i. \quad \text{Law of Reflection (2.2)}$$



**Figure 2.1:** Propagation of light at an interface. When a beam of plane waves hits an interface, two well-defined beams are formed. As illustrated at the top left, one beam is reflected off the interface, and another beam is refracted into the transmitting medium. The corresponding ray model is shown at the top right. The bottom describes the propagation behavior of light at this interface as a result of scattering. The incident plane wave sweeps across the surface of the transmitting media, generating waves along this sweep that constructively interfere into the reflected and refracted plane waves.

*Refraction* Having described the properties of the beam reflected at the interface, we now consider the light that gets scattered forward, passing the interface, as the transmitted beam. The molecules at the interface emit waves not only into the incident medium (combining into the reflected wave) but also in the transmitting medium; see Figure 2.1 again. These waves constructively interfere with each other as a well-defined transmitted wave. However, because the waves are propagating in another medium with a different refractive index, the apparent phase velocity of

the waves changes. This change in phase velocity causes the combined transmitted wave to propagate along a different direction. Hence, the beam of light bends [31], i.e. the beam gets refracted. Macroscopically this behavior is described by Snell's law, first described in [36]. For  $\theta_o, n_o, \theta_t, n_t$  being the angles and refractive indices of the incident and refracted beam, it is

$$\frac{\sin \theta_i}{\sin \theta_t} = \frac{\nu_i}{\nu_t} = \frac{n_t}{n_i}, \quad \text{Snell's law (2.3)}$$

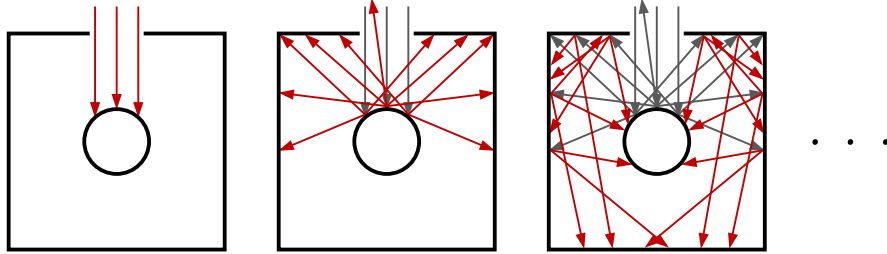
where the indices of refraction on the right are derived by division of  $c$ .

The laws of reflection and refraction, which are discussed above, can also be derived from different models for the propagation of light. A very different point of view is Fermat's Principle: The path taken by a beam of light between two points is the one that can be traversed in the least time [31]. The laws of reflection, refraction, and the concept of optical path length can be derived from Fermat's Principle [30]. Another elegant model is given by electromagnetic theory, which derives these laws from boundary conditions on the electric and magnetic fields [30]. Quantum electrodynamics, on the other hand, describes reflection and refraction with the particle behavior of light. Every possible light path is assigned a complex quantum-mechanical probability amplitude. The total probability amplitude is the sum of all possible paths and describes the macroscopic effect [34].

## 2.2.2 Global Light Transport

The propagation of light in homogeneous media and at interfaces can be macroscopically described with closed-form models, as explained in the previous paragraphs. For light transport in complex scenes, this is no longer possible. Instead, accurate simulation requires the evaluation of integrals over all possible light paths in the scene, for which no analytic approximation can be found [37]. A large body of work in computer graphics covers such global light transport simulations, i.e. rendering. A detailed introduction can be found in [38]. The geometric optics approximation, i.e. the light ray model, is used to simplify the derivations for complex scenes.





**Figure 2.2:** Global light transport in a diffuse scene. From left to right: Direct illumination only, first indirect bounce of light rays, second indirect bounce. The steady state limit of these individual ray propagations is the global light transport.

The classical *rendering equation* has been introduced in [39]. It is derived from the law of conservation of energy: The total energy of an isolated system is constant. For a ray with initial point  $x$  and direction  $\omega_o$ , this can be expressed as

$$L(x, \omega_o) = L_e(x, \omega_o) + L_r(x, \omega_o), \quad (2.4)$$

where here the Spectral Radiance  $L$  is used as an energetic quantity for a bundle of rays. Spectral radiance is the radiant power that is transmitted, refracted, absorbed or reflected by a surface per unit solid angle per unit projected area per unit wavelength [ $W \cdot sr^{-1} \cdot m^{-2} \cdot Hz^{-1}$ ]. See [38] for an introduction into radiometry and [40] for a detailed review. Eq. (2.4) states that the radiance  $L$  leaving a point  $x$  in direction  $\omega_o$  is the sum of the radiance emitted by the point  $L_e$ , and the radiance  $L_r$  reflected from other sources at the same point in the same direction. The reflected component  $L_r$  can be expressed as an integral of all components received over the hemisphere of incident angles  $\Theta$ , and Eq. (2.4) becomes the rendering equation

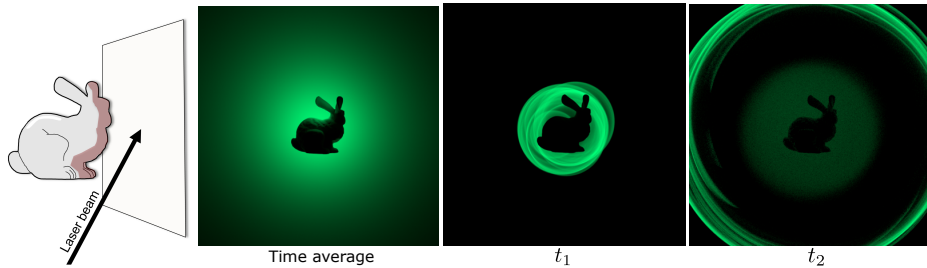
$$L(x, \omega_o) = L_e(x, \omega_o) + \int_{\Omega} L_i(x, \omega_i) \cdot f(X, \omega_i, \omega_o) \cdot \cos(\omega_i) \, d\omega_i \quad \begin{array}{l} \text{Rendering} \\ \text{Equation} \\ (2.5) \end{array}$$

The integral accumulates the incident radiance  $L_i$  from any other point in the scene in direction  $\omega_i$  toward  $x$ , weighted by the bidirectional reflectance distribution function (BRDF)  $f$  and a cosine attenuation factor due to the incident angle  $\omega$

between the surface normal at  $x$  and inward ray. The BRDF  $f$  is an intrinsic property of the surface material, which allows determining the outgoing radiance for any incident and outgoing beam geometry (by evaluating the integral from the rendering equation). The function is, in fact, not unit-less but is defined with unit  $[sr^{-1}]$  to make the rendering Eq. (2.5) consistent regarding units.

$L_i$  can either originate from a light source or from reflections off other surfaces; see Figure 2.2. The standard way of solving the rendering equation in computer graphics is by starting with a solution that contains only the emissive components and then iterating the scheme to convergence [37]. Another approach is to expand the integral and sample the space of all paths [41]. The work of [37] gives an overview of efficient physically accurate simulation methods. Model approximations and computational methods that allow for real-time rendering are described in [42].

### 2.2.3 Transient Global Light Transport



**Figure 2.3:** Time-resolved Global Light Transport. In the scene on the left, the backside of the bunny geometry is illuminated with a short temporal flash and the indirect reflections of the diffuse wall are observed. Simulations of the light transport in this scene are shown on the right. Traditional global illumination rendering assumes the speed of light to be infinite, hence computes the time average of the light transport. Time-resolving this average reveals structure in the individual components.

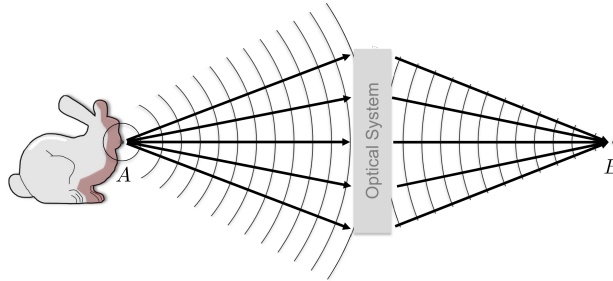
To this point, we have considered the speed of light to be infinite. The rendering equation (2.5) does not model the travel-time of photons along the integrated light ray components but assumes the transport of radiance to be instantaneous; see Figure 2.3. This is a common assumption in computer graphics, which aims at simulating steady-state global light transport [37]. This dissertation covers the

capture and inversion of the temporally resolved propagation of light. A model for temporally resolved light transport can be derived from the rendering equation (2.5): Assuming constant scene geometry and reflectance during the transport, it becomes

$$L(x, \omega_o, t) = L_e(x, \omega_o, t) + \int_0^t \int_{\Omega} L_i(y_{\omega_i, x}, -\omega_i, t_0) \cdot \delta(t_0 + \Delta(x, y_{\omega_i, x}) - t) \cdot f(x, \omega_i, \omega_o) \cdot \cos(\omega_i) \, d\omega_i \, dt_0, \quad (2.6)$$

where we use  $y_{\omega, x}$  as notational sugar for the surface point found by intersecting a ray from  $x$  along direction  $-\omega$  with the scene.  $\Delta(x, y) = \|\vec{x}\vec{y}\|/c$  is the time delay for the light transport from  $x$  to  $y$ . The delta-function makes sure that light leaving  $y$  at time  $t_0$  is registered in  $x$  only at its time of arrival  $t_0 + \Delta(x, y)$ .

## 2.3 Optics



**Figure 2.4:** Optical Systems. An illuminated object surface consists of tightly packed scattering molecules. An individual scatterer  $A$  on the surface emits spherical waves. A segment of the wave front enters an optical system that reshapes incoming wave fronts to converge at point  $B$ . Hence,  $B$  is an image of the source  $A$ .

Optical systems have been used and developed since remote antiquity, with simple lenses for vision correction, dating back to at least 1200 B.C.E., and burning-glasses to create fire, mentioned around 400 B.C.E. [30]. Such systems are designed to redirect the radiant flow in a scene. Using the wave model from the previous paragraphs, an illuminated object surface can be described as tightly packed scat-

tering molecules that emit spherical waves. Considering now a single scatterer  $A$ , as shown in Figure 2.4, its spherical wave fronts spread the emitted radiant energy with increasing distance, i.e. the corresponding rays diverge.

An optical system can be designed to reshape the wave fronts, such that they converge to a single point  $B$ . Hence, this optical system produces an image of  $A$  in the point  $B$ .

### 2.3.1 Diffraction

It is important to note that practical optical systems are limited in size. Therefore, only a portion of the wave fronts can be transformed by the system, corresponding to a cone of rays as visualized in Figure 2.4. It follows immediately from this limitation that a perfect inversion of the incoming waveforms will not be possible, but the waves will be *diffracted* at the entrance of the optical system. To see this, consider an optical system consisting of just an aperture in the same homogeneous medium in which the incident waves travel. The straight-forward propagation in homogeneous media as explained in Section 2.2.1 is not continued at the aperture as the waves blocked out by the aperture no longer contribute to the overall forward component. Hence, the direction of energy of an incoming plane wave will be spread out. Thomas Young in 1804 [43] was the first who described this diffraction effect with the wave behavior of light. His famous double slit experiment observed the redirected energy of light incident on two closely spaced slits as a diffraction (interference) pattern, which can be accurately described using the wave model of light. Therefore, it also follows that diffraction effects decrease as the wavelength of the incident light  $\lambda$  becomes small in comparison to the dimensions of the optical system. However, as the physical dimensions are limited, realizable optical systems are always *diffraction-limited*. Specifically, for a circular aperture, a perfect lens does not focus a point light to a single point but to a spread-out, symmetric, Airy pattern that can be analytically expressed [30]. Hence, the theoretical limit of a perfect imaging system can be defined based on whether two nearby imaged point sources can be resolved as distinct patterns. In 1873, Abbe defined the resolution limit of a perfect imaging system as the “radius” of the Airy disk pattern, which is

infinitely large (but quickly decaying). It is

$$r = \frac{0.5\lambda}{\text{NA}} = \frac{0.5\lambda}{n \cdot \sin(\theta)}, \quad \text{Abbe Resolution Limit (2.7)}$$

where  $\text{NA} = n \cdot \sin(\theta)$  is the numerical aperture that is defined in terms of the incident angle  $\theta$  and the refractive index of the medium surrounding the lens. Due to the fuzzy definition of when two blurred sources can be separated, slightly different criteria exist, which lead to similar analysis results in practice. See [30] for a detailed discussion on diffraction effects in optical systems.

While diffraction limits the achievable resolution of an imaging system, optics can also be deliberately designed to exploit interference, as we will show in Chapter 8. This enables ultra-compact optical systems but introduces additional imaging errors. Nevertheless, many practical optical systems can be well approximated with the light ray model, i.e. geometric optics, which ignores diffraction and thus conceptually assumes  $\lambda \rightarrow 0$ . An overview of such systems will be given in the following paragraphs.

### 2.3.2 Geometric Optics

Consider a point source  $A$  on an illuminated object surface as illustrated in Figure 2.4. A typical imaging scenario is now that this source is distanced far from the optical system (and the sensor located beneath the optics). In this case, the spherical waves reaching the system are approximately planar. Optical design aims at reshaping this beam of plane waves to make the wave fronts converge to a single focus point  $B$ . To achieve constructive interference of all incident waves in  $B$ , the number of phase distance along each path  $P$  must be equal in both media. That is  $P_i/\nu_i + P_t/\nu_t = \text{const.}$ , with  $P_i, P_t$  being the portion of the path in the incident and optic medium, respectively. Relying on Eq. (2.1) from Section 2.2.1, we have  $n_i P_i + n_t P_t = \text{const.}$ . Hence, the optimal lens surface for this (monochromatic) imaging scenario is the hyperbola.

Although hyperbolic lenses perform well in many scenarios, in fact, spherical lenses are most common in optical systems for imaging and display applications [30]. This is because spherical surfaces are relatively easy to produce by grinding two roughly spherical surfaces, one convex and one concave, against each other. High-

quality surfaces with a tolerance of less than  $\lambda/10$  can be manufactured this way [44]. For a single spherical lens, closed form expressions for the focus point and change in direction of individual rays can be easily derived from either Fermat's principle or Snell's law. Complex optical systems consisting of multiple elements, however, have to be analyzed numerically by tracing rays through all elements, using Snell's law at each interface. The paraxial approximation allows a drastically simplified analytic description for rays close to the optical axis. This approximation essentially relies on the Taylor-series expansion

$$\begin{aligned}\sin(\theta) &= \theta - \frac{\theta^3}{3!} + \frac{\theta^5}{5!} - \frac{\theta^7}{7!} + \dots, \\ \cos(\theta) &= 1 - \frac{\theta^2}{2!} + \frac{\theta^4}{4!} - \frac{\theta^6}{6!} + \dots.\end{aligned}\tag{2.8}$$

For a cone of rays impinging on the lens at small angles  $\theta \rightarrow 0$ , the first term in the Taylor-series is a good approximation, that is  $\sin(\theta) \approx \theta$  and  $\cos(\theta) \approx 1$ . For rays in this paraxial region, Snell's law simplifies to

$$\theta_i n_i \approx \theta_t n_t, \quad \text{Paraxial Approximation (2.9)}$$

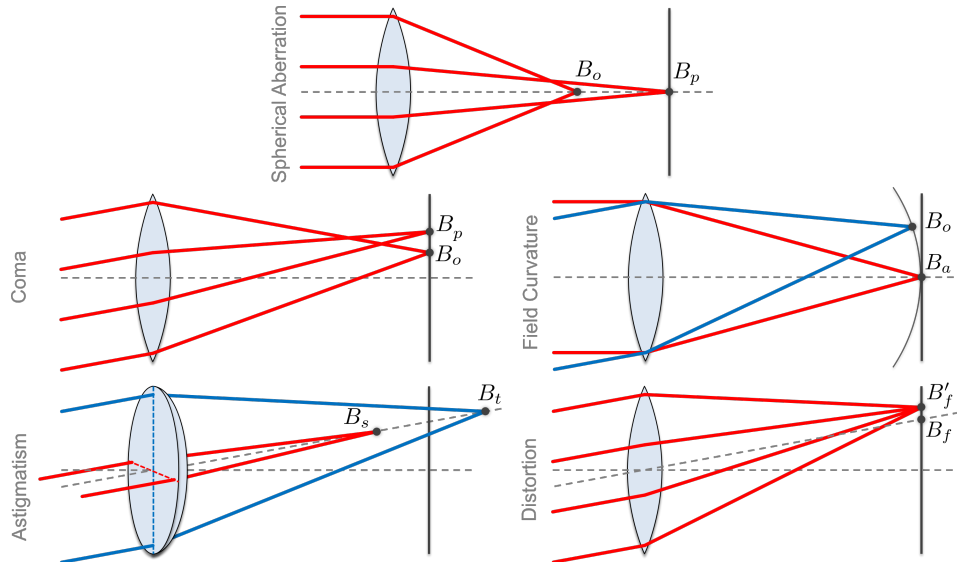
The paraxial approximation shows that closely spaced spherical interfaces, forming a thin lens, can focus parallel rays in a unique image point. In 1841, Gauss was the first to describe this linear model of optics for nearly axial rays [22], i.e. first-order or Gaussian optics. Although accurate ray-tracing simulations without the paraxial simplification are easy to obtain, the linear model using thin lenses builds intuition for many imaging tasks. Simple analytic equations describe the relation between focus-point for parallel rays, object and image point distance (Gaussian lens formula), location of focus points and planes, and combinations of different thin lenses. See [30] for an expanded discussion. The paraxial model immediately describes the macroscopic effect of focusing. With a thin-lens optical system, moving the sensor relative to the lens allows focusing on different distances, for objects at infinity with parallel incoming rays to points close to the lens. Hence, such optical systems essentially perform a perspective transform of the 3D object space to a 3D image space. The sensor records a 2D slice of the image space, with some objects in focus and others out of focus.

### 2.3.3 Aberrations

Deviations from the idealized linear model of Gaussian optics are called aberrations. They accurately describe the errors introduced by using the paraxial and thin-lens approximations. This includes the imaging artifacts of using spherical lenses instead of hyperbolic lenses, which were introduced above. Aberrations are divided into two types: chromatic and monochromatic aberrations.

Chromatic aberrations result from the wavelength-dependency of the refractive index  $n$ . Recalling Eq. (2.1), which defined  $n = c/\nu$ , the wavelength-dependency of  $n$  becomes obvious because the phase velocity is given in wavelength per period, that is  $\nu = \lambda/T$ . As a consequence, the refraction at an interface is wavelength-dependent as well, and rays of different wavelengths take different paths through an optical system. Consider imaging a point light source, the shift or spread of the imaged point along the optical axis is called axial chromatic aberration, and the remaining component in the image plane is lateral chromatic aberration. The first type is observable as chromatic halos in an image while the latter causes colorful blur and haze in a capture. Chromatic aberrations will be discussed in more detail in Chapter 8, which presents a method for aberration removal post-capture.

Monochromatic aberrations describe the deviations from linear Gaussian optics that occur even for a single wavelength (or narrow spectral band). This includes the five primary aberrations, which are spherical aberration, coma, astigmatism, field curvature, and distortion. The primary aberrations can be described with sufficient accuracy by adding the third-order term from the Taylor-series (2.8) to the paraxial approximation, that is  $\sin(\theta) \approx \theta - \theta^3/3!$ . In 1857, Seidel introduced aberration theory [23], which describes the primary aberrations with this model, i.e. Seidel aberrations. Note that these aberrations are non-linear in the ray directions, while the measured intensity transported by the rays passing through an optical system is itself linear, as we will describe further below. As the Seidel aberrations cover a vast majority of imaging scenarios, aberrations that require higher-order terms from Eq. (2.8) are usually considered jointly as higher-order aberrations. Figure 2.5 illustrates the imaging errors introduced by the Seidel aberrations. Spherical aberration describes an axial shift in focus for peripheral rays. While central rays are focused on the image plane (at  $B_p$  in Figure 2.5), peripheral rays focus in front of



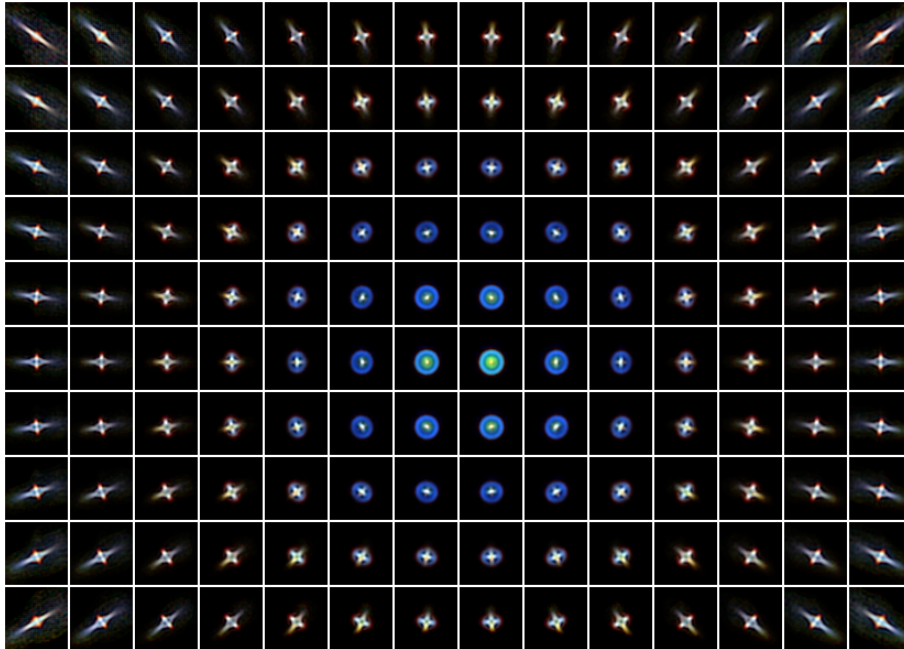
**Figure 2.5:** Seidel aberrations: These five primary aberrations explain the most severe deviations from the idealized linear Gaussian optics for monochromatic light.

or behind the image plane (at  $B_o$ ), depending on the lens being convex or concave. This causes a circularly blurred image superimposed with the focused paraxial component. Coma causes a lateral spread of the focused image point for off-axis object points, i.e. rays that impinge at an angle. This makes point sources appear to have a tail (coma) similar to a comet. Astigmatism is illustrated at the bottom left of Fig. 2.5. Peripheral rays traveling in two orthogonal planes, the tangential (including object point and optical axis) and sagittal plane, focus in different locations in front of or behind the image plane,  $B_t$  and  $B_s$  in Figure 2.5. Field curvature describes the aberrations that focus planar objects, orthogonal to the optical axis, not on a flat but a curved image plane. If this aberration is present, as for spherical lenses, only rays in the paraxial region are properly focused. Finally, the bottom right of Figure 2.5 visualizes distortion. This aberration describes for off-axis points a lateral shift in the image point  $B'_f$  from the ideal projected image  $B_f$ . The image is then distorted with pincushion or barrel distortion. See [30] for a detailed description of Seidel and chromatic aberrations.



All of the explained Seidel aberrations and chromatic aberrations, except the lateral distortion and chromatic shift, can be reduced by closing down the aperture. Distortion and lateral chromatic aberration can be mostly corrected post-capture in software. However, this introduces diffraction blur and reduces the Signal-to-Noise Ratio (SNR) of the captured image (as will be discussed in Section 2.4).

*Modeling Aberrations* In practice, all of the discussed aberrations (including diffraction blur) do not occur in isolation but jointly produce the overall imaging error. The superimposed aberrations can be elegantly described with a Point Spread Function (PSF). Figure 2.6 shows the PSFs for a simple biconvex spherical lens.



**Figure 2.6:** Patch-wise calibrated PSFs for a single biconvex lens at  $f/2.0$ . All Seidel aberrations are present, which jointly cause a spatially varying PSF. Spherical aberration and field curvature cause the rings, which vary in size. Astigmatism leads to the cross-shaped components, and coma causes an elongated tail of these crosses away from the center. Distortion leads to a shift of the PSFs' center of mass in the peripheral regions, which has been removed for a more compact display.

A PSF describes the response of an imaging system to a point source. Hence, in general, this can be a 4D function of the 3D spatial position and wavelength of this source. For a point source  $A$  with spectral distribution  $\xi_A$ , resulting from the illumination and material properties in the scene, the resulting PSF for a sensor with response  $\psi$  is

$$B_A = \int \psi(\lambda) \xi_A(\lambda) B_A(\lambda) d\lambda, \quad (2.10)$$

where  $B_A$  is here the spatially varying PSF for the point  $A$ . Common optical systems are optimized to vary little in the spectral range of the traditional RGB color filters. Hence, often PSFs are assumed to be invariant for these three channels. Common lens surfaces vary smoothly, and therefore also the resulting PSF changes smoothly over the image plane and smoothly with object distance [30]. Therefore, a widely used assumption is that PSFs are spatially invariant in a local neighborhood. Assuming invariance for a considered depth-range, the image formation including all aberrations can then be expressed as a local 2D-convolution in image tile. Considering an  $n \times m$  tile, let  $J, I \in \mathbb{R}^{n \times m}$  be the observed blurred image, and the latent sharp image (that would be captured without aberrations). For a given color channel the PSF is then a 2D convolutional kernel, and the formation of the blurred observation can be formulated as

$$J = B \otimes I, \quad (2.11)$$

$$\mathbf{j} = \mathbf{B}\mathbf{i}, \quad (2.12)$$

where  $\otimes$  is the convolution operator. Sensor noise is ignored here for simplicity. Note that this model for the measured image intensity is linear, while the ray directions in the aberrations discussed above itself are non-linear in general. In the second row,  $\mathbf{B} \in \mathbb{R}^{nm \times nm}$ , and  $\mathbf{j}, \mathbf{i} \in \mathbb{R}^{nm}$  are the corresponding quantities in matrix-vector form. The individual pixels of the 2D images are here stacked in column vectors, and the convolutional operation becomes a convolution (Toeplitz) matrix. In the remainder of this thesis, we will use either of these identical formulations depending on the context. Finally, the full image will be composed of many tiles, each with the PSF assumed constant over the tile. The transition between the tiles can be modeled with a window function  $W \in \mathbb{R}^{n \times m}$  that models the influence

of each tile on a blurred output pixel. The full image  $\tilde{J}$  is then

$$\tilde{J} = \sum_{a=1}^N W \cdot B_a \otimes I_a, \quad (2.13)$$

where a linear index  $a \in \{1, \dots, N\}$  is used for the 2D grid of tiles. A matrix-form of this equation can be formulated without much effort by adding a few linear operators for the weighting and patch extraction. It is also straightforward to express this tiled image formation in the frequency domain, which diagonalizes the convolution operation. See [45] for a detailed description and analysis of the tiled image formation model for spatially varying PSFs.

## 2.4 Imaging Sensors

Image sensors measure the intensity of light that is emitted from an active light source or sources in the scene and propagates through the potentially complex scene until reaching the imaging optics, which focus light from different scene points on different sensor locations. In particular, the optical system gathers a cone of rays that diverge toward the aperture plane and redirects them to converge onto the 2D sensor plane, as described in the previous paragraphs. Hence, the directionality (in the cone of rays) is lost in the measurement process. Instead of the directional radiance, image sensors estimate *irradiance* [ $W \cdot m^{-2}$ ].

### 2.4.1 Measuring Irradiance

Solid-state image sensors rely on the photoelectric effect to perform this measurement. As mentioned in Section 2.1, this effect refers to the generation of electron-hole pairs in a material when illuminated with light. In particular, the energy of the generated electrons is proportional to the frequency of the illuminating light but not the intensity. For low frequencies of the incident illumination, below a threshold, no electrons are generated independently of the intensity. The photoelectric effect had first been observed by Hertz in 1887 [46] but could not be described with the classical wave model of light. It was Einstein in 1905 [33] who first explained this effect with light being composed of quanta, i.e. photons [33].

The energy of a photon is

$$E = hf = \frac{hc}{\lambda}, \quad (2.14)$$

and hence proportional to the frequency is  $f = c/\lambda$ . Here, the factor  $h = 6.6260755 \cdot 10^{-34} Js$  is Planck's constant.

*Charge Generation* When a stream of photons impinges on a material, the energy  $E$  of every photon can be absorbed by electrons in the clouds around the atoms in the material, removing the electrons from their atoms as free charges that macroscopically lead to the photo-electric effect. Whether this photo-conversion happens depends on the material properties, e.g. insulators require extremely high energy to generate free-moving charges. In solids, electrons can exist in two distinct energy bands, a valence band (low energy) and a conduction band for free-moving charges (high energy). These bands are separated by an energy gap, i.e. *band gap*. For insulators, this is a large band gap (electrons do not reach the conduction band at room temperature), while the bands overlap in conductors. For silicon, the band gap is 1.11 eV. Hence, from Eq. (2.14) it follows that photons with wavelengths larger than  $1125nm$  will not interact with silicon at all.

However, even photons with higher energy do not necessarily create an electron-hole pair in the semiconductor (hole refers here to the electron-hole in the atom that contained the freed electron). Photon absorption is a statistical process depending on the energy of the photon and the material properties, which can be modeled with an exponential [47]. The penetration depth of a photon is defined as the depth where  $1/e$  (37%) of the incident radiation are absorbed. Hence, this depth exponentially depends on the wavelength of the photons. For example, red light penetrates silicon much less deeply than blue light does. Photons of longer wavelengths may not get detected at all, or may generate charges at deep penetration depths that accumulate in nearby pixels, i.e. cross-talk.

The overall charge generation process of a sensor is summarized with its quantum efficiency  $\eta$ , which is defined as the ratio of the detected electron-hole pairs  $e$  to the number of incoming photons  $i$ ; see Eq. (2.15).

$$\eta(\lambda) = \frac{e(\lambda)}{i}. \quad \text{Quantum Efficiency (2.15)}$$

Here,  $i$  and  $e$  are measured for the total area of a sensor pixel, which also includes light-insensitive areas occupied by on-sensor electronics. Hence, this definition of the quantum efficiency also models fill-factor effects, including micro-lens optics that aim at increasing it. Note that  $\eta$  depends on the wavelength of the incident radiation, including depth-dependent photon absorption effects [47].

*Charge Accumulation* Without any electric field applied to the semiconductor, the electrons generated from photon absorptions move around by thermal diffusion and self-induced drift. Thermal diffusion describes the movement in random directions due to thermal energy for temperatures larger than 0°K, and self-induced drift occurs between charges of the same polarity due to repulsive Coulomb forces [47]. Due to this movement, eventually, the electron-hole pairs recombine.

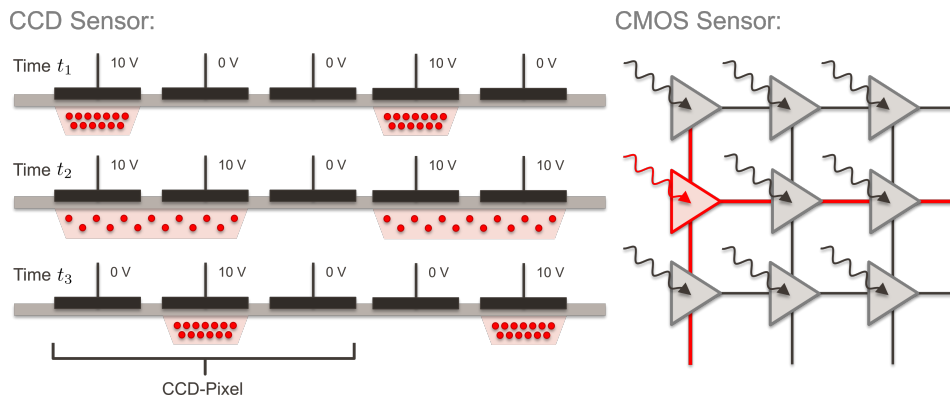
To detect them, electric fields are generated in the individual photodetectors to separate the charges. In most silicon detectors this field is created by doping narrow zones in the silicon. Doping refers to the process of introducing impurities into the silicon to alter its electrical properties. Two types of dopants exist: electron donor impurities (n-type) that contribute free electrons and electron acceptor impurities (p-type) that create holes. A very basic detector is the p–n diode. It consists of a junction of p-type and n-type doped semiconductor materials. At this junction, some electrons from the n-region (that reach the conduction band) diffuse into the p-type region to combine with holes, creating positive/negative ions on both sides, respectively. This creates an electric field in this *depletion region* [47]. If a photon gets absorbed in this depletion region, the resulting charge is moved by this electric field toward the cathode. Hence, a current is produced. In conventional CMOS sensors, p–n diodes are often reverse-biased. In this operating mode, a potential is applied to the semiconductor, which pulls electrons and holes away from the junction. Hence, the depletion region is significantly increased [47]. For further reading on semiconductor physics, we refer the reader to [47].

The number of electrons that can be retained in the depletion zone is called the well capacity. When high irradiance exceeds this capacity, charges may travel to nearby pixels, which causes so-called Blooming artifacts. Greater well capacity prevents this and increases the dynamic range of a sensor. Dynamic range is defined as the ratio of the largest detectable signal to the smallest detectable signal. This

sensor metric quantifies how well both bright and dark features can be measured. It depends not only on the well capacity but also on the dark current. Electrons that are thermally generated in the depletion zone cannot be distinguished from photoconverted electrons [27]. This causes a measurement floor for low irradiance values, i.e. the dark current. Dark current can be reduced by cooling the sensor. The well capacity can be increased by enlarging the depletion zone, which can, in general, be achieved by increasing the pixel area, and changing doping and detector-dependent properties (e.g. the gate voltage in CCD pixels [47]).

## 2.4.2 CCD and CMOS Sensors

Having described the basic principles of charge generation and accumulation in semiconductor imagers, we now review the two main sensor types: CCD sensors and CMOS Active Pixel Sensor (APS) sensors. These differ drastically in the amplification and the transfer of the photoconverted charges away from the charge bins in the photodetectors. Figure 2.7 illustrates the two sensor designs.



**Figure 2.7:** CCD and CMOS APS sensor working principle. In a CCD sensor, charges accumulated in the individual collection sites are sequentially transferred to neighboring pixels by charge-coupling. Similar to a bucket brigade, charges are moved along lines off the sensor to a single output amplifier. In contrast, CMOS APS sensors connect a buffer amplifier to each pixel's charge collection site. The locally amplified charge can then be read out over relatively long wires using column and row addressing (red wire for the red pixel).

As a very basic image sensor, one could realize a grid of p–n diodes in a silicon substrate with metal connectors (wires) from each diode to an output stage. As the metal connectors would need to cover the whole sensor area, they are relatively long compared to the pixel size. Hence, a large read-out capacitance would be introduced by these wires, causing a low SNR and low read-out speed [27]. CCD and CMOS APS sensors solve this problem using different approaches.

Charge-Coupled Device (CCD) sensors were first proposed in 1969 [48], [49]. These sensors consist of a grid of diodes as photodetectors. The diodes are biased in the deep depletion mode, which means that the depletion zone under the gate is completely depleted of free electrons. The depletion region increases with the gate voltage, similar to the p–n diode described above. For further reading on the CCD photodiodes; see [49]. During integration, photoconverted charges are accumulated in the depletion region under the photogate. See Figure 2.7 for an illustration on the top left. To read out the collected charges after the integration, the neighboring gate voltage is increased to match the bias potential. This causes the potential wells under both gates to merge. Hence, due to thermal diffusion and self-induced drift, the charges will distribute equally under both gates [27], i.e. charge-coupling will occur. This is illustrated on the center left in Figure 2.7. Afterward, the potential of the photogate is removed, causing the charges to migrate completely under the adjacent gate; see the bottom left in Figure 2.7. Repeating this process, the photogenerated charges can be moved along lines of gates to the output.

This charge transfer is very efficient: Typical CCD sensors drop only between 0.01% and 0.001% of the charges between two pixels (which mostly consist of three to four gates) [27]. Hence, for a scanline length of 1,000 pixels, a drop of only 1% can be achieved. After leaving the sensor area, the charges are converted to an output voltage sequentially in a *single*, low-capacitance output amplifier [27]. This leads to a low-noise and homogeneous output across all sensor pixels. While this is a strong benefit of CCD sensors, it also leads to low read-out speed due to the sequential output. Furthermore, high gate voltages between 10V and 20V are required for large depletion regions. Therefore, CCD sensors suffer from larger power consumption. Finally, the CCD fabrication process prohibits processing logic, such as read-out or signal processing logic, to be integrated on the same chip as the sensor [49].

CMOS APS sensors have almost completely replaced CCD sensors in the consumer market because they resolve many of the issues discussed above. In particular, CMOS technology enables the integration of processing logic along with the sensor on a single chip; it has low power consumption and drastically reduced manufacturing cost [47]. CMOS APS sensors consist of a grid of reversely biased photodiodes, each connected to a buffer amplifier. Hence, every pixel contains an active stage [27]. The locally amplified charges can then be read out over relatively long wires with large capacity. The individual pixels are addressed in a matrix with row and column decoders, as visualized on the right in Figure 2.7. Hence, this allows reading out image sub-regions at high speed, which is in contrast to CCD sensors that require full-sensor read-out. In addition, different integration start and end times can be used in each pixel.

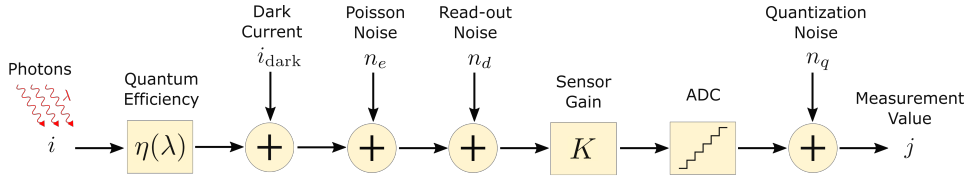
However, compared to CCD sensors, the increased flexibility in operation and manufacturing come at the cost of a reduced SNR. Due to manufacturing variation in the photo-detector and buffer amplifier (e.g. gate size and doping concentration), every pixel has a different offset and gain, resulting in fixed pattern noise. Besides this fundamentally different noise component, traditionally sensors in CMOS technology suffer from lower quantum efficiency, well capacity and higher read-out noise. However, the image quality of CMOS sensors is reaching CCD quality even in high-accuracy scientific applications [50]. Finally, it is even possible to realize CCD structures in CMOS [27]. This allows using the highly efficient charge transfer mechanism also on CMOS sensors, such as CIS TOF sensors, which will be reviewed in detail in Section 2.5.

### **2.4.3 Sensor and Noise Model**

To understand sensor measurements and perform inference tasks using measured data, it is essential to accurately model the underlying measurement process. However, as discussed in the previous paragraphs, a variety of different sensor types, operating modes and technical implementations exist. Hence, to compare different sensors for different applications, not only an accurate but also a generalizing model is necessary. The EMVA Standard 1288 [51] presents such a unified model along with calibration procedures for the model parameters. It covers all digital color (and



monochrome) cameras with linear photo response, and can even serve as a basis for emerging sensor types, such as CIS TOF cameras [52]. Since its introduction in 2005, the EMVA Standard 1288 has been adopted by a large number of sensor manufacturers [51]. In the following, we introduce a sensor and noise model that closely follows this standard.



**Figure 2.8:** Sensor model for digital cameras with linear photo response. During integration, a fraction  $\eta(\lambda)$  of the photons with wavelength  $\lambda$  generate electric charges. These charges and a signal floor of thermally generated charges (dark noise) are accumulated in the pixel. Subsequently, the collected charges are amplified with gain  $K$  and quantified, resulting in the measured digital value.

*Sensor Model* Figure 2.8 illustrates the sensor model. For simplicity, we consider spectral irradiance  $E$  of a single wavelength  $\lambda$  to be incident on the sensor. Only the quantum efficiency  $\eta$  from Eq.(2.15) is assumed to be wavelength-dependent. Hence, all wavelength-dependent effects can be modeled as linear combinations of the single wavelength model. During an integration of time  $t$  a mean of

$$i = \frac{AEt}{hf} = \frac{AE\lambda t}{hc} \quad \text{[#photons]} \quad (2.16)$$

photons hits a single pixel. Here,  $A$  is the area of the pixel and the second equation is derived from Eq. (2.14). Only a fraction  $\eta(\lambda)$  of the  $i$  photons generates charges that are accumulated in the pixels charge bin. However, thermally generated charges  $i_{\text{dark}}$  are also accumulated in the charge bins. The combined charges are amplified with the gain factor  $K$  of units of  $DN/e^-$ , which is digits per electrons. Note that this is an analog amplification. The amplified analog signal is finally converted into a digital signal in the analog-to-digital converter (ADC), resulting in the measured

digital pixel value  $j$ . This value can be modeled as

$$\begin{aligned} j &= \min ( K(\eta(\lambda)i + i_{\text{dark}}) + n, j_{\text{sat}} ) \\ &= \min ( K\eta(\lambda)i + j_{\text{dark}} + K(n_e + n_d) + n_q, j_{\text{sat}} ) \end{aligned} \quad (2.17)$$

The operator  $\min(\cdot, j_{\text{sat}})$  clamps the measurement to the saturation value  $j_{\text{sat}}$ , which is defined as the maximum measurable gray value. Note that  $j_{\text{sat}}$  is limited by the full well capacity, the amplitude gain  $K$  and the ADC. The corresponding irradiance, however, does not reach the well capacity but a slightly smaller value, as the ADC usually reaches its maximum value before the maximum well capacity is achieved. The variable  $n$  in the first row of Eq. (2.17) models additive noise introduced in the measurement process, which will be discussed in detail below. In the second row, we introduce  $j_{\text{dark}} = Ki_{\text{dark}}$  as the dark signal that is measured without any light incident on the sensor. As  $i_{\text{dark}}$  consists of thermally generated charges, this dark signal depends on the temperature (and grows proportional to the integration time  $t$ ). Note that the measurement can also be expressed in terms of incident irradiance  $E$ , combining Eq. (2.16) and Eq. (2.17) leading to

$$j = \min \left( K\eta(\lambda) \frac{AE\lambda t}{hc} + j_{\text{dark}} + K(n_e + n_d) + n_q, j_{\text{sat}} \right). \quad (2.18)$$

Using Eq. (2.18), the linearity of the sensor can easily be verified by measuring the mean response  $\mu_j$  for varying exposure  $t$  or irradiance  $E$ .

*Noise Model* To describe the stochastic variation in the measurement  $j$ , all variables in Eq. (2.17) except for  $K$  are random variables. The variable  $n = K(n_e + n_d) + n_q$  models additive noise that is introduced at three points in the measurement process. The component  $n_e$  models the shot noise fluctuations of photo-generated charges  $\eta(\lambda)i$  and the thermal noise charges  $i_{\text{dark}}$  [51] as an *additive* offset to their sum. Both, the photo-generated charges and thermal noise charges, follow a Poisson distribution. Since the sum of Poisson-distributed random variables itself is Poisson-distributed, which allows us to use a single random variable  $n_e$ . The second component  $n_d$  from Eq. (2.17) models normal-distributed noise introduced in the read-out of the charges and the amplifier. Both,  $n_e$  and  $n_d$  are amplified by

gain factor  $K$ , which also converts the fluctuations in the electrons to digits. Finally,  $n_q$  models post-amplifier noise in the ADC.  $n_q$  is uniformly distributed between the quantization levels [53]. Due to the laws of error propagation, the (temporal) variance of  $n$  is given as

$$\begin{aligned}\sigma_j^2 &= K^2(\sigma_e^2 + \sigma_d^2) + \sigma_q^2 \\ &= K^2\sigma_d^2 + \sigma_q^2 + K\mu_j,\end{aligned}\tag{2.19}$$

where  $\sigma_j^2, \sigma_e^2, \sigma_d^2, \sigma_q^2$  are the variances of  $j, n_e, n_d, n_q$ . In the second row of Eq (2.19), we have used a property of the Poisson distribution; that is its variance equals its mean, i.e.  $\sigma_e^2 = \mu_j$ . Using Eq. (2.19), it is possible to calibrate the system gain  $K$  from the linear relationship between the variance  $\sigma_j$  and  $\mu_j$ , [51]. The offset  $K^2\sigma_d^2 + \sigma_q^2$  allows calibrating  $\sigma_d^2$ , as  $\sigma_q$  is known from the ADC quantization levels. Finally, having estimated  $K$ , it is possible to calibrate the quantum efficiency  $\eta(\lambda)$  using the linear relationship between  $K\eta(\lambda)$  and  $\mu_j$  described in Eq. (2.17). For a detailed description of this calibration approach, classically known as the Photon Transfer Method, we refer the reader to [54], [55].

Above, we have discussed a sensor and noise model for an individual pixel. Note that often spatial non-uniformities between different pixels are described as an additional type of noise, so-called fixed pattern noise. However, because this effect is temporally static, it is, in fact, not noise but inhomogeneous parameters of the same per-pixel model.

The EMVA Standard 1288 describes standardized calibration methods not only for these non-uniformities but for all parameters of the sensor and noise model that we have introduced above. This also includes the spectral sensitivity, which is captured in  $\eta$  in our model. We refer the reader to [51], [56] for an in-depth discussion and a reference implementation.

*Signal-to-Noise Ratio and Model Approximations* Having introduced the sensor and noise model above, we can analyze the effects of the model parameters on the

signal quality. A common quality measure is the Signal-to-Noise Ratio, that is

$$\text{SNR}(\mu_i) = \frac{\eta(\lambda)\mu_i}{\sigma_j} = \frac{\eta(\lambda)\mu_i}{\sqrt{\eta(\lambda)\mu_i + \mu_{j\text{dark}} + \sigma_d^2 + \sigma_q^2/K^2}}, \quad (2.20)$$

which is defined as the ratio of the mean signal and the standard deviation of the noise. The effect of the quantization noise is damped by factor  $1/K^2$ , and hence is often ignored in practice. An accurate and compact noise model is the mixed Poissonian-Gaussian model from [57]. Reducing the model to only Poisson and Gaussian noise allows exploitation of the heteroskedastic normal approximation

$$P(\mu) \approx N(\mu, \mu), \quad \begin{array}{l} \text{Heteroskedastic} \\ \text{Approximation} \end{array} \quad (2.21)$$

where here  $P(\mu)$  is a Poisson distribution with mean  $\mu$  and  $N(\mu, \mu)$  is a Gaussian distribution with mean and variance  $\mu$ . This approximation becomes more accurate with increasing  $\lambda$ . It allows modeling the overall noise  $n$  as a Gaussian distribution with a spatially changing variance, depending on the (unknown) latent signal value  $\eta(\lambda)\mu_i$  [57]. While this approximation simplifies inference tasks for high and medium photon scenarios, low photon scenarios are common in many applications. The border cases are

$$\text{SNR}(\mu_i) \approx \begin{cases} \sqrt{\eta(\lambda)\mu_i} & \text{for } \eta(\lambda)\mu_i \gg \mu_{j\text{dark}} + \sigma_d^2 + \sigma_q^2/K^2 \\ \frac{\eta(\lambda)\mu_i}{\sqrt{\mu_{j\text{dark}} + \sigma_d^2 + \sigma_q^2/K^2}} & \text{for } \eta(\lambda)\mu_i \ll \mu_{j\text{dark}} + \sigma_d^2 + \sigma_q^2/K^2 \end{cases}. \quad (2.22)$$

The SNR changes from a linear dependency on  $\mu_i$  in the low photon range to a square root relationship in the high photon range. In other words, for well-exposed images, the SNR scales with square root of incoming photons. Using Eq. (2.16), this means that increasing the pixel area by 4 (e.g. doubling width and height) or increasing the exposure time by 4 will increase the SNR by  $2\times$  (given that no saturation occurs).

Finally, we can define an ideal sensor as one with no read-out noise, no quantization noise, dark current of 0, and quantum efficiency of  $\eta(\lambda) = 1$ . Using Eq. (2.20), an ideal sensor defines the upper bound  $\text{SNR}_{\text{ideal}}(\mu_i) = \sqrt{\mu_i}$ .

*Multi-Shot Capture* Due to saturation in our model from Eq. (2.18) and motion blur for dynamic scenes, the SNR cannot be arbitrarily increased by a longer exposure time. Furthermore, the square-root dependence requires long exposure times for significant improvements in SNR. A successful approach to enabling longer exposure times is using a burst of short exposure captures [58]. The burst captures have to be registered to compensate for camera or scene motion but can drastically increase the SNR [59]. Capturing several images also allows for an increased dynamic range, which was introduced above. High-dynamic-range (HDR) imaging can be implemented by capturing an image stack with different exposure times. By doing so, measurements can be reconstructed that contain more detail in the very dark and bright areas than would otherwise be possible. We refer the reader to [17] for a detailed discussion. Multiple frames can also be combined to reduce motion blur [60] or to super-resolve more detail than is available from a single image [61].

#### 2.4.4 Color Imaging

The sensor model from above has been described for incident light of a single wavelength. For a spectrum of light  $\Omega$  the model can be easily extended by integrating the photo-generated charges in  $\Omega$ , as done in the charge bucket by the accumulation. Eq. (2.17) becomes

$$j = \min \left( K \int_{\Omega} r(\lambda) \eta(\lambda) i(\lambda) d\lambda + j_{\text{dark}} + n, j_{\text{sat}} \right), \quad (2.23)$$

where  $i(\lambda)$  is the incident number of photons per wavelength  $\lambda$ . As already described in Section 2.3.3, image sensors sample the incident spectrum by adding different spectral band-pass filters on the pixels, i.e. color filters. As perfect band-pass filters are hard to realize in practice, we model the filters with a general modulation  $r : \mathbb{R} \rightarrow \mathbb{R}^+$ . In order to sample different spectral bands, the different filters are spatially multiplexed on the pixel array. Different sets of spectral filters may be used, whose measured response (the accumulated  $j$ ) eventually can be converted into a parameter vector of a chosen color space. A key result from colorimetry is that in most scenarios the response from three band-pass filters is enough to visually distinguish colors. Hence, commonly, three color filters peaking around

the red, green and blue band are used, arranged in a Bayer array [62]. The colors can then also be described with three scalars, i.e. the R, G, B component, which may be converted to different color spaces. We refer the reader to [17] for a detailed introduction to photometry and colorimetry.

*Color Image Processing* Early color image processing aims at estimating these RGB pixel values at full sensor resolution from the noisy, subsampled measurements  $j$ . Traditionally, this task is implemented as a pipeline of simple stages [63]. These image signal processors (ISPs) usually take in raw Bayer sensor measurements, interpolate over defective or stuck pixels, demosaick the sparse color samples to a dense image with RGB in every pixel [64], attempt to denoise the noisy signal [65], enhance edges, tonemap the image to 8 bits per channel, and optionally compress the image. The camera capture parameters are controlled by the auto exposure, focus, and white balancing algorithms [66].

Demosaicking and denoising play key roles in this image processing pipeline because they are both hard inverse estimation problems. We refer the reader to [64] for an overview of state-of-the-art demosaicking methods. Many approaches have been proposed for image denoising. Self-similarity and sparsity are two key concepts in modern image denoising. Non-local image modeling utilizes structural similarity between image patches. Non-local means (NLM) [67] filters a single image using a weighted average of similar patches for each pixel. Many orthogonal transforms, such as DFT, DCT, and wavelets, have good decorrelation and energy compaction properties for natural images. This property has been utilized in local transform-based denoising schemes, such as wavelet shrinkage [68] and sliding window DCT filtering [69]. BM3D [70] was the first denoising algorithm to simultaneously exploit sparse transform-domain representations and non-local image modeling. Combining internal (such as BM3D) and external (such as Total Variation) denoising has recently been shown to be advantageous; [71] run two methods separately and then merge the results.

### **2.4.5 Modified Sensor Designs**

A variety of sensor designs and capture methods has been proposed that differ from traditional Bayer array color imaging with a single main lens as discussed above. One line of research is to use sensor arrays with individual optics. An extremely thin camera design is possible by integrating several small lenses over a sensor, as in the TOMBO [72] and PiCam [73] design. By using per-lens spectral filters, the PiCam design eliminates the Bayer array. Furthermore, the optics can be drastically simplified as chromatic aberrations only need to be corrected over a small wavelength band of the per-lens filter. Tiled sensors like PiCam also allow measuring angular components of the light transport from the different tiled viewpoints in a so-called *light field* [74, 75]. Captured light field can be post-processed to allow changing of the viewpoint or focus, creating a large virtual aperture, extracting depth from the scene, etc. Traditional Bayer sensor designs have also been modified to extend the dynamic range that they capture. For example, the density of the color filter array can be varied per pixel [76], but this comes at the cost of reduced light efficiency. Alternatively, the sensor design itself can be modified to allow for per-row selection of the exposure time or sensor gain [77, 78].

## **2.5 Time-of-Flight Imaging**

Having discussed a broad variety of image sensors that measure irradiance, this section on TOF imaging describes a fundamentally different sensor type. TOF imaging systems measure the Time-of-Flight of light traveling from an active source along a global illumination path through the scene back to the sensor. Assuming a collocated source/sensor and a single reflector, such a TOF measurement immediately allows inferring the distance of this reflector.

### **2.5.1 Optical Range Imaging**

A large variety of different methods for optical range measurements exists. Stoykova et al. [79] give an extensive overview. Optical range systems can be categorized into the two main groups of triangulation-based and collinear setups, where the sub-categories single/multi-camera and passive/active exist.

*Collinear methods.* Collinear setups do not rely on a baseline (for triangulation) and can, in theory, share the optical path for illumination and camera optics (although that is in practice mostly not the case). Passive collinear single camera systems include the shape-from-X methods: shape-from-shading [80], [81], shape-from-defocus [82], [24], shape-from-texture [83]. Due to major ambiguities in the forward model describing the image formation, these techniques lead to extremely challenging inverse problems lacking efficient regularization techniques. Active collinear single-camera systems include TOF-based systems [84], [85], including also shutter-based approaches [86], [29], depth-from-defocus [87], intensity-falloff-based methods [88] and also methods based on interferometry [89], [79].

*Triangulation-based methods.* Triangulation-based methods require a baseline between multiple cameras, or between illumination and camera (if the system is active). Passive triangulation-based systems using multiple cameras [90] rely on solving the correspondence problem [91], which is computationally hard and can fail in regions with little texture. Furthermore, the depth accuracy depends on the size of the baseline, which is undesirable for small devices. Active triangulation-based systems rely on modulated illumination and a camera for triangulation, [92], [93]. The modulated illumination is introduced to solve the correspondence problem [94]. While being robust to regions with little texture, the methods suffer from accuracy for small baselines and ambient illumination, thus need high-intensity focused illumination. To achieve high accuracy, many structured light methods rely on multiple projected patterns and therefore are severely limited by acquisition speed. Spatial multiplexing [95] can overcome some of these limitations, however, at the cost of significantly decreasing spatial resolution and adding expensive computation.

*Comparison of optical range imaging methods.* Comparing all optical ranging systems listed above, passive single-camera systems represent the class with the lowest accuracy due to the major ambiguities in the image formation [79]. Passive triangulation-based systems resolve many of these ambiguities by introducing multiple views and therefore significantly improve accuracy in comparison to passive single-camera systems [91]. However, precision is still limited by solving



the correspondence problem, which is significantly simplified with active structured light illumination methods. Comparing now TOF cameras to triangulation-based systems, Beder et al. [96] show that, for indoor scenarios, TOF cameras improve depth accuracy an order of magnitude over passive stereo systems. Comparable structured light systems can achieve similar accuracy as demonstrated by Langmann et al. [97]. However, the main difference between TOF cameras and triangulation-based systems is that the accuracy of the latter is a quadratic function of the inverse depth and only linearly depending on the baseline as derived in [98]. While the range of TOF cameras can be extended by simply increasing the illumination power, this is a fundamental limit for active and passive triangulation-based systems. It also means that high accuracy requires large baselines, which severely restricts the size of the system. Because of this fundamental limitation and the drawbacks of triangulation-based methods outlined in the previous paragraph, TOF imaging is a very promising technology. Below, we review TOF imaging in detail.

### 2.5.2 Impulse Time-of-Flight Imaging

TOF range imaging systems can be classified by the type of their illumination modulation, that is impulse modulation or continuous wave modulation. In impulse Time-of-Flight imaging, a pulse of light is emitted into the scene and synchronously an accurate timer is started. The light pulse travels from the illumination source to the object and is directly reflected back to the sensor (assuming direct reflection at the scene object). Once the sensor detects the reflected light pulse, the timer is stopped, measuring the round-trip time  $\tau$ . Given the speed of light  $c = 299,792,458 \text{ m/s}$  and assuming collinear sensor and illumination, one can then calculate the depth  $d$  of the object point reflecting the pulse as  $d = c\tau/2$ . Classical Lidar range imaging [99], [26], is based on impulse Time-of-Flight imaging, with [100] among one of the first introducing a practical Lidar system.

Impulse illumination has the advantage that a high amount of energy is transmitted in a short time-window. Therefore, high SNR is achieved with regard to background illumination [27], while the mean optical power stays at low eye-safe levels. The high SNR allows for high accuracy measurements over a wide range of distances [26]. A further advantage is that Multi-Path Interference can be reduced

simply by detection of the first peak and temporal gating. We will discuss Multi-Path Interference in detail later in this section.

Drawbacks of impulse illumination are that due to varying ambient illumination, attenuation due to scattering and varying scene depth, both the peak intensity of the pulse and the ambient illumination offset change. Thus, the sensors require a large dynamic range to detect pulses, which are usually implemented by thresholding [99]. Another drawback of pulsed illumination is that currently only laser diodes deliver the required short pulse width with high optical power [27]. However, the pulse repetition rate of laser diodes is limited to 10 kHz - 150 kHz in recent systems [101]. This severely limits the frame rates of scanning systems that scan the scene point-by-point and also average multiple captures for improved SNR. Furthermore, scanners significantly increase sensitivity to vibrations, cost and system size [85]. In fact, most impulse-illuminated systems are scanning systems [26]. This is due to the difficulty in implementing high-accuracy pixel-level timers [84]. To illustrate the required precision, note that for a distance resolution of 1 mm, light pulses need to be separated with a time resolution of 6.6 ps. Only recently, advances in implementing large arrays of Single-Photon Avalanche Diodes (SPAD) in standard CMOS enable efficient scanner-less pulsed range imaging [102]. However, SPADs implemented in silicon technology at room temperature are not able to reach the timing resolutions necessary for mm-resolution [102]. Therefore, either simple averaging of many periodic pulse trains in time-correlated single-photon counting (TCSPC) or continuous wave modulation is done to improve the timing uncertainty. The principles and benefits of continuous wave modulation are described in the following paragraphs.

### **2.5.3 Continuous Wave Time-of-Flight Imaging**

In continuous wave TOF imaging, rather than using impulse illumination, the light source is modulated. The electromagnetic waves present in the illumination itself (e.g. in the case of a laser illumination only a very narrow spectrum) represent the carrier wave that gets modulated. In general, different modulation methods exist. While frequency modulation for TOF imaging has been explored [103], by far the most popular modulation technique is amplitude modulation due to its

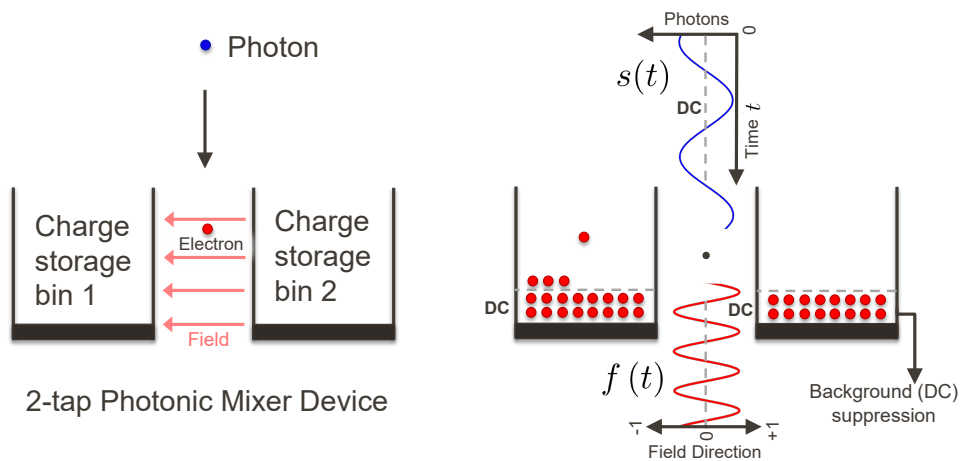
comparatively very simple implementation. As the fundamental principle of these sensors is based on temporal correlation (or convolution), we use in this dissertation the term *Correlation Image Sensors (CIS)*.

In CIS range imaging the intensity of the light source is modulated with a periodic modulation signal. Rather than directly measuring the TOF as in impulse TOF imaging, the phase difference  $\theta$  between a reference signal (often the transmitted signal) and the received signal is measured.  $\theta$  can be estimated by cross-correlation between the reference and received signal. Given the modulation frequency  $f$  of the periodic illumination, it is  $\theta = 2\pi f\tau$  and thus the depth is  $d = c\tau/2 = c/(4\pi f)\theta$ . One of the first CIS systems was introduced by Lange [27]. A survey of recent systems and their limitations can be found in [85], [84]. Using Amplitude Modulated Continuous Wave (AMCW) illumination rather than pulsed illumination has the advantage that the light source does not need to provide short pulse width at high rates, thus, in practice, inexpensive LED illumination can be used [27]. Furthermore, accurate per-pixel timing is required for pulsed TOF imaging. This is not necessary for Photonic Mixer Devices or time-gated photon counters [84], which therefore can be implemented significantly more easily.

A variety of different modulation waveforms has been proposed. Most common are sinusoidal waves or square waves [104], [27]. Frequency modulation via chirping has been explored in [105]. Pseudo-noise modulation has been proposed in [106], and pseudo-random coded illumination was used by Whyte et al. [107].

*CIS sensors* Sensors for correlation imaging are based on two working principles: correlation during or after exposure. The most popular class of sensors performs the correlation of the incoming modulated signal *during exposure*. This is achieved by using Photonic Mixer Device (PMD) pixels that direct the photo-generated charge toward multiple taps (commonly two). Such a pixel is illustrated in Figure 2.9. The charge accumulation and transport are achieved using the CCD principle discussed in Section 2.4. The main difference from a common CCD image sensor is that the charge transfer between the group of accumulation sites occurs during the exposure. The direction of the transfer can be modeled as a scalar between -1 (maximum field strength toward the left bin) and +1 (fully toward the right bin) according to a temporally varying modulation function  $f$ . The left bin corresponds then to

negative values while the right models positive values. This means that the sensor essentially *correlates* a temporally varying illumination function  $g$  with the sensor modulation function  $f$ ; see the right side of Figure 2.9. The correlation value is the subtraction of the left from the right (as we assigned negative values to the left bin). Using a zero-mean sensor modulation function  $f$ , unmodulated ambient light is removed in this subtraction as constant bin offset after or even during integration. This is also illustrated on the right side of Figure 2.9. The CIS sensor design was first proposed by [108], the application to phase imaging was shown in [109] and simultaneously a design for cm-accurate depth imaging was first presented by Schwarte et al. [110], who also introduced the term Photonic Mixer Device [111]. We refer the reader to [27] for a detailed discussion of the Photonic Mixer Device sensor design. Nearly all commercial CIS designs are currently based on PMD technology [85]. An alternative approach is to do the correlation *after exposure*



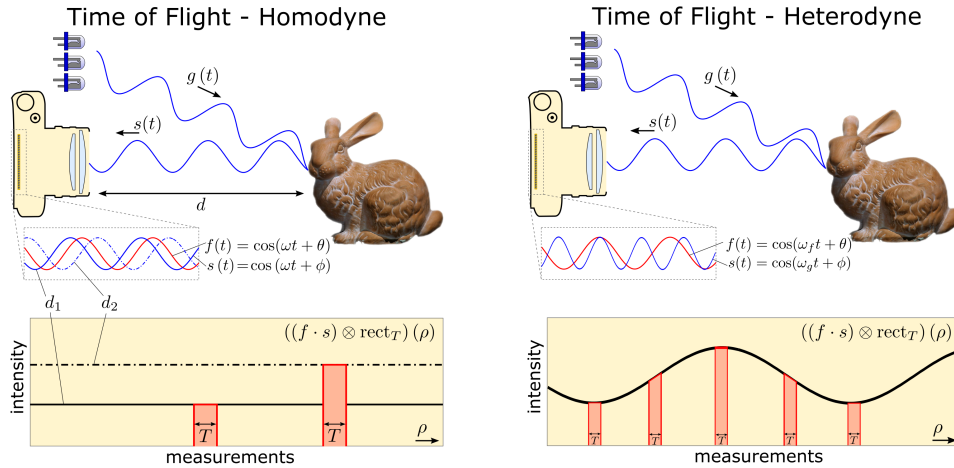
**Figure 2.9:** Working principle of a PMD pixel. Photo-generated charges are directed toward two bins using an electric field between two electrodes, i.e. the CCD transport principle (left). The intensity  $g$  of the light source and the electric field  $f$  is modulated during integration (right). The sensor *correlates* the signal  $g$  with  $f$ . Unmodulated light is accumulated equally in both buckets and can be removed during or after integration.

on the electronic signal generated by the photosensitive pixel during measurement. The correlation can be done with analog electronics using standard photodiodes as

sensors. However, this sensing approach requires complex electronics per pixel, has high-power consumption and suffers from noise in the signal processing [84]. Using SPAD sensors in Geiger mode, the correlation can be performed in the digital domain after read-out; see [112].

## 2.6 Working Principles of CIS

Having listed the benefits of continuous wave TOF imaging using CIS in the previous section, we now describe the basic working principles of this technology in detail. Sinusoidal waves are often used in the CIS literature [85] to approximate the actual shape of the modulation waveform. In the following, we assume a sine wave model for simplicity of notation. However, it can be expanded straightforwardly to a general model by using a superposition of all harmonics for the modulation waveforms. Figure 2.10 on the left presents an overview of the CIS image formation using sinusoidal modulation.



**Figure 2.10:** CIS TOF imaging with its most common capture modes. Left: Homodyne setup using the same frequency for illumination and sensor modulation. Right: Heterodyne setup using different frequencies.

Assuming a single diffuse reflector in the scene, the light source sends intensity-modulated light  $g(t)$  with a periodic modulation signal into the scene. Assuming

that the modulation signal is cosine, the signal incident on the sensor is  $s(t)$ :

$$\begin{aligned}
 g(t) &= \cos(\omega_g t) + 1 \\
 s(t) &= a \cos(\omega_g t + \phi) + I \\
 f(t) &= \cos(\omega_f t + \theta)
 \end{aligned} \tag{2.24}$$

We see that the illumination  $g(t)$  received a phase shift  $\phi$  due to the round-trip travel time of the photons, a change in amplitude  $a$  and an offset  $I$  due to ambient illumination. As discussed in the previous paragraphs, each Photonic Mixer Device can direct incoming photons into two neighboring buckets. By doing this at a certain temporal rate, the incoming signal can be multiplied with another periodic modulation function  $f(t)$ . Assuming again that this is a cosine with a phase shift  $\theta$  as in Eq. (2.25), we get the continuous signal  $p(t)$  in Eq. (2.25).

$$\begin{aligned}
 p(t) &= [\cos(\omega_f t + \theta)] \cdot [a \cos(\omega_g t + \phi) + I] \\
 &= \frac{a}{2} \cos((\omega_g - \omega_f)t - \phi + \theta) + \\
 &\quad \frac{a}{2} \cos((\omega_g + \omega_f)t + \phi + \theta) + \\
 &\quad I \cos(\omega_f t + \theta)
 \end{aligned} \tag{2.25}$$

The continuous  $p(t)$  is then integrated over the exposure time  $T$  and results in the measurement  $b$ . The reason why the sensor modulates is that due to the high frequency of the illumination (in the MHz range), one would need very high temporal sampling rate to sample the incoming signal  $s$  directly. As we will see below this is not necessary for the sampling of the modulated  $s$ . Note that CIS sensors allow for extended exposure intervals, much like regular cameras, by integrating over a very large number of periods of the wave, thus essentially *correlating*  $f$  and  $s$ . This is a key benefit of CIS sensors, in contrast to impulse-based systems.

### 2.6.1 Homodyne Measurement

In general, the angular frequencies  $\omega_g, \omega_f$  of the illumination/sensor modulation can be different. In a homodyne setup, we have  $\omega_f = \omega_g$ . In a heterodyne setup, it is  $\omega_f \neq \omega_g$ . The homodyne setting is used in nearly all commercial systems [85]. It

is illustrated on the left of Figure 2.10 with  $\omega = \omega_f = \omega_g$  and results in Eq. (2.26):

$$b_\theta(\rho) = \int_\rho^{\rho+T} p(t) dt = \frac{1}{2}a \cos(\theta - \phi), \quad (2.26)$$

where the last two terms from Eq. (2.25) disappear due the integration (which is low-pass filtering plus sampling) when assuming the  $T \gg 1/\omega$ . As illustrated in Figure 2.10,  $b$  is constant over time. This is the reason why it was useful to include the phase shift  $\theta$  in the sensor modulation. This additional shift allows us to evaluate the correlation measurement  $b$  at different samples. Evenly sampling for  $\theta_i = i \cdot 2\pi/N$  for  $i \in \{0, \dots, N-1\}$  allows us to do a frequency analysis of  $b$  by using the discrete Fourier transform. The signal of interest will be in the first frequency bin, giving the estimate in Eq. (2.27).

$$\begin{aligned} \phi_{est} &= -\text{atan} \left( \frac{\sum_i b_{\theta_i}(0) \sin(\theta_i)}{\sum_i b_{\theta_i}(0) \cos(\theta_i)} \right) \\ a_{est} &= \sqrt{\left( \sum_i b_{\theta_i}(0) \sin(\theta_i) \right)^2 + \left( \sum_i b_{\theta_i}(0) \cos(\theta_i) \right)^2} \end{aligned} \quad (2.27)$$

For  $N = 4$ , one gets the form commonly mentioned in the literature [29]. Having finally computed the estimate  $\phi_{est}$  of  $\phi$ , one can simply compute the distance  $d$  from the time of flight using  $d_{est} = \frac{c}{2 \cdot \omega} \cdot \phi_{est}$ .

## 2.6.2 Heterodyne Measurement

For the heterodyne setting [113, 114], it is  $\omega_g \neq \omega_f$ , which is illustrated on the right in Figure 2.10. In particular, let us assume a small frequency shift  $\omega_f = \omega_g - \omega_\delta$ . We then get from Eq. (2.25):

$$\begin{aligned} b_\theta(\rho) &= \int_\rho^{\rho+T} p(t) dt = \frac{a}{2} \int_\rho^{\rho+T} \cos((\omega_g - \omega_f)t - \phi + \theta) dt + \\ &\quad \frac{a}{2} \underbrace{\int_\rho^{\rho+T} \cos((\omega_g + \omega_f)t - \phi + \theta) dt}_{\approx 0} + \underbrace{\int_\rho^{\rho+T} I \cos(\omega_f t + \theta) dt}_{\approx 0} \quad (2.28) \\ &\approx \frac{a}{2} \cos(\omega_\delta \rho - \phi + \theta), \end{aligned}$$

We see that the difference in the two modulation frequencies introduces a low-frequency beating pattern that preserves the distance-dependent phase shift  $\phi$ . This is illustrated in Figure 2.10 at the bottom right. Again, as in the homodyne case, the high-frequency terms are removed in the integration. In particular, we assume here a finite integration period  $T$  with  $2\pi(1/T) \gg \omega_f$  and  $\omega_f \gg \omega_\delta$ . Integrating from  $-\infty$  to  $+\infty$  would result in  $b_\theta(\rho) = 0$ , and no meaningful signal would be extracted. In fact, the last approximation for Eq. (2.28) assumes that  $T$  is small enough so that the cosine integrand is locally linear.

Finally, the reconstruction of the unknowns  $\phi, \theta, a$  can be done by sampling the harmonic  $b$  over time with  $\theta_i = i \cdot \omega_\delta / FPS$ , where  $FPS$  are here the camera FPS. Estimating the unknowns from the samples is then done in the same way as in Eq.(2.26).

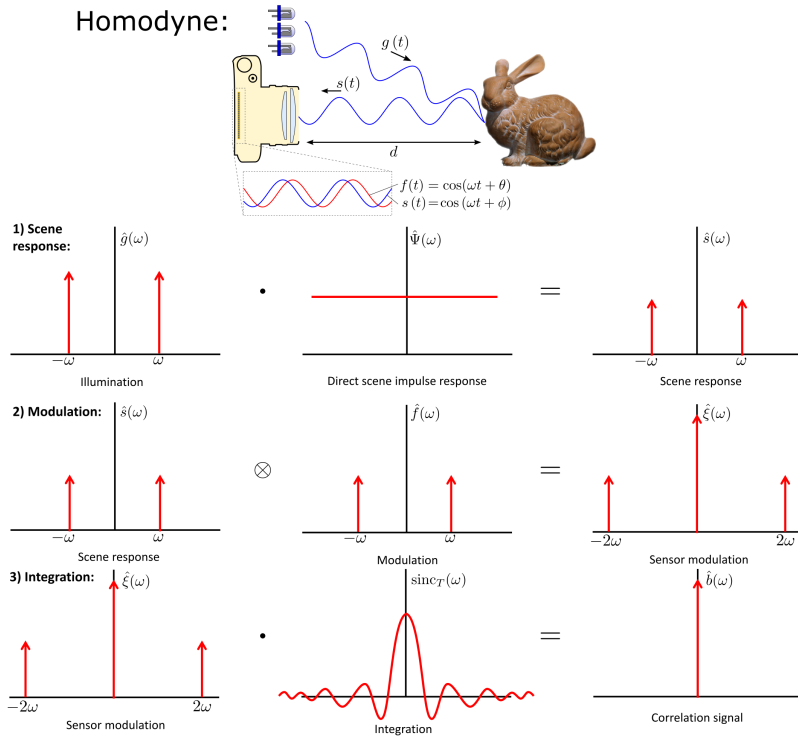
### 2.6.3 Analysis in the Spectral Domain

In the previous paragraphs, we have derived and explained the image formation for CIS in the temporal domain. The same image formation model can also be derived in the frequency domain, which can help build a deeper understanding of the measurement process. Figure 2.11 shows a spectral analysis of the homodyne and heterodyne measuring mode. The plots show the amplitude spectra. Phase spectra remain unchanged except the first row, where interaction with the scene introduces the distance-dependent phase shift  $\phi$ .

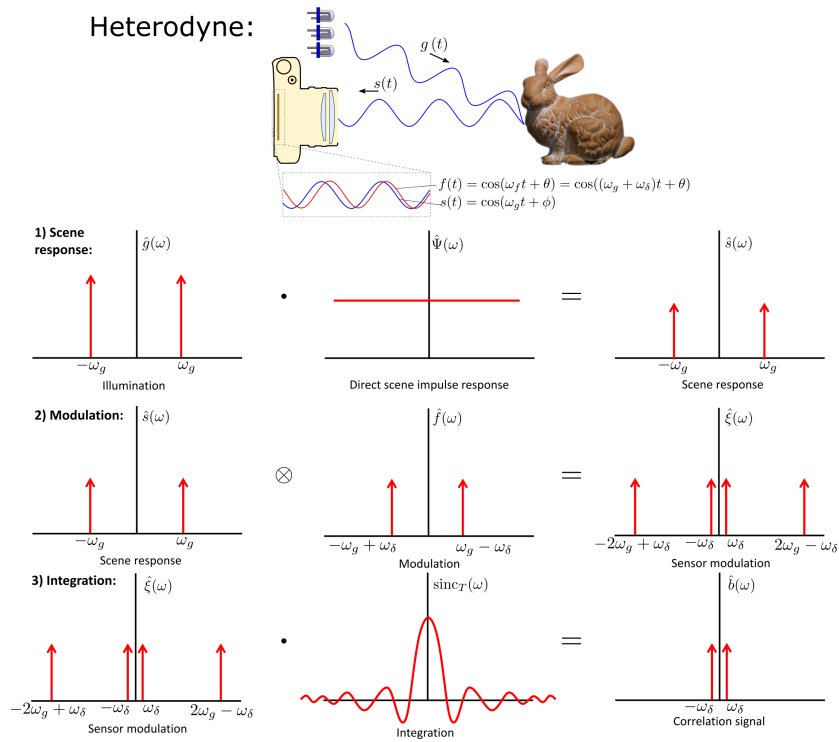
In the first row, the amplitude spectrum of the single-frequency illumination spectrum  $\hat{g}$  gets attenuated by the factor  $a$ . We absorb here illumination distance fall-off and non-perfect reflection at the scene object in this factor. The second row shows that the spectrum  $\hat{s}$  incident on the sensor gets *convolved* with the modulation spectrum  $\hat{f}$ . This is because multiplication in the spatial domain corresponds to convolution in the spectral domain. This convolution results in a high-frequency component, the sum of  $\omega_g + \omega_f$ , and a low-frequency component, the difference of  $\omega_g - \omega_f$ . Thus, in the homodyne setup, the low frequency is, in fact, 0. In the heterodyne configuration, it is a small beating frequency (assuming that  $\omega_g$  and  $\omega_f$  lie close together). The last row in Figure 2.11 shows that the integration during exposure corresponds to multiplication with a sinc low-pass filter in the spectral



### Homodyne:



### Heterodyne:



**Figure 2.11:** Spectral analysis of CIS TOF imaging for the homodyne measuring mode (top) and heterodyne measuring mode (bottom).

domain, removing the high-frequency components. A mathematical derivation of the spectral analysis discussed above can be found in the Appendix in Section A.1.

#### 2.6.4 Sensor and Noise Model

The model used so far is an idealized one. It includes neither noise, nor systematic deviations from the ideal sensor or light source. However, in practice, the most commonly used operating mode is the homodyne mode with the inverse trigonometric estimation from Eq. (2.27). Therefore, various types of errors in the depth estimation can be observed. A list of the most severe deviations is given in [85]. One can observe a systematic distance-dependent error, i.e. wiggling error [115], an intensity-dependent depth error [116], temperature-dependent depth deviations [117], exposure-time-dependent deviations [117], static pixel inhomogeneities [85], and random fluctuations (noise) in the depth estimate. Lindner and Kolb [116] and Khalmann et al. [117] identify the error components depending on distance, integration time and amplitude as the most severe ones. Both approaches reduce these systematic errors by first measuring the deviation for a range of different estimated depths, amplitudes and integration times. A depth estimate is then corrected using the interpolated deviation from a 3D lookup-table or B-spline [116]. However, this data-driven approach does not identify the inaccuracies in the model corresponding to the observed depth error, and therefore it does not generalize well beyond depth estimation (especially for higher-dimensional CIS-based imaging such as transient imaging).

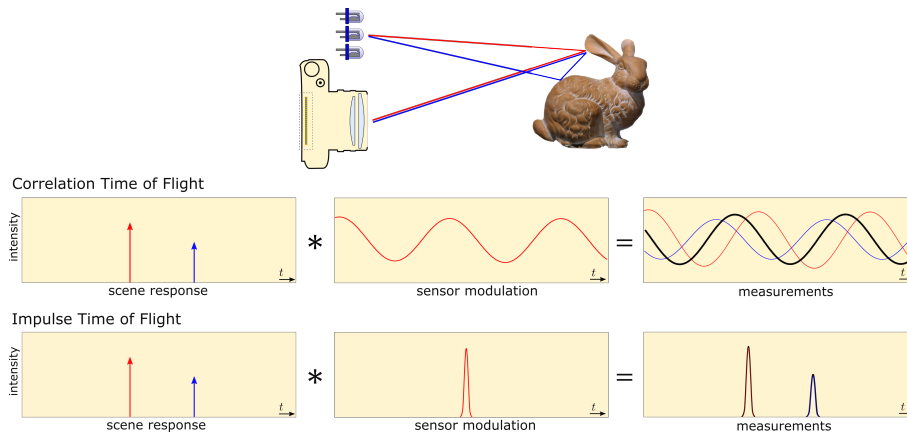
In contrast, Schmidt and Jähne [118] and Schmidt [52] propose a modified physical model to explain the observed error. As discussed in the previous section, CIS pixels are groups of collection sites with CCD transfer to the collection sites. Hence, the EMVA Standard 1288 for image sensors, reviewed in Section 2.4.3, can be extended to model CIS sensors. To represent the systematic errors, the works [118] and [52] extend it to allow for non-sinusoidal illumination  $g$  and sensor modulation  $f$ , non-linear photo-response, adaptive suppression of background illumination and spatial non-uniformities. The depth-dependent errors can then be explained by higher-order harmonics in the modulation signals, whereas the amplitude and intensity dependence are due to a non-linear photo-response and background suppression.

While a non-linear photo-response can easily be inverted after integration in each bucket [52], different approaches address non-sinusoidal modulation. Payne et al. [119] propose a phase-shifting scheme during the integration, which significantly reduces higher-order harmonics. Without changing the acquisition, the ideal sinusoidal model can be expanded to allow for arbitrary periodic modulation signals using a superposition of all harmonics present in the waveform. This harmonic CIS model is derived in Section A.2. Noise can be modeled as a sum of shot noise, read-out noise and amplifier noise, which are independent between taps, exactly as discussed for image sensors in Section 2.4.3. In the case of a sensor reading out all taps independently, the quantization noise can also be considered per tap. If a tap-difference value is read out, the quantization is applied after the tap subtraction. For a 2-tap sensor, the mixed Poisson-Gaussian noise model from Section 2.4.3 then becomes a Skellam-Gaussian noise model. We refer the reader to [52] for a discussion of calibrating the sensor and noise model parameters, again closely following the EMVA 1288 calibration procedures.

### **2.6.5 Multi-Path Interference**

In CIS TOF imaging, it is essential to handle the effects of global illumination. Describing the basic measurement principles above, we have assumed illumination along a single path from the light source to a single diffuse reflector and back to the camera pixel. In general scenes, however, emitted light can be reflected or scattered many times at different scene points until returned to the sensor. Thus, multiple returns along different paths can be superimposed at the sensor, which is the Multi-Path Interference (MPI) problem. Resolving MPI is critical, especially in CIS TOF imaging, for two reasons. First, the scene is flood-illuminated, creating significantly more potential MPI than with a scanned focused beam illumination. Second, when sine modulation is used (as in many commercial systems [85]), multiple superimposed sinusoids of the same frequency result in another sinusoid with the same frequency but different phase. See the illustration in Figure 2.12.

This means that even for a very low number of mixed paths, resolving MPI is hopeless with a single frequency. In contrast, impulse TOF could untangle a low number of separated paths using simple thresholding. Recently, many approaches



**Figure 2.12: Multi-Path Interference.** In the scene on the top, contributions from direct and indirect light paths (direct red and indirect blue) are superimposed in the measurement. In CIS TOF imaging, the sinusoids for each path result in another measured sinusoid of different phase (black), indistinguishable of a single reflector at a different depth. Impulse TOF can separate the components in principle. However, the sensor requires extremely high temporal resolution and almost continuous readout.

have been proposed to mitigate MPI effects. The methods from [120], [121], [122] all model light transport only for the piecewise Lambertian scenes. A scene approximation step followed by an MPI reduction step based on the previous scene proxy is done (either once or iteratively [122]). Another class of methods models sparse two-path MPI for specular scenes [123], [124], [125]. Reduction of MPI is performed per-pixel without estimating a scene approximation. All of the methods described so far consider only special cases of general light transport. Recently Freedman et al. [126] made an attempt to generalize per-pixel MPI reduction by assuming a temporal sparse backscatter signal with a robust data-term. The multi-frequency method can be implemented efficiently using a look-up-table. Finally, the last class of methods not only relies on CIS measurements, like all approaches above, but uses additional spatial light modulation [127],[2]. Recently, O’Toole et al. [2] achieved state-of-the-art results combining spatial light modulation and sparsely coded backscatter.

### 2.6.6 Dynamic Scenes

*Motion Artifacts* So far we have considered static scenes. Hence, if measurement time is of no concern, an arbitrary number of measurements can be acquired sequentially, i.e. temporally multiplexing. However, real-world scenes can contain motion of the camera or any of the objects in the scene. This scene motion consists of an axial component along the optical axis and a lateral component in the orthogonal plane. Due to the motion, in practice, temporal multiplexing requires the overall capture time of all  $N$  for a range estimate to be small compared to the scene motion. This can be achieved by increasing the frame-rate of the camera or the number of measurements that can be acquired simultaneously during a frame, for example through spatial multiplexing. 4-tap (or 4-bucket) sensor designs have been proposed [27], [128], which add two more charge buckets to each pixel. Hence, four correlation measurements can be acquired during a single exposure [128]. However, spatial multiplexing sacrifices spatial resolution, which is low for current CIS sensors [85] due to the large pixel structures required for the CCD-based charge transport. Hence, the majority of CIS TOF camera systems use the discussed 2-tap design as a trade-off between pure time multiplexing (1-tap) and pure spatial multiplexing (4-tap).

Different approaches addressing motion artifacts for 2-tap CIS sensors have been proposed in the literature. [52] and [129] present models for the raw measurements in each bucket and corresponding calibration procedures. Doing so, they are able to reduce the bias between the individual taps of a CIS, which is usually removed by simply using the measurements  $\theta_2, \theta_3$  from Section 2.6.1. These two measurements are  $180^\circ$  offset with regard to  $\theta_0, \theta_1$  and therefore exactly invert the role of the buckets in the correlation. Having removed the systematic bias between the two taps, the measurements for  $\theta_2, \theta_3$  are no longer necessary, and  $N$  becomes 2, that is the frame rate increases by  $2\times$ . Another approach is presented in [130] and [131]. Both works accumulate the values from both buckets of a CIS pixel, yielding an intensity image of the scene. Having computed intensity images for all sequentially captured measurements, optical flow is computed between all frames, and the raw data is warped to the first frame.

*Doppler Effect* While scene motion causes artifacts when correlating sequential frames, sinusoidal illumination, in fact, encodes both motion and depth information, even during a single frame. Radial velocity is encoded in the frequency due to the Doppler effect, and depth, as before, in the phase. The Doppler effect describes the change in frequency of a wave for the receiver of this wave that is moving relative to its source. For an emitted wave of frequency  $\omega_g$ , a reflected signal of frequency  $\omega_s = \omega_g + \Delta\omega$  is arriving on the sensor, where the frequency shift depends on the radial object velocity as well as the emitter frequency:

$$\Delta\omega = \frac{v}{c} \omega_g. \quad \text{Doppler Shift (2.29)}$$

Christian Doppler first discovered this effect, observing that the spectrum of astronomical objects shifts depending on their velocity [132]. Since then the Doppler effect has found widespread use in astronomical imaging, meteorology, traffic law enforcement, radiology, healthcare, and aviation. Doppler spectroscopy, for example, measures radial velocity of otherwise undetectable planets by observing wavelength shifts of their respective stars [133]. The rate of expansion of the universe can be estimated by Doppler spectroscopy as well [134]. Doppler velocimetry is a common technique in healthcare to measure blood flow using ultrasound or lasers. The basics of ultrasound Doppler velocimetry can be found in [135]. Albrecht [136] gives a detailed survey of laser Doppler velocimetry. Laser Doppler velocimetry also has found widespread commercial adoption with the contact-less measurement of movement in computer mice [137]. Doppler radar velocimetry uses radio waves for velocity estimation. This technology has numerous applications in aviation, meteorology, and remote sensing. Meikle [138] presents a detailed survey of state-of-the-art radar technologies. In its simplest form, continuous wave Doppler radar sends out continuous electromagnetic waves and determines the velocity of the targeted scene object directly from the returned Doppler-shifted signal. This method is commonly used in police speed guns. Doppler pulse radar simultaneously measures velocity and range and is broadly used in aviation for tracking, classification and prediction [139]. In Doppler pulse radar, short pulses of radio energy are sent out to an object. The round-trip TOF of the returned pulse determines a range estimate, while its Doppler shift provides an estimate of the velocity. Both Doppler

radar and Doppler Lidar are commonly used in meteorological applications, such as wind velocity estimation [140, 141].

All of the methods reviewed above rely on the Doppler effect of the light wave itself, that is the carrier wave for a CIS system. As the frequency of the amplitude modulation in a CIS TOF system is six orders of magnitude lower than the carrier frequency, the Doppler effect is also reduced by this factor, recalling Eq. (2.29). For example, given a typical modulation frequency of 50 MHz and a speed of 10 m/s, this results in a tiny frequency shift of  $1.67Hz$ . However, by exploiting the temporally convolutional image formation of CIS sensors, we show in Chapter 7 that it is nevertheless possible to extract the Doppler shift using inexpensive CIS TOF cameras. Our approach allows imaging of both axial velocity and range. It is a full-field imaging method, meaning that it *does not require the scene to be scanned sequentially* unlike most existing Doppler radar or Lidar systems that capture only a single scene point at a time.

## 2.7 Transient Imaging

With the light transport models from the previous sections at hand, we now review work related to transient imaging, which we realize in the following chapter using CIS sensors. Recalling Section 2.2.3, global light transport is not an instantaneous process but has a temporal dimension. Transient imaging is a new imaging modality capturing this temporal dimension. To illustrate this, imagine an illumination source emitting a short pulse of light. Having a camera with ultra-high temporal resolution, these pulses could be observed “in flight” as they traverse a scene and before the light distribution achieves a global equilibrium. A *transient image* is then the rapid sequence of frames capturing this scene response, hence representing the *impulse response of a scene*. As light transport is a linear time invariant system that is fully described by its impulse response, such a transient image fully describes the light transport in the scene. This illustrates the potential of this new image modality.

The original idea behind transient imaging goes back to work performed in the late 1970s by Abramson [142, 143] under the name “light-in-flight recording.” Abramson created holographic recordings of scenes illuminated by picosecond lasers, from which it was possible to optically reconstruct an image of the wave

front at a specific time. While the scene complexity was limited by technical constraints of the holographic setup, other researchers already used this approach for tasks such as shape measurements (e.g. [144]).

### **2.7.1 Impulse vs. Continuous Wave Transient Imaging**

Recently, interest in transient imaging has been rekindled by the development of femtosecond and picosecond lasers, allowing for ultra-short impulse illumination, as well as ultra-fast camera technologies. Velten et al. were the first to combine both technologies [145] to capture transient images, which allows for a simplified setup compared to the holographic approach, as well as significantly more general scene geometries. Transient imaging has many exciting applications. Starting with the introduction of an image formation model [146] and the pilot experiments by Kirmani et al. [147], there have been several proposals to use transient images as a means of reconstructing 3D geometry that is not directly visible to either the camera or the light sources Pandharkar et al. [148], Velten et al. [149], to capture surface reflectance [150], to perform lens-less imaging [151] or simply to visualize light transport in complex environments to gain a better understanding of optical phenomena [152]. Wu et al. [153] recently proposed the use of transient images together with models of light/object interaction to factor the illumination into direct and indirect components.

Unfortunately, impulse transient imaging relies on expensive custom hardware, namely a femtosecond laser as a light source, and a streak camera for the image capture as in Velten et al.'s setup [145]. While capturing high-resolution transient images, these components amount to hundreds of thousands of dollars' worth of equipment that is bulky, extremely sensitive, difficult to operate, potentially dangerous to the eye, and slow. For example, a streak camera measures only a single scanline of a transient image in each measurement. To obtain a full transient image, it is, therefore, necessary to mechanically scan the scene. Due to the very limited amount of light in a femtosecond pulse, averaging of multiple measurements, complicated calibration and noise suppression algorithms are required to obtain good image quality. All in all, capture times of an hour or more have been reported for a single transient image [145].



### **2.7.2 Continuous Wave Transient Imaging**

In this dissertation, we address this problem in the following Chapter 3 by introducing a novel method using continuous wave illumination with an inexpensive widespread CMOS CIS plus computation for obtaining the transient image.

A number of recent works have built on our computational CIS-based transient imaging, of which many are described later in this dissertation. In Chapter 5, we present a CIS-based transient imager able to recover geometry and albedo outside the line of sight from second-order diffuse reflections, effectively turning walls into mirrors. Lin et al. [154] and [155] present different regularization approaches for the reconstruction of transient images from the CIS measurements. We analyze the sparsity of transient images in Chapter 4 and present a new physically motivated model in Chapter 4, Chapter 6. Gupta et al. [156], Freedman et al. [126] and Kadambi et al. [157] present approaches for resolving multi-path ambiguities by reconstructing (partly complete) transient images. All of these works rely on our image formation model. In Chapter 4 we present a new approach for imaging through scattering and turbid media based on this model. Finally, Jarabo et al. [158] have investigated effective rendering of transient images, which allows for the simulation of complex transient imaging scenarios.

## **2.8 Inverse Methods**

A large variety of optical measurement systems has been described in the previous sections, ranging from traditional color cameras to impulse- or correlation-based TOF cameras. For each system, the measurement process has been explained and formalized with a mathematical model. Relying on these models, this section reviews approaches for the inference of scene features that are non-trivially encoded in the measurements, i.e. hidden or latent features. Similar to the mammalian visual system, described in the introduction of this thesis, this inference is done by adding computation to the measurement process.

### 2.8.1 Inverse Problems

The measurement process can be generalized for a large class of optical imaging systems (including the ones covered in this thesis), that is

$$\mathbf{j} = M(\mathbf{i}), \quad (2.30)$$

where here  $\mathbf{i} \in \mathbb{R}^m$  is a vector of  $m$  latent scene features and  $\mathbf{j} \in \mathbb{R}^n$  is the measurement vector consisting of a set of  $n$  observations.  $M$  models the measurements  $\mathbf{j}$ , i.e. the *forward model*. The model is parametrized by  $\mathbf{i}$ . It is important to note, that this model  $M$  is *not* a function, but is in fact a random process that generates the observations as samples from the distribution describing the measurement process. As introduced at the very beginning of this thesis in Section 2.1, the inherent particle-nature of light causes statistical fluctuations in optical imaging systems.

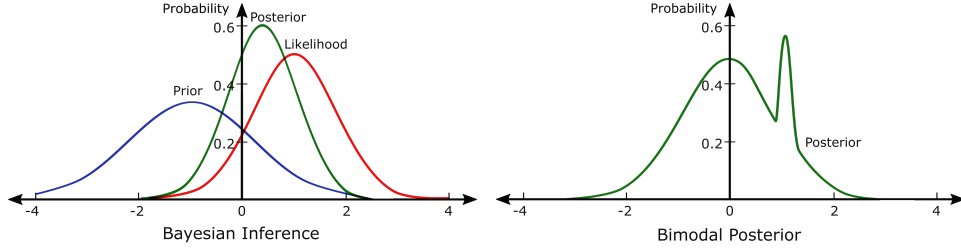
The inverse problem corresponding to Eq. (2.30) is

$$\tilde{\mathbf{i}} = M^{-1}(\mathbf{j}) \quad (2.31)$$

This means, given an observation  $\mathbf{j}$ , an inverse forward model  $M^{-1}$  is evaluated to recover an estimate  $\tilde{\mathbf{i}}$  of the hidden parameter vector  $\mathbf{i}$ . A large variety of different  $M^{-1}$  has been proposed, varying in the distance  $\tilde{\mathbf{i}} - \mathbf{i}$  depending on the properties of the forward model. Again, this is because  $M$  is not an invertible function, for which this distance could be made 0 by selecting its inverse function as the inverse forward model.

### 2.8.2 Statistical Image Recovery

To estimate the latent unknowns from the observations, we rely on Bayesian inference. By using Bayes' rule, Bayesian inference exploits both the observations and the prior knowledge on the unknowns. In contrast, Frequentist inference does not rely on priors but only on the measured observations. Both approaches have been used for inference in imaging systems, for example, traditional versus prior-based burst imaging discussed in [10]. However, in the remainder of this thesis, we show a wide variety of imaging problems that benefit from prior knowledge on the unknowns.



**Figure 2.13: Bayesian Inference.** The left plot illustrates the Bayesian approach to inference: The likelihood of observing a measurement is combined with prior knowledge on the latent variables. Using Bayes’ rule the combined belief is expressed as a posterior distribution. The right shows a more complex, bimodal, posterior.

Figure 2.13 illustrates Bayesian inference. Statistical variations in the forward model  $M$  are formalized in our approach using the following likelihood function

$$L(\mathbf{i}|\mathbf{j}) = p(\mathbf{j}|\mathbf{i}) \propto \exp(-G(\mathbf{j}, \mathbf{i})). \quad \text{Likelihood (2.32)}$$

The likelihood equals the sampling Probability Distribution Function (PDF)  $p(\mathbf{j}|\mathbf{i})$ , which describes the probability of an observed measurement given the latent variables  $\mathbf{i}$ . Without loss of generality, the sampling distribution is modeled here as an exponential on a distance function  $G : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$ . Intuitively,  $G$  assigns a distance between all possible measurements and unknowns, which is transformed into a PDF by the exponential. Prior knowledge on the unknowns can be encoded with

$$p(\mathbf{i}) \propto \exp(-F(\mathbf{i})), \quad \text{Prior (2.33)}$$

which assigns a probability to an arbitrary unknown (independent of any observations being made). Again an exponential model is used, in which intuitively  $F$  assigns a cost to less likely unknowns. Using Bayes’ rule the *posterior* probability  $p(\mathbf{i}|\mathbf{j})$  is

$$p(\mathbf{i}|\mathbf{j}) = \frac{p(\mathbf{j}|\mathbf{i})p(\mathbf{i})}{p(\mathbf{j})}. \quad \text{Bayes’ Rule (2.34)}$$

The posterior combines both, prior knowledge and observed measurements, and hence describes *all* that can be inferred about the unknowns  $\mathbf{i}$  from the  $\mathbf{j}$  (assuming all available priors are used). However, computing the full posterior PDF is infeasible for most inverse problems in imaging, which require large-scale inference with millions of latent variables and observations. Therefore, usually point estimates of the posterior are used. A strong choice for a wide variety of imaging applications is the Maximum A Posteriori (MAP) estimate

$$\mathbf{i}_{MAP} = \underset{\mathbf{i}}{\operatorname{argmax}} p(\mathbf{i}|\mathbf{j}) = \underset{\mathbf{i}}{\operatorname{argmax}} p(\mathbf{j}|\mathbf{i})p(\mathbf{i}), \quad \text{MAP estimate (2.35)}$$

$$\mathbf{i}_{MMSE} = E(\mathbf{i}|\mathbf{j}) = \int p(\mathbf{j}|\mathbf{i})\mathbf{x} \, d\mathbf{i}. \quad \text{MMSE estimate (2.36)}$$

MAP estimates the unknowns for the point where the posterior reaches its maximum probability. While this is a good strategy for unimodal posterior distributions, it can lead to undesired estimates for multimodal posteriors. Figure 2.13 shows a bimodal posterior on the right, which contains a large probability mass in the mode without the maximum. Here, other estimates, such as the Minimum Mean Square Error (MMSE) estimate in Eq. (2.36), can be beneficial. Given a point estimate, Bayesian inference can also provide measures of the uncertainty in this estimate, such as the variance or confidence intervals of the posterior. We refer the reader to [159] for a detailed discussion of Bayesian inference.

MAP estimation has been proven to be a successful approach for many imaging applications. In Chapter 9, we demonstrate that even classic image processing tasks (such as demosaicking), which traditionally are approached with specialized hand-tuned pipelines, significantly benefit from Bayesian MAP estimation. The MAP problem from Eq. (2.35) can be reformulated as

$$\begin{aligned} \mathbf{i}_{MAP} &= \underset{\mathbf{i}}{\operatorname{argmax}} p(\mathbf{i}|\mathbf{j}) \\ &= \underset{\mathbf{i}}{\operatorname{argmin}} -\log(p(\mathbf{j}|\mathbf{i})p(\mathbf{i})) \\ &= \underset{\mathbf{i}}{\operatorname{argmin}} -\log(p(\mathbf{j}|\mathbf{i})) - \log(p(\mathbf{i})) \\ &= \underset{\mathbf{i}}{\operatorname{argmin}} \underbrace{G(\mathbf{j}, \mathbf{i})}_{\text{Data Fidelity}} + \underbrace{F(\mathbf{i})}_{\text{Regularizer}}. \end{aligned} \quad (2.37)$$

The logarithmic transform in the second row can be applied since the logarithm is a monotonic function. In the last row, we have used Eq. (2.32) and Eq. (2.33). The final objective function is a sum of two terms: a *Data Fidelity* term originating from the likelihood, and a *Regularizer* term from the prior distribution. All the deterministic components of the data fidelity term, that is the ones independent of noise, are called *image formation model*. In contrast, the forward model  $M$  from Section 2.8.1 includes noise. Literature in the field often uses the term “prior” interchangeably with the term ”regularizer.” Finally, Eq. (2.38) shows an example following the Bayesian approach described in the previous paragraphs.

$$\begin{aligned} L(\mathbf{i}|\mathbf{j}) &\propto \exp\left(-\frac{\|\mathbf{j} - \mathbf{B}\mathbf{i}\|_2^2}{2\sigma^2}\right) \quad \text{and} \quad p(\mathbf{i}) \propto \exp(-\|\mathbf{i}\|_1) \\ \Rightarrow G(\mathbf{x}) &= \frac{1}{2\sigma^2}\|\mathbf{j} - \mathbf{B}\mathbf{x}\|_2^2 \quad \text{and} \quad F(\mathbf{x}) = \|\mathbf{x}\|_1 \end{aligned} \quad (2.38)$$

The likelihood models the blurred observations  $\mathbf{j}$  using the convolutional model from Eq. (2.12) in Section 2.3.3.  $\mathbf{B}$  is here the convolution matrix corresponding to a convolution with kernel  $B$ . The observations follow a normal distribution with a standard deviation of  $\sigma$ . Hence, the likelihood models Gaussian noise with the same standard deviation. The prior places a Laplacian distribution on the unknowns  $\mathbf{i}$ . Intuitively, this choice corresponds to the prior knowledge of the unknowns being sparsely distributed, e.g. astronomy images. The second row of Eq. (2.38) shows the corresponding  $\ell_2$ -norm data fidelity and  $\ell_1$ -norm regularizer. The image formation model as defined above is here  $\mathbf{B}$ .

### 2.8.3 Linear Models

A large class of inverse problems in imaging can be formulated by using linear image formation models and priors on linear transforms. Eq. (2.39) shows the modified likelihood and prior for this linear model.

$$\begin{aligned} L(\mathbf{i}|\mathbf{j}) &\propto \exp(-G(\Phi\mathbf{i})) \\ p(\mathbf{i}) &\propto \exp(-F(\Omega\mathbf{i})) \end{aligned} \quad (2.39)$$

Many inverse problems in imaging are accurately modeled using a linear image formation  $\Phi$ . Remember that both light transport and conventional image sensors are linear; see Section 2.2 and Section 2.4. Problems with linear image formation include classical image processing problems, such as demosaicking [10], deconvolution [160], [161], inpainting [162], and denoising [163]. Furthermore, the remainder of this thesis presents inverse problems for traditional and novel computational camera designs, that all can be modeled using a linear image formation. A large number of recent reconstruction approaches rely on image priors that are formulated on the linear transforms  $\Omega\mathbf{i}$ . This includes Total Variation (TV) regularizers [164], hyperlaplacian gradient priors [160], [165], mixture model priors [163], patch-based shrinkage [166], and convolutional priors [167].

Given the general linear reconstruction model from Eq. (2.39), we follow the Bayesian approach described in the previous chapter yielding the MAP estimate

$$\mathbf{i}_{MAP} = \underset{\mathbf{i}}{\operatorname{argmin}} G(\Phi\mathbf{i}) + F(\Omega\mathbf{i}). \quad (2.40)$$

### 2.8.4 Optimization Methods

Over the past years, numerical optimization has become a standard tool for solving a number of classical restoration and reconstruction problems in computational photography. Examples include the traditional image processing problems listed above as well as image editing problems such as tone mapping [168], Poisson-blending [169] and colorization [170]. Very efficient solvers have been developed for most of these problems [167, 171]. Optimization techniques are also becoming increasingly popular solutions for scientific imaging problems such as X-ray tomography [172] and phase retrieval [173].

The literature on algorithms for solving image optimization problems is extensive. A particularly fruitful line of research has focused on solving convex optimization problems with the structure of the MAP problem above from Eq. (2.40), that is problems where  $F$  and  $G$  are convex. The main challenge in solving Eq. (2.40) is that the objective has to be jointly minimized over the sum of the data and regularizer term. Splitting methods avoid solving this challenging problem by turning the joint problem into a sequence of simpler optimization problems, alternately involving

only one of the two terms. This is achieved by introducing a slack variable  $\mathbf{y}$  giving

$$\begin{aligned} \mathbf{i}_{MAP} &= \underset{\mathbf{i}}{\operatorname{argmin}} G(\Phi\mathbf{i}) + F(\Omega\mathbf{i}) \\ &= \underset{\mathbf{i}}{\operatorname{argmin}} G(\Phi\mathbf{i}) + F(\mathbf{y}) \quad \text{s.t.} \quad \mathbf{y} = \Omega\mathbf{i}. \end{aligned} \quad (2.41)$$

It is easy to see that this problem is equivalent to Eq. (2.40) by substituting the constraint. A prominent splitting method is the Alternating Direction Method of Multipliers (ADMM) [174]. This method finds a saddle point of an augmented version of the classical Lagrangian for the constrained problem from the second row of Eq (2.41). The augmented Lagrangian is

$$\mathcal{L}_\rho(\mathbf{i}, \mathbf{y}, \lambda) = G(\Phi\mathbf{i}) + F(\mathbf{y}) + \lambda^T(\Omega\mathbf{i} - \mathbf{y}) + \frac{\rho}{2}\|\Omega\mathbf{i} - \mathbf{y}\|_2^2, \quad (2.42)$$

where  $\lambda$  is here the Lagrange multiplier associated with the constraint and  $\rho \in \mathbb{R}^+$  is a penalty for the quadratic term added to the classic Lagrangian. See [174] for a detailed discussion of this augmented version. ADMM can then be formulated as the following algorithm.

---

**Algorithm 1** ADMM for Eq. (2.41)

---

- 1: **for**  $k = 1$  to  $V$  **do**
  - 2:    $\mathbf{i}^{k+1} = \underset{\mathbf{i}}{\operatorname{argmin}} \mathcal{L}_\rho(\mathbf{i}, \mathbf{y}^k, \lambda^k)$
  - 3:    $\mathbf{y}^{k+1} = \underset{\mathbf{y}}{\operatorname{argmin}} \mathcal{L}_\rho(\mathbf{i}^{k+1}, \mathbf{y}, \lambda^k)$
  - 4:    $\lambda^{k+1} = \lambda^k + \rho(\Omega\mathbf{i}^{k+1} - \mathbf{y}^{k+1})$
  - 5: **end for**
- 

In the first two steps, the method minimized the augmented Lagrangian alternately with regard to  $\mathbf{i}$  and  $\mathbf{y}$  while fixing the other variables. The last step performs a gradient ascent of the Lagrange multiplier. The Lagrange multiplier accumulates here the (scaled) consensus error. A powerful tool for the description, implementation and analysis of splitting algorithms is given by *proximal operators*. The proximal operator of a function  $f$  is defined as

$$\mathbf{prox}_{\tau f}(\mathbf{v}) = \underset{\mathbf{x}}{\operatorname{argmin}} \left( f(\mathbf{x}) + \frac{1}{2\tau}\|\mathbf{x} - \mathbf{v}\|_2^2 \right), \quad \begin{array}{l} \text{Proximal} \\ \text{Operator} \end{array} \quad (2.43)$$

where  $\tau > 0$ . The proximal operator optimizes over the function in isolation but incorporates a quadratic term that can be used to link the optimization with a broader algorithm. It can be interpreted as a gradient step for  $f$  when  $\tau$  is small and  $f$  is differentiable [175]. We can reformulate Algorithm 1 as the following Algorithm 2 using proximal operators.

---

**Algorithm 2** Proximal ADMM for Eq. (2.41)

---

- 1: **for**  $k = 1$  to  $V$  **do**
  - 2:    $\mathbf{i}^{k+1} = \underset{\mathbf{i}}{\operatorname{argmin}} G(\Phi\mathbf{i}) + \frac{1}{2\rho} \|\Omega\mathbf{i} - \mathbf{y}^k - \tilde{\lambda}^k\|_2^2$
  - 3:    $\mathbf{y}^{k+1} = \underset{\mathbf{y}}{\operatorname{argmin}} F(\mathbf{y}) + \frac{1}{2\rho} \|\Omega\mathbf{i}^{k+1} - \mathbf{y} - \tilde{\lambda}^k\|_2^2 = \mathbf{prox}_{\frac{F}{\rho}}(\Omega\mathbf{i}^{k+1} + \tilde{\lambda}^k)$
  - 4:    $\tilde{\lambda}^{k+1} = \tilde{\lambda}^k + \Omega\mathbf{i}^{k+1} - \mathbf{y}^{k+1}$
  - 5: **end for**
- 

Here, we use the scaled Lagrange multiplier  $\tilde{\lambda} = \lambda/\rho$  as a notational shortcut. We can now also understand why *proximal algorithms* like ADMM provide an efficient way to solve the considered linear MAP problems. If  $G$  is quadratic or separable, the first row becomes computationally cheap. If now the function  $F$  is separable, then also its proximal operators parallelize and yield very efficient distributed solver methods. A number of proximal algorithms have been explored; see [175]. Besides ADMM, prominent examples include the proximal point algorithm [176], forward-backward splitting [177], the Pock-Chambolle algorithm [178, 179], the split Bregman method [180], ISTA and FISTA [181], and half-quadratic splitting [182]. Recent work has applied these methods and new ones such as iPiano [183] to nonconvex optimization problems and found conditions that guarantee convergence (though not necessarily to the global optimum); see, e.g., [184–186].



## Chapter 3

# Transient Imaging

In this chapter, we demonstrate transient imaging using inexpensive CIS TOF cameras. It is shown that CIS measurements encode transient images in convolutional structure. We derive a convolutional model for the local light/object interaction. Using a sequence of differently modulated CIS images, an inverse problem can be solved to infer the latent transient image from these measurements. The resulting method produces transient images at a cost several orders of magnitude below existing impulse-based methods, listed in Section 2.7, while simultaneously simplifying and drastically speeding up the capture process.

### 3.1 Introduction

To enable fast, simple, and inexpensive transient imaging, we replace direct, impulse-based, acquisition methods with a CIS TOF camera plus computation. In particular, we attempt to substitute the complex femtosecond laser and streak camera setup discussed previously in Section 2.7.

In the following, we describe and demonstrate the approach that makes this possible. We derive a convolutional model for the relationship between transient imaging and traditional time-of-flight imaging using Photonic Mixer Devices. The estimation of transient images from CIS measurements is then formulated as an inverse problem. This problem is ill-posed but can be solved in a MAP approach, following Section 2.8, by introducing regularization terms as well as a model of



**Figure 3.1:** Left: Our capture setup for transient images (from left: computer, signal generator, power supply, modulated light source, CIS camera). Middle: A disco ball with many mirrored facets. Right: The same sphere as seen by our transient imager when illuminated from the left, colored according to the time offset of the main intensity peak.

local surface/light interactions. Finally, we demonstrate a prototype system with a modified PMD camera that allows for flexible measurements of Time-of-Flight images using a range of different modulation frequencies and phases. The system is inexpensive, portable, eye-safe, insensitive to background light, and acquires data at a much higher rate than streak cameras, enabling transient imaging even outside lab environments.

## 3.2 Transient CIS Image Formation

### 3.2.1 Generalized CIS Model

In the related work Section 2.6, we have introduced an idealized model using perfectly sinusoidal modulation in the CIS image formation. While this approximate model builds intuition and is sufficiently accurate for many applications, in general only the following assumptions on  $g, s$  have to be made. The sensor modulation  $f_\omega$  is a zero-mean periodic function with period  $T = 2\pi/\omega$ , that is

$$f_\omega(t + k \cdot T) = f_\omega(t) \quad \text{and} \quad \int_0^T f_\omega(t) dt = 0. \quad (3.1)$$

For an arbitrary incident irradiance  $s$ , the CIS measures the following *modulated exposure*

$$b_{\omega, \phi} = \int_0^{NT} s(t) f_\omega(t + \phi) dt, \quad (3.2)$$

where  $\phi$  is a programmable temporal offset. This is a temporally *convolutional* model. Recalling Section 2.6, CIS sensors integrate over a large number of  $N$  periods (ca.  $10^4$ – $10^5$ ), hence, they essentially correlate  $s$  and  $f$ . In contrast to impulse-based systems, this allows extended exposures, much like regular cameras.

### 3.2.2 Single Reflection

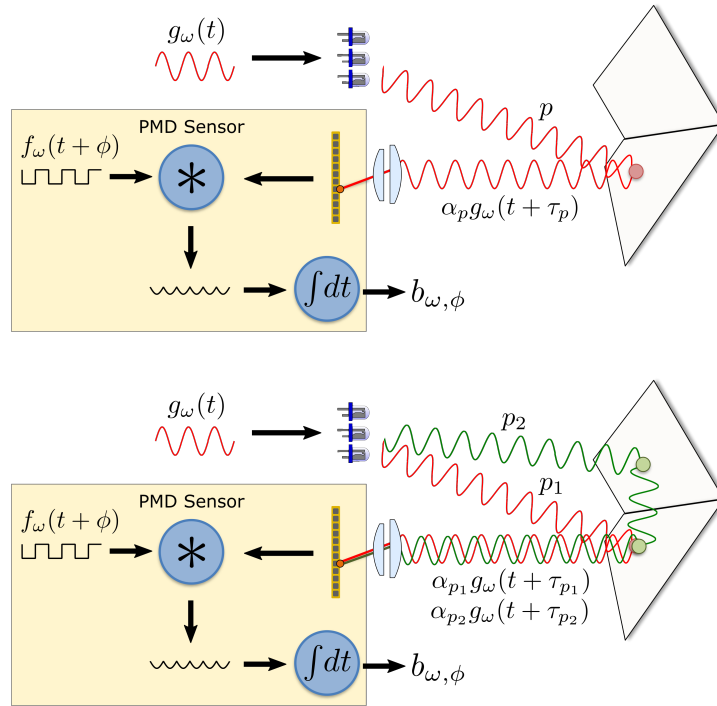
To illustrate the general model, we first consider the special case of a single reflector in the scene. Recalling Section 2.6, depth TOF imaging commonly assumes the *direct illumination* of a *single object point* by a *single point light source*, effectively meaning that only a single light path contributes to the sensor reading. This is illustrated on the top of Figure 3.2. Under this assumption we obtain the incident signal  $s_\omega$  from Eq.(3.3).

$$s_\omega(t) = I + \alpha a \cdot g_\omega(t + \tau), \quad (3.3)$$

with  $I$  being the DC component of the light source plus any ambient illumination,  $a$  is the modulation amplitude for the light source,  $\alpha$  is an attenuation term due to surface reflectance and distance-based intensity falloff, and  $\tau$  is the TOF from the light source to the object point and then to the CIS pixel. The modulated exposure measured by the sensor becomes then

$$\begin{aligned} b_{\omega,\phi} &= \int_0^{NT} (I + \alpha a g_\omega(t + \tau)) f_\omega(t + \phi) dt \\ &= I \cdot N \underbrace{\int_0^T f_\omega(t + \phi) dt}_{=0} + \alpha a \cdot N \underbrace{\int_0^T g_\omega(t + \tau) f_\omega(t + \phi) dt}_{=c_{\omega,\phi}(\tau)}. \end{aligned} \quad (3.4)$$

The correlation coefficients  $c_{\omega,\phi}(\tau)$  can either be determined analytically for specific  $f_\omega, g_\omega$  such as sinusoids, or they can be calibrated using objects at different known distances  $d$  using the relationship  $\tau = \omega/c \cdot d$ , where  $c$  is here the speed of light. Relating the general model back to range imaging, two measurements  $b_{\omega,0^\circ}$  and  $b_{\omega,90^\circ}$  are obtained for per pixel range estimation. Using the known  $c_{\omega,\phi}(\tau)$ , it is then possible to solve for the pixel intensity  $\alpha a$  and the distance  $d$  of the object visible at that pixel. In general, measurements  $b_{\omega_i,\phi_i}$  for many different modulation



**Figure 3.2:** Top: Operating principle of a conventional CIS PMD sensor. Light from a modulated light source arrives at a pixel sensor via a single light path with time delay  $\tau$ . The PMD sensor modulates the incident light with a reference signal  $f_\omega$  and integrates the resulting modulated exposure to obtain a distance-dependent correlation. Bottom: in the presence of global illumination, the incident illumination is a superposition of light with different phase shifts. Multiple measurements at different phases and frequencies are taken to analyze such light interactions.

frequencies and phases  $(\omega_i, \phi_i)$  can be acquired. This yields a collection of different, travel-time dependent correlation coefficients  $c_{\omega_i, \phi_i}(\tau)$ .

### 3.2.3 Global Illumination

Using a multitude of measurements over a wide frequency range, we can relax the requirement that light only arrives at the sensor via single light path, and move to a full global illumination model instead. Temporally-varying global light transport has been reviewed in Section 2.2.3 and is illustrated in Figure 3.2 on the bottom.

Integrating over all light paths in the scene, Eq. (3.4) then becomes

$$\begin{aligned}
b_{\omega_i, \phi_i} &= \int_0^\infty \int_{\mathcal{P}} \int_0^{NT} (I + \delta(|p| - \tau) \alpha_p \cdot a g_{\omega_i}(t + \tau)) f_{\omega}(t + \phi_i) dt dp d\tau \\
&= \int_0^\infty \int_0^{NT} (I + i(\tau) \cdot a g_{\omega_i}(t + \tau)) f_{\omega}(t + \phi_i) dt d\tau
\end{aligned} \tag{3.5}$$

We integrate now the modulated light along all ray paths, connecting the light source and the sensor pixel,  $\mathcal{P}$  is the space of all such ray paths. The factor  $\alpha_p$  is the light attenuation along such a path  $p$ , and  $|p|$  is the travel time along path  $p$ . Following Section 2.2.3, the delta-function makes sure that the path contributions with the same travel time  $\tau$  are integrated for a certain time offset  $\tau$ . In the second row of Eq.(3.5), we encapsulate all path components of the same length in  $i$ . As explained in the related work in Section 2.2.3, this is a pixel of the transient image.

$$i(\tau) = \int_{\mathcal{P}} \delta(|p| - \tau) \alpha_p dp \quad \text{Transient Pixel (3.6)}$$

$$I(x, y, \tau) = i_{x,y}(\tau) \quad \text{Transient Image (3.7)}$$

Note that above we have considered the measurement  $b_{\omega_i, \phi_i}$  for a single pixel. A regular grid of transient pixels is a *transient image*, hence it is three-dimensional, with two spatial and one time coordinate. We can see from Eq. (3.6), that the transient image is the impulse response of a scene, i.e. the intensity profile as a function of time when the scene is illuminated by a very short pulse of light. Recognizing that  $i$  does not depend on  $t$  in Eq. (3.5), we can reformulate it as Eq. (3.8) using the correlation coefficients.

$$\begin{aligned}
b_{\omega_i, \phi_i} &= \int_0^\infty \int_0^{NT} \underbrace{i(\tau)}_{\text{Transient Pixel}} \underbrace{a g_{\omega_i}(t + \tau)}_{\text{Light Modulation}} \underbrace{f_{\omega}(t + \phi_i)}_{\text{Sensor Modulation}} dt d\tau \\
&= a \int_0^\infty i(\tau) \underbrace{N \int_0^T g_{\omega_i}(t + \tau) f_{\omega}(t + \phi_i) dt d\tau}_{=c_{\omega_i, \phi_i}(\tau)} dt d\tau
\end{aligned} \tag{3.8}$$

Hence, the convolutional model allows for separating the transient component from the correlation on the CIS sensor. Discretizing  $\tau$  in  $m$  temporal bins  $\{\tau_1, \dots, \tau_m\}$  to the following linear model

$$\mathbf{b} = \mathbf{C}\mathbf{i} + \mathbf{n} \quad \text{with} \quad \mathbf{C}_{i,j} = c_{\omega_i, \phi_i}(\tau_j) \quad (3.9)$$

The vectorized transient image is  $\mathbf{i} \in \mathbb{R}^m$ , with the components stacked. The measurement  $\mathbf{b} \in \mathbb{R}^n$  denotes the vector of all  $n$  modulated exposures observed at all pixels in the image. The correlation coefficients  $c_{\omega_j, \phi_j}$  are arranged in a matrix  $\mathbf{C}$  that maps a transient image to the vector of observations  $\mathbf{b}$ . Finally,  $\mathbf{n}$  models additive noise, which we have discussed in Section 2.6.4.

### 3.3 Transient Imaging using CIS

This section describes how we estimate a transient image from the sensor measurements  $\mathbf{b}$ . We first formulate the corresponding linear inverse problem and then present a method to solve this optimization problem.

#### 3.3.1 Inverse Problem

The model from Eq. (3.9) contains a linear transform  $\mathbf{C}$  which is not well conditioned and additive noise  $\mathbf{n}$ . Approximating  $g$  and  $f$  with sinusoidal waveforms as in Section 2.5,  $\mathbf{C}$  becomes a truncated Fourier transform. The maximum frequency is limited by the sensor and light source and usually around 100 MHz [85]. This means the correlation coefficients vary slowly with distance for the range of feasible modulation frequencies and scene scales (Section Section 3.4). Due to the ill-conditioned  $\mathbf{C}$  and the noise, the inverse problem of recovering  $\mathbf{i}$  from  $\mathbf{b}$  is ill-posed, hence directly solving for  $\mathbf{i}$  fails.

The challenging inverse problem is solved following the MAP approach introduced in Section 2.8. We use spatial and temporal priors, and a likelihood approximating the noise distribution. In particular, two types of priors are used: a prior on the gradients, leading to a convex optimization problem which can be solved to high accuracy, and a Gaussian-Exponential prior leading to an objective function with local minima. Specifically, the spatial and temporal gradients are

assumed to follow a heavy-tailed distribution. In other words, we assume that the temporal and spatial profiles are sparse. The Gaussian-Exponential prior assumes that a transient pixel can be expressed using a model  $m(\cdot)$  of the *local* interaction of light with objects. Approximating the  $\mathbf{n}$  as Gaussian noise, the resulting MAP estimation problem from Section 2.8 becomes

$$\begin{aligned} \mathbf{i}_{\text{opt}} = \underset{\mathbf{i}}{\operatorname{argmin}} \quad & \frac{1}{2} \|\mathbf{C}\mathbf{i} - \mathbf{b}\|_2^2 + \lambda \sum_{\mathbf{x}} \|\nabla_z \mathbf{i}_{\mathbf{x}}\|_H + \theta \sum_{\tau} \|\nabla_{\mathbf{x}} \mathbf{i}_{\tau}\|_H \\ \text{s.t. } \quad & \mathbf{i} \in m(\mathcal{C}) \end{aligned} \quad (3.10)$$

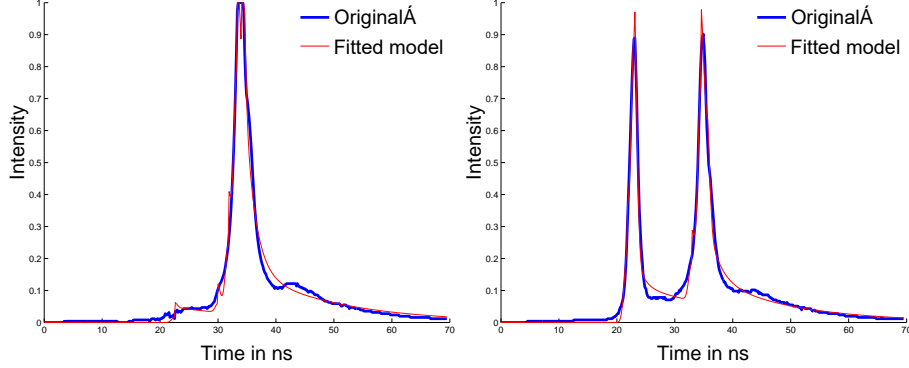
The first term is the Data Fidelity, which originates from the Gaussian likelihood.

*Gradient Priors* The other two terms in the first line respectively represent gradient regularizations along the temporal direction for each individual pixel, and along the spatial dimensions for each individual time slice. We expect the gradients for both the spatial and temporal slices to be sparse, but with occasional outliers (discontinuities). In our work we chose the Huber penalty function [187]. The Huber penalty  $\|\cdot\|_H$  is an  $\ell_1/\ell_2$  hybrid error measure and defined as

$$\|\mathbf{x}\|_H = \sum_k h_{\epsilon}(\mathbf{x}_k) \quad \text{with} \quad h_{\epsilon}(x) = \begin{cases} |x| - \frac{\epsilon}{2} & \text{if } |x| > \epsilon \\ \frac{|x|^2}{2\epsilon} & \text{else} \end{cases} \quad (3.11)$$

Huber fitting is commonly used in robust statistics since it allows for outliers in the common Gaussian models [174]. The parameter  $\epsilon > 0$  defines the trade-off between the Gaussian fit (quadratic) for small values and a Laplacian fit ( $\ell_1$ -norm). In Eq. (3.10), spatial and temporal gradients are Huber-penalized. In contrast to the Laplacian gradient penalty, the Huber gradient-penalty does not introduce non-smooth discontinuities in smooth regions. If we would allow  $\epsilon = 0$ , the spatial  $\ell_1$  gradient-penalty would become the well-known TV penalty [188], which contains such discontinuities as staircasing [178].

*Gaussian-Exponential Priors* The Gaussian-Exponential prior leads to the constraint in the MAP problem from Eq. (3.10). Here,  $m(\mathcal{C})$  is a notational shortcut for the image of our model function  $m(\cdot)$  that expresses the transient pixel profiles as a



**Figure 3.3:** Two different time profiles of the model  $m_{\mathbf{x}}(\cdot)$  fitted to ground-truth measurements, which were acquired by Velten et al. using a femtosecond laser and streak camera [152].

mixture of exponentials and Gaussians. This model is inspired by recent work by Wu et al. [189] and applies to each pixel independently. Specifically, we model the light interaction at each local surface point as a combination of surface reflection and subsurface scattering. For surface reflection, the temporal impulse response to a short pulse of light is a Dirac peak, which we represent as a Gaussian  $G_{\sigma}$ , where  $\sigma$  is related to the temporal resolution of the acquisition system. For subsurface scattering, the temporal impulse response can be modeled as an exponential decay  $E$  [189]. In a complex scene with global illumination, the model  $m_{\mathbf{x}}(\cdot)$  for the time profile at pixel  $\mathbf{x}$  can therefore be expressed as a superposition of  $K$  Gaussians and exponentials, that is

$$m_{\mathbf{x}}(\mathbf{u}) = \sum_k^{K_{\mathbf{x}}} \mathbf{r}_{\mathbf{x},k} G_{\sigma}(\tau - \mathbf{p}_{\mathbf{x},k}) + \mathbf{a}_{\mathbf{x},k} E(\mathbf{d}_{\mathbf{x},k}, \tau - \mathbf{p}_{\mathbf{x},k}), \quad (3.12)$$

where the model parameters  $\mathbf{u}$  are the set of all (temporal) positions  $\mathbf{p}_{\mathbf{x},k}$  for the Gaussians and exponentials, combined with their amplitudes  $\mathbf{r}_{\mathbf{x},k}$  and  $\mathbf{a}_{\mathbf{x},k}$ , and finally the exponential decay parameters  $\mathbf{d}_{\mathbf{x},k}$  which depend on the density of the scattering medium.

$$\mathbf{u} = \bigcup_{\mathbf{x},k} \{\mathbf{r}_{\mathbf{x},k}, \mathbf{p}_{\mathbf{x},k}, \mathbf{a}_{\mathbf{x},k}, \mathbf{d}_{\mathbf{x},k}\}.$$



To illustrate the model we have fitted ground-truth traces from a streak camera setup. For this experiment, we used the “tomato” transient image that Velten et al. obtained using a femtosecond laser and streak camera [152]. Figure 3.3 shows the fitted transient profiles for different pixel locations.

### 3.3.2 Optimization via Splitting

The objective from Eq. (3.10) is non-convex due to the constraint, which contains the non-convex set  $m(\mathcal{C})$ . The constraint even contains integer variables ( $K_x$ ). In contrast, note that all penalties from the first line are convex. To solve this problem, we introduce a variable  $\mathbf{u}$  and a new consensus constraint  $m(\mathbf{u}) = \mathbf{i}$ .

$$\begin{aligned} \mathbf{i}_{\text{opt}}, \mathbf{u}_{\text{opt}} = \underset{\mathbf{i}, \mathbf{u}}{\operatorname{argmin}} \quad & \frac{1}{2} \|\mathbf{C}\mathbf{i} - \mathbf{b}\|_2^2 + \lambda \sum_{\mathbf{x}} \|\nabla_z \mathbf{i}_{\mathbf{x}}\|_H + \theta \sum_{\tau} \|\nabla_{\mathbf{x}} \mathbf{i}_{\tau}\|_H, \\ \text{s.t.} \quad & m(\mathbf{u}) = \mathbf{i} \end{aligned} \quad (3.13)$$

with the consensus constraint this problem is equivalent to the one from Eq. (3.10). We solve this problem by first eliminating the constraint as an additional penalty term, and then perform coordinate descent w.r.t.  $\mathbf{i}$  and  $\mathbf{u}$ . See Section A.3 for a detailed discussion. The coordinate descent splits the hard joint problem into a sequence of simpler problems only involving  $\mathbf{i}$  and  $\mathbf{u}$ . The  $\mathbf{i}$ -subproblem is a convex problem since it does no longer involve the model. The  $\mathbf{u}$ -subproblem is separable per pixel. Finally, the integer variable can be iteratively estimated as part of the coordinate descent.

Summarizing, the Gaussian-exponential prior makes the optimization problem hard and costly to solve. The described coordinate-descent has weak convergence guarantees and is not guaranteed to find a global minimum. Due to the many parameters in  $\mathcal{C}$  the  $\mathbf{u}$ -subproblem is expensive. In Chapter 4 we will introduce a modified model that exploits the convolutional structure of transient images. By doing so we eliminate all of the discussed issues and combine the  $\mathbf{i}$ -subproblem with the model. Hence, in the following, we will cover the  $\mathbf{i}$ -subproblem, while the  $\mathbf{u}$ -subproblem is discussed in Section A.3. Furthermore, note that omitting the Gaussian-exponential prior also results in an optimization of the form of  $\mathbf{i}$ -subproblem.

### 3.3.3 Solving the i-subproblem

The i-subproblem is given in Eq. (3.14).

$$\begin{aligned} \mathbf{i}_{\text{opt}} = \underset{\mathbf{i}}{\operatorname{argmin}} \quad & \frac{1}{2} \|\mathbf{C}\mathbf{i} - \mathbf{b}\|_2^2 + \\ & \lambda \sum_{\mathbf{x}} \|\nabla_z \mathbf{i}_{\mathbf{x}}\|_H + \theta \sum_{\tau} \|\nabla_{\mathbf{x}} \mathbf{i}_{\tau}\|_H + \\ & \frac{\rho}{2} \|\mathbf{i} - \mathbf{m}\|_2^2 \end{aligned} \quad (3.14)$$

The last term is here a quadratic coupling term. In our coordinate descent scheme it is  $\mathbf{m} = m(\mathbf{u})$  from the previous  $\mathbf{u}$ -subproblem; see Section A.3. Eq. (3.14) is a convex problem of the form discussed in Section 2.8.3. In particular, we choose

$$\begin{aligned} \Phi &= \mathbb{I}, & G(\mathbf{v}) &= \frac{1}{2} \|\mathbf{C}\mathbf{v} - \mathbf{b}\|_2^2 + \frac{\rho}{2} \|\mathbf{v} - \mathbf{m}\|_2^2, \\ \Omega &= \mathbf{S}, & F(\mathbf{v}) &= \|\mathbf{v}\|_H, \end{aligned} \quad (3.15)$$

where here the matrix  $\mathbf{S}$  applies the temporal derivative operator to all pixel time sequences and the spatial gradient operators to all time slices (according to Eq. 3.14). Since we can express our problem in the form of Eq. (2.40), it can be solved with a proximal algorithm as discussed in Section 2.8.4. We use the first-order primal-dual framework by Chambolle and Pock [178], which is equivalent to a linearized version of ADMM [175]. We discuss various proximal algorithms, including the Chambolle-Pock method, in Chapter 10. Note that different choices of the algorithm and the functions in Eq. (3.15) can be made. Chapter 10 also presents method to automatically compare amongst different algorithms and problem formulations to yield an optimal solver method.

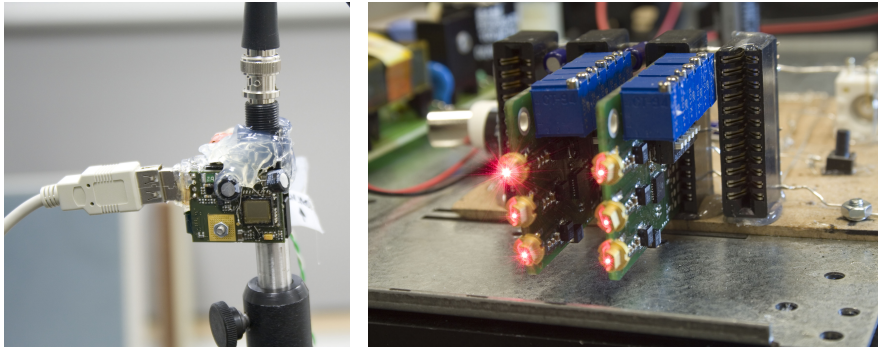
To solve the problem with the Chambolle-Pock method, the proximal operators for  $F^*$  and  $G$  have to be known. These are given by

$$\begin{aligned} \mathbf{prox}_{\tau F^*}(\mathbf{v}) &= \frac{\mathbf{v}}{\max\left(1, \left|\frac{\mathbf{v}}{1+\tau\epsilon}\right|\right)} \\ \mathbf{prox}_{\tau G}(\mathbf{v}) &= [\tau \mathbf{C}^T \mathbf{C} + (1 + \tau)\mathbb{I}]^{-1} (\tau \mathbf{C}^T \mathbf{b} + \mathbf{v} + \tau \rho \mathbf{m}), \end{aligned} \quad (3.16)$$

The Huber proximal operator amounts here to a simple pointwise evaluation. The

second proximal operator requires the solution of a linear system. However, since  $C$  is of small dimensions, it can be solved to high accuracy using a direct inversion in parallel for each pixel. The inverse can be cached which makes this step computationally efficient in the iterative scheme.

### 3.4 Setup



**Figure 3.4:** The main components of our capture setup. Left: Modified PMD imager. Right: Light source with six uncollimated laser diodes.

We now describe the physical setup we developed to acquire transient images of real scenes.

The CIS camera is based on the PMDTechnologies CamBoard nano, an evaluation platform for the PMD PhotonICs 19k-S3 image sensor. We removed the on-board light source and the default 830 nm longpass filter. Since our technique is based on the use of a wide range of modulation frequencies, we intercepted the on-board modulation signal and replaced it with our own input from an external source, and added a trigger output that signals the start of the integration phase. We confirmed that the sensor can be operated at modulation frequencies up to 180 MHz, but managed to obtain stable results only up to 110 MHz.

Our light source is a bank of six 650 nm laser diodes with a total average output power of 2.4 W. Since the beams are not collimated, eye safety is not a concern at reasonable distances from the device. Two groups of three diodes are driven through a iC-Haus 6-channel constant current driver each (type iC-HG), with each diode occupying two driver outputs for DC bias and the high-frequency modulation.

Using a fast photodiode (Thorlabs FDS010) and a 500 MHz, 2 GS/s oscilloscope, we confirm that this setup can modulate the output intensity by a full 100% up to a frequency of 180 MHz.

As our modulation source signal, we use the DDS9m/02 signal generator from Novatech Instruments, which is based on the Analog Devices AD9958 direct digital synthesizer chip. We use the digital output of the signal generator to modulate both the light source and the camera with a square wave.

A microcontroller circuit sets modulation frequencies and phases on the synthesizer board. Reacting to the trigger signal from the camera, it switches the modulation signals for light source and camera in accordance with the currently set integration time (we take an exposure series from 120 to 1920  $\mu s$  in steps of one f-stop). The sensor data is read out through the API provided with the camera board.

*Measurement Routine.* In order to complete a full measurement over a frequency range from 10 to 120 MHz in steps of 0.5 MHz, our capture system takes about 90 seconds. Note that we operate the setup at a duty cycle of less than 1%, in order to avoid overheating of the critical components (signal buffer and laser driver ICs, laser diodes, PMD sensor) that are not equipped with heatsinks or active cooling. We therefore estimate that with proper thermal management, another significant speedup will be achieved, reducing the overall acquisition time to only a few seconds.

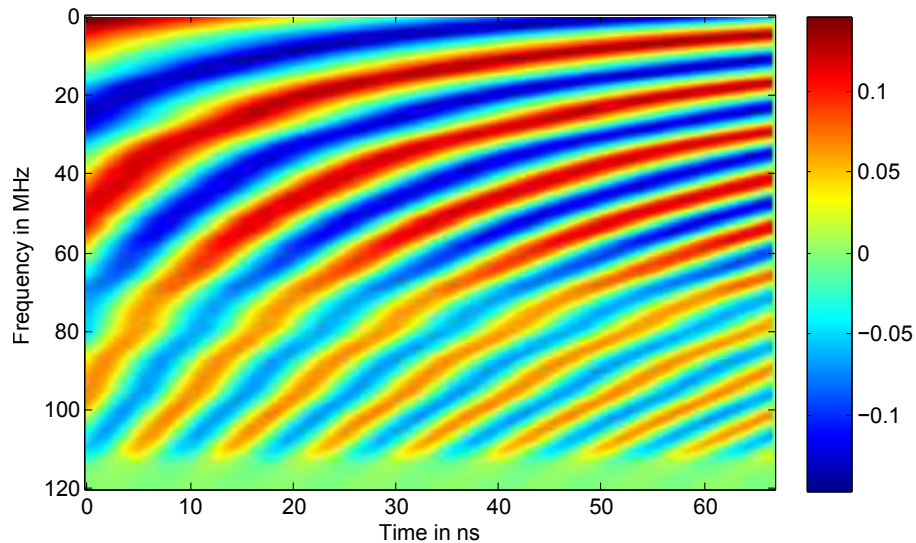
There are several ways our method deals with the dynamic range issues. The working principle of PMD sensors is itself very effective in suppressing ambient illumination, the sensor provides high-bit depth readouts (14 bits) and, finally, we do take an exposure sequence as described just above. That said, if very bright light paths and very dim light paths mix in the same pixel, the reconstruction quality of the dim paths will suffer.

*Calibration.* In order to obtain the correlation matrix  $\mathbf{C}$ , we perform a calibration step. We place camera and light source as close to each other as possible, and facing a diffuse white wall, with no other objects nearby that could scatter light into the line of sight. For each frequency, we sample the distance-dependent correlation

coefficients by varying the relative phase between the sensor and the light source modulation signals. This allows us to emulate different optical delays without mechanically moving parts.

The calibration measurements for frequencies from 10 to 120 MHz in 1 MHz steps and distances from 0 to 20 m in 0.1 m steps take about 6 hours to complete, with a further 30 minutes to extract the matrix from the raw sensor readings. We average the calibration matrices over a  $40 \times 40$  pixel region in order to reduce acquisition noise.

We note, however, that the calibrated matrix  $\mathbf{C}$  obtained in this fashion is also valid for different geometric configurations of sensor and camera, with only a change in the physical interpretation of the resulting reconstructed transient images. This means that the calibration step is essentially a one-time operation. Figure 3.5 shows a color visualization of the recovered correlation coefficients, with the vertical axis corresponding to different frequencies, and the horizontal axis corresponding to different path lengths, or travel times.



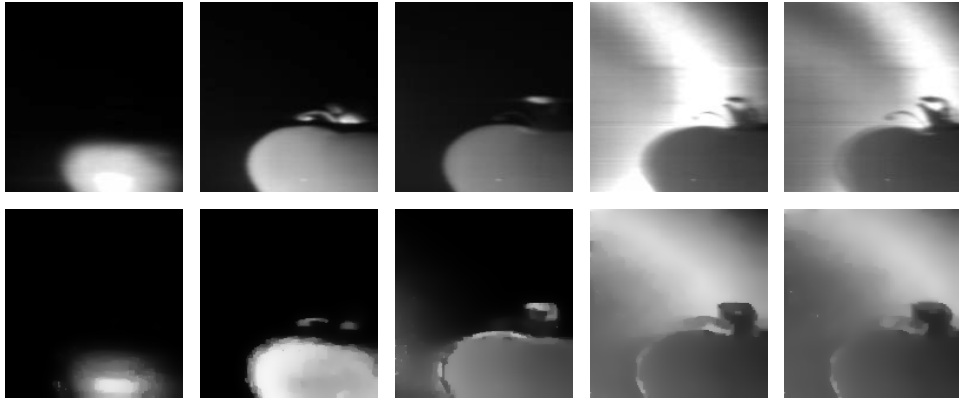
**Figure 3.5:** A visualization of the correlation coefficients for different path lengths (horizontal axis) and modulation frequencies (vertical axis).

## 3.5 Results

We evaluated our approach using both synthetic data for ground truth comparisons, and measurements using our custom setup.

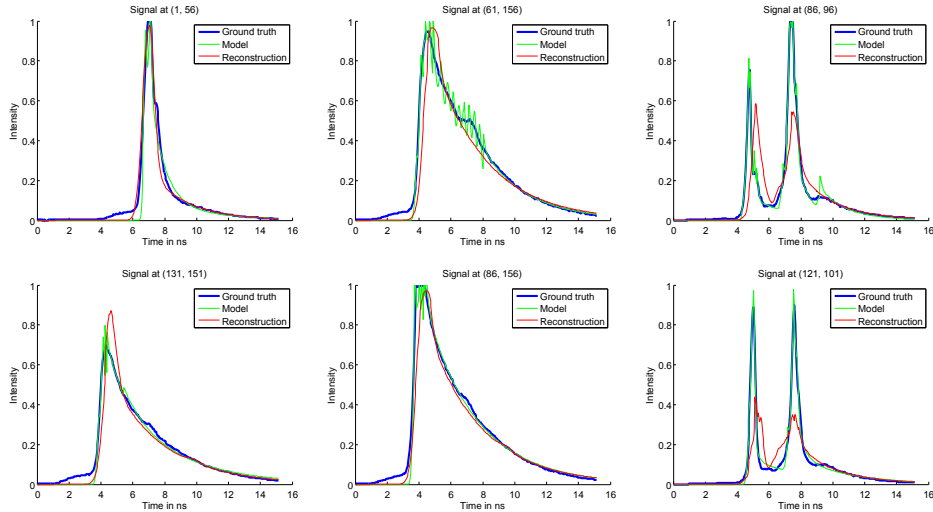
### 3.5.1 Synthetic Examples

For our synthetic experiments, we used a transient image that Velten et al. obtained using a femtosecond laser and streak camera [152]. Since this dataset was captured for a smaller scene than the size of scenes we target with our setup, we simulated a larger scene by scaling the time dimension by a factor of 20. We then used the PMD sensor model from Eq. (3.5) to simulate measurements of  $b_{\omega_i, \phi_i}$  for different frequencies and phases, also adding Gaussian noise with a sigma of 1%.



**Figure 3.6:** A couple of selected frames from simulation with ground-truth data. Top: Original ground-truth frames. Bottom: Reconstructions using our method. Temporal profiles for a few representative pixels are shown in Figure 3.7.

Figure 3.6 shows reconstructed frames (bottom row) in comparison with the ground truth (top). The key features are reproduced, although a certain amount of temporal smoothing is noticeable. An examination of time profiles for additional pixels confirms this analysis (Figure 3.7). In green, we also show a direct least-squares fit of the model to the ground truth curves. These curves demonstrate the expressiveness of the model. Even though the direct fit exhibits sharp discontinuities due to the use of exponential segments with a step function onset,



**Figure 3.7:** A selection of time profiles for different pixels from the example in Figure 3.6. Each graph shows the original time profile, which we treat as ground truth (blue), a direct fit of the model to this ground truth curve (green), and finally the result of our full optimization procedure using simulated PMD measurements (red, see text).

the examples show that the key features of the time sequences can be represented using the model. Furthermore, in the actual reconstruction from CIS measurements, the  $i$ -step smoothes out the discontinuities. These results demonstrate that our method can qualitatively reconstruct transient images, including complex scenarios with multiple discrete pulses of light arriving at a surface point, as well as broad temporal profiles resulting from indirect illumination via a diffuse surface. Note the resolution limit for our method is scene-dependent due to non-linear nature of our MAP approach.

### 3.5.2 CIS Measurements

Using our own setup, we captured a few settings with characteristic light transport. Individual time slices of three additional datasets are shown in Figure 3.9. The scenes and single-image visualizations of the recovered transient images are shown in Figure 3.1 and Figure 3.8. See the Appendix Section A.3 for an additional simple scene. The left column of Figure 3.9 shows a wavefront propagating through a scene

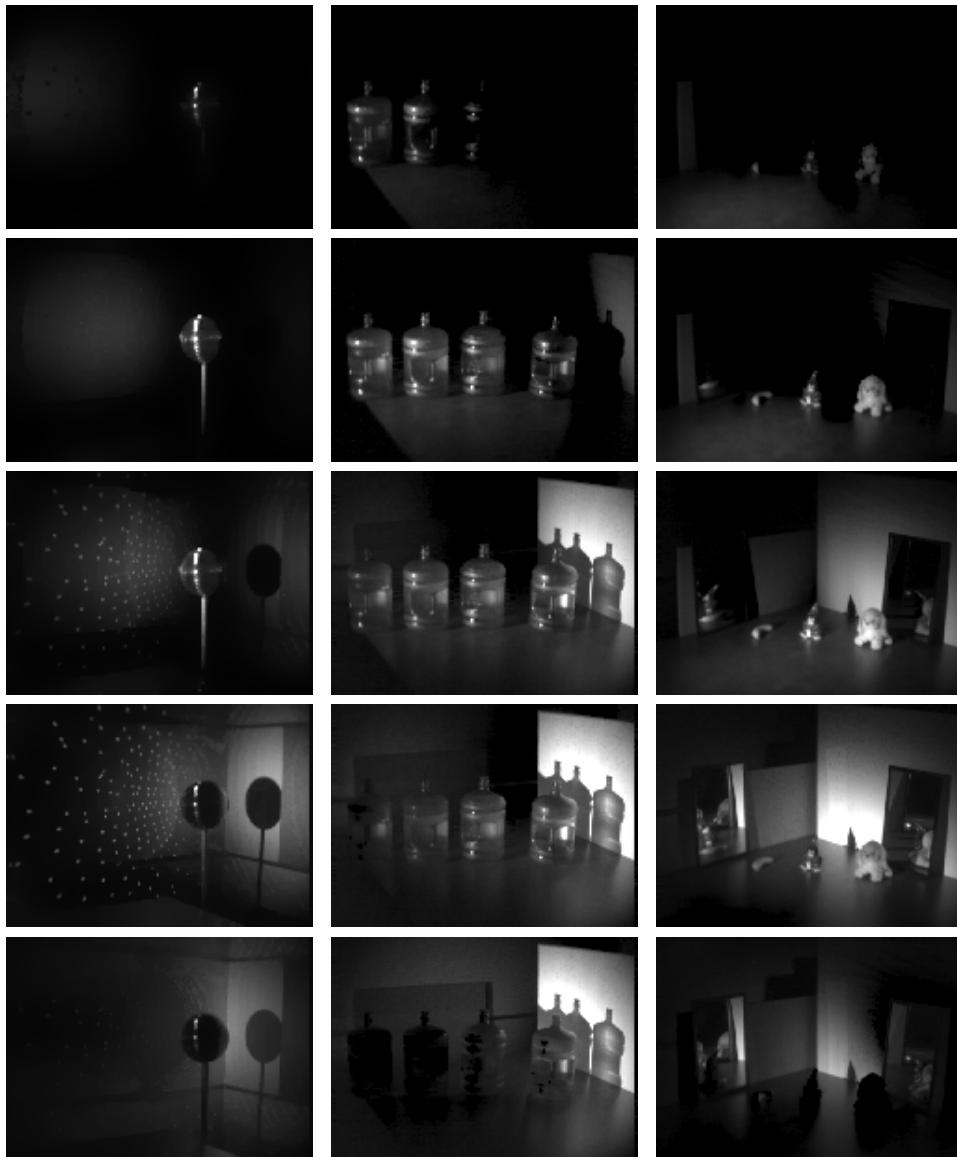


**Figure 3.8:** Our test scenes, in addition to the one from Figure 3.1. All scenes are illuminated from the left, with the rainbow images encoding the temporal location of the main intensity peak for each pixel. Top: Various objects on a table, and their reflections in mirrors placed nearby. Note how the items and mirrored counterparts light up at different times. Bottom: Four bottles filled with slightly scattering water. The specular reflections reach the camera before the light scattered inside the bottles.

with a mirrored disco ball placed in the corner of a room. In the first frame, the wavefront has just reached the front of the ball. In the second frame, the ball is now fully illuminated, and we see the wavefront propagating along the left wall. The third frame shows the first caustics generated by reflections in the mirror. More caustics appear for longer light paths near the top and bottom of the fourth image, and the direct illumination is now approaching the back of the corner from both sides. First indirect illumination in the floor is visible. In the last frame, the caustics have disappeared, and the indirect illumination is now lighting up the shadow of the ball.

The second column of the figure shows a scene with several bottles, filled with water and a small amount of milk to create scattering. In the first frame, the





**Figure 3.9:** Time slices from three transient images captured with our setup, and reconstructed with the method from Section 3.3. Each of the three columns shows a result for one of the test-scenes shown in Figure 3.1 (discoball scene on the left) and Figure 3.8 (water bottles scene in the center and mirrors scene on the right). A variety of global illumination effects can be observed. Please see the text for a detailed explanations of the different effects which we capture in each example.

wavefront has just reached the front of the leftmost bottles, and is reflecting off their surface. In the second frame, scattering effects are becoming visible in the bottles. Next, the light reaches the far wall, showing caustics of the light transport through the bottles. Indirect illumination of the back wall from light scattered in the bottles appears in the fourth frame. This light continues to illuminate the back wall even after the bottles themselves have darkened (last frame).

Finally, the last column of Figure 3.9 shows a scene with several foreground objects and two mirrors. We first see initial reflections coming off the foreground objects. As the light propagates further, the foreground objects are now fully illuminated, and the wavefront reaches the back walls, but the mirrors remain dark. In the third frame reflections of the foreground objects are starting to become visible in both the left and the right mirror. In the fourth frame, the left mirror shows a reflection of an object in the right mirror. This reflection lingers in the last frame, even after the wavefront has passed by the foreground objects.

### 3.6 Spatio-Temporal Modulation

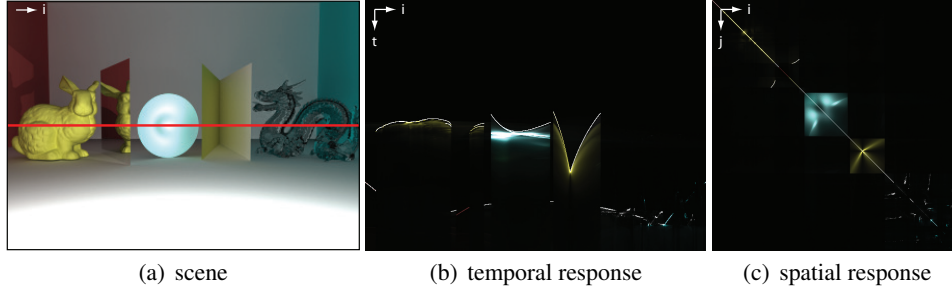
The presented approach can also be easily combined with spatial modulation. A number of recent works [190–192] analyze or capture light transport through spatial modulation using projector-camera systems, where the transport is here again assumed to be instantaneous. An established way to describe such systems is given using the light-transport matrix  $\mathbf{T}$  [192, 193], that is

$$\mathbf{i} = \mathbf{T} \mathbf{p} \quad (3.17)$$

where here now  $\mathbf{i}$  represents a 2D image with the  $X$  pixels stacked in a column vector,  $\mathbf{T}$  is the scene’s  $X \times Y$  transport matrix, and  $\mathbf{p}$  is a vector that represents the scene’s spatially-varying illumination, e.g., a 2D pattern projected onto the scene by a conventional video projector.

The spatial light transport can be interpreted as the time-average of the temporally resolved transport, that is

$$\mathbf{T} = \int_0^\infty \tilde{\mathbf{T}}(\tau) d\tau, \quad (3.18)$$



**Figure 3.10:** Simulated transient light transport, rendered using a modified path tracing integrator for non-caustic transport and a photon mapping integrator for caustic transport. For each camera pixel  $i$ , we simulate the travel time  $\tau$  of all light paths from the scene lit by a projector pixel  $j$ . **(a)** The scene contains, from left to right, a diffuse bunny, a mirror casting caustic light on the bunny, a glossy bowl, a diffuse v-shaped wedge, and a glass dragon. **(b)** A transient image of pixels  $i$  across times  $\tau$ . **(c)** Conventional light transport matrix, describing each camera pixel  $i$  in response to projector pixel  $j$ .

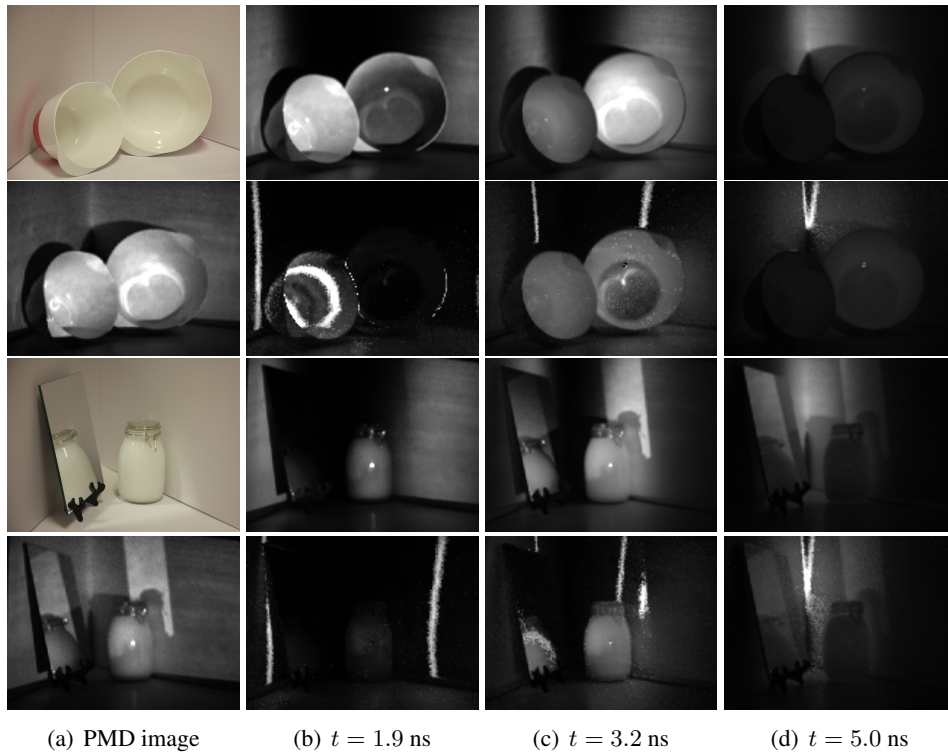
where  $\tilde{\mathbf{T}}(\cdot)$  simply extends the transient image from Eq.(3.6) by the spatial pattern dimension. Without modulation we have  $\mathbf{p} = \mathbf{1}$  and considering a pixel  $k$  it is  $i(\tau) = \sum_a \tilde{\mathbf{T}}(\tau)_{k,a}$ . Now, emitting a spatio-temporal pattern  $\tilde{\mathbf{p}}(t)$ , will produce the incident sensor signal

$$\tilde{s}(t) = \int_{-\infty}^{\infty} \tilde{\mathbf{T}}(\tau) \tilde{\mathbf{p}}(t - \tau) d\tau \stackrel{\text{def}}{=} (\tilde{\mathbf{T}} * \tilde{\mathbf{p}})(t) \quad (3.19)$$

where the operator  $*$  convolves a matrix function and a vector function over time (with notation adopted from [194]). With this temporally modulated signal incident on the sensor, we can adapt Eq. (3.8) specifically for CIS sensors and get

$$\begin{aligned} \mathbf{b}_{\omega_i, \phi_i} &= \int_0^{NT} f_{\omega_i}(t + \phi_i) \int_{-\infty}^{\infty} \tilde{\mathbf{T}}(\tau) \tilde{\mathbf{p}}(t - \tau) d\tau dt \\ &= \left[ \int_{-\infty}^{\infty} \tilde{\mathbf{T}}(\tau) \left( N \int_0^T f_{\omega_i}(t + \phi_i) g_{\omega_i}(t - \tau) dt \right) d\tau \right] \mathbf{p} \\ &= \left[ \int_{-\infty}^{\infty} \tilde{\mathbf{T}}(\tau) c_{\omega_i, \phi_i}(\tau) d\tau \right] \mathbf{p} \end{aligned} \quad (3.20)$$

It becomes obvious that adding spatial modulation leads to a straight-forward extension to Eq. (3.8). We also see that the light transport is separable for each CIS measurement. Hence, we can simply code the illumination by using a projector instead of our flood light source. For example, one can replace the RGB light source of an off-the-shelf DLP LightCrafter projector with a collimated version of the light source from Section 3.4. The spatial modulation can then, for example, be used to only selectively capture direct/retroreflective and indirect components as discussed in [190]. Since direct paths always satisfy the epipolar constraint, simply randomly turning pixels corresponding to epipolar lines on the projector



**Figure 3.11:** Transient imaging in combination with spatial modulation. **(a)** Scene captured with a normal camera under ambient illumination (rows 1, 3) and CIS camera under projector illumination (rows 2, 4). **(b-d)** Frames from the transient image reconstructed as before (rows 1, 3) and with added spatial modulation (rows 2, 4). Notice the direct component is a sharp impulse travelling along the walls.

(with probability 0.5) will suppress indirect paths from the projector to the camera. The direct/retroreflective component can then be estimated by capturing an unmasked measurement and subtracting the indirect-only component from it. We refer the reader to [190] for an in depth discussion of spatial light transport probing. Figure 3.11 shows results of transient imaging combined with this approach. The added spatial modulation aids the transient reconstruction by separating the direct and indirect components, hence simplifying the unmixing significantly. Note that the direct pulse of light traveling along the wall and the caustic light reflected by the mirror have high-temporal resolution in our reconstructions, whereas they appear very broad when reconstructed without transport decomposition. We refer the reader to our publication [2] for an in-depth analysis on the combination of transient imaging with spatial modulation, which goes beyond the scope of this dissertation.

### 3.7 Discussion

In this chapter we have presented a method for transient imaging with CIS TOF cameras. The hardware was modified to allow for different modulation frequencies and phases for both the sensor and the illumination. Solving a MAP estimation problem with spatial and temporal priors, we can robustly reconstruct transient images from CIS measurements. Unlike existing systems for the capture of transient images, ours does not require ultrafast temporal sensing or illumination. As a result, we achieve transient images with hardware costing a fraction of the price of the femtosecond laser/streak camera combinations. Our method reduced acquisition times, makes hardware more portable, and supports easier modes of operation.

A disadvantage of our system is the limited spatial and temporal resolution of CIS. The PMD PhotonICs 19k-S3 has a spatial resolution of only  $160 \times 120$  pixels. Sensors of higher resolution are available, such as the  $512 \times 424$  sensor of the Microsoft<sup>®</sup> Kinect<sup>®</sup> 2. However, most commercial camera systems like the Kinect<sup>®</sup> 2 are system-on-chip designs [195] which cannot be modified as proposed in this chapter and do not expose modulation control. The time resolution of our approach is limited by the maximum modulation frequency that can be applied to the illumination and the PMD sensor around 110 MHz, which corresponds to a spatial wavelength of about 2.70 m. The size of the investigated scene should not

be significantly below this wavelength. We therefore work with larger-scale scenes than Velten et al. [152].

The Gaussian-Exponential prior leads to a non-convex optimization problem that is hard and expensive to solve. While the convex  $i$ -substep can be solved efficiently, in under a second, the  $u$ -step takes several hours per dataset. Furthermore, the model allows for discontinuities which can be observed in Figure 3.3. We eliminate these issues in the following Chapter 4 by introducing a modified model. By exploiting the convolutional structure of transient images, a sparse model without discontinuities and corresponding convex objective can be formulated.

In summary, we have demonstrated the combination of an inexpensive hardware setup and an inverse problem to acquire transient images using CIS sensors. This approach reduces the barrier of entry for performing research in transient imaging and its applications. In the following chapters, we build on the proposed approach and demonstrate such applications. Chapter 4 demonstrates imaging in scattering and turbid media, and Chapter 5 presents a method which enables imaging of objects outside of the line of sight. In both of these traditionally challenging imaging scenarios, we show that adding computation can “make the invisible visible.”

## Chapter 4

# Imaging in Scattering Media

In this chapter, we demonstrate imaging in scattering and turbid media using CIS. This is an extremely challenging scenario since scattering events can occur not just at object surfaces, but at any location in a volume of point scatterers. Unmixing the continuum of path contributions is a hard inverse problem that is hopeless to solve without strong priors.

We solve this problem relying on a new physically-motivated model for transient images derived from an analysis of sparsity in transient images. By exploiting convolutional structure, we eliminate the issues of the Gaussian-Exponential priors mentioned before, at the very end of the previous chapter. The model leads to a convex convolutional sparse coding problem for recovering transient images from CIS. We demonstrate our method for imaging through scattering media of various densities.

### 4.1 Introduction

Imaging through scattering media has recently received a lot of attention. While many works have considered microscopic settings such as imaging in biological tissue [196, 197], we consider here the macroscopic problem, having in mind ultimate target applications such as underwater imaging or imaging through fog.

Impulse-based approaches have been reviewed in Section 2.7.1. These methods image individual femtosecond laser pulses with streak cameras [147, 149], or fast

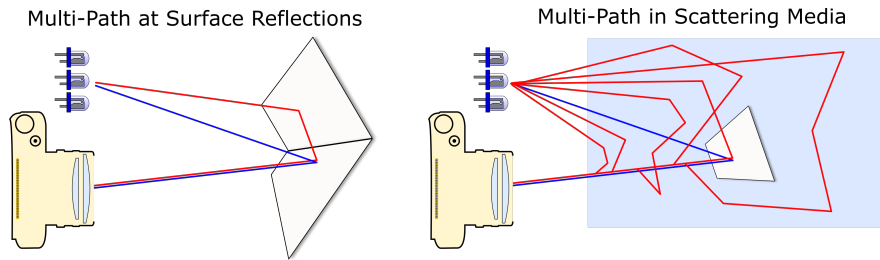


**Figure 4.1:** Example of imaging in scattering media using our approach. Left: Original scene with objects submerged in water-filled glass tank. Center: 160 ml milk added. Right: Objects that are “invisible” due to strong scattering, like the structured object on the top right or parts of the circular object become detectable using our approach.

gated cameras [196, 198, 199]. However, impulse-based approaches suffer from low SNR, since ambient illumination can easily overpower the laser pulse. Therefore, various approaches have been explored to increase the SNR. Dagleish et al. [200] combine time gated imaging with spatial modulation for an underwater imaging application. Mullen and Contarino [201] propose a hybrid system that combines gated imaging with microwave amplitude modulation per pulse see also [202–204]. The amplitude modulation in the GHz-range reduces multi-path mixing components significantly as also discussed in [156]. However, these hybrid impulse-based approaches still directly sample the modulated incident scene response which is affected by ambient light. Therefore, many repeated impulses per capture are required. In contrast, CIS TOF cameras allow for the integration over many (ca.  $10^4$ – $10^5$ ) pulses of a modulated light source. It is key to realize that these sensors perform adaptive in-pixel background suppression *during* the integration. As discussed in Section 2.6.4, and in more detail in [52, 205, 206], the unmodulated background component of photo-electrons is measured in both buckets equally and can be drained during exposure (potentially multiple times). This operating mode allows CIS sensors to amplify the signal component independently of the ambient light by increasing the exposure time, while saturation due to ambient light no longer occurs. Note also that it allows a per-pixel exposure time [205].

can be effectively used for imaging in scattering and turbid media, when combined with computational analysis based on sparse coding; see Figure 4.1. Recalling Chapter 3, the primary advantage of Correlation Image Sensors is that they have





**Figure 4.2:** Left: Surface reflections superimpose indirect path contributions (red) on the direct path (blue). As already discussed in Section 2.6.5, TOF imaging has to account for the indirect path contributions. Right: Imaging in scattering media results in indirect contributions not only from object surfaces, but due to scattering in the entire volume (light blue). Note that in both cases a continuum of indirect paths is measured.

extended exposure intervals, much like regular cameras, and they integrate over many (ca.  $10^4$ – $10^5$ ) pulses of a modulated light source, instead of a single pulse, resulting in drastically reduced capture time, as well as slower, less expensive electronics. However, for imaging in scattering media multi-path contributions are even stronger than for regular TOF imaging; see Figure 4.2. At a pixel a mixture of path lengths is measured, where only few ballistic photons directly hit objects submerged in the scattering media, and a large number of indirect paths are created by scattering events in the entire volume. This strong scattering, which makes traditional imaging very challenging, can only be handled if multi-path contributions are removed effectively. In the following, we present a sparse convolutional model that encodes strong prior knowledge on the transient image, and hence enables effective MPI suppression in scattering and turbid media.

## 4.2 Sparse Transient Mixture Model

### 4.2.1 Are Transient Images Sparse ?

The idea of using sparse representations for signal recovery was first analyzed extensively by Donoho and Candes et al. [207, 208] and since then has found use in many domains [209]. Several recent works [126, 210, 211] attempt to resolve multi-path interference by assuming sparsity of the transient profiles in the Dirac

basis. The vastly popular idea of compressed sensing [207] can be applied in a straightforward manner, by trying to solve the basis pursuit denoising problem [212]:

$$\mathbf{i}_{\text{opt}} = \underset{\mathbf{i}}{\operatorname{argmin}} \|\mathbf{i}\|_1 \quad \text{subject to} \quad \|\mathbf{C}\mathbf{i} - \mathbf{b}\|_2^2 < \epsilon, \quad (4.1)$$

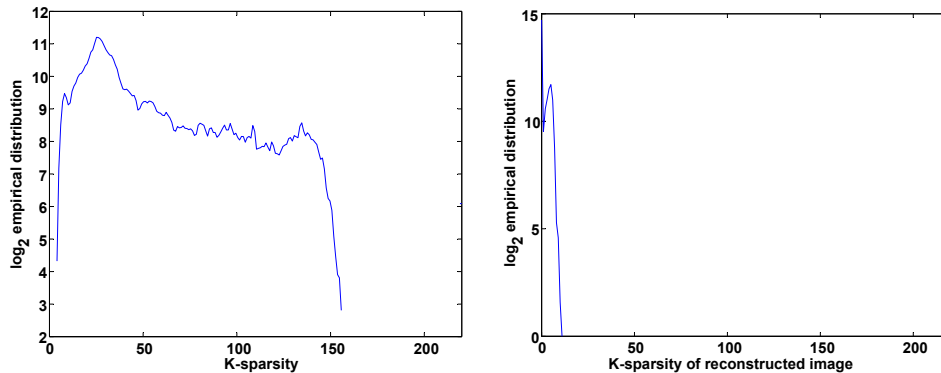
following again the notation from Chapter 3. Eq. (4.1) is a convex relaxation of the sparsity requirement  $\|\mathbf{i}\|_0 < K$ , where  $K$  determines the sparsity, and hence can be solved using convex optimization methods. The problem from Eq. (4.1) can be interpreted in our Bayesian framework from Section 2.8.1 as the MAP estimation problem

$$\mathbf{i}_{\text{opt}} = \underset{\mathbf{i}}{\operatorname{argmin}} \frac{1}{2} \|\mathbf{C}\mathbf{i} - \mathbf{b}\|_2^2 + \lambda \|\mathbf{i}\|_1 \quad (4.2)$$

The parameter  $\lambda$  is here related to the reciprocal of Lagrange multiplier of the constrained basis pursuit denoising problem from Eq. (4.1). For appropriate parameters of  $\lambda$ ,  $\epsilon$  both problems yield the same solutions [212]. This formulation has first been proposed for compressed sensing tasks in [213]. From a Bayesian perspective, this problem is MAP estimation problem with Gaussian Likelihood and a Laplacian prior distribution on the basis coefficients. As in the previous chapters, the parameter  $\lambda$  can then be simply interpreted as a weight absorbing the scaling variables of the likelihood and prior distribution, i.e. a trade-off between the data fidelity and regularizer.

The basic assumption of the compressed sensing approach from Eq. (4.1), (4.2) is  $\mathbf{i}$  being sparse. However, this is violated for many realistic environments, and in particular in the case of scattering media. For regular scenes the (intensity-modulated) radiosity leaving a single scene point integrates over a continuum of scene points, and thus in general cannot be sparse. For example, any concave object can be expected to deliver a non-sparse response. Especially for imaging in scattering media, the assumption of temporal sparsity breaks down. This is demonstrated empirically in Figure 4.3 on the left by plotting the histogram of the  $\ell_0$ -“norm”, that is the number of non-zero entries, of a high-resolution transient image captured by Velten et al. [214].

The transient image depicts a scene composed of scattering objects (a tomato) and diffuse surfaces. However, note that even pixels that do not view a scattering



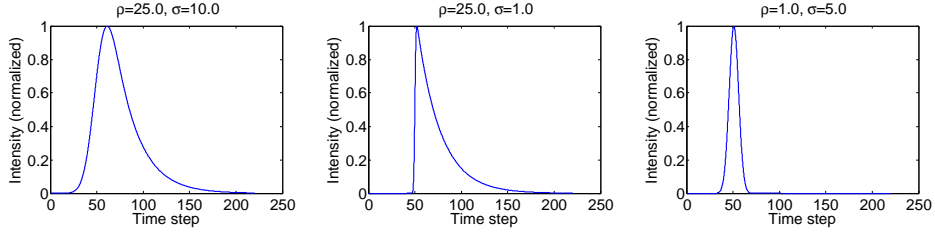
**Figure 4.3:** Sparsity of 40K signals of a transient image measured image (left). Sparsity after fitting to the convolutional basis proposed in this chapter (right).

object are not necessarily completely sparse, as we will show later in Section 4.3.2. The signal has been thresholded for 0.1 of the peak signal value along each pixel so as not to interpret sensor noise as sparse components. A large number of pixel signals have more than  $K > 100$  components, which cannot be considered sparse, given a time discretization of  $M = 220$  in this example.

## 4.2.2 Sparse Local Model for Transient Light Interaction

Having shown that the popular sparse model does not apply in the Dirac basis for realistic scenes (and especially not for scattering media), we now introduce a model for local light transport interaction. This model leads to an overcomplete basis transforming the signal into a space where it is sparse/compressible.

Building on Section 3.3, we model the temporal PSF of direct local surface reflection from a single point as a Dirac peak, while the temporal structure of scattering processes is best represented by an exponential decay. However, now, in both cases, this PSF is convolved with a Gaussian that models the temporal PSF and resolution of the correlation image sensor. It is therefore plausible to describe a transient pixel by a mixture of *exponentially modified Gaussians*. Originally developed to describe chromatographic peaks [215], this versatile model has since been adopted in different fields such as high-energy physics, where it is used to model the response of pixel detectors [216]. A single exponentially modified



**Figure 4.4:** A few samples of the exponentially modified Gaussian signals located at  $\mu = 50$ . Curves are normalized by adjusting amplitude  $a$ .

Gaussian can be defined as

$$h(\tau; a, \sigma, \rho, \mu) = a \cdot \exp\left(\frac{1}{2} \left(\frac{\sigma}{\rho}\right)^2 - \frac{\tau - \mu}{\rho}\right) \cdot \left(1 + \operatorname{erf}\left(\frac{(\tau - \mu) - \frac{\sigma^2}{\rho}}{\sqrt{2}\sigma}\right)\right), \quad (4.3)$$

where  $a$  (amplitude),  $\sigma$  (Gaussian width),  $\rho$  (skew) and  $\mu$  (peak center) are the parameters of the exponentially modified Gaussian function, and  $\tau$  is the travel time at which it is evaluated. If we stack the parameters for a single exponentially modified Gaussian in a vector  $\mathbf{u} = [a, \sigma, \rho, \mu]^T$ , then we can model the transient time-profile  $i$  as the mixture:

$$i(\tau) = \sum_{k=1}^N h(\tau; \mathbf{u}_k), \quad (4.4)$$

where  $N$  is here the number of mixture components. Figure 4.4 shows a few samples of the signals. We note that the joint modeling of the exponential and Gaussian nature of our signals has a key advantage over previous models: the basis functions and hence the reconstructions are inherently smooth. The Gaussian-Exponential prior from Section 3.3, on the other hand, produces discontinuous solutions whenever exponential components are being used.

Now, fitting this model to the data from Figure 4.3 (using the method described below), we can see that it defines a basis that transforms the transient image signals into a space that is significantly sparser than the signal itself; see Figure 4.3, right.

### 4.3 Transient Convolutional Sparse Coding

In order to find the optimal mixture parameters, commonly a non-convex non-linear problem is solved, which is prone to local minima and expensive to solve. Furthermore it leaves the number of mixing components open, which was solved in Chapter 3 using an alternate scheme that also offers no guarantee of global convergence.

We follow here a different approach and first linearize the above model by sampling the parameter space and later use it in basis pursuit fashion in a convex reconstruction problem. Linearizing the basis functions gives

$$\mathbf{D} = [h(\mathbf{s}; \mathbf{u}_0), \dots, h(\mathbf{s}; \mathbf{u}_N)] \quad (4.5)$$

where  $\mathbf{s}$  is the vector of all the sampling positions of the signal and  $N$  is the number of samples in the set containing all  $\mathcal{C} = \{\sigma, \rho, \mu\}$  with  $a = 1$ . Including this (massively overcomplete) basis  $\mathbf{D}$  in Eq. (4.2) leads to the following pursuit denoising problem.

$$\mathbf{x}_{\text{opt}} = \underset{\mathbf{x}}{\operatorname{argmin}} \frac{1}{2} \|\mathbf{CD}\mathbf{x} - \mathbf{b}\|_2^2 + \lambda \|\mathbf{x}\|_1 \quad (4.6)$$

The over-completeness of the basis is now handled by the sparsity prior. The values of  $a$  have now become the vector of basis coefficients  $\mathbf{x}$ . For simplicity, we consider only one pixel, although the approach can be trivially extended to multiple pixels/measurements.

However, the basis  $\mathbf{D}$  is still extremely large (due to the large set  $\mathcal{C}$ ), which makes the optimization inefficient even for problems of moderate size. One significant improvement to this situation is to exploit the convolutional nature of the basis. Instead of sampling the parameter space for  $\mu$  (i.e., the translation of the peak along the time axis), one can fix  $\mu$  and reformulate the problem as the following optimization problem:

$$\begin{aligned} \mathbf{x}_{\text{opt}} = \underset{\mathbf{x}}{\operatorname{argmin}} \quad & \frac{1}{2} \left\| \mathbf{C} \sum_{k=0}^K \mathbf{D}_k \otimes \mathbf{x}_k - \mathbf{b} \right\|_2^2 + \lambda \|\mathbf{x}\|_1 \\ \text{subject to} \quad & \mathbf{D}_k \in h(\mathbf{s}, \mathcal{C}') \quad \forall k \in \{1, \dots, K\} \end{aligned} \quad (4.7)$$

The basis vectors are now sampled from the space of pulse *shapes*,  $\mathcal{C}' = \{\sigma, \rho\}$ , and invariant to translation. Therefore the size of the basis is drastically reduced from previously  $N = K \cdot \dim(\mathbf{x})$  to just  $K$ , i.e., by typically around 2 to 3 orders of magnitude.

This approach is motivated by the signal processing of the acoustic nerve, where *shift-invariant* sparse coding has first been proposed by Lewicki and Sejnowski [217]. Recently, convolutional sparse coding has also been used for audio and image detection tasks [218–220] motivated by the recent success of convolutional deep neural networks in image classification and detection [18].

### 4.3.1 Reformulation in the Spectral Domain

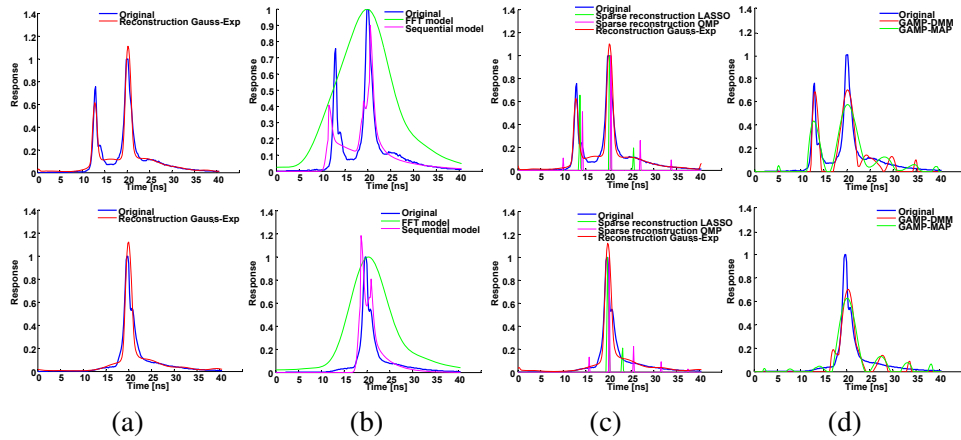
The problem from Eq. (4.7) can be defined even more compactly in the Fourier domain. The convolution reduces to pointwise multiplication, here expressed by the operator  $\odot$ :

$$\begin{aligned} \mathbf{x}_{\text{opt}} = \underset{\mathbf{x}}{\operatorname{argmin}} \quad & \frac{1}{2} \left\| \mathbf{C} \mathfrak{F}^{-1} \left( \sum_{k=0}^K \hat{\mathbf{D}}_k \odot \hat{\mathbf{x}}_k \right) - \mathbf{b} \right\|_2^2 + \lambda \sum_{k=0}^K \|\mathbf{x}_k\|_1, \\ \text{subject to} \quad & \mathbf{D}_k \in h(\mathbf{s}, \mathcal{C}') \quad \forall k \in \{1, \dots, K\} \end{aligned} \quad (4.8)$$

The operators  $\mathfrak{F}$ ,  $\mathfrak{F}^{-1}$  denote here the Fourier transform and the inverse Fourier transform, respectively. The operator  $\hat{\cdot}$  is a notational shortcut the frequency domain representation of a variable, that is  $\hat{\mathbf{v}} = \mathfrak{F}(\mathbf{v})$ . Due to the frequency domain formulation, this linear inverse problem can be solved with an efficient ADMM algorithm. In Chapter 6 we describe a generalized approach to convolutional sparse coding, which also allows to learn convolutional structure when not given by the physical model. A detailed description of frequency domain convolutional sparse coding will be given in this chapter. Since it will also cover the implementation details of our approach, the remainder of this chapter evaluates the proposed model for transient imaging in scattering media.

### 4.3.2 Synthetic Evaluation

To generate ground-truth pixel profiles, we sampled a high resolution “ground truth” transient image measured by Velten et al. using direct temporal method [214]. The observations are then generated by assuming a typical sinusoidal measurement matrix  $\mathbf{C}$  where  $f, g$  are defined as in Eq. (2.24). We sample  $\omega$  evenly spaced in 100 steps from 10 MHz to 120 MHz, which is a realistic range for recent CIS systems, recalling Chapter 3, and  $\phi$  as  $0, \pi/2$ , giving exactly  $100 \cdot 2$  measurements  $\mathbf{j}$  per pixel. The measurements are normalized and then corrupted with 1% Gaussian noise. Figure 4.5 shows synthetic results for two different pixel profiles. We compare our method (“Reconstruction Gauss-Exp”) to the ground-truth signal (“Original”) as well as two recent methods from transient imaging literature: Lin et al.’s smooth frequency-domain interpolation (“FFT model”) [221], the non-linear non-convex model-fit from Chapter 3 (“Sequential model”). For the sake of completeness, we further add comparisons to various state-of-the-art sparse reconstruction techniques, namely LASSO (“Sparse reconstruction LASSO”) [222], OMP (“Sparse reconstruction OMP”) [223] as well as Generalized Approximate Message Passing [224] using



**Figure 4.5:** Example showing the effect of our sparse coding optimization on two pixels from the “Tomato” dataset. From left to right: (a) Proposed new model, (b) FFT and sequential models, (c) state-of-the-art sparse reconstruction (LASSO and OMP), (d) two state-of-the-art compressed sensing models (GAMP).

Donoho/Maleki/Montanari-style thresholding (“GAMP-DMM”) and assuming a Laplacian signal in an MAP formulation (“GAMP-MAP”).

One can see that, although these pixels are dominated by direct reflections, the signals are not sparse at all, and time-domain sparse backscatter models as used by Freedman et al.’s SRA [126], or Bhandari et al. [211] are not capable of close reconstructions. Our method, on the other hand, produces solutions that follow the ground-truth distributions more closely than any of the competing models. In particular, the exponentially modified Gaussian basis outperforms approaches that solve non-linear non-convex optimization, such as the one from Chapter 3. Out of all the methods tested, the one delivering the poorest fit is [221], which imposes the weakest prior by just interpolating smoothly in the Fourier domain.

## 4.4 Results

In this section we show results for imaging in scattering media using our reconstruction method proposed in the previous section. The measurement setup is explained first. After that the results are analyzed qualitatively and quantitatively.

### 4.4.1 Setup

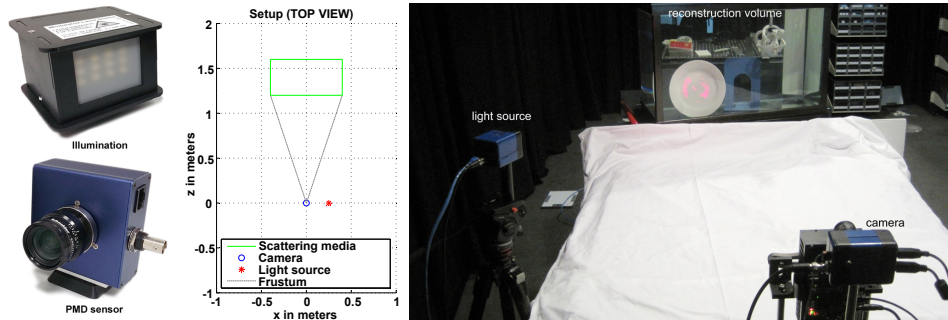
*Camera* We use a correlation image ToF camera prototype for our experiments using red laser light as illumination, see Figure 4.6 on the left.

The TOF camera consists of a modulated light source and a correlation image detector. As in the previous Chapter 5, the sensor is a PMD CamBoard nano, modified to allow for external control of the modulation signal (since access to its FPGA configuration was not available). The light source consists of an array of 650 nm, 250mW laser diodes which are operated in pulse mode, also described in the previous chapter.

We measure  $\omega$  evenly spaced in 3 steps from 20 MHz to 60 MHz and  $\phi$  evenly spaced between 0 and  $5\text{ m}/c\omega$  in 201 steps and with an additional shift of  $(0, \pi/2)$ , resulting in a measurement vector  $\mathbf{j}$  with exactly  $N = 201 \cdot 2 \cdot 3$  samples per pixel.

*Calibration* As described in Chapter 3, we calibrated the matrix  $\mathbf{C}$  by measuring a diffuse planar target that is mounted on a translation stage perpendicular to the





**Figure 4.6:** Camera prototype and setup: We use a modified PMD Technologies CamBoard nano using an array of red laser diodes for illumination (left) as described in the previous Chapter 5. In our setup we image a tank filled with a scattering medium of different concentrations frontal with the cameras. The spatial dimensions and arrangement of the setup is shown in the center. The setup is shown on the right.

z-axis. The target is translated along this axis at positions according to different travel times  $\tau_i, i \in \{1 \dots M\}$ . We populate  $\mathbf{C}$  column by column for each  $\tau_i$ .

*Measurement Setup* An image of our measurement setup is shown in Figure 4.6 on the right. We placed a water tank with glass hull at a distance of 1.2 m in front of our camera, so that the optical axis intersects the center of the largest planar side. The light source is placed at a slight offset to the right to eliminate direct reflections of the air-to-glass interface on the camera-facing wall. See the schematic in Figure 4.6 in the center for the exact spatial alignment.

*Scattering Media* We filled the tank with 80 liters of water and submerged objects at different positions in the water (and in particular at different distances to the camera). We then evaluated our approach on two different scattering materials with a series of different concentrations. First, we conducted a sequence of 100 experiments using homogenized milk from 0 ml to 500 ml in 5 ml steps. Second, we conducted a sequence of 50 experiments using Gypsum plaster with a continuum of particle sizes  $\leq 0.125$  mm. We used 0 oz to 50 oz in 1 oz steps.

For each of the 150 experiments we take a full measurement with the measure-

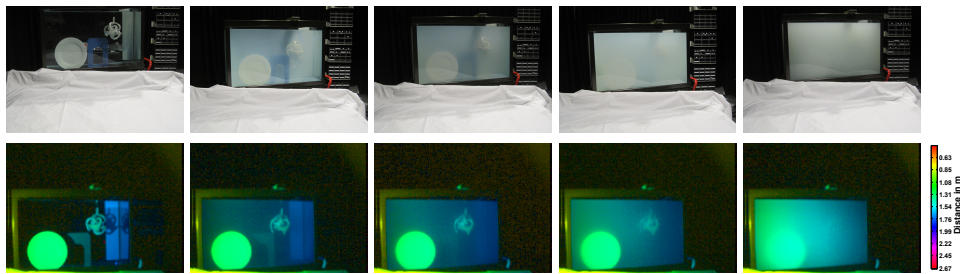
ment parameters as described above.

#### 4.4.2 Qualitative Results

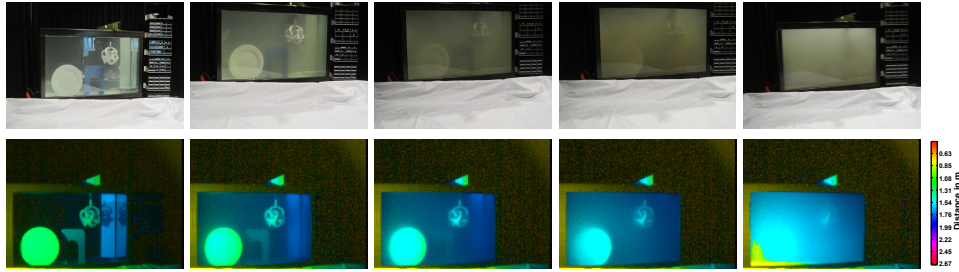
Figure 4.7 and Figure 4.8 show qualitative results for imaging through scattering media of increasing density. Objects are immersed in a tank filled with water, to which increasing concentrations of milk (Figure 4.7) or plaster (Figure 4.8) are added. See Section A.5 for the full sequence of all concentrations that we tested. With increasing concentration, visibility through the scattering medium quickly drops off for a conventional camera (top row). On the other hand, a light transport analysis based on Correlation Image Sensors, not only increases the ability to detect objects in highly turbid solutions, but also allows for a simultaneous estimation of distance (color coded images in the bottom column of each figure).

#### 4.4.3 Quantitative Results

To quantitatively analyze our results, we measure the error of the depth estimate for three distinct camera pixels with respect to the measured ground truth depths. The three camera pixels ‘plate’, ‘holder’ and ‘bar’ are chosen as representatives for objects located at different depths in the reconstruction volume. Their spatial



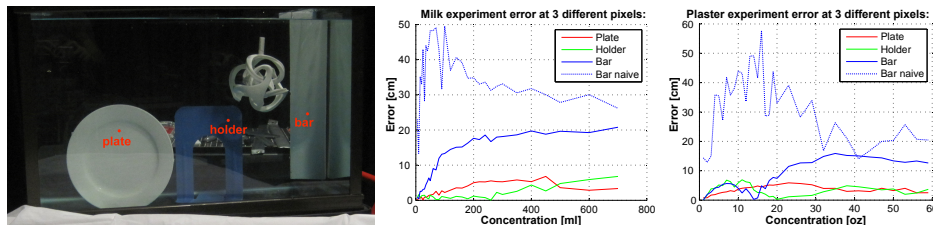
**Figure 4.7:** Qualitative results for the milk experiment described in Section 4.4.1. Experiments in each column from left to right: 0 ml, 20 ml, 40 ml, 80 ml, 300 ml of milk in water. Each column shows the regular camera image (top), peak image for red camera reconstruction (bottom). Peak images are here a parabola fit through the two nearest neighbor points of the strongest peak, where the position is encoded as hue and the intensity is encoded as value in the HSV color model.



**Figure 4.8:** Qualitative results for the plaster experiment from Section 4.4.1. Experiments from left to right: 0 oz, 4 oz, 8 oz, 16 oz, 59 oz of plaster in water. The same visualization as in Figure 4.7 has been used.

positions are shown in Figure 4.9 (left). All objects have approximately diffuse reflectance except for the ‘plate’ which does have a specular component. Ground truth depth measurements for all pixels were acquired manually. The pixel ‘plate’ is located at 5.5 cm behind front facing glass wall of the tank, pixel ‘holder’ at 21.5 cm into the tank, and ‘bar’ at 40.0 cm, touching the rear-facing wall of the tank.

The center and right portions of Figure 4.9 show the error with respect to the ground truth pixel depth for *all 100 experiments with milk as scattering media and all 50 experiments with plaster as scattering media* as described in Section 4.4.1. We can see that the error of pixels ‘plate’ and ‘holder’ has a fairly low slope and



**Figure 4.9:** Error for three specific pixels shown on the left. The error of the position is shown for all 100 experiments using milk as scattering medium and all 50 experiments using plaster as scattering medium. The error is measured with respect to 0% concentration and corrected for the speed of light in water. Standard time-of-flight depth reconstruction (atan solution from Section 2.6.1) breaks down even at very low concentrations of scattering agent.

remains almost flat around 1 cm–5 cm even for strong concentrations. These pixel depths are located close to the front and in the middle of the reconstruction volume, so the scattering is reduced in comparison to pixel ‘bar’ which is at the very rear of the reconstruction volume. Its error is significantly larger; around 10 cm–20 cm due to the increased scattering. However, performing a naïve reconstruction as described in Section 2.6.1 on the same pixel resulted in even significantly larger errors around 30 cm–60 cm for both the plaster and the milk sequence of experiments.

## 4.5 Discussion

In this chapter, we have demonstrated that CIS imagers can be used for imaging in scattering and turbid media. In the presence of scattering, light rays emitted into the scattering volume are perturbed by a large number of point scatterers, each causing a scattering event. Unlike MPI problems for range imaging, with large free-space components, the point scatterers are continuously distributed in the volume. Hence, imaging in scattering media is a hard MPI problem, which requires the unmixing of this continuum of path contributions.

The key to solving this problem using Correlation Image Sensors lies in exploiting strong prior knowledge on the transient light transport. By identifying convolutional sparse structure in transient images, we have formulated a compressive and expressive representation. This representation is based on exponentially modified Gaussians, which are tailored towards representing combinations of surface reflection and volumetric scattering. Exploiting convolutional structure of transient images is key for allowing the prior to find the sparse signal structure. By solving the resulting convolutional sparse coding problem, we have demonstrated imaging in turbid and scattering media. Our method significantly outperforms naive TOF range imaging methods that do not resolve MPI. It is robust across a large range of scattering densities, demonstrating that our model generalizes well.

In contrast to impulse-based methods, our CIS-based approach inherits the advantages mentioned in the previous Chapter 3, that is robustness to ambient illumination (all measurements presented in this chapter were taken with room lighting switched on), and works with higher light levels than approaches based on single light pulses. However, the proposed approach requires computational reconstruc-

tion, whereas impulse-based methods perform direct sampling. The quality of our multi-path unmixing is limited by the condition number of  $\mathbf{C}$ , while impulse-based methods are limited by the pulse-shape and temporal camera resolution. Since  $\mathbf{C}$  is a truncated Fourier transform, see Section 3.4, the modulation frequency determines the limit here, which for commercially available CIS imagers is up to 100 MHz. However, this means also that our method will directly benefit from increases in modulation frequency for future TOF range imagers. Furthermore, our convolutional sparse model, which enables the results in this chapter, generalizes to other TOF technologies, such as direct temporal sampling ( $\mathbf{C}$  can be calibrated with a delay-sweep, as before in Chapter 3).

## Chapter 5

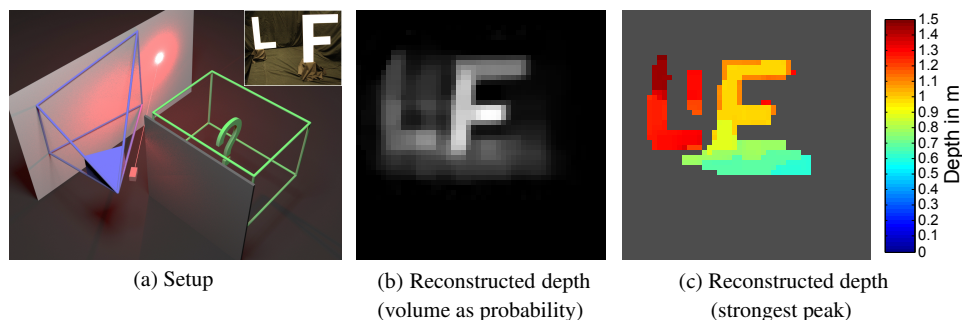
# Diffuse Mirrors

The functional difference between a diffuse wall and a mirror is well understood: one scatters back into all directions, and the other one preserves the directionality of reflected light. The temporal structure of the light, however, is left intact by both. In other words, assuming simple surface reflection, photons that arrive first are reflected first.

In this chapter, we exploit this insight to recover objects outside the line of sight from second-order diffuse reflections, effectively turning walls into mirrors. We formulate the reconstruction task as a linear inverse problem on the transient response of a scene, which we acquire, similar to the previous Chapters 4 and 3, using an affordable setup consisting of a modulated light source and a CIS TOF camera. By exploiting sparsity in the reconstruction domain, we achieve resolutions in the order of a few centimeters for object shape (depth and laterally) and albedo. Our method is robust to ambient light and works for large room-sized scenes. It is drastically faster and less expensive than previous approaches using femtosecond lasers and streak cameras, and does not require any moving parts.

### 5.1 Introduction

Object reconstruction from real-world imagery is one of the central problems in computer vision, and the very mechanism of image formation (each pixel measuring light flux as a multidimensional plenoptic integral) is one of the main reasons why it



**Figure 5.1:** (a) illustration of our measurement scenario (to scale). A diffuse wall is illuminated by a modulated laser beam and observed by a TOF camera. From diffuse reflections, we infer the geometry and albedo of objects within a bounding volume (green) that is completely occluded to both light source and camera, but visible from most locations on the wall. In this example, of two letters are placed in the unknown volume; see the scene photograph in top right inset in (a). The shape of the letters becomes clearly visible in the reconstruction (b,c).

is so challenging. To overcome the limitations of standard monocular images taken under uncontrolled illumination with respect to many vision tasks, a wide range of novel capturing approaches has emerged that extend the concept of digital imaging with structured light or new sensing techniques involving masks, filters or integral optics (light fields) [225].

The additional temporal dimension introduced by transient imaging can solve some of these challenges. By undoing the temporal mixing of the light transport, transient imaging has enabled the use of diffuse reflectors to image objects via the time profile of reflections from ultra-short laser pulses [149, 226]; see also Section 2.7. However, reconstruction of the this data from transient images is a hard inverse problem, that is sensitive to the exact parametrization of the problem as well as the priors and regularization terms that are employed. In this chapter, we develop a new parametrization for this inverse problem, and combine it with a novel set of sparsity inducing priors to achieve a robust reconstruction of geometry and albedo from transient images.

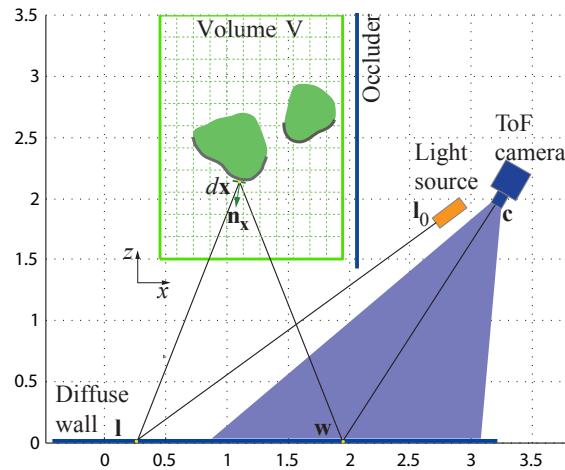
As discussed in Chapter 3, a major challenge is that the instrumentation required to measure the transient images themselves has traditionally suffered from severe

practical limitations including excessive hardware cost (hundreds of thousands of dollars), long acquisition times (hours) and the difficulty of keeping the sensitive system calibrated. We solve this problem by building on the method from the previous Chapter 3 using widespread CIS TOF sensors for obtaining the transient image. The inverse problems for transient image reconstruction and geometry recover can be merged into a single non-linear optimization problem that can be solved efficiently. The result is a system that is by several orders of magnitude more affordable and acquires data faster than previous solutions. We demonstrate the effectiveness of our setup and the computational scheme by reconstructing both a low-contrast albedo and the geometry of hidden objects.

## 5.2 Image Formation Model

### 5.2.1 Assumptions

We make several assumptions for the image formation; see Figure 5.2. The hidden



**Figure 5.2:** A schematic of our measurement setup illustrated in Figure 5.1. All distances are in meters. An isotropic point light source is created on the diffuse wall by illuminating it with a laser beam. This diffuse light source illuminates scene objects in the unknown scene volume (green), outside of the line of sight of the camera. Some of the light is reflected back to the wall (1st bounce) and can be measured by the camera bouncing of the diffuse wall (2nd bounce).



scene is modeled as a diffuse height field, which in turn is represented as a collection of differential patches  $d\mathbf{x}$  with orientation  $\mathbf{n}_\mathbf{x}$  inside a volume  $V$ . We assume that the camera points at a section of the diffuse wall, and is in focus. Light is emitted as a laser ray from position  $\mathbf{l}_0$  and illuminates a single point  $\mathbf{l}$  on the diffuse wall, outside the field of view of the camera. Radiometrically, we treat this point as a single, isotropic point light emitting a radiance  $L_e(\mathbf{l})$ . From  $\mathbf{l}$ , the light illuminates the scene, and after a single bounce returns to the diffuse wall. Patches  $d\mathbf{w}$  on the wall are chosen such that there is a one-to-one correspondence between patches and camera pixels. Finally, we ignore occlusions in the height field.

### 5.2.2 Stationary Light Transport

With these assumptions, starting from the diffuse version of the Rendering Equation introduced in Section 2.2.2, we can model the stationary (i.e. time independent) light transport as follows.

$$\begin{aligned} L(\mathbf{l}) &= L_e(\mathbf{l}) \\ L(\mathbf{x}) &= L_e(\mathbf{l})\rho(\mathbf{x})R(\mathbf{l}, \mathbf{x}) \\ L(\mathbf{w}) &= \int_V L_e(\mathbf{l})\rho(\mathbf{x})R(\mathbf{l}, \mathbf{x})\rho(\mathbf{w})R(\mathbf{x}, \mathbf{w}) d\mathbf{x} \end{aligned} \quad (5.1)$$

with  $\rho(\cdot)$  denoting the diffuse albedo of a patch, and the *unoccluded* geometry term

$$R(\mathbf{x}, \mathbf{y}) = \frac{\cos \angle(\mathbf{y} - \mathbf{x}, \mathbf{n}_\mathbf{x}) \cdot \cos \angle(\mathbf{x} - \mathbf{y}, \mathbf{n}_\mathbf{y})}{|\mathbf{y} - \mathbf{x}|^2}. \quad (5.2)$$

between the patch at  $\mathbf{x}$  and another patch at  $\mathbf{y}$ . We re-write the radiance at a wall patch (Eq. (5.1)) as

$$L(\mathbf{w}) = L_e(\mathbf{l})\rho(\mathbf{w}) \int_V r(\mathbf{x})v(\mathbf{x}) d\mathbf{x}, \quad (5.3)$$

where the geometry term

$$r(\mathbf{x}) = \frac{\cos \angle(\mathbf{x} - \mathbf{l}, \mathbf{n}_\mathbf{l}) \cdot \cos \angle(\mathbf{x} - \mathbf{w}, \mathbf{n}_\mathbf{w})}{|\mathbf{x} - \mathbf{l}|^2 \cdot |\mathbf{x} - \mathbf{w}|^2} \quad (5.4)$$

is independent of both the albedo and orientation of the patch  $d\mathbf{x}$ , while

$$v(\mathbf{x}) = \rho(\mathbf{x}) \cdot \cos \angle(\mathbf{l} - \mathbf{x}, \mathbf{n}_{\mathbf{x}}) \cdot \cos \angle(\mathbf{w} - \mathbf{x}, \mathbf{n}_{\mathbf{x}}) \quad (5.5)$$

is a term that isolates both of these unknown quantities. We can interpret  $v(\mathbf{x})$  either as a generalized albedo term or as a continuous volume occupancy that indicates whether or not a given voxel location is occupied by the surface to be reconstructed. Note that in this parametrization, the image formation is linear in  $v(\mathbf{x})$ .

### 5.2.3 Transient Light Transport

Following the approach from the previous chapter, the transient version of Eq. (5.3) is obtained by adding a time coordinate  $t$  and counting only light contributions such that  $t$  is the sum of emission time  $t_0$  and the travel time  $\tau$  for a given light path from the laser  $\mathbf{l}_0$  to a camera pixel  $\mathbf{c}$ . In our image formation model, the relevant light paths only differ in the position of the surface element  $d\mathbf{x}$ , i.e.  $t = t_0 + \tau(\mathbf{x})$ .

Recalling that we assume a one-to-one correspondence between wall patches  $\mathbf{w}$  and camera pixels  $\mathbf{c}$ , we obtain the transient image formation model

$$L(\mathbf{c}, t) = \int_0^t L_e(\mathbf{l}, t_0) \rho(\mathbf{w}) \int_V \delta(t_0 + \tau(\mathbf{x}) - t) r(\mathbf{x}) v(\mathbf{x}) d\mathbf{x} dt_0, \quad (5.6)$$

where the travel time  $\tau(\mathbf{x})$  is the total path length divided by the speed of light  $c$ :

$$\tau(\mathbf{x}) = (|\mathbf{l}_0 - \mathbf{l}| + |\mathbf{l} - \mathbf{x}| + |\mathbf{x} - \mathbf{w}| + |\mathbf{w} - \mathbf{c}|) / c \quad (5.7)$$

We note that this transient image formation model is independent of the way the transient image has been acquired. It therefore applies to all known approaches for generating transient images, including femtosecond imaging [149] as well as correlation-based measurements from the previous Chapter 3.

### 5.2.4 Discretization

The problem of reconstructing geometry from indirect light amounts to recovering the diffuse height field represented as the continuous voxel densities  $v(\mathbf{x})$ . To this end, we discretize the volume  $v(\mathbf{x})$  from Eq. (5.6) into a Euclidean voxel grid, and

represent it as a vector  $\mathbf{v}$  of size  $m$ . The transient image (radiance) is represented as a vector  $\mathbf{i} \in \mathbb{R}^{nt}$ , where  $n$  is the number of camera pixels/wall patches, and  $t$  is the number of time steps. The discrete version of Equation 5.6 is then given as

$$\mathbf{i} = \mathbf{P}\mathbf{v} \quad (5.8)$$

with the light transport tensor  $\mathbf{P} \in \mathbb{R}^{nt \times m}$ .

### 5.2.5 Transient CIS Image Formation

Unlike Velten et al. [149], in our implementation we do not measure the transient image directly. Instead, we build on-top of our work from the previous Chapter 3: using a standard CIS TOF sensor with a modulated illumination source, we obtain a sequence of modulated exposure measurements  $\mathbf{b}$  for different modulation frequencies and phases. Recalling Eq. (3.9), the transient image can then be recovered from  $\mathbf{b} = \mathbf{C}\mathbf{i} + \mathbf{n}$ , where the correlation matrix  $\mathbf{C}$  is obtained through a straightforward calibration step, and  $\mathbf{n}$  represents sensor noise. Substituting Eq. (5.8) for  $\mathbf{i}$ , we arrive at our full image formation model:

$$\mathbf{b} = \mathbf{C}\mathbf{P}\mathbf{v} + \mathbf{n} \quad (5.9)$$

## 5.3 Reconstruction from Diffuse Indirect Illumination

The image formation model from Eq. (5.9) cannot be inverted directly, due to noise, and since the light transport matrix  $\mathbf{P}$  is poorly conditioned, as is the correlation matrix  $\mathbf{C}$ ; see Section 3.3. We follow the Bayesian MAP approach from Section 2.8.3 to estimate the voxel grid. In particular, we use spatial priors on the voxel grid, and a likelihood approximating the noise distribution. Our approach is described in the following.

### 5.3.1 Inverse Problem

We solve the following MAP estimation problem

$$\mathbf{v}_{\text{opt}} = \underset{\mathbf{v}}{\operatorname{argmin}} \frac{1}{2} \|\mathbf{C}\mathbf{P}\mathbf{v} - \mathbf{b}\|_2^2 + \Gamma(\mathbf{v}), \quad (5.10)$$

where the first term is the Data Fidelity term. As in the previous chapter, this term results from a Gaussian likelihood, that is we assume  $\mathbf{n}$  to be Gaussian distributed.

*Priors* The second term implements the statistical priors on  $\mathbf{v}$  in our MAP framework. It is

$$\Gamma(\mathbf{v}) = \lambda \sum_z \|\nabla_{\mathbf{x}} \mathbf{v}_z\|_1 + \theta \|\mathbf{W}\mathbf{v}\|_1 + \sum_{x,y} \text{ind}_{\mathcal{C}}(\mathbf{v}_{x,y}) \quad (5.11)$$

We assume a sparse Laplacian distribution on the spatial gradients height field, leading to the first  $\ell_1$  penalty term on the spatial gradients for all volume depths  $z$ . Furthermore, we assume that the volume  $\mathbf{v}$  is sparsely populated, justified by our assumption of height field geometry. This second term is implemented as a weighted  $\ell_1$  norm of the volume itself. The weight matrix  $\mathbf{W}$  will be obtained using an iteratively reweighted  $\ell_1$  scheme (IRL1, see Section 5.3.3). Finally, the last term in Eq. (5.11) encodes the knowledge of the height field, by constraining the volume to have at most one non-zero entry for each 2D  $(x, y)$  coordinate. We encode this prior using a projection onto an indicator set of possible depth values for each  $(x, y)$  coordinate:

$$\text{ind}_{\mathcal{C}}(\mathbf{p}) = \begin{cases} 0 & \text{if } \mathbf{p} \in \mathcal{C} \\ \infty & \text{else} \end{cases} \quad \text{with} \quad (5.12)$$

$$\mathcal{C} = \{\mathbf{d} \in \mathbb{R}^z \mid \text{card}(\mathbf{d}) = 1 \wedge \mathbf{1}^T \mathbf{d} = \mathbf{1}^T \mathbf{p}\}$$

We note that the second and third term of the regularizer both have the purpose of encouraging a single surface reflection along the z-dimension of the reconstruction volume. The term from Eq (5.12) is stronger than the  $\ell_1$  regularization, since it prefers exactly a single-non-zero solutions (in contrast to just sparse solutions). On the other hand, it makes the overall optimization non-convex as  $\mathcal{C}$  is a non-convex set. So having both terms enables us to trade the convexity of our objective function for the sparsity of our model by adjusting the weight  $\theta$  from Equation 5.11.

### 5.3.2 Optimization

The optimization problem from Eq. (5.10) is of the form discussed in Section 2.8.3. In particular, we choose

$$\begin{aligned}\Phi &= \mathbb{I}, & G(\mathbf{p}) &= \frac{1}{2} \|\mathbf{C}\mathbf{P}\mathbf{p} - \mathbf{b}\|_2^2, \\ \Omega &= [\mathbf{D}_x^T, \mathbf{D}_y^T, \mathbf{W}\mathbb{I}^T, \mathbb{I}^T]^T, & F(\mathbf{p}) &= \lambda \|\mathbf{p}_{1,2}\|_1 + \theta \|\mathbf{p}_3\|_1 + \text{ind}_{\mathcal{C}}(\mathbf{p}_4),\end{aligned}\tag{5.13}$$

where  $\mathbf{D}_x, \mathbf{D}_y$  are derivative operators for the  $x, y$  dimensions for all  $z$  coordinates (stacked on-top of each other) and  $\mathbb{I}$  is the identity matrix. We have chosen  $F(\cdot): \Gamma(\mathbf{v}) = F(\Omega\mathbf{v})$ . The index identifies here the component stacked on-top of each other in  $\Omega\mathbf{v}$ . Note that the minimum of  $\Gamma(\mathbf{v})$  is obtained by independently minimizing  $F$  for each component of  $\Omega\mathbf{v}$ .

Eq. (5.13) represents our specific MAP estimation problem from Eq. (5.10) in the general linear MAP form from Eq. (2.40). Hence, it can be solved with a proximal algorithm as discussed in Section 2.8.4. We use a linearized version of ADMM [175]. Using  $\mathbf{v}$  as primal variable, the augmented Lagrangian from Eq. (2.42) becomes

$$\mathcal{L}_\rho(\mathbf{v}, \mathbf{y}, \lambda) = G(\Phi\mathbf{v}) + F(\mathbf{y}) + \lambda^T(\Omega\mathbf{v} - \mathbf{y}) + \frac{\rho}{2} \|\Omega\mathbf{v} - \mathbf{y}\|_2^2,\tag{5.14}$$

The ADMM method from Algorithm 1 performs three steps per iteration, each updating  $\mathbf{v}, \mathbf{y}, \lambda$  alternately. The individual steps are as follows:

**v-step** The update of the volume  $\mathbf{v}$  proceeds as follows:

$$\begin{aligned}\mathbf{v}^{k+1} &= \underset{\mathbf{v}}{\text{argmin}} \mathcal{L}_\rho(\mathbf{v}, \mathbf{y}^k, \lambda^k) \\ &= \underset{\mathbf{v}}{\text{argmin}} \frac{1}{2} \|\mathbf{C}\mathbf{P}\mathbf{v} - \mathbf{b}\|_2^2 + (\lambda^k)^T (\Omega\mathbf{v} - \mathbf{y}^k) + \frac{\rho}{2} \|\Omega\mathbf{v} - \mathbf{y}^k\|_2^2 \\ &\approx \underset{\mathbf{v}}{\text{argmin}} \frac{1}{2} \|\mathbf{C}\mathbf{P}\mathbf{v} - \mathbf{b}\|_2^2 + (\lambda^k)^T (\Omega\mathbf{v} - \mathbf{y}^k) + \\ &\quad \rho \left( \Omega^T \Omega \mathbf{v}^k - \Omega^T \mathbf{y}^k \right)^T \mathbf{v} + \frac{\mu}{2} \|\mathbf{v} - \mathbf{v}^k\|_2^2 \\ &= (\mathbf{P}^T \mathbf{C}^T \mathbf{C} \mathbf{P} + \mu \mathbb{I})^{-1} \left( \mathbf{P}^T \mathbf{C}^T \mathbf{b} + \mu \mathbf{v}^k - \rho \left( \Omega^T \Omega \mathbf{v}^k - \Omega^T \mathbf{y}^k \right) + \Omega^T \lambda^k \right).\end{aligned}\tag{5.15}$$

Note that in the third step we have made an approximation that linearizes the quadratic term from the second line in the proximity of the previous solution  $\mathbf{v}^k$ . This linearization approach is known under several different names, including Linearized ADMM or inexact Uzawa method (e.g. [175, 227, 228]). The additional parameter  $\mu$  satisfies the relationship  $0 < \mu \leq 1/(\rho\|\Omega\|_2^2)$ .

**y-step** The slack variable  $\mathbf{y}$  is updated as follows:

$$\begin{aligned} \mathbf{y}^{k+1} &= \underset{\mathbf{y}}{\operatorname{argmin}} \mathcal{L}_\rho(\mathbf{v}^{k+1}, \mathbf{y}, \lambda^k) \\ &= \underset{\mathbf{y}}{\operatorname{argmin}} F(\mathbf{y}) + (\lambda^k)^T \left( \Omega \mathbf{v}^{k+1} - \mathbf{y} \right) + \frac{\rho}{2} \left\| \Omega \mathbf{v}^{k+1} - \mathbf{y} \right\|_2^2 \quad (5.16) \\ &= \underset{\mathbf{y}}{\operatorname{argmin}} F(\mathbf{y}) + \frac{\rho}{2} \left\| \left( \Omega \mathbf{v}^{k+1} - \frac{\lambda^k}{\rho} \right) - \mathbf{y} \right\|_2^2 \end{aligned}$$

Both  $F(\cdot)$  and the least square term can be minimized independently for each component in  $\mathbf{y}$ . Using the slack variable  $\mathbf{y}$ , the minimization involving the difficult function  $F$  has now been turned into a sequence of much simpler problems in just a few variables. To derive the specific solutions to these problems, we note that the last line in Eq. (5.16) can be interpreted as a proximal operator:

$$\mathbf{y}^{k+1} = \mathbf{prox}_{(1/\rho)F} \left( \Omega \mathbf{v}^{k+1} - \frac{\lambda^k}{\rho} \right) \quad (5.17)$$

using the standard definition from Section 2.8.4. For our problem, we require the proximal operators for the  $\ell_1$  norm and for the indicator set. These are given as

$$\begin{aligned} \mathbf{prox}_{\gamma|\cdot|}(p) &= (p - \gamma)_+ - (-p - \gamma)_+ \\ \mathbf{prox}_{\gamma \operatorname{ind}_{\mathcal{C}}(\cdot)}(p) &= \Pi_{\mathcal{C}}(p) \end{aligned} \quad (5.18)$$

The first term is the well-known point-wise shrinkage [175] and the second is the projection on the set  $\mathcal{C}$ .

**$\lambda$ -step** The final step of the ADMM algorithm 1 is to update the Lagrange multiplier by adding the (scaled) error, that is  $\lambda^{k+1} := \lambda^k + \rho (\Omega \mathbf{v}^{k+1} - \mathbf{y}^{k+1})$ .

### 5.3.3 Enhancing Volume Sparsity

To further enhance the sparsity of the convex  $\ell_1$ -regularized part of our objective, we have placed a weight  $\mathbf{W}$  on the individual components of the  $\ell_1$  volume penalty (second term in Eq. (5.11)).

This approach has been proposed by [229]. The idea is that the weights  $\mathbf{W}$  capture the support of our sparse solution. This support is estimated iteratively from the last solution, which allows for improved recovery of the sparse non-negative components. As proposed in [229], we use the update rule

$$\mathbf{W}^{j+1} := \text{diag} \left( \frac{1}{|\mathbf{v}^j| + \epsilon} \right), \quad (5.19)$$

where the division is here point-wise, and  $\text{diag}(\cdot)$  denotes the diagonal matrix with the diagonal from the argument. The iteration variable  $j$  from above is for an *outer iteration on top of our original iteration* from Algorithm 1 with the steps from the previous paragraphs.

## 5.4 Implementation and Parameter Selection

*Parameters* For our specific Algorithm 1 with the steps from the previous paragraphs, we use the parameters  $\rho = 1.1$  and  $\mu = 0.5 \cdot 1 / (\rho \|\Omega\|_2^2)$ , which produced good results for all of our tested datasets. Note that  $\Omega$  changes for every outer IRL1 iteration, and thus  $\mu$  has to be recomputed for every iteration. We estimate  $\|\Omega\|_2^2$  by running the power method for  $\Omega^T \Omega$  with random initialization. We use 3 outer IRL1 iterations and an update weight of  $\epsilon = 0.1$ .

*Implementation of the  $\mathbf{v}$ -step* For a very high resolution sensor and reconstruction volume, storing  $\mathbf{P}$  would be infeasible. In this scenario one can implement  $\mathbf{P}$  as the procedural operator performing the transient light transport exactly as described in Section 5.2.3. The transient rendering operation parallelizes very well over each input pixel. One can implement its transpose  $\mathbf{P}^T$  similarly as the dot product of each transient image for a considered voxel accumulated over the whole voxel volume. Thus again only transient rendering and some additional dot-products are required.

Finally, the **v-step** from the presented linearized ADMM can be implemented using Conjugate Gradient (CG). Instead of applying explicit matrix multiplication inside CG, we replace each of the products with  $\mathbf{P}$  or  $\mathbf{P}^T$  with the operations defined above.

We implemented this version first. However, since our sensor only has a very low resolution of  $120 \times 160$ , we were actually able to fully precompute and efficiently store  $\mathbf{P}$  (in  $< 8\text{GB}$  RAM) as a sparse matrix which speeds up the reconstruction dramatically. Note that this approach would not be possible if the sensor or reconstruction resolution were significantly higher.

*Pre-factorization for Speedup* Instead of minimizing  $\|\mathbf{CP}\mathbf{v} - \mathbf{b}\|_2^2$  as a data term one can also pre-factor the optimization and first solve for a transient image  $\mathbf{C}^{-1}\mathbf{j}$  and then use this as an observation in the changed data term  $\|\mathbf{P}\mathbf{v} - \mathbf{C}^{-1}\mathbf{b}\|_2^2$ . We have used the **i**-subproblem from Section 3.3.3 to pre-factor the optimization and did not notice a strong difference in reconstruction quality in comparison to using the not pre-factored version. The advantage of pre-factoring is that the method gets sped up even more since all matrix application of  $\mathbf{C}$  have been handled before and  $\mathbf{C}$  itself can be inverted more efficiently than the full  $\mathbf{CP}$ .

## 5.5 Results

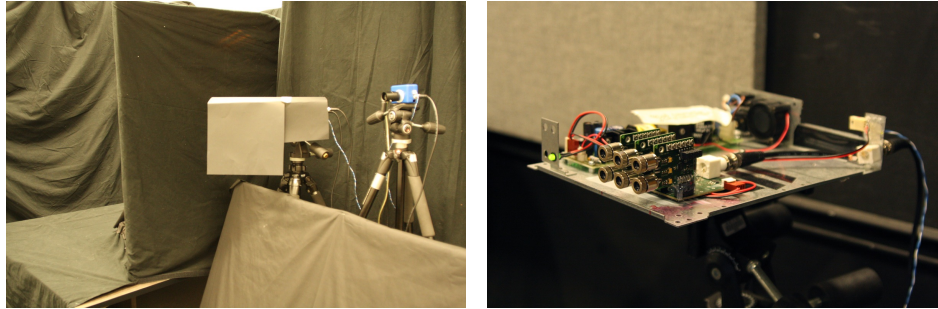
### 5.5.1 Setup

Our instrumentation comprises a modulated light source and a CIS detector, as used for the purpose of transient imaging in Chapter 3.

The detector is based on a filterless version of the TOF development kit CamBoard Nano by PMD Technologies, and extended with an external frequency-controllable modulation source (a workaround in lack of access to the FPGA configuration for the CamBoard). We determined that for our setup an integration time of 10 milliseconds delivers the optimal SNR, which we further improve by averaging over multiple measurements; see Section A.4 for details.

The light source consists of six 650 nm, 250 mW laser diodes with collimation optics and custom driving hardware to emit pulses of approximately 2-3 nanoseconds





**Figure 5.3:** Left: Photo of our capture setup facing the diffuse wall (light source covered with black photo board). To the left, behind an occluder, lies the reconstruction volume. Right: Close-up on the light source without cover.

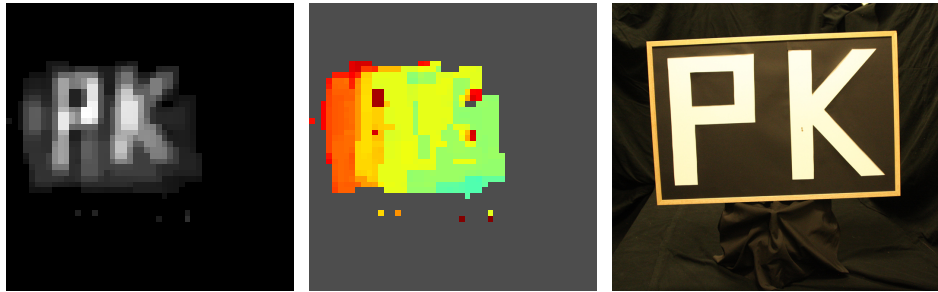
duration at variable repetition rate. The primary difference to the hardware setup from the previous chapter is that in the proposed setup, the diodes are not diffused to act as a spot light. Instead, we focus each laser diode with individual optics onto a single spot 1 on the wall (Figure 5.2, Figure 5.3). Their overall duty cycle during capture is less than 1%, allowing operation with only the lens holders doubling as heat sinks.

Our reconstruction volume has a size of  $1.5\text{ m} \times 1.5\text{ m} \times 2.0\text{ m}$  and is distanced 1.5 m from the flat, diffuse wall. The camera and illumination are about 2 m from the wall; please see Figure 5.2 for the exact spatial arrangement.

## 5.5.2 Qualitative Results

*Geometry* Our first test is to reconstruct the geometry of two letters cut out of cardboard that were painted with white color, and placed at different depths (Figure 5.1). We show two visualizations of the recovered depth information in the volume. In the center image we treat the voxels as an occupancy probability and we show the expected value of the distribution for each pixel, i.e. the sum of distances weighted by the occupancy probability.

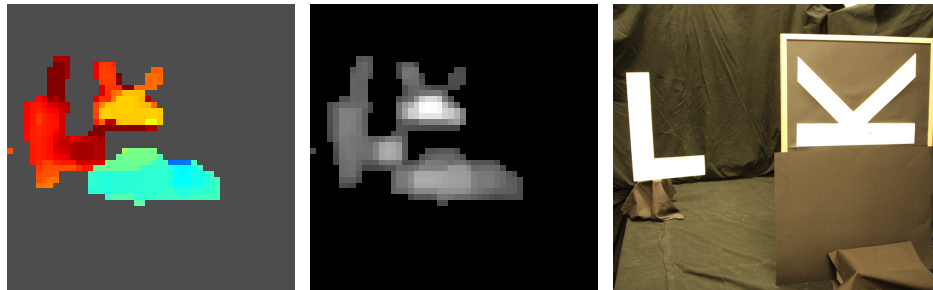
Since the expected value is not robust to outliers, we show in the right image the depth value with the strongest peak for each  $(x, y)$  pixel. This amounts to the voxel with the highest probability of occupancy in our reconstruction. Note, that



**Figure 5.4:** Albedo reconstruction example: Reconstruction of scene image with a flat surface but varying albedo (right). Center: the color-coded depth map of strongest peak along z-coordinate shows a flat geometry. Left: Albedo image at the depth map's depth.

here we threshold the volume, with very low albedo/occupancy probability as grey.

*Albedo* The next experiment Figure 5.4 shows the recovery of a spatially varying albedo on a flat surface. The color-coded depth map shows the depth of the strongest density in the reconstructed volume for each pixel  $(x, y)$  as before. The left of the figure shows the albedo  $v(\mathbf{x})$  sampled exactly at the depth map positions (the position of the strongest peak).



**Figure 5.5:** Simultaneous albedo and geometry: Reconstruction of scene image with varying albedo (letter on plane in the front) and varying depth for the letter in back (right). Albedo image, reconstruction value exactly at the depth position from the depth map (center). Color-coded depth map of strongest peak along z-coordinate (left).

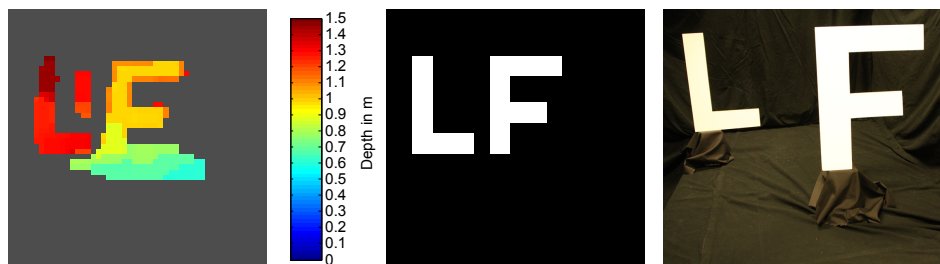
*Albedo and Geometry* Figure 5.5 shows an example of variation in both geometry and albedo. In this case, the planar surface in the front could not be reconstructed in the depth map due to the low albedo limiting the reflected light.

*Different Materials* In Section A.4 we show several examples of reconstructions with non-Lambertian surfaces. We find that Lambertian scenes result in very sparse volume reconstructions that clearly represent a height field structure. With increasingly non-Lambertian surfaces the energy is spread out more and more throughout the volume (as our model is violated).

*Effects of Ambient Light and Frame Averaging* One of the advantages of the method from the previous Chapter 3, is that it is rather insensitive to ambient illumination. We tested whether this robustness also applies to our approach for reconstructing geometry and albedo. Section A.4 presents results with strong ambient illumination which has only a minor effect on reconstruction quality.

### 5.5.3 Quantitative Results

To evaluate our reconstruction results, we compared the distance maps with manually measured scene geometry. Figure 5.6 shows a quantitative evaluation for the geometry reconstruction example shown above.



**Figure 5.6:** Quantitative evaluation: Reconstruction of scene image with letter "L" and "F" cut out of white painted cardboard (right). Color-coded depth map of strongest peak along z-coordinate visualized with color bar for depth in m (left).  $(x, y)$  ground truth geometry acquired from scene measurements (middle).

*Lateral Resolution* In order to evaluate the spatial resolution, we show an image of the measured scene geometry of the flat targets. The same discretization as for the shown depth map has been used. Having in mind that our reconstruction volume for all shown results had a size of  $1.5\text{ m} \times 1.5\text{ m} \times 2.0\text{ m}$  ( $x \times y \times z$ ), we see that we can achieve an  $(x, y)$  resolution of approximately  $\pm 5\text{ cm}$ . The accuracy of the reconstruction varies with different materials. Materials that have little or no overlap in the space-time profile (e.g. mirror example in Section A.4), allow for high reconstruction precision (around  $\pm 2\text{ cm}$  for the mirror example). The precision for more complex materials degraded to around  $\pm 15\text{ cm}$  tolerance. Overall the spatial resolution is limited by the low resolution of our sensor (which was only  $120 \times 160$  pixels).

Note that due to our robust measurement and reconstruction procedure we are able to achieve the shown results for significantly larger scenes than previously possible with the femtosecond laser approach demonstrated in [149]. Velten et al. report distances of up to  $25\text{ cm}$  from object to the wall and a reconstruction volume of  $(40\text{ cm})^3$  due to the low SNR for large distance bounces, whereas we demonstrate for the first time much larger room-sized environments.

*Depth Resolution* For the temporal resolution achieved in the above example of Figure 5.6, we see from the given color bar a depth distance of approximately  $0.6\text{ m}$ , where the measured distance was  $0.75\text{ m}$ . For all similarly diffuse materials we reach also roughly a tolerance of  $\pm 15\text{ cm}$ . For simple strong reflectors like the mirror we have less temporal superposition, so for the mirror example we obtain a high temporal resolution of below  $5\text{ cm}$  error in our  $2\text{ m}$  depth range, with more complex materials producing a precision of around  $\pm 20\text{ cm}$ .

As shown above the resolution of our approach depends on the scene content. The achievable resolution should in the future scale linearly with the availability of higher resolution TOF cameras, such as the upcoming Kinect<sup>®</sup> 2 as mentioned in Section 3.7. We have shown that our method degrades somewhat gracefully with using different materials, although a certain scene dependence is inherent in the non-linear nature of the inverse problem we solve.

## 5.6 Discussion

In this chapter, we have presented a method for reconstructing hidden, i.e. Non-Line-of-Sight, geometry and albedo values from transient images of diffuse reflections. This approach involves hard inverse problems that can only be solved using statistical priors such as sparsity in the geometry, and our primary contribution is to identify a linearized image formation model, regularization terms, and corresponding numerical solvers to recover geometry and albedo under this difficult scenario.

Despite these numerical challenges, we show that our method can be combined with the work from the previous Chapter 3 on transient imaging using inexpensive CIS cameras, which itself involves a hard inverse problem. We demonstrate that it is possible to combine these two inverse problems and solve them jointly in a single optimization method. As a result our approach has several advantages over previous methods employing femtosecond lasers and streak cameras [149]. These include a) low hardware cost, b) no moving parts and simplified calibration, c) capture times that are reduced from hours to minutes, and d) robustness under ambient illumination in large room-sized environments. We believe that, as a result, our method shows great promise for applications of indirect geometry reconstruction outside the lab.

## Chapter 6

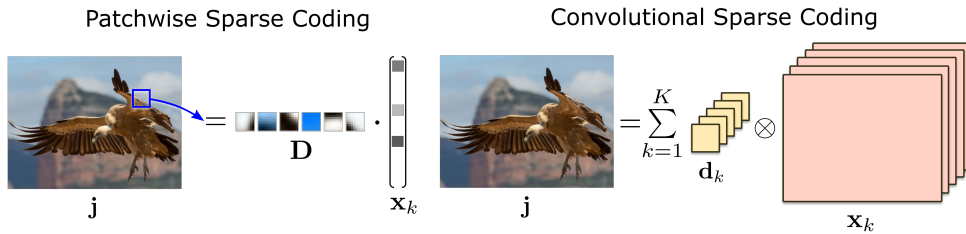
# Convolutional Sparse Coding

Convolutional Sparse Coding (CSC) has enabled exciting applications in computer vision and machine learning. The previous Chapters 3, 4 and 5 have demonstrated that physically-motivated convolutional codes allow the recovery of high-quality transient images from CIS measurements, enabling imaging of objects outside of the line of sight and imaging in scattering and turbid media. Going beyond reconstruction tasks, CSC can even serve as a strategy for unsupervised features learning, when a simple physical model cannot explain the data. In particular, it allows learning features of natural images, that subsequently are used for classification and reconstruction tasks. As opposed to patch-based methods, convolutional sparse coding operates on whole images, thereby seamlessly capturing the correlation between local neighborhoods.

In this chapter, we propose a new approach to solving CSC problems, both for learning and reconstruction, and show that our method converges significantly faster and also finds better solutions than the state of the art. In addition, the proposed method is the first efficient approach to allow for proper boundary conditions to be imposed and it also supports feature learning from incomplete data as well as general reconstruction problems.

### 6.1 Introduction

An increasing number of computer vision tasks rely on image statistics as a key prior in Bayesian inference Section 2.8. Low-level problems that benefit from



**Figure 6.1:** Patchwise Sparse Coding (left) divides the data into small patches of a given size (blue). Every patch is modeled as the superposition of a sparse set of learned dictionary atoms. The patchwise representation, however, contains shifted version of the same feature (third and last atom here). Convolutional Sparse Coding (right) finds a more compact representation by exploiting this convolutional structure in the data, and models it as a sum of sparsely-distributed convolutional features.

good priors include inpainting, denoising, deblurring, and super-resolution, while recognition, classification and other higher-level tasks often use learned features as priors for natural images. In this chapter, we revisit one strategy for unsupervised learning of image features: Convolutional Sparse Coding (CSC). CSC was introduced in the context of modeling receptive fields in human vision [230], but it has recently been demonstrated to have important applications in a wide range of computer vision problems such as low/mid-level feature learning, low-level reconstruction [231, 232], as part of more complex hierarchical structures or networks in high-level computer vision challenges [220, 233, 234], and in physically-motivated computational imaging problems as the one from the Chapter 4. Beyond these applications, CSC could find applications in many other reconstruction tasks and feature-based methods, including deblurring, denoising, inpainting, classification, localization, and tracking. While we focus on 2D image features in this chapter, our method straightforwardly generalizes to higher dimensional data, such as videos, and directly applies to 1D signals, such as the transient profiles discussed in the previous chapters.

CSC is closely related to popular patch-based learning and reconstruction methods [163, 235, 236]. However, features learned with patch-based methods often contain shifted versions of the same features and latent structures of the underlying signal may be lost when dividing it into small patches. See Figure 6.1 on the left for

an illustration. A more elegant way to model many of these problems is to use a sum of sparsely-distributed convolutional features; see Figure 6.1 on the right. The main drawback of convolutional models, however, is their computational complexity. Not only is it very challenging to find a solution to convolutional sparse coding problems in a reasonable amount of time, but finding a good local minimum is difficult as well. Generally, CSC for feature learning is a non-convex problem and many existing methods provide little to no guarantees for global convergence. Seminal advances in fast convolutional sparse coding have recently shown that feature learning via CSC can be efficiently solved in the frequency domain. Grosse et al. [237] were the first to propose a frequency domain method for 1D audio signals, while [238–240] later demonstrate efficient frequency domain approaches for 2D image data. While this is the first step towards making CSC practical, these frequency methods can introduce boundary artifacts for both learning and reconstruction [233] and, as inherently global approaches, make it difficult to work with incomplete data.

Building on recent advances in optimization [174, 175, 241–243], we propose a new splitting-based approach to convolutional sparse coding. We not only show that our formulation allows us to easily incorporate proper boundary conditions and learn from sparse observations, but we also demonstrate that the proposed method is faster and converges to better solutions than the state of the art.

In the following, we first derive a flexible formulation of the convolutional sparse coding problem, and an efficient solution by splitting the objective into a sum of simple, convex functions. This formulation fits into most recent proximal optimization frameworks. Following, we demonstrate that the proposed method allows for proper boundary conditions to be imposed without sacrificing performance; it converges faster than alternative methods and finds better solutions. The latter is verified using several low-level reconstruction problems. Finally, we show that our flexible formulation allows feature learning from incomplete observations, and yields an efficient solver when working with known features, such as the exponentially modified Gaussians from Chapter 4.



## 6.2 Mathematical Framework

Traditionally, convolutional sparse coding problems are expressed in the form of the following Eq. (6.1). See again Figure 6.1 on the right for an illustration.

$$\begin{aligned} \operatorname{argmin}_{\mathbf{d}, \mathbf{x}} \quad & \frac{1}{2} \|\mathbf{j} - \sum_{k=1}^K \mathbf{d}_k \otimes \mathbf{x}_k\|_2^2 + \beta \sum_{k=1}^K \|\mathbf{x}_k\|_1 \\ \text{subject to} \quad & \|\mathbf{d}_k\|_2^2 \leq 1 \quad \forall k \in \{1, \dots, K\}, \end{aligned} \quad (6.1)$$

where  $\mathbf{x}_k$  are sparse feature maps that approximate the data term  $\mathbf{j}$  when convolved with the corresponding filters  $\mathbf{d}_k$  of fixed spatial support. Here  $\mathbf{j} \in \mathbb{R}^D$ ,  $\mathbf{x}_k \in \mathbb{R}^D$  are vectorized images,  $\mathbf{d}_k \in \mathbb{R}^M$  are the vectorized 2D filters,  $k = 1 \dots K$ , and  $\otimes$  is the 2D convolution operator defined on the vectorized inputs. While the above equation is strictly only valid for a single target image, it can easily be generalized to multiple images  $\mathbf{j}$ .

Bristow et al. [238, 239] have shown remarkable improvements in efficiency by exploiting Parseval’s theorem for solving Eq. (6.1), which states that the energy of a signal is equivalent — up to a constant — to that of its Fourier transform. We will neglect this constant in the following. Eq. (6.1) can therefore be reformulated [238–240] as

$$\begin{aligned} \operatorname{argmin}_{\mathbf{d}, \mathbf{x}} \quad & \frac{1}{2} \|\hat{\mathbf{j}} - \sum_{k=1}^K \hat{\mathbf{d}}_k \odot \hat{\mathbf{x}}_k\|_2^2 + \beta \sum_{k=1}^K \|\mathbf{t}_k\|_1 \\ \text{subject to} \quad & \|\mathbf{s}_k\|_2^2 \leq 1 \quad \forall k \in \{1, \dots, K\} \\ & \mathbf{s}_k = \mathbf{S}\Phi^T \hat{\mathbf{d}}_k \quad \forall k \in \{1, \dots, K\} \\ & \mathbf{t}_k = \mathbf{x}_k \quad \forall k \in \{1, \dots, K\}, \end{aligned} \quad (6.2)$$

which expresses the computationally expensive convolution operations as more efficient multiplications in the frequency domain. Here,  $\hat{\cdot}$  denotes the frequency representation of a signal,  $\odot$  is the component-wise product,  $\Phi$  is the discrete Fourier transform (DFT) matrix, and  $\mathbf{S}$  projects a filter onto its (small) spatial support. The slack variables  $\mathbf{s}_k$  and  $\mathbf{t}_k$  allow Eq. (6.2) to be solved by splitting the objective into multiple subproblems that each can be solved efficiently.

It is important to note that Eqs. (6.1) and (6.2) are actually only equivalent under the assumption of circular boundary conditions [220]. Kavukcuoglu et al. [233]

point out that boundary conditions are one essential hurdle in the convolutional model that affects the optimization even for non-circular boundary conditions, because pixels close to the boundary are, in general, covered by fewer filters than center pixels. While heuristics [238] might be acceptable for learning filters with very small spatial support, this assumption does not necessarily hold for larger filters or for general reconstruction problems. We propose the following, general formulation for convolutional sparse coding:

$$\begin{aligned} \operatorname{argmin}_{\mathbf{d}, \mathbf{x}} \quad & \frac{1}{2} \|\mathbf{j} - \mathbf{M} \sum_{k=1}^K \mathbf{d}_k \otimes \mathbf{x}_k\|_2^2 + \beta \sum_{k=1}^K \|\mathbf{x}_k\|_1 \\ \text{subject to} \quad & \|\mathbf{d}_k\|_2^2 \leq 1 \quad \forall k \in \{1, \dots, K\}. \end{aligned} \quad (6.3)$$

Here,  $\mathbf{M}$  is a diagonal or block-diagonal matrix, such that it decouples linear systems of the form  $(\mathbf{M}^T \mathbf{M} + \mathbb{I})\mathbf{x} = \mathbf{b}$  into many small and independent systems that are efficiently solved. For example, for boundary handling  $\mathbf{M}$  can be a binary diagonal matrix that masks out the boundaries of the padded estimation  $\sum_{k=1}^K \mathbf{d}_k \otimes \mathbf{x}_k$ . This allows us to use unmodified filters in boundary regions, without requiring circular boundaries or other conditions. Furthermore, we show that  $\mathbf{M}$  allows for efficient learning and reconstruction from incomplete data.

Unfortunately, Eq. (6.3) cannot be solved directly with the “Fourier trick” discussed in the literature (Eq. (6.2)). In the following, we derive a formulation that is not only *flexible* in allowing us to solve Eq. (6.3) efficiently, but we also show that our formulation solves the conventional convolutional sparse coding problem (Eq. (6.1)) *faster* than previous methods and *converges to better solutions*.

### 6.2.1 Efficient Splitting of the Objective

To efficiently solve Eq. (6.3), we reformulate it such that the constraints are included in the objective via an indicator function  $\operatorname{ind}_C(\cdot)$ . This indicator function is defined on the convex set of the constraints  $C = \{\mathbf{v} \mid \|\mathbf{S}\mathbf{v}\|_2^2 \leq 1\}$ . With the constraints encoded in the indicator penalty, we get the following unconstrained objective:

$$\operatorname{argmin}_{\mathbf{d}, \mathbf{x}} \quad \frac{1}{2} \|\mathbf{j} - \mathbf{M} \sum_{k=1}^K \mathbf{d}_k \otimes \mathbf{x}_k\|_2^2 + \beta \sum_{k=1}^K \|\mathbf{x}_k\|_1 + \sum_{k=1}^K \operatorname{ind}_C(\mathbf{d}_k), \quad (6.4)$$

which can be expressed as the following sum of functions

$$\begin{aligned} \operatorname{argmin}_{\mathbf{d}, \mathbf{x}} f_1(\mathbf{D}\mathbf{x}) + \sum_{k=1}^K (f_2(\mathbf{x}_k) + f_3(\mathbf{d}_k)), \quad \text{with} \quad (6.5) \\ f_1(\mathbf{v}) = \frac{1}{2} \|\mathbf{j} - \mathbf{M}\mathbf{v}\|_2^2, \quad f_2(\mathbf{v}) = \beta \|\mathbf{v}\|_1, \quad f_3(\mathbf{v}) = \operatorname{ind}_C(\mathbf{v}). \end{aligned}$$

Here,  $\mathbf{x} = [\mathbf{x}_1^T \dots \mathbf{x}_K^T]^T$  and  $\mathbf{D} = [\mathbf{D}_1 \dots \mathbf{D}_K]$  is a concatenation of Toeplitz matrices, each one representing a convolution with the respective filter  $\mathbf{d}_k$ . Eq. (6.5) consists of a sum of functions  $f_i$ , which are all simple to optimize individually, whereas their sum is challenging. Following [242], we define  $f_1$  with  $\mathbf{M}$  included because that splits the data term into two different subproblems involving  $\mathbf{M}$  and  $\mathbf{D}$  *separately but never jointly*.

## 6.2.2 Generalization of the Objective

To derive this result more intuitively, we consider the general objective from (6.5)

$$\operatorname{argmin}_{\mathbf{x}} \sum_{i=1}^I f_i(\mathbf{K}_i \mathbf{x}), \quad (6.6)$$

where  $\mathbf{K}_i : \mathbb{R}^{b_i \times a_i}$  are arbitrary matrices,  $f_i : \mathbb{R}^{b_i} \rightarrow \mathbb{R}$  are closed, proper, convex functions, and  $i \in \{1, \dots, I\}$ , such that  $f_i(\mathbf{K}_i \cdot) : \mathbb{R}^{a_i} \rightarrow \mathbb{R}$ ;  $I$  is the number of functions in the sum.

Eq. (6.6) is motivated by recent work in image deconvolution [8, 242], which have a similar objective that consists of a sum of simple convex functions. The problem in Eq. (6.6) can be reformulated as

$$\begin{aligned} \operatorname{argmin}_{\mathbf{x}} \sum_{i=1}^I f_i(\mathbf{K}_i \mathbf{x}) = f(\mathbf{K}\mathbf{x}), \quad \text{with} \\ \mathbf{K} = \begin{bmatrix} \mathbf{K}_1 \\ \vdots \\ \mathbf{K}_I \end{bmatrix} \quad \text{and} \quad f(\mathbf{v}) = \sum_{i=1}^I f_i(\mathbf{v}_i), \quad (6.7) \end{aligned}$$

where  $\mathbf{v}_i$  selects the  $i$ -th support of  $\mathbf{v}$ . Using a formulation with the stacked matrix

$\mathbf{K}$  allows us to remap Eq. (6.6) to existing proximal algorithms, such as ADMM, which have been discussed in detail in Section 2.8.4.

### Optimization

Eq. (6.7) is an optimization problem of the form presented in Section 2.8.3. We set  $G = 0$ ,  $\Omega = \mathbf{K}$  and  $F = f$  to express our problem in the general form. This setting may seem unintuitive, but will become clear after deriving our solver method below. In particular, the general ADMM method 1 becomes now Algorithm. 3, with  $\mathbf{x}$  as primal variable. We observe that the resulting minimization becomes separable in

---

**Algorithm 3** ADMM for a sum of functions in Eq. (6.7)

---

```

1: for  $k = 1$  to  $V$  do
2:    $\mathbf{x}^{k+1} = \underset{\mathbf{x}}{\operatorname{argmin}} \|\mathbf{K}\mathbf{x} - \mathbf{y} + \lambda^k\|_2^2$ 
3:    $\mathbf{y}_i^{k+1} = \underset{\rho}{\operatorname{prox}}_{f_i}(\mathbf{K}_i\mathbf{x}_i^{k+1} + \lambda_i^k)$ 
4:    $\lambda^{k+1} = \lambda^k + (\mathbf{K}\mathbf{x}^{k+1} - \mathbf{y}^{k+1})$ 
5: end for

```

---

all the  $f_i$ . Since  $\mathbf{M}$  is included in  $f_1$ , we can solve the update in line 2 of Algorithm 3 efficiently in the Fourier domain. Note that the combination of splitting  $\mathbf{M}$  from the filter generation  $\mathbf{D}\mathbf{x}$  via  $f_1(\mathbf{D}\mathbf{x})$  leads to a non-standard application of ADMM; the standard approach from Section 2.8.4 would set  $G(\mathbf{x}) = \frac{1}{2}\|\mathbf{j} - \mathbf{M}\mathbf{D}\mathbf{v}\|_2^2$  as the data fidelity, with  $\mathbf{M}\mathbf{D}$  as a single operator.

In the following, we will describe each subproblem of Algorithm 3. Note, that although we derive an ADMM method for solving Eq. (6.7), this is equally possible for other proximal algorithms, that have been listed in Section 2.8.4. In Chapter 10, we rely on a general formulation for image optimization problems, similar to Eq. (6.6), as the basis for a corresponding domain-specific language, that allows to “compile” our optimization problem into proximal algorithms with different splittings of the objective into  $G, F$ .

**x-step** Solving the first quadratic subproblem from Algorithm 3 (line 2) gives

$$\mathbf{x}_{\text{opt}} = \underset{\mathbf{x}}{\operatorname{argmin}} \|\mathbf{K}\mathbf{x} - \xi\|_2^2 = (\mathbf{K}^T\mathbf{K})^{-1}(\mathbf{K}^T\xi) \quad (6.8)$$

Here, we have set  $\xi = \mathbf{y} - \lambda^k$  as a notational shortcut. Depending on whether we solve for the filters (i.e.  $\mathbf{x} = \mathbf{d}$ ) or for the feature maps (i.e.  $\mathbf{x} = \mathbf{x}$ ), we get:

$$\begin{aligned} \mathbf{d}_{\text{opt}} &= (\mathbf{X}^\dagger \mathbf{X} + 2\mathbb{I})^{-1} (\mathbf{X}^\dagger \xi_1 + \xi_2 + \xi_3) & \text{for } \mathbf{x} = \mathbf{d} \\ \mathbf{x}_{\text{opt}} &= (\mathbf{D}^\dagger \mathbf{D} + \mathbb{I})^{-1} (\mathbf{D}^\dagger \xi_1 + \xi_2) & \text{for } \mathbf{x} = \mathbf{x} \end{aligned} \quad (6.9)$$

Here  $\mathbf{X}$  is a concatenation of Toeplitz matrices for the respective sparse codes  $\mathbf{x}_k$  and  $\xi_i$  selects again the  $i$ -th support of  $\tau$  as defined for Eq. (6.7). The operator  $\cdot^\dagger$  defines here the conjugate transpose, with notation borrowed from [240]. In both cases, one can find a variable reordering for the equations systems in Eq. (6.9), that makes  $(\mathbf{X}^\dagger \mathbf{X} + 2\mathbb{I})$  and  $(\mathbf{D}^\dagger \mathbf{D} + \mathbb{I})$  block-diagonal [238, 240], which makes the inversion efficient by parallelization over the  $j \in \{1 \dots D\}$  different blocks. The inverse can be efficiently computed for each block  $j$  using the Woodbury formula, giving

$$\begin{aligned} (\mathbf{X}_j^\dagger \mathbf{X}_j + 2\mathbb{I})^{-1} &= \frac{1}{2}\mathbb{I} - \frac{1}{2}\mathbf{X}_j^\dagger (2\mathbb{I} + \mathbf{X}_j \mathbf{X}_j^\dagger)^{-1} \mathbf{X}_j \\ (\mathbf{D}_j^\dagger \mathbf{D}_j + \mathbb{I})^{-1} &= \mathbb{I} - \frac{\mathbf{D}_j^\dagger \mathbf{D}_j}{1 + \mathbf{D}_j \mathbf{D}_j^\dagger}, \end{aligned} \quad (6.10)$$

where the second equation holds, since a block in  $\mathbf{D}_j$  is just a row-vector. We compute the first inverse  $(2\mathbb{I} + \mathbf{X}_j \mathbf{X}_j^\dagger)^{-1}$  by computing its Cholesky factorization. In contrast to the direct inversion in [239] (due to the code update, this has to be done in each iteration of their method), caching this factorization leads to a significantly decreased running time as described below.

**y-step** The proximal operators for Algorithm 3 (line 3) are simple to derive and well known in literature [175]:

$$\begin{aligned} \mathbf{prox}_{\theta f_1}(\mathbf{v}) &= (\mathbb{I} + \theta \mathbf{M}^T \mathbf{M})^{-1} (\mathbf{v} + \theta \mathbf{M}^T \mathbf{j}) & \text{Quadratic} \\ \mathbf{prox}_{\theta f_2}(\mathbf{v}) &= \max \left( 1 - \frac{\theta \beta}{|\mathbf{v}|}, 0 \right) \odot \mathbf{v} & \text{Shrinkage} \\ \mathbf{prox}_{\theta f_3}(\mathbf{v}) &= \begin{cases} \frac{\mathbf{Sv}}{\|\mathbf{Sv}\|_2} & : \|\mathbf{Sv}\|_2^2 \geq 1 \\ \mathbf{Sv} & : \text{else} \end{cases} & \text{Projection} \end{aligned}$$

where the inverse in  $\mathbf{prox}_{\theta f_1}$  is cheap to evaluate as  $\mathbf{M}$  is usually (block-)diagonal.

### 6.2.3 Biconvex Minimization via Coordinate Descent

Above, we have described Algorithm 3, which can solve the bi-convex problem (6.3) for  $\mathbf{x}$  or  $\mathbf{d}$  when the respective other variable is fixed. To jointly solve for both, we follow the standard approach of alternating between them, yielding Algorithm 4.

---

#### Algorithm 4 CSC learning using coordinate descent

---

- 1: Algorithm penalty parameters:  $\rho_{\mathbf{d}} \in \mathbb{R}^+, \rho_{\mathbf{x}} \in \mathbb{R}^+$
  - 2: Initialize variables:  $\mathbf{d}^0, \mathbf{x}^0, \lambda_{\mathbf{d}}^0, \lambda_{\mathbf{x}}^0$
  - 3: **repeat** {Outer iterations}
  - 4:     **Kernel update:** Solve Eq. (6.5) w.r.t.  $\mathbf{d}$ :  
 $\mathbf{d}^i, \lambda_{\mathbf{d}}^i \leftarrow \operatorname{argmin}_{\mathbf{d}} f_1(\mathbf{X}\mathbf{d}) + \sum_{k=1}^K f_3(\mathbf{d}_k)$  using  
 Alg. 3 with  $\rho = \rho_{\mathbf{d}}, \lambda = \lambda_{\mathbf{d}}^{i-1}$
  - 5:     **Code update:** Solve Eq. (6.5) w.r.t.  $\mathbf{x}$ :  
 $\mathbf{x}^i, \lambda_{\mathbf{x}}^i \leftarrow \operatorname{argmin}_{\mathbf{x}} f_1(\mathbf{D}\mathbf{x}) + \sum_{k=1}^K f_2(\mathbf{x}_k)$  using  
 Alg. 3 with  $\rho = \rho_{\mathbf{x}}, \lambda = \lambda_{\mathbf{x}}^{i-1}$
  - 6: **until** No more progress in both directions.
- 

With this alternating approach, we have constructed a coordinate descent for  $\mathbf{x}$  and  $\mathbf{d}$ . The individual Lagrange multipliers are initialized with the ones from the previous iteration. In practice, we run each of the two sub-routines until sufficient progress has been made. The step-size of the coordinate descent is defined by the progress each local optimization makes. Using a constant number of  $P$  iterations for each substep gave us a sufficiently good performance. We stop the algorithm when none of the two optimizations can further decrease the objective; a local minimum is found. It also follows that our algorithm monotonically decreases the objective for its iteration sequence  $\mathbf{d}^i, \mathbf{x}^i$ .

### 6.2.4 Implementation details

For the objective in Eq. (6.3), we found that the parameter  $\beta = 1$  delivers a good tradeoff between sparsity and data fit. All results in this chapter are computed with this setting. We have verified that other settings of  $\beta$  lead to quantitatively similar results. For Algorithm 4, we have chosen the heuristic values  $\rho_{\mathbf{d}} = 1/(100 \cdot \max(\mathbf{j}))$  and  $\rho_{\mathbf{x}} = 1/(10 \cdot \max(\mathbf{j}))$ . This choice dampens variations in the optimization path of the filters more than for the codes.

## 6.3 Analysis

### 6.3.1 Complexity Analysis

This section analyzes the complexity of the proposed optimization approach and compares the theoretical runtime with alternative methods. For  $D$  being the number of pixels for a single image in  $\mathbf{j}$ ,  $N$  being the number of images,  $K$  being the number of kernels to learn,  $M$  being the size of the filter support, and  $P$  inner iterations (of the substeps in Algorithm 4), the computation cost is shown in Table 6.1.

Method	Cost (in flops)
Zeiler et al. [220]	$PN \cdot ( \underbrace{KD}_{\text{Conjugate gradient}} \cdot \underbrace{KDM}_{\text{Spatial convolutions}} + \underbrace{KD}_{\text{Shrinkage}} )$
Bristow et al. [238, 239]	$PN \cdot ( \underbrace{K^3D}_{\text{Linear systems}} + \underbrace{KD \log(D)}_{\text{FFTs}} + \underbrace{KD}_{\text{Shrinkage}} )$
Ours ( $K > N$ )	$\underbrace{KN^2D + (P-1)KND}_{\text{Linear systems}} + PN \cdot ( \underbrace{KD \log(D)}_{\text{FFTs}} + \underbrace{KD}_{\text{Shrinkage}} )$
Ours ( $K \leq N$ )	$\underbrace{K^3D + (P-1)K^2D}_{\text{Linear systems}} + PN \cdot ( \underbrace{KD \log(D)}_{\text{FFTs}} + \underbrace{KD}_{\text{Shrinkage}} )$

**Table 6.1:** Time complexity of our approach compared to other methods.

We observe immediately that Bristow’s method has significantly better performance than Zeiler et al. when  $K < DM$ . Its dominating cost is the inversion of the  $D$  linear systems. Note that in contrast to spatial methods, Bristow’s method, as well as ours, is independent of the filter size  $M$ .

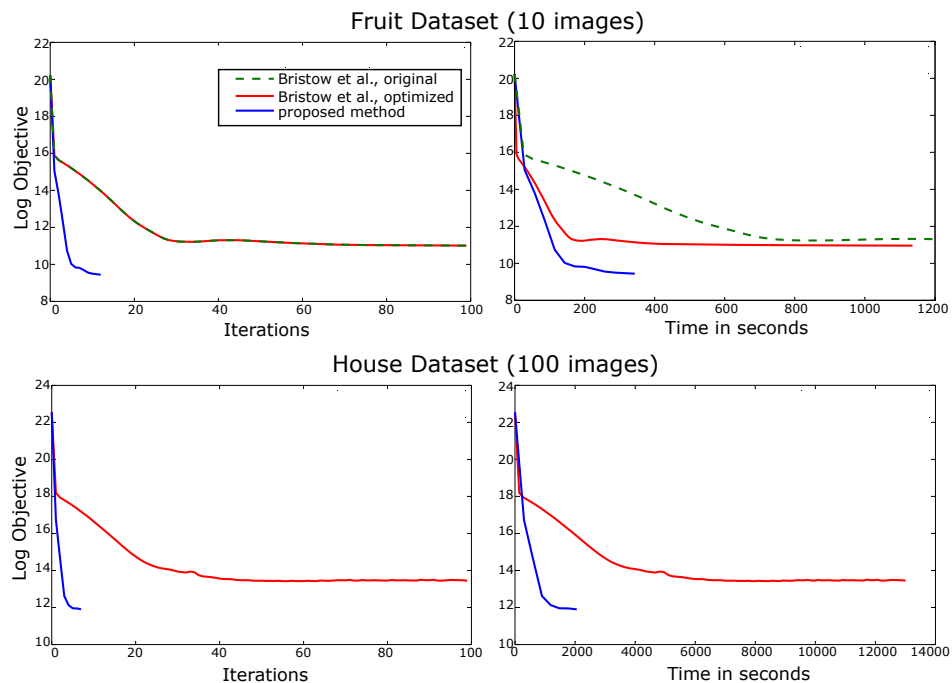
For the proposed method, we consider two cases:  $K > N$  and  $K \leq N$ . In the first case ( $K > N$ ), we exploit the inversion trick as explained in Eq. (6.10). Here, each of the  $D$  matrices  $\mathbf{X}_j$  is an  $N \times K$  matrix. Thus, by using Eq. (6.10), we reduce the cost of the inverse from  $K^3$  to  $KN^2$ . Since we cache the factorizations, this cost is only for one of the  $P$  local iterations. For the other  $(P-1)$  iterations, the back-solves cost only  $KN$  (instead of  $K^2$  for the naive inversion).

In the second case, when  $K \leq N$ , we have the full cost of the Cholesky factorization  $K^3$  of the  $D$  matrices  $\mathbf{X}_j$  once per  $P$  iterations, but again for all other  $(P-1)$  iterations, only the back-solve cost  $K^2$ . Thus, by caching the factorizations, we are able to achieve a speedup of the linear systems by  $\frac{P}{1+(P-1)/K}$  in this case.

For our setting of  $P = 10$ , even a moderate number of  $K = 100$  filters leads to a speedup of  $9\times$ . In the following, we show that not only the complexity per iteration decreases, but the convergence improves as well.

### 6.3.2 Convergence

For two datasets of different size, we plot the empirical convergence of the proposed algorithm and compare it to the state of the art in Figure 6.2. In both cases we learn  $K = 100$  filters. The first dataset is the fruit datasets [220] with  $N = 10$  images. In this case, we have  $K > N$ . The proposed algorithm converges in 13 iterations whereas [238, 239] has a slowly decreasing objective and was fully converged after about 300 iterations. To be able to compare objective values, all compared methods here are implemented using circular convolution, with edge-tapering applied to make the convolution circular. Only the central part of the convolution without



**Figure 6.2:** Convergence for two datasets (top  $N = 10$  images, bottom  $N = 100$ ). The proposed algorithm converges to a better solution in less time than competing methods.

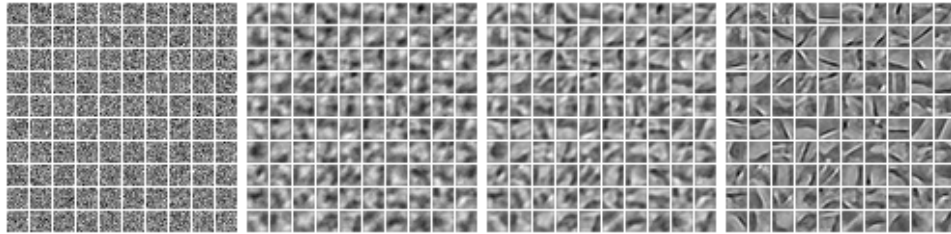


a padding of kernel size  $M$  is included in the objective values. Note that the solution of our algorithm is not only found in fewer iterations, but it also converges to a solution that has a lower objective. The objective combines the data fitting term as well as the sparsity term. We used the same sparsity weighting (from [238, 239]) for the comparison, although the same qualitative convergence was measured for different weights. We also plot the convergence in an absolute time scale demonstrating the achieved speedup. We have further compared our method to [238, 239] with our factorization strategy from the first line in Eq (6.10). While this strategy improves the speed of their method since the inverses in each iteration are done more efficiently, the performance does not reach that of our method.

For the second dataset, we have randomly sampled  $N = 100$  images of size  $100 \times 100$  from a city scene. In this case  $K \leq N$  and  $N, K$  are large. Thus, solving the linear systems becomes the dominating cost (see Table 6.1) and the benefit of caching (which cannot be done in [238, 239]) becomes even more apparent. Hence, especially for large datasets, our method converges faster and usually finds a better optimum. Note that we are already comparing to [238, 239] with our improved factorization strategy from the first line in Eq (6.10).

We also compare convolutional coding to patch-based sparse coding. One of the main challenges are large datasets, for which most patch-based methods become infeasible. Consider learning from 10 images with  $1000 \times 1000$  pixels each. Incorporating all patches into the learning requires 10 million training patches. K-SVD, for example, could not handle that much data on our computer, so we ran a comparison for 10  $100 \times 100$  pixel images. Using all patches in the training set and 100 iterations, K-SVD took 13.1 *hours* to converge whereas our method only took about 4.5 *minutes* (both on an Intel Xeon E5/Core i7 machine with 132 GB RAM).

In addition to the convergence analysis above, we also show the evolution of the filters throughout the learning process in Figure 6.3 for the fruit dataset. Initially, the filters seem random and then turn into Gaussian blob-like structures after a few iterations. After about 8 iterations, the observed structures are very similar to those frequently occurring in patch-based dictionary learning methods, whereas the filters eventually converge to Gabor-like shapes.



**Figure 6.3:** Visualization of filters learned from fruit dataset after 1, 7, 8, and 13 iterations. The evolution goes from random patterns over Gaussian blob-like structures and eventually converges to filters that bear similarity to Gabor patches.

## 6.4 Learning

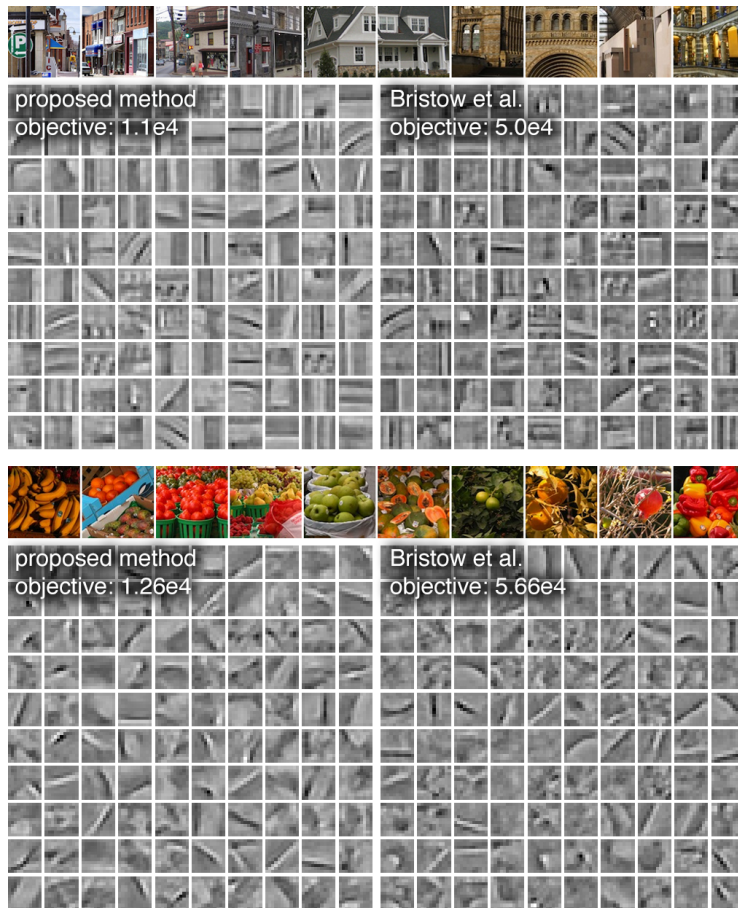
We trained our filters on the fruit and city datasets [220] with local contrast normalization applied. Figure 6.4 shows the resulting filters after convergence (ours after 13 iterations, Bristow after 300 iterations). Although the filters look similar at first glance, our results contain fewer dataset-specific features, which makes them more general as we demonstrate in Section 6.5.1.

### 6.4.1 Boundary Conditions

Eq. 6.3 is an elegant formulation that allows for general boundary conditions to be integrated into the learning and also the reconstruction steps. Usually, we mask out the boundary so that it does not contribute to the objective function. As seen in Figure 6.5, the boundary is still reconstructed via extrapolated data fitting — lines and other high-frequency structures continue across the image boundary but quickly fall off thereafter.

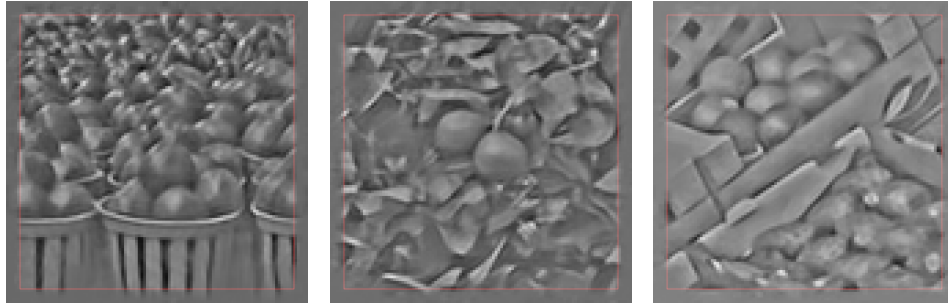
### 6.4.2 Learning from Sparse Data

The mixing matrix  $\mathbf{M}$  in Eq. 6.3 not only allows for general boundary constraints but for any type of linear operator to be applied. In Figure 6.6, we demonstrate that this can be used for learning filters from incomplete data. We subsequently use the filters learned from incomplete measurements of an image to predict the missing parts of that image and evaluate the achieved peak signal-to-noise ratio for



**Figure 6.4:** Filters learned on the city and fruit datasets [220]. We show thumbnails of the datasets along with filters learned with the proposed method (left) and with that described in [238, 239]. In both cases, our method finds a local optimum with an objective that is  $3 - 4\times$  lower than comparable methods.

varying levels of incompleteness. As is expected, the quality of the reconstructions drops with a decreasing amount of observations. Nevertheless, learning filters from incomplete measurements may be interesting for many applications (e.g. adaptive filter learning for demosaicking), but is currently not supported by any existing (efficient) method for convolutional sparse coding. Figure 6.6 shows that even for subsampling rates of up to 50%, the learned filters and quality of the inpainted reconstructions are reasonably good.



**Figure 6.5:** The proposed formulation allows us to use non-circular boundary conditions, as demonstrated for three examples. In practice, the regions outside the image boundary (red) are extrapolated but do not affect the objective.

## 6.5 Reconstruction

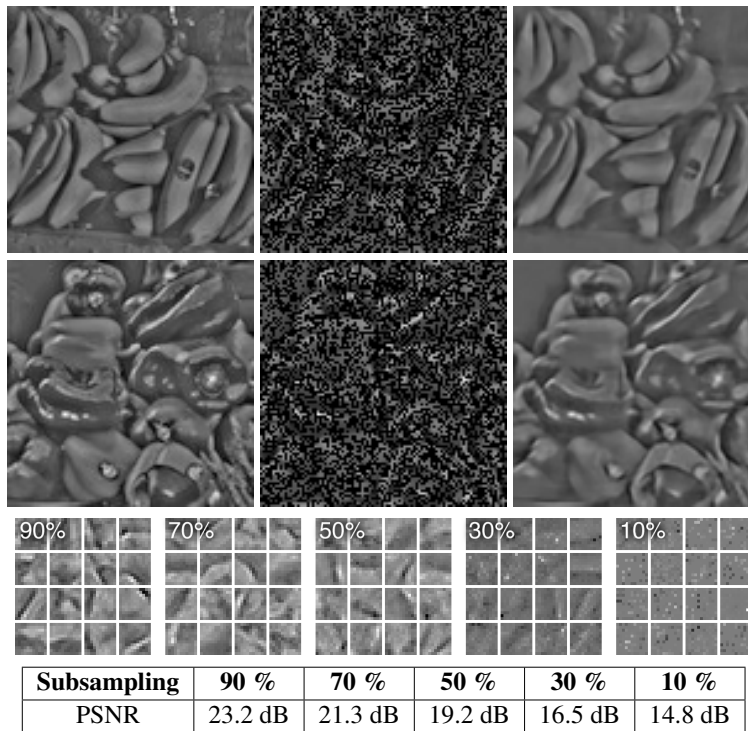
In this section, we evaluate the proposed algorithm for reconstructing signals when the filters are either already learned or known a priori. Reconstruction problems are solved for all filters with the code update step from Algorithm 4.

### 6.5.1 Validation of Reconstruction

Figure 6.7 shows an example image reconstructed from incomplete measurements. This is an inpainting problem, which we test with filters learned from the fruit database. We compare reconstruction quality using filters learned with the proposed method and filters learned with the method described in [238, 239]. Not only do our filters lead to a better reconstruction around edges and sharp features, but our framework allows us to solve this inpainting problem without sacrificing performance in the CSC solver, which was not possible in previous work due to the “Fourier trick”. See Section A.5 for an evaluation using dataset of 22 images. With a single exception, our method outperforms previous work for all test images.

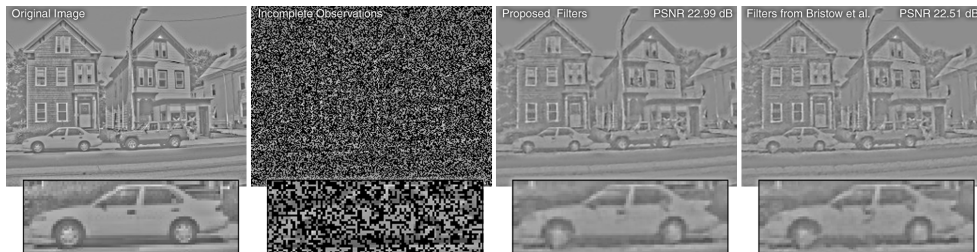
### 6.5.2 Non-Normalized Data

In most results, we show contrast-normalized examples. However, non-normalized data can be intuitively handled in two ways. Either the filters are directly learned from non-normalized training data or a low-frequency term for the DC offset is



**Figure 6.6:** Learning from sparse observations. The top two rows show examples for 50% sampling. The original images are shown in the left column, randomly subsampled observations in the center, and reconstructions of the entire image using filters learned from these sparse observations on the right. Bottom table and filters: evaluation of the learned filters for different subsampling rate of the data. *We show 16 out of the total 100 learned filters for each sampling rate above the table.* One can see that for less than 50% sampling, the reconstruction quality significantly drops due decreasing filter quality.

added to the set of filter after learning. Typical Gabor-like filters with different DC offsets are observed for the former approach. The latter approach can be interpreted as adding a smoothness prior in the form of the low-frequency term rather than rigorously enforcing sparsity. A reconstruction then has to jointly solve for the filter coefficients as well as the low-frequency term, which is shown in Figure 6.8. We have also compared this approach with a state-of-the-art compressive sensing method from Dong et al. [244]. Using the image dataset from Section A.6 (50%



**Figure 6.7:** Inpainting example showing: original image (left), randomly sampled incomplete observations (center left), reconstruction with filters learned with the proposed algorithm (center right), and filters from [238, 239] (right). In addition to this example, we evaluate the reconstruction quality for a larger dataset in Section A.6.



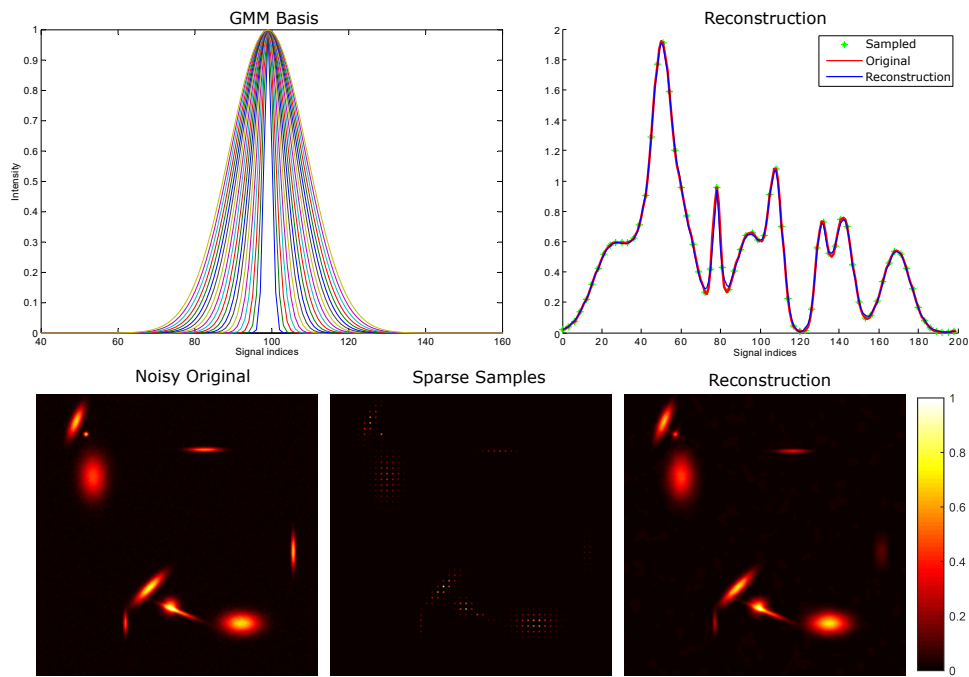
**Figure 6.8:** Inpainting non-normalized data: randomly-sampled, incomplete observations (left and center right), reconstruction with the proposed filters (center left and right).

random sampling with non-normalized data) their method achieves a mean PSNR of 23.5 dB while ours achieves 29.0 dB. This preliminary result suggests that further investigation of sparse convolutional coding might lead to many fruitful applications even outside of the low-level feature learning.

### 6.5.3 Reconstruction with Known Basis

We also evaluate the proposed method for fitting convolutional models to sparse and noisy measurements when the filters are known a priori, such as the physically moti-

vated filters presented in Chapter 4. In general, physically-motivated convolutional sparse coding may have a wide range of applications in radar, sonar, ultrasound and seismic imaging. Figure 6.9 demonstrates reconstructions of sparsely-sampled data in 1D and 2D. Filters are sampled from a Gaussian distribution and the measurements of the 2D example are further corrupted by i.i.d. Gaussian noise with a standard deviation of  $\sigma = 0.01$ . This experiment demonstrates the utility of CSC to non-feature-learning-type applications, such as general Gaussian mixture models. The proposed method is capable of recovering the latent signal with a high quality from *only 6.25% of the samples*.



**Figure 6.9:** Reconstructions for known convolutional basis. The filters in this example are sampled from 1D Gaussians (top left) and used to fit a convolution model to a sparse set of samples (top right). The same experiment is performed in 2D, where a target signal is corrupted by noise (bottom left), subsampled (bottom center), and then reconstructed from *only 6.25% of the noisy measurements* (bottom right).

## 6.6 Discussion

In this chapter, we have described a method for learning and reconstruction using convolutional sparse codes. By exploiting convolutional structure in the unknowns, Convolutional Sparse Coding has the potential to replace or supplement popular patch-based learning and reconstruction methods. The proposed method is applicable to a wide range of computer vision problems, such as feature learning, denoising, inpainting, and demosaicking.

At its core, the method relies on a generalized formulation of CSC, which allows to tackle a variety of different problems with convolutional structure in the unknowns. This general formulation enables proper handling of boundary conditions by treating boundaries as unknowns. It allows for feature learning from incomplete observations, or any type of linear operator applied to the estimation. Based on the general CSC formulation, an efficient proximal algorithm is derived, which solves for both the convolutional codes and sparse feature maps in an alternating coordinate descent scheme. The resulting method is faster than the state-of-the-art and finds better solutions.

Although already faster than existing methods, our formulation is inherently parallel and the runtime could further be significantly improved by an efficient GPU implementation. An interesting avenue for future research is to evaluate learning and reconstructing features in higher-dimensional problems, such as 3D hyperspectral image data [245] or 4D light fields [246]. For higher-dimensional problems the proposed approach is limited by its memory needs. Full resolution feature maps and kernels in the frequency domain are required. Hence, the memory requirements scale linear with the number of filters  $K$ , but exponentially with the dimensionality. Besides higher-dimensional problems, it would be interesting to apply the proposed framework to more complex hierarchical convolutional structures and networks [220, 233] that could be particularly useful for high-level computer vision applications, such as recognition.



## Chapter 7

# Doppler Velocity Imaging

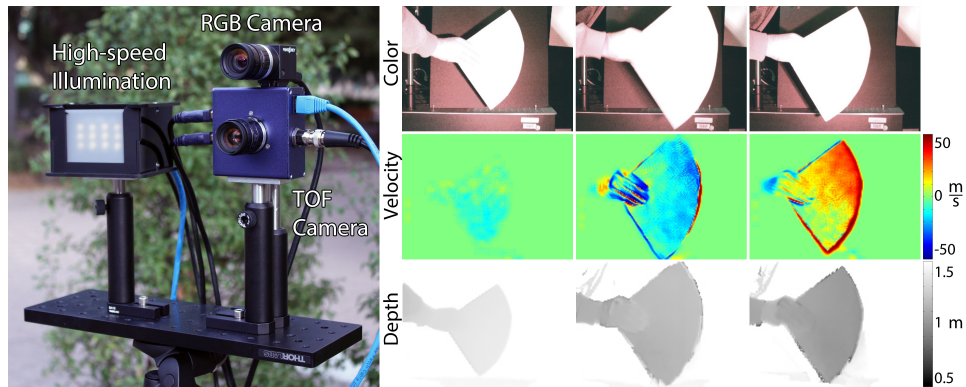
In the previous chapters, we have demonstrated that the temporal dimension of light transport, i.e. transient images, can be extracted from CIS TOF measurements. This new temporal dimension has enabled us to image non directly visible objects from indirect reflections, and imaging in scattering media.

This chapter demonstrates that we can extract a further novel imaging modality from CIS TOF measurements: per-pixel radial velocity measurement. The proposed technique exploits the Doppler effect of objects in motion, which shifts the temporal illumination frequency before it reaches the camera. Relying again on the temporally convolutional imaging formation of CIS, it is possible to code illumination and modulation frequencies of the TOF camera, such that object velocities directly map to measured pixel intensities.

Furthermore, we show that a slight modification of our imaging system allows for color, depth, and velocity information to be captured simultaneously. Combining the optical flow computed on the RGB frames with the measured metric radial velocity allows us to further estimate the full 3D metric velocity field of the scene.

### 7.1 Introduction

Pioneers of photography, including Eadweard Muybridge and Harold “Doc” Edgerton, advanced imaging technology to reveal otherwise invisible motions of high-speed events. Today, understanding the motion of objects in complex scenes is at



**Figure 7.1:** A new computational imaging system (left) that allows for metric radial velocity information to be captured instantaneously for each pixel (center row). The Doppler effect of objects in motion is detected as a frequency shift of temporally modulated illumination. By capturing a few coded CIS measurements and conventional RGB frames, we demonstrate that color, velocity, and depth information can be recorded simultaneously. For the captured example, the left-most frame shows a static object (velocity map is constant), which is then moved towards (positive radial velocity) or away from (negative velocity) the camera.

the core of computer vision, with a wide range of applications in object tracking, segmentation, recognition, motion deblurring, navigation of autonomous vehicles, and defense. Usually, object motion or motion parallax are estimated via optical flow [247]: recognizable features are tracked across multiple video frames. The computed flow field provides the basis for many computer vision algorithms, including depth estimation. Unfortunately, optical flow is computationally expensive, fails for untextured scenes that do not contain good features to track, and it only measures 2D lateral motion perpendicular to the camera’s line of sight. Further, the unit of optical flow is pixels; metric velocities cannot be estimated unless depth information of the scene is also available. For the particular application of depth estimation, many limitations of optical flow estimation can be overcome using active illumination, as done by most structured illumination and TOF cameras. With the emergence of RGB-D imaging, for example facilitated by Microsoft® Kinect®, complex and untextured 3D scenes can be tracked by analyzing both color and depth, resulting in richer visual data that is useful for many applications.

In this chapter, we introduce a new approach to directly imaging radial object velocity. Our approach analyzes the Doppler effect in CIS TOF cameras: object motion towards or away from the camera shifts the temporal illumination frequency before it is recorded. The Doppler effect has been reviewed in detail in Section 2.6.6. While conventional TOF cameras encode phase information into intensity, instead, we propose here Doppler Time-of-Flight (D-TOF) as a new imaging mode, whereby the change of illumination frequency (corresponding to radial object velocity) is directly encoded into the measured intensity. The required camera hardware is the same as for conventional TOF imaging, but illumination and modulation frequencies are carefully designed. Our approach is a full-field imaging method, meaning that it *does not require the scene to be sequentially scanned* unlike most existing Doppler radar or Lidar systems that only capture a single scene point at a time. Furthermore, we can combine depth and velocity imaging using either two TOF cameras or using the same device by alternating the modulation frequencies between successive video frames; color images can be obtained with a conventional camera.

The proposed technique offers a new imaging modality that is ideally suited for fast motion. Note that our method is fundamentally different from Optical flow, which is traditionally used to estimate apparent motion in dynamic scenes; see Section 2.6.6. Optical flow from a single camera is restricted to estimating *lateral motion* whereas the Doppler is observed only for *radial motion* towards or away from the camera. In fact, it is a complementary technique: together, optical flow on RGB frames and D-TOF allow for the metric 3D velocity field to be estimated, which is otherwise not easily possible. In general, however, D-TOF is independent of the RGB flow and works robustly for cases where optical flow often fails, including untextured scenes and extremely high object velocities. We also discuss a mode for simultaneous range and velocity imaging. As with standard TOF imaging, our method requires a few subframes to be captured with different modulation signals. In our prototype system, we use rapid time-sequential acquisition of the required subframes for simplicity, which is a common strategy for regular TOF imaging. However, appropriate hardware, such as the multi-camera extension which we will discuss in Section 7.7, the method can be implemented as a true snapshot imaging approach.

In the following, we first derive a mathematical framework for velocity esti-

mation with CIS TOF cameras, implement a prototype CIS imaging system, and validate the proposed model extensively in simulation and with the prototype. The imaging system is evaluated using a range of different types of motion, for textured and untextured surfaces as well as indoors and under strong outdoor ambient illumination. Finally, we demonstrate that the velocities measured with our system can be combined with RGB flow, allowing for the metric 3D velocity field to be estimated on a per-pixel basis.

## 7.2 Dynamic CIS Image Formation

In Section 2.6, we have explained the working principle of CIS TOF imaging for static scenes. In this section, we analyze how the image formation behaves for objects in motion.

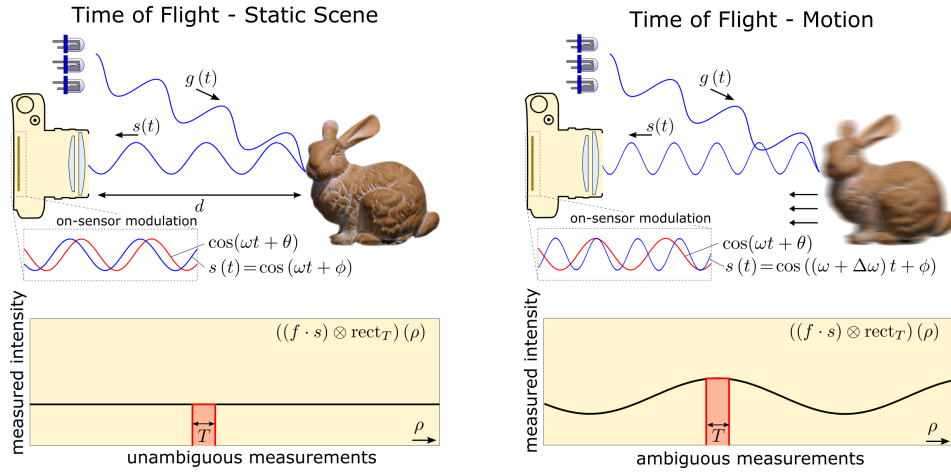
In fact, the conventional CIS image formation model breaks down when objects of interest move with a non-negligible radial velocity. In this case, the illumination frequency undergoes a Doppler shift when reflected from an object in motion; see again Section 2.6.6.

The illumination arriving at the sensor is now frequency-shifted to  $\omega_s = \omega_g + \Delta\omega$ , where the change in temporal frequency  $\Delta\omega$  depends on the radial object velocity and the illumination frequency, as defined in the definition of the Doppler shift in Eq. (2.29) from Section 2.6.6. For intuition, we consider the case of an approximately constant velocity  $v$  throughout the exposure time. If we continue to assume a homodyne setting with  $\omega_f = \omega_g = \omega$ , our measurement is still *heterodyne* (due to the frequency shift  $\Delta\omega$ ). In particular, Eq. (2.28) from the related work Section 2.6.1 on CIS heterodyning now becomes:

$$b_\theta(\rho) \approx \frac{a}{2} \cos(-\Delta\omega\rho - \phi + \theta). \quad (7.1)$$

Note that this equation is now dependent on the time of measurement. Unfortunately, the introduced temporal intensity variation makes it more difficult to estimate phase and therefore also depth. In audio signal processing, this time-dependent low-frequency artifact is known as a beating pattern. We illustrate this in Figure 7.2.

The phase estimate from Eq. (2.27) becomes then Eq. (7.2). For simplicity, we



**Figure 7.2:** Depth imaging. For static scenes, measurements are unambiguous: different phase shifts result in unique intensity measurements (left). For dynamic scenes, the Doppler shift results in a low-frequency beating pattern that makes measured intensities ambiguous, and hence prevents reliable depth estimation (right).

sample here only  $N = 2$  phases with  $\theta_i = i \cdot 2\pi/N$  for  $i \in \{0, \dots, N - 1\}$ .

$$\phi_{est}(\rho) = -\text{atan}\left(\frac{b_{\pi/2}(\rho)}{b_0(\rho)}\right) + \Delta\omega\rho, \quad (7.2)$$

where the distortion  $\Delta\omega\rho$  linearly depends on the (unknown) object velocity. Note that, in practice, the estimated phase for moving objects corresponds to its average throughout the exposure.

To summarize, in the homodyne setup, where the frequency of the light source and the frequency of the camera reference signal are identical, the Doppler shift introduced by moving objects results in mismatched frequencies on the image sensor. This situation is a heterodyne measurement. However, in heterodyne range imaging, as described in Section 2.6.1, the frequency mismatch is deliberately introduced in the sensor modulation. A major limitation of heterodyne ranging is that multiple ( $> 2$ ) measurements have to be captured to reliably estimate phase and depth. Since only low frequency beats can be detected reliably, a significant amount of time needs to pass between the two measurements for robust phase estimation.

For moving objects, the necessity to capture multiple images would place severe constraints on the velocity. To facilitate reliable velocity estimation, we derive a new computational TOF imaging methodology in the following section. Inspired by the general concept of Orthogonal Frequency-Division Multiplexing (OFDM) e.g. [248], D-TOF uses illumination and on-sensor modulation frequencies that are orthogonal within the exposure time of the camera. Using this choice of frequencies along with a newly-devised reconstruction method, we demonstrate the first approach to per-pixel radial velocity estimation.

### 7.3 Doppler Velocity Imaging

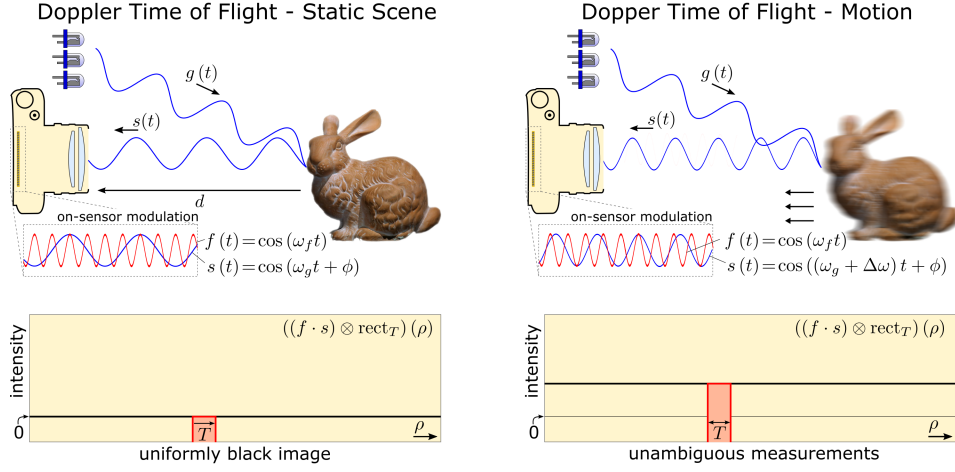
As illustrated in Figure 7.2 (right), the low-frequency beating pattern created by the Doppler effect makes it difficult or impossible to capture reliable Doppler frequency and phase information. Consider the following example: a road cyclist travels at a speed of  $v = 10 \frac{m}{s}$  towards the camera. For an illumination frequency of 50 MHz (i.e.  $\omega_g = 50 \cdot 10^6 \cdot 2\pi/s$ ), the observed Doppler shift is only

$$\Delta\omega = \frac{v}{c} \omega_g = \frac{10 \frac{m}{s}}{300 \cdot 10^6 \frac{m}{s}} \cdot 50 \cdot 10^6 \frac{2\pi}{s} \approx 1.67 \frac{2\pi}{s} \quad (7.3)$$

A frequency shift of only 1.67 Hz may seem small enough to be safely ignored. However, we show in the following that even such a minute change contains valuable information that can be used for velocity estimation.

#### 7.3.1 Velocity Imaging via Orthogonal Frequencies

Inspired by multiplexing techniques in digital communication, we devise an unconventional way to extract velocity information from the small Doppler shift observed by a TOF camera. We can interpret the camera system as a communication channel and consider the illumination a carrier signal. The carrier is optically modified by moving objects — we observe a change in carrier amplitude, phase, and frequency. The secondary modulation in the sensor followed by a low-pass filter of the exposure time corresponds to the demodulation process in communication. Conventional communication channels use orthogonal frequencies; any inter-carrier interference (which could be caused by a frequency drift) is a polluting signal (see e.g. [248]).



**Figure 7.3:** Velocity imaging. Illumination  $\omega_g$  and modulation  $\omega_f$  frequencies are designed to be orthogonal within the exposure time  $T$ . For static scenes (left), this particular choice of frequencies will integrate to zero. The Doppler shift of moving scenes destroys the orthogonality and results in an approximately linear relationship between radial velocity and recorded intensity (right).

For Doppler TOF, we deliberately design the frequencies in the receiver and transmitter to be orthogonal, such that the (usually polluting) inter-carrier interference carries the desired velocity information.

An illustration of our orthogonal frequency Doppler TOF is shown in Figure 7.3. For the application of direct velocity imaging, we would like to ensure that the measured signal for a stationary object is zero (or a constant intensity offset); see left of Figure 7.3. We can achieve this by operating the TOF camera in heterodyne mode with two orthogonal frequencies  $\omega_g$  and  $\omega_f$ . While any two sine waves with frequencies  $\omega_g \neq \omega_f$  will be orthogonal for sufficiently long integration times, this is not the case for finite integrals (exposures) in the presence of low frequency beating patterns. Designing both frequencies to be orthogonal is done by setting

$$\omega_g = k \frac{2\pi}{T} \quad \text{and} \quad \omega_f = l \frac{2\pi}{T} \quad \text{with } k, l \in \mathbb{N}, k \neq l, \quad (7.4)$$

i.e. having the exposure time  $T$  be an integer multiple of the period of both signals.

It is then easy to show from Eq. (2.25) that

$$b_\theta(\rho) = \int_\rho^{\rho+T} p(t) dt = 0 \quad (7.5)$$

for stationary objects ( $\omega_s = \omega_g$ ). In practice, we set  $l = k + 1$  and we set  $k = \omega_g T / 2\pi$ , which depends on  $T$  and the desired frequency  $\omega_g$ .

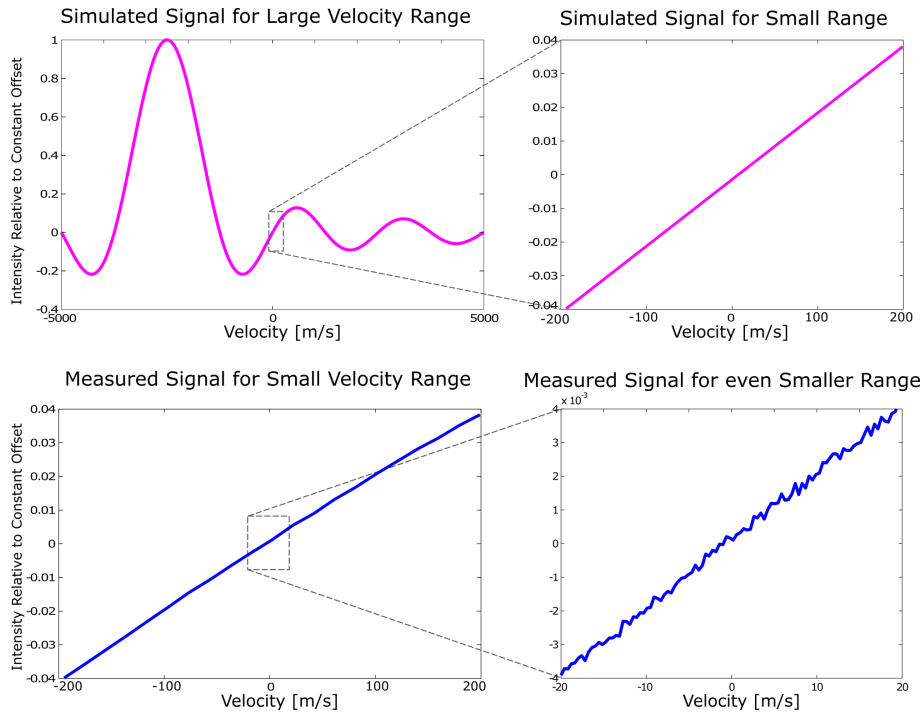
Given these two orthogonal frequencies we now use the inter-carrier interference to extract valuable information about the Doppler shift. We achieve this by computing the ratio of a heterodyne measurement and a homodyne measurement. Using only the low-frequency terms from Eq. (2.25), this ratio can be expressed as Eq. (7.6). Without loss of generality, we assume an exposure interval of  $[0 \dots T]$ .

$$\begin{aligned} r &= \frac{\int_0^T \cos(\omega_f t + \theta) \cdot (a \cos((\omega_g + \Delta\omega)t + \phi) + s_0) dt}{\int_0^T \cos(\omega_g t + \theta) \cdot (a \cos((\omega_g + \Delta\omega)t + \phi) + s_0) dt} \\ &\approx \frac{\int_0^T \frac{a}{2} \cos((\omega_f - \omega_g - \Delta\omega)t + \theta - \phi) dt}{\int_0^T \frac{a}{2} \cos(-\Delta\omega t + \theta - \phi) dt} \\ &= \frac{\frac{a}{2(\omega_f - \omega_g - \Delta\omega)} [\sin((\omega_f - \omega_g)t - \Delta\omega t + \theta - \phi)]_0^T}{\frac{a}{-2\Delta\omega} [-\Delta\omega t + \theta - \phi]_0^T} \\ &= \frac{-\Delta\omega}{\omega_f - \omega_g - \Delta\omega} \cdot \underbrace{\frac{\sin((\omega_f - \omega_g)T - \Delta\omega T + \theta - \phi) - \sin(\theta - \phi)}{\sin(-\Delta\omega T + \theta - \phi) - \sin(\theta - \phi)}}_{=1} \\ &\approx \frac{-\Delta\omega}{\omega_f - \omega_g} \end{aligned} \quad (7.6)$$

since  $(\omega_f - \omega_g)T = (k - l) 2\pi$ , and  $\Delta\omega \ll \omega_f - \omega_g$ . Note that we are only varying the sensor modulation frequency  $\omega_f$  here for the two measurements. Hence, the phases  $\phi$  in numerator and denominator are identical, and the same offsets  $\theta$  are selected in the homodyne and heterodyne measurement.

Figure 7.4 shows the model derived here. On the left side, we see the full model without any approximations (i.e. without neglecting high frequency components in Eq. (7.6)). Although the image formation is nonlinear, for a relative large range of metric velocities it is very well approximated (Figure 7.4, top right) by our linear model (Eq. (7.6)). We experimentally verify the model using our





**Figure 7.4:** Simulated and measured intensities for a range of different velocities. Although the mapping from radial velocity to measured intensity is generally nonlinear (top left), throughout a large range of velocities the conversion is approximately linear (top right). We verify the predicted mapping using our prototype camera (bottom). These particular measurements were captured with a static scene, and acquired with a modulation frequency of  $\omega_f = 60$  MHz and an illumination frequency of  $\omega_g = 60$  MHz + 1 KHz +  $\Delta\omega$ . Thus, the Doppler shift for an object moving at a specific velocity was programmed into the illumination frequency for this particular experiment.

camera prototype (Figure 7.4, bottom). With known, orthogonal illumination and modulation frequencies  $\omega_g, \omega_f$ , it is therefore straightforward to compute the Doppler shift  $\Delta\omega$  from Eq. (7.6). The ratio image  $r$  can be interpreted as a direct measurement of the instantaneous per-pixel radial velocity.

We note that this approach still requires two measurements: one heterodyne image and one homodyne image. There are several possible solutions for either acquiring these truly simultaneously, or they can be acquired in quick succession.

For instantaneous measurements, two synchronized TOF sensors can be mounted in a co-axial setup; one of the sensors is modulated with the same frequency as the light source ( $\omega_g$ ), while the other uses a slightly different frequency  $\omega_f \neq \omega_g$ . However, this approach requires accurate synchronization and control over the modulation waveforms of each camera, which is challenging for off-the-shelf camera systems. Nonetheless, we describe such a hardware system as an extension at the end of this chapter in Section 7.7.

Instead of using two distinct sensors, it would also be possible to multiplex pixels with two different modulation frequencies onto the same image sensor, either in alternating scanlines or in a checkerboard pattern. This concept is similar in spirit to techniques that have been proposed for HDR cameras [249, 250].

A third possibility is to rapidly alternate between two modulation frequencies using a single TOF camera. In this case, the measurements are not truly instantaneous, and alignment problems may occur for very fast motions. However, the two measurements can be taken immediately after each other, as fast as the camera hardware allows, e.g. at 30 or 60 Hz. We follow this approach as it only requires a single TOF camera and is therefore easy to realize. Note that, similar to heterodyne depth estimation [114], the Doppler shift can also be estimated directly from the low-frequency beating pattern, but at the cost of requiring multiple measurements that are much more widely spaced in time (hence not suitable for velocity estimation).

Finally, we note that the model from Eq. 7.6 only holds for sinusoidal modulation functions. If other periodic signals are being used, additional harmonic frequency components are introduced, which distort the measurements for both stationary and moving targets. However, these offsets are systematic and can be calibrated for a specific TOF camera/lights source combination (see Section 7.4 and Section A.7).

### 7.3.2 Simultaneous Range and Velocity

In many applications it may be useful to obtain both velocity and range measurements at the same time. As in standard TOF imaging, this can be achieved by capturing a second homodyne measurement with the phase offset by  $\pi/2$ . Simultaneous range and velocity imaging therefore requires a total of three measurements: a heterodyne image with  $\theta = 0$ , a homodyne image with  $\theta = 0$ , and a homodyne

image with  $\theta = \pi/2$ .

As discussed in Section 7.2, motion introduces a velocity-dependent distortion  $\Delta\omega\rho$  of the depth measurement (Eq. (7.2)). However, since the distortion linearly depends on the Doppler shift  $\Delta\omega$ , which is known from the velocity estimation step (Eq. (7.6)), we can now correctly estimate the phase delay (and hence the depth) from Eq. Equation 7.2. This only requires an additional calibration step to obtain  $\Delta\omega\rho$  for a specific velocity, which corresponds to estimating the time offset  $\rho$  between the start of the exposure time and the reference time for signal generation in the camera and light source.

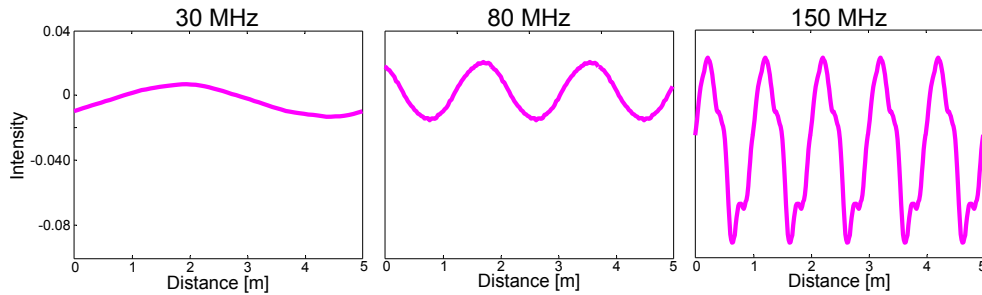
As mentioned, simultaneous velocity and range imaging requires three distinct measurements. We note that the illumination signal is the same for all three measurements, only the reference signal for the camera changes. As in the case of velocity-only imaging, this means that all three measurements can potentially be acquired at the same time using either multiple sensors with a shared optical axis, or a special sensor design with interleaved pixels. If neither option is available, rapid frame-sequential imaging is also possible.

## 7.4 Implementation

*Hardware* For all physical experiments, we use the experimental TOF CIS camera system from Chapter 4 and Chapter 5. As described in these chapters, our system comprises a custom RF modulated light source and a demodulation camera based on the PMD Technologies PhotonICs 19k-S3 sensor. The light source is again an array of 650 nm laser diodes.

We run all of the presented results with 30 MHz. The modulation signals are nearly sinusoidal, but contain multiple low-amplitude harmonic components. To avoid systematic errors in depth and velocity estimation, these components must be calibrated as described in the following.

*Correcting for Higher-order Harmonics* Our camera prototype has the drawback that the periodic modulation functions are not perfectly sinusoidal, although they are very close. In addition to the fundamental frequency, this introduces higher-order



**Figure 7.5:** Depth-dependent offset introduced by higher-order frequency components for a range of modulation frequencies. These offsets are calibrated in a one-time offline process and then used to correct the raw phase measurements on a per-pixel basis.

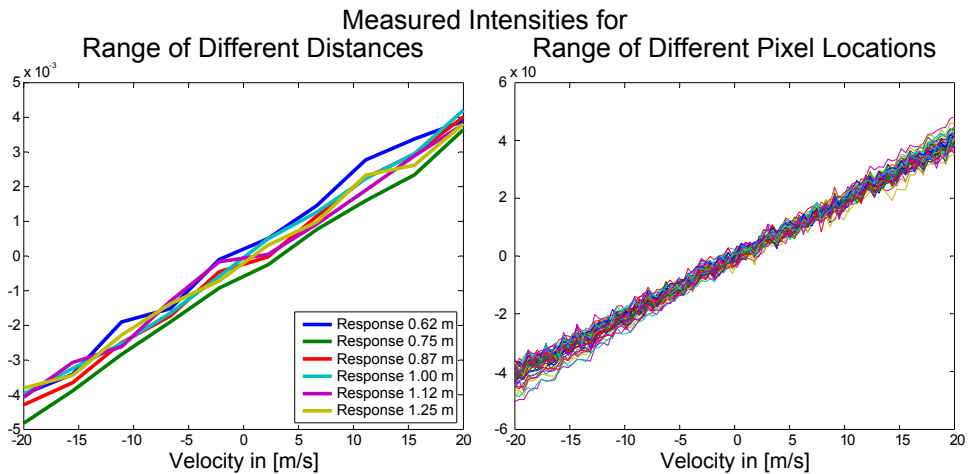
harmonic components to the modulation signal. Please refer to Section A.2 and Section A.7 for a detailed derivation of the image formation in these conditions. Unfortunately, the higher-order components are generally not orthogonal, thus they can cause a phase-dependent offset. We calibrate this offset for different modulation frequencies and phase shifts  $\theta$  using a static target. The depth-dependent offsets are plotted for different modulation frequencies in Figure 7.5.

This offset is calibrated in an offline process and raw phase measurements can be corrected digitally using a lookup table. Note that for relatively low modulation frequencies, such as 30 MHz, we find a fairly large depth range (around 1 m) to be almost independent of this offset. In practice, it is therefore relatively easy to remove the higher-order frequency components.

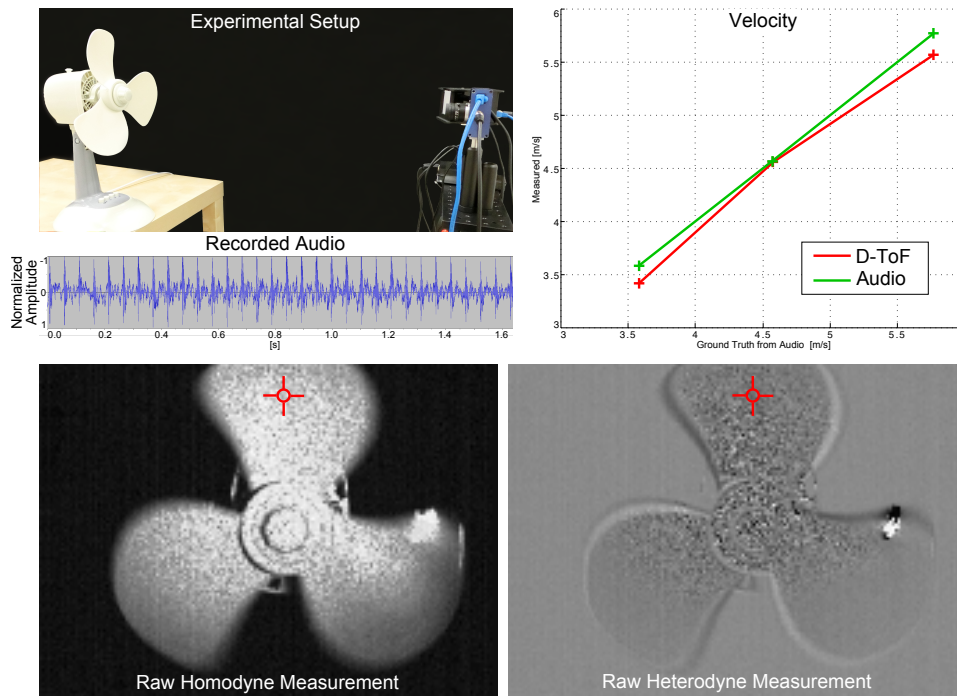
*Calibrating Phase Response* As is standard practice in time-of-flight cameras, we calibrate the physical intensity response for different phase shifts  $\phi$  in an offline calibration. Following [115], we measure the physical intensity response for a phase sweep of the illumination frequency and fit a fifth-order polynomial to the measurements. This is used as a lookup table for converting phase to depth rather than solving Eq. (2.27) directly. With our prototype, we measure a notable zeroth-order component of the fitted polynomial, corresponding to fixed pattern phase noise. This is easily corrected with the lookup table. Any other illumination-specific terms, for example introduced by the baseline between camera and light source, are

automatically calibrated with the described procedure and do not require additional processing.

*Verification of Calibration Procedure* The two calibration procedures described above are performed for all spatial locations on the sensor independently. To verify our calibration routines, we image a static target and apply a frequency and phase sweep to the modulation function, simulating objects at different velocities and depths. The results shown in Figure 7.4 (left) demonstrate that the measured intensities for a constant phase but varying Doppler shift follow the model derived in the Section 7.3. Other than a small amount of noise, which is mostly due to a relatively low signal-to-noise ratio, the curve is linear and behaves as predicted. In Figure 7.6 (left), we verify experimental measurements for a range of different phase offsets in the modulation frequency. This simulates objects at various depths, as indicated in the legend. Finally, we also test the velocity-dependent behavior for a range of different pixels over the sensor location and show results in Figure 7.6



**Figure 7.6:** Experimental verification of the imaging system for varying object velocities and depths (left) as well as velocity-dependent behavior for a range of different pixel locations on the sensor (right). All of this data is captured using a large planar target perpendicular to the camera and sweeping the illumination frequency (to simulate different Doppler shifts) and phase (to simulate different object distances).



**Figure 7.7:** Experimental validation of velocity estimation using a fan with adjustable rotation speed (three settings). We measure the ground truth velocity of the rotating blades (top left) by analyzing audio recordings (top, lower left). The top right plot shows the velocity measured by D-TOF compared to the ground truth for a varying rotation speed. As the speed becomes larger, estimation errors increase to a maximum of about 0.2 m/s. The bottom row shows the unprocessed full-field measurements of the homodyne (left) and the heterodyne (right) frequency setting with the pixel indicated for which we plotted the velocities on the top right.

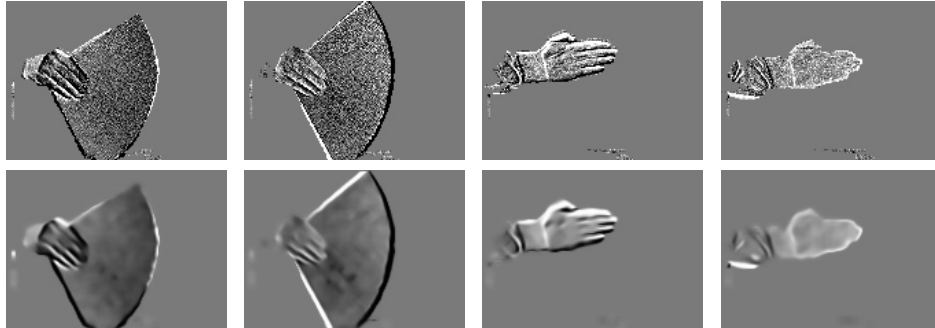
(right). The remaining variance over pixel locations and phases is minimal.

Figure 7.7 shows another experiment that we used to verify the accuracy of our prototype D-TOF camera. In this example, we adjusted the speed of a rotating fan and imaged its blades such that, throughout the time it takes for a single blade to move across a pixel, forward motion is observed by that pixel. The exposure time of the TOF camera was set to 1.5 ms and the fan was captured from a frontal perspective (raw homodyne and heterodyne measurements shown in Figure 7.7 bottom). We manually measured the slope of the fan blades, which is constant over

the entire blades. The radius of the plotted position was measured, allowing us to calculate the “ground truth” velocity when the rotation speed of the fan is known. Since the exact rotation speed is not actually known, we measure it by mounting a small pin on one of the blades and mounting a piece of flexible plastic in front of the fan, such that the rotating pin strikes the plastic exactly once per revolution, creating a distinct sound. We record the sound (sampled at 44 KHz) of this setup to estimate the ground truth velocity of the fan blades, observed by one pixel, which is compared with the corresponding D-TOF estimate (Figure 7.7, top right). For this experiment, the estimation error is always below 0.2 m/s. Errors are mainly due to the low SNR of the measured Doppler-shifted signal.

*Subframe Alignment* Although the required heterodyne and homodyne shots can be captured simultaneously as shown later in Section 7.7, the hardware modifications for an alternating capture using a single-sensor require significantly less effort. In this case, since we are dealing with moving objects, the individual shots cannot be assumed to be perfectly aligned. This results in velocity artifacts around edges in the scene. We can mitigate, although not completely remove, these artifacts by computing a SIFT flow on the raw data and warping them to a reference frame. While not perfect, the SIFT flow delivered sufficiently good warps for most captures.

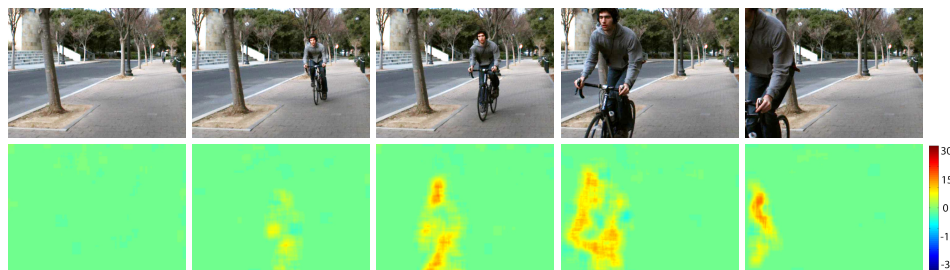
*Denoising* With our system, we capture an extremely small frequency shift (in the Hz range) relative to the modulation frequency (the MHz range). Additionally, the quantum efficiency of emerging time-of-flight sensors is still far from that of modern solid state sensors [251]. Therefore, the slight Doppler shift in our prototype is only measured by a very low photon count in the measured signal. For such low photon counts the heteroskedastic normal approximation for the Poisson noise from Section 2.4.3 is not accurate anymore. Standard denoising methods assuming Gaussian noise fail therefore in such low-light scenarios. We apply a binning-based non-local means denoising strategy to all captured velocity maps (see Figure 7.8). Please see the Section A.8 for more details and denoising comparisons.



**Figure 7.8:** Velocity maps color-coded in grayscale. The maps computed from raw measurements (top) are corrupted by Poisson noise. To account for this, we apply a binning-based non-local means-type denoiser to the reconstructed velocity images (bottom).

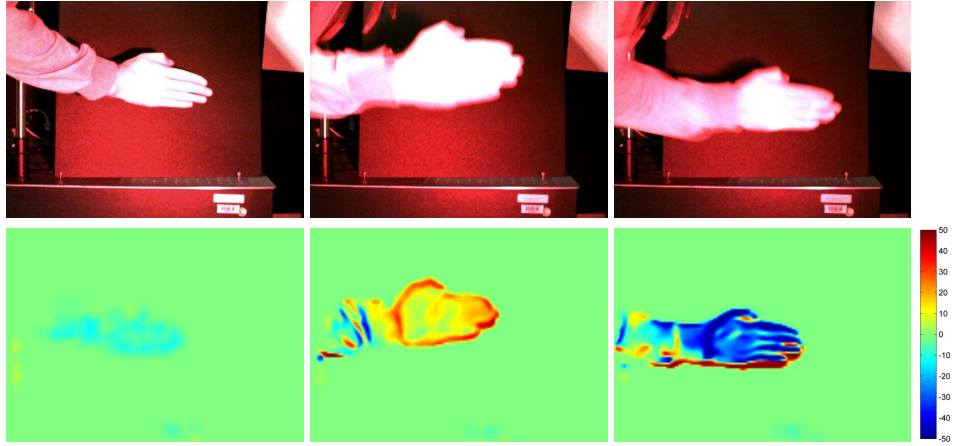
## 7.5 Results

We show results captured with our prototype imaging system in Figures 7.1, 7.9, 7.10, 7.12, 7.11, 7.13, 7.14. The results validate the proposed imaging system for a variety of challenging indoor and outdoor scenes. Color images are recorded with the same exposure time as the time-of-flight camera. Most of the scenes have a slight red tint, because we work with eye-safe red illumination in the visible spectrum. Like current commercial TOF cameras, future implementations of this system would most likely use invisible, near infrared wavelengths to encode velocity and depth



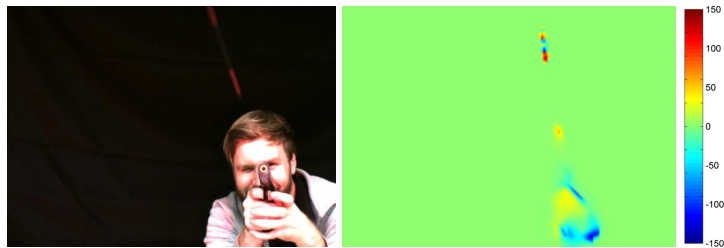
**Figure 7.9:** Complex scene with ambient illumination and a large depth range. The velocity is robustly estimated within the range of the illumination (approx. 5m inside), even in outdoor settings.



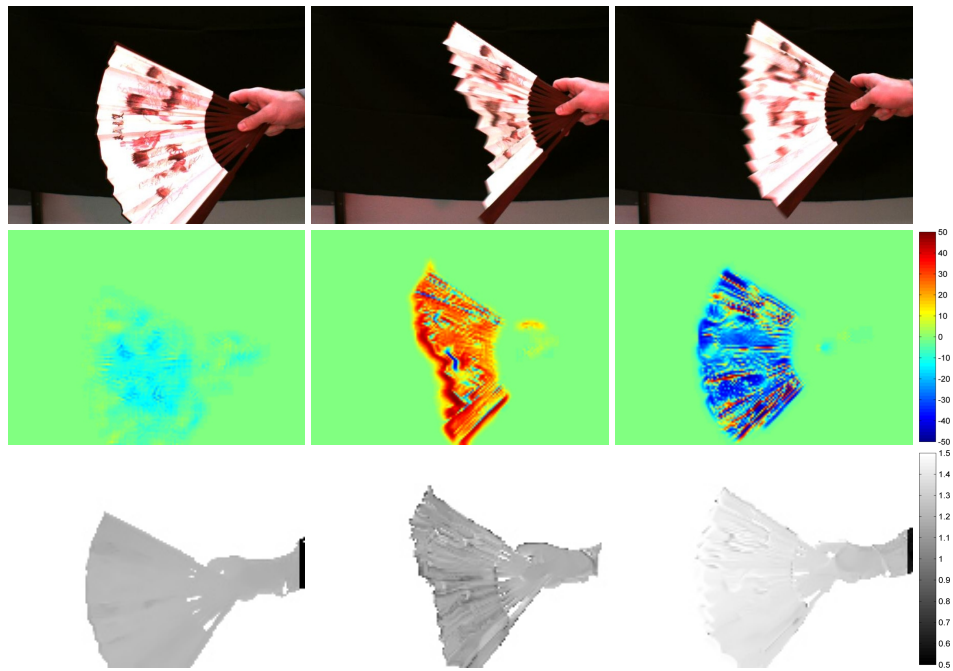


**Figure 7.10:** This result shows a periodic motion of a hand along the optical axis. The static scene on the left results in no response of the sensor, whereas forward (center) and backward (right) movement result in positive and negative responses, respectively.

information. The reconstructed velocity maps are color-coded; absolute units are indicated in the color bars. As expected, static scenes result in a constant velocity map whereas velocity is directly encoded in the measurements and subsequently reconstructed for each sensor pixel independently. In addition to the velocity maps, Figures 7.1, 7.12, 7.13, 7.14 also show the corresponding depth maps that can be estimated from an additional capture as well as the velocity maps (see Section 7.3.2).

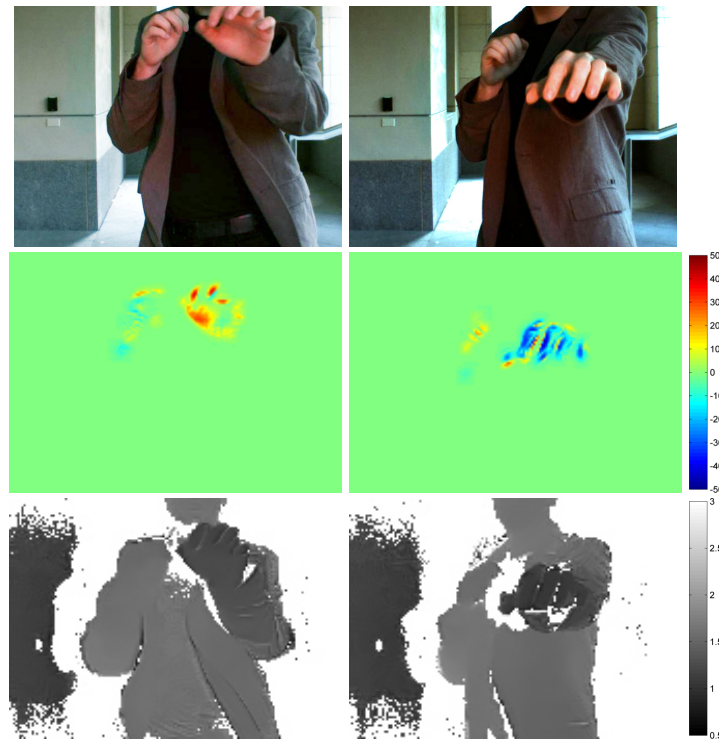


**Figure 7.11:** Even extremely fast motion, such as a bullets shot with a spring airsoft gun, can be captured with our system. The airsoft gun is being advertised as shooting bullets with 99 m/s; we measure a radial velocity of 98.2 m/s (average of the peak pixels).



**Figure 7.12:** This result shows periodic motions in  $z$  for a textured object. Although the estimated velocity is mostly correct, shadows and dark scene parts are challenging for robust velocity estimation.

The selection of scenes shows a wide range of motion types that can be reconstructed with the proposed method, but it also highlights several challenges of D-TOF and TOF in general. D-TOF requires two frames to be captured, and they must be aligned if recorded with a single camera. In some instances, such as the results from Figure 7.10 and Figure 7.11, the alignment is challenging and any errors will propagate into the velocity maps, especially around depth-discontinuities. These artifacts can be mitigated by optimizing the camera firmware to minimize switching time between the subframes or by using two co-axial TOF cameras, as demonstrated below in Section 7.7. Objects with dark albedos, as for example observed in Figure 7.12, are challenging for any TOF method because only a small amount of the coded illumination is reflected back to the camera. Similarly, shadows are very challenging and often result in either no depth/velocity estimation or errors (sweater in Figure 7.9 and regions between fingers in Fig. 7.13). Whereas some of

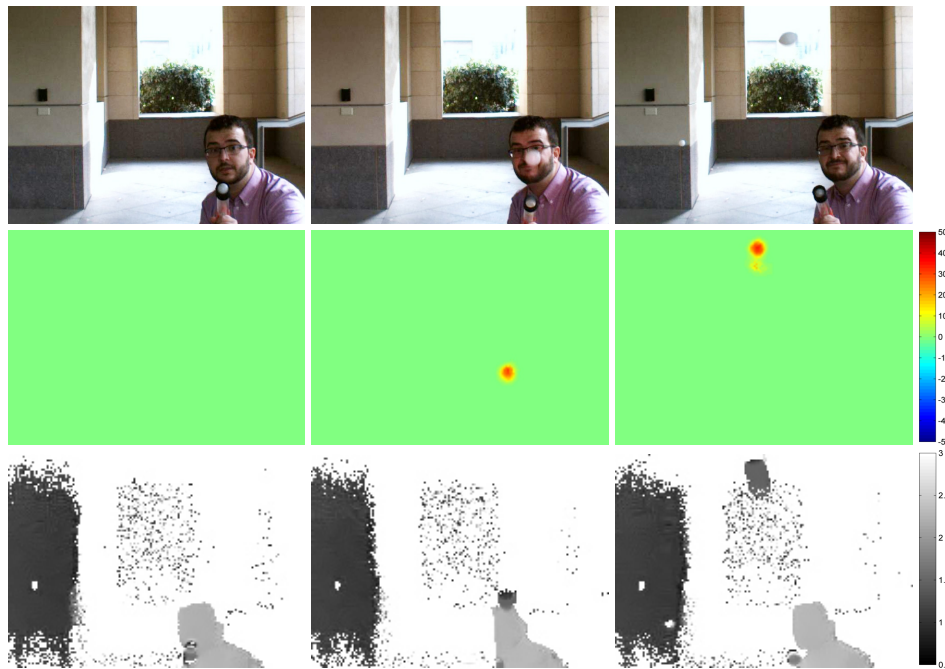


**Figure 7.13:** We envision a wide range of applications for our technique, including gaming and human-computer interaction.

these limitations can be overcome with better hardware, others are inherent to the time-of-flight approach. Please see Section 7.8 for a more detailed discussion.

## 7.6 Toward the 3D Velocity Field

Optical flow computed from conventional video sequences estimates the 2D projection of the 3D flow field onto the image plane. The radial component is usually lost. Furthermore, optical flow is an ill-posed problem and may fail in many scenarios. Doppler TOF addresses two problems of optical flow: first, it can help in cases where optical flow fails either due to large displacements or missing scene structures. Second, our technique also helps in cases where the optical flow estimation is successful; in this case, we can recover the 3D metric flow by combining metric radial velocity and the 2D optical pixel flow.

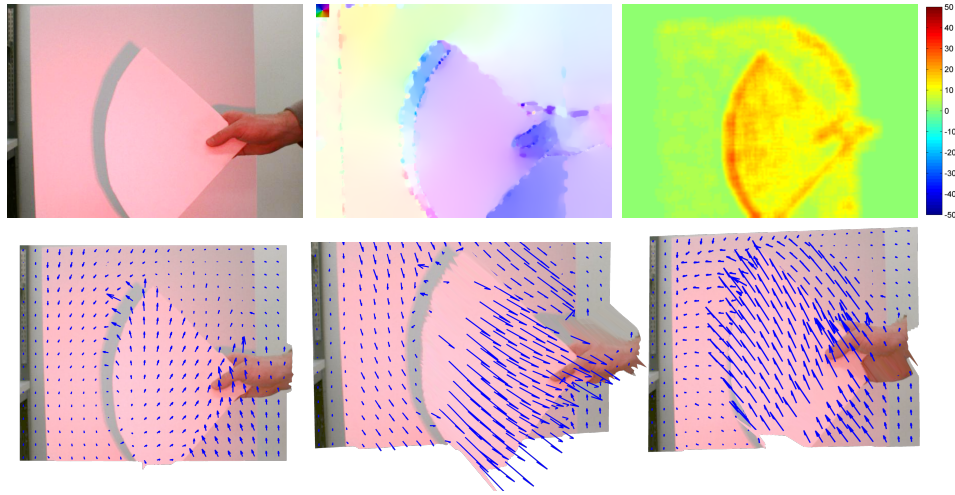


**Figure 7.14:** Physical props for gaming, such as ping pong balls fired with this toy gun, could be tracked and enable new HCI techniques.

Figure 7.15 shows an example scene where regular optical flow [252] as well as SIFT-flow [253] fail due to limited structure in the scene. Our method can successfully capture the velocity of the objects and could also lead to a proper segmentation



**Figure 7.15:** Failure case of optical flow for a moving, but untextured scene (left). Optical flow [252] and SIFT flow [253] for two succeeding color frames are shown in the second and third column; the 2D flow vectors are color-coded with the shown color wheel (insets). Both methods cannot recover the true 2D motion of the fan and wrongly segment the scene. Our orthogonal velocity estimate can resolve this problem and properly segment the scene.



**Figure 7.16:** Towards 3D flow: when optical flow succeeds, the full 3D metric flow is uniquely estimated from both 2D pixel flow and the radial velocity maps. The top images show a frame where optical flow computed reasonable estimates. The bottom shows full 3D velocity estimate for different views. Note that the optical flow helps us to determine that fan’s velocity is slightly rotated to the upper right, where the center of rotation is located (bottom left).

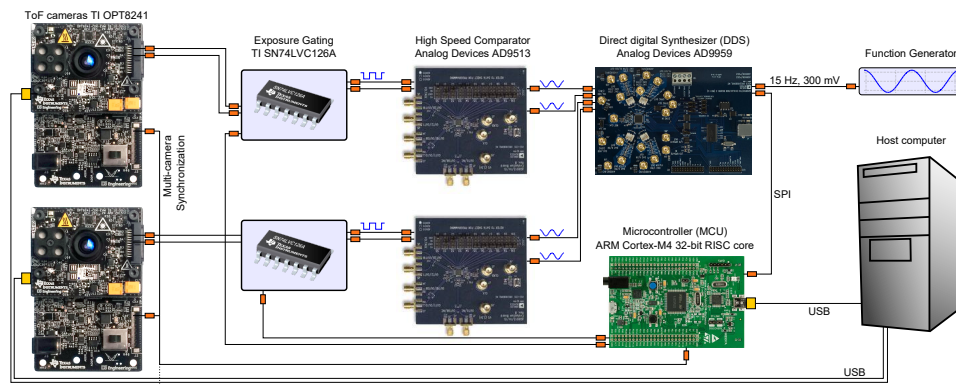
of the scene. Note that having additional depth estimates for conventional flow would also only be of limited help since flat surfaces also do not deliver enough features for correspondence matching.

Figure 7.16 shows a scene where the optical flow estimate is reasonable. In this case, the orthogonal component that our method captures completes the 2D spatial flow estimates and uniquely determines the full metric 3D flow. Given the optical flow estimates  $f_x, f_y$  for the horizontal and vertical image coordinates, one can compute the metric velocity vectors  $v_x = \frac{f_x \cdot Z}{F}, v_y = \frac{f_y \cdot Z}{F}$ , where  $F$  is the focal length of the lens and  $Z$  the corresponding depth estimate (see [254]). In conjunction with the velocity estimate  $v_z$  in the orthogonal direction along the optical axis, the full 3D metric flow is  $V = (v_x, v_y, v_z)$ . An example is shown in Figure 7.16. Please note that the 3D flow field is only as reliable as the estimated radial velocity and the RGB 2D flow. If one of them fails, so will the 3D flow.

## 7.7 Multi-Camera System

Previously, we have mentioned multi-camera CIS systems a few times as an extension to the proposed method that could remove remaining artifacts and extend the capabilities of our approach. In particular, two CIS cameras which share the same optical path allow for simultaneous capture of the two required captures per frame, thus eliminating the need for alignment. Furthermore, three cameras, each representing a stereo-pair with both other cameras, would allow for the capture of full 3D metric flow without needing to solve the ill-posed optical flow estimation problem. Stereo and multi-camera ToF arrays also have been shown to provide benefits for regular depth estimation [255–257].

However, systems for dynamic scenes have not been demonstrated before since synchronizing the exposures as well as the modulation waveforms is not straightforward. In this section, we present an extendable multi-camera TOF system that allows for synchronization, custom waveforms for each light source and camera. We also show that the orthogonal frequency modulation from this chapter can be used to eliminate multi-camera interference for depth imaging. An overview of the entire system is illustrated in Figure 7.17. We build on the Texas Instruments’



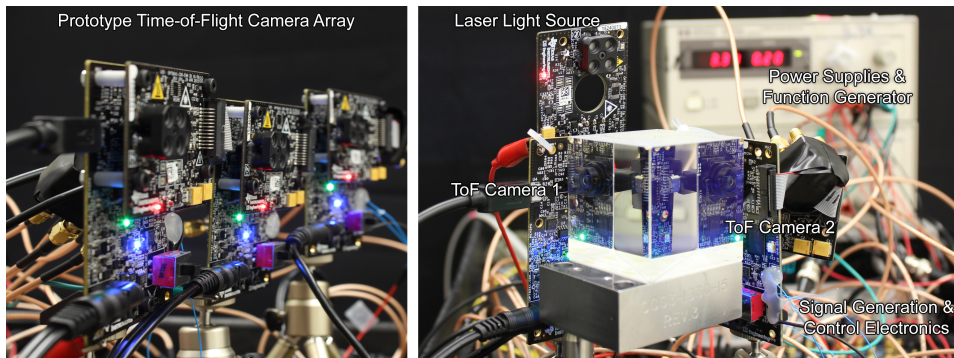
**Figure 7.17:** System overview. The proposed multi-camera time-of-flight system is built around the TI OPT8241 camera developer kit (left). All connected cameras are driven in slave mode, such that an external microcontroller (MCU) manages and synchronizes the sensor exposures of all cameras. The DDS generates analog waveforms that are digitized before being fed into the cameras. An additional exposure gating circuit guarantees that no residual signals are received unless exposing.

TOF camera development kit (CDK) OPT8241-CDK-EVM, or *Tin Tin*, and develop external control electronics for signal generation and synchronization. Tin Tin is a state-of-the-art TOF camera with sensor resolution of  $320 \times 240$ , providing external signal control as well as synchronization. In contrast to the previously discussed system from Chapters 3 and 5, which required complicated modifications to access modulation lines, this system has connections for external modulation signals. Tin Tin has a built-in laser light engine. At 850 nm, the diffused laser diodes offer a frequency of 12-80 MHz at 50% duty cycle. The on-board light engine can also be disabled and replaced by a custom light source, such as a projector which could provide spatio-temporally illumination as in Section 3.6.

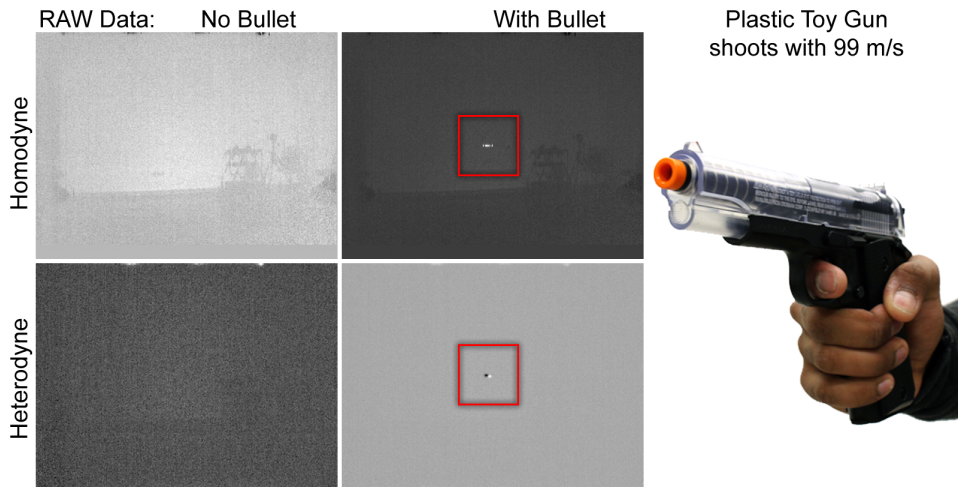
The waveform-generation circuitry in our system is built around Direct Digital Synthesis (DDS), which allows for precise waveform control in contrast to FPGA-based signal synthesis. We use an Analog Devices AD9959 four-channel DDS to generate sinusoidal waveforms with independent control over frequency, phase, and amplitude. Each of these four channels can control either a camera or light source. To digitize the analog DDS waveform, a clock fan-out buffer (Analog Devices AD9513) is used as a high-speed comparator. An additional high speed buffer chip (TI SNB8LVCFO0A) acts as an exposure gating mechanism, ensuring that no residual signal is received by the sensor unless exposing. We use an ARM Cortex M4 STM32F407VGT6 microcontroller to synchronize the exposures of all cameras and control the waveforms on the DDS. The microcontroller acts as the master for all connected devices. The frequency and phase settings of each frame in every channel can be controlled individually. Finally, the camera SDK allows to read out raw frames through the USB interface.

Figure 7.18 shows different configurations of our setup. For example, we can run the cameras in light field mode, where each sensor observes the scene from different perspectives or we can optically align them using a beam splitter so that they share the same perspective. Figure 7.19 shows a captured bullet in flight captured by the beam splitter setup. This is the same experiment as in Figure 7.11. We show the raw frames here to demonstrate our hardware setup. The homodyne and heterodyne frames showing the flying bullet are captured simultaneously and do not contain displacements, which caused the streak artifact in Figure 7.11.





**Figure 7.18:** Multi-camera prototype setup. The left shows an array of sensors observing the scene from different perspectives. On the right two cameras are optically aligned with a beam splitter. They both share the same perspective and single light source, but each of the sensors is modulated at different frequencies. In this mode, we can capture homodyne and heterodyne frames simultaneously.



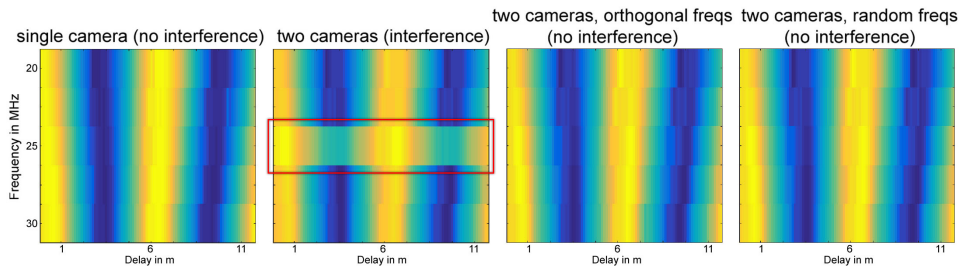
**Figure 7.19:** Example from Figure 7.11, but using the beam splitter setup shown on the right of Figure 7.18. Homodyne and heterodyne frames of a static scene are shown on the (left) and of the same scene with a bullet flying at about 99 m/s (center).



### 7.7.1 Multi-Source Interference Cancelling

The proposed hardware system also allows to eliminate multi-source interference for classical depth imaging. When multiple cameras illuminate the same scene, such as in Figure 7.18 on the left, then the temporally-coded illumination waveforms of each camera interfere with one another. Note, that this case differs significantly from the multi-path interference discussed in Section 2.6.5. Due to unsynchronized exposures and slight variation in the illumination frequencies, the multi-source interference results in a temporally beating measurement error. The frequency variation between different light sources is an inherent property of using independent sources which all have their own crystal oscillator. Using only a single light source would eliminate this problem (assuming synchronized exposures). In that case, however, parts of the imaged scene would never be illuminated, which defies the purpose of the different perspectives captured by the sensor array.

Previously, Castaneda et al. [255] proposed to capture all possible combinations of activated light sources, Li et al. [256] proposed to capture and average more than 100 frames to statistically mitigate multi-device interference. Neither option seems feasible for dynamic scenes. However, since the proposed hardware system allows to synchronize all exposures, simply changing the modulation frequencies of each camera/source pair by a large enough amount removes multi-source interference; see also Figure 7.20 on the right.



**Figure 7.20:** Multi-device interference. The left shows the reference phase and frequency sweep response of a single camera. Two CIS cameras, each running in homodyne mode, interfere at one particular frequency with one another (center left). Choosing an closely spaced orthogonal frequency pair with a difference of a few hundred Hz eliminates multi-device interference (center right), achieving the same effect as choosing two frequencies with MHz difference (right).

This is easy to see recalling Eq. (2.25) from Section 2.6: If we consider a camera/source pair with large enough difference in the modulation frequency, the continuous signal  $p(t)$  at the sensor only becomes high frequency and averages out in the integration. The smallest frequency component is exactly the difference between the light source and camera pair. Hence, selecting a large frequency difference between all camera/source pairs ensures low interference, but uses the available modulation spectrum very inefficiently, thus limiting the number of devices. The orthogonal frequency selection that we have used previously to amplify the tiny Doppler shift, allows for an efficient usage of the modulation spectrum while completely eliminating interference. In particular, we pick a frequency  $\omega_f = p \frac{2\pi}{T}$  for sensor demodulation, as well as mutually orthogonal frequencies  $l = 1 \dots M$  as  $\omega_l = (p + l - 1) \frac{2\pi}{T}$  for each light source. For this particular choice of illumination frequencies, the camera operates in homodyne mode with respect to the first light source, but in (orthogonal) heterodyne mode with respect to all the other lights. The orthogonality eliminates interference for a given integration time  $T$ , while allowing for closely spaced modulation frequencies. Figure 7.20 shows a calibration result for a modulation frequency of 22 MHz and 2 ms exposure time. The orthogonal frequencies are only separated by 508 Hz, but allow for perfect interference canceling, as achieved by MHz shifts.

We refer the reader to our publication [7] for a more detailed description of the presented multi-camera system and further applications of it which goes beyond the scope of this dissertation, such as high-speed range imaging, de-scattering of dynamic scenes and by non-line-of-sight motion detection via frequency gating.

## 7.8 Discussion

In this chapter, we propose a new computational imaging modality that directly captures radial object velocity via Doppler Time-of-Flight Imaging. We demonstrate a variety of experimental results for different types of settings. We have validated our model in simulation and experimentally. We have also demonstrated the optional combination of footage captured using an RGB camera with the depth and velocity output of our coded TOF camera. Together, this data represents simultaneous per-pixel RGB, depth, and velocity estimates of a scene and allows for the 3D velocity

field to be estimated. Finally, we have presented a hardware extension to our system enabling multi-camera/source capture.

Note that D-TOF is complimentary to optical flow. It allows for the depth bias of  $xz$ -flow to be removed and enables recording of the metric 3D velocity field of the scene. However, if only radial velocity is required, our technique can also be used stand-alone, independent of optical flow.

As a fundamentally new imaging modality, D-TOF implemented with our experimental hardware has several limitations. Foremost, the resolution of the PMD sensor in our prototype is limited to  $160 \times 120$  pixels and the SNR of the measured, Doppler-shifted signal is low. Together, these limitations result in low-resolution and noisy footage. We apply state-of-the-art denoising strategies which filter out most of the noise but do produce artifacts in some cases. Furthermore, D-TOF requires two frames to be acquired with different modulation frequencies. Currently, we capture these frames in sequence, which results in slight misalignment between the frames observed as velocity artifacts around depth discontinuities. However, as already mentioned in Section 3.7, there is a clear path to addressing all of these challenges: low-cost TOF sensors higher resolutions and significantly improved noise characteristics are already on the market. The multi-camera extension from Section 7.7 uses sensors with QVGA resolution. Some consumer cameras, such as the Microsoft<sup>®</sup> Kinect<sup>®</sup> 2, provide almost VGA resolution and also use higher modulation frequencies to increase the achievable depth resolution for range imaging. Note that increased sensor and illumination frequency would directly improve the signal-to-noise ratio in our setup, because the Doppler effect is proportional to the source frequency. With access to signal control of illumination and on-sensor modulation, D-TOF could be readily implemented on high-quality consumer TOF cameras. Nevertheless, D-TOF shares other limitations with TOF, including the need for active illumination, limited range, and problematic processing in the presence of strong ambient illumination or objects with dark albedos, and global illumination effects.

However, TOF cameras have entered the consumer market only a few years ago, but already transformed the way machines perceive the world. Human-computer interaction, robotics and machine vision, navigation for autonomous vehicles, and many other fundamental computer vision tasks have seen dramatic improvements

using these devices. With Doppler Time-of-Flight, we contribute a fundamentally new imaging modality that could impact all of these applications. By amplifying the tiny Doppler shift through carefully coded illumination and sensor modulation, our approach, much like Chapter 3, “makes the invisible visible”.

## Chapter 8

# Simple Lens Imaging

Up until now, the methods presented in this dissertation have exploited the temporally convolutional structure of the CIS measurements. In this chapter we investigate the spatially convolutional structure of aberrations in imaging optics. Optical systems and aberration theory, describing their imperfections, have been introduced in Section 2.3. Modern imaging optics are highly complex systems consisting of up to two dozen individual optical elements. This complexity is required in order to compensate for all geometric and chromatic aberrations of a single lens, including geometric distortion, field curvature, wavelength-dependent blur, and color fringing.

While traditional optical design attempts to minimize all of these aberrations equally, we demonstrate in the following that chromatic aberrations, in fact, efficiently encode scene information. Exploiting the specific structure of chromatic aberrations allows us to realize radically simplified optical designs, using cheap, simple, and compact optics.

We propose a Bayesian approach to color image deconvolution, which at its core relies on a novel statistical prior for the correlation between different spectral bands. For chromatic aberrations that allow to focus in at least one channel, this approach allows to efficiently correct images captured through uncompensated, simple optics. Finally, we present a method for the calibration of all aberrations of an optical system as a per-channel, spatially-varying PSF.

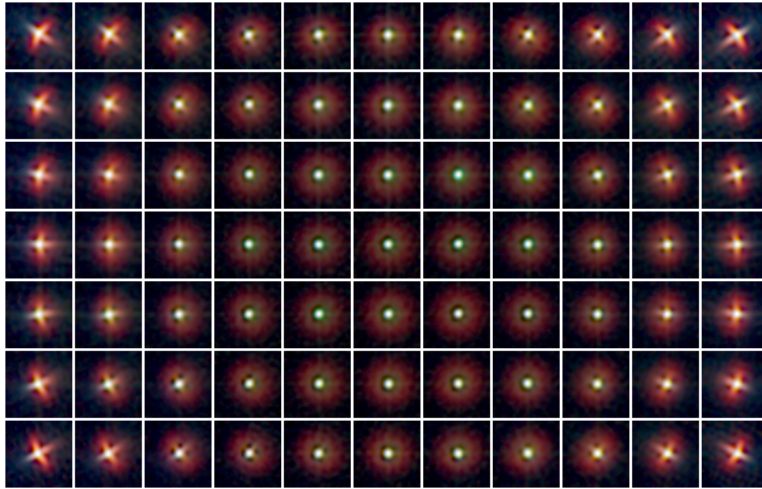


**Figure 8.1:** High-quality imagery through poorly performing lenses: Relying on the specific structure of chromatic aberrations, we present a Bayesian reconstruction method for the removal of aberrations post-capture. This enables imaging using simple lenses that suffer from severe aberrations. From left to right: Camera with our lens system containing only a single glass element (the plano-convex lens lying next to the camera), unprocessed input image, result estimated using the proposed approach.

## 8.1 Introduction

Over the past decades, camera optics have become increasingly complex. The lenses of modern single lens reflex (SLR) cameras may contain a dozen or more individual lens elements, which are used to optimize light efficiency of the optical system while minimizing *aberrations*, i.e., non-linear deviations from an idealized thin lens model. Optical systems and their aberrations have been described in detail in the related work part of this thesis. Recalling Section 2.3, aberrations include effects such as geometric distortions, chromatic aberration (wavelength-dependent focal length), spherical aberration (focal length depends on the distance from the optical axis), and coma (angular dependence on focus); see again Figure 2.5. All single lens elements with spherical surfaces suffer from these artifacts, and as a result cannot be used in high-resolution, high-quality photography. Instead, modern optical systems feature a combination of different lens elements with the intent of canceling out aberrations. For example, an achromatic doublet is a compound lens made from two glass types of different dispersion, i.e., their refractive indices depend on the wavelength of light differently. The result is a lens that is (in the first order) compensated for chromatic aberration, but still suffers from the other artifacts mentioned above.

Despite their better geometric imaging properties, modern lens designs are not



**Figure 8.2:** Patch-wise estimated PSFs for a single plano-convex lens at  $f/4.5$ . See Figure 2.6 for a single biconvex lens. The PSF estimation and non-blind deblurring in our method is done in patches to account for the PSFs’ spatial variance.

without disadvantages, including a significant impact on the cost and weight of camera objectives, as well as increased lens flare. In this chapter, we investigate a radically different approach to high-quality photography: instead of ever more complex optics, we propose to revisit much simpler optics, such as plano-convex lenses, used for hundreds of years (see, e.g., [258]), while correcting for the ensuing aberrations computationally.

The aberrations introduced by these simple lens designs follow a specific structure. Figure 2.6 and Figure 8.2 show the variation of the PSF over the image plane for two simple lens elements. We can make several observations: The blur is highly spatially varying, ranging from disc-like structures (spherical aberration) with diameters of 50 pixels and more to elongated streaks (coma and astigmatism). We can address this problem by subdividing the image into tiles over which we assume a constant PSF, as discussed in Section 2.3.3. The blur is also highly wavelength-dependent (chromatic aberration). This results in objectionable color fringing in the image. Note that these chromatic aberrations are much more severe than for partially corrected optics. For example, Kang [259] proposes a method specifically

for removing color fringing in images. However, this method is based on edge detection, which is feasible for images taken with partially corrected optics, but is impossible in the presence of very large blurs that result from the use of uncorrected optics.

Although the aberrations are severe, at the same time, the PSF of at least one of the color channels often contains more energy in high spatial frequencies than the others (one channel is usually focused significantly better than the others); note here that we do not require it to be perfectly in focus. This suggests that we can exploit cross-channel correlation and reconstruct spatial frequencies that are preserved in at least one of the channels.

In our approach we rely on a novel cross-channel prior that encodes this correlation between different spectral channels. Following our Bayesian approach from the related work Section 2.8.1, we formulate a deconvolution method which efficiently incorporates this prior in a convex optimization problem. We derive a proximal algorithm to solve this problem which is guaranteed to converge to a global optimum. The proposed approach is therefore able to handle much larger and more dense PSFs than previous methods (see, e.g., [165]), such as disk-shaped kernels with diameters of 50 pixels and more, which occur frequently in uncorrected optics unless the apertures are stopped down to an impractical size. To calibrate the kernels for a given optical system, we present a method for robust per-channel spatially-varying PSF estimation based following the same Bayesian approach as for the deconvolution problem, but treating the PSF as an unknown. Our novel cross-channel prior, which is at the core of our proposed method, can also be combined with patch-based priors, and in fact improves the robustness of patch-based reconstruction methods, which will be described in Chapter 9.

In the following, we first review the image formation model including spatially varying aberrations. Subsequently our novel cross-channel prior for color images is derived. Next, the robust Bayesian approach, and solver method for the resulting optimization problem are described. We demonstrate that our method enables high-quality photographs on modern 12 megapixel digital SLRs using only single lens elements such as plano convex or biconvex lenses, and achromatic doublets. We show high quality results comparable to conventional lenses, with apertures around  $f/4.5$ . Furthermore, we present an extension of our method that allows for color



imaging even with highly chromatic diffractive elements, such as ultra-thin Fresnel phase plates. Finally, we demonstrate that our formulation outperforms competing deconvolution approaches in terms quality and runtime. Using our method, simplified optics become viable alternatives to conventional camera objectives.

## 8.2 Image Formation Model

The color image formation in the presence of spatially varying aberrations has been described in detail in Section 2.3.3. Recalling Eq. (2.12), we can assume aberrations to be spatially invariant in a small  $n \times m$  tile on the sensor. Using  $\mathbf{j} \in \mathbb{R}^{nm}$  and  $\mathbf{i} \in \mathbb{R}^{nm}$  as the vectorized observed blurred image and latent sharp image, respectively, the image formation can be formulate in the matrix-vector form

$$\mathbf{j} = \mathbf{B}\mathbf{i} + \mathbf{n}, \quad (8.1)$$

where  $\mathbf{B} \in \mathbb{R}^{nm \times nm}$  is the matrix corresponding to the convolution with the PSF  $B$  in the tile, and  $\mathbf{n}$  represents sensor noise. As discussed in Section 2.3.3, the full image will be composed of many tiles, each with a corresponding PSF.

## 8.3 Deconvolution Using Cross-Channel Correlation

### 8.3.1 Cross-Channel Correlation

Real optical systems suffer from dispersion in the lens elements, leading to a wavelength dependency of the PSF known as *chromatic aberration*.

While complex modern lens assemblies are designed to minimize these artifacts through the use of multiple lens elements that compensate for each others' aberrations, it is worth pointing out that even very good lenses still have a residual amount of chromatic aberration. For the simple lenses we aim to use, the chromatic aberrations are very severe – one color channel is focused significantly better (although never perfectly in focus) than the other channels, which are blurred beyond recognition (of course excluding achromatic lenses which compensate for chromatic aberrations).

Given individual PSFs  $B_{\{1..3\}}$  for each color channel  $J_{\{1..3\}}$  of an image  $J$



**Figure 8.3:** Effect of using the cross-channel prior for a simulation using a strongly chromatic blur (in the very top left) Left: Blurred and noisy observation. Middle: Independently deconvolved results for each channel using [160]. Right: Deconvolution with our cross channel prior.

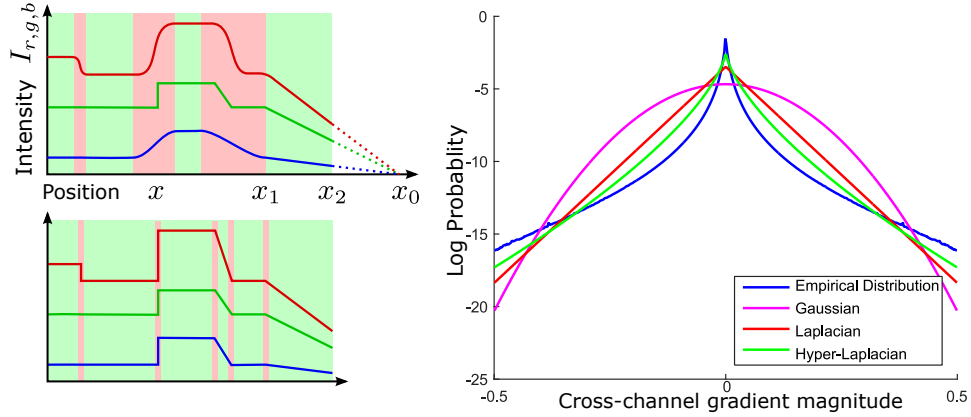
one might attempt to independently deconvolve each color channel. As Figure 8.3 demonstrates, this approach does not in general produce acceptable results, since frequencies in some of the channels may be distorted beyond recovery, leading to the severe red fringing and residual blur in this case.

We propose to share information between the deconvolution processes of the different channels, so that frequencies preserved in one channel can be used to help the reconstruction in another channel. The key observation of our cross-channel prior is that for a latent (sharp) image, changes in chroma and luma are sparsely distributed in natural images. That means, in most areas the gradients of different channels are close, except for few large changes in chroma and luma (e.g. commonly at object or material boundaries); see the left of Figure 8.4. In other words, for a pair of channels  $l, k$  we assume

$$\begin{aligned} \nabla \mathbf{i}_k / \mathbf{i}_k &\approx \nabla \mathbf{i}_l / \mathbf{i}_l \\ \Leftrightarrow \nabla \mathbf{i}_k \cdot \mathbf{i}_l &\approx \nabla \mathbf{i}_l \cdot \mathbf{i}_k, \end{aligned} \quad (8.2)$$

where the errors in the approximation are sparsely distributed around edges in the image. Note that the division and multiplication  $/, \cdot$  are pixel-wise operators. Hence, we model the difference in the normalized gradients with a heavy-tailed prior:

$$p(\mathbf{i}_c | \gamma, \mathbf{i}_l) \propto \exp \left( -\gamma \sum_{l \neq c} \sum_a \|\mathbf{H}_a \mathbf{i}_c \cdot \mathbf{i}_l - \mathbf{H}_a \mathbf{i}_l \cdot \mathbf{i}_c\|_1 \right), \quad (8.3)$$



**Figure 8.4:** Blurred scanline on the top left with different PSFs in each channel and a sharp green channel. Reconstructed scanline on the bottom left. Our cross-channel prior enforces gradient consistency between the color channels (green shaded regions) and allows sparse hue changes (red shaded regions). The right shows the cross-channel gradient statistics from 10,000 randomly selected images from the Imagenet data set [260]. The empirical distribution is heavy-tailed.

for a considered channel  $x$ . The matrices  $\mathbf{H}_a$  are different gradient operators, which will be described in detail below.

Note that the proposed prior places stronger assumptions (normalized consistent gradients) on the color-channel coupling compared to methods that assume only consistent gradients, such as [261]. This stronger assumption is verified empirically using of 10,000 images from the Imagenet data set [260]. The right of Figure 8.4 shows the empirical distribution of the accumulated cross-channel gradients, that is  $\tilde{c} = \mathbf{H}_a \mathbf{i}_c \cdot \mathbf{i}_l - \mathbf{H}_a \mathbf{i}_l \cdot \mathbf{i}_c$  for all images, channels and gradient filters, where  $\mathbf{i}_c$  is here the red or blue channel and  $\mathbf{i}_r$  is the green channel. The plot also shows a Gaussian fit  $p(\tilde{c}) \propto e^{-\gamma|\tilde{c}|^2}$ , Laplacian fit  $p(\tilde{c}) \propto e^{-\gamma|\tilde{c}|}$  and Hyper-Laplacian  $p(\tilde{c}) \propto e^{-\gamma|\tilde{c}|^{\frac{2}{3}}}$ . We can see that the empirical cross-channel gradient distribution is heavy-tailed and thus not well approximated by a normal distribution. While a Hyper-Laplacian fit best models the underlying distribution, the Laplacian is the best *convex* relaxation of the fitted distribution. We therefore chose in Eq. (8.3) the Laplacian distribution as the best convex fit.

### 8.3.2 Inverse Problem

To invert the image formation model from Eq. (8.2) using the described prior and including noise, we follow again the Bayesian MAP approach from Section 2.8.3, as for all inverse problems discussed in this dissertation. Assuming  $\mathbf{n}$  to be Gaussian distributed, we formulate the problem of jointly deconvolving all channels as

$$\begin{aligned}
 (\mathbf{i}_{1\dots 3})_{\text{opt}} = \underset{\mathbf{i}_{1\dots 3}}{\text{argmin}} & \sum_{c=1}^3 \|\mathbf{B}_c \mathbf{i}_c - \mathbf{j}_c\|_2^2 + \lambda_c \sum_{a=1}^5 \|\mathbf{H}_a \mathbf{i}_c\|_1 \\
 & + \sum_{l \neq c} \beta_{cl} \sum_{a=1}^2 \|\mathbf{H}_a \mathbf{i}_c \cdot \mathbf{i}_l - \mathbf{H}_a \mathbf{i}_l \cdot \mathbf{i}_c\|_1,
 \end{aligned} \tag{8.4}$$

with the first term being the Data Fidelity term, which results from the Gaussian likelihood. The second term enforces a heavy-tailed distribution for both gradients and curvatures of all channels. The gradient matrices  $\mathbf{H}_{\{1,2\}}$ , implement the first derivatives, while  $\mathbf{H}_{\{3\dots 5\}}$  correspond to the second derivatives. We use the same kernels as Levin et al. [161] but employ an  $\ell_1$  norm in our method rather than a fractional norm. This ensures that our problem is convex. The last term of Eq. (8.4) implements our cross-channel prior, resulting from the prior distribution Eq. (8.3).  $\lambda_c, \beta_{cl} \in \mathbb{R}$  with  $c, l \in \{1 \dots 3\}$  are weights for the image prior and cross channel prior terms, respectively.

### 8.3.3 Optimization

The joint objective from Eq. (8.4) for all channels is a hard non-convex problem. To make it feasible and efficient to solve, we minimize it by coordinate descent over the channels. That means, we alternately minimize with respect to one channel while fixing all the other channels. Similar to the optimization for CSC from Section 6.2.3, the resulting objective in each step becomes then convex.

Note that for the specific case of a given sharp channel, the objective Eq. (8.4) is separable per channel. In this case, all channels can be solved jointly leading to a complexity of our method growing linearly with the number of channels. All considered simple optics in this chapter offer an almost sharp channel for a given channel. The coordinate descent approach discussed above can be applied for

general reconstruction problems. Minimizing Eq. (8.4) with regard to a single channel  $c$  can be formulated as the problem:

$$\begin{aligned}
& (\mathbf{i}_c)_{\text{opt}} \\
&= \underset{\mathbf{x}}{\operatorname{argmin}} \|\mathbf{B}_c \mathbf{x} - \mathbf{j}_c\|_2^2 + \lambda_c \sum_{a=1}^5 \|\mathbf{H}_a \mathbf{x}\|_1 + \sum_{l \neq c} \beta_{cl} \sum_{a=1}^2 \|\mathbf{H}_a \mathbf{x} \cdot \mathbf{i}_l - \mathbf{H}_a \mathbf{i}_l \cdot \mathbf{x}\|_1 \\
&= \underset{\mathbf{x}}{\operatorname{argmin}} \|\mathbf{B}_c \mathbf{x} - \mathbf{j}_c\|_2^2 + \lambda_c \sum_{a=1}^5 \|\mathbf{H}_a \mathbf{x}\|_1 + \sum_{l \neq c} \beta_{cl} \sum_{a=1}^2 \|(\mathbf{D}_{\mathbf{i}_l} \mathbf{H}_a - \mathbf{D}_{\mathbf{H}_a \mathbf{i}_l}) \mathbf{x}\|_1 \\
&= \underset{\mathbf{x}}{\operatorname{argmin}} \|\mathbf{B}_c \mathbf{x} - \mathbf{j}_c\|_2^2 + \|\mathbf{S} \mathbf{x}\|_1 \quad \text{with} \\
& \mathbf{S} = \begin{bmatrix} \lambda_c \mathbf{H}_1 \\ \vdots \\ \lambda_c \mathbf{H}_5 \\ \beta_{cl} (\mathbf{D}_{\mathbf{i}_1} \mathbf{H}_1 - \mathbf{D}_{\mathbf{H}_1 \mathbf{i}_1}) \\ \vdots \\ \beta_{cl} (\mathbf{D}_{\mathbf{i}_2} \mathbf{H}_2 - \mathbf{D}_{\mathbf{H}_2 \mathbf{i}_2}) \\ \vdots \end{bmatrix}, \tag{8.5}
\end{aligned}$$

where  $\mathbf{D}$  denotes the diagonal matrix with the diagonal taken from the subscript.  $\mathbf{S}$  is the matrix consisting of the sequence of all  $t = 5 + 2(3 - 1)$  matrices coming from the  $\ell_1$  minimization terms in Eq. (8.4). In Eq. (8.5) it becomes clear that our deconvolution problem is of the form discussed in Section 2.8.3. In particular, we use

$$\begin{aligned}
\Phi &= \mathbb{I}, & G(\mathbf{v}) &= \|\mathbf{B}_c \mathbf{v} - \mathbf{j}_c\|_2^2, \\
\Omega &= \mathbf{S}, & F(\mathbf{v}) &= \|\mathbf{v}\|_1,
\end{aligned} \tag{8.6}$$

So, since our problem is expressible as Eq. (2.40), we can solve it using a proximal algorithm as described in Section 2.8.4. We use the Chambolle-Pock method, as in Chapter 3. This method is equivalent to a linearized version of ADMM, which has been described in Chapter 6. We discuss proximal algorithms, including the Chambolle-Pock method in detail in Chapter 10. To solve our problem with Chambolle-Pock's method, the proximal operators for  $F^*$  and  $G$  have to be given.

These operators are

$$\mathbf{prox}_{\tau F^*}(\mathbf{v}) = \frac{\mathbf{v}}{\max(1, |\mathbf{v}|)} \quad (8.7)$$

$$\begin{aligned} \mathbf{prox}_{\tau G}(\mathbf{v}) &= \tilde{\mathbf{x}} = \underset{\mathbf{x}}{\operatorname{argmin}} \|\mathbf{B}_c \mathbf{x} - \mathbf{j}_c\|_2^2 + \frac{1}{2\tau} \|\mathbf{x} - \mathbf{v}\|_2^2 \\ &\Leftrightarrow \frac{1}{\tau}(\tilde{\mathbf{x}} - \mathbf{v}) + 2(\mathbf{B}_c^T \mathbf{B}_c \tilde{\mathbf{x}} - \mathbf{B}_c^T \mathbf{j}_c) = 0 \\ &\Leftrightarrow \tilde{\mathbf{x}} = \mathfrak{F}^{-1} \left( \frac{\tau 2 \mathfrak{F}(\mathbf{B}_c)^* \mathfrak{F}(\mathbf{j}_c) + \mathfrak{F}(\mathbf{v})}{\tau 2 |\mathfrak{F}(\mathbf{B}_c)|^2 + 1} \right), \end{aligned} \quad (8.8)$$

The first proximal operator is a projection on the  $\ell_1$ -norm ball and described in detail in [178]. Note that the division is here point-wise. The second proximal operator is the solution to a linear system as shown in the second line. Since the system matrix is composed of convolution matrices with a large support we can efficiently solve this linear system in the Fourier domain (last line).

*Parameters* Adopting the notation from [178], we use  $\theta = 1$ ,  $\sigma = 10$  and  $\tau = 0.9/(\sigma L^2)$  for our specific Chambolle-Pock method with Eq. (8.7),(8.8). That only leaves the operator norm  $L = \|\Omega\|$  which is required for the method. We find the value  $L$  by using the power iteration where all matrix-vector-multiplications with  $\mathbf{S}^T \mathbf{S}$  can be decomposed into filtering operations (other eigenvalue methods like the Arnoldi iteration are consequently also suited for computing  $L$ ).

### 8.3.4 Low-Light Imaging

For image regions with very low intensity values, the method described above fails. This is since the cross-channel prior from Eq. (8.3) is not effective due to the hue normalization, which reduces the term near to zero. As a result, significant color artifacts (such as color ringing) can remain in very dark regions, see Figure 8.5. Note that by allowing luma gradients in the prior from Eq. (8.3) this is an inherent design problem of this prior and not an optimization issue.

In this scenario, we therefore propose to match absolute (rather than relative) gradient strengths between color channels using a Gaussian distribution. The additional prior can be expressed as a modification of operator  $G$  from (8.6), which



**Figure 8.5:** Chromatic artifacts in low-light areas. Left: Original blurred patch captured with a plano-convex lens. Center: Reconstruction using Eq. (8.4). Notice the chromatic artifacts in all low-intensity areas and the correct reconstruction in the other areas. Right: Regularization for low-intensity areas added as discussed below.

becomes now

$$G(\mathbf{v}) = \|\mathbf{B}_c \mathbf{v} - \mathbf{j}_c\|_2^2 + \lambda_b \sum_{l \neq c} \sum_{a=1}^2 \|\mathbf{D}_w (\mathbf{H}_a \mathbf{v} - \mathbf{H}_a \mathbf{i}_l)\|_2^2 \quad (8.9)$$

where  $\text{diag}(\mathbf{w})$  is a spatial mask that selects dark regions below a threshold  $\epsilon$ . The mask is blurred slightly with a Gaussian kernel  $B_\sigma$  with standard deviation  $\sigma = 3$  to avoid discontinuities at border regions affected by the additional prior:

$$\mathbf{w} = \left(1 - \frac{\sum_l \beta_{cl} T(\mathbf{i}_l)}{\sum_l \beta_{cl}}\right) \otimes B_\sigma \quad \text{with} \quad T(\mathbf{i})_k = \begin{cases} 1 & ; \mathbf{i}_k \leq \epsilon \\ 0 & ; \text{else} \end{cases}, \quad (8.10)$$

where  $T$  is here a simple thresholding operator with threshold  $\epsilon = 0.05$  and in our implementation. The proximal operator from Eq. (8.8) becomes then

$$\begin{aligned} \text{prox}_{\tau G}(\mathbf{v}) = \tilde{\mathbf{x}} &= \underset{\mathbf{x}}{\text{argmin}} \|\mathbf{B}_c \mathbf{x} - \mathbf{j}_c\|_2^2 + \frac{1}{2\tau} \|\mathbf{x} - \mathbf{v}\|_2^2 + \\ &\quad \lambda_b \sum_{l \neq c} \sum_{a=1}^2 \|\mathbf{D}_w (\mathbf{H}_a \mathbf{v} - \mathbf{H}_a \mathbf{i}_l)\|_2^2 \\ &\Leftrightarrow \left[ 2\tau \mathbf{B}_c^T \mathbf{B}_c + \mathbb{I} + 2\tau \lambda_b \sum_{l \neq c} \sum_{a=1}^2 \mathbf{H}_a^T \mathbf{D}_w^2 \mathbf{H}_a \right] \tilde{\mathbf{x}} \\ &= 2\tau \mathbf{B}_c^T \mathbf{j}_c + \mathbf{v} + 2\tau \lambda_b \sum_{l \neq c} \sum_{a=1}^2 \mathbf{H}_a^T \mathbf{D}_w^2 \mathbf{H}_a \mathbf{i}_l. \end{aligned} \quad (8.11)$$

The linear system from the last row cannot be solved efficiently in the Frequency domain anymore, since the spatially diagonal operator  $\mathbf{D}_w^2$  does not diagonalize in the Frequency domain. Since blur kernel sizes of the order of magnitude of  $10^2 \times 10^2$  can be expected for practical applications, the linear system is not sparse, nor separable and therefore impractical to invert explicitly. We solve the system using the Conjugate Gradient (CG) algorithm. This allows us to express the matrix-vector multiplication in CG-algorithm as a sequence of filtering operations as before.

*Residual and Scale-Space Deconvolution* The deconvolution approach described above is used in a residual framework which we adopt from [262]. By adapting the prior weights in sequential passes of our deconvolution method, residual deconvolution makes our basic deconvolution approach more robust to the parameter selection. This reduces artifacts in regions with strong gradients. Furthermore, we accelerate our method using a scale-space approach as in [263]. The residual and scale-space deconvolution technique is described in detail in the Appendix Section A.9, including a discussion on how we handle saturated regions in the residual deconvolution.

## 8.4 Results

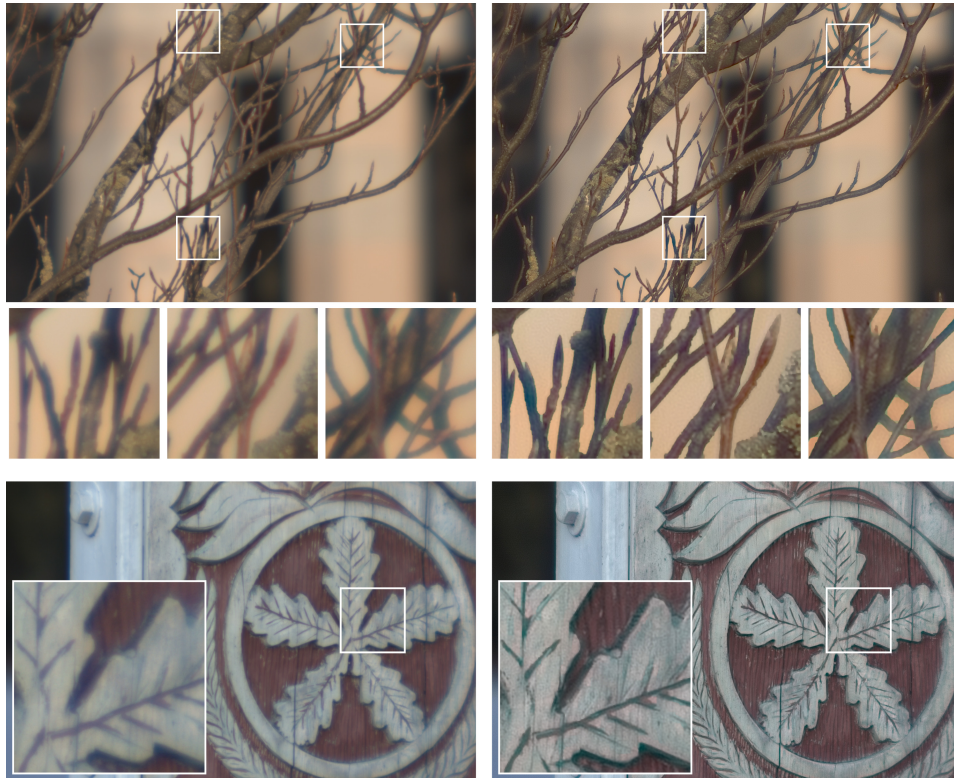
In this section we present a number of results captured with different optical systems. Specifically, we show captures using simple lens elements as well as standard consumer lens systems, and finally present an extension of our method to ultra-thin diffractive optical elements. We demonstrate full-spectrum color results and investigate multi-spectral imaging with our approach. The spatially varying PSFs of the simple lens elements are estimated using two captures of a planar target at different aperture settings. Note that any PSF estimation method can be used here and the whole estimation process is a calibration procedure, which only needs to be performed once for each lens. Our PSF estimation technique, which only requires targets printed on consumer laser-printers, is described in Section A.10. Note also that PSFs from Figure 2.6 and Figure 8.2 have been calibrated using our method.





**Figure 8.6:** Images captured with a simple plano-convex lens (left) and restored with our method (right). Note recovered spatial detail and the absence of color fringing. Also note the bokeh and graceful handling of out-of-focus blur in the top three images.

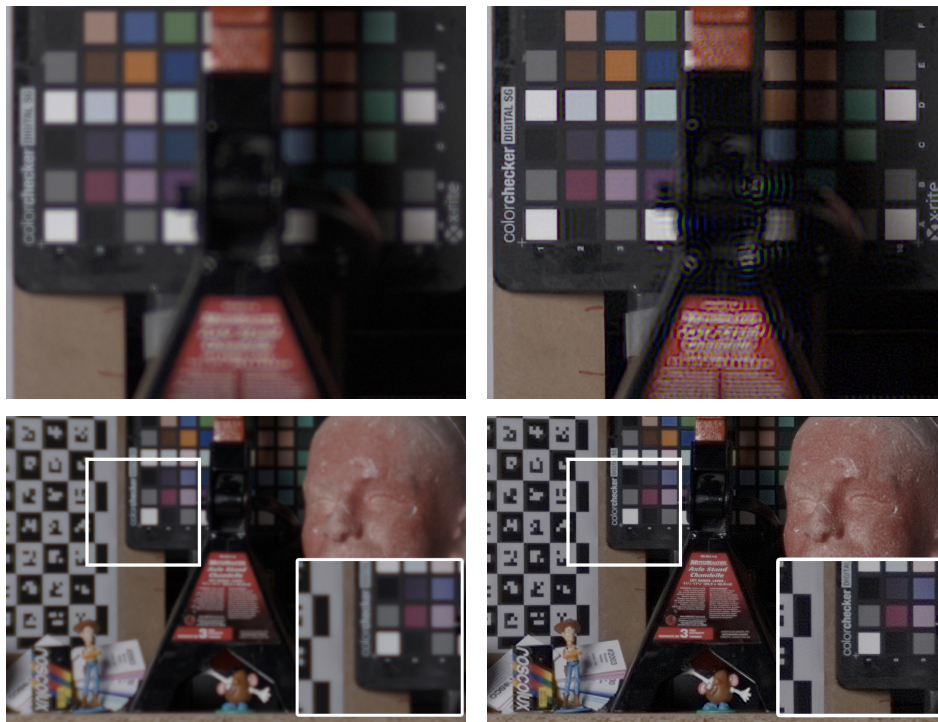
*Simple Lens Elements* Figure 8.1, Figure 8.6 and Figure 8.7 show several results from the most basic refractive optical system, a single plano-convex lens (focal length 130 mm, f/4.5) shown lying on the side in Figure 8.1. These results are captured with a Canon EOS D40 camera, using standard ISO100 and autoexposure settings. The corresponding PSFs are the ones from Figure 8.2. We note that all results show good detail down to the native sensor resolution, demonstrating that our



**Figure 8.7:** Additional scenes captured using the simple plano-convex lens as in Figure 8.6.

method is indeed capable of producing high quality digital photographs with very simple lens designs. The parking meter image and the twigs show that deconvolving with the PSF for one particular scene depth preserves the deliberate defocus blur (bokeh) for objects at different depths without introducing artifacts. The manhole cover image at the top shows focus change over a tilted plane. Saturated regions as shown in the second inset of this image are handled properly by our method. It also includes a result under low-light conditions.

Figure 8.8 test the limits of our method using lenses with a larger aperture of  $f/2$ . The top of Figure 8.8 shows the results for a 100 mm biconvex lens, with PSFs from Figure 2.6. The larger aperture increases the depth dependency of the PSF. Therefore our deconvolution method produces very sharp reconstructions in-plane

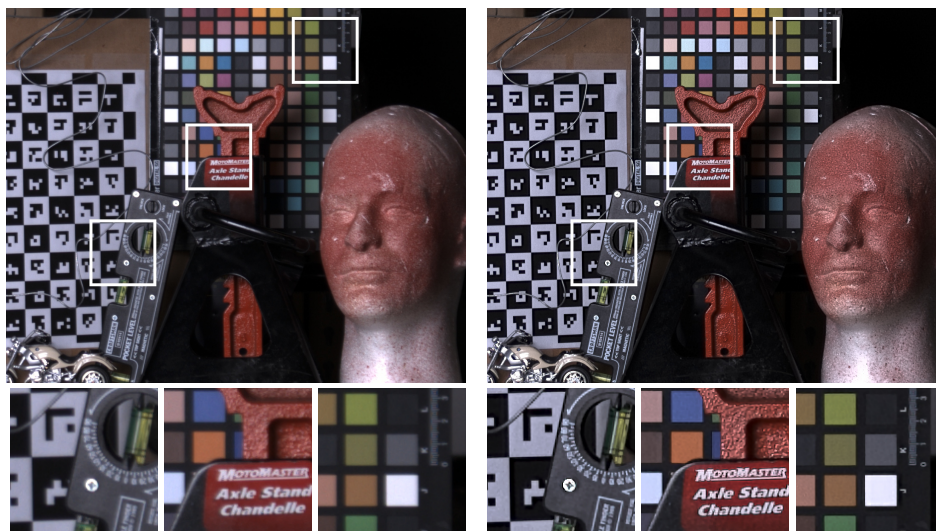


**Figure 8.8:** Limitations of our approach. The left shows captured image, and the right shows the deblurred result. Top: Results from biconvex lens at  $f/2$ . A very low cross-prior weight has been used here to illustrate the depth-dependency of the PSF. Bottom: Cropped central region of the image from achromatic lens with  $f/2.0$ .

(see color checker), but suffers from some ringing out-of-plane (see text on the foreground object). The bottom result of Figure 8.8 shows results for a 100 mm achromatic doublet, which does not contain spectrally varying blur. Considering that the PSFs exhibit disk-like structures in all channels, our deconvolution method still achieves reasonable quality, however cannot rely on the cross-channel prior.

*Commercial Lens Systems* Results for a commercial camera lens, a Canon 28–105 mm zoom lens at 105 mm and  $f/4.5$ , are shown in Figure 8.9 and Figure A.18 from the Appendix Section A.11. We use the Canon EOS D40 with the same settings as before. While this commercial lens shows much reduced aberrations compared to





**Figure 8.9:** Test scene captured with a commercial Canon lens (105 mm,  $f/4.5$ ).  
 Left: Captured input image. Right: Deblurred result.

the uncorrected optics used above, there still are some residual blurs that can be removed with our method. In particular, the vertical edges of the calibration pattern in Figure A.18 reveal a small amount of chromatic aberration that is removed by our method. The PSFs for this lens are around  $11 \times 11$  pixels in diameter.

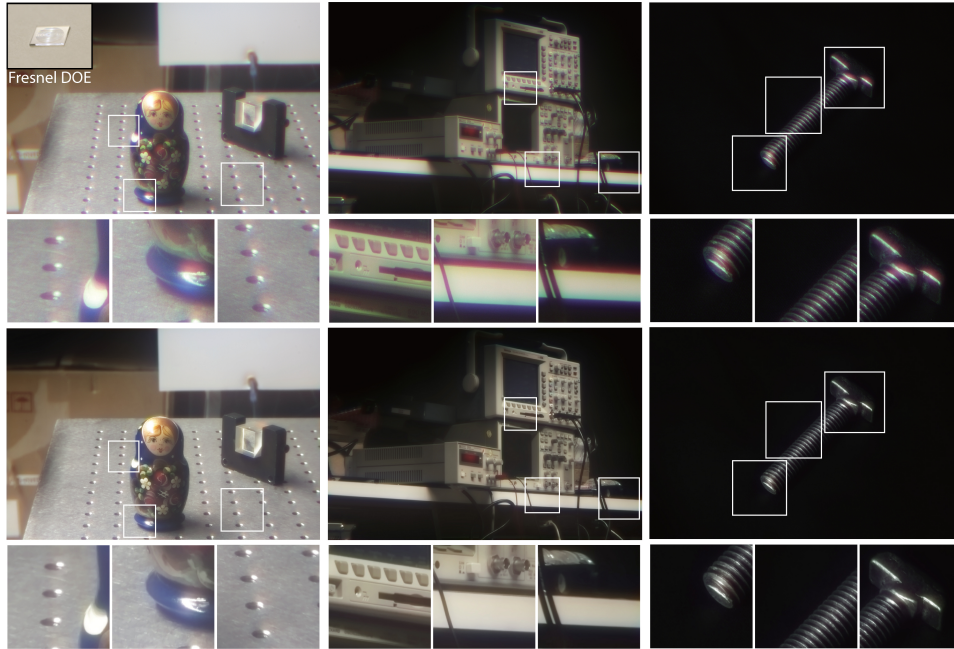
*Multi-Spectrum Imaging* Our aberration correction method cannot only be used for RGB imaging but also for multi-spectral imagers. In this case, the cross-channel prior is applied between all frequency bands pairs. Results are shown in Section A.11.

Considering the wavelength-dependent PSFs, it is important to note that the assumption of fixed PSFs for each color-channel of an RGB-sensor is often violated. This assumption is made for all the RGB sensor results in this chapter and is a classical assumption in deconvolution literature. However, one cannot tell from a tri-chromatic sensor the exact wavelength distribution. Metamers (different spectra that produce the same tri-stimulus response) will have different blur kernels, so there can always be situations where the assumption of fixed per-channel PSFs will fail, such as for example light sources with multiple narrow spectra or albedos with

very narrow color tuning. This will introduce errors in the data-fitting term of our objective function, which to a certain extent our cross-channel prior can account for. In addition to the simulated result from Figure 8.3, we have compared our method to state-of-the-art deconvolution methods for measured data in Section A.11. The additional results demonstrate that even for incorrectly estimated PSFs our method can still provide plausible estimates.

*Diffractive Optical Elements* However, for even further increased chromatic aberrations this approach fails. Diffractive optical elements (DOEs) introduce highly wavelength-dependent aberrations due to the inherent dispersion. Only relying on diffraction, such elements can be fabricated on ultra-thin planes, and hence represent a promising avenue for reducing the size and complexity of conventional lenses even further (compared to the simple refractive lenses from above). Furthermore, diffractive optics allow for a very flexible optical design since complex phase plates can be fabricated using photo lithograph, while, in contrast, high-quality refractive optics rely on spherical optics (recall Section 2.3). Recent examples include cubic phase plates for extended depth of field imaging [264], DOEs producing double-helix PSFs [20] for single-molecule microscopy beyond the diffraction limit, and anti-symmetric gratings integrated with a CMOS sensor to produce an ultra-miniature lens-less imager PicoCam [265]. Figure 8.10 shows captures acquired using a diffractive optical element which implements a Fresnel lens that is designed for the wavelength of 555 nm (center of green band). No other optical elements are used, and the results are captured with a 4 megapixel Canon 70D DSLR sensor with the same settings as before. For this optical system any incident wavelength except the single design wavelength will not be focused, depending on the spectral distributions of objects and illumination. Hence, the captures in Figure 8.10 exhibit extreme chromatic aberrations with blur kernels of diameters well over 100 pixels. Note that the PSFs are also highly spatially varying due to varying material/illumination properties.

In order to handle such aberrations, we cannot rely on pre-calibration, but estimate the spatially varying PSFs in a blind deconvolution approach only from the captured image. Using cross-channel information in the blind estimation turns out to be key for achieving good PSF estimation results, as classical methods such



**Figure 8.10:** Results capture with an ultra-thin diffractive optical element (top left inset), implementing a Fresnel phase plate optimized for the single wavelength 555 nm (green). A set of representable scenes under indoor lighting conditions are shown here. For each individual result, we compare the aberrated observation (top) and a cross-channel reconstruction (bottom). Small insets below each image highlight details and illustrate how our method removes the severe chromatic aberrations.

as [266–268] fail for the large kernels. Reconstruction results using the blind estimation are shown in Figure 8.10. To increase efficiency, we use here a slight reformulation of the prior from Eq.(8.3) without the luma normalization. While reaching almost similar reconstruction quality, the reformulation leads to 5 orders of magnitude reduced runtime, which is a critical aspect to make blind estimation methods practical. We refer the reader to our publication [9] for further details on full-spectrum imaging with diffractive optics, including an in-depth discussion of the blind extension to our cross-channel reconstruction from this chapter, which goes beyond the scope of this dissertation.

## 8.5 Discussion

In this chapter, we have exploited the structure of aberrations in imaging optics. In particular, we have presented a Bayesian method for color image deconvolution relying on the specific structure of chromatic aberrations that focus well in at least one channel. We exploit statistical correlation between the focused and the other channels to recover severely degraded color channels, even with disk-like blur kernels of diameters of 50 pixels and more. This enables high-quality imaging using very simple, uncompensated lens elements instead of complex lens systems. Going a step further, we have demonstrate that our approach can even be extended for broadband imaging with ultra-thin diffractive optics.

Overall our method produces image quality comparable to that of commercial point-and-shoot cameras even with the most basic refractive and diffractive elements: a single plano-convex lens and a Fresnel phase plate. However, for large apertures of  $f/2$  and more, the quality degrades due to the depth dependency of the PSF and significantly larger blur kernels, which destroy much of the frequency content. Furthermore, the scene and illumination determine the incident spectrum and hence make the PSF estimation challenging. While a blind extension of our approach (as discussed for the diffractive experiments) mitigates these issues, our method does still achieve the quality level of a high-end lens in combination with a good SLR camera. We conclude that in order to achieve that level of image quality it may still be necessary to optimize the lens design to partially compensate for aberrations. In particular the combination of thin refractive and diffractive elements is a promising direction for very flexible, new hybrid optical designs. Our work demonstrates that future lenses can be simpler than they are today, thus paving the road for lighter, cheaper, and more compact camera lenses.

Our method removes aberrations post-capture assuming that the optics are designed in a separate process to the best possible quality. I believe that joint optimization of the optics together with the image reconstruction can lead to exciting new areas in the optical design space, going far beyond first attempts such as [269], since the the freedom in the optical design is inherently limited by the achievable quality of the reconstruction method. Note also that this is in strong contrast to classical optics design which solely aims at minimizing the spot-size.

## Chapter 9

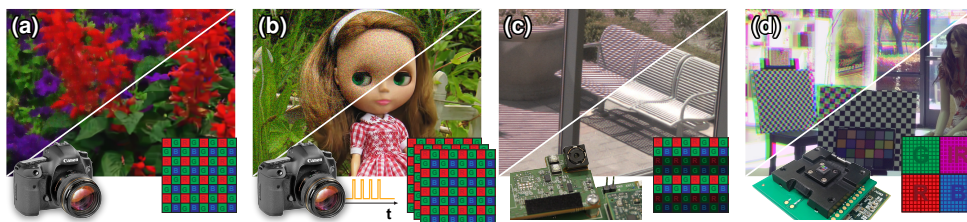
# Flexible Camera Image Processing

While the previous chapter has demonstrated that computation enables new optical designs by removing chromatic aberrations, we generalize this idea in the following and show that it enables a variety of new computational camera designs which outperform classical color imaging systems.

Color image processing is traditionally organized as a series of cascaded modules, each responsible for addressing a particular problem. Demosaicking, denoising, deconvolution and tone-mapping represent a subset of these problems, which are usually treated by independent pipeline stages; see Section 2.4.4. While this divide-and-conquer approach offers many benefits, it also introduces a cumulative error, as each step in the pipeline only considers the output of the previous step, not the original sensor data.

We propose to completely replace such pipelines by Bayesian inference. In particular, building on Section 2.8.1, we propose an end-to-end system that is aware of the camera and image model, enforces natural-image priors, while jointly accounting for common image processing steps like demosaicking, denoising, deconvolution, and so forth, all directly in a given output representation (e.g., YUV, DCT). Our system is flexible and we demonstrate it on regular Bayer images as well as emerging array sensors, such as interleaved HDR, and low-light burst imaging. In all cases, we achieve large improvements in image quality and signal reconstruction





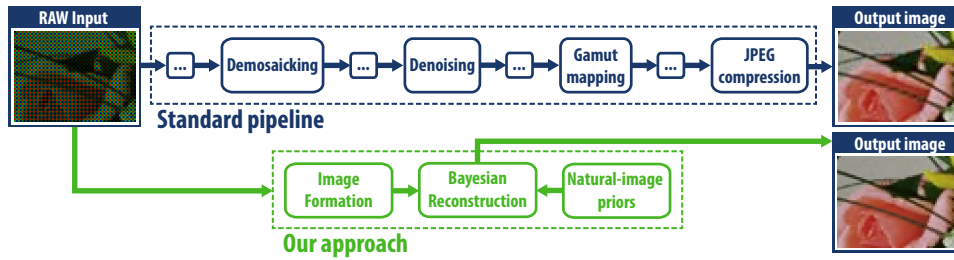
**Figure 9.1:** Our end-to-end method jointly accounts for demosaicking, denoising, deconvolution, and missing data reconstruction. Following a Bayesian approach, image formation and priors are separated, which allows us to support both conventional and unconventional sensor designs. Examples (our results at lower right): **(a)** Demosaicking+denoising of Bayer data (top: regular pipeline). **(b)** Demosaicking+denoising of a burst stack (top: first frame). **(c)** Demosaicking+denoising+HDR from interlaced exposure (top: normalized exposure). **(d)** Denoising+reconstruction of a color camera array (top: naïve reconstruction).

compared to state-of-the-art techniques, even on very classical problems such as demosaicking. This demonstrates that treating color image processing as an inverse problem in a Bayesian framework, is, in fact, a change of paradigm, enabling advances for well understood problems as well as completely new camera designs.

## 9.1 Introduction

Modern camera systems rely heavily on computation to produce high-quality digital images. See Section 2.4.4 for a review of color image processing, including non-traditional sensor designs. Even relatively simple camera designs reconstruct a photograph using a complicated process consisting of tasks such as dead-pixel elimination, noise removal, spatial upsampling of subsampled color information (e.g., demosaicking of Bayer color filter arrays), sharpening, and image compression. More specialized camera architectures may require additional processing, such as multi-exposure fusion for high-dynamic-range imaging or parallax compensation in camera arrays.

The complexity of this process is traditionally tackled by splitting the image processing into several independent pipeline stages [63]. Splitting image reconstruction into smaller, seemingly independent tasks has the potential benefit of making



**Figure 9.2:** The traditional camera processing pipeline consists of many cascaded modules, introducing cumulative errors. We propose to replace this pipeline with a unified and flexible camera processing system, leveraging natural-image priors in a Bayesian approach.

the whole process more manageable, but this approach also has severe shortcomings. First, most of the individual stages are mathematically ill-posed and rely heavily on heuristics and prior information to produce good results. The following stages then treat the results of these heuristics as ground truth input, aggregating the mistakes through the pipeline. Secondly, the individual stages of the pipeline are in fact not truly independent, and there often exists no natural order in which the stages should be processed. For example, if noise removal follows demosaicking in the pipeline, the demosaicking step must be able to deal with noisy input data when performing edge detection and other such tasks required for upsampling the color channels, and denoising is complicated as the noise statistics change due to the interpolation in demosaicking.

We present a framework that replaces the traditional pipeline with a single, integrated inverse problem; see Figure 9.2. We solve this inverse problem using our Bayesian approach from Section 2.8, relying again on efficient proximal algorithms to solve the resulting optimization problem. We separate image formation from natural image priors in our Bayesian framework, preserving the modularity of the image formation process. Instead of applying different heuristics in each stage of the traditional pipeline, our system provides a single point to inject image priors in a principled and theoretically well-founded fashion.

Despite the integration of the individual tasks into a single inverse problem, our system is flexible and we can easily extend it to include new image formation models and camera types, by simply providing a procedural implementation of the

forward image formation model. Recalling Sections 2.3.3, 2.4.4 and 2.4.3 this image formation is typically composed of a sequence of independent linear transformations (e.g., lens blur followed by spatial sampling of the color information, followed by additive noise). Hence, we generalize it to a form that many imaging systems can be represented in. To illustrate the flexibility of our system, we apply it to a number of specific image formation models, see Figure 9.1, including joint Bayer demosaicking and denoising, deconvolution of camera shake and out-of-focus blur, interlaced HDR reconstruction and motion blur reduction, image fusion from color camera arrays, joint image stack denoising and demosaicking, and optimization all the way to the output representation.

All these image formation models are combined with a set of state-of-the-art image priors, following the Bayesian approach from Section 2.8. The priors can be implemented independent of each other. This enables sharing and re-using high-performance implementations of both the image formation process stages and the image priors for different applications. We analyzed a variety of image priors and determined a single set that we use for all applications, and which consistently outperforms the best specialized algorithms for well-researched problems such as demosaicking, denoising, and deconvolution, sometimes by a significant margin. Since the optimization of the forward model and the image priors is highly parallelizable, we also provide a very efficient GPU implementation. This allows us to process high-resolution images even on mobile devices (such as modern tablets with integrated mobile GPUs). We call this flexible image signal processing system *FlexISP*.

In the following, we first introduce our flexible end-to-end camera image processing system. In particular, we formulate the generalized image formation and corresponding inverse problem for color imaging systems. Next, we describe how the inverse problem is solved in our Bayesian framework from the related work, using an efficient proximal algorithm for the resulting optimization problem. Having explained our method, we demonstrate different applications, sensor types, and priors, with minimal code changes; an in-depth analysis of the design choices made to create our system—in particular of the priors used in our approach; and an evaluation of our framework against many state-of-the-art methods, demonstrating the high image quality we achieve.

## 9.2 Image Formation Model

We adopt the linear image formation model from Section 2.8.3. A large class of imaging systems can be represented by the factorization  $\Phi = \mathbf{SB}$ . That means the observed image  $\mathbf{j}$  is modeled as

$$\mathbf{j} = \Phi \mathbf{i} + \mathbf{n} = \mathbf{SBW} \mathbf{i} + \mathbf{n}, \quad (9.1)$$

Depending on the camera lens, sensor, and scene motion, the latent image undergoes various transformations, as discussed in detail in the related work Sections 2.3.3, 2.4.4 and 2.4.3. The operator  $\mathbf{W}$  represents the registration component of the image transform as a warping matrix, resulting for example from physical camera or scene motion. The transform  $\mathbf{B}$  models the convolutional part of this transform, which models aberrations as in Section 2.3.3 or blurring due to the camera’s anti-aliasing filter. The last transform  $\mathbf{S}$  models sub-sampling on the sensor, which may occur due to the color filter array on the sensor for color imaging as discussed in Section 2.4.4. We assume the sensor noise  $\mathbf{n}$  to be Gaussian distributed in our setting. Note that the specific definition of matrix  $\Phi$  changes based on the exact application shown later in this chapter.

## 9.3 Flexible Image Reconstruction

The image formation model from Eq. (9.1) is inverted following the Bayesian MAP approach from Section 2.8.3 again, resulting in a solver very similar to the one from the previous chapter Chapter 8. However, in contrast, we do not only rely here on (cross-channel) gradient statistics, but add a patch-based image prior, which makes our method generalize across a wide variety of image reconstruction tasks.

### 9.3.1 Inverse Problem

Following Section 2.8.3, we solve the MAP estimation problem

$$\mathbf{i}_{\text{opt}} = \underset{\mathbf{i}}{\operatorname{argmin}} \frac{1}{2} \|\Phi \mathbf{i} - \mathbf{j}\|_{\mathbf{M}}^2 + \Gamma(\mathbf{i}), \quad (9.2)$$

where  $\Gamma(\mathbf{i})$  is our specific prior, and the data fidelity term results from the Gaussian likelihood. The operator  $\mathbf{M}$  is here a diagonal weighting matrix that allows to mask out different observations, e.g. saturated ones. While the data term is a standard choice, we note that finding appropriate priors  $F$  is key for yielding high-quality image reconstructions. We use the following prior

$$\Gamma(\mathbf{i}) = \phi_0 \|\nabla \mathbf{i}\|_1 + \phi_1 \|\nabla \mathbf{i}_r \cdot \mathbf{i} - \nabla \mathbf{i} \cdot \mathbf{i}_r\|_1 + \phi_2 \Lambda(\mathbf{i}). \quad (9.3)$$

The first term is a standard TV prior [178, 188]. As in Chapter 5, this prior term is based on the assumption of a sparse Laplacian distribution on the spatial gradients. In contrast to Chapter 8, we do not penalize curvature here. The second term is our cross-channel prior from Eq. (8.4) in Section 8.3.2, with  $\mathbf{i}_r$  as the current considered cross-channel as a notational shortcut. The last term represents a prior on patch statistics in natural images, as often used in image denoising methods; see Section 2.4.4. We explain this in the following.

### 9.3.2 Denoising Prior

Recalling Eq. (2.33) from the related work Section 2.8.2, the term  $\Lambda(\mathbf{i})$  in the MAP estimate (9.2) can be interpreted as the result of the exponential image prior

$$p(\mathbf{i}) \propto \exp(-\Lambda(\mathbf{i})), \quad (9.4)$$

where intuitively  $\Lambda$  assigns a cost to unknowns that are less likely, independent of any observations being made. However, following our approach from the related work Section 2.8.4, note that a proximal algorithm to solve Eq. (9.2) does not need to evaluate  $\Lambda$ , but just the proximal operator for  $\Lambda$ . This means, even if  $\Lambda$  is unknown, we solve for the Eq. (9.2) as long as its proximal operator is known.

In particular, we can show that *any* Gaussian denoiser can be expressed as a proximal operator. This allows us to use the implicit natural priors used in many complex denoising approaches, such as BM3D discussed in Section 2.4.4, without needing to evaluate Eq. (9.4). Consider now a Gaussian likelihood  $p(\mathbf{x}|\mathbf{v})$

$$p(\mathbf{x}|\mathbf{v}) \propto \exp\left(-\frac{\|\mathbf{x} - \mathbf{v}\|_2^2}{2\sigma^2}\right).$$

With the exponential image prior from Eq. (9.4) we get the following MAP estimate, that is a proximal operator

$$\mathbf{prox}_{\sigma^2\Lambda}(\mathbf{v}) = \underset{\mathbf{x}}{\operatorname{argmin}} \left( \Lambda(\mathbf{x}) + \frac{1}{2\sigma^2} \|\mathbf{x} - \mathbf{v}\|_2^2 \right) \quad (9.5)$$

Consider a family of Gaussian denoising algorithms  $\{D_\sigma^\Lambda : \sigma > 0\}$  using the prior from above, which estimate  $\mathbf{x}_0$  from  $\mathbf{x}_0 + \sigma\mathbf{z}$  with  $\mathbf{z} \sim \mathcal{N}(0, \mathbb{I})$ . We can then formulate

$$\mathbf{prox}_{\tau\Lambda}(\mathbf{v}) = D_{\sqrt{\tau}}^\Lambda(\mathbf{v}), \quad (9.6)$$

It becomes clear now, that it is not necessary to derive  $\Lambda$  in order to evaluate the proximal operator, only  $D_\sigma^\Lambda$  has to be known. Note that depending on  $\Lambda$  this operator can be non-convex. If  $\Lambda$  is the indicator function of a convex set,  $D_{\sqrt{\tau}}^\Lambda$  becomes the projection onto this set. For the method in this chapter we use the patch-based BM3D denoiser [70], which achieves high-quality results across the range of applications demonstrated in this chapter. We have experimented with various denoisers: Sliding DCT which uses collaborative filtering, NLM which uses self-similarity, and patch-based NLM and BM3D which use both. The choice depends on the application and on the desired reconstruction speed and quality, which we evaluate in detail in the appendix Section A.12.

### 9.3.3 Optimization

Having defined the proximal operator for  $\Lambda$  above, the optimization problem from Eq. (9.2) can be solved again following our approach from Section 2.8. We formulate the objective in the form of Section 2.8.3 by choosing

$$\begin{aligned} \Phi &= ?, & G(\mathbf{v}) &= \|\mathbf{v} - \mathbf{j}\|_{\mathbf{M}}^2, \\ \Omega &= [\mathbf{S}^T, \mathbb{I}^T]^T, & F(\mathbf{v}) &= \|\mathbf{v}_1\|_1 + \phi_2 \Lambda(\mathbf{v}_2) \end{aligned} \quad (9.7)$$

$\Phi$  is set based on the application. The indices in the second line identify the component stacked on-top of each other in  $\Omega\mathbf{i}$ . Note that the minimum of  $\Gamma(\mathbf{i})$  is obtained by independently minimizing  $F$  for each component of  $\Omega\mathbf{i}$ .

As in the previous Chapter 8, we use the Chambolle-Pock proximal algorithm

to solve our problem. The proximal operators for  $F$  and  $G$  are

$$\mathbf{prox}_{\tau F}(\mathbf{v}) = \left[ \mathbf{prox}_{\tau|\cdot|}(\mathbf{v}_1^T), \mathbf{prox}_{\frac{\tau}{\phi_2}\Lambda}(\mathbf{v}_2^T) \right]^T \quad (9.8)$$

$$\begin{aligned} \mathbf{prox}_{\tau G}(\mathbf{v}) &= \tilde{\mathbf{x}} = \underset{\mathbf{x}}{\operatorname{argmin}} \frac{1}{2} \|\Phi \mathbf{x} - \mathbf{j}\|_{\mathbf{M}}^2 + \frac{1}{2\tau} \|\mathbf{x} - \mathbf{v}\|_2^2 \\ &\Leftrightarrow \frac{1}{\tau}(\tilde{\mathbf{x}} - \mathbf{v}) + 2(\Phi^T \mathbf{M}^2 \Phi \tilde{\mathbf{x}} - \Phi^T \mathbf{M} \mathbf{j}) = 0 \\ &\Leftrightarrow (\tau \Phi^T \mathbf{M}^2 \Phi + \mathbb{I}) \tilde{\mathbf{x}} = (\tau \Phi^T \mathbf{M} \mathbf{j} + \mathbf{v}). \end{aligned} \quad (9.9)$$

The proximal operator for  $F$  is separable, decoupling into the shrinkage operator from Eq. (5.18) and the denoising proximal operator. Note that we have defined the proximal operator for  $F$  here, while the Chambolle-Pock method requires the evaluation of the proximal operator of the convex conjugate  $F^*$  as also defined in the previous chapter. However, using the Moreau decomposition [175, 270], this operator can be expressed in term of the proximal operator for  $F$

$$\mathbf{prox}_{\tau F^*}(\mathbf{v}) = \mathbf{v} - \tau \cdot \mathbf{prox}_{\frac{1}{\tau} F} \left( \frac{\mathbf{v}}{\tau} \right). \quad (9.10)$$

The proximal operator for  $G$  from Eq. (9.9) is a linear least-squares problem. In contrast to Chapter 8, this subproblem does not only contain convolutional operators and hence cannot be solved efficiently in the Frequency domain anymore. We use the CG algorithm. The advantage of CG in our setting is that the image formation model  $\Phi$  and its transpose can be represented algorithmically, without explicitly generating the system matrix. However, on the downside, CG does not further exploit the specific structure of  $\Phi$  from Eq. (9.1). We present a drastically more efficient approach that exploits this structure in the next Chapter 10.

*Parameters* Using the notation from [178], we use  $\theta = 1, \sigma = 40$  and  $\tau = 0.9/(\sigma L^2)$  for our implementation of the Chambolle-Pock method using Eqs. (9.8) and (9.9). Following Section 8.3.3, we estimate  $L = \|\Omega\|$  using the power iteration where all matrix-vector-multiplications are implemented as filtering operations.

## 9.4 Applications

Next we describe our applications and their individual data fidelity terms, that is the specific choice of  $G$  from Eq. (9.7). In addition to the four applications shown in the following paragraphs, Section A.13 discusses two further camera designs for which our our method enables high-quality imaging.

*Demosaicking* Our first application performs demosaicking of a Bayer raw image and simultaneous denoising, if required. The data model is quite simple: we set  $\Phi = \mathbf{S}$  as the Bayer array decimation matrix which discretizes Eq. (2.23) from Section 2.4.4 (and set the other operators  $\mathbf{B} = \mathbf{W} = \mathbb{I}$ ). The data fidelity term then becomes

$$G(\Phi\mathbf{i}) = \|\mathbf{S}\mathbf{i} - \mathbf{j}\|_2^2. \quad (9.11)$$

*Deblurring* We next demonstrate non-blind deconvolution, i.e., assuming a known (calibrated) PSF. As in Section 2.3.3, we model blur with a convolution matrix  $\Phi = \mathbf{B}$  and get

$$G(\Phi\mathbf{i}) = \|\mathbf{B}\mathbf{i} - \mathbf{j}\|_2^2. \quad (9.12)$$

As demonstrated by Schuler et al. [165], one can also incorporate Bayer array sensing into deconvolution by setting  $\Phi = \mathbf{S}\mathbf{B}$ .

*Interlaced HDR* Our third application focuses on HDR imaging. Some recent sensors (e.g., Aptina AR1331CP and Sony IMX135) can record two exposures in a single frame by allowing the exposure time to be set independently for even and odd macro-rows (a pair of consecutive rows that cover all the color samples) of an otherwise normal Bayer sensor. Figure 9.1(c) depicts the exposure pattern; odd macro-rows integrate light for a longer time than the even ones. The advantage is that two exposures, together covering a wider dynamic range than a single shot, are captured essentially at the same time, while traditional stack-based HDR allows time to pass and objects to move between the shots. However, samples are more sparse: odd macro-rows may saturate in bright regions and even macro-rows may be too noisy to be useful in dark regions. We easily adapt our framework to reconstruct



high-quality HDR images from such sensors.

We use a binary diagonal indicator as  $\mathbf{M}$  for masking bad pixels, where 0 indicates a useless pixel (either saturated or too noisy), and 1 means that the pixel has useful data. The blur matrix  $\mathbf{B}$  models the optics and sensor anti-aliasing filter by measuring the camera’s PSF; all applications model blur in this fashion, unless otherwise noted. We measure the blur like Xu and Jia [271]. Other calibration-target-based methods, such as the one mentioned in the previous Chapter 8, are also possible choices.

Finally, we set  $\Phi = \mathbf{SB}$  for the blurred, decimated, and saturated observation matrix, i.e.,

$$G(\Phi\mathbf{i}) = \|\mathbf{SB}\mathbf{i} - \mathbf{j}\|_{\mathbf{M}}^2. \quad (9.13)$$

*Burst Denoising and Demosaicking* Many compact cameras produce very noisy results in low-light situations due to their small sensor size. Instead of a single photograph, one can take a rapid burst of multiple exposures and then combine them to reduce noise. This idea has been discussed at the end of the related work Section 2.4.3. However, the previous techniques mentioned in this section demosaic each image first and then denoise the stack of images. This is not ideal since demosaicking will modify and possibly even amplify noise. Furthermore, additional frames can aid the demosaicking process [272, 273]. We demonstrate that our framework can jointly demosaic and denoise a stack of burst images, yielding results that are superior to methods that pipeline these two processes.

The image stack is likely to be slightly misaligned due to small camera and object motion. We handle this by aligning all  $k$  observations  $\mathbf{j}_i$  to the reference frame (e.g., the first one) by computing a brute-force  $1/8^{\text{th}}$  sub-pixel-accurate nearest-neighbor search ( $\ell_1$ -norm on  $15 \times 15$  patches). We represent it as a warp matrix  $\mathbf{W}_i$ . Note that we also bake the resampling filter into this matrix. While our captures did not contain any motion blur, one could easily incorporate and calibrate the blur as described for interlaced HDR above. Finally, we set  $\Phi_i = \mathbf{S}\mathbf{W}_i$  to get

$$G(\Phi\mathbf{i}) = \sum_{i=1}^k \|\mathbf{S}\mathbf{W}_i\mathbf{i} - \mathbf{j}_i\|_{\mathbf{M}_i}^2. \quad (9.14)$$

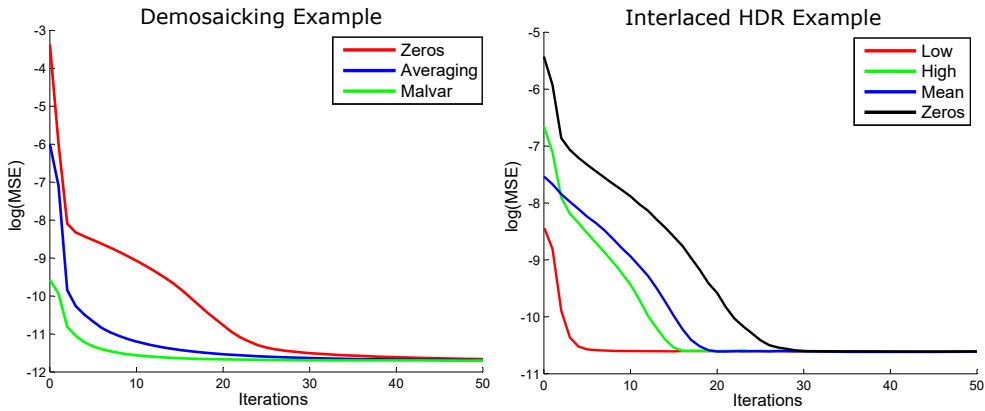
The masking matrix  $\mathbf{M}_i$  is included here since it may be used to mask out pixels for which no good alignment is found. However, in our tests this was not required.

*Applying Our Framework to New Applications* We have described a large number of different applications. In addition, Section A.13 demonstrates that our approach achieves high-quality results for color array imaging and even non-linear image formation models for JPEG deblocking. The wide variety of applications is possible due to the flexibility of our system; adapting it to a new image formation model is very easy and only requires changing the matrices  $\Phi$  and  $\mathbf{M}$ .

## 9.5 Results

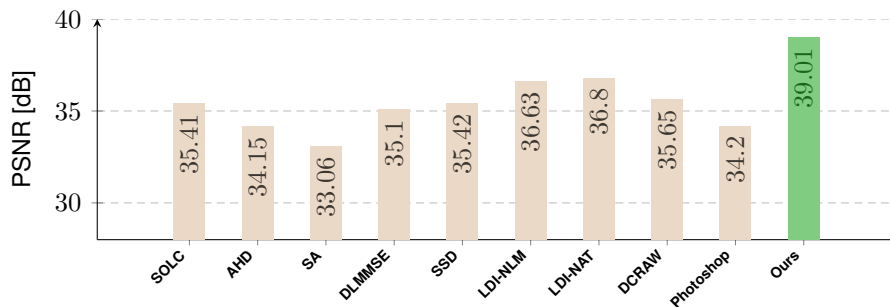
In the following, we first show empirical convergence results and next present results for each of the applications described previously. The color array camera and non-linear deblocking application are covered in the appendix Section A.13. Finally, we discuss the performance of our approach.

*Convergence* We make use of non-convex regularization terms in the prior term  $\Lambda$ , so theoretical convergence guarantees are not available. That said, we always combine both convex and non-convex terms in  $\Gamma$  from Eq. (9.3) which locally convexify the objective as discussed in the appendix Section A.12. This enables us to achieve convergence in practice for all initial iterates we tried (even for  $\mathbf{x}^0 = \mathbf{0}$ ). We present convergence plots for two different examples in Figure 9.3. The first example is demosaicking of a Bayer image, where we initialize our optimization with:  $\mathbf{x}^0 = 0$ ,  $\mathbf{x}^0 =$  average of neighboring pixels,  $\mathbf{x}^0 =$  Malvar [274] demosaicking. The second example is reconstructing an image from the interlaced HDR sensor. We use three different initializations:  $\mathbf{x}^0 = 0$ ,  $\mathbf{x}^0 =$  long exposure,  $\mathbf{x}^0 =$  short exposure,  $\mathbf{x}^0 =$  mean of long and short exposure. Notice how in all cases, we converge to the same low final error, no matter how we initialized the optimization. This indicates, that in practice our optimization converges.

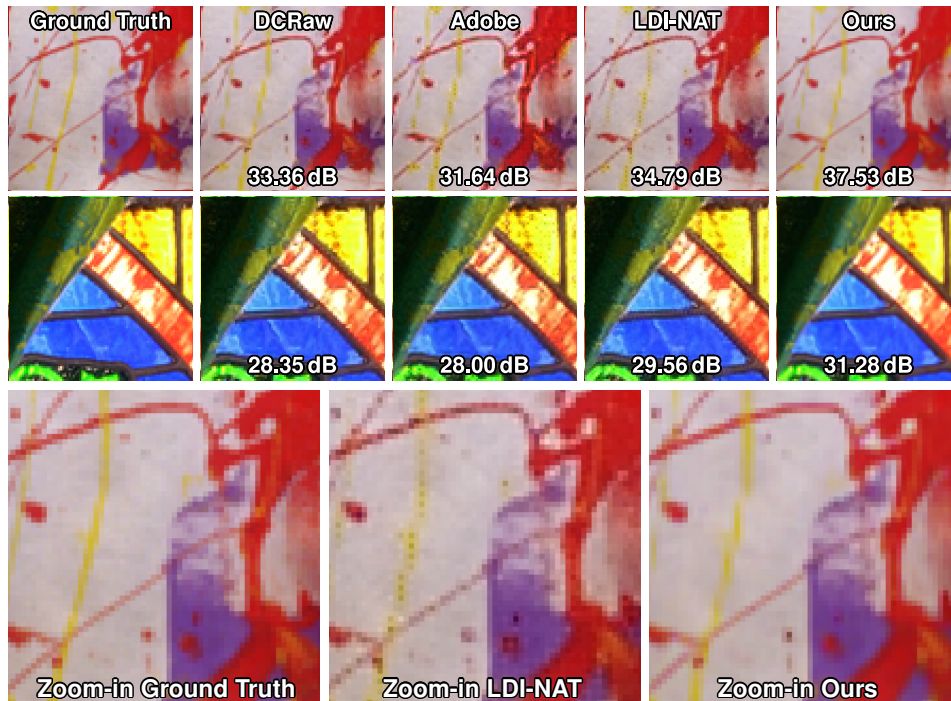


**Figure 9.3:** Effects of different initial iterate  $x^0$  on the convergence for a simple demosaicking example (left) and an interlaced HDR example (right). We compare here the convergence plots for different initial iterates (which are described in the text in detail). All cases converge to the same solution; with a good starting point, 5 iterations suffice.

*Demosaicking of Conventional Bayer Images* First, we demonstrate that our system outperforms state-of-the-art demosaicking methods. Zhang et al. [64] compared several algorithms on the McMaster color image dataset. We follow their procedure and additionally run our method, DCRaw, and Adobe Photoshop on the same dataset. Figure 9.4 lists the results of the comparison. Representative examples are shown in Figure 9.5. Our framework shows a gain of 2.23 dB over the best existing demosaicking methods. Note that this is a very significant improvement, given



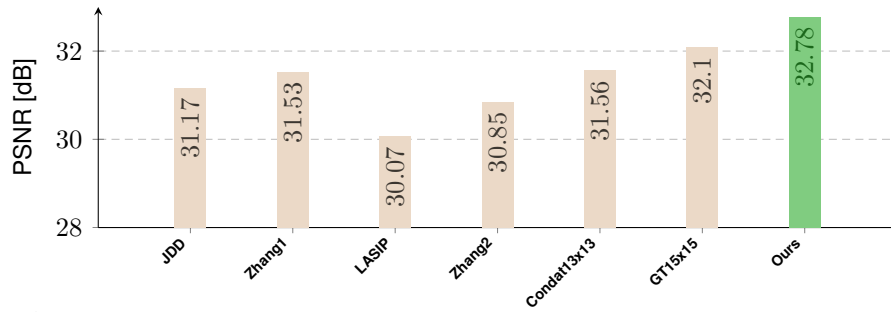
**Figure 9.4:** Demosaicking results for the McMaster color image dataset. We show average PSNR over all images and color channels.



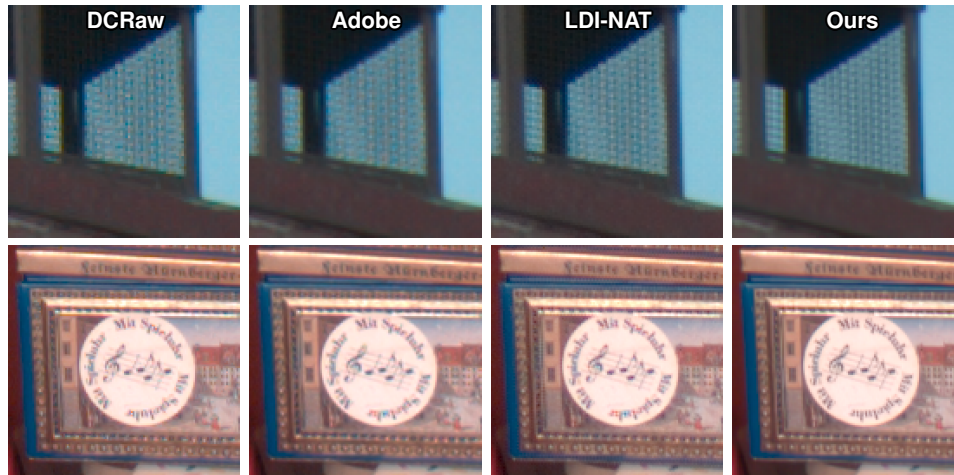
**Figure 9.5:** We compare our demosaicking method with commonly state-of-the-art demosaicking methods on two patches of the McMaster dataset (selected regions shown enlarged in the last row). Note how our method produces results virtually indistinguishable from ground truth.

that demosaicking is a mature problem with hundreds of publications over the past decade. In Figure 9.6, we compare joint demosaicking and denoising against other state-of-the-art methods, following Jeon and Dubois [275]. Our method outperforms all competing methods by at least 0.68 dB. A real-world example can be found in Figure 9.7.

*Deblurring of Out-of-focus and Camera-shake Blur* Before diving into the more complex use cases with the three new camera technologies below, we show that our approach outperforms the state-of-the-art in non-blind deconvolution. Two examples are shown in Figure 9.8. Deconvolution methods differ mostly by the image priors used. We adopt the comparison of Schuler et al. [276] with five state-of-the-art priors: EPLL (Gaussian mixture models) [163], heavy-tailed gradient



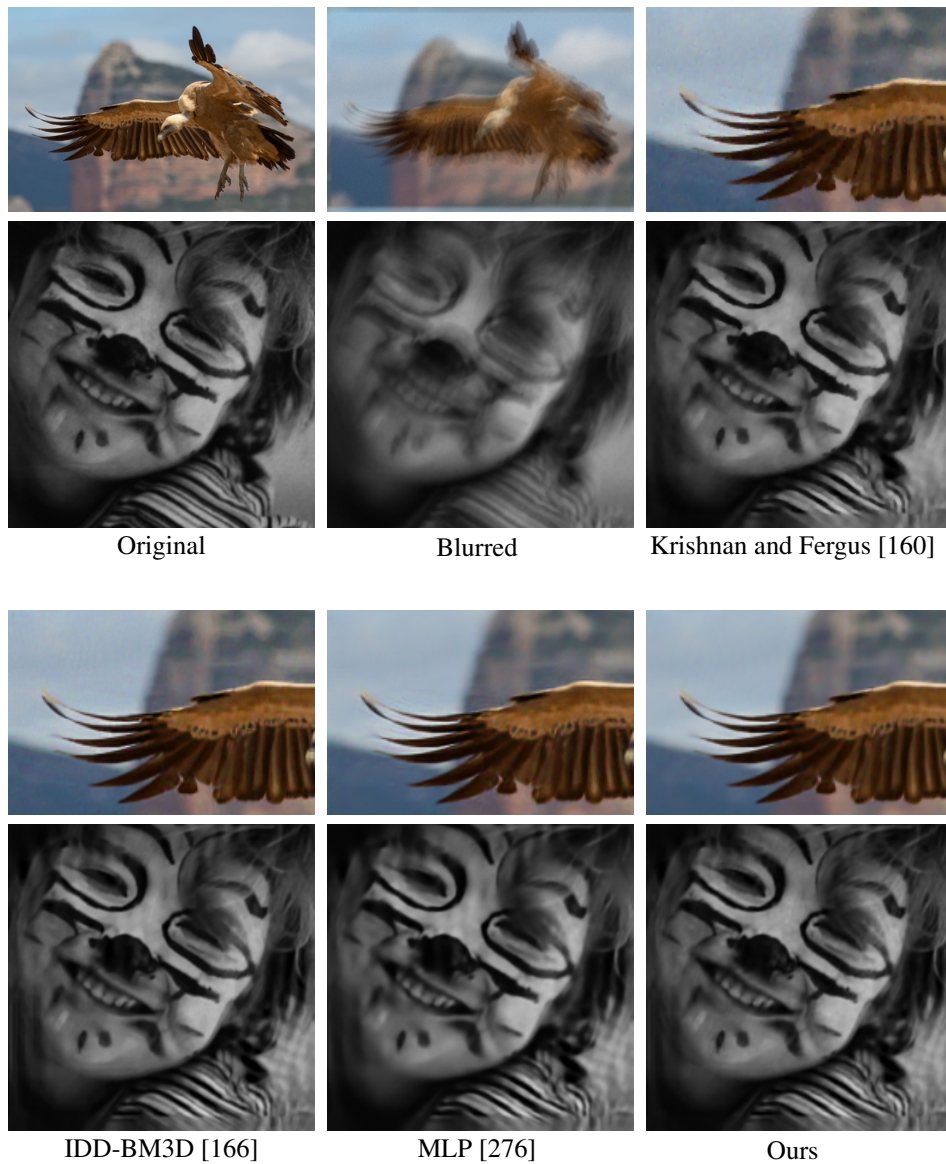
**Figure 9.6:** Joint demosaicking and denoising results for the Kodak dataset. We show average PSNR over all images, color channels, and sigmas.



**Figure 9.7:** Demosaicking comparison as in Figure 9.5, but on two real-world example captures. Note again the artifacts which are visible for the other methods.

distributions [160, 161], self-similarity [166, 277], Field-of-Experts [278], and neural networks [276].

Table 9.1 shows a quantitative comparison for five different blur and noise scenarios (a)–(e) averaged over a test-set of 11 images (see Schuler et al. [276], Table 1). We outperform MLP (and others more clearly), with the highest margin in the motion blur case (e), which probably is the most common application of deconvolution. Our mixed BM3D + TV (+ cross-channel) prior outperforms both BM3D-based and gradient-based methods. Both internal and external priors are



**Figure 9.8:** The color image example shows simulation results for a synthetically blurred image (blur and noise scenario (e) from [276]). As can be seen, our method produces the least artifacts (e.g., see smooth backgrounds). The single channel example shows results for a real-world capture from [161]. Similarly, our method produces the least artifacts.

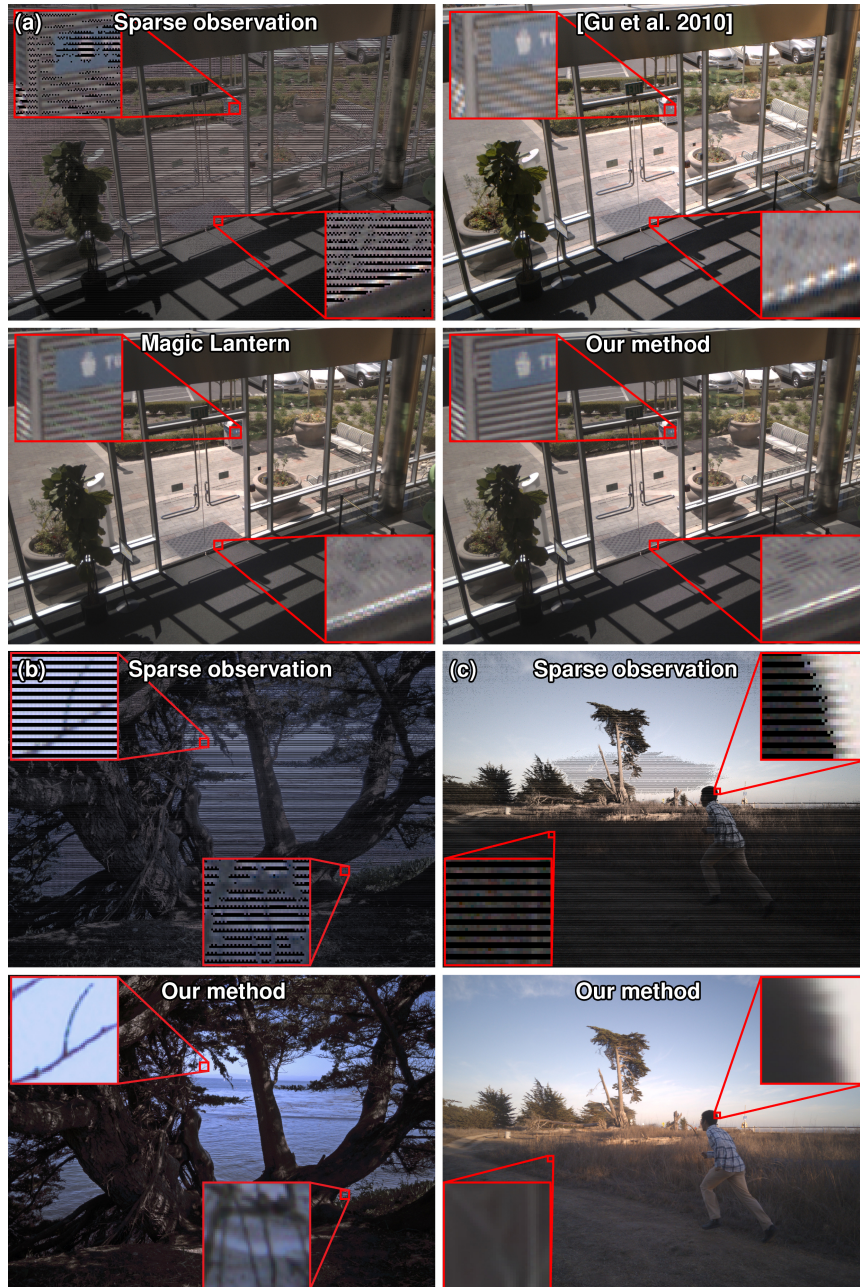
Deconvolution method	Reconstruction PSNR [ $dB$ ]				
	(a)	(b)	(c)	(d)	(e)
EPLL [163]	24.04	26.64	21.36	21.04	29.25
Levin et al. [161]	24.09	26.51	21.72	21.91	28.33
Krishnan and Fergus [160]	24.17	26.60	21.73	22.07	28.17
DEB-BM3D [277]	24.19	26.30	21.48	22.20	28.26
IDD-BM3D [166]	24.68	27.13	21.99	22.69	29.41
FoE [278]	24.07	26.56	21.61	22.04	28.83
MLP [276]	24.76	27.23	22.20	22.75	29.42
Ours	<b>24.83</b>	<b>27.31</b>	<b>22.24</b>	<b>22.89</b>	<b>29.78</b>

**Table 9.1:** Reconstruction quality for five non-blind deconvolution scenarios. We have adopted the evaluation from Schuler et al. [276]. Average results for the same representative test-set of 11 images are shown here.

needed for peak performance in image reconstruction, only one is usually not enough [71]. The images in this dataset were grayscale, so the cross-channel prior could not be used. For visual comparisons including color images, see again the two examples shown in Figure 9.8.

*HDR Image Reconstruction* The interlaced HDR sensor poses a challenging reconstruction problem, as shown in Figure 9.9. For scenes with a wide dynamic range, we use an exposure time ratio as high as 16:1, so that odd macro-rows are exposed for 16 times longer than the even macro-rows. Now the short exposure has a better chance to catch highlights and the long exposure can capture shadows; unfortunately, short exposure may be too noisy in shadows and long exposure may saturate in bright areas. While the reduced number of valid input observations increases the difficulty of the reconstruction task, our approach addresses this case efficiently. The self-similarity prior of BM3D or NLM is now crucial to meaningfully filling in the missing data. We demonstrate this in Figure 9.9 (a), where we compare our method to the standard image processing pipeline of demosaicking (DCRaw) plus HDR de-interlacing of Gu et al. [77], which is based on upsampling. We also compare against the *Magic Lantern* firmware, which provides an efficient method by Hajisharif et al. [78] for fusing interlaced HDR images (some Canon





**Figure 9.9:** Image reconstruction for interlaced HDR sensor data. The upsampling-based reconstruction [77] in (a) improves the dynamic range, but fails to recover highly structured details. The Magic Lantern method provides higher quality, but still loses resolution and produces artifacts. Our method generates an artifact-free result and provides the best overall reconstruction quality. (b, c) In addition to extending the dynamic range (bottom row), we account for local scene motion and simultaneously denoise and reduce interlacing artifacts (see insets).



EOS sensors can capture interlaced dual-ISO images)—followed by demosaicking with DCRaw, since Magic Lantern only outputs mosaicked data. Note how Magic Lantern produces some artifacts, whereas our reconstruction is virtually artifact-free.

In Figure 9.9 (b, c top) we show the sparse input observations, with pixels marked black when indicated by the matrix  $\mathbf{M}$  as too noisy or saturated (simple thresholding on intensities). Even though a significant portion of the input data is missing, our unified optimization successfully reconstructs geometric structures, including detailed textures (grass) and thin edges. Also, in Figure 9.9 (c) we demonstrate its performance on scenes with local motion (a runner), where different amounts of motion blur in short- and long-exposure rows create a challenging interpolation problem. By default, our optimizer prefers the longer blur of the longer exposure, avoiding noise from the short exposure, and producing a visually consistent result.

*Burst Denoising and Demosaicking* We demonstrate joint denoising and demosaicking on a real-world and a simulated dataset in Figure 9.10. We used here 8 images taken at ISO 12800 on a Canon EOS 650D; the ground-truth was taken at ISO 100. To account for multiplicative and Poisson noise in this application, we perform the generalized Anscombe transform to stabilize variance [279]. It is applied to the observation  $\mathbf{j}$ , transforming the observations to fulfill our additive Gaussian model assumption in the data term.

We compare our technique (column (f)) to a number of standard processing pipelines for burst denoising. Demosaicking the first frame (a) [64] is the simplest approach, but also produces the worst results. Demosaicking followed by the state-of-the-art BM3D denoising method (b) significantly improves the image quality but is still inferior to our approach, which is not surprising, as the full stack contains a lot more information than a single image. We then compare against VBM3D [280] on the (aligned) stack of 8/16 demosaicked images, which yields good but still slightly inferior results (c). Just blending the aligned, demosaicked images together using exponential weights (as used in NLM [67]) computed from a  $15 \times 15$  patch around each pixel produces surprisingly good, albeit still slightly noisy images (d). Applying BM3D to the NLM-weighted, demosaicked stack achieves good PSNR numbers (e), but is still inferior to our proposed framework (column (f)) which



**Figure 9.10: Burst Image Comparison.** We show results for real-world captures taken in low light (paintcans), and a simulated example (doll). Please see the text for a detailed description.

jointly optimizes for the best latent image given the noisy stack of images.

*Performance* We have implemented FlexISP system in both Matlab (CPU) and CUDA (on desktop and tablet GPUs). We give the run times for several different applications and denoising priors in Table 9.2. Unsurprisingly, the CPU-implementation is slow, with a single iteration of BM3D (using the implementation from authors’ web-page) taking 149.4 seconds for a 13 MPix image. Apart from a BM3D step, each iteration of our solver runs a conjugate gradient step and enforces the other priors. Full numerical convergence is usually reached in about 30 iterations as shown in Figure 9.3 (note that the vertical scale is logarithmic). The overall computation cost seems high, however, in practice running 4–5 iterations from a good starting point gets close enough to the converged result and produces images with sufficient quality.

	Demosaic	HDR	Color Array	Burst
Resolution (MPix)	5	13	0.6×3	0.4×16
Iterations	4	5	5	5
CPU BM3D	312.4 s	1471.6 s	74.2 s	303.2 s
CPU NLM	196.5 s	1094.8 s	52.4 s	291.4 s
CPU Patchwise NLM	198.7 s	1102.2 s	52.8 s	291.8 s
CPU Sliding DCT	89.6 s	747.7 s	32.3 s	280.9 s
CPU Averaging	194.2 s	1087.5 s	52.0 s	291.4 s
GTX Titan BM3D	2.1 s	7.5 s	0.9 s	0.8 s
GTX Titan NLM	0.7 s	2.8 s	0.2 s	0.3 s
GTX Titan Patchwise NLM	0.8 s	2.9 s	0.2 s	0.3 s
GTX Titan Sliding DCT	0.3 s	1.2 s	0.1 s	0.3 s
GTX Titan Averaging	0.5 s	1.9 s	0.1 s	0.3 s
Tegra K1 BM3D	40.5 s	147.4 s	18.1 s	16.7 s
Tegra K1 NLM	26.5 s	56.1 s	3.8 s	7.3 s
Tegra K1 Patchwise NLM	28.0 s	59.2 s	4.0 s	7.7 s
Tegra K1 Sliding DCT	7.0 s	30.9 s	2.8 s	7.0 s
Tegra K1 Averaging	6.7 s	33.2 s	2.2 s	6.6 s

**Table 9.2:** Running times for: desktop CPU (Core i7 2.4 Ghz), desktop GPU (NVIDIA GTX TITAN), and mobile GPU (NVIDIA Tegra K1). Results have been averaged over 10 repeated runs.

Our GPU-version is optimized for speed and memory usage. The self-similarity based denoising priors (BM3D, NLM, etc.) are accelerated with an approximate-nearest-neighbor (ANN) method [281]. Furthermore, we split large images into tiles and process these tiles separately to save memory. We found that a small overlap of only four pixels is usually sufficient to prevent visible boundaries at tile borders. Now all intermediate buffers, even combined, consume less memory than the output image, enabling a memory-friendly image reconstruction. As a result, we can process high-resolution images in a matter of seconds on a recent desktop GPU (NVIDIA GTX TITAN) and can even enable interactive previews (1.5 sec) on a recent NVIDIA Tegra K1 tablet for 1 MPix images. As expected, the use of ANN leads to a slightly reduced quality compared to the Matlab implementation (of about -0.6 dB to -1.0 dB depending on the application). We have also experimented with a GPU-based accurate nearest neighbor search. While this increases the GPU run-time, e.g., when using BM3D as a prior it increases by a factor of 4, it achieves full accuracy and is still magnitudes faster than the CPU implementation.

## 9.6 Discussion

We have presented FlexISP, a framework and a system that replaces the traditional image processing pipeline for reconstructing photographs from raw sensor data by a single, integrated, and flexible system that is based on a Bayesian approach. By defining an image formation model and statistical image priors, this approach represents a change in paradigm compared to classical pipeline processing, which consider reconstruction sub-problems isolated, but not jointly. Our method generalizes due to a strong prior for natural images exploiting self-similarity and gradient statistics. In particular we show that the implicit image prior of any Gaussian denoiser can be used in our Bayesian approach, which is a key result and enables using self-similarity-based denoising priors. We demonstrate that the proposed method allows for image reconstruction for a wide variety of novel camera designs, and it outperforms the state-of-the-art for classical camera image processing tasks.

However, while our priors are expressive, there are situations where they do not provide enough information. For instance, in the interlaced HDR application, if two consecutive rows are missing a nearly horizontal thin line, it may be difficult for the

priors to connect the segments. Future priors that take content-aware information into account could help in such situation.

Our system is fast considering the wide range of applications and problem sizes, however, on a mobile device it cannot match the processing times of an embedded ISP with a specialized ASIC. In a practical camera, a traditional ISP could preprocess the image for immediate viewing, and a higher-quality image could be processed in background, or by a cloud service. Using the ISP-processed image as initialization would also give a better starting point, requiring fewer iterations for the optimizer to converge.

While we demonstrate significant improvements in quality and simplicity for traditional camera designs, we believe that our approach will achieve its full potential with future computational cameras that have significantly more complex image formation models.

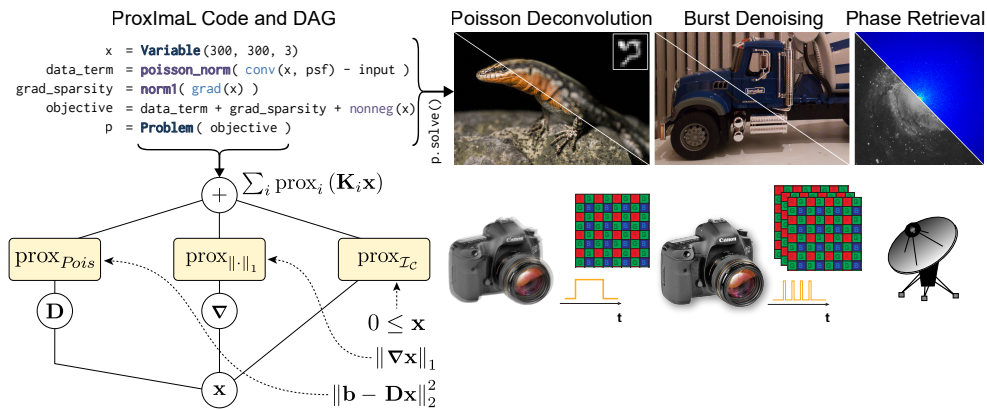
## Chapter 10

# Proximal Image Optimization

All the previous chapters in this dissertation have expressed imaging tasks as optimization problems using Bayesian inference. For each of the presented tasks, we have formulated an image formation model and image priors which model the observations, given the latent image, and prior knowledge of that image, without observations. Together, this leads to an optimization problem to find the most likely latent image given the observations and priors. As demonstrated in this dissertation, this technique generalizes across a wide range of applications, is a principled approach, and often outperforms the state-of-the-art, even on well-researched imaging problems as shown in the previous chapter.

However, while having many benefits compared to hand-engineered pipeline-based approaches, developing an image formation model into an efficient solver is not a straight-forward task. Different combinations of image priors and optimization algorithms may lead to significantly different computational cost and quality for different problems, as also demonstrated in this thesis, and implementing and testing each combination is currently a time-consuming and error-prone process.

In this chapter, we present *ProxImaL*, a domain-specific language and compiler for image optimization problems that makes it easy to explore this vast space of choices in the problem formulation and algorithm design. The language uses proximal operators as the fundamental building blocks of a variety of linear and nonlinear image formation models and cost functions, advanced image priors, and noise models. Relying on a generalized mathematical representation for image



**Figure 10.1:** As a domain-specific language and compiler, ProxImaL makes it easy to prototype a range of inverse problems in imaging. For example, we show ProxImaL code for deconvolution in the presence of Poisson-distributed shot noise and the corresponding directed acyclic graph (DAG, left, directionality from bottom to top) as well as the results generated by the compiled optimization algorithm (center left). We demonstrate state-of-the-art results for the applications shown here and other imaging problems.

optimization problems, a mathematical compiler identifies structure in the objective. The compiler intelligently chooses the best way to translate a problem formulation and choice of optimization algorithm into an efficient solver implementation, exploiting the problem structure. For a large variety of applications, we show that a few lines of ProxImaL code can generate solvers with drastically increased accuracy, convergence, and computational efficiency compared to a naive black-box optimization approach. In many cases, we can achieve multiple orders of magnitude reduction in run-time.

## 10.1 Introduction

Computational Imaging is a research area with a wide variety of applications in photography, computer vision, robotics, scientific imaging, remote sensing, microscopy, and computer graphics. Traditionally, image processing algorithms have been tailored independently to each of these applications. For example, classical image processing pipelines, discussed in Chapter 9, have to be adapted even for simple

modifications of the camera, such as changing the color filter array. In this dissertation, we have demonstrated that solving imaging tasks using optimization in a Bayesian framework is a much more principled approach. Optimization-based imaging generalizes across a wide range application areas, where the problems considered in this dissertation only represent a small subset. Furthermore, it outperforms the state-of-the-art in many classical applications. Thus, in theory, optimization-based imaging makes it easy to develop image processing techniques for new domains, that outperform hand-engineered algorithms.

In practice, however, developing image optimization methods can be difficult, because there are many choices in the Bayesian approach from Section 2.8 and it is hard to predict which framing will yield the best results. Different types of latent images may be reconstructed, which each follow different prior distributions. For example, the transient image recovery from Chapter 3 uses different priors than the color image reconstruction from Chapter 9. However, even for the same latent unknowns, different prior choices lead to varying performance. For example, the chromatic aberration removal from Chapter 8 mostly relies on the cross-channel prior and hence can achieve fast results without an additional costly patch-based prior used in Chapter 9. A similar observation can be made for the likelihood function. While an accurate noise model can improve the reconstruction quality it may also lead to a significantly more costly optimization problem than with a simple noise model. This is why we approximated noise as Gaussian distributed in Chapters 3, 5, 4, 6, 8 and 9. Note, that a more accurate prior or likelihood does not necessarily lead to better reconstruction quality, since the resulting optimization problem may be hard to solve, e.g. an unstructured and large-scale non-convex problem. Even once prior and likelihood are defined, there is ambiguity in choosing the operators and penalty functions from Section 2.8.3, which all yield equivalent objectives, but significantly different solver implementations. In each of the Chapters 3, 5, 4, 6, 8 and 9 we have chosen one of many possible translations of the respective problem into the form of Section 2.8.3. Finally, a specific proximal algorithm has to be chosen for the solver. In Chapters 4 and 6 we have used the ADMM method, Chapter 5 uses linearized ADMM, and Chapters 3, 8 and 9 implement Chambolle and Pock's method [178].

To further illustrate the various choices along the way to a solver implementation,



consider the classic problem of deconvolution: we are given measurements  $\mathbf{j}$  that satisfy  $\mathbf{j} = \mathbf{B}\mathbf{x} + \mathbf{n}$ , where  $\mathbf{B}$  is a linear operator representing convolution with a known kernel and  $\mathbf{x}$  is an unknown image. Our goal is to recover  $\mathbf{x}$  given  $\mathbf{j}$ . Following the Bayesian approach from Section 2.8 eventually leads to the MAP estimate

$$\text{minimize } g(\mathbf{B}\mathbf{x} - \mathbf{j}) + \Gamma(\mathbf{x}), \quad (10.1)$$

where in general  $g$  is an error metric, and  $\Gamma$  is a penalty function that expresses prior knowledge about the image  $\mathbf{x}$ .

There are many reasonable choices for  $g$  and  $\Gamma$ . For instance, we might define  $g$  as a sum-of-squares error, a Huber loss, or a Poisson penalty, based on the noise  $\mathbf{n}$  and expected efficiency of the solver with one of these penalties. The penalty function  $\Gamma$  could be a constraint on the range of the values of  $\mathbf{x}$ , a sparsity-inducing gradient penalty such as total-variation, a non-local patch prior as in the BM3D-based reconstruction shown in [166], or a combination of all these penalties. Each of these choices have different reconstruction qualities and computational costs associated. Furthermore, the non-local patch-based priors lead to a non-convex optimization problem.

Once we have chosen  $g$  and  $\Gamma$ , we must choose an algorithm to use for solving the optimization problem, each of them leading to different solvers subproblems. Depending on  $g$  and  $\Gamma$  these subproblems can be computationally cheap to solve, e.g. in the Frequency domain for this deconvolution example. Moreover, for each algorithm there may be many ways to translate Problem (10.1) into that algorithm's standard form. The only way to know which algorithm and translation into standard form works best for a problem is to try all of them.

In summary, finding an effective image optimization method thus requires exploring a large space of problem formulations, algorithms, and translations between standard forms. Currently, researchers must develop a new solver implementation for each point they explore in the space, which is a time-consuming and error-prone process. Developing implementations is particularly challenging for image optimization problems since these problems typically involve millions of variables and can only be solved efficiently by exploiting problem structure.

In this chapter, we address these challenges by introducing ProxImaL, a Domain-

Specific Language (DSL) for image optimization. The ProxImaL language allows users to describe image optimization problems in a few lines of code using an intuitive syntax that follows the math. Users write their problem using a fixed set of mathematical functions, whose structure can be exploited to generate an efficient solver. Most functions that occur in image optimization problems are included in the language, and it is easy to add support for more. Compositions of functions are limited by a set of simple rules that ensure the problems constructed by the user match our standard mathematical representation.

DSLs for optimization have a long history, going back to GAMS [282] in the 1970s, and including DSLs specialized for convex optimization, such as CVX [283], YALMIP [284], CVXPY [285], and Convex.jl [286]. However, in contrast to ProxImaL, these approaches reliably solve modest size problems, with on the order of 10,000s of variables, but for image optimization problems with millions of variables these solvers become completely infeasible due to their memory and computational cost. ProxImaL overcomes the limitations of these approaches by exploiting the domain-specific problem structure as described above.

The ProxImaL compiler takes the user’s problem description and choice of algorithm and automatically generates a solver implementation. The compiler considers a wide range of possible solver implementations and selects one based on expert knowledge about how to best formulate problems for the chosen solver algorithm. The user can also easily override the compiler’s default choice to try out more implementations. The solver implementations generated by the compiler are highly efficient because we created optimized code for the core mathematical operations using Halide [287].

We demonstrate the utility of ProxImaL through applications to the image processing pipeline, burst photography and denoising, deconvolution, and phase retrieval. In many cases a few lines of ProxImaL code and the default solver implementation generated by the ProxImaL compiler achieves state-of-the-art results, often with a runtime under ten seconds.

In the following, we start by developing a simple language and mathematical representation for image optimization problems that captures the problem structure needed to generate an efficient solver. Next, we present a compiler that takes the user’s problem description and choice of solver algorithm and automatically

generates an efficient solver, intelligently choosing from the many translations possible. Finally, we show that our framework can achieve state-of-the-art results on a variety of image optimization problems while also producing highly efficient solver implementations.

## 10.2 Representing Image Optimization Problems

We represent image optimization problems with the generalized objective from Section 6.2.2. This objective generalizes well across different imaging problems and proximal algorithms, as will be demonstrated in the following. To make this chapter self-contained, we discuss the generalized objective here again. We define an image optimization problem as a sum of penalties  $f_i$  on linear transforms  $\mathbf{K}_i \mathbf{x}$  with  $\mathbf{x} \in \mathbb{R}^n$  being the unknown:

$$\operatorname{argmin}_{\mathbf{x}} \sum_{i=1}^I f_i(\mathbf{K}_i \mathbf{x}) \quad \text{with} \quad \mathbf{K} = \begin{bmatrix} \mathbf{K}_1 \\ \vdots \\ \mathbf{K}_I \end{bmatrix}, \quad (10.2)$$

where here  $\mathbf{K} \in \mathbb{R}^{m \times n}$  is one large matrix that is composed of stacked linear operators  $\mathbf{K}_1, \dots, \mathbf{K}_I$ . The linear operator  $\mathbf{K}_i \in \mathbb{R}^{m_i \times n}$  selects a subset of  $m_i$  rows of  $\mathbf{K} \mathbf{x}$ . This subset of rows is then the input to the penalties  $f_i : \mathbb{R}^{m_i} \rightarrow \mathbb{R}$ .

We consider the class of linear image MAP problems. As discussed in Section 2.8.3 such image optimization problems generally contain four components: variables representing the latent image(s) to be reconstructed, a forward model of image formation in terms of linear operators, a penalty based on the noise model, and priors applied on linear transforms of the image(s). For example, consider a slightly more complex version of the deconvolution problem (10.1) where the convolved image  $\mathbf{B} \mathbf{x}$  is subsampled by a known demosaicking pattern, which we represent with the linear operator  $\mathbf{M}$ . We formulate our problem using a sum-of-squares error metric,  $g(\mathbf{x}) = \|\mathbf{M} \mathbf{B} \mathbf{x} - \mathbf{j}\|_2^2$ , and the penalty function:

$$\Gamma(\mathbf{x}) = \mu \|\nabla \mathbf{x}\|_1 + (1 - \mu) \|\nabla \mathbf{x}\|_2^2 + I_{[0, \infty)}(\mathbf{x}),$$

where  $\mu \in [0, 1]$ ,  $\nabla$  is the gradient operator, and:

$$I_{[0,\infty)}(\mathbf{x}) = \begin{cases} 0, & \text{if } \mathbf{x} \geq 0 \\ \infty, & \text{otherwise.} \end{cases}$$

The penalty function encodes a sparse gradient prior and the pixel values are nonnegative. Problem (10.3) shows the full optimization problem and how we represent it in the form of Problem (10.2).

$$\mathbf{x}_{\text{opt}} = \underset{\mathbf{x}}{\operatorname{argmin}} \|\mathbf{MB}\mathbf{x} - \mathbf{j}\|_2^2 + \Gamma(\mathbf{x}) \quad (10.3)$$

$$\Gamma(\mathbf{x}) = \mu\|\nabla\mathbf{x}\|_1 + (1 - \mu)\|\nabla\mathbf{x}\|_2^2 + I_{[0,\infty)}(\mathbf{x}) \quad (10.4)$$

$$\begin{aligned} \text{model:} \quad & f_1(\mathbf{v}) = \|\mathbf{v} - \mathbf{j}\|_2^2, & \mathbf{K}_1 &= \mathbf{MB} \\ & f_2(\mathbf{v}) = \mu\|\mathbf{v}\|_1, & \mathbf{K}_2 &= \nabla \\ & f_3(\mathbf{v}) = (1 - \mu)\|\mathbf{v}\|_2^2, & \mathbf{K}_3 &= \nabla \\ & f_4(\mathbf{v}) = I_{[0,\infty)}(\mathbf{v}), & \mathbf{K}_4 &= \mathbb{I} \end{aligned} \quad (10.5)$$

Note that there are other ways to represent the problem in our standard form. For example, we could use:

$$f_1(\mathbf{v}) = \|\mathbf{M}\mathbf{v} - \mathbf{j}\|_2^2, \quad \mathbf{K}_1 = \mathbf{B}.$$

Note, that this ambiguity also existed in Chapters 3, 5, 4, 6, 8 and 9 where we reformulated the respective objectives into the form of Section 2.8.3 is one of many. For example, the choice of the operators from Eq. (9.7) in Chapter 9. A key insight is that the choice of representation can drastically affect the performance of the solver algorithms. We take advantage of this fact and provide strategies to find an optimal reformulation.

The only assumption we make about the penalty functions  $f_1, \dots, f_I$  is that they provide a black box for evaluating the function's proximal operator; recall Eq. (2.43) from Section 2.8.4. Many algorithms can be carried out using proximal operators that cannot be carried out using the traditional approach of interacting with functions by computing their gradients and Hessians [175].

Similarly, the only assumption we make about each linear operator  $\mathbf{K}_i$  is that it provides a black box for evaluating the forward operator  $\mathbf{x} \rightarrow \mathbf{K}_i \mathbf{x}$  and the adjoint operator  $\mathbf{z} \rightarrow \mathbf{K}_i^T \mathbf{z}$ . This is a useful abstraction because many linear operators that arise in optimization problems from image processing are fast transforms, i.e., they have methods for evaluating the forward and adjoint operator that are more efficient than standard multiplication by the operator represented as a dense or sparse matrix. Common fast transforms in image processing include the discrete Fourier transform (DFT), convolution, and wavelet transforms; see [288] for many more examples. For simplicity, we assume that all linear operators are maps from a multidimensional real space  $\mathbb{R}^{n_1 \times \dots \times n_k}$  to another multidimensional real space  $\mathbb{R}^{m_1 \times \dots \times m_\ell}$ . Complex-valued linear operators such as the DFT are represented as real valued operators using the standard embedding of a complex vector in  $\mathbb{C}^{n_1 \times \dots \times n_k}$  as a real vector in  $\mathbb{R}^{2n_1 \times \dots \times n_k}$ .

All solver algorithms in ProxImaL are proximal, matrix-free solvers which only use these black boxes for the linear operators and penalty functions. ProxImaL currently supports the Chambolle-Pock algorithm, ADMM, linearized ADMM, and half-quadratic splitting. These algorithms can solve Problem (10.2) when the functions  $f_1, \dots, f_I$  are convex.

Much state-of-the-art image optimization makes use of nonconvex penalty functions; however, in applications ranging from denoising and deconvolution to burst reconstruction and registration. Patch-based approaches and hard thresholding in particular have been very successful for image reconstruction problems. Surprisingly, the same proximal, matrix-free solvers that work for convex problems yield good results for certain problems that include nonconvex penalty functions, as shown in the previous Chapter 9 and [166, 289]. There is often no guarantee that the algorithms will converge (see conditions in [183] for exceptions). Furthermore, there is no guarantee that they find the optimal  $\mathbf{x}$ , but empirically for many problems with nonconvex penalties the algorithms do produce good results in a reasonable number of iterations. We therefore allow the penalty functions  $f_1, \dots, f_I$  to be nonconvex, even though by doing so we sacrifice guarantees of optimality. For convex problems, we run the solver until convergence criteria are satisfied, but for nonconvex problems we follow the same approach as [289] and return the iteration with the lowest objective value after a fixed number of iterations.

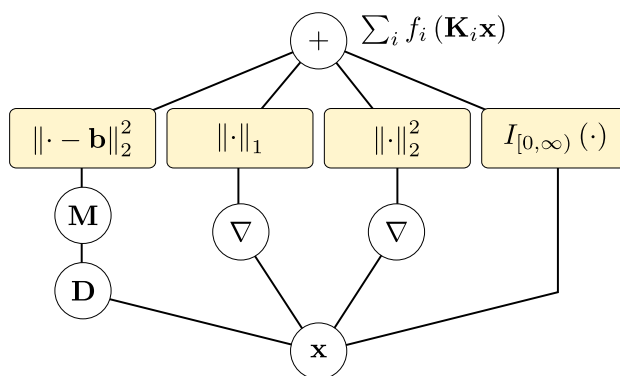
- Problem**(**ProxFn**, ...): Defines a ProxImaL problem as an objective made up of the sum of a list of proxable functions.
- ProxFn**(**LinOp**): The base type of a proxable function, applied to a linear expression. Table 10.2 shows examples from the set of predefined proxable functions.
- LinOp**(...): The base type of a linear expression, each of which has zero or more linear expressions as children. Table 10.3 shows examples from the set of predefined proxable functions.
- Variable**( $w, h, \dots$ ): Defines a set of  $w \times h \times \dots$  unknowns as a single logical multidimensional array variable. A problem may use multiple logical variables of different size and shape, which together form its complete set of unknowns  $\mathbf{x}$ .

**Table 10.1:** Core primitives in the ProxImaL language.

### 10.3 The ProxImaL Language

ProxImaL asks users to describe image optimization problems using a simple DSL, embedded in Python, which corresponds directly to the model in Problem (10.2). The basic language primitives are listed in Table 10.1. At the highest level, the user defines a **Problem** as a list of applications of proxable functions, or functions with a known proximal operator, instantiated as **ProxFn** objects. Each **ProxFn** term applies to a linear expression (**LinOp**), and each linear expression is an arbitrary sub-DAG of linear expressions, ultimately terminating in variable references. **Variables** are defined as multidimensional arrays (e.g.,  $w \times h \times 3$  for a color image). A problem can use an arbitrary number of logical variables, each referenced arbitrarily within the linear expression of any of the proximal terms. Each logical variable refers to individual subcomponents of the vector of all stacked unknowns, i.e.,  $\mathbf{x}$  from Problem (10.2). Variable references make up the leaves of the linear expression DAG, referring to individual components of the unknowns.

Operator overloading translates  $\alpha * \text{expr}$  and  $\text{expr} + \text{expr}$  into the **LinOps** `scale(alpha, expr)` and `sum(expr, expr)`, respectively. Here  $\alpha$  is a scalar constant and  $\text{expr}$  may be any **ProxFn** term or linear expression, though **ProxFn** terms cannot be multiplied by negative constants.



**Figure 10.2:** The DAG representation of Problem (10.3). Directionality from bottom to top.

For example, Problem (10.3) can be written in ProxImaL as:

```

x = Variable(300, 300, 3)
data_term = sum_squares( subsample( conv(x, psf) ) - input )
grad_term = mu * norm1( grad(x) ) + (1-mu) * sum_squares( grad(x) )
objective = data_term + grad_term + nonneg(x)
p = Problem( objective )

```

Figure 10.2 shows how ProxImaL translates the above code into a DAG representation of the problem. The problem can be solved by calling `p.solve(solver="ADMM")`, where the solver keyword specifies what solver algorithm to use, in this case ADMM.

### 10.3.1 Proxable Functions

ProxImaL provides a library of proxable functions that commonly occur in image optimization problems. Table 10.2 lists several examples from the library.

Every proxable function  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  defined in ProxImaL can be parametrized to express any function of the form:

$$\nu(\mathbf{x}) = \alpha f(\beta \mathbf{Q}\mathbf{x} - \mathbf{j}) + \langle \mathbf{c}, \mathbf{x} \rangle + \gamma \langle \mathbf{x}, \mathbf{x} \rangle,$$

where  $\mathbf{x} \in \mathbb{R}^n$  is a variable,  $\alpha > 0$ ,  $\beta \in \mathbb{R}$ ,  $\mathbf{j} \in \mathbb{R}^n$ ,  $\mathbf{c} \in \mathbb{R}^n$  and  $\gamma > 0$  are constants, and  $\mathbf{Q} \in \mathbb{R}^{n \times n}$  is an orthogonal matrix.

- `sum_squares(e)`: Defines the squared  $\ell_2$ -norm  $\|e\|_2^2$  for any linear expression  $e$ .
- `norm1(e)`: Defines an  $\ell_1$ -norm  $\|e\|_1$ .
- `poisson_norm(e, b)`: Defines a maximum-likelihood denoiser that acts as a penalty function. Not a “formal” norm.
- `patch_NLM(e, ...)`: Patch prior for image self-similarity.
- `group_norm1(e, dims)`: Flattens the dimensions `dims` of  $e$  with an  $\ell_2$ -norm, then computes an  $\ell_1$ -norm of the result. Useful to describe sparse norms over vector-valued quantities.
- `nonneg(e)`: Defines a constraint that  $e$  is non-negative (0 cost if so,  $\infty$  otherwise).

**Table 10.2:** Example proxable functions provided by ProxImaL .

The proximal operator of  $\nu$  can be evaluated using only the proximal operator of  $f$ . It is straightforward to show that:

$$\mathbf{prox}_{\tau\nu}(\mathbf{v}) = \mathbf{Q}^T (\mathbf{prox}_{\hat{\tau}f}(\hat{\mathbf{v}}) + \mathbf{j}) / \beta,$$

where  $\hat{\tau} = \frac{\alpha\beta^2\tau}{1+2\gamma\tau}$  and  $\hat{\mathbf{v}} = \frac{\beta}{1+2\gamma\tau} \mathbf{Q}(\mathbf{v} - \tau\mathbf{c}) - \mathbf{j}$ .

When  $f$  is separable, i.e.,  $f(\mathbf{v}) = \sum_{i=1}^n f_i(\mathbf{v}_i)$  for proxable scalar functions  $f_1, \dots, f_n$ , then we can replace  $\beta$  in the parametrization with a diagonal matrix  $\Delta \in \mathbb{R}^{n \times n}$ . Proxable functions may also accept multidimensional inputs, e.g.,  $f : \mathbb{R}^{n_1 \times \dots \times n_k} \rightarrow \mathbb{R}$ . In that case the parametrized form of the function is defined similarly, with  $\mathbf{j}, \mathbf{c} \in \mathbb{R}^{n_1 \times \dots \times n_k}$  and  $\mathbf{Q}$  an orthogonal linear map. In the ProxImaL language, each of these additional parameters is passed as an optional keyword argument to the proxable function constructor (e.g., `alpha=...`, `beta=...`). The ProxImaL compiler takes advantage of the parametrized form internally to rewrite optimization problems.

### 10.3.2 Linear Operators

ProxImaL provides a library of linear operators that include standard operations like addition and multiplication by a constant, as well as common image processing operations. Table 10.3 lists several examples from the library. Including linear operators for image processing operations like convolution and warp in the language,



<code>conv</code>	( $e, k$ ):	Convolve the subexpression $e$ with the kernel $k$ .
<code>subsample</code>	( $e, \text{steps}$ ):	Extracts every $\text{steps}_i$ pixel along axis $i$ , starting with the pixel $\text{steps}_i-1$ .
<code>mul_elemwise</code>	( $\text{weight}, e$ ):	Element-wise multiplication with a fixed constant weight array.
<code>scale</code>	( $c, e$ ):	Scale $e$ by fixed constant scalar $c$ .
<code>sum</code>	( $e_1, e_2, \dots$ ):	Sums input expressions into a single linear expression.
<code>vstack</code>	( $e_1, e_2, \dots$ ):	Vectorizes and stacks a list of input expressions into a single linear expression.
<code>grad</code>	( $e$ ):	Computes the gradients of $e$ , by default across all $n$ of its dimensions.
<code>warp</code>	( $e, H$ ):	Interprets $e$ as a 2D image and warps it using the homography $H$ with linear interpolation.
<code>mul_color</code>	( $e, C$ ):	Performs a blockwise $3 \times 3$ color transform using the color matrix $C$ , or the predefined opponent ( <code>opp</code> ) and YUV ( <code>yuv</code> ) color spaces.

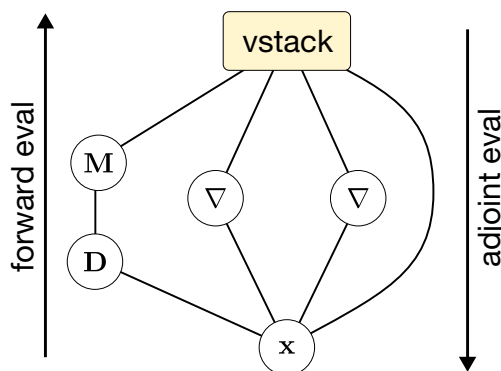
$x$ : Variable references make up the leaves of the linear expr. DAG.

**Table 10.3:** Example linear operators provided by ProxImaL .

rather than having the user write these as multiplication by a dense or sparse matrix, is crucial to generating efficient solver implementations. The high-level descriptions of the linear operators can be exploited by a matrix-free solver to evaluate the operators efficiently, as discussed in [290].

Compositions of linear operators are represented as expression DAGs. Figure 10.3 shows the DAG for the linear operators in Problem (10.3) stacked into a single operator  $K$ , as in Problem (10.2). The DAG structure makes it easy to evaluate the composition. We simply visit the nodes in topological order, reading the input to each node’s linear operator from the node’s incoming edges, applying the operator, and writing the outputs to the node’s outgoing edges. The overall input for the composition is the input to the variable nodes, and the final output is the output of the root node.

For example, to evaluate the composition in Figure 10.3 on an input  $x$ , we first evaluate the variable node by reading  $x$  as input and writing  $x$  to the node’s outgoing edges. Next, we evaluate the  $\mathbf{B}$  and  $\nabla$  nodes in any order. We can evaluate



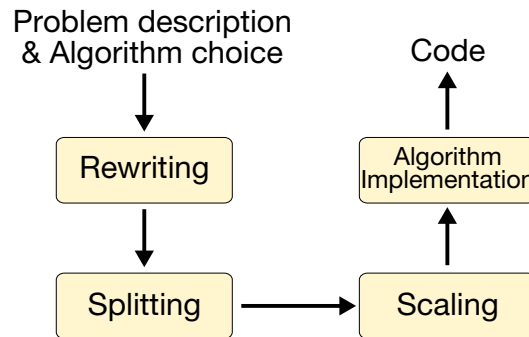
**Figure 10.3:** The DAG for the stacked linear operators in Problem (10.3).

the **M** node any time after we evaluate the **B** node. We finish by evaluating the `vstack` node to get the final output.

We can evaluate the adjoint of the composition just as easily. We follow the same algorithm as for forward evaluation, but visit the nodes in reverse topological order (starting at the root and ending with the variable nodes), reading from the node’s outgoing edges, evaluating the adjoint of the node’s linear operator, and writing the result to the node’s incoming edges.

Even for the simple example in Figure 10.3, there are many possibilities for making the forward and adjoint evaluation algorithms more efficient. The **B** and  $\nabla$  nodes could be evaluated in parallel, and the graph could be rewritten so the  $\nabla$  operator is only evaluated once. See [288] for a detailed discussion of possible optimizations, which we are planning to add in our implementation.

*Extensibility* ProxImaL also supports extending the set of proxable functions and linear operators beyond the built-ins provided. A linear operator extends the `LinOp` interface and must define `forward` and `adjoint` methods, which consume arrays of input values to produce arrays of output values. Linear operators may extend additional optional methods that provide information about how to invert the operator, as discussed in Section 10.6. A proxable function extends the `ProxFn` interface and must define the proxable operator `prox(tau, v)`.



**Figure 10.4:** The ProxImaL compiler pipeline.

## 10.4 Compiling Problems to Efficient Solvers

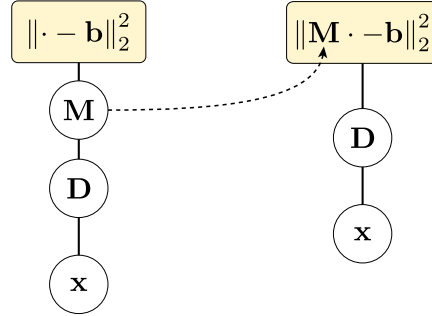
One of the main contributions in ProxImaL is a compiler that takes a problem specification and choice of algorithm and automatically generates an efficient solver implementation. In this section, we describe the stages of the compiler. Figure 10.4 shows an overview of the compiler pipeline. We use Problem (10.3) as a running example throughout our discussion.

### 10.4.1 Rewriting Problems

The first stage of the compiler attempts to rewrite the optimization problem in a form better suited for the solver algorithm. The standard form in Problem (10.2) is not unique. Any given optimization problem written in the form of Problem (10.2) can be rewritten as many equivalent problems also in the same form. Our compiler considers two kinds of rewrites: absorbing linear operators into proxable functions and merging proxable functions.

*Absorbing Linear Operators* Concretely, given a proxable function  $f$  composed with a composite linear operator  $\mathbf{K}$  represented as an expression tree  $e$ , absorbing a linear operator means removing the root linear operator  $\tilde{\mathbf{K}}$  of  $e$  and replacing  $f$  with the composition  $f \circ \tilde{\mathbf{K}}$ . Absorbing a linear operator is only possible when the root of the expression tree has exactly one child.

Whether absorbing a linear operator is a good idea depends on whether the



**Figure 10.5:** Absorbing a linear operator. Directionality from bottom to top.

composition  $f \circ \tilde{\mathbf{K}}$  has an efficient proximal operator. For example, consider the proxable function  $f_1(\mathbf{v}) = \|\mathbf{v} - \mathbf{j}\|_2^2$  and linear operator  $\mathbf{K}_1 = \mathbf{M}\mathbf{B}$  from Problem (10.3). We can absorb a linear operator by replacing  $f_1$  with  $\tilde{f} = f_1 \circ \mathbf{M}$  and  $\mathbf{K}_1$  with  $\tilde{\mathbf{K}} = \mathbf{B}$  Figure 10.5 shows the expression trees for  $f_1 \circ \mathbf{K}_1$  and  $\tilde{f} \circ \tilde{\mathbf{K}}$ . In this case,  $\tilde{f}$  has the proximal operator

$$\mathbf{prox}_{\tau\tilde{f}}(\mathbf{v}) = \left( \frac{1}{2\tau}\mathbb{I} + \mathbf{M}^T\mathbf{M} \right)^{-1} (\mathbf{M}^T\mathbf{j} + \mathbf{v}/\sqrt{2\tau}),$$

which can be computed efficiently because  $\mathbf{M}^T\mathbf{M}$  is diagonal.

We could absorb a linear operator again by replacing  $\tilde{f}(\mathbf{v}) = \|\mathbf{M}\mathbf{v} - \mathbf{j}\|_2^2$  and  $\tilde{\mathbf{K}} = \mathbf{B}$  with  $\hat{f}(\mathbf{v}) = \|\mathbf{M}\mathbf{B}\mathbf{v} - \mathbf{j}\|_2^2$  and  $\hat{\mathbf{K}} = \mathbb{I}$ . In this case, the proximal operator:

$$\mathbf{prox}_{\tau\hat{f}}(\mathbf{v}) = \left( \frac{1}{2\tau}\mathbb{I} + \mathbf{M}^T\mathbf{B}^T\mathbf{B}\mathbf{M} \right)^{-1} (\mathbf{B}^T\mathbf{M}^T\mathbf{j} + \mathbf{v}/\sqrt{2\tau}),$$

is not as efficient. We could compute the proximal operator using an iterative method such as the conjugate gradient method (CG) [291] or LSQR [292], which only interact with  $\mathbf{B}$  and  $\mathbf{M}$  by evaluating the linear operators and their adjoints. However, such iterative methods are more computationally expensive and less accurate than the methods for computing the proximal operator of  $f_1$  or  $\tilde{f}$ .

*Merging Proxable Functions* Merging proxable functions means replacing two proxable functions  $f_i$  and  $f_j$  that are composed with the same linear operator (i.e.,

$\mathbf{K}_i = \mathbf{K}_j$ ) with a new function  $g(\mathbf{v}) = f_i(\mathbf{v}) + f_j(\mathbf{v})$ . In Problem (10.3), we can merge the proxable functions  $f_2(\mathbf{v}) = \mu\|\mathbf{v}\|_1$  and  $f_3(\mathbf{v}) = (1 - \mu)\|\mathbf{v}\|_2^2$  because they are both composed with the linear operator  $\mathbf{K}_2 = \mathbf{K}_3 = \nabla$ . As with absorbing linear operators, merging proxable functions is only a good idea if the new function still has an efficient proximal operator. Note that our conditions for merging functions also include the common case when both functions are parametrized forms of the same function. We exploit here the general formulation of a proxable function from Section 10.3.1. For example, two functions  $f_a(\mathbf{v}) = \mu\|\mathbf{v}\|_2$  and  $f_b(\mathbf{v}) = \rho\|\mathbf{Q}\mathbf{v}\|_2^2$ , where  $\mu, \rho > 0$  and  $\mathbf{Q}$  is an orthogonal matrix, turn out to be parametrizations of the  $\ell_2$ -norm using the general definition. In addition, this example demonstrates that merging functions and absorbing operators work together symbiotically: the operator  $\mathbf{Q}$  can be absorbed to make  $\mathbf{K}_a = \mathbf{K}_b$ . In general the parametrized form provides many opportunities to absorb linear operators and merge proxable functions into efficient compound proxable functions. In our example (10.3) the merged function  $g(\mathbf{v}) = \mu\|\mathbf{v}\|_1 + (1 - \mu)\|\mathbf{v}\|_2^2$  has an efficient proximal operator because it is a parametrized form of  $\tilde{f}(\mathbf{v}) = \|\mathbf{v}\|_1$  discussed previously in Section 10.3.1.

*Default Choices* Absorbing linear operators and merging proxable functions generally simplifies the optimization problem and makes it easier to solve. At a minimum, absorbing linear operators makes multiplication by the overall linear operator  $\mathbf{K}$  from Problem (10.2) more efficient.

Our compiler by default iterates over the proxable functions, and for each one repeatedly absorbs linear operators until doing so would substantially increase the cost of the proximal operator. The compiler then considers all pairs of proxable functions and greedily merges them whenever it can do so and still have an efficient proximal operator. In the context of Problem (10.3), our compiler would absorb the linear operator  $\mathbf{M}$  but not the linear operator  $\mathbf{B}$  and would merge the two functions  $\alpha\|\nabla\mathbf{x}\|_1$  and  $(1 - \alpha)\|\nabla\mathbf{x}\|_2^2$ .

### 10.4.2 Problem Splitting

The second step in the compilation is problem splitting. The optimization algorithms in ProxImaL are operator splitting methods that solve problems in the standard form:

$$\begin{aligned} & \text{minimize} && g(\mathbf{x}) + h(\mathbf{z}) \\ & \text{subject to} && \mathbf{K}\mathbf{x} = \mathbf{z}, \end{aligned} \tag{10.6}$$

where  $\mathbf{x} \in \mathbb{R}^n$  and  $\mathbf{z} \in \mathbb{R}^m$  are variables,  $\mathbf{K} \in \mathbb{R}^{m \times n}$  is a known linear operator, and  $g : \mathbb{R}^n \rightarrow \mathbb{R}$  and  $h : \mathbb{R}^m \rightarrow \mathbb{R}$  have known proximal operators. Some of the algorithms in fact allow a more general standard form with a linear constraint of the form  $\mathbf{K}\mathbf{x} + \mathbf{B}\mathbf{z} = \mathbf{c}$ , but our compiler does not currently take advantage of this, though future, more sophisticated versions of the compiler will.

The compiler expresses  $g$  and  $h$  as:

$$g(\mathbf{x}) = \sum_{f_i \in \Omega} f_i(\mathbf{x}), \quad h(\mathbf{z}) = \sum_{f_i \in \Psi} f_i(\mathbf{z}),$$

where  $\Omega$  and  $\Psi$  are a partition of the set of functions  $\{f_1, \dots, f_I\}$  from Problem (10.2). Problem splitting means choosing  $\Omega$  and  $\Psi$ .

The splitting  $\Psi = \{f_1, \dots, f_I\}$  is always valid for the algorithms in ProxImaL, so the compiler always has at least one choice of splitting. The functions  $f_i \in \Omega$  must have the identity as their linear operator, i.e.,  $\mathbf{K}_i = \mathbb{I}$ . This may seem unduly restrictive, but recall that the compiler tries to absorb linear operators, which means that penalty functions that began with complex  $\mathbf{K}_i$  may end up with  $\mathbf{K}_i = \mathbb{I}$  by the problem splitting stage. Just because a function has the identity as its linear operator does not mean including it in  $\Omega$  is a good idea. Each algorithm has its own logic for choosing when to include a function in  $\Omega$ .

### 10.4.3 Problem Scaling

The third stage of the compiler is problem scaling. Problem scaling replaces Problem (10.6) with the equivalent problem:

$$\begin{aligned} & \text{minimize} && g(\mathbf{A}\hat{\mathbf{x}}) + h(\mathbf{B}^{-1}\hat{\mathbf{z}}) \\ & \text{subject to} && \mathbf{B}\mathbf{K}\mathbf{A}\hat{\mathbf{x}} = \hat{\mathbf{z}}, \end{aligned} \tag{10.7}$$

where  $\mathbf{A} \in \mathbb{R}^{n \times n}$  and  $\mathbf{B} \in \mathbb{R}^{m \times m}$  are invertible and the new variables  $\hat{\mathbf{x}} \in \mathbb{R}^n$  and  $\hat{\mathbf{z}} \in \mathbb{R}^m$  are related to  $\mathbf{x}$  and  $\mathbf{z}$  in Problem (10.6) via:

$$\mathbf{x} = \mathbf{A}\hat{\mathbf{x}}, \quad \mathbf{z} = \mathbf{B}^{-1}\hat{\mathbf{z}}.$$

Problem scaling can substantially affect the number of iterations our solver algorithms take to converge [293, 294]. Our compiler by default sets:

$$\mathbf{A} = \frac{1}{\sqrt{\|\mathbf{K}\|_2}}\mathbb{I}, \quad \mathbf{B} = \frac{1}{\sqrt{\|\mathbf{K}\|_2}}\mathbb{I},$$

where  $\|\mathbf{K}\|_2$  is the spectral norm, or maximum singular value of  $\mathbf{K}$ . We use the implicitly restarted Arnoldi method to compute  $\|\mathbf{K}\|_2$  using only multiplication by  $\mathbf{K}$  and  $\mathbf{K}^T$  [295].

## 10.5 Proximal Algorithms

The final stage of our compiler is generating (or calling) an optimization algorithm. This is a major task in the overall solver development; recall the derivations by hand in Chapters 3, 5, 4, 6, 8 and 9. We provide a detailed description of the optimization methods in this section. ProxImaL currently provides the Chambolle-Pock method, ADMM, linearized ADMM, and half-quadratic splitting. We describe the algorithms of these four methods, discuss convergence and stopping criteria, and how our compiler chooses default problem splittings and hyper-parameters.

*Chambolle-Pock Implementation* The pseudo-code for the Chambolle-Pock algorithm is given in Algorithm 5. Our compiler uses the default  $\tau = \sigma = 1/\|\mathbf{K}\|_2$ ,  $\theta = 1$ ,  $\mathbf{x}^0 = 0$ , and  $\mathbf{z}^0 = 0$ . With default scaling, we have  $\|\mathbf{K}\|_2 = 1$ .

Our compiler only allows at most one penalty function  $f_i$  to be included in  $\Omega$ , and the operator  $\mathbf{K}_i$  must be the identity. The restriction on  $\Omega$  ensures that the algorithm can be carried out using known proximal operators. The compiler's default problem splitting is to include one penalty function in  $\Omega$  whenever possible. For the example Problem (10.3), the penalty function  $f_4(\mathbf{v}) = I_{[0,\infty)}(\mathbf{v})$  would be included in  $\Omega$  and all other penalty functions would be in  $\Psi$ .

---

**Algorithm 5** Chambolle-Pock to solve Problem (10.2)

---

1: Initialization:  $\sigma\tau\|\mathbf{K}\|_2^2 < 1$ ,  $\theta \in [0, 1]$ ,  $(\mathbf{x}^0, \mathbf{z}^0)$ ,  $\bar{\mathbf{x}}^0 = \mathbf{x}^0$ .  
2: **for**  $k = 1$  to  $V$  **do**  
3:  $\mathbf{z}_j^{k+1/2} = \mathbf{z}_j^k + \sigma\mathbf{K}_j\bar{\mathbf{x}}^k \quad \forall j \in \Psi$   
4:  $\mathbf{z}_j^{k+1} = \mathbf{z}_j^{k+1/2} - \sigma\mathbf{prox}_{f_j/\sigma}(\mathbf{z}_j^{k+1/2}/\sigma) \quad \forall j \in \Psi$   
5: **if**  $\Omega = \{f_i\}$  **then**  
6:  $\mathbf{x}^{k+1} = \mathbf{prox}_{\tau f_i}(\mathbf{x}^k - \tau\mathbf{K}^T\mathbf{z}^{k+1})$   
7: **else**  
8:  $\mathbf{x}^{k+1} = \mathbf{x}^k - \tau\mathbf{K}^T\mathbf{z}^{k+1}$   
9: **end if**  
10:  $\bar{\mathbf{x}}^{k+1} = \mathbf{x}^{k+1} + \theta(\mathbf{x}^{k+1} - \mathbf{x}^k)$   
11: **end for**

---

Line 4 of Algorithm 5 uses Moreau's Identity [270]. We will show later that the Chambolle-Pock algorithm and linearized ADMM from further below are equivalent to an augmented/preconditioned version of ADMM [178, 296], described next.

*ADMM Implementation* The pseudo-code for ADMM is given in Algorithm 6. Our compiler uses the default hyper-parameters  $\rho = 1$ ,  $\alpha = 1$ ,  $\mathbf{x}^0 = 0$ ,  $\mathbf{z}^0 = 0$ , and  $\lambda^0 = 0$ . The choice of problem splitting in ADMM is more complex than in Chambolle-Pock, since computing step 3 requires more than the proximal operators of the  $f_i \in \Omega$ . Our compiler only allows quadratic functions to be included in  $\Omega$  because then step 3 reduces to solving a least squares problem.

We can use iterative methods such as CG and LSQR to solve the least squares problem using only multiplication by the linear operators and their adjoints. We use

---

**Algorithm 6** ADMM to solve Problem (10.2)

---

1: Initialization:  $\rho > 0$ ,  $\alpha \in (0, 2)$ ,  $(\mathbf{x}^0, \mathbf{z}^0, \lambda^0)$ .  
2: **for**  $k = 1$  to  $V$  **do**  
3:  $\mathbf{x}^{k+1} = \underset{\mathbf{x}}{\operatorname{argmin}} \sum_{i \in \Omega} f_i(\mathbf{x}) + \sum_{j \in \Psi} (\rho/2) \|\mathbf{K}_j\mathbf{x} - \mathbf{z}_j^k + \lambda_j^k\|_2^2$   
4:  $\mathbf{z}_j^{k+1} = \mathbf{prox}_{\frac{f_j}{\rho}}(\mathbf{K}_j(\alpha\mathbf{x}_j^{k+1} + (1-\alpha)\mathbf{x}_j^k) + \lambda_j^k) \quad \forall j \in \Psi$   
5:  $\lambda_j^{k+1} = \lambda_j^k + \mathbf{K}_j(\alpha\mathbf{x}_j^{k+1} + (1-\alpha)\mathbf{x}_j^k) - \mathbf{z}_j^{k+1} \quad \forall j \in \Psi$   
6: **end for**

---



several tricks to accelerate these iterative methods. First, we initialize the iterative methods with the previous solution  $\mathbf{x}^k$ . Second, we initially solve the least squares problem with low accuracy and increase the required accuracy each iteration. This approach keeps the number of linear operator evaluations in the iterative methods relatively constant across iterations, since the previous solution  $\mathbf{x}^k$  is increasingly close to the next solution  $\mathbf{x}^{k+1}$  [297].

*Linearized ADMM Implementation* The pseudo-code for linearized ADMM is given in Algorithm 7. Our compiler uses the default hyper-parameters  $\rho = 1/\|\mathbf{K}\|_2$ ,  $\mu = \|\mathbf{K}\|_2$ ,  $\mathbf{x}^0 = 0$ ,  $\mathbf{z}^0 = 0$ , and  $\lambda^0 = 0$ . With the default problem scaling, we have  $\|\mathbf{K}\|_2 = 1$ . In linearized ADMM the terms  $f_i$  of  $g = \sum_{i \in \Omega} f_i$  can be any function with an efficient proximal operator. The default splitting into  $\Omega$  and  $\Psi$  is to include all quadratic functions in  $\Omega$ .

---

**Algorithm 7** Linearized ADMM to solve Problem (10.2)

---

- 1: Initialization:  $\mu > \rho\|\mathbf{K}\|_2^2$ ,  $(\mathbf{x}^0, \mathbf{z}^0, \lambda^0)$ .
  - 2: **for**  $k = 1$  to  $V$  **do**
  - 3:    $\mathbf{x}^{k+1} = \mathbf{prox}_{\frac{g}{\mu}}(\mathbf{x}^k - (\rho/\mu)\mathbf{K}^T(\mathbf{K}\mathbf{x}^k - \mathbf{z}^k + \lambda^k))$
  - 4:    $\mathbf{z}_j^{k+1} = \mathbf{prox}_{f_j}(\mathbf{K}_j\mathbf{x}_j^{k+1} + \lambda_j^k) \quad \forall j \in \Psi$
  - 5:    $\lambda_j^{k+1} = \lambda_j^k + (\mathbf{K}_j\mathbf{x}_j^{k+1} - \mathbf{z}_j^{k+1}) \quad \forall j \in \Psi$
  - 6: **end for**
- 

Linearized ADMM is a natural variant of standard ADMM. Recall that ADMM applied to Problem (10.2) solves the optimization problem

$$\text{minimize } g(\mathbf{x}) + (\rho/2)\|\mathbf{K}\mathbf{x} - \mathbf{z}^k + \lambda^k\|_2^2$$

as a subroutine. In linearized ADMM we replace the term

$$(\rho/2)\|\mathbf{K}\mathbf{x} - \mathbf{z}^k + \lambda^k\|_2^2$$

with its linearization plus quadratic regularization:

$$\rho\mathbf{K}^T(\mathbf{K}\mathbf{x} - \mathbf{z}^k + \lambda^k) + (\mu/2)\|\mathbf{x} - \mathbf{x}^k\|_2^2.$$

The Chambolle-Pock algorithm is in fact linearized ADMM applied to the dual of Problem (10.2) [178]. Chambolle-Pock could therefore be implemented in ProxImaL as a symbolic transformation that converts the problem representation into its dual, combined with the linearized ADMM implementation. The dual problem involves the conjugates of  $f_1, \dots, f_I$  from Problem (10.2) (see Problem (10.12)). The proximal operator for the conjugate  $f_i^*$  can be evaluated using the proximal operator for  $f_i$  and Moreau's Identity [270].

*Half-Quadratic Splitting Implementation* The pseudo-code for half-quadratic splitting is given in Algorithm 8. Our compiler uses the default hyper-parameters  $\rho^0 = 1$ ,  $\alpha = 2$ ,  $\rho_{\max} = 2^8$ ,  $\mathbf{x}^0 = 0$ , and  $\mathbf{z}^0 = 0$ . Our compiler only allows quadratic functions to be included in  $\Omega$  because then step 3 reduces to solving a least squares problem. The default splitting into  $\Omega$  and  $\Psi$  is the same as for ADMM.

---

**Algorithm 8** Half-Quadratic Splitting to solve Problem (10.2)

---

- 1: Initialization:  $\rho^0 > 0$ ,  $\rho_{\max} > 0$ ,  $\alpha > 1$ ,  $(\mathbf{x}^0, \mathbf{z}^0)$ .
  - 2: **for**  $k = 1$  to  $V$  **do**
  - 3:  $\mathbf{x}^{k+1} = \underset{\mathbf{x}}{\operatorname{argmin}} \sum_{i \in \Omega} f_i(\mathbf{K}_i \mathbf{x}) + \rho^k \sum_{j \in \Psi} \|\mathbf{K}_j \mathbf{x} - \mathbf{z}_j\|_2^2$
  - 4:  $\mathbf{z}_j^{k+1} = \operatorname{prox}_{\frac{f_j}{\rho^k}}(\mathbf{K}_j \mathbf{x}_j^{k+1}) \quad \forall j \in \Psi$
  - 5:  $\rho^{k+1} = \max\{\rho^k * \alpha, \rho_{\max}\}$
  - 6: **end for**
- 

On the first glance, the half-quadratic splitting (HQS) implementation looks similar to our ADMM implementation. However, the key difference is the Lagrange multipliers in ADMM have been eliminated in HQS. The use of the Lagrange multipliers in ADMM allows to have a fixed  $\rho$ . In HQS, by contrast, we need to scale  $\rho \rightarrow \infty$ . This scaling can cause both the quadratic step in line 3 and the proximal operator step in line 4 to be unstable. Thus, for the HQS method to be stable, it is crucial to minimize the number of splitting variables and find numerically accurate solutions to both steps (which is not necessary for ADMM).

*Direct Methods* Often the least squares problem can be solved using a simple direct method, which makes the solver implementation much faster and more reliable. For

example, recall that after the rewriting stage our example problem (10.3) has the form:

$$\mathbf{x}_{\text{opt}} = \underset{\mathbf{x}}{\operatorname{argmin}} \|\mathbf{M}\mathbf{B}\mathbf{x} - \mathbf{j}\|_2^2 + \Gamma(\mathbf{x}) \quad (10.8)$$

$$\Gamma(\mathbf{x}) = \mu\|\nabla\mathbf{x}\|_1 + (1 - \mu)\|\nabla\mathbf{x}\|_2^2 + I_{[0,\infty)}(\mathbf{x}) \quad (10.9)$$

$$\begin{aligned} f_1(\mathbf{v}) &= \|\mathbf{M}\mathbf{v} - \mathbf{j}\|_2^2, & \mathbf{K}_1 &= \mathbf{B} \\ \text{model: } f_2(\mathbf{v}) &= \mu\|\mathbf{v}\|_1 + (1 - \mu)\|\mathbf{v}\|_2^2, & \mathbf{K}_2 &= \nabla \\ f_3(\mathbf{v}) &= I_{[0,\infty)}(\mathbf{v}), & \mathbf{K}_3 &= \mathbb{I} \end{aligned} \quad (10.10)$$

If we choose the problem splitting:

$$\Omega = \emptyset, \quad \Psi = \{f_1, f_2, f_3\},$$

and consider solving it using ADMM, then step 3 of ADMM becomes:

$$\begin{aligned} \mathbf{x}^{k+1} &= \underset{\mathbf{x}}{\operatorname{argmin}} \left\| \begin{bmatrix} \mathbf{B} \\ \nabla \\ \mathbb{I} \end{bmatrix} \mathbf{x} - \mathbf{z}^k + \lambda^k \right\|_2^2 \\ &= (\mathbf{B}^T\mathbf{B} + \nabla^T\nabla + \mathbb{I})^{-1}(\mathbf{B}^T + \nabla^T + \mathbb{I})(\mathbf{z}^k - \lambda^k). \end{aligned}$$

The linear operators  $\mathbf{B}$ ,  $\nabla$ , and  $\mathbb{I}$  are convolutions and thus diagonal in the frequency domain. Note that arbitrary boundary conditions can be supported by padding and masking the observations [242]. Given the convolutional operations, we can compute a diagonal matrix  $\Delta \in \mathbb{C}^{n \times n}$  for which:

$$(\mathbf{B}^T\mathbf{B} + \nabla^T\nabla + \mathbb{I})^{-1} = \Phi^{-1}\Delta\Phi,$$

where  $\Phi \in \mathbb{C}^{n \times n}$  is the DFT matrix. The upshot is that we can solve the least squares problem exactly in  $O(n \log n)$  operations using the FFT.

Our compiler includes a system for automatically detecting when a linear operator  $\mathbf{K}$  has a Gram matrix  $\mathbf{K}^T\mathbf{K}$  that is diagonal in the spatial or frequency domain and obtaining the Gram matrix's diagonal representation; see Section 10.6 for details. This allows the compiler to automatically exploit the fast direct method.

*Default Splitting* The compiler uses the following procedure to choose the problem splitting. We consider each subset  $S$  of quadratic penalty functions in the problem. We check whether for  $\Omega = S$  the least squares problem (step 3 for ADMM) has a Gram matrix that is diagonal in the spatial or frequency domain, in which case we can solve the problem with a fast direct method. We then set  $\Omega$  to be a maximal cardinality subset for which the Gram matrix is diagonal. If there is no subset for which the Gram matrix is diagonal, we include all the quadratic penalty functions in  $\Omega$  and solve the least squares problem using CG.

One might be concerned about the runtime of this method since the number of subsets is exponential in the number of quadratic penalty functions. However, hardly any problems will have more than a few (often one or two) quadratic penalty functions, and our method for checking whether the Gram matrix is diagonal is linear time in the number of linear operators.

For the rewriting of the example Problem (10.3) chosen by our compiler, given in Equation (10.10), our method would choose  $\Omega = \emptyset$  and solve the least squares problem directly. For the original, naive formulation in Equation (10.5) our method would choose  $\Omega = \{f_1, f_3\}$  and use CG to solve the least squares problem. The choices made by our compiler can thus dramatically improve the performance of the solver implementation over a naive approach that always uses an iterative method or does not consider rewrites. Problem (10.3) may seem contrived, but solving the least squares problem with a direct method when possible is necessary for our system to be competitive with specialized solvers, and in fact was sufficient for state-of-the-art performance on a demosaicking problem; see Section 10.8 for details.

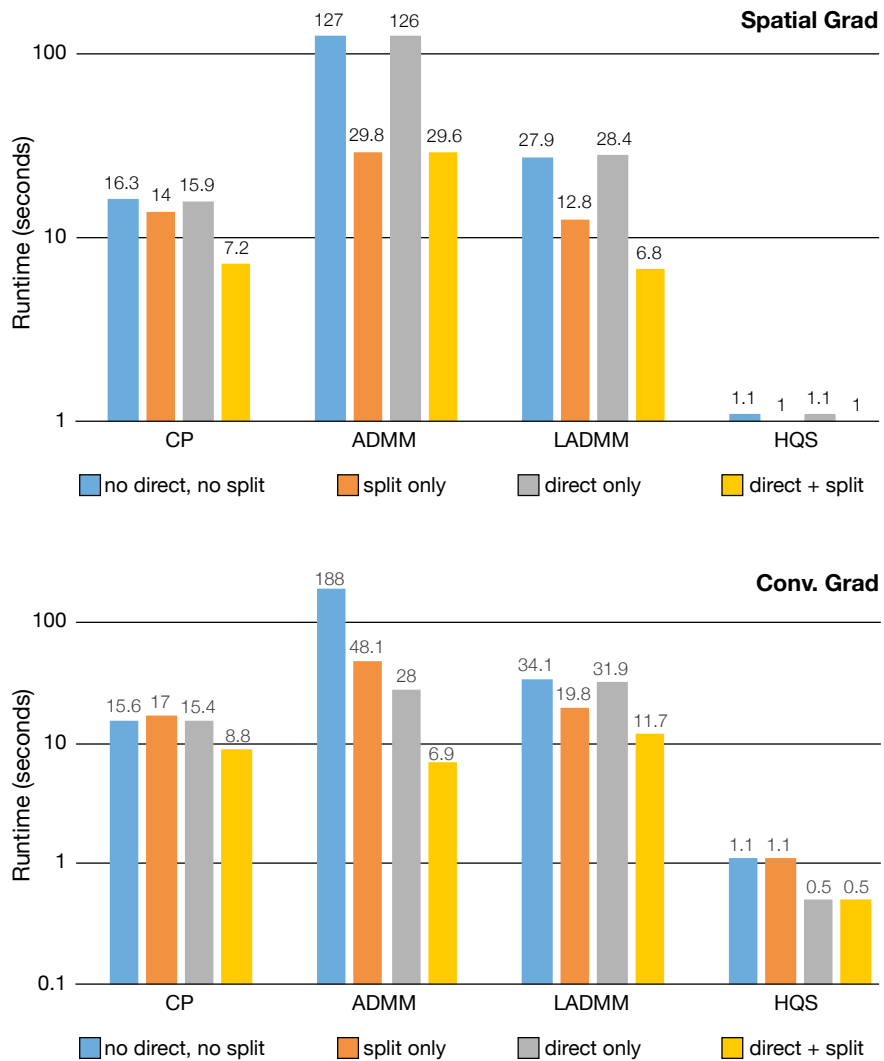
*Manual Options* We allow the user to override any of the default compiler choices. The user can manually rewrite the problem by changing the problem definition, choosing the problem splitting and least squares solver, and setting the problem scaling and algorithm hyper-parameters. An important example of where manual input is helpful is specifying the starting iterates in the solver algorithm (e.g.,  $(\mathbf{x}^0, \mathbf{z}^0)$  for Chambolle-Pock). Starting from a good iterate can dramatically reduce the number of solver iterations needed and the quality of the solution for nonconvex problems as shown in the previous Chapter 9.

*Empirical Validation* Figure 10.6 shows that the choices made by our compiler can dramatically reduce the solver runtime. We list the average runtime for a deconvolution problem regularized by total variation (TV) for each of the four algorithms in ProxImaL: Chambolle-Pock (CP), ADMM, linearized ADMM (LADMM), and half-quadratic splitting (HQS). Note that HQS does not have convergence guarantees for arbitrary convex problems fitting into Problem (10.2), but only for a subset [298], however, including common problems such as the considered TV-regularized deconvolution. Each of the four algorithms is tested with and without the compiler’s intelligent rewritings and splitting, as well as with and without the compiler automatically replacing CG with a fast direct method by detecting diagonal matrices. Further, we evaluate implementations of the TV in the spatial domain (Figure 10.6 left) and as a multiplication in the Fourier domain (Figure 10.6 right). The latter case assumes circular boundary conditions but can sometimes be faster. Assuming circular boundary conditions slightly modifies the optimization problem being solved, but we evaluate all our results using the original objective, with non-circular boundary conditions. All algorithms are run until convergence to the exact same objective function value. The results show that the choices made by the compiler for all algorithms improve the runtime, often substantially. Please note that the absolute runtime depends in general on the algorithm-specific implementation and parameters, including effects of potentially adaptive parameter schedules as for example discussed in [299]. However, Figure 10.6 shows that independent of the algorithm choice, the choice made by our compiler significantly improves runtime.

### 10.5.1 Stopping Criteria

There are many possible ways to determine when the solver algorithms have converged, and different choices may work better for different problems. A reasonable, general purpose stopping criteria is that proposed in [174], in which the algorithm terminates when the norms of the primal residual  $r^{k+1} = \mathbf{K}\mathbf{x}^{k+1} - z^{k+1}$  and dual residual  $s^{k+1} = \rho\mathbf{K}^T(z^{k+1} - z^k)$  fall below certain thresholds. The threshold for the norm of the primal residual is given by

$$\epsilon^{\text{pri}} = \sqrt{m}\epsilon^{\text{abs}} + \epsilon^{\text{rel}} \max\{\|\mathbf{K}\mathbf{x}^k\|_2, \|\mathbf{z}^k\|_2\},$$



**Figure 10.6:** Runtime (in seconds) of a TV-regularized deconvolution problem in ProxImaL. When the *direct* parameter is on, the ProxImaL compiler automatically replaces CG with a fast direct method. The *split* option further indicates whether the compiler’s intelligent rewriting and splitting are used. We evaluate four different algorithms that are implemented in ProxImaL. Note that independent of the algorithm choice, our compiler choice improves runtime. We evaluate two implementations of the TV prior: a finite differences implementation in the spatial domain (top) and convolutional implementation (bottom).

where  $\epsilon^{\text{abs}}$  and  $\epsilon^{\text{rel}}$  are chosen by the user. By default ProxImaL uses  $\epsilon^{\text{abs}} = \epsilon^{\text{rel}} = 10^{-3}$ . Similarly, the threshold for the norm of the dual residual is given by

$$\epsilon^{\text{pri}} = \sqrt{n}\epsilon^{\text{abs}} + \epsilon^{\text{rel}}\|\mathbf{K}^T\lambda^k\|_2.$$

This criterion is used in our framework for ADMM, LADMM and PC. Note that all of these methods are variants of the ADMM method [178]. HQS suffers from instabilities for  $\rho \rightarrow \infty$ . Hence, in practice often a stopping criterion is used that limits  $\rho$ , see [160]. In our implementation we use default value of  $\rho_{\text{max}} = 2^8$ . To add robustness for small values of  $\alpha$  we add a progress-based stopping criterion

$$\|\mathbf{x}^{k+1} - \mathbf{x}^k\|_2 + \|\mathbf{z}_j^{k+1} - \mathbf{z}_j^k\|_2 < n\epsilon^{\text{hq}},$$

where our default value for  $\epsilon^{\text{hq}}$  is  $1e - 6$ . Finally, we allow the user to change all parameters of the stopping criteria for each of the implemented methods. In practice often an estimate of modest accuracy, but low computational cost, might be desired.

## 10.5.2 Convergence Properties

Finally, we discuss the conditions under which the solver algorithms in ProxImaL converge. We assume that problem (10.2) has a global minimizer and observes certain regularity conditions. Formally, we assume that the problem

$$\begin{aligned} & \text{minimize} && \sum_{i=1}^I f_i(\mathbf{z}_i) \\ & \text{subject to} && \mathbf{K}_i\mathbf{x} = \mathbf{z}_i, \quad i = 1, \dots, I, \end{aligned} \tag{10.11}$$

is feasible and bounded below, with optimal value equal to the optimal value of its Lagrange dual

$$\begin{aligned} & \text{maximize} && -\sum_{i=1}^I f_i^*(\lambda_i) \\ & \text{subject to} && \mathbf{K}_i^T\lambda_i = 0, \quad i = 1, \dots, I. \end{aligned} \tag{10.12}$$

Under our assumption on problem (10.2), our implementation of Chambolle-Pock is guaranteed to converge to a global minimizer of problem (10.2) if the functions  $f_1, \dots, f_I$  are proper, convex, and lower semicontinuous. The hyperparameters  $\tau$  and  $\sigma$  must be chosen such that  $\sigma\tau\|\mathbf{K}\|_2^2 < 1$  [178].

Our implementation of ADMM is guaranteed to converge to a global minimizer of problem (10.2) if the functions  $f_1, \dots, f_I$  are closed, proper, and convex. Any choice of  $\rho > 0$  is valid [174]. Variations of ADMM where we use a relaxation parameter  $\alpha \in (0, 2)$  or compute the variable updates inexactly still guarantee convergence to a global minimizer under reasonable conditions [300].

Our implementation of linearized ADMM (and hence also the Chambolle-Pock method) is guaranteed to converge to a global minimizer of problem (10.2) if the functions  $f_1, \dots, f_I$  are closed, proper, and convex. The hyper-parameters  $\rho$  and  $\mu$  must be chosen such that  $\rho > 0$  and  $\mu > \rho \|\mathbf{K}\|_2^2$  [228]. The convergence analysis for our implementation of half-quadratic splitting is more complex than for the other algorithms in ProxImaL. When  $\Omega = \{f_1, \dots, f_I\}$ , and the  $\mathbf{x}$ -update is computed exactly, convergence to a global minimizer is guaranteed for  $\rho_{\max} = \infty$  [301, Chap. 17]. If the functions  $f_1, \dots, f_I$  are closed, proper, and convex, and  $\mathbf{K}_1, \dots, \mathbf{K}_I$  are full rank, then for constant  $\rho^k$  (i.e.,  $\alpha = 1$ ), our implementation converges to a global minimizer of the problem

$$\text{minimize } \sum_{i=1}^I f_i(\mathbf{z}_i) + \rho^k \|\mathbf{K}_i \mathbf{x} - \mathbf{z}_i\|_2^2,$$

where the equality constraints have been relaxed [302]. In the fully general scenario where  $f_1, \dots, f_I$  are convex but  $\Psi \neq \emptyset$  and  $\rho^k$  increases, our implementation is a heuristic, but one used successfully in applications, and strongly motivated by theoretical analysis [303].

For our implementation of Half-Quadratic Splitting, convergence results for a large class of objectives have been shown in [298]. However, in general convergence can not be guaranteed (for objectives not in this class). Nevertheless, in practice HQS can provide good results even for problems where convergence is not guaranteed [160].

We have only discussed convergence when problem (10.2) is convex. Recent work has shown, however, that for nonconvex problems under certain conditions Chambolle-Pock, ADMM, linearized ADMM, and variants of half-quadratic splitting are guaranteed to converge to a stationary point [184–186, 298].

ProxImaL does not force the user to obey the restrictions on the functions  $f_1, \dots, f_I$  and the algorithm hyper-parameters needed to guarantee convergence.



Many state-of-the-art results in image optimization involve applying proximal algorithms to nonconvex problems for which it is difficult to provide guarantees about convergence (e.g., [10, 160]).

## 10.6 Analysis of Linear Systems

In this section, we explain our method for automatically detecting when a least squares problem:

$$\text{minimize } \|\mathbf{K}\mathbf{x} - \mathbf{j}\|_2^2, \quad (10.13)$$

can be solved using a fast direct method. Specifically, our system detects when the Gram matrix  $\mathbf{K}^T\mathbf{K}$  is diagonal in the spatial or frequency domain and computes its diagonal representation.

*Diagonal  $\mathbf{K}^T\mathbf{K}$*  We first explain how we determine when  $\mathbf{K}^T\mathbf{K}$  is diagonal. We use the subroutines `is_diag` and `is_gram_diag`. These subroutines take as argument an expression DAG  $e$  of linear operators. The subroutine `is_diag` returns true if the composite linear operator  $\mathbf{K}_e$  defined by  $e$  is diagonal, and false otherwise. The subroutine `is_gram_diag` returns true if  $\mathbf{K}_e^T\mathbf{K}_e$  is diagonal, which is always true if  $\mathbf{K}_e$  is diagonal, but is also true for non-diagonal  $\mathbf{K}_e$  such as:

$$\mathbf{K}_e = \begin{bmatrix} I \\ 0 \end{bmatrix},$$

which represents zero-padding. The output of the subroutines depends on the type of  $e$ 's root node  $n$  and the results of applying the subroutines to the subDAGs rooted at  $n$ 's children. Table 10.4 gives examples of the logic for various linear operator types. Recall that leaves of an expression DAG are always variable nodes, so the behavior for variables is the base case.

We simply apply `is_gram_diag` to the expression DAG representing  $\mathbf{K}$  in Problem (10.13) to determine whether  $\mathbf{K}^T\mathbf{K}$  is diagonal. To find the value of  $\mathbf{K}^T\mathbf{K}$ , we use the `get_diag` subroutine, which takes as argument a variable node  $v$  and an expression DAG  $e$  for which `is_gram_diag(e)` returns true. The subroutine returns the diagonal of  $(\mathbf{K}_{e,v}^T\mathbf{K}_{e,v})^{1/2}$ , where  $\mathbf{K}_{e,v}$  is the linear function of the

<i>type</i>	<i>subroutine logic</i>
x (variable)	is_diag: Always true. is_gram_diag: Always true.
mul_elemwise	is_diag: True if is_diag is true for children. is_gram_diag: True if is_diag is true for children.
subsample	is_diag: Always false. is_gram_diag: True if is_diag is true for children.
sum	is_diag: True if is_diag is true for children. is_gram_diag: True if is_diag is true for children.
vstack	is_diag: Always false. is_gram_diag: True if is_gram_diag is true for children.

**Table 10.4:** Logic for the `is_diag` and `is_gram_diag` subroutines.

variable  $v$  defined by  $e$ . The full diagonal of  $(\mathbf{K}_e^T \mathbf{K}_e)^{1/2}$  is obtained by evaluating `get_diag( $v, e$ )` for all variable nodes  $v$  in  $e$  and stacking the results into a single vector.

The output of `get_diag` depends on  $e$ 's root node  $n$  and the results of applying `get_diag` to the subDAGs rooted at  $n$ 's children. The parent call passes its variable node argument to the recursive calls. The output of `get_diag( $v, e$ )` when  $e$  is a variable node is a vector of ones if  $e = v$  and a vector of zeros otherwise. For other linear operators the logic of `get_diag` is straightforward and does not depend on the variable node argument.

*Diagonal  $\mathbf{K}^T \mathbf{K}$  in the Frequency Domain* We take an analogous approach to determine whether  $\mathbf{K}^T \mathbf{K}$  is diagonal in the frequency domain and, if so, get its diagonal representation. We use the subroutines `is_fdiag`, `is_gram_fdiag`, and `get_fdiag`, which are the same as `is_diag`, `is_gram_diag`, and `get_diag`, respectively, except defined for the frequency domain.

*Extensions* Our approach easily generalizes to linear operators  $\mathbf{K}$  for which  $\mathbf{K}^T \mathbf{K}$  is block diagonal in the spatial or frequency domain. For example, this includes color transformations in the image formation model. We would simply extend

is\_diag, is\_gram\_diag, and their counterparts for the frequency domain to track the dimensions of the blocks on the diagonal. When the blocks are small, it would be worthwhile to factor the blocks and find the solution  $x^*$  to Problem (10.13) directly via  $x^* = \mathbf{K}^\dagger \mathbf{j}$ . A further extension will determine whether  $\mathbf{K}^T \mathbf{K}$  is banded.

## 10.7 Implementation

We implemented ProxImaL as a Python library with syntax inspired by CVXPY [285]. Each solver algorithm is implemented as a Python driver that schedules and evaluates the series of proximal and linear operators which make up the objective, as split and scheduled by our compiler logic. Our implementation only evaluates one operator at a time; a more sophisticated approach would evaluate multiple operators simultaneously. The proximal and linear operators are evaluated using a combination of NumPy and Halide-generated parallel and vectorized x86 code. We are planning to extend our framework to compile into Halide-generated GPU code.

We applied traditional Halide scheduling techniques to optimize the proximal and linear operators and other kernels. Table 10.5 gives an overview of timings for several linear and proximal operators for NumPy and Halide implementations. Most are simply parallelized, vectorized, unrolled, and occasionally with loops reordered. A few (particularly those involving `warp`) also block and fuse one or two stages for locality. We paid particular attention to the common  $\ell_2$ -norm and dot product, which we parallelized using two-phase reductions to expose both vector and multicore parallelism in the main phase. This improved performance by at least an order of magnitude over a basic serial implementation. The current implementation left some

	$\ell_2$ -norm	dot product	subsample	subsample*	grad	grad*	convolution
Halide	41.6	15.6	72.6	72.6	94.8	237.4	121.4
NumPy	245.8	96.6	356.0	356.0	1188.0	713.1	7790.9
	convolution*	warp	warp*	norm1	group norm1	Poisson prox	FFT inversion
Halide	121.4	153.1	367.8	27.1	67.8	44.7	9.4
NumPy	7790.9	457.6	474.4	201.8	1036.6	265.2	23.4

**Table 10.5:** Runtimes (in ms) for linear operators and some of the proximal functions implemented with NumPy and Halide. `subsample*`, `grad*` *etc.* are the adjoint operators.

performance on the floor by conservatively generalizing all kernels to accept any input with either row- or column-major order for compatibility with arbitrary BLAS and Fortran code in NumPy. A number of kernels also still use simple NumPy implementations, without any Halide-generated code. Above all, there is significant opportunity to schedule and fuse the resulting pipelines *across* operators for each solver, drastically improving on our separate scheduling.

## 10.8 Results

In this section, we evaluate ProxImaL for a range of inverse problems in imaging. In particular, we show that ProxImaL exploits problem-specific structure to improve quality and intuitiveness of camera image processing as presented in the previous Chapter 9. In the following, first, the applications from Section 9.4 are demonstrated using ProxImaL. Next, we show state-of-the-art results for deconvolution in the presence of Poisson-distributed shot noise, which makes low-light photography so difficult. Lastly, we show applications to nonlinear image formations for the example of phase retrieval, which is an important problem in scientific imaging. For all applications shown in this section, ProxImaL allows for a very compact representation as shown in Table 10.6.

	Reference	ProxImaL
Burst	1020 (FlexISP)	6
Demosaicking	1020 (FlexISP)	6
IHdr	1020 (FlexISP)	6
Phase retrieval	300 (Matlab)	6
Poisson deconvolution	510 (Matlab)	6
$\ell_2$ deconvolution	360 (Krishnan and Fergus [160])	6

**Table 10.6:** Lines of code comparisons: We compare high-level (Matlab) code of reference methods with splitting done by hand to ProxImaL. For all example applications shown in this section, problems can be expressed in a very compact way using ProxImaL. Different splitting approaches that require large restructuring in the method can be expressed with just a few changed lines of ProxImaL code.

*Demosaicking* The ProxImaL code for the demosaicking application is:

```

x = Variable(300, 300, 3)
data_term = sum_squares( subsample(x, bayer) - input )
patch_similarity = patch_BM3D( tonemap(x) )
grad_sparsity = norm1( grad(x) )
objective = data_term + patch_similarity + grad_sparsity
p = Problem( objective )

```

ProxImaL automatically compiles this expression into the optimization routines required to solve this problem and it also detects the diagonal matrix structure of the subsampling operator, which can be inverted in closed form. As illustrated in Table 10.7, a few lines of ProxImaL code are sufficient to generate the state-

Average PSNR	Red	Green	Blue
FlexISP	38.50	41.84	36.50
ProxImaL	<b>38.54</b>	<b>41.90</b>	<b>36.53</b>

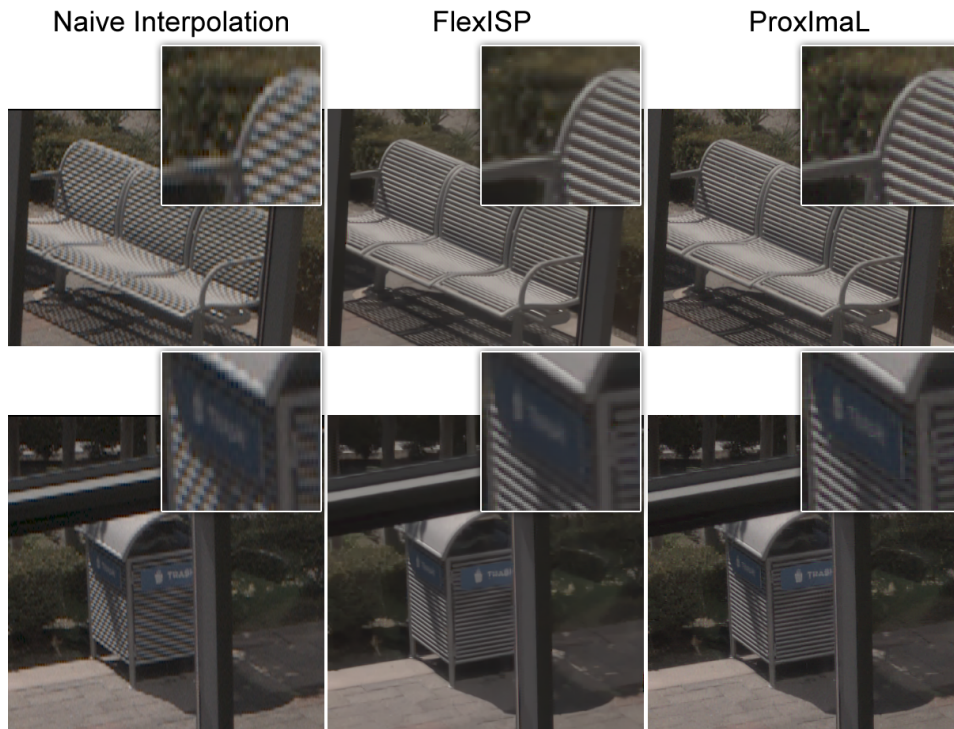
**Table 10.7:** Demosaicking. From only a few lines of code, ProxImaL automatically compiles a solver that achieves qualitatively better results than FlexISP (example on top), averaged over 12 test images quantitatively better (table) and significantly faster (see text).

of-the-art ISP. Due to the fact that FlexISP or any generic solver is oblivious to problem-specific structure, we achieve faster runtimes because ProxImaL detects the closed-form solution of the subsampling operator and generates the most efficient solver. This makes the ProxImaL code faster and also slightly better, on average, than using a proximal operator based on iterative conjugate gradient (e.g., FlexISP). Other than the closed-form inverse of the subsampling step, both approaches are equivalent.

The runtime of our splitting-based demosaicking is, for the most part, determined by two steps: the inversion of the subsampling and blur operators; and the BM3D denoiser. We improve upon the inversion step, which was previously implemented with the conjugate gradient method and takes 0.2 seconds per iteration for a  $512 \times 512$  image and 9 seconds per iteration for a 16 megapixel image on our test computer. The closed-form inverse discovered by ProxImaL basically makes this step “free” and directly finds the optimal solution without any iterations. For the BM3D step, we use a Matlab implementation of BM3D since the code is closed source, which takes about 4 seconds per iteration for a  $512 \times 512$  image; we use 15 iterations in total. Relying on [281] the previous Chapter 9 demonstrated speedups of up to  $1000\times$  using GPU-optimized implementations.

*Interlaced HDR* Next, we consider the interlaced HDR problem described in Section 9.4. The only modification to the demosaicking code from above is a change in the masking operator. The results in Figure 10.7 demonstrates that ProxImaL achieves slightly sharper reconstructions than FlexISP while providing similar performance benefits as for the deconvolution.

*Burst Denoising* High-quality photography in low-light conditions is one of the most challenging problems in computational photography due to the observed noise. Next, we consider the burst image denoising application for low-light imaging, which has been described in Section 9.4. In our implementation we use an additional non-local means (NLM) denoising prior on the reconstructed image, instead of the BM3D prior which has been suggested in the previous Chapter 9. Recall again, that the proximal operator corresponding to any denoising prior can be evaluated by



**Figure 10.7:** Interlaced High Dynamic Range as in Section 9.4. Compared to FlexISP, ProxImaL further improves recovered image sharpness, runtime, and convenience.

calling the denoiser itself. In particular, we formulate the burst denoising problem in ProxImaL as:

```

x = Variable(300, 300, 3)
data_term = sum_squares( vstack( warp(x, H_1),
                                ...,
                                warp(x, H_n) ) - input )
patch_similarity = patch_NLM( tonemap(x) )
objective = data_term + patch_similarity
p = Problem( objective )

```

In addition to the denoiser and the NLM prior, we include the image alignment as part of the optimization routine via the `warp` operator. This operator makes ProxImaL ideally suited for a range of burst photography applications—we simply specify that the global image alignment is part of the problem and the homography





**Figure 10.8:** Comparison of image priors for burst denoising & demosaicking. For this simulation, a non-local means (NLM) prior outperforms the BM3D prior. The approximate NLM implemented in OpenCV is faster than, but not quite as good as, a full implementation.

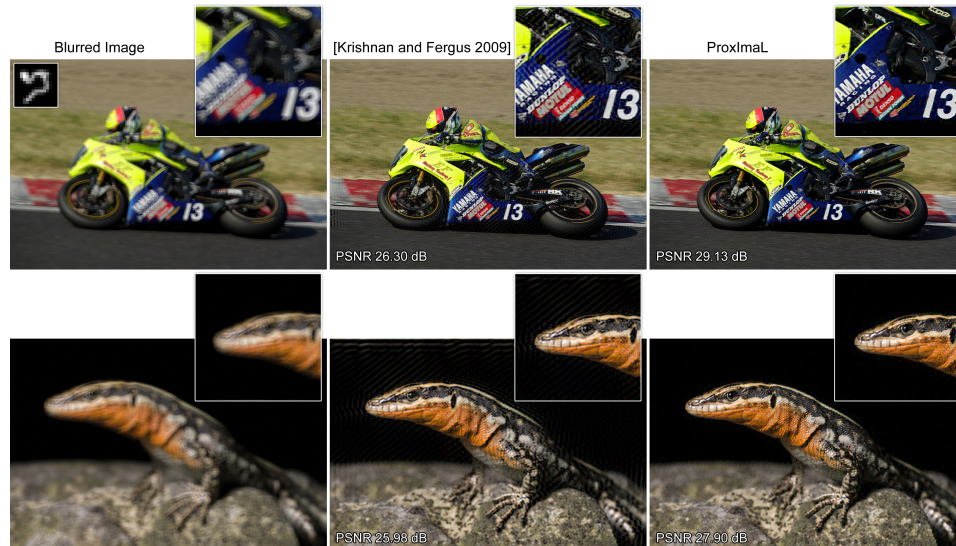
$H$  is automatically estimated using [304]. Feature-based homography estimation also produced results of equal quality.

As discussed before, ProxImaL makes it very easy to evaluate different image priors for a particular reconstruction problem. For example, instead of NLM we can use BM3D, total variation, or other priors. Figure 10.8 evaluates several different priors for burst denoising in simulation. We compare BM3D, the fast approximation of NLM implemented by the OpenCV library, and a full NLM implementation. The latter is the slowest but also the best prior for this particular image. Some of the insights made in the presented experiments include the fact that different priors lead to vastly different results and implementing all of them separately takes a lot of time and effort. ProxImaL makes it very easy to evaluate all of them or combinations of them and it allows different priors to be combined with different solvers. We believe



that this flexibility is one of the strongest benefits of a domain-specific language like ProxImaL.

*Poisson Deconvolution* In low-light conditions, photographs are often not only noisy but also blurry. Especially in cellphone cameras, blur due to hand motion is almost unavoidable. In addition, the noise in images captured in low-light conditions is dominated by Poisson-distributed shot noise. We have assumed this noise to be Gaussian noise in Chapters 3, 5, 4, 6, 8 and 9, which simplified the derivation and efficiency of the optimization method. Instead we can include a Poisson likelihood in our model; see Figure 10.9. The corresponding proximal operator is implemented as a maximum likelihood solution for Poisson noise, given in [305], [306], which we discuss in detail in the appendix Section A.14. With this proximal operator in



**Figure 10.9:** Poisson deconvolution. We compare the deconvolution technique described in the text with the hyper-Laplacian prior described by Krishnan and Fergus [160]. Although the latter method is efficient, it fails in the presence of Poisson-distributed shot noise, which often dominates image noise in low-light photography. An appropriate proximal operator for the noise term can mitigate ringing. This proximal operator is easily compiled with ProxImaL. However, a Poisson-penalty cannot be easily integrated into the approach from [160].

	Kernel 1	Kernel 2	Kernel 3	Kernel 4	Kernel 5
[Krishnan and Fergus 2009]	21.68	21.08	22.70	23.10	25.19
[Figueiredo and Jose 2010]	22.15	<b>22.02</b>	23.59	24.08	26.43
ProxImaL	<b>22.19</b>	21.87	<b>24.11</b>	<b>24.31</b>	<b>26.58</b>

**Table 10.8:** Quantitative evaluation of Poisson deconvolution. We show average peak signal-to-noise-ratios (PSNRs) for 12 example images and for 5 different blur kernels. In almost all cases, ProxImaL reconstructs a higher-quality image compared to previous work.

hand, the deconvolution problem can additionally benefit from advanced image priors, such as self-similarity. Usually, this would require the entire solver to be rewritten. Using ProxImaL, we can simply add the appropriate proximal operator to the problem formulation. The ProxImaL code for the Poisson Deconvolution application is:

```

x = Variable(300, 300, 3)
data_term = poisson_norm( conv(x, psf) - input )
grad_sparsity = norm1( grad(x) )
objective = data_term + grad_sparsity + nonneg(x)
p = Problem( objective )

```

The method proposed by Krishnan and Fergus [160] is efficient, but it assumes Gaussian noise. It is not easily possible to reformulate this reconstruction method without proximal operators, which is precisely what ProxImaL does. The resulting deconvolved images exhibit significantly less ringing. In Table 10.8, we also show comparisons of peak signal-to-noise-ratios (PSNRs) averaged over 12 images and 5 different kernels. We include an additional comparison to the method proposed by Figueiredo and Bioucas-Dias [307]. For most cases, ProxImaL produces superior results. Note that the conceptual approach of Figueiredo and Jose is the same as ours, but we replace the conjugate gradient updates with the closed-form inverse of the diagonalized matrix. Yet again, exploiting problem-specific structure leads to better and faster solutions.

*Phase Retrieval* All of the applications discussed above use a linear image formation model. Some imaging problems, however, have to deal with nonlinear and non-convex image formations. An example of such a problem is phase retrieval.

As one of the most important problems in electron microscopy, wavefront sensing, astronomy, crystallography, and other scientific imaging areas, it is interesting to evaluate ProxImaL for this challenging problem. The phase retrieval problem is the recovery of a real or complex-valued optical field or image  $\mathbf{x}$  from measurements of its Fourier amplitudes:

$$\text{minimize } f(|\Phi\mathbf{x}| - \mathbf{j}) + r(\mathbf{x}), \quad \text{s.t. } 0 \leq \mathbf{x} \quad (10.14)$$

Here,  $\Phi$  is the Fourier transform,  $f$  is the error metric, and  $r$  is an optional image prior. The most common approach to solving Problem (10.14) is the iterative hybrid input-output (HIO) algorithm [308]. In the spirit of HIO, we can approach this problem by adding a simple total variation image prior as:

$$\text{minimize } \|\Phi\mathbf{x} - \mathbf{j}\|_2^2 + \lambda \|\nabla\mathbf{x}\|_1 + \mathcal{I}_{[0,\infty)}(\mathbf{x}), \quad (10.15)$$

where  $\lambda > 0$ ,  $\|\nabla\cdot\|_1$  is the total variation ( $\ell_1$ -norm of the image gradients), and  $\mathcal{I}_{[0,\infty)}(\cdot)$  is the indicator function that enforces the constraint that  $\mathbf{x} \geq 0$ . We implement the proximal operator of the nonlinear data term as a simple iterative nonlinear gradient descent update that is inspired by HIO. The corresponding ProxImaL code is:

```

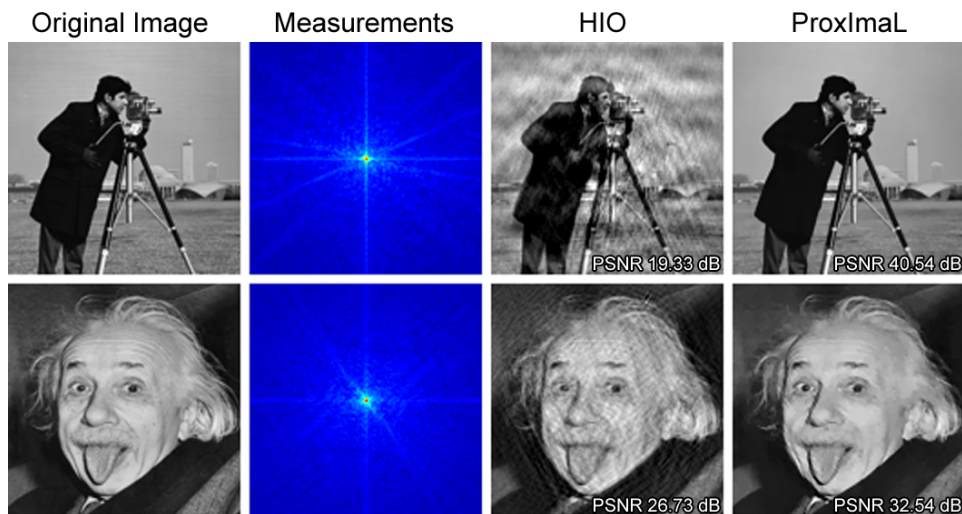
x = Variable(300, 300)
data_term = phase_ret(x, input)
grad_sparsity = norm1( grad(x) )
objective = data_term + patch_similarity + nonneg(x)
p = Problem( objective )

```

We show two phase retrieval results in Figure 10.10. Whereas the HIO solution suffers from strong ringing artifacts, the TV-regularized solution computed by ProxImaL is almost free of artifacts. ProxImaL makes it convenient to evaluate modern image priors developed in the computer vision and computational photography communities for scientific imaging applications.

## 10.9 Discussion

Using a domain-specific language and compiler for image optimization, ProxImaL allows for rapid prototyping of inverse problems in imaging while providing high-



**Figure 10.10:** Phase retrieval. In this nonlinear and nonconvex problem, we measure the Fourier magnitudes of the image (center left column). ProxImaL solves this problem with a nonlinear gradient descent update, which can be combined with any of the image priors previously discussed.

performance execution. The presented mathematical compiler identifies structure in the objective, which can be of the two following types: structure in the unknowns and structure in the objective itself. ProxImaL exploits it in the generation of the solver and also the implementation of the individual sub-problems. It is this structure-driven problem translation that enables the high-performance execution for a wide range of image optimization problems.

However, this strategy may not necessarily apply to other problem domains. For example, Natural Language Processing exhibits structure that differs from imaging problems. Even if the considered problem is an image optimization problem and fits into the generalized objective from Problem (10.2), it may contain non-separable, global penalties that cannot be decomposed into functions with efficient proximal operators, e.g. parameter tuning for large-scale physical simulations. Furthermore, for very simple objectives, such as unconstrained unstructured least-squares problems, ProxImaL does not yield worse solvers than traditional approaches, but also does not improve on them (no structure in the image formation or objective can be

exploited). The biggest limitation of the proposed framework is the selection of objective and algorithm parameters. For the objective parameters, learning-based approaches have been proposed for denoising problems [309], while adaptive schedules have shown to be good recipes for the algorithm parameters [299]. Automatic parameter estimation is an exciting area for future research.

A key extension that we plan to add is fully-automatic robust parameter estimation and problem scaling, beginning with diagonal scaling methods [293, 294]. There are also a number of other directions that are interesting to explore. We plan to extend the set of proximal operators that are currently implemented (e.g., separable quasi-newton methods). We hope to extend both the set of image formation models, and also the list of supported priors, including other patch-based methods and local low-rank models.

Summarizing, ProxImaL allows to efficiently express and solve imaging optimization problems. All of the previous chapter in this dissertation have expressed the respective imaging task as an optimization problem using Bayesian inference. ProxImaL provides an easy way to prototype efficient solver methods for this wide range of applications, exploiting the specific structure of each problem. In essence, it is ProxImaL that makes the principled, optimization-based approach from the previous chapters truly practical and easy to adopt for a new applications.

## Chapter 11

# Conclusion

In this dissertation, we have demonstrated that exploiting structure in the image formation allows us to decode scene information that is commonly considered lost in this process and thus regarded as noise. Specifically, relying on temporal and spatial structure, in both the measurements and the unknowns, we have introduced two completely new image modalities: a new temporal dimension of light as it propagates, and a new per-pixel radial velocity dimension. We have shown that these two novel types of images can make the inversion of light transport feasible in certain scenarios. In particular, we have demonstrated Non-Line-of-Sight imaging and imaging in scattering media, both of which have been traditionally considered extremely hard inverse problems since an immense amount of light transport components are mixed. Furthermore, we have shown that exploiting image structure represents, in fact, a completely new approach to traditional color image processing. We have replaced hand-crafted image processing pipelines with a principled, optimization-based approach. The resulting method has been shown to outperform the state-of-the-art solutions for well-researched image processing problems, such as demosaicking, and we have illustrated that it enables new color camera designs by showing selected examples. Finally, this dissertation has presented a mathematical framework following a Bayesian approach, and a set of tools for the generation of efficient optimization methods. We have shown that, together, both facilitate the development of structure-exploiting methods for all of the previously described imaging problems. In addition, we demonstrate the easy adoption of this approach for solving new imaging problems using a variety of examples.

In the following, we first re-iterate the key findings of each chapter. After that, we highlight promising future directions of research which could build on this thesis.

Chapter 3 demonstrates that a new transient image dimension can be extracted from consumer TOF camera measurements by adding computation in the form of a Bayesian inference. This transient image is the temporal impulse response of the linear light transport in a scene, and hence fully describes it. The temporally convolutional structure of the global light transport make the recovery of high-quality transient images from CIS measurements possible. In contrast to a brute-force direct acquisition using ultra-short pulse lasers, this approach is shown to require several orders of magnitude less capture time and setup cost, is robust to ambient illumination, and allows for significantly increased scene sizes.

Chapter 5 and Chapter 4 have demonstrated that the challenging problem of inverting light transport can be made feasible under certain conditions using this additional temporal dimension. Specifically, Chapter 4 shows that it can enable imaging in scattering and turbid media. Imaging in volumetric scattering is challenging since it mixes a large subset of transport components, leaving only the paths of ballistic photons undisturbed. Our progress in solving this problem was made by exploiting the convolutional structure in the transient images. Chapter 5 considers diffuse reflections, which preserve the temporal dimension but not directionality of the reflected light. We show that, by relying on the additional temporal dimension, we can recover geometry and albedo outside the direct line-of-sight from second-order diffuse reflections, effectively turning walls into mirrors.

Chapter 6 generalizes this approach. While the convolutional structure was previously given by the physical image formation, the proposed method allows us to learn convolutional structure from data directly. Although being a general tool, we demonstrate the approach for efficient feature learning and reconstruction for low level vision tasks.

Chapter 7 demonstrates that a further novel image modality can be extracted from CIS measurements using their convolutional image formation: per-pixel radial velocity. In particular, we have shown that carefully designing illumination and sensor modulation frequencies makes extracting the tiny Doppler shift caused by objects in motion possible. Having just recently demonstrated that this is feasible, we imagine that this new image dimension has many applications for high-level

vision and dynamic reconstruction tasks.

Chapter 8 and Chapter 9 show that exploiting structure in imaging tasks also allows us to rethink camera designs for classical color imaging. First, Chapter 8 considers optical systems for color imaging, which are highly complex systems that eliminate the aberrations of simple optics using a large number of optical elements. Relying on cross-channel structure in the aberrations, we are able to demonstrate high-quality color imaging with a radical change in optical design, using only uncompensated simple optics. Chapter 9 generalizes this co-design of optics and sensing to universal color image processing. While traditional color image processing is organized as a pipeline that causes a cumulative error, we demonstrate that such pipelines can be completely replaced with a single optimization problem using Bayesian inference. This approach represents, in fact, a change of paradigm, outperforming state-of-the-art methods by a large margin, while being extremely flexible and applicable to wide range of classical and novel camera designs, from Bayer image demosaicking to ultra-low-light burst denoising.

Finally, Chapter 10 addresses the computational challenges of the Bayesian, optimization-based approach that we adopted in all of the previous chapters. We introduced a framework consisting of a domain-specific language and compiler that enables the rapid development of efficient solvers for image optimization problems. The compiler automatically identifies and exploits the structure of individual objectives represented in our language. Our framework produces highly-efficient, tailored solver with just a few lines of code for a wide range of imaging problems, the ones presented in Chapters 3–9 being only a small subset.

This dissertation makes first promising steps toward the vision of inverting of light transport under arbitrary conditions. However, to make this problem tractable at all, we have made strong assumptions about the scene in all considered cases, and we only analyzed part of the light transport. In the Non-Line-of-Sight scenario, we assumed only diffuse second-order reflections, and we relied on paths corresponding to ballistic photons for imaging in scattering media. Hence, a promising direction for future research would be to investigate inverse methods that use a more accurate forward model and include more complex light paths. While the resulting optimization problems would become challenging non-convex problems, the methods presented in this work offer good initializations and local convexifications that



can be leveraged. In addition to expanding the forward model, a key argument of this dissertation is to exploit the structure of an inverse problem. In the future, this approach could be drastically extended. Only little of the rich spatio-temporal structure from Section 3.6 and our work [310] has been exploited in the transient model presented in this thesis. In addition to filtering out the direct component, the structure of a specific scenario, e.g., for imaging in scattering media, may be exploited. Over the last few decades, the computer graphics community has become the leading scientific community in simulating light transport with ultra-compact scene representations. In the future, even such representations may be used in reconstruction methods, which illustrates that there is a tremendous amount of structure that is not currently being used.

However, even with improved inverse methods, inverting light transport quality is fundamentally limited by the resolution of the transient measurements, both in the temporal and spatial domain. A big benefit of using TOF depth cameras, as suggested in this thesis, is that they have already become widely adopted consumer products, with emerging applications in gaming, virtual reality and robotics. These growing markets have driven the cost of CIS sensors down and they have increased the resolution and sensor characteristics. While we rely on sensors with only 10 kilopixels in this dissertation, VGA-resolution depth cameras with an order of magnitude more sensels are already commercially available, such as Microsoft's<sup>®</sup> Kinect<sup>®</sup> 2. Once control over the sensor modulation is made available for researchers, drastically increased spatial resolution can be leveraged. However, higher temporal modulation frequencies cannot be immediately expected in consumer products. Due to the CCD charge transfer, at higher modulation frequencies power consumption drastically increases and modulation contrast decreases [27]. While this represents a severe limitation of the CIS sensor, it is important to note that this limitation does not exist in the illumination; commercial laser diodes allow for amplitude modulation in the gigahertz range. An interesting line of research could be the combination of high-frequency modulated illumination with electro-absorption modulators that rely on the Franz-Keldysh effect, which have been used in optical fiber communications at gigahertz frequencies at low voltages [311]. Another promising direction for future research is to rely on emerging SPAD technology and to perform the modulation digitally after the readout. Finally, we relied on amplitude modulation in

this dissertation. Combining this modulation type with interference-based methods, e.g., as shown in [312], could rely on the amplitude modulation for low frequency samples and the carrier for high frequency sampling.

A core concept of this thesis is the Bayesian approach to inverse problems, that has allowed for the statistical modeling of extremely flexible image formations, both for the novel camera designs imagined above as well as classical designs. In particular, the distinction between image formation models, sensor noise characteristics, and the priors allows for the easy reuse of the same structure-exploiting priors or noise models for completely new optical and sensor designs. In the future, this Bayesian approach could be drastically expanded. The MAP estimation that we propose in this thesis only recovers a point-estimate of the posterior distribution. This distribution could be better characterized, which, depending on the distribution, may reveal that other estimates are more robust. Credible intervals quantify the uncertainty of the estimator, which should be computed along with the estimate in the future. Furthermore, the variance of the estimator and the theoretical achievable bounds can provide well-understood tools to compare the quality of different reconstruction techniques. Going beyond classical Bayesian methods, a key area of future research would be hyper-parameters tuning for the likelihood, priors, and optimization algorithms. In order to achieve robust solvers, these parameters should be treated as unknowns and estimated along with the latent image. In fact, without optimal hyper-parameter settings, comparisons between established methods are only somewhat indicative of the achievable performance, and approaches that currently achieve sub-par quality may, in fact, become the state-of-the-art using improved parameters. A promising direction of research on hyper-parameter estimation would be the investigation of empirical Bayes methods, which, however, are computationally expensive. Recent bilevel optimization approaches [309],[313] have presented an efficient approach to parameter estimation by solving a higher-level data fitting problem that takes the result of a specific solver method as an input. Using a large set of training data in this higher-level fitting problem, these methods then effectively learn the parameters from the training data. This approach could also be directly applied to the proximal algorithms from this work [313]. I believe that unrolled, tuned proximal methods are a very promising architecture that will lead to improved inverse methods whose computational costs can be bound, improving on recent

efforts of adapting discriminative convolutional neural networks for reconstruction tasks which fail to provide generative models and are not well understood. However, it might still be fruitful to combine such discriminative models with the proposed proximal methods. Replacing the generative convolutional network in [314] by an unrolled proximal method with convolutional priors may enable learned priors while still allowing for parametrized physical image formation models. The goal of such a combined approach would be to exploit high-level scene classification in the reconstruction process. This could be a feasible step along to full reliance on rendered generative models which we discussed above as a long-term goal.

While computational methods are essential for making advances in imaging, this thesis makes a stronger statement: rather than optimizing optics, sensors and computation separately, they should be jointly optimized. In other words, this dissertation demonstrates that an “optimal camera” should not be defined as a camera that captures every dimension of the plenoptic function at the highest possible resolution, but as an imaging system that recovers latent variables at the highest possible resolution for the lowest total cost, a weighted combination of the footprint, power-consumption, sensor, optical characteristics, capture time, cost, etc. In general, solving this saddle-point problem is extremely challenging since it contains many categorical variables for the different realizable hardware and software components of the camera system, not to mention the large-scale continuous design parameters. However, I believe that this joint optimization can be made feasible by exploiting structure in the design of imaging systems. The imaging systems in Chapters 3, 8 and 9 all rely on structure in the image formation, outperform corresponding “brute force” imaging systems, and have a significantly lower total cost with only slight loss in resolution. Using parametrized architectures, e.g., fixing most of the optical parameters, some sensor characteristics, and algorithm architecture, some of these systems could potentially be found as the solution to a higher-level optimization problem. Exploring this objective, automatically or by hand, could lead to fundamentally new imaging systems. As a local minimum one might imagine application-specific imaging systems using fewer but drastically better pixels in combination with custom optics. Co-designing optics and sensing for a given application has already made a large impact in scientific imaging, specifically in super-resolved fluorescence microscopy, which recently was awarded the Nobel

Prize in Chemistry. Imaging systems optimized for specific tasks in vision, health, virtual reality, gaming and consumer photography could be imagined. Furthermore, an end-to-end optimized camera system would also take the scene into account. We can think of the NLOS imaging in this thesis as an example, where the diffuse reflector acts as a large sensor in the scene. This interpretation could go even further, where every reflection in the scene at every surface essentially represents a sensor. We imagine all of these imaging systems as potential local optima of the higher-level camera design objective. Being able to solve this objective numerically would essentially allow us to evolve cameras computationally, in contrast to the natural evolution of the mammalian vision system. This also means that we will be able to drastically outperform natural vision systems for a wide range of tasks. This thesis has already demonstrated a small set of such tasks.

# Bibliography

- [1] F. Heide, M. B. Hullin, J. Gregson, and W. Heidrich. Low-budget transient imaging using photonic mixer devices. *ACM Transactions on Graphics (TOG)*, 32(4):45, 2013. → pages iii, 4, 8
- [2] M. O’Toole, F. Heide, L. Xiao, M. B. Hullin, W. Heidrich, and K. N. Kutulakos. Temporal Frequency Probing for 5D Transient Analysis of Global Light Transport. *ACM Transactions on Graphics (SIGGRAPH)*, 33(4), 2014. → pages iii, 8, 53, 86
- [3] F. Heide, L. Xiao, W. Heidrich, and M. B. Hullin. Diffuse mirrors: 3D reconstruction from diffuse indirect illumination using inexpensive time-of-flight sensors. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 3222–3229. IEEE, 2014. → pages iv, 4, 8
- [4] F. Heide, L. Xiao, A. Kolb, M. B. Hullin, and W. Heidrich. Imaging in scattering media using correlation image sensors and sparse convolutional coding. *Opt. Express*, 22(21):26338–26350, Oct 2014. → pages iv, 4, 8
- [5] F. Heide, W. Heidrich, and G. Wetzstein. Fast and flexible convolutional sparse coding. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, pages 5135–5143. IEEE, 2015. → pages iv, 4, 9
- [6] F. Heide, W. Heidrich, M. Hullin, and G. Wetzstein. Doppler Time-of-Flight Imaging. *ACM Transactions on Graphics (SIGGRAPH)*, (4), 2015. → pages iv, 4, 9
- [7] S. Shrestha, F. Heide, W. Heidrich, and G. Wetzstein. Computational Imaging with Multi-Camera Time-of-Flight Systems. *ACM Transactions on Graphics (SIGGRAPH)*, 35(4), 2016. → pages v, 9, 163
- [8] F. Heide, M. Rouf, M. B. Hullin, B. Labitzke, W. Heidrich, and A. Kolb. High-quality computational imaging through simple lenses. *ACM Transactions on Graphics (TOG)*, 32(5):149, 2013. → pages v, 4, 9, 124

- [9] F. Heide, Q. Fu, Y. Peng, and W. Heidrich. Encoded diffractive optics for full-spectrum computational imaging. *Scientific Reports*, 6, 2016. → pages v, 9, 183
- [10] F. Heide, M. Steinberger, Y.-T. Tsai, M. Rouf, D. Pająk, D. Reddy, O. Gallo, J. Liu, W. Heidrich, K. Egiazarian, J. Kautz, and K. Pulli. FlexISP: A Flexible Camera Image Processing Framework. *ACM Transactions on Graphics (SIGGRAPH Asia)*, 33(6), 2014. → pages vi, 4, 10, 59, 63, 234
- [11] F. Heide, S. Diamond, M. Nießner, J. Ragan-Kelley, W. Heidrich, and G. Wetzstein. ProxImaL: Efficient Image Optimization using Proximal Algorithms. *ACM Transactions on Graphics (SIGGRAPH)*, 35(4), 2016. → pages vi, 4, 10
- [12] A. D. Milner and M. A. Goodale. The Visual Brain in Action. *PSYCHE*, 4: 12, 1998. → pages 1
- [13] S. Zeki. A vision of the brain. 1993. → pages 1
- [14] D. H. Hubel and T. N. Wiesel. Receptive fields and functional architecture of monkey striate cortex. *The Journal of physiology*, 195(1):215–243, 1968. → pages 1
- [15] B. A. Olshausen and D. J. Field. Sparse coding with an overcomplete basis set: A strategy employed by V1? *Vision research*, 37(23):3311–3325, 1997. → pages 1
- [16] L. Tian, X. Li, K. Ramchandran, and L. Waller. Multiplexed coded illumination for Fourier Ptychography with an LED array microscope. *Biomedical optics express*, 5(7):2376–2389, 2014. → pages 1
- [17] E. Reinhard, W. Heidrich, P. Debevec, S. Pattanaik, G. Ward, and K. Myszkowski. *High dynamic range imaging: acquisition, display, and image-based lighting*. Morgan Kaufmann, 2010. → pages 1, 38, 39
- [18] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. → pages 1, 95
- [19] A. Kolb, E. Barth, and R. Koch. ToF-sensors: New dimensions for realism and interactivity. In *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW'08. IEEE Computer Society Conference on*, pages 1–6. IEEE, 2008. → pages 1

- [20] S. R. P. Pavani, M. A. Thompson, J. S. Biteen, S. J. Lord, N. Liu, R. J. Twieg, R. Piestun, and W. Moerner. Three-dimensional, single-molecule fluorescence imaging beyond the diffraction limit by using a double-helix point spread function. *Proceedings of the National Academy of Sciences*, 106(9):2995–2999, 2009. → pages 1, 2, 182
- [21] V. F. Pamplona, A. Mohan, M. M. Oliveira, and R. Raskar. NETRA: interactive display for estimating refractive errors and focal range. In *ACM Transactions on Graphics (TOG)*, volume 29, page 77. ACM, 2010. → pages 1
- [22] C. F. Gauss. *Dioptrische Untersuchungen von CF Gauss*. in der Dieterichschen Buchhandlung, 1843. → pages 2, 23
- [23] L. Seidel. Über die Theorie der Fehler, mit welchen die durch optische Instrumente gesehenen Bilder behaftet sind, und über die mathematischen Bedingungen ihrer Aufhebung. *Abhandlungen der Naturwissenschaftlich-Technischen Commission bei der Königl. Bayerischen Akademie der Wissenschaften in München*. Cotta, 2:4, 1857. → pages 2, 24
- [24] A. Levin, R. Fergus, F. Durand, and W. T. Freeman. Image and Depth from a Conventional Camera with a Coded Aperture. *ACM Trans. Graph.*, 26(3), July 2007. ISSN 0730-0301. → pages 2, 41
- [25] J. A. Thomas, C. F. Moss, and M. Vater. *Echolocation in bats and dolphins*. University of Chicago Press, 2004. → pages 2
- [26] C. Weitkamp. *Lidar: range-resolved optical remote sensing of the atmosphere*, volume 102. Springer Science & Business Media, 2005. → pages 3, 42, 43
- [27] R. Lange. 3D time-of-flight distance measurement with custom solid-state image sensors in CMOS/CCD-technology. *Diss., Department of Electrical Engineering and Computer Science, University of Siegen*, 2000. → pages 3, 31, 32, 33, 42, 43, 44, 45, 54, 250
- [28] A. Payne, A. Daniel, A. Mehta, B. Thompson, C. S. Bamji, D. Snow, H. Oshima, L. Prather, M. Fenton, L. Kordus, et al. 7.6 A 512 × 424 CMOS 3D Time-of-Flight image sensor with multi-frequency photo-demodulation up to 130MHz and 2GS/s ADC. In *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2014 IEEE International*, pages 134–135. IEEE, 2014. → pages 3

- [29] A. Kolb, E. Barth, R. Koch, and R. Larsen. Time-of-flight sensors in computer graphics. In *Proc. Eurographics (State-of-the-Art Report)*, volume 6, 2009. → pages 3, 41, 48
- [30] E. Hecht. Optics, 4th. *International edition, Addison-Wesley, San Francisco*, 3, 2002. → pages 12, 13, 14, 15, 17, 20, 21, 22, 23, 25, 27
- [31] R. P. Feynman, R. B. Leighton, and M. Sands. *The Feynman Lectures on Physics, Desktop Edition Volume I*, volume 1. Basic books, 2013. → pages 12, 15, 17
- [32] J. C. Maxwell. *A treatise on electricity and magnetism*, volume 1. Clarendon press, 1881. → pages 12
- [33] A. Einstein. Über einen die Erzeugung und Verwandlung des Lichtes betreffenden heuristischen Gesichtspunkt. *Annalen der Physik*, 322(6): 132–148, 1905. → pages 13, 28
- [34] R. P. Feynman and A. Zee. *QED: The strange theory of light and matter*. Princeton University Press, 2006. → pages 13, 17
- [35] J. W. Strutt. LVIII. On the scattering of light by small particles. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 41 (275):447–454, 1871. → pages 14
- [36] C. Huygens. *Traite de la lumière*, 1678. → pages 17
- [37] M. Pharr and G. Humphreys. *Physically based rendering: From theory to implementation*. Morgan Kaufmann, 2004. → pages 17, 19
- [38] P. Dutre, P. Bekaert, and K. Bala. *Advanced global illumination*. CRC Press, 2006. → pages 17, 18
- [39] J. T. Kajiya. The rendering equation. In *ACM Siggraph Computer Graphics*, volume 20, pages 143–150. ACM, 1986. → pages 18
- [40] J. M. Palmer and B. G. Grant. *The art of radiometry*. SPIE Press Bellingham, 2010. → pages 18
- [41] E. Veach and L. J. Guibas. Metropolis light transport. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 65–76. ACM Press/Addison-Wesley Publishing Co., 1997. → pages 19



- [42] T. Akenine-Möller, E. Haines, and N. Hoffman. *Real-time rendering*. CRC Press, 2008. → pages 19
- [43] T. Young. The Bakerian lecture: Experiments and calculations relative to physical optics. *Philosophical transactions of the Royal Society of London*, 94:1–16, 1804. → pages 21
- [44] E. Optics. Optics and Optical Instruments Catalogue. → pages 23
- [45] M. Hirsch, S. Sra, B. Schlkopf, and S. Harmeling. Efficient filter flow for space-variant multiframe blind deconvolution. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 607–614, June 2010. → pages 28
- [46] H. Hertz. Ueber einen Einfluss des ultravioletten Lichtes auf die electrische Entladung. *Annalen der Physik*, 267(8):983–1000, 1887. → pages 28
- [47] S. M. Sze. *Semiconductor devices: physics and technology*. John Wiley & Sons, 2008. → pages 29, 30, 31, 33
- [48] W. S. Boyle and G. E. Smith. Charge coupled semiconductor devices. *Bell System Technical Journal*, 49(4):587–593, 1970. → pages 32
- [49] A. J. Theuwissen. *Solid-state imaging with charge-coupled devices*, volume 1. Springer Science & Business Media, 2006. → pages 32
- [50] G. Köklü. Design and Implementation of CMOS Image Sensors for Biomedical Applications. 2014. → pages 33
- [51] E. M. V. Association. EMVA Standard 1288 - Standard for Characterization of Image Sensors and Cameras, Release 3.0. [www.standard1288.org](http://www.standard1288.org) (accessed on January 1st, 2016). → pages 33, 34, 35, 36
- [52] M. Schmidt. Analysis, modeling and dynamic optimization of 3d time-of-flight imaging systems. 2011. → pages 34, 51, 52, 54, 89
- [53] B. Jähne. Digital Image Processing. 2005. → pages 36
- [54] J. R. Janesick. *Photon transfer*. SPIE press San Jose, 2007. → pages 36
- [55] J. R. Janesick. CCD characterization using the photon transfer technique. In *Proc. SPIE*, volume 570, pages 7–19, 1985. → pages 36
- [56] EMVA Standard 1288 - Reference Implementation. <https://github.com/EMVA1288/emva1288> (accessed on January 1st, 2016). → pages 36

- [57] A. Foi, M. Trimeche, V. Katkovnik, and K. Egiazarian. Practical Poissonian-Gaussian noise modeling and fitting for single-image raw-data. *Image Processing, IEEE Transactions on*, 17(10):1737–1754, 2008. → pages 37
- [58] Z. Liu, L. Yuan, X. Tang, M. Uyttendaele, and J. Sun. Fast burst images denoising. *ACM Transactions on Graphics (TOG)*, 33(6):232, 2014. → pages 38
- [59] T. Buades, Y. Lou, J. Morel, and Z. Tang. A note on multi-image denoising. In *Local and Non-Local Approximation in Image Processing, 2009. LNLA 2009. International Workshop on*, pages 1–15. IEEE, 2009. → pages 38
- [60] M. Tico and K. Pulli. Image Enhancement Method via Blur and Noisy Image Fusion. In *ICIP*, 2009. → pages 38
- [61] M. Irani and S. Peleg. Improving resolution by image registration. *Graphical Models and Image Processing*, 53(3), 1991. → pages 38
- [62] B. E. Bayer. Color imaging array, 1976. US Patent 3,971,065. → pages 39
- [63] R. Ramanath, W. E. Snyder, Y. Yoo, and M. S. Drew. Color image processing pipeline in digital still cameras. *IEEE Signal Processing Magazine*, 22(1), 2005. → pages 39, 186
- [64] L. Zhang, X. Wu, A. Buades, and X. Li. Color demosaicking by local directional interpolation and nonlocal adaptive thresholding. *Journal of Electronic Imaging*, 20(2), 2011. → pages 39, 196, 202
- [65] L. Shao, R. Yan, X. Li, and Y. Liu. From Heuristic Optimization to Dictionary Learning: A Review and Comprehensive Comparison of Image Denoising Algorithms. *IEEE Transactions on Cybernetics*, (99), 2013. ISSN 2168-2267. → pages 39
- [66] A. Adams, E.-V. Talvala, S. H. Park, D. Jacobs, B. Ajdin, N. Gelfand, J. Dolson, D. Vaquero, J. Baek, M. Tico, H. P. A. Lensch, W. Matusik, K. Pulli, M. Horowitz, and M. Levoy. The Frankencamera: an experimental platform for computational photography. *ACM TOG*, 29(4), 2010. → pages 39
- [67] A. Buades, B. Coll, and J. M. Morel. A non-local algorithm for image denoising. In *CVPR*, 2005. → pages 39, 202, 313

- [68] R. R. Coifman and D. L. Donoho. Translation-invariant denoising. Technical Report 475, Stanford University, 1995. → pages 39
- [69] K. O. Egiazarian, J. Astola, M. Helsingius, and P. Kuosmanen. Adaptive denoising and lossy compression of images in transform domain. *J. Electronic Imaging*, 8(3), 1999. → pages 39, 313
- [70] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image Denoising by Sparse 3-D Transform-Domain Collaborative Filtering. *IEEE TIP*, 16(8), 2007. → pages 39, 191, 313
- [71] I. Mosseri, M. Zontak, and M. Irani. Combining the power of Internal and External denoising. In *ICCP*, 2013. → pages 39, 200
- [72] J. Tanida, T. Kumagai, K. Yamada, S. Miyatake, K. Ishida, T. Morimoto, N. Kondou, D. Miyazaki, and Y. Ichioka. Thin observation module by bound optics (TOMBO): concept and experimental verification. *Applied Optics*, 40(11), 2001. → pages 40
- [73] K. Venkataraman, D. Lelescu, J. Duparré, A. McMahan, G. Molina, P. Chatterjee, R. Mullis, and S. Nayar. PiCam: an ultra-thin high performance monolithic camera array. *ACM TOG*, 32(6), 2013. → pages 40, 315, 319
- [74] M. Levoy and P. Hanrahan. Light field rendering. In *SIGGRAPH*, 1996. → pages 40
- [75] B. Wilburn, N. Joshi, V. Vaish, E.-V. Talvala, E. Antunez, A. Barth, A. Adams, M. Horowitz, and M. Levoy. High performance imaging using large camera arrays. *ACM TOG*, 24(3), 2005. → pages 40
- [76] S. Nayar and V. Branzoi. Adaptive Dynamic Range Imaging: Optical Control of Pixel Exposures over Space and Time. In *ICCV*, Oct 2003. → pages 40
- [77] J. Gu, Y. Hitomi, T. Mitsunaga, and S. Nayar. Coded Rolling Shutter Photography: Flexible Space-Time Sampling. In *ICCP*, 2010. → pages 40, 200, 201
- [78] S. Hajisharif, J. Kronander, and J. Unger. HDR reconstruction for alternating gain (ISO) sensor readout. In *Proc. of Eurographics (short paper)*, 2014. → pages 40, 200

- [79] E. Stoykova, A. A. Alatan, P. Benzie, N. Grammalidis, S. Malassiotis, J. Ostermann, S. Piekh, V. Sainov, C. Theobalt, T. Thevar, et al. 3-D time-varying scene capture technologies A survey. *Circuits and Systems for Video Technology, IEEE Transactions on*, 17(11):1568–1586, 2007. → pages 40, 41
- [80] B. K. Horn. Obtaining shape from shading information. In *Shape from shading*, pages 123–171. MIT press, 1989. → pages 41
- [81] R. Zhang, P.-S. Tsai, J. E. Cryer, and M. Shah. Shape-from-shading: a survey. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 21(8):690–706, 1999. → pages 41
- [82] P. Favaro and S. Soatto. A geometric approach to shape from defocus. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(3):406–417, 2005. → pages 41
- [83] J. Aloimonos. Shape from texture. *Biological cybernetics*, 58(5):345–360, 1988. → pages 41
- [84] F. Remondino and D. Stoppa. *ToF range-imaging cameras*, volume 68121. Springer, 2013. → pages 41, 43, 44, 46
- [85] S. Foix, G. Alenya, and C. Torras. Lock-in time-of-flight (ToF) cameras: a survey. *Sensors Journal, IEEE*, 11(9):1917–1926, 2011. → pages 41, 43, 44, 45, 46, 47, 51, 52, 54, 71
- [86] G. J. Iddan and G. Yahav. Three-dimensional imaging in the studio and elsewhere. In *Photonics West 2001-Electronic Imaging*, pages 48–55. International Society for Optics and Photonics, 2001. → pages 41
- [87] B. Girod and S. Scherock. Depth from defocus of structured light. In *1989 Advances in Intelligent Robotics Systems Conference*, pages 209–215. International Society for Optics and Photonics, 1990. → pages 41
- [88] S. Fanello, C. Keskin, S. Izadi, P. Kohli, D. Kim, D. Sweeney, A. Criminisi, J. Shotton, S. B. Kang, and T. Paek. Learning to be a Depth Camera for Close-Range Human Capture and Interaction. In *Journal ACM Transactions on Graphics (TOG)*. ACM Association for Computing Machinery, July 2014. → pages 41
- [89] M. C. Teich and B. Saleh. Fundamentals of photonics. *Canada, Wiley Interscience*, page 3, 1991. → pages 41

- [90] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003. → pages 41
- [91] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International journal of computer vision*, 47(1-3):7–42, 2002. → pages 41
- [92] M. MIMOU, T. Kanade, and T. SAKAI. A method of time-coded parallel planes of light for depth measurement. *IEICE TRANSACTIONS (1976-1990)*, 64(8):521–528, 1981. → pages 41
- [93] J. Salvi, S. Fernandez, T. Pribanic, and X. Llado. A state of the art in structured light patterns for surface profilometry. *Pattern recognition*, 43(8): 2666–2680, 2010. → pages 41
- [94] J. Batlle, E. Mouaddib, and J. Salvi. Recent progress in coded structured light as a technique to solve the correspondence problem: a survey. *Pattern recognition*, 31(7):963–982, 1998. → pages 41
- [95] M. O’Toole, J. Mather, and K. N. Kutulakos. 3d shape and indirect appearance by structured light transport. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 3246–3253. IEEE, 2014. → pages 41
- [96] C. Beder, B. Bartczak, and R. Koch. A comparison of PMD-cameras and stereo-vision for the task of surface reconstruction using patchlets. In *Computer Vision and Pattern Recognition, 2007. CVPR’07. IEEE Conference on*, pages 1–8. IEEE, 2007. → pages 42
- [97] B. Langmann, K. Hartmann, and O. Loffeld. Depth Camera Technology Comparison and Performance Evaluation. In *ICPRAM (2)*, pages 438–444, 2012. → pages 42
- [98] D. Gallup, J.-M. Frahm, P. Mordohai, and M. Pollefeys. Variable baseline/resolution stereo. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008. → pages 42
- [99] M.-C. Amann, T. Bosch, M. Lescure, R. Myllyla, and M. Rioux. Laser ranging: a critical review of usual techniques for distance measurement. *Optical Engineering*, 40(1):10–19, 2001. → pages 42, 43

- [100] W. Koechner. Optical Ranging System Employing a High Power Injection Laser Diode. *Aerospace and Electronic Systems, IEEE Transactions on*, AES-4(1):81–91, Jan 1968. ISSN 0018-9251. → pages 42
- [101] N. Csanyi and C. K. Toth. LiDAR Data Accuracy: The Impact of Pulse Repetition Rate. In *ASPRS/MAPPS Conference*, November 2006. → pages 43
- [102] E. Charbon, M. Fishburn, R. Walker, R. Henderson, and C. Niclass. SPAD-Based Sensors. In F. Remondino and D. Stoppa, editors, *TOF Range-Imaging Cameras*, pages 11–38. Springer Berlin Heidelberg, 2013. ISBN 978-3-642-27522-7. → pages 43
- [103] G. Beheim and K. Fritsch. Range finding using frequency-modulated laser diode. *Appl. Opt.*, 25(9):1439–1442, May 1986. → pages 43
- [104] Z. Xu. *Investigation of 3D-imaging Systems Based on Modulated Light and Optical RF-interferometry (ORFI) (ZESS-forschungsberichte)*. Shaker Verlag GmbH, Germany, 1999. ISBN 3826567366. → pages 44
- [105] C. Baetzel, J. K. Schaller, and C. G. Stojanoff. Fiber optical multi-channel laser radar with parallel processing. *Proc. SPIE*, 3100:366–374, 1997. → pages 44
- [106] J. Li, O. Loffeld, R. Schwarte, B. Buxbaum, and Z. Xu. Optimization of laser ranging system in the case of colored noise. *Proc. SPIE*, 2904:108–116, 1996. → pages 44
- [107] R. Z. Whyte, A. D. Payne, A. A. Dorrington, and M. J. Cree. Multiple range imaging camera operation with minimal performance impact. *Proc. SPIE*, 7538:75380I–75380I–10, 2010. → pages 44
- [108] H. Povel, H. Aebersold, and J. O. Stenflo. Charge-coupled device image sensor as a demodulator in a 2-D polarimeter with a piezoelastic modulator. *Appl. Opt.*, 29(8):1186–1190, Mar 1990. → pages 45
- [109] T. Spirig, P. Seitz, O. Vietze, and F. Heitger. The lock-in CCD-two-dimensional synchronous detection of light. *Quantum Electronics, IEEE Journal of*, 31(9):1705–1708, Sep 1995. ISSN 0018-9197. → pages 45
- [110] R. Schwarte, H.-G. Heinol, Z. Xu, and K. Hartmann. New active 3D vision system based on rf-modulation interferometry of incoherent light. In *Photonics East'95*, pages 126–134. International Society for Optics and Photonics, 1995. → pages 45

- [111] R. Schwarte, Z. Xu, H.-G. Heinol, J. Olk, R. Klein, B. Buxbaum, H. Fischer, and J. Schulte. New electro-optical mixing and correlating sensor: facilities and applications of the photonic mixer device (PMD). In *Lasers and Optics in Manufacturing III*, pages 245–253. International Society for Optics and Photonics, 1997. → pages 45
- [112] C. Niclass, C. Favi, T. Kluter, F. Monnier, and E. Charbon. Single-photon synchronous detection. *Solid-State Circuits, IEEE Journal of*, 44(7): 1977–1989, 2009. → pages 46
- [113] R. M. Conroy, A. A. Dorrington, R. Künemeyer, and M. J. Cree. Range imager performance comparison in homodyne and heterodyne operating modes. In *IS&T/SPIE Electronic Imaging*, pages 723905–723905. International Society for Optics and Photonics, 2009. → pages 48
- [114] A. A. Dorrington, M. J. Cree, A. D. Payne, R. M. Conroy, and D. A. Carnegie. Achieving sub-millimetre precision with a solid-state full-field heterodyning range imaging camera. *Measurement Science and Technology*, 18(9):2809, 2007. → pages 48, 147
- [115] M. Lindner and A. Kolb. Lateral and depth calibration of pmd-distance sensors. In *Advances in Visual Computing*, pages 524–533. Springer, 2006. → pages 51, 149
- [116] M. Lindner and A. Kolb. Calibration of the intensity-related distance error of the PMD ToF-camera. In *Proc. SPIE, Intelligent Robots and Computer Vision*, volume 6764, page 67640W, 2007. → pages 51
- [117] T. Khalmann, H. Ingensand, and F. Remondino. Calibration for increased accuracy of the range imaging camera Swissranger<sup>TM</sup>. *ISPRS Archives*, 36 (Part 5):136–141. → pages 51
- [118] M. Schmidt and B. Jähne. A physical model of time-of-flight 3D imaging systems, including suppression of ambient light. In *Dynamic 3D Imaging*, pages 1–15. Springer, 2009. → pages 51
- [119] A. D. Payne, A. A. Dorrington, M. J. Cree, and D. A. Carnegie. Improved measurement linearity and precision for AMCW time-of-flight range imaging cameras. *Applied optics*, 49(23):4392–4403, 2010. → pages 52
- [120] S. Fuchs. Multipath interference compensation in time-of-flight camera images. In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pages 3583–3586. IEEE, 2010. → pages 53

- [121] S. Fuchs, M. Suppa, and O. Hellwich. Compensation for multipath in ToF camera measurements supported by photometric calibration and environment integration. In *Computer Vision Systems*, pages 31–41. Springer, 2013. → pages 53
- [122] D. Jiménez, D. Pizarro, M. Mazo, and S. Palazuelos. Modeling and correction of multipath interference in time of flight cameras. *Image and Vision Computing*, 32(1):1–13, 2014. → pages 53
- [123] J. P. Godbaz, M. J. Cree, and A. A. Dorrington. Closed-form inverses for the mixed pixel/multipath interference problem in amcw lidar. In *IS&T/SPIE Electronic Imaging*, pages 829618–829618. International Society for Optics and Photonics, 2012. → pages 53
- [124] A. A. Dorrington, J. P. Godbaz, M. J. Cree, A. D. Payne, and L. V. Streeter. Separating true range measurements from multi-path and scattering interference in commercial range cameras. In *IS&T/SPIE Electronic Imaging*, pages 786404–786404. International Society for Optics and Photonics, 2011. → pages 53
- [125] A. Kirmani, A. Benedetti, and P. A. Chou. Spumic: Simultaneous phase unwrapping and multipath interference cancellation in time-of-flight cameras using spectral methods. In *Multimedia and Expo (ICME), 2013 IEEE International Conference on*, pages 1–6. IEEE, 2013. → pages 53
- [126] D. Freedman, Y. Smolin, E. Krupka, I. Leichter, and M. Schmidt. SRA: Fast removal of general multipath for tof sensors. In *Computer Vision–ECCV 2014*, pages 234–249. Springer, 2014. → pages 53, 58, 90, 97
- [127] D. Falie and V. Buzuloiu. Further investigations on ToF cameras distance errors and their corrections. In *Circuits and Systems for Communications, 2008. ECCSC 2008. 4th European Conference on*, pages 197–200, July 2008. → pages 53
- [128] R. Kaufmann, M. Lehmann, M. Schweizer, M. Richter, P. Metzler, G. Lang, T. Oggier, N. Blanc, P. Seitz, G. Gruener, et al. A time-of-flight line sensor: development and application. In *Photonics Europe*, pages 192–199. International Society for Optics and Photonics, 2004. → pages 54
- [129] J.-M. Gottfried, R. Nair, S. Meister, C. S. Garbe, and D. Kondermann. Time of flight motion compensation revisited. In *Image Processing (ICIP), 2014 IEEE International Conference on*, pages 5861–5865. IEEE, 2014. → pages 54



- [130] M. Lindner and A. Kolb. Compensation of motion artifacts for time-of-flight cameras. In *Dynamic 3D Imaging*, pages 16–27. Springer, 2009. → pages 54
- [131] D. Lefloch, T. Hoegg, and A. Kolb. Real-time motion artifacts compensation of tof sensors data on gpu. In *SPIE Defense, Security, and Sensing*, pages 87380U–87380U. International Society for Optics and Photonics, 2013. → pages 54
- [132] C. J. Doppler. Über das farbige Licht der Doppelsterne und einiger anderer Gestirne des Himmels. *Abhandlungen der Königl. Böhm. Gesellschaft der Wissenschaften*, 12(2):465–482, 1842. → pages 55
- [133] T. Guillot, A. Burrows, W. Hubbard, J. Lunine, and D. Saumon. Giant planets at small orbital distances. *The Astrophysical Journal Letters*, 459(1): L35, 1996. → pages 55
- [134] S. Weinberg. *Cosmology*. Oxford Univ. Press, 2008. → pages 55
- [135] B. G. Lipták and B. G. Lipták. *Process measurement and analysis*. Butterworth Heinemann, 1995. → pages 55
- [136] H.-E. Albrecht. *Laser Doppler and phase Doppler measurement techniques*. Springer Science & Business Media, 2003. → pages 55
- [137] F. J. Snow. Laser optical mouse, Dec. 28 1993. US Patent 5,274,361. → pages 55
- [138] H. Meikle. *Modern radar systems*. Artech House, 2008. → pages 55
- [139] W. H. Long, D. H. Mooney, and W. A. Skillman. Pulse doppler radar. *Radar Handbook*, 2, 1990. → pages 55
- [140] R. J. Doviak and D. S. Zrníc. *Doppler Radar & Weather Observations*. Academic press, 2014. → pages 56
- [141] R. Frehlich, S. M. Hannon, and S. W. Henderson. Performance of a 2-um Coherent Doppler Lidar for Wind Measurements. *Journal of Atmospheric And Oceanic Technology*, 11(6):1517–1528, 1994. → pages 56
- [142] N. Abramson. Light-in-flight recording by holography. *Optics Letters*, 3(4): 121–123, 1978. → pages 56
- [143] N. Abramson. Light-in-flight recording: high-speed holographic motion pictures of ultrafast phenomena. *Applied optics*, 22(2):215–232, 1983. → pages 56

- [144] B. Nilsson and T. E. Carlsson. Direct three-dimensional shape measurement by digital light-in-flight holography. *Applied Optics*, 37(34):7954–7959, 1998. → pages 57
- [145] A. Velten, R. Raskar, and M. Bawendi. Picosecond Camera for Time-of-Flight Imaging. In *OSA Imaging Systems and Applications*, 2011. → pages 57
- [146] A. Smith, J. Skorupski, and J. Davis. Transient Rendering. Technical Report UCSC-SOE-08-26, University of California Santa Cruz, School of Engineering, 2008. → pages 57
- [147] A. Kirmani, T. Hutchison, J. Davis, and R. Raskar. Looking around the corner using transient imaging. In *Proc. ICCV*, pages 159–166, 2009. → pages 57, 88
- [148] R. Pandharkar, A. Velten, A. Bardagjy, E. Lawson, M. Bawendi, and R. Raskar. Estimating motion and size of moving non-line-of-sight objects in cluttered environments. In *Proc. CVPR*, pages 265–272, 2011. → pages 57
- [149] A. Velten, T. Willwacher, O. Gupta, A. Veeraraghavan, M. Bawendi, and R. Raskar. Recovering three-dimensional shape around a corner using ultrafast time-of-flight imaging. *Nature Communications*, 3:745, 2012. → pages 57, 88, 104, 107, 108, 117, 118, 294
- [150] N. Naik, S. Zhao, A. Velten, R. Raskar, and K. Bala. Single view reflectance capture using multiplexed scattering and time-of-flight imaging. *ACM Trans. Graph.*, 30(6):171, 2011. → pages 57
- [151] D. Wu, G. Wetzstein, C. Barsi, T. Willwacher, Q. Dai, and R. Raskar. Ultra-fast lensless computational imaging through 5D frequency analysis of time-resolved light transport. *International Journal of Computer Vision*, 110(2):128–140, 2014. → pages 57
- [152] A. Velten, D. Wu, A. Jarabo, B. Masia, C. Barsi, C. Joshi, E. Lawson, M. Bawendi, D. Gutierrez, and R. Raskar. Femto-Photography: Capturing and Visualizing the Propagation of Light. *ACM Trans. Graph.*, 32, 2013. → pages 57, 73, 74, 79, 87
- [153] D. Wu, M. O’Toole, A. Velten, A. Agrawal, and R. Raskar. Decomposing global light transport using time of flight imaging. In *Proc. CVPR*, pages 366–373, 2012. → pages 57

- [154] J. Lin, Y. Liu, M. B. Hullin, and Q. Dai. Fourier analysis on transient imaging with a multifrequency time-of-flight camera. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 3230–3237. IEEE, 2014. → pages 58
- [155] H. Qiao, J. Lin, Y. Liu, M. B. Hullin, and Q. Dai. Resolving transient time profile in ToF imaging via log-sum sparse regularization. *Opt. Lett.*, 40(6): 918–921, Mar 2015. → pages 58
- [156] M. Gupta, S. K. Nayar, M. Hullin, and J. Martin. Phasor Imaging: A Generalization Of Correlation-Based Time-of-Flight Imaging. *ACM Transactions on Graphics (TOG)*, page 18, Feb 2015. → pages 58, 89
- [157] A. Kadambi, A. Bhandari, R. Whyte, A. Dorrington, and R. Raskar. Demultiplexing illumination via low cost sensing and nanosecond coding. In *Computational Photography (ICCP), 2014 IEEE International Conference on*, pages 1–10. IEEE, 2014. → pages 58
- [158] A. Jarabo, B. Masia, A. Velten, C. Barsi, R. Raskar, and D. Gutierrez. Rendering relativistic effects in transient imaging. *Proceedings of CEIG*, 2013. → pages 58
- [159] K. P. Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012. → pages 61
- [160] D. Krishnan and R. Fergus. Fast Image Deconvolution using Hyper-Laplacian Priors. In *NIPS*, 2009. → pages 63, 171, 198, 199, 200, 232, 233, 234, 237, 242, 243, 309
- [161] A. Levin, R. Fergus, F. Durand, and W. T. Freeman. Deconvolution using natural image priors. *ACM Transactions on Graphics (TOG)*, 26(3), 2007. → pages 63, 173, 198, 199, 200, 307, 308
- [162] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester. Image inpainting. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 417–424. ACM Press/Addison-Wesley Publishing Co., 2000. → pages 63
- [163] D. Zoran and Y. Weiss. From learning models of natural image patches to whole image restoration. In *Proc. IEEE ICCV*, pages 479–486, 2011. → pages 63, 120, 197, 200, 311

- [164] T. F. Chan, G. H. Golub, and P. Mulet. A nonlinear primal-dual method for total variation-based image restoration. *SIAM Journal on Scientific Computing*, 20(6), 1999. → pages 63
- [165] C. J. Schuler, M. Hirsch, S. Harmeling, and B. Schölkopf. Non-stationary correction of optical aberrations. In *ICCV*, 2011. → pages 63, 169, 193, 303, 308, 309, 310
- [166] A. Danielyan, V. Katkovnik, and K. Egiazarian. BM3D frames and variational image deblurring. *IEEE TIP*, 21(4), 2012. → pages 63, 198, 199, 200, 210, 214
- [167] U. Schmidt and S. Roth. Shrinkage fields for effective image restoration. In *Proc. IEEE CVPR*, pages 2774–2781, 2014. → pages 63
- [168] R. Fattal, D. Lischinski, and M. Werman. Gradient domain high dynamic range compression. In *ACM Trans. Graph.*, volume 21, pages 249–256. ACM, 2002. → pages 63
- [169] A. Levin, A. Zomet, S. Peleg, and Y. Weiss. Seamless image stitching in the gradient domain. In *Proc. ECCV*, pages 377–389. 2004. → pages 63
- [170] A. Levin, D. Lischinski, and Y. Weiss. Colorization using optimization. In *ACM Trans. Graph.*, volume 23, pages 689–694, 2004. → pages 63
- [171] D. Krishnan and R. Szeliski. Multigrid and multilevel preconditioners for computational photography. *ACM Trans. Graph.*, 30(6):177, 2011. → pages 63
- [172] E. Y. Sidky and X. Pan. Image reconstruction in circular cone-beam computed tomography by constrained, total-variation minimization. *Physics in medicine and biology*, 53(17):4777, 2008. → pages 63
- [173] L. Tian and L. Waller. 3D intensity and phase imaging from light field measurements in an LED array microscope. *Optica*, 2(2):104–111, 2015. → pages 63
- [174] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011. → pages 64, 72, 121, 230, 233
- [175] N. Parikh and S. Boyd. Proximal algorithms. *Foundations and Trends in Optimization*, 1(3):123–231, 2013. → pages 65, 75, 110, 111, 121, 126, 192, 213

- [176] R. Rockafellar. Augmented Lagrangians and Applications of the Proximal Point Algorithm in Convex Programming. *Mathematics of Operations Research*, 1(2):97–116, 1976. → pages 65
- [177] R. Bruck. An iterative solution of a variational inequality for certain monotone operators in Hilbert space. *Bulletin of the American Mathematical Society*, 81(5):890–892, Sept. 1975. → pages 65
- [178] A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40(1):120–145, 2011. → pages 65, 72, 75, 175, 190, 192, 209, 225, 227, 232
- [179] T. Pock, D. Cremers, H. Bischof, and A. Chambolle. An algorithm for minimizing the Mumford-Shah functional. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1133–1140, 2009. → pages 65
- [180] T. Goldstein and S. Osher. The Split Bregman Method for  $\ell_1$ -Regularized Problems. *SIAM Journal on Imaging Sciences*, 2(2):323–343, 2009. → pages 65
- [181] A. Beck and M. Teboulle. A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009. → pages 65
- [182] D. Geman and C. Yang. Nonlinear image recovery with half-quadratic regularization. *IEEE Trans. Image Processing*, 4(7):932–946, 1995. → pages 65
- [183] P. Ochs, Y. Chen, T. Brox, and T. Pock. iPiano: Inertial proximal algorithm for nonconvex optimization. *SIAM Journal on Imaging Sciences*, 7(2):1388–1419, 2014. → pages 65, 214
- [184] H. Attouch, J. Bolte, and B. F. Svaiter. Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forward–backward splitting, and regularized Gauss–Seidel methods. *Mathematical Programming*, 137(1):91–129, 2011. → pages 65, 233
- [185] T. Möllenhoff, E. Strelakovski, M. Moeller, and D. Cremers. The Primal-Dual Hybrid Gradient Method for Semiconvex Splittings. *SIAM Journal on Imaging Sciences*, 8(2):827–857, 2015. → pages

- [186] G. Li and T. K. Pong. Global convergence of splitting methods for nonconvex composite optimization. *arXiv e-Print 1407.0753*, 2015. → pages 65, 233
- [187] P. J. Huber et al. Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, 35(1):73–101, 1964. → pages 72
- [188] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1):259–268, 1992. → pages 72, 190, 311
- [189] D. Wu, A. Velten, M. O’Toole, B. Masia, A. Agrawal, Q. Dai, and R. Raskar. Decomposing global light transport using time of flight imaging. *International journal of computer vision*, 107(2):123–138, 2014. → pages 73
- [190] M. O’Toole, R. Raskar, and K. N. Kutulakos. Primal-dual coding to probe light transport. 2012. → pages 83, 85, 86
- [191] P. Sen, B. Chen, G. Garg, S. Marschner, M. Horowitz, M. Levoy, and H. P. A. Lensch. Dual photography. *ACM SIGGRAPH*, pages 745–755, 2005. → pages
- [192] P. Debevec, T. Hawkins, C. Tchou, H.-P. Duiker, W. Sarokin, and M. Sagar. Acquiring the reflectance field of a human face. *ACM SIGGRAPH*, pages 145–156, 2000. → pages 83
- [193] R. Ng, R. Ramamoorthi, and P. Hanrahan. All-frequency shadows using non-linear wavelet lighting approximation. *ACM SIGGRAPH*, pages 376–381, 2003. → pages 83
- [194] M. Hlawitschka, J. Ebling, and G. Scheuermann. Convolution and fourier transform of second order tensor fields. In *Proc. IASTED VIIP*, pages 78–83, 2004. → pages 84
- [195] C. S. Bamji, P. O’Connor, T. Elkhatib, S. Mehta, B. Thompson, L. A. Prather, D. Snow, O. C. Akkaya, A. Daniel, A. D. Payne, et al. A 0.13  $\mu\text{m}$  CMOS system-on-chip for a  $512 \times 424$  time-of-flight image sensor with multi-frequency photo-demodulation up to 130 MHz and 2 GS/s ADC. *Solid-State Circuits, IEEE Journal of*, 50(1):303–319, 2015. → pages 86
- [196] B. Das, K. Yoo, and R. Alfano. Ultrafast time-gated imaging in thick tissues: a step toward optical mammography. *Opt. Lett.*, 18(13):1092–1094, 1993. → pages 88, 89

- [197] P. Han, G. Cho, and X.-C. Zhang. Time-domain transillumination of biological tissues with terahertz pulses. *Opt. Lett.*, 25(4):242–244, 2000. → pages 88
- [198] J. B. Schmidt, Z. D. Schaefer, T. R. Meyer, S. Roy, S. A. Danczyk, and J. R. Gord. Ultrafast time-gated ballistic-photon imaging and shadowgraphy in optically dense rocket sprays. *Appl. Opt.*, 48(4):B137–B144, 2009. → pages 89
- [199] BrightWay Vision Active Gated Imaging. <http://www.brightwayvision.com/Technology> (accessed on September 21st, 2016). → pages 89
- [200] F. R. Dalgleish, A. K. Vuorenkoski, and B. Ouyang. Extended-range undersea laser imaging: Current research status and a glimpse at future technologies. *Marine Technology Society Journal*, 47(5):128–147, 2013. → pages 89
- [201] L. J. Mullen and V. M. Contarino. Hybrid lidar-radar: seeing through the scatter. *IEEE Microwave Magazine*, 1(3):42–48, 2000. → pages 89
- [202] L. J. Mullen, V. M. Contarino, A. Laux, B. M. Concannon, J. P. Davis, M. P. Strand, and B. W. Coles. Modulated laser line scanner for enhanced underwater imaging. In *SPIE’s International Symposium on Optical Science, Engineering, and Instrumentation*, pages 2–9. International Society for Optics and Photonics, 1999. → pages 89
- [203] L. J. Mullen, A. J. Vieira, P. Herezfeld, and V. M. Contarino. Application of RADAR technology to aerial LIDAR systems for enhancement of shallow underwater target detection. *IEEE Transactions on microwave theory and techniques*, 43(9):2370–2377, 1995. → pages 89
- [204] L. Mullen, A. Laux, and B. Cochenour. Propagation of modulated light in water: implications for imaging and communications systems. *Applied optics*, 48(14):2607–2612, 2009. → pages 89
- [205] B. Büttgen, T. Oggier, M. Lehmann, R. Kaufmann, and F. Lustenberger. CCD/CMOS lock-in pixel for range imaging: Challenges, limitations and state-of-the-art. *1st range imaging research day*, pages 21–32, 2005. → pages 89
- [206] B. Buttgen and P. Seitz. Robust optical time-of-flight range imaging based on smart pixel structures. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 55(6):1512–1525, 2008. → pages 89

- [207] D. L. Donoho. Compressed sensing. *IEEE Trans. Inform. Theory*, 52(4): 1289–1306, 2006. → pages 90, 91
- [208] E. J. Candes, J. K. Romberg, and T. Tao. Stable signal recovery from incomplete and inaccurate measurements. *Comm. Pure Appl. Math.*, 59(8): 1207–1223, 2006. → pages 90
- [209] Y. C. Eldar and G. Kutyniok. *Compressed Sensing: Theory and Applications*. Cambridge University, 2012. → pages 90
- [210] A. Kadambi, R. Whyte, A. Bhandari, L. Streeter, C. Barsi, A. Dorrington, and R. Raskar. Coded time of flight cameras: sparse deconvolution to address multipath interference and recover time profiles. *ACM Transactions on Graphics (TOG)*, 32(6):167, 2013. → pages 90
- [211] A. Bhandari, A. Kadambi, R. Whyte, C. Barsi, M. Feigin, A. Dorrington, and R. Raskar. Resolving multipath interference in time-of-flight imaging via modulation frequency diversity and sparse regularization. *Opt. Lett.*, 39(6):1705–1708, Mar 2014. → pages 90, 97
- [212] E. Van Den Berg and M. P. Friedlander. Probing the Pareto frontier for basis pursuit solutions. *SIAM Journal on Scientific Computing*, 31(2):890–912, 2008. → pages 91
- [213] S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. *SIAM review*, 43(1):129–159, 2001. → pages 91
- [214] A. Velten, R. Raskar, and M. Bawendi. Picosecond Camera for Time-of-Flight Imaging. In *Imaging and Applied Optics*, page IMB4. Optical Society of America, 2011. → pages 91, 96
- [215] E. Grushka. Characterization of exponentially modified Gaussian peaks in chromatography. *Anal. Chem.*, 44(11):1733–1738, 1972. → pages 92
- [216] P. Datte, A. M. Manuel, M. Eckart, M. Jackson, H. Khater, and M. Newton. Evaluating radiation induced noise effects on pixelated sensors for the National Ignition Facility. volume 8850, pages 885003–885003–9, 2013. → pages 92
- [217] M. S. Lewicki and T. J. Sejnowski. Coding Time-varying Signals Using Sparse, Shift-invariant Representations. In *Proceedings of the 1998 Conference on Advances in Neural Information Processing Systems II*, pages 730–736, Cambridge, MA, USA, 1999. MIT. ISBN 0-262-11245-0. → pages 95



- [218] M. Mørup, M. N. Schmidt, and L. K. Hansen. Shift Invariant Sparse Coding of Image and Music Data. Technical report, Technical University of Denmark, Richard Petersens Plads bld. 321, 2800 Kgs. Lyngby, Denmark, 2008. → pages 95
- [219] M. D. Zeiler and R. Fergus. Learning Image Decompositions with Hierarchical Sparse Coding. Technical Report TR2010-935, Courant Institute of Mathematical Science, New York University, 2010. → pages
- [220] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus. Deconvolutional networks. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2528–2535. IEEE, 2010. → pages 95, 120, 122, 128, 129, 131, 132, 137, 299
- [221] J. Lin, Y. Liu, M. B. Hullin, and Q. Dai. Fourier Analysis on Transient Imaging with a Multifrequency Time-of-Flight Camera. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014. → pages 96, 97
- [222] R. Tibshirani. Regression shrinkage and selection via the lasso. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, pages 267–288, 1996. → pages 96
- [223] J. A. Tropp and A. C. Gilbert. Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Trans. Inform. Theory*, 53(12): 4655–4666, 2007. → pages 96
- [224] S. Rangan. Generalized approximate message passing for estimation with random linear mixing. In *Information Theory Proceedings (ISIT), 2011 IEEE International Symposium on*, pages 2168–2172. IEEE, 2011. → pages 96
- [225] R. Raskar and J. Tumblin. *Computational Photography: Mastering New Techniques For Lenses, Lighting, and Sensors*. A K Peters, Limited, 2007. ISBN 9781568813134. → pages 104
- [226] O. Gupta, T. Willwacher, A. Velten, A. Veeraraghavan, and R. Raskar. Reconstruction of hidden 3D shapes using diffuse reflections. *Opt. Express*, 20(17):19096–19108, Aug 2012. → pages 104
- [227] X. Zhang, M. Burger, and S. Osher. A unified primal-dual algorithm framework based on Bregman iteration. *J. Sci. Comp.*, 46(1):20–46, 2011. → pages 111

- [228] E. Esser, X. Zhang, and T. Chan. A general framework for a class of first order primal-dual algorithms for convex optimization in imaging science. *SIAM J. Imag. Sci*, 3(4):1015–1046, 2010. → pages 111, 233
- [229] E. J. Candes, M. B. Wakin, and S. P. Boyd. Enhancing sparsity by reweighted L1 minimization. *J. Fourier Analysis and Applications*, 14(5-6): 877–905, 2008. → pages 112
- [230] B. A. Olshausen and D. J. Field. Sparse coding with an overcomplete basis set: A strategy employed by V1? *Vision Research*, 37(23):3311 – 3325, 1997. ISSN 0042-6989. → pages 120
- [231] A. Szlam, K. Kavukcuoglu, and Y. LeCun. Convolutional matching pursuit and dictionary training. *arXiv:1010.0422*, 2010. → pages 120
- [232] B. Chen, G. Polatkan, G. Sapiro, D. Blei, D. Dunson, and L. Carin. Deep Learning with Hierarchical Convolutional Factor Analysis. *IEEE Trans. PAMI*, 35(8):1887–1901, 2013. → pages 120
- [233] K. Kavukcuoglu, P. Sermanet, Y. Boureau, K. Gregor, M. Mathieu, and Y. LeCun. Learning Convolutional Feature Hierachies for Visual Recognition. In *Proc. NIPS*, 2010. → pages 120, 121, 122, 137
- [234] M. D. Zeiler, G. W. Taylor, and R. Fergus. Adaptive deconvolutional networks for mid and high level feature learning. In *Proc. ICCV*, pages 2018–2025, 2011. → pages 120
- [235] A. M. Bruckstein, D. L. Donoho, and M. Elad. From sparse solutions of systems of equations to sparse modeling of signals and images. *SIAM review*, 51(1):34–81, 2009. → pages 120
- [236] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online dictionary learning for sparse coding. In *Proc. ICML*, pages 689–696. ACM, 2009. → pages 120
- [237] R. B. Grosse, R. Raina, H. Kwong, and A. Y. Ng. Shift-Invariance Sparse Coding for Audio Classification. In *Proc. UAI*, pages 149–158, 2007. → pages 121
- [238] H. Bristow, A. Eriksson, and S. Lucey. Fast convolutional sparse coding. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 391–398. IEEE, 2013. → pages 121, 122, 123, 126, 128, 129, 130, 132, 133, 135, 299

- [239] H. Bristow and S. Lucey. Optimization Methods for Convolutional Sparse Coding. *arXiv:1406.2407*, 2014. → pages 122, 126, 128, 129, 130, 132, 133, 135, 299
- [240] B. Kong and C. C. Fowlkes. Fast Convolutional Sparse Coding (FCSC). Technical report, UCI, May 2014. → pages 121, 122, 126
- [241] V. Cevher, S. Becker, and M. Schmidt. Convex Optimization for Big Data: Scalable, randomized, and parallel algorithms for big data analytics. *IEEE Signal Processing Magazine*, 31(5):32–43, 2014. → pages 121
- [242] M. S. Almeida and M. A. Figueiredo. Frame-based image deblurring with unknown boundary conditions using the alternating direction method of multipliers. In *Proc. ICIP*, pages 582–585, 2013. → pages 124, 228
- [243] M. Schmidt, N. L. Roux, and F. Bach. Minimizing finite sums with the stochastic average gradient. *arXiv:1309.2388*, 2013. → pages 121
- [244] W. Dong, G. Shi, X. Li, Y. Ma, and F. Huang. Compressive Sensing via Nonlocal Low-rank Regularization. *IEEE Transactions on Image Processing*, 23(8):3618 – 3632, 2014. → pages 134
- [245] X. Lin, G. Wetzstein, Y. Liu, and Q. Dai. Dual-coded compressive hyperspectral imaging. *OSA Opt. Lett.*, 39(7):2044–2047, 2014. → pages 137
- [246] K. Marwah, G. Wetzstein, Y. Bando, and R. Raskar. Compressive Light Field Photography Using Overcomplete Dictionaries and Optimized Projections. *ACM Trans. Graph. (SIGGRAPH)*, 32(4):46:1–46:12, 2013. → pages 137
- [247] B. Horn and B. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185203, 1981. → pages 139
- [248] Y. Li and G. Stuber. *Orthogonal Frequency Division Multiplexing for Wireless Communications*. Springer, 2006. → pages 143
- [249] F. Yasuma, T. Mitsunaga, D. Iso, and S. K. Nayar. Generalized assorted pixel camera: postcapture control of resolution, dynamic range, and spectrum. *IEEE TIP*, 19(9):2241–2253, 2010. → pages 147
- [250] J. Gu, Y. Hitomi, T. Mitsunaga, and S. Nayar. Coded Rolling Shutter Photography: Flexible Space-Time Sampling. In *Proc. ICCP*, 2010. → pages 147

- [251] M. Erz and B. Jähne. Radiometric and spectrometric calibrations, and distance noise measurement of ToF cameras. In *Dynamic 3D Imaging*, pages 28–41. Springer, 2009. → pages 152, 300
- [252] C. Liu. *Beyond pixels: exploring new representations and applications for motion analysis*. PhD thesis, MIT, 2009. → pages 157
- [253] C. Liu, J. Yuen, A. Torralba, J. Sivic, and W. T. Freeman. SIFT flow: Dense correspondence across different scenes. In *Computer Vision–ECCV 2008*, pages 28–42. Springer, 2008. → pages 157
- [254] D. Honegger, L. Meier, P. Tanskanen, and M. Pollefeys. An open source and open hardware embedded metric optical flow CMOS camera for indoor and outdoor applications. In *Proc. ICRA*, pages 1736–1741. IEEE, 2013. → pages 158
- [255] V. Castaneda, D. Mateus, and N. Navab. Stereo Time-of-Flight with Constructive Interference. *IEEE Trans. PAMI*, 36(7):1402–1413, 2014. → pages 159, 162
- [256] L. Li, S. Xiang, Y. Yang, and L. Yu. Multi-camera interference cancellation of time-of-flight (TOF) cameras. In *Proc. IEEE ICIP*, pages 556–560, 2015. → pages 162
- [257] S. Jayasuriya, A. Pediredla, S. Sivaramakrishnan, A. Molnar, and A. Veeraraghavan. Depth Fields: Extending Light Field Techniques to Time-of-Flight Imaging. In *Proc. 3DV*, pages 1–9, 2015. → pages 159
- [258] R. Rashed. A Pioneer in Anaclastics: Ibn Sahl on Burning Mirrors and Lenses. *Isis*, 81(3):464–491, 1990. ISSN 00211753. → pages 168
- [259] S. B. Kang. Automatic Removal of Chromatic Aberration from a Single Image. In *Computer Vision and Pattern Recognition, IEEE Conference on*, pages 1–8, 2007. → pages 168
- [260] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proc. IEEE CVPR*, pages 248–255. IEEE, 2009. → pages 172
- [261] R. Kaftory, Y. Y. Schechner, and Y. Y. Zeevi. Variational distance-dependent image restoration. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007. → pages 172

- [262] L. Yuan, J. Sun, L. Quan, and H.-Y. Shum. Image deblurring with blurred/noisy image pairs. *ACM TOG (SIGGRAPH)*, 26(3):1, 2007. → pages 177, 301, 303
- [263] L. Yuan, J. Sun, L. Quan, and H.-Y. Shum. Progressive inter-scale and intra-scale non-blind image deconvolution. *ACM Trans. Graph.*, 27(3): 74:1–74:10, Aug. 2008. → pages 177, 301
- [264] D. L. Marks, R. A. Stack, D. J. Brady, and J. van der Gracht. Three-dimensional tomography using a cubic-phase plate extended depth-of-field system. *Opt. Lett.*, 24(4):253–255, 1999. → pages 182
- [265] D. G. Stork and P. R. Gill. Lensless ultra-miniature CMOS computational imagers and sensors. In *Proc. SENSORCOMM*, pages 186–190. IARIA, 2013. → pages 182
- [266] T. Michaeli and M. Irani. Blind deblurring using internal patch recurrence. In *IEEE ECCV*, pages 783–798. Springer, 2014. → pages 183
- [267] L. Sun, S. Cho, J. Wang, and J. Hays. Edge-based blur kernel estimation using patch priors. In *Proc. IEEE ICCP*, pages 1–8. IEEE, 2013. → pages
- [268] C. J. Schuler, M. Hirsch, S. Harmeling, and B. Schölkopf. Blind correction of optical aberrations. In *IEEE ECCV*, pages 187–200. Springer, 2012. → pages 183
- [269] A. Alon and I. Alon. Combined design of optical and image processing elements, 2006. US Patent App. 11/278,255. → pages 184
- [270] J.-J. Moreau. Décomposition orthogonale d’un espace Hilbertien selon deux cônes mutuellement polaires. *CR Acad. Sci. Paris*, 255, 1962. → pages 192, 225, 227
- [271] L. Xu and J. Jia. Two-phase kernel estimation for robust motion deblurring. In *ECCV*, 2010. → pages 194
- [272] X. Wu and D. Zhang. Improvement of Color Video Demosaicking in Temporal Domain. *IEEE TIP*, 15(10), 2006. ISSN 1057-7149. → pages 194
- [273] E. P. Bennett, M. Uyttendaele, C. Zitnick, R. Szeliski, and S. Kang. Video and Image Bayesian Demosaicing with a Two Color Image Prior. In *ECCV*, 2006. → pages 194
- [274] H. S. Malvar, L.-w. He, and R. Cutler. High-quality linear interpolation for demosaicing of Bayer-patterned color images. In *ICASS*, 2004. → pages 195

- [275] G. Jeon and E. Dubois. Demosaicking of noisy Bayer-sampled color images with least-squares luma-chroma demultiplexing and noise level estimation. *IEEE TIP*, 22(1), 2013. → pages 197
- [276] C. J. Schuler, H. C. Burger, S. Harmeling, and B. Schölkopf. A machine learning approach for non-blind image deconvolution. In *CVPR*, 2013. → pages 197, 198, 199, 200
- [277] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image restoration by sparse 3D transform-domain collaborative filtering. In *Electronic Imaging*, 2008. → pages 198, 200
- [278] S. Roth and M. J. Black. Fields of experts. *International Journal of Computer Vision*, 82(2), 2009. → pages 198, 200
- [279] J.-L. Starck, F. Murtagh, and A. Bijaoui. *Image Processing and Data Analysis: The Multiscale Approach*. Cambridge University Press, Cambridge, 1998. ISBN 0-521-59914-8. → pages 202
- [280] K. Dabov, A. Foi, and K. Egiazarian. Video denoising by sparse 3D transform-domain collaborative filtering. In *European Signal Processing Conference*, 2007. → pages 202
- [281] Y.-T. Tsai, M. Steinberger, D. Pająk, and K. Pulli. Fast ANN for High-Quality Collaborative Filtering. In *High-Performance Graphics*, 2014. → pages 205, 239
- [282] A. Brooke, D. Kendrick, A. Meeraus, and R. Rosenthal. *GAMS: A user's guide*. Course Technology, 1988. → pages 211
- [283] M. Grant and S. Boyd. CVX: MATLAB Software for Disciplined Convex Programming, version 2.1. <http://cvxr.com/cvx> (accessed on January 1st, 2016). → pages 211
- [284] J. Lofberg. YALMIP: A Toolbox for Modeling and Optimization in MATLAB. In *Proc. IEEE Int. Symp. Computed Aided Control Systems Design*, pages 294–289, 2004. → pages 211
- [285] S. Diamond and S. Boyd. CVXPY: A Python-Embedded Modeling Language for Convex Optimization. *Journal of Machine Learning Research*, 2016. To appear. → pages 211, 236
- [286] M. Udell, K. Mohan, D. Zeng, J. Hong, S. Diamond, and S. Boyd. Convex optimization in Julia. *Workshop on High Performance Technical Computing in Dynamic Languages*, 2014. → pages 211

- [287] J. Ragan-Kelley, C. Barnes, A. Adams, S. Paris, F. Durand, and S. Amarasinghe. Halide: a language and compiler for optimizing parallelism, locality, and recomputation in image processing pipelines. *ACM SIGPLAN*, 48(6):519–530, 2013. → pages 211
- [288] S. Diamond and S. Boyd. Matrix-Free Convex Optimization Modeling. In *Optimization and Applications in Control and Data Sciences*. Springer, 2016. To appear. → pages 214, 219
- [289] D. Hallac, J. Leskovec, and S. Boyd. Network lasso: Clustering and optimization in large graphs. In *Proc. ACM SIGKDD*, pages 387–396, 2015. → pages 214
- [290] S. Diamond and S. Boyd. Convex Optimization with Abstract Linear Operators. In *Proc. IEEE ICCV*, 2015. → pages 218
- [291] M. Hestenes and E. Stiefel. Methods of Conjugate Gradients for Solving Linear Systems. *J. Res. N.B.S.*, 49(6):409–436, 1952. → pages 221
- [292] C. Paige and M. Saunders. LSQR: An Algorithm for Sparse Linear Equations and Sparse Least Squares. *ACM Trans. Mathematical Software*, 8(1):43–71, 1982. → pages 221
- [293] T. Pock and A. Chambolle. Diagonal preconditioning for first order primal-dual algorithms in convex optimization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1762–1769, 2011. → pages 224, 246
- [294] P. Giselsson and S. Boyd. Diagonal scaling in Douglas-Rachford splitting and ADMM. In *Proceedings of the 53rd IEEE Conference on Decision and Control*, 2014. → pages 224, 246
- [295] R. Lehoucq and D. Sorensen. Deflation Techniques for an Implicitly Restarted Arnoldi Iteration. *SIAM Journal on Matrix Analysis and Applications*, 17(4):789–821, 1996. → pages 224
- [296] Y. Zhu. An augmented ADMM algorithm with application to the generalized lasso problem. *Journal of Computational and Graphical Statistics*, (just-accepted), 2015. → pages 225
- [297] B. O’Donoghue, E. Chu, N. Parikh, and S. Boyd. Operator Splitting for Conic Optimization via Homogeneous Self-Dual Embedding. *arXiv e-Print 1312.3039*, 2015. → pages 226

- [298] M. C. Robini and Y. Zhu. Generic half-quadratic optimization for image reconstruction. *SIAM Journal on Imaging Sciences*, 8(3):1752–1797, 2015. → pages 230, 233
- [299] C. Fougner and S. Boyd. Parameter Selection and Pre-Conditioning for a Graph Form Solver. *arXiv e-Print 1503.08366*, 2015. → pages 230, 246
- [300] J. Eckstein and D. Bertsekas. On the Douglas—Rachford splitting method and the proximal point algorithm for maximal monotone operators. *Mathematical Programming*, 55(1):293–318, 1992. → pages 233
- [301] J. Nocedal and S. Wright. *Numerical Optimization*. Springer Science, 2006. → pages 233
- [302] A. Beck. On the Convergence of Alternating Minimization for Convex Programming with Applications to Iteratively Reweighted Least Squares and Decomposition Schemes. *SIAM Journal on Optimization*, 25(1):185–209, 2015. → pages 233
- [303] Y. Wang, J. Yang, W. Yin, and Y. Zhang. A New Alternating Minimization Algorithm for Total Variation Image Reconstruction. *SIAM Journal on Imaging Sciences*, 1(3):248–272, 2008. → pages 233
- [304] G. D. Evangelidis and E. Z. Psarakis. Parametric image alignment using enhanced correlation coefficient maximization. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(10):1858–1865, 2008. → pages 241
- [305] F.-X. Dupe, M. Fadili, and J.-L. Starck. Inverse problems with Poisson noise: Primal and primal-dual splitting. In *Proc. ICIP*, 2011. → pages 242, 322
- [306] M. Figueiredo and J. Bioucas-Dias. Deconvolution of Poissonian images using variable splitting and augmented Lagrangian optimization. In *SWorkshop on Statistical Signal Processing*, pages 733–736, 2009. → pages 242, 322
- [307] M. Figueiredo and J. Bioucas-Dias. Restoration of Poissonian images using alternating direction optimization. *IEEE Trans. Image Processing*, 19(12): 3133–3145, 2010. → pages 243
- [308] J. R. Fienup. Phase retrieval algorithms: a comparison. *Applied Optics*, 21(15):2758–2769, 1982. → pages 244



- [309] K. Kunisch and T. Pock. A bilevel optimization approach for parameter learning in variational models. *SIAM Journal on Imaging Sciences*, 6(2): 938–983, 2013. → pages 246, 251
- [310] M. O’Toole, F. Heide, L. Xiao, M. B. Hullin, W. Heidrich, and K. N. Kutulakos. Temporal frequency probing for 5D transient analysis of global light transport. *ACM Transactions on Graphics (TOG)*, 33(4):87, 2014. → pages 250
- [311] E. Lach, K. Schuh, and M. Schmidt. Application of electroabsorption modulators for high-speed transmission systems. In *Ultrahigh-Speed Optical Transmission Technology*, pages 347–377. Springer, 2005. → pages 250
- [312] I. Gkioulekas, A. Levin, F. Durand, and T. Zickler. Micron-scale light transport decomposition using interferometry. *ACM Transactions on Graphics (TOG)*, 34(4):37, 2015. → pages 251
- [313] P. Ochs, R. Ranftl, T. Brox, and T. Pock. Bilevel optimization with nonsmooth lower level problems. In *International Conference on Scale Space and Variational Methods in Computer Vision*, pages 654–665. Springer, 2015. → pages 251
- [314] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014. → pages 252
- [315] C. Audet and J. Dennis Jr. Mesh adaptive direct search algorithms for constrained optimization. *SIAM Journal on optimization*, 17(1):188–217, 2006. → pages 288
- [316] M. Schmidt. minConf. <http://www.cs.ubc.ca/~schmidtm/Software/minConf.html> (accessed on January 1st, 2016). → pages 288
- [317] O. Shcherbakova. *3D Camera Based on Gain-Modulated CMOS Avalanche Photodiodes*. PhD thesis, University of Trento, 2013. → pages 300
- [318] J. Salmon, Z. Harmany, C.-A. Deledalle, and R. Willett. Poisson noise reduction with non-local PCA. *Journal of mathematical imaging and vision*, 48(2):279–294, 2014. → pages 300, 301

- [319] M. Trimeche, D. Paliy, M. Vehviläinen, and V. Katkovnik. Multichannel image deblurring of raw color components. In *Proc. Computational Imaging III*, pages 169–178, 2005. → pages 304
- [320] N. Joshi, R. Szeliski, and D. J. Kriegman. PSF estimation using sharp edge prediction. In *Computer Vision and Pattern Recognition, IEEE Conference on*, pages 1–8, 2008. → pages 304
- [321] J. Brauers, C. Seiler, and T. Aach. Direct PSF Estimation Using a Random Noise Target. In *Proceedings of SPIE, the International Society for Optical Engineering*, page 75370B, 2010. → pages 304
- [322] G. Wetzstein, I. Ihrke, A. Gukov, and W. Heidrich. Towards a database of high-dimensional plenoptic images. → pages 306
- [323] N. Gat. Imaging spectroscopy using tunable filters: a review. In *AeroSense 2000*, pages 50–64. International Society for Optics and Photonics, 2000. → pages 306
- [324] K. Mitra, O. Cossairt, and A. Veeraraghavan. A Framework for Analysis of Computational Imaging Systems: Role of Signal Prior, Sensor Noise and Multiplexing. *IEEE PAMI*, 36(10), 2014. → pages 311
- [325] E. Candès and D. Donoho. Curvelets: A Surprisingly Effective Nonadaptive Representation of Objects with Edges. In *Curves and Surfaces*. Vanderbilt University Press, 1999. → pages 311
- [326] A. Buades, B. Coll, and J.-M. Morel. Non-Local Means Denoising. *Image Processing On Line*, 2011. → pages 313
- [327] C. Liu. *Beyond Pixels: Exploring New Representations and Applications for Motion Analysis*. PhD thesis, MIT, 2009. → pages 317
- [328] A. Joulin and S. B. Kang. Recovering Stereo Pairs from Anaglyphs. In *CVPR*, 2013. → pages 318, 319
- [329] C. Liu, J. Yuen, A. Torralba, J. Sivic, and W. T. Freeman. SIFT Flow: Dense Correspondence Across Different Scenes. In *ECCV*, 2008. → pages 319
- [330] G. K. Wallace. The JPEG still picture compression standard. *Communications of the ACM*, 34(4), 1991. → pages 319
- [331] A. Foi, V. Katkovnik, and K. Egiazarian. Pointwise shape-adaptive DCT for high-quality denoising and deblocking of grayscale and color images. *IEEE TIP*, 16(5), 2007. → pages 320, 321

## Appendix A

# Supporting Materials

### A.1 Detailed Spectral Analysis of the CIS Image Formation

In Section 2.6.3 we have derived and explained the image formation for CIS in the spectral domain. Figure 2.11 illustrates this spectral analysis. The following Eq. (A.1) provides the corresponding detailed mathematical derivation.

$$\begin{aligned}
 b(\rho, \theta) &= \int_{\rho}^{\rho+T} p(t) dt \\
 &= \left( \text{rect}_T \otimes (f \cdot s) \right) (\rho) \\
 &= \mathfrak{F}^{-1} \left( \text{sinc}_T \cdot \hat{f} \otimes \hat{s} \right) (\rho) \\
 &= \mathfrak{F}^{-1} \left( \text{sinc}_T \cdot \left( e^{-2\pi i \theta \xi} \frac{\delta(\xi - \frac{\omega_f}{2\pi}) + \delta(\xi + \frac{\omega_f}{2\pi})}{2} \right) \otimes \left( a \cdot e^{-2\pi i \theta \xi} \frac{\delta(\xi - \frac{\omega_g}{2\pi}) + \delta(\xi + \frac{\omega_g}{2\pi})}{2} + I \delta(\xi) \right) \right) (\rho) \\
 &= \mathfrak{F}^{-1} \left( \text{sinc}_T \cdot \left( a \cdot e^{-2\pi i(\phi+\theta)\xi} \cdot \frac{\delta(\xi - \frac{\omega_g + \omega_f}{2\pi}) + \delta(\xi + \frac{\omega_g + \omega_f}{2\pi})}{2} + \delta(\xi - \frac{\omega_g - \omega_f}{2\pi}) + \delta(\xi + \frac{\omega_g - \omega_f}{2\pi}) + I e^{-2\pi i \theta \xi} \frac{\delta(\xi - \frac{\omega_f}{2\pi}) + \delta(\xi + \frac{\omega_f}{2\pi})}{2} \right) \right) (\rho) \\
 &\approx \mathfrak{F}^{-1} \left( a \cdot e^{-2\pi i(\phi+\theta)\xi} \cdot \frac{\delta(\xi - \frac{\omega_g - \omega_f}{2\pi}) + \delta(\xi + \frac{\omega_g - \omega_f}{2\pi})}{2} + 0 \right) (\rho) \\
 &= \begin{cases} \mathfrak{F}^{-1} \left( a \cdot e^{-2\pi i(\phi+\theta)\xi} \cdot \delta(\xi) \right) (\rho) & \text{if } \omega_f = \omega_g \\ \mathfrak{F}^{-1} \left( a \cdot e^{-2\pi i(\phi+\theta)\xi} \cdot \frac{\delta(\xi - \frac{\omega_g}{2\pi}) + \delta(\xi + \frac{\omega_g}{2\pi})}{2} \right) (\rho) & \text{if } \omega_f = \omega_g - \omega_\delta \end{cases}
 \end{aligned} \tag{A.1}$$

### A.2 Harmonic CIS Image Formation

In this section we expand on the sinusoidal model derived in Section 2.6 to allow for arbitrary periodic modulation functions for both the illumination and the camera reference signal. As before, we assume periodically modulated illumination with an

angular frequency  $\omega$ . In the general case, this illumination signal can be described as a superposition of all the harmonics of this frequency at different phases  $\theta_k$ :

$$g(t) = \sum_{k=0}^{\infty} g_k \cos(k\omega t + \theta_k)$$

Assuming a stationary object, the illumination arriving at the sensor can then be expressed as

$$s(t) = \sum_{k=0}^{\infty} s_k \cos(k\omega t + \phi_k)$$

where  $s_0$  includes ambient illumination, and  $s_k$  for  $k > 0$  includes geometric terms and the surface albedo as before. The phase shifts are given as

$$\phi_k = \theta_k + k\omega \frac{2d}{c}$$

Assuming a zero-mean modulation function  $f$ , the modulated signal at the sensor is then

$$\begin{aligned} \tilde{i}_\theta(t) &= f_\theta(t) \cdot s(t) \\ &= \sum_{j=1}^{\infty} f_j \cos(j\omega t + \theta_j) \cdot \sum_{k=0}^{\infty} s_k \cos(k\omega t + \phi_k) \\ &= \frac{1}{2} \sum_{j=1}^{\infty} \sum_{k=0}^{\infty} f_j s_k \cos((k-j)\omega t + \phi_k - \theta_j) + \\ &\quad \frac{1}{2} \sum_{j=1}^{\infty} \sum_{k=0}^{\infty} f_j s_k \cos((k+j)\omega t + \phi_k + \theta_j) \\ &= \frac{1}{2} \sum_{k=0}^{\infty} f_k s_k \cos(\phi_k - \theta_k) + \\ &\quad \frac{1}{2} \sum_{j=1}^{\infty} \sum_{k \neq j}^{\infty} f_j s_k \cos((k-j)\omega t + \phi_k - \theta_j) + \\ &\quad \frac{1}{2} \sum_{j=1}^{\infty} \sum_{k=0}^{\infty} f_j s_k \cos((k+j)\omega t + \phi_k + \theta_j). \end{aligned}$$

Low-pass filtering due to the finite exposure time  $T$  eliminates all but the first term, resulting in Equation A.2.

$$p_\theta(t) = \tilde{i}_\theta(t) * \text{rect}(Tt) \approx \frac{1}{2} \sum_{k=0}^{\infty} f_k s_k \cos(\phi_k - \theta_k). \quad (\text{A.2})$$

In order to recover depth and albedo from Eq. A.2, the camera needs to be calibrated for different depths, either as a dense lookup table of depth, or by estimating the coefficients  $f_k$  and  $s_k$  from sparser measurements.

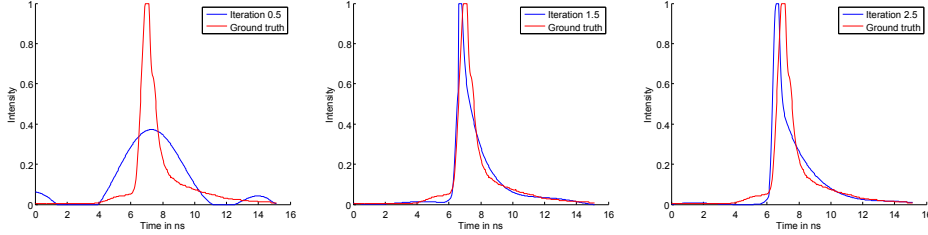
### A.3 Transient Imaging with Gaussian-Exponential Priors

To handle the non-convexity of the inverse problem from Section 3.3, we propose to solve the following problem using coordinate descent.

$$\begin{aligned}
(\mathbf{u}_{\text{opt}}, i_{\text{opt}}) = \underset{\mathbf{u}, \mathbf{i}}{\text{argmin}} \quad & \frac{1}{2} \|\mathbf{C}\mathbf{i} - \mathbf{b}\|_2^2 + \\
& \lambda \sum_{\mathbf{x}} \|\nabla_z \mathbf{i}_{\mathbf{x}}\|_H + \theta \sum_{\tau} \|\nabla_{\mathbf{x}} \mathbf{i}_{\tau}\|_H + \\
& \frac{1}{2} \|\mathbf{C}m(\mathbf{u}) - \mathbf{b}\|_2^2 + \frac{\rho}{2} \|\mathbf{i} - m(\mathbf{u})\|_2^2 + \\
& \frac{\beta}{2} \sum_{\mathbf{x}} \|\text{maxima}(\mathbf{i}_{\mathbf{x}}) - \mathbf{p}_{\mathbf{x}}\|_2^2 \\
\text{s.t.} \quad & \forall \mathbf{x}, i : K_{\mathbf{x}} = \#\text{maxima}(\mathbf{i}_{\mathbf{x}}) \wedge \\
& \mathbf{r}_{\mathbf{x},i}, \mathbf{a}_{\mathbf{x},i}, \mathbf{d}_{\mathbf{x},i} \in [0, 1]
\end{aligned} \quad (\text{A.3})$$

Here  $\text{maxima}(\mathbf{i}_{\mathbf{x}})$  are the positions of the temporal maxima for a pixel  $\mathbf{x}$ , and  $\#\text{maxima}(\mathbf{i}_{\mathbf{x}})$  is number of such local maxima. The additional constraints therefore express that we prefer locations of Gaussians and exponentials close to the onset positions, that we restrict the number  $K_{\mathbf{x}}$  to be the number of onsets, and constrain the numerical range of Gaussian and exponential amplitudes and decays. Hence, solving this problem restricts the large parameter space of the model to a local region around the fixed variables. Note that we have duplicated the data fidelity term here for  $\mathbf{i}$  and  $\mathbf{u}$ . We found this to be beneficial for the convergence behavior of the  $\mathbf{u}$ -subproblem. Note that this does not affect the separability of  $\mathbf{u}$  in the coordinate descent. The proposed coordinate descent does not have convergence guarantees.

However, we found it to perform well in practice. Figure A.1 shows iterations of the coordinate descent method.



**Figure A.1:** A single synthetic pixel signal (over time) reconstructed with the proposed coordinate descent. Left: Solution obtained by solving for  $\mathbf{i}$  only. Notice that the solution is smooth in temporal and spatial domain and contains the right number of modes. Center: solution after one full iteration plus one extra  $\mathbf{i}$ -step to reconstruct the transient image from the mixed Gaussian and exponential model parameters  $\mathbf{u}$ . Notice the drastically improved reconstruction fidelity. Right: an additional iteration does not improve the result significantly.

In the  $\mathbf{i}$ -subproblem from Section 3.3, we can neglect the maxima terms. This is justified by the fact that these terms will be dominated by  $\frac{\rho}{2} \|\mathbf{i} - m(\mathbf{u})\|_2^2$  for large  $\rho$  (recall that we only introduced the extra terms to account for the non-convexity of the  $\mathbf{u}$ -subproblem). The resulting problem then becomes a convex optimization problem. It is discussed in Section 3.3. In the following we describe the second subproblem, the  $\mathbf{u}$ -subproblem.

### Solving the $\mathbf{u}$ -subproblem

In the second subproblem, we fix  $\mathbf{i}$  in Eq. (A.3) and solve  $\mathbf{u}$ , leading to the following problem:

$$\begin{aligned}
 \mathbf{u}_{\text{opt}} = \underset{\mathbf{u}}{\text{argmin}} \quad & \frac{1}{2} \|\mathbf{C}m(\mathbf{u}) - \mathbf{b}\|_2^2 + \frac{\rho}{2} \|\mathbf{i} - m(\mathbf{u})\|_2^2 + \\
 & \frac{\beta}{2} \sum_{\mathbf{x}} \|\text{maxima}(\mathbf{i}_{\mathbf{x}}) - \mathbf{p}_{\mathbf{x}}\|_2^2 \\
 \text{s.t.} \quad & \forall \mathbf{x}, i : K_{\mathbf{x}} = \#\text{maxima}(\mathbf{i}_{\mathbf{x}}) \wedge \\
 & \mathbf{r}_{\mathbf{x},i}, \mathbf{a}_{\mathbf{x},i}, \mathbf{d}_{\mathbf{x},i} \in [0, 1]
 \end{aligned} \tag{A.4}$$

This is a regularized data-fitting problem that is globally still non-convex, so there are no guarantees of obtaining a global optimum. However, by constraining the parameters, and especially the onset  $\mathbf{p}$ , we limit the search to regions where we expect locally convex behavior.

We first determine  $K_{\mathbf{x}} = \#\text{maxima}(\mathbf{i}_{\mathbf{x}})$  by explicitly detecting the number of peaks in time for each pixel signal  $\mathbf{i}_{\mathbf{x}}$ . No noise suppression or further smoothing is needed since we require only a good lower bound for  $K_{\mathbf{x}}$ . Accurate fitting is then handled by solving Eq. (A.4).

The resulting problem (with  $K_{\mathbf{x}}$  now fixed) can be solved independently for each pixel. To account for the non-convexity we perform a two-step optimization per pixel signal: first, several steps of a global search in the constrained parameter space are done using the derivative-free Mesh Adaptive Direct Search method for non-linear constrained optimization (MADS) [315]. This direct search method models the parameter space using a mesh. By adaptively probing the mesh and also adaptively changing the mesh refinement this method makes good progress in regions around local minimizers. We use the implementation in the MATLAB generalized pattern search function "patternsearch" in the Global Optimization Toolbox, random polling and a maximum number of 1000 iterations.

We expect the parameters found by this derivative-free global optimization to lie at least in a locally convex region. To effectively find the according minimum in this region we now use the gradient and do local constrained gradient-descent iterations using the spectral projected gradient method from the minConf package [316]. The projection operator that encodes our constraints is straightforward to derive since in our case we only have simple box constraints.

We use here analytic gradient information of the unconstrained version of Eq. (A.4), i.e.,

$$\mathbf{u}_{\text{opt}} = \underset{\mathbf{u}}{\text{argmin}} \frac{1}{2} \|\mathbf{C}m(\mathbf{u}) - \mathbf{b}\|_2^2 + \frac{\rho}{2} \|\mathbf{i} - m(\mathbf{u})\|_2^2 + \underbrace{\frac{\beta}{2} \sum_{\mathbf{x}} \|\text{maxima}(\mathbf{i}_{\mathbf{x}}) - \mathbf{p}_{\mathbf{x}}\|_2^2}_{\Phi(\mathbf{u})} \quad (\text{A.5})$$

For this subproblem, all the components  $\mathbf{u}_{\mathbf{x}}$  for a pixel  $\mathbf{x}$  are independent of the

components for other pixels. So, for a single component  $j$  of  $\mathbf{u}$ , we obtain the following gradient component:

$$\nabla_j \Phi(\mathbf{u}) = 2 \left[ \frac{\partial m_{\mathbf{x}}(\mathbf{u})}{\partial \mathbf{u}_j} \right]^T \mathbf{C}^T (\mathbf{C}m_{\mathbf{x}}(\mathbf{u}) - \mathbf{b}) \quad (\text{A.6})$$

Thus, with Eq. A.6, we have reduced the gradient computation for  $\Phi(\mathbf{u})$  to the evaluation of the partial derivatives of a pixel time sequence  $m_{\tau}(\mathbf{u})$  with respect to its parameters (that is the Gaussian amplitude, exponential amplitude, exponential decay and the position of the Gaussian and exponential). These partial derivatives are straightforward to derive from Eq. (3.12).

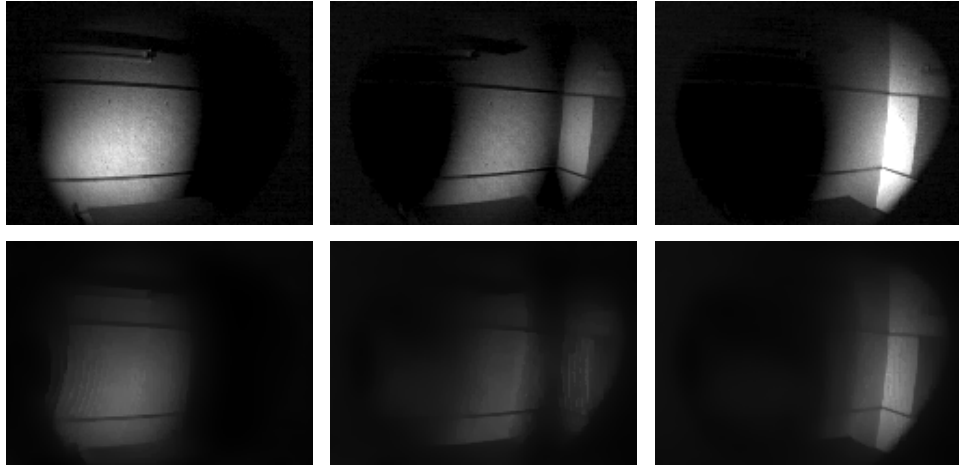
## Discussion

With optimization split into two stages as described above, the  $\mathbf{i}$ -subproblem is fast and convex, and provides a data term plus spatial and temporal regularization, with the tendency to overly smoothen the results. The  $\mathbf{u}$ -subproblem, on the other hand, is a very expensive, non-convex fitting of the model parameters, that produces higher temporal resolution, but may in turn introduce some visually objectionable spatial artifacts. After the final  $\mathbf{u}$ -step, we always perform another  $\mathbf{i}$ -step to obtain a final transient image from the model parameters.

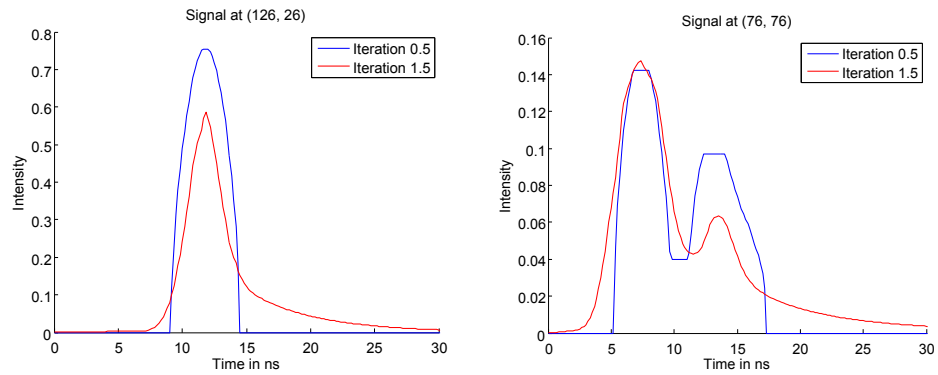
Figure A.1 compares the time sequences obtained for a single pixel with ground truth data for a synthetic example (more details on the dataset can be found in Section 3.5). The left image shows the over-smooth reconstruction after solving just the  $\mathbf{i}$  subproblem. The center and right image show reconstructions after one and two full iterations, respectively (each followed by a final  $\mathbf{i}$ -step). We can see that the method is converged after only a single iteration. This is consistent with behavior we observe for the time sequences of other pixels and datasets.

Figure A.2 shows results for a simple scene where a wavefront travels along a diffuse wall, captured using our PMD camera setup (Section 3.4). Using the model, pixel time profiles (Figure A.3) show a similar sharpening effect as in the synthetic data set. However, as the full-frame reconstructions show, this extra sharpness comes at the cost of some spatial quantization artifacts. We refer to the accompanying video to observe the full dataset. Because of the tradeoff between





**Figure A.2:** Reconstructions for a simple real dataset of a wavefront traveling along a wall. Top row: image reconstruction after only solving the  $i$  subproblem. Bottom: the same images after one iteration of the full optimization problem.



**Figure A.3:** Pixel time sequences for two selected pixels of Figure A.2 after a full iteration and after only solving the  $i$  subproblem.

temporal sharpness and spatial artifacts, it may in some cases be preferable to only run the  $i$  subproblem, depending on whether the goal is to obtain the most detailed measurements possible, or simply to produce visually pleasing results.

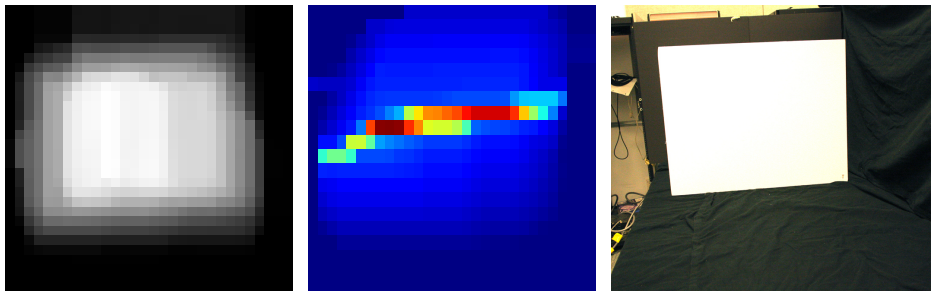
## A.4 Indirect Imaging of Specular Object, and with Ambient Light

Here we show additional experiments that illustrate the performance of our method from Chapter 5 for non-Lambertian surfaces. In addition, we examine the sensitivity to ambient illumination, and the impact of the sparsity inducing priors.

### Non-Lambertian Surfaces

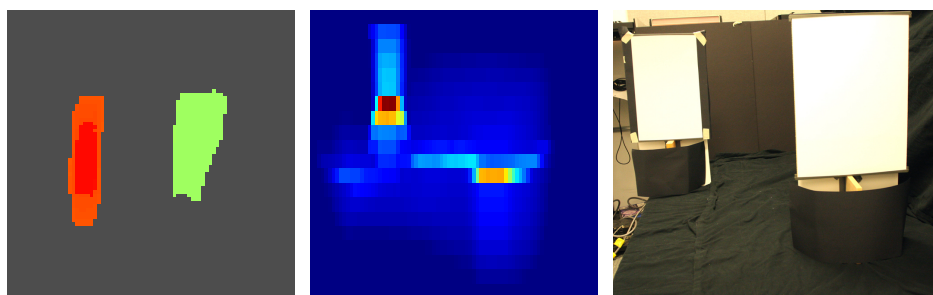
Figures A.4 to A.7 show a progression of results for surfaces that violate the assumption of diffuse Lambertian reflectors. In these images, we show both the color-coded depth map as well as a cross-section through the volume densities  $v(\mathbf{x})$  for one scan line.

Figure A.4 shows a single foam board. This material is quite diffuse, and as a result the volume cross-section shows a good localization of the depth value, revealing the board was slightly tilted in depth.



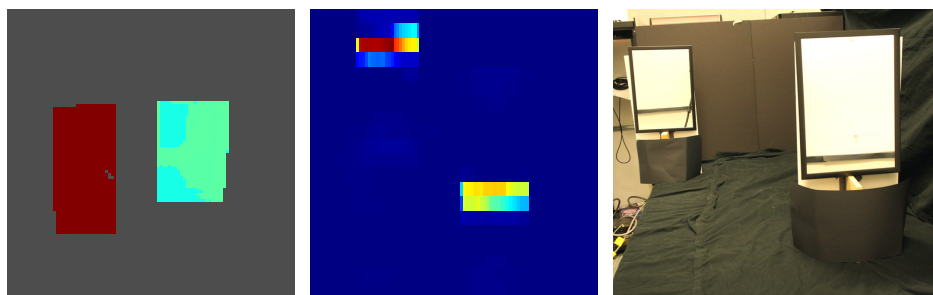
**Figure A.4:** Reconstruction of scene image with diffuse foamboard (right). Smooth depth map, occluder-probability weighted depth along  $z$ -coordinate visualized in jet color map (left). Slice through the reconstruction volume along  $y$ -coordinate (middle).

Figure A.5 shows results for two small whiteboards, which are slightly more shiny than the foam boards, but still very diffuse. We see that the sharpness of depth values suffers a bit from the glossiness of the scene, although isolating the strongest peak for each pixel still yields a good depth map.



**Figure A.5:** Reconstruction of scene image with two whiteboards, that are more reflective than the foam board (right). Color-coded depth map of strongest peak along  $z$ -coordinate visualized (left) (left). Slice through the reconstruction volume along  $y$ -coordinate (middle).

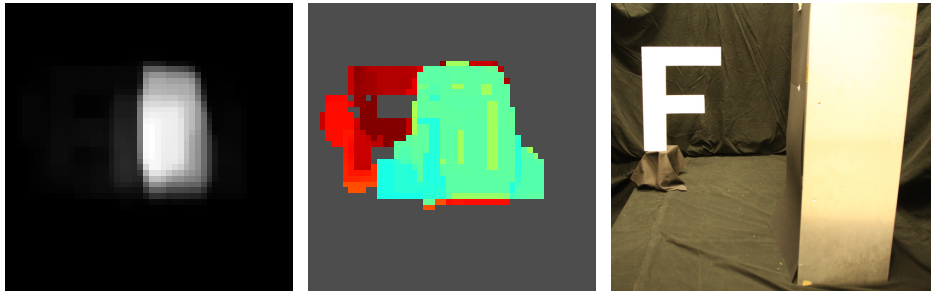
Figure A.6 shows an extreme example with two mirrors. The geometry of the two surfaces can be recovered well, since the reflections from the mirrors do not interfere with each other on the diffuse wall - they create two spatially separated caustics. Our optimization procedure was adapted to handle this special case, where everything stays as before except for the bounce in the volume which is then made specular.



**Figure A.6:** Reconstruction of scene image with two mirrors, that are the most reflective objects tested (right). Color-coded depth map of strongest peak along  $z$ -coordinate visualized (left). Slice through the reconstruction volume along  $y$ -coordinate (middle).

Finally, Figure A.7 shows the result of mixing Lambertian and non-Lambertian surfaces. The F-shape is made out of cardboard material painted with white fairly diffuse paint, while the large rectangular surface is made out of brushed metal. The

reflection from the brushed metal surface is extremely strong and glossy, and thus overlaps with the one from the “F” in both space and time. This causes our method to fail to reconstruct the dimmer “F”-shape.

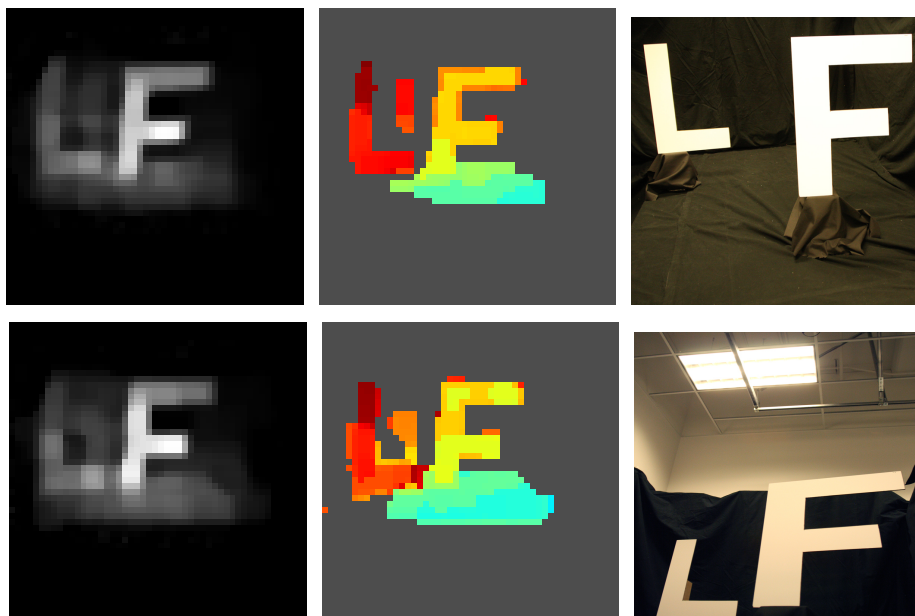


**Figure A.7:** Reconstruction of challenging scene with a strong quite diffuse reflector made from brushed metal and a letter made from painted cardboard (right). Smooth depth map, occluder-probability weighted depth (left). Color-coded depth map of strongest peak along z-coordinate visualized (middle).

### Effects of Ambient Light and Frame Averaging

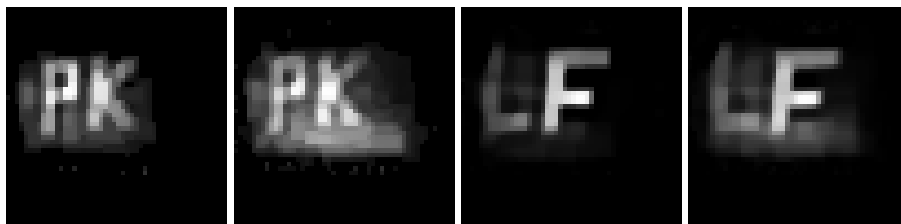
Our method from Chapter 5 is robust to ambient illumination. We have tested the sensitivity to strong ambient illumination for reconstructing geometry (Figure A.8) and albedo (Figure A.9). In both cases we performed our capture once with the ceiling lights in the room switched off, and once with them switched on. We can see that there is only a minor effect on the overall sharpness and reconstruction quality in both cases.

As already mentioned in Section 5.5.1, we average several measurements before reconstruction. This improves SNR, since the measured indirect reflection results in very low light levels. Figure A.10 shows different depth and albedo reconstructions, where each measurement respectively is the average of 10 or 500 individual TOF images with a specific modulation frequency and phase. We see that we still get a reasonable reconstruction by averaging only 10 images. The corresponding capture time of 4 minutes (200 minutes for averaging 500 measurements) could be significantly improved by better synchronizing the PMD camera and light source so that the camera can capture at video rates. Still, even with the current setup,

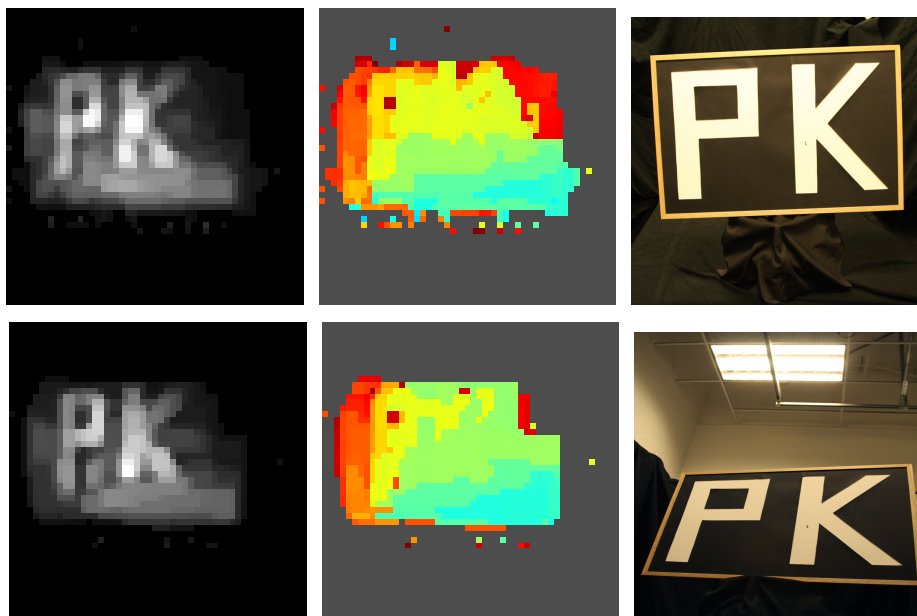


**Figure A.8:** Effects of ambient illumination on geometry reconstruction: All lights in room off (top) and lights on(bottom). We see that we still get a reasonable reconstruction with strong ambient illumination

our capture times compare very favorably to those reported for femtosecond laser setups [149].



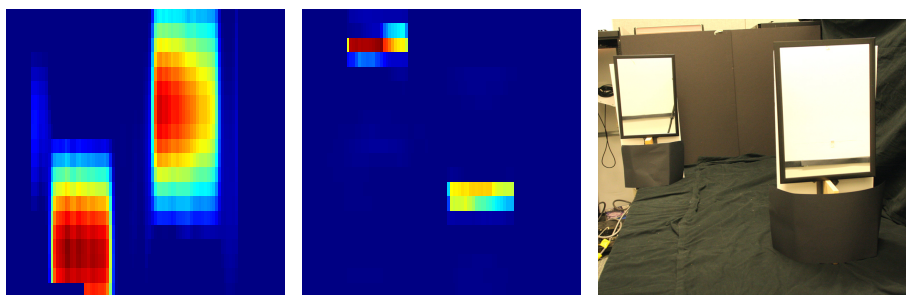
**Figure A.10:** Effects of frame averaging on albedo (left) and geometry (right). The left image in each pair is based on averaging 500 ToF images for each measurement, while the right image in each pair uses only 10.



**Figure A.9:** Effects of ambient illumination on albedo reconstruction: All lights in room off (top) and lights on(bottom). We see that we still get a reasonable reconstruction with strong ambient illumination

### Sparsity Analysis

Figure A.11 shows the effect of the sparsity inducing terms in the optimization on the mirror scene from above.

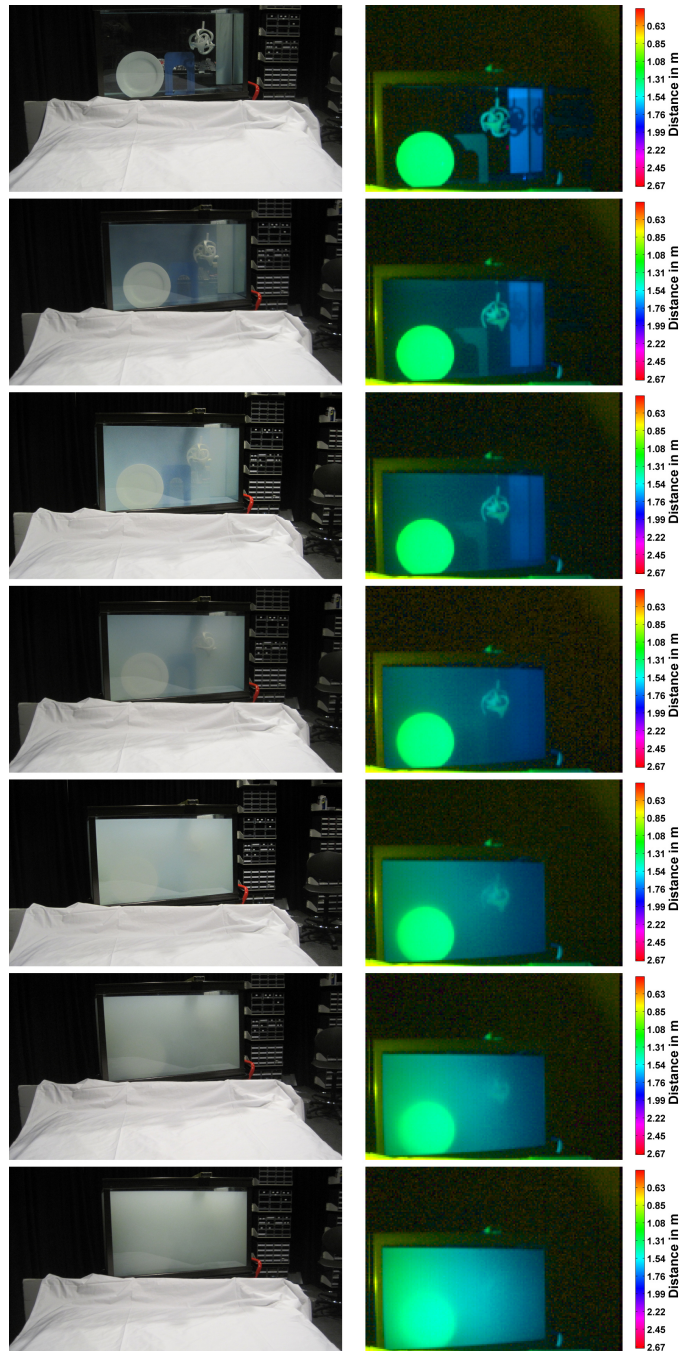


**Figure A.11:** Effect of our sparse optimization: Scene with two planar mirror surfaces shown on the right. Slice through the reconstruction volume computed with *simple backprojection* along  $y$ -coordinate (middle) shown on the left. Slice through the reconstruction volume computed our *whole sparsity preferring optimization* shown in the middle.

We show slices through the recovered volume without the sparsity terms (left) and with the full objective function (center). We can see that our method nicely prefers a very sparse solution and therefore finds good solutions that represent our assumption of height field geometry well.

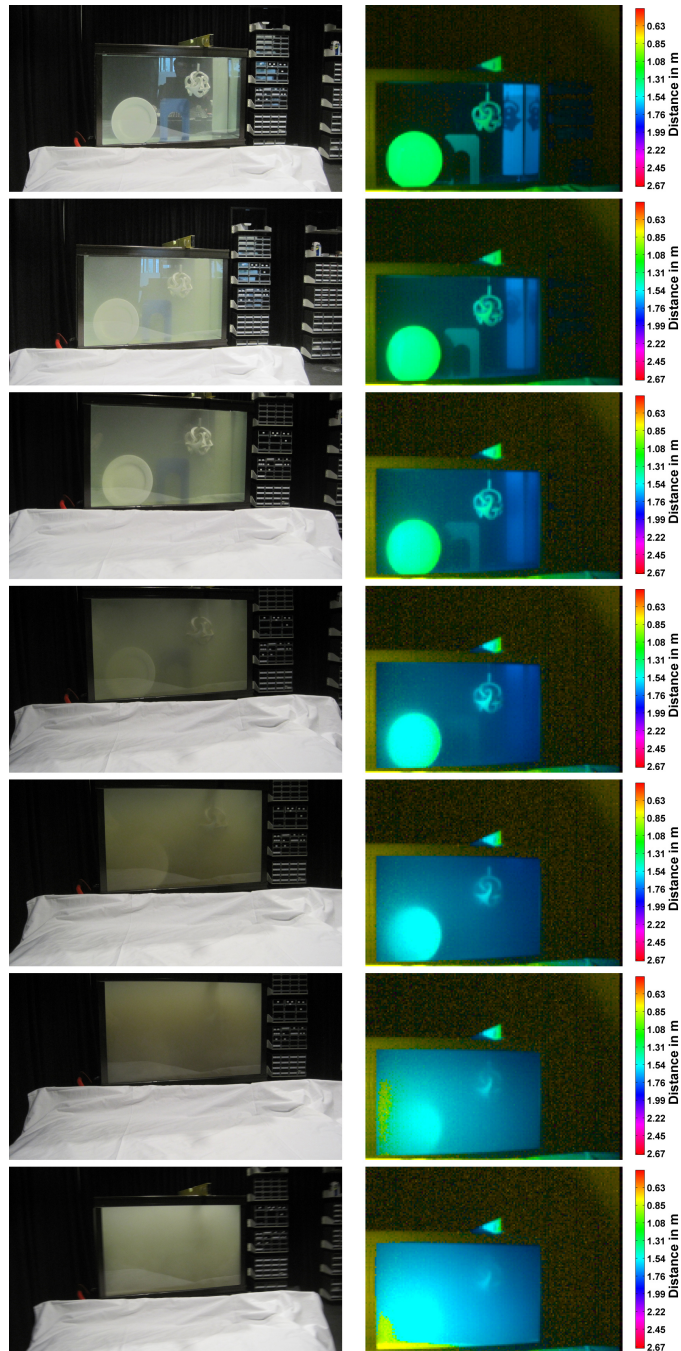
## **A.5 Imaging in Scattering Media of Various Densities**

This section presents additional results demonstrating the robustness of our method from Chapter 4 imaging in scattering and turbid media. Figure A.12 and Figure A.13 show qualitative results for imaging through scattering media of increasing density. See Section 4.4.1 for a description of the experimental setup.



**Figure A.12:** Larger sequence of the milk experiments shown in Figure 4.7. Experiments in each row from top down: 0 ml, 10 ml, 20 ml, 40 ml, 80 ml, 160 ml, 300 ml of milk in water.





**Figure A.13:** Larger sequence of the plaster experiments shown in Figure 4.8. Experiments in each row from top down: 0 oz, 2 oz, 4 oz, 8 oz, 16 oz, 32 oz, 59 oz of plaster in water.

## A.6 Validation of Convolutional Sparse Coding

This section shows additional experiments which validate our Convolutional Sparse Coding method from Chapter 6. As described in Section 6.5.1, we solve an inpainting problem to verify if our approach can properly predict unknown data from local context. We use the filters learned from the fruit and city database shown in Section 6.4. The experiment in Figure A.14 evaluates reconstruction quality for a dataset consisting of 22 image and, with a single exception, shows improved quality for both filter sets learned on the city and fruit dataset compared to [238, 239].




Image	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
PSNR ours	<b>23.46</b>	<b>25.60</b>	<b>24.64</b>	24.66	<b>30.13</b>	<b>28.10</b>	<b>24.42</b>	<b>24.97</b>	<b>22.07</b>	<b>26.15</b>	<b>25.90</b>	<b>20.76</b>	<b>24.20</b>	<b>24.06</b>	<b>23.60</b>	<b>24.28</b>	<b>22.26</b>	<b>26.03</b>	<b>21.26</b>	<b>28.36</b>	<b>22.89</b>	<b>21.52</b>
PSNR [238, 239]	23.06	24.58	24.28	<b>24.71</b>	29.40	27.27	23.99	24.62	21.79	25.13	25.22	20.50	23.92	23.57	23.37	23.91	21.77	25.74	21.10	27.80	22.74	21.42

Image	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
PSNR ours	<b>23.21</b>	<b>25.35</b>	<b>24.69</b>	24.70	<b>29.88</b>	<b>27.82</b>	<b>24.24</b>	<b>25.44</b>	<b>21.88</b>	<b>26.29</b>	<b>26.05</b>	<b>20.55</b>	<b>24.26</b>	<b>23.80</b>	<b>23.46</b>	<b>24.58</b>	<b>21.90</b>	<b>25.86</b>	<b>21.16</b>	<b>28.11</b>	<b>22.96</b>	<b>21.43</b>
PSNR [238, 239]	22.91	24.68	24.44	<b>24.72</b>	29.33	27.28	23.81	25.21	21.60	25.72	25.58	20.33	23.92	23.44	23.03	24.32	21.55	25.70	21.00	27.72	22.62	21.35

**Figure A.14:** Reconstruction quality for filters learned with the proposed algorithm (tables, center row) and the filters proposed in [238, 239] (tables, bottom row). All reconstructions are performed for 50% subsampling. The upper table shows the reconstruction results with the filters learned from the fruit dataset from [220], the lower one shows the reconstructions with the filters from the city dataset. The dataset consists of 22 images, *none of which are part of the training set for learning the filters*. With the exception of image 4, our algorithm results in higher-quality reconstructions for both filter sets.

## A.7 Non-Sinusoidal Doppler Time-of-Flight Imaging

In this section we expand on the harmonic model from Section A.2, which allows arbitrary waveforms for the sensor and illumination modulation. For moving objects, we can again analyze the Doppler shift, and obtain

$$s(t) = \sum_{k=0}^{\infty} s_k \cos(k(\omega + \Delta\omega)t + \phi_k)$$

and

$$p_\theta(t) \approx \frac{1}{2} \sum_{k=0}^{\infty} f_k s_k \cos(\Delta\omega t + \phi_k - \theta_k). \quad (\text{A.7})$$

Finally, we can use Equation (A.7) to analyze inter-carrier interference in the case of general periodic functions. As in Section 7.3.1, we define the ratio image of a heterodyne and a homodyne image, and obtain:

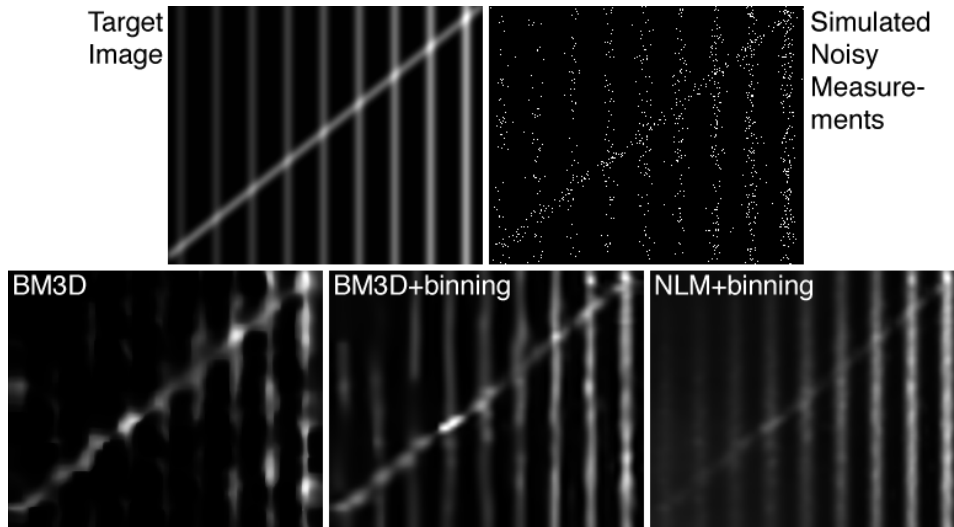
$$r \approx \frac{\int_0^T \frac{1}{2} \sum_{k=0}^{\infty} f_k s_k \cos((\omega_f - \omega_g - \Delta\omega)t + \theta - \phi) dt}{\int_0^T \frac{1}{2} \sum_{k=0}^{\infty} f_k s_k \cos(-\Delta\omega t + \theta - \phi) dt} \quad (\text{A.8})$$

Due to the multiple terms in both numerator and the denominator this equation cannot be simplified such that the dependencies on the phases  $\theta$  and  $\phi$  cancel out. Unlike the sinusoidal model, general periodic functions thus introduce a phase- or depth-dependency to the velocity measurements. These need to be calibrated; see Section 7.4.

## A.8 Velocity Map Denoising

With our system from Chapter 7, we capture an extremely small frequency shift (in the Hz range) relative to the modulation frequency (the MHz range). Additionally, the quantum efficiency of emerging TOF sensors is still far from that of modern solid state sensors [251], and the demodulation contrast drops significantly for high frequencies [317]. Under these conditions, the few photons collected during a measurement are strongly affected by Poisson noise. To account for this, we apply a denoising strategy to all captured velocity maps.

We test several denoising strategies for a synthetic example in Figure A.15. The ground truth test pattern (top left) is synthetically corrupted by Poisson noise (top right). Standard denoising approaches, such as BM3D fail in these conditions, even when variance-stabilizing transforms are applied (bottom left). As recently discussed by Salmon et al. [318], a simple improvement of BM3D results in significant improvements. First, one accumulates the photon counts locally in bins (for example with a size of  $3 \times 3$  pixels) that now have much higher photon counts (but lower spatial resolution). The denoiser is subsequently applied to the binned counts. Finally, the image is upsampled. We show simulated results using this method



**Figure A.15:** Synthetic test pattern (top left) and simulated measurement that is heavily corrupted by Poisson noise (top right). Various modern denoising strategies applied, including BM3D (bottom left), BM3D with binning (bottom center), and non-local means (NLM) with binning (bottom right). To account for the high level of Poisson noise in the measurements, we apply the latter denoising strategy to all captured velocity data.

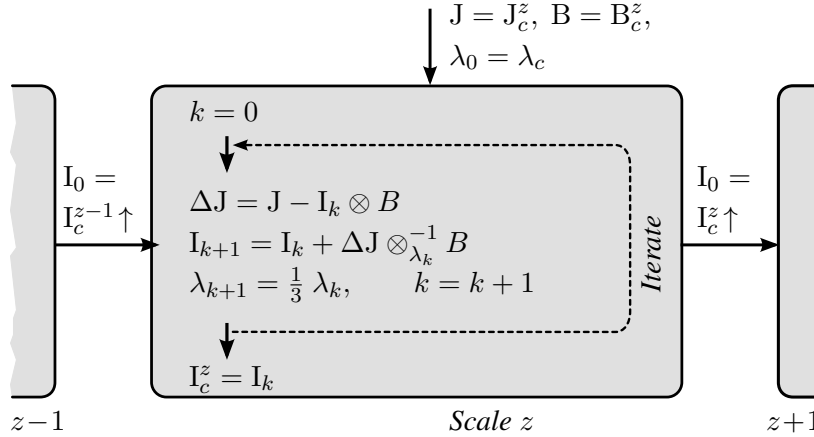
in Figure A.15 (bottom center). Unfortunately, we observe completely missing structures in the image, which is undesirable. Using the binning strategy [318] but replacing BM3D by a non-local means denoiser results in slightly blurrier but overall favorable results (bottom right). We apply this denoising strategy to all results presented in Chapter 7.

## A.9 Residual and Scale-Space Deconvolution

In this section we describe the residual and scale-space deconvolution technique for the basic deconvolution method presented in Section 8.3.2. The residual deconvolution scheme is adopted from [262], and a scale-space implementation has been proposed in [263].

The idea behind residual deconvolution is to iteratively deconvolve the residual image starting with a large regularization weight which is progressively decreased.

Since the residual image has a significantly reduced amplitude, its deconvolved reconstruction contains less ringing which is proportional to the amplitude. The iteration is shown in the center of Figure A.16.



**Figure A.16:** Iterative deconvolution of the residual image which eliminates ringing artifacts, center. The operator  $\otimes_{\lambda_k}^{-1}$  is the deconvolution of the image to the left using the kernel to the right and a regularizer-weight  $\lambda_k$ . For performance reasons the algorithm is implemented in scale space. This figure shows the deconvolution on a level  $z$  in the pyramid. We set  $I_c^{-1} = 0$ .

Our method handles saturation in the blurred image by removing the rows where  $\mathbf{j}$  is saturated from the data fitting term. This is done by pre-multiplying the residual  $\mathbf{B}_c \mathbf{x} - \mathbf{j}_c$  with a diagonal weighting matrix whose diagonal is 0 for saturated rows and 1 else; the derivation from Eq. (8.11) is changed straightforwardly.

To increase the performance of the algorithm by using good starting points for Eq. (8.5), the method is performed in scale space. See again Figure A.16, where  $\uparrow$  is an upsampling operator to the next finer scale. We use nearest neighbor since it preserves edges. The pyramids  $\{J^z\}_{z=0}^Z, \{B^z\}_{z=0}^Z$  of the blurred image/kernel pairs are computed by bicubic downsampling of  $J, B$  with the scale factor  $\frac{1}{2}$ . The reconstruction pyramid  $\{I^z\}_{z=0}^Z$  is progressively recovered from coarse (scale 0) to fine, where at each scale the initial iterate is the upsampled result of the next coarser level. Note, that our scale space implementation purely serves as a performance improvement. In particular we do not need to overcome local minima in our

minimization problem since it is convex.

Contrary to [262], we are also not using any information from coarser scales in the deblurring at a considered scale. Since the reconstructions can contain significantly less detail we found that guiding fine scale deblurring with coarser scale information is problematic in many cases.

## A.10 PSF Estimation using Calibration Targets

This section explains the PSF estimation method for our imaging system from Chapter 8, which uses uncompensated simple optics instead of complex lens systems for high-quality imaging. The aberrations for these simple optics are spatially varying as shown in Figure 2.6 and Figure 8.2. However, for a small tile the corresponding PSF can be assumed to be spatially invariant, see Section 8.2, and the deconvolution method from Section 8.3 is formulated for small tiles.

While the PSF for a tile can be obtained by any existing technique, we use a calibration-pattern-based approach. In contrast to methods that directly measure the PSF like Schuler et al. [165], no pinhole-light source and a dark-room lab is necessary. Instead we just use a consumer laser-printer to make our targets. To estimate the PSFs from the target images, it is natural to apply the same optimization framework that was used for deblurring in Chapter 8 also for the PSF estimation step.

### PSF Estimation as Deconvolution

The PSF estimation problem can be posed as a deconvolution problem, where both a blurred image and a sharp image of the same scene are given. The blurred image is simply the scene imaged through the simple lens, with the aperture open, while the sharp image can be obtained by stopping the lens down to a small, almost pinhole aperture, where the lens aberrations no longer have an effect. By acquiring a sharp image this way (as opposed to a synthetic sharp image) we avoid both geometric and radiometric calibration issues in the sharp reference image.

Let  $J$  be an image patch in a considered blurred channel,  $I$  the corresponding sharp pinhole-aperture patch. We estimate a PSF  $B_{\text{opt}}$  describing the blur in  $J$  by

solving the minimization problem

$$\mathbf{b}_{\text{opt}} = \underset{\mathbf{b}}{\text{argmin}} \quad \|\mathbf{I}\mathbf{b} - s \cdot \mathbf{j}\|_2^2 + \lambda \|\nabla \mathbf{b}\|_1 + \mu \|\mathbf{1}^T \mathbf{b} - 1\|_2^2, \quad (\text{A.9})$$

where the first term is a linear least-squares data fitting term, and the scalar  $s = \sum_{k,l} I(k,l) / \sum_{k,l} J(k,l)$  accounts for the difference in exposure between the blurred and pinhole image. The second term represents a standard TV prior on the gradients of the recovered PSF, and the third term smoothly penalizes deviations from the energy conservation constraint, i.e.,  $\sum_{k,l} B(k,l) = 1$ .

Note that Eq. (A.9) is a convex optimization problem. To solve it we use a proximal algorithm which is similar to the deconvolution method from Section 8.3.3. Specifically, Eq. (A.9) is expressed in the form of Section 2.8.3, using:

$$\begin{aligned} \Phi &= \mathbb{I}, & G(\mathbf{v}) &= \frac{1}{\lambda} \|\mathbf{I}\mathbf{v} - s \cdot \mathbf{j}\|_2^2 + \frac{\mu}{\lambda} \|\mathbf{1}^T \mathbf{v} - 1\|_2^2, \\ \Omega &= \nabla, & F(\mathbf{v}) &= \lambda \|\mathbf{v}\|_1. \end{aligned} \quad (\text{A.10})$$

As in Section 8.3.3, we use Chambolle-Pock's proximal algorithm to solve Eq. (2.40) with the specific operators from Eq. (A.10). The proximal operators  $F^*$  has already been defined in Eq. (8.7), and the proximal operator for  $G$  is

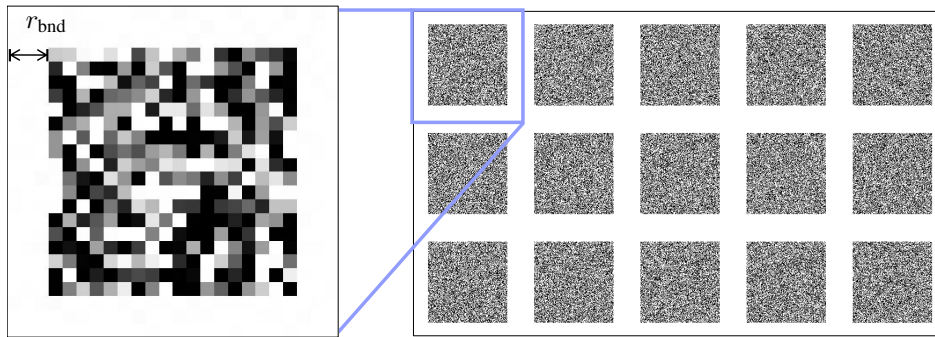
$$\begin{aligned} \text{prox}_{\tau G}(\mathbf{v}) &= \tilde{\mathbf{x}} = \underset{\mathbf{x}}{\text{argmin}} \quad \frac{1}{2\tau} \|\mathbf{x} - \mathbf{v}\|_2^2 + \frac{1}{\lambda} \|\mathbf{I}\mathbf{x} - s \cdot \mathbf{j}\|_2^2 + \frac{\mu}{\lambda} \|\mathbf{1}^T \mathbf{x} - 1\|_2^2 \\ &\Leftrightarrow \frac{\lambda}{2} \tilde{\mathbf{x}} - \frac{\lambda}{2} \mathbf{v} + \tau (\mathbf{I}^T \mathbf{I} \tilde{\mathbf{x}} - s \cdot \mathbf{I}^T \mathbf{j}) + \tau \mu (\mathbf{O} \tilde{\mathbf{x}} - \mathbf{1}) = 0 \\ &\Leftrightarrow \tilde{\mathbf{x}} = \mathfrak{F}^{-1} \left( \frac{\tau s \mathfrak{F}(\mathbf{I})^* \mathfrak{F}(\mathbf{j}) + \frac{\lambda}{2} \mathfrak{F}(\mathbf{v}) + \tau \mu \mathfrak{F}(\mathbf{1})}{\tau |\mathfrak{F}(\mathbf{I})|^2 + \frac{\lambda}{2} + \tau \mu \mathfrak{F}(\mathbf{O})} \right) \end{aligned} \quad (\text{A.11})$$

The computation of  $L$  for Chambolle and Pock's method has been described in Section 8.3.3, and the same  $\tau$  and  $\sigma$  can be used.

## Calibration Pattern

To allow for robust PSF estimation, the scene used for this purpose should have a broad spectrum. We therefore decide against both natural scenes and edge based patterns (e.g., [319, 320]), and instead adopt a white noise pattern. See [321] for

a comparison of a white-noise pattern to different other calibration patterns. Our specific pattern, shown in Figure A.17, is subdivided into several noise patches, separated by a white border. Each noise tile will produce a PSF for a single image tile, while the white frame helps suppress boundary effects in the PSF estimation process. This is because it enables a physically correct constant boundary condition around each patch if the frame width is  $r_{\text{bnd}} \geq \text{blur-radius}$  (see again Figure A.17).



**Figure A.17:** Proposed PSF calibration target consisting of a grid of framed white noise patches. If the frame width is  $r_{\text{bnd}} \geq \text{blur-radius}$ , a boundary condition propagating the boundary pixels outside is a physically correct model for the blurred and sharp noise patch.

### Full PSF Estimation Framework

To increase the efficiency of the PSF estimation, we apply the algorithm introduced above in scale space. We initialize our iterative minimization at each scale with the upsampled results from the next coarser scale, which yields a good starting point for the convex objective function from Eq. (A.9) and thus speeds up the minimization.

After the initial PSF estimation, we perform an additional step of smoothing by computing weighted averages of PSFs for a  $3 \times 3$  set of neighboring tiles. Although the PSFs may contain high frequency features, these tend to change smoothly over the image plane. Combined with the relatively small tile size, we found that this spatial filtering does not cause feature loss, but reduces noise significantly.



## A.11 Cross-Channel Deconvolution for Various Optical Systems

This section presents additional results that evaluate our method from Chapter 8 for various optical systems, and in comparison to other deconvolution algorithms.

### Complex Lens Systems

Figure A.18 shows an additional result for commercial camera lens, a Canon 28–105 mm zoom lens at  $f/4.5$ , which has been discussed in Section 8.4.

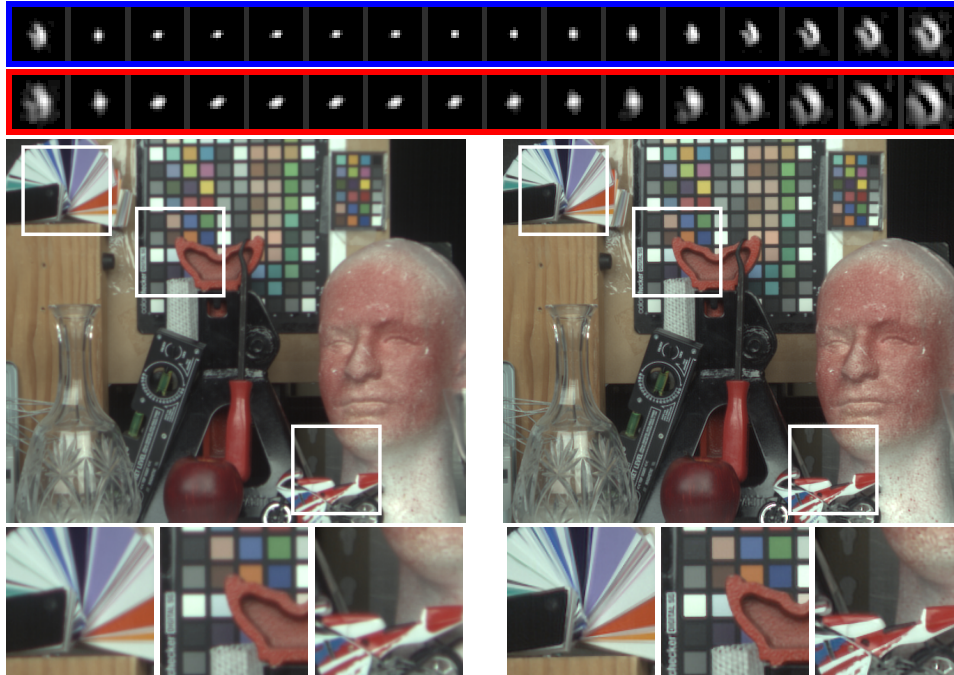


**Figure A.18:** Patches from an image captured with a commercial Canon lens (105 mm,  $f/4.5$ ). First and third column: Captured input patches. Second and last column: reconstructed sharp result.

### Multispectral Camera

Figure A.19 shows results for a multi-spectral camera with custom multi-element optics. The camera prototype is described in [322]. Using the Liquid crystal tunable filter (LCTF) we capture narrow spectral bands whose center wavelength can be electronically tuned. See [323] for a review of LCTF-based multi-spectral cameras. The prototype system allows for 20 nm wavelength increments in the spectral range from 420 to 720 nm.

The results shown in Figure A.19 demonstrate that, as for conventional cameras,

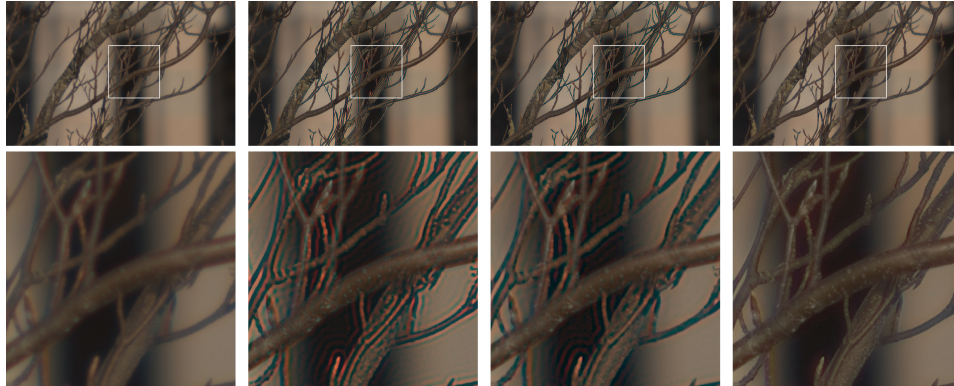


**Figure A.19:** Results for a multispectral camera with custom optics. The top row shows PSF estimation results for 20 nm wavelength increments in the spectral range from 420 to 720 nm (blue: image center, red: upper left corner). The bottom set of images and insets shows the captured image (left) and the deconvolved result (right) after mapping to sRGB space.

our method successfully removes chromatic aberration and restores lost image detail in the blurred channels over the full range of the visible spectrum.

### Comparisons

We have compared our method against several state-of-the-art deconvolution algorithms, including the one from Levin et al. [161]. Figure A.20 shows a comparison. Compared to our result, it shows more prevalent ringing artifacts, which are particularly noticeable because they also have chromatic component. Our cross-channel prior successfully suppresses chromatic artifacts, but also helps to transfer information between channels for reduced ringing and increased detail overall. For the



**Figure A.20:** Top row from left to right: Captured image, deconvolved result from [161] independently on each channel, IRLS method from [161] extended with our prior, our result. Bottom row: Patches in same order of the methods.

result in Figure A.20 we also implemented our cross-channel prior as an additional regularization term for Levin et al.’s method. While this improves the result, it does not match the quality of our method. The IRLS solver is unstable, not guaranteed to converge, does not minimize the sparse norm close to its non differentiable origin, and thus cannot achieve the results of our method. We especially found robustness to be a general issue once the cross-channel prior was added to IRLS, with the resulting method being very sensitive to parameter selection and often failing completely. We have also compared our method to Schuler et al. [165]. First, in Figure A.21 we show results using their datasets (images and estimated PSFs). We show results with two different weights for the regularizer. Note that in *both* cases, our method recovers more fine-scale structure than their approach. Since Schuler et al. use a Hyper-Laplacian non-convex objective, their method is not guaranteed to converge to the desired accuracy. However, the results for this dataset do not match the quality we can achieve with our own datasets – for varying regularization weight always a strong residual blur remains, especially around the leaves and window edges. These are due to PSF estimation errors rather than issues in the actual deconvolution method. We note that the PSFs lack a very low-frequency component, which we have observed with every single lens element we have tested in our setup. Instead, their blur kernels exhibit very thin structures akin to motion blur kernels and contain



**Figure A.21:** Top row from left to right: Captured image, our deconvolved result, result from [165]. Patch matrix below from top left to bottom right: Captured image, result from [165], our result with strong regularization (smoother reconstruction), our result with low regularization (more detailed reconstruction).

very little chromatic variation. A missing low-frequency component in the kernels explains the poor quality of the deconvolved results. Unfortunately, Schuler et al. do not specify details about the lens, so we are unable to obtain similar images using our setup.

We note that Schuler et al.’s YUV-cross-channel regularization *breaks down when the chromatic aberrations increase*. It completely fails for many of the large aberrations that we can handle. In Figure A.21 and Figure A.22 we compare our cross-channel prior against their YUV-cross-channel regularization. In this case we did not have access to their original implementation, so we reimplemented their method. To solve Eq.(5) from their paper, we have adapted the code from the Hyper-Laplacian solver [160] as suggested by the authors. All parameters documented in their paper have been used. Note that our prior removes nearly all chromatic aberrations, while Schuler’s YUV regularization fails with the large aberrations that we can handle.





**Figure A.22:** Images from Figure 8.6, Figure 8.1 reconstructed using our cross-channel prior (left) and restored with the YUV-regularization from [165] (right).

## A.12 Natural Image Prior Design Choices

This section discusses the design choices for the natural image priors from Eq. (9.3) used in our Bayesian approach to camera image processing from Chapter 9.

### Choice and Importance of Priors

The choice and combination of priors is crucial for high-quality results [324]. The priors can be divided into two main categories: *internal* priors use intra-image information, while *external* priors use external knowledge about natural images. While most priors fall in one or the other category, a few have mixed characteristics, so the boundary is somewhat vague. We have experimented with many internal and external priors to yield a combination that achieves the highest quality but remains efficient to compute. We next analyze the internal and external priors separately.

### Choice of External Priors

We considered three different external priors (in addition to the complementary cross-channel prior which has been analyzed in the previous Chapter 8): the simple TV prior [188], a curvelet prior [325], and the EPLL prior [163]. We explore their use for reconstructing interlaced HDR (Section 9.4) images with a simulated lens blur (Gaussian, varying  $\sigma$ ) using a synthetic dataset consisting of 12 different images. This dataset provides ground truth and helps to determine which priors are adequate. We set the internal prior to BM3D and reconstruct the 12 images by varying the weights between the internal (BM3D) and the three investigated external priors. The reconstruction quality is listed in Table A.1.

The complex EPLL prior can give a slight image quality increase (f) in the case of a large blur ( $\sigma = 1.83$ ), but its computational cost is prohibitive (60 times higher than TV). Just using TV gives a similar boost. For a smaller lens blur ( $\sigma = 0.14$ ), EPLL also only yields a minor increase in image quality, but larger gains are achieved by the much cheaper TV prior. The curvelet prior does not provide any benefit for  $\sigma = 1.83$ ; on the contrary, the quality decreases slightly with its use. For  $\sigma = 0.14$ , the curvelet prior can achieve gains, but is quite sensitive to the chosen weighting. Furthermore, using the curvelet prior is about 15% slower than using TV.

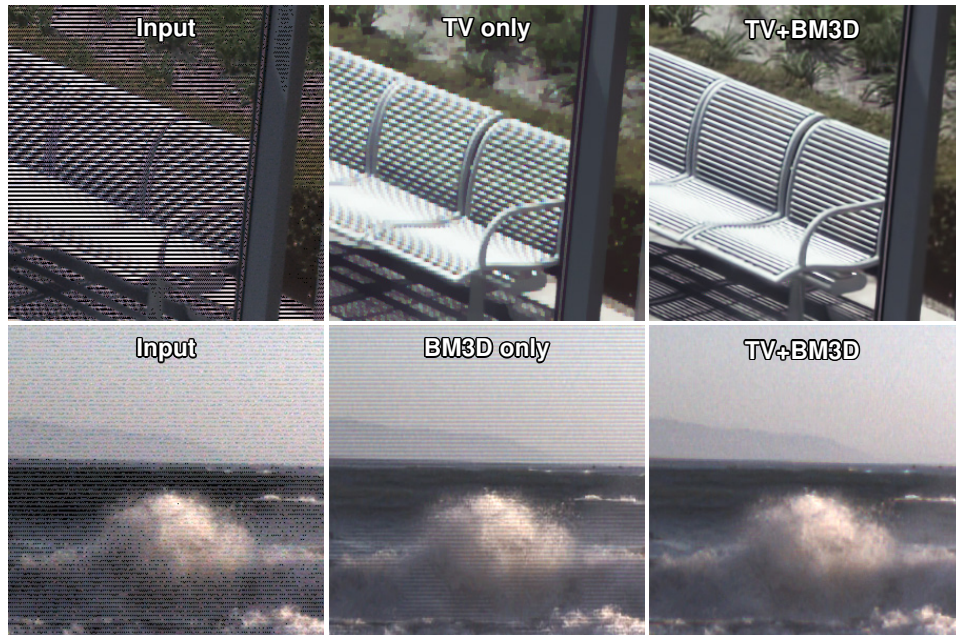
	(a)	(b)	(c)	(d)	(e)	(f)	(g)
$\phi_0^{\text{ext.}}$	0	1/32	1/16	1/4	1/2	1/2	1
$\phi_2^{\text{BM3D}}$	1	1	1	1	1	1/2	0
EPLL ( $\sigma = 1.83$ )	0.00	+0.01	0.00	0.00	-0.03	+0.07	-0.34
Curvelets ( $\sigma = 1.83$ )	0.00	-0.02	-0.01	-0.41	-1.03	-0.90	-1.78
TV ( $\sigma = 1.83$ )	0.00	+0.03	+0.04	+0.05	-0.02	-0.19	-0.61
EPLL ( $\sigma = 0.14$ )	0.00	0.00	+0.03	+0.04	+0.07	+0.05	-0.82
Curvelets ( $\sigma = 0.14$ )	0.00	-0.02	-0.02	-0.25	+0.12	-0.05	-0.40
TV ( $\sigma = 0.14$ )	0.00	+0.04	+0.13	+0.16	+0.02	-0.39	-0.95
TV ( $\sigma = 1.83$ )	0.00	+0.03	+0.04	+0.05	-0.02	-0.19	-0.61
TV ( $\sigma = 1.38$ )	0.00	+0.05	+0.10	+0.12	+0.03	-0.18	-0.79
TV ( $\sigma = 1.00$ )	0.00	+0.05	+0.12	+0.14	+0.05	-0.21	-1.12
TV ( $\sigma = 0.55$ )	0.00	+0.02	+0.05	+0.10	0.00	-0.40	-1.23
TV ( $\sigma = 0.14$ )	0.00	+0.04	+0.13	+0.16	+0.02	-0.39	-0.95

**Table A.1:** Different weighting between internal and external prior and the resulting change in SNR. To analyze the effect isolated from the cross-channel prior we set  $\phi_1 = 0$ .

We note that the benefit of using TV in addition to BM3D lies in making the problem more convex. To further illustrate this, we show results for five different Gaussian blurs (bottom of Table A.1). Smaller blurs make the data term less convex, thus making the problem successively more difficult to solve by only using BM3D. Adding TV convexifies the problem, and image quality increases.

Figure A.23 demonstrates the complementary benefits of using BM3D and TV. The top row illustrates that by exploiting structural self-similarity (BM3D) we can reconstruct significantly more detail than just by using a TV prior. The bottom row illustrates that the BM3D prior may fail for correlated noise or reconstruction artifacts, but those problems can be fixed by the external TV prior.

We conclude that TV is the most cost-effective external prior giving similar performance as much more computationally complex priors and can help to increase convexity. In all our applications, a small amount of TV has improved the resulting reconstruction quality.



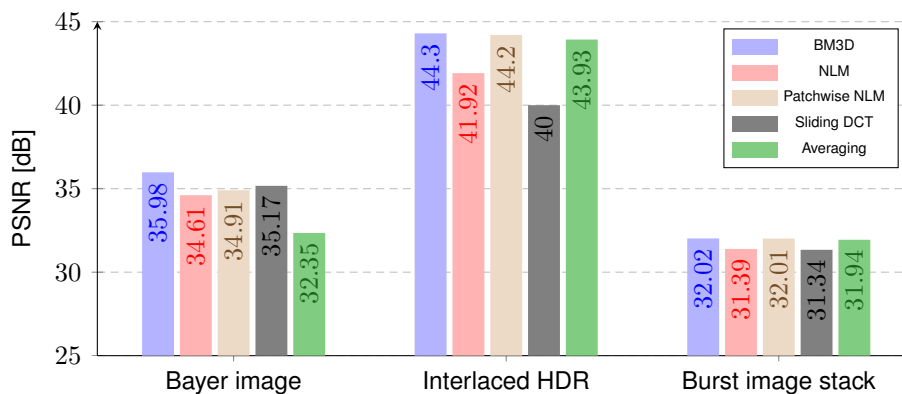
**Figure A.23:** Illustration of using both the TV and BM3D priors for reconstructing images from the interlaced HDR sensor (Section 9.4).

### Choice of Internal Priors

We showed in Section 9.3.2 that we can incorporate *any* Gaussian denoiser into our framework. We are particularly interested in *internal* denoising priors that exploit self-similarity within the image [67, 70]. We experimented with the most-commonly-used non-local operators: BM3D [70], NLM [67], patchwise NLM [326], Sliding DCT [69], and simple averaging of similar patches, which we compared for demosaicking, interlaced HDR, and burst image stack processing in Figure A.24.

All denoisers except Sliding DCT operate on a stack of similar patches. For this experiment, we configured all of them to search for the 16 most-similar  $8 \times 8$  patches in a  $15 \times 15$  pixel vicinity. BM3D performs a 3D transform (DCT on  $xy$  and Haar on  $z$ ) on this stack and thresholds all coefficients followed by Wiener filtering; NLM computes a weighted average of the patch centers based on patch similarity; patchwise NLM computes the weighted average for all  $8 \times 8$  pixels; simple averaging assigns a uniform weight to all patches. Sliding DCT simply





**Figure A.24:** The impact of selected natural-image priors on the reconstruction quality of various image models.

thresholds DCT-transformed patches for each pixel and aggregates the results. We only use the denoising prior for this comparison.

Figure A.24 shows the resulting PSNR values for each denoiser. Overall, BM3D and patchwise NLM achieve very similar results, with BM3D being slightly better in all applications. Hence, all results in this paper were generated using BM3D. Simple averaging works well for reconstructing interlaced HDR images and burst denoising, but not for Bayer demosaicking. Pure NLM achieves reasonable PSNR values in all applications, but is always worse than BM3D and patchwise NLM. Sliding DCT achieves fairly good PSNR for demosaicking and burst denoising, but not when applied to interlaced HDR with a significant number of pixels being either too dark or bright. Since significant information is missing in interlaced HDR, a self-similarity prior is needed to recover it.

*BM3D modifications* One can expect improved results by modifying the BM3D parameters for a given application, and we experimented with this for demosaicking. Note that we only varied the prior parameters (just like the data-term weights for different applications), but not the prior itself. We found that two changes to the standard BM3D parameters both improve the running time and increase the result image quality. The first parameter is the size of the patch to be matched. Instead of using the standard  $8 \times 8$  patch, we use much smaller  $4 \times 4$  patches. Obviously,

smaller patches are faster to compare, but more importantly, it is likely that we can find more patches that are similar. This is particularly important in areas with irregular textures, such as foliage or grass.

The second parameter relates to the color spaces that we operate in, and the 2D transformations we use. In the first stage, when patches are matched, the modified version uses the YUV color space, and matches patches based on the luminance component only; we use DCT as the 2D spatial transform to sparsify the signal, and 1D Haar for thresholding. For the second stage, Wiener filtering, we use the 3-point DCT to decorrelate colors and then use DST (Discrete Sine Transform) as the 2D spatial transformation, and again Haar for thresholding. Use of different color spaces and transformations decorrelates the two processing steps, yielding improved results. The average improvement is +1.73 dB in pure demosaicking and +0.6 dB in joint demosaicking and denoising over the default BM3D. Please note that, *even with the default settings, our technique still beats the competing methods.*

### **Choice of Weights**

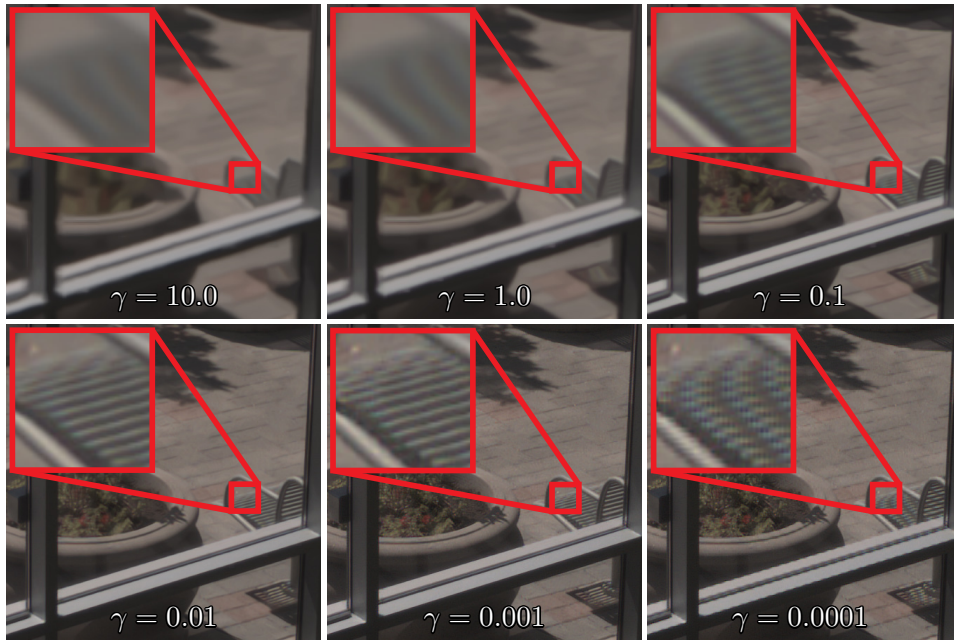
We found that a single setting for the prior weights is sufficient for each application—no image-dependent modifications are necessary. Furthermore, we need to find the right overall weighting between the data term and the priors, as illustrated in Figure A.25. We experimented with different weights, and again found that a single setting per application works well for any given image. User preferences such as sharp but grainy vs. smooth with less noise, do influence the settings.

## **A.13 Bayesian Color Arrays Imaging and Deblocking**

This section presents further application of our Bayesian approach to camera image processing from Chapter 9 to color array cameras. In particular, we demonstrate color array cameras and JPEG deblocking.

### **Color Array Camera**

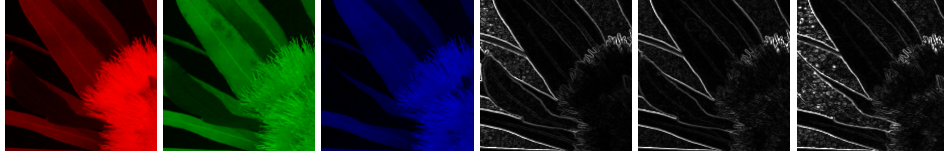
Single-chip array cameras provide thin camera designs and a potential for depth-map-based applications [73]. The color filters are per lens and not per pixel, and each lens can be optimized for the wavelength passed by the respective filter. Since



**Figure A.25:** Impact of the regularizer weights (relative to the data term) on the solution. The weights are  $\gamma_i = \gamma \phi_i$ :  $\gamma$  is shown in the images, while  $\phi_i$  are  $\phi_0 = 0.1$  (TV),  $\phi_2 = 1.0$  (BM3D), and  $\phi_1 = 0.1$  (cross-channel). High  $\gamma$ -values impair the reconstruction process by enforcing too much sparsity in the output. On the other hand, a too small  $\gamma$  ignores the priors, leading to sharp but aliased results.

each sub-image has only one color, no demosaicking is required. Instead, the images have to be registered, but this is challenging since each sub-image has a different color and viewpoint, leading to parallax and occlusions.

We use our framework to fuse images from a  $2 \times 2$  single-chip color array camera, shown in Figure 9.1(d). This prototype uses R, G, and B sensors, with a 1.4mm baseline. Additionally, we use a second  $2 \times 2$  camera array. We first estimate the registration between the sub-images, which we encode in the matrix  $\Phi$ , and use the matrix  $M$  as a confidence mask modeling the uncertainty in this registration. This uncertainty occurs both in flow-based or depth-based registration. We have opted for a novel cross-channel optical flow for registration across different color channels, as it avoids any geometric calibration. Each channel is replaced with a



**Figure A.26:** R, G, and B color planes, and corresponding normalized images. The normalized gradient magnitude images look similar, apart from noise. Prominent edges show up on all channels, helping the optical flow algorithm to converge on a good motion estimate.

normalized image  $\tilde{\mathbf{j}} = \frac{\|\nabla \mathbf{j}\|}{\mathbf{j} + \epsilon}$  (A.26), and we solve for the cross-channel optical flow  $\Delta v$  between the color channels  $\mathbf{j}_C$  (for  $C = \{R, B\}$ ) and the fixed green channel  $\mathbf{j}_G$  with

$$\operatorname{argmin}_{\Delta v} \left\| \tilde{\mathbf{j}}_C + (\nabla \tilde{\mathbf{j}}_C)^T \Delta v - \tilde{\mathbf{j}}_G \right\|_2^2 + \|\Delta v\|_1 \quad (\text{A.12})$$

using a standard optical flow algorithm [327].

We detect the reliable flow estimates through forward-backward consistency check and mark them in the indicator matrix  $\mathbf{M}$ . Having computed the flows  $\Delta v_R$  and  $\Delta v_B$ , we form flow matrices  $\mathbf{W}_i$  for each capture  $i$ . We also account for the optical blur in the  $2 \times 2$  camera and recover some of the sharpness in the reconstructed image. The blur is encoded in the matrix  $\mathbf{B}$  as before. The mosaicking matrix is not needed for this camera.

We set  $\Phi_i = \mathbf{M}_i \mathbf{B} \mathbf{W}_i$  for each capture  $i$  (out of  $k$ ), and get

$$G(\mathbf{i}) = \sum_{i=1}^k \|\mathbf{B} \mathbf{W}_i \mathbf{i} - \mathbf{j}_i\|_{\mathbf{M}_i}^2. \quad (\text{A.13})$$

## Results

We perform the experiment on both a large and a small baseline design for a  $2 \times 2$  color camera array. In Figure A.27 we show the large-baseline (top) and the small-baseline results (bottom). The first column shows an overlay of the different channels to illustrate the extent of chromatic aberration due to the parallax. The second column shows the images warped by the computed cross-channel flow. Note that there are still color artifacts in the occluded and poorly-estimated flow regions. In the third column we show the result by a method adapted from Joulin and



**Figure A.27:** Image reconstruction for color-array camera. **(a)** Raw input red, green, and blue images overlaid. **(b)** Naïve registration of color channels via optical-flow produces color leaks. **(c)** Anaglyph-based reconstruction [328] partially reduces the color leakage. **(d)** Our method reconstructs the image without such artifacts.

Kang [328], which was originally proposed to reconstruct full color stereo images from anaglyph (red/cyan) images. This method works by iteratively computing SIFTFlow [329] between the channels, detecting poorly matching regions through flow consistency check, and then colorizing these regions. But the method does not enforce the cross-channel edge consistency illustrated in Figure A.27, and produces visible artifacts. In the fourth column we show that our combination of priors manages to maintain the consistency of R and B channels with the G channel in just a single pass and avoids color fringing. While some small artifacts remain, this problem is hard and we believe we have made a large step towards a robust solution. *Note that we do not require a perfect registration*, but rely on our image priors to refine the effective registration. This is an important advantage, because registration errors are unavoidable—whether images are registered via optical flow or via depth estimates [73]—and may lead to very visible color bleeding at occlusion edges (Figure A.27). Our conclusion is that the proposed framework is suitable for processing color camera images, even without first extracting per-pixel depths.

### Beyond Linear RGB

We next demonstrate that we do not need to stop at linear RGB, or YUV, but that we can extend our processing all the way to the JPEG-compressed image.

JPEG compression works by taking an RGB image  $\mathbf{i}$ , converting it to YUV space with a linear color transform  $\mathbf{E}$ , downsampling the chroma components ( $\mathbf{S}_J$ ), performing a block-based DCT on the individual channels ( $\Phi$ ), and reweighting the individual frequencies according to a quantization matrix  $\mathbf{Q}$  [330]. Finally, the integer JPEG coefficients are obtained through rounding:

$$[\mathbf{c}] = \text{round}(\underbrace{\mathbf{Q}\Phi\mathbf{S}_J\mathbf{E}}_{\mathbf{R}} \mathbf{i}). \quad (\text{A.14})$$

We can see that all steps except for the final quantization can be expressed as a linear operator. Likewise, JPEG decompression can be expressed as a linear operator  $\mathbf{R}^{-1} = \mathbf{E}^{-1}\mathbf{U}_J\Phi^{-1}\mathbf{Q}^{-1}$ , where the upsampling operator  $\mathbf{U}_J = \mathbf{S}_J^\dagger$  is a pseudo-inverse of the chroma subsampling.

With this observation it becomes possible to directly perform image reconstruc-

tion in the JPEG transform space, by setting  $\Phi = \mathbf{MSBR}^{-1}$ , as below:

$$\operatorname{argmin}_{\mathbf{c}} \quad \|\mathbf{SBR}^{-1}\mathbf{c} - \mathbf{j}\|_{\mathbf{M}}^2 + \Gamma(\mathbf{R}^{-1}\mathbf{c}), \quad (\text{A.15})$$

where  $\Gamma(\mathbf{R}^{-1}\mathbf{c})$  is simply  $F(\Omega\mathbf{R}^{-1}\mathbf{c})$ .

Hence, we can add this compression-space optimization to any image-formation model and priors, directly optimizing for the best JPEG DCT coefficients  $\mathbf{c}$ . If we do not want to optimize all the way to the JPEG coefficients, but only to a YUV or other color space representation, we can set  $\Phi = \mathbf{MSBE}^{-1}\mathbf{U}_J$ .

Note that we can use a similar approach for JPEG image decompression in an image viewer. Depending on the quantization matrix  $\mathbf{Q}$ , JPEG-compressed images may exhibit severe artifacts, including blocking, ringing, and loss of texture. We can use our usual image priors to alleviate these artifacts by using  $\Phi = \mathbf{R}$  to get

$$G(\mathbf{i}) = \|\mathbf{R}\mathbf{i} - \mathbf{c}\|_2^2, \quad (\text{A.16})$$

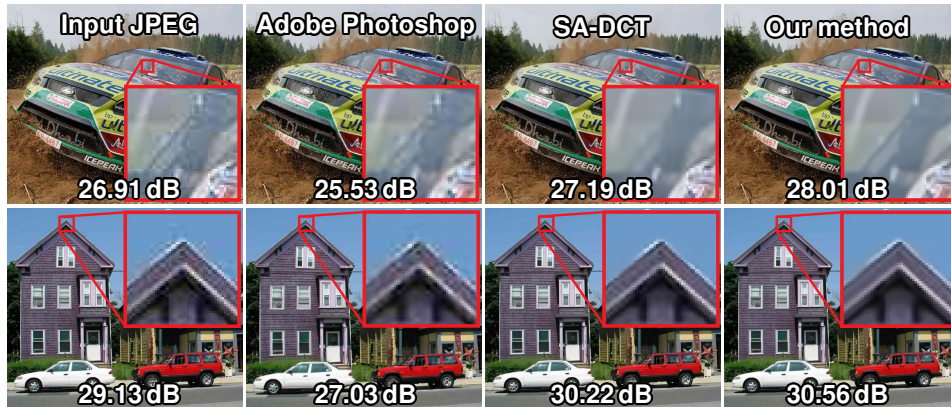
where  $\mathbf{c}$  are the JPEG coefficients, and  $\mathbf{i}$  the reconstructed image. Intuitively, using Eq. (A.16) as the data term in our minimization scheme determines the image that best matches the image priors while still compressing to the given JPEG coefficients  $\mathbf{c}$ . In effect, we are deblocking the displayed JPEG image [331].

## Results

To evaluate our joint optimization with JPEG’s DCT encoding, we have reconstructed 12 images from the interlaced HDR camera application. We first compare our joint optimization by directly reconstructing these images in YUV420 vs. first reconstructing them as full RGB and then subsampling and converting to YUV420. On these 12 images, our method achieves an average PSNR of 28.83 dB, in contrast to the pipeline approach, which only achieves 28.45 dB.

We also compare JPEG images reconstructed directly from the interlaced HDR inputs by our framework (quality 50) to the standard pipeline approach of first reconstructing in full RGB, followed by regular JPEG encoding (using the quality factor that best matches the file size of the first JPEG). PSNR values of the decoded JPEG images are then computed w.r.t. the ground-truth image. On average, we





**Figure A.28:** Extending the reconstruction pipeline to the receiver. We compare the JPEG (quality = 30), Adobe Photoshop deblocking, SA-DCT, and our reconstruction on the display side (PSNR in images).

achieve an improvement of 0.1 dB over the sequential approach.

The improvement is modest, as the JPEG coefficients do not couple much with the rest of the pipeline. However, strong JPEG compression inevitably creates blocking and quantization artifacts, which this approach cannot address. To overcome such issues, we can extend the image pipeline all the way to decompression and solve Eq. A.16 on the receiver side, in essence deblocking the JPEG image [331]. We compare our method to Adobe Photoshop CC’s deblocking algorithm and SA-DCT [331] in Figure A.28. Our method removes most of the blocking artifacts and ringing, and yields a significant PSNR improvement ( $\sim 1.0$  dB). Photoshop removes some artifacts, but does not reconstruct the original image as faithfully as our method does. SA-DCT achieves good quality, but 0.3–0.6 dB worse than ours.

## A.14 Poisson Data Fidelity

This section describes the data fidelity term resulting from a Poisson likelihood which models Poisson distributed additive noise. In particular, we model the observed image  $\mathbf{j}$  as a sample of a random variable  $\tilde{\mathbf{j}}$ :

$$p(\tilde{\mathbf{j}} = \mathbf{j} \mid \lambda) = \prod_{l=1}^n \frac{\lambda_l^{j_l} e^{-\lambda_l}}{j_l!}, \quad (\text{A.17})$$



where here the notation  $(\cdot)_l$  denotes the selection of the  $l$ -th component of the image vector given as argument. Following the Bayesian MAP approach from Section 2.8.1, which is also proposed in [305] and [306], the quadratic dataterm  $\|\Phi\mathbf{i} - \mathbf{j}\|_2^2$  then becomes the negative log-likelihood of  $p(\tilde{\mathbf{j}} = \mathbf{j} \mid \mathbf{i})$ , that is

$$-\log \left( p(\tilde{\mathbf{j}} = \mathbf{j} \mid \mathbf{i}) \right) = \sum_{l=1}^n \Gamma((\Phi\mathbf{i})_l, \mathbf{j}_l) \quad \text{with} \quad (\text{A.18})$$

$$\Gamma(a, b) = a - b \log(a) + \text{ind}_{\mathbb{R}^+}(a),$$

where  $\text{ind}_{\mathbb{R}^+}(a)$  is the indicator function for the positive orthant. Setting now the negative log-likelihood as

$$f_{\text{Poisson}}(\mathbf{x}) = \mathbf{x} - \mathbf{j} \log(\mathbf{x}) + \text{ind}_{\mathbb{R}^+}(\mathbf{x}), \quad (\text{A.19})$$

then, the corresponding proximal operator for the changed  $f_1$  is

$$\mathbf{prox}_{\theta f_{\text{Poisson}}(\cdot)}(\mathbf{v}) = \frac{\mathbf{v} - \theta}{2} + \sqrt{\theta \mathbf{j} + \frac{(\theta - \mathbf{v})^2}{4}}, \quad \text{Poisson Prox} \quad (\text{A.20})$$

This analytic solution for the proximal operator of  $f_1$  results in a root-finding problem of a second-order polynomial as shown in Eq. (A.21). Due to the positivity constraint in  $f_1$  the minimum is uniquely defined by the positive root

$$\begin{aligned} \mathbf{prox}_{\theta f_1(\cdot)}(\mathbf{v}) &= \mathbf{x}_{\text{opt}} = \underset{\mathbf{x}}{\text{argmin}} \quad f_1(\mathbf{x}) + \frac{1}{2\theta} \|\mathbf{x} - \mathbf{v}\|_2^2 \\ &= \underset{\mathbf{x} \in \mathbb{R}^+}{\text{argmin}} \quad \underbrace{\mathbf{x} - \mathbf{j} \log(\mathbf{x}) + \frac{1}{2\theta} \|\mathbf{x} - \mathbf{v}\|_2^2}_{\Upsilon(\mathbf{x})} \\ &\Leftrightarrow \frac{\partial \Upsilon(\mathbf{x}_{\text{opt}})}{\partial \mathbf{x}} = \mathbf{1} - \frac{\mathbf{j}}{\mathbf{x}_{\text{opt}}} + \frac{1}{\theta} \mathbf{x}_{\text{opt}} - \frac{1}{\theta} \mathbf{v} \stackrel{!}{=} 0 \quad \text{s.t.} \quad \mathbf{x}_{\text{opt}} \in \mathbb{R}^+ \\ &\Leftrightarrow \mathbf{x}_{\text{opt}}^2 + (\theta - \mathbf{v}) \cdot \mathbf{x}_{\text{opt}} - \theta \mathbf{j} = 0 \quad \text{s.t.} \quad \mathbf{x}_{\text{opt}} \in \mathbb{R}^+ \\ &\Leftrightarrow \mathbf{x}_{\text{opt}} = \frac{\mathbf{v} - \theta}{2} + \sqrt{\theta \mathbf{j} + \frac{(\theta - \mathbf{v})^2}{4}}. \end{aligned}$$