On Natural Parametrizations of Motion and Structure Manifolds

Guy Rosman

On Natural Parametrizations of Motion and Structure Manifolds

Research Thesis

In Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy

Guy Rosman

Submitted to the Senate of the Technion —
Israel Institute of Technology

Elul 5713 Haifa August 2013

The Research Thesis Was Done Under The Supervision of Prof. Ron Kimmel

in the Faculty of Computer Science at the Technion

Acknowledgements

I would like to thank all the wonderful people who have accompanied me on this journey from our lab, faculty and the Technion, as well as other universities – you have made this journey an exciting and pleasurable one.

I would like to especially thank my advisor, prof. Ron Kimmel, for guidance throughout the years, and for teaching me so much on so many topics, both academic and otherwise – I cannot thank you enough, well beyond words.

An additional acknowledgement goes to prof. Xue-Cheng Tai – you have given so much, in depth and breath, inspired, and support me throughout the journey.

Extended gratitude goes to profs. Alex and Michael Bronstein who have been a continuous source of both knowledge and inspiration since my undergraduate project under their guidance and have made my life so much richer and I thank them for it.

I would like to also wholeheartedly thank prof. Freddie Bruckstein who has encouraged me to ask questions and never give up the pursuit for knowledge and mathematical beauty – I would have not seen the same world without you and I thank you for everything.

Finally, I would like to thank my family and friends, who have supported me throughout my studies and throughout the years – you know who you are and I owe you everything.

The generous financial help of the Technion and the Israeli Ministry of Science is gratefully acknowledged, as well as the European Community's FP7- ERC program, grant agreement no. 267414. The author would further like to thank Jacobs-Qualcomm scholarship for their support and Intel for the Intel PhD award.

Publications List

The work described in this thesis is based partially on the following publications:

[1] G. Rosman, A. M. Bronstein, M. M. Bronstein, and R. Kimmel. Articulated motion segmentation of point clouds by group-valued regularization. In *Eurographics Workshop on 3D Object Retrieval*, pages 77–84, 2012.

[2] G. Rosman, A. M. Bronstein, M. M. Bronstein, X.-C. Tai, and R. Kimmel. Group-valued regularization for analysis of articulated motion. In *NORDIA workshop, ECCV*, pages 52–62, Berlin, Heidelberg, 2012. Springer-Verlag.

[3] G. Rosman, M. M. Bronstein, A. M. Bronstein, A. Wolf, and R. Kimmel. Group-valued regularization framework for motion segmentation of dynamic non-rigid shapes. In *Scale Space and Variational Methods in Computer Vision*, volume 6667 of *Lecture Notes on Computer Science*, pages 725–736, 2011.

[4] G. Rosman, L. Dascal, X.-C. Tai, and R. Kimmel. On semi-implicit splitting schemes for the beltrami color image filtering. *Journal of Mathematics in Imaging and Vision*, 40(2):199–213, 2011.

[5] G. Rosman, A. Dubrovina, and R. Kimmel. Sparse modeling of shape from structured light. In *3DIMPVT*, pages 456–463, Washington, DC, USA, 2012. IEEE Computer Society.

[6] G. Rosman, A. Dubrovina, and R. Kimmel. Patch-collaborative spectral point-cloud denoising. *Computer Graphics Forum*, 2013.

[7] G. Rosman, A. Dubrovina, and R. Kimmel. Sparse modeling of shape from structured light. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2013. submitted.

[8] G. Rosman, S. Shemtov, D. Bitton, T. Nir, G. Adiv, R. Kimmel, A. Feuer, and A. M. Bruckstein. Over-parameterized optical flow using a stereoscopic constraint. In *Scale Space and Variational Methods in Computer Vision*, volume

6667 of *Lecture Notes on Computer Science*, pages 761–772, 2011.

[9] G. Rosman, X.-C. Tai, L. Dascal, and R. Kimmel. Polyakov action for efficient color image processing. In *ECCV workshop on color and reflectance in comp. vision*, 2010.

[10] G. Rosman, X.-C. Tai, L. Dascal, and R. Kimmel. Polyakov action for efficient color image processing. In *ICNAAM*, volume 1281 of *AIP*, pages 1018–1021, 2010.

[11] G. Rosman, Y. Wang, X.-C. Tai, R. Kimmel, and A. M. Bruckstein. Fast regularization of matrix-valued images. In *European Conf. Computer Vision*, volume 7574 of *Lecture Notes on Computer Science*, pages 173–186. Springer, 2012.

[12] G. Rosman, Y. Wang, X.-C. Tai, R. Kimmel, and A. M. Bruckstein. Fast regularization of matrix-valued images. *Methods and Applications of Analysis*, 2013. accepted.

Contents

1 Introduction **7**

1.1 Regularization Techniques 8

1.1.1 Regularization Functionals 10

1.1.2 Dirichlet Energy 10

1.1.3 Total Variation 10

1.1.4 Second-Order Total Variation 11

1.1.5 Ambrosio-Tortorelli Regularization 12

1.1.6 Non-local Regularizers 12

1.1.7 Sparsity-based Regularizers 14

1.2 Optimization Techniques for Variational Regularizers 14

1.2.1 Minimizing Flow Approaches 14

1.2.2 Steady-State Approaches 16

1.2.3 Direct Optimization Approaches 16

1.3 Motion Manifolds 17

1.3.1 Three-Dimensional Rigid Motion 18

1.3.2 Stereoscopic Motion 20

1.3.3 Depth Priors 20

2 Over-parameterized Optical Flow using a Stereoscopic Constraint **23**

2.1 Introduction 23

2.2 Background 25

2.2.1 The Variational Framework 25
2.2.2 Epipolar Geometry 26
2.3 Estimation of the Fundamental Matrix 27
2.4 A Flow Model Based on Local Homographies 28
2.4.1 Euler-Lagrange Equations 31
2.4.2 Implementation 32
2.5 Experimental results 33
2.6 Conclusions 35

3 Group-valued Regularization for Motion Segmentation of Articulated Shapes 38

3.1 Introduction 38
3.1.1 Main Contribution. 39
3.1.2 Relation to Prior Work. 40
3.2 Problem Formulation 41
3.2.1 Articulation Model 42
3.2.2 Motion Segmentation 42
3.2.3 Lie-Groups 44
3.3 Regularization of Group-Valued Functions on Surfaces 46
3.3.1 Ambrosio-Tortorelli Scheme 47
3.3.2 Diffusion of Lie-Group Elements 48
3.4 Numerical Considerations 49
3.4.1 Initial Correspondence Estimation 49
3.4.2 Diffusion of Lie-Group Elements 50
3.4.3 Visualizing Lie-Group Clustering on Surfaces 52
3.5 Results 53
3.5.1 Triangulated Meshes 54
3.5.2 Point-Clouds 55
3.6 Conclusion 57

4 Fast Regularization of Matrix-Valued Images **62**

4.1 Introduction 62

4.2 A Short Introduction to Lie-Groups and Matrix Manifolds 63

4.3 An Augmented Lagrangian Regularization Algorithm for Matrix-valued Images 64

4.3.1 Minimization w.r.t. v 67

4.3.2 Minimization w.r.t. u 67

4.3.3 Split-Bregman method for matrix-valued regularization 69

4.3.4 Regularization of maps onto $SO(n)$ 70

4.3.5 Regularization of maps onto $SE(n)$ 73

4.3.6 Regularization of maps onto $SPD(n)$ 74

4.3.7 A Higher-Order Prior for Group-Valued Images 77

4.4 Numerical Results 78

4.4.1 Directions Regularization 79

4.4.2 $SE(n)$ Regularization 80

4.4.3 DT-MRI Regularization 83

4.5 Regularized DTI Reconstruction 84

4.6 Conclusions 86

5 Sparse Priors for Structured-Light Reconstruction **94**

5.1 Introduction 94

5.2 Regularized Structured-Light Model 96

5.2.1 Regularization Terms for Depth Images 100

5.3 Alternating Minimization Algorithm for Regularized Structured-Light . 103

5.3.1 Learning a Depth Dictionary 105

5.3.2 Learning a Gaussian-Mixture Model for Depth Images 106

5.4 Results 107

5.4.1 Color Structured-Light Example 111

5.5 Conclusions 112

6 Discussion and Conclusions **118**

6.1 The Generality of the Proposed Approach 118

6.2 Optimization Techniques 119

References **122**

List of Figures

1.1 An illustration of over-parameterized motion spaces. 18

2.1 Epipolar geometry 28
2.2 AAE values on the middlebury test set 35
2.3 Grove2 and Urban2 sequence results 36

3.1 Segmenting a human figure 55
3.2 Segmenting a horse figure 56
3.3 Segmenting a camel figure 57
3.4 Segmenting a horse based on motion capture markers 58
3.5 Visualization of the transformations before and after smoothing 58
3.6 $SE(3)$ segmentation based on Kinect frames 59
3.7 Visualization of the transformations before and after regularization . . . 60

4.1 Shrinkage Cost Functions Illustration 68
4.2 Residual plots for a variety of θ values, for the optimization problem given in Figure 4.3 73
4.3 TV regularization of $SO(n)$ synthetic data 87
4.4 $SO(2)$ regularization in fingerprint images 88
4.5 $SE(3)$ regularization from kinect frames using ICP 89
4.6 $SE(3)$ regularization and segmentation, rendered hand model 89
4.7 $SE(3)$ regularization and segmentation from Kinect video 90
4.8 $SE(3)$ segmentation and the effect of regularization, Kinect frames . . . 91

4.9 $SE(3)$ scale-space, Kinect video of a deforming object 91
4.10 TV denoising for DTI data, glyphs visualization 91
4.11 TV-regularized reconstruction of DTI data, glyph visualization 92
4.12 TV-regularized reconstruction of DTI data, principal directions/FA/mean diffusivity visualization 92

5.1 An example of a structured-light system setup. 97
5.2 Sample dictionary from depth images 107
5.3 Sample GMM model from depth images 108
5.4 Sample reconstruction of the eye region, colored structured-light 109
5.5 Plane sweeping vs. decoding SL 110
5.6 SL reconstruction with sparse prior, single-color gray-code patterns . . 111
5.7 3D reconstruction comparison 114
5.8 Motion artifacts reconstruction comparison 115
5.9 Reconstruction results with GMM prior and color patterns 116

List of Tables

2.1 AAE comparison for the Middlebury training set and the Yosemite sequence 34

4.1 Processing times for $SE(3)$ regularization 82

5.1 SL Reconstruction error comparison 112

Abstract

One of the most important aspects of solving a problem is that of choosing an appropriate parameterization. This trivial observation can be seen in many forms in image processing and computer vision. Global parametrizations include the Hough and Fourier transforms, whereas local parameterizations include sparsity-based patch models and over-parameterized approaches. This research explores important cases in motion analysis and 3D reconstruction where a careful choice of the parameterization matters. It leads, in these cases, to simple and yet generic formulations that can be efficiently implemented.

The first part of the work related to 2D stereovision, where we suggest to use the plane equation and planar homographies as a basis for an over-parameterized optical flow estimation. The algorithm achieves state of the art results in term of accuracy in optical flow computation. The regularization term has a physically meaningful interpretation bridging the gap between optical flow computation and scene understanding.

The second part of the the dissertation relates to 3D motion understanding, where we reformulate articulated motion as edge-preserving smoothing of Lie-group-valued images of two types. By choosing carefully the parameterization and regularization terms, the resulting algorithms obtain results comparable to those of domain specific tools, on 3D range data. One of these algorithms can be implemented at real-time speeds due to a novel formulation. Furthermore, it applies also to other inverse problems such as diffusion tensor imaging reconstruction, and direction diffusion.

In the third and final part of the dissertation, we show how structured light recon-

struction can be formulated as probability maximization with respect to the scene geometry, given the camera and projector images. This allows us to incorporate sparse priors for the surface into the non-linear reconstruction process itself. These priors, resulting from the data, have a natural and intuitive interpretation, and in themselves parameterize epipolar motion between the camera and projector. Furthermore, they help us obtain 3D reconstruction that is robust to low sensor exposure and motion artifacts.

List of Symbols

u	A scale- or vector- valued function over the image or surface domain, representing a motion field
v	An auxiliary function for u
μ	Lagrange multipliers
g	A group-valued function
a	An over-parameterized 3-DOF representation of stereoscopic motion; An over-parameterized motion vector
S	The special homography matrix between two stereoscopic images
$\mathbb{G}$	A Lie-group
$\mathfrak{g}$	Lie-algebra associated with the Lie-group $\mathbb{G}$
$N(x)$	A neighborhood of a point x in the image or surface domain
$\mathcal{S}$	A surface
D	A dictionary
α	A coefficient vector of a dictionary
$H(u)$	The Hessian matrix of scalar- or vector- function u as a vector-valued function
$J(u)$	The Jacobian of a scalar- or vector- function u as a vector-valued function
q	A quaternion describing 3D motion

v_{AT}	A diffusivity function associated with an Ambrosio-Tortorelli regularization
$\Psi(\cdot)$	A robust fitting function
$\mathbf{e}'$	The epipole of an image
$\mathcal{F}$	The fundamental matrix
H	The Hessian operator
$\mathbb{R}^n$	n-dimensional Euclidean space
$SO(n)$	The special-orthogonal group in n dimensions
$SE(n)$	The special-Euclidean group in n dimensions
$\mathrm{Aff}(\mathbb{R}^n)$	The affine group over $\mathbb{R}^n$

List of Abbreviations

AAE	Average Angular Error
ANN	Approximate Nearest Neighbor library
AT	Ambrosio-Tortorelli
BM3D	Block-Matching 3D denoising
GMM	Gaussian-Mixture Model
ICP	Iterative Closest Point
IRLS	Iterative Reweighted Least-Squares
MRI	Magnetic Resonance Imaging
DT-MRI	Diffusion Tensor Magnetic Resonance Imaging
FA	Fractional Anisotropy
MA	Mean Diffusivity
SL	Structured Light
LMedS	Least Median of Squares
MSE	Mean Squared Error
MLS	Moving Least-Squares
NLM	Non-local Means

ODE	Ordinary Differential Equation
PCA	Principal Component Analysis
PDE	Partial Differential Equation
PSNR	Peak Signal-to-Noise Ratio
RANSAC	Random Sampling Consensus
SPD	Symmetric Positive Definite
SNR	Signal-to-Noise Ratio
TV	Total Variation
ISTA	Iterative Shrinkage Thresholding Algorithm

Chapter 1

Introduction

The choice of a natural measure of regularity is important in many areas of engineering such as signal processing and numerical analysis, and has been part of computer vision and control theory since its early days (see for example [65, 225, 240]). Regularization of images and multidimensional signals has been also given significant attention since the beginning of modern computer vision. The connection between regularization and understanding and interpretation of images has been thoroughly studied in the context of scale-space theories [4, 139, 149, 242].

Yet in computer vision, there are often several ways to represent some aspect of the scene as a map. This is especially true when the relevant information is given as a vector-, matrix- or group-valued map. For cases in which several value sets can be used to describe the scene, we refer to this decision of representation as a *choice of parameterization*. Often in these cases, the exact choice of parameterization for the problem at hand is crucial. Its interplay with the regularization and data-dependent terms is especially important when trying to solve inverse problems in computer vision, as we will demonstrate.

In this thesis we demonstrate several aspects in computer vision relating to understanding of motion and structure where the right choice of parameterization, coupled with axiomatic approaches for regularization provides us with well-founded methods of solution.

Specifically, we propose algorithmic solutions for the problems of 2D optical flow estimation, 3D articulated motion understanding and 3D reconstruction from structured light. In Chapter 2, based on an SSVM conference paper [185], we suggest a paraemterization for optical flow estimation that is based on 2D homographies between image regions. These homographies are intimately linked to planar regions in the scene. Thus, motion estimation in this case parallels segmentation of the scene into planar patches, as we will show. In the case of 3D articulated motion understanding, in Chapter 3, based on conference papers [180, 182], we demonstrate how 3D articulated motion segmentation can be reformulated as regularization of a mapping from the scene to the rigid transformations group $SE(3)$. Chapter 4, based on a conference paper [188], further extends this regularization into a fast framework for matrix-valued image regularization. In 3D reconstruction from structured light, we demonstrate how reformulating the problem as probability maximization of the estimated depth. Besides the known camera and projector image inserting regularization for the depth provides us with superior reconstruction results. This allows us to handle low exposure times and motion artifacts in the reconstruction. These results have been published as a conference paper [183]. Chapter 6 concludes the thesis and discusses future directions of research.

We now proceed to describe the main regularization terms utilized in this thesis, the parameterizations used for motion and structure, and the optimization techniques utilized throughout this work.

1.1 Regularization Techniques

The topic of regularization is an important one in solving inverse problems. In our setting of inverse problems we are looking at a map, $u : \mathcal{X} \rightarrow \mathcal{Y}$, describing some aspect of the real world – this can be the actual intensity image of the scene, a description of the motion field between two images of the scene, a volumetric map of the diffusion tensor in the tissue, and so forth. We denote $\mathcal{X}$ as the domain, and $\mathcal{Y}$ the range of u, and specifically in most of the examples we will show, one of them or both will not be

a Euclidean manifold, but rather a manifold equipped with a more general Riemannian metric (we refer the interested reader to standard books such as [49, 70] for a survey of differential and Riemmanian geometry). An inverse problem can be loosely defined as minimizing a distance between $F_D(u)$ (the application of a forward operator, describing the measurement process) and f_D, the measurements. In many problems, this distance is the squared Euclidean distance, essentially minimizing the L_2 norm between $F_D(u)$ and f_D. Other possibilities involve either other different norms of the L_p family, or different comparison measurements altogether such as more generic robust fitting terms, measurements between point clouds, Wasserstein metrics and so forth. These fitting terms are also known as the data terms.

Many inverse problems do not have a single solution for the instance data; these problems are called *ill-posed* problems. One solution is to add to the optimization problem a term whose minimizing solutions favor some properties of the solution. This term is known as a regularization term, and its choice greatly affects the resulting solution. The regularization term is usually the result of our assumption on the model of reality, allowing us to obtain a reasonable solution with respect to this model. For specific types of problems, regularization allows us to obtain a unique solution (for example, if both the data and the regularization terms are convex).

Problems that are not ill-posed are called *well-posed*. Indeed, some problems may be well-posed, but the sensitivity of the solution to inaccuracies in the problem data may have a significant gain factor associated with it. Such problems are *ill-conditioned*. In these cases as well, more accurate and suitable regularization terms allow us to handle stronger inaccuracies and noise in the data we sense.

We now describe several of the standard terms used in regularization, and then proceed to describe the associated optimization techniques.

1.1.1 Regularization Functionals

Regularization approaches go back to Tikhonov [225] and earlier works. In this subsection we describe the main functionals discussed in this thesis.

1.1.2 Dirichlet Energy

The simplest regularization functional in used in signal processing beyond standard Tikohnov regularization is the Dirichlet energy

$$\int \|\nabla u\|^2 d\mathcal{X}, \tag{1.1}$$

where $d\mathcal{X}$ is a measure associated with $\mathcal{X}$. This energy in itself is useful for many signals that are assumed to be smooth. The Euler-Lagrange equation for this functional is the heat equation

$$\Delta u = 0. \tag{1.2}$$

For the case in which the domain or range are non-Euclidean manifolds, the resulting flow has a subject of intense research. For the general case, see [75]. Additionally, evolution according to this equation can be related in signal processing to Gaussian filtering [123]. It therefore has strong ties to the Gaussian scale-space, whose properties have been thoroughly studied [139, 242].

Yet in most signals, the smoothness assumption is far too simplistic. In the last two decades, numerous regularization functions have been suggested in both the discrete and continuous setting. We now describe a few of these regularization functionals, especially those that relate to the algorithms shown in this dissertation.

1.1.3 Total Variation

For signals that are piecewise-smooth Rudin, Osher and Fatemi (ROF) introduced [192] the *total variation* (TV) measure. This regularization term penalizes the L_1 norm of the

derivative of the signal for 1D signals.

$$\int |u_x| d\mathcal{X} \tag{1.3}$$

For a multivariate scalar signal u over a domain $\Omega \in \mathbb{R}^d$, the total variation functional is written as

$$\int \|\nabla u\| d\Omega. \tag{1.4}$$

For vector-valued signal $u = (u_1, \cdots, u_n)$, a straightforward extension is by integrating the Frobenious norm of the Jacobian operator,

$$\int \|Ju\| d\Omega, (Ju)_{ij} = \frac{\partial u_i}{\partial x_j}. \tag{1.5}$$

Additional important extensions to the case of color images are Di Zenzo's gradient extension [257], and the Beltrami cost function [209], for which several efficient computational schemes have been proposed [66, 186], as part of this thesis. The Beltrami functional expresses the area of the map between the image domain and the spatial-chromal range, defined via the induced metric of each space.

In order to obtain a continuous approximation to the TV functional, it is customary to use the Charbonnier regularization term

$$\int \Psi\left(\|\nabla\|^2\right) d\Omega, \quad \Psi\left(s^2\right) = \sqrt{s^2 + \epsilon^2}, \tag{1.6}$$

where ϵ is some small constant. Typically ϵ is chosen so as to distinguish small errors in the measurements from outliers, signifying the typical "small" noise level. Additional robust fitting terms used in the context of computer vision can be found in [34].

1.1.4 Second-Order Total Variation

In order to alleviate staircasing artifacts, and to accomodate the solution of problems involving signals that are approximately piecewise-linear, the second-order total variation,

$$\int \|\nabla^2 u\| d\Omega, \tag{1.7}$$

has been suggested.

While the Euler-Lagrange equation for this functional is well-defined, the higher derivatives tend to make the solution of problems involving such terms less straightforward and specific numerical schemes are needed.

Other high-order regularizers are also available. These include, for example the total generalized variation (TGV, [36]), and the unbiased second-order prior suggested by Trobin et al. [228].

1.1.5 Ambrosio-Tortorelli Regularization

Ambrosio and Tortorelli [5] proved the Γ-convergence of a sequence of regularization problems approximating the Mumford-Shah functional [157]. In the Mumford-Shah functional, the image is approximated by a piecewise-smooth model, and regularization is given by

$$\int_{\Omega\setminus\gamma} \|\nabla u\|^2 d\Omega + \int_{\Omega} (u-u_0)^2 \, d\Omega + \mathcal{H}(\gamma), \tag{1.8}$$

where γ is the discontuity set, $\mathcal{H}$ is the *Housdorff measure* associated with the domain Ω, and u_0 is the original image.

Ambrosio and Tortorelli suggested to approximate the space by a set of subproblems, and proved convergence in the sense of Γ-convegence [15]. Among these are functionals of the form

$$\int_{\Omega} v_{AT}^2 \|\nabla u\|^2 + \epsilon \|\nabla v_{AT}\|^2 + \alpha^2 \frac{(v_{AT}-1)^2}{4\epsilon} d\Omega, \tag{1.9}$$

where v_{AT} is the diffusivity function, serving as a phase-discontinuity indicator. Ambrosio-Tortorelly regularization has been incorporated into many application domains, such as image processing [19] and motion estimation [45], among others.

1.1.6 Non-local Regularizers

Bilateral Regularizers

Motivated by signal smoothing in the histogram domain [253], the bilateral filter has been suggested in several formulations [167, 208, 226, 253]. The filter itself operating over an input image u_0 can be written as

$$u(x) = \frac{\int k(x, x'; u_0) u_0(x') d\Omega}{\int k(x, x'; u_0) d\Omega}. \tag{1.10}$$

The standard choice of a kernel $k(x, x'; u_0)$ is a Gaussian one,

$$k(x, x'; u) = e^{-\left(\|u_0(x) - u_0(x')\|^2 / \sigma_R^2 + \|x - x'\|^2 / \sigma_D^2\right)}, \tag{1.11}$$

where σ_R and σ_D denote the average scales associated with the range and domain of u, respectively. The distance $d^2_{\sigma_R, \sigma_D}(x, x'; u_0) = \frac{\|x - x'\|^2}{\sigma_D^2} + \frac{\|u_0(x) - u_0(x')\|^2}{\sigma_R^2}$ can be shown to be the Euclidean approximation of the geodesic distance associated with the Beltrami filter [209]. This has been used for fast Beltrami filtering by Spira et al. [210].

By freezing the role of u_0, the basic functional behind the bilateral filter can be written as

$$\int k(x, x'; u_0) \|u(x) - u_0(x')\|^2 d\Omega, \tag{1.12}$$

where k is a weight function comparing the positions x, x', and the value at $u_0(x)$,$u_0(x')$, usually taken to be decreasing with the distance between x, x', $u_0(x), u_0(x')$. It is usually assumed that $u_0(x)$,$u_0(x')$ are close enough to have a meaningful k associated with u_0.

Non-local Means Regularizers

An intuitive and yet very powerful extension of the bilateral filter is given by extending k to describe the similarity between the neighborhoods of x, x' and their map.

$$\int k(N(x), N(x'); u_0) \|u(x) - u_0(x')\|^2 d\Omega \tag{1.13}$$

The classical example for this [46] was suggested by Buades, Coll and Morel,

$$\int e^{-\int_N (u_0(x+y) - u_0(x'+y))^2 d\Omega} \|u(x) - u_0(x')\|^2 d\Omega. \tag{1.14}$$

A generalization of this filter to other kernel choices is given by Goossens et al. [93]. the resulting *non-local means* filter has been used for texture and image completion [76], pixel classification [63, 87], depth-image denoising [110] and other applications.

1.1.7 Sparsity-based Regularizers

Sparse Dictionary Regularizers

A different family of regularizers, usually given for discretely sampled signals, is the family of sparsity-based regularizers. In these regularizers, the representation of the signal is assumed to be a locally sparse one. This representation is set by a dictionary of examplar patches. The coefficients associated with this dictionary for local patches of the signal can be the coefficients of a linear combination used to obtain the patch (this is known as a synthesis approach), or the response of the signal to a matched set of filters (this is known as the analysis approach [88]). We refer the interested reader to the literature (see for example [78]) for a more complete view of this developingc topic.

Structured Sparsity Regularizers and GMMs

In a few recent papers [256, 260], the relation between local *Gaussian-mixture models* (GMM) and structured-sparsity approaches has been investigated. The structured sparsity interpretation of GMM priors suggests to pick first the support set of the patch (i.e the Gaussian component) and then decide on the coefficient values themselves.

1.2 Optimization Techniques for Variational Regularizers

Given a functional with a data and regularization terms, several methods are available for obtaining the optimal u. Here, we describe the main optimization approaches used in the proceeding algorithms, with an emphasis on continuous variational approaches.

1.2.1 Minimizing Flow Approaches

Explicit Minimizing Flows

A classical set of approaches for minimizing variational functionals involves expressing the minimizing flow of the functional,

$$u_t = -\frac{\delta E(u)}{\delta u}, \tag{1.15}$$

where $\frac{\delta E(u)}{\delta u}$ is the first variation of the functional E with respect to u. The resulting *partial differential equation* (PDE) can now be discretized, and its steady state computed. The simplest approach for this involves explicit time iterations, such as the explicit forward-Euler scheme (we refer the reader to standard literature for discussion of standard discretization schemes [155]). Assuming a small enough time step Δt, we have

$$\frac{u^t - u^{t-1}}{\Delta t} = -\frac{\delta E(u)}{\delta u}\left(u^{t-1}\right), \tag{1.16}$$

Semi-implicit and Implicit Schemes

In semi-implicit schemes, Equation 1.15 is linearized (based on time $t-1$) with respect to u^t for the right hand side terms. Specifically, we write the time evolution equation as

$$\frac{u^t - u^{t-1}}{\Delta t} = A\left(u^{t-1}\right)u^t, \tag{1.17}$$

where $A(\cdot)$ is an operator that depends on the previous time step for linearization. For the current time step solution is obtained by solving the system

$$\left(Id - \Delta t A\left(u^{t-1}\right)\right)u^t = u^{t-1}, \tag{1.18}$$

where the key to obtaining an overall efficient scheme is efficiently inverting the system matrix $(Id - \Delta t A(u^{t-1}))$. This inverted operator can be shown to be a contraction, ensuring the stability of this scheme.

Semi-implicit schemes enabled the development of efficient solution methods for nonlinear functionals in computer vision, coupled with additive [144, 238] and multiplicative [20] splitting.

Implicit schemes for variational regularization functinals have been suggested. Using such schemes, however, is more involved and requires equation-specific solutions for the time-stepping equations, as in this methods $A(\cdot)$ is given in terms of u^t.

1.2.2 Steady-State Approaches

Another set of approaches involves using the Euler-Lagrange formula as a steady-state equation and solving the (spatially-dependent only) PDE.

Classical approaches such as *Gauss-Seidel* (GS), and *successive over-relaxation* (SOR) have been used extensively in computer vision. For example, in optical flow estimation [42], with red-black GS/SOR solvers being quite effective due to the support of the discretized diffusion operators.

Multigrid techniques have allowed further efficiency improvement under a well-defined basis [43], and allows us to further exploit the convergence properties of the problem.

1.2.3 Direct Optimization Approaches

Returning to optimization of the cost function, several efficient techniques have been suggested for optimization of the variational nonlinear regularizers described above. Specifically, based on the augmented Lagrangian technique [103, 175], differential operators can be replaced with auxiliary variables during the optimization. The complete optimization is now performed on both sets of variables, but the additional complexity incurred by adding more variables is offset by the simpler optimization steps done for each variable or sets of variables. The resulting optimization can be shown [244] to be equivalent to split-Bregman optimization approaches in certain cases [91, 254]. Usually, the information flow between signal neighborhood is now embodied in a linear PDE, and the auxiliary variables are updated via a per-pixel equation. This results in numerical schemes that are highly-parallelizable, and easily achieve real-time performance on *graphics processing units* (GPUs).

1.3 Motion Manifolds

Motions in 2D and 3D have several typical parameterizations and models. One way to view a motion field sees it as a map between a domain $\mathcal{X}$ and a range $\mathcal{Y}$. An alternative view considers a map between the domain $\mathcal{X}$ and a *motion parameterization space* representing the local motion model $\mathcal{P}$. For example, in optical flow the domain $\mathcal{X}$ and range $\mathcal{Y}$ are images, with the flow defined between them. $\mathcal{P}$ is a motion model which locally defines an intermediate representation of the motion from $\mathcal{X}$ into $\mathcal{Y}$.

On one hand, the motion field in optical flow can be expressed either as a map from an image into the manifold of all motion vectors, $\mathbb{R}^2$. On the other hand, it can also be expressed via a map from the image domain into the manifold of all affine transformations. Hence, both $\mathbb{R}^2$ and $\mathrm{Aff}(\mathbb{R}^2)$ can serve as $\mathcal{P}$ in this case. Embedding a manifold such as $\mathrm{Aff}(\mathbb{R}^2)$ into $\mathbb{R}^6$ gives us a simple way of describing this model using six parameters. The standard metric *induced* from the embedding onto $\mathbb{R}^6$ gives us a way of quantifying discontinuities in the motion model. This results in the affine over-parameterized optical flow model [163] for the choice of $\mathcal{P} = \mathrm{Aff}(\mathbb{R}^2) \subseteq \mathbb{R}^6$, but other choices would have given us different over-parameterized models, equipped with different measures of model discontinuities.

As an example of such a different choice, normalizing the image coordinates, which is done in practice [163], changes the specific parameterization, and this is known to be essential for affine over-parameterized optical flow to work. But this normalization step, however, is not an arbitrary step – it merely reflects a different embedding of $\mathrm{Aff}(\mathbb{R}^2)$ into $\mathbb{R}^6$, with a different resulting induced metric, and hence, in practice, a different regularization effect for the term $\|\nabla a\|^2$, where a is the over-parameterized vector (see Chapter 2 and [163] for more details).

Thus, the parameterization space $\mathcal{P}$ and its metric capture the essence of over-parameterized models of motion by relating the choice of parameterization to the regularization used. In this section we discuss a few of these models, relevant to the problems we describe in the following chapters, such as 3D rigid motion segmentation and

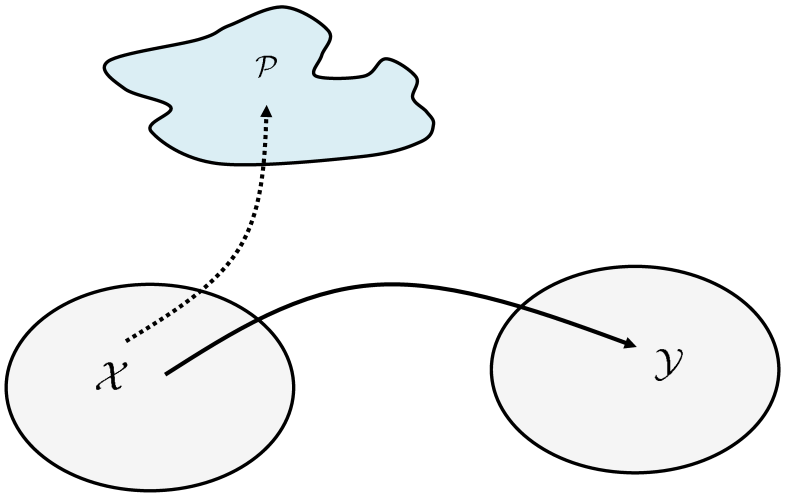


Figure 1.1: An illustration of over-parameterized motion spaces.

1.3.1 Three-Dimensional Rigid Motion

Models characterising rigid motions in Euclidean space have been around for quite a while (see for example [18, 127, 137]). We now describe a few of these models – special emphasis is given to rigid motion in the three-dimensional Euclidean space, but some of the principles involved are much more general and apply to any finite dimension.

One theory that allows us to analyze rigid motions in a concise manner that lends itself to differential operators and discussion of regularity is the theory of Lie-groups. Lie-groups are topological groups with a smooth manifold structure such that the group action $\mathbb{G} \times \mathbb{G} \mapsto \mathbb{G}$ and the group inverse are differentiable maps.

Because of the group structure and the existence of an inverse element, the neighborhood of each point on the manifold can be mapped onto a neighborhood of the identity element. This vector space allows us a way of adding and subtracting neighboring group elements, and thus allows us to define derivatives, regularity, and diffusion operators on

the group valued data. This vector space is known as the *Lie-algebra* $\mathfrak{g}$ associated with the Lie-group $\mathbb{G}$.

More generally, A Lie-algebra is a vector space endowed with a *Lie-brackets operator* $[\cdot,\cdot] : \mathfrak{g} \times \mathfrak{g} \rightarrow \mathfrak{g}$, describing the local structure of the group. The Lie-algebra associated with a Lie-group can be mapped diffeomorphically via the *exponential map* onto a neighborhood of the identity operator and its tangent space.

There are many Lie-groups being investigated in various subfields of mathematics, physics and engineering. Of these, two such important groups relate to 3D motion – the group of all rotations in $\mathbb{R}^3$, $SO(3)$, and the group of all rigid motions in $\mathbb{R}^3$, $SE(3)$. We further detail Lie-groups that pertain to this work in Chapters 3 and 4.

Another method for representing rotations in 3D are quaternions. Quaternions can be approached in several forms which result in slightly different parameterizations. For the relation of quaternions to Lie-groups theory we refer the reader to [213].

A relatively intuitive route taken from Euler's rotation theorem: any displacement of a rigid body such that a point on the rigid body remains fixed, is equivalent to a single rotation about some axis that runs through the fixed point.

Thus, one way to view quaternions is as a 4-element complex number, where the 3 imaginary elements describe the rotation axis, whereas the fourth element describes the amount of rotation [102]. If normalized properly, a quaternion can be written as $q = \cos\frac{1}{2}\theta + \left(u_x\hat{i} + u_y\hat{j} + u_z\hat{k}\right)\sin\frac{1}{2}\theta$, where $\hat{i},\hat{j},\hat{k}$ denote the components of the complex parts of the quaternion, (u_x, u_y, u_z) describe the direction vector, and θ denotes the amount of rotation applied. This quaternion can operate on a 4 vector describing a point $p = x\hat{i} + y\hat{j} + z\hat{k}$ in 3D-space using Hamilton's product

$$p' = qpq^*, \tag{1.19}$$

where q^* denotes the conjugate of q,

$$q^* = cos\frac{1}{2}\theta - \left(u_x\hat{i} + u_y\hat{j} + u_z\hat{k}\right)sin\frac{1}{2}\theta. \tag{1.20}$$

In order to describe free rigid motion, including translation, a product space with the

translations space $\mathbb{R}^3$ can be used.

Another representation, motivated by Chasles' theorem is the screw parameterization [18, 158]. Chasles' theorem states that every rigid motion in 3D-space can be described by a rotation around an axis, followed by motion along this axis. Screws parameterization is the combination (l, λ, t) where l describe the line in 3D (4 DOF), λ describes the pitch, or ratio between rotation and translation along the axis. t describes the amount of overall motion.

1.3.2 Stereoscopic Motion

In the case of stereoscopic image pairs and the disparity fields they describe, several parameterizations can also be used. While the flow field itself, given in Cartesian coordinates is a standard choice in optical flow and registration algorithms, alternative parameterizations are constantly being applied to the problem. Since motion between two images of the same scene is restricted to the epipolar lines with the depth of each point determining the amount of motion, optical flow models based on epipolar lines [202, 207, 237] have proven to be quite accurate and robust. Yet, such algorithms do not try to incorporate higher-level priors on the scene structure.

In the case where camera calibration is known, the plane equation of the scene surface can determine a 3×3 homography between the two images [100]. In the normalized case, where the intrinsic parameters matrices are the identify matrix $K_1 = K_2 = Id_{3\times3}$, we can easily develop the expression for this homography to be $\mathrm{H} \propto \mathrm{R} - \mathbf{t}\mathbf{v}^{\mathrm{T}}$, as will be shown in Chapter 2.

1.3.3 Depth Priors

A related topic is that of priors for depth images and surfaces. In surface processing, numerous algorithms have emerged from the field of spectral analysis [59, 69, 71, 201, 222]. Others are motivated by algorithms from image processing [169], often looking at the surface height field with respect to a local tangent plane. Yet another family of

algorithms [82, 135] stems from approximation theory and views the discrete surface as a sampled version of a continuous or piecewise-continuous object. As part of this thesis, a novel algorithm was suggested that draws upon these approaches [184]. Even in the case of explicit range images, several priors have been suggested, with close correlation to developments in image processing.

Chapter 5 explores the use of various priors inside the nonlinear reconstruction process used in shape-from-structured light. The resulting sparse dictionaries and Gaussian mixture model components favor local piecewise linear and polynomial models, similar to moving least squares estimators. Specifically in the inverse problem of structured light reconstruction as well as the problem of structure from multiple views, these local estimators define a motion parameterization space describing motion along the projected patterns code. A discussion relating variational methods to MLS estimators for the case of signal and surface reconstruction is available as a book chapter we published [205].

Chapter 2

Over-parameterized Optical Flow using a Stereoscopic Constraint

2.1 Introduction

We now proceed to describe a few application domains where a principled outlook relating overparameterized regularization and careful choice of parameterization can benefit 3D reconstruction and motion estimation. We start with optical flow estimation in stereoscopic image pairs, as we now describe.

A fundamental type of motion estimation is optical flow. Optical flow is defined as the motion field between consecutive frames in a video sequence. Its computation often relies on the brightness constancy assumption [107], which states that pixel brightness corresponding to a given scene point is constant throughout the sequence. Optical flow computation is a notoriously ill-posed problem. Hence, additional assumptions on the motion are made in order to regularize the problem. Early methods assumed spatial smoothness of the optical flow [107, 145]. Parametric motion models [29, 154], and more recently machine learning [190] were introduced in order to take into account the specificity of naturally occurring video sequences. In parallel, the regularization process was made much more robust [35, 42, 44, 60].

In this chapter, we focus on optical flow computation in stereoscopic image pairs,

given a reliable estimation of the fundamental matrix. We assume two images depicting the same scene – either these are two images taken simultaneously, or two images of a static scene. This problem has already been addressed in [32, 207, 232, 237]. The papers [32, 207] expressed the optical flow as a one-dimensional problem. This was done either by working on a rectified image pair [32], or by solving for the displacement along the epipolar lines [207]. A different approach [232, 237] merely penalized deviation from the epipolar constraint. In addition, [232] proposed a joint estimation of the stereoscopic optical flow and the fundamental matrix. Finally, in order to treat the problem of occluded areas and object boundaries, Ben-Ari and Sochen [27] suggest to explicitly account for regions of discontinuities.

Yet, a third body of works turned to a complete modeling of the scene flow [22, 109, 173]. While this approach is the most general, we focus in this chapter on static scenes, for which a more specific parameterization can be found.

While the reported experimental results in the aforementioned papers are very convincing, their regularization methods still rely on the traditional assumption that optical flow should be piecewise smooth. Here, motivated by the over-parameterization approach presented in [163], the optical flow is obtained by estimation of the space-time dependent parameters of a motion model, the regularization being applied to the model parameters. In [33], we used homogeneous coordinates to express a homography model, which allows to select a geometrically meaningful coordinate systems for this problem. Here we elaborate upon this model by adding an Ambrosio-Tortorelli scheme, which gives a physically meaningful interpretation for the minima obtained in the optimization process.

In the case of a static scene, the optical flow can be factored into a model determined by the camera motion and an over-parameterized representation of the scene. The scene motion is described locally as a homography satisfying the epipolar constraint and parameterized by the equation of a local planar approximation of the scene. Assuming that the scene can be approximated by a piecewise smooth manifold, enforcing piecewise

spatial smoothness on the homography parameters becomes an axiomatically justified regularization criterion which favors piecewise smooth planar regions.

2.2 Background

2.2.1 The Variational Framework

In the variational framework for optical flow, brightness constancy and smoothness assumptions are integrated in an energy functional. Let $(u(x,y,t), v(x,y,t))$ denote the optical flow at pixel coordinates (x,y) and time t. Brightness constancy determines the data term of the energy functional

$$E_D(u,v) = \int \Psi\left(I_z^2\right), \tag{2.1}$$

where

$$I_z = I(x+u, y+v, t+1) - I(x,y,t) \tag{2.2}$$

and $\Psi(s^2) = \sqrt{s^2+\varepsilon^2}$ is a convex approximation of the L_1 norm for a small ε, as in Equation 1.6.

$\mathcal{M}(\mathrm{a}, x, y, t)$ denotes a generic model of the optical flow at pixel (x,y) and time t, where $\mathrm{a} = (\mathrm{a_i}(x,y,t))_{i\in\{1,\dots,n\}}$ is a family of functions parameterizing the model, i.e.,

$$\begin{pmatrix} u(x,y,t) \\ v(x,y,t) \end{pmatrix} = \mathcal{M}(\mathrm{a}, x, y, t). \tag{2.3}$$

We begin with the smoothness term proposed by Nir et al. in [163],

$$E_S(\mathrm{a}) = \int \Psi\left(\sum_{i=1}^{n} ||\nabla \mathrm{a_i}||^2\right). \tag{2.4}$$

In order to refine the discontinuities and obtain a physically meaningful regularization, we extend the smoothness prior using the Ambrosio-Tortorelli scheme [5, 203].

$$E_{S,AT}(\mathrm{a}) = \int v_{AT}^2 \Psi\left(\sum_{i=1}^{n} ||\nabla \mathrm{a_i}||^2\right) + \epsilon_1(1-v_{AT})^2 + \epsilon_2\|\nabla v_{AT}\|^2, \tag{2.5}$$

where v_{AT} is a diffusivity function, ideally serving as an indicator of the discontinuities set in the flow field. Choosing $\epsilon_1 = \frac{1}{\epsilon_2}$ and gradually decreasing ϵ_2 towards 0 can be

used to approximate the Mumford-Shah [157] model via Γ-convergence process, but we do not pursue this direction in this chapter.

While the Ambrosio-Tortorelli scheme has been used in the context of optical flow [6, 28, 45], in our case this seemingly arbitrary choice of regularization and segmentation has a physical meaning. The regularization of the flow becomes a segmentation process of the *visible surface* in the scene into planar patches, each with his own set of plane parameters. In addition, it helps us obtain accurate edges in the resulting flow.

Furthermore, the generalized Ambrosio-Tortorelli scheme allows us to explicitly reason about the places in the flow where the nonlinear nature of the motion parameterization space manifests itself. Suppose we have a piecewise-planar, static, scene, and an ideal solution (a^*, v_{AT}^*) where a^* is piecewise constant, and the diffusivity function v_{AT}^* is 0 at planar region boundaries and 1 elsewhere. At such a solution, we expect two neighboring points which belong to different regions to have a very small diffusivity value v_{AT} connecting them, effectively nullifying the interaction between different planes' parameters. Furthermore the cost associated with this solution is directly attributed to the discontinuity set measure in the image. The proposed ideal solution therefore becomes a global minimizer of the functional, as determined by the measure of discontinuities in the $2\frac{1}{2}$-*D sketch* [149]. This is directly related to the question raised by Trobin et al. [228] regarding the over-parameterized affine flow model and its global minimizers.

The complete functional now becomes:

$$E(\mathrm{a}) = E_D(\mathcal{M}(\mathrm{a}, x, y, t)) + \alpha E_{S,AT}(\mathrm{a}). \tag{2.6}$$

In the remainder of this chapter, we will propose a motion model enforcing the epipolar constraint and show how to minimize the proposed functional.

2.2.2 Epipolar Geometry

Let us introduce some background on epipolar geometry, so as to motivate the choice of the motion model. A complete overview can be found in [80, 100].

Given two views of a static scene, the optical flow is restricted by the epipolar constraint. Figure 2.1 shows that a pixel $\mathbf{m}$ in the left image is restricted to a line $\mathbf{l}'$ called an epipolar line in the right image. All the epipolar lines in the left (resp. right) image go through $\mathbf{e}$ (resp. $\mathbf{e}'$), which is called the left (resp. right) epipole.

In projective geometry, image points and lines are often represented by 3D homogeneous coordinates

$$\mathbf{m} = \left\{ \lambda \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} | \lambda \in \mathbb{R}^{\star} \right\}. \tag{2.7}$$

Image points and their corresponding epipolar lines are related by the fundamental matrix $\mathcal{F}$

$$\mathbf{l}' = \mathcal{F}\mathbf{m}. \tag{2.8}$$

Consider a plane π, visible from both cameras, and the planar homography H_{π} which corresponds to the composition of the back-projection from the left view to a plane (π) and the projection from (π) to the right view (see Figure 2.1). The homography H_{π} gives rise to a useful decomposition of the fundamental matrix

$$\mathcal{F} = [\mathbf{e}']_{\times}\mathrm{H}_{\pi}, \tag{2.9}$$

where $[\mathbf{e}']_{\times}$ is a matrix representation of the cross product with $\mathbf{e}'$.

2.3 Estimation of the Fundamental Matrix

One of the main challenges in estimating optical flow using the epipolar geometry is to retrieve an accurate and robust estimation of the fundamental matrix. Mainberger et. al. [148] showed that robustness of the fundamental matrix estimation could be achieved by using dense optical flow instead of applying RANSAC or LMedS methods to a sparse set of matches. Hence, we use as initialization the Horn-Schunck with Charbonnier penalty function optical flow implementation provided by Sun et al. [217], modified to use color images. This represents a baseline nonlinear optical flow method, as in [217].

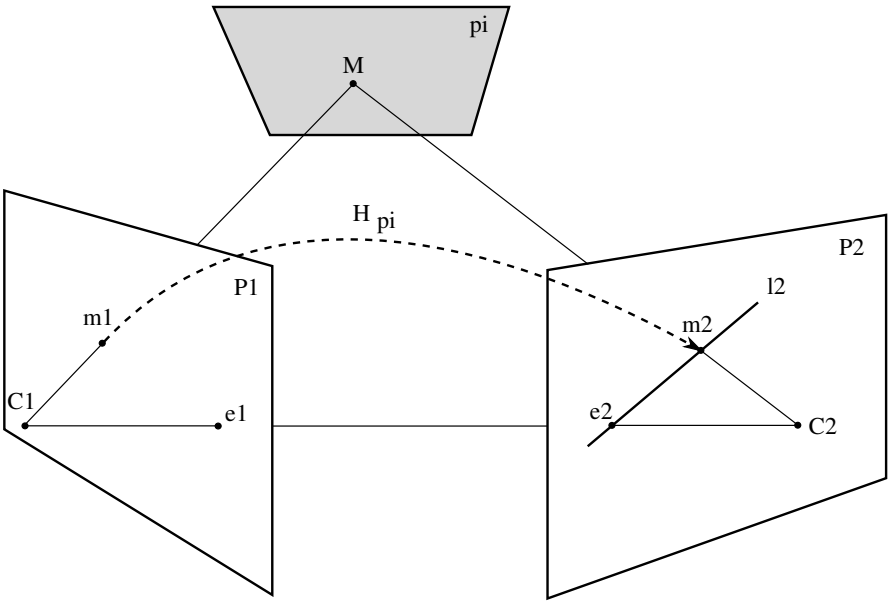


Figure 2.1: Epipolar geometry

In addition to allowing the computation of the fundamental matrix, this initialization also serves as a starting point for our optical flow computation algorithm.

Many methods aimed at estimating the fundamental matrix can handle large numbers of correspondences. Among those, we choose a robust M-estimation method based on the symmetric epipolar distance, the implementation of which is made very efficient by the use of the Levenberg-Marquardt algorithm, as explained in [124].

2.4 A Flow Model Based on Local Homographies

We now proceed to develop the model and motivation for the flow equations. Suppose the camera is calibrated, with projection matrices

$$\mathrm{P}(t) = \mathrm{P}_0 = \begin{pmatrix} \mathrm{I} & |\mathbf{0} \end{pmatrix}, \quad \mathrm{P}(t+1) = \mathrm{P}_1 = \begin{pmatrix} \mathrm{R} & |\mathbf{t} \end{pmatrix}. \tag{2.10}$$

where R is a rotation matrix and $\mathbf{t}$ is a translation vector expressing camera motion between the two consecutive frames at t and $t+1$. We assume that locally, the scene is well approximated by the plane

$$\mathbf{v}^{\mathrm{T}}\mathbf{x} + d = 0 \tag{2.11}$$

where $(\mathbf{x}^{\mathrm{T}}, d)^{\mathrm{T}} = (x, y, 1, d)^{\mathrm{T}}$ denotes the 3D scene point visible at pixel $\mathbf{x}$ in homogeneous coordinates. The corresponding point of $\mathbf{x}$ at time $t+1$ is

$$\mathbf{x}' = \mathrm{P}_1 \begin{pmatrix} \mathbf{x} \\ d \end{pmatrix} = \mathrm{R}\mathbf{x} + \mathbf{t}d = (\mathrm{R} - \mathbf{t}\mathbf{v}^{\mathrm{T}})\mathbf{x} \tag{2.12}$$

in homogeneous coordinates. $\mathbf{v}$ designates the normal of the local planar approximation of the scene, and $-(\mathbf{v}^{\mathrm{T}}\mathbf{x})^{-1}$ is the depth of the scene at time t. The planar homography expressed in (2.12) gives a geometrically meaningful motion model parameterized by $\mathbf{v}$. From now on, consider $\mathbf{v}$ as a function of the pixel coordinates. Under the assumption that the scene can be approximated by a piecewise smooth manifold, $\mathbf{v}$ must be piecewise smooth.

We now derive the motion parameterization. In general, the camera parameters are not known, but we can re-parameterize the planar homography using $\mathbf{e}'$ and $\mathcal{F}$. In the following derivation we assume a calibrated view for simplicity's sake. Let $\mathrm{H}(x, y, t)$ denote the planar homography motion model. We have

$$\mathrm{H} \propto \mathrm{R} - \mathbf{t}\mathbf{v}^{\mathrm{T}}. \tag{2.13}$$

For any compatible planar homography H_0 (cf. [100], 13.1.1.1, we will provide a specific choice later on),

$$\exists(\mathbf{v_0}, \mu): \quad \mathrm{H}_0 = \mu(\mathrm{R} - \mathbf{t}\mathbf{v_0}^{\mathrm{T}}) \tag{2.14}$$

$$\mathrm{H} = \mathrm{H}_0 - \mu\mathbf{t}(\mathbf{v} - \mathbf{v_0})^{\mathrm{T}}. \tag{2.15}$$

As $\mathbf{t}$ and $\mathbf{e}'$ are parallel, we can also write

$$\mathrm{H} = \mathrm{H}_0 + \mathbf{e}'\frac{-\mu\mathbf{e}'^{\mathrm{T}}\mathbf{t}}{||\mathbf{e}'||^2}(\mathbf{v} - \mathbf{v_0})^{\mathrm{T}}. \tag{2.16}$$

Hence, $\mathrm{H}(x, y, t)$ can be parameterized by the function

$$\mathbf{a}(x, y, t) = \frac{-\mu\mathbf{e}'^{\mathrm{T}}\mathbf{t}}{||\mathbf{e}'||^2}(\mathbf{v}(x, y, t) - \mathbf{v_0}), \tag{2.17}$$

so that

$$\mathrm{H}(x, y, t) = \mathrm{H}_0 + \mathbf{e}'\mathbf{a}(x, y, t)^{\mathrm{T}}. \tag{2.18}$$

The parameterization $\mathbf{a}$ is the unknown field we want to compute in order to model and estimate the optical flow. The piecewise smoothness of $\mathbf{a}$ is a direct consequence of the piecewise smoothness of $\mathbf{v}$, as testified by (2.17). More precisely, minimization of the Ambrosio-Tortorelli regularization term favors segmentation of the visible surface into planar patches where the data evidence permits it.

When the cameras are not calibrated, the relationship between the parameterization $\mathbf{a}$ and $\mathbf{v}$ is still linear. In fact, the calibration matrices mainly affect the relative weighting of the model parameters smoothness. Our experiments show that even without controlling the relative smoothness of the model parameters, the optical flow can be estimated accurately.

Note that the parameterization $\mathbf{a}$ can also be derived directly from the fundamental matrix decomposition (2.9).

For H_0, we can choose the special matrix

$$H_0 = \mathrm{S} = [\mathbf{e}']_\times \mathcal{F}. \tag{2.19}$$

Each column of S with the corresponding column of $\mathcal{F}$ and $\mathbf{e}'$ form an orthogonal basis of $\mathbb{R}^3$ so that (2.9) is satisfied. S is a degenerate homography which projects points in the left image to points of the line represented by $\mathbf{e}'$ in the right image. Next, we use the notations

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}, \quad \mathbf{e}' = \begin{pmatrix} x_{e'} \\ y_{e'} \\ z_{e'} \end{pmatrix}, \quad \mathrm{H}_0 = \begin{pmatrix} \mathbf{h_1}^{\mathrm{T}} \\ \mathbf{h_2}^{\mathrm{T}} \\ \mathbf{h_3}^{\mathrm{T}} \end{pmatrix}, \tag{2.20}$$

to signify the 3D point coordinates, the epipole's 2D homogeneous coordinates, and the homography matrix rows, respectively. The parameterization of H is introduced into the expression of the optical flow

$$\mathcal{M}(\mathrm{a}, x, y, t) = \begin{pmatrix} u \\ v \end{pmatrix} = \lambda \begin{pmatrix} \mathbf{h_1}^{\mathrm{T}}\mathbf{x} + x_{e'}\mathbf{a}^{\mathrm{T}}\mathbf{x} \\ \mathbf{h_2}^{\mathrm{T}}\mathbf{x} + y_{e'}\mathbf{a}^{\mathrm{T}}\mathbf{x} \end{pmatrix} - \begin{pmatrix} x \\ y \end{pmatrix}, \quad \lambda = \frac{1}{\mathbf{h_3}^{\mathrm{T}}\mathbf{x} + z_{e'}\mathbf{a}^{\mathrm{T}}\mathbf{x}}. \tag{2.21}$$

where $\begin{pmatrix} x \\ y \end{pmatrix}$ are the corresponding pixels in the left image.

2.4.1 Euler-Lagrange Equations

By interchangeably fixing $a_i, i = 1...n$ and v_{AT}, we obtain the Euler-Lagrange equations which minimize the functional.

Minimization with respect to a_i.

Fixing v_{AT}, we obtain

$$\forall i, \quad \nabla_{\mathrm{a_i}}(E_D + \alpha v_{AT}^2 E_S) = 0. \tag{2.22}$$

the variation of the data term with respect to the model parameter function $\mathrm{a_i}$ is given by

$$\nabla_{\mathrm{a_i}} E_D(u, v) = 2\Psi'\left(I_z^2\right) I_z \nabla_{\mathrm{a_i}} I_z, \tag{2.23}$$

where

$$\nabla_{\mathrm{a_i}} I_z = \lambda^2 x_i (x_{e'} \mathbf{h_3}^{\mathrm{T}}\mathbf{x} - z_{e'}\mathbf{h_1}^{\mathrm{T}}\mathbf{x}) I_x^+ + \lambda^2 x_i (y_{e'}\mathbf{h_3}^{\mathrm{T}}\mathbf{x} - z_{e'}\mathbf{h_2}^{\mathrm{T}}\mathbf{x}) I_y^+, \tag{2.24}$$

and

$$I_x^+ = I_x(x + u, y + v, t + 1) \tag{2.25}$$

$$I_y^+ = I_y(x + u, y + v, t + 1). \tag{2.26}$$

For the smoothness term, the Euler-Lagrange equations are

$$\nabla a_i E_s = 2 v_{AT} \Psi\left(\sum_{i=1}^{n} ||\nabla a_i||^2\right) + 2 v_{AT}^2 \operatorname{div}\left(\Psi'\left(\sum_j ||\nabla a_j||^2\right) \nabla a_i\right) \tag{2.27}$$

thus, the energy is minimized by solving the nonlinear system of equations

$$\Psi'\left(I_z^2\right) I_z \nabla_{a_i} I_z - \alpha \nabla\left(v_{AT}^2 \Psi'\left(\sum_{i=1}^{n} ||\nabla a_i||^2\right)\right)^T \nabla a_i - \alpha v_{AT}^2 \operatorname{div}\left(\Psi'\left(\sum_j ||\nabla a_j||^2\right) \nabla a_i\right) = 0. \tag{2.28}$$

Minimization with respect to v_{AT}.

Fixing a_i, we obtain

$$2\alpha v_{AT}\Psi\left(\sum_{i=1}^{n}||\nabla a_i||^2\right)+2\epsilon_1(v_{AT}-1)-\epsilon_2\Delta v_{AT}=0 \tag{2.29}$$

2.4.2 Implementation

Minimization with respect to v_{AT} is straightforward, as the equations are linear with respect to v_{AT}, therefore we will only elaborate on the minimization with respect to a_i

The nonlinear Euler-Lagrange equation minimizing a_i, are linearized by adopting three embedded loops, similarly to [163]. First, the warped image gradient (I_x^+, I_y^+) is frozen, and so is λ. At each iteration k, we have

$$(\nabla_{\mathrm{a_i}} I_z)^k = x_i d^k \tag{2.30}$$

where

$$\begin{aligned} d^k &= (\lambda^k)^2(x_{e'}\mathbf{h_3}^{\mathrm{T}}\mathbf{x}-z_{e'}\mathbf{h_1}^{\mathrm{T}}\mathbf{x})(I_x^+)^k \\ &\quad +(\lambda^k)^2(y_{e'}\mathbf{h_3}^{\mathrm{T}}\mathbf{x}-z_{e'}\mathbf{h_2}^{\mathrm{T}}\mathbf{x})(I_y^+)^k, \end{aligned}$$

and the following approximation is made using first order Taylor expansions

$$I_z^{k+1}\approx I_z^k+d^k\sum_{i=1}^{3}x_i d\mathrm{a_i}^k \tag{2.31}$$

where

$$d\mathbf{a}^k=\mathbf{a}^{k+1}-\mathbf{a}^k. \tag{2.32}$$

The system of equations (2.28) becomes

$$\Psi'\left((I_z^{k+1})^2\right)\left(I_z^k+d^k\sum_{j=1}^{3}x_j d\mathrm{a_j}^k\right)x_i d^k-\alpha\ \mathrm{div}\left(\Psi'\left(\sum_j||\nabla \mathrm{a_j}^{k+1}||^2\right)\nabla \mathrm{a_i}^{k+1}\right)=0.$$

A second loop with superscript l is added to cope with the nonlinearity of Ψ'.

$$(\Psi')^{k,l}_{\mathrm{Data}}\left(I_z^k+d^k\sum_{j=1}^{3}x_j d\mathrm{a_j}^{k,l+1}\right)x_i d^k-\alpha\ \mathrm{div}\left((\Psi')^{k,l}_{\mathrm{Smooth}}\nabla \mathrm{a_i}^{k,l+1}\right)=0$$

where

$$(\Psi')^{k,l}_{\text{Data}} = \Psi'\left(\left(I_z^k + d^k \sum_{i=1}^{3} x_i d\mathrm{a_i}^{k,l}\right)^2\right), \quad (\Psi')^{k,l}_{\text{Smooth}} = \Psi'\left(\sum_j ||\nabla \mathrm{a_j}^{k,l}||^2\right).$$

At this point, the system of equations is linear and sparse in the spatial domain. The solution $\mathbf{a}$, as well as the diffusivity term v_{AT} are obtained through Gauss-Seidel iterations. In the case of the Ambrosio-Tortorelli regularization term, the diffusion term of the equation is modulated by v_{AT}.

2.5 Experimental results

We now demonstrate motion estimation results using our algorithm, both visually and in terms of the average angular error (AAE). No post-processing was applied to the optical flow field obtained after energy minimization. The algorithm was tested on image pairs from the Middlebury optical flow test set [17], as well as all images with a static scene and publicly available ground truth optical flow from the training set. Results from the training set are presented in Table 2.1.

The flow, parameters, and diffusivity field resulting from our method are presented in Figure 2.3. The optical flow is shown with color encoding and a disparity map.

Results from the test set are shown in Figure 2.2. A smoothness parameter α of 400 was used in all experiments, and the Ambrosio-Tortorelli coefficients were set to $\epsilon_1 = 20, \epsilon_2 = 5 \times 10^{-5}$. The proposed method produced the best results to date on the static Yosemite and Urban scenes. The algorithm is not designed, however, for non-static scenes, where the computed epipolar lines have no meaning. One possible solution to this shortcoming is to return to a 2D search [237]. Such a combined approach is left for future work.

In the Teddy and Grove test images, the initialization of our algorithm introduced errors in significant parts of the image, which our method could not overcome. This behavior is related to the problem of finding a global minimum for the optical flow, which is known to have several local minima. Improving the global convergence using

	AAE	STD
Grove2	2.41	7.16
Grove3	5.53	15.76
Urban2	2.15	9.22
Urban3	3.84	16.88
Venus	4.29	12.01
Yosemite	0.85	1.24

(a) Middlebury training set

Method	AAE	Method	AAE
Brox et al. [42]	1.59	Roth/Black [190]	1.43
Mémin/Pérez [154]	1.58	Valgaerts et al. [232]	1.17
Bruhn et al. [44]	1.46	Nir et al. [163]	1.15
Amiaz et al. [7]	1.44	Our method	0.85

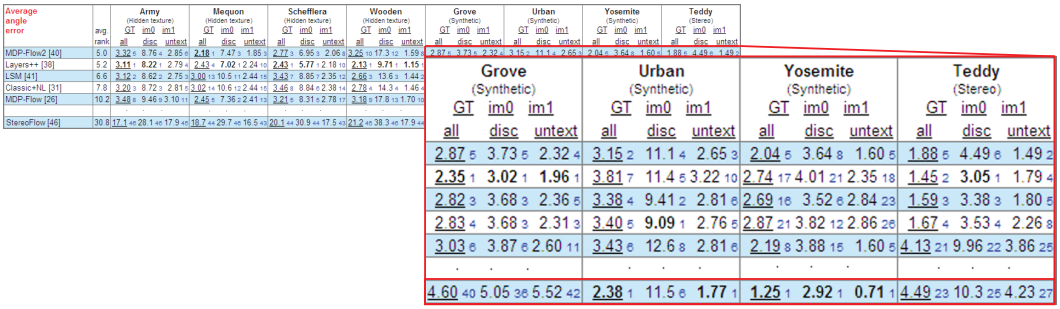
(b) Yosemite sequence

Table 2.1: AAE comparison for static scenes of the Middlebury training set and for the Yosemite sequence

discrete graph-based techniques, has been the focus of several papers (see [120, 133, 141], for example), and is beyond the scope of this work. We expect better initialization to improve the accuracy to that of the Yosemite and Urban image pairs.

Our optical flow estimation for the Yosemite and Urban sequences gives the best results to date, achieving an AAE of 1.25 for the Yosemite sequence test pair and 2.38 for the Urban sequence, as shown in Figure 2.2. When the fundamental matrix estimate was improved (by estimating from the ground truth optical flow), we reduced the AAE to 0.66 for Yosemite!

It is interesting to look at the results obtained for scenes with planar regions, such as the Urban2 (Figure 2.3) image pair. In Urban2, the scene is composed of many planar patches, modeled by constant patches in the model parameters. In both these scenes, as



Average angle error	avg. rank	Grove (Synthetic) GT all	Grove (Synthetic) GT disc	Grove (Synthetic) GT untext	Grove (Synthetic) im0 all	Grove (Synthetic) im0 disc	Grove (Synthetic) im0 untext	Grove (Synthetic) im1 all	Grove (Synthetic) im1 disc	Grove (Synthetic) im1 untext
MDP-Flow2 [40]	5.0	2.87 5	3.73 5	2.32 4	3.15 2	11.1 4	2.65 3	2.04 5	3.64 8	1.60 5
Layers++ [38]	5.2	2.35 1	3.02 1	1.96 1	3.81 7	11.4 6	3.22 10	2.74 17	4.01 21	2.35 18
LSM [41]	6.6	2.82 3	3.68 3	2.36 5	3.38 4	9.41 2	2.81 6	2.69 16	3.52 6	2.84 23
Classic+NL [31]	7.8	2.83 4	3.68 3	2.31 3	3.40 5	9.09 1	2.76 5	2.87 21	3.82 12	2.86 26
MDP-Flow [26]	10.2	3.03 6	3.87 6	2.60 11	3.43 6	12.6 8	2.81 6	2.19 8	3.88 15	1.60 5
		.	.	.	.	.	.	.	.	.
StereoFlow [46]	30.8	4.60 40	5.05 36	5.52 42	2.38 1	11.5 6	1.77 1	1.25 1	2.92 1	0.71 1

Average angle error	avg. rank	Urban (Synthetic) GT all	Urban (Synthetic) GT disc	Urban (Synthetic) GT untext	Urban (Synthetic) im0 all	Urban (Synthetic) im0 disc	Urban (Synthetic) im0 untext	Urban (Synthetic) im1 all	Urban (Synthetic) im1 disc	Urban (Synthetic) im1 untext
MDP-Flow2 [40]	5.0	3.15 2	11.1 4	2.65 3	2.04 5	3.64 8	1.60 5	1.88 5	4.49 6	1.49 2
Layers++ [38]	5.2	3.81 7	11.4 6	3.22 10	2.74 17	4.01 21	2.35 18	1.45 2	3.05 1	1.79 4
LSM [41]	6.6	3.38 4	9.41 2	2.81 6	2.69 16	3.52 6	2.84 23	1.59 3	3.38 3	1.80 5
Classic+NL [31]	7.8	3.40 5	9.09 1	2.76 5	2.87 21	3.82 12	2.86 26	1.67 4	3.53 4	2.26 8
MDP-Flow [26]	10.2	3.43 6	12.6 8	2.81 6	2.19 8	3.88 15	1.60 5	4.13 21	9.96 22	3.86 25
		.	.	.	.	.	.	.	.	.
StereoFlow [46]	30.8	2.38 1	11.5 6	1.77 1	1.25 1	2.92 1	0.71 1	4.49 23	10.3 25	4.23 27

Average angle error	avg. rank	Yosemite (Synthetic) GT all	Yosemite (Synthetic) GT disc	Yosemite (Synthetic) GT untext	Yosemite (Synthetic) im0 all	Yosemite (Synthetic) im0 disc	Yosemite (Synthetic) im0 untext	Yosemite (Synthetic) im1 all	Yosemite (Synthetic) im1 disc	Yosemite (Synthetic) im1 untext
MDP-Flow2 [40]	5.0	2.04 5	3.64 8	1.60 5	1.88 5	4.49 6	1.49 2	2.87 5	3.73 5	2.32 4
Layers++ [38]	5.2	2.74 17	4.01 21	2.35 18	1.45 2	3.05 1	1.79 4	2.35 1	3.02 1	1.96 1
LSM [41]	6.6	2.69 16	3.52 6	2.84 23	1.59 3	3.38 3	1.80 5	2.82 3	3.68 3	2.36 5
Classic+NL [31]	7.8	2.87 21	3.82 12	2.86 26	1.67 4	3.53 4	2.26 8	2.83 4	3.68 3	2.31 3
MDP-Flow [26]	10.2	2.19 8	3.88 15	1.60 5	4.13 21	9.96 22	3.86 25	3.03 6	3.87 6	2.60 11
		.	.	.	.	.	.	.	.	.
StereoFlow [46]	30.8	1.25 1	2.92 1	0.71 1	4.49 23	10.3 25	4.23 27	4.60 40	5.05 36	5.52 42

Average angle error	avg. rank	Teddy (Stereo) GT all	Teddy (Stereo) GT disc	Teddy (Stereo) GT untext	Teddy (Stereo) im0 all	Teddy (Stereo) im0 disc	Teddy (Stereo) im0 untext	Teddy (Stereo) im1 all	Teddy (Stereo) im1 disc	Teddy (Stereo) im1 untext
MDP-Flow2 [40]	5.0	1.88 5	4.49 6	1.49 2	2.87 5	3.73 5	2.32 4	3.15 2	11.1 4	2.65 3
Layers++ [38]	5.2	1.45 2	3.05 1	1.79 4	2.35 1	3.02 1	1.96 1	3.81 7	11.4 6	3.22 10
LSM [41]	6.6	1.59 3	3.38 3	1.80 5	2.82 3	3.68 3	2.36 5	3.38 4	9.41 2	2.81 6
Classic+NL [31]	7.8	1.67 4	3.53 4	2.26 8	2.83 4	3.68 3	2.31 3	3.40 5	9.09 1	2.76 5
MDP-Flow [26]	10.2	4.13 21	9.96 22	3.86 25	3.03 6	3.87 6	2.60 11	3.43 6	12.6 8	2.81 6
		.	.	.	.	.	.	.	.	.
StereoFlow [46]	30.8	4.49 23	10.3 25	4.23 27	4.60 40	5.05 36	5.52 42	2.38 1	11.5 6	1.77 1

Figure 2.2: Average angular error values of our algorithm, compared on the middlebury test set. The smoothness coefficient was set to $\alpha = 400$ in all experiments. Red marks the row of the suggested algorithm.

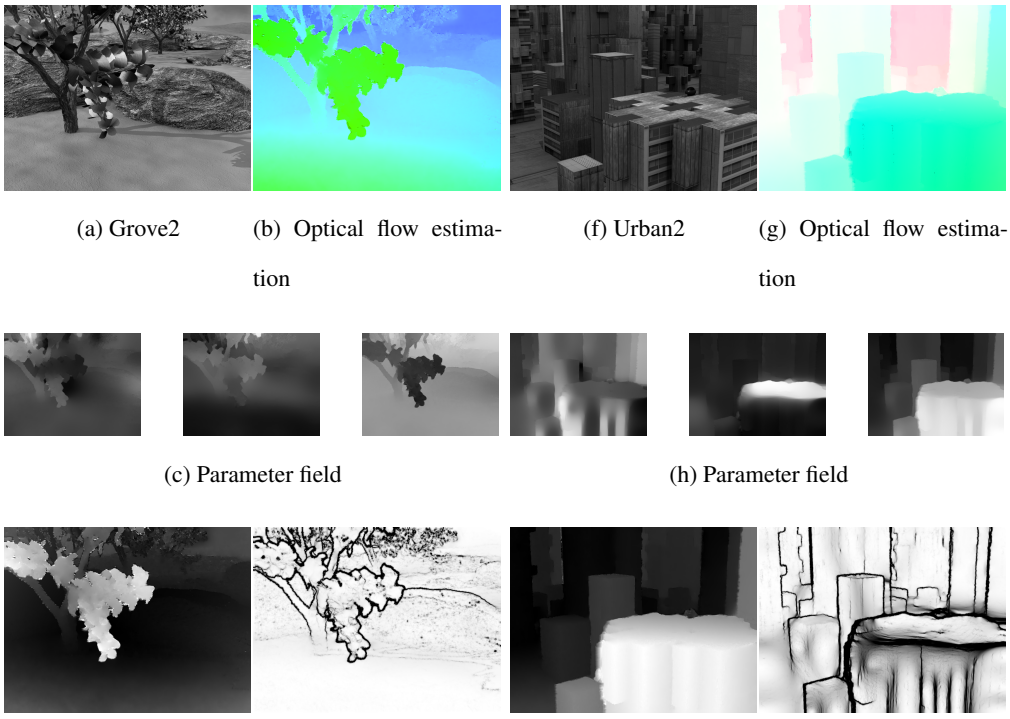
well as others, the resulting diffusivity field clearly marks the contours of planar regions in the image such as the buildings in Urban2 and the tree and soil ridges in Grove2.

2.6 Conclusions

In this chapter we have presented a new method for optical flow computation, based on a local parametric model of homography. We note that we use the minimal number of degrees of freedom while still having a model the encompasses the complete range of 2D motions spanned by stereo motion. The method is applicable to static scenes and retrieves meaningful local motion parameters related to the scene geometry. The over-parameterized framework allows future extensions to models involving dynamic scenes as well.

An interesting aspect of our energy functional, which was already mentioned in [163], is that given a carefully selected over-complete parameter field, the different parameters support each other to find a smooth piecewise constant parameter patches, while the incorporated Ambrosio-Tortorelli scheme prevents diffusion across discontinuities. Furthermore, the Ambrosio-Tortorelli scheme allows us to combine regularization and segmentation, resulting in a physically meaningful regularization process, while minimizing the dependency on the relative scaling of the coefficients.

Finally, although the performance demonstrated already goes beyond the latest pub-



(a) Grove2 (b) Optical flow estimation (f) Urban2 (g) Optical flow estimation

(c) Parameter field (h) Parameter field

(d) Disparity estimation (e) Diffusivity function (i) Disparity estimation (j) Diffusivity function

Figure 2.3: Grove2 and Urban2 sequence results

lished results, there is still much gain to be expected from better fundamental matrix estimation and algorithm initialization, as well as better navigating the optimization landscape. In addition, when more than two frames are available and the camera pose is known, augmenting the model with time-smoothness is expected to systematically improve the results.

Chapter 3

Group-valued Regularization for Motion Segmentation of Articulated Shapes

3.1 Introduction

Articulated objects segmentation is a key problem in biomechanics [3], mechanical engineering, computer vision [13, 101, 128, 150, 198], and computer graphics [11, 115, 126, 132, 224, 247, 248]. Related problems of deformation analysis [8, 243] and motion segmentation [9, 64] have also been studied extensively in these disciplines. Algorithms solving these problems try to infer the articulated motion of an object, given several instances of the object in different poses. Simultaneously, the segmentation of the object into rigid parts takes place along with motion estimation between the corresponding parts in the various poses.

Most motion analysis techniques make some assumptions on the object to be segmented. These usually concern the number or location of rigid parts in the articulated object. This can be in the form of a skeleton describing the topology of the shape, or some other prior on the object structure. Such priors are usually formulated in an *ad hoc* manner, but not based on the kinematic model commonly assumed for near-rigid

objects [3, 8]. In cases where such a prior is not available for the objects in question, or where assumptions about the data are only approximate, this can lead to errors in the segmentation and motion estimation.

Another common assumption, especially in graphics applications, is that of known correspondences. In computer graphics, the problem is usually referred to as *dynamic mesh segmentation*. While a matching technique between poses can be combined with existing motion segmentation tools, a more complete formulation for motion segmentation should handle the correspondence problem implicitly.

Clearly, the above assumptions are often too limiting in real-world applications, and should be avoided as part of the basic problem formulation. We would like instead to apply the intuition often used when studying real-life near-rigid objects, about the existence of a representative rigid motion existing for each body part. We wish, however, to avoid detecting the articulated parts in advance. Furthermore, in some object, a clear partition into rigid parts may not exist for all of the surface. We wish to obtain reasonable results in such a case. In other words, we would like to obtain a "soft" segmentation of the surface, without knowing the number or location of regions in advance, an explicit analysis of the surface features, or having additional priors on the various object parts. Also, we strive towards a formulation of motion segmentation that incorporates an implicit handling of the correspondence problem, given a reasonable initialization.

3.1.1 Main Contribution.

In this chapter we try to remedy the shortcoming of existing approaches to articulated motion estimation by combining the two tasks of motion estimation and segmentation into a single functional. This scheme has been described in a recent conference paper [182] and we now slightly expand upon it. Unlike existing methods, we propose a principled variational approach, attempting to find a rigid transformation at each surface point, between the instance surfaces, such that the overall transformation is described by a relatively sparse set of such transformations, each matching a rigid part of the

object. The functional we propose regularizes the motion between the surfaces, and is guided by the fact that the parameters of the motion transformations

(i) should describe the motion at each point with sufficient accuracy.

(ii) should vary smoothly within the (unknown) rigid parts.

(iii) can vary abruptly between rigid parts.

The main contribution of this chapter are in these :

A new framework: First, we propose an axiomatic variational framework for articulated motion segmentation. While focusing on the segmentation problem in this chapter, our framework is more general and the proposed functionals can be easily incorporated into other applications such as motion estimation, tracking, and surface denoising.

Variational segmentation: We claim that using the right parameterization, taken from the specific domain of rigid motion analysis, we can formulate the articulated motion segmentation problem as a generalization of classical tools in variational computer vision. This allows for an elegant and simple solution within the proposed framework, obtaining results competitive with domain-specific state-of-the-art tools.

A novel visualization algorithm: Third, we suggest a spatially-coherent algorithm for spatial visualization of group valued data on manifolds, which draws from the same variational principles.

3.1.2 Relation to Prior Work.

Several previous works have attempted motion based segmentation of surfaces. We mention but a few of these. Kompatsiaris et al. [128] use an estimation of the rigid motion at each segment in order to segment the visible surface in a coarse-to-fine manner. Arcila et al. [11] iteratively refine the segmentation for segments whose transformation error is too large. Wuhrer and Brunton [247] use a dual tree representation of the surface with weights between triangles set according to the dihedral angles. Lee et al.

[132] use a similar graph-based formulation, looking at deformation matrices around each triangle.

The scheme we propose involves diffusing the transformations between poses along the surface, in the spirit of the Ambrosio-Tortorelli scheme ([5], see also [19]) for Mumford-Shah segmentation [157]. The diffusion component of our scheme is a diffusion process of Lie-group elements, which has recently attracted significant attention in other applications [72, 96, 215]. In diffusing transformations on the surface, our work is similar to that of Litke et al. [140], although the parameterization of the motion and of the surface is different. In addition, we do not make an assumption on the surface topology; to that end, the proposed method diffuses transformations along the surface, rather than representing the surface in an evenly sampled 2D parametrization plane. When dealing with real-life deformable objects that seldom admit regular global parametrization, such an assumption could be too restrictive.

The idea of combining soft segmentation and motion estimation has been attempted before in the case of optical flow computation (see, e.g., [6, 45]). In optical flow fields, however, the motion field is merely expected to be piecewise smooth. For truly articulated objects one would expect piecewise-constant flow fields, when expressed in the correct parametrization.

Finally, the functional can be extended with priors from general mesh segmentation techniques. These are usually based on the geometry of the surface itself, and obtain remarkable results for a variety of objects. We point the reader to [14, 56, 117, 204], and references therein, for additional examples of mesh segmentation algorithms. We do not, however, use an additional prior as such an addition will prevent the isolated examination of the principles shown in this chapter.

3.2 Problem Formulation

We now proceed to define the problem we try to solve and the proposed model.

3.2.1 Articulation Model

We denote by X a 2-manifold representing a three-dimensional shape. We assume X to have several embeddings into $\mathbb{R}^3$. Each of these embeddings constitutes a pose of the surface. In the following, we will denote by $\mathbf{x} : X \to \mathbb{R}^3$ the embedding of X into $\mathbb{R}^3$, and use synonymously the notation x and $\mathbf{x}$ referring to a point on the manifold and its Euclidean embedding coordinates, for a specific pose.

In the setting of rigid motion segmentation, we assume that X represents an *articulated* shape, i.e., it can be decomposed into *rigid parts* $S_1, \ldots, S_p$. These are transformed between different poses of the objects by a rigid transformation. This transformation, a rotation and a translation, is an isometry of $\mathbb{R}^3$. The rigid parts are connected by nonrigid *joints* $J_1, \ldots, J_q$, such that $X = \bigcup_{i=1}^{p} S_i \cup \bigcup_{k=1}^{q} J_k$. An *articulation* $Y = \mathbf{A}X$ is obtained by applying rigid motions $\mathbf{T}_i \in \mathrm{Iso}(\mathbb{R}^3)$ to the rigid parts, and non-rigid deformations $\mathbf{Q}_k$ to the joints, such that $\mathbf{A}X = \bigcup_{i=1}^{p} \mathbf{T}_i S_i \cup \bigcup_{k=1}^{q} \mathbf{Q}_k J_k$.

3.2.2 Motion Segmentation

The problem of *motion-based segmentation* can be described as follows: given two articulations of the shape, X and Y, extract its rigid parts. An extension to the case of multiple shape poses is straightforward. We therefore consider in the following only a pair of shapes for the sake of simplicity and without loss of generality. A strongly related question attempts to determine, given these articulations, the motion parameters linking the poses of the object.

Assuming that the correspondence between the two poses X and Y is known, given a point $x \in X$ and its correspondent point $y(x) \in Y$, we can find a motion $g \in \mathbb{G}$ such that $g\mathbf{x} = \mathbf{y}$, where $\mathbb{G}$ is some representation of coordinate transformations in $\mathbb{R}^3$. This motion g may change, in the setting described above, for each surface point. We therefore consider g to be a function $g : X \to \mathbb{G}$. We will simultaneously use $g\mathbf{x} \in \mathbb{R}^3$ to denote the action of $g(x)$ on the coordinates of the point x, as well as consider the mapping given by $g : X \to \mathbb{G}$ and its properties.

We note that typical representations of motion in $\mathbb{R}^3$ contain more than 3 degrees of freedom. In this sense, they are over-parameterized [164], and thus some measure of regularity is required in order to avoid ambiguity as well as favor a meaningful solution. On the other hand, we note that since the articulated parts of the shape move rigidly, if we choose an appropriate motion representation (as detailed below), two points $x, x' \in S_i$ will undergo the same transformation, from which it follows that $g(x)|_{x \in S_i} = \text{const}$. One possibility is to adopt a constrained minimization approach, forcing $g(X) = Y$, where $g(X)$ is a notation for the set $g(x)\mathbf{x}(x)$ for all $x \in X$. This approach, however, needs to somehow handle the set of joints, for which such a constraint may be meaningless. In general, restricting the feasible set of solutions by such constraints or even constraints involving an error in the data may be harmful for the overall result. In order to avoid this, another possible approach is to take an unconstrained, yet regularized, variational formulation,

$$\min_{g:X \to \mathbb{G}} \lambda E_{\mathrm{D}}(g) + \rho(g), \tag{3.1}$$

where ρ denotes a smoothness term operating on the motion parameters field. This term is expected to be small for fields g which are piecewise constant on the manifold X. While an appropriate parameterization of motion g, and regularization term $\rho(g)$ are crucial, we also require a data term that will encourage consistency of the transformation field g with the known surface poses. Specifically, we wish to favor a transformation field where the point $\mathbf{x}$ is taken by its transformation $g(x)$ to a point on the other surface. $E_{\mathrm{D}}(g)$ is our fitting term which measures this consistency with the data.

$$E_{\mathrm{D}}(g) = \int_X \|g(x)\mathbf{x} - \mathbf{y}(\mathbf{x})\|^2 da, \tag{3.2}$$

where $\mathbf{y}(x) \in \mathbb{R}^3$ denotes the coordinate of the point $y(x) \in Y$ corresponding to x, $g(x)$ is the transformation at x, and da is a measure on X. We have assumed in the discussion so far that the correspondence between X and Y is known, which is usually not true. We can solve for the correspondence as part of the optimization in an efficient

manner. We will mention this issue in Section 3.4.1. We use the term corresponding point $y(x)$ since, as in the case of *iterative closest point* (ICP) algorithms [31, 57], several approaches for pruning erroneous or ineffective matches exist [194].

Minimizing the functional with respect to $g, y(x)$ from a reasonable initial solution allows recovery of the articulated parts by clustering g into regions of equal value. Yet another choice of a data term is a semi-local fitting term, is a semi-local one,

$$E_{\mathrm{D,SL}}(g) = \int_X \int_{y \in \mathcal{N}(x)} \|g(x)\mathbf{x}' - \mathbf{y}(\mathbf{x}')\|^2 da' da, \tag{3.3}$$

where $\mathcal{N}(x)$ denotes a small neighborhood around the point x (we took $\mathcal{N}(\mathbf{x})$ to be the 12 nearest neighbors). This fitting term, by itself, formulates a local ICP process. The functional (3.1) equipped with the semi-local data term can be considered as the geometrical fitting equivalent of the combined global-local approach for optic flow estimation [44].

The simplest representation of motion is a *linear motion* model, affectively setting $\mathbb{G}$ to be the group of translation, or $\mathbb{G} = \mathbb{R}^3$. This results in the motion model $g\mathbf{x} = \mathbf{x} + \mathbf{t} = \mathbf{y}$ for some $\mathbf{t} \in \mathbb{R}^3$. However, such a simplistic model fails to capture the piecewise constancy of the motion field in most cases. Instead of turning to a higher order approximation model such as the affine over-parameterized model [163], or to more elaborate smoothness priors [228], we look for a relatively simple model that will capture natural, piecewise-rigid motions with a simple smoothness prior. Thus we turn to a slightly different motion model, naturally occuring in motion research.

3.2.3 Lie-Groups

One parametrization often used in computer vision and robotics [96, 126, 156, 231] is the representation of rigid motions by the Lie-group $SE(3)$ and the corresponding Lie-algebra $se(3)$, respectively. In general, given two coordinate frames, an element in $SE(3)$ describes the transformation between them. Works by Brockett [39], Park et al. [168] and Zefran et al. [234, 235] strongly relate Lie-groups, both in their global and differential description, to robotics and articulated motions. We give a very brief

introduction to the subject and refer the reader to standard literature on the subject (e.g., [97, 158]) for more information.

In this chapter, we are specifically interested in the special orthogonal (rotation) matrix group $SO(3)$ and the Euclidean group $SE(3)$ to represent rigid motions. These can be represented in matrix forms, where $SO(3)$ is given as

$$SO(3) = \left\{ \mathbf{R} \in \mathbb{R}_{3\times 3}, \mathbf{R}^T\mathbf{R} = \mathbf{I} \right\}, \tag{3.4}$$

and $SE(3)$ is given by

$$SE(3) = \left\{ \begin{pmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{pmatrix}, \mathbf{R} \in SO(3), \mathbf{t} \in \mathbb{R}^3 \right\}. \tag{3.5}$$

The Lie-algebra of $SO(3)$, $so(3)$ consists of skew-symmetric matrices,

$$so(3) = \left\{ \mathbf{A} \in \mathbb{R}_{3\times 3}, \mathbf{A}^T = -\mathbf{A} \right\}, \tag{3.6}$$

whereas the Lie-algebra of $SE(3)$ can be identified with the group of 4×4 matrices of the form

$$se(3) = \left\{ \begin{pmatrix} \mathbf{A} & \mathbf{t} \\ \mathbf{0} & 0 \end{pmatrix}, \mathbf{A} \in so(3), \mathbf{t} \in \mathbb{R}^3 \right\}, \tag{3.7}$$

where $so(3)$ is the set of 3×3 skew-symmetric matrices. For both groups, the group action is given by matrix multiplication.

In order to obtain piecewise constant description over the surface for the relatively simple case of articulated object, we would like the points at each object part to have the same representative. Under the assumption of $\mathbb{G} = SE(3)$, this desired property holds. We note, however, that the standard parameterization of small rigid motions has 6 degrees of freedom, while the number of degrees of freedom required to describe the motion of point is mere 3. Thus, this parameterization clearly constitutes an over-parameterized motion field [163] for articulated surfaces.

We now turn to the regularization term, $\rho(g)$, and note that the formulation given in Equation 3.1 bears much resemblance to *total variation* (TV) regularization common

in signal and image processing [192]. Total variation regularization does not, however, favor distinct discontinuity sets. This property of TV regularization is related to the *staircasing effect*. Furthermore, in the scalar case, discontinuity sets form closed curves, which may not be the case in some surfaces with large joint areas. Instead, a model that better suits our segmentation problem is the Mumford-Shah segmentation model [157]. This model can be implemented using an Ambrosio-Tortorelli scheme [5], which can be easily generalized for the case of maps between general manifolds such as maps from surfaces into motion manifolds. We further describe the regularization chosen in Section 3.3.

We also note that due to the non-Euclidean structure of the group, special care should be taken when parameterizing such a representation [96, 126, 156, 215], as discussed in Section 3.4.2.

3.3 Regularization of Group-Valued Functions on Surfaces

Ideally, we would like the transformation field defined on the articulated surface to be piecewise smooth, if not piecewise constant. Therefore, a suitable regularization of the transformation parameters is required. Since the Lie-group $\mathbb{G}$ as a Riemannian manifold, it is only natural to turn to regularization functionals defined on maps between manifolds of the form $g : X \to \mathbb{G}$.

A classical functional defined over such maps is the well-known *Dirichlet energy* [75],

$$\rho_{\mathrm{DIR}}(g) = \frac{1}{2}\int_X \langle \nabla g, \nabla g \rangle_{g(x)} da = \frac{1}{2}\int_X \mathrm{tr}\left(g^{-1}\nabla g\right)^2 da, \tag{3.8}$$

where ∇g denotes the *intrinsic gradient* of g on X, $\langle \cdot, \cdot \rangle_{g(x)}$ is the Riemannian metric on $\mathbb{G}$ at a point $g(x)$, and da is the area element of X. This functional is the more general form of the (gradient) Tikhonov regularization (for Euclidean spaces X and $\mathbb{G}$), and its properties are well defined for general manifolds, as studied by Eells [75].

Minimizers of the Dirichlet energy are called *harmonic maps*. These result from a

diffusion process, and are often used for surface matching [236, 258].

3.3.1 Ambrosio-Tortorelli Scheme

Unfortunately, the Dirichlet energy favors smooth maps defined on X, whereas our desired solution has discontinuities at the boundaries of rigid parts. We would, intuitively, want to prevent diffusion across these discontinuity curves. This can be obtained by adding a diffusivity function $v_{AT} : X \to [0, 1]$ to the Dirichlet functional, leading to the generalized Ambrosio-Tortorelli scheme [5] for Mumford-Shah regularization [157].

$$\rho_{\mathrm{AT}}(g) = \int_X \left(\frac{1}{2} v_{AT}^2 \langle \nabla g, \nabla g \rangle_g + \epsilon \langle \nabla v_{AT}, \nabla v_{AT} \rangle + \frac{(1 - v_{AT})^2}{4\epsilon} \right) da, \tag{3.9}$$

where ϵ is a small positive constant. This allows us to extend our outlook in several ways. The Mumford-Shah functional replaces the notion of a set of regions with closed simple boundary curves with that of a general discontinuity sets. It furthermore generalizes our notion of constant value regions with that of favored smoothness inside the areas defined by these discontinuity curves. This is in order to handle objects which deviate from articulated motion, for example in flexible regions or joints.

Furthermore, the generalized Ambrosio-Tortorelli scheme allows us to explicitly reason about places in the flow where the nonlinear nature of the data manifold manifests itself. Suppose we have a solution (g^*, v_{AT}^*) satisfying our piecewise-constancy assumptions of g, and a diffusivity function with 0 at region boundaries and 1 elsewhere. At such a solution, we expect two neighboring points which belong to different regions to have a very small diffusivity value v_{AT} connecting them, effectively nullifying the interaction between far-away group elements which is dependent on the mapping used for the logarithm map at each point, and hence can be inaccurate [112, 156]. While such a solution (g^*, v_{AT}^*) may not be a minimizer of the functional, it serves well to explain the intuition motivating the choice of the functional.

3.3.2 Diffusion of Lie-Group Elements

In order to efficiently compute the Euler-Lagrange equation corresponding to the generalized Ambrosio-Tortorelli functional (3.9), we transform the neighborhood of each point into the corresponding Lie-algebra elements before applying the diffusion operator. Using Lie-algebra representation of differential operators for rigid motion has been used before in computer vision [215], numerical PDE computations [112], path planning and optimal control theory [126, 156].

The Euler-Lagrange equation for the generalized Dirichlet energy measuring the map between two manifolds is given as [75]

$$\Delta_X g^\alpha + \Gamma^\alpha_{\beta\gamma} \left\langle \nabla g^\beta, \nabla g^\gamma \right\rangle_{g(x)} = 0, \tag{3.10}$$

where $\alpha, \beta\, \gamma$ enumerate the local coordinates of our group manifold, $se(3)$, and we use Einstein's notation according to which corresponding indices are summed over. $\Gamma^\alpha_{\beta\gamma}$ are the *Christoffel symbols* of $SE(3)$, which express the Riemannian metric's local derivatives. We refer the reader to [70] for an introduction to Riemannian geometry. Finally, Δ_X denotes the Laplace-Beltrami operator on the surface X.

In order to avoid computation of the Christoffel symbols, we transform the point and its neighbors using the logarithm map at that point in $SE(3)$. The diffusion operation is now affected only by the structure of the surface X. After applying the diffusion operator, we use the exponential map in order to return to the usual representation of the transformation. While this approach may suffer at discontinuities, where the logarithm and exponential maps are less accurate, it is at these continuities that we expect the diffusivity function v_{AT} to be very small, perventing numerical instability. In practice, as we will demonstrate, this did not a significant problem.

3.4 Numerical Considerations

We now describe the algorithm for articulated motion estimation based on the minimization of the functional

$$E(g, v_{AT}) = \lambda E_{\mathrm{DATA}}(g) + \rho_{AT}(g, v_{AT}), \tag{3.11}$$

where $E_{\mathrm{DATA}}(g)$ is the matching term defined by Equation 3.2, and $\rho_{AT}(g, v_{AT})$ is defined in Equation 3.9. The main steps of the algorithm are outlined as Algorithm 3.1. Throughout the algorithm we sample $g(x)$ based on the first surface, given as a mesh with vertices $\{x_i\}_{i=1}^{N}$, and an element from $SE(3)$ defined at each vertex. The triangulation of the mesh is used in Subsection 3.5.1 merely as a mean of obtaining a more consistent numerical diffusion operator, and can be avoided, for example by point-cloud based Laplacian approximations [24]. Special care is made in the choice of coordinates during the optimization as explained in Section 3.4.2.

3.4.1 Initial Correspondence Estimation

As in other motion segmentation and registration algorithms, some initialization of the matching between the surfaces must be used. One approach [11] is to use nonrigid surface matching for initialization. Another possibility, in the case of high framerate range scanners [248], is to exploit temporal consistency by 3D tracking. Yet another possible source for initial matches incorporates motion capture marker systems. Such sparse initial correspondence lends itself to interpolation of the motion field, in order to initialize a local ICP algorithm, and match the patch around each source point to the target mesh. In Figure 3.4, we use 30 matched points for initialization. This number of points is within the scope of current motion capture marker systems, or of algorithms for global nonrigid surface matching such as spectral methods [114, 150, 179, 193], or the *generalized multidimensional scaling* (GMDS) algorithm [40].

We expect that a better initial registration, as can be obtained e.g. using a smoothness assumption, or by pruning unsuitable candidates [194], will reduce the number of

markers needed.

3.4.2 Diffusion of Lie-Group Elements

Rewriting the optimization over the functional in Equation 3.11 in a fractional step approach [249], we update the parameters w.r.t. each term of the functional in a suitable representation. The treatment of regularized data fitting in a fractional step approach with respect to different forces has been used before for rigid body motion [51], and is also similar to the approach taken by Thirion's demons algorithm [170, 223] for registration.

Using the transformation described in Section 3.3, the update step with respect to the regularization now becomes

$$g^{k+1/2} = g^k \exp\left(-dt\frac{\delta\rho_{\mathrm{AT}}}{\delta\tilde{g}}\right), v_{AT}^{k+1} = v_{AT}^k - dt\frac{\delta\rho_{\mathrm{AT}}}{\delta v_{AT}} \tag{3.12}$$

where $\exp(A) = I + A + A^2/2! + A^3/3! + \ldots$ denotes the matrix exponential, $\tilde{g}$ denotes the logarithm transform of g, and dt denotes the time step. $\frac{\delta\rho_{\mathrm{AT}}}{\delta\tilde{g}}$ denotes the variation of the regularization term $\rho_{AT}(g)$ w.r.t. the Lie-algebra local representation of the solution, describing the Euler-Lagrange descent direction. $g(x)$ and the neighboring transformations are parameterized by a basis for matrices in $se(3)$, after applying the logarithm map at $g(x)$. The descent directions are given by

$$\begin{aligned} \frac{\delta\rho_{\mathrm{AT}}}{\delta\tilde{g}_i} &= v_{AT}^2\Delta_X(\tilde{g}_i) + v_{AT}\left\langle\nabla v_{AT}, \nabla\tilde{g}_i\right\rangle \\ \frac{\delta\rho_{\mathrm{AT}}}{\delta v_{AT}} &= \left\langle\nabla g, \nabla g\right\rangle_{g(x)} v_{AT} + 2\epsilon\Delta_X(v_{AT}) + \frac{(v_{AT}-1)}{2\epsilon}, \end{aligned} \tag{3.13}$$

where $\tilde{g}_i$ denote the components of the logarithmic representation of g. The discretization we use for Δ_X for triangulated meshes is a cotangent one suggested by [69], which has been shown to be convergent for relatively smooth and well-parameterized surfaces. It is expressed as

$$\Delta_X(u) \approx \frac{3}{\mathcal{A}_i}\sum_{j\in\mathcal{N}_1(i)}\frac{\cot\alpha_{ij} + \cot\beta_{ij}}{2}\left[u_j - u_i\right], \tag{3.14}$$

for a given function u on the surface X, where $\mathcal{N}_1(i)$ denotes the mesh neighbors of point i, and α_{ij}, β_{ij} are the angles opposing the edge ij in its neighboring faces. $\mathcal{A}_i$ denotes the area of the 1-ring around i in the mesh. For unstructured meshes (point-clouds), we use the Laplacian approximation suggested by Belkin et al. [24]. After a gradient descent step w.r.t. the diffusion term, we take a step w.r.t. the data term.

$$g^{k+1} = P_{SE(3)}\left(g^{k+1/2} - dt\frac{\delta E_{DATA}}{\delta g}\right), \tag{3.15}$$

where $P_{SE(3)}(\cdot)$ denotes a projection onto the group $SE(3)$ obtained by correcting the singular values of the rotation matrix [26]. We compute the gradient w.r.t. a basis for small rotation and translation matrices comprised of the regular basis for translation and the skew-matrix approximation of small rotations. We then reproject the update onto the manifold. This keeps the inaccuracies associated with the projecting manifold-constrained data [52, 96, 112, 156] at a reasonable level, and leads to a first-order accuracy method. As noted by Belta and Kumar [25] in the context of trajectory planning and ODEs over Lie-groups, this method is reasonably accurate. In practice the time-step is limited in our case by the data-fitting ICP term and the explicit diffusion scheme. We expect improved handling of these terms to allow faster implementation of the proposed method.

Finally, we note that we may not know in advance the points $y(x)$ which match X in Y. The correspondence can be updated based on the current transformations in an efficient manner similarly to the ICP algorithm. In our implementation we used the ANN library [12] for approximate nearest-neighbor search queries. We did not incorporate, however, any selective pruning of the matching candidates. These are often used in order to robustify such the ICP algorithm against ill-suited matches but are beyond the scope of this chapter.

Algorithm 3.1 Articulated Surface Segmentation and Matching

1: Given an initial correspondence.
2: **for** $k = 1, 2, \ldots,$ until convergence **do**
3: Update $g^{k+1/2}, v_{AT}^{k+1}$ w.r.t. the diffusion term, according to Equation 3.12.
4: Obtain g^{k+1} according to the data term, using Equation 3.15.
5: Update $y^{k+1}(x)$, the current estimated correspondence of the deformed surface.
6: **end for**

3.4.3 Visualizing Lie-Group Clustering on Surfaces

Finally, we need to mention the approach taken to visualize the transformations as the latter belong to a six-dimensional non-Euclidean manifold. Motivated by the widespread use of vector quantization in such visualizations, we use a clustering algorithm with spatial regularization. Instead of minimizing the Lloyd-Max quantization [113] cost function, we minimize the function

$$E_{VIS}(g_i, R_i) = \sum_i \int_{R_i} \|g - g_i\|^2 da + \int_{\partial R_i} v_{AT}^2(s) ds, \tag{3.16}$$

where ∂R_i denotes the set of boundaries between partition regions $\{R_i\}_{i=1}^N$, g_i are the group representatives for each region, and $v_{AT}^2(s)$ denotes the diffusivity term along the region boundary. The representation of members in $SE(3)$ is done via its embedding into $\mathbb{R}^{12}$, with some weight given to spatial location, by looking at the product space $\mathbb{R}^3 \times SE(3) \subset \mathbb{R}^{15}$. Several (about 50) initializations are performed, as is often customary in clustering, with the lowest cost hypothesis kept. The visualization is detailed as Algorithm 3.2

While this visualization algorithm coupled with a good initialization at each point can be considered as a segmentation algorithm in its own right, it is less general as it assumes a strict separation between the parts. One possible question that can be raise concerned the meaning behind vector quantization of points belonging to a manifold through its embedding into Euclidean space. In our case, since we are dealing with relatively well-clustered points (most of the points in a part move according to a single

Algorithm 3.2 Spatially-consistent clustering algorithm

1: **for** $j = 1, 2, \ldots,$ for a certain number of attempts **do**

2: Use k-means on the spatial-feature space embedding, $\mathbb{R}^3 \times SE(3) \subset \mathbb{R}^{15}$, to get an initial clustering.

3: Use the clusters in order to optimize a spatially-regularized vector quantization measure,

$$C = \min_{g_i, \partial R_i} \int_X \|g - g_i\|^2 da + \int_{\partial R_i} v_{AT}^2(s) ds,$$

where ∂R_i denotes the set of boundaries between clustered regions, g_i are the transformation representatives for each region, and $v_{AT}^2(s)$ denotes the diffusivity term along the region boundary.

4: If C is lower than the lowest C found so far, keep the hypothesis.

5: **end for**

6: **return** current best hypothesis.

transformation in $SE(3)$), the distances on the manifold are not large and are therefore well-approximated by Euclidean ones. We further note, however, that the diffusion process lowered the score obtained in Equation 3.16 in the experiments we conducted, indicating a consistency between the two algorithms in objects with well-defined rigid parts.

3.5 Results

We now show the results of our method, in terms of the obtained transformations clusters and the Ambrosio-Tortorelli diffusivity function. We first demonstrate results on standard datasets of triangulated meshes. We then demonstrate results on point cloud data.

3.5.1 Triangulated Meshes

In Figure 3.1 we show the segmentation obtained by matching two human body poses taken from the TOSCA dataset [41]. We visualize the transformations obtained using the clustering algorithm described in subsection 3.4.3. We initialized the transformations on the surface by matching the neighborhood of each surface point to the other poses using the true initial correspondence. The results of our method seem plausible, except for the missing identification of the right leg, which is due to the fact that its motion is limited between the two poses.

Figure 3.1 also demonstrates the results of comparing four poses of the same surface, this time with the patch-based data term described by (3.3). In our experiments the patch-based term gave a cleaner estimation of the motion, as is observed in the diffusivity function. We therefore demonstrate the results of minimizing the functional incorporating this data term. We also show the diffusivity function, which hints at the location of boundaries between parts, and thus justifies the assumption underlying Algorithm 3.2.

In Figure 3.2,3.3 we show the results of our algorithm on a set of 6 poses of a horse and camel surfaces taken from [216]. In this figure we compare our results to those of Wuhrer and Brunton [247], obtained on a similar set of poses with 10 frames. The results of our method seem to be quite comparable to those obtained by Wuhrer and Brunton, despite the fact that we use only 6 poses. We also note that both the diffusion scheme and the visualization algorithm gave a meaningful result for the tail part, which is not rigid and does not have a piecewise-rigid motion model.

In Figure 3.4 we demonstrate our algorithm, with an initialization of 30 simulated motion capture marker points, where the displacement is known. The relatively monotonous motion range available in the dynamic mesh sequence leads to a less complete, but still quite meaningful, segmentation of the horse, except for its head.

We also note the relatively low number of poses required for segmentation – in both Figure 3.2 and Figure 3.4 we obtain good results despite the fact that we use only a few

poses, six and eight respectively.

Finally, in Figure 3.4 we demonstrate initialization of our method based on a sparse point set, with 30 known correspondence points. The points are arbitrarily placed using farthest point sampling [92, 106]. This demonstrates a possibility of initializing the algorithm using motion capture markers, coupled with a 3D reconstruction pipeline, for object part analysis. While the large-motion examples shown in this chapter are synthetic, this example shows that the algorithm can be initialized with data obtained in a realistic setup.

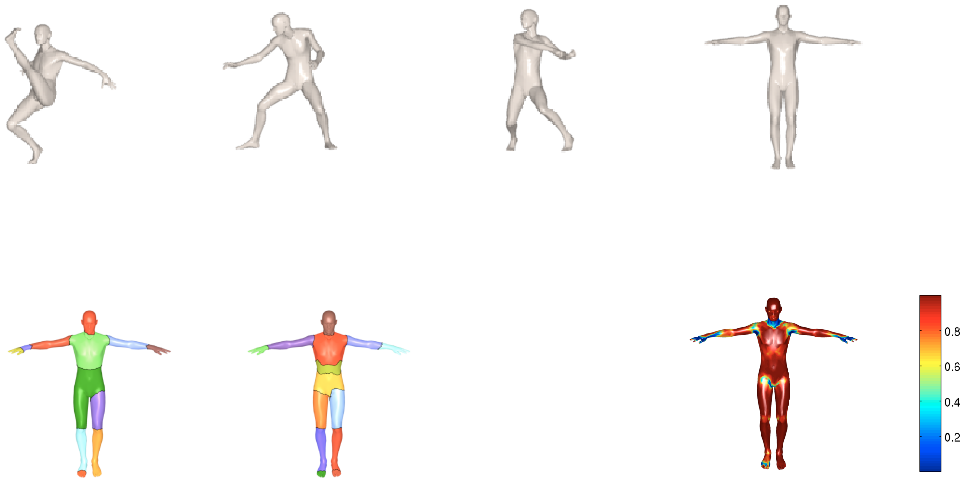


Figure 3.1: Segmenting a human figure. Top row: the set of poses used. Bottom row, left to right: the transformations obtained from the two left most poses, the transformations obtained from all four poses using Equation 3.3 as a data term, and the Ambrosio-Tortorelli diffusivity function based on four poses.

3.5.2 Point-Clouds

We now show a few results of our algorithm. We demonstrate segmentation of real point-clouds obtained from laser scanners and Microsoft Kinect depth sensors. We first describe the visualization technique used. In Figures 3.5–3.7 we use vector quantization (VQ, [142, 152]), in terms of the embedding $SE(3) \subset \mathbb{R}^{12}$, with multiple initializations in order to visualize the resulting transformations. In the examples shown, 40 initial-

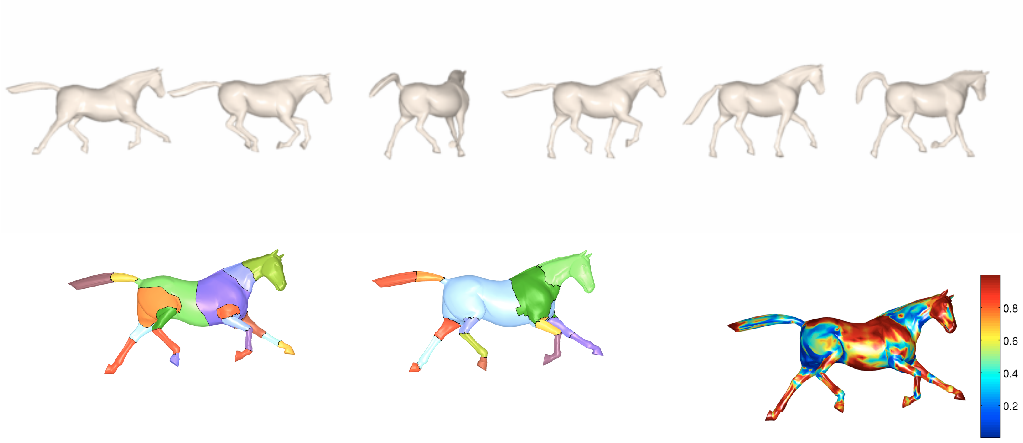


Figure 3.2: Segmenting a horse dynamic surface motion based on six different poses. Top row: the poses used. Bottom row, left to right: a visualization of the transformations of the surface obtained by our method, and the segmentation results obtained by [247], and the diffusivity function v_{AT}.

izations of vector quantization are used, at which point a minimal quantization cost is practically achieved and new hypotheses do not feature lower costs.

While vector quantization can be used in itself to provide segmentation of motion, using it over the raw estimated transformation creates various artifacts due to the noisy initial estimation and the inaccuracy of using the embedding-space (Euclidean) metric to reason about matrix-manifold distances. These are seen in the examples, where our piece-wise smooth regularization solution manages to fix these artifacts.

In addition, we show the Ambrosio-Tortorelli diffusivity field, where several of the main boundaries between parts can be seen.

In Figure 3.5 we demonstrate results from the SCAPE dataset [10]. The results are based on the algorithm with initialization using 200 initial matches, and use the first 5 frames of the dataset.

In Figures 3.6,3.7 we demonstrate results from a Kinect sensor. The transformation maps were initialized using frame-to-frame 3D tracking. Figure 3.6 demonstrates results on arbitrarily segmented part of the upper arm, with initialization based on local, patch-based, ICP between frames. For this experiment, 4 frames were taken. Figure 3.7

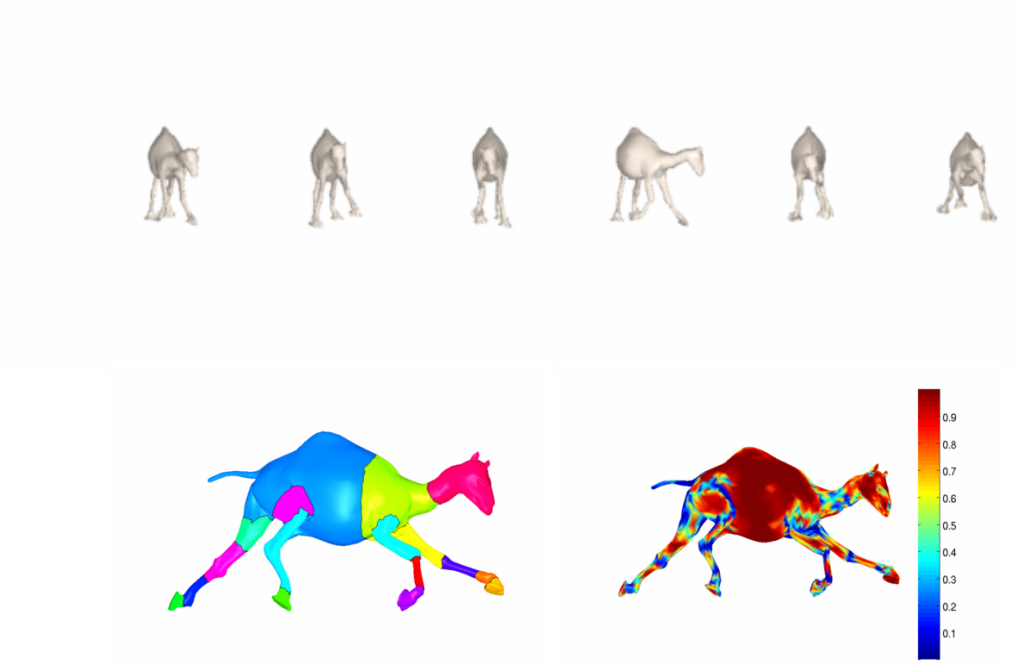


Figure 3.3: Segmenting a camel dynamic surface motion based on six different poses. Top row: the poses used. Bottom row, left to right: a visualization of the transformations of the surface obtained by our method and the diffusivity function v_{AT}.

demonstrates results on a human hand doing a waving motion, with initialization based on the coherent point drift algorithm [159], with 6 frames taken for the segmentation. These results show the applicability of the proposed framework also for analysis of depth video from noisy data sources in an automated manner.

3.6 Conclusion

In this chapter we present a new method for motion-based segmentation of articulated objects, in a variational framework with axiomatic motivation. The method is based on minimizing a generalized Ambrosio-Tortorelli functional regularizing a map from the surface onto the Lie-group $SE(3)$ – extending the generalized Dirichlet functional. We demonstrate the method's effectiveness, and compare it with state-of-the-art articulated

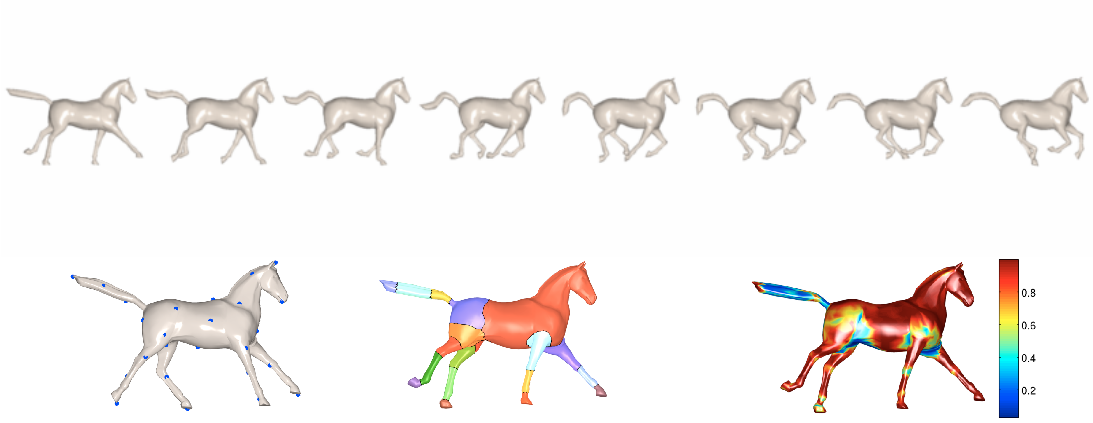


Figure 3.4: Segmenting a horse dynamic surface motion with a given sparse initial correspondences. Top row: the eight random poses used. Bottom row, left to right: the set of points used for initializing the transformations, and a visualization of the transformations obtained, and the diffusivity function v_{AT}.

motion segmentation algorithms, showing it to be competitive with domain-specific algorithms. The variational formulation of the problem allows us to easily tailor the functional to specific problems where it can be combined with domain-specific terms. In future work we intend to adapt the proposed algorithm to depth videos, which will allow more robust detection of motion estimates, and a more complete detection of the overall set of degrees of freedom. Another promising direction is to explore global

Figure 3.5: Visualization of the detected transformations before and after smoothing, using 6 frames from the SCAPE dataset. Colors show the vector quantization results on the transformations embedded into $\mathbb{R}^{12}$

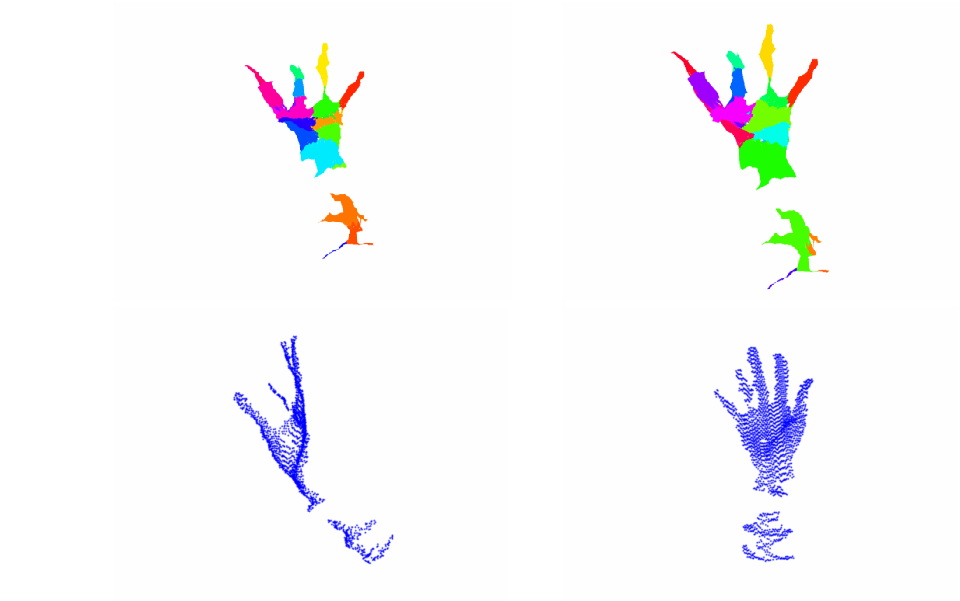


Figure 3.6: Top row: Visualization of the detected transformations before and after regularization, based on a point cloud from a Kinect sensor at 70cm, using local ICP for initialization. Colors show the vector quantization results on the transformations embedded into $\mathbb{R}^{12}$. Left: visualization of the initial solution based on local-ICP between frames. Right: the result after optimization. Bottom row: the first input frame from the front/side. Note the fragmented surface

optimization methods as well as to use the proposed framework in other applications such as surfaces denoising in order to handle noisy data.

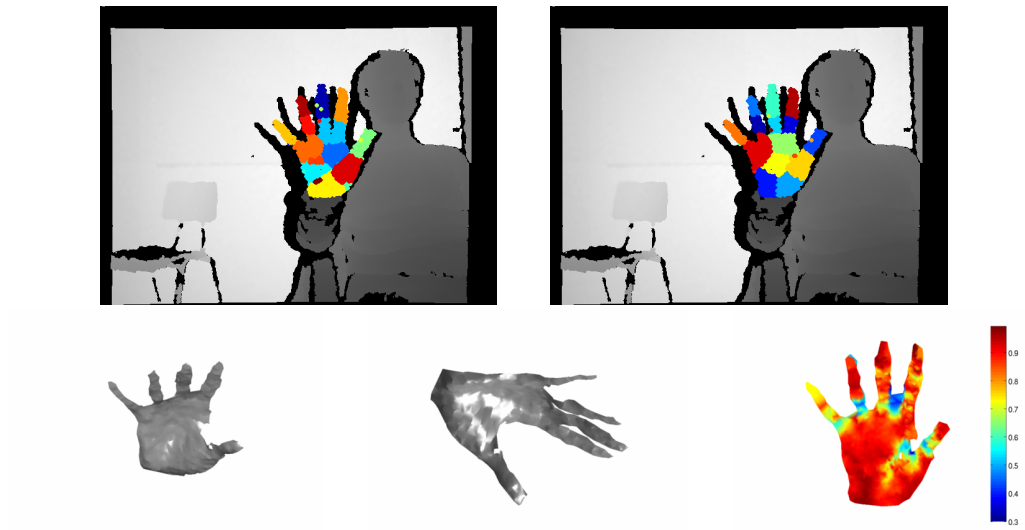


Figure 3.7: Top row: Visualization of the detected transformations before and after regularization, based on a point cloud from a Kinect sensor at 70cm. Colors show the vector quantization results on the transformations embedded into $\mathbb{R}^{12}$, greyscale shows the depth in regions that were not subject to the algorithm. Left: VQ visualization of the initial state obtained by the CPD algorithm. Right: visualization of the resulting state after optimization. Note the merged sections of the ring and middle finger, as well as additional artifacts vector quantization before the regularization. Bottom row: Left: Two surface reconstructions of the point cloud obtained from the Kinect. Note the relatively high noise level in the surface reconstructions. Right: The diffusivity function v_{AT}.

Chapter 4

Fast Regularization of Matrix-Valued Images

4.1 Introduction

Matrix-manifolds such as Lie-groups, and Matrix-valued images have become an integral part of computer vision and image processing. Matrix-manifolds and groups have been used for tracking [178, 231], robotics [52, 168, 234], motion analysis [84, 182], image processing and computer vision [47, 172, 177, 182, 241], as well as medical imaging [23, 171]. Efficient regularization of matrix-valued images is therefore highly important in the fields of for image analysis and computer vision. This includes applications such as direction diffusion [121, 220, 239] and scene motion analysis [138] in computer vision, as well as diffusion tensor MRI (DT-MRI) regularization [30, 72, 95, 211, 229] in medical imaging.

We present an augmented Lagrangian method for efficient regularization of matrix-valued images, or maps. We assume the matrix-manifold to have an efficient projection operator onto it from some embedding into a Euclidean space, and that the distortion associated with this mapping is not too large in term of the metric accompanying these spaces.

Examples for such matrix-manifolds of interest include the special-orthogonal, special-

Euclidean Lie-groups and the symmetric positive-definite matrices. We show that the augmented Lagrangian technique allows us to separate the optimization process into a regularization update step of a map onto an embedding-space, and a per-pixel projection step. An efficient regularization step is shown for the *total-variation* (TV, [192]) regularization, and a second-order regularization penalizing the Hessian norm. Both the regularization step and the projection steps are simple to compute, fast and easily parallelizable using consumer graphic processing units (GPUs), achieving real-time processing rates. The resulting framework unifies algorithms using in several application domains into one framework, since they differ only in the choice of projection operator. While such an optimization problem could have been approached by general saddle-point solvers such as [55], the domain of our problem is not convex, requiring such algorithms to be revisited in order to prove their convergence.

In order to obtain fast update steps we add two auxiliary fields, with appropriate constraints. One field approximates the gradient of the image and simplifies the total-variation cost function minimization, as done, for example, in [53, 90, 166, 219]. Another field approximates the image, but is forced during its update to stay on matrix manifold, turning the group constraint into a simple projection operator. This results in a unified framework for processing of $SO(n)$, $SE(n)$ and $SPD(n)$ images, as we describe in Section 4.3. This framework was initially presented in a recent conference paper [189] and we now expand upon it, with additional explanations and a partial convergence proof. In addition, we relate in this section the proposed algorithms to split-Bregman iterations and describe their convergence properties. In Section 4.4 we demonstrate a few results of our method, for regularization of 3D motion analysis, medical image analysis, and direction diffusion. Section 4.6 concludes the chapter.

4.2 A Short Introduction to Lie-Groups and Matrix Manifolds

We now shortly describe the matrix manifolds we deal with in our algorithm. Their structure allows us to define priors on matrix-valued data in computer vision and has

been the subject of intense research efforts, especially involving statistics of matrix-valued data [171], and regularization of matrix-valued images [229], as well as describing the dynamics of spatial processes involving *Lie-group* data [138]. Lie-groups are algebraic groups endowed with a differentiable manifold structure and an appropriate group action. We briefly describe the matrix manifolds our algorithm deals with, and refer the reader to the literature for an introduction to Lie-groups [97].

The rotations group $SO(n)$ - The group $SO(n)$ describes all rotation matrices of the n-dimensional Euclidean space,

$$SO(n) = \left\{ \mathbf{R} \in \mathbb{R}_{n\times n}, \mathbf{R}^T\mathbf{R} = \mathbf{I}, \det(\mathbf{R}) = 1 \right\}. \tag{4.1}$$

The special-Euclidean group $SE(n)$ - This group represents rigid transformations of the n-dimensional Euclidean space. This group can be thought of as the product manifold of the rotations manifold $SO(n)$ and the manifold $\mathbb{R}^n$ representing all translations of the Euclidean space. In matrix form this group is written as

$$SE(n) = \left\{ \begin{pmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{pmatrix}, \mathbf{R} \in SO(n), \mathbf{t} \in \mathbb{R}^n \right\}. \tag{4.2}$$

The symmetric positive definite set $SPD(n)$ - Another matrix manifold that has an efficient projection operator is the cone of symmetric positive definite matrices. This matrix manifold has been studied extensively in control theory (see [83] for example), as well as in the context of diffusion tensor images [171], where the matrices are used to describe the diffusion coefficients along each direction. By definition, this set is given in matrix form as

$$SPD(n) = \left\{ \mathbf{A} \in \mathbb{R}_{n\times n}, \mathbf{A} \succeq 0 \right\}. \tag{4.3}$$

4.3 An Augmented Lagrangian Regularization Algorithm for Matrix-valued Images

We now proceed to describe a fast regularization algorithm for images with matrix-valued data, referred to as Algorithm 4.1. The standard regularization problem of Lie-

groups maps is formulated in terms of the Lie-algebra,

$$\underset{u\in\mathcal{G}}{\operatorname{argmin}} \int \|u^{-1}\nabla u\| + \lambda\|u - u_0\|^2 dx, \tag{4.4}$$

where $\|\cdot\|$ is the Frobenius norm, u represents an element in an embedding of the Lie-group $\mathcal{G}$ into Euclidean space. We use the notation ∇u to denote the Jacobian of u, described as a column-stacked vector. We note that we use the same notation to represent the Lie-group element, its matrix representation, and the embedding into Euclidean space, as specified in each case we explore.

The term $\|u^{-1}\nabla u\|$ can be thought of as a regularization term placed on elements of the Lie algebra about each pixel. This formulation parallels the time derivatives in construction of Lie-group integrators [52], and defines smoothness in the tangent space of the Lie-group. Smoothness expressed in the regularization term $\|u^{-1}\nabla u\|$ is in sense of the geometry of the Lie-group, via the Lie-algebra, but this may not generalize to other matrix groups. Furthermore, its minimization is inefficient as it requires computing the logarithm and exponential maps at each pixel update. In order to obtain a fast regularization scheme that applies for all matrix groups, we look instead at the regularity of an embedding of the Lie-group into Euclidean space,

$$\underset{u\,\in\,\mathcal{G}}{\operatorname{argmin}} \int \|\nabla u\| + \lambda\|u - u_0\|^2 dx, \tag{4.5}$$

where $\|\nabla u\|$ denotes (by abuse of notation) the Frobenius norm of the Jacobian of the map from the domain ($\mathbb{R}^2$ or $\mathbb{R}^3$) into the embedding space. This allows us to consider also matrix groups that are not Lie-groups, such as symmetric positive-definite matrices and Stiefel matrices, as part of the same framework. In our formulation, elements of $SO(n)$ can be embedded into $\mathbb{R}^m, m = n^2$, and elements of $SE(n)$ can similarly be embedded into $\mathbb{R}^m, m = n(n+1)$. The elements of $SPD(n)$ can be embedded into $\mathbb{R}^m, m = n(n+1)/2$.

The rationale behind the different regularization term $\|\nabla u\|$ stems from the fact that $SO(n)$ and $SE(n)$ are isometries of Euclidean space, but such a regularization is

possible whenever the data consists of nonsingular matrices. This regularization term has also been used for SPD matrices [233]. We refer the reader to our technical report [187] for a more in-depth discussion of this important point. Next, instead of restricting u to $\mathcal{G}$, we add an auxiliary variable, v, at each point, such that $u = v$, and restrict v to $\mathcal{G}$, where the equality constraint is enforced via augmented Lagrangian terms [103, 175]. The suggested augmented Lagrangian optimization now reads

$$\begin{aligned} &\min_{v\in\mathcal{G},u\in\mathbb{R}^m} \max_{\mu} \mathcal{L}(u,v;\mu) = \\ &\min_{v\in\mathcal{G},u\in\mathbb{R}^m} \max_{\mu} \int \begin{bmatrix} \|\nabla u\| + \lambda\|u-u_0\|^2 + \\ \frac{r}{2}\|u-v\|^2 + \operatorname{tr}(\mu^T(u-v)) \end{bmatrix} dx. \end{aligned} \tag{4.6}$$

Given a fixed Lagrange multiplier μ, the minimization w.r.t. u, v can be split into alternating minimization steps with respect to u and v, both of which lend themselves to an efficient and parallel optimization. Specifically, we can further reformulate the regularization of u in the same way as Wu and Tai [245], by introducing an auxiliary variable p

$$\begin{aligned} &\min_{v\in\mathcal{G},u\in\mathbb{R}^m,p\in\mathbb{R}^{mn}} \max_{\mu} \mathcal{L}(u,v,p;\mu,\mu_2) = \\ &\min_{v\in\mathcal{G},u\in\mathbb{R}^m,p\in\mathbb{R}^{mn}} \max_{\mu} \int \begin{bmatrix} \|p\| + \lambda\|u-u_0\|^2 + \\ \frac{r}{2}\|u-v\|^2 + \operatorname{tr}(\mu^T(u-v)) + \\ \frac{r_2}{2}\|\nabla u - p\|^2 + \operatorname{tr}(\mu_2^T(\nabla u - p)) \end{bmatrix} dx, \end{aligned} \tag{4.7}$$

where μ_2 is the Lagrange multiplier associated with the constraint $p = \nabla u$, and r_2 is the related penalty coefficient.

4.3.1 Minimization w.r.t. v

The advantage of adding the auxiliary variable v is that minimization w.r.t v becomes a simple projection problem per pixel,

$$\begin{aligned}
&\underset{v\in\mathcal{G}}{\operatorname{argmin}} \frac{r}{2}\|v-u\|^2 + \operatorname{tr}(\mu^T(u-v)) \\
&= \underset{v\in\mathcal{G}}{\operatorname{argmin}} \frac{r}{2}\left\|v-\left(\frac{\mu}{r}+u\right)\right\|^2 \\
&= \underset{\mathcal{G}}{\operatorname{Proj}}\left(\frac{\mu}{r}+u\right),
\end{aligned} \tag{4.8}$$

where $\operatorname{Proj}_{\mathcal{G}}$ denotes a projection operator onto the specific matrix-group $\mathcal{G}$. The numerical update step for $SO(n)$, $SE(n)$ and $SPD(n)$ will be explicitly given later on.

4.3.2 Minimization w.r.t. u

The update step w.r.t u in Equation 4.6 is a vectorial TV denoising problem

$$\underset{u\in\mathbb{R}^m}{\operatorname{argmin}} \int \|\nabla u\| + \tilde{\lambda}\,\|u-\tilde{u}\,(u_0,v,\mu,r)\|^2\,dx, \tag{4.9}$$

with $\tilde{u} = \frac{(2\lambda u_0 + rv + \mu)}{(2\lambda + r)}$. This problem can be solved via fast minimization techniques for TV regularization of vectorial images, such as [38, 74, 89]. In our case, we regularize the image using the algorithm [219], as we now describe. In order to obtain fast optimization of the problem with respect to u, we add an auxiliary variable p, along with a constraint that $p = \nabla u$. Again, the constraint is enforced in an augmented Lagrangian manner. The optimal u now becomes a saddle point of the optimization problem

$$\begin{array}{ll}\min\limits_{\substack{u\in\mathbb{R}^m \\ p\in\mathbb{R}^{2m}}} & \max\limits_{\mu_2} \int \begin{bmatrix} \tilde{\lambda}\,\|u-\tilde{u}\,(u_0,v,\mu,r)\|^2 + \|p\| \\ +\mu_2^T(p-\nabla u) + \frac{r_2}{2}\|p-\nabla u\|^2 \end{bmatrix} dx. \end{array} \tag{4.10}$$

We solve for u using the Euler-Lagrange equation,

$$2\tilde{\lambda}(u-\tilde{u}) + (\operatorname{div}\mu_2 + r_2\operatorname{div} p) + \Delta u = 0, \tag{4.11}$$

for example, in the Fourier domain, or by Gauss-Seidel iterations. We have chosen Gauss-Seidel iterations since complete minimization of the functional is not required at each substep.

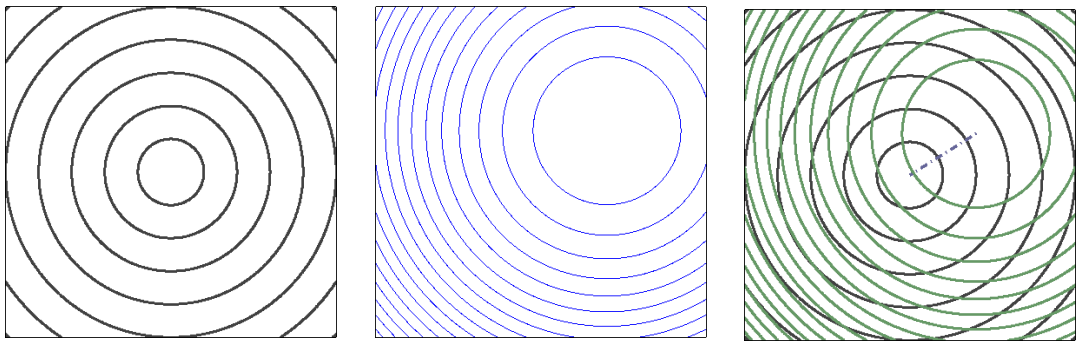
The auxiliary field p is updated by rewriting the minimization w.r.t. p as

$$\underset{p \in \mathbb{R}^{2m}}{\operatorname{argmin}} \int \|p\| + \mu_2^T p + \frac{r_2}{2}\|p - \nabla u\|^2, \tag{4.12}$$

with the closed-form solution obtained by shrinkage [219]

$$p = \frac{1}{r_2} \max\left(1 - \frac{1}{\|w\|}, 0\right) w, \quad w = r_2 \nabla u - \mu_2. \tag{4.13}$$

It is easy to see that the solution should lie on the segment connecting 0 and w, see Figure 4.1. Hence the solution comes from minimizing a relatively 1D problem. See [219] for more details, where the level-lines of the two terms of the functional involving p are shown.



Regularization Term | Coupling Term | Combined Terms

Figure 4.1: Level-lines of the two cost function terms involving p. Left to right: the regularization term, the coupling term, an overlay of both these terms together.

Hence, the main part of the proposed algorithm is to iteratively update v, u, and p respectively. Also, according to the optimality conditions, the Lagrange multipliers μ and μ_2 should be updated by taking

$$\begin{aligned} \mu^k &= \mu^{k-1} + r\left(v^k - u^k\right), \\ \mu_2^k &= \mu_2^{k-1} + r_2\left(p^k - \nabla u^k\right). \end{aligned} \tag{4.14}$$

An algorithmic description is summarized as Algorithm 4.1.

Algorithm 4.1 Fast TV regularization of matrix-valued data

1: **for** $k = 1, 2, \ldots,$ until convergence **do**

2: Update $u^k(x)$, according to Equations (4.11).

3: Update $p^k(x)$, according to Equations (4.13).

4: Update $v^k(x)$, by projection onto the matrix group,

- For $SO(n)$ matrices, according to Equation (4.19).
- For $SE(n)$ matrices, according to Equation (4.28).
- For $SPD(n)$ matrices, according to Equation (4.29).

5: Update $\mu^k(x), \mu_2^k(x)$, according to Equation (4.14).

6: **end for**

4.3.3 Split-Bregman method for matrix-valued regularization

An additional interpretation of augmented-Lagrangian total variation regularization has been suggested in [219], relating it to split-Bregman techniques, and specifically, the method suggested in [130]. In fact, for the set of unit vectors, $\mathcal{G} = S^m$, the projection operator for v would be the same as the one used in [130]. In split-Bregman iterations [91], in each iteration a *Bregman distance* [37] is minimized.

Looking at the function

$$E\left(p, u, v\right) = \int \|p\| + \|u - u_0\|^2 , \tag{4.15}$$

and vector sequence

$$d_{BREG}^k = \left(d_u^k, d_p^k, d_v^k\right) = -\left(\operatorname{div} \mu_2^k + \mu^k, \mu_2^k, -\mu^k\right), \tag{4.16}$$

we use the Bregman distance

$$\begin{aligned} &D^{d_{BREG}^k}\left((u, q, v), \left(u^k, p^k, v^k\right)\right) = \\ &\int \|p\| + \|u - u_0\|^2 - \|p^k\| - \left\|u^k - u_0\right\|^2 - \left\langle d_{BREG}^k, (u, p, v)\right\rangle . \end{aligned} \tag{4.17}$$

Re-examining our inner update step in algorithm 4.1, we can reformulate our algorithm (between updates of the Lagrange multipliers) in the form of a split-Bregman iterations

$$\begin{aligned}
&\left(u^{k+1}, p^{k+1}, v^{k+1}\right) = \\
&\underset{u,p,v}{\operatorname{argmin}} \begin{array}{l} D^{\left(d_u^k, d_p^k, d_v^k\right)}\left((u,p,v), \left(u^k, p^k, v^k\right)\right) \\ +\frac{r}{2}\int \|p - \nabla u\|^2 + \frac{r_2}{2}\int \|u - v\|^2 \end{array} = \\
&\underset{u,p,v}{\operatorname{argmin}} \int \begin{array}{l} \|p\| + \|u - u_0\|^2 + \left\langle \operatorname{div} \mu_2^k + \mu^k, u \right\rangle \\ + \left\langle \mu^k, p \right\rangle + \left\langle -\mu_2^k, v \right\rangle + \frac{r_2}{2}\|p - \nabla u\|^2 + \frac{r}{2}\|u - v\|^2 \end{array} = \\
&\underset{u,p,v}{\operatorname{argmin}} \int \begin{array}{l} \|p\| + \|u - u_0\|^2 + \left\langle \mu_2^k, p - \nabla u \right\rangle \\ + \left\langle \mu^k, u - v \right\rangle + \frac{r_2}{2}\|p - \nabla u\|^2 + \frac{r}{2}\|u - v\|^2 \end{array}.
\end{aligned} \tag{4.18}$$

We note that we take into account the nature of v while minimizing the Bregman distance only. In this sense, and in the choice of vector d_{BREG}^k the algorithm differs from the split-Bregman method. This is not surprising as our domain is not necessarily convex, and the convergence properties of the split-Bregman iteration cannot automatically hold for nonconvex domains. Partial convergence proofs for the algorithm can be obtained in the context of the augmented-Lagrangian formulation, with some small modification, as mentioned in the following section for nonconvex matrix manifolds.

4.3.4 Regularization of maps onto $SO(n)$

In the case of $\mathcal{G} = SO(n)$, Although the embedding of $SO(n)$ in Euclidean space is not a convex set, the projection onto the matrix manifold is easily achieved by means of the singular value decomposition [86]. Let $\mathbf{USV}^T = \left(\frac{\mu}{r} + u^k\right)$ be the SVD decomposition of $\frac{\mu}{r} + u^k$, we update v by

$$\begin{aligned}
v^{k+1} &= \underset{SO(n)}{\operatorname{Proj}}\left(\frac{\mu}{r} + u^k\right) = \mathbf{U}(x)\mathbf{V}^T(x), \\
\mathbf{USV}^T &= \left(\frac{\mu}{r} + u^k\right).
\end{aligned} \tag{4.19}$$

Other possibilities include using the Euler-Rodrigues formula, quaternions, or the polar decomposition [131]. We note that the nonconvex domain $SO(n)$ makes the analysis of global convergence for this scheme quite elaborate.

Convergence proof for the Inner Iterations

Despite the non-convex domain, and non-continuous projection operator, the inner iterations (update of u, v, p) of a modified variant of the algorithm, in the case of $\mathcal{G} = SO(n)$ (and $\mathcal{G} = SE(n)$) can be easily made convergent in a weak sense using the method inspired by the work of Attouch et al. [16], as described in our technical report [187]. In this limited setup, we hold the Lagrange multipliers μ, μ_2 fixed, and update p, u, v, minimizing them as described above. Adapting our notation to that of Attouch et al. [16], we rewrite

$$\begin{aligned} f(u,p) &= \|p\| + \mu_2^T (p - \nabla u) + \frac{r_2}{2}\|p - \nabla u\|^2, \\ g(v) &= i_g(v), \\ Q(u,p,v) &= \mu^T (u - v) + \frac{r}{2}\|u - v\|^2, \\ L(u,v,p) &= f(u,p) + Q(u,p,v) + g(v), \end{aligned} \tag{4.20}$$

where Q, f, g would fill similar roles as in [16], and $i_g(v)$ is the indicator function for the group g. We note that care needs to be taken when adapting the proofs from [16], as in our case the minimization step of v is confined to a subset of the Euclidean space which is not an affine subspace, and hence many of the proofs utilizing the subgradient w.r.t v need to be significantly revised. We can still, however, prove convergence of the residuals towards zero in the following manner. We modify our update steps to be

$$u^k = \underset{u}{\operatorname{argmin}}\, L(u, p^{k-1}, v^{k-1}) + \frac{1}{2\theta}\|u - u^{k-1}\|^2, \tag{4.21}$$

$$v^k = \underset{v}{\operatorname{argmin}}\, L(u^k, p^{k-1}, v) + \frac{1}{2\theta}\|v - v^{k-1}\|^2, \tag{4.22}$$

$$p^k = \underset{p}{\operatorname{argmin}}\, L(u^k, p, v^k), \tag{4.23}$$

where θ is a (finite and positive) constant coefficient. We begin by noting according to Equation 4.23 that

$$L\left(u^k, p^k, v^k\right) \leq L\left(u^k, p^{k-1}, v^k\right). \tag{4.24}$$

Using Equation 4.22 we get

$$L\left(u^k, p^{k-1}, v^k\right) + \frac{1}{2\theta}\|v^k - v^{k-1}\|^2 \leq L\left(u^k, p^{k-1}, v^{k-1}\right).$$

Finally, using Equation 4.21 we get

$$L\left(u^k, p^{k-1}, v^{k-1}\right) + \frac{1}{2\theta}\|u^k - u^{k-1}\|^2 \leq L\left(u^{k-1}, p^{k-1}, v^{k-1}\right),$$

resulting in the inequality

$$L\left(u^k, p^k, v^k\right) + \frac{1}{2\theta}\|u^k - u^{k-1}\|^2 + \frac{1}{2\theta}\|v^k - v^{k-1}\|^2 \leq L\left(u^{k-1}, p^{k-1}, v^{k-1}\right),$$

or alternatively

$$\|u^k - u^{k-1}\|^2 + \|v^k - v^{k-1}\|^2 \leq 2\theta\left(L\left(u^{k-1}, p^{k-1}, v^{k-1}\right) - L\left(u^k, p^k, v^k\right)\right). \quad (4.25)$$

Since the sequence of $L\left(u^k, p^k, v^k\right)$ is non-increasing, and is bounded from below (because of the quadratic structure of L with respect to u, v, p) for a given set of multipliers, we know that

$$2\theta\left(L\left(u^{k-1}, p^{k-1}, v^{k-1}\right) - L\left(u^k, p^k, v^k\right)\right) \to 0, \quad (4.26)$$

and therefore

$$\|u^k - u^{k-1}\|^2 + \|v^k - v^{k-1}\|^2 \to 0. \quad (4.27)$$

Thus, we can show that the residual decreases towards 0, and give some assurance as to the convergent behavior of the algorithm. A complete convergence analysis is not straightforward, as mentioned in [130]. Empirical results seem to demonstrate strong convergence properties in a variety of applications and scenarios, for a wide variety of θ values. Convergence plots for a range of θ values is shown in Figure 4.2. As can be seen, there is a slight advantage in terms of convergence speed for high θ values and weak coupling, but in general, the method works well for a large variety of θ values.

Moreover, the case where only partial updates of u, v, p are performed is more elaborate and its analysis is left as future work. We note that for the case of total-variation regularization, properties of the split-Bregman iterations with partial accuracy have been analyzed by Yin and Osher [254].

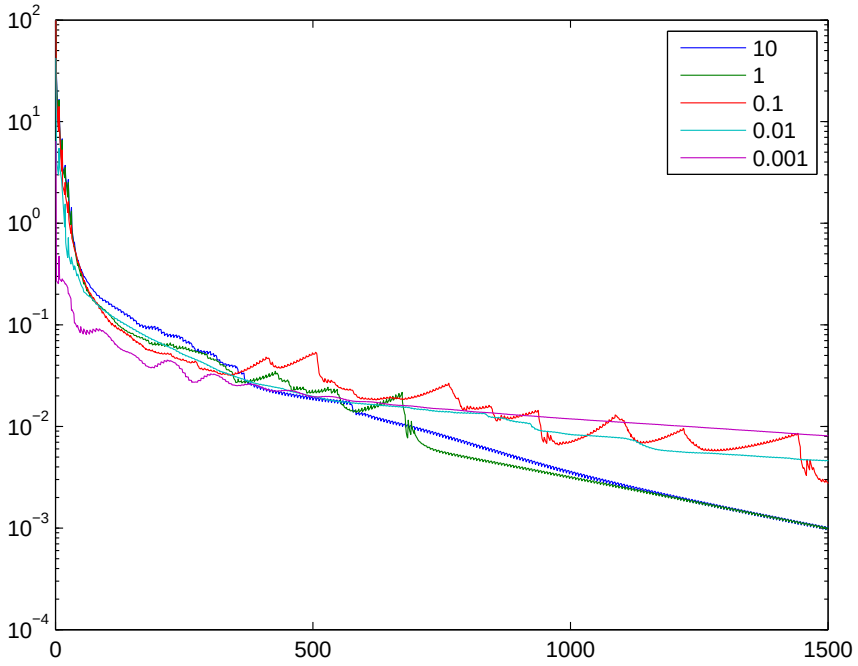


Figure 4.2: Residual plots for a variety of θ values, for the optimization problem given in Figure 4.3

4.3.5 Regularization of maps onto $SE(n)$

In order to regularize images with values in $SE(n)$, we use the embedding of $SE(n)$ into $\mathbb{R}^{n(n+1)}$ as our main optimization variable, u, per pixel.

The projection step w.r.t. v applies only for the n^2 elements of v describing the rotation matrix, leaving the translation component of $SE(n)$ unconstrained.

Specifically, let $v = (v_R, v_t), v_R \in \mathbb{R}^{n^2}, v_t \in \mathbb{R}^n$ denotes the rotation and translation parts of the current solution, with a similar partition for the Lagrange multipliers $\mu = (\mu_R, \mu_t)$. Updating v in line 4 of Algorithm 4.1 assumes the form

$$v_R^{k+1} = \underset{SO(n)}{\text{Proj}} \left(\frac{\mu_R}{r} + u_R^k \right), \; v_t^{k+1} = \left(\frac{\mu_t}{r} + u_t^k \right) \tag{4.28}$$

$$v^{k+1} = \underset{SE(n)}{\text{Proj}}(v^k) = (v_R^{k+1}, v_t^{k+1}).$$

4.3.6 Regularization of maps onto $SPD(n)$

The technique described above can be used also for regularizing symmetric positive-definite matrices. Here, the intuitive choice of projecting the eigenvalues of the matrices onto the positive half-space is shown to be optimal [105]. Many papers dealing with the analysis of DT-MRI rely on the eigenvalue decomposition of the tensor as well, i.e. for tractography, anisotropy measurements, and so forth.

For $\mathcal{G} = SPD(n)$, the minimization problem w.r.t. v in step 3 of Algorithm 4.1 can be solved by projection of eigenvalues. Let $\mathbf{U}\,\mathrm{diag}\,(\boldsymbol{\lambda})\,\mathbf{U}^T$ be the eigenvalue decomposition of the matrix $\frac{\mu}{r} + u^k$. v is updated according to

$$v^{k+1} = \underset{SPD(n)}{\mathrm{Proj}}\,(v^k) = \mathbf{U}(x)\,\mathrm{diag}\left(\hat{\boldsymbol{\lambda}}\right)\mathbf{U}^T(x), \tag{4.29}$$

$$\mathbf{U}\,\mathrm{diag}\,(\boldsymbol{\lambda})\,\mathbf{U}^T = \left(\frac{\mu}{r} + u^k\right), \left(\hat{\boldsymbol{\lambda}}\right)_i = \max\left((\boldsymbol{\lambda})_i, 0\right),$$

where the matrix U is a unitary one, representing the eigenvectors of the matrix, and the eigenvalues $\left(\hat{\boldsymbol{\lambda}}\right)_i$ are the positive projection of the eigenvalues $(\boldsymbol{\lambda})_i$. Optimization w.r.t. u is done as in the previous cases, as described in Algorithm 4.1.

Furthermore, the optimization w.r.t. u, v is now over the domain $\mathbb{R}^m \times SPD(n)$, and the cost function is convex, resulting in a convex optimization problem. The convex domain of optimization allows us to formulate a convergence proof for the algorithm similar to the proof by Tseng [230]. We refer the interested reader to our technical report [187]. An example of using the proposed method for DT-MRI denoising is shown in Section 4.4.

Global Convergence for $SPD(n)$ Regularization

For $SPD(n)$ regularization we basically do a coordinate descent on a convex domain [230] and therefore can show global convergence of our method. At each step of the inner iteration, we do a full minimization with respect to the selected variables block u,

v and p. Using the notation provided by [230], we can rewrite our functional as

$$\mathcal{F}_{\mu,\mu_2}(u,v,p) = f_0(u,v,p) + f_1(u) + f_2(v) + f_3(p), \tag{4.30}$$

where

1. f_0 is a convex, smooth, function.

$$f_0(u,v,p) = \frac{r}{2}\|v-u\|^2 + \langle \mu, v-u\rangle + \frac{r_2}{2}\|p-\nabla u\|^2 + \langle \mu_2, p-\nabla u\rangle$$

2. f_1, f_2 and f_3 are convex, lower-semiconinuous, continuous in their effective domain,

$$f_1(u) = \|u-u_0\|^2 \tag{4.31}$$

$$f_2(v) = 0 \tag{4.32}$$

$$f_3(p) = \|p\|. \tag{4.33}$$

By [230, Proposition 1], it can be shown that the alternating minimization will converge to a minimizer of $\mathcal{F}_{\mu,\mu_2}(u,v,p)$. Along the same proof in [246], it can be proved the whole algorithm converges. For completeness we repeat the proof here. The following characterization for the minimizers of functional $\mathcal{F}(u,v,p;\mu,\mu_2)$ will be used. Assume that (u^*,v^*,p^*) is one of the minimizers, and for arbitrary (u',v',p') we have,

$$\begin{aligned}
&\lambda\|u^*-u_0\|^2 - \lambda\|u'-u_0\|^2 + r_2(p^*-\nabla u^*, -(\nabla u^*-\nabla u')) \\
&+r(u^*-v^*, u^*-u') + (\mu^*, u^*-u') + (\mu_2^*, -(\nabla u^*-\nabla u')) \le 0 \qquad (4.34)\\
&-r(u^*-v^*, v^*-v') - (\mu^*, v^*-v') \le 0 \qquad (4.35)\\
&\|p^*\| - \|p'\| + r_2(p^*-\nabla u^*, p^*-p') + (\mu_2^*, p^*-p') \le 0 \qquad (4.36)
\end{aligned}$$

(see [77],p.38 Proposition 2.2)

Theorem 4.3.1. *The sequence $(u^k,v^k,p^k;\mu^k,\mu_2^k)$ generated by Algorithm 4.1 converges to the saddle-point $(u^*,v^*,p^*;\mu^*,\mu_2^*)$ of the functional $\mathcal{F}(u,v,p;\mu,\mu_2)$*

Proof. Let $\bar{u}^k = u^* - u^k$,$\bar{v}^k = v^* - v^k$,$\bar{p}^k = p^* - p^k$, $\bar{\mu}^k = \mu^* - \mu^k$, and $\bar{\mu}_2^k = \mu_2^* - \mu_2^k$ Since $(u^*, v^*, p^*; \mu^*, \mu_2^*)$ is the saddle point of $\mathcal{F}(u, v, p; \mu, \mu_2)$, we have

$$\mathcal{F}(u^*, v^*, p^*; \mu^*, \mu_2^*) \leq \mathcal{F}(u', v', p'; \mu^*, \mu_2^*), \forall u, v, p \tag{4.37}$$

In particular when $u' = u^k$ (4.34) still holds

$$\begin{aligned} &\lambda\|u^* - u_0\|^2 - \lambda\|u^k - u_0\|^2 + r_2(p^* - \nabla u^*, -\nabla(u^* - u^k)) \\ &+r(u^* - v^*, u^* - u^k) + (\mu^*, u^* - u^k) + (\mu_2^*, -\nabla(u^* - u^k)) \leq 0 \end{aligned} \tag{4.38}$$

On the other hand, since the $(u^k, v^k, p^k; \mu^k, \mu_2^k)$ is the minimizer of $\mathcal{F}(u, v, p; \mu^k, \mu_2^k)$, u^k will also satisfy (4.34) and after substituting $u' = u^*$ we obtain

$$\begin{aligned} &\lambda\|u^k - u_0\|^2 - \lambda\|u^* - u_0\|^2 + r_2(p^k - \nabla u^k, -\nabla(u^k - u^*)) \\ &+r(u^k - v^k, u^k - u^*) + (\mu^k, u^k - u^*) + (\mu_2^k, -\nabla(u^k - u^*)) \leq 0. \end{aligned} \tag{4.39}$$

Adding the two inequalities yields

$$r_2(\bar{p}^k - \nabla\bar{u}^k, -\nabla\bar{u}^k) + r(\bar{u}^k - \bar{v}^k, \bar{u}^k) + (\bar{\mu}^k, \bar{u}^k) + (\bar{\mu_2}^k, -\nabla\bar{u}^k) \leq 0 \tag{4.40}$$

Similarly, w.r.t v^*, v^k using the same argument to (4.35) we have

$$-r(u^* - v^*, v^* - v^k) - (\mu^*, v^* - v^k) \leq 0 \tag{4.41}$$

$$-r(u^k - v^k, v^k - v^*) - (\mu^k, v^k - v^*) \leq 0 \tag{4.42}$$

adding two inequalities yields

$$-r(\bar{u}^k - \bar{v}^k, \bar{v}^k) - (\bar{\mu}^k, \bar{v}^k) \leq 0 \tag{4.43}$$

w.r.t p^*, p^k, the same argument is applied to (4.36)

$$\|p^*\| - \|p^k\| + r_2(p^* - \nabla u^*, p^* - p^k) + (\mu_2^*, p^* - p^k) \leq 0 \tag{4.44}$$

$$\|p^k\| - \|p^*\| + r_2(p^k - \nabla u^k, p^k - p^*) + (\mu_2^k, p^k - p^*) \leq 0 \tag{4.45}$$

thus

$$r_2(\bar{p}^k - \nabla\bar{u}^k, \bar{p}^k) + (\bar{\mu_2}^k, \bar{p}^k) \leq 0 \tag{4.46}$$

Adding (4.40), (4.43) and (4.46) we have

$$r_2\|\bar{p}^k - \nabla\bar{u}^k\|^2 + r\|\bar{u}^k - \bar{v}^k\|^2 + (\bar{\mu_2}^k, \bar{p}^k - \nabla\bar{u}^k) + (\bar{\mu}^k, \bar{u}^k - \bar{v}^k) \leq 0 \tag{4.47}$$

By the way of updating multipliers, also note that $u^* = v^*$ and $p^* = \nabla u^*$ we obtain

$$\bar{\mu}^{k+1} = \bar{\mu}^k + r(\bar{u}^k - \bar{v}^k) \tag{4.48}$$

$$\bar{\mu}_2^{k+1} = \bar{\mu}_2^k + r_2(\bar{p}^k - \nabla\bar{u}^k) \tag{4.49}$$

therefore by (4.47) we have

$$\begin{aligned}
&r_2\|\bar{\mu}^{k+1}\|^2 + r\|\bar{\mu}_2^{k+1}\|^2 - r_2\|\bar{\mu}^k\|^2 - r\|\bar{\mu}_2^k\|^2 \\
&= 2rr_2(\bar{\mu}^k, \bar{u}^k - \bar{v}^k) + 2rr_2(\bar{\mu}_2^k, \bar{p}^k - \nabla\bar{u}^k) + r^2r_2\|\bar{u}^k - \bar{v}^k\|^2 + rr_2^2\|\bar{p}^k - \nabla\bar{u}^k\| \\
&\leq -r^2r_2\|\bar{u}^k - \bar{v}^k\|^2 - rr_2^2\|\bar{p}^k - \nabla\bar{u}^k\| \leq 0
\end{aligned} \tag{4.50}$$

This actually implies μ^k and μ_2^k are bounded, and

$$\lim_{k\to\infty} \|p^k - \nabla u^k\| = 0 \tag{4.51}$$

$$\lim_{k\to\infty} \|u^k - v^k\| = 0 \tag{4.52}$$

With this in mind, it is not hard to show that $(u^k, v^k, p^k; \mu^*, \mu_2^*)$ converge to the saddle-point of the functional □

4.3.7 A Higher-Order Prior for Group-Valued Images

We note that the scheme we describe is susceptible to the staircasing effect, since it minimizes a total variation regularization of the map u. Several higher-order priors can be incorporated into our scheme, that avoid this effect. One such possible higher-order term generalizes the scheme presented by Wu and Tai [245], by replacing the per-element gradient operator with a Hessian operator. The resulting saddle-point problem

becomes

$$\begin{aligned} &\min_{u \in \mathbb{R}^m} \max_{\mu_2} \int \begin{bmatrix} \|p\| + \tilde{\lambda} \|u - \tilde{u}(u_0, v, \mu, r)\|^2 \\ +\mu_2^T (p - Hu) + \frac{r_2}{2} \|p - Hu\|^2 \end{bmatrix} dx, \\ &p \in \mathbb{R}^{4m}, \\ &v \in \mathcal{G} \end{aligned} \tag{4.53}$$

where H denotes the per-element Hessian operator,

$$(H(u))_{i,\cdot} = \left(D_{xx}^{-+} u_i, D_{xy}^{++} u_i, D_{yx}^{++} u_i, D_{yy}^{-+} u_i \right), \tag{4.54}$$

where we use, for example D_{xx}^{-+} to describe the second order derivative obtained by first applying the forward and then the backward first order derivative. Minimizing this functional with respect to p is done by shrinkage, as described in [245]. Solving with respect to u given p is done by solving the resulting optimality system of equations [245],

$$2\tilde{\lambda}(u) + r_2 H^* (H(u)) = H^*(\mu_2) + r_2 H^* (p) + 2\tilde{\lambda}\tilde{u}, \tag{4.55}$$

where $H^*(\cdot)$ denotes the adjoint operator for the operator $H(\cdot)$,

$$H^*(p) = D_{xx}^{+-} p^1 + D_{xy}^{--} p^2 + D_{yx}^{--} p^3 + D_{yy}^{+-} p^4, \tag{4.56}$$

where p^i denotes the ith element of the per-pixel vector p for each element in u, using scalar notations in order to avoid further complicating the notation. We refer the reader to [245] for the complete discussion and definition of these operators, and remark that in our case, Gauss-Seidel iteration were used instead of a Fourier-domain solver.

We show an example using the appropriately modified scheme, for the case of $\mathcal{G} = SO(2)$ in Figures 4.3,4.4

4.4 Numerical Results

As discussed above, the proposed algorithmic framework is considerably general and suitable for various applications. We show several examples from different application domains to demonstrate our algorithm

4.4.1 Directions Regularization

Analysis of principal directions in an image or video is an important aspect of modern computer vision, in fields such as video surveillance [162, and references therein], vehicle control [73], crowd behaviour analysis [153], and other applications[172].

The input in this problem is a set of normalized / unnormalized direction vectors located throughout the image domain, either in a dense or sparse set of locations. The goal is to obtained a smoothed version of the underlying direction field. Since $SO(2)$ is isomorphic to S^1, the suggested regularization scheme can be used for regularizing directions, such as *principal motion directions* in a video sequence. A reasonable choice for a data term that does not depend on the vector lengths would try to align the rotated first coordinate axis with the motion directions in the neighborhood,

$$E_{PMD}(U) = \sum_{(x_j,y_j)\in\mathcal{N}(i)} \left(U_{1,1} \left(v_j\right)_x + U_{1,2} \left(v_j\right)_y \right),$$

where $\left(x_j, y_j, \left(v_j\right)_x, \left(v_j\right)_y\right)$ represent a sampled motion particle [153] in the video sequence (location and velocity), and $U_{i,j}$ represent elements of the solution u at each point.

In Figure 4.3 we demonstrate two sparsely sampled, noisy, motion fields, and a dense reconstruction of the main direction of motion at each point. The data for the direction estimation was corrupted by adding component-wise Gaussian noise. In the first image, the motion field is comprised of 4 regions with a different motion direction at each region. The second image contains a sparse sampling of an expansion motion field of the form $\vec{v}(x,y) = \frac{(x,y)^T - \mathbf{c}}{\|(x,y)^T - \mathbf{c}\|}$, where $\mathbf{c}$ denotes the center of the image. Such an expansion field is often observed by forward-moving vehicles. Note that despite the fact that a vanishing point of the flow is clearly not smooth in terms of the motion directions, the estimation of the motion field remains reasonable, due to the robust nature of total-variation regularization.

Another classical example of direction diffusion is in denoising of directions in fingerprint images. An example for direction diffusion on a fingerprint image taken from

the Fingerprint Verification Competition datasets [1] can be seen in Figure 4.4. Adding a noise of $\sigma = 0.05$ to the image and estimating directions based on the structure tensor, we smoothed the direction field and compared it to the field obtained from the original image. We used our method with $\lambda = 3$, and the modified method based on Equation 4.53 with $\tilde{\lambda} = 10$, as well as the method suggested by Sagiv et al. [195] with $\beta = 100, T = 425$. The resulting MSE values of the tensor field are 0.0317, 0.0270 and 0.0324, respectively, compared to an initial noisy field with and MSE of 0.0449. The results demonstrate the effectiveness of our method for direction diffusion, even in cases where the staircasing effect may cause unwanted artifacts.

4.4.2 $SE(n)$ Regularization

An example of $SE(3)$ valued images can be obtained by doing local matches between two range scans obtained from a Kinect device. For each small surface patch from the depth image we use an iterative closest point algorithm [31, 57] to match the surface from the previous frame. This obtains for us a field over $SE(3)$ over the image plane for every time frame. The method is described as Algorithm 4.2

Algorithm 4.2 Regularized 3D rigid motion estimation

1: **for** each frame t, and the scanned 3D surface at time t, S_t **do**
2: **for** each pixel $\mathbf{x}_i \in S_t$ **do**
3: Crop a small patch around $\mathbf{x}_i$, $P_i = S_t \cap B_r(\mathbf{x}_i)$
4: Estimate the rigid motion R_i, t_i that minimizes the L_2 error between P_i and S_{t-1}, by running the iterative closest point algorithm.
5: Set $u_0(\mathbf{x}_i) = (\mathbf{R}_i, \mathbf{t}_i)$
6: **end for**
7: Perform regularization of u_0 using Algorithm 4.1.
8: **end for**

We note that this measurement process is highly contaminated by non-Gaussian noise. Despite this high level of noise we can use our algorithm to smooth this $SE(3)$

image, obtaining a scale-space of $SE(3)$ images, as shown in Figure 4.5. It can be seen that for a careful choice of the regularization parameter, total variation in the group elements is seen to significantly reduce rigid motion estimation errors. Visualization is accomplished by projecting the embedded matrix onto 3 different representative vectors in $\mathbb{R}^{12}$.

In order to demonstrate the efficiency of this method and its parallelizable nature, we implement it using the CUDA framework, and measure the computational time required in order to obtain practical levels of convergence, at least 3 orders of magnitude. The computation times are shown in Table 4.1, for various image sizes and iterations. Most of the examples shown are with only 1 inner iteration since this has given us the fastest convergence, without artifacts in the final result. In the GPU implementation the polar decomposition was chosen for its simplicity and efficiency. In practice, one Gauss-Seidel iteration sufficed to update u. Using 15 outer iterations, practical convergence is achieved in 49 milliseconds on an NVIDIA GTX-580 card for QVGA-sized images, demonstrating the efficiency of our algorithm and its potential for real-time applications. This is especially important for applications such as gesture recognition where fast computation is crucial. We note that we do not use here the information between more than two frames, as is often done in optical flow. Furthermore, using nonrigid deformation rather than ICP in order to compute u_0 leads to significantly better results using the same regularization scheme, as we demonstrated [181], including segmentation of articulated motion. Such discussion of the initialization method is beyond the scope of this work.

We now demonstrate the results of our algorithm using a more accurate initial motion estimation technique.

In order to estimate the non-rigid motion occuring between two subsequent time-frames of a depth video, we first apply a simple non-rigid registration process, similar to the approach suggested by Li et al. [136], followed by the estimation of a locally-rigid motion model, as described in the supplementary material. In general, any motion

Outer iterations	15	15	25	50	100
GS iterations	1	3	1	1	1
320×240	49	63	81	160	321
640×480	196	250	319	648	1295
1920×1080	1745	2100	2960	5732	11560

Table 4.1: Processing times (ms) for various sizes of images, with various iteration counts.

estimation method can be used.

Since the overall motion field can involve both piecewise rigid and non-rigid motion components, and because of the noisy scan results often obtained from commodity depth scanners, the estimated instantaneous motion is quite noisy, as can be seen in Figure 4.7. The motion field should be post-processed so as to obtain locally-rigid interpretation.

In Figure 4.6, we demonstrate results based on a synthetic hand model undergoing motion. We used the non-rigid registration model to track the surface over several frames so as to obtain a sufficiently large motion. While the detected motion is not completely piecewise-rigid due to skinning artifacts, occlusions, etc, the fingers are detected quite well. Using a standard mean-shift algorithm on the log-coordinates of the rotation matrices, we obtain segmentation of the fingers and the phalanges that undergo motion.

In Figure 4.7, we demonstrate TV regularization of $SE(3)$ for several frame pairs in a depth sequence by a Kinect sensor. Visualization is done using log-coordinates of the rotation matrix. The resulting estimated rigid motion allows segmentation of body parts, or finger phalanges in the (synthetically rendered) hand. In Figure 4.8 we use the mean-shift clustering algorithm [62] on the $SE(3)$ images' projection onto the small rotations standard linearization basis in order to segment the main moving parts. Despite the simple choice of the segmentation algorithm, that does not take into account the geometry of the surface and linearizes the Lie-group in the simplest possible man-

ner, the segmentation of the moving parts is clear. It is expected that utilizing geometric prior on the regions size will prevent artifacts such as oversegmentation. In the examples shown here, QVGA resolution (320×240) was used. The estimation of motion coefficients takes in Matlab about 5 seconds on an Intel i3 CPU. The regularization is on the order of a tenth of a second to compute on GPU. Preliminary results support this efficiency claim.

4.4.3 DT-MRI Regularization

In Figure 4.10 we demonstrate a smoothing of DT-MRI data from [146], based on the scheme suggested in Section 4.3.6. We show an axial view of the brain, glyph-based visualization using Slicer3D [2], with anisotropy-based color coding. In this visualization, the color of the glyphs marks isotropic (red) to anisotropic (blue/violet) tensors. The relative size of the ellipsoids denotes the amount of diffusion (trace of the tensor), and the directions of the ellipsoids' principal axes align with the eigenvectors of the tensor at each point.

The noise added is an additive Gaussian noise in each of the tensor elements with $\sigma = 0.1$. Note that while different noise models are often assumed for diffusion-weighted images, at high noise levels the Gaussian model is a reasonable approximation. Regularization with $\lambda = 30$ is able to restore a significant amount of the white matter structure. At such levels of noise, the TV-regularized data bias towards isotropic tensors (known as the *swell effect* [68]) is less significant. The RMS of the tensor representation was 0.0406 in the corrupted image and 0.0248 in the regularized image. An additional application of our method is to perform regularized reconstruction of DT-MRI signals from diffusion-weighted images (DWI). This is done by replacing the quadratic fidelity term with a fitting term based on the Stejskal-Tanner equation [212]. Demonstrating this application is shown in our technical report [187], as discussion of such reconstruction terms is beyond the scope of this chapter.

4.5 Regularized DTI Reconstruction

There are several possibilities of using the proposed regularization scheme for DT-MRI reconstruction from diffusion-weighted measurements. Instead of adding a fidelity term as in Equation (4.5), we add a term for fitting the Stejskal-Tanner equations [212], based on a set of measurements describing the diffusion in specific directions, and reconstruct the full diffusion tensor at each voxel. The fitting term can be written as

$$\sum_i \left\| b_i \mathbf{g}_i^T u \mathbf{g}_i - \log\left(\frac{S_i}{S_0}\right) \right\|^2,$$

where b_i and $\mathbf{g}_i$ are the b-values and gradient vectors, u is the diffusion tensor reconstructed at each voxel, and $\frac{S_i}{S_0}$ define the relative signal ratio for each direction at each voxel. The complete minimization problem reads

$$\begin{array}{c} \operatorname{argmin} \\ v \in SPD(n) \\ \\ u \end{array} \int \sum_i \left\| b_i \mathbf{g}_i^T u \mathbf{g}_i - \log\left(\frac{S_i}{S_0}\right) \right\|^2 + \lambda \|\nabla u\| + \frac{r}{2}\|v-u\|^2 + \langle \mu, v-u \rangle dx. \tag{4.57}$$

While the memory requirements seem less favorable for fast optimization, looking closely at the quadratic penalty data term, we see it can be expressed by looking at a fitting term for the Stejskal-Tanner equations ,

$$\sum_i \left\| b_i \mathbf{g}_i^T u \mathbf{g}_i - \log\left(\frac{S_i}{S_0}\right) \right\|^2 = u^T \mathbf{A} u + \mathbf{b}^T u + c, \tag{4.58}$$

where $\mathbf{A}$ is a constant matrix over the whole volume,

$$\mathbf{A} = \sum_i b_i^2 \begin{pmatrix} g_1^4 & 2g_1^3g_2 & 2g_1^3g_3 & g_1^2g_2^2 & 2g_1^2g_2g_3 & g_1^2g_3^2 \\ 2g_1^3g_2 & 4g_1^2g_2^2 & 4g_1^2g_2g_3 & 2g_1g_2^3 & 4g_1g_2^2g_3 & 2g_1g_2g_3^2 \\ 2g_1^3g_3 & 4g_1^2g_2g_3 & 4g_1^2g_3^2 & 2g_1g_2^2g_3 & 4g_1g_2g_3^2 & 2g_1g_3^3 \\ g_1^2g_2^2 & 2g_1g_2^3 & 2g_1g_2^2g_3 & g_2^4 & 2g_2^3g_3 & g_2^2g_3^2 \\ 2g_1^2g_2g_3 & 4g_1g_2^2g_3 & 4g_1g_2g_3^2 & 2g_2^3g_3 & 4g_2^2g_3^2 & 2g_2g_3^3 \\ g_1^2g_3^2 & 2g_1g_2g_3^2 & 2g_1g_3^3 & g_2^2g_3^2 & 2g_2g_3^3 & g_3^4 \end{pmatrix} \tag{4.59}$$

and $\mathbf{b}$ is the vector

$$\mathbf{b} = \sum_i b_i \log\left(\frac{S_i}{S_0}\right) \begin{pmatrix} 2g_1^2 & 4g_1g_2 & 4g_1g_3 & 2g_2^2 & 4g_2g_3 & 2g_3^2 \end{pmatrix}^T, \tag{4.60}$$

and c is the scalar image

$$c = \sum_i \left(\log\left(\frac{S_i}{S_0}\right)\right)^2. \tag{4.61}$$

We note that, unlike the denoising case, in the reconstruction case it is the data term that couples together the elements of the tensor together. Care must be taken so as to handle this coupled data term.

Reconstruction with the new data term can be computed using several techniques.

- Freezing all elements of the tensor but one, we obtain from the Euler-Lagrange equations pertaining to Equation 4.57 an update rule for the image, to be computed in the Fourier domain, or via Gauss-Seidel iterations. While the coupling between the tensor elements (expressed via the non-diagonal matrix $\mathbf{A}$) prevents us from treating each tensor element separately, the optimization w.r.t. each of the elements converges quite rapidly.

- Another possibility is to take a block Gauss-Seidel approach, and optimize each tensor separately, going over all the voxels one-by-one.

- Yet another possibility is to further decouple the TV and data term, using separate variables and constraining them using an augmented Lagrangian approach.

Of the above techniques, we have tried the first one. The reconstruction obtained is the spatially-regularized version of the *linear-least-squares* (LLS) method. One can incorporate a weighted least-squares (WLS, [196]), or nonlinear-least-squares (NLS) [125] data term instead. Combining such data terms and exploring the interaction between the regularization and nonlinear terms is beyond the scope of this work.

In Figures 4.11,4.12 we demonstrate reconstruction of the DT-MRI tensors, again based data from Lundervold et al. [146], using a set of 30 directional measurements.

The measure ratios $\log\left(\frac{S_i}{S_0}\right)$ were added a Gaussian additive noise of standard deviation 100. The reconstructed image obtained by regularized reconstruction with $\lambda = 1\times10^{-3}$ had an MSE of 2.1×10^{-4}, compared to 8.9×10^{-3} without regularization.

4.6 Conclusions

We propose in this chapter a general framework for matrix-valued image regularization. Using the augmented Lagrangian technique, we separate the optimization problem into a TV-regularization step and a projection step, both of which can be solved in an easy-to-implement and parallel way. We also demonstrate how to reformulate the optimization in terms of split-Bregman iterations, and relate it to existing works on split-Bregman iterations. Furthermore, we show the efficiency and effectiveness of the resulting scheme through several examples whose data taken from $SO(2)$, $SE(3)$, and $SPD(3)$ respectively. Our algorithms allow real-time regularization for tasks in image analysis and computer vision.

In an extension work we intend to explore other applications for matrix-valued image regularization as well as generalize our method to other modalities and data of maps. Additional research paths incorporates higher level priors, replacing locally-acting priors with patch-based and global shape priors.

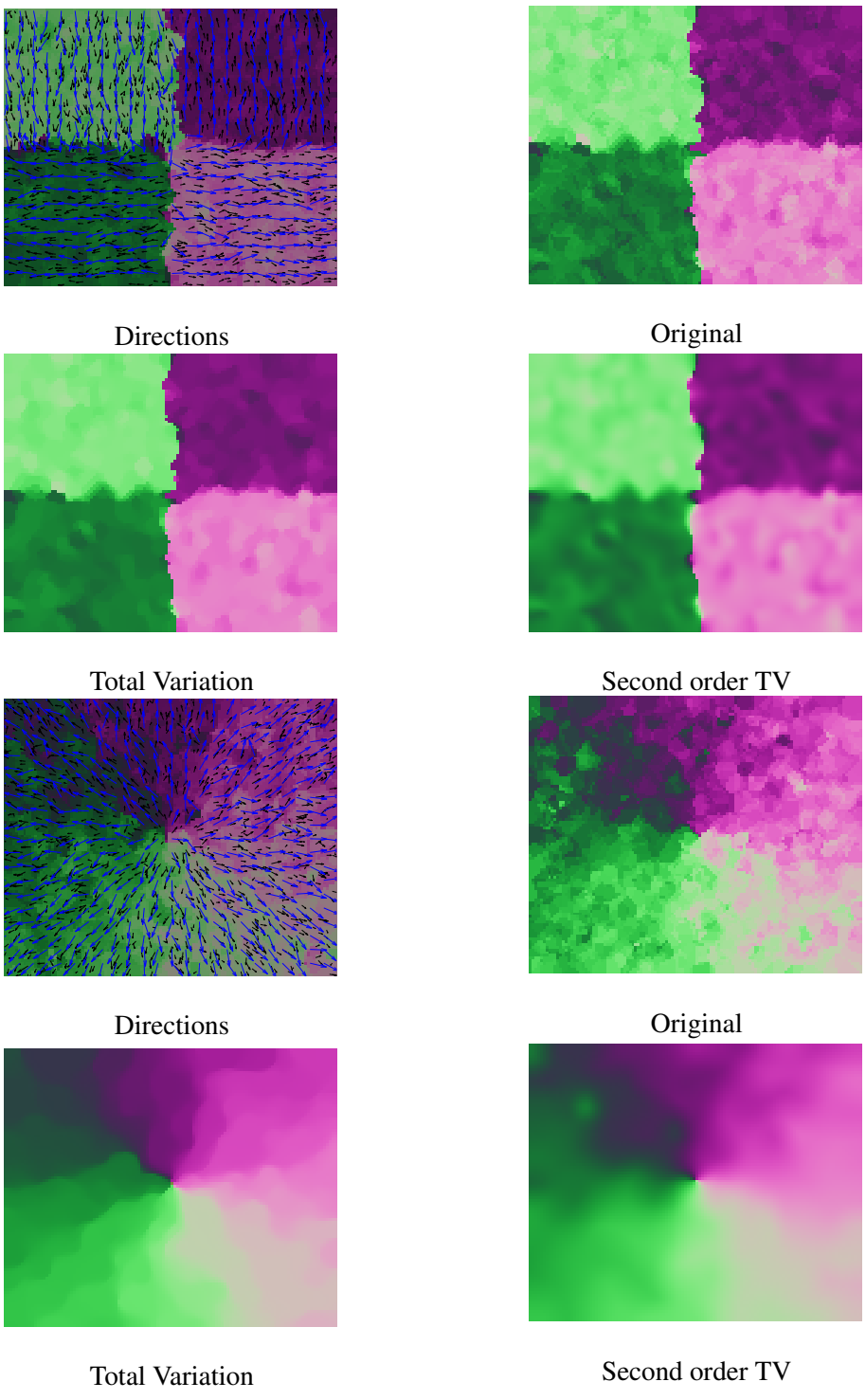


Figure 4.3: TV regularization of $SO(n)$ data. Left-to-right, top-to-bottom: a noisy, TV-denoised, and higher-order regularized (minimizing Equation 4.53) version of a piecewise constant $SO(2)$ image, followed by a expansion field direction image. Different colors mark different orientations of the initial/estimated dense field, black arrows signify the measured motion vectors, and blue arrows demonstrate the estimated field

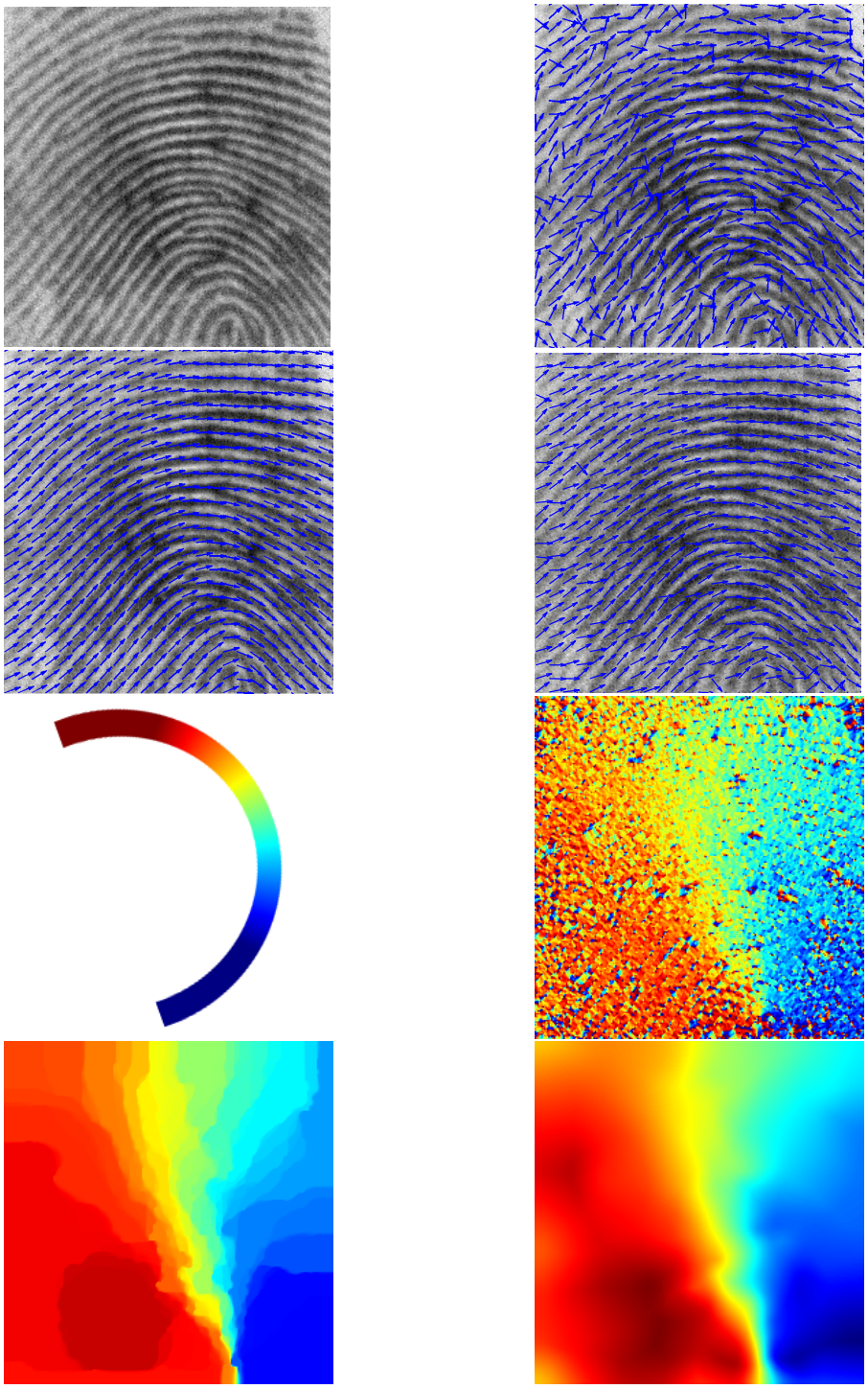


Figure 4.4: TV regularization of $SO(2)$ data based on fingerprint direction estimation. Top two rows,left-to-right: The fingerprint image with added Gaussian noise of $\sigma = 0.05$, the detected direction angles displayed as arrows, the detected directions after regularization with using a higher-order regularization term shown in Equation 4.53 with $\lambda = 6$, the regularization result by Sochen et al. [195]. Bottom two rows: color legend for the directionality images, initial estimated field, result of TV regularization with $\lambda = 3$, higher-order regularization with $\lambda = 6$.

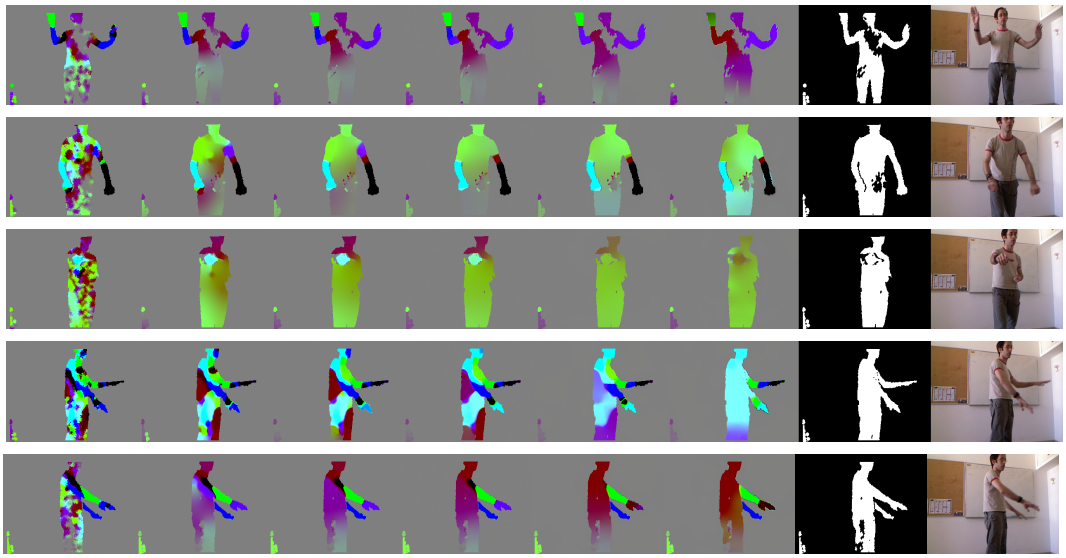


Figure 4.5: Regularization of $SE(3)$ images obtained from local ICP matching of the surface patch between consecutive Kinect depth frames. Left-to-right: diffusion scale-space obtained by different values of λ: $1.5, 1.2, 0.7, 0.2, 0.1, 0.05$, the foreground segmentation based on the depth, and an intensity image of the scene. Top-to-bottom: different frames from the depth motion sequence.

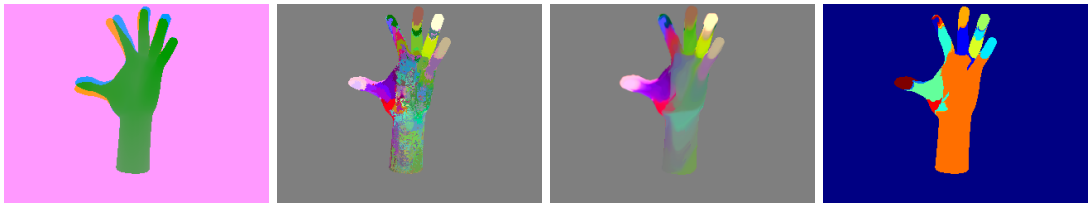


Figure 4.6: TV regularization based of an $SE3$-valued image placed on a rendered depth surface. Left-To-Right, Top-To-Bottom: An overlay of the two consecutive time-frames used to obtain motion estimation, the estimated and regularized $SE(3)$ images, and a resulting segmentation using mean-shift.

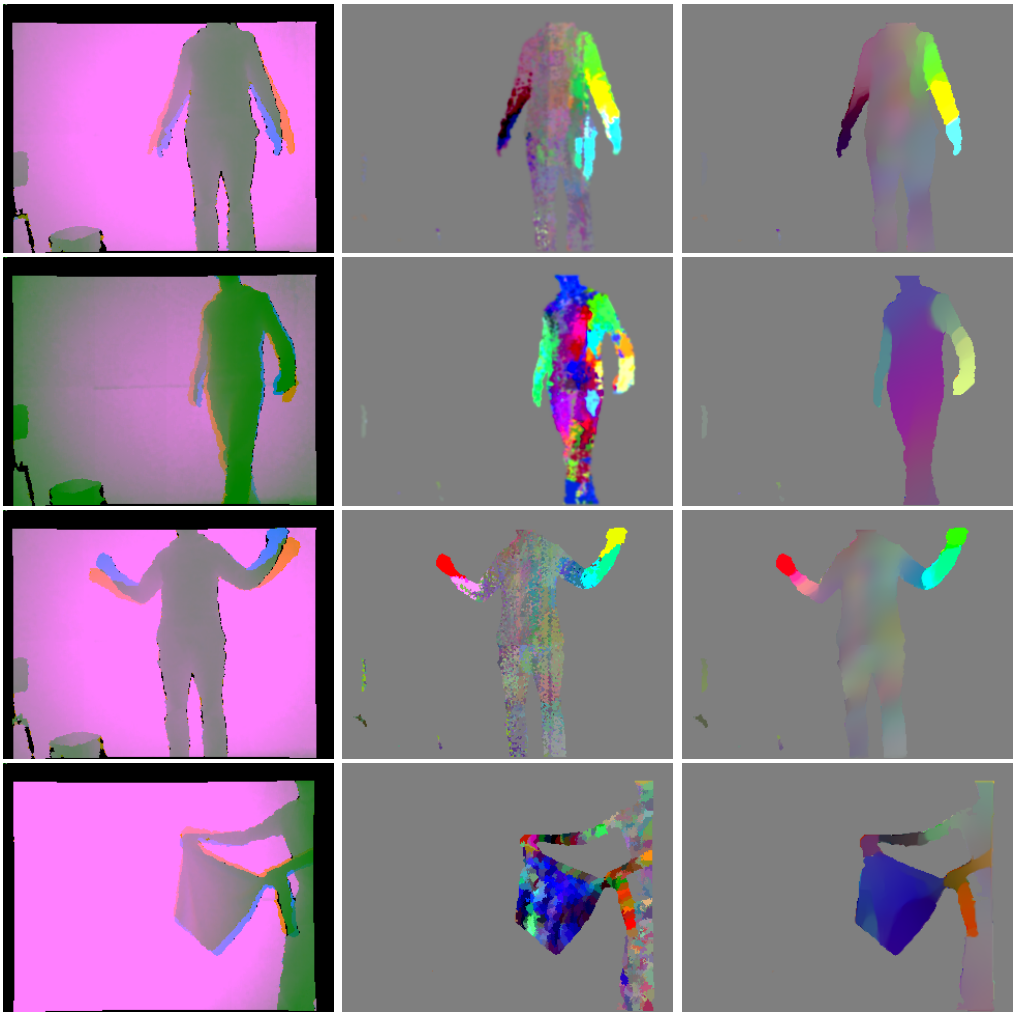


Figure 4.7: TV regularization of an $SE3$-valued image given on a scanned depth surface. . Each row represents results on two different frames from a depth sequence. Left-To-Right, for each frame: An overlay of the two consecutive time-frames used to obtain motion estimation, the estimated $SE(3)$ measurement, and regularized image. Raw depth data is used to estimate the motion. The regularized $SE(3)$ image hints at joint locations for parts that were moving at the time the depth frames were taken. Note in the last example, using a slightly stronger regularization, a nonrigid object (a shirt) is still separated clearly from the arms.

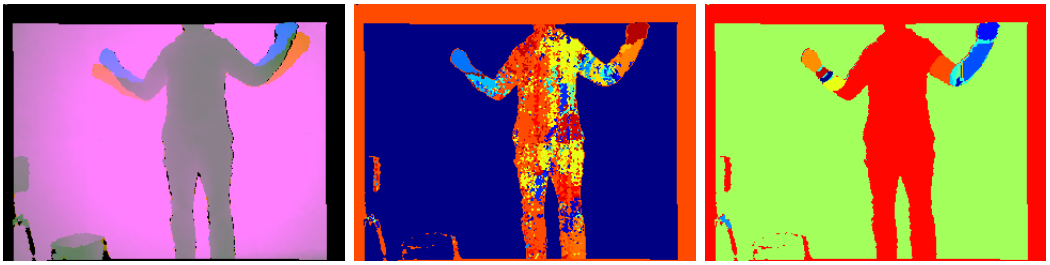


Figure 4.8: Segmentation based on mean-shift clustering of the $SE(3)$ image. Left-to-right: The motion between the two frames, the segmentation obtained using the raw estimated $SE(3)$, and the segmentation obtained using the regularized result, showing a segmentation of the moving limb parts.

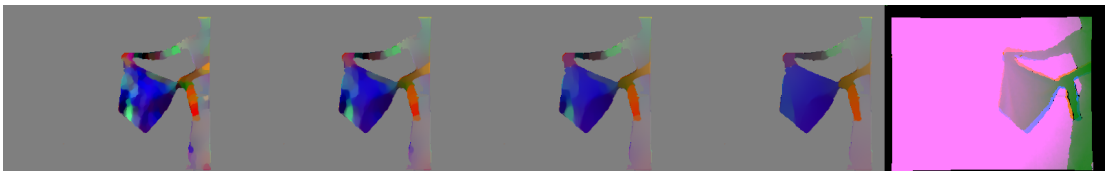


Figure 4.9: A scale-space obtained by changing the fidelity coefficient through the values $\lambda = 5, 2.5, 1.5, 0.8$.

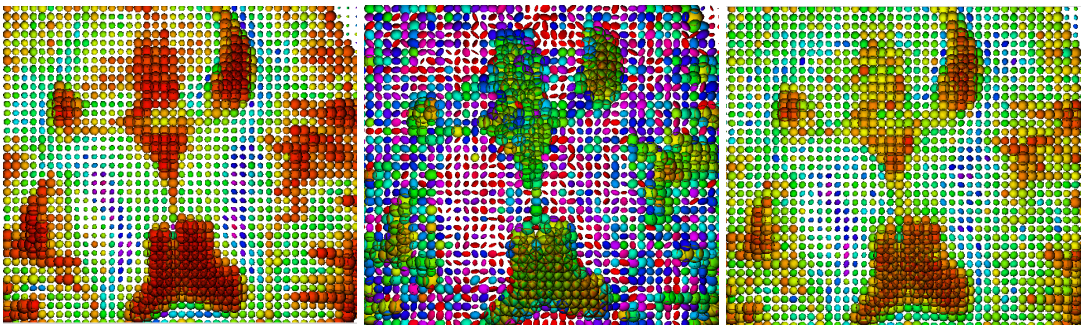


Figure 4.10: TV denoising of images with diffusion tensor data, visualized by 3D tensor ellipsoid glyphs colored by fractional anisotropy. Left-to-right: the original image, an image with added component-wise Gaussian noise of $\sigma = 0.1$, and the denoised image with $\lambda = 30$.

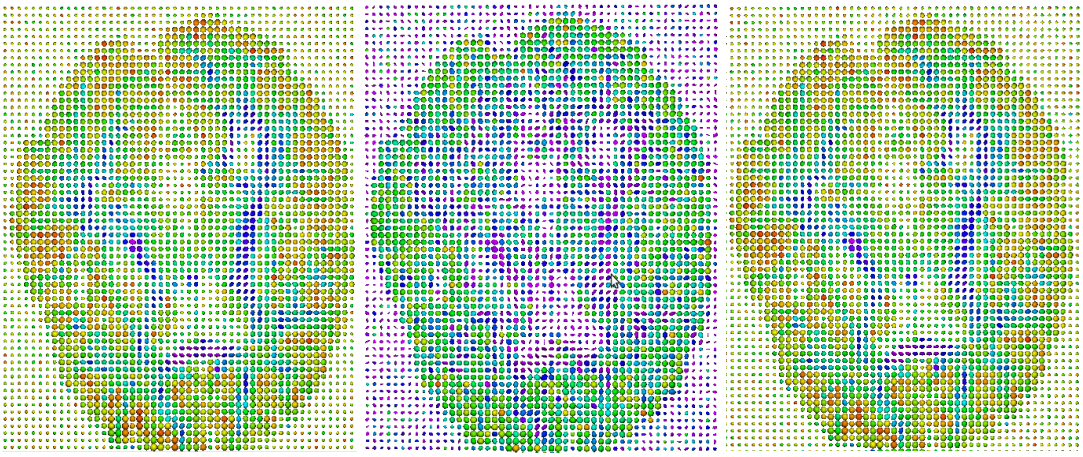


Figure 4.11: TV-regularized reconstruction of images with diffusion tensor data. Left-to-right: the original image, an image with added component-wise Gaussian noise, and the denoised image. Noise was of standard deviation 100, $\lambda = 1 \times 10^{-3}$.

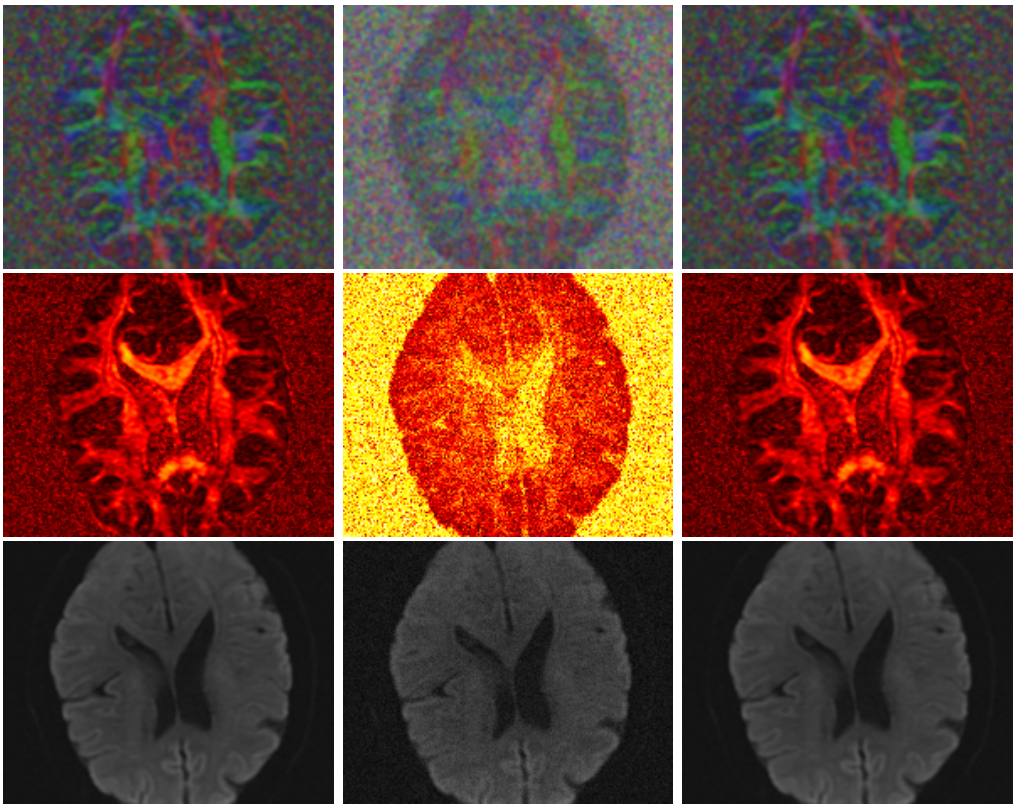


Figure 4.12: TV-regularized reconstruction of diffusion tensor data. Left-to-right: the original reconstruction without noise, the noisy least-squares fitting solution (used as initialization), and the regularized reconstruction result. Top-to-bottom: a visualization of the principal directions, the fractional anisotropy, and the mean diffusivity. The noise added to the field ratio logarithm was of strength 100, $\lambda = 1 \times 10^{-3}$.

Chapter 5

Sparse Priors for Structured-Light Reconstruction

5.1 Introduction

With ever more prevalent sources for 3D data, 3D acquisition and processing is an increasingly important part of scene analysis. Active illumination range scanners are used for scene understanding [104, 119, 143], robotics [94, 151, 176], object modeling [48, 81], indoor scene mapping [160], and human computer interaction [206], among other tasks.

Structured-light systems usually consist of a calibrated camera-projector pair, where coded light pattern sequences emitted by the projector are acquired by the camera, allowing robust triangulation and depth reconstruction. Time-multiplexed structured-light systems trade-off spatial for temporal resolution. They allow us to obtain dense and accurate reconstruction at low cost, with relatively simple hardware and without too many limiting assumptions on the scene. Other alternatives for structured-light attempt to trade-off resolution for coding robustness by incorporating decoding schemes for larger neighborhoods which add a certain assumption of regularity. For a review of existing structured-light techniques see, for example, [197].

In order to improve reconstruction robustness, many of the techniques used to re-

construct 3D depth via structured-light incorporate ad-hoc assumptions on the scene structure and the 3D imaging process. These include, for instance, smoothness of the acquired surface [129, 259], or temporal objects behavior [98, 129, 259]. This regularity, however, is usually based on channel decoding error approaches (see for example [50, 108]), and does not relate to the geometry of the scene or the image formation model. As such, its optimality is often limited due to the inaccurate reconstruction error model.

Yet, modeling these assumptions in a more complete way is crucial when the captured illumination patterns are of low SNR, for example due to long scanning range and short camera exposure times. In the case of dynamic scenes, where some of the captured images are subject to abrupt intensity changes due to motion of depth discontinuities or albedo boundaries, failing to model the imaging process in a realistic manner may cause more reconstruction artifacts.

The probabilistic model we present here relates the time-multiplexed structured-light to methods for spatio-temporal stereo reconstruction [67, 218]. In our case, however, we are estimating the expected camera luminous intensity, rather than assuming brightness constancy.

Here, we obtain improved reconstruction results from structured-light scanners [174, 197], in face of challenging illumination conditions and motion artifacts, by providing strong priors for the imaging model and surface shape. Instead of using strong shape priors for range image correction, the approach we suggest incorporates shape and illumination priors into the reconstruction itself, giving us a principled approach of combining powerful surface priors and probabilistic understanding of the acquisition process. We use patch-based range image priors, similar to those successfully utilized for images, depth images, and surface processing [46, 79, 99, 134, 200, 227, 255, 256]. We demonstrate the priors obtained from range images to be quite intuitive and meaningful.

This chapter builds upon a previous conference paper [183], discussing more completely the reconstruction model and demonstrating additional priors. Furthermore, we

add more examples in order to test the behaviour of the algorithm in real-life low SNR conditions, and add an additional structured-light patterns scheme.

Specifically, in Section 5.2 we develop our reconstruction model. In Section 5.3 we describe the resulting reconstruction algorithm. We demonstrate our results and several aspects of the model's behavior on real images in Section 5.4. Section 5.5 concludes the chapter and discusses future venues of research.

5.2 Regularized Structured-Light Model

In shape from structured-light, we reconstruct the geometric structure of the scene based on active illumination. We illuminate the scene with projected patterns $I_P = \{I_P^{(i)}\}_{i=1}^N$, where N is the number of patterns, and capture a sequence of images $I_C = \{I_C^{(i)}\}_{i=1}^N$ with a camera. Let us denote the optical centers of the camera and projector by points $\mathcal{C}$ and $\mathcal{P}$ respectively. The overall setup is shown in Figure 5.1. In our formulation, we denote the estimated range image as $z(\mathbf{x})$. $\mathbf{x} \in \mathbb{R}^2$ is the (two-dimensional) camera image coordinates vector.

In this work we assume a Lambertian surface model for objects, and a projector emitting directional light in a temporal sequence of patterns. The main source of image noise is assumed to be the sensor/imaging process. Although other sources of deviations from the model exist (for example object motion), in many cases they can be overcome as we will show. Since structured-light systems decode a set of patterns and need all of the patterns to be decoded correctly, we can assume relatively low noise levels – the photon count per image sensor pixel is high enough so that the image noise model is approximately Gaussian, yet the signal is weak enough so that correctly decoding the coded light patterns poses a challenge. This is the typical scenario in real structured-light systems with temporal multiplexed code, aimed for example at capturing dynamic scenes, and thus requiring short exposure intervals.

Assuming a global illumination component and a projector illumination component,

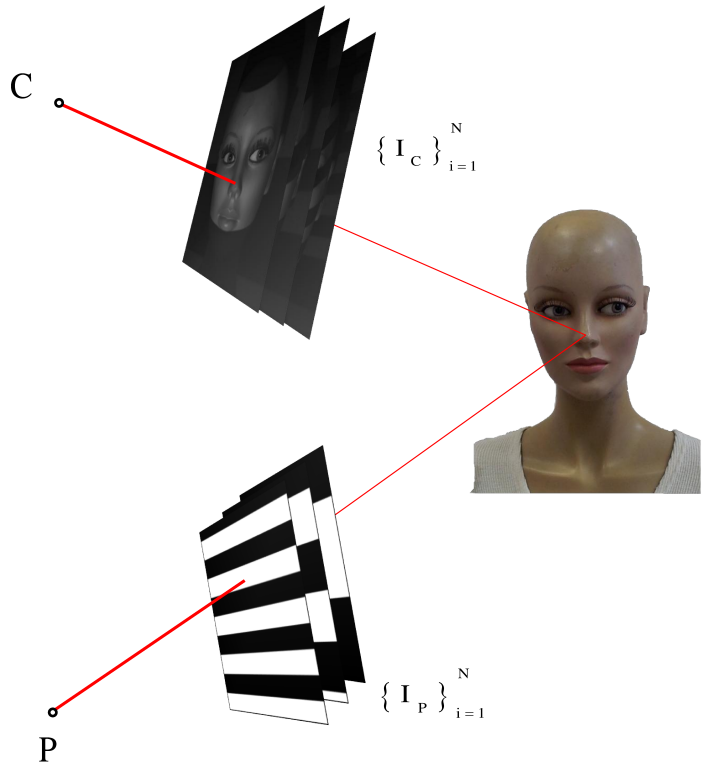


Figure 5.1: An example of a structured-light system setup.

we can model every pixel's intensity at each frame i as

$$I_C^{(i)}(\mathbf{x}) = a(\mathbf{x}) I_P^{(i)}\left(\Pi_z(\mathbf{x})\right) + b(\mathbf{x}) + n^{(i)}(\mathbf{x}), \tag{5.1}$$
$$n^{(i)}(\mathbf{x}) \sim N\left(0, \sigma_I^2\right).$$

$a(\mathbf{x})$ and $b(\mathbf{x})$ are pixel-wise coefficients that depend on the global illumination of the scene, the surface properties, object albedo, projector properties, and so forth. $\Pi_z(\mathbf{x})$ denotes the depth-dependent intensity transformation from pixel $\mathbf{x}$ to a corresponding pixel on the projector image. It is obtained by backprojecting the camera ray to depth z and projecting the point into the projector optical center. $n^{(i)}(\mathbf{x})$ is the pixel noise, assumed to be additive white Gaussian noise, independent and identically distributed

(i.i.d.) in space and time. The above model assumes a linear camera gain model. In practice, this approximate model works well enough so as to obtain good reconstruction results. Incorporating camera gain nonlinearity is deferred as future work since it is not necessary for this model.

We wish to formulate and maximize a probability function of the depth given the known camera images and projected textures. In reconstruction we are looking for the depth value $z(\mathbf{x})$ that maximizes the probability

$$\begin{aligned} z &= \underset{z}{\operatorname{argmax}} \min_{a,b} P\left(z, a, b | I_P, I_C\right) \\ &= \underset{z}{\operatorname{argmax}} \min_{a,b} \frac{P\left(z, a, b, I_P, I_C\right)}{P\left(I_P, I_C\right)} \\ &= \underset{z}{\operatorname{argmax}} \min_{a,b} \frac{P\left(I_P, I_C, a, b | z\right) P\left(z\right)}{P\left(I_P, I_C\right)} \\ &= \underset{z}{\operatorname{argmax}} \min_{a,b} P\left(I_P, I_C, a, b | z\right) P\left(z\right) \\ &= \underset{z}{\operatorname{argmin}} \min_{a,b} \left(-logP\left(I_P, I_C, a, b | z\right) - logP\left(z\right)\right), \end{aligned} \tag{5.2}$$

where we have applied Bayes' rule, and switched to log-probability domain. In order to obtain an efficient algorithm for computing and optimizing photoconsistency in the structured-light case, we note that we can incorporate the computation of the maximum-likelihood expressions for a, b into a plane-sweep operation [61] when seeking the optimum value of z. In the framework of probabilistic inference, this is known as max-sum elimination. Minimizing the negative log-probability over a and b, we have

$$\begin{aligned} &\min_{z} \min_{a,b} \left[-\log\left(P\left(I_P, I_C, a, b | z\right)\right)\right] = \\ &\min_{z} \left(\min_{a,b} \left[\sum_{i} \frac{\left(a(\mathbf{x}) I_P^{(i)}\left(\Pi_z(\mathbf{x})\right) + b(\mathbf{x}) - I_C^{(i)}\left(\mathbf{x}\right)\right)^2}{\sigma_I^2} \right] \right). \end{aligned} \tag{5.3}$$

The optimal values of a and b for this least-squares fitting problem are given in analytical form by solving the normal equations using I_C,I_P at points $\mathbf{x}, \Pi_z(\mathbf{x})$, respec-

tively,

$$\begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} \mu_{PP} & \mu_P \\ \mu_P & N \end{pmatrix}^{-1} \begin{pmatrix} \mu_{CP} \\ \mu_C \end{pmatrix}, \tag{5.4}$$

$$\mu_P = \sum I_P^{(i)}\left(\Pi_z(\mathbf{x})\right),\ \mu_C = \sum I_C^{(i)}\left(\mathbf{x}\right),$$

$$\mu_{CP} = \sum I_C^{(i)}\left(\mathbf{x}\right) I_P^{(i)}\left(\Pi_z(\mathbf{x})\right),$$

$$\mu_{PP} = \sum \left(I_P^{(i)}\left(\Pi_z(\mathbf{x})\right)\right)^2.$$

Inserting the optimal a, b as a function of z and noting the conditional independence (given z) of neighboring pixel values $I_C(\mathbf{x}), I_P\left(\Pi_z(\mathbf{x})\right)$ provides us with a functional to minimize with respect to $z(\mathbf{x})$, similar to [221],

$$\begin{aligned} &\underset{z}{\operatorname{argmin}} \int_{\mathbf{x}} \min_{a,b} \left(-\log\left(P\left(I_P, I_C, a, b|z\right)\right)\right) d\mathbf{x} + \psi\left(z\right) = \\ &\underset{z}{\operatorname{argmin}} \int_{\mathbf{x}} \rho_{SL}\left(z; I_C, I_P, \mathbf{x}\right) dx + \psi\left(z\right). \end{aligned} \tag{5.5}$$

The expression $\rho_{SL}\left(z; I_C, I_P, \mathbf{x}\right)$ denotes a penalty for the photoconsistency assumption. In standard structured-light techniques, this term is often optimized per pixel in several steps, including binarization of the code letters, decoding of the code, and depth reconstruction. These separate steps, however (for any specific code) are sub-optimal, even if efficient to compute. In order to achieve robustness to noise and computational efficiency, these approaches treat binarization or code-word identification errors as general channel decoding errors, using robust codes which have a firm theoretical background, but which do not really model the channel characteristics for this specific problem. These characteristics should result from the imaging model and geometric relations, and should not be ignored.

The term $\psi\left(z\right)$ denotes our choice for approximating the negative log-probability prior for the surface shape, $-\log P(z)$. There are several possible choices of surface shape priors. These can incorporate either smoothness assumptions and more elaborate geometric priors, assumptions on local shape of patches on surfaces, or reasoning on natural depth image statistics [252]. In Section 5.2.1 we describe several possible

regularization priors for depth images.

Incorporating Illumination and Reflectance Characteristics An additional improvement to the model can be made if we take into account the regularity of scene characteristics such as object albedo, illumination and normals. These assumptions have been utilized in the context of shape-from-single-image reconstruction [21], but in our case a simpler model suffices. We augment the photoconsistency assumption by adding a prior for the choice of a, b,

$$\min_{z}\left(\min_{a,b}\left[\sum_{i}\frac{\left(a(\mathbf{x})I_{P}^{(i)}\left(\Pi_{z}(\mathbf{x})\right)+b(\mathbf{x})-I_{C}^{(i)}\left(\mathbf{x}\right)\right)^{2}}{\sigma_{I}^{2}}+\frac{\left(a(\mathbf{x})-\mu_{a}\right)^{2}}{\sigma_{a}^{2}}+\frac{\left(b(\mathbf{x})-\mu_{b}\right)^{2}}{\sigma_{b}^{2}}\right]\right), \quad (5.6)$$

where μ_a, μ_b are taken from a locally computed average. σ_a, σ_b are constants set manually, since estimating second-order moments from a small neighborhood of a noisy signal can be quite sensitive. This allows us to compute a, b even if only a few noisy frames are available, as is often the case with multiple color structured-light systems, as shown for example in Figure 5.5.

5.2.1 Regularization Terms for Depth Images

We now describe a few possible regularization terms for the depth image, representing various tradeoffs between model robustness and computational efficiency.

Total-Variation Regularization The minimum area [58] and *total-variation* [192] (TV) priors, and related smoothness measures have been suggested in several forms for regularization of range images [161] and surface reconstruction [118, 122, 214]. TV regularization for structured-light can be expressed as

$$\underset{z}{\operatorname{argmin}}\int_{\mathbf{x}}\rho_{SL}\left(z;I_{C},I_{P},\mathbf{x}\right)+\tilde{c}\|\nabla z\|dx, \quad (5.7)$$

where $\|\nabla z\|$ is the total variation of the range image, for some coefficient $\tilde{c}$. This form of regularization is strongly related to MRF-based structured-light [221]. A related

prior is the second order total variation,

$$\underset{z}{\operatorname{argmin}} \int_{\mathbf{x}} \rho_{SL}\left(z; I_C, I_P, \mathbf{x}\right) + \tilde{c}\|Hz\|dx, \tag{5.8}$$

where H denotes the matrix of second order derivatives of z, $H(z) = (z_{xx}, \sqrt{2}z_{xy}, z_{yy})$. This prior can be computed quite efficiently and lends itself to parallel computation [244]. Furthermore, it is well suited to the often-made approximation of the scene as a piecewise-linear surface.

Patch-based L_1 Prior for Structured-Light Another possibility for modeling range images involves assuming a local model for each patch of the surface. Regularizing the surface then expresses itself via the parameters of this model. This includes modelling via polynomials or similar functions, leading to the moving-least-squares [135] approach, or expressing the patch via a functional basis with sparse coefficients, leading to sparsity-based regularization. Priors for depth images based on patch-estimators are described, for example, in [111, 147, 200, 227].

In our case, we assume that the depth image can be locally viewed as a sparse combination of basis functions. We note by $\tilde{\psi}\left(\cdot\right)$ our prior for surface patches. This leads to a patch-based regularizer of the reconstruction,

$$\underset{z}{\operatorname{argmin}} \int_{\mathbf{x}} \rho_{SL}\left(z; I_C, I_P, \mathbf{x}\right) d\mathbf{x} + \tilde{c}_1 \sum_j \tilde{\psi}\left(P_j z\right), \tag{5.9}$$

where $P_i z$ denotes extraction of a small neighborhood i from the surface z. For example, for an L_1-sparse representation prior, Equation 5.5 becomes

$$\underset{z,\alpha_j}{\operatorname{argmin}} \int_{\mathbf{x}} \rho_{SL}\left(z; I_C, I_P, \mathbf{x}\right) + \tilde{c}_1 \left(\sum_j \|P_j z - D\alpha_j\|^2 + \lambda\|\alpha_j\|_1 \right), \tag{5.10}$$

where D denotes a dictionary for depth image patches, P_j denotes a matrix extracting block j from the image in column-stacked notation, and α_j denotes the representation coefficients of patch $P_j z$ in that dictionary.

Gaussian Mixtures Prior for Structured-Light Since depth images are expected locally to be very sparse, another approach of modeling them is by a Gaussian-mixture

patch model, which in a sense first selects the support set of atoms from a structured-sparsity dictionary where each Gaussian component defines a support set, and then estimates its coefficients. In this approach, data patches are assumed to be generated from a sparse Gaussian-Mixture model in patch-space, similar to the approach suggested by Yu et al. [256], and Zoran and Weiss [260]. The relation of this image model to sparsity has been thoroughly discussed in [256]. In our case, Gaussians are pre-learned from a set of depth images, although an adaptive approach, learning the component distributions from the processed image itself is also possible. Unlike the case of natural images, such a learning process would have to account for the bias of the depth image patches, and the nature of the noise in the initial reconstruction results, as described in Subsection 5.3.1. As the reconstruction errors are far from the standard additive noise model, learning under such an outliers noise is not trivial and is left for future research. The components of the Gaussian components form natural features of the range images, adapted to edges and corners. This is not surprising, and has often been demonstrated in sparsity-related literature. The components of the patch distribution obtained by a Gaussian mixture model (GMM) are given in Figure 5.3. The optimization problem can be written as

$$\begin{aligned} &\underset{z,\alpha_j,k_j}{\operatorname{argmin}} \int_{\mathbf{x}} \rho_{SL}\left(z; I_C, I_P, \mathbf{x}\right) + \\ &\sum_j \tilde{c}_1 \|P_j z - U_j f_j\|^2 + \\ &\lambda \left(f_j^k\right)^T \Sigma_k^{-1} \left(f_j^k\right) + \frac{\lambda}{2} log\left(\pi^N \left|\Sigma_k^{-1}\right|\right), \end{aligned} \tag{5.11}$$

where f_j^k denotes the coefficients used to represent patch j in terms of Gaussian component k_j. Σ_k denotes the covariance matrix of component k, N is the number of pixel in each patch, and U_j define the principal directions of the Gaussian component used for patch j. As is often the case in patch-based priors, the patches' mean is subtracted before coefficients estimation, and added before the synthesis of the new patch.

Change in the Dictionary Coefficients under Surface Transformations

The use of the sparse model for depth images revives the discussion of invariance often held in visual images reconstruction. The standard invariant properties often sought in visual image processing are those of scale, translation, and rotation. These degrees of freedom are not accounted for in the standard models for sparsity-based image pro-

cessing. It is, however, instructive to look at standard geometric primitives and reason about their transformation. Occlusion edges and corners are handled in depth images in the same way as in visual images – translational and in-plane rotational invariances are handled by multiple dictionary atoms. The robustness of the model to out-of-plane rotationals is partially obtained by changing the coefficients of the atoms, but in general, it is obtained in the same way, by multiplicity of the atoms in the dictionary - the standard sparse model is not invariant to rotations of the object and a rotated object will result in different atoms at each patch.

5.3 Alternating Minimization Algorithm for Regularized Structured-Light

We now describe the specific algorithm used to solved structured-light reconstruction with a sparsity-based prior, as shown in Equation 5.10. A complete algorithmic description is given as Algorithm 5.1.

We assume the coded light pattern can be initially reconstructed by minimizing per-pixel the decoding error function $\rho_{SL}(\mathbf{x}, I_C, I_P; z)$. While this reconstruction is usually obtained by binarization and decoding of the time-multiplexed code, we view it as a photoconsistency term between the structured-light patterns and the resulting camera image intensities [165], when estimating the illumination conditions. Note that this function depends only on the depth value and camera intensities per pixel. In order to obtain the regularized solution we suggest to use an alternating minimization. By adding a set of auxiliary variables, we decouple the problems of regularization and structured-light decoding. This is done by minimizing the functional in Equation 5.5, which is of a half-quadratic form [85]. Minimization with respect to the regularization term given z results in a denoising problem. For patch-based priors, the resulting approach is similar to the one shown in [116]. We now detail each of the minimization steps.

Solving for z The update of z depends on the structured-light patterns, and may not even be continuous. We note that for all of the regularization terms presented in subsection 5.2.1, the term coupling the regularization to the photoconsistnecy term is quadratic in $z_i(\mathbf{x})$, the patch-dependent representation of $z(\mathbf{x})$ in patch i. Therefore we can rewrite the term for each pixel $\mathbf{x}$ in z as the sum of a photoconsistency measure and a sum of squared distances from $\tilde{z}(\mathbf{x})$, an averaged version of $z(\mathbf{x})$ in all of the patches containing this pixel, with an aggregate weight $w(\mathbf{x})$, for every $\mathbf{x}$

$$z^{n+1} = \underset{z}{\operatorname{argmin}}\, \rho_{SL}(z) + \tilde{c}_1 w \|z - \tilde{z}\|^2. \tag{5.12}$$

A solution can be obtained by sweeping the set of possible z values, similar to stereo algorithms [61]. Doing this plane-sweep is highly suitable for parallel implementation on graphics processing units (GPUs) [251]. Note that plane-sweeps are discrete by nature, as are the coded patterns in many cases. This does not constitute a further disadvantage as they are of approximately the same resolution. In order to obtain convergence, however, and in order to allow sub-pixel precision, we minimize a linearly-interpolated photoconsistency, along with the quadratic distance in the second term of Equation 5.12. The depth estimated at each pixel is set according to the minimum of the interpolated cost function, allowing us to incoporate sub-pixel precision into the plane-sweeping operator, as can be seen in the results section. Achieving sub-pixel resolution is important both in terms of accuracy and in terms of the visual artifacts that accompany discrete-pixel reconstruction, as seen for example in Figure 5.4, where the staircasing effect in the noise of the median-filtered reconstruction is typical of discrete-patterns structured-light systems.

Solving an L_1 regularization Given a patch estimate $P_j z$, an update of its representation becomes a standard sparse approximation problem. Specifically, if we take our sparse prior to be of an L_1 regularity type, we can update α_j using iterative shrinkage [54],

$$\alpha_j^{n+1} = S_{\lambda t}\left(\alpha_j^n - 2tD^T\left(D\alpha_j^n - P_j z\right)\right), \tag{5.13}$$

where t is a gradient descent step, chosen to be small enough, and $S_{\lambda t}(\cdot)$ denotes the soft shrinkage operator,

$$S_{\lambda t}(y) = \begin{cases} 0, & |y| \leq \lambda t \\ y - \lambda t, & y > \lambda t \\ y + \lambda t, & y < -\lambda t \end{cases} \tag{5.14}$$

While faster iterative methods exist for L_1 minimization (see [250] for a few examples), because of the alternating minimization nature of our scheme, more complex steps may not lead to faster convergence. We therefore chose to use the original iterative shrinkage scheme. We note that the dictionary in our case is pretrained from a set of depth images. The exact training procedure is defined in Subsection 5.3.1.

Solving a GMM regularization prior In this case, the choice of Gaussian component and its coefficients are given by going over the Gaussian components, computing the corresponding linear estimator, and the resulting log-probability term. Given that patch $P_j z$ belongs to Gaussian mixture component k with basis U_k and covariance matrix Σ_k, the linear estimator for the coefficients f_j is given by

$$f_j^k = \left(\tilde{c}_1 U_k^T U_k + \lambda \Sigma_k^{-1}\right)^{-1} \left(\tilde{c}_1 U_k^T\right) P_j z. \tag{5.15}$$

The component k for each patch is chosen so that the minimum regularized error is achieved over all components,

$$\underset{k}{\operatorname{argmin}} \begin{pmatrix} \tilde{c}_1 \|P_j z - U_j f_j^k\|^2 + \\ \lambda \left(f_j^k\right)^T \Sigma_k^{-1} \left(f_j^k\right) + \frac{\lambda}{2} log\left(\pi^N \left|\Sigma_k^{-1}\right|\right) \end{pmatrix}, \tag{5.16}$$

where the GMM coefficients f_j^k are computed according to Equation 5.15. We refer the reader to Yu et al. [256] for further elaboration on the method, and comment that the Gaussian mixture component are pretrained on a dataset of images, as defined in Subsection 5.3.2.

5.3.1 Learning a Depth Dictionary

In order to learn a surface model from range images, several properties of the data must be taken into account. Since reconstruction errors are of an outlier nature, algorithms

Algorithm 5.1 Alternating Minimization Sparse Structured-Light

1: Compute initial reconstruction z by plane-sweeping, according to Equation 5.6.

2: **for** $k = 1, 2, \ldots,$ until convergence **do**

3: Update auxiliary variable for regularization:

- Update $\tilde{z}^k$ by TV denoising, or second order TV denoising, according to [244], or
- Update $\alpha_j^k(\mathbf{x})$ for all j, according to Equation (5.13), for L_1 regularization, or
- Update $\tilde{f_j^k}$ by GMM component selection and linear estimation according to Equation 5.15,5.16, for GMM regularization.

5: Update $z^k(\mathbf{x})$, according to Equation (5.12).

6: **end for**

such as KSVD [79] that assume an additive white Gaussian noise model. Such algorithms require some form of pre-processing and outlier removal in order to train on data with outliers. Furthermore, since many of the patches in range scans are of smooth surfaces, and since the KSVD algorithm is initialization-dependent, care must be taken to provide a diversified initial dictionary. We focus the algorithm on the less-frequent edge patches by clustering the data first using the mean-shift algorithm [62]. The resulting dictionary obtained from a set of 50 range scans is shown in Figure 5.2. We note that the examples used for testing are not part of this dataset. Thus we avoid overfitting for a specific subject. While the training data is from a specific class of human faces, the learned primitives are quite general, as can be seen in Figure 5.2. We leave the effect of different dictionary and training data choices for future research.

5.3.2 Learning a Gaussian-Mixture Model for Depth Images

For the GMM prior we have used 200 Gaussian components, learned from the same dataset as the sparse dictionary prior, and using the same type of pruning for flat and outlier-containing patches. Learning the GMM component was done in a standard way

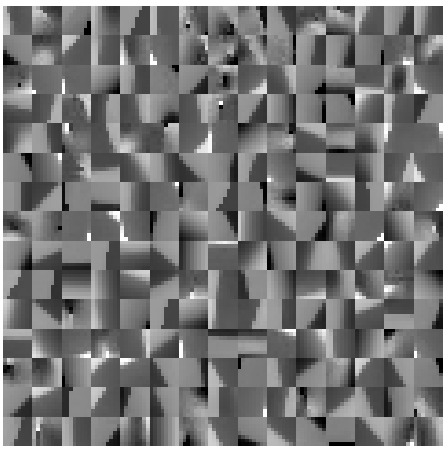


Figure 5.2: Example atoms from a dictionary of 300 words obtained from a set of 50 range scans.

(see [256] for more details). The components of the patch distribution obtained by a Gaussian mixture model are given in Figure 5.3.

5.4 Results

We now proceed to demonstrate the results of the proposed scheme. We first note that merely by using a sweeping approach instead of the usual decoding approach, we can improve the reconstruction. This is not surprising since the channel noise model used in the standard gray-code reconstruction was inaccurate to begin with. This improvement is obtained even without an additional regularization term, as shown in Figure 5.5. In this figure, in order to measure the amount of reconstruction outliers, we measure the deviation of the current depth beyond the $(0.4, 0.6)$ quantiles of the local neighborhood depth for a small (9×9) region. As can be seen in the zoomed images, there are significantly more outliers in the decoding-based reconstruction at areas with low illumination intensity such as the sides of the object, where the incident angle is large.

The importance of using a better per-pixel model can be clearly observed around the eyes of a reconstructed face, which is often a problematic area in 3D reconstruction due to the low reflection coefficient of the pupil. This is demonstrated in Figure 5.4, where reconstructing and then post-processing the depth image does not provide reasonable

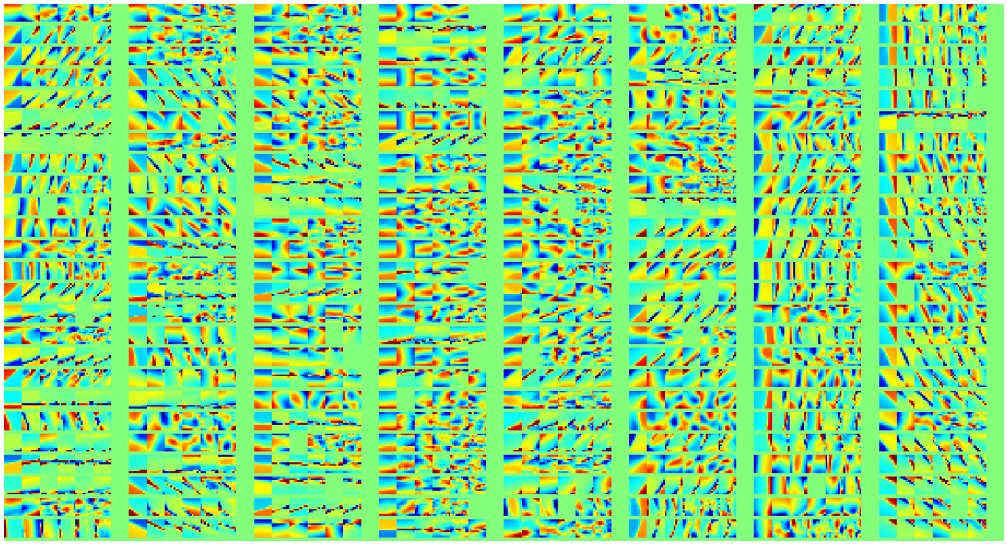


Figure 5.3: An example of 200 Gaussian mixture components obtained from a set of 50 range scans. Each 6-columns group of columns represents the principle directions of Gaussian components. Each row represents a Gaussian component, with the leftmost columns representing the more variable directions in the mixture.

reconstruction of the eyes region. Similarly, the sides of the face which are poorly illuminated by the projector suffer from reconstruction artifacts as well.

The main structured-light patterns scheme we experimented with is a standard structured-light setup similar to [199], with 10 striped black and white patterns, along with an all-ones and all-zeros pattern. The camera images are sampled at a resolution of 320×240, and projector patterns are shot using a 1024×768 DLP projector. In order to simulate low-SNR conditions, we have added Gaussian noise to the camera images before reconstruction. Results are shown in Figure 5.6,5.7 for the case of structured-light images with intensity Gaussian noise of standard deviations 5 and 10.

In order to quantitatively validate our method, we take as ground truth an almost-noiseless range image of the head statue, and measure range errors compare to it. We compare both L_1 and robustified L_2, truncated at 10 millimeters. The error measurements are performed over a manually segmented mask of the 3D object in the image domain. The results of this comparison are given in Table 5.1. For all of the images, the dictionary trained for patched-based priors was of patch size 8×8. As can be seen,

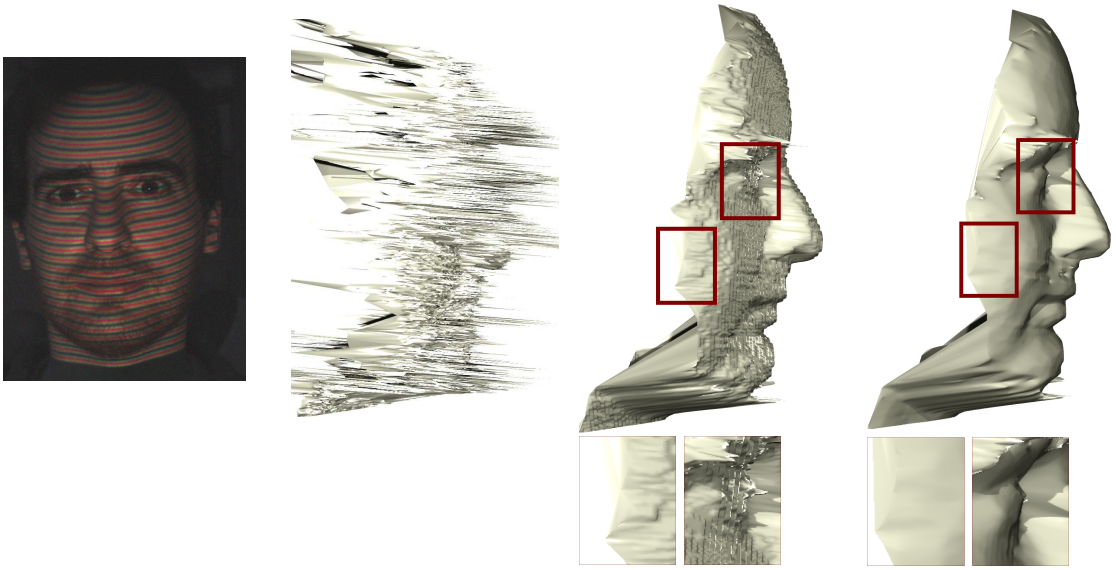


Figure 5.4: An example reconstruction of the eye region of a person. Left-to-right: the intensity image based on the structured-light setup of [191], the result obtained by plane-sweeping according to Equation 5.6 with no post-processing, the result after median filtering, and the result of regularized reconstruction using Equation 5.8.

the error of the median filtered result is smaller than those of sparse denoising with robust fitting term, or that of TV regularized reconstruction. This is due to the fact that TV regularization is too weak to overcome errors in the data term, and denoising with an L_1 term is still somewhat sensitive to the strong outliers found in structured-light reconstructed depth images.

We compare our results to several approaches. A common way of removing reconstruction artifacts is by median filtering, as was done in [191]. We compared to median post-processing, taken with the smallest filter size that removed range outliers from the face, in order to avoid oversmoothing. Yet another approach treats the problem as a denoising problem with a strong prior and impulse noise assumption. An example of this type of method would be to take the same depth prior we use, but solve a denoising problem with an L_1 fidelity term

$$\underset{z}{\operatorname{argmin}} \int_{\mathbf{x}} \|z - z_0\| d\mathbf{x} + \tilde{c}_1 \sum_j \tilde{\psi}\left(P_j z\right), \tag{5.17}$$

where z_0 is the reconstruction results without a prior. This approach would be similar,

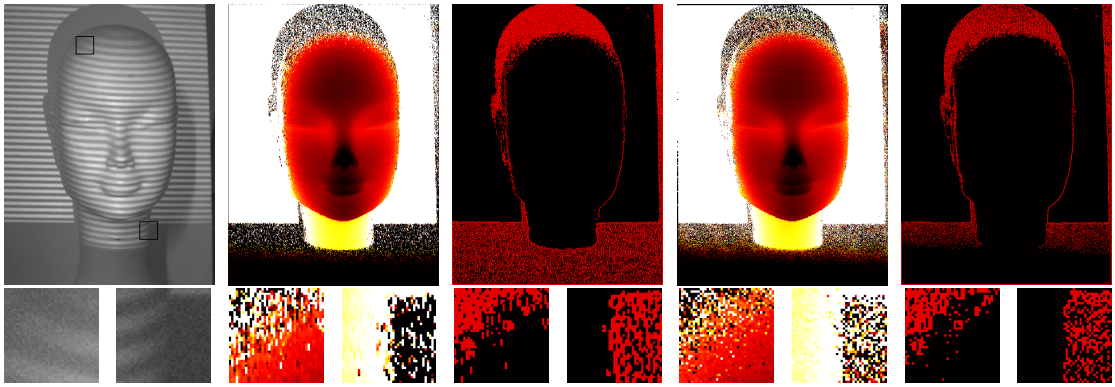


Figure 5.5: Top row, left-to-right: One of the texture images, the result obtained by the method of [191], and an outlier map (red pixels signify gross errors in the reconstruction), the result obtained by plane-sweeping according to Equation 5.6, and an outlier map. Bottom row: two zoomed-in areas of low SNR, marked as boxes in the texture intensity image. In these areas of weak illumination, plane-sweeping results in fewer outliers compared to a standard decoding approach for structured-light.

in a sense, to the depth image denoising suggested in [227]. This approach is marked in Table 5.1 under the *Sparse Denoise* column. In addition, it would be interesting to try a weaker prior for reconstruction such as TV regularization as suggested in Section 5.3. This approach is shown in the table as column *TV*. For all of the methods, parameters were chosen so as to obtain optimal robust L_2 results, while preventing remaining depth outliers. The table demonstrates the effectiveness of the proposed algorithm. While the computational cost of our algorithm is quite high with current Matlab code, the algorithm is highly parallelizable and one future line of work involves fast parallel implementation of this algorithm.

In Figure 5.8 we demonstrate the results of our algorithm on artifacts caused by head motion in the vertical direction. Even though the assumption of constant $a(\mathbf{x}), b(\mathbf{x})$ breaks down, the algorithm overcomes many of the errors caused by a decoding-based reconstruction followed by outlier removal. The size of the median filter is chosen to be the smallest size that filters the motion artifacts over the eyes and mouth regions, a 7×7 filter in this case. We note that at this filter size, the mouth and nose areas merge, while artifacts remain on the eyelids.

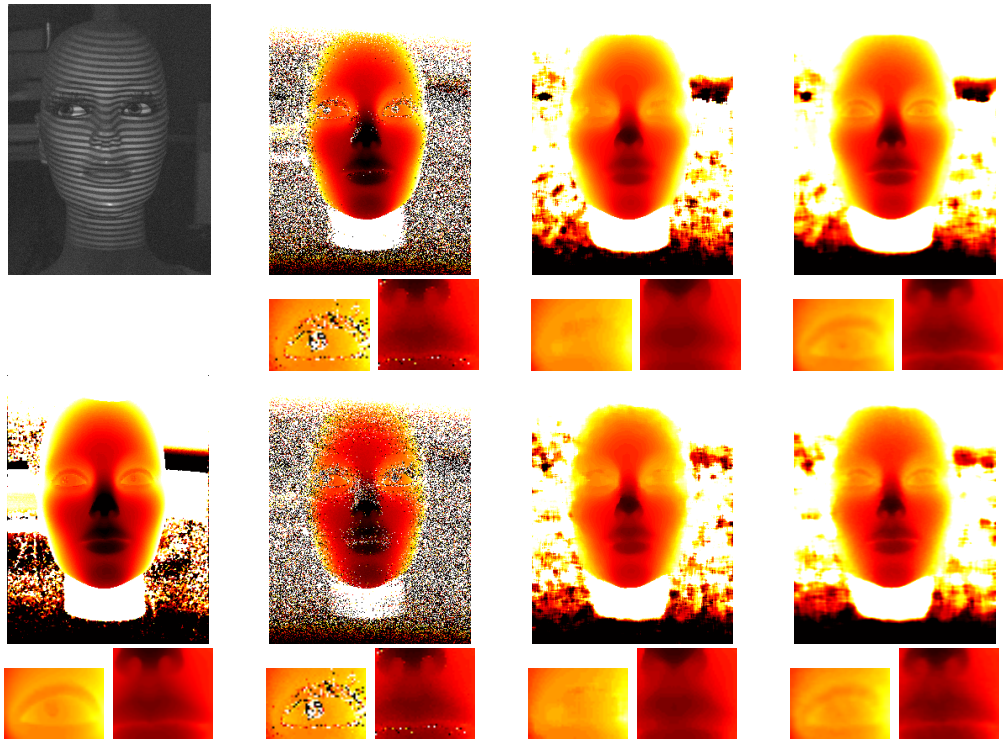


Figure 5.6: First row, left-to-right: An example textured pattern, reconstruction results, reconstruction with median filtering, reconstruction with sparse prior, where camera images were added Gaussian noise with standard deviation of 5, with close-up on the right eye region and the nose and mouth region. Second row, left-to-right: ground-truth reconstruction obtained from noiseless reconstruction, same sequence of results, where camera images were added Gaussian noise with standard deviation of 10. In order to view the range images, color and/or online viewing is suggested.

5.4.1 Color Structured-Light Example

Another example patterns scheme we used involves a color pattern projector, similar to [191]. In this setup, a single grayscale camera is used, operating at a resolution of 480×360, at 180 frames-per-second. The exposure time is $5.56ms$, due to synchronization between the projector and camera (see [191] for more details). A DLP projector emits color patterns sequentially in each cycle, and 12 patterns are used, 4 at each channel. In such a patterns set, since only 4 patterns are available per color channel, estimation of a, b is sensitive to image noise.

Noise Level	Raw	Median	TV	Sparse Denoising	Sparse Reconst.	Raw	Median	TV	Sparse Denoising	Sparse Reconst.
	L_2 error	L_2 error	L_2 error	L_2 error	L_2 error	L_1 error	L_1 error	L_1 error	L_1 error	L_1 error
2.5	1.4608	0.8411	0.8744	0.8680	0.8191	0.5996	0.4255	0.4240	0.4298	0.3379
5	2.6443	1.1033	1.1508	1.1768	0.9584	1.2013	0.5696	0.5689	0.6356	0.4135
7.5	3.9080	1.5315	1.715	1.8136	1.3489	2.1032	0.7384	0.7164	0.9489	0.5603
10	4.9841	1.9399	2.3866	2.758	1.7490	3.0949	0.9840	1.216	1.288	0.7571

Table 5.1: Error measurement at various noise levels, for structured-light reconstruction, and noise reduction by median post-processing, reconstruction with TV prior, reconstruction followed by sparse denoising, and reconstruction using a sparse prior as shown in Algorithm 5.1. Errors are shown as robust L_2 (truncated at 10mm) and L_1 errors, in millimeters, over the region of the scanned object.

It is quite important in this setup to have a prior for a, b as part of the model. Incorporating such a prior as shown in Equation 5.6 contributed greatly to the reconstruction performance. The result of the reconstruction is shown in Figures 5.4,5.9. The noise levels in these examples are not very high, but these examples are important because they demonstrate a real structured-light scenario, with real sensor short exposure artifacts. The frame-rate of the camera, about 15Hz, is still relatively low. It is therefore important to stress that in faster scanners short exposure time (and the resulting artifacts) is likely to play an even more significant role.

In Figure 5.9 we demonstrate the results using the Gaussian mixture model prior shown in Figure 5.3. This result demonstrates the generality of the proposed framework using a different regularization term. The Gaussian mixture components we used are shown in Figure 5.3.

5.5 Conclusions

In the chapter we presented a novel model for regularized structured-light reconstruction. Incorporating a sparse surface prior into a physically-motivated probabilistic outlook on structured-light decoding, we demonstrate accurate results in scenarios where the usual approach for decoding structured-light tends to fail.

The results obtained merit the coupling of a strong surface prior with a probabilis-

tic model for structured-light reconstruction, and motivate further exploration of the benefits of the proposed method as well as investigating the use of this approach for different types of depth scanners. Even in the case of no regularization, incorporating a realistic illumination model into the reconstruction cost function leads to a more robust reconstruction of each range pixel. An additional line of work involves implementing the current algorithm in an efficient manner, exploiting the high level of parallelism available in each phase. Other relevant venues of research include online learning of the surface model, and the incoproration of a more complete illumination model into the reconstruction.

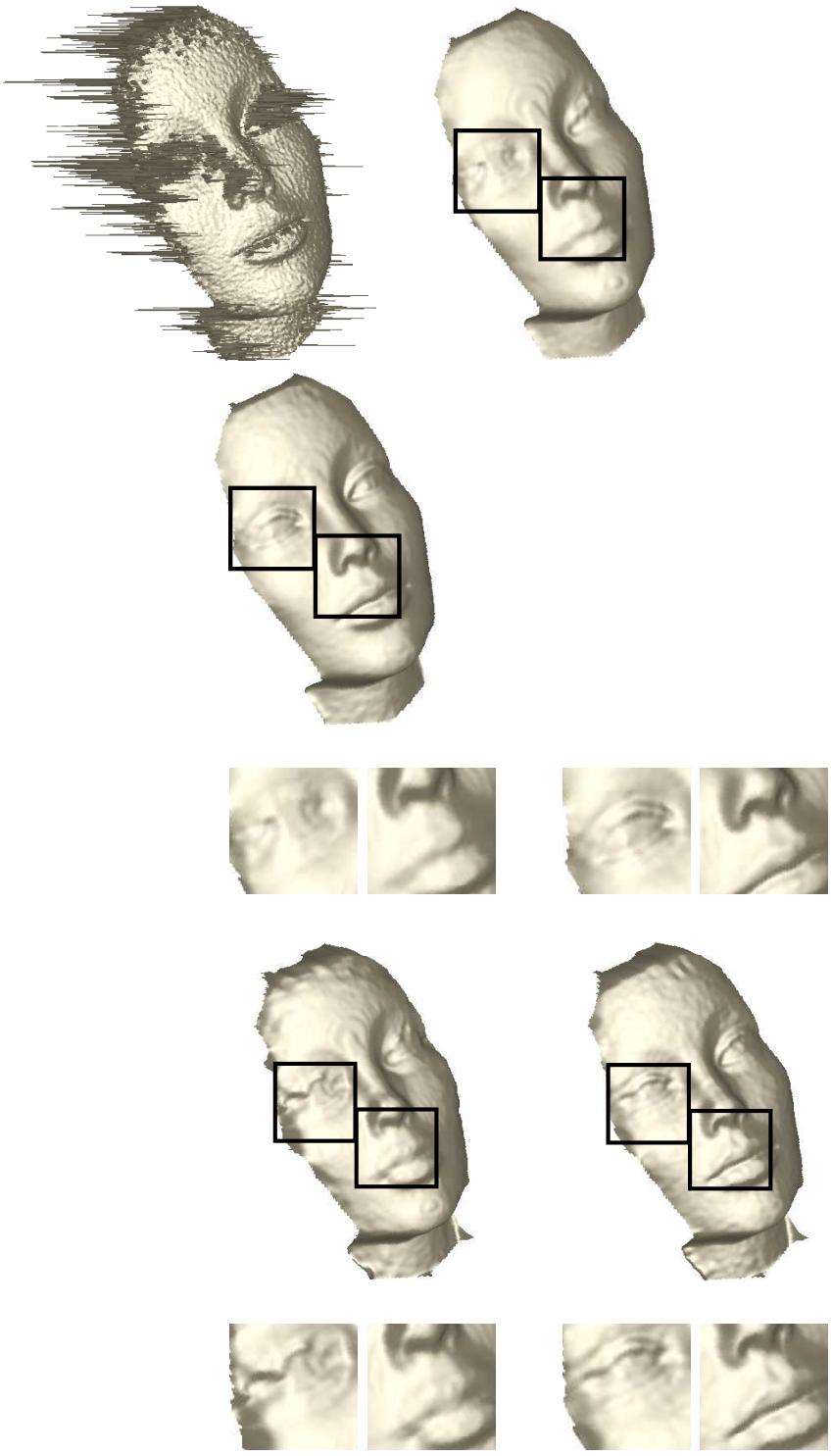


Figure 5.7: First row, left-to-right: 3D raw reconstruction results, reconstruction with median post-processing and with a sparse prior for the case of $\sigma = 5$ noise. Second row, left-to-right: (3D raw reconstruction omitted since it was too noisy), reconstruction with median post-processing and with a sparse prior for the case of $\sigma = 10$ noise. In order to view the range images, color and/or online viewing is suggested.

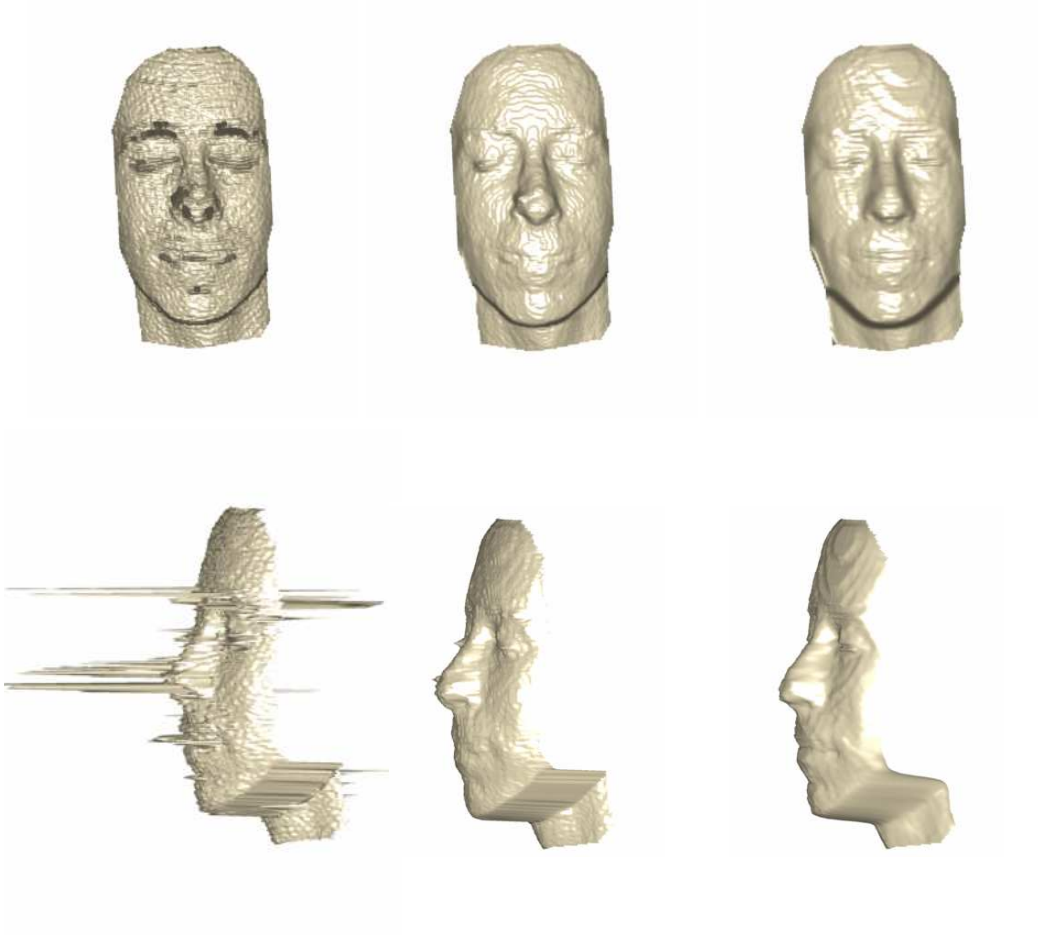


Figure 5.8: Left-to-right: An example with artifacts caused by vertical head motion, a median-filtered result, the result of the proposed method. Note the merging of the mouth and nose area in the median filter, and the remaining artifacts around the left eye and nose area.

Figure 5.9: Reconstructed surface based on the structured-light setup of [191], with GMM prior. Top, Left-to-right: one of the camera images in all 3 channels, raw reconstruction front view, median-filtered initial solution, regularized reconstruction with GMM prior. Bottom: reconstruction, side view.

Chapter 6

Discussion and Conclusions

In this work we discussed an extension to the over-parameterized framework for the case of non-linear parameterization models and various regularization terms. The resulting flexibility of the framework provides for various new applications for this framework, while still using axiomatically developed regularization terms and well-founded parameterization models.

6.1 The Generality of the Proposed Approach

We demonstrated several cases where a well-informed choice of parameterization for motion and depth allows us to solve key problems in structure and motion estimation. The resulting framework demostrate the power of a careful choice of parameterization, coupled with regularization of the model parameters.

In the case of 2D stereo motion, a new method for optical flow computation was presented, which hinges on a guiding principle that optic flow regularization should have a strong theoretical foundation related to the scene geometry, while having as few parameters as possible. The method is applicable to static scenes and retrieves meaningful local motion parameters related to the scene geometry. At each pixel, the parameters provide an estimation of the plane tangent to the scene manifold, up to a fixed shift and scale. To that extent, they can be seen as a higher level output than

optical flow in the computer vision hierarchy.

In 3D articulated motion segmentation, we have demonstrated a clear axiomatic development of a segmentation functional for this case, based on the Mumford-Shah framework. The highly-related TV regularization term can be generalized in our case into fast regularization schemes for various types of matrix-valued images. This results in various applications, both in smoothing and denoising, as well as more general inverse problems. Using the augmented-Lagrangian technique, we split the optimization problem into highly-parallelizable and efficient to solve subproblems. This results in an algorithm that achieves real-time speed for a nonlinear inverse problem with several possible applications.

In the case of 3D reconstruction, we demonstrate how the structured-light reconstruction problem can be rewritten as stereo motion estimation based on a probabilistic model. This probabilistic model described the geometry and illumination model for the object and the scanner system. This outlook connects this problem to the discussion of motion parameterization and allows us to incorporate high-level priors on the scene geometry and plausible shapes to be reconstructed – despite the nonlinearity of the inverse problem. We have examined several such priors, and shown them to allow great improvement in the reconstruction robustness, while expressing a quite intuitive model of the local 3D structure.

These examples highlight the interplay of parameterization and regularization in motion and structure estimation problems, and demonstrate the possible future benefits of such approaches. The generality of the proposed framework suggests new ways to extend the over-parameterized approach well beyond linear estimation models, and places the emphasis on the choice of parameters and nonlinear regularization terms.

6.2 Optimization Techniques

The optimization techniques demonstrated in the algorithms presented are local methods, acting on nonconvex functionals. Hence, they achieve a local minimum, which

may not be always optimal, as noted before in the case of over-parameterized optical flow. We expect global optimization techniques being developed to provide us with better results in the future.

Another question regarding the optimization of the solution involves the the question of whether the global minimum for a more elaborate model is the correct one. This question has been previously raise by Wedel et al. [237], and it is an important one. There are two prominent ways to tackle this question while handling overparameterized models. One approach is by verifying the properties of the solution under a well defined problem instance. We have done this in Chapter 2, where the optimal solution is shown to be a global minimizer under conditions relating to the scene geometry. Another approach is by choosing the optimization steps in ways that lead to a descent step, but incorporate additional knowledge on the problem instance. Since we have focused on axiomatic approaches for formulating and solving the problems described, we found these techniques to be less aligned with the focus of this thesis, but they remain a valid and important research direction.

Indeed, much work is required if novel parameterizations and over-parameterizations are to be fully utilized. The resulting cost functions have many local minima, and generic global minimization algorithms cannot solve them in reasonable time. We expect specially tailored global optimization techniques with domain-specific update steps to help reduce the computational complexity of the search.

Finally, in the way of more advanced models, we believe more natural parameterization can be proposed for addtional, more elaborate, problems. Such problems may include higher-order tensors, and stereoscopic models that include motion between the frames, helping strengthen the connection between low-level vision and high-level scene understanding.

We expect the methods and models presented in this thesis to allow solutions of these problems and more, extending the basis of axiomatic over-parameterized approaches

and help bridging these approaches to more data-driven parameterizations, resulting in significant benefits to various fields of computer vision, machine intelligence, and computer science.

References

[1] Fingerprints Verification Competition database.

[2] 3DSlicer software package.

[3] E. J. Alexander, C. Bregler, and T. P. Andriacchi. Non-rigid modeling of body segments for improved skeletal motion estimation. *Computational Modeling Engineering Science*, 4:351–364, 2003.

[4] L. Alvarez, F. Guichard, P.-L. Lions, and J.-M. Morel. Axioms and fundamental equations of image processing. *Archive for Rational Mechanics and Analysis*, 123:199–257, Sept. 1993.

[5] L. Ambrosio and V. M. Tortorelli. Approximation of functional depending on jumps by elliptic functional via Γ-convergence. *Communications on Pure and Applied Mathematics*, 43(8):999–1036, 1990.

[6] T. Amiaz and N. Kiryati. Piecewise-smooth dense optical flow via level sets. *International Journal of Computer Vision*, 68(2):111–124, 2006.

[7] T. Amiaz, E. Lubetzky, and N. Kiryati. Coarse to over-fine optical flow estimation. *Pattern Recognition*, 40(9):2496–2503, 2007.

[8] M. S. Andersen, D. L. Benoit, M. Damsgaarda, D. K. Ramsey, and J. Rasmussen. Do kinematic models reduce the effects of soft tissue artefacts in skin marker-based motion analysis an in vivo study of knee kinematics. *Journal of Biomechanics*, 43:268–273, 2010.

[9] D. Anguelov, D. Koller, H.-C. Pang, P. Srinivasan, and S. Thrun. Recovering articulated object models from 3D range data. In *Proc. Conf. on Uncertainty in Artificial Intelligence*, pages 18–26. AUAI Press, 2004.

[10] D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, J. Rodgers, and J. Davis. Scape: shape completion and animation of people. *ACM Trans. Graph.*, 24(3):408–416, 2005.

[11] R. Arcila, S. K. Buddha, F. Hétroy, F. Denis, and F. Dupont. A framework for motion-based mesh sequence segmentation. In *Int. Conf. on Comp. Graphics, Visual. and Comp. Vision*, pages 33–40, Plzen, Tchéquie, 2010.

[12] S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Y. Wu. An optimal algorithm for approximate nearest neighbor searching fixed dimensions. *J. ACM*, 45(6):891–923, 1998.

[13] A. P. Ashbrook, R. B. Fisher, C. Robertson, and N. Werghi. Segmentation of range data into rigid subsets using surface patches. In *International Conference on Computer Vision*, pages 201–206, 1998.

[14] M. Attene, S. Katz, M. Mortara, G. Patane, M. Spagnuolo, and A. Tal. Mesh segmentation - a comparative study. In *Proc. IEEE Int. Conf. on Shape Modeling and Applications*, pages 7–18, Washington, DC, USA, 2006. IEEE Computer Society.

[15] H. Attouch. *Variational convergence for functions and operators*. Applicable mathematics series. Pitman Advanced Publishing Program, 1984.

[16] H. Attouch, J. Bolte, P. Redont, and A. Soubeyran. Proximal alternating minimization and projection methods for nonconvex problems: An approach based on the Kurdyka-Lojasiewicz inequality. *Math. Oper. Res.*, 35:438–457, 2010.

[17] S. Baker, S. Roth, D. Scharstein, M. Black, J. Lewis, and R. Szeliski. A Database and Evaluation Methodology for Optical Flow. In *International Conference on Computer Vision*, pages 1–8, 2007.

[18] R. Ball. *A treatise on the theory of screws*. Cornell University Library historical math monographs. University Press, 1900.

[19] L. Bar, N. A. Sochen, and N. Kiryati. Variational pairing of image segmentation and blind restoration. In *European Conf. Computer Vision*, pages 166–177, 2004.

[20] D. Barash, T. Schlick, M. Israeli, and R. Kimmel. Multiplicative operator splittings in nonlinear diffusion: from spatial splitting to multiple timesteps. *Journal of Mathematics in Imaging and Vision*, 19(16):33–48, 2003.

[21] J. T. Barron and J. Malik. Shape, albedo, and illumination from a single image of an unknown object. In *CVPR*, pages 334–341, 2012.

[22] T. Basha, Y. Moses, and N. Kiryati. Multi-view scene flow estimation: A view centered variational approach. In *Computer Vision and Pattern Recognition*, pages 1506–1513, 2010.

[23] P. J. Basser, J. Mattiello, and D. LeBihan. MR diffusion tensor spectroscopy and imaging. *Biophysical journal*, 66(1):259–267, Jan. 1994.

[24] M. Belkin, J. Sun, and Y. Wang. Constructing Laplace operator from point clouds in rd. In *Symposium on Discrete Algorithms*, pages 1031–1040, Philadelphia, PA, USA, 2009. SIAM.

[25] C. Belta and V. Kumar. On the computation of rigid body motion. *Electronic Journal of Computational Kinematics*, 1, 2001.

[26] C. Belta and V. Kumar. An svd-based project method for interpolation on se(3). *IEEE Trans. Robotics and Automation*, 18(3):334–345, June 2002.

[27] R. Ben-Ari and N. Sochen. Variational stereo vision with sharp discontinuities and occlusion handling. In *International Conference on Computer Vision*, volume 0, pages 1–7, 2007.

[28] R. Ben-Ari and N. A. Sochen. Stereo matching with mumford-shah regularization and occlusion handling. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(11):2071–2084, 2010.

[29] J. Bergen, P. Anandan, K. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. In *ECCV*, volume 2, pages 237–252. Springer Verlag, 1992.

[30] Ø. Bergmann, O. Christiansen, J. Lie, and A. Lundervold. Shape-adaptive DCT for denoising of 3D scalar and tensor valued images. *J. Digital Imaging*, 22(3):297–308, 2009.

[31] P. J. Besl and N. D. McKay. A method for registration of 3D shapes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14(2):239–256, 1992.

[32] S. Birchfield and C. Tomasi. Depth discontinuities by pixel-to-pixel stereo. *International Journal of Computer Vision*, 35(3):269–293, Dec. 1999.

[33] D. Bitton, G. Rosman, T. Nir, A. M. Bruckstein, A. Feuer, and R. Kimmel. Over-parameterized optical flow using a stereoscopic constraint. Technical report, Technion, 2009. CIS2009-18.

[34] M. J. Black. *Robust Incremental Optical Flow*. Dissertation, Yale University, December 1992.

[35] M. J. Black and P. Anandan. A framework for the robust estimation of optical flow. In *International Conference on Computer Vision*, pages 231–236, 1993.

[36] K. Bredies, K. Kunisch, and T. Pock. Total generalized variation. *SIAM J. Imaging Sciences*, 3(3):492–526, 2010.

[37] L. Bregman. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR Computational Mathematics and Mathematical Physics*, 7(3):200 – 217, 1967.

[38] X. Bresson and T. Chan. Fast dual minimization of the vectorial total variation norm and applications to color image processing. *Inverse Problems and Imaging*, 2(4):455–484, 2008.

[39] R. Brockett. Robotic manipulators and the product of exponential formula. In P. Fuhrman, editor, *Proc. Mathematical Theory of Networks and Systems*, pages 120–129, 1984.

[40] A. M. Bronstein, M. M. Bronstein, and R. Kimmel. Generalized multidimensional scaling: a framework for isometry-invariant partial surface matching. *Proc. Natl. Acad. Sci. USA*, 103(5):1168–1172, January 2006.

[41] A. M. Bronstein, M. M. Bronstein, and R. Kimmel. *Numerical geometry of non-rigid shapes*. Springer Verlag, 2008.

[42] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In *European Conf. Computer Vision*, pages 25–36, Prague, Czech Republic, May 2004. Springer Verlag.

[43] A. Bruhn, J. Weickert, T. Kohlberger, and C. Schnörr. A multigrid platform for real-time motion computation with discontinuity-preserving variational methods. *Int. J. Comput. Vision*, 70(3):257–277, Dec. 2006.

[44] A. Bruhn, J. Weickert, and C. Schnörr. Lucas/Kanade meets Horn/Schunck: combining local and global optic flow methods. *International Journal of Computer Vision*, 61(3):211–231, 2005.

[45] C. Brune, H. Maurer, and M. Wagner. Detection of intensity and motion edges within optical flow via multidimensional control. *SIAM J. Imag. Sci.*, 2(4):1190–1210, 2009.

[46] A. Buades, B. Coll, and J.-M. Morel. A non-local algorithm for image denoising. In *Computer Vision and Pattern Recognition*, pages 60–65, 2005.

[47] A. D. Bue, X. J. ao, L. Agapito, and M. Paladini. Bilinear factorization via augmented Lagrange multipliers. In *European Conf. Computer Vision*, pages 283–296. Springer-Verlag, 2010.

[48] Y. Cai, A. Nee, and H. Loh. Geometric feature detection for reverse engineering using range imaging. *J. of Vis. Comm. and Image Repres.*, 7(3):205–216, September 1996.

[49] M. P. D. Carmo. *Differential Geometry of Curves and Surfaces*. Prentice-Hall, 1976.

[50] D. Caspi, N. Kiryati, and J. Shamir. Range imaging with adaptive color structured light. *IEEE Trans. Pattern Anal. Mach. Intell.*, 20(5):470–480, May 1998.

[51] E. Celledoni, F. Fassò, N. Säfström, and A. Zanna. The exact computation of the free rigid body motion and its use in splitting methods. *SIAM Journal on Scientific Computing*, 30(4):2084–2112, 2008.

[52] E. Celledoni and B. Owren. Lie group methods for rigid body dynamics and time integration on manifolds. *Computer Methods in Applied Mechanics and Engineering*, 19:421–438, 1999.

[53] A. Chambolle. An algorithm for total variation minimization and applications. *J. Math. Imaging Vis.*, 20(1-2):89–97, 2004.

[54] A. Chambolle, R. A. DeVore, N.-Y. Lee, and B. J. Lucier. Nonlinear wavelet image processing: variational problems, compression, and noise removal through wavelet shrinkage. *IEEE Trans. Image Process.*, 7(3):319–335, 1998.

[55] A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematics in Imaging and Vision*, 40(1):120–145, 2011.

[56] X. Chen, A. Golovinskiy, and T. Funkhouser. A benchmark for 3D mesh segmentation. *ACM Trans. on Graphics*, 28(3), Aug. 2009.

[57] Y. Chen and G. Medioni. Object modelling by registration of multiple range images. *Image Vision Comput.*, 10:145–155, April 1992.

[58] D. L. Chopp. Computing minimal surfaces via level set curvature flow. *J. Comput. Phys.*, 106(1):77–91, May 1993.

[59] U. Clarenz, U. Diewald, and M. Rumpf. Anisotropic geometric diffusion in surface processing. In *Proceedings of the 11th IEEE Visualization Conference*, pages 397–405. IEEE, 2000.

[60] I. Cohen. Nonlinear variational method for optical flow computation. In *Proc. Eighth Scandinavian Conference on Image Analysis*, volume 1, pages 523–530, 1993.

[61] R. T. Collins. A space-sweep approach to true multi-image matching. In *CVPR*, pages 358–363, 1996.

[62] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24:603–619, May 2002.

[63] P. Coupé, J. V. Manjón, V. Fonov, J. Pruessner, M. Robles, and D. L. Collins. Patch-based segmentation using expert priors: application to hippocampus and ventricle segmentation. *NeuroImage*, 54(2):940–954, Jan. 2011.

[64] D. Cremers and S. Soatto. Motion competition: A variational framework for piecewise parametric motion segmentation. *International Journal of Computer Vision*, 62(3):249–265, 2005.

[65] J. Cullum. Numerical differentiation and regularization. *SIAM Journal on Numerical Analysis*, 8(2):pp. 254–265, 1971.

[66] L. Dascal, G. Rosman, X.-C. Tai, and R. Kimmel. On semi-implicit splitting schemes for the Beltrami color flow. In *Scale Space and Variational Methods in Computer Vision*, pages 259–270, Berlin, Heidelberg, 2009. Springer Verlag.

[67] J. Davis, D. Nehab, R. Ramamoorthi, and S. Rusinkiewicz. Spacetime stereo: A unifying framework for depth from triangulation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(2):296–302, Feb. 2005.

[68] R. Deriche, D. Tschumperle, and C. Lenglet. DT-MRI estimation, regularization and fiber tractography. In *ISBI*, pages 9–12, 2004.

[69] M. Desbrun, M. Meyer, P. Schröder, and A. H. Barr. Implicit fairing of irregular meshes using diffusion and curvature flow. *SIGGRAPH*, pages 317–24, 1999.

[70] M. P. do Carmo. *Riemannian Geometry*. Birkhäuser Verlag, Boston, MA, 1992.

[71] M. Droske and M. Rumpf. A level set formulation for Willmore flow. *Interfaces and Free Boundaries*, 6(3):361–378, 2004.

[72] R. Duits and B. Burgeth. Scale spaces on Lie groups. In *Scale Space and Variational Methods in Computer Vision*, pages 300–312, Berlin, Heidelberg, 2007. Springer Verlag.

[73] Y. Dumortier, I. Herlin, and A. Ducrot. 4D tensor voting motion segmentation for obstacle detection in autonomous guided vehicle. In *IEEE Int. Vehicles Symp.*, pages 379–384, 2008.

[74] V. Duval, J.-F. Aujol, and L. A. Vese. Mathematical modeling of textures: Application to color image decomposition with a projected gradient algorithm. *Journal of Mathematics in Imaging and Vision*, 37(3):232–248, 2010.

[75] J. J. Eells and J. H. Sampson. Harmonic mappings of Riemannian manifolds. *American J. of Math*, 86(1):106–160, 1964.

[76] A. A. Efros and T. K. Leung. Texture synthesis by non-parametric sampling. In *Proceedings of the International Conference on Computer Vision-Volume 2 - Volume 2*, ICCV '99, pages 1033–, Washington, DC, USA, 1999. IEEE Computer Society.

[77] I. Ekeland and R. Temam. *Convex Analysis and Variational Problems*. CMS Books in Mathematics. SIAM, Philadelphia, 1999.

[78] M. Elad. *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*. Springer Science+Business Media, LLC, 2010.

[79] M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *Image Processing, IEEE Transactions on*, 15(12):3736–3745, 2006.

[80] O. Faugeras and Q. T. Luong. *The Geometry of Multiple Images*. The MIT Press, ISBN: 0262062208, 2001.

[81] A. W. Fitzgibbon, D. W. Eggert, and R. B. Fisher. High-level model acquisition from range images. *Computer-Aided Design*, 29(4):321–330, 1997.

[82] S. Fleishman, D. Cohen-Or, and C. T. Silva. Robust moving least-squares fitting with sharp features. In *SIGGRAPH*, pages 544–552, New York, NY, USA, 2005. ACM.

[83] R. Fletcher. Semi-definite matrix constraints in optimization. *SIAM J. on Cont. and Optimization*, 23(4):493–513, 1985.

[84] O. Freifeld and M. J. Black. Lie bodies: A manifold representation of 3D human shape. In *European Conf. Computer Vision*, pages 1–14, 2012.

[85] D. Geman and C. Yang. Nonlinear image recovery with half-quadratic regularization. *IEEE Transactions on Image Processing*, 5(7):932–946, 1995.

[86] W. Gibson. On the least-squares orthogonalization of an oblique transformation. *Psychometrika*, 27:193–195, 1962.

[87] G. Gilboa and S. Osher. Nonlocal Linear Image Regularization and Supervised Segmentation. *Multiscale Modeling & Simulation*, 6(2):595–630, 2007.

[88] R. Giryes, S. Nam, M. Elad, R. Gribonval, and M. E. Davies. Greedy-Like Algorithms for the Cosparse Analysis Model. partially funded by the ERC, PLEASE project, ERC-2011-StG-277906.

[89] B. Goldluecke, E. Strekalovskiy, and D. Cremers. The natural vectorial variation which arises from geometric measure theory. *SIAM J. Imag. Sci.*, 2012.

[90] T. Goldstein, X. Bresson, and S. Osher. Geometric applications of the split Bregman method: Segmentation and surface reconstruction. *J. of Sci. Comp.*, 45(1–3):272–293, 2010.

[91] T. Goldstein and S. Osher. The split bregman method for l1-regularized problems. *SIAM J. Imag. Sci.*, 2(2):323–343, Apr. 2009.

[92] T. F. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theor. Comput. Sci.*, 38:293–306, 1985.

[93] B. Goossens, Q. Luong, A. Pizurica, and W. Philips. An improved non-local denoising algorithm. In J. Astola, K. Egiazarian, and V. Katkovnik, editors, *Local and Non-Local Approximation in Image Processing, International Workshop, Proceedings*, pages 143–156, 2008.

[94] S. Gould, P. Baumstarck, M. Quigley, A. Y. Ng, and D. Koller. Integrating visual and range data for robotic object detection. In *ECCV Workshop on Multi-camera and Multi-modal Sensor Fusion Algorithms and Applications (M2SFA2)*, 2008.

[95] Y. Gur and N. Sochen. Fast invariant Riemannian DT-MRI regularization. In *International Conference on Computer Vision*, pages 1–7, 2007.

[96] Y. Gur and N. A. Sochen. Regularizing flows over lie groups. *Journal of Mathematics in Imaging and Vision*, 33(2):195–208, 2009.

[97] B. C. Hall. *Lie Groups, Lie Algebras,and Representations, An Elementary Introduction*. Springer Verlag, 2004.

[98] O. Hall-Holt and S. Rusinkiewicz. Stripe boundary codes for real-time structured-light range scanning of moving objects. In *International Conference on Computer Vision*, volume 2, pages 359–366. IEEE, 2001.

[99] C. Hane, C. Zach, B. Zeisl, and M. Pollefeys. A patch prior for dense 3d reconstruction in man-made environments. *2012 International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission*, 0:563–570, 2012.

[100] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2004.

[101] S. Hauberg, S. Sommer, and K. S. Pedersen. Gaussian-like spatial priors for articulated tracking. In *European Conf. Computer Vision*, pages 425–437. Springer Verlag, 2010.

[102] D. Hearn and M. P. Baker. *Computer graphics with OpenGL (3. ed.)*. Pearson Education, 2004.

[103] M. R. Hesteness. Multipliers and gradient methods. *Journal of Optimization Theory and Applications*, 4:303–320, 1969.

[104] G. Hetzel, B. Leibe, P. Levi, and B. Schiele. 3D object recognition from range images using local feature histograms. In *Computer Vision and Pattern Recognition*, volume 2, pages II–394 – II–399 vol.2, 2001.

[105] N. J. Higham. Matrix nearness problems and applications. In *Applications of Matrix Theory*, pages 1–27. Oxford University Press, Oxford, 1989.

[106] D. Hochbaum and D. Shmoys. A best possible approximation for the k-center problem. *Mathematics of Operations Research*, 10(2):180–184, 1985.

[107] B. K. Horn and B. G. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.

[108] E. Horn and N. Kiryati. Toward optimal structured light patterns. In *3-D Digital Imaging and Modeling, 1997. Proceedings., International Conference on Recent Advances in*, pages 28–35, 1997.

[109] F. Huguet and F. Devernay. A variational method for scene flow estimation from stereo sequences. *Computer Vision and Pattern Recognition*, 0:1–7, 2007.

[110] B. Huhle, T. Schairer, P. Jenke, and W. Straíer. Fusion of range and color images for denoising and resolution enhancement with a non-local filter. *Comput. Vis. Image Underst.*, 114(12):1336–1345, Dec. 2010.

[111] B. Huhle, T. Schairer, P. Jenke, and W. Straßer. Fusion of range and color images for denoising and resolution enhancement with a non-local filter. *Computer Vision and Image Understanding*, 114(12):1336 – 1345, 2010. Special issue on Time-of-Flight Camera Based Computer Vision.

[112] A. Iserles, H. Z. Munthe-kaas, S. P. Nørsett, and A. Zanna. Lie group methods. *Acta Numerica*, pages 215–365, 2000.

[113] A. K. Jain. *Fundamentals of digital image processing*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1989.

[114] V. Jain and H. Zhang. Robust 3D shape correspondence in the spectral domain. In *Proc. of Shape Modeling International*, pages 118–129, 2006.

[115] D. L. James and C. D. Twigg. Skinning mesh animations. *SIGGRAPH*, 24(3):399–407, Aug. 2005.

[116] K. Jia, X. Wang, and X. Tang. Optical flow estimation using learned sparse model. *Computer Vision, IEEE International Conference on*, 0:2391–2398, 2011.

[117] E. Kalogerakis, A. Hertzmann, and K. Singh. Learning 3D Mesh Segmentation and Labeling. *ACM Trans. on Graphics*, 29(3), 2010.

[118] R. Keriven and O. Faugeras. Complete dense stereovision using level set methods. In *5th European Conference on Computer Vision*, 1998.

[119] E. Kim and G. G. Medioni. 3D object recognition in range images using visibility context. In *IROS*, pages 3800–3807, 2011.

[120] W. Kim, J. Park, and K. Lee. Stereo matching using population-based mcmc. *International Journal of Computer Vision*, 83(2):195–209, June 2009.

[121] R. Kimmel, , and N. Sochen. Orientation diffusion or how to comb a porcupine. *special issue on PDEs in Image Processing, Computer Vision, and Computer Graphics, Journal of Visual Communication and Image Representation*, 13:238–248, 2002.

[122] R. Kimmel. 3D shape reconstruction from autostereograms and stereo. *Journal of Visual Communication and Image Representation*, 13(1-2):324–333, 2002.

[123] R. Kimmel. *Numerical Geometry of Images: Theory, Algorithms, and Applications*. Springer Verlag, 2003.

[124] J. Klappstein. *Optical-Flow Based Detection of Moving Objects in Traffic Scenes*. PhD thesis, Ruprecht-Karls-Universität, Heidelberg, July 2008.

[125] C. Koay, J. Carew, A. Alexander, P. Basser, and M. Meyerand. Investigation of anomalous estimates of tensor-derived quantities in diffusion tensor imaging. *Magn Reson Med*, 2006.

[126] M. Kobilarov, K. Crane, and M. Desbrun. Lie group integrators for animation and control of vehicles. *ACM Trans. Graph.*, 28(2):1–14, 2009.

[127] T. Koetsier. Euler and kinematics. In R. E. Bradley and C. E. Sandifer, editors, *Leonhard Euler: Life, Work and Legacy*, volume 5 of *Studies in the History and Philosophy of Mathematics*, pages 167 – 194. Elsevier, 2007.

[128] I. Kompatsiaris, D. Tzovaras, and M. G. Strintzis. Object articulation based on local 3D motion estimation. In *Proc. of ECMAST*, pages 378–391, London, UK, 1999. Springer Verlag.

[129] T. Koninckx and L. Van Gool. Real-time range acquisition by adaptive structured light. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(3):432–445, 2006.

[130] R. Lai and S. Osher. A splitting method for orthogonality constrained problems. Technical report, UCLA, 2012.

[131] P. M. Larochelle, A. P. Murray, and J. Angeles. *On advances in robot kinematics*, chapter SVD and PD based projection metrics on SE(N), pages 13–22. Kluwer, 2004.

[132] T.-Y. Lee, Y.-S. Wang, and T.-G. Chen. Segmenting a deforming mesh into near-rigid components. *Vis. Comput.*, 22(9):729–739, 2006.

[133] V. Lempitsky, C. Rother, S. Roth, and A. Blake. Fusion moves for markov random field optimization. *IEEE Trans. Pattern Anal. Mach. Intell.*, 99(PrePrints), 2009.

[134] F. Lenzen, K. I. Kim, R. Nair, S. Meister, H. Schafer, F. Becker, C. Garbe, and C. Theobalt. Denoising strategies for time-of-flight data. In *Time-of-Flight Imaging: Algorithms, Sensors and Applications*, 2012.

[135] D. Levin. The approximation power of moving least-squares. *Math. Comput.*, 67(224):1517–1531, 1998.

[136] H. Li, R. W. Sumner, and M. Pauly. Global correspondence optimization for non-rigid registration of depth scans. *Computer Graphics Forum*, 27(5), July 2008.

[137] S. Lie. Die grundlagen f ur die theorie der unendlichen kontinuierli chen transformationsgruppen. i. treatise. In *Gesammelte Abhandlunge*, volume 6, pages 300–330, 1891.

[138] D. Lin, W. Grimson, and J. Fisher. Learning visual flows: A Lie algebraic approach. In *Computer Vision and Pattern Recognition*, pages 747–754, 2009.

[139] T. Lindeberg. *Discrete Scale-Space Theory and the Scale-Space Primal Sketch*. PhD thesis, Royal Institute of Technology, Stockholm, 1991.

[140] N. Litke, M. Droske, M. Rumpf, and P. Schröder. An image processing approach to surface matching. In *Eurographics Symp. on Geom. Proc.*, pages 207–216, Aire-la-Ville, Switzerland, Switzerland, 2005. Eurographics Association.

[141] Y. Liu, X. Cao, Q. Dai, and W. Xu. Continuous depth estimation for multi-view stereo. In *Computer Vision and Pattern Recognition*, pages 2121–2128, 2009.

[142] S. P. Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, IT-28(2):129–137, 1982.

[143] T.-W. R. Lo and J. P. Siebert. Local feature extraction and matching on range images: 2.5d sift. *Comput. Vis. Image Underst.*, 113:1235–1250, December 2009.

[144] T. Lü, P. Neittaanmaäki, and X.-C. Tai. A parallel splitting up method and its application to Navier-Stokes equations. *Applied Mathematics Letters*, 4(2):25–29, 1991.

[145] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *International Joint Conference on Artificial Intelligence*, pages 674–679, april 1981.

[146] A. Lundervold. On consciousness, resting state fMRI, and neurodynamics. *Nonlinear Biomed Phys*, 4 Suppl 1, 2010.

[147] M. Mahmoudi, G. Sapiro, M. Mahmoudi, and G. Sapiro. Sparse representations for three-dimensional range data restoration. IMA Preprint 2280, University of Minnesota, 2009.

[148] M. Mainberger, A. Bruhn, and J. Weickert. Is dense optic flow useful to compute the fundamental matrix? In *ICIAR*, pages 630–639. Springer Verlag, 2008.

[149] D. Marr. *Vision: a computational investigation into the human representation and processing of visual information*. W. H. Freeman, San Francisco, 1982.

[150] D. Mateus, R. Horaud, D. Knossow, F. Cuzzolin, and E. Boyer. Articulated shape matching using laplacian eigenfunctions and unsupervised point registration. In *Computer Vision and Pattern Recognition*, 2008.

[151] L. Matthies, T. Balch, and B. Wilcox. Fast optical hazard detection for planetary rovers using multiple spot laser triangulation. In *ICRA*, volume 1, pages 859–866. IEEE Press, 1997.

[152] J. Max. Quantizing for minimum distortion. *IRE Transactions on Information Theory*, IT-6(1):7–12, 1960.

[153] R. Mehran, B. E. Moore, and M. Shah. A streakline representation of flow in crowded scenes. In *European Conf. Computer Vision*, 2010.

[154] E. Mémin and P. Pérez. Hierarchical estimation and segmentation of dense motion fields. *International Journal of Computer Vision*, 46(2):129–155, 2002.

[155] K. W. Morton and D. F. Mayers. *Numerical Solution of Partial Differential Equations*. Cambridge University Press, 2005.

[156] A. Müller and Z. Terze. Differential-geometric modelling and dynamic simulation of multibody systems. *Strojarstvo: Journal for Theory and Application in Mechanical Engineering*, 51(6):597–612, 2009.

[157] D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on Pure and Applied Mathematics*, 42(5):577–685, 1989.

[158] R. M. Murray, Z. Li, and S. S. Sastry. *A Mathematical Introduction to Robotic Manipulation*. CRC, 1st edition, March 1994.

[159] A. Myronenko and X. Song. Point set registration: Coherent point drift. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(12):2262–2275, Dec. 2010.

[160] R. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *ISMAR*, pages 127–136, 2011.

[161] R. A. Newcombe, S. Lovegrove, and A. J. Davison. DTAM: Dense tracking and mapping in real-time. In *International Conference on Computer Vision*, pages 2320–2327, 2011.

[162] M. Nicolescu and G. Medioni. A voting-based computational framework for visual motion analysis and interpretation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27:739–752, May 2005.

[163] T. Nir, A. M. Bruckstein, and R. Kimmel. Over-parameterized variational optical flow. *International Journal of Computer Vision*, 76(2):205–216, 2008.

[164] T. Nir, R. Kimmel, and A. M. Bruckstein. Variational approach for joint optic-flow computation and video restoration. Technical Report CIS-05, Technion, 2005.

[165] M. Okutomi and T. Kanade. A multiple-baseline stereo. *IEEE Trans. Pattern Anal. Mach. Intell.*, 15(4):353–363, Apr. 1993.

[166] S. Osher, M. Burger, D. Goldfarb, J. Xu, and W. Yin. An iterative regularization method for total variation-based image restoration. *Simul*, 4:460–489, 2005.

[167] S. Paris and F. Durand. A fast approximation of the bilateral filter using a signal processing approach. In A. Leonardis, H. Bischof, and A. Pinz, editors, *European Conf. Computer Vision*, volume 3951 of *Lecture Notes on Computer Science*, pages 568–580. Springer Verlag, 2006.

[168] F. C. Park, J. E. Bobrow, and S. R. Ploen. A Lie group formulation of robot dynamics. *Int. J. Rob. Res.*, 14:609–618, December 1995.

[169] J. Peng, V. Strela, and D. Zorin. A simple algorithm for surface denoising. In *Proceedings of the conference on Visualization '01*, pages 107–112, Washington, DC, USA, 2001. IEEE Computer Society.

[170] X. Pennec, P. Cachier, and N. Ayache. Understanding the "demon's algorithm": 3D non-rigid registration by gradient descent. In *MICCAI*, pages 597–605, 1999.

[171] X. Pennec, P. Fillard, and N. Ayache. A Riemannian framework for tensor computing. *International Journal of Computer Vision*, 66(1):41–66, 2006.

[172] P. Perona. Orientation diffusions. *IEEE Trans. Image Process.*, 7(3):457–467, 1998.

[173] J.-P. Pons, R. Keriven, O. Faugeras, and G. Hermosillo. Variational stereovision and 3d scene flow estimation with statistical similarity measures. *International Conference on Computer Vision*, 1:597, 2003.

[174] J. Posdamer and M. Altschuler. Surface measurement by space-encoded projected beam systems. *Computer Graphics and Image Processing*, 18(1):1 – 17, 1982.

[175] M. J. Powell. *Optimization*, chapter A method for nonlinear constraints in minimization problems, pages 283–298. Academic Press, 1969.

[176] M. Quigley, S. Batra, S. Gould, E. Klingbeil, Q. Le, A. Wellman, and A. Y. Ng. High-accuracy 3d sensing for mobile manipulation: improving object detection and door opening. In *ICRA*, pages 3604–3610, Piscataway, NJ, USA, 2009. IEEE Press.

[177] I. U. Rahman, I. Drori, V. C. Stodden, D. L. Donoho, and P. Schroeder. Multiscale representations of manifold-valued data. Technical report, Stanford, 2005.

[178] M. Raptis and S. Soatto. Tracklet descriptors for action modeling and video analysis. In *European Conf. Computer Vision*, pages 577–590, Sep. 2010.

[179] D. Raviv, A. Dubrovina, and R. Kimmel. Hierarchical shape matching. In *Scale Space and Variational Methods in Computer Vision*, 2011. Accepted.

[180] G. Rosman, A. M. Bronstein, M. M. Bronstein, and R. Kimmel. Articulated motion segmentation of point clouds by group-valued regularization. In *Eurographics Workshop on 3D Object Retrieval*, pages 77–84, 2012.

[181] G. Rosman, A. M. Bronstein, M. M. Bronstein, X.-C. Tai, and R. Kimmel. Group-valued regularization for analysis of articulated motion. In *NORDIA workshop, ECCV*, pages 52–62, Berlin, Heidelberg, 2012. Springer-Verlag.

[182] G. Rosman, M. M. Bronstein, A. M. Bronstein, A. Wolf, and R. Kimmel. Group-valued regularization framework for motion segmentation of dynamic non-rigid shapes. In *Scale Space and Variational Methods in Computer Vision*, volume 6667 of *Lecture Notes on Computer Science*, pages 725–736, 2011.

[183] G. Rosman, A. Dubrovina, and R. Kimmel. Sparse modeling of shape from structured light. In *3DIMPVT*, pages 456–463, Washington, DC, USA, 2012. IEEE Computer Society.

[184] G. Rosman, A. Dubrovina, and R. Kimmel. Patch-collaborative spectral point-cloud denoising. *Computer Graphics Forum*, 2013.

[185] G. Rosman, S. Shemtov, D. Bitton, T. Nir, G. Adiv, R. Kimmel, A. Feuer, and A. M. Bruckstein. Over-parameterized optical flow using a stereoscopic constraint. In *Scale Space and Variational Methods in Computer Vision*, volume 6667 of *Lecture Notes on Computer Science*, pages 761–772, 2011.

[186] G. Rosman, X.-C. Tai, L. Dascal, and R. Kimmel. Polyakov action for efficient color image processing. In *ECCV workshop on color and reflectance in comp. vision*, 2010.

[187] G. Rosman, Y. Wang, X.-C. Tai, R. Kimmel, and A. M. Bruckstein. Fast regularization of matrix-valued images. Technical Report CAM11-87, UCLA, 2011.

[188] G. Rosman, Y. Wang, X.-C. Tai, R. Kimmel, and A. M. Bruckstein. Fast regularization of matrix-valued images. In *European Conf. Computer Vision*, volume 7574 of *Lecture Notes on Computer Science*, pages 173–186. Springer, 2012.

[189] G. Rosman, Y. Wang, X.-C. Tai, R. Kimmel, and A. M. Bruckstein. Fast regularization of matrix-valued images. In *European Conf. Computer Vision*, pages 173–186, Berlin, Heidelberg, 2012. Springer-Verlag.

[190] S. Roth and M. J. Black. On the spatial statistics of optical flow. *International Journal of Computer Vision*, 74(1):33–50, 2007.

[191] O. Rubinstein, Y. Honen, A. Bronstein, M. Bronstein, and R. Kimmel. 3D-color video camera. In *3DIM*, pages 1505 –1509, 272009-oct.4 2009.

[192] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D Letters*, 60:259–268, 1992.

[193] M. R. Ruggeri, G. Patanè, M. Spagnuolo, and D. Saupe. Spectral-driven isometry-invariant matching of 3D shapes. *International Journal of Computer Vision*, 89(2-3):248–265, 2010.

[194] S. Rusinkiewicz and M. Levoy. Efficient variants of the ICP algorithm. In *Third International Conference on 3D Digital Imaging and Modeling (3DIM)*, June 2001.

[195] C. Sagiv, N. A. Sochen, and R. Kimmel. Stereographic combing a porcupine or studies on direction diffusion in image processing. *SIAM J. Appl. Math.*, 64(5):1477–1508, 2004.

[196] R. Salvador, A. Pena, D. K. Menon., T. Carpenter, J. Pickard, and E. Bullmore. Formal characterization and extension of the linearized diffusion tensor model. *Human brain mapping*, 24(2):144–155, Feb. 2005.

[197] 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.

[198] B. Sapp, A. Toshev, and B. Taskar. Cascaded models for articulated pose estimation. In *European Conf. Computer Vision*, pages 406–420. Springer Verlag, 2010.

[199] K. Sato and S. Inokuchi. Three-dimensional surface measurement by space encoding range imaging. *JRobS*, 2(1):27–39, 1985.

[200] O. Schall, A. Belyaev, and H.-P. Seidel. Adaptive feature-preserving non-local denoising of static and time-varying range data. *Computer-Aided Design*, 40(6):701–707, 2008.

[201] R. Schneider, L. Kobbelt, and H.-P. Seidel. Mathematical methods for curves and surfaces. chapter Improved bi-Laplacian mesh fairing, pages 445–454. Vanderbilt University, Nashville, TN, USA, 2001.

[202] H. Sekkati and A. Mitiche. Concurrent 3D motion segmentation and 3D interpretation of temporal sequences of monocular images. *IEEE Trans. Image Process.*, 15(3):641–653, March 2006.

[203] J. Shah. A common framework for curve evolution, segmentation and anisotropic diffusion. In *Proceedings of the 1996 Conference on Computer Vision and Pattern Recognition (CVPR '96)*, Computer Vision and Pattern Recognition, pages 136–, Washington, DC, USA, 1996. IEEE Computer Society.

[204] A. Shamir. A survey on mesh segmentation techniques. *Computer Graphics Forum*, 27(6):1539–1556, 2008.

[205] S. Shemtov, G. Rosman, G. Adiv, R. Kimmel, and A. M. Bruckstein. On globally optimal local modeling: From moving least squares to over-parametrization. In M. Breuß, A. Bruckstein, and P. Maragos, editors, *Innovations for Shape Analysis*, Mathematics and Visualization, pages 379–405. Springer, 2012.

[206] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-Time human pose recognition in parts from single depth images. June 2011.

[207] N. Slesareva, A. Bruhn, and J. Weickert. Optic flow goes stereo: A variational method for estimating discontinuity-preserving dense disparity maps. In *DAGM Symposium*, volume 3663, pages 33–40. Springer Verlag, 2005.

[208] S. M. Smith and J. Brady. SUSAN – A new approach to low level image processing. *International Journal of Computer Vision*, 23:45–78, 1997.

[209] N. Sochen, R. Kimmel, and R. Maladi. From high energy physics to low level vision. In B. M. ter Haar Romeny, L. Florack, J. J. Koenderink, and M. A. Viergever, editors, *International Conference on Scale Space Theory in Computer Vision*, volume 1252 of *Lecture Notes on Computer Science*, pages 236–247, Utrecht, The Netherlands, July 1997. Springer Verlag.

[210] A. Spira and R. Kimmel. Geometric curve flows on parametric manifolds. *J. Comput. Phys.*, 223:235–249, April 2007.

[211] G. Steidl, S. Setzer, B. Popilka, and B. Burgeth. Restoration of matrix fields by second-order cone programming. *Computing*, 81(2-3):161–178, 2007.

[212] E. O. Stejskal and J. E. Tanner. Spin diffusion measurements: Spin echoes in the presence of a time-dependent field gradient. *Journal of Chemical Physics*, 42:288–292, 1965.

[213] J. Stillwell. *Naive Lie Theory*. Undergraduate texts in mathematics. Springer-Verlag New York, 2008.

[214] J. Stuehmer, S. Gumhold, and D. Cremers. Real-Time Dense Geometry from a Handheld Camera. In *Pattern Recognition (Proc. DAGM)*, pages 11–20, Darmstadt, Germany, Sept. 2010.

[215] R. Subbarao and P. Meer. Nonlinear mean shift over Riemannian manifolds. *International Journal of Computer Vision*, 84(1):1–20, 2009.

[216] R. W. Sumner and J. Popović. Deformation transfer for triangle meshes. In *SIGGRAPH*, pages 399–405, New York, NY, USA, 2004. ACM.

[217] D. Sun, S. Roth, and M. J. Black. Secrets of optical flow estimation and their principles. In *Computer Vision and Pattern Recognition*, pages 2432–2439, 2010.

[218] Y. Swirski, Y. Y. Schechner, and T. Nir. Variational stereo in dynamic illumination. In *International Conference on Computer Vision*, pages 1124–1131, Washington, DC, USA, 2011. IEEE Computer Society.

[219] X.-C. Tai and C. Wu. Augmented Lagrangian method, dual methods and split Bregman iteration for ROF model. In *Scale Space and Variational Methods in Computer Vision*, pages 502–513, 2009.

[220] B. Tang, G. Sapiro, and V. Caselles. Diffusion of general data on non-flat manifolds viaharmonic maps theory: The direction diffusion case. *International Journal of Computer Vision*, 36:149–161, February 2000.

[221] J. P. Tardif and S. Roy. A MRF formulation for coded structured light. In *3DIM*, pages 22–29, Washington, DC, USA, 2005. IEEE Computer Society.

[222] G. Taubin. A signal processing approach to fair surface design. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, SIGGRAPH '95, pages 351–358, New York, NY, USA, 1995. ACM.

[223] J. P. Thirion. Image matching as a diffusion process: an analogy with Maxwell's demons. *Medical Image Analysis*, 2(3):243–260, Sept. 1998.

[224] J. Tierny, J.-P. Vandeborre, and M. Daoudi. Fast and precise kinematic skeleton extraction of 3D dynamic meshes. In *ICPR*, pages 1–4, 2008.

[225] A. Tikhonov. On the stability of inverse problems. *Doklady Akademii nauk SSSR*, 39(5):195–198, 1943.

[226] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. *International Conference on Computer Vision*, pages 836–846, 1998.

[227] I. Tosic, B. A. Olshausen, and B. J. Culpepper. Learning sparse representations of depth. *CoRR*, abs/1011.6656, 2010.

[228] W. Trobin, T. Pock, D. Cremers, and H. Bischof. An unbiased second-order prior for high-accuracy motion estimation. In *DAGM Symposium*, pages 396–405. Springer Verlag, 2008.

[229] D. Tschumperlé and R. Deriche. Vector-valued image regularization with pdes: A common framework for different applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27:506–517, 2005.

[230] P. Tseng. Coordinate ascent for maximizing nondifferentiable concave functions. LIDS-P 1940, MIT, 1988.

[231] O. Tuzel, F. Porikli, and P. Meer. Learning on lie groups for invariant detection and tracking. In *Computer Vision and Pattern Recognition*, 2008.

[232] L. Valgaerts, A. Bruhn, and J. Weickert. A variational model for the joint recovery of the fundamental matrix and the optical flow. In *DAGM Symposium*, pages 314–324. Springer Verlag, 2008.

[233] B. C. Vemuri, Y. Chen, M. Rao, T. McGraw, Z. Wang, and T. Mareci. Fiber tract mapping from diffusion tensor MRI. In *Proceedings of the International Conference on Variational, Geometry and Level Sets Methods in Computer Vision*, pages 81–88. IEEE Computer Society, 2001.

[234] M. Žefran, V. Kumar, and C. Croke. On the generation of smooth three-dimensional rigid body motions. *Robotics and Automation, IEEE Transactions on*, 14(4):576 –589, Aug. 1998.

[235] M. Žefran, V. Kumar, and C. Croke. Metrics and connections for rigid-body kinematics. *I. J. Robotic Res.*, 18(2):242, 1999.

[236] S. Wang, Y. Wang, M. Jin, X. D. Gu, and D. Samaras. Conformal geometry and its applications on 3D shape matching, recognition, and stitching. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(7):1209–1220, 2007.

[237] A. Wedel, T. Pock, J. Braun, U. Franke, and D. Cremers. Duality TV-L1 flow with fundamental matrix prior. *Image Vision and Computing New Zealand*, pages 1–6, 2008.

[238] J. Weickert. *Anisotropic Diffusion in Image Processing*. Teubner-Verlag, 1998.

[239] J. Weickert and T. Brox. Diffusion and regularization of vector- and matrix-valued images. volume 313 of *Inverse problems, image analysis, and medical imaging*, 2002.

[240] N. Weiner. The extrapolation, interpretation and smoothing of stationary time series with engineering applications. Technical Report DIC Contract 6037, MIT, 1942.

[241] Z. Wen, D. Goldfarb, and W. Yin. Alternating direction augmented Lagrangian methods for semidefinite programming. CAAM TR09-42, Rice university, 2009.

[242] A. P. Witkin. Scale-space filtering. In *Proceedings of the 8th International Joint Conference on Artificial Intelligence*, volume 2, pages 1019–1022, Karlsruhe, August 1983. William Kaufmann.

[243] A. Wolf, I. Sharf, and M. B. Rubin. Using Cosserat point theory for estimating kinematics and soft-tissue deformation during gait analysis. In *Advances in Robot Kinematics: Motion in Man and Machine*, pages 63–70. Springer Netherlands, 2010.

[244] C. Wu and X.-C. Tai. Augmented lagrangian method, dual methods, and split bregman iteration for ROF, vectorial TV, and high order models. *SIAM J. Img. Sci.*, 3:300–339, July 2010.

[245] C. Wu and X.-C. Tai. Augmented lagrangian method, dual methods, and split Bregman iteration for ROF, vectorial TV, and high order models. *SIAM J. Imaging Sciences*, 3(3):300–339, 2010.

[246] C. Wu, J. Zhang, and X. Tai. Augmented lagrangian method for total variation restoration with non-quadratic fidelity. *Inverse Problems and Imaging*, 5(1):237–261, 2011.

[247] S. Wuhrer and A. Brunton. Segmenting animated objects into near-rigid components. *The Visual Computer*, 26:147–155, 2010.

[248] T. Yamasaki and K. Aizawa. Motion segmentation for time-varying mesh sequences based on spherical registration. *EURASIP Journal on Applied Signal Processing*, 2009.

[249] N. N. Yanenko. *The method of fractional steps: solution of problems of mathematical physics in several variables*. Springer Verlag, 1971. Translated from Russian.

[250] A. Y. Yang, A. Ganesh, Z. Zhou, S. Sastry, and Y. Ma. A review of fast l1-minimization algorithms for robust face recognition. *CoRR*, abs/1007.3753, 2010.

[251] R. Yang, G. Welch, and G. Bishop. Real-time consensus-based scene reconstruction using commodity graphics hardware. In *Pacific Conf. on Comp. Graphics and Applications*, PG '02, pages 225–, Washington, DC, USA, 2002. IEEE Computer Society.

[252] Z. Yang and D. Purves. A statistical explanation of visual space. *Nat. Neuroscience*, 6(6):632–40, June 2003.

[253] L. P. Yaroslavsky. *Digital Picture Processing*. Springer Verlag New York, Inc., Secaucus, NJ, USA, 1985.

[254] W. Yin and S. Osher. Error forgetting of Bregman iteration. *J. Sci. Comput.*, 54(2-3):684–695, Feb. 2013.

[255] S. Yoshizawa, A. Belyaev, and H. P. Seidel. Smoothing by example: mesh denoising by averaging with similarity-based weights. *Proceedings of International Conference on Shape Modelling and Applications*, pages 38–44, 2006.

[256] G. Yu, G. Sapiro, and S. Mallat. Solving inverse problems with piecewise linear estimators: From Gaussian mixture models to structured sparsity. *CoRR*, abs/1006.3056, 2010.

[257] S. D. Zenzo. A note on the gradient of a multi-image. *Computer Vision, Graphics, and Image Processing*, 33(1):116–125, 1986.

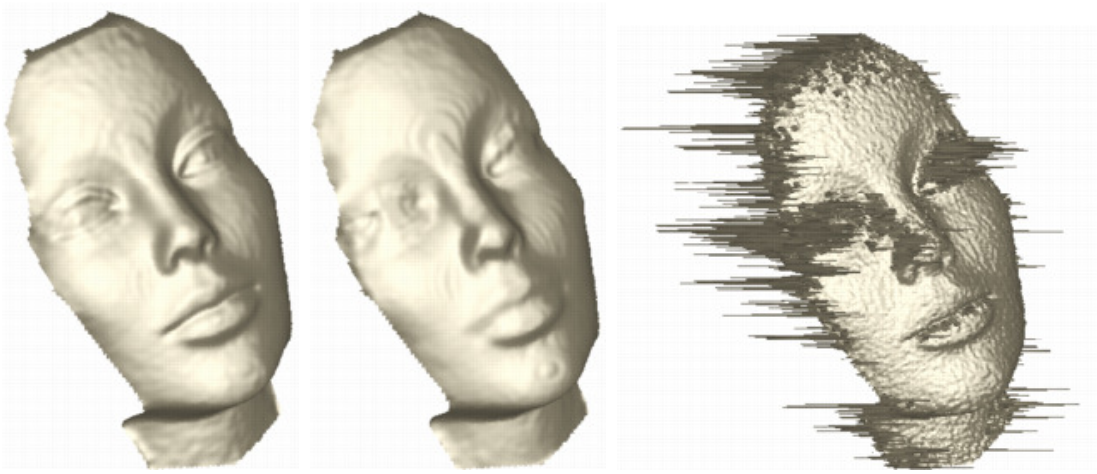
[258] D. Zhang and M. Hebert. Harmonic maps and their applications in surface matching. In *Computer Vision and Pattern Recognition*, volume 2, pages 524–530, 1999.

[259] L. Zhang, B. Curless, and S. Seitz. Rapid shape acquisition using color structured light and multi-pass dynamic programming. In *3DPVT*, pages 24–36. IEEE, 2002.

[260] D. Zoran and Y. Weiss. From learning models of natural image patches to whole image restoration. In *International Conference on Computer Vision*, pages 479–486, 2011.

בפרויקט שביצענו, ואילו הפריורים מבוססי הדלילות מגיעים לתוצאות משמעותית יותר טובות משיטות אחרות בגלל התמך הרחב שלהם ודיוק המודל. אנו מגדירים כיצד ניתן ללמוד מילונים דלילים לתמונות עומק, המאופיינות על ידי עיוותים השונים מהותית ממודל רעש לבן גאוסי אדיטיבי שמשמש בדרך כלל במידול אלגוריתמים כמו KSVD, ועל ידי הטיה חזקה של פילוג ה-patches בתמונות, דבר שמקשה על אלגוריתמים סטנדרטיים לבניית מילון.

השחזור שאנו מבצעים מודגם בשני סטים שונים של תבניות אור מקודד, שמשמשות בסורק התלת-מימד שבשימוש במעבדתנו. חלק מהניסוחים ניתנים למימוש מקבילי יעיל על כרטיסים גרפיים בזמן אמת, כפי שהדגמנו בפרויקט שבוצע במעבדה שלנו. התוצאות מראות על שיפור בתוצאות הסריקה, שנעשית חסינה לרעשי תמונה וחשיפה נמוכה של החיישנים, כמו גם שיחזור של משטחים עם מקדם החזר נמוך, תופעות לא-למברטיות, וקלקולים בשחזור שנובעים מתזוזת האוביקט הנסרק במהלך הקרנת התמונות.

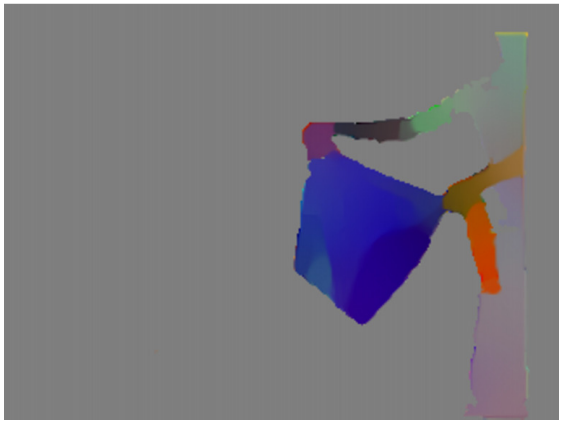


למקוטעין בתור החלקות משמרות שפה, של המיפוי מהמשטח אל חבורת לי של מטריצות סיבוב והזזה, $SE(3)$.

בתחילת חלק זה אנו מדגימים חלקות מסוג Mumford-Shah על שדה התנועה, שמשלבת סגמנטציה לפי הטרנספורמציה הקשיחה המקורבת בכל נקודה, בצורה סתומה וללא אילוץ של חלוקה מוגדרת, ביחד עם התאמת משטחים בדומה לתהליך ICP. אנו מגדירים דיסקרטיזציה לתהליך החלקה זה הן על משטחים משולשיים לפי נוסחת משקלות קוטנגנס, והן על ענני נקודות ותמונות עומק, לפי הסכמה של בלקין וסון ללאפלסיאן. אנו מראים כיצד בעזרת סכמת Ambrosio-Tortorelli ניתן להתאים את פונקציונל אנרגית Dirichlet המוכללת לפונקציונל המעודד חלקות למקוטעין. החלקה זאת ממומשת בסכמה מפורטת של תהליך זרימה שממזער את הפונקציונל המתאים, הכולל גורם חלקות וגורם התאמה לנתונים. האלגוריתמים שמתקבלים משיגים תוצאות דומות לכלים יעודיים מהתחום על משטחים סטנדרטיים מתחום עיבוד המשטחים, וממידע תלת-מימדי ממצלמות עומק.

בהמשך החלק השני אנו מדגימים חלקות מסוג Total Variation של התמרות קשיחות על גבי תמונות עומק. בהינתן וידאו של תמונות עומק אנו מגדירים כיצד ניתן לקבל שיערוך של ההתמרה המקומית בין שתי תמונות עומק עוקבות בכל נקודה, ולמעשה ליצור מפה עם ערך מטריצי אותו אנו מנסים להחליק בצורה משמרת שפות לצורך סגמנטציה. הבעיה מנוסחת בתור בעית אופטימיזציה עם אילוצים שנאכפים בשיטת Augmented Lagrangian. אחד מהאלגוריתמים ניתן למימוש מקבילי בעזרת הניסוח שבנינו. אלגוריתם זה, במימוש על גבי יחידה לחישוב מקבילי (GPU) מאפשר להשיג זמן חישוב מתאים לזמן-אמת. בנוסף, האלגוריתם מבוסס Total Variation מגדיר מסגרת שכוללת מספר בעיות היפוך אחרות. עבור מטריצות חיוביות אנו מראים אפליקציות בתחום הדימות הרפואי כגון ניקוי תמונות Diffusion Tensor MRI, ושיחזור של תמונות DT-MRI מתוך סט של תמונות MRI. יישום נוסף הוא החלקת שדה כיוונים המשוערכים מתוך תמונות של טביעות אצבע.

בנוסף אנו מוכיחים בחלק זה את ההתכנסות של התהליך באיטרציות הפנימיות של סכמת ה- Augmented Lagrangian. למרות שהן כוללות אופרטורים לא רציפים, ניתן לשנות במעט את התהליך לקבלת סכמה עם הבטחת התכנסות והוכחה שמתבסס על הקשר בין המרחקים בין האיטרנטים וערך הפונקציה. כמו כן אנו מראים את ההתכנסות הגלובלית במקרים שמרחב הפרמטריזציה קמור, כמו לדוגמא במקרה של –DT-MRI, ומטריציות חיוביות. לאחר מכן אנו מכלילים את הפונקציונל לכדי חלקות מסדר שני, דבר המאפשר להמנע מתופעת מדרגות (staircasing) שמאפיינת סכמות Total Variation.



בחלק השלישי והאחרון של התזה אנו מדגימים כיצד שחזור תלת-מימד מאור מקודד ניתן לניסוח בתור מקסימיזציה של הסתברות המודל התלת-מימדי בהינתן תמונות המצלמה והמקרן, לפי מודל תאורה למברטי. רכיבי התאורה משוערכים בכל נקודה כתלות בעומק, ולמעשה מקבלים מקסימום על פני רכיבי התאורה והעומק. ניסוח הסתברותי זה מאפשר לנו לשלב פריורים לצורת המשטח לתוך תהליך החישה הלא-לינארי. למרות שהבעיה לא לינארית, ניתן להראות התכנסות מקומית טובה מעבר לדיוק של רזולוציות המקרן והמצלמה, ומתוך איתחול סביר ופשוט.

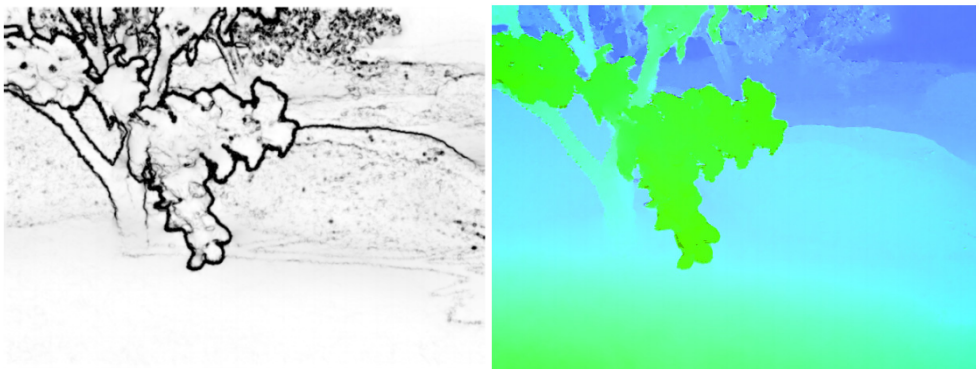
הפריורים שאנו משתמשים בהן, שנלמדו מנתוני עומק, מתגלים כבעלי מבנה טבעי ואינטואיטיבי שמתאר שפות ופינות בתמונת העומק. אנו מדגימים בחלק זה כ-4 פריורים שונים לתמונות רעש – שני פריורים ווריאציונים (Total Variation, Second-order TV), ושני פריורים מבוססי דלילות (Shrinkage L1, Structured Sparsity/GMM). הפריורים הווריאציונים ניתנים למימוש בזמן אמת על גבי כרטיס גרפים, כפי שהוגדרים

<u>תקציר בעברית</u>

שאלה בסיסית וחשובה המופיעה כאשר ניגשים לפתרון בעיה היא הייצוג, או הפרמטריזציה, של הפתרון לבעיה. ייצוג מתאים לפתרון מאפשר לעיתים פתרון יותר קל, או אף נכון במובנים מסוימים, לבעיה. לתובנה בסיסית זאת מגוון מופעים בעיבוד תמונה, בראייה ממוחשבת, ובבעיות היפוך בתחומים הנ"ל. בין הפרמטריזציות הגלובליות שמשמשות בתחומים אלה אנו כוללים את התמרת Hough, התמרות ספקטרליות של אותות כמו התמרת פורייה, וייצוגי רב-שריג כמו התמרות wavelets, או ההתמרה למרחב צבע-מיקום שמשמשת כבסיס למסננים בילטרליים מהירים לעיבוד תמונה, ולמסנני בלטרמי. פרמטריזציות מקומיות של אותות כוללות מודלים וייצוגים דלילים לתמונות, וייצוגים ייתירים, המשמשים לשיחזור, ניקוי, אינטרפולציה, השלמה והבנת תמונות.

מחקר הזה עוסק בייצוגים חשובים המופיעים במקרים שונים של אנליזת תנועה בדו-מימד ובתלת-מימד, ובשחזור תלת-מימד, שניתן לקשור אותו ישירות להבנת תנועה בין מספר מצלמות. מקרים אלו הם מקרים בהם יש חשיבות רבה לבחירת הפרמטריזציה, ולהגדרה של מרחקים בתוך מרחב הפרמטריזציה. הגדרת הפרמטריזציה משפיעה ומושפעת מבחירת הרגולרציה לאותה בעיה, דבר העומד בבסיס של שיטות יתירות למידול. בחירות מתאימות של פרמטריזציה ורגולריזציה יכולות במקרים אלה להוביל לניסוחים פשוטים, אקסיומטיים ויעילים לחישוב פתרונות לבעיות אלה. ניסוחים אלה יכולים במקרים מסוימים לקשר את תהליך החישה לתהליכים גבוהים יותר של הבנת התמונה וזיהוי אוביקטים בסצינה. במהלך הדוקטורט חקרנו מספר ייצוגים המשמשים בבעיות אלה, וניסחנו אלגוריתמים לפי המודלים שמשתמשים בייצוגים האלה, כולל סכמות נומריות יעילות לבעיות אלה.

בחלק הראשון של העבודה אנו עוסקים בראייה סטריאוסקופית ושטף אופטי בין זוג תמונות של אותה סצנה. בחלק זה אנו מציעים להשתמש במשוואת המישור בתלת-מימד ובהומוגרפיות שמוגדרות על ידה בתור ייצוג יתיר לשטף האופטי, כהמשך לא לינארי לייצוג היתיר שהוצג בעבודתו על שטף אופטי של טל ניר ושותפיו. הפרמטריזציה הלא-לינארית דורשת הוספת איטרציה פנימית נוספת למבנה הסטנדרטי של פתרון בעיות שטף אופטי לצורך לינאריזציה של מודל התנועה. בחירה מקומית של פרמטר עם שלוש דרגות חופש מגדירה את משוואת המישור וההומוגרפיה שפועלת בנקודה בין שתי התמונות. בהתאם, סצינה שמקורבת על ידי חלוקה למישורים תתואר על ידי שדה פרמטרים עם מעט שינויים חדים, ורובה תהיה חלקה או קבועה. האלגוריתם שמתקבל משיג תוצאות מהשורה הראשונה במונחים של דיוק של השטף האופטי המחושב, כפי שנבדקו ב-Middlebury Dataset ועל ה-Yosemite sequence. לגורם הרגולריזציה יש במקרה שלנו פירוש גיאומטרי-פיזיקלי פשוט שנובע ממודל Mumford-Shah והפרמטריזציה שבחרנו, והוא מקשר בין ראייה ממוחשבת ברמה נמוכה של חישוב שטף אופטי ובין רמות הבנה כלליות יותר של מידול והבנת הסביבה, כמו גם סגמנטציה של הסביבה למשטחים מישוריים.



החלק השני של התזה מתייחס להבנת תנועה בתלת-מימד ובפרט סגמנטציה של תנועה קשיחה למקוטעין – זיהוי התנועה והחלוקה שלה למקטעים שנעים בצורה קשיחה. בחלק זה אנו מנסחים מחדש הבנת תנועה קשיחה

על פרמטריזציות טבעיות של מרחבי תנועה ומבנה

חיבור על מחקר

לשם מילוי חלקי של הדרישות לקבלת התואר

דוקטור לפילוסופיה

גיא רוסמן

הוגש לסנט הטכניון – מכון טכנולוגי לישראל

אלול תשע''ג חיפה אוגוסט 2013