DISSERTATION

submitted to the
Combined Faculty for the Natural Sciences and Mathematics
of
Heidelberg University, Germany
for the degree of
Doctor of Natural Sciences

Put forward by

Diplom-Mathematiker Johannes Peter Berger
born in Bad Dürkheim / Weinstraße, Germany

Second Order Minimum Energy Filtering

of Joint Variational Camera Motion
and Depth Map Reconstructions

Johannes Berger

Advisors: Prof. Dr. Christoph Schnörr
Prof. Dr. Rainer Dahlhaus

Date of the oral examination:

Abstract

Camera motion estimation and dense scene reconstruction are essential for modern computer vision applications such as autonomous driving, robot navigation and virtual reality. State-of-the-art methods are usually based on stereo camera systems that use the information about the distance between the two cameras to uniquely estimate the depth map. However, these systems need to be calibrated and are too expensive for some special industrial applications. Thus, we focus in this work on monocular camera systems that consist of a single moving camera. To increase the robustness of the method we use temporal information in terms of filters. These use temporal consistency to improve the accuracy of the estimation of the current state of a system, e.g. the unknown camera motion or the depth map. Instead of using established stochastic filters such as extended Kalman filters, unscented Kalman filters or particle filters we use novel minimum energy filters that do not base on a stochastic model but on the minimization of an energy function.

In a first step we derive the minimum energy filter and provide differential equations for the optimal state and the corresponding second-order operator. We demonstrate that this filter is as exact as state-of-the-art stochastic filters for most problems and, in addition to it, superior in more involved scenarios. Then we consider a simple geometric setting for the reconstruction of the camera motion within a static scene based on stereo image data. There we formulate a non-linear filtering problem on the special Euclidean group based on non-linear observations of optical flow and depth map. In experiments we show that the underlying camera motion can be reconstructed with minimum energy filters as accurate as in other state-of-the-art stereo methods. Finally, we present an approach for the *joint* reconstruction of camera motion and disparity map (inverse of depth map) in a monocular approach by means of minimum energy filters. By introducing a novel disparity group we can derive the filter without additional constraints or barrier functions. Further we generalize the used energy function to a Charbonnier penalty function which is robuster against outliers in the optical flow. We also demonstrate that additional regularizers can be easily integrated within the overall filtering problem providing a rich basis for many applications. From the mathematical point of view we solve by means of *minimum energy filters* a *non-linear* filtering problem on a *Lie group* for a *high dimensional* problem – thus a problem which is infeasible for most stochastic filter.

Zusammenfassung

Die Schätzung der Kamerabewegung und dichte Szenen-Rekonstrution sind wesentlich für Anwendungen im Bereich des maschinellen Sehens, wie etwa autonomes Fahren, Roboter-Navigation und virtuelle Realität. Aktuelle Methoden basieren normalerweise auf Stereo-Systemen welche die Information über die Entfernung der Kameras nutzen, um Tiefenkarten eindeutig zu schätzen. Allerdings müssen diese Systeme kalibriert werden und sind zu teuer für spezielle industrielle Anwendungen. Daher fokussieren wir uns in dieser Arbeit auf monokulare Kamera-Systeme, bestehend aus nur einer sich bewegenden Kamera. Um die Robustheit der Methode zu erhöhen, verwenden wir zeitliche Informationen mithilfe von Filtern. Diese nutzen zeitliche Konsistenz, um die Genauigkeit der Schätzung des aktuellen Zustands des Systems zu erhöhen, z.B. die unbekannte Kamerabewegung oder die Tiefenkarte. Anstatt etablierter stochastischer Filter, nutzen wir neue minimale Energie Filter, welche nicht auf einem stochastischen Modell basieren, sondern auf der Minimierung einer Energie Funktion.
In einem ersten Schritt leiten wir den Minimale Energie Filter her, einschließlich der Differentialgleichungen für den optimalen Zustand und dem entsprechenden Operator zweiter Ordnung. Experimente zeigen, dass dieser Filter für die meisten Problemstellungen genauso exakt ist wie stochastische Filter des Standes der Forschung und darüber hinaus in komplizierteren Szenarien sogar überlagen ist. Danach betrachten wir eine einfache geometrische Situation für die Rekonstruktion der Kamerabewegung in einer statischen Szene. Wir formulieren ein nichtlineares Filter-Problem auf der speziellen Euklidischen Gruppe, welches auf nichtlineare Beobachtungen vom optischen Fluss und der Tiefenkarte basiert. In Experimenten zeigen wir, dass wir die zugrunde liegende Kamerabewegung mindestens so genau schätzen können wie alternative Methoden des Standes der Forschung. Zum Schluss präsentieren wir einen Ansatz für die *gemeinsame* Rekonstruktion der Kamerabewegung und der Disparitäten-Karte (Inverse der Tiefen) in einem monokularen Ansatz mit Hilfe von Minimale Energie Filtern. Durch Einführung einer neuen Disparitäten-Gruppe können wir den Filter ohne zusätzliche Nebenbedingungen oder Barriere-Funktionen herleiten. Ferner verallgemeinern wir die Energie Funktion zu einer Charbonnier Straf-Funktion welche robuster gegenüber Ausreißern im optischen Fluss sind. Wir zeigen auch, dass sich zusätzliche Regularisierer leicht in das gesamte Filterungs-Problem einbetten lassen, was eine ergiebige Basis für viele Anwendungen ergibt. Vom mathematischen Standpunkt aus betrachtet, lösen wir mithilfe *minimaler Energie Filter* ein *nichtlineares* Filter-Problem auf einer *Lie Gruppe* für ein *hochdimensionales* Problem – also ein Problem das für die meisten stochastischen Filter unlösbar ist.

Acknowledgements

First of all, I would like to thank my advisor Christoph Schnörr for the supervision of my PhD project. He inspired this work and gave me a lot of background knowledge in the lectures and classes that he prepared meticulously. He was always answering immediately to my concerns and questions, helping to bring this work to success. Second, I thank Rainer Dahlhaus for being prepared to be part of my examination board. Additionally, I would like to thank the German Research Foundation (DFG, GRK 1563) for the funding of my PhD.
Special thanks go to my colleagues Florian Becker, Frank Lenzen, Andreas Neufeld who published some articles with me. They gave a lot of inspiration on how to improve these publications.
For the detailed proofread of this work, I thank Niko Krasowski.
I want to thank all my fellows for the wonderful time, the experiences and discussions that we shared together in Heidelberg and in the whole world, on conferences and workshops. It was a very nice time with them. Particularly, I thank Andreea Deniţiu for being a nice roommate and for her social commitment within the whole group.
Sincere thanks are given to my family – my mother Beatrice, my father Armin, my sister Charlotte and my brother Thomas – for the assistance and the support during my whole life and especially during my study and the PhD.
Finally, I would like to thank my wife Steffi for her assistance in difficult times when my work got stuck, for the celebration of successful moments when my work improved steadily as well as for her love and the beautiful time we've been spending together.

Contents

Chapter 1

Introduction

1.1 Overview and Motivation

Accurate camera motion estimation and three dimensional scene reconstruction from observed image or video data is a fundamental building block of computer vision. There are broad applications in many areas such as autonomous driving, robot navigation, 3D reconstruction of large outdoor objects as well as small scale medical applications e.g. manufacturing of dental crowns.

Reconstruction methods that use multiple cameras in parallel gain precise scene representation. First applications which use these methods are already available on the market. However, multiple camera systems generate a big amount of data that need to be processed. The relative position of one camera to the others has to be known precisely such that these methods require a good calibration before they can be used. On the other side industrial applications are requiring cheap sensors and prefer systems with as little hardware costs as possible. Therefore in the last decade reconstruction methods that require only a single camera (also known as *monocular methods*) gained center stage. Monocular methods require less calibration effort but have the drawback that the monocular reconstruction is ill posed because of the unknown scene scale. It cannot be reconstructed uniquely without additional information about invariants within the scene. Furthermore, monocular methods have a less beneficial motion parallax compared to stereo methods. Through this fact, state-of-the-art methods often use temporal consistency assumptions on monocular camera systems. By recording consecutive frames they increase accuracy and robustness. One can distinguish between offline methods such as *bundle adjustment,* which require a full video, and online methods that can (in principle) process the data in

real time, e.g. sliding window methods and *filtering methods.*
In this work we focus on filtering methods for the problem of camera motion estimation and monocular scene reconstruction. Unlike sliding window approaches, stochastic filters provide a broad mathematical framework and are well understood within the last half century outbound the seminal work of Rudolf Kálmán in 1960. This so called *Kálmán filter* was frequently used for many important engineering tasks in the past, such as for the Apollo 13 mission to the moon, in the navigation of satellites, aircraft steering and many other tasks in sciences and economy down to the present day. Mathematically spoken, Kálmán found a solution to the following problem: Given observations of a system at different time points t, e.g. radar signals of a plane, which we denote by y_t, we want to recover the most likely position of the plane $x(t)$. More exactly, we want to determine the whole *a posteriori* probability distribution of the state process, denoted by π_t which is given through

$$\pi_t(A) := \mathbb{P}(x_t \in A | y_s, s \leq t)\,. \tag{1.1}$$

We assume that the movement of the plane can be described by a mathematical equation, that describes the temporal change rate of the position of the aircraft. This can be expressed by a continuous function f that depends on the current position $x(t)$. This behavior can be written as ordinary differential equation $\dot{x}(t) = f(x(t))$. We allow the state to differ from this system by introducing a noise term $\delta(t)$ and obtain the following equation

$$\dot{x}(t) = f(x(t)) + \delta(t)\,, \quad x(t_0) = x_0\,. \tag{1.2}$$

Here we also demand the existence of an initial state x_0 in order to obtain a well defined solution. The observations depend on the unknown state $x(t)$ by a function $h(x(t))$ and some measurement noise $\epsilon(t)$. We can write this equation as follows:

$$y(t) = h(x(t)) + \epsilon(t)\,. \tag{1.3}$$

Kálmán searched for a solution of the following problem: Given observations $y(s)$, $s \leq t$ up to a time t, what is the most likely state $x^*(t)$ that fits to these observations? Mathematically spoken, we want to find the following expression, also known as *maximum* a posteriori (MAP) estimate, i.e.

$$x^*(t) = \arg\max_{x(t)} \pi_t(x)\,. \tag{1.4}$$

For the case of linear functions f and h and Gaussian noise processes $\delta(t)$ and $\epsilon(t)$, Kálmán found an exact solution for this problems by minimizing the least squares error on the residuals. The resulting *a posteriori* distribution is

again normally distributed (Gaussian), and the generating first and second-order moment can be calculated explicitly. Due to the great success of the *Kálmán filter* this approach was extended in the following decade to plenty of problems in different areas of sciences and engineering. Many modern problems require a solution for the filtering problem for scenarios of non-linear dependencies between that state and the observations. However, for the general filtering problem with non-linear functions f and h it was shown that no explicit finite dimensional solution for the *a posteriori* distribution π_t exists [63]. Besides, the demand for the error terms $\epsilon(t)$ and $\delta(t)$ to be normally distributed does often not mirror the correct description of the considered physical system.
Therefore a couple of strategies have been developed to tackle the non-linearities in the filtering approach: By using linear approximations the *extended Kálmán Filter* is relatively similar to the classical (linear) Kálmán filter. *Unscented Kálmán filters* sample the mean and variance of the *a posteriori* distribution by using *sigma-points* and the *unscented transformation*. *Particle filters* find a Monte-Carlo-approximation of the *a posteriori* distribution by using sampling methods and variance reduction strategies. Many other engineering tricks (e.g. coordinate transform) were adapted to specific problems and led to convincing results.
Today's challenges in image processing, such as the reconstruction of three dimensional structure of a scene or the estimation of camera motion often require accurate methods to find suitable solutions for the filtering problem to cope with such problems as follows:

- non-linearities between the observations (e.g. image features) and the latent variables (camera motion, depth map) on state space (through a projective camera),
- non-Euclidean geometry of the state space (rotation of camera motion, positiveness constraints of depth maps),
- high-dimensional state space (each pixel of an image correspond to a dimension),
- higher-order differential equations (e.g. separation of the kinematics of the camera motion into different orders),
- and outliers or ambiguous data points within the observations.

In the last years modern filters have been established that support a broader class of *a posteriori* distributions, e.g. particle filters, that are very successful

in image processing. They are not limited to the Gaussian case but unfortunately these methods require a careful design and sophisticated variance reducing techniques to gain good results. Other methods try to find a good (finite dimensional) approximation of the true *a posteriori* distribution that is given through the *Kushner-Stratonovich equation* by numerical considerations or projection onto a suitable set of exponential families [18]. A good overview about these non-linear methods can be found in [24]. Due to this variety of non-linear filtering methods, many problems can be solved today with a high accuracy. For problems containing a non-trivial geometry, as appearing in modern applications (e.g. robitocs), non-linear filters were generalized for special Riemannian manifolds and Lie groups. As an example we mention particle filters on SE_3 [53] and extended Kálmán filters on Lie groups [17] for e.g. application to monocular SLAM (Simultaneous Localization And Mapping).
Although for each *single* problem can be solved with methods, it becomes apparent that solving *all* problems at once seems to be involved for most existing methods. The non-linear problem which we want to solve requires a filter that copes with a curved geometry of Lie groups and with high dimensional state spaces. Moreover, it would be beneficial to use an alternative to the least square error minimization that is used within extended Kálmán filters. Least squares minimization tends to be sensitive to outliers within the data, and does often not lead to the desired results in image processing applications.
As a consequence we will take another path in this work: Instead of using a probabilistic interpretation of the filtering problem we will investigate the same problem from the point of view of optimal control following the work of Mortensen [58]. This results in an optimal control problem in which a certain energy function is minimized that penalizes deviations from the model of the state (1.2) and the observation (1.3). Interpreting the model noise $\delta(t)$ as the control variable of the system one can recursively approximate the optimal state $x^*(t)$ of (1.2) by a *second-order Minimum Energy Filter* using *Hamiltonian dynamics* and the well known *Hamilton-Jacobi-Bellman* equation.

1.2 Related Work

Incorporation of temporal context – in terms of (partial) differential equations – into the estimation of latent variables has a long tradition in many common applications, e.g. robotics, aviation and astronautics. Starting from the seminal work of Kálmán [49] considering Gaussian noise and linear filtering

equations, stochastic filters had have great success in many important areas of mathematics, computer sciences and engineering during the last fifty years. The filtering methods have been improved during the last decades to cope with non-linearities of state and observation equations, such as extended Kálmán filters [46], unscented Kálmán filters [47] and particle filters [4]. For a detailed overview of these methods we refer to [7, 24].
However, one strong limitation of stochastic filters represents the fact that the a posteriori distribution is usually unknown and, in general, is infinite dimensional due to the non-linear dependencies. To cover a large bandwidth of *a posteriori* distributions Brigo et al. approximated these by distributions of the exponential family [18]. In contrast, particle filters try to sample from these [4]. Extended and unscented Kálmán filters, on the other hand, only allow distributions that are Gaussian.
Although these methods work successfully for many real-valued problems, they cannot be easily transferred to filtering problems which are constrained to manifolds, appearing in many modern engineering and robotic applications. Therefore, in the last decade, several strategies have been developed to adapt classical unconstrained filters to filtering problems on specific Lie groups and Riemannian manifolds: Kálmán filters were transferred to the manifold of symmetric positive definite matrices [81]. Extended Kálmán filters on SO_3 [56] with symmetry preserving observers [15] were elaborated. Particle filters on SO_3 and SE_3 were proposed in [53] as well on Stiefel [79] and on Grassman manifolds [68]. An application of particle filters to monocular SLAM is reported in [54].
Recently, unscented Kálmán filters were generalized to Riemannian manifolds [42]. Since then, extended Kálmán filters for constrained model and observation equations were developed [17] for general Lie groups based on the idea of the Bayesian fusion [86].
However, although stochastic filters have been adapted to curved spaces and non-linear measurement equations, they still require assumptions about the a posteriori distributions, e.g. to be Gaussian. Furthermore, while transferring related concepts of probability theory and stochastic analysis to Riemannian manifolds is mathematically feasible [45, 21, 20], exploiting these computationally for stochastic filtering seems involved. The widely applied particle filters also have limitations in connection with manifolds since the sampling requirements of particles become expensive [54].
A different way to approach a solution to the filtering problem was proposed by Mortensen [58]. Rather than trying to cope with the probabilistic setting of the filtering problem, he investigated the filtering problem from the viewpoint of optimal control. By using the control parameter to model noise and by integrating a quadratic penalty function over the time, he foufnd a

first-order optimal *Minimum Energy Filter*. The advantage of this method is that it does not rely on assumptions about, or approximations of, the a posteriori distribution and that *Hamilton-Jacobi-Bellman equation* provides a well-defined optimality criterion. It was shown theoretically in [50] that the minimum energy estimator converges with *exponential* speed for control systems on $\mathbb{R}^n$ that are uniformly observable.
The first article applying the minimum energy filters to geometrically constrained problems used perspective projections in the case of vectorial measurements [3]. The minimum energy filters were generalized to *second-order* filters on specific Lie groups with the help of geometric control theory in [48, 2, 70]. The Minimum Energy Filter, as introduced by Mortensen [58], was generalized to the Lie group SO_3 for the case of linear observation equations [88] and for attitude estimation [87]. Further follow-up work [71] generalized the filter to non-compact Lie groups [72].

1.3 Contribution

First, we introduce the main concepts of minimum energy filtering which are based on *optimal control theory* and derive the second-order minimum energy filter for the case of Euclidean state spaces. We also provide comparisons with state-of-the-art stochastic filters for difficult filtering problems (e.g. with sinusoidal sensor) demonstrating that the proposed method is superior. Then we extend the minimum energy filtering approach of Saccon et al. [73], which is itself based on the recursive filtering principle of Mortensen [58], to the problem of camera motion estimation on the special Euclidean group SE_3 . The corresponding non-linear filtering problem is based on observations of optical flow and depth map, which are calculated in a preprocessing step. This problem implicitly corresponds to a stereo camera system. We derive the second-order optimal minimum energy filter, which results in a differential equation for the optimal state and its second-order operator. Instead of a constant velocity model, where the camera motion is propagated constantly, we use a higher-order kinematic model that also respects constant accelerations or higher-order moments. We demonstrate that this model is beneficial in situations where the camera motion changes quickly, e.g. in curves or due to accelerations of the camera. Numerical experiments on the challenging KITTI benchmark demonstrate that our method reconstructs the camera motions as good as state-of-the-art methods. We also provide comparisons between the extended Kálmán filter and the proposed minimum energy filters. This shows that the minimum energy filter is superior which confirms the results of [88].

In a second step we generalize this approach from a stereo approach to a monocular approach which is based on non-linear observations of optical flow. The state variables which need to be reconstructed are the camera motion and higher-order kinematics as well as the disparity map. Here we use a disparity map instead of a depth map, which has a better numerical behavior. To avoid constraints such as barrier functions within the optimization, we define a novel *disparity group* and derive the corresponding exponential map and logarithmic map. Thus, the state space of the overall filtering problem can be expressed as a product Lie group. Since the observations of optical flow are not always reliable, e.g. in regions without texture, we extend the approach of Saccon et al. [73] from a quadratic to a *generalized Charbonnier* energy function. This energy function is a smooth approximation to the L^1 norm and gains accurate reconstructions in contrast to a quadratic L^2 energy function. In the sequel we provide the derivation of the minimum energy filter. The calculations result in a recursive description of the optimal state of the system as well as a second-order operator. Finally, we briefly discuss how to add a spatial regularizer to the filtering problem resulting in a better reconstruction of the disparity map.
Overall we solve in the last chapter a non-linear filtering problem on a (product) Lie group for a high dimensional problem – a problem which is infeasible for most stochastic filters. We also showed that the minimum energy filter enables to solve even more complicated problems that also use additional regularization terms.

1.4 Organisation

This work is structured as follows: In the chapter 2 we introduce the fundamental concepts of stochastic filtering theory in terms of the well-known *Kushner-Stratnonovich* equation. Then we give an overview about stochastic filters and briefly discuss the benefits and drawbacks of these. We continue with stating some of the fundamental concepts of optimal control theory which we will use then to derive the second-order optimal minimum energy filters in the case of an Euclidean state space. Afterwards we consider a case study where we compare the minimum energy filter with classical stochastic filters. Chapter 3 continues with a short introduction about fundamental concepts of differential geometry and Lie groups that we will need later for the derivation of the main results of this thesis. Then we discuss the concepts of how the minimum energy filtering principle formulated on a Euclidean space can be assigned to a Lie group. Chapter 4 contains the first main contribution of this work – the derivation of the second-order optimal min-

imum energy filter on the special Euclidean group SE_3 for camera motion estimation within a stereo approach. Finally, this approach is generalized in chapter 5 for the problem of joint monocular filtering of camera motion and disparity map. We conclude our work in chapter 6 with a summary about the contributions of this work and a brief discussion about what this work can be extended to in the future.
Please take into account that we only proof the main Theorems within the chapters for a better readability of this work. Thus, we shifted the the most technical proofs of Propositions and Lemmas to the appendix. However, these proofs belong to the main contributions of this work.

1.5 Notation

Sets

$\mathbb{N} := \{1, 2, 3, \dots\}$	natural numbers
$\mathbb{R}$	real numbers
$[n] := \{1, \dots, n\}$	set of integer numbers from 1 to n
Ω	(discrete and finite) image domain
$z \in \Omega$	point in image domain
$\lvert\Omega\rvert$	number of elements in Ω (if finite)
n_Ω	number of pixels in image domain

Differential Geometry and Lie groups

$\mathcal{M}$	differentiable manifold
$T_x\mathcal{M}$	tangent space of $\mathcal{M}$ at $x \in \mathcal{M}$
$T\mathcal{M}$	tangent bundle on $\mathcal{M}$
η, χ, ξ	tangent vectors $\in T_x\mathcal{M}$
$\langle \xi, \eta \rangle_x$	Riemannian metric at $x \in \mathcal{M}$
$\nabla_\cdot \cdot$	Levi-Civita connection on $T\mathcal{M}$
$\mathcal{G}$	general Lie group
$\mathfrak{g}$	Lie algebra of $\mathcal{G}$
$\mathrm{Id}_\mathcal{G}, \mathrm{Id}$	identity of considered Lie group
$\langle \xi, \eta \rangle = \langle \xi, \eta \rangle_{\mathrm{Id}}$	Riemannian metric on Lie algebra $\mathfrak{g}$
$\mathrm{vec}_\mathfrak{g} : \mathfrak{g} \to \mathbb{R}^n$	vectorization operator
$\mathrm{mat}_\mathfrak{g} =: \mathrm{vec}_\mathfrak{g}^{-1}$	inverse of $\mathrm{vec}_\mathfrak{g}$
$\mathrm{Exp}_\mathcal{G}$	exponential map on $\mathcal{G}$
$\mathrm{Log}_\mathcal{G}$	logarithmic map on $\mathcal{G}$
GL_4	General Linear group
SO_3	Special Orthogonal group
$\mathfrak{so}_3$	Lie algebra of SO_3

SE_3	Special Euclidean group
$\mathfrak{se}_3$	Lie algebra of SE_3
$\mathrm{vec}_{\mathfrak{se}} : \mathfrak{se}_3 \to \mathbb{R}^6$	vectorization operator
$\mathrm{mat}_{\mathfrak{se}} = \mathrm{vec}_{\mathfrak{se}}^{-1}$	inverse of $\mathrm{vec}_{\mathfrak{se}}$
$\mathrm{Pr} : \mathbb{R}^{4\times 4} \to \mathfrak{se}_3$	projection onto Lie algebra $\mathfrak{se}_3$
$L_G H := GH$	left translation
$T_H L_G$	tangent map of left translation at H
$x\eta := T_{\mathrm{Id}} L_x \eta$	shorthand for tangent map
$x^{-1}\eta := T_{\mathrm{Id}} L_x^* \eta$	shorthand for dual of tangent map
$\omega_\chi \eta := \omega(\chi, \eta) := \nabla_\chi \eta$	connection function for $\chi, \eta \in \mathfrak{g}$
$\omega_\chi^{\leftrightharpoons} \eta := \omega_\eta \chi$	swap operator
$\langle \omega_\chi^* \eta, \xi \rangle := \langle \eta, \omega_\chi \xi \rangle$	dual of connection function
$\langle \omega_\chi^{\leftrightharpoons *} \eta, \xi \rangle := \langle \eta, \omega_\xi \chi \rangle$	dual of swap operator
$[\cdot, \cdot]$	Lie bracket on considered Lie group, matrix commutator
$\mathbf{D} f(x)$	differential/Riemannian gradient of f at x
$\mathbf{D} f(x)[\eta] := \langle \mathbf{D} f(x), \eta \rangle$	directional derivative of f in direction η
$\mathrm{Hess}\, f(x)$	Hessian of a twice differentiable function $f : \mathcal{G} \to \mathbb{R}$
$\mathbf{D}_i f$	differential resp. i-th component of f
$\mathbf{D}_x$	differential of an expression resp. x

Control Theory and Stochastic Filtering

$x = x(t)$	state space variable
f	function that models changes of x
$y = y(t)$	observation space variable
h	function that models dependency of state space variable x
$\delta = \delta(t)$	control variable (model noise)
$\epsilon = \epsilon(t)$	observation noise (residual)
μ	costate / dual variable
$\dot{x}(t) = \frac{d}{dt} x(t)$	(total) time derivative of x
$\mathcal{H}$	Hamiltonian
$\mathcal{J}$	energy function

Matrix and Vector Notations

$\mathbf{x}$	column vector (bold printed)
$\mathbf{x}_i$	i-th to j-th component of $\mathbf{x}$
$\langle \mathbf{x}, \mathbf{y} \rangle$	scalar product on $\mathbb{R}^n$
A	matrix (capitals)
A_{ij}	element of A in row i and column j

$A_{i:j,k:l}$	block matrix with rows from i to j and columns from k to l from A
$\mathrm{vec}(A)$	vectorization (stacking columns of matrix A)
$\mathbb{1}_n$	$n \times n$ identity matrix
$\|x\|_Q^2 := \langle x, Qx \rangle$	quadratic form regarding Q
e_i^n	i-th unit vector in $\mathbb{R}^n$
$\mathbf{0}_{n \times m}$	zero matrix of dimension $n \times m$
$\mathbf{1}_{n \times m}$	one matrix of dimension $n \times m$
$\mathbf{0}_n$	zero vector of dimension n
$\mathbf{1}_n$	one vector of dimension n

Chapter 2

Stochastic Filtering

In this chapter we summarize the classical filtering problem that consists of finding the *a posteriori* distribution π_t of the unknown state $x(t)$ given observations $y(s), s \leq t$ in the past. We will require the theoretical background in the chapters below and demonstrate that non-linear filtering problems are involved to solve at large. The general solution of this filtering problem is implicitly given by the famous *Kushner-Stratonovich equation* which is also known as *Fokker-Planck equation* for physicists. This stochastic partial differential equation is the normalized version of the *Zakai equation* and these equations are connected by the *Kallianpur-Striebel* Theorem. For brevity we do not provide the derivation of these Theorems but want to give a little insight to filtering theory to explain why the general problem is hard.
Up to some carefully designed filtering problems, e.g. with linear dependencies between the unknown state and observations and with additive Gaussian noise, it is hard to find an explicit expression for the *a posteriori* distribution π_t. In fact, many non-linear filtering problems result in an infinite dimensional *a posteriori* distribution and thus cannot be solved exactly. Hence we introduce in this chapter some well known strategies to approximate the non-linear filtering problem and briefly discuss the benefits and drawbacks of these methods.

2.1 Theoretical Backgrounds

In this section we introduce the general filtering problem with the unknown state variable that is given through a stochastic differential equation (stochastic integral equation) in continuous time as well as the given observations $y(t)$ that depend on $x(t)$ and the *a posteriori* distribution $\pi(t)$. For details we refer to [7].

2.1.1 State and Observation Equations

In this work we only consider the case where the unknown state x is a diffusion process which can be described by a stochastic differential equation, i.e.

$$x_t = x_0 + \int_{t_0}^{t} f(x_s)\, ds + \int_{t_0}^{t} \sigma(x_s)\, dB_s, \tag{2.1}$$

where the second integral is stochastic integral by a Brownian motion B_t. This can also be expressed as differential equation as follows

$$dx(t) = f(x(t))\, dt + \sigma(x(t))\, dB_t, \quad x(t_0) = x_0. \tag{2.2}$$

This stochastic differential equation means, that the change in x retains like the function $f(x)$ disturbed by a noise term σ that might also be dependent on x and a Brownian motion. As in equation (1.2) we assume that the stochastic part of this stochastic differential equation is simply a white noise process. This means that $\sigma \equiv 1$. Then the differential of the Brownian motion is the white noise processs $\delta(t)dt = dB_t$. We do not known the state of x directly but some observations that depend on x by a measurable function $\tilde{h}$, such that the observations can be expressed as

$$y_t = y_0 + \int_{t_0}^{t} \tilde{h}(x_s)\, ds + W_t. \tag{2.3}$$

$W = (W_t)_{t\in\mathbb{R}}$ again denotes a brownian motion, which models noise on the observations. Since the function $\tilde{h}$ does not depend on y, this equation is often represented as

$$y_t = h(x(t)) + W_t, \tag{2.4}$$

where we set $h(x(t)) = y_0 + \int_{t_0}^{t} \tilde{h}(x_s)\, ds$. Take into account that h depends on the whole trajectory of x rather than the single state $x(t)$. For the mathematical investigations one usually prefers the first representation (2.3), however for practical purposes the second expression (2.4) is more suitable. As most filters usually use the representation (2.4) we will also use this convention.

2.1.2 The *A Posteriori* Distribution

The filtering problem consists of estimating the conditional distribution π_t of a signal x at time t given the information accumulated from observing y in the interval $[0, t]$. This means for a bounded and measurable function ϕ, computing the a posteriori distribution

$$\pi_t\phi := \mathbb{E}(\phi(x_t)|\mathcal{Y}_t), \tag{2.5}$$

where $\mathcal{Y}_t := \sigma(y_s, s \in [0,t]) \vee \mathcal{N}$ denotes the filtration generated by y_s augmented with all sets of measure null.

2.1.3 The Kushner-Stratonovich equation

In the 1960s Harold Kushner found a rigorous mathematical description for the general filtering problem that can be expressed as a partial differential equation, if some regularity assumptions are fulfilled [51, 52]. Using the second-order differential operator

$$A = f\mathbf{D} + \frac{1}{2}\sigma^2\mathbf{D}^2\,,$$

we can formulate the Kushner-Stratonovich partial differential equation which implicitly describes the *a posteriori* distribution of the general filtering problem.

Theorem 2.1.1 (Kushner-Stratonovich)**.** *Under suitable conditions [7, Eq. (3.25), (3.42)] the conditional distribution of the signal π_t satisfies the following evolution formula, called* Kushner-Stratonovich equation*,*

$$\begin{aligned}\pi_t(\phi) =& \pi_{t_0}(\phi) + \int_{t_0}^{t} \pi_s(A\phi)\,ds \\ &+ \int_{t_0}^{t} \Big(\pi_s(\phi h) - \pi_s(h\pi_s(\phi))\Big)(dy_s - \pi_s(h)\,ds)\end{aligned} \tag{2.6}$$

for any function ϕ for which $A\phi$ is bounded.

Remark 2.1.2. Note, that this partial differential equation cannot be solved explicitly in most cases and that solutions of (2.6) are in general infinite dimensional. This means that the true solution of (2.6), which is a probability distribution, cannot be expressed with a finite number of parameters [63]. However, there are non-linear filtering problems for which the solution of (2.6) has a finite dimensional representation.

2.2 Common Stochastic Filters

As the general solution of the non-linear filtering problem is in general infinite dimensional, suitable approximations of the *a posteriori* distribution have been developed in the past yielding a large bandwidth of stochastic filters. In this section we summarize the most important of them and briefly discuss their benefits and drawbacks. Depending on the situation and the specific

design of the filter one can gain excellent reconstruction results for many filtering problems. There exist a lot of strategies to improve the performance of a filter to a specific situation. Therefore we focus only on the general properties of the methods that we present and discuss typical problems of these methods. A good overview about non-linear filters can be found in [24].

2.2.1 The Kálmán Filter

The classical *Kálmán filter* [49] was derived in 1960 for the linear filtering problem with the following state- and observation equations

$$\begin{aligned} x_{k+1} &= Fx_k + \delta_k \,, \\ y_z &= Hx_k + \epsilon_z \,, \end{aligned}$$

with Gaussian model noise $\delta(\cdot)$ and Gaussian observation noise $\epsilon(\cdot)$. This means that δ and ϵ are fully described by the mean value which is zero and the covariance matrices R and Q, respectively. The functions $f(x) = Fx$ and $h(x) = Hx$ can be expressed by matrices F and H which represent linear transformations of the state. In this scenario, the *a posteriori* distribution can be understood as a linear transformation of Gaussians such that it is also a Gaussian. It can be shown that the filter is optimal with respect to the mean square error of the residuals. The discrete Kálmán filter for the linear filtering problem is given through the following propagation and update equations (cf. [84]):

$$\begin{aligned} x_{k+1}^- &= Fx_k && \text{(state propagation)} \\ P_{k+1}^- &= FP_kF^\top + R && \text{(covariance propagation)} \\ K_{k+1} &= P_{k+1}^- H^\top \left(HP_{k+1}^- H^\top + Q\right)^{-1} && \text{(Kálmán gain update)} \\ x_{k+1} &= x_{k+1}^- + K_{k+1}(y_{k+1} - Hx_{k+1}^-) && \text{(state update)} \\ P_{k+1} &= (I - K_{k+1}H)P_{k+1}^- && \text{(covariance update)} \end{aligned}$$

Benefits: The Kálmán filter leads to optimal results in the case of linear functions f and h as introduced above and Gaussian noise terms.

Drawbacks: The Kálmán filter is limited to the linear case. Additionally, Kálmán filter cannot be easily extended to filtering problems on Lie groups because the filtering problem becomes non-linear due to the curvature of the state space.

2.2.2 The Extended Kálmán Filter

The *extended Kálmán filter* (EKF), proposed in 1974, is a non-linear filter which adapts the concepts of the (linear) Kálmán filter [77]. The solution of the non-linear filtering problem

$$\begin{aligned} x_{k+1} =& f(x_k) + \delta_k \,, \\ y_k =& h(x_k) + \epsilon_k \,, \end{aligned}$$

can be approximated by similar propagation and update equations as in the linear case presented above. By linear approximations of the non-linear functions f and h the resulting recursive equations for mean value and the covariance of the *a posteriori* distribution are similar to those of the Kálmán filter.

Benefits: Simple generalization of the classical Kálmán filter to the non-linear filtering problem that results in good reconstructions. For many applications methods were found to increase the accuracy of the filter [24]. Using the Baker-Campbell-Hausdorff formula, the extended Kálmán filter was generalized to Lie groups [17].

Drawbacks: The assumption that the *a posteriori* distribution is a Gaussian is in general violated, especially if there exist multiple modes in the general solution. Adaption of the extended Kálmán filter to a novel non-linear problem might fail since there exists no general theory that states how accurate an approximation is. For involved non-linear filtering problems the linearization technique of the extended Kálmán filter often fails. This is also known as *divergence problem* [33].

2.2.3 The Unscented Kálmán Filter

The unscented Kálmán filter (UKF), introduced in 2000 [47], is a generalization of the classical Kálmán filter for non-linear filtering problems, where the *a posteriori* distribution is assumed to be a Gaussian [47, 22]. The key idea is to carefully select a number of samples (*sigma points*) which correctly describe the mean value and covariance in the one domain and to transfer these points by the non-linear function f and h into the measurement space. From these transformed points the parameters of the *a posteriori* distribution can be estimated. Since the full algorithm is relatively elongated we refer to the original article [47] where the single steps of the algorithm are listed.

Benefits: The *unscented transformation* of *sigma points* usually leads to better result than the linearization techniques of the extended Kálmán filter. Moreover, the unsented Kálmán filter was derived for general Riemannian manifolds in [42]. Therefore it is widely applicable in contrast to the extended Kálmán filter. Due to the statistical simplicity of the filter it is easy to transfer the concepts to filtering problems defined on Riemannian manifolds [42].

Drawbacks: Although the unscented Kálmán filter counts to the state-of-the-art non-linear filters, it is assumed that the *a posteriori* distributions is a Gaussian. For highly non-linear problems this approach often fails which will be shown in the experimental section of this chapter.

2.2.4 The Particle Filter

In 2001 *particle filters* (PFs) were proposed which are similar to unscented Kálmán filters; the main idea is to apply non-linear transformations to particles to gain a statistical description of the *a posteriori* distribution by Monte-Carlo sampling [29, 4]. In contrast to the unscented Kálmán filter the *a posteriori* is not restricted to be a Gaussian. In fact, particle filters cover a large bandwidth of different *a posteriori* distributions depending of the chosen number of desired particles.

Benefits: In contrast to extended and unscented Kálmán filters, particle filters cover a large bandwidth of distributions. Variance reduction techniques, such as importance sampling, enable to keep the computational effort relatively small. This cannot be taken for granted within general approaches. This makes the particle filter to the mean of choice for many applications e.g. [23].

Drawbacks: Although PFs deal with high dimensions they do not beat the so-called *"curse of dimensionality"* [25]. For moderate filtering problems in high dimensions other filters might perform better than PFs since for each single dimension at least a few particles need to be generated which is computationally expensive.

2.2.5 The Projection Filter

A differential geometric approach for solving the non-linear filtering problem is given by Brigo et al. [18] published in 1999. The key idea within this

approach is to project the infinite dimensional solution of (2.6) onto a suitable manifold of finite dimensional distributions on $\mathbb{R}^n$

$$S := \{p(\cdot, \theta), \theta \in \Theta\},$$

where $\Theta \subset \mathbb{R}^d$ is a parameter set. One can show that the set of square roots of these densities $S^{1/2} := \{\sqrt{p(\cdot, \theta), \theta \in \Theta}\}$ is an $d-$dimensional submanifold of $L^2(\mathbb{R}^n)$ where the inner product on L^2 gives the Fisher information matrix $g(\theta) = (g_{ij}(\theta))$. This inner product provides $S^{1/2}$ with the structure of a Riemannian manifold. A suitable choice of parametrized density functions are exponential families that cover a large bandwidth of distributions, i.e.

$$S = \{p(x, \theta) = \exp(\theta^\top c(x) - \psi(\theta)), \theta \in \Theta\},$$

where $c_1, \ldots, c_n$ are scalar functions (*sufficient statistics*) such that $\{1, c_1, \ldots, c_n\}$ are linearly independent and ψ is a normalizing function.
Brigo et al. [18] showed that the solution of (2.6) (in Stratonovich form) can be projected onto a manifold of exponential families. The resulting density function is parametrized by a vector $\theta(\cdot) \in \mathbb{R}^d$ that can be obtained by solving a simple stochastic (ordinary) differential equation. This is much easier to solve than the stochastic *partial* differential equation (2.6).

Benefits: The projection filter provides a quite general framework that covers a large bandwidth of *a posteriori* distributions (no only Gaussian), i.e. the sufficient statistics can be adapted in such a manner to fit to a specific filtering problem that result in an almost optimal filter. Additionally, for some specific non-linear problems also *exact* solutions can be found.

Drawbacks: It is difficult to generalize the projection filter to distributions on Riemannian manifolds although there are approaches that define suitable exponential families on compact manifolds. A more practical problem is to find suitable exponential families for a given problem that leads to good results.

2.2.6 Discussion on Common Non-Linear Filters

The filters summarized above show how a solution of the general filtering problem (2.6) can be found. However, extended Kálmán filter suffer from the *divergence problem*, whereas unscented Kálmán filter result in better reconstruction. Due to the fact that the *a posteriori* distribution of the general filtering problem is not Gaussian in general, there exists problems

where these methods fail. Particle filters are more flexible in finding the correct solution but do not work in high dimensions. In practice also a lot of tuning is required to find good parameters for the filter. A positive result is that most of these methods were generalized to non-Euclidean geometry. In contrast, projection filters, provided by an exact mathematical theory, will result in better reconstructions but are usually limited to Euclidean spaces. Due to these limitations we consider a different strategy based on optimal control theory on Lie groups and the dynamical programming principle. Therefore we introduce basic concepts of optimal control theory in the next section and derive the so-called minimum energy filter at the end of this section.

2.3 Control Theory: A Brief Introduction

For the derivation of the minimum energy filter we require some basic results from optimal control theory such as the famous *Pontryagin Minimum Principle* and the *Hamilton-Jacobi-Bellman equation*. We begin with some essential definitions and continue with the statement of the optimal control theory and the corresponding concepts that are required for solving it. The following definitions and Theorems from optimal control theory are well-known and can be found in Athans and Falb [5] as well as Fleming and Rishel [34] which provide an introduction to optimal control theory. In the sections belows we will follow the approach of Evans [32, chapter 10] in which a rigorous introduction to the field of optimal control is given.

2.3.1 Fixed-Time Optimal Control Problem

Let us consider the following dynamical system that can be understood as a differential equation of the unknown state x. $\delta(t)$ denotes the control variable.

$$\dot{x}(t) = f(x(t), \delta(t)) \tag{2.7}$$

$$x(t_0) = x_0\,. \tag{2.8}$$

Here, $x_0 \in \mathbb{R}^n$ is a given initial point and $f : \mathbb{R}^n \times D \to \mathbb{R}^n$ is a given bounded and Lipschitz-continuous function. D is a compact subset of $\mathbb{R}^m$ for a $m \in \mathbb{N}$. In the following the variable $\delta = \delta(t)$ is a *control variable* and the set of all measurable controls is the set of *admissible controls*, denoted by

$$\mathcal{D} := \{\delta : [t_0, t] \to D | \delta \text{ is measurable } \}\,.$$

The objective of optimal control theory is to find an admissible control δ such that a certain optimality criterion is fulfilled and such that the control

variable sufficies (2.7). The optimality criterion is usually given by a cost function on the control variable δ that needs to be minimized. For each admissible control $\delta(\cdot) \in \mathcal{D}$ we define the corresponding *cost* or *energy* function

$$\mathcal{J}_{x,t_0}(\delta) := \int_{t_0}^{t} r(x(\tau), \delta(\tau))\, d\tau + g(x(t))\,, \tag{2.9}$$

where $x(\cdot) = x^{\delta(\cdot)}(\cdot)$ solves the ordinary differential equation (ODE) (2.7). The functions $r : \mathbb{R}^n \times D \to \mathbb{R}$ and $g : \mathbb{R}^n \to \mathbb{R}$ are given cost functions for the *running costs* and *terminal costs*, respectively. Following Evans [32] these functions are assumed to be bounded and Lipschitz-continuous.
The goal of optimal control theory is to determine the *value function* that is defined through

$$\mathcal{V}(x,\tau) := \inf_{\delta\in\mathcal{D}} \mathcal{J}_{x,\tau}(\delta), \quad x \in \mathbb{R}^n,\, 0 \le \tau \le t, \tag{2.10}$$

such that the state equation (2.7) is fulfilled. It can be shown that the value function $\mathcal{V}$ satisfies a specific kind of Hamilton-Jacobi equation which can be used to find an optimal control. This procedure is also known as *Pontryagin Minimum Principle*. This is a result of the following Theorem.

Theorem 2.3.1. *The value function $\mathcal{V}$ is the unique viscosity solution of this terminal-value problem for the* Hamilton-Jacobi-Bellman *equations:*

$$\frac{\partial}{\partial\tau}\mathcal{V}(x,\tau) + \min_{d\in D}\left\{\langle f(x,d), \mathbf{D}_x\mathcal{V}(x,\tau)\rangle + r(x(\tau),d)\right\} = 0\,, \tag{2.11}$$

$$\mathcal{V}(\cdot,t) = g(\cdot)\,. \tag{2.12}$$

Proof. See [32, chapter 10, Theorem 2]. □

Remark 2.3.2. Below we will use the term "pre-Hamiltonian" which is defined by

$$\tilde{\mathcal{H}}(x,p,\delta,t) := \langle f(x(t),\delta(t)), p(t)\rangle + r(x(t),\delta(t))\,. \tag{2.13}$$

The Hamiltonian can then be defined as the minimum of the pre-Hamiltonian regarding δ, i.e.

$$\begin{aligned}\mathcal{H}(x,p,t) :=& \min_{\delta\in D}\left\{\langle f(x(t),\delta(t)), p(t)\rangle + r(x(t),\delta(t))\right\}\\ =& \min_{\delta\in D}\tilde{\mathcal{H}}(x,p,\delta,t)\end{aligned} \tag{2.14}$$

With the definition of the Hessian, the Hamilton-Jacobi-Bellman equation in (2.11) can also be written as

$$\frac{\partial}{\partial t}\mathcal{V}(x,t) + \mathcal{H}(x, \mathbf{D}_x\mathcal{V}(x,t), t) = 0\,. \tag{2.15}$$

Below we will introduce the main concepts of minimum energy filter. Unfortunately, these require a different energy function than the function introduced in (2.9), where a condition on the terminal value is given by the function g. For the minimum energy filter, however, we need a condition on the *initial value*. The corresponding energy is

$$\mathcal{J}_{x,t}(\delta) := \int_{t_0}^{t} r(x(\tau), \delta(\tau))\, d\tau + g(x(t_0)), \tag{2.16}$$

where $x(\cdot) = x^{\delta(\cdot)}(\cdot)$ solves the ordinary differential equation (ODE) (2.7). By switching the integration bounds we obtain an expression, which is similar to the original one:

$$\mathcal{J}_{x,t}(\delta) := -\int_{t}^{t_0} r(x(\tau), \delta(\tau))\, d\tau + g(x(t_0)). \tag{2.17}$$

The corresponding optimal control problem can be considered to run *backward in time* such that the pre-Hamiltonian needs to be defined as

$$\tilde{\mathcal{H}}(x, p, \delta, \tau) := \langle f(x(\tau), \delta(\tau)), p(\tau)\rangle - r(x(\tau), \delta(\tau))\,. \tag{2.18}$$

Using the Pontryagin's minimum principle we find the Hamiltonian for the control problem (2.17) is given as in (2.14) by

$$\begin{aligned}\mathcal{H}(x,p,r) :=& \min_{\delta\in D}\left\{\langle f(x(\tau),\delta(\tau)), p(\tau)\rangle - r(x(\tau),\delta(\tau))\right\}\\ =& \min_{\delta\in D}\tilde{\mathcal{H}}(x,p,\delta,\tau)\,.\end{aligned} \tag{2.19}$$

Theorem 2.3.1 states the Hamilton-Jacobi-Bellman equation for control problems that are solved forward in time. However, we require a similar statement for control problems that are solved *backwards in time*. This state can be derived in a similar way as in Theorem 2.3.1 but the proof is involved. The most important part is, to replace the terminal value problem by the initial value problem. The corresponding value function $\mathcal{V}$ is defined through

$$\mathcal{V}(x,t) = \inf_{\delta\in\mathcal{D}} \int_{t_0}^{t} r(x(\tau),\delta(\tau))\, d\tau + g(x(t_0)), \quad \text{s.t. } (2.7)\,. \tag{2.20}$$

As in Evans [32] one can derive a similar Hamilton-Jacobi-Bellman equation as in Theorem 2.3.1 for the value function (2.20) which reads

$$\frac{\partial}{\partial t}\mathcal{V}(x,t) - \mathcal{H}(x, \mathbf{D}_x\mathcal{V}(x,t), t) = 0\,. \tag{2.21}$$

For brevity we omit the proof of this equation which is beyond the scope of this work.

2.3.2 Observability and Controllability

Definition 2.3.3 (Observability)**.** We say that a state x_0 is observable at t_0 if, given any control δ, there is a time $t_1 > t_0$ such that knowledge $\delta(\tau)$ and the output $y(\tau)$ for $\tau \in (t_0, t_1]$ is sufficient to determine x_0. If every state x_0 is observable at every time t_0 in the interval of definition of the system, then we say that the system is observable.

This definition means that the state variable can be uniquely determined from the observations $y = y(t)$ and the knowledge about the trajectories of the noise processes $\delta = \delta(t)$. When the function f is not injective, e.g. $f(x) = \sin(x)$ one can reconstruct the value of x only modulo 2π which is not unique. We provide another counter-example where consider the observation of the position of the epipole (focus of expansion) $y(t) \in \mathbb{R}^2$ which is not sufficient to uniquely reconstruct the underlying camera motion.

Example 2.3.4. Let us consider the following filtering problem in which we want to determine the ego-motion of the camera $E(t) = (R(t), w(t)) \in \mathrm{SE}_3$, where $R(t) \in \mathrm{SO}_3$ is a rotation matrix and $w(t) \in \mathbb{R}^3$ is a translation vector. We assume that only observations of the epipole (focus of expansion) $y(t) \in \mathbb{R}^2$ are given and that E is constant. This can be expressed as the following system:

$$\dot{E}(t) = E(t)\delta(t)\,, \qquad E(t_0) = E_0\,, \tag{2.22}$$

$$y(t) = \pi(R(t)^\top w(t))\,, \tag{2.23}$$

where $\pi : \mathbb{R}^3 \to \mathbb{R}^2$, $(x_1, x_2, x_3)^\top \mapsto x_3^{-1}(x_1, x_2)^\top$ denotes the perspective projection onto the image plane. δ is a noise process which evolves on the tangent space of SE_3 at identity, such that $E\delta \in T_E\,\mathrm{SE}_3$. Since the Lie group SE_3 is a six dimensional manifold, but the observations are only two dimensional, there are four degrees of freedom for the camera motion that cannot be determined from epipole observations uniquely. Indeed the scale of the scene (magnitude of translation) get lost due to the projection or the rotation of the camera around the z-axis that leaves the epipole invariant.

Furthermore, changes in the translational and rotational components can be compensated by each other without changing the location of the epipole. Therefore, the underlying motion cannot be reconstructed uniquely and the control problem is not observable.

Definition 2.3.5 (Controllability)**.** If the state $x_1 = 0$ is reachable from x_0 at t_0, then we say that x_0 is controllable at time t_0. In other words, x_0 is controllable at t_0 if there exists a piecewise continuous (control) function δ such that

$$\Phi(T; \delta_{(t_0,t]}, x_0) = 0\,, \tag{2.24}$$

for some $T \geq t_0$, where Φ denotes the transition function of the system. If every state x_0 is controllable at any time t_0 in the interval of definition of the system, then we say that the system is (completely) controllable.

2.4 The Minimum Energy Filter (MEF)

In this section, we will derive the second-order minimum energy filter that is based on optimal control theory. Our presentation is oriented towards the original paper [58] from 1968 together with some modifications, in agreement with the more recent literature like e.g. [88] which was published in 2012. Furthermore, we will also provide an evolution equation of the so-called gain operator, which has not been provided in the original paper [58]. We again consider the filtering problem

$$\dot{x}(t) = f(x(t)) + \delta(t)\,, \quad x(t_0) = x_0\,, \tag{2.25}$$
$$y(t) = h(x(t)) + \epsilon(t)\,, \tag{2.26}$$

where we assume that the functions f and h are two times continuously differentiable functions and $\delta = \delta(t)$ and $\epsilon = \epsilon(t)$ are noise processes. Next, we will introduce an energy function on these noise processes that needs to be minimized.

2.4.1 Objective Function and Control Problem

Given the state (2.25) and observation (2.26) equations, we wish to find the optimal state of the system. To this end, we introduce a quadratic energy function that penalizes the model noise δ and the observation noise ϵ, as an optimality criterion. Thus, given an initial state x_0, we suggest to minimize

the energy function

$$\mathcal{J}(t; x_0, \delta, \epsilon) := \tfrac{1}{2}\|x(t_0) - x_0\|^2_{R_0} + \tfrac{1}{2}\int_{t_0}^{t} \|\delta(\tau)\|^2_R + \|\epsilon(\tau)\|^2_Q \, d\tau \,, \tag{2.27}$$

subject to the state (2.25) and observation (2.26) equations. Here, we assume that $R, R_0 \in \mathbb{R}^{n\times n}$ and $Q \in \mathbb{R}^{m\times m}$ are symmetric and positive definite. The main difficulty for minimum energy filtering is to find a *recursive* expression for the optimal state x, to avoid the need to minimize the energy (2.27) for each fixed point of time t separately.

Theorem 2.4.1. *The second-order minimum energy filter for the state* (2.25) *and observation* (2.26) *with energy* (2.27) *and initial state* x_0 *is given by the solution of the coupled differential equations*

$$\dot{x}^*(t) = f(x^*, t) + P(t)\mathbf{D}_1 h(x^*, t) Q\big(y(t) - h(x^*, t)\big) \,, \quad x^*(t_0) = x_0 \,, \tag{2.28}$$

$$\begin{aligned} \dot{P}(t) =& R^{-1} + \mathbf{D}_{x^*} f(x^*, t) P(t) + P(t)(\mathbf{D}_{x^*} f(x^*, t))^\top \\ &+ P(t) E(x^*, t) P(t) \,, \qquad\qquad P(t_0) = R_0^{-1} \,, \end{aligned} \tag{2.29}$$

where $E(x^*, t) := \mathbf{D}_{x^*}\big(\mathbf{D}h(x^*, t) Q(y(t) - h(x^*(t)))\big)$.

Before providing a sketch of the proof at the end of this section, we introduce the some required major concepts of optimal control theory first.

Remark 2.4.2. In practice, measurements of a system are only available at discrete points of time rather than continuously. Accordingly, we assume the measurement equation to be piecewise constant, i.e. the function $y(t)$ has the value $y(t_{k-1})$ for $t \in [t_{k-1}, t_k)$. For a continuous-discrete filter one requires a discrete control theory whose development is beyond the scope of this work.

Following Mortensen [58], we replace the observation noise by the residual given by (2.26),

$$\epsilon(t) = y(t) - h(x(t)) \,. \tag{2.30}$$

Insertion of equation (2.30) into the energy (2.27) provides us with an energy that depends on x_0, δ and x, i.e. $\mathcal{J}(t; x_0, \delta, x) := \mathcal{J}(t; x_0, \delta, y - h(x))$, where $x = x(\tau)|_{\tau\in[t_0,t]}$, and $\delta = \delta(\tau)|_{\tau\in[t_0,t]}$ are considered as full trajectories. Our optimization problem then reads

$$\min_{x,\delta} \mathcal{J}(t; x_0, \delta, x) \,, \qquad \text{subject to (5.1)} \,. \tag{2.31}$$

This optimization problem can be split up into two parts as each admissible control variable $\delta(t)$ will lead to a unique path $x(t)$ if we incorporate the

state equation (5.1) into the optimization. The first part consists of finding the *value function* that returns the value of the energy function after minimization subject to (5.1), given for a fixed point of time t by

$$\mathcal{V}(x,t;x_0,t_0) := \min_{\delta(\tau),t_0\leq\tau\leq t} \mathcal{J}(t;x_0,\delta,x) \qquad \text{subject to (5.1)}\,. \tag{2.32}$$

The second part focuses on the derivation of a recursive expression of the optimal solution – see Eq. (2.43) below.
The optimization problem (2.32) is solved by classical Hamilton-Jacobi theory. We start with defining the (pre-)Hamiltonian of the control problem that consists of two expressions: The integrand of the energy function (2.27) as well as the constraint equation (5.1), together with the costate variable $p \in \mathbb{R}^n$, that can be understood as a Lagrangian multiplier.

$$\begin{aligned}\mathcal{H}^-(x,p;\delta,t) =& \tfrac{1}{2}\|\delta(t)\|_R^2 + \tfrac{1}{2}\|y(t)-h(x(t))\|_Q^2 - \langle p(t),\dot{x}(t)\rangle \\ \overset{(5.1)}{=}& \tfrac{1}{2}\|\delta(t)\|_R^2 + \tfrac{1}{2}\|y(t)-h(x(t))\|_Q^2 - \langle p(t), f(x(t))+\delta(t)\rangle\,.\end{aligned} \tag{2.33}$$

Since we defined a cost function on the inital state $x(t_0)$ in (2.27), the corresponding control problem has to be solved backwards in time (cf. [73]). This explains the minus sign in (2.33). Applying Pontryagin's maximum/minimum principle [66], which states that the constrained optimization problem (2.32) is solved if

$$\mathbf{D}_\delta \mathcal{H}^-(x,p,\delta;t) = 0\,, \tag{2.34}$$

finally gives us the optimal control δ^*, i.e.

$$0 \overset{!}{=} \mathbf{D}_\delta \mathcal{H}^-(x,p,\delta^*;t) = R\delta^*(t) - p(t) \tag{2.35}$$

$$\Leftrightarrow \qquad \delta^*(t) = R^{-1}p(t)\,. \tag{2.36}$$

Insertion of the optimal control δ^* into the (pre-)Hamiltonian (2.33) leads to the optimal Hamiltonian $\mathcal{H} : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}$, that we define through $\mathcal{H}(x,p;t) := \mathcal{H}^-(x,p,\delta^*;t)$ and which is given by the following calculation.

$$\begin{aligned}\mathcal{H}(x,p;t) =& \tfrac{1}{2}\|R^{-1}p(t)\|_R^2 + \tfrac{1}{2}\|y(t)-h(x(t))\|_Q^2 - \langle p, f(x(t)) + R^{-1}p(t)\rangle \\ =& \tfrac{1}{2}(p(t))^\top (R^{-1})^\top p(t) - (p(t))^\top R^{-1}p(t) + \tfrac{1}{2}\|y(t)-h(x(t))\|_Q^2 \\ & - (p(t))^\top f(x(t)) \\ =& -\tfrac{1}{2}\|p(t)\|_{R^{-1}}^2 + \tfrac{1}{2}\|y(t)-h(x(t))\|_Q^2 - \langle p(t), f(x(t))\rangle\,.\end{aligned} \tag{2.37}$$

In the last line we used the symmetry of R. Now, we can write down Hamilton's equations

$$\dot{x}(t) = \mathbf{D}_2\mathcal{H}(x,p;t) = -f(x(t)) - R^{-1}p(t)\,, \tag{2.38}$$
$$\dot{p}(t) = -\,\mathbf{D}_1\mathcal{H}(x,p;t) = \mathbf{D}f(x(t))p(t) - \mathbf{D}h(x(t))Q(y(t)-h(x(t)))\,. \tag{2.39}$$

2.4.2 Hamilton-Jacobi-Bellmann Equation

From classical control theory [5] we obtain the Hamilton-Jacobi-Bellman partial differential equation as given in (2.21)

$$\frac{\partial \mathcal{V}}{\partial t}(x,t;x_0,t_0) - \mathcal{H}(x,\mathbf{D}_1\mathcal{V}(x,t;x_0,t_0),t) = 0\,. \tag{2.40}$$

If it is solved locally, the Hamilton-Jacobi-Bellman equation will result in a necessary condition for optimality. For a linear-quadratic control problem (i.e. with linear dynamics and quadratic energy function), it will be a necessary and sufficient optimality condition for optimality provided it is globally solved for all t. But even in the absence of only linear dependencies between the state of the system and the observations, condition (2.40) will be evaluated in the proof of Theorem 2.4.1 in order to obtain a recursive solution of the filtering problem.

Proof of Theorem 2.4.1. The initial optimal state $x^*(t_0)$ can be found by minimizing the value function for $t = t_0$.

$$\mathcal{V}(x,t_0;x_0,t_0) = \tfrac{1}{2}\|x(t_0)-x_0\|^2_{R_0} \tag{2.41}$$

As R_0 is symmetric and positive definite, the optimal state for the initialization is $x^*(t_0) = x_0$. Next, we consider the necessary condition of the state process $x^*(t)$ for an arbitrary $t \in \mathbb{R}$ in order to be optimal, i.e.

$$\mathbf{D}_1\mathcal{V}(x^*,t;x_0,t_0) = 0\,. \tag{2.42}$$

All solutions of this equation match as optimal trajectories $x(\tau)|_{\tau\in[t_0,t]}$. However, the computation of these solutions is expensive. Since a considerable effort is required to compute the optimal trajectory for each single $t \in T$, we are interested in a *recursive rule* for updating an optimal state. This can be found by computing the total time derivative of the necessary condition to the optimal trajectory (2.42), restricted to an optimal path x^*. Thus, after inserting the Hamilton-Jacobi-Bellman equation (2.40) into the resulting condition, we obtain the evolution equation for the optimal state

$$\dot{x}^*(t) = f(x^*(t)) + \big(\mathrm{Hess}_1\,\mathcal{V}(x^*,t;x_0,t_0)\big)^{-1}\mathbf{D}_1h(x^*(t))Q\big(y(t)-h(x^*(t))\big)\,. \tag{2.43}$$

The derivation of this equation can be found in Appendix A.1.1. The calculations there are based on the original approach of Mortensen [58] but we will additionally derive also the evolution equation of the second-order operator explicitly which is missing in the original work. Additionally, we provide a derivation that is consistent with more recent work of Zamani et al. [88]. The evolution equation of x^* (2.43) still depends on the Hessian of the value function, which is unknown. Following [58], one may compute the total time derivative of the Hessian, which results in a differential equation that contains third order tensors. By continuing this iteratively, we find that the computation of the optimal estimate x^* will contain derivatives of the value function of any order, since the filtering problem generally is infinite dimensional. As a consequence, an approximation is conducted as described next.

2.4.3 Truncation and Evolution of Second-Order Operator

Since calculation of the infinite recursion, which is required for exact filtering, is impossible, we require an approximation of the true solution. Thus, we exemplarily restrict the recursion to a *second-order* minimum energy filter that contains derivatives of first and second-order only. All higher-order derivatives that appear are neglected. Using the notation

$$\hat{Z}(x^*, t) := \operatorname{Hess} \mathcal{V}(x^*, t; t_0, x_0)\,, \tag{2.44}$$

we compute the total time derivative of $\hat{Z}$. The corresponding calculations can be found in section A.1.2. As proposed in [88, 73], we find an approximation $Z(x^*, t)$ to the optimal operator $\hat{Z}(x, t)$ that develops from the following matrix differential equation by omitting higher-order terms:

$$\begin{aligned}\frac{d}{dt} Z(x^*, t) = &\operatorname{Hess}_1 \mathcal{H}(x^*, 0, t) + (\mathbf{D}_2 \mathbf{D}_1 \mathcal{H}(x^*, 0, t))^\top \cdot Z(x^*, t) \\ &+ Z(x^*, t) \cdot \mathbf{D}_1 \mathbf{D}_2 \mathcal{H}(x^*, 0, t) + Z(x^*, t) \cdot \operatorname{Hess}_2 \mathcal{H}(x^*, 0, t) \cdot Z(x^*, t)\,.\end{aligned} \tag{2.45}$$

Since the Hessian of the value function must be inverted in (2.43), we replace the corresponding expression by

$$\big(\operatorname{Hess}_1 \mathcal{V}(x^*, t; x_0, t_0)\big)^{-1} \approx Z(x^*, t)^{-1} =: P(t)\,. \tag{2.46}$$

By evaluating the Hessian of the value function at $t = t_0$, we find the initial-

ization for the operator P,

$$\begin{aligned} P(t_0) =& \big(\mathrm{Hess}_1\, \mathcal{V}(x^*, t_0; x_0, t_0)\big)^{-1} \\ \overset{(2.41)}{=}& \big(\mathrm{Hess}_{x^*}(\tfrac{1}{2}\|x^*(t_0) - x_0\|^2_{R_0}\big)^{-1} \\ =& R_0^{-1}\,. \end{aligned}$$

Using the well known rule for computing the derivative of the inverse of a matrix, we obtain the evolution equation for the gain operator P.

$$\begin{aligned} \dot{P}(t) =& \frac{d}{dt}(Z(t))^{-1} = -Z(x^*,t)^{-1}\dot{Z}(x^*,t)Z(x^*,t)^{-1} \\ =& -P(t)\Big(\mathrm{Hess}_1\, \mathcal{H}(x^*,0,t) + (\mathbf{D}_2\mathbf{D}_1\mathcal{H}(x^*,0,t))^\top \cdot Z(x^*,t) \\ & + Z(x^*,t)\cdot \mathbf{D}_1\mathbf{D}_2\mathcal{H}(x^*,0,t) \\ & + Z(x^*,t)\cdot \mathrm{Hess}_2\, \mathcal{H}(x^*,0,t)\cdot Z(x^*,t)\Big)P(t) \\ =& -P(t)\,\mathrm{Hess}_1\, \mathcal{H}(x^*,0,t)P(t) - P(t)(\mathbf{D}_2\mathbf{D}_1\mathcal{H}(x^*,0,t))^\top \\ & - \mathbf{D}_1\mathbf{D}_2\mathcal{H}(x^*,0,t)P(t) - \mathrm{Hess}_2\, \mathcal{H}(x^*,0,t). \end{aligned} \tag{2.47}$$

Finally, the calculation of the expressions in (2.47) can be written explicitly as follows:

$$\begin{aligned} \mathrm{Hess}_1\, \mathcal{H}\big(x^*,0,t\big) =& \mathbf{D}_{x^*}\Big(\mathbf{D}h(x^*(t))Q(y(t) - h(x^*(t))) - \mathbf{D}f(x^*,t)p\Big)\Big|_{p=0} \\ =& \mathbf{D}_{x^*}\Big(\mathbf{D}h(x^*,t)Q^{-1}(y(t) - h(x^*(t)))\Big) =: E(x^*,t)\,, \\ \mathbf{D}_2\mathbf{D}_1\mathcal{H}(x^*,0,t) =& \mathbf{D}_p\Big(-\mathbf{D}f(x^*(t))p + \mathbf{D}h(x^*(t))Q(y(t) - h(x^*(t)))\Big)\Big|_{p=0} \\ =& -\mathbf{D}f(x^*(t))\,, \\ \mathbf{D}_1\mathbf{D}_2\mathcal{H}(x^*,0,t) =& \mathbf{D}_{x^*}\Big(-f(x^*(t)) - R^{-1}p\Big) = -\mathbf{D}_{x^*}f(x^*(t))\,, \\ \mathrm{Hess}_2\, \mathcal{H}(x^*,0,t) =& \mathbf{D}_p\Big(f(x^*(t)) - R^{-1}p\Big)\Big|_{p=0} = -R^{-1}\,. \end{aligned}$$

After insertion of these expressions into (2.47), we obtain the second-order minimum energy filter, which completes the proof of Theorem 2.4.1. □

2.5 Case Studies

Before considering the actual filtering problem for camera motion reconstruction and disparity map estimation in the following chapters, we provide

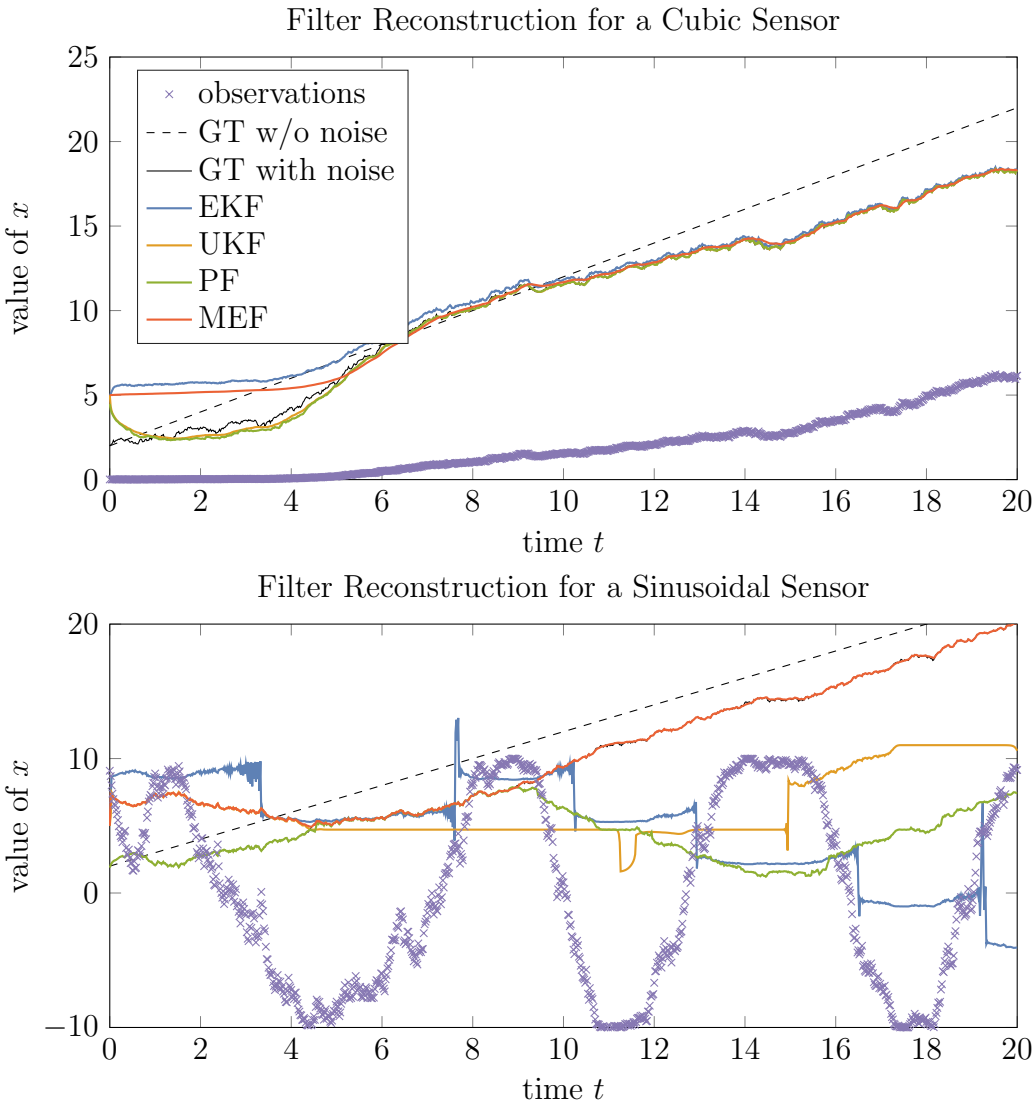


Figure 2.1: Comparison of the reconstructed trajectories of extended Kalman filter (EKF), unscented Kalman filter (UKF), particle filter (PF) (with 100 particles) and the minimum energy filter (MEF) given the observations of the cubic sensor (left) and a sinusoidal sensor (right). In the case of the cubic sensor one can observe that the EKF slightly worse. UKF, PF and MEF perform almost similarly after convergence. In the more difficult setting of the sinusoidal sensor, EKF and UKF diverge relatively fast and result in piecewise constant reconstructions. The PF converges faster than the MEF but the MEF stays longer at the true solution (dashed line). In this experiment we use for all stochastic filters the true covariance matrices and all filters have the same initialization.

a small academic example on an Euclidean state space. We use the minimum energy filter that we derived above. The case study contains two non-linear filtering problems to compare the proposed minimum energy filter with the extended Kalman filter [35], the unscented Kalman filter [47] as well as the standard particle filter [38] directly. For the extended Kalman filter we use our own implementation whereas the code for the unscented Kalman filter and the particle filter is from [74] and [75], respectively. The state- and observation equations are given through

$$\dot{x}(t) = 1 + \sigma\epsilon(t)\,, \quad x(t_0) = 2\,, \tag{2.48}$$

$$y(t) = h(x(t)) + \delta(t)\,, \tag{2.49}$$

where the processes $\delta(t)$ and $\epsilon(t)$ correspond to white noise processes with fixed covariance matrices R and Q, respectively. In the experiments we use moderate model noise ($\sigma = 0.5$) and consider two non-linear scenarios:

$$h(x) = 10^{-3}x^3\,, \qquad \text{(cubic sensor)} \tag{2.50}$$

$$h(x) = 10\sin(x)\,. \qquad \text{(sinusoidal sensor)} \tag{2.51}$$

Here we added the coefficients to enable the representation of the observations on the same scale as the state. The reconstructions of the trajectory of the optimal state x gained by extended Kalman filter (EKF), unscented Kalman filter (UKF) and minimum energy filter (MEF) are depicted in Fig. 2.1, where we use the true covariance matrices for all stochastic filters; all filters were initialized equally with $x_0 = 5$. We also evaluate the cumulative asymptotic error after convergence of the filters ($t = 1$) in Fig. 2.2. In the simpler case of the cubic sensor the MEF is as good as UKF and PF; in the more difficult case of a sinusoidal sensor the MEF outperforms the stochastic filters clearly.

2.6 Derivation of Third-Order Filters

In this section we will consider third-order optimal minimum energy filters. For that reason we extend the approach from section 2.4 and explicitly derive evolution equations of the third order operator of the optimal state. The derivation of higher-order minimum energy filters was not done before and therefore we provide the full derivations below. On multi-dimensional Euclidean spaces this would require tensor calculus, which is beyond the scope of this work. Thus, we consider only the one-dimensional case, i.e. the state variable x and the observation y are real valued and the functions f and h map to one-dimensional spaces. For simplicity we will use the prime-notation f' for derivatives of a function f regarding x. Additionally, we assume that

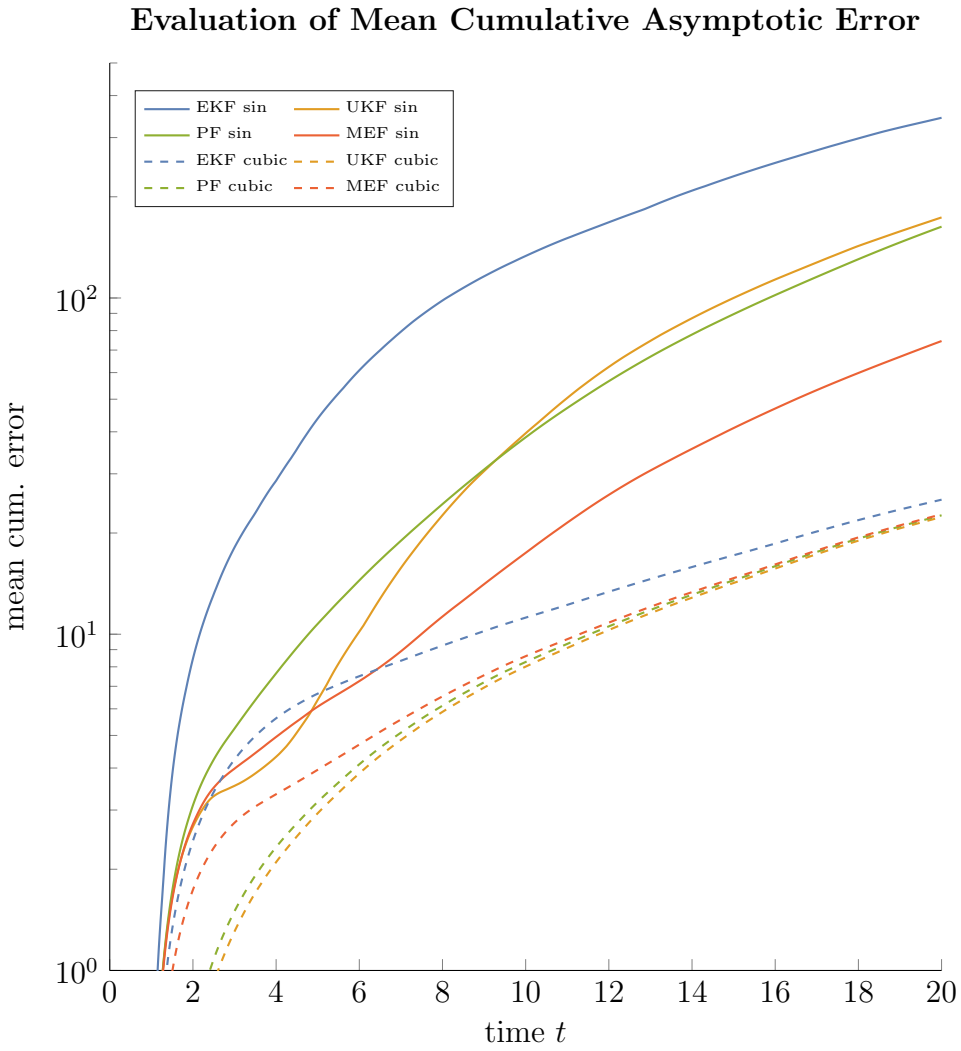


Figure 2.2: Evaluation of the mean cumulative asymptotic error after convergence of the filters ($t = 1$) on a logarithmic scale. The mean error is averaged over 100 sample tracks for the filtering problem in (2.48) and (2.49). In the scenario of a cubic sensor (cubic), the UKF, PF and MEF are similar, only the EKF is slightly worse. In the more difficult case of the sinusoidal sensor (sin), the MEF outperforms the PF, UKF and EKF clearly. However, in this setting the error of all filters is quite large.

the Hamiltonian in (2.37) is three times continuously differentiable in the arguments x and p such that all partial derivatives commute.
The equation for the optimal state stays equal to (2.43), which reads in the one-dimensional case:

$$\dot{x}^*(t) = f(x^*(t)) + \big(\mathcal{V}''(x^*, t; x_0, t_0)\big)^{-1} h'(x^*(t)) Q\big(y(t) - h(x^*(t))\big)\,. \quad (2.52)$$

Using again the notation $Z(t) = \mathcal{V}''(x^*(t), t)$ we continue with the calculation of the total time derivative of Z to find a recursive representation:

$$\begin{aligned}
&\frac{d}{dt} Z(x^*(t), t) = \frac{d}{dt} \mathcal{V}''(x, t)\big|_{x=x^*} \\
&= \Big(\mathcal{V}'''(x, t)\dot{x}(t) + \big(\tfrac{\partial}{\partial t} \mathcal{V}(x, t)\big)'' \Big)\Big|_{x=x^*} \\
&\overset{(2.40)}{=} \Big(\mathcal{V}'''(x, t)\dot{x}(t) + \big(\mathcal{H}(x, \mathcal{V}'(x, t), t)\big)'' \Big)\Big|_{x=x^*} \\
&= \mathcal{V}'''(x^*, t)\dot{x}^*(t) + \Big(\mathbf{D}_1 \mathcal{H}(x, \mathcal{V}'(x, t), t) + \mathbf{D}_2 \mathcal{H}(x, \mathcal{V}'(x, t), t) \mathcal{V}''(x, t) \Big)'\Big|_{x=x^*} \\
&= \mathcal{V}'''(x^*, t)\dot{x}^*(t) + \Big(\mathbf{D}_1^2 \mathcal{H}(x, \mathcal{V}'(x, t), t) + (\mathbf{D}_2 \mathbf{D}_1 \mathcal{H}(x, \mathcal{V}'(x, t), t)) \mathcal{V}''(x, t) \\
&\qquad + \big(\mathbf{D}_1 \mathbf{D}_2 \mathcal{H}(x, \mathcal{V}'(x, t), t) + \mathbf{D}_2^2 \mathcal{H}(x, \mathcal{V}'(x, t), t) \mathcal{V}''(x, t) \big) \mathcal{V}''(x, t) \\
&\qquad + \mathbf{D}_2 \mathcal{H}(x, \mathcal{V}'(x, t), t) \mathcal{V}'''(x, t) \Big)\Big|_{x=x^*} \\
&\overset{(2.42)}{=} \mathcal{V}'''(x^*, t)\dot{x}^*(t) + \mathbf{D}_1^2 \mathcal{H}(x^*, 0, t) + \mathbf{D}_2 \mathbf{D}_1 \mathcal{H}(x^*, 0, t) \mathcal{V}''(x^*, t) \\
&\qquad + \big(\mathbf{D}_1 \mathbf{D}_2 \mathcal{H}(x^*, 0, t) + \mathbf{D}_2^2 \mathcal{H}(x^*, 0, t) \mathcal{V}''(x^*, t) \big) \mathcal{V}''(x^*, t) \\
&\qquad + \mathbf{D}_2 \mathcal{H}(x^*, 0, t) \mathcal{V}'''(x^*, t) \\
&= \mathbf{D}_1^2 \mathcal{H}(x^*, 0, t) + \mathbf{D}_2 \mathbf{D}_1 \mathcal{H}(x^*, 0, t) Z(t) \\
&\qquad + \mathbf{D}_1 \mathbf{D}_2 \mathcal{H}(x^*, 0, t) Z(t) + \mathbf{D}_2^2 \mathcal{H}(x^*, 0, t) Z(t)^2 \\
&\qquad + \big(\dot{x}^* + \mathbf{D}_2 \mathcal{H}(x^*, 0, t) \big) \mathcal{V}'''(x^*, t) \\
&\overset{(2.52)}{=} \mathbf{D}_1^2 \mathcal{H}(x^*, 0, t) + \mathbf{D}_2 \mathbf{D}_1 \mathcal{H}(x^*, 0, t) Z(t) \\
&\qquad + \mathbf{D}_1 \mathbf{D}_2 \mathcal{H}(x^*, 0, t) Z(t) + \mathbf{D}_2^2 \mathcal{H}(x^*, 0, t) Z(t)^2 \\
&\qquad + \big(f(x^*(t)) + \big((Z(t))^{-1} h'(x^*(t)) Q\big(y(t) - h(x^*(t))\big) \\
&\qquad\quad + \mathbf{D}_2 \mathcal{H}(x^*, 0, t) \big) \mathcal{V}'''(x^*, t) \\
&= \mathbf{D}_1^2 \mathcal{H}(x^*, 0, t) + 2\mathbf{D}_2 \mathbf{D}_1 \mathcal{H}(x^*, 0, t) Z(t) + \mathbf{D}_2^2 \mathcal{H}(x^*, 0, t) Z(t)^2 \\
&\qquad + \big((Z(t))^{-1} h'(x^*(t)) Q\big(y(t) - h(x^*(t))\big) \big) \mathcal{V}'''(x^*, t)\,.
\end{aligned} \quad (2.53)$$

Here we used in the last step the equality $\mathbf{D}_2 \mathcal{H}(x^*, 0, t) = -f(x^*, t)$ and that the partial derivatives commute. Note that the last summand of (2.53) is

not contained in the truncated evolution equation in (2.45). By substitution of $P(t) := (Z(t))^{-1}$ we avoid the inversion of Z. By using the derivative of the inverse, i.e.

$$\frac{d}{dt}P(t) = \frac{d}{dt}(Z(t))^{-1} = -(P(t))^2\dot{Z}(t)\,, \tag{2.54}$$

and by inserting of (2.53) into (2.54) we gain the following differential equation:

$$\begin{aligned}\dot{P}(t) = \Big(&- (P(t))^2\mathbf{D}_1^2\mathcal{H}(x^*,0,t) - 2\mathbf{D}_2\mathbf{D}_1\mathcal{H}(x^*,0,t)P(t) - \mathbf{D}_2^2\mathcal{H}(x^*,0,t)\\ &- \big((P(t))^3h'(x^*(t))Q\big(y(t)-h(x^*(t))\big)\mathcal{V}'''(x^*,t)\Big)\\ =&Q\big((y(t)-h(x^*(t)))h''(x^*(t)) - (h'(x^*(t)))^2\big)(P(t))^2 + 2f'(x^*(t))P(t)\\ &+ R - \big((P(t))^3h'(x^*(t))Q\big(y(t)-h(x^*(t))\big)\mathcal{V}'''(x^*,t)\Big)\end{aligned} \tag{2.55}$$

2.6.1 Evolution of Third-Order Operator

For the derivation of a third-order optimal filter it remains to calculate the total time derivative of the third order operator $Y(t) := Y(t,x^*) = \mathcal{V}'''(x^*,t)$, which can again be obtained by differentiation and insertion of the existing equations of x^*, Z and Y. This results in the following differential equation for Y :

$$\begin{aligned}&\frac{d}{dt}Y(x^*(t),t) = \frac{d}{dt}\mathcal{V}'''(x,t)\big|_{x=x^*}\\ =&\mathcal{V}''''(x^*,t)\dot{x}^*(t) + \Big(\tfrac{\partial}{\partial t}\mathcal{V}(x,t)\Big)'''\Big|_{x=x^*}\\ =&\mathcal{V}''''(x^*,t)\dot{x}^* + \Big(\mathcal{H}(x,\mathcal{V}'(x,t),t)\Big)'''\Big|_{x=x^*}\\ =&\mathcal{V}''''(x^*,t)\dot{x}^* + \Big(\mathbf{D}_1^2\mathcal{H}(x,\mathcal{V}'(x,t),t) + 2\mathbf{D}_1\mathbf{D}_2\mathcal{H}(x,\mathcal{V}'(x,t),t)\mathcal{V}''(x,t)\\ &+ \mathbf{D}_2^2\mathcal{H}(x,\mathcal{V}'(x,t),t)(\mathcal{V}''(x,t))^2 + \mathbf{D}_2\mathcal{H}(x,\mathcal{V}'(x,t),t)\mathcal{V}'''(x,t)\Big)'\Big|_{x=x^*}\\ =&\mathcal{V}''''(x^*,t)\dot{x}^*(t) + \Big(\mathbf{D}_1^3\mathcal{H}(x,\mathcal{V}'(x,t),t) + \mathbf{D}_2\mathbf{D}_1^2\mathcal{H}(x,\mathcal{V}'(x,t),t)\mathcal{V}''(x,t)\\ &+ 2\mathbf{D}_1^2\mathbf{D}_2\mathcal{H}(x,\mathcal{V}'(x,t),t)\mathcal{V}''(x,t) + 2\mathbf{D}_1\mathbf{D}_2^2\mathcal{H}(x,\mathcal{V}'(x,t),t)(\mathcal{V}''(x,t))^2\\ &\quad + 2\mathbf{D}_1\mathbf{D}_2\mathcal{H}(x,\mathcal{V}'(x,t),t)\mathcal{V}'''(x,t)\\ &+ \mathbf{D}_1\mathbf{D}_2^2\mathcal{H}(x,\mathcal{V}'(x,t),t)(\mathcal{V}''(x,t))^2 + \mathbf{D}_2^3\mathcal{H}(x,\mathcal{V}'(x,t),t)(\mathcal{V}''(x,t))^3\\ &\quad + 2\mathbf{D}_2^2\mathcal{H}(x,\mathcal{V}'(x,t),t)\mathcal{V}'''(x,t)\mathcal{V}''(x^*,t)\\ &+ \mathbf{D}_1\mathbf{D}_2\mathcal{H}(x,\mathcal{V}'(x,t),t)\mathcal{V}'''(x,t) + \mathbf{D}_2^2\mathcal{H}(x,\mathcal{V}'(x,t),t)\mathcal{V}''(x,t)\mathcal{V}'''(x,t)\end{aligned}$$

$$
\begin{aligned}
&\quad + \mathbf{D}_2\mathcal{H}(x,\mathcal{V}'(x,t),t)\mathcal{V}''''(x,t)\Big)\Big|_{x=x^*}\\
=&\mathcal{V}''''(x^*,t)\dot{x}^*(t) + \mathbf{D}_1^3\mathcal{H}(x^*,0,t) + \mathbf{D}_2\mathbf{D}_1^2\mathcal{H}(x^*,0,t)\mathcal{V}''(x^*,t)\\
&\quad + 2\mathbf{D}_1^2\mathbf{D}_2\mathcal{H}(x^*,0,t)\mathcal{V}''(x^*,t) + 2\mathbf{D}_1\mathbf{D}_2^2\mathcal{H}(x^*,0,t)(\mathcal{V}''(x^*,t))^2\\
&\quad\quad + 2\mathbf{D}_1\mathbf{D}_2\mathcal{H}(x^*,0,t)\mathcal{V}'''(x^*,t)\\
&\quad + \mathbf{D}_1\mathbf{D}_2^2\mathcal{H}(x^*,0,t)(\mathcal{V}''(x^*,t))^2 + \mathbf{D}_2^3\mathcal{H}(x^*,0,t)(\mathcal{V}''(x^*,t))^3\\
&\quad\quad + 2\mathbf{D}_2^2\mathcal{H}(x^*,0,t)\mathcal{V}'''(x^*,t)\mathcal{V}''(x^*,t)\\
&\quad + \mathbf{D}_1\mathbf{D}_2\mathcal{H}(x^*,0,t)\mathcal{V}'''(x^*,t) + \mathbf{D}_2^2\mathcal{H}(x^*,0,t)\mathcal{V}''(x^*,t)\mathcal{V}'''(x^*,t)\\
&\quad\quad + \mathbf{D}_2\mathcal{H}(x^*,0,t)\mathcal{V}''''(x^*,t)
\end{aligned}
$$

After insertion of the expressions for the second and third order operator we obtain the following equality:

$$
\begin{aligned}
\frac{d}{dt}Y(x^*(t),t) &= \mathcal{V}''''(x^*,t)\dot{x}^* + \mathbf{D}_1^3\mathcal{H}(x^*,0,t) + \mathbf{D}_2\mathbf{D}_1^2\mathcal{H}(x^*,0,t)Z(t)\\
&\quad + 2\mathbf{D}_1^2\mathbf{D}_2\mathcal{H}(x^*,0,t)Z(t) + 2\mathbf{D}_1\mathbf{D}_2^2\mathcal{H}(x^*,0,t)(Z(t))^2\\
&\quad\quad + 2\mathbf{D}_1\mathbf{D}_2\mathcal{H}(x^*,0,t)Y(t)\\
&\quad + \mathbf{D}_1\mathbf{D}_2^2\mathcal{H}(x^*,0,t)(Z(t))^2 + \mathbf{D}_2^3\mathcal{H}(x^*,0,t)(Z(t))^3\\
&\quad\quad + 2\mathbf{D}_2^2\mathcal{H}(x^*,0,t)Y(t)Z(t)\\
&\quad + \mathbf{D}_1\mathbf{D}_2\mathcal{H}(x^*,0,t)Y(t) + \mathbf{D}_2^2\mathcal{H}(x^*,0,t)Z(t)Y(t)\\
&\quad\quad + \mathbf{D}_2\mathcal{H}(x^*,0,t)\mathcal{V}''''(x^*,t)\,.
\end{aligned}
$$

This expression contains also fourth-order terms, i.e. $\mathcal{V}''''$. By omitting these we obtain an third-order approximation of Y which we denote by $\tilde{Y}$. It is given through:

$$
\begin{aligned}
\frac{d}{dt}\tilde{Y}(t) &= \mathbf{D}_1^3\mathcal{H}(x^*,0,t) + 3\mathbf{D}_1^2\mathbf{D}_2\mathcal{H}(x^*,0,t)Z(t)\\
&\quad + 3\mathbf{D}_1\mathbf{D}_2^2\mathcal{H}(x^*,0,t)(Z(t)^2 + \tilde{Y}(t)) + \mathbf{D}_2^3\mathcal{H}(x^*,0,t)(Z(t))^3\\
&\quad + 3\mathbf{D}_2^2\mathcal{H}(x^*,0,t)Z(t)\tilde{Y}(t)
\end{aligned}
\tag{2.56}
$$

The evaluation of the partial derivatives of the Hamiltonian results in the following expressions:

$$
\begin{aligned}
\mathbf{D}_1^3\mathcal{H}(x^*,0,t) &= 3Qh'(x^*)h''(x^*) - Q(y-h(x^*))h'''(x^*) && (2.57)\\
\mathbf{D}_1^2\mathbf{D}_2\mathcal{H}(x^*,0,t) &= -f''(x^*) && (2.58)\\
\mathbf{D}_1\mathbf{D}_2^2\mathcal{H}(x^*,0,t) &= 0 && (2.59)\\
\mathbf{D}_2^3\mathcal{H}(x^*,0,t) &= 0 && (2.60)\\
\mathbf{D}_2^2\mathcal{H}(x^*,0,t) &= -R && (2.61)
\end{aligned}
$$

This gives the final differential equation equation for the approximate third order operator $\tilde{Y}$ which is

$$\begin{aligned}\frac{d}{dt}\tilde{Y} = 3Qh'(x^*)h''(x^*) - Q(h(x^*) - y)h'''(x^*) - 3f''(x^*)Z \\ - 3RZ\tilde{Y}, .\end{aligned} \tag{2.62}$$

The initial values for the optimal state is already given in (2.28), which is $x^*(t_0) = x_0$. The optimal state of the second-order operator can be calculated as

$$\begin{aligned}Z(t_0) =& \mathcal{V}''(x^*(t_0), t_0) \\ =& \frac{\partial^2}{\partial (x^*)^2} \tfrac{1}{2} R_0^{-1}(x^* - x_0)^2 \\ =& \frac{\partial}{\partial x^*} R_0^{-1}(x^* - x_0) \\ =& R_0^{-1} .\end{aligned}$$

Similarly, the initial state of the third-order operator can be calculated, which is zero, i.e. $\tilde{Y}(t_0) = 0$. The expressions (2.52), (2.55) and (2.62) together with the initial values gives the third-order optimal minimum energy filter which is summarized in the following Theorem:

Theorem 2.6.1. *The third-order optimal minimum energy filter for the* one-dimensional *state and observation equations* (2.25) *and* (2.26) *with energy function* (2.27) *is given through the following coupled evolution equations:*

$$\begin{aligned}\dot{x}^*(t) =& f(x^*(t)) + P(t)h'(x^*(t))Q\big(y(t) - h(x^*(t))\big) , \\ \dot{P}(t) =& Q\big((y(t) - h(x^*(t)))h''(x^*(t)) - (h'(x^*(t)))^2\big)(P(t))^2 + 2f'(x^*(t))P(t) \\ &+ R - \Big((P(t))^3 h'(x^*(t))Q\big(y(t) - h(x^*(t))\big)\tilde{Y}(t)\Big) \\ \dot{\tilde{Y}}(t) =& 3Qh'(x^*(t))h''(x^*(t)) - Q(y(t) - h(x^*(t)))h'''(x^*(t)) - 3f''(x^*(t))/P(t) \\ &- 3R\tilde{Y}(t)/P(t) .\end{aligned}$$

The initial states are given through $x^*(t_0) = x_0$, $Z(t_0) = R_0^{-1}$ *and* $\tilde{Y}(t_0) = 0$, *where the expressions* x_0 *and* R_0 *are from the energy function* (2.27).

In a similar way one could proceed and calculate the total time derivative of the fourth-order operator $X(t) := \mathcal{V}''''(x^*, t)$, and the insertion of the Hamilton-Jacobi-Bellman equation to find a fourth-order optimal minimum energy filter. The corresponding expressions become large that is why we omit them.

2.7 Experiments on Higher-Order Filters

In this section we compare the second-order optimal minimum energy filter presented in Theorem 2.4.1 with the third-order optimal minimum energy filter from Theorem 2.6.1. For the sake of simplicity we use the same filtering problems as in the case studies above, i.e. a simple differential equation for the optimal state and a sinusoidal sensor. As reference methods we also compared to the extended Kálmán filter, unscented Kálmán filter and particle filter which were used for the experiments in section 2.5. We plotted sample paths and filter reconstructions of the filters in Fig. 2.3. As one can observe are the reconstructions of the second-order and third-order optimal minimum energy filter very similar, yielding almost the same mean cumulative asymptotic error. Only at some selected areas the methods differ from each other. Although we think that there are problems which require higher-order optimal filtering (e.g. with a non-quadratic energy function), in most cases the second-order optimal filter results in excellent results and represents a good tradeoff between complexity and accuracy.

2.8 Summary

In this chapter we introduced the general filtering problem from stochastic filtering theory with its fundamental solution given by the Kushner-Stratonovich equation which is a partial differential equation (PDE) on the *a posteriori* distribution. Since this PDE cannot be solved exactly we presented state-of-the-art methods that try to find a suitable approximation of this distribution. We briefly discussed the benefits and drawbacks of these filters. We continued with some basics about control theory and derived the second-order optimal minimum energy filter which does not rely on stochastic filtering theory but on optimal control theory. After a complete derivation of the minimum energy filter which is consistent with the general approach of Saccon et al. [73], we compared this filter with extended Kálmán filter, unscented Kálmán filter and particle filter. We demonstrate that the minimum energy filter is as good as the reference filters for simple non-linear scenarios but superior in the case of more involved filtering problems. Finally, we contributed the derivation of a third-order optimal filter for the scalar Euclidean case, extending the work of Mortensen [58]. In experiments we showed that second-order and third-order minimum energy filters differ only slightly from each other, which does not justify the effort of calculating the third-order operator in general. We also think, that calculating third order tensors in scenarios with high-dimensional filtering problems become computationally

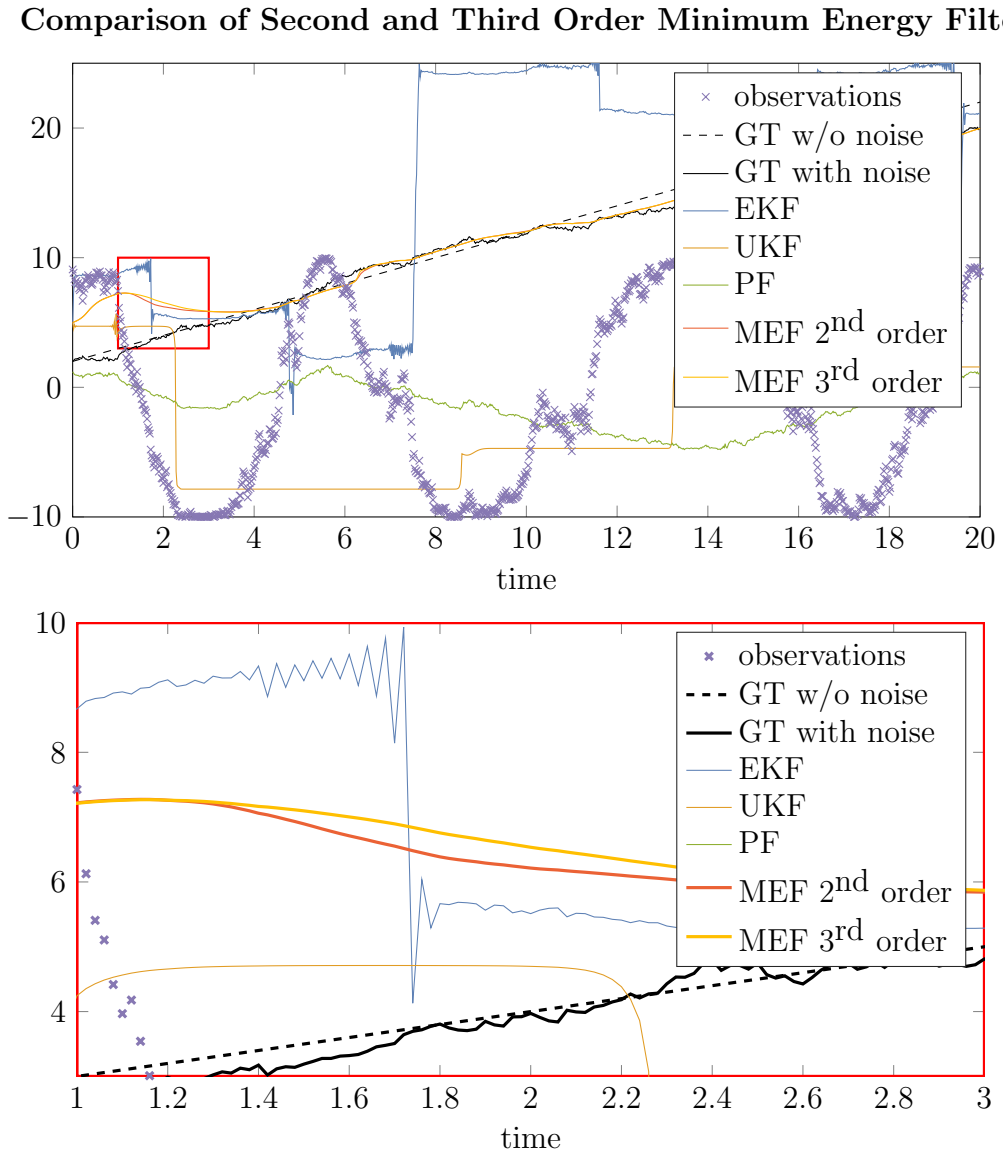


Figure 2.3: Top (full view): typical filter reconstructions of stochastic filters (extended Kálmán filter (EKF), unscented Kálmán filter(UKF) and particle filter (PF)) and minimum energy filter (MEF) of second and third order regarding to a sinusoidal sensor filtering problem, (cf. (2.48) and (2.49) with $h(x) = 10 \cdot \sin(x)$). We observe that the minimum energy filters of different orders return almost exact filter reconstructions. Bottom (detailed view): between the time points 1 and 3 (red box) the minimum energy filter of second and third order differ slightly from each other. However, this little difference does not justify the increased computation effort of the third-order filter, especially in high dimensions.

infeasible. This is the reason why we restrict to second-order optimal filters in the chapters below.

Chapter 3

Differential Geometry

In this chapter we introduce fundamental concepts of differential geometry with focus on Lie groups. Then we briefly summarize the main concepts of the article of Saccon et al. [73] of how to apply the minimum energy filter, which was derived in the last chapter, to the more general case of Lie groups. The resulting second-order optimal filter consists of two coupled differential equations that require to be integrated. Thus, we conclude this chapter with a section about numerical integration on Lie groups and numerical integration of matrix Riccati equations.

3.1 Riemannian Manifolds

The definitions and Theorems of this chapter correspond to the classical definitions from literature. For the introduction of Riemannian manifolds, especially matrix manifolds we used definitions from [1].

3.1.1 Charts, Atlases and Manifolds

Before introducing manifolds we start with the basic definition of topological spaces.

Definition 3.1.1 (topological space)**.** A pair $(\mathcal{M}, \mathcal{A})$ is called a *topological space* with a set $\mathcal{M}$ and a family $\mathcal{A}$ of subsets of $\mathcal{M}$ such that

1. $A_1, A_2 \in \mathcal{A} \implies A_1 \cap A_2 \in \mathcal{A}$,
2. For any index set I holds

$$(A_i, i \in I) \subset \mathcal{A} : \quad \bigcup_{i \in I} A_i \in \mathcal{A},$$

3. $\emptyset, \mathcal{M} \in \mathcal{A}$.

If there exists for each point $p \in \mathcal{M}$ a neighborhood $A \in \mathcal{A}$ and a diffeomorphism $\phi : A \to B$, where B is an open subset of $\mathbb{R}^n$, we call the pair (A, ϕ) a *chart* at p and A is called a *coordinate neighborhood*.

Definition 3.1.2. A (C^∞) atlas is a collection of charts $(A_i, \phi_i)_i$ of the topological space $\mathcal{M}$ such that

1. $\bigcup_i A_i = \mathcal{M}$,
2. for any i, j and $A_i \cap A_j \neq \emptyset$ the charts (A_i, ϕ_i) and (A_j, ϕ_j) are compatible with each other in the following sense: The set $\phi_i(A_i \cap A_j)$ and $\phi_j(A_i \cap A_j)$ are open sets in $\mathbb{R}^d$ and the mapping

$$\phi_j \circ \phi_i^{-1} : \mathbb{R}^d \to \mathbb{R}^d$$

is smooth (C^∞).

An atlas $\mathcal{A}$ can be augmented by all charts (U, ϕ) such that $\mathcal{A} \cup \{(U, \phi)\}$ is still an atlas. This is called a *maximal atlas*. A *smooth manifold* is a couple $(\mathcal{M}, \mathcal{A})$ where $\mathcal{M}$ is a set and $\mathcal{A}$ is a maximal atlas of $\mathcal{M}$ into $\mathbb{R}^d$ where $\mathcal{A}$ is Hausdorff and second-countable. For details we refer to [1].

3.1.2 Tangent Spaces

Most important for optimization on a smooth manifold as introduced above is the notion of tangent vectors at a specific point $p \in \mathcal{M}$. There are different concepts of how to introduce tangent vectors. We will consider tangent vectors as equivalence classes of differentiable curves $\gamma : \mathbb{R} \to \mathcal{M}, t \mapsto \gamma(t)$ with $\gamma(0) = x$ that have the the same differential. However, calculation of this differential requires a vector space structure which is not given on $\mathcal{M}$ itself. Instead one requires a smooth real-valued mapping $f : \mathcal{M} \to \mathbb{R}$ such that $f \circ \gamma : t \mapsto f(\gamma(t))$ is a smooth mapping from $\mathbb{R}$ to $\mathbb{R}$. This leads to the following definition from [1]:

Definition 3.1.3. A tangent vector ξ_x to a manifold $\mathcal{M}$ at a point $x \in \mathcal{M}$ is a mapping from all smooth functions on $\mathcal{M}$ that are defined in a neighborhood of x, denoted by $\mathfrak{F}_x(\mathcal{M})$, such that there exists a curve γ on $\mathcal{M}$ with $\gamma(0) = x$, satisfying

$$\mathbf{D}f(x)[\xi_x] = \mathbf{D}f(x)[\dot{\gamma}(0)] := \frac{\mathrm{d}(f(\gamma(t)))}{\mathrm{d}\,t}\Big|_{t=0}, \tag{3.1}$$

for all $f \in \mathfrak{F}_x(\mathcal{M})$. Such a curve γ is said to realize the tangent vector ξ_x.

The *tangent space* at a point $x \in \mathcal{M}$, denoted by $T_x\mathcal{M}$ is the space of all tangents vectors ξ_x to $\mathcal{M}$ at x. For a manifold $\mathcal{M}$ we denote the tangent bundle by the union of all tangent spaces, which is

$$T\mathcal{M} := \bigcup_{x \in \mathcal{M}} T_x\mathcal{M}. \tag{3.2}$$

Example 3.1.4 (Tangent spaces of matrix Lie groups)**.** We consider the *general linear group* of all quadratic $n \times n$ matrices which are invertible which is denoted by GL_n . Together with the matrix multiplication this gives a matrix Lie group. Since GL_n is an embedded submanifold of $\mathbb{R}^{n\times n}$ which itself is a Euclidean space, the tangent spaces can directly calculated as the limit

$$\gamma'(0) := \lim_{t\to 0} \frac{\gamma(t) - \gamma(0)}{t}\,, \tag{3.3}$$

where $\gamma : \mathbb{R} \to \mathrm{GL}_n$ is a curve with $\gamma(0) = x$. The resulting tangent space is

$$T_x \,\mathrm{GL}_n := \mathbb{R}^{n\times n}\,. \tag{3.4}$$

Below we will introduce subgroups of GL_n that are also considered as embedded submanifolds of $\mathbb{R}^{n\times n}$.

3.1.3 Riemannian Metric

Definition 3.1.5. A differentiable manifold $\mathcal{M}$ equipped with a scalar product $g = g_x : T\mathcal{M} \times T\mathcal{M} \to \mathbb{R}_{\geq 0}$ that varies smoothly on the manifold is called Riemannian manifold. For each tangent space $T_x\mathcal{M} \ni \xi_x, \eta_x$ the Riemannian metric $g_x(\xi_x, \eta_x)$ is an *inner product*, which is a bilinear and symmetric positive form. We use the following notations: $g_x(\xi, \eta) = g(\xi, \eta) = \langle \xi, \eta \rangle = \langle \xi, \eta \rangle_x$.

Riemannian Metric on Embedded Submanifolds If $\mathcal{N}$ is an embedded submanifold of $\mathcal{M}$ and g be a Riemannian metric on $\mathcal{M}$ then the corresponding Riemannian metric $\bar{g} = \bar{g}_x$ on $\mathcal{N}$ is equal to $g = g_x$ since each tangent space $T_x\mathcal{N}$ is an subspace of $T_x\mathcal{M}$. Thus

$$\bar{g}_x(\xi, \eta) = g_x(\xi, \eta) \quad \forall \xi, \eta \in T_x\mathcal{N}.$$

Definition 3.1.6 (Riemannian gradient)**.** The Riemannian gradient, denoted by $\mathbf{D}f(x)$, is defined as the unique elements of $T_x\mathcal{M}$ that satisfies

$$\langle \mathbf{D}f(x), \eta \rangle_x := \mathbf{D}f(x)[\eta], \qquad \forall \eta \in T_x\mathcal{M}, \tag{3.5}$$

where the **D** on the right hand side denotes the directional derivative. Note, that we will use the notation **D** for both, the directional derivative and the Riemannian gradient, or more abstract, for differentials.

Remark 3.1.7. If we have functions of multiple variables, i.e. $f(x_1, \ldots, x_n)$ we denote with $\mathbf{D}_i f(x_1, \ldots, x_n)$ the corresponding (partial) derivative or the (partial) gradient. For expressions we will use $\mathbf{D}_y (x^2 + y)$ to identify the element for which the derivative has to be computed.

Riemannian Gradients on Embedded Submanifolds If $(\mathcal{N}, g)$ is an embedded submanifold of a Riemannian manifold $(\mathcal{M}, g)$ (with the same Riemannian metric), then each tangent vector $\xi \in T_x\mathcal{M}$ can be decomposed in an element of $T_x\mathcal{N}$ and and an element on the normal space of $T_x\mathcal{N}$, denoted by $(T_x\mathcal{N})^\perp$ as follows

$$\xi = \mathbf{P}_x(\xi) + \mathbf{P}_x^\perp(\xi)\,,$$

where $\mathbf{P}_x$ denotes the orthogonal projection onto $T_x\mathcal{N}$ and $\mathbf{P}_x^\perp$ denotes the orthogonal projection onto the normal space of $T_x\mathcal{N}$.
If $f(x)$ is a real-valued function on $\mathcal{N} \subset \mathcal{M}$ we find the Riemannian gradient $\overline{\text{grad}} f(x)$ on the tangent space $T_x\mathcal{N}$ by computing the Riemannian gradient $\text{grad}\, f(x)$ on $T_x\mathcal{M}$ and projecting it onto $T_x\mathcal{N}$, i.e.

$$\overline{\text{grad}} f(x) = \mathbf{P}_x(\text{grad}\, f(x))\,. \tag{3.6}$$

3.1.4 Second-Order Geometry

In order to adapt the notion of the Hessian to a manifold, we introduce affine connections.

Definition 3.1.8. Let $\mathcal{X}(\mathcal{M})$ denote the set of all vector fields on $\mathcal{M}$. An *affine connection* ∇ on a manifold $\mathcal{M}$ is a mapping

$$\nabla : \mathcal{X}(\mathcal{M}) \times \mathcal{X}(\mathcal{M}) \to \mathcal{X}(\mathcal{M}),$$

which is denoted by $(\eta, \xi) \mapsto \nabla_\eta \xi$ and satisfies the following properties.

1. $\mathcal{F}(\mathcal{M})$-linearity in η: $\nabla_{f\eta + g\chi}\xi = f\nabla_\eta \xi + g\nabla_\chi \xi\,,$
2. $\mathbb{R}$-linearity in ξ : $\nabla_\eta(a\xi + b\zeta) = a\nabla_\eta \xi + b\nabla_\eta \xi\,,$
3. Product rule: $\nabla_\eta(f\xi) = (\eta f)\xi + f\nabla_\eta \xi\,,$

where we used $\eta, \chi, \xi, \zeta \in \mathcal{X}(\mathcal{M})$, $f, g \in \mathfrak{F}(\mathcal{M})$, and $a, b \in \mathbb{R}$.

For the introduction of second-order geometry we require the symmetry of a connection to be expressed in a coordinate free way which needs the concept of the *Lie bracket* of two vector fields.

Definition 3.1.9 (Lie bracket for vector fields)**.** Let ξ and η be two vector fields on $\mathcal{M}$ with common open domain U and let $\mathfrak{F}(U)$ denote the space of all real-valued and smooth functions defined on U. The *Lie bracket* $[\xi, \eta]$ is defined as the function $\mathfrak{F}(U)$ into itself defined by

$$[\xi, \eta]f := \xi(\eta f) - \eta(\xi f). \tag{3.7}$$

On matrix manifolds the Lie bracket is simply the matrix commutator $[A, B] = AB - BA$. Below we will also consider the Lie bracket from the point of view of Lie group theory in order to define the *Lie algebra.*
For the definition of a Riemannian Hessian we need to select a specific affine connection with properties that are desirable for a Riemannian calculus, i.e. the properties *symmetry* and *compatibility with the Riemannian metric* which are given through the following Theorem:

Theorem 3.1.10. *On a Riemannian manifold $\mathcal{M}$ there exists an unique affine connection ∇, called the* Levi-Civita connection *or the* Riemannian connection *that satisfies*

1. $\nabla_\eta \xi - \nabla_\xi \eta = [\eta, \xi]$, *(symmetry), and*
2. $\chi\langle \eta, \xi\rangle = \langle \nabla_\chi \eta, \xi\rangle + \langle \eta, \nabla_\chi \xi\rangle$ *(compatibility with the Riemannian metric),*

for all vector fields χ, ξ, η on $\mathcal{M}$.

The existence of this connection was shown by Levi and Cicita [1, Thm. 5.3.1]. In the following we think of the Riemannian connection if we use the notation ∇.

Affine Connections on Embedded Submanifolds Following [1, Prop. 5.3.2] the affine connection $\overline{\nabla}$ on an embedded submanifold $\mathcal{N}$ of a Riemannian manifold $\mathcal{M}$ with Riemannian connection ∇ can be computed by projection, i.e.

$$\overline{\nabla}_{\eta_x} \xi = \mathbf{P}(\nabla_{\eta_x} \xi) \tag{3.8}$$

In this work we will mainly focus on matrix manifolds that are embedded submanifolds of Euclidean spaces. In this case we can compute the affine connection directly by

$$\overline{\nabla}_{\eta_x} \xi = \mathbf{P}_x(\mathbf{D}\xi(x)[\eta]). \tag{3.9}$$

Definition 3.1.11. Let f be a real-valued function on a Riemannian manifold $\mathcal{M}$, then the Riemannian Hessian at a point $x \in \mathcal{M}$ is the linear mapping $\operatorname{Hess} f(x)$ of $T_x\mathcal{M}$ into itself defined by

$$\operatorname{Hess} f(x)[\xi_x] = \nabla_{\xi_x} \operatorname{grad} f \tag{3.10}$$

for all $\xi_x \in T_x\mathcal{M}$, where ∇ denotes the Riemannian connection on $\mathcal{M}$.

With this definition the Riemannian Hessian can also be expressed as

$$\langle \operatorname{Hess} f(x)[\xi], \eta \rangle := \mathbf{D}(\mathbf{D}f(x)[\xi])[\eta] - \mathbf{D}f(x)[\nabla_\eta \xi]. \tag{3.11}$$

3.2 Lie Groups

In this section we will consider specific Riemannian manifolds that contain a smooth group action, which are known as *Lie groups*. In this section we will introduce the main concepts of Lie groups and Lie algebras that are required for the following mathematical treatment of filters on Lie groups. An extensive overview about general Lie groups and representation theory can be found in [43, 19]. An introduction to matrix Lie groups is given in [40]. All definitions and Theorems in this sections are basics from general Lie group theory and are thus provided without references.

3.2.1 Lie Algebras

Before considering Lie groups we start investigating Lie algebras, which can be considered as the infinitesimal counterpart of Lie groups. Most important for studying Lie algebras is the *Lie bracket.*

Definition 3.2.1. Let $\mathfrak{g}$ be a vector space. A *Lie bracket* on $\mathfrak{g}$ is a bilinear map $[\cdot,\cdot] : \mathfrak{g} \times \mathfrak{g} \to \mathfrak{g}$ satisfying

1. $[x, y] = -[y, x]$ for $x, y \in \mathfrak{g}$,
2. $[x, [y, z]] + [y, [z, x]] + [z, [x, y]] = 0$ for $x, y, z \in \mathfrak{g}$ (Jacobi identity).

For any Lie bracket on $\mathfrak{g}$, the pair $(\mathfrak{g}, [\cdot,\cdot])$ is called a *Lie algebra.*

We also require to define the adjoint mapping on Lie groups which are derivations

Definition 3.2.2. Let $\mathfrak{g}$ be a Lie algebra. A linear map $\delta : \mathfrak{g} \to \mathfrak{g}$ is called a *derivation* if

$$\delta([x, y]) = [\delta(x), y] + [x, \delta(y)], \quad \text{for} \quad x, y \in \mathfrak{g}. \tag{3.12}$$

A typical example for derivations is the adjoint representation:

Definition 3.2.3. Let $\mathfrak{g}$ be a Lie algebra. Then we obtain with the Jacobi identity that the linear map

$$\mathrm{ad}_x : \mathfrak{g} \to \mathfrak{g}, y \mapsto \mathrm{ad}_x(y) := [x, y], \tag{3.13}$$

is a derivation.

3.2.2 Lie Groups

Many problems arising in the real world can be described as Lie groups, since problems are usually symmetric in the sense, that they have a forward and an inverse backward action.

Definition 3.2.4. A *Lie group* is a group $\mathcal{G}$ endowed with the structure of a Riemannian manifold such that the group actions

$$m : \mathcal{G} \times \mathcal{G} \to \mathcal{G}, \quad (x, y) \mapsto xy \quad \text{and} \quad i(x) : \mathcal{G} \to \mathcal{G}, \quad x \mapsto x^{-1}, \tag{3.14}$$

are smooth. The neutral element of the Lie group $\mathcal{G}$ is denoted by $\mathrm{Id}_{\mathcal{G}} = xx^{-1}$. We omit the subscript $\mathcal{G}$ when the Lie group is clear from the context.

We will write $L_x : \mathcal{G} \to \mathcal{G}$, $y \mapsto xy$ for the left translation map and $R_x : \mathcal{G} \to \mathcal{G}$, $y \mapsto yx$ for the right translation map.

Example 3.2.5. We consider the following matrix Lie groups

1. Gl_n : general linear group of all invertible $n \times n$ matrices

2. SO_n : the group of rotations in $\mathbb{R}^n$, that consists of all orthogonal matrices with positive determinant 1, i.e.

$$\mathrm{SO}_n := \left\{ R \in \mathbb{R}^{n \times n} | R^\top R = \mathbb{1}_n, \det(R) = 1 \right\}. \tag{3.15}$$

   Further, SO_n is a compact Lie group.

3. SE_n : special euclidean group that consists of all affine motions, with rotations and translations. It is defined through

$$\mathrm{SE}_n := \left\{ \begin{pmatrix} R & w \\ \mathbf{0}_{1\times n} & 1 \end{pmatrix} \in \mathbb{R}^{(n+1)\times(n+1)} \middle| R \in \mathrm{SO}_n, w \in \mathbb{R}^n \right\}. \tag{3.16}$$

3.2.3 Left Invariant Vector Fields

The basic tool of Lie group theory is to translate problems on the Lie group to problems to the corresponding Lie algebra. To do this we require *left-invariant* vector fields. A fundamental Lemma of Lie group theory is that the tangent map of the group action also defines a group action on the tangent bundle $T\mathcal{G}$ of $\mathcal{G}$ which can be found in [43, Lemma 9.6.1]:

Lemma 3.2.6. *The tangent map of the left translation L*

$$T(L) : T(\mathcal{G} \times \mathcal{G}) \cong T\mathcal{G} \times T\mathcal{G} \to T\mathcal{G}, \quad (\xi, \eta) \mapsto TL(\xi, \eta)\,,$$

defines a Lie group structure on the tangent bundle $T\mathcal{G}$ with identity element $0_{\mathrm{Id}} \in T_{\mathrm{Id}}\mathcal{G}$. The map

$$\Phi : \mathcal{G} \times T_{\mathrm{Id}}(\mathcal{G}) \to T\mathcal{G}, \quad (x, \eta) \mapsto x\eta = T_{\mathrm{Id}}L_x\eta := TL_x\eta(\mathrm{Id})\,,$$

is a diffeomorphism.

Proof. See [43, Lemma 9.6.1]. □

Definition 3.2.7. A vector field $\xi \in \mathcal{X}(\mathcal{G})$, where $\mathcal{X}(\mathcal{G})$ denotes the set of smooth vector fields on $\mathcal{G}$, is called *left-invariant if*

$$\xi = (L_x)_*\xi := T_{\mathrm{Id}}L_x \circ \xi \circ L_x^{-1} \tag{3.17}$$

holds for each $x \in \mathcal{G}$.

The left-invariance of the vector field implies together with Lemma 3.2.6 that for each $x \in \mathcal{G}$ we have $\xi(x) = x\xi = T_{\mathrm{Id}}L_x\xi = TL_x\xi(\mathrm{Id})$, such that the vector field ξ is completely determined by its value $\xi(\mathrm{Id}) \in T_{\mathrm{Id}}\mathcal{G}$. Conversely, we obtain for each $\eta \in T_{\mathrm{Id}}\mathcal{G}$ a left-invariant vector field ξ with $\xi(\mathrm{Id}) = \eta$ by the relation $\xi(x) = T_{\mathrm{Id}}L_x\eta$. The left-invariance follows from

$$\xi \circ L_y x = \xi(yx) = (yx)\eta = y(x\eta) = T_{\mathrm{Id}}L_y\xi(x)\,, \tag{3.18}$$

for all $x, y \in \mathcal{G}$. From this follows that the mapping from $T_{\mathrm{Id}}\mathcal{G}$ to all left-invariant vector fields on $\mathcal{G}$ given through $\eta \mapsto \xi$ is a linear bijection. From this fact we obtain the Lie bracket $[\cdot, \cdot]$ on $T_{\mathrm{Id}}G$ which satisfies the relation

$$T_{\mathrm{Id}}L_x[\xi, \eta] = [T_{\mathrm{Id}}L_x\xi, T_{\mathrm{Id}}L_x\eta] \quad \text{for all } \xi, \eta \in T_{\mathrm{Id}}\mathcal{G} \text{ and } x \in \mathcal{G}. \tag{3.19}$$

Definition 3.2.8. The Lie algebra $\mathfrak{g} := (T_{\mathrm{Id}}\mathcal{G}, [\cdot, \cdot])$ which is associated with the left-invariant vector fields is called *the Lie algebra of $\mathcal{G}$*.

In the next chapters below we will often use this fact to pull operations on tangent spaces back to operations on the Lie algebra.

Example 3.2.9. We continue the example 3.2.5:

1. The Lie algebra of GL_n is the set of all $n \times n$ matrices, i.e. $\mathfrak{g}l_n := \mathbb{R}^{n\times n}$.
2. The Lie algebra to SO_n is the set of all skew-symmetric matrices
$$\mathfrak{so}_n := \{X - X^\top | X \in \mathbb{R}^{n\times n}\}.$$
3. The Lie algebra of SE_n is
$$\mathfrak{se}_n := \left\{ \begin{pmatrix} \omega & t \\ \mathbf{0} & 0 \end{pmatrix} \middle| \omega \in \mathfrak{so}_n, t \in \mathbb{R}^n \right\}. \tag{3.20}$$

By definition the Lie algebra $\mathfrak{g}$ can also be understood as the tangent space at the identity element $\mathrm{Id}_\mathcal{G}$ of the Lie group $\mathcal{G}$. Since $\mathfrak{g}$ has the structure of a vector-space, for a $n-$dimensional Lie group $\mathcal{G}$ there exists a linear and invertible mapping $\mathrm{vec}_\mathfrak{g} : \mathfrak{g} \to \mathbb{R}^n$ such that each element in $\mathfrak{g}$ can be uniquely represented as a vector in $\mathbb{R}^n$. The inverse operation is denoted by $\mathrm{mat}_\mathfrak{g} := \mathrm{vec}_\mathfrak{g}^{-1}$. As this mapping is not unique we will define the corresponding maps in the following chapters for the required Lie groups.

3.2.4 The Exponential Map

On of the most important structure in Lie group theory is the so called *exponential map*, that maps a tangent vector in the Lie algebra to an element of the Lie group.

Definition 3.2.10. We define the exponential function on a Lie group $\mathcal{G}$ as in [43, sec. 9.2] through

$$\mathrm{Exp}_\mathcal{G} : \mathfrak{g} \to \mathcal{G}, \quad \mathrm{Exp}_\mathcal{G}(\eta) := \gamma_\eta(1), \tag{3.21}$$

where $\gamma_\eta : \mathbb{R} \to \mathcal{G}$ is the unique maximal integral curve of the left invariant vector field η_l, satisfying $\gamma_\eta(0) = \mathrm{Id}_\mathcal{G}$. This means that γ_η is the unique solution of the initial value problem

$$\gamma(0) = \mathrm{Id}_\mathcal{G}, \quad \dot\gamma(t) = T_{\mathrm{Id}} L_{\gamma(t)} \eta, \quad \text{for all } t \in \mathbb{R}. \tag{3.22}$$

Here $T_{\mathrm{Id}} L_x$ denotes the tangent map of the left translation of x evaluated at identity.

Remark 3.2.11. We also require the so called *logarithmic map* denoted by $\mathrm{Log}_{\mathcal{G}} : \mathcal{G} \to \mathfrak{g}$ which is the inverse of the exponential map, i.e. $\mathrm{Log}_{\mathcal{G}} := \mathrm{Exp}_{\mathcal{G}}^{-1}$.

Example 3.2.12. Let $\mathcal{G} := \mathrm{GL}_n(\mathbb{R})$ the Lie group of invertible $n \times n$ matrices. The tangent map of the left translation of x at identity is given through

$$T_{\mathrm{Id}} L_x \eta = \mathbf{D}_y L_x(y)[\eta]|_{y=\mathrm{Id}} = x\eta. \tag{3.23}$$

The unique solution $\gamma_\eta(t)$ of the initial value problem

$$\gamma_\eta(0) = 1, \quad \dot{\gamma}_\eta = \gamma(t)\eta\,, \tag{3.24}$$

is given by the fundamental system of linear differential equations in terms of the matrix exponential, i.e.

$$\gamma_\eta(t) = e^{t\eta} = \sum_{k=0}^{\infty} t^k \frac{\eta^k}{k!}\,. \tag{3.25}$$

This exponential series converges for any matrix η. Thus, the exponential map of the Lie group $\mathrm{GL}_n(\mathbb{R})$ coincides with the usual matrix exponential, i.e. $\mathrm{Exp}_{\mathrm{GL}_n(\mathbb{R})}(\eta) = e^\eta$. This is the reason for the term "exponential map".

3.2.5 Product Lie Groups

In this section we investigate product Lie groups with their induced product group action. For this purpose let $\mathcal{G}_1, \ldots, \mathcal{G}_n$ be Lie groups and $\mathcal{G} := \mathcal{G}_1 \times \cdots \times \mathcal{G}_n$ endowed with the direct product group structure

$$(x_1, \ldots, x_n)(y_1, \ldots, y_n) := (x_1 y_1, \ldots, x_n y_n)\,. \tag{3.26}$$

Proposition 3.2.13. *$\mathcal{G} := \mathcal{G}_1 \times \cdots \times \mathcal{G}_n$ is a Lie group with Lie algebra*

$$\mathfrak{g} = \mathfrak{g}_1 \times \cdots \times \mathfrak{g}_n \tag{3.27}$$

where $\mathfrak{g}_1, \ldots, \mathfrak{g}_n$ denote the Lie algebras of $\mathcal{G}_1, \ldots, \mathcal{G}_n$.

Proof. By the definition of the group action as direct product we see that $\mathcal{G}$ is a Lie group with identity element $\mathrm{Id}_{\mathcal{G}} = (\mathrm{Id}_{\mathcal{G}_1}, \ldots, \mathrm{Id}_{\mathcal{G}_n})$. Since we have a product Riemannian manifold, the tangent spaces also factorize, i.e. $T_{\mathrm{Id}}\mathcal{G} = T_{\mathrm{Id}_{\mathcal{G}_1}}\mathcal{G}_1 \times \cdots \times T_{\mathrm{Id}_{\mathcal{G}_n}}\mathcal{G}_n$. It remains to calculate the Lie bracket by determining the left-invariant vector fields on $\mathcal{G}$. By definition of left invariant vector fields we obtain for each $\eta = (\eta_1, \ldots, \eta_n) \in T_{\mathrm{Id}}\mathcal{G}$ a vector field $\xi \in \mathcal{X}(\mathcal{G})$ with

$\xi(x) = T_{\mathrm{Id}} L_x \xi$. The left-translation of the tangent map at identity of $\mathcal{G}$ can be calculated as

$$\begin{aligned} T_{\mathrm{Id}} L_x \xi =& T_{(\mathrm{Id}_{\mathcal{G}_1},\dots,\mathrm{Id}_{\mathcal{G}_n})} L_{(x_1,\dots,x_n)}(\xi_1,\dots,\xi_n) \\ =& (T_{\mathrm{Id}} L_{x_1}\xi_1,\dots,T_{\mathrm{Id}} L_{x_n}\xi_n). \end{aligned}$$

Thus, the Lie bracket on $\mathcal{G}$ also fulfills the equality

$$x[\eta,\chi] = ([x_1\eta_1, x_1\chi_1],\dots,[x_n\eta_n, x_n\chi_n])$$

for $\eta, \chi \in T_{\mathrm{Id}}\mathcal{G}$. It follows that the Lie algebra is by definition given through the product topology, i.e. $\mathfrak{g} = \mathfrak{g}_1 \times \cdots \mathfrak{g}_n$. □

Example 3.2.14. In the chapters below we consider different product Lie groups. On of them is the group $\mathcal{G} = \mathrm{SE}_3 \times \mathbb{R}^6$ with the group action which is defined for $x_1 = (E_1, v_1)$, $x_2 = (E_2, v_2) \in \mathcal{G}$ through

$$(x_1, x_2) \mapsto (E_1E_2, v_1 + v_2) \in \mathcal{G}\,. \tag{3.28}$$

In Fig. 3.1 we depicted this Lie group (represented as a sphere) and also inserted the relations to the tangent map and the exponential map on $\mathcal{G}$. We can also define the following Riemannian metric and the Riemannian gradient on this group using the product topology as follows:

Riemannian metric on product Lie group. On SE_3 as submanifold of GL_4, the Riemannian metric at $E \in \mathrm{SE}_3$ for $\xi, \eta \in T_E\,\mathrm{SE}_3$ is given by $\langle \xi, \eta\rangle_E := \langle E^{-1}\xi, E^{-1}\eta\rangle_{\mathbb{1}_4}$ where $\langle A, B\rangle_{\mathbb{1}_4} := \mathrm{tr}(A^\top B)$ is the usual inner matrix product. Thus the Riemannian metric on $T_x\mathcal{G}$ for two tangent vectors $\xi = (E\xi_1, \xi_2), \eta = (E\eta_1, \eta_2) \in T_x\mathcal{G} = T_E\,\mathrm{SE}_3 \times \mathbb{R}^6$ is given through

$$\langle \xi, \eta\rangle_x = \langle E\xi_1, E\eta_1\rangle_E + \langle \xi_2, \eta_2\rangle = \langle \xi_1, \eta_1\rangle_{\mathbb{1}_4} + \langle \xi_2, \eta_2\rangle, \tag{3.29}$$

where we used the components $x = (E, v) \in \mathcal{G}$.

Riemannian Gradient. For a real-valued function $f : \mathcal{G} \to \mathbb{R}$, the Riemannian gradient $\mathbf{D}f(x)$ is defined through the relation $\langle \mathbf{D}f(x), \eta\rangle_x := \mathbf{D}f(x)[\eta]$ for all $\eta \in T_x\mathcal{G}$. For the product Lie group $\mathcal{G} = \mathrm{SE}_3 \times \mathbb{R}^6$ and $x = (E, v) \in \mathcal{G}, \eta = (E\eta_1, \eta_2) \in T_x\mathcal{G}$ we calculate the Riemannian gradient as follows:

$$\begin{aligned} &\mathbf{D}f(x)[\eta] = \langle \mathbf{D}f(x), \eta\rangle_x \\ =& \langle E^{-1}\mathbf{D}_E f((E,v)), \eta_1\rangle_{\mathbb{1}_4} + \langle \mathbf{D}_v f((E,v)), \eta_2\rangle\,, \end{aligned}$$

where $\mathbf{D}_E f((E, v))$ is the partial Riemannian gradient on SE_3 and $\mathbf{D}_v f((E, v))$ is the Euclidean partial gradient on $\mathbb{R}^6$.

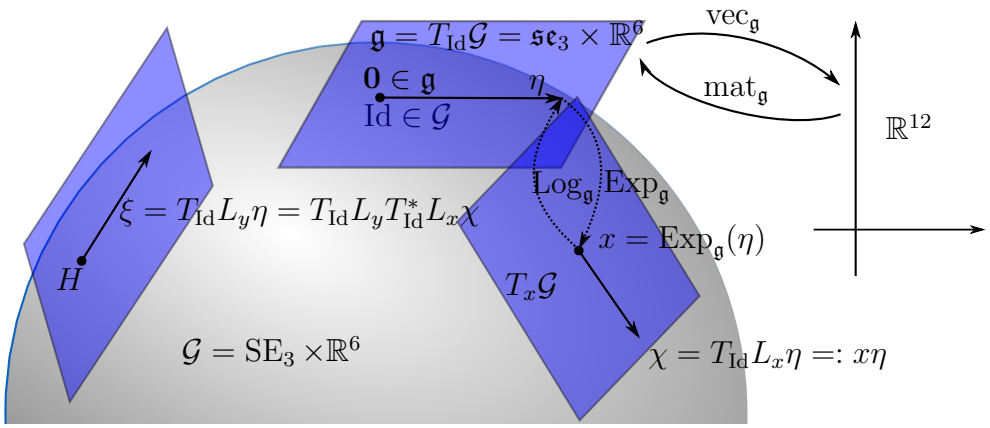


Figure 3.1: Illustration of the Lie group $\mathcal{G}$ (represented as sphere) with its Lie algebra $\mathfrak{g}$ and tangent spaces at different points. A tangent vector χ at a point x can be expressed as tangent map at identity of the left translation at x of a vector $\eta \in \mathfrak{g}$, i.e. $\chi = x\eta$. Since the Lie algebra $\mathfrak{g}$ can be identified by the $\mathrm{vec}_{\mathfrak{g}}$ mapping with $\mathbb{R}^{12}$, we can express each tangent vector at a point x as a pair $(x, \mathrm{vec}_{\mathfrak{g}}(\eta))$. Each tangent vector on any tangent space may be mapped to the manifold using the exponential map Exp.

3.2.6 Levi-Civita Connection on Lie Groups

Since each Lie group is a Riemannian manifold it is naturally equipped with the Levi-Civita connection ∇. Due to the structure of Lie groups it is additionally possible to express the Levi-Civita connection in terms of the *connection function* which is defined through

$$x\omega(\xi, \eta) := x\nabla_\xi \eta = \nabla_{x\xi} x\eta\,, \tag{3.30}$$

for all $x \in \mathcal{G}$ and tangent vectors $\xi, \eta \in \mathfrak{g}$. We also introduce the "swap"-operator that interchanges the arguments, i.e $\omega^{\leftrightharpoons}_\xi \eta := \omega_\eta \xi$.

3.2.7 Duality on Lie Groups

Tangential spaces on Riemannian manifolds and especially on Lie groups are linear spaces, endowed with a scalar product – the Riemannian metric. Thus, these tangent spaces of finite-dimensional Lie groups are finite-dimensional Hilbert spaces, which imply duality: In this work we denote by $T_x\mathcal{G}^*$ the dual space of the tangent space of $T_x\mathcal{G}$ at a point $x \in \mathcal{G}$. For a Lie group $\mathcal{G}$ we also have the dual of the Lie algebra $\mathfrak{g}$ denoted by $\mathfrak{g}^*$. The dual space $T_x\mathcal{G}^*$ denotes the set of all linear mappings from $T_x\mathcal{G}$ to $\mathbb{R}$. For $\mu \in T_x\mathcal{G}^*$ and $\eta \in T_x\mathcal{G}$

we use the natural pairing $\langle \mu, \eta \rangle_x = \mu(\eta)$. Below we use the identification of the Lie algebra $\mathfrak{g}$ with the bidual $\mathfrak{g}^{**}$. We say that a operator $\phi : \mathfrak{g} \to \mathfrak{g}^*$ is symmetric with respect to the Riemannian metric (dual pairing) if $\phi = \phi^*$. Using the duality concept we introduce the following operators on the connection function in (3.30). The dual of the connection function $\omega_\xi^* : \mathfrak{g}^* \to \mathfrak{g}^*$ is defined through the relation $\langle \omega_\xi^*(\mu), \eta \rangle := \langle \mu, \omega_\xi \eta \rangle$. Additionally we require expressions that were introduced in [73]: The swapped dual of the connection function is given through $\omega_\mu^{*\leftrightharpoons} : \mathfrak{g} \to \mathfrak{g}^*$ which is defined by $\langle \omega_\mu^{*\leftrightharpoons}(\xi), \eta \rangle := \langle \omega_\xi^*(\mu), \eta \rangle = \langle \mu, \omega_\xi \eta \rangle$ as well as the dual of the "swap"-operator $\omega_\eta^{\leftrightharpoons *} : \mathfrak{g}^* \to \mathfrak{g}^*$ defined by the relation $\langle \omega_\eta^{\leftrightharpoons *} \mu, \xi \rangle := \langle \mu, \omega_\eta^{\leftrightharpoons} \xi \rangle = \langle \mu, \omega_\xi \eta \rangle$.

3.3 The Minimum Energy Filter on Lie Groups

In this section we summarize the main ideas that are required for the generalization of the minimum energy filter for Lie groups. These include the concept of left-trivialization of the appearing expression which allow calculations on the Lie algebra. We follow the approach of Saccon et al. [72, 73]. Instead of providing the full derivation of the second-order optimal filter, we will only focus on the main results. We begin with the problem formulation and the definition of the energy function and proceed with the solution of the corresponding optimal control problem. Finally we demonstrate the recursive description of the optimal state and the corresponding second-order operator which results in the minimum energy filter.

3.3.1 Filtering Problem

In opposite to the filtering problem on an Euclidean space which was formulated in (2.25) and (2.26) we require a different formulation on Lie groups. The corresponding differential equations can be expressed be means of the tangent map of the left translation in terms of left invariant vector fields as follows. This results in the following state equation:

$$\dot{x}(t) = x(t)\big(f(x(t)) + \delta(t)\big)\,. \tag{3.31}$$

The idea behind this representation is, that each tangent vector $\eta \in T_x\mathcal{G}$ can be expressed in terms of an element of the Lie algebra $\xi \in \mathfrak{g}$ and the tangent map of the left translation of x at identity, i.e. $\eta = x\xi = T_{\mathrm{Id}} L_x \xi$. Thus the functions $f : \mathcal{G} \to \mathfrak{g}$ as well as the noise term δ map onto an element of the

Lie algebra. The observation stays unchanged in comparison to the euclidean case, i.e.

$$y(t) = h(x(t)) + \epsilon(t)\,. \tag{3.32}$$

3.3.2 Objective Function

The objective function can be formulated similarly to the energy function on the Euclidean space. However, instead of δ we use its vectorized representation within the quadratic form. The final energy function reads

$$\mathcal{J}(\delta, \epsilon, t; t_0) := m_0(x) + \int_{t_0}^{t} \|\delta(\tau)\|_R^2 + \|\mathrm{vec}_{\mathfrak{g}}(\epsilon(\tau))\|_Q^2 \, d\tau\,. \tag{3.33}$$

In this energy function $\|\mathrm{vec}_{\mathfrak{g}}(\delta(\tau))\|_R^2$ and $\|\epsilon(\tau)\|_Q^2$ denote quadratic forms on the noise processes with symmetric and positive definite matrices R and Q. The *initial value* function m_0 is a smooth and bounded function with an unique minimum x_0 on $\mathcal{G}$. The only difference to the energy in (2.27) consists of the vectorization operator for δ.

3.3.3 Optimal Control Problem

We continue with the replacement of the observation noise $\epsilon = \epsilon(t)$ by the residual

$$\epsilon(t) = \epsilon(t, x(t)) := y(t) - h(x(t))\,, \tag{3.34}$$

that is gained from the observation equation (3.32). Insertion of this expression into the energy function (3.33) gives the modified energy that now depends on x rather than ϵ

$$\mathcal{J}(\delta, x, t; t_0) = m_0(x) + \int_{t_0}^{t} \|\mathrm{vec}_{\mathfrak{g}}(\delta(\tau))\|_R^2 + \|y(\tau) - h(x(\tau))\|_Q^2 \, d\tau\,. \tag{3.35}$$

As we already included the observation equation (3.32) into this energy it remains to minimize it with respect to δ and x as well as the state equation (3.31). Below we address this problem in terms of optimal control and interpret the model noise as a control parameter. As every (admissible) control $\delta = \delta(t)$ generates a trajectory of x, we first minimize the energy function with respect to δ and such that the state equation (3.31) is fulfilled. From the point of view of optimal control we want to obtain the *value function* which returns the value of the energy function of the optimal control δ^*, i.e.

$$\mathcal{V}(x, t) := \inf_{\delta|_{[t_0,t]}} \mathcal{J}(\delta, x, t; t_0) \qquad \text{subject to (3.31)}\,. \tag{3.36}$$

Finding the optimal control on the Lie algebra is mathematically involved and requires geometric control theory and a geometric formulation of the Pontryagin maximum (minimum principle) (cf. [48, chapter 11]). As the proof of this maximum principle is already elongated for the scenario of Euclidean space it will be even more involved on Lie groups since one requires the so called *symplectic structure of the cotangent bundle*. Even textbooks provide a simplified proof of the maximum principle, cf. [5, chapter 5]. Therefore we will not provide the proofs of the maximum principle; it is beyond the scope of this work. Instead we refer to the results of geometric control theory and the work of Saccon et al. [72]. The (time-varying) Hamiltonian $\tilde{\mathcal{H}} : T^*\mathcal{G} \times \mathfrak{g} \times \mathbb{R} \to \mathbb{R}$ for the above control problem (3.36) is given through

$$\tilde{\mathcal{H}}(p, \delta, t) := \|\mathrm{vec}_{\mathfrak{g}}(\delta(t))\|_R^2 + \|y(t) - h(x(t))\|_Q^2 - \langle p(t), T_{\mathrm{Id}} L_x(f(x(t)) + \delta(t))\rangle_x \,. \tag{3.37}$$

Instead of the scalar product in the Euclidean, the Riemannian metric is used here. Thus the costate variable p (which can be interpreted as a Lagrangian multiplier) lies in the dual of the tangent space $T_x\mathcal{G}$, i.e. $p \in T_x\mathcal{G}^*$. The key concept of the approach of [72] is to trivialize this Hamiltonian such that is is independent of the point $x \in \mathcal{G}$. Each element of the cotangent bundle $p \in T_x\mathcal{G}^*$ can be represented uniquely by a $\mu \in \mathfrak{g}^*$ by means of left translation such that $\mu = T_{\mathrm{Id}} L_x^* p$ resulting in the left-trivialized Hamiltonian $\tilde{\mathcal{H}}^- : \mathcal{G} \times \mathfrak{g}^* \times \mathbb{R}^n \times \mathfrak{g} \to \mathbb{R}$ proposed in [72, Eq. (19)].

$$\tilde{\mathcal{H}}^-(x, \mu, \delta, t) = \|\mathrm{vec}_{\mathfrak{g}}(\delta(t))\|_R^2 + \|y(t) - h(x(t))\|_Q^2 - \langle \mu(t), f(x(t)) + \delta(t)\rangle_{\mathrm{Id}} \,. \tag{3.38}$$

Using the *geometric Pontryagin minimum principle* [48, chapter 11], the value function can be found by minimization of the Hamiltonian (3.38). Since this Hamiltonian is convex in δ, we obtain a unique solution δ^*, of the control problem (3.36). Insertion of this solution into the (time-varying) Hamiltonian gives the *optimal* Hamiltonian

$$\mathcal{H}(x, \mu, t)^- := \tilde{\mathcal{H}}^-(x, \mu, \delta^*, t) \,. \tag{3.39}$$

As in the case of an Euclidean state space we require the Hamilton-Jacobi-Bellman equation for the recursive filtering principle in the sequel. This differs from the classical theory and needs to be generalized to Riemannian manifolds. The corresponding *left-trivialized Hamilton-Jacobi-Bellman equation* (HJB) provided by [72] is

$$\frac{\partial}{\partial t}\mathcal{V}(x, t) - \mathcal{H}(x, T_{\mathrm{Id}} L_x^*(\mathbf{D}_1 \mathcal{V}(x, t)), t) = 0. \tag{3.40}$$

Most of the other concepts of the minimum energy filter presented in the last chapter stay almost unchanged: After starting with the necessary condition

for optimality, i.e.

$$\mathbf{D}_1 \mathcal{V}(x^*, t) = 0\,, \tag{3.41}$$

one can calculate the total time derivation of the condition (3.41). Then the HJB equation (3.40) is inserted as an optimality criterion, which includes the constraints of the dynamical system (3.31). After truncation of higher-order derivatives and a variable substitution one gains the following two differential equations, cf. [73]:

$$\dot{x}^*(t) = T_{\mathrm{Id}} L_{x^*(t)} \big(f(x^*(t)) + K(t) r_t(x^*) \big), \quad x^*(t_0) = x_0\,, \tag{3.42}$$

where $K(t) : \mathfrak{g}^* \to \mathfrak{g}$ is the second-order operator satisfying the operator Riccati equation below. The residual $r_t(x^*)$ is given through

$$r_t(x^*) = T_{\mathrm{Id}} L^*_{x^*(t)} \Big(Q(y(t) - h(x^*(t))) \circ \mathbf{D}h(x^*(t)) \Big)\,, \tag{3.43}$$

where Q is the weighting matrix in the energy function (3.35). The second-order optimal symmetric gain operator $K(t)$ fulfills the following Riccati equation

$$\dot{K}(t) = A \circ K + K \circ A^* - K \circ E \circ K + \mathrm{mat}_{\mathfrak{g}} \circ R^{-1} \circ \mathrm{vec}_{\mathfrak{g}} - \omega_{Kr} \circ K - K \circ \omega^*_{Kr}\,, \tag{3.44}$$

with initial condition $K(t_0) = \big(T_{\mathrm{Id}} L^*_{x^*(t_0)} \circ \mathrm{Hess}\, m_0(x^*(t_0)) \circ T_{\mathrm{Id}} L_{x^*(t)} \big)^{-1}$. The operators $A(t) : \mathfrak{g} \to \mathfrak{g}$ and $E(t) : \mathfrak{g} \to \mathfrak{g}^*$ are given by

$$A(t) = \mathbf{D}_1 f(x^*(t)) \circ T_{\mathrm{Id}} L_{x^*(t)} - \mathrm{ad}_{f(x^*(t))} - T_{f(x^*(t))}\,, \tag{3.45}$$

$$\begin{aligned} E(t) = &- T_{\mathrm{Id}} L^*_{x^*(t)} \circ \Big(\big(Q(y(t) - h(x^*(t)) \big)^{T_{x^*}\mathcal{G}} \Big) \circ \mathrm{Hess}\, h(x^*(t)) \\ &\circ (\mathbf{D}h(x^*(t)))^* Q \circ Dh(x^*(t)) \Big) \circ T_{\mathrm{Id}} L_{x^*}\,, \end{aligned} \tag{3.46}$$

where ω is the connection function, defined in (3.30) and Kr is a shorthand $K(t)r_t(x^*)$. The functions ad. and $T.$ denote the adjoint representation and the torsion function associated with the connection function ω. The expression $(\cdot)^W$ is the so-called *exponential functor* which maps a linear map $\phi : U \to V$ to the linear map $\phi^W : \mathfrak{L}(W, U) \to \mathfrak{L}(W, V)$ defined by $\phi^W(\eta) = \phi \circ \eta$. For details on functor theory of Lie groups we refer the reader to [43, chapter 9.1].
Since the details of the above expression are involved we refer the reader to the original article of Saccon et al. [73] where also the proofs can be found. In the chapters below we will adapt this second-order optimal filter to specific problems of image processing and scene understanding, where we will derive the expressions in (3.42) and (3.44) explicitly.

3.4 Numerical Integration on Lie Groups

An overview about numerical integration on Lie groups can be found in [39] for right invariant vector fields. However, we will consider only Lie groups that are generated by left invariant vector fields to be consistent with our main reference [72]. Therefore we introduce in this section the adapted numerical methods for Lie groups that are generated by left invariant vector fields. We distinguish between explicit and implicit methods that are used for solving the following differential equation on a Lie group $\mathcal{G}$ with Lie algebra $\mathfrak{g}$

$$\dot{x}(t) = T_{\mathrm{Id}} L_{x(t)} f(x(t), t) \qquad x(t_0) = x_0 \,, \tag{3.47}$$

where $f : \mathcal{G} \to \mathfrak{g}$ is a continuous function. $T_{\mathrm{Id}} L_x f(x) =: xf(x)$ is the tangent map of the left translation of x evaluated at the identity element Id of $\mathcal{G}$.

3.4.1 Explicit Schemes

Similarly to explicit Runge-Kutta methods which act on Euclidean spaces one can apply the corresponding concepts to Lie group. The numerical integration consists of two steps: The propagation of the vector field and an update step. *Crouch-Crossman* methods generalize these operations to Lie groups by using the exponential map to project the "update"-step onto the Lie group. The first stage Crouch-Crossman method is given by

$$x_{k+1} = x_k \operatorname{Exp}_{\mathcal{G}}(\Delta f(x_k)) \,, \tag{3.48}$$

where $\Delta = t_{k+1} - t_k$ denote the integration step size. This methods corresponds to a simple explicit Euler scheme on an Euclidean space. On Lie groups this scheme is also known as *Lie-Euler method*. The second stage Crouch-Crossman method reads

$$x_{k+1} = x_k \operatorname{Exp}_{\mathcal{G}}\left(\tfrac{\Delta}{2} K_1\right) \operatorname{Exp}_{\mathcal{G}}\left(\tfrac{\Delta}{2} K_2\right) , \tag{3.49}$$

where $K_1 = f(x_k)$ and $K_2 = f(x_k \operatorname{Exp}_{\mathcal{G}}(\Delta K_1))$. One can continue these procedures to third order stages. As remarked in [39], the construction of "algorithms with order greater than three" will be very complex.

3.4.2 Implicit Schemes

Based on *Munthe-Kaas* methods that are based on the *Magnus Series Expansion* one can derive implicit numerical integration procedures. The simplest of the implicit integration schemes is the *Lie midpoint rule* which is given by

$$x_{k+1} = x_k \operatorname{Exp}_{\mathcal{G}}(\Omega), \quad \Omega = \Delta f(x_k \operatorname{Exp}(\tfrac{\Omega}{2})) \,. \tag{3.50}$$

The value of Ω can be found by fixed point iterations.

3.4.3 Numerical Integration of Matrix Riccati Equations

The minimum energy filter that we derived within the last chapter requires to integrate the matrix Riccati equation (2.29), which is a second-order matrix differential equation on the second-order operator P. Because P is the matrix representation of the inverse Hessian of the value function for the optimal state x^*, it is symmetric and positive definite. Thus, it is important that this property is preserved during numerical integration. It was shown in [27] that positive definiteness is preserved if the numerical integration scheme (explicit or implicit) has at most order one. Therefore we propose to use an explicit or implicit first order scheme. The explicit scheme can be derived relatively simple but the implicit scheme of a matrix Riccati equation which is given through the solution of

$$P_k = S_k + A_k P_k + P_k A_k^\top - P_k H_k P_k \tag{3.51}$$

requires to solve an *algebraic Riccati equation*. Therefore the equation (3.51) can be reformulated into the following canonical form:

$$S_k + (A_k - \tfrac{1}{2}\mathbb{1})P_k + P_k(A_k - \tfrac{1}{2}\mathbb{1})^\top - P_k E_k P_k = \mathbf{0}\,. \tag{3.52}$$

If $E_k \in \mathbb{R}^{n\times n}$ has a low rank and thus can be decomposed into $E_k = F_k^\top F_k$ with $F_k \in \mathbb{R}^{m\times n}$ with $m \ll n$, the equation (3.52) can be solved exactly. Typical method for solving (3.52) are based on *Krylov-subspace* methods [11] or *Kleinmann iterations* [10]. For dense situations where $m = n$, implicit methods for solving the algebraic Riccati equation cannot be applied. The corresponding optimization problem is infeasible for large n. We propose to use explicit numerical integration schemes instead.

Chapter 4

Recursive Filtering on SE_3

4.1 Introduction

4.1.1 Overview and Motivation

Camera motion estimation is a fundamental task in many important applications in computer vision (e.g. autonomous driving, robotics). It is an essential component of structure-from-motion, simultaneous localization and mapping (SLAM) and odometry tasks. Furthermore it aids as additional prior for e.g. optical flow methods. In the proposed approach, the ego-motion of the camera is fully determined solely by the apparent motion of visual features (optical flow) as recorded by the camera without needing additional sensors such as GPS or acceleration sensors.

Although the camera motions can be reconstructed correctly from only two consecutive frames [62, 41], the best performing methods take into account multiple frames. They are more robust against the influence of erroneous correspondence estimates. Two approaches to making use of the temporal context can be distinguished: batch approaches – such as bundle adjustment methods [80] – first record all the frames and fit in a smooth camera path afterwards. They sometimes also incorporate loop closure constraints [85] to further improve camera motion accuracy. Factorization methods [78, 60] create the problem of jointly estimating camera poses and scene points as a matrix decomposition problem. These batch approaches have the potential of working exactly as they make use of all available information. On the other hand, they hardly work in real-time applications, as the volume of incorporated information increases linearly with time.

In contrast, *online* approaches apply sliding window techniques [6, 9, 16] that track features on multiple frames to increase robustness and compute the best fitting motion.

A mathematical description of (online) temporal smoothing is given by the notion of (stochastic) filtering [7]. In case of an ODE describing the behavior of a latent variable, and observations that depend on the latent variable, the goal is to estimate the most likely value of the unknowns. However, stochastic filters suffer from non-linear dependencies of latent variables and observations as well as geometric constraints and unknown probability distributions.
That is the reason why we chose deterministic *Minimum Energy Filters* that do not need information about distributions and cope with the non-linearities of the observer equation and the geometry of the state space SE_3 in [13]. Since the state equation of the ego-motion in [13] is simple and requires small weights on the penalty term for the model noise, however, this approach is sensitive against noise and requires good observation data.
Therefore, in this chapter, we extend our previous work [13] to a second-order model with constant *acceleration* assumption which is more stable and shows better convergence. In our experiments, we demonstrate significantly improved performance both on synthetic data with higher-order kinematic scenarios and on the challenging KITTI benchmark [36]. Comparison with novel *continuous/discrete extended Kálmán filters on Lie groups* [17] shows that our approach – although being less general than [17] – leads to better results and is robust against imperfect initializations.

4.1.2 Contribution and Organization

Our contributions reported in this chapter amount

- to generalize the constant camera *velocity* model from [13] (non-linear measurement model) to polynomial models, in particular the constant *acceleration* model;
- to provide a complete derivation of the second-order minimum energy filter [72] as applied to camera motion estimation together with robust numerics that are consistent with the geometry and the structure of matrix Riccati equations;
- to report experiments demonstrating that *higher-order kinematic models* are more accurate than the *constant velocity model* [13] on synthetic (with kinematic camera tracks) and real world data and that they enable to reconstruct higher-order information;
- to report experiments comparing our approach to state-of-the-art extended Kálmán Filters on Lie groups [17], indicating that our method is superior in coping with non-linearities of the observation function as well as in being more robust against imperfect initializations.

In the next section, we introduce the filtering equations related to our problem of camera motion reconstruction. Next, we describ how to apply the (operator-valued) minimum energy filter derived from [72], which was introduced in section 3.3 , to our scenario. The numerical integration schemes of the ODEs for the optimal state will be given in section 4.5. We will confirm the theoretical results in section 4.6 by experiments on synthetic and real world data and thus underline the applicability of our approach.

4.2 Minimum Energy Filtering Approach

As introduced in example 3.2.5, will denote by $E(t) \in \mathrm{SE}_3$ the time-dependent (external) camera parameter that can be expressed in terms of a rotation matrix $R(t) \in \mathrm{SO}_3$ and a translation vector $w(t) \in \mathbb{R}^3$ as a 4×4 matrix

$$E(t) = \begin{pmatrix} R(t) & w(t) \\ \mathbf{0}_{1\times 3} & 1 \end{pmatrix}, \tag{4.1}$$

for which we also use the shorthand $E(t) = (R(t), w(t))$. Since the ego-motion of a camera is generally not constant, the model $\dot{E} = 0$ assumed in previous work [13] does not hold in real-world problems, where a camera fixed to a car rotates and accelerates in different directions. The *constant acceleration* assumption, however, is more suited in this cases. It can be described by the second-order differential equation $\ddot{E}(t) = 0$ for all t with initial pose $E(t_0) = E_0$ and velocity $\dot{E}(t_0) = V_0$. In general, one can consider a polynomial model of even higher-order for $E(t)$. In the following, we will focus on the assumption that $E(t)$ is quadratic in t. We will comment on generalizations at the end of section 4.3.
The equation $\ddot{E}(t) = 0$ can be prescribed as a system of first-order differential equations

$$\begin{aligned} \dot{E}(t) =& V(t)\,, \\ \dot{V}(t) =& 0\,, \end{aligned} \tag{4.2}$$

where $V(t) \in T_{E(t)}\,\mathrm{SE}_3$ and $\dot{V}(t) \in T_{V(t)}T_{E(t)}\,\mathrm{SE}_3 = T_{E(t)}\,\mathrm{SE}_3$. However, since the tangent bundle of a Lie group can be expressed in terms of the product $T\,\mathrm{SE}_3 \sim \mathrm{SE}_3 \times \mathfrak{se}_3$, we obtain a more compact expression, i.e.

$$\begin{aligned} \dot{E}(t) =& E(t)\,\mathrm{mat}_{\mathfrak{se}}(v(t)), \\ \dot{v}(t) =& \mathbf{0}_6 \in \mathbb{R}^6, \end{aligned} \tag{4.3}$$

where the operator $\mathrm{mat}_{\mathfrak{se}} : \mathbb{R}^6 \to \mathfrak{se}_3$ is defined by

$$(\eta_1, \eta_2, \eta_3, \eta_4, \eta_5, \eta_6)^\top \mapsto \begin{pmatrix} 0 & -\frac{\eta_3}{\sqrt{2}} & \frac{\eta_2}{\sqrt{2}} & \eta_4 \\ \frac{\eta_3}{\sqrt{2}} & 0 & -\frac{\eta_1}{\sqrt{2}} & \eta_5 \\ -\frac{\eta_2}{\sqrt{2}} & \frac{\eta_1}{\sqrt{2}} & 0 & \eta_6 \\ 0 & 0 & 0 & 0 \end{pmatrix}. \tag{4.4}$$

The inverse operation is denoted by $\mathrm{vec}_{\mathfrak{se}} : \mathfrak{se}_3 \to \mathbb{R}^6$. Note that this operation is consistent with the usual scalar product, i.e. for $\chi, \eta \in \mathfrak{se}_3$ it holds

$$\langle \chi, \eta \rangle_{\mathrm{Id}} := \mathrm{tr}(\chi^\top \eta) = \langle \mathrm{vec}_{\mathfrak{se}}(\chi), \mathrm{vec}_{\mathfrak{se}}(\eta) \rangle. \tag{4.5}$$

Since SE_3 is a Lie groups regarding the matrix multiplication and $v \in \mathbb{R}^6$ is a Lie group regarding addition, we can understand the system (4.3) as a first-order differential equation on a product Lie group

$$\mathcal{G} := \mathrm{SE}_3 \times \mathbb{R}^6. \tag{4.6}$$

For two elements $x_1 = (E_1, v_1), x_2 = (E_2, v_2) \in \mathcal{G}$ we define the left translation L_{x_1} by $L_{x_1} x_2 := (E_1 E_2, v_1 + v_2) \in \mathcal{G}$. Since the tangent bundle $T\mathbb{R}^6$ can be identified with $\mathbb{R}^6$, we obtain the Lie algebra

$$\mathfrak{g} = \mathfrak{se}_3 \times \mathbb{R}^6. \tag{4.7}$$

In turn, we can take down (4.3) compactly as

$$\dot{x}(t) = \left(E(t)\, \mathrm{mat}_{\mathfrak{se}}(v(t)), \mathbf{0}_6\right), \tag{4.8}$$

where E and v denote the first and second element of $x \in \mathcal{G}$, respectively. On matrix Lie groups, one can express kinematics directly as matrix multiplication (cf. [88]), i.e. $\dot{E} = E\Gamma$ for $\Gamma \in \mathfrak{se}_3, E \in \mathrm{SE}_3$, which is not valid for general Lie groups. The rigorous way to describe kinematics is to use the *tangent map* (cf. [72]) of the left translation which is given by the following Proposition:

Proposition 4.2.1. *The tangent map of the left translation regarding $x = (E, v) \in \mathcal{G}$ at identity, i.e. $T_{\mathrm{Id}} L_x : \mathfrak{g} \to T_x \mathcal{G}$, can be computed for $\eta = (\eta_1, \eta_2) \in \mathfrak{g}$ as*

$$T_{\mathrm{Id}} L_x \eta = (E\eta_1, \eta_2) = L_{(E,0)} \eta =: x\eta. \tag{4.9}$$

With Proposition 4.2.1 we can write down (4.8) as

$$\dot{x}(t) = T_{\mathrm{Id}} L_{x(t)} f(x(t)) = x(t) f(x(t)), \tag{4.10}$$

where $f : \mathcal{G} \to \mathfrak{g}$ is given by

$$f(x) = f((E, v)) = (\mathrm{mat}_{\mathfrak{se}}(v), \mathbf{0}_6). \tag{4.11}$$

Remark 4.2.2. During the further development, the notation $x\eta$ for a Lie group element $x \in \mathcal{G}$ and $\eta \in \mathfrak{g}$ must always be understood as the tangent map of the left translation at identity. Similarly, we denote $x^{-1}\eta := T_{\mathrm{Id}}L_x^*\eta$ for the dual of the tangent map of L_x at identity.

4.2.1 Optical Flow Induced by Ego-Motion

Within a static scene, the optical flow $u : \Omega \times \mathbb{R} \to \mathbb{R}^2$ on an image sequence $\{I(t), t \in \mathbb{R}\}$ can be computed in terms of the underlying scene structure. This is given by a depth map $d : \Omega \times \mathbb{R} \to \mathbb{R}_{>1}$ and the camera motion $E : \mathbb{R} \to \mathrm{SE}_3$, i.e. $E(t) = (R(t), w(t))$, where $R(t)$ and $w(t)$ denote the camera rotation and translation, respectively, by the following relation:

$$u(z,t;d(z,t),(R(t),w(t))) = \pi(R(t)^\top(\begin{pmatrix} z \\ 1 \end{pmatrix} d(z,t) - w(t))) - z, \quad z \in \Omega \tag{4.12}$$

whereas $\pi : \mathbb{R}^3 \to \mathbb{R}^2$ is the projection $(z_1, z_2, z_3)^\top \mapsto z_3^{-1}(z_1, z_2)^\top$ as depicted in Fig. 4.1. Note that $z \in \mathbb{R}^3$ indicates *inhomogenous* coordinates rather than *homogenous* coordinates on the projective space.
We can also express (4.12) directly in terms of $E(t)$ for each $z \in \Omega$:

$$u(z,t;d(z,t),E(t)) + z = \pi((E^{-1}(t)g_z(t))_{1:3}), \tag{4.13}$$

where $g_z(t) := (d(z,t)(z)^\top, d(z,t), 1)^\top$ denotes the data vector containing depth information of pixel z below.

4.3 Minimum Energy Filter Derivation

In this section, we will determine the problem of camera motion estimation with filtering equations, and we will summarize the most important steps for the derivation of the minimum energy filter.
By denoting the left side of (4.13) by $z \in \mathbb{R}^2$ which is the observation, i.e.

$$y_z(t) := u(z,t;d(z,t),E(t)) + z\,, \tag{4.14}$$

and defining

$$h_z(E,t) := \pi((E^{-1}(t)g_z(t))_{1:3}) \tag{4.15}$$

as the right hand side of (4.13), together with (4.3) and (4.10), we obtain the following state- and observation system by setting $x = (E, v) \in \mathcal{G}$:

$$\dot{x}(t) = x(t)(f(x(t)) + \delta(t)), x(t_0) = x_0\,, \qquad \text{(state)} \tag{4.16}$$

$$y_z = h_z(E(t),t) + \epsilon_z(t), z \in \Omega\,, \qquad \text{(observation)} \tag{4.17}$$

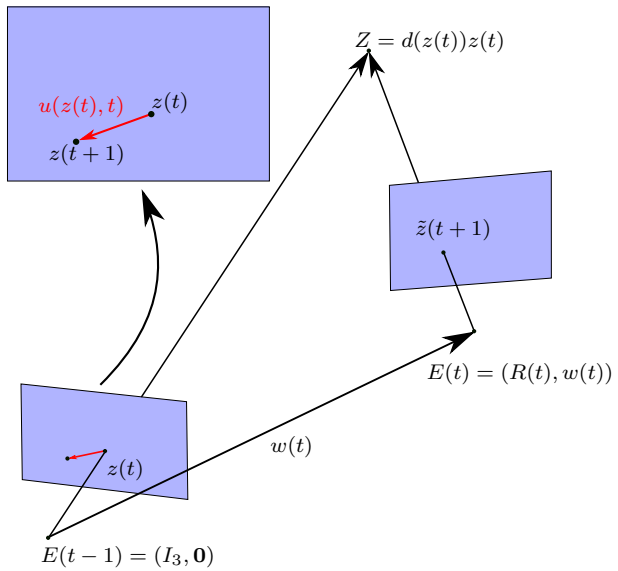


Figure 4.1: Camera model for the monocular approach: A static scene point Z is projected onto the plane at $z(t)$ of the first camera $E(t-1)$ which is mounted at the origin with rotation $\mathbb{1}_3$ such that $Z = d(z(t))z(t)$. By moving the camera into position $E(t) = (R(t), w(t))$, the scene point is projected onto $\pi(R^\top(t)(Z - w(t))) = z(t+1)$ which is at the same (relative) image position as $\tilde{x}(t+1)$ on the second image plane. The induced optical flow is given by the difference $u(z(t), t) = z(t+1) - z(t)$.

where $f(x)$ is defined as in (4.11). The functions $\delta : \mathbb{R} \to \mathfrak{g}$ and $\epsilon_z : \mathbb{R} \to \mathbb{R}^2$, $z \in \Omega$, are noise processes that model deviations from state and observations, respectively.

4.3.1 Energy Function

Given a depth map, which is contained in the function $g_z(t)$ in (4.15) and the optical flow $u(z,t)$ in terms of the observations y_z in (4.14), we want to find the camera motion and its velocity in terms of $x(t) \in \mathcal{G}$ such that the observation error ϵ_z in (4.17) is minimal and such that (4.16) is fulfilled with minimal deviations $\delta(t)$ for all $t \in \mathbb{R}$.
To this end, we consider the penalization of $\delta = (\delta_1, \delta_2) \in \mathfrak{g}$ and $\epsilon = \{\epsilon_z\}_{z\in\Omega}$ by a quadratic function $c : \mathfrak{g} \times \mathbb{R}^{2n} \times \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ given as

$$c(\delta, \epsilon, \tau, t) := \tfrac{1}{2}\Big(\|\mathrm{vec}_{\mathfrak{se}}(\delta_1(\tau))\|^2_{S_1} + \|\delta_2(\tau)\|^2_{S_2} + \sum_{z\in\Omega} \|\epsilon_z(\tau)\|^2_Q\Big), \tag{4.18}$$

where $S_1, S_2 \in \mathbb{R}^{6\times 6}$ and $Q \in \mathbb{R}^{2\times 2}$ are symmetric, positive definite weighting matrices. From [72] we adopt the idea of a decay rate $\alpha > 0$, and thus we introduce the weighting factor $e^{-\alpha(t-t_0)}$ on the right-hand side of (4.18):

$$c(\delta, \epsilon, \tau, t) := \tfrac{1}{2} e^{-\alpha(t-t_0)} \Big(\|\mathrm{vec}_{\mathfrak{se}}(\delta_1(\tau))\|^2_{S_1} + \|\delta_2(\tau)\|^2_{S_2} + \sum_{z\in\Omega} \|\epsilon_z(\tau)\|^2_Q \Big). \tag{4.19}$$

Based on the penalty function (4.19), we define the energy:

$$\mathcal{J}(\delta, \epsilon, t_0, t) := m_0(x(t), t, t_0) + \int_{t_0}^{t} c(\delta, \epsilon, \tau, t)\, d\tau\,, \tag{4.20}$$

where m_0 is a quadratic penalty function for the initial state. For our model we set

$$m_0(x, t, t_0) := \tfrac{1}{2} e^{-\alpha(t-t_0)} \langle x - \mathrm{Id}, x - \mathrm{Id} \rangle_{\mathrm{Id}}, \tag{4.21}$$

where the difference is canonical, i.e. $x - \mathrm{Id} = (E - \mathbb{1}_4, v)$ for $x = (E, v)$.

Remark 4.3.1. Instead of using two quadratic forms with matrices S_1, S_2, we can use more generally a symmetric and positive weighting matrix $S \in \mathbb{R}^{12\times 12}$ if we want to couple δ_1 and δ_2. In the upper case we find that $S = \left(\begin{smallmatrix} S_1 & 0 \\ 0 & S_2 \end{smallmatrix}\right)$.

4.3.2 Optimal Control Problem

The optimal control theory allows us to determine the optimal control input $\delta : \mathbb{R} \to \mathfrak{g}$ that minimizes the energy $\mathcal{J}(\delta, \epsilon, t_0, t)$ for each $t \in \mathbb{R}$ subject to the state constraints (4.16). To be precise, we want to find for all $t \in \mathbb{R}$ and fixed $x(t)$ the control input $\delta|_{[t_0,t]}$ defining

$$\mathcal{V}(x(t), t) := \min_{\delta|_{[t_0,t]}} \mathcal{J}(\delta, \epsilon(x(t), t), t_0, t), \text{ s.t. } (4.16)\,. \tag{4.22}$$

The optimal trajectory is

$$x^*(t) := \arg\min_{x(t)\in\mathcal{G}} \mathcal{V}(x(t), t)\,, \tag{4.23}$$

for all $t \in \mathbb{R}$ and $\mathcal{V}(x, t_0) = m_0(x_0, t_0, t_0)$. This problem is a classical optimal control problem, for which the standard Hamilton-Jacobi theory [48, 5] under appropriate conditions results in the well-known *Hamilton-Jacobi-Bellman* equation. Pontryagin [5] proved that the minimization of the Hamiltonian provides a solution to the corresponding optimal control problem (*Pontryagin's Minimum Principle*).

However, since $\mathcal{G}$ is a non-compact Riemannian manifold, we cannot apply the classical Hamilton-Jacobi theory for real-valued problems (cf. [5]). Instead we follow the approach of Saccon *et al.* [72] who derived a *left-trivialized optimal Hamiltonian* based on control theory on Lie groups [48]. This left-trivialized optimal Hamiltonian is defined by $\tilde{\mathcal{H}}^- : \mathcal{G} \times \mathfrak{g} \times \mathfrak{g} \times \mathbb{R} \to \mathbb{R}$,

$$\tilde{\mathcal{H}}^-(x, \mu, \delta, t) := c(\delta, \epsilon(x,t), t_0, t) - \langle \mu, f(x(t)) + \delta(t) \rangle_{\mathrm{Id}}. \tag{4.24}$$

The minimization of (4.24) w.r.t. the variable $\delta = (\delta_1, \delta_2)$ leads [72, Proposition 4.2] to the optimal Hamiltonian

$$\mathcal{H}^-(x, \mu, t) := \tilde{\mathcal{H}}^-(x, \mu, \delta^*, t)\,, \tag{4.25}$$

where $\delta^* = (\delta_1^*, \delta_2^*)$ is given by

$$\begin{aligned} \mathrm{vec}_{\mathfrak{se}}(\delta_1^*) &= e^{\alpha(t-\tau)} S_1^{-1} \,\mathrm{vec}_{\mathfrak{se}}(\mu_1), \text{ and} \\ \delta_2^* &= e^{\alpha(t-t_0)} S_2^{-1} \mu_2\,. \end{aligned} \tag{4.26}$$

Examining the right-hand side of (4.25) in detail, we obtain

$$\begin{aligned} \mathcal{H}^-((E,v),\mu,t) = {} & \tfrac{1}{2} e^{-\alpha(t-t_0)} \Big(\sum_{z \in \Omega} \| y_z - h_z(E) \|_Q^2 \Big) \\ & - \tfrac{1}{2} e^{\alpha(t-t_0)} \Big(\langle \mu_1, \mathrm{mat}_{\mathfrak{se}}(S_1^{-1} \,\mathrm{vec}_{\mathfrak{se}}(\mu_1)) \rangle_{\mathrm{Id}} \\ & + \langle \mu_2, S_2^{-1} \mu_2 \rangle \Big) - \langle \mu_1, \mathrm{mat}_{\mathfrak{se}}(v) \rangle_{\mathrm{Id}}\,, \end{aligned} \tag{4.27}$$

where we used $\epsilon(x(t),t) = \{ y_z - h_z(E(t),t) \}_{z \in \Omega}$. Here we introduced on the left hand the variable x since the right hand side depends on $x = (E, v)$.
In the next section, we will compute explicit ordinary differential equations regarding the optimal state $x^*(t)$ for each $t \in \mathbb{R}$ that consists of different derivatives of the left trivialized Hamilton function (4.27).

4.3.3 Recursive Filtering Principle by Mortensen

In order to find a recursive filter, we compute the total time derivative of the optimality condition on the value function, which is

$$\mathbf{D}_1 \mathcal{V}(x^*, t) = 0\,, \tag{4.28}$$

for each $t \in \mathbb{R}$. This equation must be fulfilled by an optimal solution $x^* \in \mathcal{G}$ of the filtering problem. Unfortunately, because the filtering problem is in general infinite dimensional, this leads to an expression containing derivatives

of every order. In practice (cf. [88, 72]), derivatives of third order and higher are neglected, since they require tensor calculus. Omitting these leads to a second-order approximation of the optimal filter. The following Theorem is an adaption of [72, Theorem 4.1]:

Theorem 4.3.2. *The differential equations of the second-order Minimum Energy Filter for state* (4.16) *and non-linear observer model* (4.17) *are given by*

$$\begin{aligned}
&(x^*)^{-1}\dot{x}^* = \big(f(x^*) - \mathrm{mat}_{\mathfrak{g}}(P(t)\,\mathrm{vec}_{\mathfrak{g}}(r_t(x^*)))\big),\\
&x^*(t_0) = \mathrm{Id}\,, \qquad (4.29)\\
&\dot{P}(t) = -\alpha\cdot P + S^{-1} + CP + PC^\top\\
&- P\begin{pmatrix} \sum_{z\in\Omega}(\tilde{\Gamma}_{\mathrm{vec}_{\mathfrak{se}}(\mathrm{Pr}(A_z(E^*)))} + D_z(E^*)) & \mathbf{0}_{6\times 6}\\ \mathbf{0}_{6\times 6} & \mathbf{0}_{6\times 6}\end{pmatrix} P\,, \qquad (4.30)\\
&P(t_0) = \mathbb{1}_{12}\,,
\end{aligned}$$

where $r_t(x^*) := \big(\sum_{z\in\Omega}\mathrm{Pr}(A_z(E^*), \mathbf{0}_6\big)$ *and with the orthogonal projection* $\mathrm{Pr}: \mathbb{R}^{4\times 4} \to \mathfrak{se}_3$ *defined in section A.2.2. The expressions* C *and* Ψ *are given through*

$$C(x^*, t) := \begin{pmatrix} -\Psi(x^*, t) & \mathbb{1}_6\\ \mathbf{0}_{6\times 6} & \mathbf{0}_{6\times 6}\end{pmatrix},$$

with

$$\Psi(x^*, t) := \mathrm{ad}^{\mathrm{vec}}_{\mathfrak{se}_3}(f(x^*)) + \tilde{\Gamma}^*_{\mathrm{vec}_{\mathfrak{se}}(P\, r_t(x^*))}\,. \qquad (4.31)$$

The function $A_z : \mathrm{SE}_3 \to \mathbb{R}^{4\times 4}$ *is given by*

$$\begin{aligned}
A_z(E) = D_z(E, g_z) := \big(\kappa_z^{-1}\hat{I} - \kappa_z^{-2}\hat{I}E^{-1}e_3^4 g_z^\top \hat{I}\big)^\top\\
\cdot Q(y_z - h_z(E))g_z^\top E^{-\top},
\end{aligned} \qquad (4.32)$$

where $\kappa_z := \kappa_z(E) := (e_3^4)^\top E^{-1} g_z$. *The second-order operator* $D_z : \mathrm{SE}_3 \to \mathbb{R}^{6\times 6}$ *is given by* (A.37)*, see Appendix A.3.1.*
The matrix valued functions $\tilde{\Gamma}_z, \tilde{\Gamma}^*_z : \mathbb{R}^6 \to \mathbb{R}^{6\times 6}$ *are obtained from the vectorization of the connection functions. Their components are given by* $(\tilde{\Gamma}_z)_{ij} := \sum_{k=1}^6 \Gamma^i_{jk} z^k$ *and* $(\tilde{\Gamma}^*_z)_{ik} := \sum_{j=1}^6 \Gamma^i_{jk} z^j$ *with* $z \in \mathbb{R}^6$ *and the Christoffel-Symbols* Γ^i_{jk} *are given in Appendix A.2.5.*

This Theorem will be proven at the end of the section after we provided some essential Lemmas.

Remark 4.3.3. A generalization of this Theorem is published in Saccon *et al.* [72] for a larger class of filtering problems. However, the application of the Theorem is not straightforward since the appearing expressions, e.g. exponential functor, cannot be evaluated directly. Furthermore, the adaption to non-linear filtering problems has not been considered in the literature yet. Besides, we show how to find explicit expressions in terms of matrices for the general operators in [72].

In our previous work [13] we presented a theory regarding the case of constant *velocity*. This theory can be derived directly from Theorem 4.3.2 by neglecting the velocity v i.e. the second component of $x = (E, v) \in \mathcal{G}$ (thus changing from Lie group $\mathrm{SE}_3 \times \mathbb{R}^6$ to SE_3) and by setting $f(x) \equiv 0$. In this case, the state and observation equations are reduced to

$$\dot{E}(t) = E(t)\delta(t),\ E(t_0) = E_0, \quad (\text{state}), \tag{4.33}$$

$$y_z = h_z(E(t), t) + \epsilon_z(t),\ z \in \Omega. \quad (\text{observation}). \tag{4.34}$$

For the reader's convenience, we state the theory under the assumption of constant *velocity* as a corollary:

Corollary 4.3.4. *The differential equations of the second-order Minimum Energy Filter for our state* (4.33) *and non-linear observer model* (4.34) *are given by*

$$\begin{aligned}
&(E^*)^{-1}\dot{E}^* = -\operatorname{mat}_{\mathfrak{se}}(P(t)\operatorname{vec}_{\mathfrak{se}}(\sum_{z\in\Omega}\Pr(A_z(E^*)))),\\
&E^*(t_0) = \mathrm{Id}\,, \qquad (4.35)\\
&\dot{P}(t) = -\alpha\cdot P + S_1^{-1}\\
&- \tilde{\Gamma}^*_{\operatorname{vec}_{\mathfrak{se}}((E^*)^{-1}\dot{E}^*)}P - P(\tilde{\Gamma}^*_{\operatorname{vec}_{\mathfrak{se}}((E^*)^{-1}\dot{E}^*)})^\top\\
&- P\big(\sum_{z\in\Omega}(\tilde{\Gamma}_{\operatorname{vec}_{\mathfrak{se}}(\Pr(A_z(E^*)))} + D_z(E^*))\big)P, \qquad (4.36)\\
&P(t_0) = \mathbb{1}_6.
\end{aligned}$$

Remark 4.3.5. We compare the computational complexity for the cases of constant *velocity* and constant *acceleration*. By considering the difference between Theorem 4.3.2 and Corollary 4.3.4, we see that the only differences are a larger state space and the occurrence of the additional operator $f(x^*)$ in (4.31). However, this does not change the computational effort significantly. Thus, we suggest to use the second-order state equation since it is more robust but computational only slightly more complex as we will see in the experiments.

Before we will turn to proving Theorem 4.3.2, we first provide some Lemmas that are based on the general approach of [72]. However, we cannot use the main result of [72] directly, since the appearing general operators are complicated to evaluate. Instead, we provide the corresponding expressions in such a way that they can be easily implemented. Thus, following [72, Eq. (37)] the estimate of the optimal state x^* is given by

$$(x^*)^{-1}\dot{x}^* = -\mathbf{D}_2\mathcal{H}^-(x^*,0,t) - Z(x^*,t)^{-1} \circ (x^*)^{-1}\mathbf{D}_1\mathcal{H}^-(x^*,0,t)\,. \tag{4.37}$$

This expression contains the second-order information matrix $Z(x,t) : \mathfrak{g} \to \mathfrak{g}$ of the value function $\mathcal{V}$ as defined in (4.22), defined through

$$Z(x,t) \circ \eta = x^{-1} \circ \operatorname{Hess}_1 \mathcal{V}(x,t)[x\eta]\,. \tag{4.38}$$

An explicit expression for the gradient of the Hamiltonian in (4.37) is provided in the following Lemma:

Lemma 4.3.6. *The Riemannian gradient $\mathbf{D}_1\mathcal{H}^-(x,\mu,t)$ on $T_x\mathcal{G}$ for $x = (E,v)$ can be calculated as*

$$\mathbf{D}_1\mathcal{H}^-(x,\mu,t) = x\Big(e^{-\alpha(t-t_0)}\sum_{z\in\Omega}\Pr(A_z(E)), -\operatorname{vec}_{\mathfrak{se}}(\mu_1)\Big), \tag{4.39}$$

where the function $A_z(E) = A_z(E,g_z) : \mathrm{SE}_3 \times \mathbb{R}^4 \to \mathrm{GL}_4$ is defined in (4.32).

By insertion of (4.39) in (4.37) and usage of the definition of $r_t(x^*)$ from Theorem 4.3.2 we obtain

$$(x^*)^{-1}\dot{x}^* = -\mathbf{D}_2\mathcal{H}^-(x^*,0,t) - e^{-\alpha(t-t_0)}Z(x^*,t)^{-1} \circ r_t(x^*). \tag{4.40}$$

Following the calculus in [72], the evolution equation for the trivialized Hessian $Z(x,t) : \mathfrak{g} \to \mathfrak{g}^*$ is given through [72, Eq. (51)] which is:

$$\begin{aligned}
&\frac{d}{dt}Z(x^*(t),t)\\
\approx&Z(x^*,t)\circ\omega_{(x^*)^{-1}\dot{x^*}}\\
&+ Z(x^*,t)\circ\omega^{\rightleftharpoons}_{\mathbf{D}_2\mathcal{H}^-(x^*,0,t)}\\
&+ \omega^*_{(x^*)^{-1}\dot{x^*}}\circ Z(x^*,t)\\
&+ \omega^{\rightleftharpoons *}_{\mathbf{D}_2\mathcal{H}^-(x^*,0,t)}\circ Z(x^*,t)\\
&+ T_{\mathrm{Id}}L^*_{x^*}\circ\operatorname{Hess}_1\mathcal{H}^-(x^*,0,t)\circ T_{\mathrm{Id}}L_{x^*}\\
&+ T_{\mathrm{Id}}L^*_{x^*}\circ\mathbf{D}_2(\mathbf{D}_1\mathcal{H}^-)(x^*,0,t)\circ Z(x^*,t)\\
&+ Z(x^*,t)\circ\mathbf{D}_1(\mathbf{D}_2\mathcal{H}^-)(x^*,0,t)\circ T_{\mathrm{Id}}L_{x^*}\\
&+ Z(x^*,t)\circ\operatorname{Hess}_2\mathcal{H}^-(x^*,0,t)\circ Z(x^*,t)\,.
\end{aligned} \tag{4.41}$$

(cf. [72, Eq. (51)]).
The "swap"-operators $\omega_{\cdot}^{\leftrightharpoons}\cdot, \omega_{\cdot}^{\leftrightharpoons *}\cdot$ in this expression are defined in section 1.5, i.e. $\omega_{\eta}^{\leftrightharpoons}\xi := \omega_{\xi}\eta$ and $\langle \omega_{\eta}^{\leftrightharpoons *}\xi, \chi\rangle_{\mathrm{Id}} := \langle \xi, \omega_{\eta}^{\leftrightharpoons}\chi\rangle_{\mathrm{Id}} = \langle \xi, \omega_{\chi}\eta\rangle_{\mathrm{Id}}$. By considering the standard basis of $\mathfrak{g}$, there exists a matrix representation $K \in \mathbb{R}^{12\times 12}$, such that for all $\eta = (\eta_1, \eta_2) \in \mathfrak{g}$ we receive

$$\mathrm{vec}_{\mathfrak{g}}(Z(x^*, t) \circ \eta) = K(t)\,\mathrm{vec}_{\mathfrak{g}}(\eta)\,. \tag{4.42}$$

Similarly to [13] we need to evaluate the right-hand side of the evolution equation at $\eta \in \mathfrak{g}$ and to vectorize it. The single expressions are shown in the following Lemma.

Lemma 4.3.7 (Matrix representations of Z)**.** *Let $Z(x^*, t) : \mathfrak{g} \to \mathfrak{g}$ be the operator* (4.38)*. Then there exists a matrix $K = K(t) \in \mathbb{R}^{12\times 12}$ yielding*

$$\mathrm{vec}_{\mathfrak{g}}(Z(x^*, t)(\eta)) = K(t)\,\mathrm{vec}_{\mathfrak{g}}(\eta), \tag{4.43}$$

and thus

$$\mathrm{vec}_{\mathfrak{g}}(d/dt Z(x^*, t)(\eta)) = \dot{K}(t)\,\mathrm{vec}_{\mathfrak{g}}(\eta)\,, \tag{4.44}$$
$$\mathrm{vec}_{\mathfrak{g}}(Z^{-1}(x^*, t)(\eta)) = K^{-1}(t)\,\mathrm{vec}_{\mathfrak{g}}(\eta)\,, \tag{4.45}$$

as well as

1. $\mathrm{vec}_{\mathfrak{g}}(Z(x^*, t) \circ \omega_{(x^*)^{-1}\dot{x^*}}\eta$
$+ Z(x^*, t) \circ \omega_{\mathbf{D}_2\mathcal{H}^-(x^*,0,t)}^{\leftrightharpoons}\eta) = K(t)B\,\mathrm{vec}_{\mathfrak{g}}(\eta)$

2. $\mathrm{vec}_{\mathfrak{g}}(\omega^*_{(x^*)^{-1}\dot{x^*}} \circ Z(x^*, t) \circ \eta$
$+ \omega^{\leftrightharpoons *}_{\mathbf{D}_2\mathcal{H}^-(x^*,0,t)} \circ Z(x^*, t) \circ \eta) = B^\top K(t)\,\mathrm{vec}_{\mathfrak{g}}(\eta)$

3. $\mathrm{vec}_{\mathfrak{g}}(T_{\mathrm{Id}}L^*_{x^*} \circ \mathrm{Hess}_1\,\mathcal{H}^-(x^*, 0, t)[T_{\mathrm{Id}}L_{x^*}\eta]) = e^{-\alpha(t-t_0)}$
$\cdot \begin{pmatrix} \sum_{z\in\Omega}(\tilde{\Gamma}_{\mathrm{vec}_{\mathfrak{se}}(\mathrm{Pr}(A_z(E)))} + D_z(E)) & \mathbf{0}_{6\times 6} \\ \mathbf{0}_{6\times 6} & \mathbf{0}_{6\times 6} \end{pmatrix} \mathrm{vec}_{\mathfrak{g}}(\eta)$

4. $\mathrm{vec}_{\mathfrak{g}}(Z(x^*, t) \circ \mathbf{D}_1(\mathbf{D}_2\mathcal{H}^-)(x^*, 0, t) \circ T_{\mathrm{Id}}L_{x^*}\eta)$
$= -K(t) \begin{pmatrix} \mathbf{0}_{6\times 6} & \mathbb{1}_6 \\ \mathbf{0}_{6\times 6} & \mathbf{0}_{6\times 6} \end{pmatrix} \mathrm{vec}_{\mathfrak{g}}(\eta)$

5. $\mathrm{vec}_{\mathfrak{g}}(T_{\mathrm{Id}}L^*_{x^*} \circ \mathbf{D}_2(\mathbf{D}_1\mathcal{H}^-)(x^*, 0, t) \circ Z(x^*, t) \circ \eta)$
$= -\begin{pmatrix} \mathbf{0}_{6\times 6} & \mathbf{0}_{6\times 6} \\ \mathbb{1}_6 & \mathbf{0}_{6\times 6} \end{pmatrix} K(t)\,\mathrm{vec}_{\mathfrak{g}}(\eta)$

6. $\mathrm{vec}_{\mathfrak{g}}(Z(x^*, t)(\mathrm{Hess}_2\,\mathcal{H}^-(x^*, 0, t)[Z(x^*, t)(\eta)]))$
$= -e^{\alpha(t-t_0)}K(t)S^{-1}K(t)\,\mathrm{vec}_{\mathfrak{g}}(\eta)\,,$

with $\tilde{\Gamma}_\cdot$, $\tilde{\Gamma}^*_\cdot$, *and functions* A_z, D_z *from Theorem 4.3.2 and*

$$B := \begin{pmatrix} \Psi(x^*,t) & \mathbf{0}_{6\times 6} \\ \mathbf{0}_{6\times 6} & \mathbf{0}_{6\times 6} \end{pmatrix}, \tag{4.46}$$

with Ψ *from Theorem 4.3.2.*

With these Lemmas we are able to prove our main result in Theorem 4.3.2:

Proof of Theorem 4.3.2. We can easily compute the differential of Hamiltonian in (4.27) which is

$$-\mathbf{D}_2\mathcal{H}^-(x^*,0,t) = \big(\mathrm{mat}_{\mathfrak{se}}(v^*),\mathbf{0}\big) = f(x^*)\,. \tag{4.47}$$

By inserting expression (4.47) into the optimal state equation (4.40) together with the definition of the operator $\mathrm{vec}_{\mathfrak{g}}(Z(x^*,t)^{-1}\circ x^*\eta) = K^{-1}(t)\,\mathrm{vec}_{\mathfrak{g}}(\eta)$, we find that

$$\begin{aligned}(x^*)^{-1}\dot{x}^* =& f(x^*)\\ &- e^{-\alpha(t-t_0)}\,\mathrm{mat}_{\mathfrak{g}}\big(\mathrm{vec}_{\mathfrak{g}}(Z(x^*,t)^{-1}\circ r_t(x^*))\big)\\ =& f(x^*) - e^{-\alpha(t-t_0)}\,\mathrm{mat}_{\mathfrak{g}}\big(K^{-1}(t)\,\mathrm{vec}_{\mathfrak{g}}(r_t(x^*))\big)\,.\end{aligned} \tag{4.48}$$

The application of the $\mathrm{vec}_{\mathfrak{g}}$ −operation onto the equation (4.41) evaluated for a direction η, together with Lemma 4.3.7 results in

$$\begin{aligned}\dot{K}(t)\,\mathrm{vec}_{\mathfrak{g}}(\eta) = \Big[&K(t)B + B^\top K(t)\\ &+ e^{-\alpha(t-t_0)}\begin{pmatrix}\sum_{z\in\Omega}(\tilde{\Gamma}_{\mathrm{vec}_{\mathfrak{se}}(\mathrm{Pr}(A_z(E)))}+D_z(E)) & \mathbf{0}_{6\times 6}\\ \mathbf{0}_{6\times 6} & \mathbf{0}_{6\times 6}\end{pmatrix}\\ &- K(t)\begin{pmatrix}\mathbf{0}_{6\times 6} & \mathbb{1}_6\\ \mathbf{0}_{6\times 6} & \mathbf{0}_{6\times 6}\end{pmatrix} - \begin{pmatrix}\mathbf{0}_{6\times 6} & \mathbf{0}_{6\times 6}\\ \mathbb{1}_6 & \mathbf{0}_{6\times 6}\end{pmatrix}K(t)\\ &- e^{\alpha(t-t_0)}K(t)S^{-1}K(t)\Big]\,\mathrm{vec}_{\mathfrak{g}}(\eta),\end{aligned} \tag{4.49}$$

where on the right-hand side we assume that $K(t)$ is an approximation of the vectorized operator $Z(x^*(t),t)$. This is the reason why we replace the approximation by an equality sign in (4.49). With a change of variables (cf. [72])

$$P(t) := e^{-\alpha(t-t_0)}K(t)^{-1}\,, \tag{4.50}$$

and the formula for the derivative of the inverse of a matrix [64], we obtain

$$\begin{aligned}\dot{P}(t) =& -\alpha e^{-\alpha(t-t_0)}K(t)^{-1} - e^{-\alpha(t-t_0)}K(t)^{-1}\dot{K}(t)K(t)^{-1}\\ =& -\alpha P(t) - e^{\alpha(t-t_0)}P(t)\dot{K}(t)P(t)\,.\end{aligned} \tag{4.51}$$

Insertion of (4.49) (after omitting the direction $\mathrm{vec}_{\mathfrak{g}}(\eta)$ that was chosen arbitrarily) into (4.51) leads to the differential equation (4.30) in Theorem 4.3.2. Therefore, we also find that

$$C(x^*, t) = \begin{pmatrix} \mathbf{0}_{6\times 6} & \mathbb{1}_6 \\ \mathbf{0}_{6\times 6} & \mathbf{0}_{6\times 6} \end{pmatrix} - B(t)\,. \tag{4.52}$$

The differential equation of the optimal state (4.29) follows from inserting (4.50) into (4.48), which completes the proof. □

4.3.4 Generalization to Higher-Order Models

In the previous section, we discussed minimum energy filters to estimate ego-motion under the assumption of *constant acceleration*. We saw that changing the assumption of *constant velocity* to *constant acceleration* requires extending the Lie group and adopting the functions $f(x)$ and $C(x)$.
The generalization to higher polynomial models regarding camera motion, where we assume that the m-th order derivative of the ego-motion should be zero, i.e.

$$\frac{d^m}{dt^m} E(t) = 0\,, \tag{4.53}$$

is straightforward. Again, the approach can be described by a system of first-order ODEs as follows. Note that in the constant acceleration model (second-order), only the first-order model needs to respect manifold structures, whereas all the other derivatives are trivial since they evolve on Euclidean spaces:

$$\begin{aligned} \dot{E}(t) =& E(t)\big(\mathrm{mat}_{\mathfrak{se}}(v_1(t)) + \delta_1(t)\big), \\ \dot{v}_1(t) =& v_2(t) + \delta_2(t), \\ &\vdots \\ \dot{v}_{m-2}(t) =& v_{m-1} + \delta_{m-1}(t), \\ \dot{v}_{m-1}(t) =& \delta_m(t)\,. \end{aligned} \tag{4.54}$$

To achieve a unique solution we require initial values, i.e. $v_1(0) = v_1^0, \ldots, v_{m-1}(0) = v_{m-1}^0 \in \mathbb{R}^6$. Again, the observation equations (4.17) stay unchanged. The minimum energy filter for this model is provided by the following Theorem. By using once again

$$x = (E, v_1, \ldots, v_{m-1}) \in \mathcal{G}_m := \mathrm{SE}_3 \times \mathbb{R}^6 \times \cdots \times \mathbb{R}^6\,, \tag{4.55}$$

the corresponding minimum energy filter can be obtained easily from Theorem 4.3.2.

Theorem 4.3.8 (Minimum energy filter for $m-$th order state equation)**.** *The differential equations of the second-order Minimum Energy Filter for the state equation* (4.54) *and the observation equations* (4.17) *are given by the equations* (4.29) *and*

$$\begin{aligned}
&\dot{P}(t) = -\alpha \cdot P + S^{-1} + CP + PC^\top \\
&- P \begin{pmatrix} \sum_{z\in\Omega}(\tilde{\Gamma}_{\mathrm{vec}_{\mathfrak{se}}(\Pr(A_z(E^*)))} + D_z(E^*)) & \mathbf{0}_{6\times(m-1)6} \\ \mathbf{0}_{(m-1)6\times 6} & \mathbf{0}_{(m-1)6\times(m-1)6} \end{pmatrix} P\,, \qquad (4.56)\\
&P(t_0) = \mathbb{1}_{6m}\,,
\end{aligned}$$

where we assume that the expressions x^* and P lie in the spaces $\mathcal{G}_m$ and $\mathbb{R}^{6m\times 6m}$, respectively. The appearing expressions in Theorem 4.3.2 are replaced by

$$\begin{aligned}
f(x) :=& (\mathrm{mat}_{\mathfrak{se}}(v_1), v_2, \dots, v_{m-1}, \mathbf{0_{6\times 1}}),\\
r_t(x^*) :=& \big(\sum_{z\in\Omega} \Pr(A_z(E^*), \mathbf{0}_{(m-1)6\times 1}\big),\\
C(x^*,t) :=& \begin{pmatrix} \begin{pmatrix} -\Psi(x^*,t) \\ \mathbf{0}_{6(m-2)\times 6} \end{pmatrix} & \mathbb{1}_{6(m-1)} \\ \mathbf{0}_{6\times 6} & \mathbf{0}_{6\times 6(m-1)} \end{pmatrix}.
\end{aligned}$$

All the other expressions from Theorem 4.3.2 stay unchanged.

Proof. Since product Lie groups are simply Lie groups with the product topology, we can still apply the general minimum energy filter of Saccon *et al.* [72]. The Lie group $\mathcal{G}_m$ has dimension $6m$ such that the vectorized bilinear operator Z from (4.38), i.e. P results in a $6m \times 6m$ matrix. The definition of the function f follows from the differential equations in (4.54). Similarly to Theorem 4.3.2, the observations do not depend on the whole state $x = (E, v_1, \dots, v_{m-1})$, but only on E. This leads to the fact that r_t, which is essentially the left-trivialized differential of the Hamiltonian (i.e. $x^{-1}\mathbf{D}_1\mathcal{H}^-(x,\mathbf{0},t)$), vanishes after calculating the differentials regarding $v_1, \dots, v_{m-1}$. Similarly, the Hessian $x^{-1}\,\mathrm{Hess}_1\,\mathcal{H}^-(x,\mathbf{0},t)[x\eta]$ in Lemma 4.3.7 can be extended by zeros. Furthermore, components $v_1, \dots, v_{m-1} \in \mathbb{R}^6$ have a trivial geometry and do not contribute to curvature and thus the corresponding connection functions in Lemma 4.3.7 also do not influence curvature. Finally, we can compute the expression

$$\begin{aligned}
\mathbf{D}_1(\mathbf{D}_2\mathcal{H}^-(x,\mathbf{0},t))[x\eta] &= -\mathbf{D}f(x)[\eta]\\
&= -(\mathrm{mat}_{\mathfrak{se}}(v_2), v_3, \dots, v_{m-1}, \mathbf{0})
\end{aligned}$$

and thus

$$\mathrm{vec}_{\mathfrak{g}}\big(\mathbf{D}_1(\mathbf{D}_2\mathcal{H}^-(x,\mathbf{0},t))[x\eta]\big) = \begin{pmatrix} \mathbf{0}_{6(m-1)\times 6} & \mathbb{1}_{6(m-1)} \\ \mathbf{0}_{6\times 6} & \mathbf{0}_{6\times 6(m-1)} \end{pmatrix}, \qquad (4.57)$$

as we did in Lemma 4.3.7 for the special case. Together with the adjoint operator in $\Psi(x,t)$, we obtain the expression C. □

4.4 Comparison with Extended Kálmán Filters

As an alternative to the proposed approach, we also suggest considering extended Kálmán filters, which are established within the stochastic filtering community for many decades. The extended Kálmán filter was also generalized to Lie groups within the last years yielding to a *discrete / continuous extended Kálmán filter on Lie groups* [17]. We provide the corresponding calculations in section 4.6. The Kálmán filter approach is valid in a more generalized scenario compared to ours because the state space as well as the observation space are matrix Lie groups, whereas we only consider real-valued observations in $\mathbb{R}^n$. On the other hand, one needs to know that the covariance matrices of the model and observation noise and the a posteriori distribution are assumed to be Gaussian, which is in general not true for non-linear observation dynamics.

Algorithm 1 Extended Kálmán Filter for Lie groups

Require: State $x(t_{l-1})$, Covariance $P(t_{l-1})$, Observations $y_k(t_l), k = 1,\dots,n$

1: **procedure** PROPAGATION ON $[t_{l-1}, t_l]$: *Integrate the following differential equations*

2: $\quad \dot{x}(t) = x(t)f(x(t))$

3: $\quad \dot{P}(t) = J(t)P(t) + P(t)(J(t))^\top + S$
$\quad\quad +\frac{1}{4}\mathbb{E}(\mathrm{ad}_{\mathfrak{g}}(\epsilon(t))S\,\mathrm{ad}_{\mathfrak{g}}(\epsilon(t))^\top)$
$\quad\quad +\frac{1}{12}\mathbb{E}\big(\mathrm{ad}_{\mathfrak{g}}(\epsilon(t))^2\big)S + \frac{1}{12}S\mathbb{E}\big(\mathrm{ad}_{\mathfrak{g}}(\epsilon(t))^2\big)^\top$

4: $\quad x^-(t_l) = x(t_l),\ P^-(t_l) = P(t_l)$

5: ***procedure*** UPDATE:

6: $\quad K_l = P^-(t_l)H_l^\top\big(H_lP^-(t_l)H_l^\top + Q_l\big)^{-1}$

7: $\quad m_{l|l}^- = K_l\sum_{z\in\Omega}\big(y_z(t_l) - h_z(x^-(t_l))\big)$

8: $\quad x(t_l) = x^-(t_l)\,\mathrm{Exp}(\mathrm{mat}_{\mathfrak{g}}(m_{l|l}^-))$

9: $\quad P(t_l) = \Phi(m_{l|l}^-)\big(\mathbb{1}_{12} - K_lH_l\big)P^-(t_l)\Phi(m_{l|l}^-)^\top$

The extended Kálmán Filter from [17] is summarized in Algorithm 1 and has already been adapted to our problem for real-valued observations. In line 7 the residual is expressed as direct difference which is a special case of [17]. The function Φ in line 9 on $\mathcal{G}$ is shown in Appendix A.4.

In the next section, we will adapt the Algorithm 1 to different scenarios: to a filtering problem with linear observations as well as to our non-linear filtering problem with a projective camera (cf. (4.16), (4.17)).

Remark 4.4.1. Note that the extended Kálmán filter from [17] requires a differential equation (that is not only driven by noise) in order to propagate the state, i.e. $\dot{E}(t) = E(t)\big(f(E) + \delta(t)\big)$, where f is non-trivial. Otherwise the update step of the extended Kálmán filter is not significant because update and correction steps in the extended Kálmán filter are separated. This is the reason why we only compare it to the second-order state equation where $f \not\equiv 0$.

4.4.1 Derivations for Linear Observations

In the scenario of linear observations the state equation stays unchanged, i.e. is identical to (4.16). Similarly to [88] we use the following linear observation equations:

$$y_k(t) = E(t)a_k + \epsilon_k(t), \quad k \in [n], \tag{4.58}$$

where $E(t) \in \mathrm{SE}_3$ is the first component of $x(t) \in \mathcal{G}$ and $a_k \in \mathbb{R}^4$ are vectors that model the linear transformation of the state x. Again, $\epsilon_k(t) \in \mathbb{R}^4$ are the observation noise vectors. In this case, the minimum energy filter can be derived much more easily than in the non-linear case. Thus, for the compactness of presentation, we will skip the proof of the following Propositions.

Proposition 4.4.2. *The minimum energy filter for the constant acceleration model* (4.16) *and linear observation equations* (4.58) *is given by the equations* (4.29) *and* (4.30) *where the function A_k for $x = (E, v)$ is replaced by*

$$A_k(x) = E^\top Q(Ea_k - y_k)a_k^\top, \tag{4.59}$$

and the components $(i, j), i, j = 1, \ldots, 6$ of the matrix $D_k(x) \in \mathbb{R}^{6\times 6}$ are given by

$$(D_k(x))_{i,j} = \zeta_i^k(E)(E^j), \quad E^j := \mathrm{mat}_{\mathfrak{se}}(e_j^6)\,, \tag{4.60}$$

with $\zeta^k(E)(\cdot) : \mathfrak{se}_3 \to \mathbb{R}^6$ given by

$$\mathrm{mat}_{\mathfrak{se}}(\zeta^k(E)(\eta_1)) := \mathrm{Pr}\big(\eta_1^\top Q(Ea_k - y_k)a_k^\top + E^\top Q\eta_1 a_k a_k^\top\big)\,. \tag{4.61}$$

Here, $Q \in \mathbb{R}^{4\times 4}$ is a symmetric and positive definite matrix (cf. (4.19)*). All other expressions from Theorem 4.3.2 stay unchanged.*

Since the linear observation model is a special case of the approach in [17] we only need to modify the corresponding expressions in Algorithm 1 which we summarize in the following Proposition.

Proposition 4.4.3. *The Extended Kálmán Filter for the constant acceleration model* (4.16) *and linear observation equations* (4.58) *is given by Algorithm 1 where the matrix* $H_l := \sum_{z\in\Omega} H_l^k$ *is given by*

$$H_l^k = \begin{pmatrix} \mathrm{vec}_{\mathfrak{se}}(\Pr(E(t_l)^\top e_1^4 a_k^\top))^\top & \mathbf{0}_{1\times 6} \\ \mathrm{vec}_{\mathfrak{se}}(\Pr(E(t_l)^\top e_2^4 a_k^\top))^\top & \mathbf{0}_{1\times 6} \\ \mathrm{vec}_{\mathfrak{se}}(\Pr(E(t_l)^\top e_3^4 a_k^\top))^\top & \mathbf{0}_{1\times 6} \\ \mathrm{vec}_{\mathfrak{se}}(\Pr(E(t_l)^\top e_4^4 a_k^\top))^\top & \mathbf{0}_{1\times 6} \end{pmatrix} \in \mathbb{R}^{4\times 12} \tag{4.62}$$

and the function $J(t)$ *([17, Eq. (52)]) is provided by* (A.50) *in Appendix A.4.*

Remark 4.4.4. Note that (4.62) is different from [17, Eq. (111)] because of the additive instead of multiplicative noise term, and consequently is not consistent with the group structure of SE_3.

4.4.2 Derivations for Non-linear Observations

The adaption of the extended Kálmán Filter [17] to our state (4.16) and observation (4.17) equation is provided by the following Proposition:

Proposition 4.4.5. *The extended Kálmán filter from [17] for our state* (4.16) *and observation* (4.17) *equation is given by Algorithm 1 where the expressions* $J(t)$ *and* H_l *are provided in the equations* (A.50) *and* (A.49)*, respectively, see Appendix A.4.*

4.5 Numerical Geometric Integration

The numerical integration of the optimal state differential equation (4.29) requires respecting the geometry of the Lie group. We use the implicit Lie midpoint rule for integration of the differential equation of the optimal state x^* (4.29) as proposed in [39]. We need to modify the method since we defined state space $\mathcal{G}$ as left invariant Lie group. Instead, in [39], only right-invariant Lie groups are investigated. The adaption to left-invariant Lie groups is straightforward and leads to the following integration schemes: for a discretization $t_0 < t_1 < \cdots < t_n$ with equidistant step size $\delta = t_k - t_{k-1}$ for all k, we integrate the differential equation of the optimal state (4.29) using

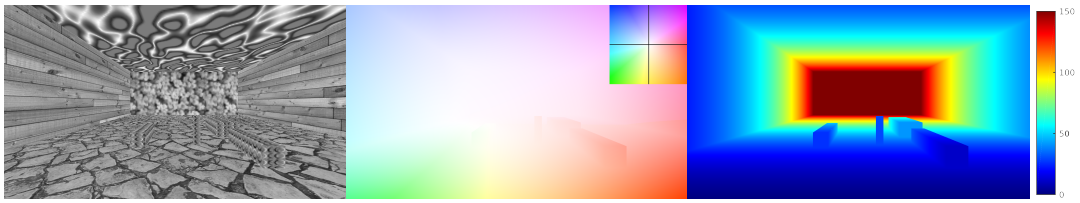


Figure 4.2: Synthetic sequence (left) generated by a simple ray tracer. To provide realistic camera tracks we used ground truth trajectories from the KITTI odometry benchmark and computed the corresponding induced optical flow (mid) and the depth map (right). The corresponding color encodings for direction of optical flow and depth map are on the right hand side.

the *Lie midpoint rule* (3.50):

$$x(t_{k+1}) = x(t_k)\operatorname{Exp}(\Xi)\,, \tag{4.63}$$

$$\begin{aligned}\text{with } \Xi = \delta\big(&f(x(t_k)\operatorname{Exp}(\Xi/2))) \\ &- \operatorname{mat}_{\mathfrak{g}}(P(t_k)\operatorname{vec}_{\mathfrak{g}}(r_t(x(t_k)\operatorname{Exp}(\Xi/2))))\big)\,.\end{aligned} \tag{4.64}$$

For each k the matrix Ξ is received by a fixed point iteration of (4.64). For the integration of equation (4.30), we need to consider that this is a special kind of the *matrix Riccati differential equation* for which methods exist that ensure that the solution is positive definite. As shown in [27], a numerical integration method will preserve positive definiteness if and only if the order of the method is one. By taking down (4.30) as general Riccati differential equation

$$\dot{P}(t) = A(t)P(t) + P(t)A(t)^\top - P(t)B(t)P(t) + C(t)\,, \tag{4.65}$$

with symmetric matrices $B(t)$ and $C(t)$, the implicit Euler integration method is given by

$$\begin{aligned}P(t_{k+1}) =& P(t_k) + \delta\big(AP(t_{k+1}) + P(t_{k+1})A^\top \\ &- P(t_{k+1})BP(t_{k+1}) + C\big),\end{aligned} \tag{4.66}$$

which can be expressed by the *algebraic Riccati equation* for which an unique solution exists [55] that can be found by standard solvers, e.g. CARE.

4.6 Experiments

In this experimental section, we will evaluate the accuracy of the proposed minimum energy filter for ego-motion estimation. First we will provide ex-

periments on synthetic data to exclude external influences and to show robustness against measurement noise. Then we will consider real world experiments on the challenging KITTI benchmark and compare our method with a state-of-the-art method [37]. Finally, to evaluate the theoretical performance of the filter, we will also compare to the state-of-the-art extended Kálmán filter [17] in a controlled environment.

4.6.1 Synthetic Data

Before considering real-life sequences, we first evaluate synthetic scenes to have full control on the regularity on the camera track. We generate 3D scenes by raytracing simple geometric objects (cf. Fig. 4.2), which also enables us to acquire correctly induced optical flow and depth maps. In order to gain a realistic camera behavior, we use the tracks from the KITTI visual odometry training benchmark which were determined by an inertial navigation system in a real moving car. We start with considering the case of perfect measurements (section 4.6.1) and demonstrating robustness against different kinds of noise in section 4.6.1.

Evaluation on Noiseless Measurements First, we evaluate the proposed filter on the true optical flow. To avoid overfitting, we set a relatively small weight onto the weighting matrix for the data term, i.e. $Q = 0.1/n$, where n is the number of observations. We set the weighting matrix S to the block diagonal matrix containing the matrices S_i, i.e.

$$S = \text{blockdiag}(S_1, \dots, S_m)\,, \tag{4.67}$$

where m denotes the order of the kinematic. The single blocks are given for $i = 1, \dots, m$ through

$$S_i = \text{diag}(s_1, s_1, s_1, s_2, s_2, s_2)$$

with $s_1 = 10^{-2}$ and $s_2 = 10^{-5}$. The decay rate is set to $\alpha = 2$ and the integration step size to $\delta = 1/50$.
As demonstrated in Fig. 4.3, the proposed filters of different order show a similar rotational error since the ground truth rotation is often constant and influenced by (physical) noise. That is possibly caused by the low temporal resolution of 10 Hz, not being able to give sufficient information on the kinematics. On the contrary, in the translational part we can see that the higher-order models work significantly better than our first-order model [13], but that third- and fourth-order methods perform fairly the same. From this we can conclude that kinematics of fifth- or even higher-order will not improve performance regarding this kind of camera tracks.

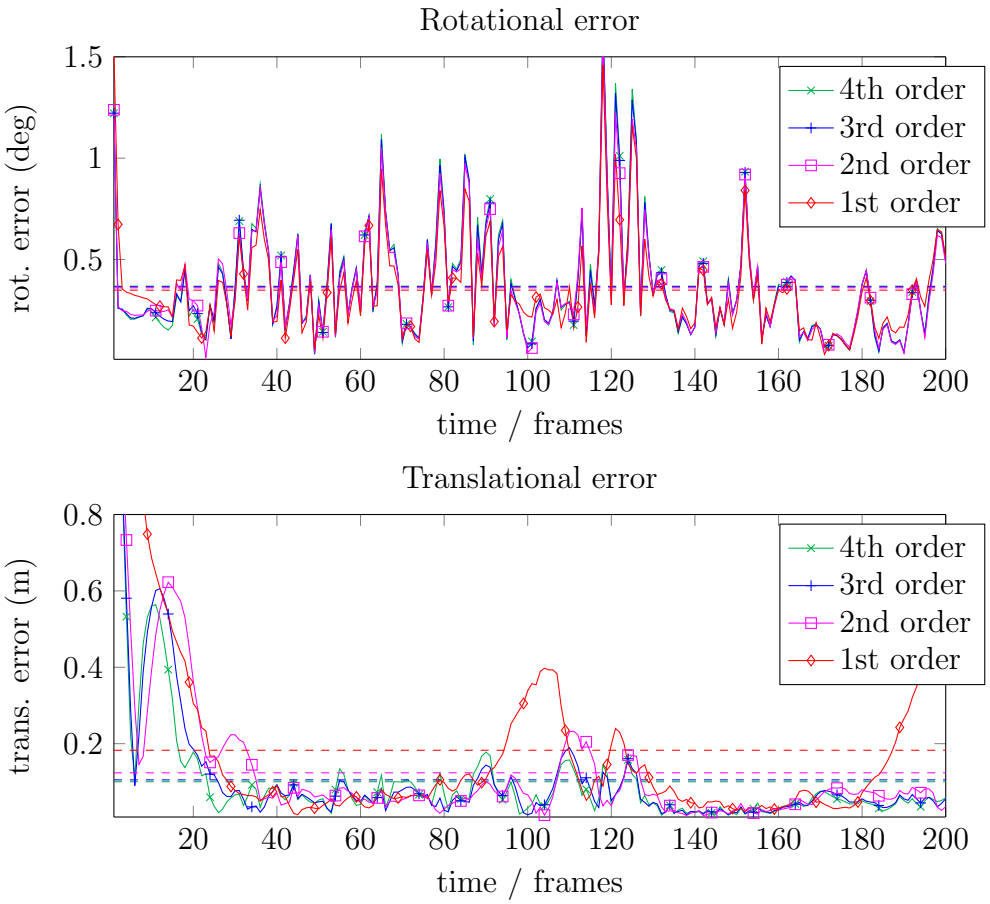


Figure 4.3: Comparison of the rotational error in degree (top) and the translational error in meters (bottom) of the proposed minimum energy filters with kinematic state equations of orders one (see [13]) and two, three and four (this work). The dotted lines show the error averaged over all frames. We used a real camera track from sequence 0 of the KITTI visual odometry benchmark and used it to generate synthetic sequences with induced depth map and optical flow. The rotational errors are similar through all frames although the higher-order methods converge faster in the first iterations. In frames 20–90, the motion of the camera is almost constant and the filters perform similarly. However, the translational error of the first order method significantly changes in frames 90–150 and 175–200 because the constant velocity assumption is violated by curves in the trajectory.

Figure 4.4: Different noise models for the observed data (optical flow, cf. Fig. 4.2): top left: additive Gaussian noise ($\mu = 0$, $\sigma^2 = 0.001$), top right: additive uniform noise ($\mu = 0$, $\sigma^2 = 0.001$), bottom left: multiplicative Gaussian noise ($\mu = 1$, $\sigma^2 = 1$), bottom right: multiplicative uniform noise ($\mu = 1$, $\sigma^2 = 1$).

Evaluation on Noisy Measurements To evaluate the robustness against noise, we altered the true optical flow measurements by multiplicative and additive noise, each being distributed uniformly or Gaussian, see Fig. 4.4. The proposed method determines camera motion using the same parameters as in section 4.6.1. Comparison to the ground truth is achieved using the geodesic distance on SE_3 in order to avoid two separate error measures for translation and rotation, i.e.

$$d_{\mathrm{SE}_3}(E_1, E_2) := \|\mathrm{vec}_{\mathfrak{se}}(\mathrm{Log}(E_1^{-1}E_2))\|_2 \,. \tag{4.68}$$

The results in Tab. 4.1 show that higher-order models outperform the first-order model with the exception of very high noise levels where the data does not contain sufficient information to correctly estimate a higher-order kinematic.

Remark 4.6.1. Note that our model currently does not model noise on depth maps explicitly since it only allows additive noise on the *flow measurements* as introduced in (4.17). However, we think that the noise term ϵ should also compensate small deviations of the depth.

Table 4.1: Quantitative evaluation of proposed methods (order 1 to 4) measuring the geodesic error (cf. (4.68)) w.r.t. ground truth camera motion. As input data we used noisy flow observations with the following noise models: additive Gaussian (AG, $\mu = 0$), additive uniform (AU, $\mu = 0$), multiplicative Gaussian (MG, $\mu = 1$) and multiplicative uniform (MU, $\mu = 1$) for different variances σ^2. For intense noise (multiplicative: $\sigma^2 > 10^{-1}$, additive: $\sigma^2 > 10^{-4}$), the first-order method performs better than higher-order models since it is more robust against noise. In contrast, for moderate noise levels, higher-order kinematics are more appropriate.

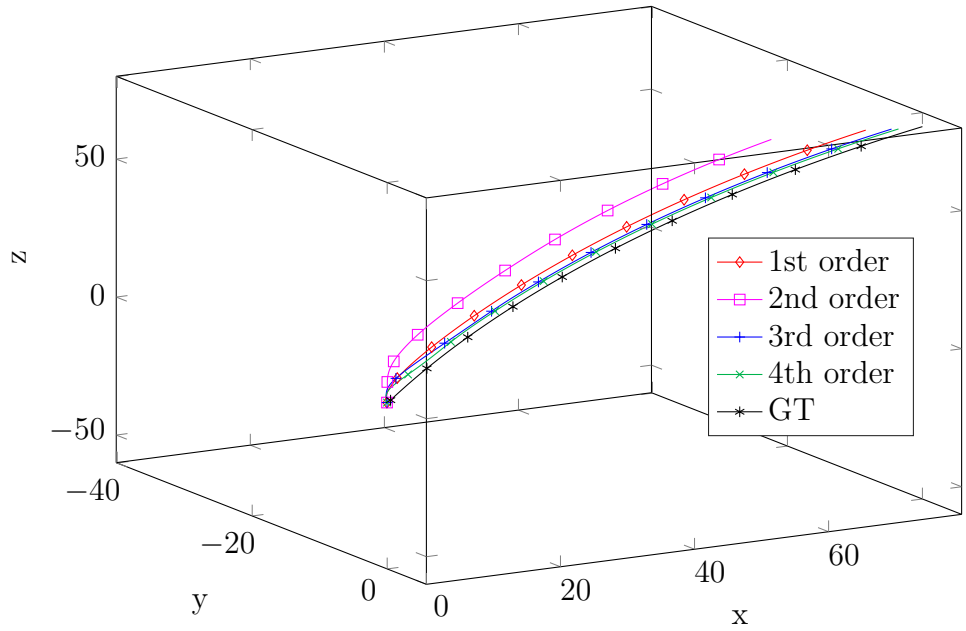
noise	σ^2	1st order	2nd order	3rd order	4th order
MG	10^0	**0.2162**	0.2759	0.2821	0.2866
MU		**0.2856**	0.3840	0.3705	0.3705
MG	10^{-1}	0.1597	0.1644	0.1485	**0.1423**
MU		**0.2072**	0.2596	0.2367	0.2287
MG	10^{-2}	0.1417	0.1184	0.1041	**0.1011**
MU		0.1517	0.1353	0.1143	**0.1082**
MG	10^{-3}	0.1283	0.0987	0.0844	**0.0808**
MU		0.1300	0.0952	0.0808	**0.0777**
noise	σ^2	1st order	2nd order	3rd order	4th order
AG	10^{-3}	**0.2859**	0.4355	0.4318	0.4385
AU		**0.4835**	0.7431	0.7175	0.7071
AG	10^{-4}	**0.1598**	0.1695	0.1688	0.1701
AU		**0.2176**	0.2341	0.2216	0.2193
AG	10^{-5}	0.1384	0.1157	0.1010	**0.0974**
AU		0.1263	0.1130	0.1009	**0.0968**
w/o	0	0.1264	0.0893	0.0783	**0.0757**

Evaluation of Kinematics In the last section we showed that the proposed method is robust against different kinds of measurement noise. Now we evaluate the proposed minimum energy filters with higher-order kinematic model for camera tracks of different complexity. For this purpose, we generate camera tracks for the kinematic models (first to fourth order) by (geometric) numerical integration of corresponding differential equation (4.54) for $m \in \{1, 2, 3, 4\}$ where we set $v_0 \equiv 0$. In order to obtain reasonable paths we use non-trivial initializations for $(E_0, v_1^0, v_2^0, v_3^0)$. Then we generate synthetic sequences for the different kinematic tracks and use the ground truth optical flow and depth maps as input for the proposed filters.
The proposed method uses the parameters $Q = 0.1n^{-1}\mathbb{1}_2$ with $n = 1000$; and S was chosen as in (4.67), whereas $s_1 = 1$, $s_2 = 0.001$ and $\alpha = 0$.
In Fig. 4.2 we visualize the geodetical error (4.68) as well as the camera track reconstructions. It becomes apparent that for a camera track with constant velocity (Fig. 4.5b) the minimum energy filter with first-order kinematics [13] performs best and reaches the highest accuracy. For the other tracks with higher-order kinematics (cf. Figures 4.4d, 4.3f and 4.2h), the proposed filters with higher-order kinematic model work superiorly to [13].

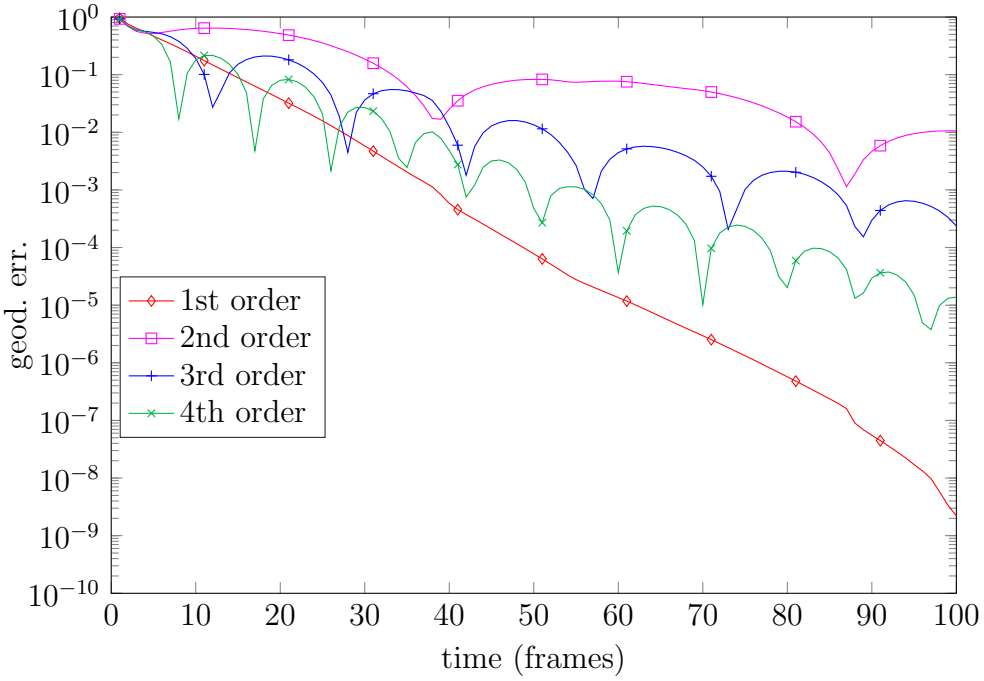
4.6.2 Evaluation with Realistic Observations

In order to demonstrate that the minimum energy filter with higher-order state equations also works under real world conditions, we evaluate our approach on the challenging KITTI odometry benchmark [36]. This benchmark does not contain ground truth data for optical flow, and depth maps can only be obtained from external laser scanners. Thus, we compute optical flow and depth maps in a preprocessing step using the freely available method by Vogel *et al.* [82] which only requires image data. Although this method is the top ranked method on the KITTI optical flow benchmark, its results still contain relevant deviations from the true solution and thus provide realistic observation noise to evaluate the performance of our proposed filter. As the preprocessed data of [82] is dense optical flow, it causes a high computational effort. Therefore, we only use a sparse subset of data points which are selected randomly. In section 4.6.2 we will show that a small number of observations is sufficient for good reconstructions.

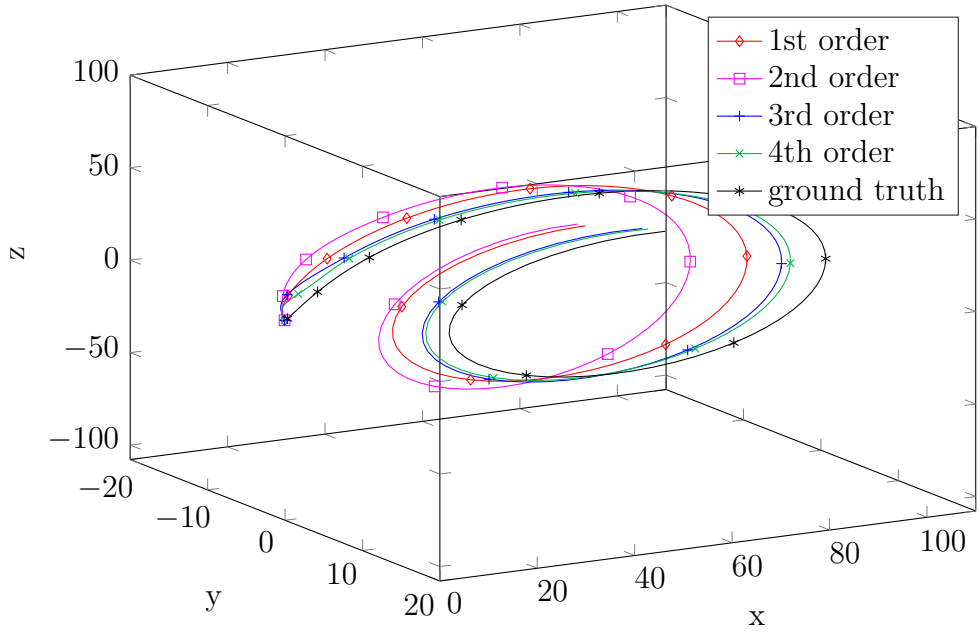
Quantitative Evaluation of First and Higher-Order Models For our quantitative evaluations on the KITTI benchmark in Table 4.2, we initialize our first [13] and higher-order approaches with the corresponding identity element on the Lie group, i.e. $x_0 = \mathrm{Id}$, and set the corresponding matrices P_0 to the identity matrices. The quadratic forms of the penalty term of the



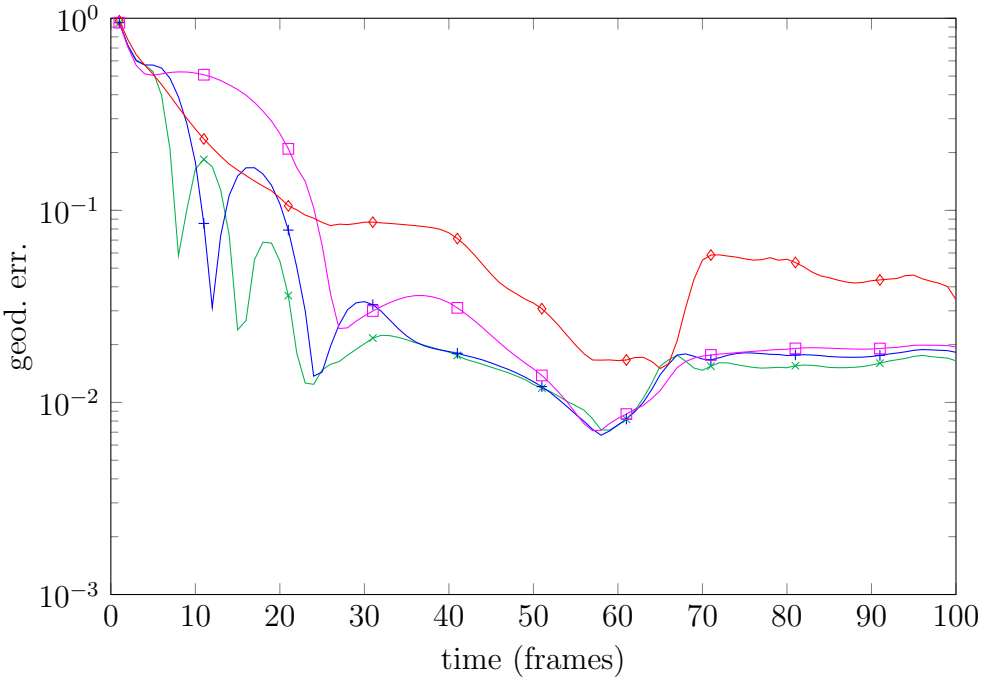
(a) reconstruced track: first-order kinematics



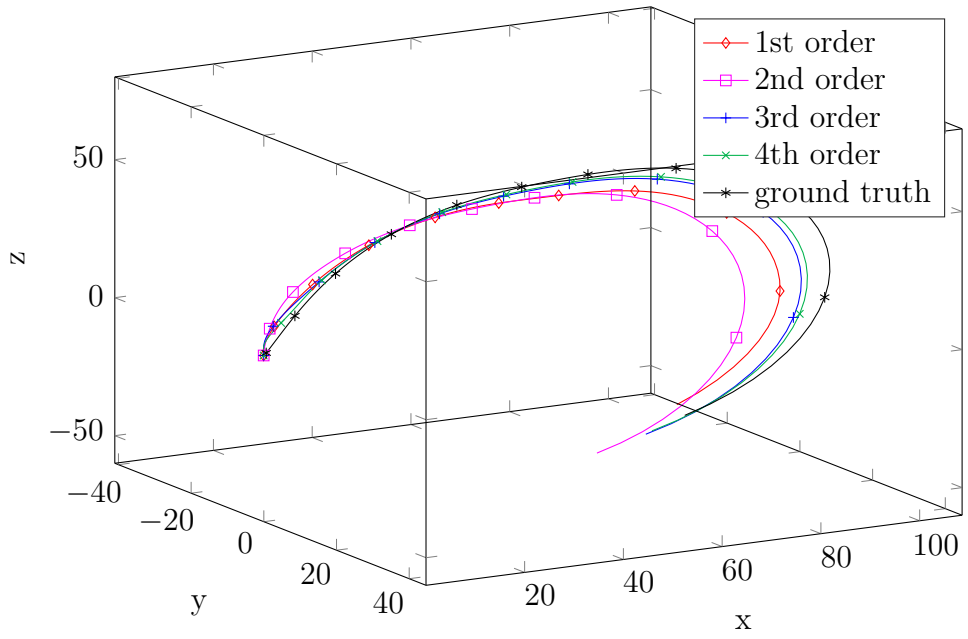
(b) geodetical error on a first-order kinematic track



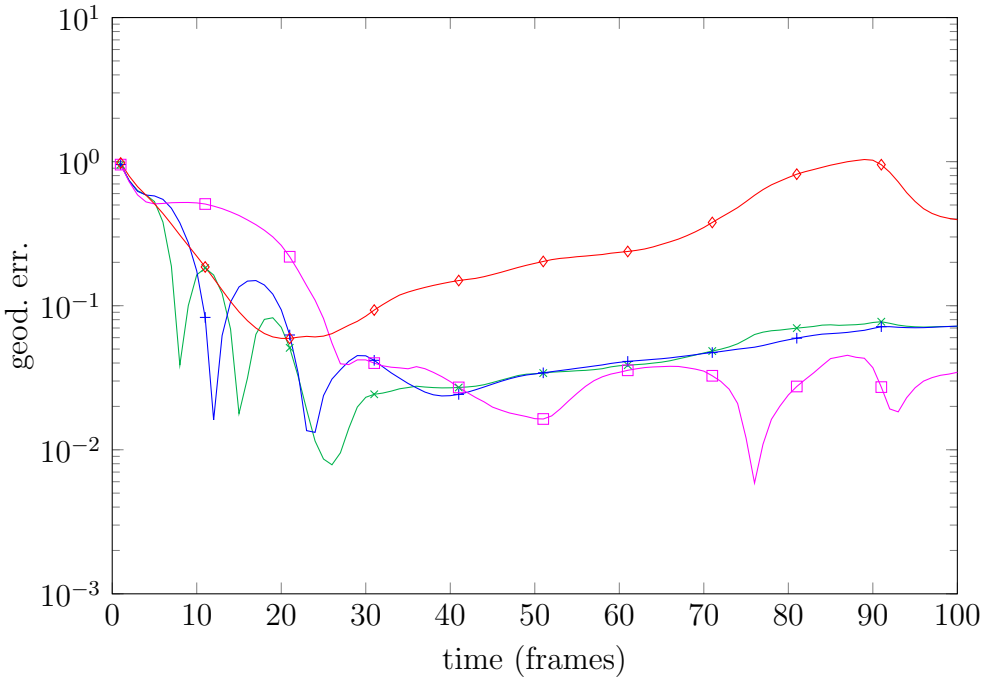
(c) reconstruced track: second-order kinematics



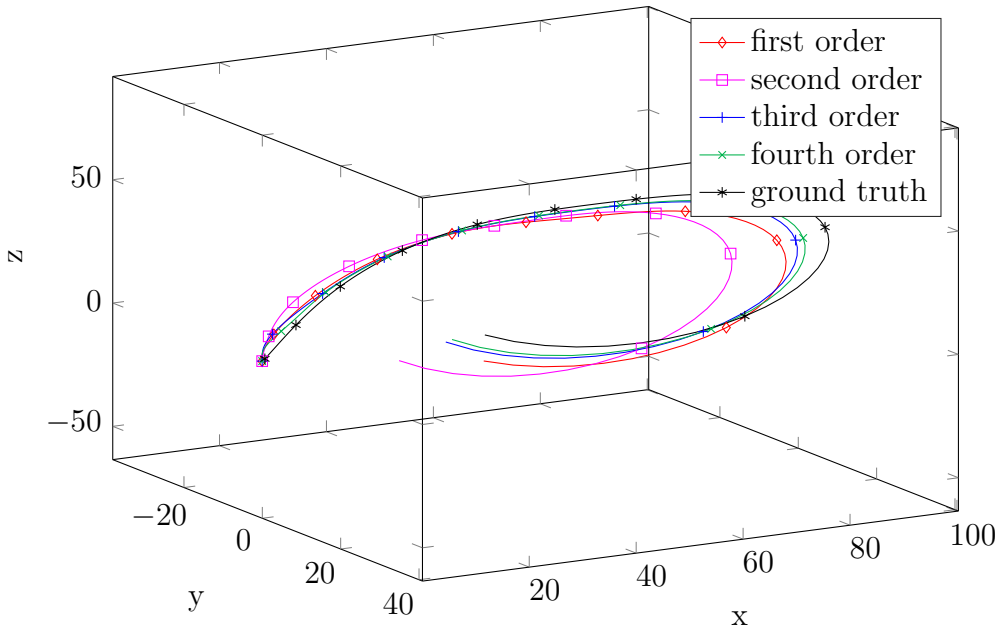
(d) geodetical error on a second-order kinematic track



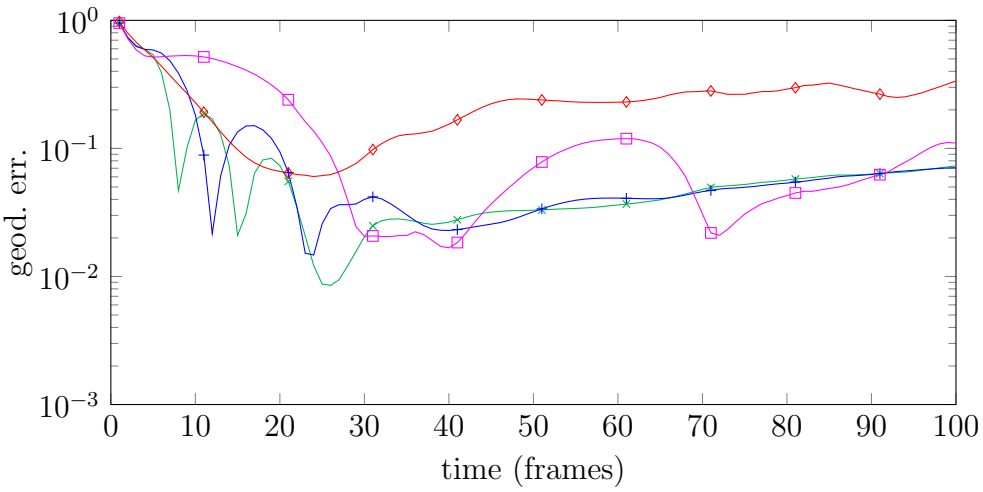
(e) reconstruced track: third order kinematics



(f) geodetical error on a third order kinematic track



(g) reconstruced track: fourth order kinematics



(h) geodetical error on a fourth order kinematic track

Figure 4.2: Reconstruction of the camera tracks (top row) and evaluation of the geodetical error w.r.t. ground truth (bottom row) as computed by the proposed filter with kinematics of order 1, 2, 3 and 4. We evaluated the performance on simulated camera tracks with kinematic models of different orders: constant velocity (b),(a), constant acceleration (d),(c) as well as third (f),(e) and fourth (h),(g) order kinematics. In the constant velocity scenario (b), the first-order filter performs best. On the other scenarios (d), (f), (h), the higher-order methods are superior and lead to the best path reconstructions.

model noise δ are set as shown in (4.67) with $s_1 = 10^{-2}$ and $s_2 = 10^{-5}$. To increase the influence of the data term, we set the weighting matrix to

$$Q := \tfrac{1}{n}\mathbb{1}_2\,, \quad n = 1000\,. \tag{4.69}$$

On the one hand, this high-weighting leads to less smoothed camera trajectories, but on the other hand minimizes the observation error, which is desirable for visual odometry applications. For comparison we also present in Table 4.2 the performance measures of the odometry method [37].
We emphasize that the first-order approach [13] and second-order method from Theorem 4.3.2 perform better in the case of camera motion reconstruction than the proposed higher-order (> 2) models with generalized kinematics from Theorem 4.3.8. The reason for that is that the real camera motion is influenced by model noise, induced by jumps of the camera, to which the first-order method can adapt faster. Higher-order models smooth the camera trajectories, which in this case is unfortunate. However, they will be beneficial if the actual camera motion behaves according to the models, as shown in the experiments in section 4.6.1.
Note that our method currently is not designed to be robust against outliers in the observation. Using outlier rejection models would increase the accuracy of our method. In contrast, the approach of Geiger *et al.* [37] uses additional precautions to eliminate violation of the assumption of a single rigid body motion, see e.g. sequence 3 in Table 4.2. However, in this work we focus more on the modeling and optimization than on optimal results because the latter require assumptions and heuristics making the whole model involved.

Determination of Optimal Number of Observations Since the evaluation of the functions A_z and D_z in Theorem 4.3.2 as well as the accurate numerical integration in section 4.5 are expensive, we are looking for a good trade-off between the number of required measurements and accuracy. In Table 4.3 we evaluate the geodetical error for a different number of observations n. For $n = 1$, our proposed filters do not converge since they are numerically instable. For $n = 5, \ldots, 20$, the geodetical error is fairly small but reaches a minimum for $n = 50$. For $n < 5$, the error increases because the ego-motion cannot be reconstructed uniquely (cf. Five-point-algorithm [62]). Likewise, for $n > 50$, the error rises due to noisy measurements averaged by the filter.

Influence of the Decay Rate α In real sequences, the motion is usually not uniform and changes due to acceleration and curves. As demonstrated earlier, higher-order state equations that model accelerations, jerks,

Table 4.2: Quantitative evaluation of rotational (in degrees) and translational (in meters) error on the first 200 frames of the training set of the KITTI odometry benchmark. We compared the proposed higher-order method (i.e. 2nd to 4th) with our first-order method from [13]. As a reference method, we also evaluated the approach by Geiger *et al.* [37]. The first and second-order methods outperform the higher-order methods since they can fit more easily to the non-smooth ego-motion data.

	sequence	00	01	02	03	04	05
trans. error	(Geiger [37])	**0.0272**	**0.0572**	0.0255	**0.0175**	**0.0161**	**0.0185**
	1st order [13]	0.0284	0.0759	**0.0188**	0.0804	0.0165	0.0188
	2nd order	0.0356	0.0786	0.0289	0.0938	0.0210	0.0288
	3rd order	0.0358	0.0784	0.0290	0.0924	0.0216	0.0286
	4th order	0.0347	0.0782	0.0275	0.0918	0.0211	0.0277
rot. error	(Geiger [37])	**0.1773**	**0.1001**	0.1552	**0.1829**	0.0970	0.1539
	1st order [13]	**0.1773**	0.1139	0.1504	0.2246	0.0836	**0.1454**
	2nd order	0.1996	0.1183	**0.1430**	0.2448	**0.0805**	0.1566
	3rd order	0.2402	0.1348	0.1872	0.2719	0.1090	0.1971
	4th order	0.2795	0.1466	0.2223	0.3120	0.1479	0.2335
	sequence	06	07	08	09	10	
trans. error	(Geiger [37])	**0.0118**	**0.0160**	0.1166	**0.0175**	**0.0147**	
	1st order [13]	0.0122	0.0174	**0.1142**	0.0193	0.0205	
	2nd order	0.0153	0.0284	0.1153	0.0293	0.0417	
	3rd order	0.0175	0.0268	0.1153	0.0258	0.0342	
	4th order	0.0140	0.0257	0.1155	0.0240	0.0317	
rot. error	(Geiger [37])	0.0829	0.1770	0.1589	0.1166	0.2001	
	1st order [13]	0.0765	**0.1654**	**0.1444**	**0.0911**	0.1829	
	2nd order	**0.0703**	0.2113	0.1676	0.1167	0.2388	
	3rd order	0.0875	0.2362	0.2053	0.1335	0.2628	
	4th order	0.1045	0.2709	0.2318	0.1630	0.2956	

Table 4.3: Determination of the optimal number of measurements n. We evaluated the mean geodetical error of our filter with different kinematic models (first to fourth order) on a short sequence (10 frames) for different numbers n of observations. Since the n observations are selected randomly, we repeated the experiment 50 times and averaged finally, to find a representative value. We found an optimal number of measurements for $n = 50$.

n	1st order	2nd order	3rd order	4th order
1000	0.1205	0.1361	0.1311	0.1290
500	0.1070	0.1174	0.1116	0.1096
200	0.0915	0.0945	0.0902	0.0890
100	0.0764	0.0764	0.0739	0.0733
50	**0.0667**	**0.0651**	**0.0638**	**0.0637**
20	0.0715	0.0703	0.0687	0.0684
15	0.0709	0.0691	0.0674	0.0672
12	0.0718	0.0720	0.0702	0.0699
10	0.0749	0.0735	0.0716	0.0712
9	0.0751	0.0747	0.0726	0.0722
8	0.0772	0.0762	0.0742	0.0738
7	0.0735	0.0733	0.0717	0.0714
6	0.0786	0.0776	0.0757	0.0753
5	0.0789	0.0797	0.0778	0.0774
4	0.0856	0.0859	0.0837	0.0831
3	0.0917	0.0951	0.0928	0.0921
2	0.1005	0.1085	0.1058	0.1051

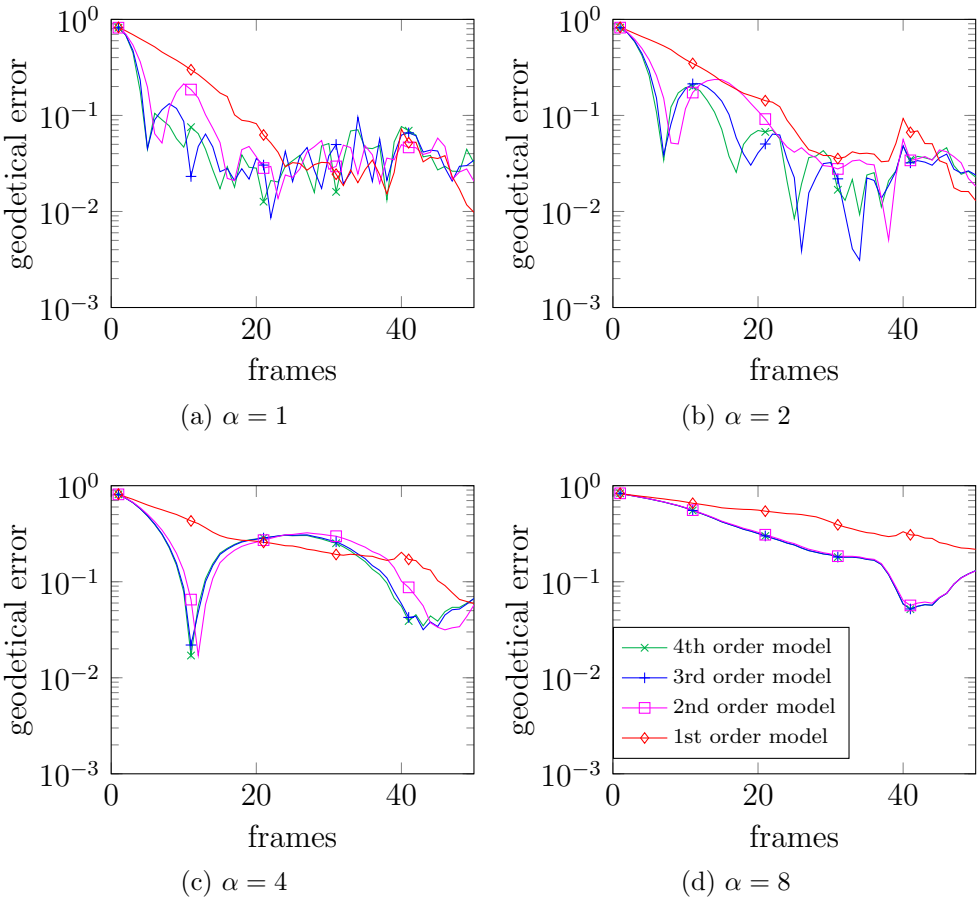
etc. usually converge faster and yield a better accuracy. However, higher-order models are delayed since it takes some time until the information from the observation is transported to the lowest layer. Furthermore, if the motion changes quickly, then higher-order models will still propagate wrong kinematics. For this reason, in [72] a decay $\alpha > 0$ rate is introduced and also adopted to our model. For $\alpha = 0$, all past information is preserved in the propagation within the filter. For larger values of α, old information about the trajectory has lower influence on the filter and is less respected in future. For the experiments we use the weighting matrix $Q = n^{-1}\mathbb{1}_2$, where n is the number of measurements. Furthermore, we use S as in (4.67) with the values $s_1 = 5 \cdot 10^{-2}$, $s_2 = 5 \cdot 10^{-4}$. The integration step size is set to $\delta = 1/50$.
In Fig. 4.3, we visualize the influence of different values of α on the minimum energy filters of order 1 to 4. For small decay rates α, the filters will converge faster over time, but will also cause errors if the kinematics change. On the other hand, large decay rates adapt more easily to spontaneous changes of kinematics. The filters take longer to converge, however.

4.6.3 Comparison with the Extended Kálmán Filters

Experiments with Linear Observation Equation For the experiments in Fig. 4.4 we use four observation equations ($n = 4$), and the vectors a_k in (4.58) are chosen as

$$a_k = e_k^4\,, \quad k \in [4]\,, \tag{4.70}$$

to extract information from all directions. We generate the ground truth from an arbitrary initialization by integration of (4.16) with multivariate Gaussian noise with mean $\mathbf{0}_{12}$ and diagonal covariance matrix $S = \mathbb{1}_{12}$. As shown in [17], we integrate the ground truth with ten times smaller step sizes than the filtering equations of extended Kálmán and minimum energy filter. Afterwards we generate the observations with (4.62) and Gaussian noise with covariance $Q = 10^{-8}\mathbb{1}_4$ and set the covariance matrices S and Q in Algorithm 1 to the same values. However, the matrix Q for the minimum energy filter in Proposition 4.4.2 is set to $Q = 100\,\mathbb{1}_4$ to give more weight to the observations for faster convergence. Note that for the extended Kálmán Filter the choice $Q = 100\,\mathbb{1}_4$ leads to a worse performance, which is why we use the true covariance instead.
As a reference, we apply our own implementation of the method by Bourmaud *et al.* [17] adapted to our model. The results are demonstrated in Fig. 4.4. We suppose that the main reason for the different performances is that we compare a *second-order* (minimum energy filter) with a *first-order* (extended Kálmán) filter.



(a) $\alpha = 1$ (b) $\alpha = 2$

(c) $\alpha = 4$ (d) $\alpha = 8$

Figure 4.3: Evaluation of the translational error (in meters) of the minimum energy filter regarding the first, second, third and fourth order state equation on the first 50 frames of sequence 0 of the KITTI odometry sequence. For small values of the decay rate α, the filter memorizes past information and converges fast, see Fig. (4.3a). Although higher-order filters converge faster, they cause oscillation due to the time delay that is required to propagate information into higher-order derivatives of the kinematics. Since for large values of α past information is neglected, the filters converge slower and the difference between second, third and fourth order models become smaller, while the oscillations disappear. Please note that for this experiments the weighting matrices S and Q are kept fixed. To further reduce the error for large α we propose to adapt the weights.

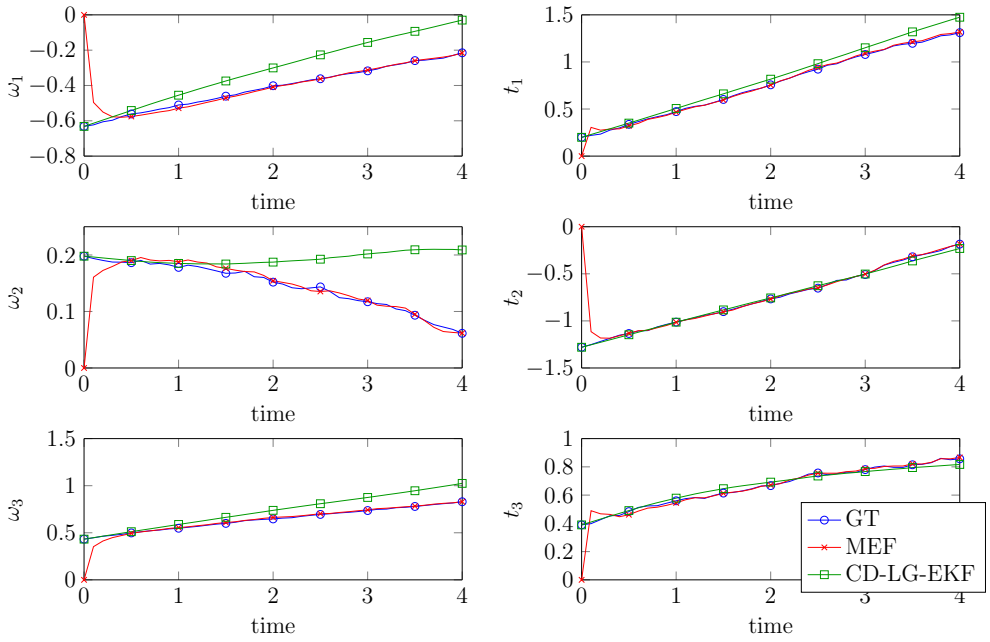


Figure 4.4: Comparison between minimum energy filter with second-order kinematics (MEF) (red, cross) and extended Kálmán filter (CD-LG-EKF) [17] (green, square) with state equation (4.16) and observation equation (4.58) as derived in Properties 4.4.2 and 4.4.3, respectively. We plotted the six components of the rigid motion of the ground truth (GT) (blue, circle), the extended Kálmán filter, and the minimum energy filter, i.e. $(\omega_1, \omega_2, \omega_3, t_1, t_2, t_3)^\top := (\mathrm{vec}_{\mathfrak{g}}(\mathrm{Log}_{\mathcal{G}}(G))))_{1:6}$. Here, G is the corresponding element of the Lie group $\mathcal{G}$. Further, we set the discretization step size to $\delta = 0.1$. Although we initialized the extended Kálmán filter with the ground truth solution and added only little observation noise, it diverges after a few steps whereas the minimum energy filter converges from a wrong initialization to the correct solution within a few steps. The reason for that is that the approach [17] only uses first-order approximation, whereas the minimum energy filter also includes second-order derivatives of the observation function.

Extended Kálmán Filter for Non-linear Observations We were not able to obtain convergence of this filter from a trivial (chosen as identity element of the Lie group) or ground truth initialization. Since the extended Kálmán did not converge for linear observations (4.4.1) from wrong initializations, we presume that the non-linearities of our observation equations are intractable for the approach from [17].

4.7 Limitations

Our proposed method requires good measurements in terms of optical flow and depth maps in order to reconstruct the camera motion correctly. Although we showed on synthetic data that the proposed method is robust against different kinds of noise, it is not robust against outliers, caused by independently moving objects that violate the static scene assumption, or simply wrong computations of optical flow and depth maps. Making our approach robust as component of a superordinate processing stage, however, is beyond the scope of this work and left for future work.
In addition to optical flow, the proposed method requires depth information which is expensive to obtain if not available anyway, e.g. in stereo camera setups.

4.8 Summary

In this chapter we generalized the camera motion estimation approach [13] from a model with constant velocity assumption to a more realistic model with constant acceleration assumption as well as to a kinematic model which respects derivatives of any (fixed) order. For the resulting second-order minimum energy filter with higher-order kinematics, we provided all necessary derivations and demonstrated that our approach is superior to our previous method [13] for both synthetic and real-life data. We also compared our approach to the state-of-the-art *continuous-discrete extended Kálmán filter on connected unimodular matrix Lie groups* [17] and showed that in both cases the minimum energy filters is superior since it converges from imperfect initializations to the correct solutions.

Chapter 5

Joint Filtering of Disparity Map and Camera Motion

5.1 Introduction

5.1.1 Overview

In the last chapter we considered the scenario of reconstructing the unknown camera motion from observed optical flow and depth maps. In practice, this usually requires a stereo approach to find the depth map. In this chapter, we want to address the reconstruction of the scene structure of images and videos itself, which is a fundamental building block in computer vision and is required for plenty of applications, e.g. autonomous driving, robot vision and augmented reality. Although stereo methods usually lead to exact reconstruction and work fast, they require calibration of the camera setup and, due to the second camera, these systems are more expensive than single camera systems. Therefore, in this chapter, we will focus on the monocular approach that consists of reconstructing the scene structure based on the data gained by a *single* moving camera. In contrast to the stereo setting, this problem is ill-posed because of the unknown motion parallax. On the other hand, monocular approaches can be used as fallback level in a stereo system if one camera fails.
Similarly to the approach within the last chapter, we want to use temporal information for smoothing and propagation. Again, in this scenario, the state variables, e.g. camera motion, do not evolve on an Euclidean space but a more general Lie group, we cannot use classical filters, such as *extended Kálmán* filters [35]. Moreover, other state-of-the-art non-linear filters, such as *particle filters* [29], that can be applied to specific Lie groups [53], cannot be easily extended to high dimensional problems [25]. Due to these math-

ematical problems we will use the recently proposed *minimum energy filter* on compact Lie groups [73] that minimizes a quadratic energy function to penalize deviations of the filtering equations by means of optimal control theory. This filter was shown to be superior to extended Kálmán filters on the low dimensional Lie group SE_3 [12]. We will demonstrate that this approach can also be successfully applied to high dimensional problems, enabling *joint* optimization of camera motion and disparity map. As in [12], we will also incorporate higher-order kinematics of the camera motion. To be robust against outliers, we will extend the approach of [73] from quadratic energy function to a generalized Charbonnier energy function.

5.1.2 Related Work

Plenty of methods for depth or disparity map estimation were published during the last decade. We distinguish between stereo methods (that benefit from the additional information gained from the camera setup) and monocular methods. Recognized stereo methods include [44, 67, 82] that use the known distance of the cameras (baseline) for accurate triangulation of the scene. These methods also enable reducing the computational effort by using epipolar geometry and by combining local and global optimization schemes. Monocular methods [26, 8, 31, 30, 65, 61, 16] benefit from less calibration effort in comparison to stereo methods, but suffer from a peculiarity of the mathematical setup that prevents to reconstruct the scale of the scene uniquely. To increase the robustness and the accuracy of the reconstruction, modern methods incorporate multiple consecutive frames into the optimization procedure. Well-known is bundle adjustment [80] which optimizes a whole trajectory but cannot be used in online approaches such as sliding window [9] or filtering methods [8, 16]. Filtering methods usually require a suitable modeling of the unknown a posteriori distribution. However, they suffer from the drawback that the definition of probability densities on curved spaces, such as Lie groups, is complicated, although successful strategies to find a solution to this problem have been developed [17, 20, 53]. Zamani et al. [88] introduces so-called *minimum energy filters* for linear filtering problems for compact Lie groups based on optimal control theory and the recursive filtering principle of Mortensen [58]. This approach was generalized to (non-) compact Lie groups in [73] and applied to a *non-linear* filtering problem on SE_3 for camera motion estimation [13].

5.1.3 Contributions

In this chapter our main contributions add up

- to provide a mathematical filtering framework for *joint* monocular camera motion and disparity map estimation including higher-order kinematics,
- to introduce a *novel disparity Lie group for inverse depth maps* which avoids additional positive depth constraints such as barrier functions,
- to solve the corresponding challenging *non-linear* and *high-dimensional* filtering problem on a *product Lie group* by using novel *minimum energy filters*,
- to provide a *generalized Charbonnier* energy function instead of a quadratic energy function [73], which results in robustness against outliers,
- to incorporate a *spatial regularizer* within the energy function to increase the accuracy of the disparity map in regions close to the epipole.

5.2 Model

In this section we introduce the mathematical framework of joint monocular camera motion and disparity map estimation from the point of view of temporal filtering. Note, that we use the notion *disparity map* for the inverse of the depth map in this work without using the baseline that is required in stereo settings. In classical filtering theory one wants to determine the most likely state of an unknown process $x = x(t)$ modeled by a perturbed differential equation $\dot{x}(t) = f(x(t)) + \delta(t)$ based on prior perturbed observations $y(s) = h(x(s)) + \epsilon(s)$ for $s \leq t$, which results in a *maximum* a posteriori problem. In this work, we require the state space of x to be a Lie group $\mathcal{G}$ which we need to describe non-Euclidean expressions such as camera motions. Using the expressions $\delta = \delta(t)$ and $\epsilon = \epsilon(t)$ to represent model noise and observations noise, respectively, the resulting filtering equations can be written as

$$\dot{x}(t) = x(t)\big(f(x(t)) + \delta(t)\big)\,, \quad x(t_0) = x_0\,, \tag{5.1}$$

$$y(t) = h(x(t)) + \epsilon(t)\,. \tag{5.2}$$

The state equation (5.1) is modeled on a Lie group $\mathcal{G}$ by means of the tangent map of the left translation at identity and functions $f, \delta \in \mathfrak{g}$ such that $\dot{x}(t) \in T_x\mathcal{G}$. In the following sections we will introduce the state space of x, the propagation functions f and the observation function h.

5.2.1 State Space

The camera motion is modeled on the Special Euclidean group $\mathrm{SE}_3 := \{\left(\begin{smallmatrix} R & w \\ 0 & 1 \end{smallmatrix}\right) | R \in \mathrm{SO}_3, w \in \mathbb{R}^3\}$, and we also use a higher-order kinematics (e.g. acceleration of camera) modeled by a vector $v \in \mathbb{R}^6$. The disparity map can be represented by a large vector $d_i \in \mathbb{R}^{|\Omega|}$, resulting in an own dimension for each pixel in the image. However, the depth must always be positive and we want to avoid additional constraints within our optimization. Therefore, we introduce a novel Lie group for the inverse of the depth, denoted by $(0,1)^{|\Omega|}$ which is defined as follows:

Definition 5.2.1 (Lie group $(0,1)^n$ (Disparity group))**.** By denoting $d_i(z,t) := \frac{1}{d(z,t)} \in (0,1)$ the inverse of the depth we define the Lie group $(0,1)^n$ with group action for $x, y \in (0,1)^n$ as

$$(x,y) \mapsto \big((x^{-1} - \mathbf{1}) \cdot (y^{-1} - \mathbf{1}) + \mathbf{1}\big)^{-1} = \frac{xy}{\mathbf{1} - x - y + 2xy}\,.$$

All operations apply component-wise to the vectors involved. The (Lie group inverse) can be computed as $i(x) := \mathbf{1} - x$. This results in the identity element $\mathrm{Id} = \frac{\mathbf{1}}{\mathbf{2}}$, i.e. a vector full of $1/2$.

For the following calculations and numerical treatment such as numerical integration we require to calculate the unique exponential and logarithmic maps of the proposed Lie group $(0,1)^n$. These are provided by the following Proposition.

Proposition 5.2.2. *The exponential on the Lie group* $(0,1)^n$, $\mathrm{Exp}_{(0,1)^n} : \mathbb{R}^n \to (0,1)^n$ *is given through*

$$x \mapsto \frac{e^{4x}}{e^{4x} + 1}\,. \tag{5.3}$$

Proof. The derivation of the exponential maps requires to calculate the geodesics γ on the Lie group $(0,1)^n$ as described in definition 3.2.10. Thus, we need to solve the following initial value problem for $\gamma_\eta(t)$:

$$\gamma_\eta(0) = \mathrm{Id}_{\mathcal{G}}, \quad \dot{\gamma}_\eta(t) = T_{\mathrm{Id}} L_{\gamma(t)} \eta, \quad \text{for all } t \in \mathbb{R}. \tag{5.4}$$

The tangent map of the left translation can be calculated as follows:

$$
\begin{aligned}
T_{\mathrm{Id}}L_x =& \mathbf{D}_y L_x y|_{y=\mathrm{Id}} \\
=& \mathbf{D}_y \big((x^{-1}-\mathbf{1})(y^{-1}-\mathbf{1})+\mathbf{1}\big)^{-1}|_{y=\mathrm{Id}} \\
=& (-1)\big((x^{-1}-\mathbf{1})(y^{-1}-\mathbf{1})+\mathbf{1}\big)^{-2}(x^{-1}-\mathbf{1})\mathbf{D}_y(y^{-1}-1)|_{y=\mathrm{Id}} \\
=& \big((x^{-1}-\mathbf{1})(y^{-1}-\mathbf{1})+1\big)^{-2}(x^{-1}-\mathbf{1})y^{-2}|_{y=\mathrm{Id}} \\
=& \big((x^{-1}-\mathbf{1})((\tfrac{\mathbf{1}}{\mathbf{2}})^{-\mathbf{1}}-\mathbf{1})+\mathbf{1}\big)^{-2}(x^{-1}-\mathbf{1})(\tfrac{\mathbf{1}}{\mathbf{2}})^{-2} \\
=& x^2(x^{-1}-\mathbf{1})\mathbf{4} \\
=& \mathbf{4}(x-x^2)\,.
\end{aligned}
$$

Now, the ordinary differential equation

$$
\gamma_\eta(0) = \mathrm{Id}_{(0,1)^n}, \quad \dot{\gamma}_\eta(t) = \mathbf{4}(\gamma_\eta(t) - \gamma_\eta(t)^2)\eta, \quad \text{for all } t \in \mathbb{R}. \tag{5.5}
$$

has the solution

$$
\gamma_\eta(t) = \frac{e^{\mathbf{4}t\eta}}{\mathbf{1}+e^{\mathbf{4}t\eta}}\,. \tag{5.6}
$$

Thus, by definition, we obtain the exponential map $\mathrm{Exp}_{(0,1)^n} : \mathbb{R}^n \to (0,1)^n$ for $t = 1$ which is

$$
\mathrm{Exp}_{(0,1)^n}(\eta) = \gamma_\eta(1) = \frac{e^{\mathbf{4}\eta}}{\mathbf{1}+e^{\mathbf{4}\eta}}\,, \tag{5.7}
$$

where all operations apply component-wise to the vectors involved. □

The logarithmic map, denoted by $\mathrm{Log}_{(0,1)^n} : (0,1)^n \to \mathbb{R}^n$ which is defined as the inverse function of the exponential map can be calculated as

$$
\mathrm{Log}_{(0,1)^n}(x) := \mathrm{Exp}^{-1}_{(0,1)^n}(x) = \tfrac{1}{4}\log\Big(\frac{x}{\mathbf{1}-x}\Big)\,. \tag{5.8}
$$

Both, exponential and logarithmic map are depicted in Fig. 5.1
Using SE_3 for the camera motion, $\mathbb{R}^6$ for the acceleration of the camera and the Lie group given through definition 5.2.1 for the disparity map, we find the product Lie group $\mathcal{G}$ for our state space, i.e.

$$
\mathcal{G} := \mathrm{SE}_3 \times \mathbb{R}^6 \times (0,1)^{|\Omega|}\,. \tag{5.9}
$$

5.2.2 Propagation of the Camera Motion

For propagation of the camera we will use a second-order kinematic model that can be expressed as second-order differential equation as in [12], which is

$$
\begin{aligned}
\dot{E}(t) =& E(t)\,\mathrm{mat}_{\mathfrak{se}}(v(t)), \quad E(t_0) = E_0\,, \\
\dot{v}(t) =& \mathbf{0}, \quad v(t_0) = v_0\,,
\end{aligned} \tag{5.10}
$$

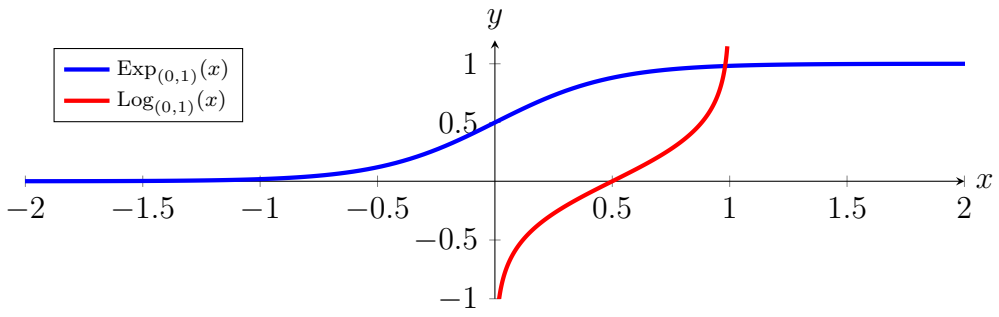


Figure 5.1: Exponential and logarithmic map of the Lie group $(0,1)^n$ for $n = 1$.

where $E = E(t) \in \mathrm{SE}_3$ and $v = v(t) \in \mathbb{R}^6$. The linear operator $\mathrm{mat}_{\mathfrak{se}}$ was defined in (4.4).

Remark 5.2.3. Since E describes the local camera motion from frame to frame, a first order model $\dot{E}(t) = \mathbf{0}$ corresponds to a constantly moving camera, i.e. with constant velocity. Thus, the model (5.10) describes a constant acceleration in the global camera frame.

5.2.3 Propagation of the Disparity Map

For the filter design we require differential equations for propagation of the states. The camera motion is propagated using the generalized kinematic model that can be written as dynamical system as in (4.54).
Propagation of the disparity map on an image grid, however, is non trivial. To this end we proposed two methods for propagation of the scene's depth map: 1. *A piecewise affine model* and 2. *A warping and interpolation technique.*

Discrete Propagation By Warping and Interpolation The propagation consists of mapping the image grid forward by an estimate of the motion $\hat{E} = (\hat{R}, \hat{w})$, by cubic interpolation of the depth on the irregular grid and back-projection of the resulting scene points. This leads to the following algorithm, which is also depicted in Figure 5.2.

1. Start with the current disparity map d_i on regular image grid in camera $(I, 0)$.

2. Warp the image grid forward into next image (camera estimate $\hat{E}(t)$) by using current disparity map d_i to get a grid with points

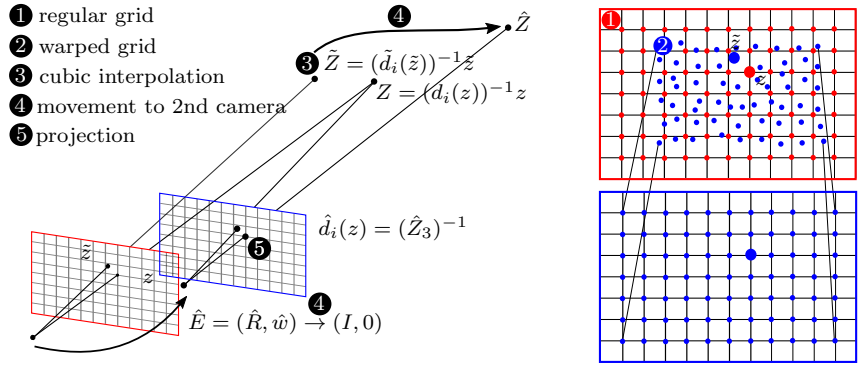


Figure 5.2: Discrete propagation of the disparity map

$\tilde{z} = \pi(\hat{R}\binom{z}{1}(d_i(z))^{-1} + \hat{w})$, where $\pi : \mathbb{R}^3 \to \mathbb{R}^2$ is given through $(z_1, z_2, z_3)^\top \mapsto (z_3)^{-1}(z_1, z_2)^\top$.

3. Perform cubic interpolation on the warped grid $\tilde{z}$ given the values $(z, d_i(z))$ which gives the new depth map $(\tilde{z}, \tilde{d}_i(\tilde{z}))$ in frame $(I, 0)$.

4. Move $\tilde{Z} = \binom{\tilde{z}}{1}(\tilde{d}_i(\tilde{z}))^{-1}$ to second camera to obtain $\hat{Z} = \hat{R}^\top\big(\tilde{Z} - \hat{w}\big)$.

5. Recognize the propagated disparity map as third component, i.e. $\hat{d}_i(z) = (\hat{Z}_3)^{-1}$.

Continuous Propagation Using a Affine Model This model assumes a piecewise affine scene structure and uses spatial gradients for propagating the scene's disparity map forward using a continuous framework. However, as we will see later, this method works poor since large spatial gradients (e.g. depth jumps at occlusions) violate the planar scene assumptions causing oscillations within numerical integration.
Let $d_i(\cdot, t) : \Omega \to (0, 1)^{|\Omega|}$ be the inverse depth map, where the subscript i denotes the inverse depth in the sequel and let $x = (E, v, d_i) \in \mathcal{G}$. Then for a (piecewise) planar scene the following differential equation for the inverse depth holds:

$$\dot{d}(z, t) = f_{d_i}(x(t), t)_z, \quad d_i(z, t_0) = d_i^0(z)\,. \tag{5.11}$$

The function on the right side is defined through

$$(f_{d_i}(x))_z := -\big(\mathrm{Log}(E)g_z\big)^\top F(z)\begin{pmatrix}\nabla_z d_i(z,t)\\ d_i(z,t)\end{pmatrix} \tag{5.12}$$

where

$$g_z := g_z(x) := (z_1, z_2, 1, d_i(z,t))^\top , \tag{5.13}$$

and the matrix $F(z)$ is defined through

$$F(z) := \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -z_1 & -z_2 & 1 \\ 0 & 0 & 0 \end{pmatrix}. \tag{5.14}$$

A detailed derivation of this formula is given in appendix A.5.1.

Comparison of Propagation Models For evaluation of the propagation models we use a synthetic scene and a given camera motion which is used for propagation. The piecewise affine model integrates the differential equation (5.12) to gain the propagated depth map, whereas the warping and interpolation technique only uses two discrete time points. In Figure 5.3 we see the results of the two methods. The discrete method is more exact and much cheaper, since the continuous needs 100 time steps for a good reconstruction. The assumption of a piecewise planar scene is in general not true. E.g. depth discontinuities result in large spatial gradients resulting in a worse reconstruction than the discrete propagation method.

5.2.4 Camera Motion and Disparity Map Induced Optical Flow

Since the state space $\mathcal{G}$ consists of the camera motion $E(t)$ and the disparity map $d_i(\cdot, t)$, we require observations that depend on both variables. It is well-known that from a given disparity map and a given camera motion the correspondences between a pair of consecutive images expressed as *optical flow* can be uniquely determined if the scene is static. To be precise, the dependency between the optical flow vector $u(z,t)$ at a position $z \in \Omega$ can be expressed with the following *non-linear* relation, where we denote by $R(t) \in \mathrm{SO}_3$ and $w(t) \in \mathbb{R}^3$ the rotational and translational component of the camera motion $E(t) = (R(t), w(t))$, respectively:

$$u(z,t) - z = \pi\Big(R(t)\,\tbinom{z}{1}\,(d_i(z,t))^{-1} + w(t)\Big). \tag{5.15}$$

This equation can also be written in terms of the matrix $E(t) \in \mathrm{SE}_3$ as follows:

$$u(z,t) - z = \pi\Big(E(t)\begin{pmatrix} \binom{z}{1}(d_i(z,t))^{-1} \\ 1 \end{pmatrix}\Big). \tag{5.16}$$

Remark 5.2.4. In comparison to chapter 4 we used a different modeling since we used a backward optical flow model that avoids inversion of the

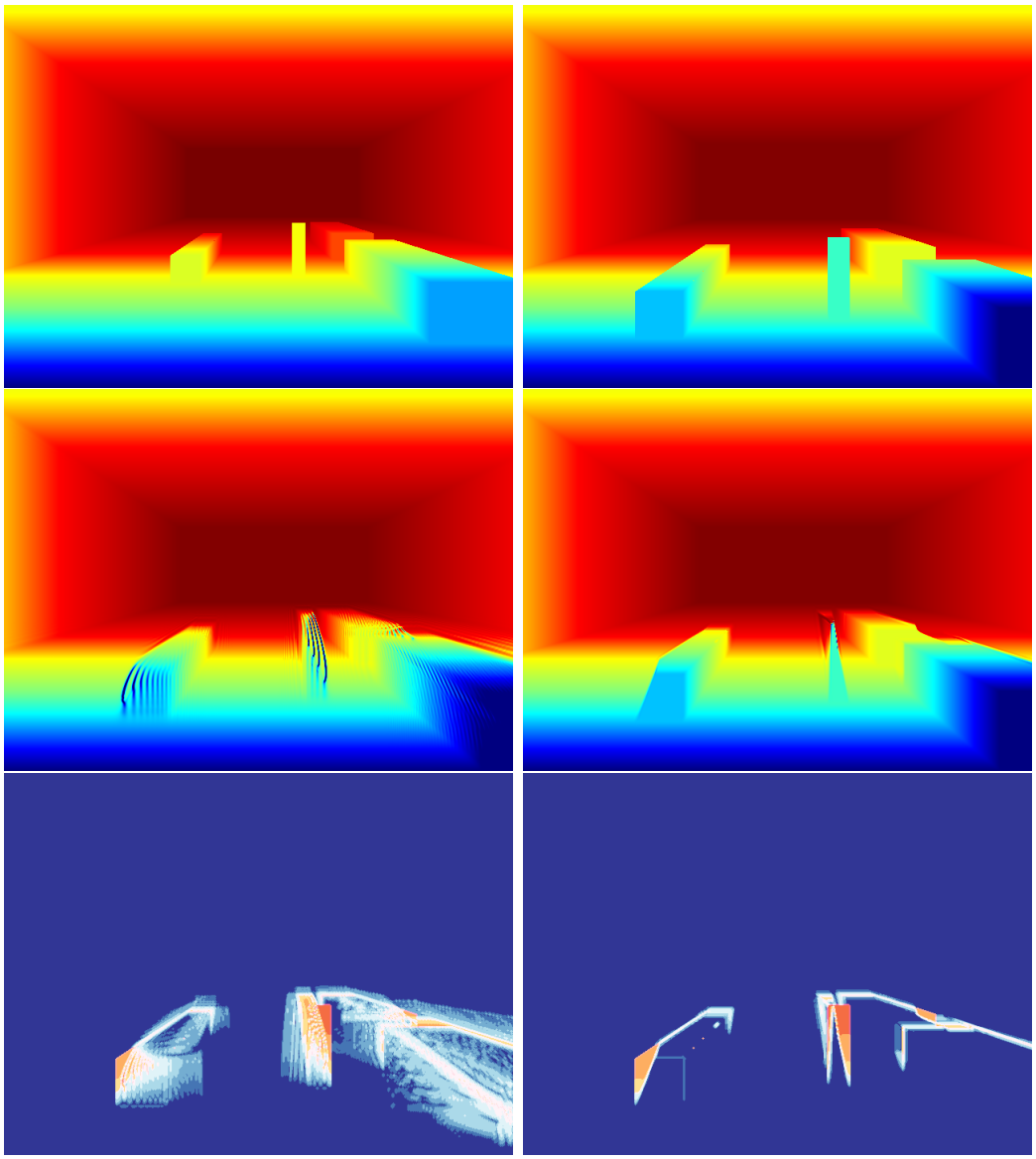


Figure 5.3: Comparison of the propagation methods. Top left: ground truth disparity map at time t_0 (initialization for the presented propagation methods), top right: ground truth disparity map at time t_{20}, mid left: continuous propagation method at time t_{20}, mid right: discrete-time propagation method at time t_{20}, bottom left: error map of continuous method, bottom right: error map of discrete method (blue: correct reconstruction, red: wrong reconstruction). The discrete time propagation methods leads to a better reconstruction and has a much smaller run time of factor 300. At positions with high disparity gradient, the assumptions of the continuous methods are violated which results in a worse reconstruction.

matrix $E \in \mathrm{SE}_3$, instead of the forward optical flow model. This modeling has two benefits:

1. The calculations of the gradient and Hessian become simpler,
2. the reconstruction will be better since in the forward optical flow model (divergent) usually cannot reconstruct the margins of the images correctly due to occlusions. This can be avoided using a backward flow model.

5.2.5 Overall Filtering Model

The function $f(x(t)) : \mathcal{G} \to \mathfrak{g}$ in (5.1) can now be defined as follows:

$$f(x(t)) := (f_E(x(t)), f_v(x(t)), f_{d_i}(x(t)))\,, \tag{5.17}$$

with component functions $f_E(x(t)) := \mathrm{mat}_{\mathfrak{se}}(v(t))$, and $f_v(x(t)) := \mathbf{0_6}$ as in (5.10) as well as $f_{d_i}(x(t)) := \mathbf{0}_{|\Omega|}$. Beside this continuous propagation step we also incorporate discrete updates of the disparities as described in section 5.2.3.
By setting $y_z(t) := u(z,t) - z$ and $h_z : \mathcal{G} \to \mathbb{R}^2$, as the right hand side of (5.15), we find the following observation equations by adding noise $\epsilon_z(t) \in \mathbb{R}^2$ for all $z \in \Omega$.

$$y_z(t) = h_z(x(t)) + \epsilon_z(t)\,, \qquad z \in \Omega\,. \tag{5.18}$$

5.2.6 Objective Function

Minimum energy filtering requires to define an energy function that penalizes the model and observation noise. In contrast to [73], that we will follow in this chapter, we will not use quadratic energy functions but an energy function that is a smooth approximation of the L^1-norm. The reason is that we want to reduce the influence of outliers in the observations that may cause numerical problems because the gradient grows linearly. This problem is depicted in Fig. 5.4. The norm of the gradient of the proposed L^1 penalty function is bounded. A smooth approximation to the non-differentiable L^1-norm is the generalized charbonnier penalty function that is smooth (C^∞) and has linear growth, such that we use it for ϕ, i.e. $\phi(x) := (x+\nu)^\beta - \nu^\beta$. With this notation and the shorthand $\|x\|_Q^2 := x^\top Q x$ the energy function reads

$$\mathcal{J}(\delta,\epsilon,x;t) := \tfrac{1}{2}\|x - x_0\|^2_{R_0^{-1}} + \int_{t_0}^{t} \Big(\tfrac{1}{2}\|\mathrm{vec}_{\mathfrak{g}}(\delta(\tau))\|^2_{R^{-1}} + \sum_{z\in\Omega} \phi(\tfrac{1}{2}\|\epsilon_z(\tau)\|^2_{Q_z}) \Big) d\tau, \tag{5.19}$$

where Q_z, R_0 and R are symmetric and positive definite matrices.

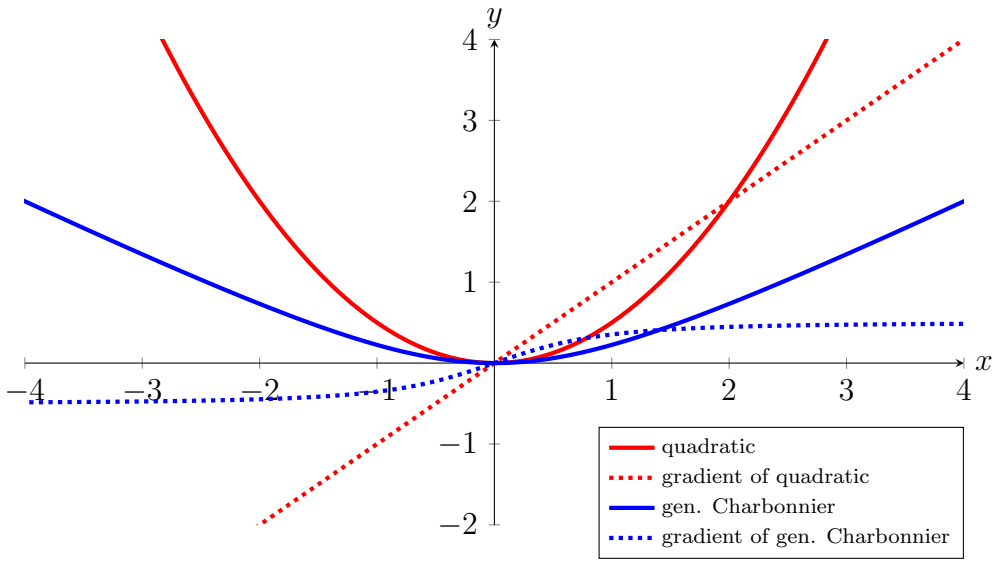


Figure 5.4: Comparison of a quadratic penalty function (red curve) and the generalized Charbonnier penalty function (5.31) (blue curve) with $\beta = \frac{1}{2}$ and $\nu = 1$ and their gradients. On can recognize that for outliers (e.g $x > 3$) the gradient of the quadratic function grows linearly whereas the gradient of the generalized Charbonnier function stays constant, what is beneficial for exact numerics.

5.2.7 Optimal Control Problem

After replacing the observation noise $\epsilon_z(t)$ by the residual $\epsilon_z(t) = \epsilon_z(x(t), t) := y_z(t) - h_z(x, t)$ in (5.30) we want to minimize the energy function $\mathcal{J}(\delta, x, x(t_0); t) = \mathcal{J}(\delta, \epsilon(x), x(t_0); t)$ regarding the model noise $\delta(t)$ with respect to the differential equation (5.1) yielding the *value function*

$$\mathcal{V}(x(t), t, x(t_0)) := \min_{\delta|_{[t_0,t]}} \mathcal{J}(\delta, x; t) \quad \text{subject to (5.1).} \tag{5.20}$$

Calculation of the value function requires to introduce the time-varying (left-trivialized) Hamiltonian function $\tilde{\mathcal{H}} : \mathcal{G} \times \mathfrak{g}^* \times \mathfrak{g} \times \mathbb{R} \to \mathbb{R}$ that is given through

$$\begin{aligned} \tilde{\mathcal{H}}(x, \mu, \delta, t) :=& \big(\tfrac{1}{2}\|\mathrm{vec}_{\mathfrak{g}}(\delta(t))\|^2_{R^{-1}} + \sum_{z\in\Omega} \phi(\tfrac{1}{2}\|y_z(t) - h_z(x(t)))\|^2_{Q_z}\big) \\ & - \langle \mu, f(x(t)) + \delta(t)\rangle_{\mathrm{Id}}\,. \end{aligned} \tag{5.21}$$

Owing to the *Pontryagin minimum principle* [66] we find the minimizing argument of the value function (5.20) by minimizing the Hamiltonian $\tilde{\mathcal{H}}$ with respect to δ. Since this Hamiltonian is convex with respect to δ the minimum is unique and can be calculated as follows:

$$\begin{aligned} \mathbf{0} \overset{!}{=}& \mathbf{D}_{\delta^*} \tilde{\mathcal{H}}^-(x, \mu, \delta^*, t) \\ =& \mathbf{D}_{\delta^*} \big(\tfrac{1}{2}\|\mathrm{vec}_{\mathfrak{g}}(\delta(t)^*)\|^2_R - \langle \mu, \delta(t)^*\rangle\big) \\ =& R\,\mathrm{vec}_{\mathfrak{g}}(\delta^*) - \mathrm{vec}_{\mathfrak{g}}(\mu) \\ \Leftrightarrow \qquad \delta^* =& \mathrm{mat}_{\mathfrak{g}}\left(R^{-1}\,\mathrm{vec}_{\mathfrak{g}}(\mu)\right). \end{aligned}$$

The optimal Hamiltonian $\mathcal{H} : \mathcal{G} \times \mathfrak{g}^* \times \mathbb{R} \to \mathbb{R}$ defined through $\mathcal{H}(x, \mu, t) := \min_\delta \tilde{\mathcal{H}}^-(x, \mu, \delta, t)$ is found by insertion of δ^*. This results in $\mathcal{H}(x, \mu, t) := \tilde{\mathcal{H}}(x, \mu, \delta^*, t)$ such that

$$\mathcal{H}(x, \mu, t) = -\langle \mu, f(x(t)\rangle_{\mathrm{Id}} - \tfrac{1}{2}\|\mathrm{vec}_{\mathfrak{g}}(\mu)\|^2_R + \sum_{z\in\Omega} \big(\phi\big(\tfrac{1}{2}\|y_z(t) - h_z(x(t))\|^2_{Q_z}\big)\big)\,. \tag{5.22}$$

In the case of a linear-quadratic control problem this optimal Hamiltonian satisfies the (left-trivialized) Hamilton-Jacobi-Bellman equation (cf. (3.40)), i.e.

$$\frac{\partial}{\partial t}\mathcal{V}(x, t) - \mathcal{H}(x, x^{-1}\mathbf{D}_1\mathcal{V}(x, t), t) = 0\,. \tag{5.23}$$

Here, $\mathbf{D}_1\mathcal{V}(x, t) \in T^*_x\mathcal{G}$ is an element of the cotangent space.

Remark 5.2.5. Note that our control problem has neither *linear control dynamics* nor a *quadratic* energy function. Thus, we have no guarantee that the HJB equation is a necessary *and sufficient* condition for optimality. Instead we require a good initialization to gain an optimal reconstruction. However, we will show that in parctice a fairly general initialization will lead to good reconstructions.

5.2.8 Recursive Filtering Principle and Truncation

Computation of the total time derivative of the necessary condition

$$\mathbf{D}_1\mathcal{V}(x,t,x(t_0)) = \mathbf{0}\,, \tag{5.24}$$

and insertion of the HJB equation (5.23) leads to the following Lemma that gives a recursive description of the optimal state $x^* = x^*(t)$ (cf. [73, Eq. (37)]).

Lemma 5.2.6. *The evolution equation of the optimal x^* state is given through*

$$\dot{x}^*(t) = x(t)\Big(f(x^*(t)) - \hat{Z}(x^*(t),t)^{-1} \circ x^{-1}(\mathbf{D}_1\mathcal{H}(x^*(t),\mathbf{0},t))\Big)\,, \tag{5.25}$$

where $\hat{Z} : \mathfrak{g} \to \mathfrak{g}^*$ is the left-trivialized Hessian of the value function given through

$$\hat{Z}(x^*,t)\circ\eta = (x^*)^{-1}\operatorname{Hess}\mathcal{V}(x^*(t),t,x(t_0))[x^*\eta]\,, \quad \eta\in\mathfrak{g}\,. \tag{5.26}$$

Proof. See appendix A.5.3. □

Because the non-linear filtering problem is infinite dimensional we will replace the exact operator $\hat{Z}$ by an approximation $Z : \mathfrak{g} \to \mathfrak{g}^*$ which can be obtained by truncation of the full evolution equation of Z. But still the operator $Z(x^*,t)$ on $\mathfrak{g}$ is complicated such that we introduce a matrix representation $P(t)$ that is defined through the relation $\operatorname{vec}_{\mathfrak{g}}(Z(x^*,t)^{-1}\circ\eta) =: P(t)\operatorname{vec}_{\mathfrak{g}}(\eta)$.

Lemma 5.2.7. *The matrix representation of the approximation of the operator $\hat{Z}$ evolves regarding the following matrix Riccati equation*

$$\dot{P}(t) = R + C(x^*,t)P(t) + P(t)C(x^*,t)^\top - P(t)H(x^*,t)P(t), \tag{5.27}$$

where the matrix R is the weighting matrix in the energy function (5.30) *and the matrices C and H are given for $\eta\in\mathfrak{g}$ through*

$$\begin{aligned} C(x^*,t)P(t)\operatorname{vec}_{\mathfrak{g}}(\eta) &:= \operatorname{vec}_{\mathfrak{g}}\big((x^*)^{-1}\mathbf{D}_2(\mathbf{D}_1\mathcal{H}(x^*,\mathbf{0},t))[Z(x^*,t)\circ\eta]\big)\\ &\quad + \operatorname{vec}_{\mathfrak{g}}(\omega^*_{\overleftrightarrow{\mathbf{D}_2\mathcal{H}(x^*,\mathbf{0},t)}}\circ Z(x^*,t)\circ\eta) + \operatorname{vec}_{\mathfrak{g}}(\omega^*_{(x^*)^{-1}\dot{x^*}}\circ Z(x^*,t)\circ\eta)\,,\\ H(x^*,t)\operatorname{vec}_{\mathfrak{g}}(\eta) &:= \operatorname{vec}_{\mathfrak{g}}((x^*)^{-1}\operatorname{Hess}_1\mathcal{H}(x^*,\mathbf{0},t)[x\eta])\,. \end{aligned}$$

Here, $x\omega_\xi\eta := \nabla_{x\xi}x\eta$ denotes the connection function on the Lie algebra $\mathfrak{g}$ of the Levi-Civita connection $\nabla_\cdot\cdot$ for $\xi, \eta \in \mathfrak{g}$ and $x \in \mathcal{G}$, and $\omega_{\overleftarrow{\xi}}^*$ is the dual of the "swaped" connection function $\omega_{\overleftarrow{\xi}}\eta := \omega_\eta\xi$ (cf. [72]).

Proof. See appendix A.5.3. □

By insertion of the expression P into (A.87) and by evaluation of the expressions in Lemma 5.2.6 and 5.2.7 we obtain the final minimum energy filter that consists of continuous propagation of the states with a discrete update of the disparity map.

Theorem 5.2.8. *The second-order minimum energy filter with additional discrete propagation step for the disparity map is given through the following evolution equations of the optimal state $x^* \in \mathcal{G}$ as well as the second-order operator $P \in \mathbb{R}^{(12+|\Omega|)\times(12+|\Omega|)}$.*

$$\dot{x}^*(t) = x^*(t)\big(f(x^*(t)) - \mathrm{mat}_{\mathfrak{g}}(P(t)\,\mathrm{vec}_{\mathfrak{g}}(G(x^*(t),t)))\big), \tag{5.28}$$

$$\dot{P}(t) = R + C(x^*,t)P(t) + P(t)C(x^*,t)^\top - P(t)H(x^*,t)P(t), \tag{5.29}$$

with initial conditions $x^*(t_0) = x_0$ and $P(t_0) = R_0$, where R_0 is the matrix in (5.30). $G(x^*,t) = (G_E(x^*), \mathbf{0}, G_{d_i}(x^*)) \in \mathfrak{g}$ denotes the Riemannian gradient of the Hamiltonian in (A.87) with components G_E and G_{d_i}.

Proof. See appendix A.5.3. □

The numerical integration of these equations between the time steps t_{k-1} and t_k correspond to the update step of a filter, where the updates are assumed to be piecewise constant. After each update step the disparity map is propagated forward using the procedure in Fig. 5.2 that result in the final filter.

5.2.9 Numerical Integration

In contrast to the low-dimensional filtering presented in chapter 4 where we used implicit numerical integration that we presented in section 3.4, these methods cannot be used for high-dimensional problems. The main problem is that the operator H in Theorem 5.2.8 is symmetric and strictly positive definite. However, implicit methods for solving the algebraic Riccati equation [10] require a low-rank approximation of H. In the considered case this cannot be done without loosing essential information. Therefore we decided to use explicit numerical integration schemes, such as Crouch-Crossman methods that we presented in section 3.4.1. During numerical integration it is important to keep the matrix P sparse, therefore we set the off-diagonal entries of the lower right part of P (that addresses the disparities) after each iteration to zero.

5.3 Adding of Spatial Regularization

Since the approach that we presented within the last section does not consider spatial regularization which causes incorrect results in regions close to the epipole, we propose to add a spatial regularizer into the optimization problem. Thus, the resulting minimum energy filter will consider both, *temporal and spatial* regularization. Although the resulting optimal filter can again be derived in a mathematical correct way, the numerics becomes more involved as in the case without spatial regularizer: Temporal and spatial regularization will lead to a second operator (inverse of the Hessian of the value function) which becomes dense relative quickly. Since our problem consists of a high dimensional state space $N > 500.000$, an exact numerical treatment becomes infeasible. Instead, we need to find suitable approximations that enable a sparse representation of the Hessian. We discuss this special issue after derivation of the corresponding minimum energy filter.
For this derivation we leave the original filtering problem in (5.1) and (5.2) unchanged but add an additional spatial regularization term into the energy function which now reads

$$\begin{aligned}\mathcal{J}(\delta,x,x(t_0);t) := m_0(x(t_0)) + \Big(&\tfrac{1}{2}\|\mathrm{vec}_{\mathfrak{g}}(\delta(\tau))\|_Q^2 \\ &+ \sum_{z\in\Omega} \phi_{\text{data}}(\tfrac{1}{2}\|y_z(\tau) - h_z(x(\tau))\|_{S_z}^2) + \lambda\, \phi_{\text{reg}}\big(\tfrac{1}{2}\|\nabla_z d_i(z,\tau)\|^2\big)\Big)\, d\tau\,,\end{aligned} \tag{5.30}$$

where $\lambda > 0$ is a positive constant and ϕ is a two times continuously differentiable penalty function. As in the previous section ϕ_{data} and ϕ_{reg} denotes the generalized Charbonnier penalty function which is given through

$$\phi(\cdot) := (\cdot + \nu)^\beta - \nu^\beta\,. \tag{5.31}$$

For a small $\nu > 0$ and the choice $\beta = 1/2$ we gain an approximation of L^1. Below we will denote the parameters of the corresponding data term by ν_d and β_d and for the regularizer by ν_r and β_r.

5.3.1 Optimal Control Problem

In the case of a spatial regularizer the corresponding minimization of the energy function (5.30) regarding the control variable $\delta(t)$ subject to the differential equation (5.1) stays almost unchanged and reads

$$\mathcal{V}(x(t),t,x(t_0)) := \min_{\delta|_{[t_0,t]}} \mathcal{J}(\delta,x;t) \quad \text{subject to (5.1).} \tag{5.32}$$

The model noise process $\delta(t)$ is considered as control parameter of the corresponding minimum energy control problem that is solved backwards in time (cf. [72]). Since the energy function (5.30) is still convex in δ the optimal value δ^* can again be found by minimizing the pre-Hamiltonian (5.21) which needs to be modified by adding an extra term for the spatial gradient. As the spatial gradient does not depend on δ the minimizing parameter of the optimization problem (5.32) is again $\delta^* = \mathrm{mat}_{\mathfrak{g}}(R^{-1}\,\mathrm{vec}_{\mathfrak{g}}(\mu))$. The resulting Hamiltonian is given through

$$\begin{aligned}\mathcal{H}(x,\mu,t) := &-\langle \mu, f(x(t)\rangle - \tfrac{1}{2}\|\mathrm{vec}_{\mathfrak{g}}(\mu)\|^2_{R^{-1}} \\ &+ \Big(\sum_{z\in\Omega}\big(\phi_{\mathrm{data}}\big(\tfrac{1}{2}\|y_z(t) - h_z(x(t))\|^2_{Q_z}\big) + \lambda\phi_{\mathrm{reg}}\big(\tfrac{1}{2}\|\nabla_z d_i(z,t)\|^2\big)\big)\Big),\end{aligned} \tag{5.33}$$

where $x = x(t)$ consists of the components $E(t) \in \mathrm{SE}_3$, $v(t) \in \mathbb{R}^6$ and $(d_i(z,t))_{z\in\Omega} = d_i(t) \in (0,1)^{|\Omega|}$.

5.3.2 Recursive Filtering Principle and Truncation

Similarly to section 5.2.8 we compute the total time derivative of the necessary condition and use the Hamilton-Jacobi-Bellman equation to find the optimal filtering equations. For brevity we omit the corresponding calculations which require a discretization of the spatial gradient of the disparities $\nabla_z d_i(z,t)$ in the energy function (5.30). Finally, we obtain the minimum energy filter which is given in Theorem 5.2.8. The only difference is that the Riemannian gradient and the Riemannian Hessian need to be calculated regarding the Hamiltonian in (5.33) instead of the Hamiltonian in (5.22). Since only few additional terms appear we refer the reader for the corresponding expressions in the appendix.

5.4 Experiments

5.4.1 Preprocessing

As stated above, our method requires precise optical flow as input. Since we propose a monocular method we also demand that the optical flow is computed from two consecutive image frames without stereo information. For this reason we used the well-known *EpicFlow* approach [69]. The matches are computed with *Deep Matching* [83]; the required edges are from [28].

5.4.2 Choice of the Weighting Matrices

Monocular methods suffer from the fact that observations that appear close to the epipole (focus of expansion) are orthogonal to the camera motion such that these regions cannot be reconstructed correctly. Therefore we use the weighting term from [8, Eq. (14)] for the weighting matrix Q that decreases the influence of the data term in regions close to the epipole.

5.4.3 Outlier Detection

To remove outliers, we compute the backward flow from frame i to $i+1$ as well as the forward flow from frame $i+1$ to i. In regions where these flows are not consistent with each other, we decrease the weight of the term R such that the filter has less ability to fit to the data and the discrete disparity map propagation from section 5.2.3 reduces the error.

5.4.4 Scale Correction

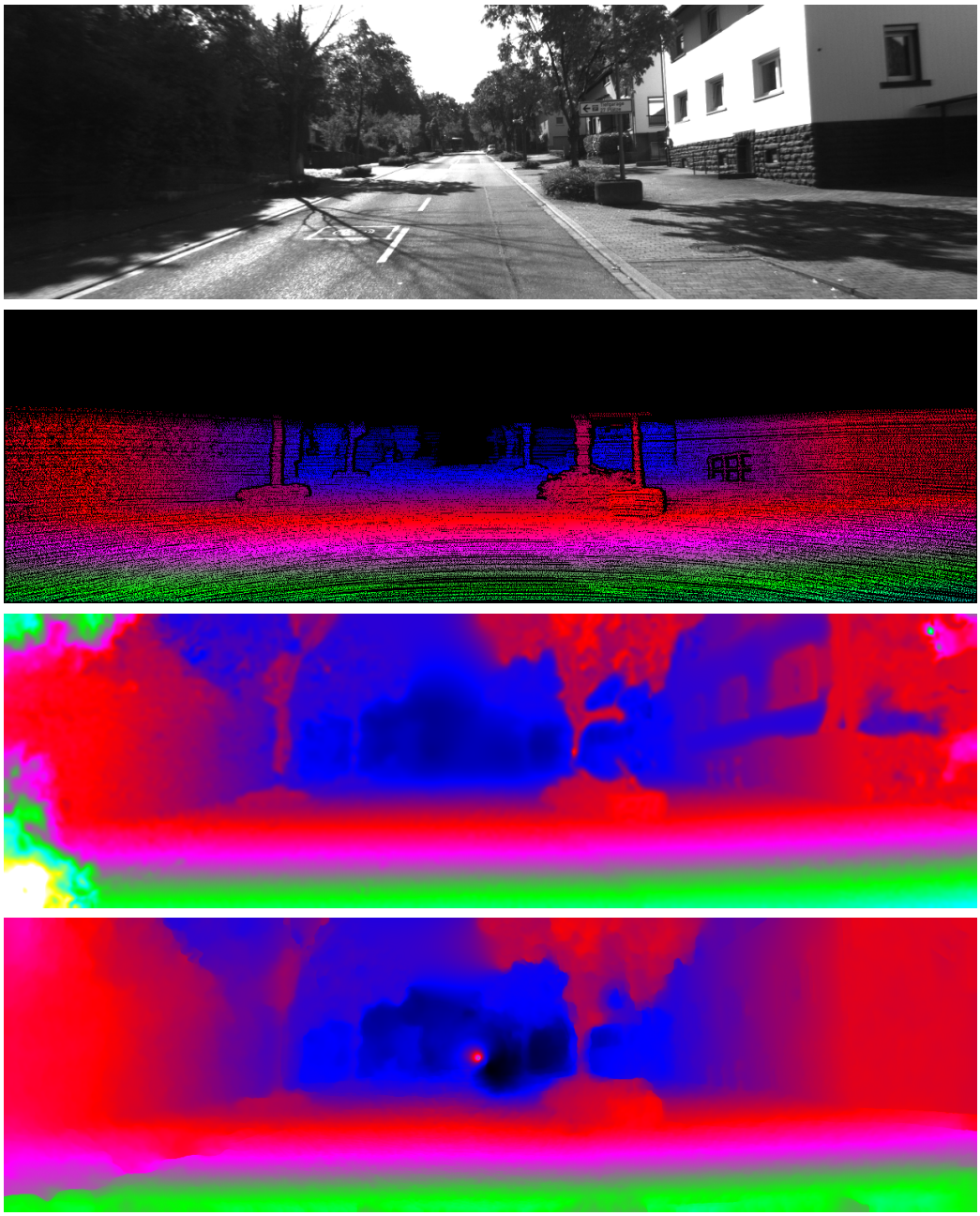
As monocular approaches cannot estimate the scale of a scene without prior knowledge about invariants in the scene, we corrected the scale of the scene by calculating of the pixel-wise quotient of the disparities and taking its median as scale $s := \text{median}\{d_i^{\text{gt}}(z,t)/d_i^{\text{est}}(z,t) | z \in \Omega^*\}$, where Ω^* denotes the image domain without points which are close to the epipole (< 50 pixel distance).

5.4.5 Qualitative Results without Regularization

In Fig. 5.5 we compare the reconstruction of the disparity map for different scenes (a) – (k) of our method with the results from [8] and the ground truth without using a spatial regularization (i.e. $\lambda = 0$ in (5.30)). One can observe in Fig. 5.5 that our method preserves small details and depth discontinuities better than [8] and returns sharper edges. However, regions close to the epipole cannot be reconstructed correctly, because of the monocular approach with induces a less beneficial motion parallax.

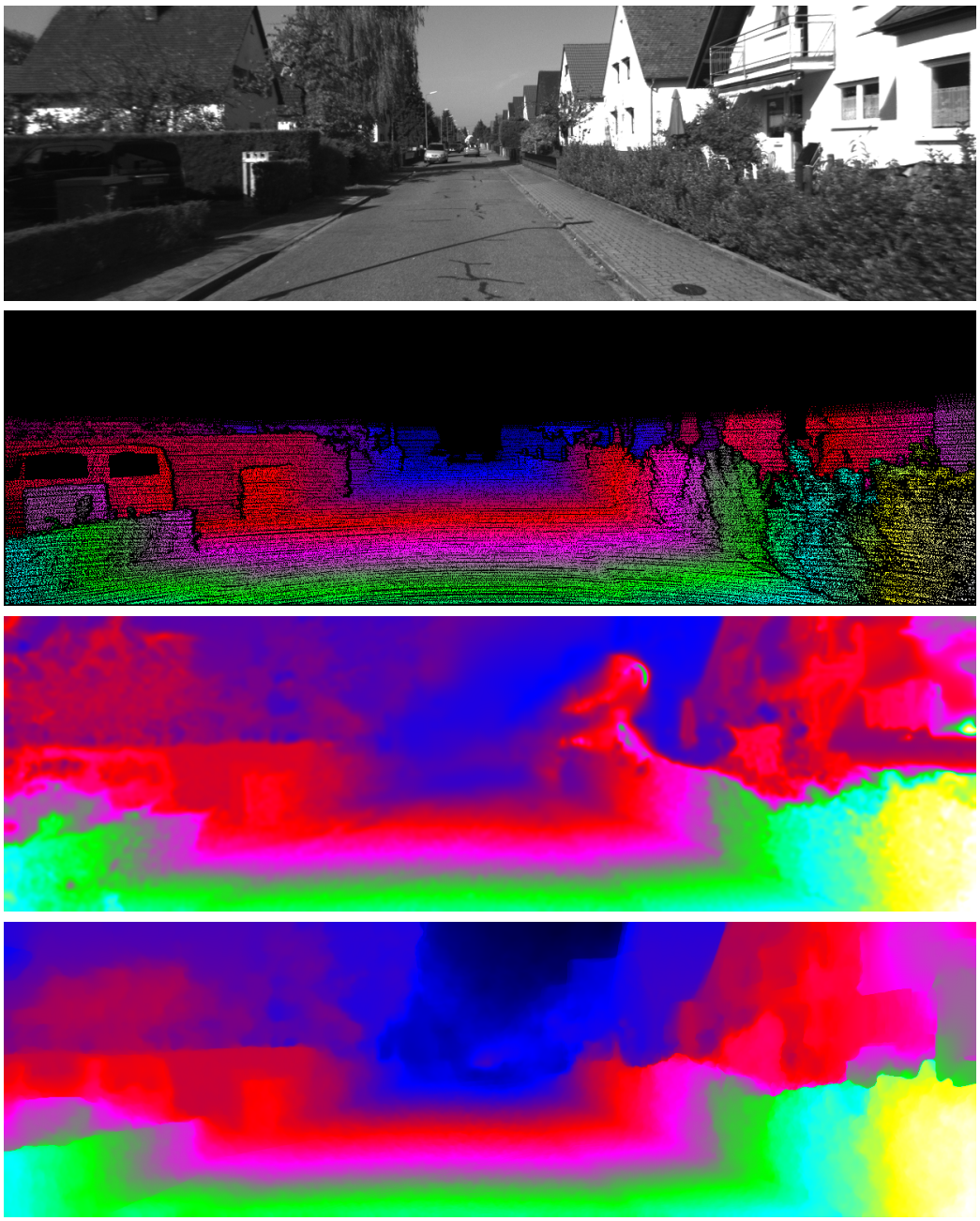
5.4.6 Quantitative Results without Regularizer

We evaluated the mean amount of pixels in Ω^* with a disparity error larger than three pixel for both occluded and not occluded scenarios in Table 5.1. We are slightly inferior towards Becker et al. [8]; however, unlike [8] we do



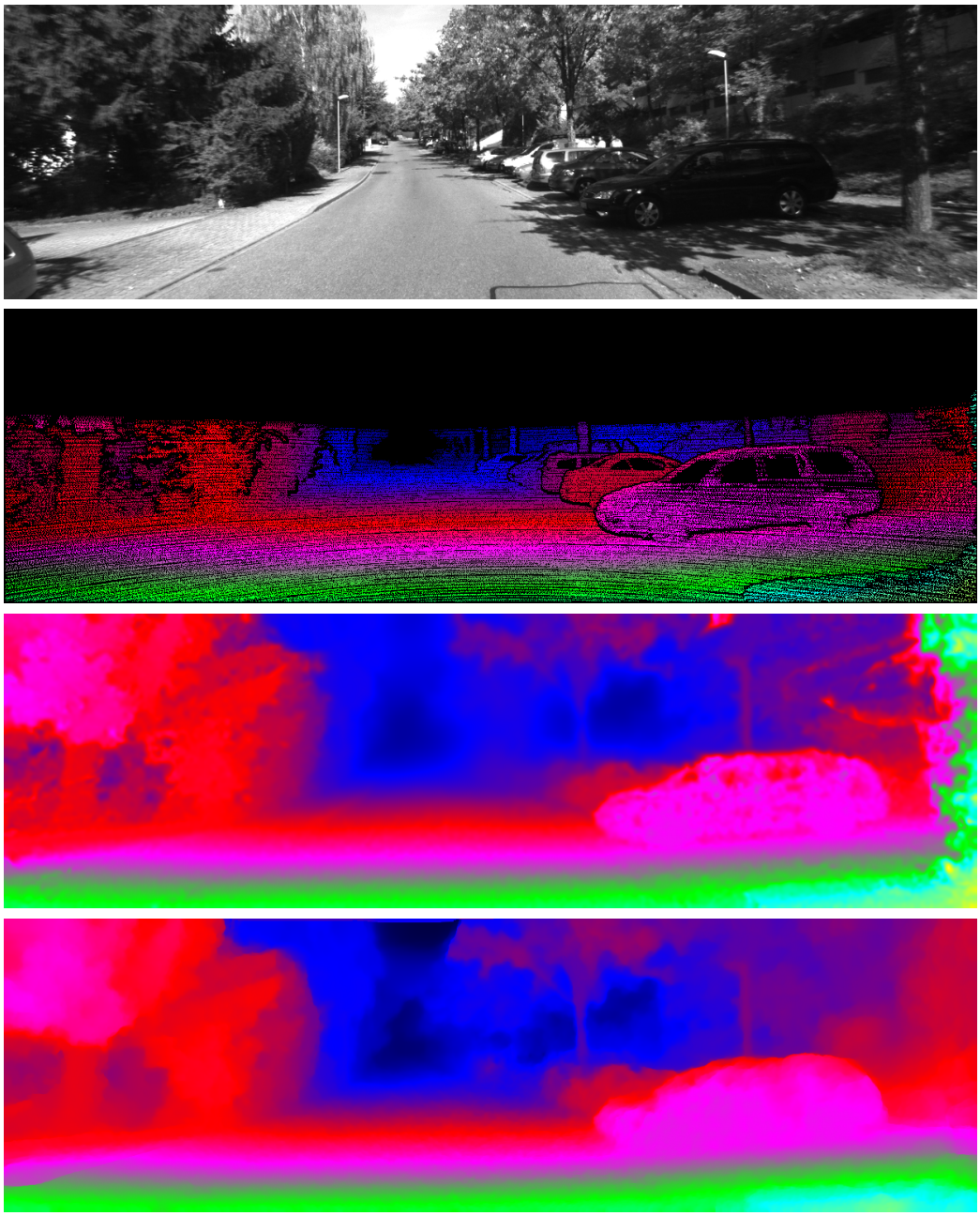
(a)

Figure 5.5: Top: gray value image, second row: ground truth disparity map, third row: Becker et al. [8], fourth row: our method.



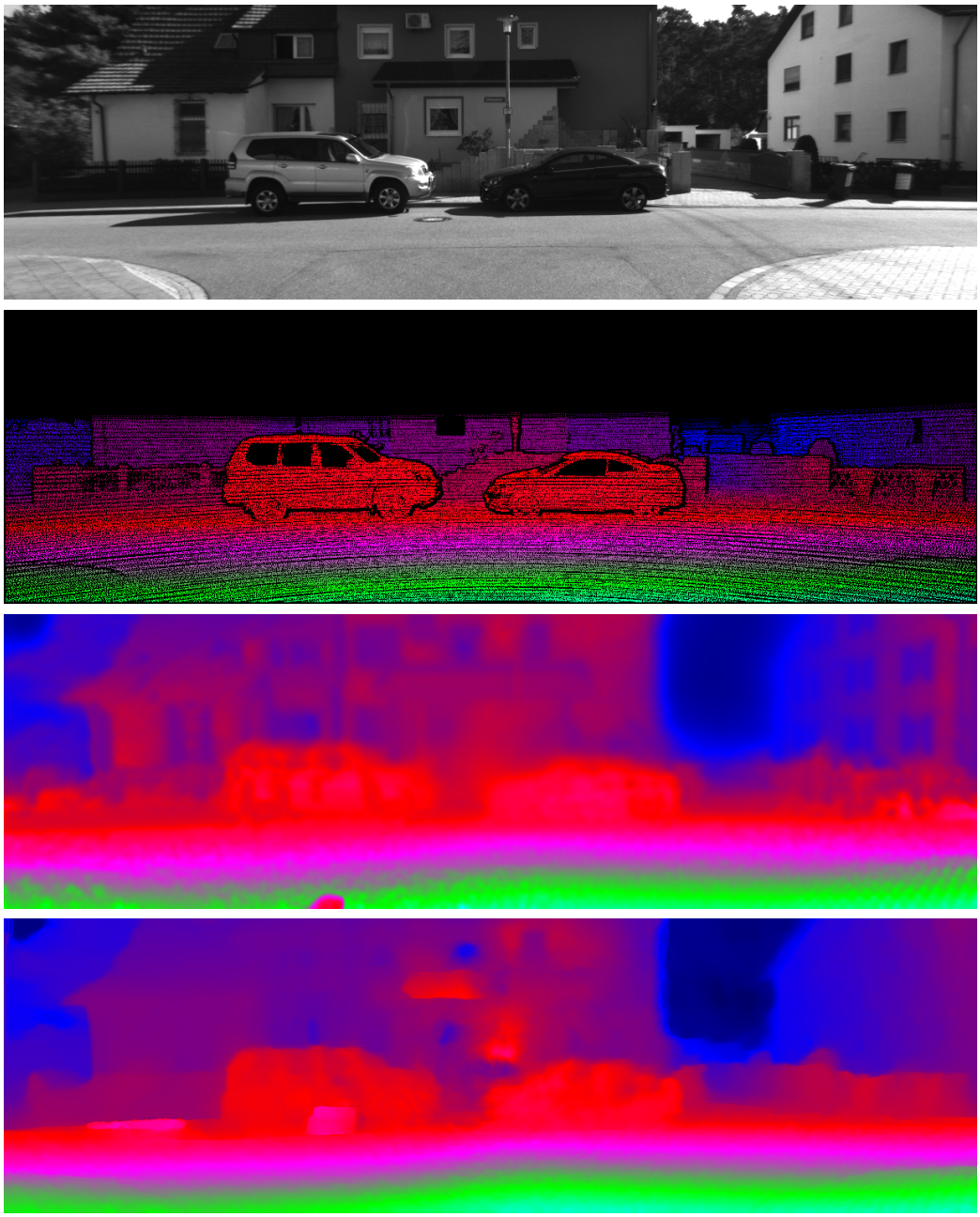
(b)

Figure 5.5: Top: gray value image, second row: ground truth disparity map, third row: Becker et al. [8], fourth row: our method.



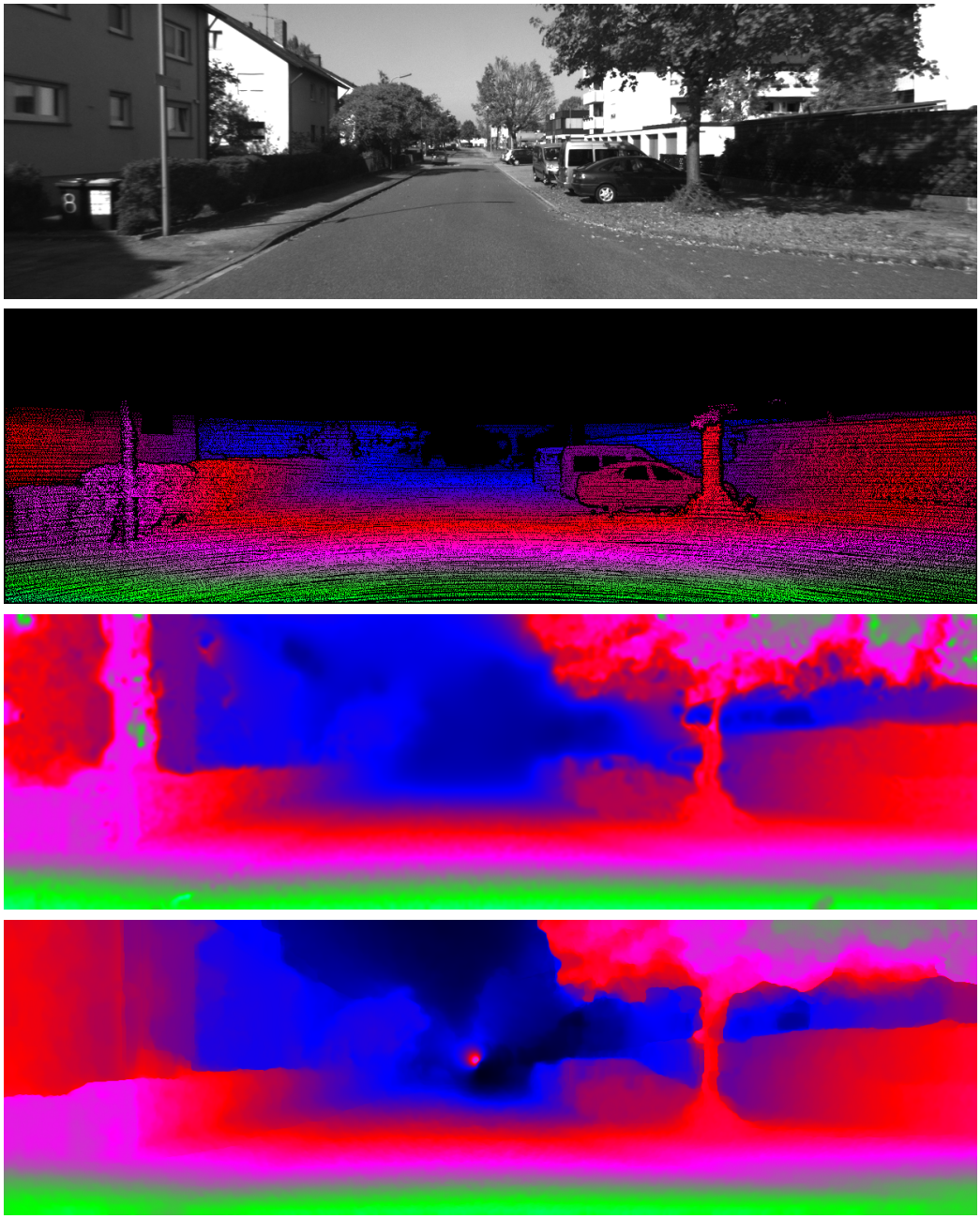
(c)

Figure 5.5: First row: gray value image, second row: ground truth disparity map, third row: Becker et al. [8], fourth row: our method.



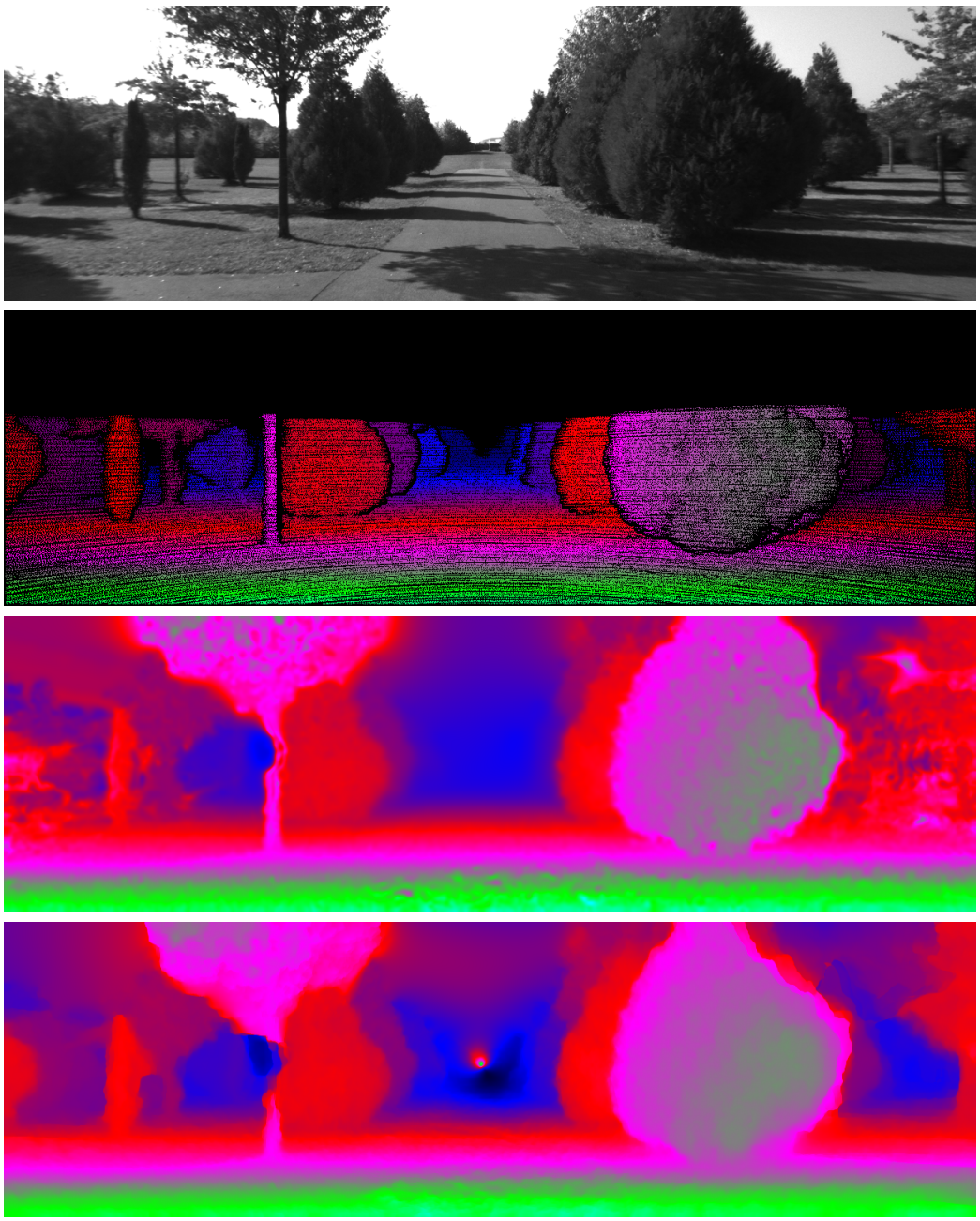
(d)

Figure 5.5: First row: gray value image, second row: ground truth disparity map, third row: Becker et al. [8], fourth row: our method.



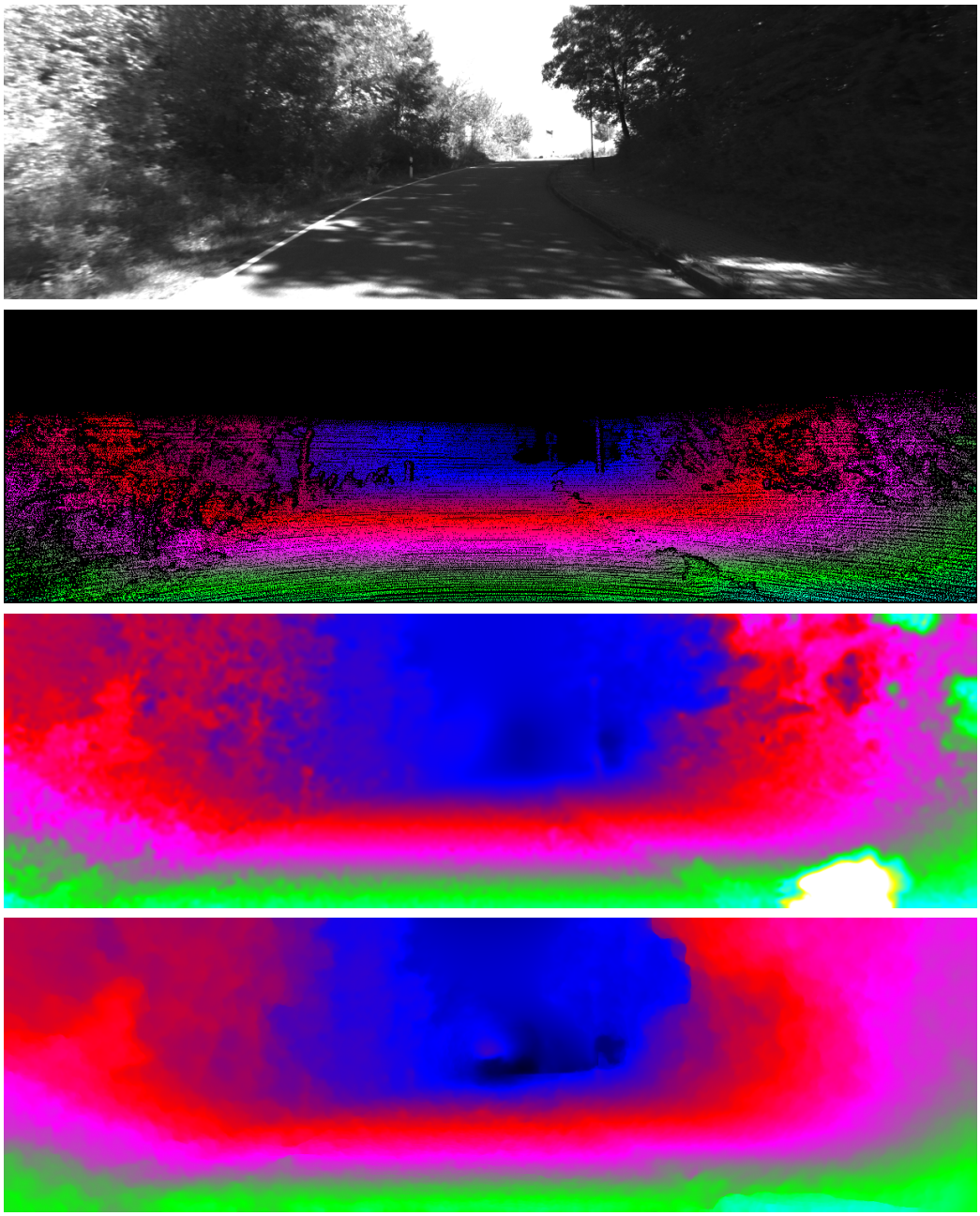
(e)

Figure 5.5: First row: gray value image, second row: ground truth disparity map, third row: Becker et al. [8], fourth row: our method.



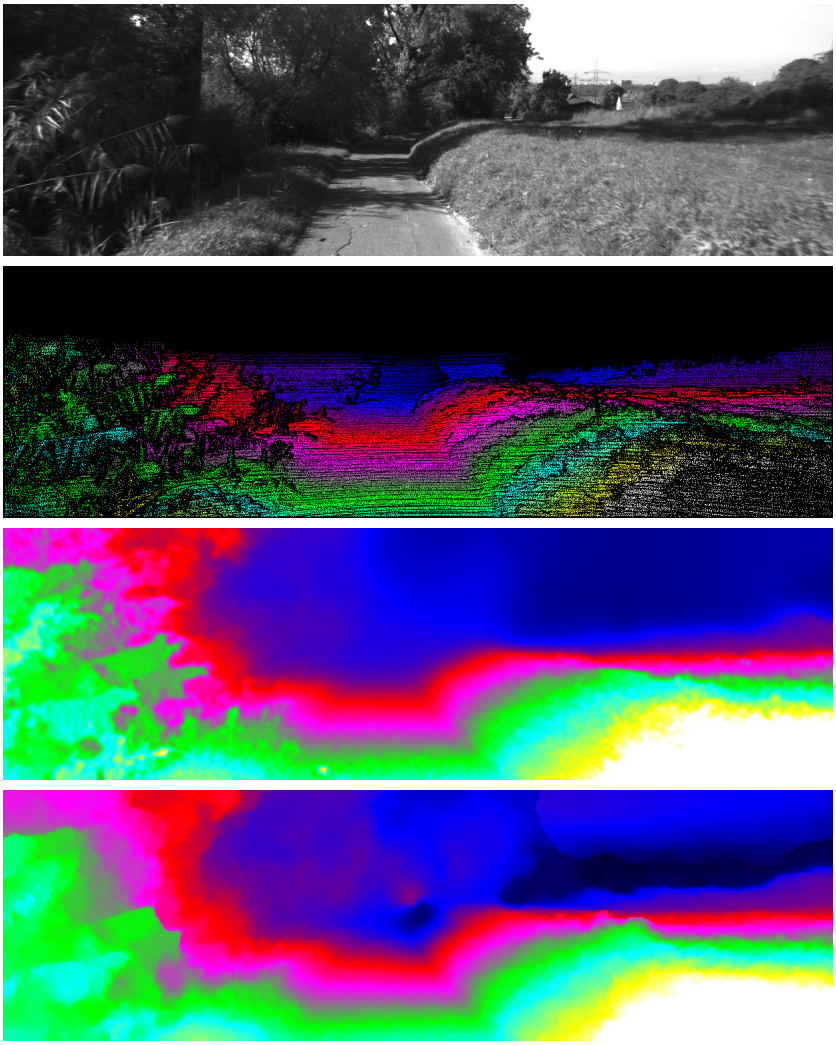
(f)

Figure 5.5: First row: gray value image, second row: ground truth disparity map, third row: Becker et al. [8], fourth row: our method.



(g)

Figure 5.5: First row: gray value image, second row: ground truth disparity map, third row: Becker et al. [8], fourth row: our method.



(h)

Figure 5.5: Reconstruction of selected disparity maps (a) – (h) of the KITTI stereo benchmark [36] with the corresponding color encoding (green is near, blue/black is far): first row: gray value image of scene, second row: ground truth from the KITTI stereo benchmark, third row: monocular method of Becker et al. [8], fourth row: reconstruction with our monocular method. Although in the quantitative evaluation both methods perform equally, one can recognize that our method results in sharper corners. Due to spatial regularization [8] reconstructs regions close to the epipole better.

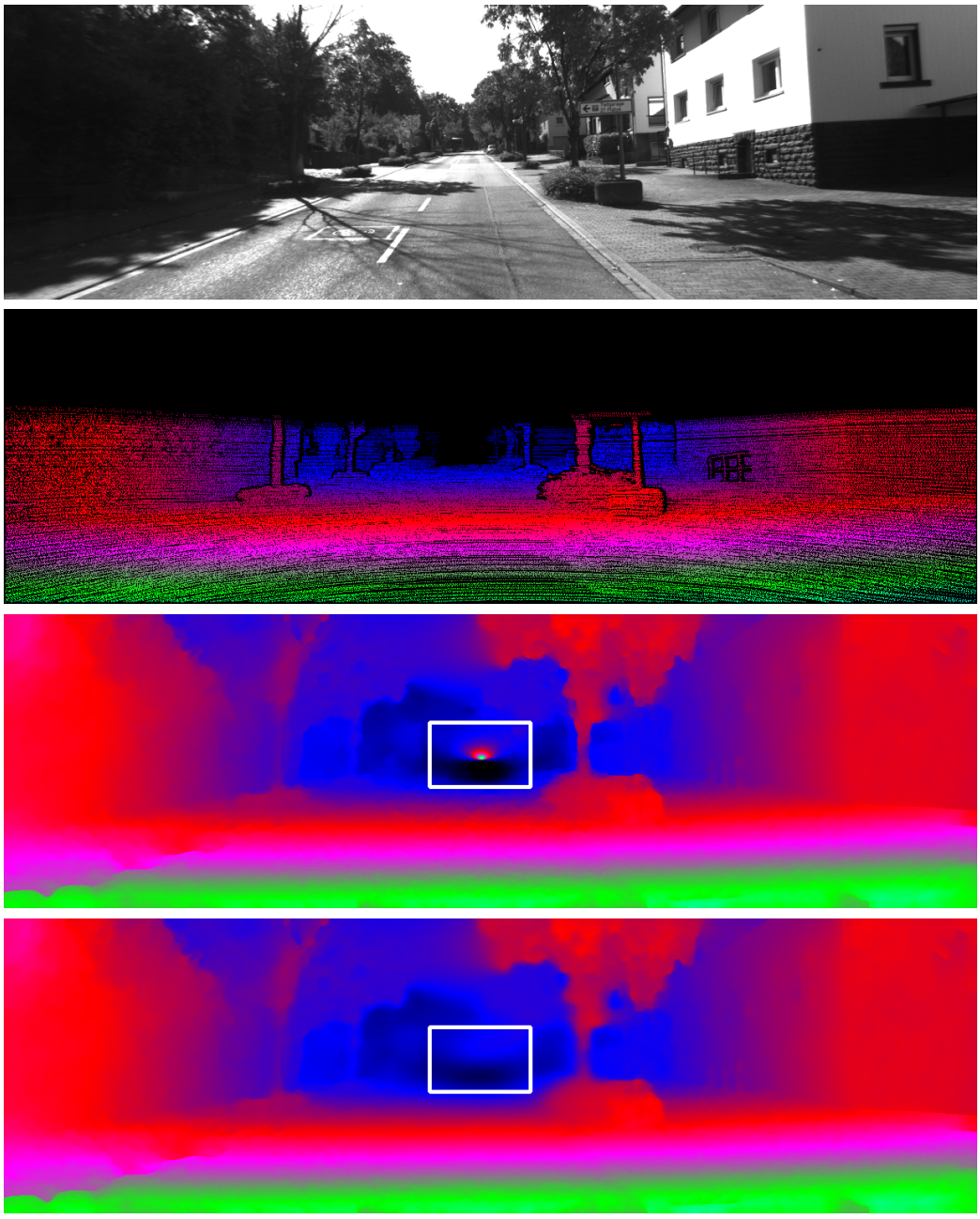
not have spatial regularization within our optimization which explains the differences.

Table 5.1: Evaluation of the mean disparity errors of the method of Becker et al. [8] and our approach. We calculated the disparity maps on all training sequences and evaluated the amount of pixels with a disparity error larger than three ($p_{3px}[\%]$) or five $p_{5px}[\%]$) pixels. We also evaluated to occluded (occ) and non-occluded (noc) ground truth data. Our approach is inferior in the category of three pixel error but slightly superior in the relation of five pixel errors meaning that our approach has less large errors than [8].

	$p_{3px}[\%]$ (occ)	$p_{5px}[\%]$ (occ)	$p_{3px}[\%]$ (noc)	$p_{5px}[\%]$ (noc)
Becker et al. [8]	**17.74**	10.82	**17.63**	10.72
our approach	19.24	**10.69**	19.14	**10.59**

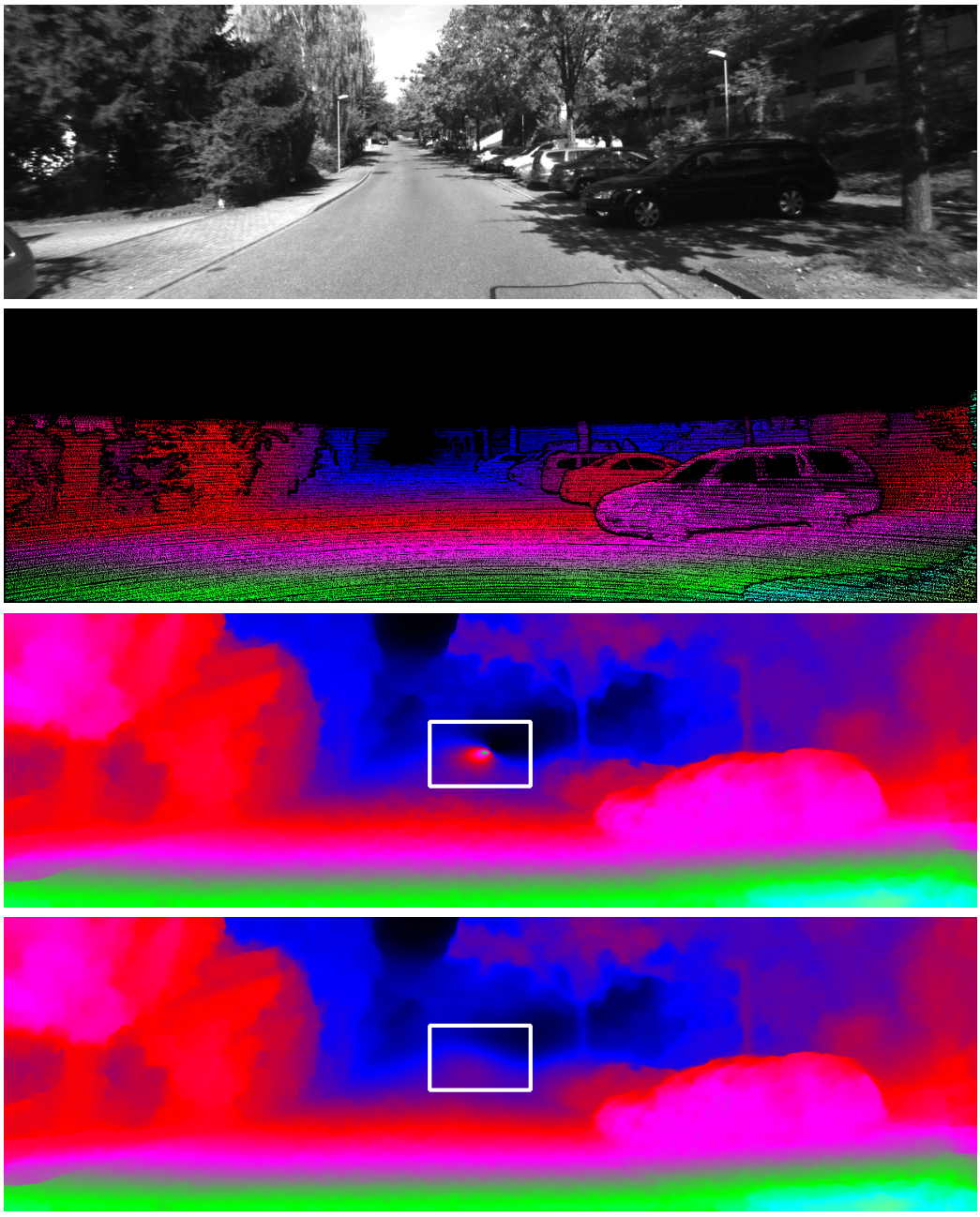
5.4.7 Qualitative Results with Spatial Regularizer

State-of-the-art optical flow methods usually include spatial regularization, such that the optical flow field that we need for our method is good enough for a good reconstruction. However, outliers in the optical flow that appear within low textured regions or because of repetitive patterns cause large errors within the depth map reconstruction. Often the influence of erroneous optical flow even results in an non-perfect reconstruction of the camera motion. As mentioned above, the minimum energy filtering approach has limitations in reconstructing regions to the epipole correctly (cf. Fig. 5.5), because the induced energy at these regions is close to zero. Therefore, we present a comparison between the minimum energy filter with regularization (presented in section 5.3) and without in Fig. 5.6. Although it is not possible to estimate the disparity map at the epipole within a monocular approach (without using additional information), one can observe that the reconstruction error in this region is reduced compared to the minimum energy filter without spatial regularization.



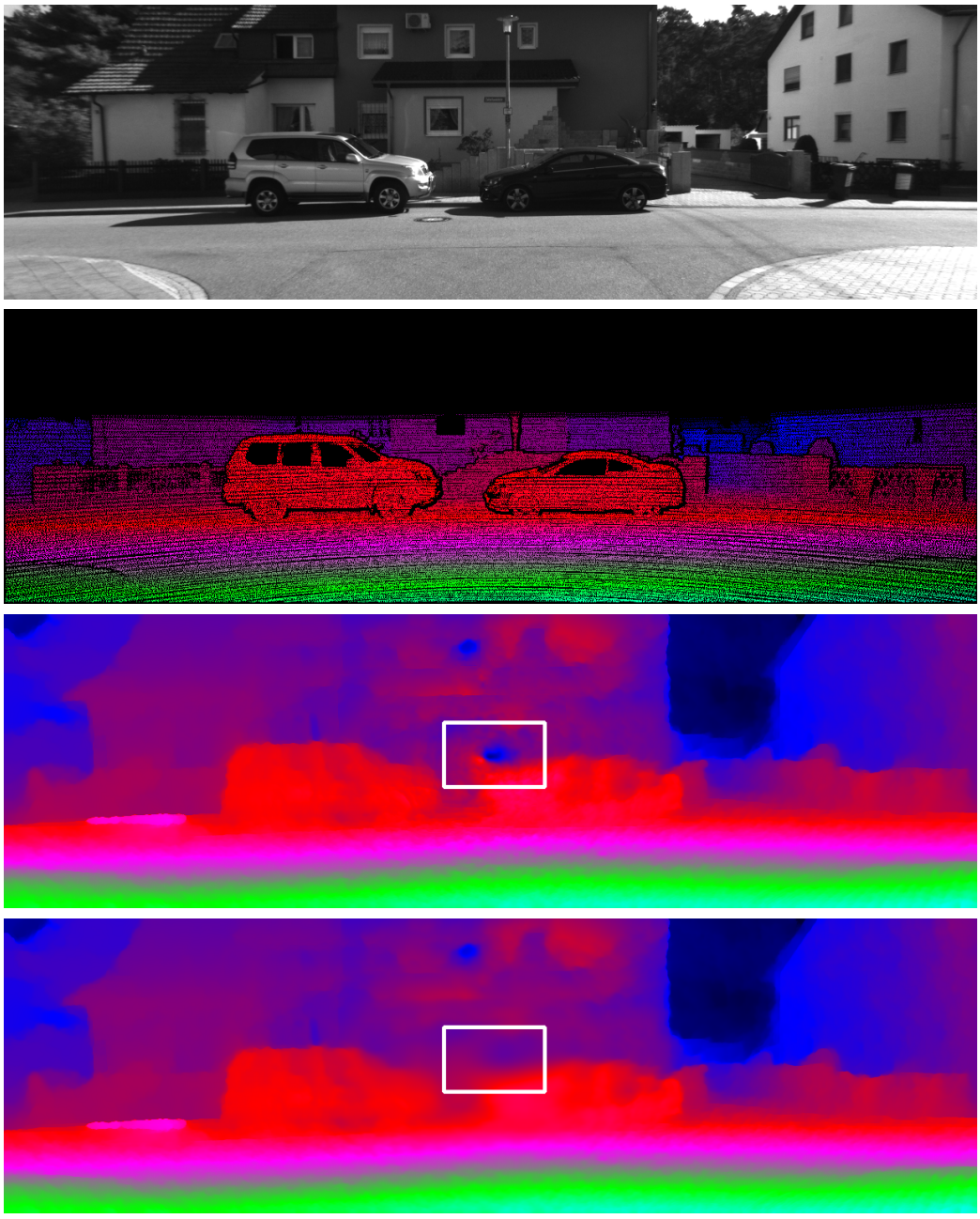
(a)

Figure 5.6: First row: gray value image, second row: ground truth disparity map, third row: minimum energy filter without spatial regularization, fourth row: minimum energy filter with spatial regularization.



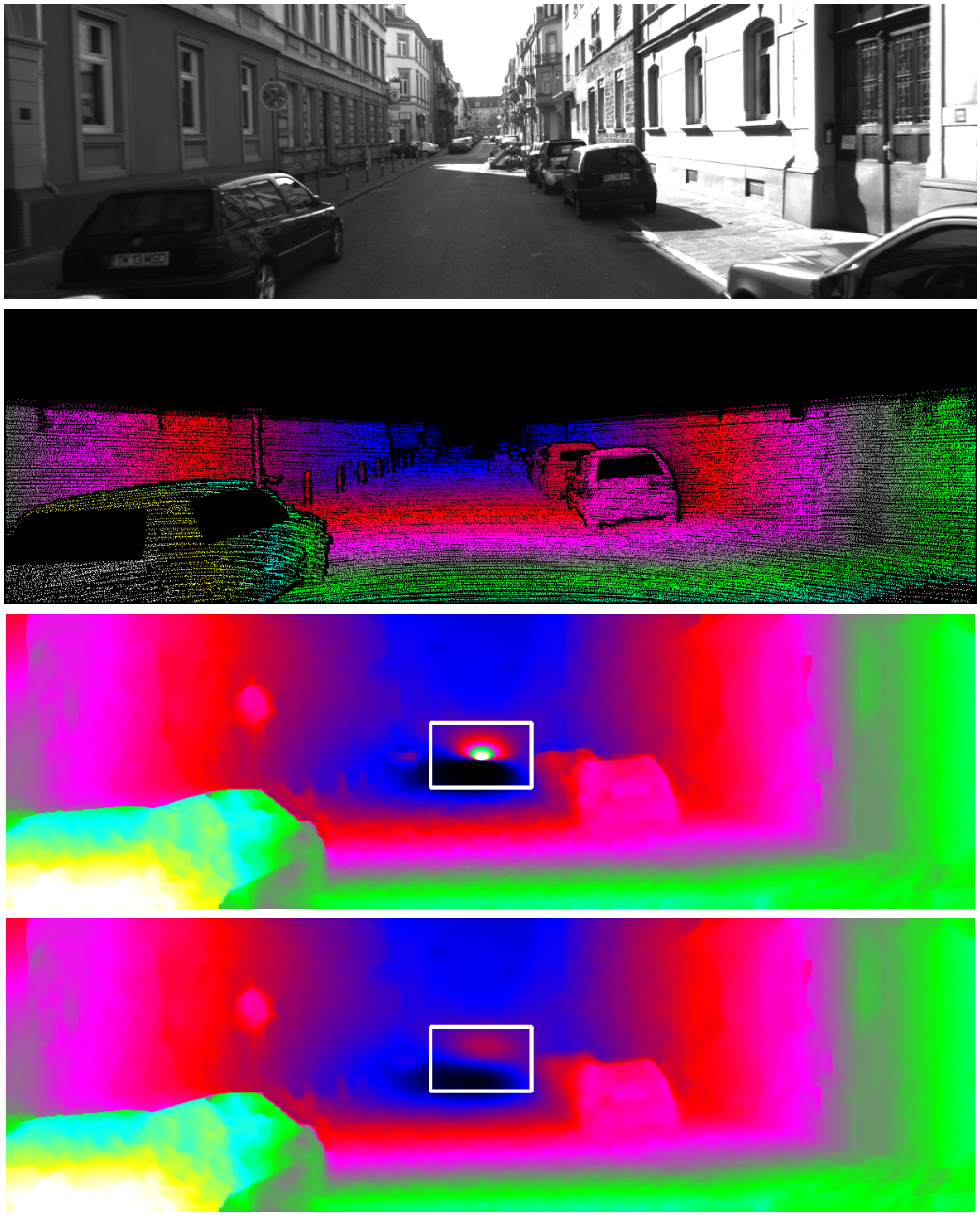
(b)

Figure 5.6: First row: gray value image, second row: ground truth disparity map, third row: minimum energy filter without spatial regularization, fourth row: minimum energy filter with spatial regularization.



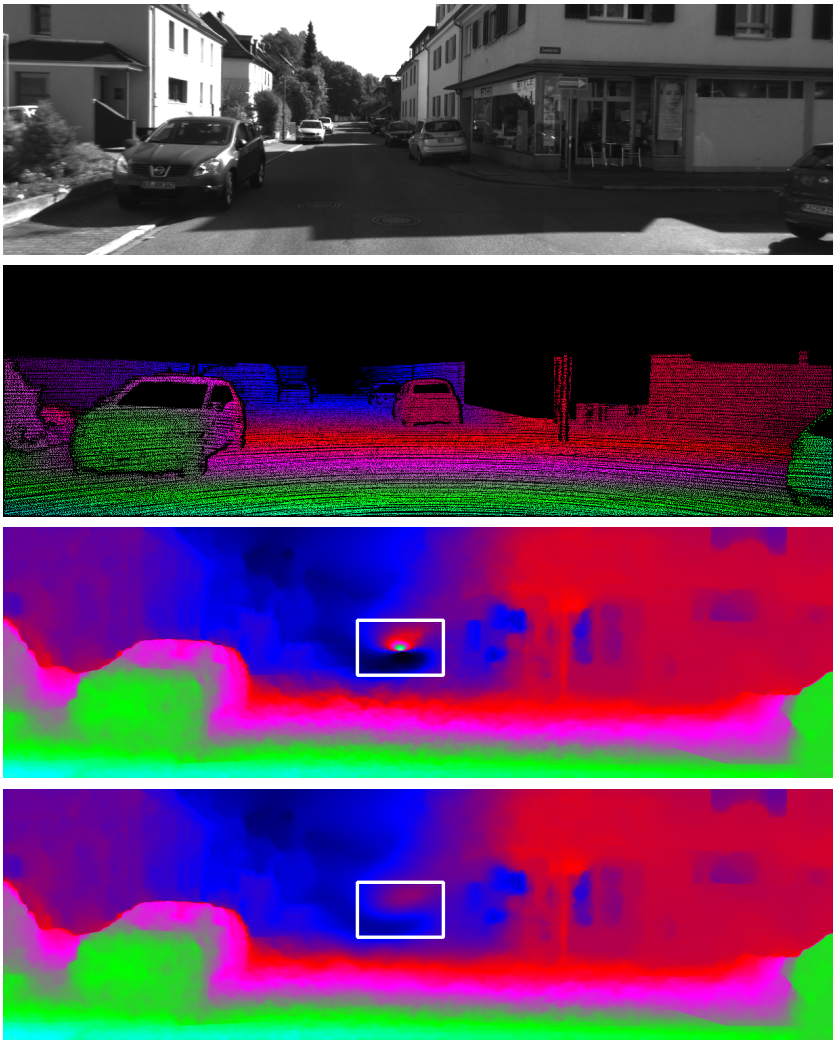
(c)

Figure 5.6: First row: gray value image, second row: ground truth disparity map, third row: minimum energy filter without spatial regularization, fourth row: minimum energy filter with spatial regularization.



(d)

Figure 5.6: First row: gray value image, second row: ground truth disparity map, third row: minimum energy filter without spatial regularization, fourth row: minimum energy filter with spatial regularization.



(e)

Figure 5.6: Reconstruction of selected disparity maps (a) – (e) of the KITTI stereo benchmark [36] with the corresponding color encoding (yellow/green is near, blue/black is far). For each reconstruction the minimum energy filter used data from nine consecutive frames. The top two images of each subfigure show the original gray value image and the ground truth disparity map. The third row in each subfigure corresponds to the proposed minimum energy filter without using a spatial regularizer (i.e. $\lambda = 0$ in (5.30)). The last row presents the reconstruction of the minimum energy filter with spatial regularizer ($\lambda > 0$). The white boxes indicate regions that are close to the epipole. One can easily recognize that these regions are better reconstructed by using the spatial regularizer, which we introduced in section 5.3.

5.5 Summary

In this chapter we extended the minimum energy filter, which was presented in the chapters before, to the monocular approach, where only data of one moving camera is available. In order to solve the corresponding filtering problem jointly (with respect to the unknown camera motion, higher order kinematics and the disparity map), we introduced a novel disparity group. This allows to model the whole problem on one specific product Lie group without the need of constraints (e.g. positive depth priors). Instead of a quadratic energy function we used the more general Charbonnier penalty function enabling robustness against outliers within the observations. In experiments we demonstrated that our approach is as accurate as state-of-the-art methods and additionally preserves small details. In total, we solved a non-linear filtering problem on a non-trivial product Lie group for high dimensional problems by means of second-order minimum energy filters. We also pointed out that the presented minimum energy filter is a rich tool for the solutions many filtering problems by adding additional regularizers within the energy function. Finally, we would like to refer the reader to [14] where some parts of this chapter are published.

Chapter 6

Conclusion

In this work we introduced the main concepts of minimum energy filtering theory based on the previous work of Mortensen [58], Zamani et al. [88] and Saccon et al. [72, 73] and applied the filter to problems of camera motion estimation and depth/disparity map estimation. First, we considered the Euclidean case and showed how the second-order optimal filter can be derived for this case. We introduced the main ideas of optimal control theory such as the energy function, the value function, Pontryagin's minimum principle and the Hamilton-Jacobi-Bellman equation. By calculation of the total time derivative of the necessary condition on the value function and omitting third and higher-order derivatives we obtained a second-order approximation of the optimal state. Finally, the second-order optimal state can be found by integration of a coupled system of ordinary differential equations. One might ask if it is possible to gain a better approximation of the optimal solution of the filtering problem which itself cannot be expressed explicitly. Mortensen [58] addressed this important point: "If one seeks a differential equations for these quantities [(third order tensor)], one needs to know the components of a tensor of the fourth rank, and so on *ad infinitum*." Thus, it is indeed possible to derive approximation of any (finite) order of the exact solution of the filtering problem. However, as mentioned by Zamani et al. [88] it is involved to evaluate tensors on Lie groups: "Similar to our previous filter derivation [...] we could continue with Mortensen's approach to derive a higher-order filter. However, this would require some tedious tensor algebra." Due to the long and technical evaluations of the second-order tensor across the whole work we can confirm this statement of Zamani et al. [88]. However, as an example we provided the *third-order optimal filter* for a one-dimensional state space which does not require tensor calculus. Experiments show that the third-order-optimal filter differs only slightly from the second-order optimal filter, which does not influence the mean cumulative error. On the other

hand, the computational effort of evaluating the third order operators grows quickly in high dimensions such that we think that the second-order optimal filter provides a good tradeoff between accuracy and runtime.
More important than comparing the different orders of approximation of the minimum energy filters, is the comparison to state-of-the-art stochastic filters, such as extended Kálmán filter, unscented Kálmán filter or particle filter. Within an academic example we showed that for usual filtering problems the proposed second-order minimum energy filter is as good as established methods. But for more involved filters, e.g. sinusoidal sensor, we showed that the minimum energy filter is superior. To the best knowledge of the author, this is the first experiment where the minimum energy filter was compared directly to established methods. We were surprised that the minimum energy filter performs so well in comparison to stochastic filters, since the latter are more flexible to correctly describe the true *a posteriori* distribution. The reason for this is, that the Hessian of the value function is computed explicitly for the second-order optimal minimum energy filter. In contrast, the presented stochastic filters only use the non-linear transformations (particle filter, unscented Kálmán filter) or first-order approximations (extended Kálmán filter), which result less accurate results.
In chapter 4 we considered the minimum energy filter for the application of estimating the unknown camera motion on the Lie group SE_3, which is also known as visual odometry. In this filtering problem the observations are given by the optical flow and the depth map, resulting in a non-linear dependence of the observations from the state. In contrast to a constant velocity model which is assumed in state-of-the-art visual odometry methods, we considered a *constant acceleration* model which is more suitable for real-world applications. Although this model is relatively simple and does not incorporate physical constraints, such as the Newtonian force laws [57], it results in accurate results that are as exact as modern visual odometry methods. In addition, we compared our approach to the approach of Bourmaud et al. [17] which is a generalized version of the extended Kálmán filter for Lie groups. In the case of a filtering problem with a linear dependence between state and observation equation we showed that the minimum energy filter is superior to the extended Kálmán filter in accuracy and the asymptotic error. Together with the results in chapter 2 this experiment indicates that the proposed filter outperforms the extended Kálmán filter for filtering problems on the Euclidean space as well as on Lie groups.
We extended this approach for joint recursive filtering of camera motion and scene structure in chapter 5. Unlike the stereo approach, which was implicitly used for the visual odometry estimation, the joint reconstruction disposes of less observations because only the monocular optical flow is given; simulta-

neously it needs to reconstruct a high-dimensional state, which makes this procedure difficult. For better numerical properties we used disparities (inverse of depth map) and introduced a novel disparity group. This allows to conduct all mathematical operations directly on this Lie group without needing projections, barrier functions or additional terms within the energy function. Moreover this representation is more suitable for regularization since the deviation of disparity differences is smaller than the deviation of depth differences. Next, we derived the state equation for the propagation of the optimal state of camera motion and disparity map. For the camera motion we used the higher-order kinematic model presented in chatper 4. For the propagation of the disparity map we investigated two approaches: a propagation with a continuous-time model based on planar scene assumptions and a discrete-time model. Although the continuous propagation model seems to fit better to the time-continuous minimum energy filter, we learned that for practical application the discrete-time propagation performs much better; the assumption of a piecewise planar scene is violated too often in real-world problems. After defining the energy function of the filtering problem we spotted that a quadratic energy function causes numerical problems because of outliers in the measurements and the properties of the exponential map of the Lie group. We solved this problems by replacing the quadratic energy by a generalized Charbonnier penalty function, which can be understood as a smooth approximation of the L^1 norm. Subsequently, we showed that Hamilton-Jacobi theory can be applied successfully to solve the corresponding optimal control problem. One takes note of the fact that the observed filtering problem is neither a linear filtering problem nor with quadratic (L^2) penalty function such that Hamilton-Jacobi theory only provides a necessary condition for optimality. Fortunately this is not a limitation for the considered application. By using a sparse approximation of the second-order operator we showed that numerical integration of the differential equations of optimal state and second-order operator is feasible if we use explicit integration schemes. However, implicit integration schemes which are numerically stabler, cannot be used by reason of the high dimension of the state space. Using explicit numerical schemes, we demonstrated that the derived filter gains accurate reconstructions of the disparity map and camera motion. By comparison with the monocular methods of Becker et al. [8] it was manifested that our method is better in most scenarios. Only regions close to the epipole (focus of expansion) were better reconstructed by the method of [8] due to an additional regularization terms. Thus, in a last step, we also incorporated a spatial regularization term to replace missing data around the epipole. The corresponding minimum energy filter contains both, temporal and spatial regularizers as well as propagation functions for the camera motion and the

disparity map. On the one hand this allows an improved reconstruction of the disparity map but on the other hand the second-order operator become dense, which is numerically infeasible. To cope this problem we proposed a simple numerical approximation which is numerically feasible and leads to state-of-the-art scene reconstructions.

Future Work Because the method of minimum energy filtering is relatively new compared to established stochastic filters, such as the extended Kálmán filter, the unscented Kálmán filter or the particle filter, there are plenty of applications where this approach can be adapted to. We showed that the minimum energy filter is superior to classical filters in the observed non-linear scenarios and we think that it will return accurate results in many modern applications. Especially for non-linear filtering problems on Lie groups or problems in high dimensions this filter leads to good results.
For gaining reconstructions with highest accuracy, we require exact numerical integration with many fine integration steps which is intensive in run-time. In practice, however, real-time solutions for filtering problems are often required. Therefore, for the classical filters (extended Kálmán filters, unscented Kálmán filters, particle filters) discrete-time filters were derived, which are much simpler and thus less time consuming. Although these methods are very fast, they are limited in accuracy and do not lead to state-of-the-art results. If runtime is the limiting resource, we propose to use a discrete-time version of the minimum energy filter. This can by easily derived by means of discrete optimal control theory, but we think that the accuracy will be inferior to the time-continuous minimum energy filter which we presented in this work.

Appendix A

Proofs regarding the Minimum Energy Filter

A.1 Minimum Energy Filter on Euclidean Space

A.1.1 Derivation of the Evolution Equation of the Optimal State

By using the chain rule we compute the total time derivative of the necessary condition (2.42) we gain the following equalities.

$$\begin{aligned}
0 \overset{!}{=} & \frac{d}{dt}\Big(\mathbf{D}_1\mathcal{V}(x,t;x_0,t_0)\Big|_{x=x^*}\Big) \\
= & \Big(\mathbf{D}_x\Big(\frac{\partial}{\partial t}\mathcal{V}(x,t;x_0,t_0) + \big(\mathbf{D}_1\mathcal{V}(x,t;x_0,t_0)\big)^\top \dot{x}(t)\Big)\Big)\Big|_{x=x^*} \\
\overset{(2.40)}{=} & \Big(\mathbf{D}_x\Big(\mathcal{H}\big(x,\mathbf{D}_1\mathcal{V}(x,t;x_0,t_0),t\big) + \big(\mathbf{D}_1\mathcal{V}(x,t;x_0,t_0)\big)^\top \dot{x}(t)\Big)\Big)\Big|_{x=x^*} \\
= & \Big(\mathbf{D}_x\mathcal{H}\big(x,\mathbf{D}_1\mathcal{V}(x,t;x_0,t_0),t\big) + \operatorname{Hess}_1\mathcal{V}(x,t;x_0,t_0)\dot{x}(t) \\
& \quad + (\mathbf{D}_x\dot{x}(t))^\top\mathbf{D}_1\mathcal{V}(x,t;x_0,t_0)\Big)\Big|_{x=x^*} \\
\overset{(2.42)}{=} & \Big(\mathbf{D}_x\mathcal{H}\big(x,\mathbf{D}_1\mathcal{V}(x,t;x_0,t_0),t\big) + \operatorname{Hess}_1\mathcal{V}(x,t;x_0,t_0)\dot{x}(t)\Big)\Big|_{x=x^*}
\end{aligned} \tag{A.1}$$

Here, $\mathbf{D}_x g(x)$ denotes also the Jacobian matrix of a function $g(x)$ as well as the gradient if the function is real valued. We require to compute the gradient of the Hamiltonian. It can be computed using the right hand side

of the Hamilton's equations (2.38) and (2.39) as well as the chain rule such that

$$\begin{aligned}\mathbf{D}_x\mathcal{H}\big(x,\mathbf{D}_1\mathcal{V}(x,t;x_0,t_0),t\big) &= \big(\mathbf{D}_1\mathcal{H}(x,\mathbf{D}_1\mathcal{V}(x,t;x_0,t_0),t)\big)\\ &\quad+\operatorname{Hess}_1\mathcal{V}(x,t;x_0,t_0)\big(\mathbf{D}_2\mathcal{H}(x,\mathbf{D}_1\mathcal{V}(x,t;x_0,t_0),t)\big)\\ \overset{(2.39),(2.38)}{=}&-\mathbf{D}_1 f(x,t)\mathbf{D}_1\mathcal{V}(x,t;x_0,t_0)+\mathbf{D}h(x(t))Q(y(t)-h(x(t)))\\ &\quad+\operatorname{Hess}_1\mathcal{V}(x,t;x_0,t_0)\big(-f(x(t))-R^{-1}\mathbf{D}_1\mathcal{V}(x,t;x_0,t_0)\big)\,.\end{aligned} \tag{A.2}$$

By insertion of this expression into (A.1) and by using the optimality condition (2.42) we obtain the following expression

$$\begin{aligned}0\overset{!}{=}&\Big(\operatorname{Hess}_1\mathcal{V}(x,t;x_0,t_0)\frac{d}{dt}x(t)-\big(\mathbf{D}f(x(t))\mathbf{D}_1\mathcal{V}(x,t;x_0,t_0)\\ &\quad-\mathbf{D}h(x(t))Q(y(t)-h(x(t))\big)\\ &\quad+\operatorname{Hess}_1\mathcal{V}(x,t;x_0,t_0))\big(-f(x(t))-R^{-1}\mathbf{D}_1\mathcal{V}(x,t;x_0,t_0)\big)\Big)\Big|_{x=x^*}\\ \overset{(2.42)}{=}&\operatorname{Hess}_1\mathcal{V}(x^*(t),t;x_0,t_0)\frac{d}{dt}x^*(t)-\mathbf{D}h(x^*(t))Q(y(t)-h(x^*(t)))\\ &\quad-\operatorname{Hess}_1\mathcal{V}(x^*,t;x_0,t_0)f(x^*,t)\\ \Leftrightarrow\quad&\operatorname{Hess}_1\mathcal{V}(x^*(t),t;x_0,t_0)\frac{d}{dt}x^*(t)=\operatorname{Hess}_1\mathcal{V}(x^*,t;x_0,t_0)f(x^*(t),t)\\ &\qquad+\mathbf{D}h(x^*(t))Q\big(y(t)-h(x^*(t))\big)\\ \Leftrightarrow\quad&\frac{d}{dt}x^*=f(x^*(t))+\big(\operatorname{Hess}_1\mathcal{V}(x^*,t;x_0,t_0)\big)^{-1}\mathbf{D}_1h(x^*(t))Q\big(y(t)-h(x^*(t))\big)\,.\end{aligned} \tag{A.3}$$

A.1.2 Derivation of the Evolution Equation of $\hat{Z}(t)$

The evaluation of the total time derivative of the operator $\hat{Z}(x,t)$ defined in (2.44) results in the following equality:

$$\begin{aligned}\frac{d}{dt}\hat{Z}(x^*,t)=&\frac{d}{dt}\big(\operatorname{Hess}_1\mathcal{V}(x^*,t;t_0,x_0)\big)\\ =&\operatorname{Hess}\Big(\frac{\partial}{\partial t}\mathcal{V}(x^*,t;t_0,x_0)\Big)+\mathbf{D}^3_{x^*}\mathcal{V}(x^*,t;t_0,x_0)\frac{d}{dt}x^*(t)\\ =&\operatorname{Hess}\Big(\frac{\partial}{\partial t}\mathcal{V}(x^*,t;t_0,x_0)\Big)+h.o.t.\end{aligned} \tag{A.4}$$

Here, $\mathbf{D}^3_{x^*}$ denotes the third order differential tensor of the value function that we simply replace by higher-order terms (*h.o.t.*).

The Hessian in (A.4) can be computed by differentiation of the expression (A.2) that results in the following calculation.

$$
\begin{aligned}
&\mathrm{Hess}\Big(\frac{\partial}{\partial t}\mathcal{V}(x^*,t;t_0,x_0)\Big) = \mathbf{D}_{x^*}\Big(\mathbf{D}_{x^*}\Big(\mathcal{H}(x^*,\mathbf{D}_1\mathcal{V}(x^*,t;x_0,t_0),t)\Big)\Big)\\
&\overset{(A.2)}{=} \mathbf{D}_{x^*}\Big(\mathbf{D}_1\mathcal{H}\big(x^*,\mathbf{D}_1\mathcal{V}(x^*,t;x_0,t_0),t\big)\\
&\qquad\qquad + \mathrm{Hess}_1\,\mathcal{V}(x^*,t;x_0,t_0)\cdot\mathbf{D}_2\mathcal{H}\big(x^*,\mathbf{D}_1\mathcal{V}(x^*,t;x_0,t_0),t\big)\Big)\\
&\quad = \mathrm{Hess}_1\,\mathcal{H}\big(x^*,\mathbf{D}\mathcal{V}(x^*,t;x_0,t_0),t\big)\\
&\qquad + (\mathbf{D}_2\mathbf{D}_1\mathcal{H}\big(x^*,\mathbf{D}_1\mathcal{V}(x^*,t;x_0,t_0),t\big))^\top\cdot\mathrm{Hess}_1\,\mathcal{V}(x^*,t;x_0,t)\\
&\qquad + \mathrm{Hess}_1\,\mathcal{V}(x^*,t;x_0,t_0)\cdot\mathbf{D}_1\mathbf{D}_2\mathcal{H}\big(x^*,\mathbf{D}\mathcal{V}(x^*,t;x_0,t_0),t\big)\\
&\qquad + \mathrm{Hess}_1\,\mathcal{V}(x^*,t;x_0,t_0)\cdot\mathrm{Hess}_2\,\mathcal{H}\big(x^*,\mathbf{D}_1\mathcal{V}(x^*,t;x_0,t_0),t\big)\cdot\mathrm{Hess}_1\,\mathcal{V}(x^*,t;x_0,t_0)\\
&\qquad + h.o.t.\\
&\overset{(2.42)}{=} \mathrm{Hess}_1\,\mathcal{H}(x^*,0,t)\\
&\qquad + (\mathbf{D}_2\mathbf{D}_1\mathcal{H}(x^*,0,t))^\top\cdot\mathrm{Hess}_1\,\mathcal{V}(x^*,t;x_0,t_0)\\
&\qquad + \mathrm{Hess}_1\,\mathcal{V}(x^*,t;x_0,t_0)\cdot\mathbf{D}_1\mathbf{D}_2\mathcal{H}(x^*,0,t)\\
&\qquad + \mathrm{Hess}_1\,\mathcal{V}(x^*,t;t_0,x_0)\cdot\mathrm{Hess}_2\,\mathcal{H}(x^*,0,t)\cdot\mathrm{Hess}_1\,\mathcal{V}(x^*,t;t_0,x_0) + h.o.t.\\
&\quad = \mathrm{Hess}_1\,\mathcal{H}(x^*,0,t) + (\mathbf{D}_2\mathbf{D}_1\mathcal{H}(x^*,0,t))^\top\cdot\hat{Z}(x^*,t)\\
&\qquad + \hat{Z}(x^*,t)\cdot\mathbf{D}_1\mathbf{D}_2\mathcal{H}(x^*,0,t) + \hat{Z}(x^*,t)\cdot\mathrm{Hess}_2\,\mathcal{H}(x^*,0,t)\cdot\hat{Z}(x^*,t) + h.o.t.
\end{aligned}
$$

One can check that the transposed matrices are correct. This can be verified by component-wise evaluation of the derivatives. Here, we also used the symmetry of the Hessian und thus omitted the corresponding transposed signs. Insertion of the full expression into (A.4) leads to

$$
\begin{aligned}
\frac{d}{dt}\hat{Z}(x^*,t) = {}& \mathrm{Hess}_1\,\mathcal{H}(x^*,0,t) + (\mathbf{D}_2\mathbf{D}_1\mathcal{H}(x^*,0,t))^\top\cdot\hat{Z}(x^*,t)\\
&+ \hat{Z}(x^*,t)\cdot\mathbf{D}_1\mathbf{D}_2\mathcal{H}(x^*,0,t) + \hat{Z}(x^*,t)\cdot\mathrm{Hess}_2\,\mathcal{H}(x^*,0,t)\cdot\hat{Z}(x^*,t)\\
&+ h.o.t.
\end{aligned}
\tag{A.5}
$$

A.2 Properties of SE_3 and $\mathcal{G}$

A.2.1 Kronecker Products on $\mathfrak{se}_3$

The Kronecker products $\otimes_{\mathfrak{se}}, \otimes_{\mathfrak{se}}^\top : \mathbb{R}^{4\times4}\times\mathbb{R}^{4\times4}\to\mathbb{R}^{6\times6}$ on $\mathfrak{se}_3$ are defined for matrices $A, B\in\mathbb{R}^{4\times4}$ and $\eta\in\mathfrak{se}_3$ through $\mathrm{vec}_{\mathfrak{se}}(A\eta B) =: (A\otimes_{\mathfrak{se}} B)\,\mathrm{vec}_{\mathfrak{se}}(\eta)$

and $\mathrm{vec}_{\mathfrak{se}}(A\eta^\top B) =: (A \otimes^\top_{\mathfrak{se}} B)\,\mathrm{vec}_{\mathfrak{se}}(\eta)$. Since the explicit formulas for $\otimes_{\mathfrak{se}}, \otimes^\top_{\mathfrak{se}}$ are quite uninformative, we do not provide them here.

A.2.2 Projection onto $\mathfrak{se}_3$

The projection $\mathrm{Pr} : \mathbb{R}^{4\times4} \to \mathfrak{se}_3$ is given by

$$\mathrm{Pr}(A) := \tfrac{1}{2}\,\mathrm{diag}((1,1,1,0)^\top)\big(A\,\mathrm{diag}((1,1,1,2)^\top) - A^\top\,\mathrm{diag}((1,1,1,0)^\top)\big)\,. \tag{A.6}$$

A.2.3 Adjoints, Exponential and Logarithmic Map

The adjoint operator $\mathrm{ad}_{\mathfrak{se}}(\mathrm{mat}_{\mathfrak{se}}(v))$ can be computed for a vector $v \in \mathbb{R}^6$ as follows

$$\begin{aligned}\mathrm{vec}_{\mathfrak{se}}\big(\mathrm{ad}_{\mathfrak{se}}(\mathrm{mat}_{\mathfrak{se}}(v))\eta\big) &= \mathrm{ad}^{\mathrm{vec}}_{\mathfrak{se}}(\mathrm{mat}_{\mathfrak{se}}(v))\,\mathrm{vec}_{\mathfrak{se}}(\eta)\\ &:= \begin{pmatrix}\mathrm{mat}_{\mathfrak{so}}(v_{1:3}) & \mathbf{0}_{3\times3}\\ \mathrm{mat}_{\mathfrak{so}}(v_{4:6}) & \mathrm{mat}_{\mathfrak{so}}(v_{1:3})\end{pmatrix}\mathrm{vec}_{\mathfrak{se}}(\eta)\,,\end{aligned} \tag{A.7}$$

where $\mathrm{mat}_{\mathfrak{so}}(v_{1:3}) := (\mathrm{mat}_{\mathfrak{se}}(v))_{1:3,1:3}$. This directly follows from the definition of the adjoint as Lie bracket, i.e. $\mathrm{ad}_{\mathfrak{se}}(\xi)\eta := [\xi,\eta]$ where the Lie bracket $[\cdot,\cdot] : \mathfrak{se}_3 \times \mathfrak{se}_3 \to \mathfrak{se}_3$ is simply the matrix commutator on $\mathfrak{se}_3$.

$$\mathrm{vec}_{\mathfrak{se}}(\mathrm{ad}_{\mathfrak{se}_3}(\mathrm{mat}_{\mathfrak{se}}(v))\eta) = \mathrm{vec}_{\mathfrak{se}}([\mathrm{mat}_{\mathfrak{se}}(v),\eta]) \tag{A.8}$$

$$= \mathrm{vec}_{\mathfrak{se}}(\mathrm{mat}_{\mathfrak{se}}(v)\eta\mathbb{1}_4 - \mathbb{1}_4\eta\,\mathrm{mat}_{\mathfrak{se}}(v)) \tag{A.9}$$

$$= \big(\mathrm{mat}_{\mathfrak{se}}(v) \otimes_{\mathfrak{se}} \mathbb{1}_4 - \mathbb{1}_4 \otimes_{\mathfrak{se}} \mathrm{mat}_{\mathfrak{se}}(v)\big)\,\mathrm{vec}_{\mathfrak{se}}(\eta)\,. \tag{A.10}$$

A componentwise evaluation of (A.10) leads to (A.7). Since $\mathbb{R}^6$ is trivial, the adjoint representation on $\mathfrak{g}$ parametrized by a vector $v \in \mathbb{R}^{12}$ is

$$\mathrm{ad}^{\mathrm{vec}}_{\mathfrak{g}}(\mathrm{mat}_{\mathfrak{g}}(v)) = \begin{pmatrix}\mathrm{ad}^{\mathrm{vec}}_{\mathfrak{se}}(v_{1:6}) & \mathbf{0}_{6\times6}\\ \mathbf{0}_{6\times6} & \mathbf{0}_{6\times6}\end{pmatrix}. \tag{A.11}$$

The exponential map $\mathrm{Exp}_{\mathrm{SE}_3} : \mathfrak{se}_3 \to \mathrm{SE}_3$ and the logarithmic map on SE_3 can be computed by the matrix exponential and matrix logarithm or more efficiently by the *Rodrigues' formula* as in [59, p. 413f].
Then the exponential map $\mathrm{Exp}_{\mathcal{G}} : \mathfrak{se}_3 \to \mathrm{SE}_3$ for a tangent vector $\eta = (\eta_1,\eta_2) \in \mathfrak{g}$ and the logarithmic map $\mathrm{Log}_G : \mathrm{SE}_3 \to \mathfrak{se}_3$ for $x = (E,v) \in \mathcal{G}$ are simply

$$\mathrm{Exp}_{\mathcal{G}}(\eta) = (\mathrm{Exp}_{\mathrm{SE}_3}(\eta_1),\eta_2) \in \mathcal{G}\,, \tag{A.12}$$

$$\mathrm{Log}_{\mathcal{G}}(x) = (\mathrm{Log}_{\mathrm{SE}_3}(E),v) \in \mathfrak{g}\,, \tag{A.13}$$

and similar for higher-order state spaces.

A.2.4 Vectorization of the Connection Function

Following [1, section 5.2], we can vectorize the connection function ω of the Levi-Civita connection ∇ for *constant* η, $\xi \in \mathfrak{g}$ in the following way:

$$\mathrm{vec}_{\mathfrak{g}}(\omega_\eta \xi) = \mathrm{vec}_{\mathfrak{g}}(\omega(\eta, \xi)) = \mathrm{vec}_{\mathfrak{g}}(\nabla_\eta \xi) = \tilde{\Gamma}_{\mathrm{vec}_{\mathfrak{g}}(\xi)} \, \mathrm{vec}_{\mathfrak{g}}(\eta)\,, \tag{A.14}$$

where $\tilde{\Gamma}_x$ is the matrix whose (i, j) element is the real-valued function

$$(\tilde{\Gamma}_\gamma)_{i,j} := \sum_k (\gamma_k \Gamma^i_{jk})\,, \tag{A.15}$$

and Γ^i_{jk} are the *Christoffel symbols* of the connection function ω for a vector $\gamma \in \mathbb{R}^{12}$. Similarly, permuting indices, we can define the adjoint matrix $\tilde{\Gamma}^*_\gamma$ whose (i, j)-th element is given by

$$(\tilde{\Gamma}^*_\gamma)_{i,j} := \sum_k (\gamma_k \Gamma^i_{kj})\,. \tag{A.16}$$

This leads to the following equality:

$$\mathrm{vec}_{\mathfrak{g}}(\omega_\eta \xi) = \tilde{\Gamma}^*_{\mathrm{vec}_{\mathfrak{g}}(\eta)} \, \mathrm{vec}_{\mathfrak{g}}(\xi)\,. \tag{A.17}$$

If the expression ξ in (A.14) is *non-constant*, we obtain the following vectorization from [1, Eq. (5.7)], for the case of the Lie algebra $\mathfrak{se}_3$, i.e.

$$\begin{aligned}
&\mathrm{vec}_{\mathfrak{se}}(\nabla_{\eta_x} \xi(x)) \\
=&\tilde{\Gamma}_{\mathrm{vec}_{\mathfrak{g}}(\xi(x))} \, \mathrm{vec}_{\mathfrak{se}}(\eta_x) + \mathbf{D} \, \mathrm{vec}_{\mathfrak{se}}(\xi(x))[\mathrm{vec}_{\mathfrak{se}}(\eta_x)] \\
=&\tilde{\Gamma}_{\mathrm{vec}_{\mathfrak{g}}(\xi(x))} \, \mathrm{vec}_{\mathfrak{se}}(\eta_x) + \sum_i (\eta_x)_i \, \mathrm{vec}_{\mathfrak{se}}(\mathbf{D}\xi(x))[E^i]) \\
=&\tilde{\Gamma}_{\mathrm{vec}_{\mathfrak{g}}(\xi(x))} \, \mathrm{vec}_{\mathfrak{se}}(\eta_x) + D \, \mathrm{vec}_{\mathfrak{se}}(\eta_x)\,,
\end{aligned} \tag{A.18}$$

where the entries of the matrix $D \in \mathbb{R}^{6\times 6}$ can be computed as

$$(D)_{i,j} = (\mathrm{vec}_{\mathfrak{se}}(\mathbf{D}\xi(x)[E^j]))_i\,, \quad E^j = \mathrm{mat}_{\mathfrak{se}}(e^6_j)\,, \tag{A.19}$$

where e^6_j denotes the j-th unit vector in $\mathbb{R}^6$.

A.2.5 Christoffel symbols of SE_3

The Christoffel symbols $\Gamma^k_{ij}, i, j, k \in \{1, \ldots, 6\}$ for the Riemannian connection on SE_3 are given by

$$\begin{aligned}
\Gamma^3_{12} = \Gamma^1_{23} = \Gamma^2_{31} &= \tfrac{1}{2}\,, \\
\Gamma^2_{13} = \Gamma^3_{21} = \Gamma^1_{32} &= -\tfrac{1}{2}\,, \\
\Gamma^6_{15} = \Gamma^4_{26} = \Gamma^5_{34} &= 1\,, \\
\Gamma^5_{16} = \Gamma^6_{24} = \Gamma^4_{35} &= -1\,.
\end{aligned}$$

and zero otherwise. Note that this Christoffel symbols are similar to these of the *kinematic* connection in [89]. However, for the *Riemannian* connection, we need to switch the indexes i and j.

A.3 Supplemental Material of Chapter 4

A.3.1 Proofs

Proof of Proposition 4.2.1. The tangent map is simply the differential or directional derivative. For $x_1 = (E_1, v_1), x_2 = (E_2, v_2) \in \mathcal{G}$ it holds $T_{x_2}L_{x_1} : T_{x_2}\mathcal{G} \to T_{L_{x_1}(x_2)}\mathcal{G}$. Thus, we can compute it for a $\eta = (E_2\eta_1, \eta_2) \in T_{x_2}\mathcal{G} = T_{E_2}\,\mathrm{SE}_3 \times \mathbb{R}^6$ as follows

$$\begin{aligned}
T_{x_2}L_{x_1} \circ \eta &= \mathbf{D}L_{x_1}(x_2)[\eta] \\
&= \lim_{\tau \to 0^+} \tau^{-1}\big(L_{x_1}(x_2 + \tau\eta) - L_{x_1}(x_2)\big) \\
&= \lim_{\tau \to 0^+} \tau^{-1}\big(L_{(E_1,v_1)}((E_2 + \tau E_2\eta_1, v_2 + \tau\eta_2)) \\
&\qquad - (E_1E_2, v_1 + v_2)\big) \\
&= \lim_{\tau \to 0^+} \tau^{-1}\big((E_1E_2 + \tau E_1E_2\eta_1, v_1 + v_2 + \tau\eta_2) \\
&\qquad - (E_1E_2, v_1 + v_2)\big) \\
&= (x_1x_2\eta_1, \eta_2) \in T_{x_1x_2}\mathcal{G} = T_{L_{x_1}(x_2)}\mathcal{G}\,.
\end{aligned}$$

For $x_2 = \mathrm{Id} = (\mathbb{1}_4, \mathbf{0}_6)$ and $\eta = (\eta_1, \eta_2) \in \mathfrak{g}$, it follows

$$T_{\mathrm{Id}}L_{x_1} \circ \eta = (E_1\eta_1, \eta_2) = L_{(E_1,\mathbf{0}_6)}(\eta_1, \eta_2) =: x_1\eta \in T_{x_1}\mathcal{G}\,.$$

Note that the adjoint of the tangent map of L_x at identity can be expressed as inverse of $x = (E, v)$, i.e. for $\eta = (\eta_1, \eta_2) \in T_x\mathcal{G}$ and $\xi = (\xi_1, \xi_2) \in \mathfrak{g}$

$$\begin{aligned}
\langle T_{\mathrm{Id}}L_x^*\eta, \xi\rangle_{\mathrm{Id}} &= \langle \eta, T_{\mathrm{Id}}L_x\xi\rangle_x \\
&= \langle \eta_1, E\xi_1\rangle_E + \langle \eta_2, \xi_2\rangle \\
&= \langle E^{-1}\eta_1, \xi_1\rangle_{\mathrm{Id}_{\mathfrak{se}_3}} + \langle \eta_2, \xi_2\rangle \\
&= \langle L_{(E^{-1},\mathbf{0}_6)}\eta, \xi\rangle_{\mathrm{Id}}\,.
\end{aligned}$$

Thus, $T_{\mathrm{Id}}L_x^*\eta = L_{(E^{-1},\mathbf{0}_6)}\eta$. We will use the shorthand $x^{-1}\eta := T_{\mathrm{Id}}L_x^*$ for the dual of the tangent map of L_x at identity. □

Proof of Lemma 4.3.6. Since $\mu = (\mu_1, \mu_2), v$ are independent of E the gradient $\mathbf{D}_1\mathcal{H}^-(x = (E, v), \mu, t)$ can be computed separately in terms of E, i.e.

for $\eta = (E\eta_1, \eta_2) \in T_x\mathcal{G}$

$$\begin{aligned}\mathbf{D}_1\mathcal{H}^-(x,\mu,t)[\eta] =&\Big(\mathbf{D}_E \tfrac{1}{2} e^{-\alpha(t-t_0)}\Big(\sum_{z\in\Omega}\|y_z - h_z(E)\|_Q^2\Big)[\eta_1],\\ &- \mathbf{D}_v\langle\mu_1, \mathrm{mat}_{\mathfrak{se}}(v)\rangle[\eta_2]\Big)\,.\end{aligned}$$

The directional derivative regarding v can be computed by the usual gradient on $\mathbb{R}^6$ which is given by

$$\begin{aligned}-\mathbf{D}_v\langle\mu_1, \mathrm{mat}_{\mathfrak{se}}(v)\rangle[\eta_2] =& -\langle \mathrm{vec}_{\mathfrak{se}}(\mu_1), \mathbf{D}_v v[\eta_2]\rangle\\ =&\langle -\mathrm{vec}_{\mathfrak{se}}(\mu_1), \eta_2\rangle\,,\end{aligned} \tag{A.20}$$

such that $\mathbf{D}_v\langle\mu_1, \mathrm{mat}_{\mathfrak{se}}(v)\rangle = -\mathrm{vec}_{\mathfrak{se}}(\mu_1)$. For the directional derivative of $\mathcal{H}^-$ we first consider the directional derivative of $h_z(E)$. Since $h_z(E)$ can also be written as

$$h_z(E) := ((e_3^4)^\top E^{-1} g_z(t))^{-1} \hat{I} E^{-1} g_z(t)\,, \quad \hat{I} := \left(\begin{smallmatrix}1&0&0&0\\0&1&0&0\end{smallmatrix}\right)\,, \tag{A.21}$$

the directional derivative (into direction ξ) can be derived by the following matrix calculus.

$$\begin{aligned}&\mathbf{D}h_z(E)[\xi] &\text{(A.22)}\\ =&\mathbf{D}\big(((e_3^4)^\top E^{-1} g_z)^{-1}\big)[\xi]\hat{I}E^{-1}g_z + ((e_3^4)^\top E^{-1} g_z)^{-1}\mathbf{D}\big(\hat{I}E^{-1}g_z\big)[\xi]\\ =&-\kappa_z^{-1}\mathbf{D}((e_3^4)^\top E^{-1} g_z)[\xi]\kappa_z^{-1}\hat{I}E^{-1}g_z + \kappa_z^{-1}\hat{I}\mathbf{D}(E^{-1})[\xi]g_z\\ =&-\kappa_z^{-1}(e_3^4)^\top\mathbf{D}(E^{-1})[\xi]g_z\kappa_z^{-1}\hat{I}E^{-1}g_z + \kappa_z^{-1}\hat{I}\mathbf{D}(E^{-1})[\xi]g_z\\ =&-\kappa_z^{-1}(e_3^4)^\top(-1)E^{-1}\mathbf{D}(E)[\xi]E^{-1}g_z\kappa_z^{-1}\hat{I}E^{-1}g_z + \kappa_z^{-1}\hat{I}(-1)E^{-1}\mathbf{D}(E)[\xi]E^{-1}g_z\\ =&\kappa_z^{-2}(e_3^4)^\top E^{-1}\xi E^{-1}g_z\hat{I}E^{-1}g_z - \kappa_z^{-1}\hat{I}E^{-1}\xi E^{-1}g_z\,, &\text{(A.23)}\end{aligned}$$

where $\kappa_z = \kappa_z(E) := (e_3^4)^\top E^{-1} g_z$. Then for the choice $\xi = E\eta_1$ we find that

$$\begin{aligned}&e^{\alpha(t-t_0)}\mathbf{D}_1\mathcal{H}^-(x,\mu,t)[E\eta_1] &\text{(A.24)}\\ =&-\sum_{z\in\Omega}\mathrm{tr}\big(\mathbf{D}h_z(E)[E\eta_1](y_z - h_z(E))^\top Q\big)\\ =&-\sum_{z\in\Omega}\mathrm{tr}\big(\big(\kappa_z^{-2}((e_3^4)^\top\eta_1 E^{-1}g_z)\hat{I}E^{-1}g_z - \kappa_z^{-1}\hat{I}\eta_1 E^{-1}g_z\big)(y_z - h_z(E))^\top Q\big)\\ =&\sum_{z\in\Omega}\mathrm{tr}\big(\big(\kappa_z^{-1}\hat{I}\eta_1 E^{-1}g_z - \kappa_z^{-2}((e_3^4)^\top\eta_1 E^{-1}g_z)\hat{I}E^{-1}g_z\big)(y_z - h_z(E))^\top Q\big)\\ =&\sum_{z\in\Omega}\mathrm{tr}\big(\big(\kappa_z^{-1}\hat{I}\eta_1 E^{-1}g_z - \kappa_z^{-2}\hat{I}E^{-1}g_z(e_3^4)^\top\eta_1 E^{-1}g_z\big)(y_z - h_z(E))^\top Q\big)\end{aligned}$$

$$
\begin{aligned}
&= \sum_{z\in\Omega} \operatorname{tr}\big(\big(\kappa_z^{-1}\hat{I} - \kappa_z^{-2}\hat{I}E^{-1}g_z(e_3^4)^\top\big)\eta_1 E^{-1}g_z(y_z - h_z(E))^\top Q\big) \\
&= \sum_{z\in\Omega} \operatorname{tr}\big(E^{-1}g_z(y_z - h_z(E))^\top Q\big(\kappa_z^{-1}\hat{I} - \kappa_z^{-2}\hat{I}E^{-1}g_z(e_3^4)^\top\big)\eta_1\big) \\
&= \sum_{z\in\Omega} \Big\langle \underbrace{\big(\kappa_z^{-1}\hat{I} - \kappa_z^{-2}\hat{I}E^{-1}g_z(e_3^4)^\top\big)^\top Q(y_z - h_z(E))g_z^\top E^{-\top}}_{=:A_k(E)}, \eta_1 \Big\rangle_{\mathrm{Id}}. \qquad \text{(A.25)}
\end{aligned}
$$

Here we used that the trace is cyclic. We obtain the Riemannian gradient on SE_3 by projecting (cf. [1, section 3.6.1]) the left side of the Riemannian metric in (A.25) onto $T_E\,\mathrm{SE}_3$, which is for $x = (E, v)$

$$
\begin{aligned}
\mathbf{D}_E\mathcal{H}^-(x,\mu,t) =& e^{-\alpha(t-t_0)}\operatorname{Pr}_E\big(EA_z(E)\big) \\
=& e^{-\alpha(t-t_0)}\sum_k E\operatorname{Pr}\big(A_z(E)\big)\,, \qquad \text{(A.26)}
\end{aligned}
$$

with $A_z(E) := \big(\kappa_z^{-1}\hat{I} - \kappa_z^{-2}\hat{I}E^{-1}g_z(e_3^4)^\top\big)^\top Q(y_z - h_z(E))g_z^\top E^{-\top}$, and $\operatorname{Pr}_E : \mathrm{GL}_4 \to T_E\,\mathrm{SE}_3$ denotes the projection onto the tangential space $T_E\,\mathrm{SE}_3$ that can be expressed in terms of $\operatorname{Pr}_E(E\cdot) = E\operatorname{Pr}(\cdot)$. Besides, $\operatorname{Pr} : \mathrm{GL}_4 \to \mathfrak{se}_3$ denotes the projection onto the Lie algebra $\mathfrak{se}_3$ as given in (A.6). Putting together (A.20) and (A.26) results in

$$
\mathbf{D}_1\mathcal{H}^-(x,\mu,t) = \Big(e^{-\alpha(t-t_0)}\sum_{k=1}^{n} E\operatorname{Pr}\big(A_z(E)\big), -\operatorname{vec}_{\mathfrak{se}}(\mu_1)\Big) \in T_x\mathcal{G}\,. \qquad \text{(A.27)}
$$

□

Proof of Lemma 4.3.7. Eq. (4.43) can be easily found by considering a basis of $\mathfrak{se}_3$ and the fact that Z is a linear operator on the Lie algebra. Since the resulting matrix $K(t)\operatorname{vec}_{\mathfrak{g}}(\eta) := Z(x^*, t)\circ\eta$ depends only on t, the equation (4.44). Eq.(4.45) is trivial since Z is linear.

1. With the symmetry of the Levi-Civita connection, i.e.

$$
[\eta, \xi] = \nabla_\eta\xi - \nabla_\xi\eta\,, \qquad \text{(A.28)}
$$

we gain the following equalities

$$
\begin{aligned}
&\operatorname{vec}_{\mathfrak{g}}(Z(x^*,t)\circ\omega_{(x^*)^{-1}\dot{x^*}}\eta + Z(x^*,t)\circ\omega_{\overleftrightarrow{\mathbf{D}_2\mathcal{H}^-(x^*,0,t)}}\eta)\\
&\overset{(4.43)}{=} K(t)\operatorname{vec}_{\mathfrak{g}}(\omega_{(x^*)^{-1}\dot{x^*}}\eta + \omega_{\overleftrightarrow{\mathbf{D}_2\mathcal{H}^-(x^*,0,t)}}\eta)\\
&\overset{(4.40)}{=} K(t)\operatorname{vec}_{\mathfrak{g}}(\nabla_{-\mathbf{D}_2\mathcal{H}^-(x^*,0,t)}\eta\\
&\qquad - \nabla_{e^{-\alpha(t-t_0)}Z(x^*,t)^{-1}\circ r_t(x^*)}\eta + \nabla_\eta \mathbf{D}_2\mathcal{H}^-(x^*,0,t))\\
&\overset{(A.28)}{=} K(t)\operatorname{vec}_{\mathfrak{g}}(-[\mathbf{D}_2\mathcal{H}^-(x^*,0,t),\eta]) - \nabla_{e^{-\alpha(t-t_0)}Z(x^*,t)^{-1}\circ r_t(x^*)}\eta\\
&\overset{(A.17)}{=} K(t)\big(\operatorname{vec}_{\mathfrak{g}}([f(x^*),\eta]) - \tilde{\Gamma}^*_{\operatorname{vec}_{\mathfrak{g}}(e^{-\alpha(t-t_0)}Z(x^*,t)^{-1}\circ r_t(x^*))}\operatorname{vec}_{\mathfrak{g}}(\eta)\big)\\
&\overset{(4.43)}{=} K(t)\big(\operatorname{vec}_{\mathfrak{g}}([f(x^*),\eta]) + \tilde{\Gamma}^*_{-e^{-\alpha(t-t_0)}K(t)^{-1}\operatorname{vec}_{\mathfrak{g}}(r_t(x^*))}\big)\operatorname{vec}_{\mathfrak{g}}(\eta)\big)\\
&\overset{(A.11)}{=} K(t)\big(\operatorname{ad}^{\operatorname{vec}}_{\mathfrak{g}}(f(x^*)) + \tilde{\Gamma}^*_{-e^{-\alpha(t-t_0)}K(t)^{-1}\operatorname{vec}_{\mathfrak{g}}(r_t(x^*))}\big)\operatorname{vec}_{\mathfrak{g}}(\eta)\\
&\quad =: K(t)B\operatorname{vec}_{\mathfrak{g}}(\eta)\,.
\end{aligned}
\tag{A.29}
$$

The claim follows from the fact that the adjoints and the Christoffel symbols on $\mathbb{R}^6$ are zero.

2. Since this expression is dual to the expression in 1. the claim follows by using its transpose.

3. Recall that the Hamiltonian in (4.27) is given by

$$
\begin{aligned}
\mathcal{H}^-((E,v),\mu,t) = \tfrac{1}{2}e^{-\alpha(t-t_0)}\Big(\sum_{z\in\Omega}\|y_z - h_z(E)\|^2_Q\Big)\\
- \tfrac{1}{2}e^{\alpha(t-t_0)}\Big(\langle\mu_1, \operatorname{mat}_{\mathfrak{se}}(S_1^{-1}\operatorname{vec}_{\mathfrak{se}}(\mu_1))\rangle_{\mathrm{Id}}\\
+ \langle\mu_2, S_2^{-1}\mu_2\rangle\Big) - \langle\mu_1, \operatorname{mat}_{\mathfrak{se}}(v)\rangle_{\mathrm{Id}}\,.
\end{aligned}
$$

The Riemannian Hessian w.r.t. the first component can be computed for $x = (E,v) \in \mathcal{G}$, $\eta = (\eta_1,\eta_2) \in \mathfrak{g}$ and the choice $\mu = (\mu_1,\mu_2) = (\mathbf{0}_{4\times4}, \mathbf{0}_6)$ as

$$
\begin{aligned}
&e^{\alpha(t-t_0)}\operatorname{vec}_{\mathfrak{g}}(x^{-1}\operatorname{Hess}_1\mathcal{H}^-(x,\mu,t)[x\eta])\\
&= e^{\alpha(t-t_0)}\operatorname{vec}_{\mathfrak{g}}\Big(x^{-1}\nabla_{x\eta}\mathbf{D}_1\mathcal{H}^-(x,\mathbf{0},t)\Big) &\text{(A.30)}\\
&= e^{\alpha(t-t_0)}\operatorname{vec}_{\mathfrak{g}}\Big(\nabla_\eta x^{-1}\mathbf{D}_1\mathcal{H}^-(x,\mathbf{0},t)\Big) &\text{(A.31)}\\
&= \operatorname{vec}_{\mathfrak{g}}\Big(\nabla_\eta\big(\sum_{z\in\Omega}\Pr(A_z(E)), -e^{\alpha(t-t_0)}\operatorname{vec}_{\mathfrak{se}}(\mathbf{0}_{4\times4})\big)\Big) &\text{(A.32)}
\end{aligned}
$$

$$
\begin{aligned}
&= \Big(\sum_{z\in\Omega} \operatorname{vec}_{\mathfrak{se}}\big(\nabla_{\eta_1} \Pr\big(A_z(E)\big)\big), \mathbf{0}_6\Big) \\
&= \sum_{z\in\Omega}\Big(\tilde{\Gamma}_{\operatorname{vec}_{\mathfrak{g}}\left(\Pr(A_z(E))\right)} \operatorname{vec}_{\mathfrak{se}}(\eta_1) \\
&\qquad + \sum_i (\operatorname{vec}_{\mathfrak{se}}(\eta_1))_i \operatorname{vec}_{\mathfrak{se}}(\mathbf{D}\Pr\big(A_z(E)\big))[E^i])\Big)\,.
\end{aligned} \tag{A.33}
$$

Here, line (A.30) follows from the general definition of the Hessian (cf. [1, Def. 5.5.1]). Line (A.31) holds because of the linearity of the affine connection, the equation (A.32) results from insertion of the expression in Lemma 4.3.6 and (A.33) can be achieved with (A.18).

As next we calculate the differential $\mathbf{D}\Pr(A_z(E))[\eta_1]$ in (A.33) for an arbitrary direction η_1. Since the projection is a linear operation (cf. (A.6)), i.e. $\mathbf{D}\Pr(A_z(E))[\eta_1] = \Pr(\mathbf{D}A_z(E)[\eta_1])$, we require to calculate $\mathbf{D}A_z(E)[\eta_1]$. By using the product rule and the definition of A_z from (4.32) we obtain

$$
\begin{aligned}
&\mathbf{D}A_z(E)[\eta_1] \\
=&\mathbf{D}\big(\big(\kappa_z^{-1}\hat{I} - \kappa_z^{-2}\hat{I}E^{-1}g_z(e_3^4)^\top\big)^\top Q(y_z - h_z(E))g_z^\top E^{-\top}\big)[\eta_1] \\
=&\big(\mathbf{D}\big(\kappa_z^{-1}\hat{I} - \kappa_z^{-2}\hat{I}E^{-1}g_z(e_3^4)^\top\big)^\top[\eta_1]Q(y_z - h_z(E))g_z^\top E^{-\top}\big) \\
+&\big(\kappa_z^{-1}\hat{I} - \kappa_z^{-2}\hat{I}E^{-1}g_z(e_3^4)^\top\big)^\top Q\Big((-\mathbf{D}h_z(E)[\eta_1])g_z^\top E^{-\top}\big) \\
+&\big((y_z - h_z(E))g_z^\top \mathbf{D}E^{-\top}[\eta_1]\big)\Big)\,.
\end{aligned} \tag{A.34}
$$

The directional derivative of $\big(\kappa_z^{-1}\hat{I} - \kappa_z^{-2}\hat{I}E^{-1}g_z(e_3^4)^\top\big)$ is

$$
\begin{aligned}
&\mathbf{D}\big(\kappa_z^{-1}\hat{I} - \kappa_z^{-2}\hat{I}E^{-1}g_z(e_3^4)^\top\big)[\eta_1] \\
=& -\kappa_z^{-2}(e_3^4)^\top \mathbf{D}E^{-1}[\eta_1]g_z\hat{I} \\
&\quad + 2\kappa_z^{-3}(e_3^4)^\top \mathbf{D}E^{-1}[\eta_1]g_z\hat{I}E^{-1}g_z(e_3^4)^\top \\
&\quad\quad - \kappa_z^{-2}\hat{I}\mathbf{D}E^{-1}[\eta_1]g_z(e_3^4)^\top \\
=&\kappa_z^{-2}(e_3^4)^\top E^{-1}\eta E^{-1}g_z\hat{I} \\
&\quad - 2\kappa_z^{-3}(e_3^4)^\top E^{-1}\eta_1 E^{-1}g_z\hat{I}E^{-1}g_z(e_3^4)^\top \\
&\quad\quad + \kappa_z^{-2}\hat{I}E^{-1}\eta_1 E^{-1}g_z(e_3^4)^\top\,.
\end{aligned} \tag{A.35}
$$

By inserting the directional derivatives (A.35), (A.23) and $\mathbf{D}E^{-\top}[\eta_1] = -(E^{-1}\eta_1 E^{-1})^\top$ into (A.34), we obtain the vector-valued function

$\zeta^k(E)(\cdot) : \mathfrak{se}_3 \to \mathbb{R}^6$ defined as

$$\begin{aligned}\operatorname{mat}_{\mathfrak{se}}(\zeta^k(E)(\eta_1)) &:= \operatorname{Pr}\big(\mathbf{D}A_z(E)[\eta_1]\big) \\ &= \operatorname{Pr}\Bigg(\Big(\kappa_z^{-2}(e_3^4)^\top E^{-1}\eta_1 E^{-1} g_z \hat{I} \\ &\quad - 2\kappa_z^{-3}(e_3^4)^\top E^{-1}\eta_1 E^{-1} g_z \hat{I} E^{-1} g_z (e_3^4)^\top \\ &\quad + \kappa_z^{-2}\hat{I}E^{-1}\eta_1 E^{-1} g_z (e_3^4)^\top\Big)^\top Q\big(y_z - h_z(E)\big) g_z^\top E^{-\top} \\ &\quad + \big(\kappa_z^{-1}\hat{I} - \kappa_z^{-2}\hat{I}E^{-1}g_z(e_3^4)^\top\big)^\top Q\Big(\big(\kappa_z^{-1}\hat{I}E^{-1}\eta_1 E^{-1} g_z \\ &\quad - \kappa_z^{-2}(e_3^4)^\top E^{-1}\eta_1 E^{-1} g_z \hat{I} E^{-1} g_z\big) g_z^\top E^{-\top} \\ &\quad - \big(y_z - h_z(E)\big) g_z^\top E^{-\top}\eta_1^\top E^{-\top}\Big)\Bigg).\end{aligned} \tag{A.36}$$

Using the basis $\{E^j\}_{j=1}^6$ of $\mathfrak{se}_3$, with $E^j := \operatorname{mat}_{\mathfrak{se}}(e_j^6)$ we define, as in (A.19), the following matrix $D_z(E) \in \mathbb{R}^{6\times 6}$ with components

$$(D_z(E))_{i,j} := \zeta_i^k(E^j)\,. \tag{A.37}$$

By using the equation (A.18) we find that

$$\operatorname{vec}_{\mathfrak{se}}\big(\nabla_{\eta_1}\operatorname{Pr}(A_z(E))\big) = \big(\tilde{\Gamma}_{\operatorname{Pr}(A_z(E))} + D_z(E)\big)\operatorname{vec}_{\mathfrak{se}}(\eta_1)\,.$$

Insertion of this expression into (A.33) leads finally to the desired result, i.e.

$$\begin{aligned}&e^{\alpha(t-t_0)}\operatorname{vec}_{\mathfrak{g}}(x^{-1}\operatorname{Hess}_1\mathcal{H}^-(x,\mu,t)[x\eta]) \\ &= \begin{pmatrix}\sum_{z\in\Omega}(\tilde{\Gamma}_{\operatorname{vec}_{\mathfrak{se}}(\operatorname{Pr}(A_z(E)))} + D_z(E)) & \mathbf{0}_{6\times 6} \\ \mathbf{0}_{6\times 6} & \mathbf{0}_{6\times 6}\end{pmatrix}\operatorname{vec}_{\mathfrak{g}}(\eta)\,.\end{aligned}$$

4. The Riemannian gradient of the Hamiltonian regarding the second component is at zero, thus we obtain

$$\mathbf{D}_2\mathcal{H}^-(x,\mathbf{0},t) = \big(-\operatorname{mat}_{\mathfrak{se}}(v),\mathbf{0}\big) = -f(x)\,. \tag{A.38}$$

Computation of differential regarding the first component at $\eta = (E\eta_1,\eta_2) \in T_x\mathcal{G}$ results in

$$\begin{aligned}\mathbf{D}_1(\mathbf{D}_2\mathcal{H}^-(x,\mathbf{0},t))[\eta] &= -\mathbf{D}f(x)[\eta] \\ &= -\mathbf{D}_{(E,v)}(\operatorname{mat}_{\mathfrak{se}}(v),\mathbf{0})[\eta] \\ &= -(\operatorname{mat}_{\mathfrak{se}}(\eta_2),\mathbf{0})\,.\end{aligned}$$

Finally, we compute the complete expression which is for $\eta = (\eta_1, \eta_2) \in \mathfrak{g}$ and $x^* = (E, v) \in \mathcal{G}$

$$\begin{aligned}\operatorname{vec}_{\mathfrak{g}}(&Z(x^*,t) \circ \mathbf{D}_1(\mathbf{D}_2\mathcal{H}^-)(x^*,0,t) \circ T_{\mathrm{Id}}L_{x^*}\eta) \\ &= K(t)\operatorname{vec}_{\mathfrak{g}}(\mathbf{D}_1(\mathbf{D}_2\mathcal{H}^-)(x^*,0,t)[E\eta_1,\eta_2]) \\ &= -K(t)\operatorname{vec}_{\mathfrak{g}}((\operatorname{mat}_{\mathfrak{se}}(\eta_2),\mathbf{0})) \\ &= -K(t)\begin{pmatrix}\mathbf{0}_{6\times6} & \mathbb{1}_6 \\ \mathbf{0}_{6\times6} & \mathbf{0}_{6\times6}\end{pmatrix}\operatorname{vec}_{\mathfrak{g}}(\eta)\,.\end{aligned} \tag{A.39}$$

5. The following duality holds

$$\begin{aligned}\mathbf{D}_2(\mathbf{D}_1\mathcal{H}^-(x^*,0,t)) =&(\mathbf{D}_1(\mathbf{D}_2\mathcal{H}^-(x^*,0,t)))^* \\ =&-(\mathbf{D}_{x^*}f(x^*))^*\,,\end{aligned} \tag{A.40}$$

as well as the following duality rule for linear operators $f, g : \mathfrak{g} \to \mathfrak{g}^*$ (i.e. $f^*, g^* : \mathfrak{g} \to \mathfrak{g}^*$ by the identification $\mathfrak{g}^{**} = \mathfrak{g}$) and $\eta, \xi \in \mathfrak{g}$,

$$\begin{aligned}\langle(g^* \circ f^*)(\eta),\xi\rangle_{\mathrm{Id}} &= \langle f^*(\eta), g(\xi)\rangle_{\mathrm{Id}} \\ =\langle\eta,(f\circ g)(\xi)\rangle_{\mathrm{Id}} &= \langle(f\circ g)^*(\eta),\xi\rangle_{\mathrm{Id}}\,,\end{aligned} \tag{A.41}$$

from which follows

$$(g^* \circ f^*) = (f \circ g)^*\,. \tag{A.42}$$

Note that for $\mathfrak{g} = \mathfrak{se}_3$ we replace the Riemannian metric $\langle\cdot,\cdot\rangle$ by the trace, and that the dual notation can be replaced by the transpose.

Applying the $\operatorname{vec}_{\mathfrak{g}}-$ operation for $\eta \in \mathfrak{g}$ gives

$$\begin{aligned}&\operatorname{vec}_{\mathfrak{g}}(T_{\mathrm{Id}}L^*_{x^*} \circ \mathbf{D}_2(\mathbf{D}_1\mathcal{H}^-)(x^*,0,t) \circ Z(x^*,t)\circ\eta) \\ &\overset{(A.40)}{=} -\operatorname{vec}_{\mathfrak{g}}(T_{\mathrm{Id}}L^*_{x^*} \circ (\mathbf{D}f(x^*))^* \circ Z(x^*,t)\circ\eta) \\ &\overset{(A.42)}{=} -\operatorname{vec}_{\mathfrak{g}}((\mathbf{D}f(x^*) \circ T_{\mathrm{Id}}L_{x^*})^* \circ Z(x^*,t)\circ\eta) \\ &\overset{(A.39)}{=} -\begin{pmatrix}\mathbf{0}_{6\times6} & \mathbf{0}_{6\times6} \\ \mathbb{1}_6 & \mathbf{0}_{6\times6}\end{pmatrix}\operatorname{vec}_{\mathfrak{g}}(Z(x^*,t)\circ\eta) \\ &= -\begin{pmatrix}\mathbf{0}_{6\times6} & \mathbf{0}_{6\times6} \\ \mathbb{1}_6 & \mathbf{0}_{6\times6}\end{pmatrix}K(t)\operatorname{vec}_{\mathfrak{g}}(\eta)\,.\end{aligned}$$

6. It holds for $\eta = (\eta_1, \eta_2) \in \mathfrak{g}$ and the definition of the Riemannian Hessian that

$$\operatorname{Hess}_2\mathcal{H}^-(x,\mu,t)[\eta] = \nabla_{(\eta_1,\eta_2)}\mathbf{D}_2\mathcal{H}^-(x,\mu,t)\,. \tag{A.43}$$

The Riemannian gradient of the Hamiltonian regarding the second component can be computed for $x = (E, v) \in \mathcal{G}$ as

$$\begin{aligned}&\mathbf{D}_2\mathcal{H}^-(x,\mu,t)\\ =&\big(-e^{\alpha(t-t_0)}\operatorname{mat}_{\mathfrak{se}}(S_1^{-1}\operatorname{vec}_{\mathfrak{se}}(\mu_1)) - \operatorname{mat}_{\mathfrak{se}}(v), -e^{\alpha(t-t_0)}S_2^{-1}\mu_2\big)\,. \quad \text{(A.44)}\end{aligned}$$

Inserting (A.44) into (A.43) results in

$$\begin{aligned}&e^{-\alpha(t-t_0)}\operatorname{Hess}_2\mathcal{H}^-(x,\mu,t)[\eta]\\ =&-\nabla_{(\eta_1,\eta_2)}\big(\operatorname{mat}_{\mathfrak{se}}(S_1^{-1}\operatorname{vec}_{\mathfrak{se}}(\mu_1)) + \operatorname{mat}_{\mathfrak{se}}(v), S_2^{-1}\mu_2\big)\\ =&-\operatorname{Pr}_{\mathfrak{g}}\Big(\mathbf{D}_\mu(\operatorname{mat}_{\mathfrak{se}}(S_1^{-1}\operatorname{vec}_{\mathfrak{se}}(\mu_1)) + \operatorname{mat}_{\mathfrak{se}}(v))[\eta],\\ &\qquad\qquad \mathbf{D}_\mu(S_2^{-1}\mu_2)[\eta]\Big)\\ =&-\Big(\operatorname{Pr}\big(\operatorname{mat}_{\mathfrak{se}}(S_1^{-1}\operatorname{vec}_{\mathfrak{se}}(\eta_1))\big), S_2^{-1}\eta_2\Big)\\ =&-\Big(\operatorname{mat}_{\mathfrak{se}}(S_1^{-1}\operatorname{vec}_{\mathfrak{se}}(\eta_1)), S_2^{-1}\eta_2\Big)\,,\end{aligned}$$

where $\operatorname{Pr}_{\mathfrak{g}} : \mathbb{R}^{4\times4} \times \mathbb{R}^6 \to \mathfrak{g}$ denotes the projection onto the Lie algebra $\mathfrak{g}$. Note that the second component of the projection is trivial.

This result coincides with [72] where the Hessian of the Hamiltonian regarding the second component is computed directly. Applying the $\operatorname{vec}_{\mathfrak{g}}$ −operation leads to

$$\begin{aligned}&\operatorname{vec}_{\mathfrak{g}}(\operatorname{Hess}_2\mathcal{H}^-(x,\mu,t)[T_{\mathrm{Id}}L_x\eta])\\ &\quad=-e^{\alpha(t-t_0)}\operatorname{vec}_{\mathfrak{g}}\Big(\operatorname{mat}_{\mathfrak{se}}(S_1^{-1}\operatorname{vec}_{\mathfrak{se}}(\eta_1)), S_2^{-1}\eta_2\Big)\\ &\quad=-e^{\alpha(t-t_0)}((S_1^{-1}\operatorname{vec}_{\mathfrak{se}}(\eta_1))^\top, (S_2^{-1}\eta_2)^\top)^\top\\ &\quad=-e^{\alpha(t-t_0)}\underbrace{\begin{pmatrix}S_1^{-1} & \mathbf{0}_{6\times6}\\ \mathbf{0}_{6\times6} & S_2^{-1}\end{pmatrix}}_{=:S^{-1}}\operatorname{vec}_{\mathfrak{g}}(\eta)\,.\end{aligned}$$

Now we apply the $\operatorname{vec}_{\mathfrak{g}}$-operation to the expression $Z(x^*,t) \circ \operatorname{Hess}_2\mathcal{H}^-(x^*,0,t) \circ Z(x^*,t)$:

$$\begin{aligned}&\operatorname{vec}_{\mathfrak{g}}\Big(Z(x^*,t)\circ\operatorname{Hess}_2\mathcal{H}^-(x^*,0,t)[Z(x^*,t)(\eta)]\Big)\\ &\quad=K(t)\operatorname{vec}_{\mathfrak{g}}\Big(\operatorname{Hess}_2\mathcal{H}^-(x^*,0,t)[Z(x^*,t)(\eta)]\Big)\\ &\quad=-e^{\alpha(t-t_0)}K(t)S^{-1}\operatorname{vec}_{\mathfrak{g}}(Z(x^*,t)(\eta))\\ &\quad=-e^{\alpha(t-t_0)}K(t)S^{-1}K(t)\operatorname{vec}_{\mathfrak{g}}(\eta)\,.\end{aligned}$$

□

A.4 Derivations for Extended Kálmán Filter

The function $\Phi : \mathbb{R}^{12} \to \mathbb{R}^{12\times 12}$ in Alg. 1 is

$$\Phi(v) = \begin{pmatrix} \Phi_{\mathrm{SE}_3}(v_{1:6}) & \mathbf{0}_{6\times 6} \\ \mathbf{0}_{6\times 6} & \mathbb{1}_6 \end{pmatrix},$$

whereas the function Φ_{SE_3} is given in [76, section 10] (cf. [17, Eq. (17)]).

A.4.1 Derivations for Non-Linear Observations

The expression of H_l that is defined in [17, Eq. (59)] is simply the Riemannian gradient of the observation function h_z, i.e.

$$H_l := \sum_{z\in\Omega} \mathbf{D}h_z(G(t_l)),$$

where h_k is defined as in (A.21); and the $\mathbf{D}h_z$ can be computed componentwise (for $j = 1, 2$) for $x(t_l) = (E(t_l), v(t_l))$ by the directional derivative for a direction $x\eta \in T_x\mathcal{G}$.

$$\begin{aligned}
\mathbf{D}h_z^j(x)[x\eta] &= \mathbf{D}\big((e_3^4 E^{-1} g_z)^{-1} e_j^4 E^{-1} g_z\big)[(E\eta_1, \eta_2)] && \text{(A.45)}\\
&= \kappa_z^{-2} e_3^4 \eta_1 E^{-1} g_z e_j^4 E^{-1} g_z - \kappa_z^{-1} e_j^4 \eta_1 E^{-1} g_z && \text{(A.46)}\\
&= \langle \big(\kappa_z^{-2} E^{-1} g_z e_j^4 E^{-1} g_z e_3^4 - \kappa_z^{-1} E^{-1} g_z e_j^4\big)^\top, \eta_1 \rangle && \text{(A.47)}\\
&=: \langle \rho_k^j(x), \eta_1 \rangle, && \text{(A.48)}
\end{aligned}$$

where the second last line follows from the definition of the Riemannian metric on SE_3, i.e. $\langle \eta, \xi \rangle_{\mathrm{Id}} = \eta^\top \xi$, and the fact that the trace is cyclic. By projection of $\rho_k^1(x(t_l))$ onto the Lie algebra $\mathfrak{se}_3$ and by vectorization, we obtain the Riemannian gradient. Stacking the vectors leads to the Jacobian $H_l \in \mathbb{R}^{2\times 12}$, which is provided through

$$H_l = \sum_{k=1}^{l} \begin{pmatrix} \mathrm{vec}_{\mathfrak{se}}(\mathrm{Pr}(\rho_k^1(t_l)))^\top & \mathbf{0}_{1\times 6} \\ \mathrm{vec}_{\mathfrak{se}}(\mathrm{Pr}(\rho_k^2(t_l)))^\top & \mathbf{0}_{1\times 6} \end{pmatrix}. \tag{A.49}$$

Next, we consider the calculation of the function $J(t)$ in Alg. 1 in line 3. Following [17], $J(t)$ can be calculated as

$$J(t) = F(t) - \mathrm{ad}_{\mathfrak{g}}(f(x(t))) + \tfrac{1}{12} C(S), \tag{A.50}$$

where the differential of $F(t) = \mathbf{D}f(x(t))$ can be computed as

$$F(t) = \begin{pmatrix} \mathbf{0}_{6\times 6} & \mathbb{1}_6 \\ \mathbf{0}_{6\times 6} & \mathbf{0}_{6\times 6} \end{pmatrix}. \tag{A.51}$$

For a diagonal weighting matrix S, we find that in (A.50) the function C can be computed for diagonal weighting matrices S as

$$C(S) = \begin{pmatrix} \begin{pmatrix} \Xi & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \Xi \end{pmatrix} & \mathbf{0}_{6\times6} \\ \mathbf{0}_{6\times6} & \mathbf{0}_{6\times6} \end{pmatrix}, \tag{A.52}$$

where $\Xi = -\operatorname{diag}((S_{22}+S_{33}, S_{11}+S_{33}, S_{11}+S_{22})^\top)$, and the adjoint in (A.50) can be computed with (A.11).

A.5 Supplemental Material of Chapter 5

A.5.1 Derivation of the Piecewise Affine Model

We assume to have a given depth map $d = (d(z), z \in \Omega)$ that we want to propagate in the future corresponding a given camera motion $E(t) = (R(t), w(t)) \in \mathrm{SE}_3$. With the depth map we can assign to each image point $z \in \Omega$ on the image grid a space point $Z(z, d(z)) = \binom{z}{1} d(z)$. We suppose that the scene structure around $Z(z, d(z))$ is piecewise planar and can be described by vector $p(z) \in \mathbb{R}^3$ such that the surface around Z is the set

$$\{Z \in \mathbb{R}^3 | \langle Z, p(z) \rangle = 1\}, \tag{A.53}$$

To determine the vector p we choose the constraints

$$\begin{aligned} \left\langle d(z) \binom{z}{1}, p(z) \right\rangle &= 1 \\ \nabla_{z_1,z_2} \left\langle d(z) \binom{z}{1}, p(z) \right\rangle &= \binom{0}{0}. \end{aligned}$$

Given a known depth $d(z)$ and a gradient $\nabla_{z_1,z_2} d(z)$ the solution can be computed as

$$p(z) = d^{-2}(z) \begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ z_1 & z_2 & 1 \end{pmatrix} \begin{pmatrix} \nabla_z d(z) \\ d(z) \end{pmatrix}. \tag{A.54}$$

When we move the camera by the rotation $R \in \mathrm{SO}_3$ and translation $w \in \mathbb{R}^3$ the induced viewing ray $\{R\lambda \binom{z}{1} + w | \lambda > 0\}$ intersects the plane at

$$\lambda = \frac{1 - p(z)^\top w}{p(z)^\top R \binom{z}{1}}, \tag{A.55}$$

that leads to a new space point (from the point of view of the moved camera)

$$\hat{Z}(z, R, w) := \lambda \binom{z}{1} = \frac{1 - p(z)^\top w}{p(z)^\top R \binom{z}{1}} \binom{z}{1} \tag{A.56}$$

The corresponding depth $\tilde{d}$ in z for an arbitrary camera motion $E = (R, w)$ is just the third component of $\hat{Z}(z, R, w)$, i.e.

$$\hat{d}(z, R, w) = \left(\begin{smallmatrix} 0 & 0 & 1 \end{smallmatrix}\right) \hat{Z}(z, R, w) = \frac{1 - p(z)^\top w}{p(z)^\top R \left(\begin{smallmatrix} z \\ 1 \end{smallmatrix}\right)} = \lambda \,. \tag{A.57}$$

For the choice $(R, w) = (\mathbb{1}, \mathbf{0})$ we find that $\hat{d}(z, \mathbb{1}, \mathbf{0}) = d(z)$. This means that $\hat{d}$ is accurate next to a constant camera motion, that can be described by $\eta \in \mathfrak{se}_3$, such that

$$E(t) = \begin{pmatrix} R(t) & w(t) \\ \mathbf{0} & 1 \end{pmatrix} = \mathrm{Exp}(t\eta) \in \mathrm{SE}_3 \ .$$

We want to compute the total time derivative $\hat{d}(z, t) := \hat{d}(z, R(t), w(t))$ at $t = 0$ which is given through

$$\frac{d}{dt} \hat{d}(z, t)\big|_{t=1} = \lim_{t \to 0} t^{-1} \big(\hat{d}(z, t) - \hat{d}(z, 0) \big) \tag{A.58}$$

by using the matrix differential $\frac{d}{dt} E(t)|_{t=0} = \eta$. Evaluating (A.58) gives for $\eta_1 = \eta_{1:3,1:3}$ and $\eta_2 = \eta_{1:3,4}$

$$\frac{d}{dt} \hat{d}(z, t)\Big|_{t=0} = - \frac{p(z)^\top \eta_2 p(z)^\top \left(\begin{smallmatrix} z \\ 1 \end{smallmatrix}\right) + p(z)^\top \eta_2 \left(\begin{smallmatrix} z \\ 1 \end{smallmatrix}\right)}{(p(z)^\top \left(\begin{smallmatrix} z \\ 1 \end{smallmatrix}\right))^2}, \tag{A.59}$$

where $p(z)$ was determined on (A.54) and depends on $d(z)$ and $\nabla d(z)$. Thus, substituting (A.54) finally gives

$$\dot{d}(z, t) = - \left(d(z, t)^{-1} \eta_2 + \eta_1 \left(\begin{smallmatrix} z \\ 1 \end{smallmatrix}\right) \right)^\top \begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ z_1 & z_2 & 1 \end{pmatrix} \begin{pmatrix} \nabla d(z,t) \\ d(z,1) \end{pmatrix} \tag{A.60}$$

$$= - \left(\eta \begin{pmatrix} z \\ 1 \\ d(z,t)^{-1} \end{pmatrix} \right)^\top \begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ z_1 & z_2 & 1 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \nabla d(z,t) \\ d(z,1) \end{pmatrix}, \tag{A.61}$$

where $\eta = \left(\begin{smallmatrix} \eta_1 & \eta_2 \\ \mathbf{0} & 0 \end{smallmatrix}\right)$ and $z = \left(\begin{smallmatrix} z_1 \\ z_2 \end{smallmatrix}\right)$. In this work we use the inverse depth representation. Therefore we replace $d_i(z) = (d(z))^{-1}$. With the formula

$$\frac{d}{dt} d_i(z, t) = -d(z, t)^{-2} \dot{d}(z, t) \,, \tag{A.62}$$

we find the differential equation

$$\begin{aligned} \frac{d}{dt} d_i(z, t) =& d_i(z, t)^2 \left(\eta \begin{pmatrix} z \\ 1 \\ d_i(z,t) \end{pmatrix} \right)^\top \begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ z_1 & z_2 & 1 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \nabla (d_i(z,t))^{-1} \\ (d_i(z,t))^{-1} \end{pmatrix} \\ =& - \left(\eta \begin{pmatrix} z \\ 1 \\ d_i(z,t) \end{pmatrix} \right)^\top \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -z_1 & -z_2 & 1 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \nabla d_i(z,t) \\ d_i(z,t) \end{pmatrix} . \end{aligned}$$

A.5.2 Vectorization of Adjoint Representation

For the considerations below we require to introduce the vectorization of the adjoint operator on the Lie group $\mathcal{G} = \mathrm{SE}_3 \times \mathbb{R}^6 \times (0,1)^{n_\Omega}$ which is given for a vector $v \in \mathbb{R}^{12+n_\Omega}$ by the expression

$$\mathrm{ad}_{\mathfrak{g}}^{\mathrm{vec}}(\mathrm{mat}_{\mathfrak{g}}(v)) = \begin{pmatrix} \mathrm{ad}_{\mathfrak{se}}^{\mathrm{vec}}(v_{1:6}) & \mathbf{0}_{6\times 6} & \mathbf{0}_{6\times|\Omega|} \\ \mathbf{0}_{6\times 6} & \mathbf{0}_{6\times 6} & \mathbf{0}_{6\times|\Omega|} \\ \mathbf{0}_{|\Omega|\times 6} & \mathbf{0}_{|\Omega|\times 6} & \mathbf{0}_{|\Omega|\times|\Omega|} \end{pmatrix}. \tag{A.63}$$

Here $\mathrm{ad}_{\mathfrak{se}}^{\mathrm{vec}}$ denotes the operator defined in (A.7).

A.5.3 Proofs

Proof of Lemma 5.2.6. The proof requires to compute the total time derivative of the necessary condition for the optimal state x^* which is given through

$$\mathbf{D}_1 \mathcal{V}(x^*, t) = \mathbf{0}. \tag{A.64}$$

The calculation of the time derivative of (A.64) is already given in [72, Eq. (26)–(37)] and results in the following evolution equation:

$$(x^*(t))^{-1}\dot{x}^*(t) = -\mathbf{D}_2\mathcal{H}(x^*(t), \mathbf{0}, t) - Z(x^*(t), t)^{-1} \circ (x^*)^{-1}(\mathbf{D}_1\mathcal{H}(x^*(t), \mathbf{0}, t)). \tag{A.65}$$

The derivative of the Hamiltonian regarding the second component is simply

$$\mathbf{D}_2\mathcal{H}(x^*, \mathbf{0}, t) = -f(x^*)\,, \tag{A.66}$$

such that the evolution equation for the optimal state x^* reads

$$(x^*(t))^{-1}\dot{x}^*(t) = f(x^*) - Z(x^*(t), t)^{-1} \circ (x^*)^{-1}(\mathbf{D}_1\mathcal{H}(x^*(t), \mathbf{0}, t)), \tag{A.67}$$

which is Lemma 5.2.6.
The calculation of the differential of the Hamiltonian

$$\mathbf{D}_1\mathcal{H}(x^*(t), \mathbf{0}, t)$$

is a bit involved but can be calculated component-wise. We will use the shorthands $I := \left(\begin{smallmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{smallmatrix}\right) \in \mathbb{R}^{2\times 4}$, $e = \left(\begin{smallmatrix} 0 & 0 & 1 & 0 \end{smallmatrix}\right) \in \mathbb{R}^{1\times 4}$, $g_z = \begin{pmatrix} \left(\begin{smallmatrix} z \\ 1 \end{smallmatrix}\right)(d_i(z,t))^{-1} \\ 1 \end{pmatrix} \in \mathbb{R}^4$ and $\kappa_z = eEg_z \in \mathbb{R}$. With these expressions we can write the function h as $h(z,t) := \kappa_z^{-1} I E g_z$.

We begin with the directional derivative of the Hamiltonian $\mathcal{H}(x, \mathbf{0}, t)$ regarding the camera motion E for $x = (E, v, d_i)$ in a specific direction $E\eta \in T_E \operatorname{SE}_3$ which is

$$\begin{aligned}
\mathbf{D}_E\mathcal{H}(x,\mathbf{0},t)[E\eta] &= \sum_z \mathbf{D}_E\phi(\tfrac{1}{2}\|y_z - h_z(x,t)\|^2_{Q_z})[E\eta] \\
&= \sum_z \beta(\tfrac{1}{2}\|y_z - h_z(x,t)\|^2_{Q_z} + \nu)^{\beta-1}(y_z - h_z(x,t))^\top Q_z(-1)\mathbf{D}_E h_z(x,t)[E\eta] \\
&= \sum_z \beta(\cdots)^{\beta-1}(y_z - h_z(x,t))^\top Q_z(-1)\mathbf{D}_E\big(eEg_zIEg_z\big)[E\eta] \\
&= \sum_z \beta(\cdots)^{\beta-1}(-1)\operatorname{tr}\Big((y_z - h_z(x,t))^\top Q_z(\kappa_z^{-1}IE\eta g_z - \kappa_z^{-2}eE\eta g_zIEg_z)\Big) \\
&= \sum_z \beta(\cdots)^{\beta-1}(-1)\operatorname{tr}\Big((y_z - h_z(x,t))^\top Q_z\big(\kappa_z^{-1}I - \kappa_z^{-2}IEg_ze\big)E\eta g_z\Big) \\
&= \sum_z \beta(\cdots)^{\beta-1}(-1)\Big\langle g_z(y_z - h_z(x,t))^\top Q_z\big(\kappa_z^{-1}I - \kappa_z^{-2}IEg_ze\big)E\eta\Big\rangle \\
&= \sum_z \beta(\cdots)^{\beta-1}\Big\langle \big(g_z(y_z - h_z(x,t))^\top Q_z\big(\kappa_z^{-2}IEg_z(e_3^4)^\top - \kappa_z^{-1}I\big)E\big)^\top, \eta\Big\rangle .
\end{aligned}$$

From the definition of the Riemannian gradient on SE_3 follows that it can be computed by the orthogonal projection $\operatorname{Pr} : \mathbb{R}^{4\times4} \to \mathfrak{se}_3$ which reads

$$A \mapsto \tfrac{1}{2}\begin{pmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&0\end{pmatrix}\left(A\begin{pmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&2\end{pmatrix} - A^\top\begin{pmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&0\end{pmatrix}\right). \tag{A.68}$$

The resulting Riemannian gradient of the Hamiltonian regarding E is

$$\begin{aligned}
\mathbf{D}_E\mathcal{H}(x,0,t) &= \sum_z \beta(\cdots)^{\beta-1}E\operatorname{Pr}\Big(\big(g_z(y_z - h_z(x,t))^\top Q_z\big(\kappa_z^{-2}IEg_ze - \kappa_z^{-1}I\big)E\big)^\top\Big) \\
&=: EG_E(x) \in T_E\operatorname{SE}_3 .
\end{aligned} \tag{A.69}$$

The gradient of the Hamiltonian can be calculated component-wise, i.e. for each z in the image domain Ω separately. We will use the shorthand $g_z' := -\begin{pmatrix}\binom{z}{1}(d_i(z,t))^{-2}\\0\end{pmatrix}$ for the partial derivative $\frac{\partial}{\partial d_i(z,t)}g_z$.
Then the components of $\mathbf{D}_{d_i}\mathcal{H}(x,\mathbf{0},t)$ read

$$\begin{aligned}
\frac{\partial}{\partial d_i(z,t)}\mathcal{H}(x,\mathbf{0},t) &= \frac{\partial}{\partial d_i(z,t)}\sum_{z\in\Omega}\phi(\tfrac{1}{2}\|y_z - h_z(x,t)\|^2_{Q_z}) \\
&= \frac{\partial}{\partial d_i(z,t)}\big(\tfrac{1}{2}\|y_z - h_z(x,t)\|^2_{Q_z} + \nu\big)^\beta + \nu^\beta
\end{aligned}$$

$$
\begin{aligned}
&= \beta(\cdots)^{\beta-1}(y_z - h_z(x,t))^\top Q_z(-1)\frac{\partial}{\partial d_i(z,t)}h_z(x,t)\\
&= \beta(\cdots)^{\beta-1}(y_z - h_z(x,t))^\top Q_z(-1)\big(-\kappa_z^{-2}eEg_z'IEg_z + \kappa_z^{-1}IEg_z'\big)\\
&= \beta(\cdots)^{\beta-1}(y_z - h_z(x,t))^\top Q_z\big(\kappa_z^{-2}eEg_z'IEg_z - \kappa_z^{-1}IEg_z'\big)\\
&=: (G_{d_i}(x))_{p(z)} \in \mathbb{R}\,, \qquad \text{(A.70)}
\end{aligned}
$$

where $p(z)$ denotes the index of the pixel z after stacking the image domain Ω column-wise. Thus, by combining all these entries to a vector (given a fixed ordering of $z \in \Omega$, e.g. column-wise), we obtain the expression $G_{d_i}(x)$. Since the Hamiltonian does not depend on the vector v, the corresponding entries are zero such that we finally obtain the gradient of the Hamiltonian regarding $x = (E, v, d_i)$ which is

$$
\mathbf{D}_1\mathcal{H}(x, \mathbf{0}, t) = T_{\mathrm{Id}}L_x(G_E(x), \mathbf{0}_6, G_{d_i}(x)) \in T_x\mathcal{G}. \qquad \text{(A.71)}
$$

□

Proof of Lemma 5.2.7. Following Saccon et al. [72, Eq. (51)] the evolution equation of the operator $Z(x^*, t) : \mathfrak{g} \to \mathfrak{g}^*$ is given through

$$
\begin{aligned}
&\frac{d}{dt}Z(x^*,t) && \text{(A.72a)}\\
&\approx Z(x^*,t)\circ\omega_{(x^*)^{-1}\dot{x^*}} && \text{(A.72b)}\\
&\quad + Z(x^*,t)\circ\omega_{\overleftrightarrow{\mathbf{D}_2\mathcal{H}}(x^*,0,t)} && \text{(A.72c)}\\
&\quad + \omega^*_{(x^*)^{-1}\dot{x^*}}\circ Z(x^*,t) && \text{(A.72d)}\\
&\quad + \omega^*_{\overleftrightarrow{\mathbf{D}_2\mathcal{H}}(x^*,0,t)}\circ Z(x^*,t) && \text{(A.72e)}\\
&\quad + T_{\mathrm{Id}}L^*_{x^*}\circ \mathrm{Hess}_1\,\mathcal{H}(x^*,0,t)\circ T_{\mathrm{Id}}L_{x^*} && \text{(A.72f)}\\
&\quad + T_{\mathrm{Id}}L^*_{x^*}\circ \mathbf{D}_2(\mathbf{D}_1\mathcal{H})(x^*,0,t)\circ Z(x^*,t) && \text{(A.72g)}\\
&\quad + Z(x^*,t)\circ \mathbf{D}_1(\mathbf{D}_2\mathcal{H})(x^*,0,t)\circ T_{\mathrm{Id}}L_{x^*} && \text{(A.72h)}\\
&\quad + Z(x^*,t)\circ \mathrm{Hess}_2\,\mathcal{H}(x^*,0,t)\circ Z(x^*,t)\,. && \text{(A.72i)}
\end{aligned}
$$

Since $Z \in \mathfrak{g}^*$ and $\mathfrak{g}^*$ is a vector space we can represent $Z(x^*, t)$ as a matrix $K(t) \in \mathbb{R}^{(|\Omega|+12)\times(|\Omega|+12)}$ by evaluating Z for a specific $\eta \in \mathfrak{g}$ and vectorization, i.e.

$$
\mathrm{vec}_{\mathfrak{g}}(Z(x^*,t)(\eta)) = K(t)\,\mathrm{vec}_{\mathfrak{g}}(\eta)\,. \qquad \text{(A.73)}
$$

Similarly we can vectorize the full differential equation of Z to find the evolution equation of the operator $P(t)$ in Lemma 5.2.7.
With (A.73) it is obvious to see that the expression in (A.72a) can be vectorized as

$$
\mathrm{vec}_{\mathfrak{g}}\Big(\frac{d}{dt}Z(x^*,t)(\eta)\Big) = \dot{K}(t)\,\mathrm{vec}_{\mathfrak{g}}(\eta)\,. \qquad \text{(A.74)}
$$

Second, we consider the expressions in (A.72b) and (A.72c).

$$\begin{aligned}
&\operatorname{vec}_{\mathfrak{g}}(Z(x^*,t)\circ\omega_{(x^*)^{-1}\dot{x^*}}\eta + Z(x^*,t)\circ\omega^{\rightleftharpoons}_{\mathbf{D}_2\mathcal{H}(x^*,0,t)}\eta)\\
&\overset{(A.73)}{=} K(t)\operatorname{vec}_{\mathfrak{g}}(\omega_{(x^*)^{-1}\dot{x^*}}\eta + \omega^{\rightleftharpoons}_{\mathbf{D}_2\mathcal{H}(x^*,0,t)}\eta)\\
&\overset{(A.67)}{=} K(t)\operatorname{vec}_{\mathfrak{g}}(\nabla_{f(x^*)}\eta - \nabla_{Z(x^*,t)^{-1}\circ(x^*)^{-1}\mathbf{D}_1\mathcal{H}(x^*,\mathbf{0},t)\eta}\\
&\qquad + \nabla_\eta \mathbf{D}_2\mathcal{H}(x^*,0,t))\\
&\overset{(A.66)}{=} K(t)\operatorname{vec}_{\mathfrak{g}}(\nabla_{f(x^*)}\eta - \nabla_{Z(x^*,t)^{-1}\circ(x^*)^{-1}\mathbf{D}_1\mathcal{H}(x^*,\mathbf{0},t)\eta}\\
&\qquad - \nabla_\eta f(x^*))\\
&\overset{(A.28)}{=} K(t)\operatorname{vec}_{\mathfrak{g}}([f(x^*),\eta] - \nabla_{Z(x^*,t)^{-1}\circ(x^*)^{-1}\mathbf{D}_1\mathcal{H}(x^*,\mathbf{0},t)\eta})\\
&\overset{(A.17)}{=} K(t)\big(\operatorname{vec}_{\mathfrak{g}}([f(x^*),\eta]) - \tilde{\Gamma}^*_{\operatorname{vec}_{\mathfrak{g}}(Z(x^*,t)^{-1}\circ(x^*)^{-1}\mathbf{D}_1\mathcal{H}(x^*,\mathbf{0},t))}\operatorname{vec}_{\mathfrak{g}}(\eta)\big)\\
&\overset{(A.73)}{=} K(t)\big(\operatorname{vec}_{\mathfrak{g}}([f(x^*),\eta]) - \tilde{\Gamma}^*_{K(t)^{-1}\operatorname{vec}_{\mathfrak{g}}((\dot{x})^{-1}\mathbf{D}_1\mathcal{H}(x^*,\mathbf{0},t))}\operatorname{vec}_{\mathfrak{g}}(\eta)\big)\\
&\overset{(A.63)}{=} K(t)\big(\operatorname{ad}^{\operatorname{vec}}_{\mathfrak{g}}(f(x^*)) - \tilde{\Gamma}^*_{K(t)^{-1}\operatorname{vec}_{\mathfrak{g}}((\dot{x})^{-1}\mathbf{D}_1\mathcal{H}(x^*,\mathbf{0},t))}\big)\operatorname{vec}_{\mathfrak{g}}(\eta)\\
&\quad =: K(t)C_1(x^*,t)\operatorname{vec}_{\mathfrak{g}}(\eta)\,.
\end{aligned} \tag{A.75}$$

By duality we find that the lines (A.72d) and (A.72e) can be represented as

$$\operatorname{vec}_{\mathfrak{g}}\big(\omega^*_{(x^*)^{-1}\dot{x^*}}\circ Z(x^*,t)(\eta) + \omega^{\rightleftharpoons *}_{\mathbf{D}_2\mathcal{H}(x^*,0,t)}\circ Z(x^*,t)(\eta)\big) = C_1(x^*,t)^\top K(t)\eta. \tag{A.76}$$

The vectorization of (A.72h) can be simply achieved as

$$\begin{aligned}
&\operatorname{vec}_{\mathfrak{g}}\big(Z(x^*,t)\circ\mathbf{D}_1(\mathbf{D}_2\mathcal{H})(x^*,0,t)\circ T_{\mathrm{Id}}L_{x^*}\eta\big)\\
&\overset{(A.73)}{=} K(t)\operatorname{vec}_{\mathfrak{g}}\big(\mathbf{D}_1(\mathbf{D}_2\mathcal{H})(x^*,0,t)\circ x^*\eta\big)\\
&\quad = K(t)\operatorname{vec}_{\mathfrak{g}}\Big(-\mathbf{D}_1 f(x^*)[x^*\eta]\big)\\
&\quad = -K(t)\operatorname{vec}_{\mathfrak{g}}(\operatorname{mat}_{\mathfrak{se}}((\eta_2,\mathbf{0}_6,\mathbf{0}_{|\Omega|})))\\
&\quad = -K(t)\begin{pmatrix}\mathbf{0}_{6\times 6} & \mathbb{1}_6 & \mathbf{0}_{6\times|\Omega|}\\ \mathbf{0}_{6\times 6} & \mathbf{0}_6 & \mathbf{0}_{6\times|\Omega|}\\ \mathbf{0}_{|\Omega|\times 6} & \mathbf{0}_{|\Omega|\times 6} & \mathbf{0}_{|\Omega|\times|\Omega|}\end{pmatrix}\operatorname{vec}_{\mathfrak{g}}(\eta)\\
&\quad =: -K(t)C_2(x^*,t)\operatorname{vec}_{\mathfrak{g}}(\eta)\,,
\end{aligned}$$

where f is the function in (5.17). Again, by duality follows

$$\operatorname{vec}_{\mathfrak{g}}\big(T_{\mathrm{Id}}L^*_{x^*}\circ\mathbf{D}_2(\mathbf{D}_1\mathcal{H})(x^*,0,t)\circ Z(x^*,t)\circ\eta\big) = C_2(x^*,t)^\top K(t)\operatorname{vec}_{\mathfrak{g}}(\eta) \tag{A.77}$$

for the expression in (A.72g). Setting $C(x^*,t) := C_2(x^*,t) - C_1(x^*,t)$ gives the matrix $C(x^*,t)$ in Lemma 5.2.7.

The expression in (A.72i) can be calculated as

$$\operatorname{vec}_{\mathfrak{g}}\big(Z(x^*,t)\circ \operatorname{Hess}_2 \mathcal{H}(x^*,0,t)\circ Z(x^*,t)\circ \eta\big) \tag{A.78}$$

$$\overset{(A.73)}{=} K(t)\operatorname{vec}_{\mathfrak{g}}\big(\operatorname{Hess}_2 \mathcal{H}(x^*,0,t)\circ Z(x^*,t)\big) \tag{A.79}$$

$$= -K(t)RK(t)\operatorname{vec}_{\mathfrak{se}}(\eta)\,, \tag{A.80}$$

where R is the weighting matrix in the energy function in (**??**).
It remains to calculate the matrix representation of the Hessian of the Hamiltonian in (A.72f), i.e.

$$\operatorname{vec}_{\mathfrak{g}}\big(T_{\mathrm{Id}}L^*_{x^*}\circ \operatorname{Hess}_1 \mathcal{H}(x^*,0,t)\circ T_{\mathrm{Id}}L_{x^*}\circ \eta\big) =: H(x^*,t)\operatorname{vec}_{\mathfrak{g}}(\eta). \tag{A.81}$$

The single blocks of the Hessian of the Hamiltonian can be calculated again separately, i.e.

$$H = \begin{pmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & H_{33} \end{pmatrix}.$$

Note that the entries that address the variable v are all zero since the Hamiltonian does not depend on v for $x = (E, v, d_i)$. Thus, the entries $H_{12}, H_{13}, H_{21}, H_{22}, H_{31}$ are all zero. As we consider the Riemannian Hessian regarding the symmetric Levi-Civita connection, it is sufficient to calculate H_{11}, H_{33} and $H_{13} = H_{31}^\top$. The matrix H_{33} is a diagonal matrix containing the partial derivatives

$$\frac{\partial}{\partial d_i(z,t)}(G_{d_i}(x))_z\,, \tag{A.82}$$

where $(G_{d_i}(x))_z$ was calculated in (A.70). The columns of H_{13} can be obtained similarly by calculation of the partial derivatives of the vector representation of the gradient $G_E(x^*)$ in (A.69), i.e.

$$(H_{13})_{p(z)\bullet} = \frac{\partial}{\partial d_i(z)}\operatorname{vec}_{\mathfrak{se}}(\Pr(G_E(x^*)))\,, \tag{A.83}$$

where $\bullet$ denotes the full column, and $p(z)$ denotes the position of z in a fixed ordering (e.g. column-wise). The matrix H_{11} is a bit more complicated because the *Christoffel symbols* on SE_3 are not zero. It can be calculated by the general definition of the Riemannian Hessian (cf. [1, Def. 5.5.1]). For $\eta \in \mathrm{SE}_3$ we find the following equalities

$$\begin{aligned} &\operatorname{vec}_{\mathfrak{se}}(E^{-1}\operatorname{Hess}_E \mathcal{H}(x^*,\mathbf{0},t)[E\eta]) \\ &\quad\overset{\text{Def.}}{=} \operatorname{vec}_{\mathfrak{se}}(E^{-1}\nabla_{E\eta}\mathbf{D}_E\mathcal{H}(x^*,\mathbf{0},t)) \\ &\quad\overset{\text{Lin.}}{=} \operatorname{vec}_{\mathfrak{se}}(\nabla_{\eta}E^{-1}\mathbf{D}_E\mathcal{H}(x^*,\mathbf{0},t)) \\ &\quad\overset{(A.69)}{=} \operatorname{vec}_{\mathfrak{se}}(\nabla_{\eta}G_E(x^*))\,. \end{aligned}$$

The last equation can be evaluated by using the formula in (A.18). For brevity we omit the full calculations.
Thus, evaluation of the differential equation of Z for a $\eta \in \mathfrak{g}$ and vectorization with the operator $\mathrm{vec}_{\mathfrak{g}}$ gives us the following equation for K :

$$\dot{K}(t) = -K(t)RK(t) - K(t)C(x^*,t) - C(x^*,t)^\top K(t) + H(x^*,t). \quad \text{(A.84)}$$

We want to avoid the inversion of the operator K in (A.65). Therefore we replace $P(t) := K(t)^{-1}$ in (A.84). By using the well-known rule for the calculation of the differential of an inverse matrix we obtain finally the expression in Lemma 5.2.7.

$$\begin{aligned}\dot{P}(t) =& \frac{d}{dt}K(t)^{-1} \\ =& -K(t)^{-1}\dot{K}(t)K(t)^{-1} \\ =& -P(t)(-1)K(t)RK(t)P(t) - (-1)P(t)K(t)C(x^*,t)P(t) \\ & \quad - (-1)P(t)C(x^*,t)^\top K(t)P(t) - P(t)H(x^*,t)P(t) \\ =& R + C(x^*,t)P(t) + P(t)C(x^*,t)^\top - P(t)H(x^*,t)P(t)\end{aligned} \quad \text{(A.85)}$$

□

Proof of Theorem 5.2.8. By replacing the expression

$$Z(x^*,t)^{-1} \circ \eta = \mathrm{mat}_{\mathfrak{g}}(P(t)\,\mathrm{vec}_{\mathfrak{g}}(G(x^*,t))$$

in (A.65) we obtain the equation (15) in the paper where

$$G(x^*,t) := (G_E(x^*), \mathbf{0}_6, G_{d_i}(x^*))$$

denotes the gradient of the Hamiltonian as calculated in (A.71). The initial condition of equation (15) can be found by minimizing the value function (9) for $t = t_0$. Then we find that the optimal initial state is $x^*(t_0) = x_0$. The differential equation for P was already calculated in Lemma 5.2.7 in (A.85) and the initial state of P is the inverse of the Hessian of the value function (9) at $t = t_0$ which is R_0. This completes the proof. □

A.5.4 Calculations for the Filtering Problem with Spatial Regularization

In this section we derive the minimum energy filter for joint camera motion and disparity estimation which was discussed in section 5.3. We recall the

(left-trivialized) Hamiltonian (5.33) which includes the spatial regularizer. It is given through

$$\begin{aligned}\mathcal{H}(x,\mu,t) := &-\langle \mu, f(x(t)\rangle - \tfrac{1}{2}\|\mathrm{vec}_{\mathfrak{g}}(\mu)\|^2_{R^{-1}} \\ &+ \Big(\sum_{z\in\Omega}\big(\phi_{\mathrm{data}}\big(\tfrac{1}{2}\|y_z(t) - h_z(x(t))\|^2_{Q_z}\big) + \lambda\phi_{\mathrm{reg}}\big(\tfrac{1}{2}\|\nabla_z d_i(z)\|^2\big)\big)\Big)\,. \end{aligned} \tag{A.86}$$

Here, $x = (E, v, d_i) \in \mathcal{G}$ consists of the camera motion $E \in \mathrm{SE}_3$, $v \in \mathbb{R}^6$ and the disparity map $d_i \in (0,1)^{|\Omega|}$.
Computation of the total time derivative of the necessary condition (5.24) again leads to a differential equation for the optimal state (cf. (A.87)).

$$\dot{x}^*(t) = x(t)\Big(-\mathbf{D}_2\mathcal{H}(x^*(t),\mathbf{0},t) - Z(x^*(t),t)^{-1}\circ x^{-1}(\mathbf{D}_1\mathcal{H}(x^*(t),\mathbf{0},t))\Big) \tag{A.87}$$

where $Z : \mathfrak{g} \to \mathfrak{g}^*$ is the left-trivialized Hessian of the value function given through

$$Z(x,t)\circ\eta = x^{-1}\,\mathrm{Hess}\,\mathcal{V}(x(t),t,x(t_0))[x\eta], \quad \eta \in \mathfrak{g}\,. \tag{A.88}$$

The differential of the Hamiltonian regarding the second component is $\mathbf{D}_2\mathcal{H}(x^*,\mathbf{0},t) = -f(x^*)$. We continue with the differential of the Hamiltonian regarding the first component. As the differential of the data term was already derived in (A.71), we only calculate the differential of the regularizer, which we denote for $x = (E, v, d_i)$ by

$$\Phi(x) := \lambda\sum_{z\in\Omega}\phi_{\mathrm{reg}}\big(\tfrac{1}{2}\|\nabla_z d_i(z)\|^2\big)\,. \tag{A.89}$$

The image domain Ω is discrete, therefore we require a discretization of the spatial gradient $\nabla_z d_i(z)$. For this reason we introduce difference matrices which can be expressed with help of Kronecker products. If n_1 denotes the number of rows and n_2 denotes the number of columns of the image domain Ω, we have

$$D_1 := \mathbb{1}_{n_2}\otimes\tilde{D}^1\,, \quad D_2 := \tilde{D}^2\otimes\mathbb{1}_{n_1}\,, \tag{A.90}$$

where $\tilde{D}^1 \in \{-1,0,1\}^{n_1\times n_1}$ and $\tilde{D}^2 \in \{-1,0,1\}^{(n_2-1)\times n_2}$ are given through

$$\tilde{D}^1_{ij} := \begin{cases} 1, & j = i+1, i < n_1, \\ -1, & j = i, i < n_1, \\ 0, & \text{otherwise}\,, \end{cases}, \quad \tilde{D}^2_{ij} := \begin{cases} 1, & j = i+1, i < n_2, \\ -1, & j = i, i < n_2, \\ 0, & \text{otherwise}\,. \end{cases}$$

As an example the difference matrices of a 3×4 image are given through

$$D_1 = \begin{pmatrix}
-1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix} \in \{-1, 0, 1\}^{3\cdot4 \times 3\cdot4}$$

$$D_2 = \begin{pmatrix}
-1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix} \in \{-1, 0, 1\}^{3\cdot4 \times 3\cdot4}$$

The discrete gradient $\nabla_z d_i(z)$ can now be expressed as follows

$$\nabla_z d_i(z) = \begin{pmatrix} (D_1 d_i)_{p(z)} \\ (D_2 d_i)_{p(z)} \end{pmatrix}, \tag{A.91}$$

where d_i is the vector representation (stacked column wise) of the inverse depth map $(d_i(z))_{z\in\Omega}$ and $p(z)$ denotes position (index) of z in the vector d_i, i.e. $(d_i)_{p(z)} = d_i(z)$. The derivative of (A.89) can be calculated regarding $d_i(\tilde{z})$:

$$\begin{aligned}
\frac{\partial}{\partial d_i(\tilde{z})}\Phi(x) =& \lambda \frac{\partial}{\partial d_i(\tilde{z})} \sum_{z\in\Omega} \phi_{\mathrm{reg}}\big(\tfrac{1}{2}\|\nabla_z d_i(z)\|^2\big) \\
=& \lambda \sum_{z\in\Omega} \frac{\partial}{\partial d_i(\tilde{z})} \left(\big(\tfrac{1}{2}\| \begin{pmatrix} (D_1 d_i)_{p(z)} \\ (D_2 d_i)_{p(z)} \end{pmatrix} \|^2 + \nu_r\big)^{\alpha_r} - \nu_r^{\alpha_r} \right) \\
=& \lambda \sum_{z\in\Omega} \alpha_r \big(\tfrac{1}{2}\| \begin{pmatrix} (D_1 d_i)_{p(z)} \\ (D_2 d_i)_{p(z)} \end{pmatrix} \|^2 + \nu_r\big)^{\alpha_r - 1} \\
& \cdot \big((D_1 d_i)_{p(z)} (D_1 e_{p(\tilde{z})})_{p(z)} + (D_2 d_i)_{p(z)} (D_2 e_{p(\tilde{z})})_{p(z)} \big) \\
:=& (G_{d_i}^{\mathrm{reg}}(x))_{p(\tilde{z})}
\end{aligned} \tag{A.92}$$

Stacking these partial derivatives into a vector results in the gradient regarding the depth, i.e. $D_{d_i}\Phi(x) =: G_{\mathrm{reg}}(x)$. By using the previously calculated expressions G_E from (A.69) and G_{d_i} from (A.70) and combining them with the expression (A.92) we find the differential of the expanded Hamiltonian in (A.86) which reads for $x = (E, v, d_i) \in \mathcal{G}$

$$\mathbf{D}_1 \mathcal{H}(x, 0, t) = T_{\mathrm{Id}} L_x \big(G_E(x), 0, T_{\mathrm{Id}_{(0,1)^{|\Omega|}}} L_{d_i}^* \big(G_{d_i}(x) + G_{d_i}^{\mathrm{reg}}(x) \big) \big) . \tag{A.93}$$

For the second-order optimal filter it remains to compute the Hessian of the Hamiltonian regarding the first component $\mathbf{D}_1\mathcal{H}(x,0,t)$ which was partially done within the last section for the data term. Therefore, we continue separately with the regularizer Φ in (A.89). This expression does not depend on the camera motion $E \in \mathrm{SE}_3$ and the velocity $v \in \mathbb{R}^6$. Thus, we only require to calculate the mixed derivatives of Φ since the geometry of the disparity group $(0,1)^{|\Omega|}$ is trivial (the corresponding Christoffel-symbols are all zero).

$$
\begin{aligned}
\frac{\partial^2}{\partial d_i(\hat{z})\partial d_i(\tilde{z})}\Phi(x) = \frac{\partial}{\partial d_i(\hat{z})}\lambda\sum_{z\in\Omega}\alpha_r\big(\tfrac{1}{2}\|\begin{pmatrix}(D_1d_i)_{p(z)}\\(D_2d_i)_{p(z)}\end{pmatrix}\|^2+\nu_r\big)^{\alpha_r-1}\\
\cdot\big((D_1d_i)_{p(z)}(D_1e_{p(\tilde{z})})_{p(z)}+(D_2d_i)_{p(z)}(D_2e_{p(\tilde{z})})_{p(z)}\big)\\
=\lambda\sum_{z\in\Omega}\alpha_r(\alpha_r-1)\big(\tfrac{1}{2}\|\begin{pmatrix}(D_1d_i)_{p(z)}\\(D_2d_i)_{p(z)}\end{pmatrix}\|^2+\nu_r\big)^{\alpha_r-2}\\
\cdot\big((D_1d_i)_{p(z)}(D_1e_{p(\tilde{z})})_{p(z)}+(D_2d_i)_{p(z)}(D_2e_{p(\tilde{z})})_{p(z)}\big)\\
\cdot\big((D_1d_i)_{p(z)}(D_1e_{p(\hat{z})})_{p(z)}+(D_2d_i)_{p(z)}(D_2e_{p(\hat{z})})_{p(z)}\big)\\
+\alpha_r\big(\tfrac{1}{2}\|\begin{pmatrix}(D_1d_i)_{p(z)}\\(D_2d_i)_{p(z)}\end{pmatrix}\|^2+\nu_r\big)^{\alpha_r-1}\\
\cdot\big((D_1e_{p(\hat{z})})_{p(z)}(D_1e_{p(\tilde{z})})_{p(z)}+(D_2e_{p(\hat{z})})_{p(z)}(D_2e_{p(\tilde{z})})_{p(z)}\big)\\
=:(H_{\mathrm{reg}}(x))_{p(\hat{z})p(\tilde{z})}.
\end{aligned}
$$

Adding the Hessian of the regularization term to the Hessian of the data term, which was calculated above, finally results in the Hessian of the Hamiltonian regarding the first component. All the other expressions which are required for the minimum energy filter with spatial regularization were already calculated in appendix A.5.3. By adding the corresponding gradients and Hessians of the regularizers we finally obtain the minimum energy filter for the problem of joint monocular disparity map and camera motion estimation which also includes a spatial regularizer. This completes the calculations.

Bibliography

[1] P. A. Absil, R. Mahony, and R. Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, 2008.

[2] A. A. Agrachev and Y. Sachkov. *Control Theory from the Geometric Viewpoint*, volume 2. Springer, 2004.

[3] A. P. Aguiar and J.P. Hespanha. Minimum-Energy State Estimation for Systems with Perspective Outputs. *Automatic Control, IEEE Transactions on*, 51(2):226–241, Feb 2006.

[4] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking. *Signal Processing, IEEE Transactions on*, 50(2):174–188, 2002.

[5] M. Athans and P. Falb. *Optimal Control. An Introduction to the Theory and Its Applications*. McGraw-Hill, 1966.

[6] H. Badino, A. Yamamoto, and T. Kanade. Visual Odometry by Multi-Frame Feature Integration. In *Computer Vision Workshops (ICCVW), 2013 IEEE International Conference on*, pages 222–229. IEEE, 2013.

[7] A. Bain and D. Crisan. *Fundamentals of Stochastic Filtering*. Springer, 2009.

[8] F. Becker, F. Lenzen, J. H. Kappes, and C. Schnörr. Variational Recursive Joint Estimation of Dense Scene Structure and Camera Motion from Monocular High Speed Traffic Sequences. *IJCV*, 105:269–297, 2013.

[9] F. Bellavia, M. Fanfani, F. Pazzaglia, and C. Colombo. Robust Selective Stereo SLAM without Loop Closure and Bundle Adjustment. In *Image Analysis and Processing–ICIAP 2013*, pages 462–471. Springer, 2013.

[10] P. Benner, J. R. Li, and T. Penzl. Numerical Solution of Large-Scale Lyapunov Equations, Riccati Equations, and Linear-Quadratic Optimal Control Problems. *Numerical Linear Algebra with Applications*, 15(9):755–777, 2008.

[11] P. Benner and J. Saak. Numerical Solution of Large and Sparse Continuous Time Algebraic Matrix Riccati and Lyapunov Equations: A State of The Art Survey. *GAMM-Mitteilungen*, 36(1):32–52, 2013.

[12] J. Berger, F. Lenzen, F. Becker, A. Neufeld, and C. Schnörr. Second-Order Recursive Filtering on the Rigid-Motion Lie Group SE(3) Based on Nonlinear Observations, 2015. ArXiv, preprint.

[13] J. Berger, A. Neufeld, F. Becker, F. Lenzen, and C. Schnörr. Second Order Minimum Energy Filtering on SE(3) with Nonlinear Measurement Equations. In *Scale Space and Variational Methods in Computer Vision*, pages 397–409. Springer, 2015.

[14] J. Berger and C. Schnörr. Joint Recursive Monocular Filtering of Camera Motion and Disparity Map. In *38th German Conference on Pattern Recognition*, 2016. in press.

[15] S. Bonnabel. Left-Invariant Extended Kalman Filter and Attitude Estimation. In *IEEE conference on decision and control*, pages 1027–1032, 2007.

[16] G. Bourmaud and R. Mégret. Robust Large Scale Monocular Visual SLAM. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1638–1647, 2015.

[17] G. Bourmaud, R. Mégret, M. Arnaudon, and A. Giremus. Continuous-Discrete Extended Kalman Filter on Matrix Lie Groups Using Concentrated Gaussian Distributions. *Journal of Mathematical Imaging and Vision*, 51(1):209–228, 2015.

[18] D. Brigo, B. Hanzon, and F. Le Gland. Approximate Nonlinear Filtering by Projection on Exponential Manifolds of Densities. *Bernoulli*, 5(3):495–534, 1999.

[19] D. Bump. *Lie Groups*. Springer, 2004.

[20] Y. Chikuse. *Statistics on Special Manifolds*, volume 174. Springer Science & Business Media, 2012.

[21] G. S. Chirikjian. *Stochastic Models, Information Theory, and Lie Groups, Volume 2: Analytic Methods and Modern Applications*, volume 2. Springer Science & Business Media, 2011.

[22] J. L. Crassidis and F. L. Markley. Unscented Filtering for Spacecraft Attitude Estimation. *Journal of guidance, control, and dynamics*, 26(4):536–542, 2003.

[23] R. Dahlhaus and J. C. Neddermeyer. Online Spot Volatility-Estimation and Decomposition with Nonlinear Market Microstructure Noise Models. *Journal of Financial Econometrics*, 12(1):174–212, 2014.

[24] F. Daum. Nonlinear Filters: Beyond the Kalman Filter. *IEEE A&E Systems Magazin*, 20(8, Part 2):57–69, 2005.

[25] F. Daum and J. Huang. Curse of Dimensionality and Particle Filters. In *Aerospace Conference*, 2003.

[26] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse. MonoSLAM: Real-Time Single Camera SLAM. *PAMI*, 29(6):1052–1067, 2007.

[27] L. Dieci and T. Eirola. Positive Definiteness in the Numerical Solution of Riccati Differential Equations. *Numerische Mathematik*, 67(3):303–313, 1994.

[28] P. Dollár. Piotr's Computer Vision Matlab Toolbox (PMT). http://vision.ucsd.edu/ pdollar/toolbox/doc/index.html.

[29] A. Doucet, N. Freitas, and N. Gordon. *Sequential Monte Carlo Methods in Practice*, chapter An Introduction to Sequential Monte Carlo Methods, pages 3–14. Springer New York, New York, NY, 2001.

[30] J. Engel, T. Schöps, and D. Cremers. LSD-SLAM: Large-Scale Direct Monocular SLAM. In *Computer Vision–ECCV 2014*, pages 834–849. Springer, 2014.

[31] J. Engel, J. Sturm, and D. Cremers. Semi-Dense Visual Odometry for a Monocular Camera. In *ICCV*, pages 1449–1456. IEEE, 2013.

[32] L. C. Evans. *Partial Differential Equations*. Graduate studies in mathematics. American Mathematical Society, 2010.

[33] R. J. Fitzgerald. Divergence of the Kalman Filter. *Automatic Control, IEEE Transactions on*, 16(6):736–747, 1971.

[34] W. H. Fleming and R. W. Rishel. *Deterministic and Stochastic Optimal Control*, volume 1. Springer Science & Business Media, 1982.

[35] P. Frogerais, J. Bellanger, and L. Senhadji. Various Ways to Compute the Continuous-Discrete Extended Kalman Filter. *Automatic Control, IEEE Transactions on*, 57:1000–1004, 2012.

[36] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for Autonomous Driving? the KITTI Vision Benchmark Suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[37] A. Geiger, J. Ziegler, and C. Stiller. Stereoscan: Dense 3D Reconstruction in Real-Time. In *Intelligent Vehicles Symposium (IV), IEEE*, 2011.

[38] N.J. Gordon, D. J Salmond, and A. F. M. Smith. Novel Approach to Nonlinear/Non-Gaussian Bayesian State Estimation. In *Radar and Signal Processing, IEE Proceedings F*, volume 140, 1993.

[39] E. Hairer, C. Lubich, and G. Wanner. *Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations*. Springer, 2006.

[40] B.C. Hall. *Lie Groups, Lie Algebras, and Representations: An Elementary Introduction*, volume 222. Springer, 2015.

[41] R. Hartley. In Defense of The Eight-Point Algorithm. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(6):580–593, 1997.

[42] S. Hauberg, F. Lauze, and K. Pedersen. Unscented Kalman Filtering on Riemannian Manifolds. *Journal of Mathematical Imaging and Vision*, 46(1):103–120, 2013.

[43] J. Hilgert and K.-H. Neeb. *Structure and Geometry of Lie Groups*. Springer Science & Business Media, 2011.

[44] H. Hirschmüller. Stereo Processing by Semiglobal Matching and Mutual Information. *PAMI*, 30(2):328–341, 2008.

[45] E. P. Hsu. *Stochastic Analysis on Manifolds*, volume 38. American Mathematical Soc., 2002.

[46] A. H. Jazwinski. *Stochastic Processes and Filtering Theory*. Courier Corporation, 2007.

[47] S. J. Julier and J. K. Uhlmann. New Extension of the Kalman Filter to Nonlinear Systems. In *AeroSense'97*, pages 182–193. International Society for Optics and Photonics, 1997.

[48] V. Jurdjevic. *Geometric Control Theory*. Cambridge university press, 1997.

[49] R. E. Kalman. A New Approach to Linear Filtering and Prediction Problems. *Journal of Fluids Engineering*, 82(1):35–45, 1960.

[50] A. J. Krener. The Convergence of the Minimum Energy Estimator. In *New Trends in Nonlinear Dynamics and Control and their Applications*. Springer, 2003.

[51] H. J. Kushner. On the Differential Equations Satisfied by Conditional Probablitity Densities of Markov Processes, with Applications. *SIAM Journal on Control and Optimization*, 2(1):106, 1964.

[52] H. J. Kushner. Approximations to Optimal Nonlinear Filters. *Automatic Control, IEEE Transactions on*, 12(5):546–556, 1967.

[53] J. Kwon, M. Choi, F. C. Park, and C. Chun. Particle Filtering on the Euclidean Group: Framework and Applications. *Robotica*, 25(6):725–737, 2007.

[54] J. Kwon and K. M. Lee. Monocular SLAM with Locally Planar Landmarks via Geometric Rao-Blackwellized Particle Filtering on Lie Groups. In *CVPR 2010*, pages 1522–1529. IEEE, 2010.

[55] P. Lancaster and L. Rodman. Existence and Uniqueness Theorems for the Algebraic Riccati Equation. *International Journal of Control*, 32(2):285–309, 1980.

[56] F. L. Markley. Attitude Error Representations for Kalman Filtering. *Journal of guidance, control, and dynamics*, 26(2):311–317, 2003.

[57] M. I. Miller, A. Srivastava, and U. Grenander. Conditional-Mean Estimation via Jump-Diffusion Processes in Multiple Target Tracking/Recognition. *Signal Processing, IEEE Transactions on*, 43(11):2678–2690, 1995.

[58] R. E. Mortensen. Maximum-Likelihood Recursive Nonlinear Filtering. *J. Opt. Theory Appl.*, 2(6):386–394, 1968.

[59] R. M. Murray, Z. Li, and S. S. Sastry. *A Mathematical Introduction to Robotic Manipulation*. CRC press, 1994.

[60] B. Nasihatkon, R. Hartley, and J. Trumpf. A Generalized Projective Reconstruction Theorem and Depth Constraints for Projective Factorization. *International Journal of Computer Vision*, pages 1–28, 2015.

[61] A. Neufeld, J. Berger, F. Becker, F. Lenzen, and C. Schnörr. Estimating Vehicle Ego-Motion and Piecewise Planar Scene Structure from Optical Flow in a Continuous Framework. In *GCPR*, 2015.

[62] D. Nistér. An Efficient Solution to the Five-Point Relative Pose Problem. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(6):756–770, 2004.

[63] D. Ocone and E. Pardoux. A Lie Algebraic Criterion for Non-Existence of Finite Dimensionally Computable Filters. In *Stochastic Partial Differential Equations and Applications II*, pages 197–204. Springer, 1989.

[64] K. B. Petersen and M. S. Pedersen. The Matrix Cookbook. *Technical University of Denmark*, 7:15, 2008.

[65] M. Pizzoli, C. Forster, and D. Scaramuzza. REMODE: Probabilistic, Monocular Dense Reconstruction in Real Time. In *Proc. IEEE Int. Conf. on Robotics and Automation*, 2014.

[66] L. S. Pontryagin, V. Boltyanskii, R. Gamkrelidze, and E. Mishchenko. The Mathematical Theory of Optimal Processes. *Interscience Publishers, Inc.*, 1962.

[67] E. T. Psota, J. Kowalczuk, M. Mittek, and L. C. Perez. MAP Disparity Estimation Using Hidden Markov Trees. In *ICCV*, pages 2219–2227, 2015.

[68] Q. Rentmeesters, P. A. Absil, P. Van Dooren, K. Gallivan, and A. Srivastava. An Efficient Particle Filtering Technique on the Grassmann Manifold. In *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, pages 3838–3841. IEEE, 2010.

[69] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid. EpicFlow: Edge-Preserving Interpolation of Correspondences for Optical Flow. In *CVPR*, 2015.

[70] A. Saccon, A. P. Aguiar, and J. Hauser. Lie Group Projection Operator Approach: Optimal Control on t so (3). In *CDC*, pages 6973–6978, 2011.

[71] A. Saccon, J. Hauser, and A. P. Aguiar. Optimal Control on Lie Groups: The Projection Operator Approach. *Automatic Control, IEEE Transactions on*, 58(9):2230–2245, Sept 2013.

[72] A. Saccon, J. Trumpf, R. Mahony, and A. P. Aguiar. Second-Order-Optimal Filters on Lie groups. In *CDC*, 2013.

[73] A. Saccon, J. Trumpf, R. Mahony, and A. P. Aguiar. Second-Order-Optimal Minimum-Energy Filters on Lie Groups. *IEEE TAC*, PP(99):1–1, 2015.

[74] S. Särkkä, J. Hartikainen, and A. Solin. EKF/UKF Toolbox for Matlab V1.3, 2011.

[75] T. Schön, F. Gustafsson, and P. J. Nordlund. Marginalized Particle Filters for Mixed Linear/Nonlinear State-Space Models. *IEEE Transactions on Signal Processing*, 53(7):2279–2289, July 2005.

[76] J. M. Selig. Lie Groups and Lie Algebras in Robotics. In *Computational Noncommutative Algebra and Applications*, pages 101–125. Springer, 2004.

[77] H. W. Sorenson. On The Development of Practical Nonlinear Filters. *Information Sciences*, 7:253 – 270, 1974.

[78] P. Sturm and B. Triggs. A Factorization Based Algorithm for Multi-Image Projective Structure and Motion. In *ECCV*, pages 709–720, Cambridge, England, 1996. Springer.

[79] F. Tompkins and P. J. Wolfe. Bayesian Filtering on the Stiefel Manifold. In *Computational Advances in Multi-Sensor Adaptive Processing, 2007. CAMPSAP 2007. 2nd IEEE International Workshop on*, pages 261–264. IEEE, 2007.

[80] B. Triggs, P. F. McLauchlan, R. I Hartley, and A. W. Fitzgibbon. Bundle AdjustmentA Modern Synthesis. In *Vision Algorithms: Theory and Practice*, pages 298–372. Springer, 2000.

[81] A. Tyagi and J. W. Davis. A Recursive Filter for Linear Systems on Riemannian Manifolds. In *CVPR*, pages 1–8. IEEE, 2008.

[82] C. Vogel, K. Schindler, and S. Roth. 3D Scene Flow Estimation with a Piecewise Rigid Scene Model. *IJCV*, 115(1):1–28, 2015.

[83] P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid. Deepflow: Large Displacement Optical Flow with Deep Matching. In *ICCV*, pages 1385–1392, 2013.

[84] G. Welch and G. Bishop. An Introduction to the Kalman Filter, 1995.

[85] T. Whelan, M. Kaess, J. J. Leonard, and J. McDonald. Deformation-Based Loop Closure for Large Scale Dense RGB-D SLAM. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 548–555. IEEE, 2013.

[86] K. C. Wolfe, M. Mashner, and G. Chirikjian. Bayesian Fusion on Lie Groups. *J. Algebraic Statistics*, 2(1):75–97, 2011.

[87] J. Zamani, M.and Trumpf and R. Mahony. Minimum-Energy Filtering for Attitude Estimation. *Automatic Control, IEEE Transactions on*, 58(11):2917–2921, 2013.

[88] M. Zamani, J. Trumpf, and M. Mahoney. A Second Order Minimum-Energy Filter on the Special Orthogonal Group. In *Proc. ACC*, 2012.

[89] M. Žefran, V. Kumar, and C. Croke. Metrics and Connections for Rigid-Body Kinematics. *The International Journal of Robotics Research*, 18(2):242–1, 1999.