Institut für Geodäsie und Geoinformation
Bereich Photogrammetrie

D 98

Visual SLAM from image sequences acquired by unmanned aerial vehicles

Inaugural-Dissertation

zur

Erlangung des Grades

Doktor-Ingenieur

(Dr.-Ing.)

der

Hohen Landwirtschaftlichen Fakultät

der

Rheinischen Friedrich-Wilhelms-Universität

zu Bonn

vorgelegt am 27. Juli 2009 von

Richard Steffen

aus Schwerin

Referent:	Prof. Dr. Wolfgang Förstner
Korreferent:	Prof. Dr. Wolf-Dieter Schuh
Tag der mündlichen Prüfung:	20.11.2009
Erscheinungsjahr:	2009
Gedruckt bei:	Diese Dissertation ist auf dem Hochschulschriftenserver der ULB Bonn http://hss.ulb.uni-bonn.de/diss_online elektronisch publiziert.

Never try, never know ...

This dissertation is dedicated to my parents and to my sister. Thank you for giving me the strength and hope. Foremost, this work would not have been possible without Prof. Wolfgang Förstner. It was the best decision in my life to join your research group. Thank you for your excellent advise, patience and all your support. I will never forget this time. A special thanks goes to my colleagues at the Department of Photogrammetry for all the productive discussions. The genial work atmosphere was one of the reasons that made this thesis a success. In particular, I thank Hanns-Florian Schuster, Mark Luxen, Christian Beder and Jochen Meidow for giving me the opportunity to learn from you. A special thanks to Heidi for checking the English spellings. I will remember you as the good spirit of the department. Finally, I am grateful to all my friends who shared the path of my lifeThis is much more important to me, than you can imagine.

Zusammenfassung

Visuelle gleichzeitige Lokalisierung und Kartierung aus Bildfolgen von unbemannten Flugkörpern

Diese Arbeit zeigt, dass die Kalmanfilter basierte Lösung der Triangulation zur Lokalisierung und Kartierung aus Bildfolgen von unbemannten Flugkörpern realisierbar ist. Aufgrund von Echtzeitanforderungen autonomer Systeme erreichen rekursive Schätz-verfahren, insbesondere Kalmanfilter basierte Ansätze, große Beliebheit. Bedauerlicherweise treten dabei durch die Nichtlinearität der Triangulation einige Effekte auf, welche die Konsistenz und Genauigkeit der Lösung hinsichtlich der geschätzten Parameter maßgeblich beeinflussen.

Der *erste Beitrag* dieser Arbeit besteht in der Herleitung eines generellen Verfahrens zum rekursiven Verbessern im Kalmanfilter mit impliziten Beobachtungsgleichungen. Wir zeigen, dass die klassischen Verfahren im Kalmanfilter eine Spezialisierung unseres Ansatzes darstellen.

Im *zweiten Beitrag* erweitern wir die klassische Modellierung für ein Einkameramodell zu einem Mehrkameramodell im Kalmanfilter. Diese Erweiterung erlaubt es uns, die Prädiktion für eine lineares Bewegungsmodell vollkommen linear zu berechnen.

In einem *dritten Hauptbeitrag* stellen wir ein neues Verfahren zur Initialisierung von Neupunkten im Kalmanfilter vor. Anhand von empirischen Untersuchungen unter Verwendung simulierter und realer Daten einer Bildfolge eines photogrammetrischen Streifens zeigen und vergleichen wir, welchen Einfluß die Initialisierungsmethoden für Neupunkte im Kalmanfilter haben und welche Genauigkeiten für diese Szenarien erreichbar sind.

Am Beispiel von Bildfolgen eines unbemannten Flugkörpern zeigen wir in dieser Arbeit als *vierten Beitrag*, welche Genauigkeit zur Lokalisierung und Kartierung durch Triangulation möglich ist. Diese theoretische Analyse kann wiederum zu Planungszwecken verwendet werden.

Abstract

Visual SLAM from image sequences acquired by unmanned aerial vehicles

This thesis shows that Kalman filter based approaches are sufficient for the task of simultaneous localization and mapping from image sequences acquired by unmanned aerial vehicles. Using solely direction measurements to solve the problem of simultaneous localization and mapping (SLAM) is an important part of autonomous systems. Because the need for real-time capable systems, recursive estimation techniques, Kalman filter based approaches are the main focus of interest. Unfortunately, the non-linearity of the triangulation using the direction measurements cause decrease of accuracy and consistency of the results.

The *first contribution* of this work is a general derivation of the recursive update of the Kalman filter. This derivation is based on implicit measurement equations, having the classical iterative non-linear as well as the non-iterative and linear Kalman filter as specializations of our general derivation.

Second, a new formulation of linear-motion models for the single camera state model and the sliding window camera state model are given, that make it possible to compute the prediction in a fully linear manner.

The *third major contribution* is a novel method for the initialization of new object points in the Kalman filter. Empirical studies using synthetic and real data of an image sequence of a photogrammetric strip are made, that demonstrate and compare the influences of the initialization methods of new object points in the Kalman filter.

Forth, the accuracy potential of monoscopic image sequences from unmanned aerial vehicles for autonomous localization and mapping is theoretically analyzed, which can be used for planning purposes.

Contents

Chapter 1

Introduction

Orbis Typus Universalis, Waldseemüller 1506

A major goal of photogrammetry and computer vision is a fully automatic reconstruction of an environment from images. An important aspect is the determination of the image location and orientation. Although much progress has been made, the problem still remains an active research field. A lot of attention to this problem has been received by the robotics community simply because images carry manifold information about the scene. Hence, autonomous systems are able to use the image information for different tasks, even for their location determination.

The human ability to learn maps of the surrounding environment and to use them for localization has inspired researchers in the robotics community over the last decades. The understanding and acquisition of this ability has been identified as a fundamental problem in robotics. The process of incremental map construction is called *simultaneous localization and mapping (SLAM)*. This task is a key component for building autonomous mobile devices, that for many applications has been a dream of researchers. In the field of civil applications service, robots perform cleaning, inspection and transportation tasks or medical and construction assistance. Robots also operate in dangerous environments, say for life rescue or pollution control. Robots can be used in a wide field of research applications, for instance space and deep sea exploration. Moreover, in military applications for instance robots are useful in investigating areas and for transportation and rescue tasks.

The research community is focused on different kinds of aspects concerning the mapping ability. Various sensor types and configurations to obtain map information as well as knowledge about the robots location will be used. Of course, the developed methods depend on the application environment and the a-priori information about it. The use of multiple cooperating robots is of interest, caused by its advantages in case of fault-tolerance and to accomplish a task faster. Available solutions for narrowed environments are well understood. For unstructured large-scale environments open problems remain. These are issues regarding map representation and its uncertainty, real-time capable update methods and sensor fusion algorithms. Almost every technique assumes the environment to be static. A more difficult problem occurs if the environment changes over time.

A module to solve the task of simultaneous localization and mapping can be understood as a subcomponent of an entire system for autonomous robots. Its main function is to aggregate observations obtained by sensors in order to obtain information of the environment stored in a map and provide this informations to other subsystems. The map and the relative location of the robot may be used for interaction with the environment depending on the robots task. This interaction comprises for instance safe navigation, exploration of areas where the robots knowledge about the map is uncertain, identifying and following dynamic objects or manipulation of the environment.

1.1 Motivation

The task of simultaneous localization and mapping is an old problem. Since the computer technology provides sufficient computation power and data from sensor systems is digitalized automatically, classical techniques becomes practical for real-time applications. One of the well-established techniques is named as *triangulation*, where direction measurements from different locations are used to determine the location of an observed object and the location of the observer himself. In the area of nautical navigation this task is called *dead reckoning* and is known since millenniums.

In classical photogrammetry a realization incorporating uncertainties of the direction measurements into the triangulation is given by a least square solution known as *bundle adjustment*. This technique can be adapted to the real-time localization and mapping problem of a robot. It becomes even more complex in the case of large environments and a huge number of observations and mapping parameters. To handle this complexity a reformulation of the least square solution to a recursive update that is part of the *Kalman filter* can be used. Several researchers adapted this technique to recover the map structure as well as the localization of the robot at any point in time. One argument to use the recursive update is always the high computational performance. Unfortunately, many researchers observe disadvantages concerning the consistency of the resulting estimates and its uncertainty. These inconsistencies appear dominant in the case when only direction measurements are used.

The contribution of this thesis is to derive real-time capable models based on a recursive update technique using image measurements only and to compare their benefit to the results concerning consistency and precision. In practice we are interested in the domain of lightweight *unmanned aerial vehicles (UAV)*. Intentionally we do not incorporate additional sensors like global positioning systems (e.g. GPS) and inertial systems (INS) to determine the location of the robot directly. There are several applications to use a UAV without access to a global positing system for a specific task supposable, for instance indoor applications, between skyscrapers, under bridges, in tunnels or in military faced approaches where the GPS signals are jammed as well as extra-planetary missions. High accurate INS sensors are still heavy and expensive and therefore not usable in all kinds of applications. It is certainly advisable to incorporate additional sensor information if available. This work may be considered as a continuation of the work of Ackermann (1965) and Li (1987). Both highlighted aspects about the accuracy and the sensitivity analysis of the outlier detection of the triangulation using high resolution wide baseline aerial images. Our approach will deal with a low resolution image sequence. This thesis will examine the questions of

- how to represent the environment and the robots dynamics in a Kalman filter,
- how the initialization of new object points in the map influences the overall consistency and precision of the results,
- how to deal with outliers in the observations, and
- how to find a general law to assess the achievable accuracy in case of aerial image sequences.

The common method to represent the environment and the robots dynamics in a Kalman filter follows Davison (2003). We will improve and extend this representation to be more flexible. To our knowledge, the inconsistency of the Kalman filter based results observed by many researchers e. g. Castellanos *et al.* (2004), Bailey *et al.* (2006) has not yet been analyzed for photogrammetric strips. Furthermore, the known previous work assumes that the observations are free of gross errors in the observations. This assumption cannot be guaranteed. Hence, we will present a method to reduce the outlier influence in a Kalman filter based approach. The question about the achievable accuracy in case of low resolution aerial image sequences becomes interesting to assess the approach for new application tasks.

This thesis is organized as follows. First, we will give an overview about the meaning and available techniques of the simultaneous localization and mapping problem and review the previous work in a broader context.

In Chapter 3 the relevant methods to this thesis in the area of projective geometry, image geometry and least squares solutions are presented. A novel algorithm for a recursive update will be introduced. It will be shown, that this novel algorithm is a generalization of the classical recursive update of the Kalman filter. In addition, we will examine the task of outlier detection and elimination.

Chapter 4 addresses the problem how the environment can be represented in a Kalman filter. We will introduce a new state representation for the robot pose and a completely linear dynamic model. In the second part we will highlight the problem of parameter initialization in terms of a variable state according to the extended knowledge of the environment during exploration. In this context we will introduce a new approach to initialize new object points into the filter state.

The proposed algorithms will be evaluated in Chapter 5 using synthetic datasets and real datasets. We develop a model for the theoretical accuracy as a function of the design parameters of a flight line. In a second step the influence of the initialization method of new object points to the resulting observers trajectory will be analyzed in case of full synthetic observations, synthetic image sequences and real image sequences. Finally we will conclude this thesis with an outlook to future work.

1.2 Collaborations and publications

Some algorithms proposed in this thesis were developed in cooperation with other people. The idea of a non-linear recursive update as well as a quality measure for the intersection of projection rays have been presented by Christian Beder in Steffen & Beder (2007) and Beder & Steffen (2006). A new solution to initialize object points in a Kalman filter based approach for the mapping task was done in collaboration with Wolfgang Förstner, presented in Steffen & Förstner (2008). Parts of this thesis have been published in the following articles:

- Christian Beder and Richard Steffen. Determining an initial image pair for fixing the scale of a 3d reconstruction from an image sequence. In K. Franke, K.-R. Müller, B. Nickolay, and R. Schäfer, editors, *Pattern Recognition*, number 4174 in LNCS, pages 657–666. Springer, 2006.

- Christian Beder and Richard Steffen. Incremental estimation without specifying a-priori covariance matrices for the novel parameters. *VLMP Workshop on CVPR. Anchorage, USA*, 2008.

- Wolfgang Förstner and Richard Steffen. Online geocoding and evaluation of large scale imagery without GPS. *Photogrammetric Week, Heidelberg*, Wichmann Verlag, 2007.

- Richard Steffen and Christian Beder. Recursive estimation with implicit constraints. In F.A. Hamprecht, C. Schnörr, and B. Jähne, editors, *Proceedings of the DAGM 2007*, number 4713 in LNCS, pages 194–203. Springer, 2007.

- Richard Steffen and Wolfgang Förstner. On visual real time mapping for unmanned aerial vehicles. In *21st Congress of the International Society for Photogrammetry and Remote Sensing (ISPRS)*, Beijing, China, 2008.

1.3 Notation

symbol	**meaning**
general writing style	
x, X	scalar value
$\boldsymbol{x}, \boldsymbol{X}$	vector
$\mathit{\mathsf{X}}$	matrix
$\mathbf{x}, \mathbf{X}$	homogeneous vector
X	homogeneous matrix
$\widehat{x}$	estimated variable
$\widetilde{x}$	true value of stochastical variable
$\vec{\boldsymbol{x}}$	predicted vector
$\mathit{\mathsf{S}}(\bullet)$	skew matrix of a 3-vector
reserved symbols	

L	roundness measure
R	rotation matrix
M	homogeneous motion matrix
K	camera calibration matrix
P	camera projection matrix
$P(\cdot)$	probability density function of a variable
$\boldsymbol{p}$	parameter vector or state
D	duality matrix for homogeneous elements
V	unit quaternion inversion matrix
U	concatenation matrix for small quaternion
Υ	quaternion representation in matrix form
$\breve{\Upsilon}$	reinverse quaternion representation in matrix form
$\boldsymbol{z}$	observation vector
ω, ϕ, κ	Euler angles for rotations around X, Y, Z - axis
Θ	threshold of a scalar variable
statistic symbols	
σ	standard deviation
C	covariance matrix
ϵ	white noise vector with mean zero
χ^2	χ-square distribution
$N(\boldsymbol{\mu}, \mathsf{C})$	normal distribution with mean vector $\boldsymbol{\mu}$ and covariance C
C_{XX}	covariance matrix of parameter vector $\boldsymbol{X}$
mathematic operators and indices	
$\boldsymbol{X}^{\mathsf{T}}$	transpose
X^{-1}	inverse of a matrix
$\mathsf{X}^{-\mathsf{T}}$	inverse transpose of a matrix
$\boldsymbol{X}^{(\nu)}$	iteration ν counter
$\boldsymbol{X}_t$	time index
$\|\mathsf{A}\|$	determinant of the matrix A
$\|\boldsymbol{X}\|$	$L2$-norm of the vector X
$\mathrm{tr}(\mathsf{A})$	trace or sum of the diagonal elements of A

reference indices	
${}^{c}\boldsymbol{X}$	variable in camera coordinate system
${}^{o}\boldsymbol{X}$	variable in object coordinate system
${}^{w}\boldsymbol{X}$	variable in world coordinate system or geo-referenced system
${}^{w}\mathsf{M}_{o}$	homogeneous motion from object to world system
${}^{w}\mathsf{M}_{o}^{-1} = {}^{o}\mathsf{M}_{w}$	inverse homogeneous motion from object to world system is equivalent to the transformation from world to object system

Chapter 2

Previous work

In this chapter we will present the previous work that has been done in the field of simultaneous localization and mapping. To classify the available literature we have to clarify the meaning of *localization*, *mapping* and *simultaneous localization and mapping* as a joint problem.

The *localization* in a known environment as a single problem is to determine the pose of an observer relative to a given map. The pose can usually not be sensed directly and has to be inferred from sensor data. For the localization determination almost every system is based on triangulation, where directions, distances or both of a known object are measured. The real problem is to establishing correspondences between the map and the acquired sensor data. In case of active systems like GPS the object identification is simple as the GPS satellites communicate its identification. Using image measurements or range scanners the identification, also known as the *data association problem*, becomes challenging.

The *mapping* task as a single problem determines the locations of objects in the observers environment. The determination assumes to know the observers location. In case the sensor system to obtain the map gets multiple observations of an object, then the map estimation method has to be taken into account the uncertainty of the observations as well as the uncertainty of the observers location. Again, the data association problem has to be solved. The mapping approach should be able to identify map changes and update the map respectively.

Simultaneous localization and mapping as a joined problem is much more complicated. This task was identified as one of the fundamental problems in the field of autonomous robotics. The problem arises when an observer does not have any information about its location and does not know anything about its environment. The task of simultaneous localization and mapping can be considered as a time driven process. We distinguish between two major forms following Thrun *et al.* (2005). First, the observer will determine its momentary location only, also termed as *online SLAM*. Second, the observer will determine the entire path of the observers trajectory at every point in time, known as *full SLAM*. Both concepts combined with the ability to fall back on previous observations afford different solutions of the problem. In the area of photogrammetry the aero-triangulation is a well understood example for a *full SLAM* approach. But, in case of large maps and real-time tasks *full SLAM* may not be applicable. Due to large maps implicating a huge number of observations and therefore a huge storage space, they are therefore usually not real-time capable. Instead, *online SLAM* performs the integration of new observations one-at-a-time. Typically, a fall back to previous observations is not possible. The information of the observations will be accumulated and represented by a posterior distribution of the map and of the observer's location. Some solutions also neglect the information of past locations which are not necessary anymore. Depending on the task, map information can also be discarded, e. g. if only a save motion is required. Often, these systems operate in an egocentric system comparable to human perception. Approaches, which do not neglect information and resolve inconsistencies of previous and actual observations are called optimal. The "'gold standard"' is to estimate the full posterior distribution about the environment. However, approximative approaches are able to approximate this full posterior distribution in a sufficient manner. The gain is a speed up of the computation and a reduction of the necessary storage space.

The main problem to classify the available literature arises from the wide field of applications, environments, sensors and their combinations. We identified two main areas caused by the sensor type, namely visual sensors and range sensors. The first induces direction measurements. Using multiple image sensors, distances can also be determined at one point in time. Secondly, range sensors measure distances and directions directly. Both will divide the proposed concepts into *visual SLAM* and *scan-matching SLAM*.

In the following sections we will review general estimation techniques in the context of metric simultaneous localization and mapping. The influence of different sensor types and metric map representations on the complexity of the joined location and map estimation will be highlighted and we will outline the benefits of these approaches in various ambiances of particular publications. The main leading works in the already addressed problem of huge maps will be presented.

2.1 Previous work on general estimation techniques

In the task of simultaneous localization and mapping orientation parameters of an observer and parameters of the observed environment have to be estimated. At the moment there are three general methods in the focus of the community, namely least squares, recursive estimation and particle based methods. Thrun (2002) presents an overview of the state of the art techniques. The paper points out the relative strength and weakness of the proposed algorithms.

1. *Least squares* is a commonly-used technique for the parameter estimation tasks (c. f. section 3.3). Here, the posteriori distribution will be approximated by a Gaussian distribution. In case of direction measurements the method will be addressed as *bundle adjustment* and usually will solve the *full SLAM* problem. A very good overview of the bundle adjustment techniques in terms of estimation theory and robustification, solving large normal equation systems efficiently, incremental updates, gauge problems, outlier detection and sensitivity analysis, model selection and network design is given by Triggs *et al.* (1999).

   To achieve real-time capability, algorithms were developed to reduce the computational cost of solving the normal equation system of the least squares solution. These take advantage of the special structure of the normal equation system. Grün (1982) discusses a normal equation factorization method based on the *Schur*-complement[1], which is able to subdivide the parameter vector into two independent solutions. Additionally, a formalism to detect blunder via Baarda's data-snooping is introduced. Also Thrun & Montemerlo (2005) presented this factorization method of Grün (1982) for large scale mapping. To obtain the data association a statistically motivated method based on incrementally updated maps is introduced. Besides the factorization method, a compromise between between *full* and *online SLAM* can be used to significantly reduce the parameter space. According to a hierarchical map representation, the estimation will be performed only on a local map which will be adjusted at any point in time. This means only the last few orientation parameters of the observer and only a fragment of the environment will be used in the estimation process. This adaptation is called *sliding window*, e. g. Bibby & Reid (2007). Mouragnon *et al.* (2006) propose a local

[1] In the following we will refer to the Schur-complement as a factorization.

bundle adjustment using observations of a set of arbitrary chosen images (keyframes) to reduce the number of parameters. Correlations between previous orientation parameters (the observer trajectory) will be ignored. This makes it impossible to resolve the inconsistencies in case of a loop closing in an optimal manner. Additionally, estimated uncertainties of the parameters are too optimistic.

Another important aspect to the time consuming complexity of the least squares solution is the re-linearization of the non-linear observation model in an iterative manner. The convergence of the iteration depends on the degree of the non-linearity as well as on the accuracy of the initial approximate values. Sibley (2006) and McLauchlan (2000) argue, that a re-linearization on previous orientation parameters is not necessary. However, their described methods use a sliding window for the orientation parameters to enhance the solution.

Sometimes only information about the trajectory of the observer is desired. Using image data only the task can be denoted as *visual odometry* estimation. Sünderhauf *et al.* (2005) describes a sparse bundle approach using a sliding window to estimate the motion of the observer only.

As a compendium, Dellaert (2005) reviews the factorization techniques used in bundle adjustment and enhanced the classical model by a dynamic model of an autonomous system. Additionally, he compared some implementations in case of complexity and runtime.

Faugeras *et al.* (1998) introduce an algorithm to recover the structure of an image sequence with uncalibrated cameras. The reconstruction leads to an unknown projective transformation, which can be recovered up to an affine transformation using three pairs of parallel lines and to a similarity transformation using orthogonal lines.

2. *Recursive methods* are based on an update technique of already estimated parameters using new observations to increase its accuracy that can be used to solve the *online-SLAM* problem. The most popular technique is the Kalman filter[2] introduced by Kalman (1960), which will approximate the posteriori distribution of the parameters using a Gaussian distribution. An good overview about Kalman filter techniques can

[2] We do not distinguish between the linear Kalman filter, the extended Kalman filter and the iterative extended Kalman filter in the review section.

be found in Simon (2006). The Kalman filter update can be derived from the least squares solution as well as from the weighted mean of the Bayes theorem, e. g. Koch (1997) p. 182ff and Thrun *et al.* (2005) p. 45ff.

A well-established method using a Kalman filter based technique is proposed for instance by Davison (2003). A linear motion model to get approximate values is used. Civera *et al.* (2007a) extend the approach to an unknown scale factor of the environment. This is useful if no a-priori metric information is available. Julier & Uhlmann (2007) introduced a Kalman filter technique with the update complexity of O(1) ignoring the correlations inside the covariance matrix and a special update procedure. In their experiments the uncertainty is growing three times faster.

In McLauchlan & Murray (1995) and McLauchlan (2000) the interconnections between least squares and recursive estimation for the localization and mapping problem are analyzed. Here observer localization and map parameters will be separated into two groups of parameters. There are three update methods proposed: 1) A recursive update according to the factorization method of Grün (1982) is introduced. For a large number of localization parameters this can be expensive in terms of computational complexity. 2) Only the new localization parameters will be updated, the previous localization parameters are fixed and the correlations are neglected. 3) The new localization parameters will be determined in larger time steps, where the approximate values will be obtained by a prediction using a linear motion model. In all three cases the map parameters will be updated using the factorization method. In contrast, Beder & Steffen (2008) calculate the update for the localization parameters similar to the second method proposed by McLauchlan & Murray (1995), though the previous localization parameters will not be fixed. As an advantage this will consider the correlations.

In case of a highly non-linear prediction and measurement model the Kalman filter can diverge caused by the linear error propagation. To reduce this effect Julier & Uhlmann (1997) introduced an error propagation method for non-linear functions called the unscented transformation as it does reduce the bias in propagation the first and second moments of the distribution. This non-linear error propagation for Gaussian distribution guarantees to keep either the second in mean or fourth moments in variance. It can be applied to the Kalman filter resulting in the so-called *sigma point Kalman filter*. As an extension, Sibley *et al.* (2006) introduced an iterated version of the *sigma* *point Kalman filter*. The computational overhead of calculation is the computation of the inverse of the whole covariance matrix of the state vector.

An alternative way is to approximate non-Gaussian distributions of the parameters, which will usually arise if new parameters have to be initialized, by a sum of Gaussian distributions. Solà *et al.* (2005) introduce a multi-hypothesis initialization of new object points in a Kalman filter based approach.

3. *Particle filter* as non-parametric filter are the common way to represent the posterior distribution by a set of random states assigned with a probability. A good overview on localization and mapping using non-parametric filter can be found in Thrun *et al.* (2005) p. 85ff. However, the number of particles will increase exponentially with the number of parameters and is therefore not suitable for the *SLAM* task.

   Hence, another idea to solve the *online-SLAM* problem is to decouple the map and observer trajectory into two separated parts. This decoupling will be used in the so-called *Fast-SLAM* algorithm introduced by Montemerlo *et al.* (2002). The trajectory is represented by a *Particle filter*. Every particle is associated by an individual map, stored in a tree based data structure for fast access and memory usage reduction. The environment object locations are estimated by separated Kalman filter for every particle and every object simultaneously. Spero & Jarvis (2005) extended the *Fast-SLAM* algorithm by the use of robot hypotheses with no underlying distribution. The hypotheses are computed and valuated in a feature matching process in a RANSAC based manner. Eliazar & Parr (2003) introduced a particle filter based method according to *Fast-SLAM* using range scanners. Here multi hypothesis grid maps for multi-hypothesis trajectories of the observer are associated with a balanced tree, that reduces the memory consumption significantly.

   In Pupilli & Calway (2005) and Pupilli & Calway (2006) also the trajectory parameters of an observer are represented by a particle filter. Here in contrast, one single map is updated by a sigma point Kalman filter with weighted multi hypothesis updates of the object locations according to every particle of the observers trajectory. Täubig & Schröter (2004) extended the approach of Montemerlo *et al.* (2002) by using a particle filter also for the environment parameter estimation. This enables multi-modal distributions of object locations.

2.2 Previous work with different sensor types

Simultaneous localization and mapping approaches are strongly influenced by the used sensor system. In the case of the environment perception this can be monocular or stereo image sensors, range sensors like laser scanners, sonar or radar sensors. Additionally odometry sensors like wheel sensors, GPS or initial sensors for rotation and velocity are used.

The integration of odometry sensors will always stabilize a system, especially in case of weak sensors to acquire information about the environment. For example, Davison & Kita (2001) demonstrate the benefit of the integration of roll and pitch sensors. The propagation of the observers location will be much more accurate. Therefore, gross errors in the data association can be determined much more precisely and the data association search space will be reduced.

In close range environments autonomous systems can carry stereo and multi stereo vision systems. Comparable with range sensors, these systems are used to determine depth informations at any point in time.

One method to solve the *SLAM* task is to extend the monocular vision approaches by incorporating the new sensor information of a second camera using the same observation model. In Davison & Murray (2002) a stereo vision system is used. The Kalman filter based model represents the observers trajectory in a 2d environment and the object locations in a 3d environment.

Another method is to determine a dense depth map at a discrete point of time. The observers motion will be determined by a scan matching of at least two depth maps of the same parts of the environment. The key problem is to find correspondences between the depth maps. A popular technique to do this is the *iterative closest point* algorithm and its extensions, c. f. Besl & D.McKay (1992), Rusinkiewicz & Levoy (2001). The work of Akbarzadeh *et al.* (2006) is based on a multi stereo image sensor system. First a dense map will be obtained and used for scan matching. The map will be represented by a point cloud, obtained by a median fusion algorithm of different scans. Scan matching can be performed with 3d sonar sensors Ribas *et al.* (2006), laser scanners Brennecke *et al.* (2003) as well as radar sensors Dissanayake *et al.* (2001). The scan matching can also be performed using 2d range scans, e. g. Lu & Milios (1994).

Simple multiple scan matching will result in discrepancies arising from the inaccuracy of the matching process and the sensor noise. To incorporate the uncertainty the relative motion of the observer derived by the scan matching can be used as a measurement in a least squares solution. Nüchter (2007) showed, that the inconsistencies in this case can be minimized.

The idea of Nieto *et al.* (2005) combines scan matching and Kalman filter based SLAM. Here objects will be separated in the range data. The location of an object in the observer's coordinate system will be used as a direct observation of a landmark in a Kalman filter based approach.

2.3 Previous work with different map representations

The scope of all approaches is to acquire map information which has to be represented. In figure 2.1 different schemes of map representations are shown. First, one commonly used representation is feature maps, using points as well as line features. Euclidean coordinates are used as the usual parameterization. Montiel *et al.* (2006) introduced the inverse distance parametrization for the monocular visual SLAM problem to reduce the effect of linearization errors and to deal with points on infinity. It is shown, that the non-Gaussian distribution of a ray intersection with small parallaxes results in a bias. Related to this representation, Civera *et al.* (2007b) proposed a linearity measurement based on the disparity angle to decide when to switch from inverse distance to Euclidean representation. Trawny & Roumeliotis (2006) introduced an over represented feature initialization using a vantage point. Nearby points and points at infinity can be represented in the same way. In the work of Lemaire & Lacroix (2007) line segments and point feature as map elements are used simultaneously in a Kalman filter based approach.

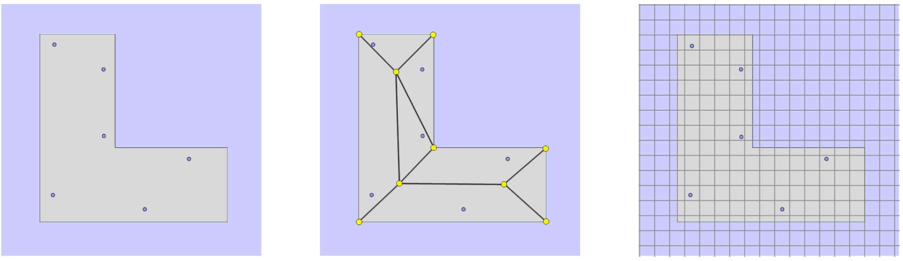


Figure 2.1: **Left:** Feature map, **Middle:** Topological map, **Right:** Grid map

In Marzorati *et al.* (2007) a solution for the integration into a Kalman filter of 3d lines and points acquired by a stereo camera system is described. The model is generated using homogeneous representations of geometric entities. Rodríguez-Losada *et al.* (2006) introduced a sub-map representation in an Kalman filter. For every new set of features its own reference coordinate system is defined. The actual location of the observer can also be transformed to this coordinate system. One positive effect is the reduction of the linearizion error, but it increases the computational costs.

Second, topological maps are typically used if the environment can be separated into distinct locations. In Kuipers *et al.* (2000) the connections between spacial semantic and localization and mapping using autonomous systems is outlined. The topological map represented by a Voronoi graph is introduced in Choset & Nagatani (2001), which also encodes metric information about the environment. The key feature of this approach is to construct the graph incrementally using a set of basic control laws. Another example using a topological graph representation has been shown by Folkesson & Christensen (2004). The approach is able to impose global constraints (e. g. loop closing) and to represent inconsistencies using an energy term of the graph nodes.

Third, grid maps can be used to represent map information. Most of these implementations are limited to 2d environments, for example Howard & Kitchen (1997), Montemerlo & Thrun (2003), Eliazar & Parr (2004), where the grid cells are signed as occluded or not occluded. Grisetti *et al.* (2006) for example extends the grid map approach using particles associated with a probability for the map occlusion. Using an elevation grid map based approach is outlined in Pfaff *et al.* (2007). Also a dimension extension to 3d grid maps (voxel maps) will be feasible, e. g. Yu & Zhang (2006) and Zask & Dailey (2009). Grid maps in general are limited in their accuracy depending on the chosen grid resolution, which will be a trade-off between memory usage and sensor accuracy.

2.4 Previous work with different ambiances

Different kinds of solution can be separated into different ambiances. Very popular are systems for indoor environments, such as Davison (2003) using a single camera approach, Harati *et al.* (2007) using laser scans or Tardós *et al.* (2002) using sonar measurements.

In an outdoor environment most systems focus on safe navigation to autonomous vehicles, e. g. Holz *et al.* (2008), Nüchter *et al.* (2006). Particularly detecting dynamical objects is one of the major problems Liao *et al.* (2003). The localization and mapping problem can be simplified using GPS.

Using unmanned aerial vehicles as a sensor platform in any kind of altitude becomes more and more affordable. Kim & Sukkarieh (2006) described an airborne platform with range, bearing and inertial sensors. A single camera system combined with GPS and inertial sensors was used in Sünderhauf *et al.* (2007) to demonstrate the capability of the sigma point Kalman Filter with inverse distance parametrization in the context of autonomous airships. Also Schlaile *et al.* (2006) presented a vision based framework for small unmanned aerial vehicles in indoor environments. Visual data can be used to achieve odometer information of a UAV proposed by Ollero *et al.* (2004) and Benítez *et al.* (2005). In case of stratospheric UAV system, e. g. Everaerts *et al.* (2004), the mapping task will be simplified into a 2d map problem.

Sonar sensors makes it possible to perform the localization and mapping problem for autonomous underwater vehicles (AUV), e. g. P. M. Newman & Rikoski (2003), Fairfield *et al.* (2006) and Ribas *et al.* (2008).

Extra-planetary missions will be the most spectacular environment for the simultaneous localization and mapping task. Primarily camera sensors will be used, because of its hardware robustness, lightweight construction and high information gain, e. g. Se *et al.* (2005). Future missions will also incorporate satellite images. For instance Li *et al.* (2005) propose a network vision system using ground image series and satellite images. The challenge is a robust, fully autonomous system for outdoor environments, which do not have any access to a global positioning system.

2.5 Previous work for large scenes

All presented approaches in the previous sections are very suitable in small areas. In case of large maps Kalman filter based algorithms shows inconsistencies between the estimated uncertainties and the parameters and becomes instable in terms of its numerical stability. Castellanos *et al.* (2004) investigated the inconsistency of a Kalman filter approach versus a local map joining. They introduced a robocentric mapping approach, where the world coordinate system is defined by the observer. It is shown, that a map-consistency improvement for this approach can be obtained. The reason is, that higher order terms of the linearization becomes smaller due to the robocentric coordinate system. Also Bailey *et al.* (2006) analyzed the consistency of Kalman filter based methods for large loops. It is pointed out, that the angular uncertainty of measurements has the overall influence to the divergence of the Kalman filter approaches. In J. A. Castellanos & Tardós (1999) a framework is introduced to deal with uncertain geometric symmetry of segments (e. g. walls) using a Kalman filter method. It is shown that symmetry information improves significantly the consistency of the filter.

Simultaneous localization and mapping for large scenes typically can be solved by a hierarchical mapping. Small local maps with high inner accuracy and consistency can be joined. In case of an unmanned aerial vehicle, classical algorithms using image pair geometry can be used to obtain an independent *photogrammetric model*. These local maps can be joined to make large maps as shown in Kanade *et al.* (2004). Clemente *et al.* (2007) also used a hierarchical map approach. For obtaining independent local maps with a Kalman filter based approach with inverse distance map representation, an overall transformation of the generated local maps will be computed to yield an optimal solution, if a loop closing is detected. Paz *et al.* (2007) proposed a technique to join local independent maps at fixed intervals using a binary tree to reduce the complexity of joining the maps. Martinez-Cantin & Castellanos (2005) showed implementation details on the integration of the sigma point Kalman filter to the SLAM problem and introduced an innovation-based consistency checking for large scale outdoor navigation using a 2d laser range sensor. In Estrada *et al.* (2005) an example for combining several local maps under the loop closing constraint is shown. All local maps are adjusted, but the inner precision of every local map does not change. Blanco *et al.* (2007) also introduce a hierarchical SLAM approach to combine small local maps with a superior Bayesian network. Sub-maps are computed by the *FAST-SLAM* algorithm. Guivant *et al.* (2004) introduce a new hybrid map representation. The global map is partitioned in *Local Triangular Regions* (LTR) based on selected landmarks, which defines independent skew symmetric coordinate systems. The observer's location is embedded in this system. The observed landmarks therefore can be updated using separated Kalman filter.

In this chapter we specified the meaning of localization, mapping and its joint problem as a time dependent process. As outlined, the goal of a simultaneous localization and mapping approach is to estimate the full posterior distribution of an observers location and its environment. In a review part we gave a broad overview about different solutions distinguishable by the parameter estimation technique, map representation and used sensor systems. It can be observed that existing approaches score well for different kinds of ambiances in small areas. However, building larger maps still seems to be challenging.

Chapter 3

Background theory

3.1 Geometric entities and transformations

In this section we will present the construction and relationship of geometric entities and basic coordinate transformation techniques. We will briefly describe the representation of rotations, the extension to motions and homographies as a general coordinate transformation in the 3d space. Using homogeneous coordinates we are able to formulate rotations, motions as well as homographies as linear transformation in a 2- and 3-dimensional projective space. For a broad introduction to homogeneous coordinates, its transformations and its historical background please refer to Heuel (2004). For an interpretation of the different writing styles of variables please refer to the notation definition.

3.1.1 Geometric entities

In this section we will present the fundamentals of algebraic projective geometry which we use in this dissertation. In a first step we introduce homogeneous coordinates to represent points and lines in the 2-dimensional space. Secondly, we present the extension to tree-dimensional space using points, planes and lines followed by construction[1] elements and incidence contradictions.

[1] The construction of an element from two elements will be indicated by the operator $\wedge$ and the intersection of two elements by $\cap$.

3.1.1.1 Points and lines in the 2d space

Homogeneous coordinates in general are invariant with respect to multiplication by a scalar factor $\lambda \neq 0$. Every Euclidean coordinate $\boldsymbol{x} = [x, y]^{\mathsf{T}}$ can be represented as a homogeneous vector

$$\mathbf{x} = \begin{bmatrix} \boldsymbol{x}_0 \\ x_h \end{bmatrix} = \begin{bmatrix} u \\ v \\ w \end{bmatrix}. \tag{3.1}$$

Normalizing the homogeneous vector by dividing $\mathbf{x}$ by w, than $\boldsymbol{x}_0$ represents the Euclidean part of $\mathbf{x}$. All points $[u, v, w]^{\mathsf{T}} \neq 0$ build the so called projective plane consisting of all points $[x, y]^{\mathsf{T}}$ of the Euclidean plane together with the points at infinity $[u, v, 0]^{\mathsf{T}}$.

Every line in in $\mathbb{R}^2$ can be represented in *Hessian normal form*

$$x \cos\phi + y \sin\phi - d = 0 \tag{3.2}$$

and has the homogeneous coordinate vector

$$\mathbf{l} = \begin{bmatrix} \boldsymbol{l}_h \\ l_0 \end{bmatrix} = \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \sqrt{a^2 + b^2} \begin{bmatrix} \cos\phi \\ \sin\phi \\ -d \end{bmatrix} \tag{3.3}$$

as long as $\sqrt{a^2 + b^2} \neq 0$. Every line with the same coordinates as a point up to an arbitrary factor $\lambda \neq 0$ is called the dual element of the point. The line at infinity is $[0, 0, 1]^{\mathsf{T}}$.

3.1.1.2 Points, planes and lines in the 3d space

In the 3-dimensional space a point $\boldsymbol{X} = [X, Y, Z]^{\mathsf{T}}$ can be represented similar to the two-dimensional space by

$$\mathbf{X} = \begin{bmatrix} \boldsymbol{X}_0 \\ X_h \end{bmatrix} = \begin{bmatrix} U \\ V \\ W \\ T \end{bmatrix}. \tag{3.4}$$

The dual element of a 3d point in homogeneous coordinates is a plane. From the implicit plane representation

$$AX + BY + CZ + D = 0 \tag{3.5}$$

or its analogon in *Hessian normal form* $\boldsymbol{X}^\mathsf{T}\boldsymbol{N} - S = 0$, with $\boldsymbol{N}$ as the unit normal vector of the plane and S as the distance from the origin, the homogeneous representation vector is defined as

$$\mathbf{A} = \begin{bmatrix} \boldsymbol{A}_h \\ A_0 \end{bmatrix} = \begin{bmatrix} A \\ B \\ C \\ D \end{bmatrix}. \tag{3.6}$$

Normalizing $\mathbf{A}$ in a way that the length of the subvector is $|\boldsymbol{A}_h| = 1$, the absolute value of the fourth element of $\mathbf{A}$ represent the distance to the origin.

Lines in $\mathbb{R}^3$ can be represented by four independent parameters, for instance by one point in the XY-plane and a direction represented by azimuth and elevation. A common representation is the Plücker representation, namely a 6d-vector containing the *Plücker* coordinates. We will use this representation, because there are mathematical connections to the homogeneous representation. One can show that the construction of a 3d line using two homogeneous 3d points $\mathbf{L} = \mathbf{X} \wedge \mathbf{Y}$ can be obtained by

$$\mathbf{L} = \begin{bmatrix} \boldsymbol{L}_h \\ \boldsymbol{L}_0 \end{bmatrix} = \begin{bmatrix} L_1 \\ L_2 \\ L_3 \\ L_4 \\ L_5 \\ L_6 \end{bmatrix} = \begin{bmatrix} X_4Y_1 - Y_4X_1 \\ X_4Y_2 - Y_4X_2 \\ X_4Y_3 - Y_4X_3 \\ X_2Y3 - Y_2X_3 \\ X_3Y_1 - Y_3X_1 \\ X_1Y_2 - Y_1X_2 \end{bmatrix} = \begin{bmatrix} X_h\boldsymbol{Y}_0 - Y_h\boldsymbol{X}_0 \\ \mathsf{S}(\boldsymbol{X}_0)\boldsymbol{Y}_0 \end{bmatrix}. \tag{3.7}$$

Because a 3d line can be defined by 4 independent parameters and the *Plücker* vector is homogeneous, obviously there is an additional constraint for all 6-vectors represents a 3d-line, the so called *Plücker* constraint:

$$\boldsymbol{L}_0^\mathsf{T}\boldsymbol{L}_h = L_1L_4 + L_2L_5 + L_3L_6 = 0. \tag{3.8}$$

The homogeneous part of the Plücker line $\boldsymbol{L}_h$ can be interpreted as the line direction vector, the Euclidean part $\boldsymbol{L}_0$ is the normal vector of a plane spanning of the 3d line and the origin.

Like the dual element of a 3d point is a plane with the same coordinate vector there exists a dual element of the Plücker line $\mathbf{L}$. This dual element is called a dual line $\overline{\mathbf{L}}$ and can be constructed using a dual operator D by

$$\overline{\mathbf{L}} = \mathsf{D}\mathbf{L} \qquad \text{with} \qquad \mathsf{D} = \begin{bmatrix} \mathit{0} & I_3 \\ I_3 & \mathit{0} \end{bmatrix}. \tag{3.9}$$

3.1.1.3 Incidence constrains and entity construction

Now we will introduce the basic formulations for checking the incidence of geometric entities and their construction. Using the *Hessian normal form* of equation (3.2) we can easily check the incidence of a 2d point $\mathbf{x}$ lies on the 2d line $\mathbf{l}$ by

$$\mathbf{l}^{\mathsf{T}}\mathbf{x} = \mathbf{x}^{\mathsf{T}}\mathbf{l} = 0. \tag{3.10}$$

Equivalently, a 3d point $\mathbf{X}$ lies on a plane $\mathbf{A}$ if

$$\mathbf{A}^{\mathsf{T}}\mathbf{X} = \mathbf{X}^{\mathsf{T}}\mathbf{A} = 0. \tag{3.11}$$

The construction of entities in the 2-dimensional space is restricted to points and lines only. Given two points $\mathbf{x}, \mathbf{y}$ the line intersect both can be constructed by

$$\mathbf{x} \wedge \mathbf{y} = \mathbf{l} = S(\mathbf{x})\mathbf{y} = S(\mathbf{y})\mathbf{x}. \tag{3.12}$$

Using the duality relationship, the intersection of two lines $\mathbf{l}$ and $\mathbf{m}$ is a point $\mathbf{x}$ and can be expressed by

$$\mathbf{l} \cap \mathbf{m} = \mathbf{x} = S(\mathbf{l})\mathbf{m} = S(\mathbf{m})\mathbf{l}. \tag{3.13}$$

In 3-dimensional space the incidence relationships and the construction of entities need to introduce some matrix expressions. Rewriting equation (3.7) to a matrix vector multiplication $\mathbf{L} = \Pi(\mathbf{X})\mathbf{Y}$ we introduce

$$\Pi(\mathbf{X}) = \begin{bmatrix} X_h I_3 & -\boldsymbol{X}_o \\ S(\boldsymbol{X}_0) & \mathbf{0} \end{bmatrix} = \begin{bmatrix} T & 0 & 0 & -U \\ 0 & T & 0 & -V \\ 0 & 0 & T & -W \\ 0 & -W & V & 0 \\ W & 0 & -U & 0 \\ -V & U & 0 & 0 \end{bmatrix}. \tag{3.14}$$

Using the plane $\mathbf{A}$ as the dual element of $\mathbf{X}$ we can construct a dual element of Π written as

$$\overline{\Pi}(\mathbf{A}) = \mathsf{D}\Pi(\mathbf{A}) = \begin{bmatrix} 0 & -C & B & 0 \\ C & 0 & -A & 0 \\ -B & A & 0 & 0 \\ D & 0 & 0 & -A \\ 0 & D & 0 & -B \\ 0 & 0 & D & -C \end{bmatrix}. \tag{3.15}$$

Additionally we will introduce a proper representation of a 3d line in a matrix expression, the so-called *Plücker matrix* representation

$$\Gamma(\mathbf{L}) = \begin{bmatrix} 0 & L_6 & -L_5 & -L_1 \\ -L_6 & 0 & L_4 & -L_2 \\ L_5 & -L_4 & 0 & -L_3 \\ L_1 & L_2 & L_3 & 0 \end{bmatrix}. \tag{3.16}$$

One can show that the dual matrix relation holds

$$\overline{\Gamma}(\mathbf{L}) = \Gamma(\overline{\mathbf{L}}) = \begin{bmatrix} 0 & L_3 & -L_2 & -L_4 \\ -L_3 & 0 & L_1 & -L_5 \\ L_2 & -L_1 & 0 & -L_6 \\ L_4 & L_5 & L_6 & 0 \end{bmatrix} \tag{3.17}$$

with the Plücker constraint in matrix expression $\Gamma^{\mathsf{T}}(\mathbf{L})\overline{\Gamma}(\mathbf{L}) = \mathbf{0}$.

Without any proof and claim of completeness we summarize the construction of main entities in table 3.1, Heuel (2004).

entities	construction	expression
points $\mathbf{x}, \mathbf{y}$	$\mathbf{l} = \mathbf{x} \wedge \mathbf{y}$	$\mathbf{l} = \mathsf{S}(\mathbf{x})\mathbf{y} = -\mathsf{S}(\mathbf{y})\mathbf{x}$
lines $\mathbf{l}, \mathbf{m}$	$\mathbf{x} = \mathbf{l} \cap \mathbf{m}$	$\mathbf{x} = \mathsf{S}(\mathbf{l})\mathbf{m} = -\mathsf{S}(\mathbf{m})\mathbf{l}$
points $\mathbf{X}, \mathbf{Y}$	$\mathbf{L} = \mathbf{X} \wedge \mathbf{Y}$	$\mathbf{L} = \Pi(\mathbf{X})\mathbf{Y} = -\Pi(\mathbf{Y})\mathbf{X}$
planes $\mathbf{A}, \mathbf{B}$	$\mathbf{L} = \mathbf{X} \cap \mathbf{Y}$	$\mathbf{L} = \overline{\Pi}^{\mathsf{T}}(\mathbf{A})\mathbf{B} = -\overline{\Pi}(\mathbf{B})\mathbf{A}$
point $\mathbf{X}$, line $\mathbf{L}$	$\mathbf{A} = \mathbf{X} \wedge \mathbf{L}$	$\mathbf{A} = \overline{\Pi}^{\mathsf{T}}(\mathbf{X})\mathbf{L} = \overline{\Gamma}^{\mathsf{T}}(\mathbf{L})\mathbf{X}$
plane $\mathbf{A}$, line $\mathbf{L}$	$\mathbf{X} = \mathbf{A} \cap \mathbf{L}$	$\mathbf{X} = \Pi^{\mathsf{T}}(\mathbf{A})\mathbf{L} = \Gamma^{\mathsf{T}}(\mathbf{L})\mathbf{A}$

Table 3.1: Construction of geometric entities

3.1.2 Rotations

In geometry rotations are one central transformation operation. In the following we will review rotation representations and some additional mathematical aspects for the 3-dimensional case we will use in this thesis. A rotation matrix R is a special, orthogonal n-dimensional linear transformation with the restrictions $|R| = 1$ and $R^{\mathsf{T}} = R^{-1}$, which results in $RR^{\mathsf{T}} = I$.

A Euclidean point $\boldsymbol{X} = \{X, Y, Z\}$ in 3d can be transformed by

$$\boldsymbol{X}' = \mathsf{R}\boldsymbol{X} \quad \text{with} \quad \mathsf{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}. \tag{3.18}$$

Elementary rotations using Euler angles denoted as ω, ϕ, κ around the three axis X, Y, Z of a right hand 3d-coordinate system are

$$\mathsf{R}_1(\omega) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\omega & -\sin\omega \\ 0 & \sin\omega & \cos\omega \end{bmatrix} \tag{3.19}$$

$$\mathsf{R}_2(\phi) = \begin{bmatrix} \cos\phi & 0 & \sin\phi \\ 0 & 1 & 0 \\ -\sin\phi & 0 & \cos\phi \end{bmatrix} \tag{3.20}$$

$$\mathsf{R}_3(\kappa) = \begin{bmatrix} \cos\kappa & -\sin\kappa & 0 \\ \sin\kappa & \cos\kappa & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{3.21}$$

In this dissertation we define the concatenation of the rotation by left multiplication

$$\mathsf{R}(\omega, \phi, \kappa) = \mathsf{R}_3(\kappa)\mathsf{R}_2(\phi)\mathsf{R}_1(\omega) = \begin{bmatrix} c_\kappa c_\phi & -s_\kappa c_\omega + c_\kappa s_\phi s_\omega & s_\kappa s_\omega + c_\kappa s_\phi c_\omega \\ s_\kappa c_\phi & c_\kappa c_\omega + s_\kappa s_\phi s_\omega & -c_\kappa s_\omega + s_\kappa s_\phi c_\omega \\ -s_\phi & c_\phi s_\omega & c_\phi c_\omega \end{bmatrix}. \tag{3.22}$$

Note, the order of the concatenation is crucial for an interpretation of an estimated rotation. Instead of the Euler angle representation, a rotation in 3d-space can also be defined by a rotation axis $\boldsymbol{n}$ through the origin with $|\boldsymbol{n}| = 1$ and a rotation angle α. The corresponding rotation matrix is

$$\mathsf{R}_{n,\alpha} = \cos\alpha \ \mathsf{I}_3 + (1 - \cos\alpha)\mathsf{D}_n + \sin\alpha \ \mathsf{S}(\boldsymbol{n}) \tag{3.23}$$

with I_3 as the identity matrix, $\mathsf{D}_n = \boldsymbol{n}\boldsymbol{n}^{\mathsf{T}}$ as the dyadic product of $\boldsymbol{n}$ and

$$\mathsf{S}(\boldsymbol{n}) = \begin{bmatrix} 0 & -n_3 & n_2 \\ n_3 & 0 & -n_1 \\ -n_2 & n_1 & 0 \end{bmatrix} \tag{3.24}$$

as the skew-symmetric matrix of $\boldsymbol{n}$. From a given rotation matrix we can compute the angle and rotation vector in the following way. Given the vector $\boldsymbol{a}$ from the elements of R

$$\boldsymbol{a} = -\begin{bmatrix} r_{23} - r_{32} \\ r_{31} - r_{13} \\ r_{12} - r_{21} \end{bmatrix} \tag{3.25}$$

the rotation angle is given by

$$\alpha = \text{atan2}\ (|\boldsymbol{a}|, tr(\mathsf{R}) - 1) \tag{3.26}$$

and the rotation vector $\boldsymbol{n}$ by

$$\boldsymbol{n} = \frac{\boldsymbol{a}}{|\boldsymbol{a}|} \qquad \text{if} \quad |\boldsymbol{a}| \neq 0. \tag{3.27}$$

Rotations represented by Euler angle as well as axis and angle require trigonometric functions which may be a disadvantage. The quaternion representation introduced by *W. R. Hamilton* is closely related to the axis and angle representation, but there is no need for trigonometric terms. Quaternions are also closely related to complex numbers and they define their own algebra. Furthermore the quaternion representation for rotations is unique except for a sign and shows no singularities. A quaternion consists of a scalar part $q = q_0$ and a vector part $\boldsymbol{q} = \{q_1, q_2, q_3\}$. We can write a quaternion as a 4-vector

$$\mathbf{q} = \begin{bmatrix} q \\ \boldsymbol{q} \end{bmatrix} = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix}. \tag{3.28}$$

The corresponding rotation matrix can be computed by

$$\mathsf{R}(\mathbf{q}) = \frac{1}{q_0^2 + q_1^2 + q_2^2 + q_3^2} \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1 q_2 - q_0 q_3) & 2(q_1 q_3 + q_0 q_2) \\ 2(q_2 q_1 + q_0 q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2 q_3 - q_0 q_1) \\ 2(q_3 q_1 - q_0 q_2) & 2(q_3 q_2 + q_0 q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix}. \tag{3.29}$$

We observe, that $\mathsf{R}(\mathbf{q})$ is invariant with respect to a multiplication of $\mathbf{q}$ with a scalar $\neq 0$. It can be shown, that the vector part $\boldsymbol{q}$ is parallel to the rotation axis $\boldsymbol{n}$. Using the unit quaternion $|\mathbf{q}| = 1$ the relationship between quaternions and the axis-angle representation is defined by

$$\mathbf{q} = \begin{bmatrix} \cos\frac{\alpha}{2} \\ \sin\frac{\alpha}{2}\boldsymbol{n} \end{bmatrix}. \tag{3.30}$$

We can easily derive the concatenation of two rotations $\mathbf{q}$ and $\mathbf{r}$ represented by quaternion as a quaternion multiplication $\mathbf{p} = \mathbf{q}\mathbf{r}$ defined as the matrix vector multiplication

$$\mathbf{p} = \mathbf{q}\mathbf{r} = \Upsilon(\mathbf{q})\mathbf{r} = \breve{\Upsilon}(\mathbf{r})\mathbf{q} \tag{3.31}$$

with the 4×4 matrix

$$\Upsilon(\mathbf{q}) = \begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & -q_3 & q_2 \\ q_2 & q_3 & q_0 & -q_1 \\ q_3 & -q_2 & q_1 & q_0 \end{bmatrix} \tag{3.32}$$

and

$$\breve{\Upsilon}(\mathbf{r}) = \begin{bmatrix} r_0 & -r_1 & -r_2 & -r_3 \\ r_1 & r_0 & r_3 & -r_2 \\ r_2 & -r_3 & r_0 & r_1 \\ r_3 & r_2 & -r_1 & r_0 \end{bmatrix} . \tag{3.33}$$

In case of $|\mathbf{q}| = 1$ the inverse element of $\boldsymbol{q}$ is defined by

$$\mathbf{q}^{-1} = \begin{bmatrix} q \\ -\boldsymbol{q} \end{bmatrix} \qquad \Upsilon_{\mathbf{q}^{-1}} = \Upsilon_{\mathbf{q}}^{-1} \tag{3.34}$$

or as matrix-vector multiplication

$$\mathbf{q}^{-1} = V\mathbf{q} \qquad \text{with} \qquad V = \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0}^\mathsf{T} & -I_3 \end{bmatrix} . \tag{3.35}$$

For a small rotation $\mathbf{q}^k$ which depends on k concatenations of a small rotation $\mathbf{q}$ around the same axis a sufficient approximation can be obtained by

$$\mathbf{q}^k \approx \begin{bmatrix} 1 \\ k\boldsymbol{q} \end{bmatrix} \tag{3.36}$$

or written as matrix vector product

$$\mathbf{q}^k \approx U_k\mathbf{q} \qquad \text{with} \qquad U_k = \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0}^\mathsf{T} & kI_3 \end{bmatrix} . \tag{3.37}$$

This approximation is very useful for the representation of small angular velocities and accelerations we will use in chapter 4.

In the following we will show for the special case of small circular motions, that the quaternion concatenation can be used to obtain a rotation from angular velocity. First, lets assume an angle α is a function of time t depends on an initial angle α_0 and an angular velocity $\dot{\alpha}$, so

$$\alpha_t = \alpha_0 + \dot{\alpha}t. \tag{3.38}$$

Substitute α_t into the unit quaternion representation in equation (3.30) we get

$$\mathbf{q}(t) = \begin{bmatrix} \cos(\frac{\alpha_0+\dot{\alpha}t}{2}) \\ \sin(\frac{\alpha_0+\dot{\alpha}t}{2})\boldsymbol{n} \end{bmatrix} \tag{3.39}$$

as the quaternion as a function of time. Using the theorems

$$\begin{aligned} \cos(x+y) &= \cos(x)\cos(y) - \sin(x)\sin(y) && (3.40) \\ \sin(x+y) &= \sin(x)\cos(y) + \cos(x)\sin(y) && (3.41) \end{aligned}$$

and reformulate equation (3.39) we get

$$\mathbf{q}(t) = \begin{bmatrix} \cos(\frac{\alpha_0}{2})\cos(\frac{\dot{\alpha}t}{2}) - \sin(\frac{\alpha_0}{2})\sin(\frac{\dot{\alpha}t}{2}) \\ \left(\sin(\frac{\alpha_0}{2})\cos(\frac{\dot{\alpha}t}{2}) + \cos(\frac{\alpha_0}{2})\sin(\frac{\dot{\alpha}t}{2})\right)\boldsymbol{n} \end{bmatrix}. \tag{3.42}$$

Evaluate (3.42) at $t_0 = 0$ and $\alpha_0 = 0$ and approximate $\sin(\frac{\dot{\alpha}t}{2}) \approx \frac{\dot{\alpha}t}{2}$ and $\cos(\frac{\dot{\alpha}t}{2}) \approx 1$ we get

$$\mathbf{q}(t) = \begin{bmatrix} 1 \\ (\frac{\dot{\alpha}t}{2})\boldsymbol{n} \end{bmatrix} = \begin{bmatrix} 1 \\ (\frac{\dot{\alpha}}{2}\boldsymbol{n})t \end{bmatrix}, \tag{3.43}$$

which can be expressed using equations (3.36) and (3.37) with $k = t$ and $\boldsymbol{q} = \frac{\dot{\alpha}}{2}\boldsymbol{n}$. A derivation using a Taylor expansion and incorporation of acceleration can be found in the appendix in section A.1.

3.1.3 Motions and homographies

In many geometric tasks linear transformations are necessary. The projective transformation, also known as a homography, can be applied to a homogeneous point as a 4×4 matrix multiplication with 15 degrees of freedom. As the homography H is homogeneous, a normalization with respect to the last element is useful.

$$\mathbf{X}' = \mathsf{H}\mathbf{X} \rightarrow \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}' = \begin{bmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ m & n & o & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}. \tag{3.44}$$

This general linear transformation can be specialized to the motion of coordinate systems. In the Euclidean 3d space a motion is defined by a rotation R and a translation $\boldsymbol{T}$ of a 3-dimensional point $\boldsymbol{X}$

$$\boldsymbol{X}' = \mathsf{R}\boldsymbol{X} + \boldsymbol{T} \tag{3.45}$$

and its inversion

$$\boldsymbol{X} = \mathsf{R}^{\mathsf{T}}(\boldsymbol{X} - \boldsymbol{T}). \tag{3.46}$$

Using homogeneous coordinates for the point $\mathbf{X} = [X, Y, Z, 1]^{\mathsf{T}}$ we can rewrite the motion in (3.45) to a linear mapping using a matrix vector multiplication

$$\mathbf{X}' = \mathsf{M}\mathbf{X} \tag{3.47}$$

with the homogeneous motion matrix

$$\mathsf{M} = \begin{bmatrix} \mathsf{R} & \boldsymbol{T} \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix}. \tag{3.48}$$

From equation (3.46) we can show that the inversion of the motion can be computed by

$$\mathsf{M}^{-1} = \begin{bmatrix} \mathsf{R}^{\mathsf{T}} & -\mathsf{R}^{\mathsf{T}}\boldsymbol{T} \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix} \tag{3.49}$$

and so is $\mathsf{M}\mathsf{M}^{-1} = \mathsf{I}_4$. We are able to concatenate motion similar to rotations.

Example: As an example for the concatenation of motions we will rotate a point-cloud $\mathbf{X}_i$ around its centroid T_c. First we translate the centroid to the origin, rotate the point-cloud and translate back by using the motion matrix

$$\mathsf{M} = \mathsf{M}_{T_c}\mathsf{M}_R\mathsf{M}_{T_c}^{-1} = \begin{bmatrix} \mathsf{I}_3 & \boldsymbol{T}_c \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix} \begin{bmatrix} \mathsf{R} & \mathbf{0} \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix} \begin{bmatrix} \mathsf{I}_3 & -\boldsymbol{T}_c \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix} = \begin{bmatrix} \mathsf{R} & -\mathsf{R}\boldsymbol{T}_c + \boldsymbol{T}_c \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix}. \tag{3.50}$$

A second specialization is the 7-parameter *Helmert*-transformation (similarity transformation) as a concatenation of a motion and a scale factor. This restricted homography can be derived by the following concatenation:

$$\mathsf{H} = \mathsf{H}_s\mathsf{M}_R\mathsf{M}_T = \begin{bmatrix} \lambda\mathsf{I}_3 & \mathbf{0} \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix} \begin{bmatrix} \mathsf{R}^{\mathsf{T}} & \mathbf{0} \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix} \begin{bmatrix} \mathsf{I}_3 & -\boldsymbol{T} \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix}. \tag{3.51}$$

In the same manner also a 3d affine transformation with 12 degrees of freedom can be represented. The rotation, motion and the homography can also be applied to 2d space coordinates by canceling the third row and column of the transformation matrices. As an example the 2d affine transformation can be represented by a matrix multiplication as a concatenation of base transformations, namely a translation H_t, a rotation H_r, a scaling H_s with individual scales for both vector components and an unsymmetric shearing H_{sh}

$$\mathbf{x}' = \mathsf{H}\mathbf{x} = \mathsf{H}_t\mathsf{H}_r\mathsf{H}_s\mathsf{H}_{sh}\mathbf{x} = \begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix} \mathbf{x}. \tag{3.52}$$

For an overview of possible transformations have a look to McGlone *et al.* (2004) page 143ff. Rotations in the 2d space are embedded in the 3d space as a rotation only around the Z-Axis using equation (3.21).

3.2 Basic image geometry

In this section we will briefly introduce the basic issues that deal with images. The term image will be used in a broad context. Usually an image is a mapping of the 3d world to a 2d image using a camera. There are numbers of real world camera with different constructions and projection principles Sturm *et al.* (2006). In this work we will assume cameras with a single projection center and a perspective projection. However, for every kind of cameras given an image point $\boldsymbol{x}'$ it is possible to reconstruct at least one projection ray, which is a 3d line $\mathbf{L}(\boldsymbol{x}')$ from an object point to an arbitrary projection center given by a measured image point $\boldsymbol{x}'$. This principle allows to assign the geometry reconstruction principle in this work to all different kinds of camera models.

3.2.1 The geometry of the single image

In general, we distinguish between the interior and exterior orientation of a camera. The interior orientation describes the transformation of an image point $\boldsymbol{x}'$ to the projection ray in a fixed camera coordinate system ${}^{c}S$ and vice versa, the exterior orientation describes the location of the camera system ${}^{c}S$ in a world system ${}^{w}S$.

The exterior orientation can be represented by the position and angular orientation of the camera. The interior orientation of perspective cameras can be modeled using an affine sensor. This is caused from the principle of the manufacturing of sensors and/or scanner technologies. Moreover, caused by lens distortions and different physically effects non-linear distortion terms have to be applied to correct them.

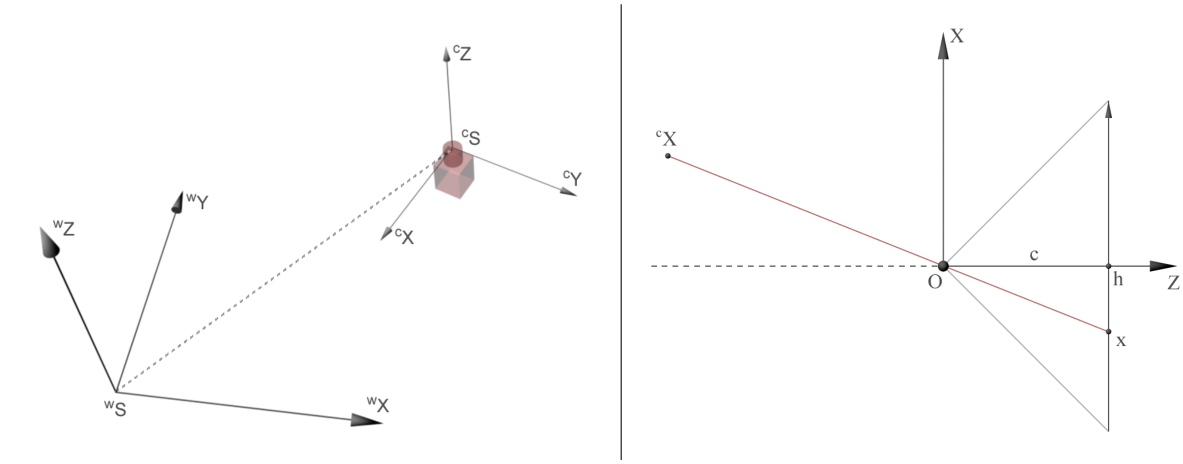


Figure 3.1: **Left:** Exterior orientation definition **Right:** Central perspective mapping scheme

A given 3d point ${}^c\boldsymbol{X}$ in the camera coordinate system can be projected into the image plane resulting in the ideal image point $\bar{\mathbf{x}}'$ using the intercept theorems

$$ {}^c\bar{\boldsymbol{x}} = c\frac{{}^c\boldsymbol{X}}{{}^c\boldsymbol{Z}} \quad \text{and} \quad {}^c\bar{\boldsymbol{y}} = c\frac{{}^c\boldsymbol{Y}}{{}^c\boldsymbol{Z}}. \tag{3.53} $$

This can be rewritten using homogeneous coordinates

$$ {}^c\bar{\mathbf{x}} = \begin{bmatrix} \bar{u} \\ \bar{v} \\ \bar{w} \end{bmatrix} = {}^c\bar{\mathsf{P}}_{\boldsymbol{c}}\,{}^c\mathbf{X} = \begin{bmatrix} c & 0 & 0 & 0 \\ 0 & c & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} {}^cX \\ {}^cY \\ {}^cZ \\ 1 \end{bmatrix}. \tag{3.54} $$

Here c is the principal distance of the projection center to the image plane. The ideal image point $\bar{\mathbf{x}}'$ can be mapped using an affine sensor model by a 2d homography for an affine sensor

$$ \mathbf{x}' = \mathsf{H}_c\,{}^c\bar{\mathbf{x}}' = \begin{bmatrix} 1 & s & x_h \\ 0 & 1+m & y_h \\ 0 & 0 & 1 \end{bmatrix} {}^c\bar{\mathbf{x}}. \tag{3.55} $$

Using the position and orientation of the camera in a world coordinate system ${}^wS$ we can transform a homogeneous 3d-point $\mathbf{X}$ applying the inverse motion matrix from equation

(3.48). We obtain a point ${}^c\mathbf{X}$ in the camera system by

$$ {}^c\mathbf{X} = {}^c\mathsf{M}\mathbf{X} = \begin{bmatrix} R^\mathsf{T} & -R^\mathsf{T}\boldsymbol{T} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{X} \\ 1 \end{bmatrix}. \tag{3.56} $$

Combining the equations (3.54), (3.55) and (3.56) and substituting the commonly named calibration matrix

$$ \mathsf{K} = \begin{bmatrix} c & sc & x_h \\ 0 & (1+m)c & y_h \\ 0 & 0 & 1 \end{bmatrix} \tag{3.57} $$

we get the projection of a 3d point in a world coordinate system to the image plane by

$$ \mathbf{x}' = \mathsf{P}\mathbf{X} = \mathsf{K}R^\mathsf{T}[I_3 \mid -\boldsymbol{T}]\mathbf{X}. \tag{3.58} $$

The Euclidean coordinates of the image point can be derived by normalizing $\mathbf{x}'$. Assuming zero shear and scale differences we get the collinearity equations

$$ x' = c\frac{r_{11}(X - X_T) + r_{21}(Y - Y_T) + r_{31}(Z - Z_T)}{r_{13}(X - X_T) + r_{23}(Y - Y_T) + r_{33}(Z - Z_T)} + x_h \tag{3.59} $$

$$ y' = c\frac{r_{12}(X - X_T) + r_{22}(Y - Y_T) + r_{32}(Z - Z_T)}{r_{13}(X - X_T) + r_{23}(Y - Yv) + r_{33}(Z - Z_T)} + y_h \tag{3.60} $$

with $\boldsymbol{T} = [X_T, Y_T, Z_T]^\mathsf{T}$ as the camera projection center and r_{ij} are the entries in R. We can see from equation (3.54) that the projection is not invertible, because we lose the information about the depth of the object point. Splitting the projection matrix into two parts

$$ \mathsf{P} = [\mathsf{H}_\infty|\mathbf{h}] = [\mathsf{K}R^\mathsf{T} \mid -\mathsf{K}R^\mathsf{T}\boldsymbol{T}] \tag{3.61} $$

as H_∞ map points at the plane of infinity into the image plane, we can derive the direction to a projected object point in the camera system by

$$ \mathbf{d} = \mathsf{H}_\infty^{-1}\mathbf{x}'. \tag{3.62} $$

Assuming a distance d to the object points $\boldsymbol{X}$ we can calculate its location by

$$ \boldsymbol{X} = d\frac{\mathbf{d}}{|\mathbf{d}|} + \boldsymbol{T}. \tag{3.63} $$

So far, from the principle of linear transformations the camera model is straight-line-preserving. Real cameras are not straight line preserving, due to several physical influences such as lens distortions or refraction. In the literature different non-linear distortion models are available.

For a overview please c. f. Abraham (1999). A general way to represent all these models is by using a distortion lockup-table. This means, for every real image point $\boldsymbol{x}'$ exists a distortion vector to get the straight-line-preserving image point coordinate and vice versa. The distortion lookup-table can be obtained in a camera calibration process. A measured Euclidean image point $\boldsymbol{x}^m$ can be corrected by

$$\boldsymbol{x}' = \boldsymbol{x}^m + \Delta\boldsymbol{x}(\boldsymbol{x}^m) \tag{3.64}$$

to get image coordinates following the model above. Typically, for non-integer coordinates the distortion are interpolated.

3.2.2 The geometry of the image pair

In this section we introduce the basic concept for 3d geometry retrieval from two images. For a general introduction we refer to Hartley & Zisserman (2000).

In general the two cameras have different exterior and interior orientations. In the following we assume to know the interior orientation of the cameras, which can be obtained by a previous calibration Abraham (1999). The proposed approaches are useful for consistency checks of automatic feature measurements which we will use in subsequent algorithms.

Forward intersection Knowing the interior and exterior orientation of at least two cameras given by their projection matrices P we can derive the 3d point coordinate as the intersection of the projection rays from corresponding image measurements $\mathbf{x}$ in both images. This can be formulated in an equation system as follows:

The incidence of the projected point $\mathbf{X}$ can be implicitly rewritten as

$$\mathsf{S}(\mathbf{x})\mathsf{P}\mathbf{X} = \mathbf{0}. \tag{3.65}$$

Using two rows[2] of (3.65) for two cameras and stack them to

$$\underbrace{\begin{bmatrix} \mathsf{S}(\mathbf{x}')\mathsf{P}' \\ \mathsf{S}(\mathbf{x}'')\mathsf{P}'' \end{bmatrix}}_{\underset{4\times4}{\mathsf{A}}} \mathbf{X} \overset{!}{=} \mathbf{0} \tag{3.66}$$

we can derive $\mathbf{X}$ as an approximate solution of (3.66) using for instance the singular value decomposition as a method of an equation solver. With more cameras two additional lines can be added to A for each additional ray.

[2] Only two rows of equation (3.65) are independent.

Coplanarity constraint As shown in figure 3.2 the image points $\mathbf{x}'$ and $\mathbf{x}''$ and the object point $\mathbf{X}$ are coplanar. This plane $\mathbf{A}$ intersects the image planes in two lines $\mathbf{l}'$ and $\mathbf{l}''$. Obviously, the corresponding point to $\mathbf{x}'$ lies on the line $\mathbf{l}''$. It can be shown, that this constraint can be formulated as

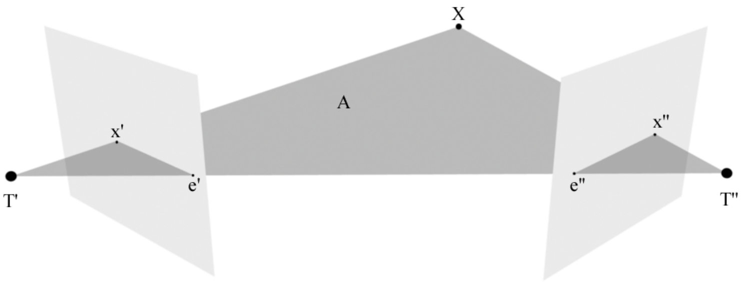


Figure 3.2: Epipolar geometry, $\boldsymbol{x}'$ and $\boldsymbol{x}''$ are the images of a 3d object point $\boldsymbol{X}$, the projection centers of the two cameras are $\boldsymbol{T}'$ and $\boldsymbol{T}''$ and their images (epipole) in the opposite cameras are $\boldsymbol{e}'$ and $\boldsymbol{e}''$. The 2d line constructed from the image point $\boldsymbol{x}$ and the epipole $\boldsymbol{e}$ is the *epipolar line*.

$$\mathbf{x}'^{\mathsf{T}} \mathsf{K}'^{-\mathsf{T}} R' S(\mathbf{b}) R''^{\mathsf{T}} \mathsf{K}''^{-1} \mathbf{x}'' = 0 \tag{3.67}$$

with $\mathbf{b}$ as the base vector between the projection centers of the two cameras. Substitute ${}^{c}\overline{\mathbf{x}}' = \mathsf{K}^{-1}\mathbf{x}'$ as the normalized image coordinates and $\mathsf{E} = R' S(\mathbf{b}) R''^{\mathsf{T}}$ as the essential matrix we get

$$ {}^{c}\overline{\mathbf{x}}'^{\mathsf{T}} \mathsf{E} \, {}^{c}\overline{\mathbf{x}}'' = 0. \tag{3.68}$$

Note, this constraint also holds for multiplying the equation with any scalar, so the coplanarity equation is independent of the length of the base vector. Fixing the world coordinate system at the first camera, we can easily see that the essential matrix is defined by five parameters. These five parameters, the base direction and the rotation between the first and the second camera represents the relative orientation. Using five arbitrary point correspondences we can derive the relative orientation using the algorithm of Nistér (2004). In case of a given relative orientation represented by E the correspondence of the image measurements can be checked using equation (3.68). The corresponding image point has to be on the corresponding epipolar line.

Homography induced by a planar object As introduced in section 3.1.3 the concatenation of homographies is also a homography. The mapping between two images observing the same plane is then also a homography. In a world coordinate system at the first camera the mapping from the first to the second image using the homogeneous plane representation $\mathbf{A} = [\boldsymbol{A}_h, A_0]^\mathsf{T}$ is given by

$$\mathbf{x}'' = \mathsf{K}'\mathsf{H}_p\mathsf{K}''^{-1}\mathbf{x}' = \mathsf{H}\mathbf{x}' \qquad \text{and} \qquad \mathsf{H}_p = R'' + \frac{R''\boldsymbol{T}''\boldsymbol{A}_h^\mathsf{T}}{A_0}. \tag{3.69}$$

The homography H can be derived using at least four corresponding image points. Using the *Kroenecker product*

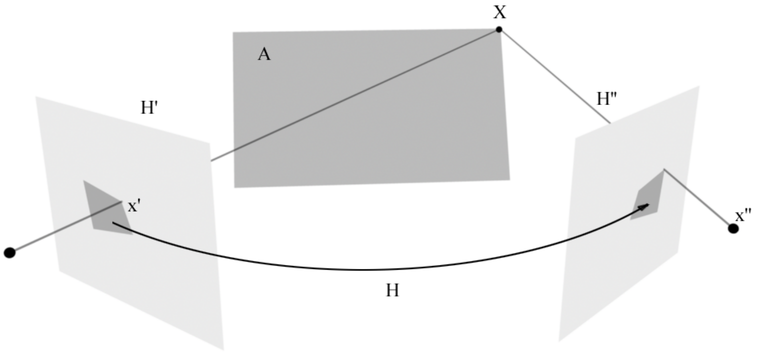


Figure 3.3: Homography induced by a plane

$$ABC = (C^\mathsf{T} \otimes A)\mathrm{vec}(B) \tag{3.70}$$

we can rewrite equation (3.69) to

$$\mathbf{0} = \mathbf{x}'' \times \mathsf{H}\mathbf{x}' = S(\mathbf{x}'')\mathsf{H}\mathbf{x}' = (S(\mathbf{x}'') \otimes \mathbf{x}')^\mathsf{T}\mathrm{vec}(\mathsf{H}). \tag{3.71}$$

Note, that only two of these constraints are independent. Using the homogeneous coordinates for corresponding points in equation (3.71) we obtain

$$\begin{bmatrix} 0 & 0 & 0 & -w''u' & -w''v' & w''w' & v''u' & v''v' & v''w' \\ w''u' & w''v' & w''w' & 0 & 0 & 0 & -u''u' & -u''v' & -u''w' \end{bmatrix} \mathrm{vec}(\mathsf{H}) = \mathbf{0}. \tag{3.72}$$

Collecting these equations for at least four points we finally obtain an equation system

$$A\mathrm{vec}(\mathsf{H}) = \mathbf{0}, \tag{3.73}$$

that we can solve with respect to the elements of H.

The proposed algorithms starting from equation (3.66) assume independent and equal measurement accuracy. Here we neglected statistical conditions, which can be easily introduced by weights for the observations. The accuracy of the obtained parameters can be derived using error propagation. Furthermore, the extension for the case of image blocks is possible. To consider the different uncertainty of observations and achieve multi-image integration we propose a least square solution, which will be introduced in the next section.

3.3 Least squares

Finding a set of parameters of a model given a set of observations is known as an optimization problem. To do this, the least square method is a very effective numerical method. The proposed algorithms are embedded in the broad context of probability theory. For linear relationships of Gaussian distributed observations and unknown parameters, the least squares method leads to *best unbiased estimators*. In the physical world, measurements can well be approximated by Gaussian distributions. Because least square methods approximate probability density function by Gaussian distributions, least squares methods becomes very popular.

In this section we will derive various forms of functional models. This derivation follows the concept of specialization of a universal form to specific specializations. These derivations can be found in several publications, for instance Mikhail & Ackermann (1976), Koch (1988), Niemeier (2001) or Förstner & Wrobel (2004). Our motivation to derive these algorithms again lies in a novel derivation of recursive least square estimation in the subsequent section. Furthermore we will present standard methods for outlier detection and robustification as well as a technique for parameter reduction.

3.3.1 Least squares optimization

Optimizing linear Gauss-Helmert-model with constraints At the beginning we start with the definition of two linear constraints

$$\boldsymbol{c}_g = \mathsf{A}\widehat{\boldsymbol{p}} + \mathsf{B}^{\mathsf{T}}(\widehat{\boldsymbol{z}} - \boldsymbol{z}) \tag{3.74}$$

$$\boldsymbol{c}_h = \mathsf{H}^{\mathsf{T}}\widehat{\boldsymbol{p}} \tag{3.75}$$

where $\widehat{\boldsymbol{p}}$ of size U is the parameter vector to be optimized. The contradiction vectors $\boldsymbol{c}_g$ and $\boldsymbol{c}_h$ of size G and H therefore depend on the estimated parameter vector as well as on the fitted observations $\widehat{\boldsymbol{z}}$ with size N. Therefore, the linear model in A has the size $G \times U$, H has the size $U \times H$ and B has the size $N \times G$. The basic concept of the least square method is to find $\widehat{\boldsymbol{p}}$ by minimizing the function

$$\Omega^2 = \frac{1}{2}\widehat{\boldsymbol{v}}^{\mathsf{T}} \mathsf{C}_{zz}^{-1} \widehat{\boldsymbol{v}} = \frac{1}{2}(\widehat{\boldsymbol{z}} - \boldsymbol{z}) \mathsf{C}_{zz}^{-1} (\widehat{\boldsymbol{z}} - \boldsymbol{z})^{\mathsf{T}} \tag{3.76}$$

with $\boldsymbol{z}$ as the observation vector with its covariance matrix C_{zz} and $\widehat{\boldsymbol{z}}$ as the estimated observation vector under the constrains in (3.74) and (3.75). To incorporate the contradictions it is necessary that we minimize

$$\Omega^2 = \frac{1}{2}(\widehat{\boldsymbol{z}} - \boldsymbol{z}) \mathsf{C}_{zz}^{-1} (\widehat{\boldsymbol{z}} - \boldsymbol{z})^{\mathsf{T}} + \boldsymbol{\lambda}^{\mathsf{T}} (\mathsf{A}\widehat{\boldsymbol{p}} + \mathsf{B}^{\mathsf{T}}(\widehat{\boldsymbol{z}} - \boldsymbol{z}) - \boldsymbol{c}_g) + \boldsymbol{\mu}^{\mathsf{T}} (\mathsf{H}^{\mathsf{T}} \widehat{\boldsymbol{p}} - \boldsymbol{c}_h) \tag{3.77}$$

by introducing the Lagrangian multipliers $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$. To find the minimum of this function we set the first derivation with respect to all parameters to zero. We obtain four equations

$$\left(\frac{\partial \Omega}{\partial \widehat{\boldsymbol{z}}}\right)^{\mathsf{T}} = \mathsf{C}_{zz}^{-1} \widehat{\boldsymbol{v}} + \mathsf{B}\boldsymbol{\lambda} = \mathbf{0} \tag{3.78}$$

$$\left(\frac{\partial \Omega}{\partial \widehat{\boldsymbol{p}}}\right)^{\mathsf{T}} = \mathsf{A}^{\mathsf{T}} \boldsymbol{\lambda} + \mathsf{H}\boldsymbol{\mu} = \mathbf{0} \tag{3.79}$$

$$\left(\frac{\partial \Omega}{\partial \lambda}\right)^{\mathsf{T}} = \mathsf{A}\widehat{\boldsymbol{p}} + \mathsf{B}^{\mathsf{T}} \widehat{\boldsymbol{v}} - \boldsymbol{c}_g = \mathbf{0} \tag{3.80}$$

$$\left(\frac{\partial \Omega}{\partial \mu}\right)^{\mathsf{T}} = -\boldsymbol{c}_h + 2\mathsf{H}^{\mathsf{T}} \widehat{\boldsymbol{p}} = \mathbf{0} \tag{3.81}$$

Solving (3.78) with respect to the residuals $\widehat{\boldsymbol{v}}$, substitute $\widehat{\boldsymbol{v}}$ into (3.80) and solve with respect to $\boldsymbol{\lambda}$ we get

$$\boldsymbol{\lambda} = (\mathsf{B}^{\mathsf{T}} \mathsf{C}_{zz} \mathsf{B})^{-1} (\mathsf{A}\widehat{\boldsymbol{p}} - \boldsymbol{c}_g) \tag{3.82}$$

and for the residuals

$$\widehat{\boldsymbol{v}} = -\mathsf{C}_{zz} \mathsf{B} (\mathsf{B}^{\mathsf{T}} \mathsf{C}_{zz} \mathsf{B})^{-1} (\mathsf{A}\widehat{\boldsymbol{p}} - \boldsymbol{c}_g). \tag{3.83}$$

The sub-term $\mathsf{B}^{\mathsf{T}} \mathsf{C}_{zz} \mathsf{B}$ can be interpreted as a projection of the covariance of the observations to the space of the contradictions. The inverse can only be computed if the column rank of B is N. This is a special case of the general Gauss-Helmert model. In the following we will assume our model guarantees full rank of the sub-term. Now, substitute $\boldsymbol{\lambda}$ into (3.79) and use equation (3.81) we can arrange these two equations to the normal equation system

$$\begin{bmatrix} \mathsf{A}^{\mathsf{T}} (\mathsf{B}^{\mathsf{T}} \mathsf{C}_{zz} \mathsf{B})^{-1} \mathsf{A} & \mathsf{H} \\ \mathsf{H}^{\mathsf{T}} & \mathsf{0} \end{bmatrix} \begin{bmatrix} \widehat{\boldsymbol{p}} \\ \boldsymbol{\mu} \end{bmatrix} = \begin{bmatrix} \mathsf{A}^{\mathsf{T}} (\mathsf{B}^{\mathsf{T}} \mathsf{C}_{zz} \mathsf{B})^{-1} \boldsymbol{c}_g \\ \boldsymbol{c}_h \end{bmatrix} \tag{3.84}$$

The inverse of the left side of the normal equation system is the theoretical covariance matrix of the parameter vector, so

$$\begin{bmatrix} N & H \\ H^\mathsf{T} & 0 \end{bmatrix}^{-1} = \begin{bmatrix} C_{\widehat{p}\widehat{p}} & S \\ S^\mathsf{T} & T \end{bmatrix} \tag{3.85}$$

and can only be computed if the stacked matrix $[A^\mathsf{T} H]^\mathsf{T}$ has full column rank. Substituting the solution for $\widehat{\boldsymbol{p}}$ into (3.83) we can compute the variance factor

$$\widehat{\sigma}_0^2 = \frac{\widehat{\boldsymbol{v}}^\mathsf{T} C_{zz}^{-1} \widehat{\boldsymbol{v}}}{G + H - U} \tag{3.86}$$

with G as the number of constraints, H as the number of contradictions and U as the number of unknown parameters inside the vector $\boldsymbol{p}$. The estimated covariance matrix for the unknown parameters is therefore

$$\widehat{C}_{\widehat{p}\widehat{p}} = \widehat{\sigma}_0^2 C_{\widehat{p}\widehat{p}} \tag{3.87}$$

By error propagation of (3.83) we can show, that the covariance matrix of the residuals is

$$C_{\widehat{v}\widehat{v}} = C_{zz} B (B^\mathsf{T} C_{zz} B)^{-1} (I - A C_{\widehat{p}\widehat{p}} A^\mathsf{T} (B^\mathsf{T} C_{zz} B)^{-1}) B^\mathsf{T} C_{zz} \tag{3.88}$$

c. f. Mikhail & Ackermann (1976) p. 197.

Optimizing Nonlinear Gauss-Helmert-Model with constraints Now let us assume we have a nonlinear model with the nonlinear implicit contradictions and constrains according to equations (3.74) and (3.75)

$$\boldsymbol{g}(\widetilde{\boldsymbol{p}}, \widetilde{\boldsymbol{z}}) = \mathbf{0} \tag{3.89}$$

$$\boldsymbol{h}(\widetilde{\boldsymbol{p}}) = \mathbf{0} \tag{3.90}$$

These equations hold for the true parameters $\widetilde{\boldsymbol{p}}$ and observations $\widetilde{\boldsymbol{z}}$. Using estimated parameters and observations the following system holds

$$\boldsymbol{g}(\widehat{\boldsymbol{p}}, \widehat{\boldsymbol{z}}) = \mathbf{0} \tag{3.91}$$

$$\boldsymbol{h}(\widehat{\boldsymbol{p}}) = \mathbf{0} \tag{3.92}$$

with $\widehat{\boldsymbol{z}} = \boldsymbol{z} + \widehat{\boldsymbol{v}}$. Approximate these functions using a Taylor expansion on the approximate values $\boldsymbol{p}_0$ and $\boldsymbol{z}_0$ up to the first order we get

$$\boldsymbol{g}(\widehat{\boldsymbol{p}}, \widehat{\boldsymbol{z}}) \approx \boldsymbol{g}(\boldsymbol{p}_0, \boldsymbol{z}_0) + A(\widehat{\boldsymbol{p}} - \boldsymbol{p}_0) + B^\mathsf{T}(\widehat{\boldsymbol{z}} - \boldsymbol{z}_0) \tag{3.93}$$

$$\boldsymbol{h}(\widehat{\boldsymbol{p}}) \approx \boldsymbol{h}(\boldsymbol{p}_0) + H^\mathsf{T}(\widehat{\boldsymbol{p}} - \boldsymbol{p}_0) \tag{3.94}$$

with the Jacobians

$$\underset{G\times U}{A} = \left.\frac{\partial \boldsymbol{g}(\boldsymbol{p},\boldsymbol{z})}{\partial \boldsymbol{p}}\right|_{\widehat{\boldsymbol{p}},\widehat{\boldsymbol{z}}} \tag{3.95}$$

$$\underset{N\times G}{B} = \left.\frac{\partial \boldsymbol{g}(\boldsymbol{p},\boldsymbol{z})^{\mathsf{T}}}{\partial \boldsymbol{z}}\right|_{\widehat{\boldsymbol{p}},\widehat{\boldsymbol{z}}} \tag{3.96}$$

$$\underset{U\times H}{H} = \left.\frac{\partial \boldsymbol{h}(\boldsymbol{p}}{\partial \boldsymbol{p}}\right|_{\widehat{\boldsymbol{p}}} \tag{3.97}$$

taken at the current approximation of $\widehat{\boldsymbol{p}}$ and $\widehat{\boldsymbol{z}}$. Now we enlarge the third term of equation (3.93) by $\boldsymbol{z}-\boldsymbol{z}$ and we get

$$\boldsymbol{g}(\widehat{\boldsymbol{p}},\widehat{\boldsymbol{z}}) \approx \boldsymbol{g}(\boldsymbol{p}_0,\boldsymbol{z}_0) + A(\widehat{\boldsymbol{p}}-\boldsymbol{p}_0) + B^{\mathsf{T}}(\widehat{\boldsymbol{z}}+\boldsymbol{z}-\boldsymbol{z}-\boldsymbol{z}_0) = \boldsymbol{0}. \tag{3.98}$$

Split equation (3.98) into two separated parts c_g and substitute $\widehat{\Delta\boldsymbol{p}} = \widehat{\boldsymbol{p}} - \boldsymbol{p}_0$ we define a nonlinear contradiction

$$\boldsymbol{c}_g = -\boldsymbol{g}(\boldsymbol{p}_0,\boldsymbol{z}_0) + B^{\mathsf{T}}(\boldsymbol{z}_0-\boldsymbol{z}) \tag{3.99}$$

and a linear contradiction

$$\boldsymbol{c}_g = A(\widehat{\boldsymbol{p}}-\boldsymbol{p}_0) + B^{\mathsf{T}}(\widehat{\boldsymbol{z}}-\boldsymbol{z}) = A\widehat{\Delta\boldsymbol{p}} + B^{\mathsf{T}}\widehat{\boldsymbol{v}}. \tag{3.100}$$

Note, both terms are equal if and only if the linearization point for the observation is $\boldsymbol{z}_0 = \widehat{\boldsymbol{z}}$ and for the unknown parameters is $\boldsymbol{p}_0 = \widehat{\boldsymbol{p}}$ so

$$\boldsymbol{0} = A\widehat{\Delta\boldsymbol{p}} = -\boldsymbol{g}(\boldsymbol{p}_0 = \widehat{\boldsymbol{p}}, \boldsymbol{z}_0 = \widehat{\boldsymbol{z}}). \tag{3.101}$$

The residuals $\widehat{\boldsymbol{v}}$ are equal to equation (3.83) replacing $\widehat{\boldsymbol{p}}$ with $\widehat{\Delta\boldsymbol{p}}$. Because we do not know a-priori $\widehat{\boldsymbol{p}}$ and $\widehat{\boldsymbol{z}}$ we have to find the solution by iteration. After rewriting equation (3.94) and using (3.100) we get the linearized normal equation system

$$\begin{bmatrix} A^{\mathsf{T}}(B^{\mathsf{T}}C_{zz}B)^{-1}A & H \\ H^{\mathsf{T}} & 0 \end{bmatrix} \begin{bmatrix} \widehat{\Delta\boldsymbol{p}} \\ \boldsymbol{\mu} \end{bmatrix} = \begin{bmatrix} A^{\mathsf{T}}(B^{\mathsf{T}}C_{zz}B)^{-1}\boldsymbol{c}_g \\ \boldsymbol{c}_h \end{bmatrix} \tag{3.102}$$

with $\boldsymbol{c}_g$ from equation (3.99) and

$$\boldsymbol{c}_h = -\boldsymbol{h}(\boldsymbol{p}_0). \tag{3.103}$$

In every iteration ν we update $\boldsymbol{p}^{(\nu+1)} = \boldsymbol{p}^{(\nu)} + \widehat{\Delta\boldsymbol{p}}^{(\nu)}$ and $\boldsymbol{z}^{(\nu+1)} = \boldsymbol{z} + \widehat{\boldsymbol{v}}^{(\nu)}$. The convergence of this iteration scheme depends on the linearity of the optimization space around the approximate values $\boldsymbol{p}^{(\nu)}$ and $\boldsymbol{z}^{(\nu)}$. Note, that this scheme does not guaranty to find the optimal minimum, because the optimization space depends on nonlinear functions and therefore can be non-convex. There are several methods to apparently overcome this limitation e. g. the *Levenberg-Marquardt* algorithm.

Optimizing Gauss-Markov-Model with constraints Now we assume that the relationship between the unknown parameters and the observations can be formulated as an explicit function of the form $\widetilde{\boldsymbol{z}} = \boldsymbol{f}(\widetilde{\boldsymbol{p}})$. We can always rewrite these functions to implicit functions by subtracting $\boldsymbol{z}$ from both sides. After replacing the true values against the estimated values we get

$$\mathbf{0} = \boldsymbol{g}(\widehat{\boldsymbol{p}}, \widehat{\boldsymbol{z}}) = \boldsymbol{f}(\widehat{\boldsymbol{p}}) - \widehat{\boldsymbol{z}}. \tag{3.104}$$

Applying the estimation algorithm of the nonlinear Gauss-Helmert model we always get the Jacobian $\mathsf{B} = -\mathsf{I}$. Setting the approximate value for the observations as $\boldsymbol{z}_0 = \boldsymbol{f}(\boldsymbol{p}_0)$, substitute (3.104) and B in equations (3.102), (3.99) and (3.100) we get

$$\begin{aligned} c_g &= -(\boldsymbol{f}(\boldsymbol{p}_0) - \boldsymbol{z}_0) - (\boldsymbol{z}_0 - \boldsymbol{z}) & (3.105)\\ &= \Delta\boldsymbol{z} = \boldsymbol{z} - \boldsymbol{z}_0 & (3.106)\\ &= \mathsf{A}\widehat{\Delta\boldsymbol{p}} - \widehat{\boldsymbol{v}} & (3.107)\end{aligned}$$

and therefore the normal equation system is

$$\begin{bmatrix} \mathsf{A}^{\mathsf{T}}\mathsf{C}_{zz}^{-1}\mathsf{A} & \mathsf{H} \\ \mathsf{H}^{\mathsf{T}} & \mathit{0} \end{bmatrix} \begin{bmatrix} \widehat{\Delta\boldsymbol{p}} \\ \boldsymbol{\mu} \end{bmatrix} = \begin{bmatrix} \mathsf{A}^{\mathsf{T}}\mathsf{C}_{zz}^{-1}\Delta\boldsymbol{z} \\ \boldsymbol{c}_h \end{bmatrix}. \tag{3.108}$$

The residuals in equation (3.83) simplify to

$$\widehat{\boldsymbol{v}} = \mathsf{A}\widehat{\Delta\boldsymbol{p}} - \Delta\boldsymbol{z}. \tag{3.109}$$

As we can see the Gauss-Markov model is a simplification of the more general Gauss-Helmert model.

3.3.2 Outlier detection and robustification

The least square estimation schemes presented so far minimize the square residuals of the observations. This is known to be extremely sensitive to outliers. In general we are not able to discriminate between systematic errors and gross errors. Systematic error can be caused by errors in the functional model or by non-modeled effects of the observation process. The last one can be a non-Gaussian distribution of the observations or neglected correlations. To analyze where these residuals do come from we have to distinguish between systematic and gross errors. In the following we assume that our functional model holds and therefore no systematic error appears. We will now show how gross errors, also called outliers, may be detected by looking at the plausibility of the computed residuals with respect to the expected uncertainty. Outliers can be eliminated canceling the observation, or their influence on the estimation process can be reduced.

Outlier detection by testing the residuals Following Baarda (1967) a test value t_n for uncorrelated observations in the Gauss-Markov-model can by obtained by

$$t_n = \frac{-\widehat{v}_n}{\sigma_{v_n}}. \tag{3.110}$$

If there are no gross errors in the observations $t_n \sim N(0,1)$ is normally distributed. The computation of σ_{v_n} can be expensive as the covariance of the parameters is required. In the case that all observation can be assumed to have the same influence to the parameter vector, it is suitable to use σ_{z_n} instead. In various cases, groups of observations are strongly correlated to each other but uncorrelated to the remaining observations. In this case we cannot distinguish which of the observations in the group i causes the error, but we can test the group itself Stefanovic (1978) with

$$T_i = \widehat{\boldsymbol{v}}_i^{\mathsf{T}} \mathsf{C}_{v_i v_i}^{-1} \widehat{\boldsymbol{v}}_i \sim \chi_I^2 \tag{3.111}$$

where I is the size of the group i and $\mathsf{C}_{v_i v_i}$ is the regular sub-matrix of the covariance of the residuals.

Using implicit functions in the Gauss-Helmert model we can formulate functional models, where a number of contradiction groups depend on a set of observations. In this case we cannot test the observation groups themselves according to equation (3.111) because of the functionally dependent on the contradictions. Using the relation (3.100) in the convergence point that $\widehat{\Delta \boldsymbol{p}} = \boldsymbol{0}$, we get

$$\boldsymbol{c}_g = \mathsf{B}^{\mathsf{T}}(\boldsymbol{z} - \boldsymbol{z}_0) = \mathsf{B}^{\mathsf{T}} \widehat{\boldsymbol{v}} \quad \text{with} \quad \mathsf{C}_{c_g c_g} = \mathsf{B}^{\mathsf{T}} \mathsf{C}_{v_i v_i} \mathsf{B} \tag{3.112}$$

and a test value for the groups of observations composed of the contradiction relations with

$$T_i = \boldsymbol{c}_{g_i}^{\mathsf{T}} \mathsf{C}_{c_{g_i} c_{g_i}}^{-1} \boldsymbol{c}_{g_i} \sim \chi_I^2. \tag{3.113}$$

Corresponding groups of observations than can be determined as outliers if the test value T_i exceeds a chosen threshold. The sub-covariance matrix $\mathsf{C}_{v_i v_i}$ can be singular and we are not able to compute the test value. However, if we assume that all observations have the same influence to the parameter vector, it is suitable to use C_{zz} instead.

Influence reduction Outliers can be eliminated in the estimation process. Alternatively, their influence on the minimization function can be reduced. This technique is called reweighting. The key idea is to adjust the assumed weight of the observations or rather groups by an influence function, so that their influence on the minimization function is not square. If there is a-priori information about the distribution of the gross errors, the function should be chosen accordingly. The influence of the square residuals to the estimated parameter vector can be adjusted using different functions ρ (c. f. Hampel *et al.* (1986). In table (3.3.2) we can see commonly used influence and corresponding weighting functions. Assuming uncorrelated and independent observations than equation (3.76) can be rewritten as

$$\Omega^2 = \sum_n \Omega_n^2 = \sum_n \frac{1}{2}\left(\frac{v_n}{\sigma_{z_n}}\right)^2 = \sum_n \rho(x) \tag{3.114}$$

This means that the influence function for the least square case is $\rho(x) = \frac{1}{2}x^2$. Instead to replace the influence function, we can reweight the variances of the observations by its corresponding weighting function. The weighting function for the observations according to the influence function can be derived by

$$w(x) = \frac{\frac{d\rho(x)}{dx}}{x} \tag{3.115}$$

which leads to weighting factors w_n for individual uncorrelated observations n. To reduce the influence of outliers we multiply the covariance of the observation group by

$$\mathsf{C}_{z_i z_i}^{(\nu)} = \frac{1}{w(x_i^{(\nu)})}\mathsf{C}_{z_i z_i}^{(0)} \tag{3.116}$$

in the νth iteration.

Outlier detection by testing the variance components In case that a group of observations belongs together in a functional manner, alternatively the variance component of this group can be tested to detect that the group does not fulfill the functional model and should be eliminated from the optimization process. The variance factor of the i-th group can be computed following Förstner (1987) by

$$\widehat{\sigma}_{0i}^2 = \frac{\widehat{\boldsymbol{v}}_i^{\mathsf{T}} \mathsf{C}_{z_i z_i}^{-1} \widehat{\boldsymbol{v}}_i}{r_i} \tag{3.117}$$

with the redundancy numbers r_i

$$r_i = \mathrm{tr}(\mathsf{C}_{v_i v_i} \mathsf{C}_{z_i z_i}^{-1}). \tag{3.118}$$

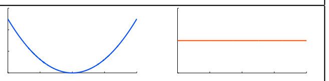
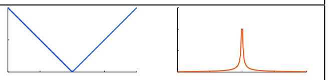
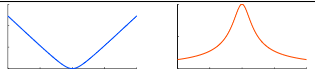
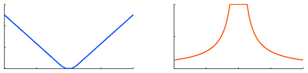
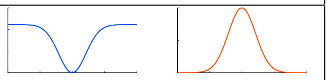
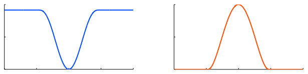
Norm	influence function	weight function	visualizations
L_2	$\rho(x) = \frac{1}{2}x^2$	$w(x) = 1$	
L_1	$\rho(x) = \|x\|$	$w(x) = \frac{\text{sign}(x)}{x} = \frac{1}{\|x\|}$	
L_1-L_2-mixture	$\rho(x) = 2(\sqrt{1+\frac{x^2}{2}}-1)$	$w(x) = \frac{1}{\sqrt{1+\frac{x^2}{2}}}$	
Huber $\begin{cases} \text{if}\|x\| \leq k \\ \text{if}\|x\| \geq k \end{cases}$	$\rho(x) = \begin{cases} \frac{x^2}{2} \\ k(\|x\| - \frac{k}{2}) \end{cases}$	$\psi(x) = \begin{cases} 1 \\ \frac{k}{\|x\|}) \end{cases}$	
Welch	$\rho(x) = \frac{k^2}{2}(1-\exp(-\frac{x^2}{k^2})$	$\psi(x) = \exp(-\frac{x^2}{k^2})$	
Tukey $\begin{cases} \text{if}\|x\| \leq k \\ \text{if}\|x\| \geq k \end{cases}$	$\rho(x) = \begin{cases} \frac{k^2}{6}(1-1-\frac{x^2}{k^2})^3) \\ \frac{k^2}{6} \end{cases}$	$\psi(x) = \begin{cases} (1-\frac{x^2}{k^2})^2 \\ 0 \end{cases}$	

Table 3.2: Optimization and corresponding weight functions for different norms for robust estimation (M-estimators). The tuning factors k can be adjusted for the specific distribution of the task. Substitute $x^2 = \widehat{\boldsymbol{v}}_i^{\mathsf{T}} \mathsf{C}_{z_i z_i}^{-1} \widehat{\boldsymbol{v}}_i$ in the Gauss-Markov-model or $x^2 = \boldsymbol{c}_{g_i}^{\mathsf{T}} \mathsf{C}_{c_{g_i} c_{g_i}}^{-1} \boldsymbol{c}_{g_i}$ in the Gauss Helmert model.

If the variance factor of the group exceeds an a-priori assumed variance factor, the observations of the group can be determined as outliers. In real implementations one will not eliminate all groups determined as outliers simultaneously from the estimation process. Instead, one should eliminate the worst group or groups and iterate the estimation process.

3.3.3 Parameter elimination in the normal equation system

In some cases we are only interested in a subset of parameters inside the parameter vector. For instance in case of the camera orientation problem the object coordinates can be of no interest. According to the special structure of the normal equation system of this task, the computation time of solving this for the camera parameters can be speeded up dramatically. In the Gauss-Helmert-model and the Gauss-Markov model the normal equation system has the form

$$N\widehat{p} = \boldsymbol{n} \tag{3.119}$$

ignoring contradictions on the parameter. Let us assume we can split the unknown parameter vector into two parts

$$\boldsymbol{p} = \begin{bmatrix} \boldsymbol{p}_1 \\ \boldsymbol{p}_2 \end{bmatrix}. \tag{3.120}$$

The normal equation system can then be rewritten as

$$\begin{bmatrix} N_{11} & N_{12} \\ N_{21} & N_{22} \end{bmatrix} \begin{bmatrix} \widehat{\boldsymbol{p}}_1 \\ \widehat{\boldsymbol{p}}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{n}_1 \\ \boldsymbol{n}_2 \end{bmatrix} \tag{3.121}$$

which leads to the equations

$$N_{11}\widehat{\boldsymbol{p}}_1 + N_{12}\widehat{\boldsymbol{p}}_2 = \boldsymbol{n}_1 \tag{3.122}$$

$$N_{21}\widehat{\boldsymbol{p}}_1 + N_{22}\widehat{\boldsymbol{p}}_2 = \boldsymbol{n}_2. \tag{3.123}$$

Drawing $N_{21}\widehat{\boldsymbol{p}}_1$ from the left hand side of (3.123) and multiplying the left hand side with N_{22}^{-1} we get the subvector solution with respect to $\widehat{\boldsymbol{p}}_1$ as

$$\widehat{\boldsymbol{p}}_2 = N_{22}^{-1}\boldsymbol{n}_2 - N_{22}^{-1}N_{21}\widehat{\boldsymbol{p}}_1. \tag{3.124}$$

Now substitute (3.124) into (3.122) we get

$$N_{11}\widehat{\boldsymbol{p}}_1 + N_{12}\widehat{\boldsymbol{p}}_2 = \boldsymbol{n}_1 \tag{3.125}$$

$$N_{11}\widehat{\boldsymbol{p}}_1 + N_{12}(N_{22}^{-1}\boldsymbol{n}_2 - N_{22}^{-1}N_{21}\widehat{\boldsymbol{p}}_1) = \boldsymbol{n}_1 \tag{3.126}$$

$$(N_{11} - N_{12}N_{22}^{-1}N_{21})\widehat{\boldsymbol{p}}_1 = \boldsymbol{n}_1 - N_{12}N_{22}^{-1}\boldsymbol{n}_2 \tag{3.127}$$

$$\bar{N}_{11}\widehat{\boldsymbol{p}}_1 = \bar{\boldsymbol{n}}_1. \tag{3.128}$$

Observe, that the inverse of N_{22} has to be computed in equation 3.127 and therefore has to be full rank. Without any limitation we can expand (3.128) using contradiction $H\widehat{\boldsymbol{p}} = \boldsymbol{0}$, if the common normal equations system is regular.

3.4 Recursive state estimation

In this section we will introduce the concept of estimating states from sensor models in a recursive manner. State here means a time varying unknown parameter vector. The aim is to derive a recursive least square estimator and embedding it into a broader context of recursive state estimation.

3.4.1 General recursive state estimation

In general the real world changes over time. For example objects change their position, light becomes darker or wind changes its strength. On the other hand, some objects or occurrences are static over a segment of time. Describing these phenomenons we distinguish between *dynamic state variables* and *static state variables*. Describing the evolution of the real world as a temporal process depend on the state of the world only on variables prior to the actual time t we get a so called *Markov chain* or *dynamic Bayes networks*. Measurements depends stochastically on the state of the world.

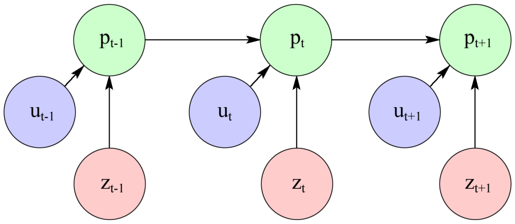


Figure 3.4: The dynamic Bayes network of the evolution of the world depends on the description of the world $\boldsymbol{p}$, controls $\boldsymbol{u}$ and measurements $\boldsymbol{z}$.

Figure 3.4 illustrates the evolution of the knowledge of the world coded in the state vector $\boldsymbol{p}$. We distinguish between control data $\boldsymbol{u}$ and measurements $\boldsymbol{z}$, although both introduce information into the system. Controls are actions which influence the state directly, which interact with the system dynamics and are separable in time from the measurements.

The basic concept for a realization of the *dynamic Bayes network* is called *general Bayes filter*. In the following, we denote $P(\boldsymbol{p})$ as the probability density function of the state $\boldsymbol{p}$. The *sum rule* or joint probability has to be used for the dynamic system, because the new state $\boldsymbol{p}_t$ depends on the old state $\boldsymbol{p}_{t-1}$ so that the predicted probability $\vec{P}(\boldsymbol{p}_t)$ can be obtained by

$$\vec{P}(\boldsymbol{p}_t) = \int P(\boldsymbol{p}_t|\boldsymbol{p}_{t-1}, \boldsymbol{u}_t)P(\boldsymbol{p}_{t-1})d\boldsymbol{p}_{t-1}. \tag{3.129}$$

Under the assumption that the observations $\boldsymbol{z}_t$ are independent of the state $\boldsymbol{p}_t$, the updated probability density can be obtained by

$$P(\boldsymbol{p}_t) = \eta P(\boldsymbol{z}_t|\boldsymbol{p}_t)\vec{P}(\boldsymbol{p}_t). \tag{3.130}$$

The factor η is just a normalization factor, that $\int P(\boldsymbol{p}_t) = 1$. Real implementations of equation (3.129) and (3.130) require a dynamic and a measurement model as well as a representation of the probabilistic state space.

3.4.2 The derivation of Kalman filtering

In the following we will assume that the probability density over the variable $\boldsymbol{p}_t$ can be approximated using a Gaussian distribution. One realization of the *Bayes filter* using a Gaussian approximation will be called Kalman filter. The probability density $P(\boldsymbol{p})$ can be represented using the Gauss-function

$$P(\boldsymbol{p}) = \det(2\pi\Sigma)^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(\boldsymbol{p}-\boldsymbol{\mu})^{\mathsf{T}} \mathsf{C}^{-1}(\boldsymbol{p}-\boldsymbol{\mu})\right\} \tag{3.131}$$

with its mean $\boldsymbol{\mu}$ and covariance C. We distinguish between the dynamic model and the measurement model. Figure 3.5 illustrates the filter scheme. The transition of the probability density in the dynamic model can be computed using error propagation. The dynamic model is given in the general non-linear function to get the predicted[3] state vector $\vec{\boldsymbol{p}}_t$ by

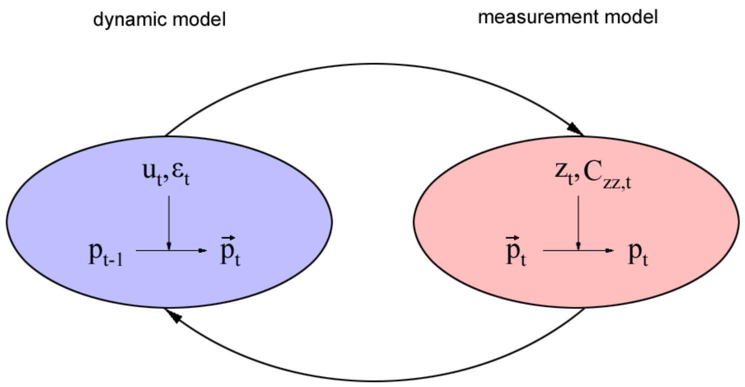


Figure 3.5: Kalman filter scheme. The dynamic model is the realization of equation (3.129), the measurement model of equation (3.130) of the Bayes filter.

$$\vec{\boldsymbol{p}}_t = \boldsymbol{h}(\boldsymbol{p}_{t-1}, \boldsymbol{u}_t) + \boldsymbol{s}(\boldsymbol{\epsilon}_t) \tag{3.132}$$

with $\boldsymbol{u}_t$ as control variables and $\boldsymbol{\epsilon}_t$ as additional Gaussian random noise influences the state according to a non-linear function $\boldsymbol{s}$. The non-linear error propagation can be achieved for instance by a linearization of the dynamic model using a Taylor expansion. The mean of

[3] Predicted variables are indicated by an arrow.

the posterior distribution can be transformed using the non-linear model and the Gaussian distribution C_{pp} of the state using the linearized error propagation

$$C_{\vec{p}\vec{p},t} = HC_{pp,t-1}H^{\mathsf{T}} + SC_{\epsilon\epsilon,t}S^{\mathsf{T}} \tag{3.133}$$

with

$$H = \left. \frac{\partial \boldsymbol{h}(\boldsymbol{p}_{t-1}, \boldsymbol{u}_t)}{\partial \boldsymbol{p}_{t-1}} \right|_{\boldsymbol{p}_{t-1}, \boldsymbol{u}_t} \tag{3.134}$$

$$S = \left. \frac{\partial \boldsymbol{s}(\boldsymbol{p}_{t-1})}{\partial \boldsymbol{p}_{t-1}} \right|_{\boldsymbol{p}_{t-1}} . \tag{3.135}$$

After the prediction we obtain measurements from the world to update or correct the predicted state. This process can be interpreted as a weighted mean of a direct measurement of the state itself and an indirect measurements from the sensor data.

Typically, in the common literature the derivation of the update equation will be formulated specializing the linear case going to non-linear iterative case. Here we will derive a general novel form of an implicit update equation similar to the Gauss-Helmert model in section 3.3.1 and we will show, that the classical linear, non-linear and iterative non-linear update equations are specializations of our formulation.

Recursive update in the Gauss-Helmert-model[4] From equation (3.91) we can split the parameter estimation into two groups of implicit constraints

$$\begin{bmatrix} \boldsymbol{g}_1(\widehat{\boldsymbol{p}}, \widehat{\boldsymbol{z}}_1) \\ \boldsymbol{g}_2(\widehat{\boldsymbol{p}}, \widehat{\boldsymbol{z}}_2) \end{bmatrix} = \boldsymbol{0} \tag{3.136}$$

with $\boldsymbol{g}_1$ as a direct observation of the state and $\boldsymbol{g}_2$ as an indirect observation. In the following $\boldsymbol{p}$ is equal to $\boldsymbol{p}_t$. Using the normal equation system in equation (3.102) and neglecting the constraints on the parameter vector, we can rewrite the left side as

$$A^{\mathsf{T}}(B^{\mathsf{T}}C_{zz}B)^{-1}A = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix}^{\mathsf{T}} \left(\begin{bmatrix} B_{11}^{\mathsf{T}} & B_{12}^{\mathsf{T}} \\ B_{21}^{\mathsf{T}} & B_{22}^{\mathsf{T}} \end{bmatrix} \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} \right)^{-1} \begin{bmatrix} A_1 \\ A_2 \end{bmatrix} \tag{3.137}$$

and the right side as

$$A^{\mathsf{T}}(B^{\mathsf{T}}C_{zz}B)^{-1}\boldsymbol{c}_g = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix}^{\mathsf{T}} \left(\begin{bmatrix} B_{11}^{\mathsf{T}} & B_{12}^{\mathsf{T}} \\ B_{21}^{\mathsf{T}} & B_{22}^{\mathsf{T}} \end{bmatrix} \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} \right)^{-1} \begin{bmatrix} \boldsymbol{c}_{g_1} \\ \boldsymbol{c}_{g_2} \end{bmatrix} . \tag{3.138}$$

[4] First publication in Steffen & Beder (2007)

In the following we will assume that the two observation blocks are stochastically independent, i.e. $C_{12} = C_{21} = 0$ as well as functionally independent, i.e. $B_{12} = B_{21} = 0$. Observe, that these boundary conditions are implicitly defined in the Bayes filtering scheme. Then we can rewrite the left and right side as

$$A^{\mathsf{T}}(B^{\mathsf{T}}C_{zz}B)^{-1}A = A_1^{\mathsf{T}}(B_{11}^{\mathsf{T}}C_{11}B_{11})^{-1}A_1 + A_2^{\mathsf{T}}(B_{22}^{\mathsf{T}}C_{22}B_{22})^{-1}A_2 \tag{3.139}$$

and

$$A^{\mathsf{T}}(B^{\mathsf{T}}C_{zz}B)^{-1}\boldsymbol{c}_g = A_1^{\mathsf{T}}(B_{11}^{\mathsf{T}}C_{11}B_{11})^{-1}\boldsymbol{c}_{g_1} + A_2^{\mathsf{T}}(B_{22}^{\mathsf{T}}C_{22}B_{22})^{-1}\boldsymbol{c}_{g_2}. \tag{3.140}$$

Using the substitution $W = B^{\mathsf{T}}C_{zz}B$, the final solution, that incorporates both observation $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$, may be obtained iteratively at the approximate values $\boldsymbol{p}^{(\nu)}$ and $\boldsymbol{z}^{(\nu)}$ using the update

$$\widehat{\Delta\boldsymbol{p}} = (A_1^{\mathsf{T}}W_{11}^{-1}A_1 + A_2^{\mathsf{T}}W_{22}^{-1}A_2)^{-1}(A_1^{\mathsf{T}}W_{11}^{-1}\boldsymbol{c}_{g_1} + A_2^{\mathsf{T}}W_{22}^{-1}\boldsymbol{c}_{g_2}). \tag{3.141}$$

In the following the dependency on the first set of contradiction $\boldsymbol{c}_{g_1}$ and therefore on the first set of observation $\boldsymbol{z}_1$ should be removed. The goal of recursive estimation is now to derive such a solution $\widehat{\Delta\boldsymbol{p}}_2$ for the combined constraints using the solution of the first constraint block $\boldsymbol{g}_1$ represented by $\widehat{\Delta\boldsymbol{p}_1}$ and its covariance matrix $C_{p_1p_1}$ as well as the new constraint block $\boldsymbol{g}_2$ together with the new observations $\boldsymbol{z}_2$ and their covariance matrix C_{22}. In order to achieve this goal equation (3.141) may be re-written as

$$\widehat{\Delta\boldsymbol{p}}_2 = \underbrace{(\underbrace{A_1^{\mathsf{T}}W_{11}^{-1}A_1}_{C_{p_1p_1}^{-1}} + A_2^{\mathsf{T}}W_{22}^{-1}A_2)^{-1}}_{C_{p_2p_2}}(\underbrace{A_1^{\mathsf{T}}W_{11}^{-1}\vec{\boldsymbol{c}}_{g_1}}_{C_{p_1p_1}^{-1}\widehat{\Delta p_1}} + A_1^{\mathsf{T}}W_{11}^{-1}\Delta\boldsymbol{c}_{g_1} + A_2^{\mathsf{T}}W_{22}^{-1}\boldsymbol{c}_{g_2}) \tag{3.142}$$

with the contradictions for the first block being separated into

$$\boldsymbol{c}_{g_1} = \vec{\boldsymbol{c}}_{g_1} + \Delta\boldsymbol{c}_{g_1}. \tag{3.143}$$

Observe that the contradictions[5] $\vec{\boldsymbol{c}}_{g_1}$ for the first contradiction block $\boldsymbol{g}_1$ at $\boldsymbol{p}_1$ change due to the change of parameters resulting from the new contradiction block $\boldsymbol{g}_2$, due to the dependence on $\boldsymbol{p}$ of equation (3.99). As a consequence, the residuals for the observations of the first contradiction block change as well

$$\boldsymbol{v}_1 = C_{11}B_1W_{11}^{-1}(\vec{\boldsymbol{c}}_{g_1} + \Delta\boldsymbol{c}_{g_1} - A_1\widehat{\Delta\boldsymbol{p}_2}) \tag{3.144}$$

$$= \underbrace{C_{11}B_1W_{11}^{-1}(\vec{\boldsymbol{c}}_{g_1} - A_1\widehat{\Delta\boldsymbol{p}_1})}_{\vec{\boldsymbol{v}}_1} + \underbrace{C_{11}B_1W_{11}^{-1}(\Delta\boldsymbol{c}_{g_1} - A_1(\widehat{\Delta\boldsymbol{p}}_2 - \widehat{\Delta\boldsymbol{p}}_1))}_{\Delta\boldsymbol{v}_1}. \tag{3.145}$$

[5] Here we use $\vec{(\cdot)}$ to indicate, that the vector depends on a prediction to be consistent in the following equations.

The expression $C_{p_2p_2}$ in equation (3.142) is the inverse of a sum and can be decomposed using the matrix inversion identity (c.f. Koch (1988) page 37)

$$(K + LN^{-1}M)^{-1} = K^{-1} - K^{-1}L(N + MK^{-1}L)^{-1}MK^{-1} \tag{3.146}$$

that follows

$$\begin{aligned} C_{p_2p_2} &= C_{p_1p_1} - \underbrace{C_{p_1p_1}A_2^{\mathsf{T}}(W_{22} + A_2C_{p_1p_1}A_2^{\mathsf{T}})^{-1}}_{F}A_2C_{p_1p_1} && (3.147)\\ &= C_{p_1p_1} - FA_2C_{p_1p_1} && (3.148)\\ &= (I - FA_2)C_{p_1p_1} && (3.149)\\ &= C_{p_1p_1} - F(W_{22} + A_2C_{p_1p_1}A_2^{\mathsf{T}})F^{\mathsf{T}} && (3.150) \end{aligned}$$

with F being the gain or influence matrix. We can see, that this update of $C_{p_2p_2}$ does not involve the inversion of the full normal equation matrix. Substituting this back into equation (3.142) we obtain

$$\begin{aligned} \widehat{\Delta\boldsymbol{p}}_2 &= (I - FA_2)C_{p_1p_1}C_{p_1p_1}^{-1}\widehat{\Delta\boldsymbol{p}}_1 + (I - FA_2)C_{p_1p_1}A_2^{\mathsf{T}}W_{22}^{-1}\boldsymbol{c}_{g_2} + && (3.151)\\ &\quad (I - FA_2)C_{p_1p_1}A_1^{\mathsf{T}}W_{11}^{-1}\Delta\boldsymbol{c}_{g_1} &&\\ &= \widehat{\Delta\boldsymbol{p}}_1 - FA_2\widehat{\Delta\boldsymbol{p}}_1 + F\boldsymbol{c}_{g_2} + (I - FA_2)C_{p_1p_1}A_1^{\mathsf{T}}W_{11}^{-1}\Delta\boldsymbol{c}_{g_1} && (3.152) \end{aligned}$$

using the identity Koch (1988) page 37

$$F = (I - FA_2)C_{p_1p_1}A_2^{\mathsf{T}}W_{22}^{-1} = C_{p_1p_1}A_2^{\mathsf{T}}(W_{22} + A_2C_{p_1p_1}A_2^{\mathsf{T}})^{-1}. \tag{3.153}$$

The only remaining part still depending on $\boldsymbol{z}_1$ is now the change of the contradictions (see equation (3.99))

$$\Delta\boldsymbol{c}_{g_1} = \boldsymbol{c}_{g_1} - \vec{\boldsymbol{c}}_{g_1} = -\boldsymbol{g}_1(\widehat{\boldsymbol{p}}_2, \widehat{\boldsymbol{z}}_1) + B_1^{\mathsf{T}}\boldsymbol{v}_1 + \boldsymbol{g}_1(\widehat{\boldsymbol{p}}_1, \boldsymbol{z}_1) - B_1^{\mathsf{T}}\vec{\boldsymbol{v}}_1. \tag{3.154}$$

In order to remove this remaining dependence on the previous observations, we bring in mind that the whole first contradiction block is encoded in the first two moments of the parameter vector only. We therefore replace the first contradiction block by a direct observation of the parameters itself from the prediction, i.e. $\boldsymbol{z}_1 = \vec{\boldsymbol{p}}_t = \widehat{\boldsymbol{p}}_1$ and $C_{p_1p_1} = C_{\vec{p}\vec{p},t}$, so that

$$\boldsymbol{g}_1(\widehat{\boldsymbol{p}}_1, \boldsymbol{z}_1) = \widehat{\boldsymbol{p}}_1 - \boldsymbol{z}_1 = \boldsymbol{0} \tag{3.155}$$

immediately fulfills the constraint and therefore

$$\begin{aligned} \vec{\boldsymbol{c}}_{g_1} &= \boldsymbol{0} && (3.156)\\ \vec{\boldsymbol{v}}_1 &= \boldsymbol{0} && (3.157)\\ \widehat{\Delta \boldsymbol{p}}_1 &= \boldsymbol{0}. && (3.158) \end{aligned}$$

Furthermore the Jacobians are given by $A_1 = I$ and $B_1 = -I$ independently of the linearization point. Now equation (3.152) simplifies to

$$\widehat{\Delta \boldsymbol{p}}_2 = F\boldsymbol{c}_{g_2} + (I - FA_2)\Delta \boldsymbol{c}_{g_1} \tag{3.159}$$

with

$$\begin{aligned} \Delta \boldsymbol{c}_{g_1} &= -\boldsymbol{g}_1(\boldsymbol{p}^{(\nu)}, \widehat{\boldsymbol{z}}_1) - \boldsymbol{v}_1\\ &= -\boldsymbol{p}^{(\nu)} + \vec{\boldsymbol{p}}_t + \boldsymbol{v}_1 - \boldsymbol{v}_1\\ &= -\boldsymbol{p}^{(\nu)} + \vec{\boldsymbol{p}}_t. && (3.160) \end{aligned}$$

For the second contradiction block we can compute the residuals

$$\boldsymbol{v}_2 = C_{22} B_2 W_{22}^{-1}(\boldsymbol{c}_{g_2} - A_2 \widehat{\Delta \boldsymbol{p}}_2) \tag{3.161}$$

and the contradictions

$$\boldsymbol{c}_{g_2} = -\boldsymbol{g}_2(\boldsymbol{p}^{(\nu)}, \boldsymbol{z}^{(\nu)}) + B_2^{\mathsf{T}} \boldsymbol{v}_2 \tag{3.162}$$

where $\boldsymbol{p}_0^{(\nu+1)} = \boldsymbol{p}^{(\nu)} + \widehat{\Delta \boldsymbol{p}}_2^{(\nu)}$ and $\boldsymbol{z}^{(\nu+1)} = \boldsymbol{z}_2 + \boldsymbol{v}_2^{(\nu)}$ has to be iterated until convergence is reached. We will summarize the update algorithm in 3.1.

Recursive update in the Gauss-Markov-model Similar to the derivation of the least square estimation we can derive the Kalman filter update equations for the Gauss-Markov model as a specialization of the Gauss-Helmert model. Again beginning at the explicit observation model $\widetilde{\boldsymbol{z}} = \boldsymbol{f}(\widetilde{\boldsymbol{p}})$, which can be rewritten to the implicit model using (3.104). The Jacobian is with respect to the observations $B_2 = -I$. The Gain matrix from equation (3.153) simplifies to

$$F = C_{p_1 p_1} A_2^{\mathsf{T}} (C_{zz} + A_2 C_{p_1 p_1} A_2^{\mathsf{T}})^{-1}. \tag{3.163}$$

Substitute the model into (3.162) we get

$$\begin{aligned} \boldsymbol{c}_{g_2} &= -\boldsymbol{g}_2(\boldsymbol{p}^{(\nu)}, \boldsymbol{z}^{(\nu)}) + B_2^{\mathsf{T}} \boldsymbol{v}_2\\ &= -(\boldsymbol{f}(\boldsymbol{p}^{(\nu)}) - (\boldsymbol{z}_2 + \boldsymbol{v}_2)) - \boldsymbol{v}_2\\ &= \boldsymbol{z}_2 - \boldsymbol{f}(\boldsymbol{p}^{(\nu)}). && (3.164) \end{aligned}$$

Algorithm 3.1 Iterative Kalman filter update in the Gauss-Helmert-model

```
given observation vector $\boldsymbol{z}_2$, its covariance $C_{zz}$ and predicted state vector $\vec{\boldsymbol{p}}_t$
set $\nu = 0$ as the iteration counter
set the initial state correction $\widehat{\Delta \boldsymbol{p}}_2^{(\nu)} = 0$
set initial $\boldsymbol{p}^{(\nu)} = \vec{\boldsymbol{p}}_t$ and $C_{p_1 p_1} = C_{\vec{p}\vec{p},t}$
set $\boldsymbol{v}_2 = \boldsymbol{0}$, hence $\boldsymbol{z}^{(\nu)} = \boldsymbol{z}_2$
repeat
    compute Jacobians $A_2$ and $B_2$ at $\boldsymbol{p}^{(\nu)}$ and $\boldsymbol{z}^{(\nu)}$
    compute the gain matrix $F^{(\nu)}$ as shown in equation (3.153)
    compute $\boldsymbol{c}_{g_2}^{(\nu)}$ according to equation (3.162)
    compute $\Delta \boldsymbol{c}_{g_1}^{(\nu)}$ according to equation (3.160)
    compute $\widehat{\Delta \boldsymbol{p}}_2^{(\nu)}$ according to equation (3.159)
    update $\boldsymbol{p}^{(\nu+1)} = \boldsymbol{p}^{(\nu)} + \widehat{\Delta \boldsymbol{p}}_2^{(\nu)}$
    compute $\boldsymbol{v}_2^{(\nu)}$ according to equation (3.161)
    update $\boldsymbol{z}^{(\nu+1)} = \boldsymbol{z}_2 + \boldsymbol{v}_2^{(\nu)}$
    increase $\nu$ by 1
until $\widehat{\Delta \boldsymbol{p}}_2^{(\nu)}$ is sufficiently small
compute the updated covariance $C_{pp,t} = C_{p_2 p_2}$ according to equation (3.149)
```

This results in the iterative update

$$\begin{aligned}\widehat{\Delta\boldsymbol{p}}_2 &= \mathsf{F}\boldsymbol{c}_{g_2} + (\mathsf{I} - \mathsf{F}\mathsf{A}_2)\Delta\boldsymbol{c}_{g_1} \\ &= \mathsf{F}(\boldsymbol{z}_2 - \boldsymbol{f}(\boldsymbol{p}^{(\nu)})) + (\mathsf{I} - \mathsf{F}\mathsf{A}_2)(\vec{\boldsymbol{p}}_t - \boldsymbol{p}^{(\nu)}) \\ &= (\vec{\boldsymbol{p}}_t - \boldsymbol{p}^{(\nu)}) + \mathsf{F}(\boldsymbol{z}_2 - \boldsymbol{f}(\boldsymbol{p}^{(\nu)}) - \mathsf{A}_2(\vec{\boldsymbol{p}}_t - \boldsymbol{p}^{(\nu)}))\end{aligned} \tag{3.165}$$

to the approximate values $\boldsymbol{p}^{(\nu+1)} = \boldsymbol{p}^{(\nu)} + \widehat{\Delta\boldsymbol{p}}_2^{(\nu)}$. Instead we can use the predicted values as the reference state which results in the update equation

$$\boldsymbol{p}^{(\nu+1)} = \vec{\boldsymbol{p}}_t + \mathsf{F}(\boldsymbol{z}_2 - \boldsymbol{f}(\boldsymbol{p}^{(\nu)}) - \mathsf{A}_2(\vec{\boldsymbol{p}}_t - \boldsymbol{p}^{(\nu)}) \tag{3.166}$$

which is known in the literature as the *iterative extended Kalman filter update*. In many practical applications the approximate values from the prediction $\boldsymbol{p}^{(0)} = \vec{\boldsymbol{p}}_t$ in the first iteration is good enough and there is no iteration necessary. In this case equation (3.166) simplifies to

$$\widehat{\boldsymbol{p}} = \vec{\boldsymbol{p}}_t + \mathsf{F}(\boldsymbol{z}_2 - \boldsymbol{f}(\vec{\boldsymbol{p}}_t)) \tag{3.167}$$

and is known as the *extended Kalman filter update*. We summarize the update algorithm in 3.2.

Algorithm 3.2 Iterative Kalman filter update in the Gauss-Markov-model

```
set $\nu = 0$ as the iteration counter
set initial state $\boldsymbol{p}^{(\nu)} = \vec{\boldsymbol{p}}_t$ and state covariance $\mathsf{C}_{p_1p_1} = \mathsf{C}_{\vec{p}\vec{p},t}$
repeat
    compute Jacobian $\mathsf{A}_2$ for $\boldsymbol{f}(\boldsymbol{p}^{(\nu)})$
    compute the gain matrix $\mathsf{F}$ as shown in equation (3.163)
    update $\boldsymbol{p}^{(\nu+1)}$ according to equation (3.166)
    increase $\nu$ by 1
until the difference of $\Delta^{(\nu)} = \boldsymbol{p}^{(\nu+1)} - \boldsymbol{p}^{(\nu)}$ is sufficiently small
compute the updated covariance $\mathsf{C}_{pp,t} = \mathsf{C}_{p_2p_2}$ according to equation (3.149)
```

3.4.3 The particle based Kalman filtering

As mentioned before the Kalman filter is a realization of the general Bayes filter using a Gaussian approximation for the posterior distribution and assumes a Gaussian distribution after the correction step. The transition and update through non-linear functions are realized by a Taylor expansion. The Gaussian distribution is transformed using the non-linear transformation for its mean and by a linear transformation for its covariance. Higher degrees of the non-linear transformation are neglected. Another error propagation method is the so-called *unscented transformation* introduced by Julier & Uhlmann (1996). Instead of approximating the non-linear function by a Taylor series the posterior Gaussian distribution can be approximated using a series of particles, which will be passed through the non-linear function itself. Instead of using randomly selected particles according to a Monte Carlo simulation, Julier & Uhlmann (1996) choose deterministically particles. These particles are called *Sigma points*. First we will briefly describe the error propagation through the *unscented transformation* and then apply this algorithm to a *dynamic system* resulting in the so-called *Unscented Kalman filter*. For a more detailed explanation c. f. van der Merwe (2004).

The posterior distribution is assumed to be n-dimensional Gaussian by its mean μ and its covariance C. We compute $2n+1$ sigma points $\boldsymbol{\chi}^{[i]}$:

$$\begin{aligned}
\boldsymbol{\chi}^{[1]} &= \mu \\
\boldsymbol{\chi}^{[i]} &= \mu + (\sqrt{(n+\lambda)\mathsf{C}})_i \qquad i = 1, ..., n \\
\boldsymbol{\chi}^{[i]} &= \mu - (\sqrt{(n+\lambda)\mathsf{C}})_i \qquad i = n+1, ..., 2n
\end{aligned} \tag{3.168}$$

where the scaled root of the covariance $(\sqrt{(n+\lambda)\mathsf{C}})$ is a matrix square root computed by a *Cholesky-decomposition*[6]. The sigma points are associated with weights $W^{[i]}$, that reflect the influence of the i-th sigma point to the Gaussian approximation.

$$\begin{aligned}
W^{[0]} &= \frac{\lambda}{n+\lambda} \\
W^{[i]} &= \frac{1}{2(n+\lambda)} \qquad i = 1, .., 2n.
\end{aligned} \tag{3.169}$$

Here λ is a tuning factor. For Gaussian distributions select $\lambda = 3 - n$ (c. f. Julier & Uhlmann (1996)). The general non-linear error propagation is summarized in algorithm 3.4.3.

As we have shown before, the Kalman filter consists of two parts, a dynamic model performed by error propagation using the general model in equation (3.132) and a measurement model. The measurement update using the *unscented transformation* is applicable for explicit measurement functions (Gauss-Markov-model) only.

[6] In case of a singular covariance matrix C, we use an incomplete Cholesky decomposition.

Algorithm 3.3 Unscented Transformation

```
1: compute *sigma points* and their weights according to equations (3.168) and (3.169)
2: transform all sigma points with the non-linear system
       $\boldsymbol{\gamma}^{[i]} = \boldsymbol{f}(\boldsymbol{\chi}^{[i]})$    (3.170)
3: compute the mean by
       $\bar{\boldsymbol{\gamma}} = \sum_i W^{[i]} \boldsymbol{\gamma}^{[i]}$    (3.171)
4: compute the covariance by the weighted sum of the dyadic product
       $C_{\gamma\gamma} = \sum_i W^{[i]} (\boldsymbol{\gamma}^{[i]} - \bar{\boldsymbol{\gamma}})(\boldsymbol{\gamma}^{[i]} - \bar{\boldsymbol{\gamma}})^{\mathsf{T}}$    (3.172)
```

The dynamic model has to take into account the effect of process noise $\boldsymbol{\epsilon}$. In a first step the state vector is augmented with the process noise with mean zero and covariance $C_{\epsilon\epsilon,t}$

$$\boldsymbol{p}^*_{t-1} = \begin{bmatrix} \boldsymbol{p}_{t-1} \\ \boldsymbol{0} \end{bmatrix} \qquad \text{with} \qquad C^*_{pp,t-1} = \begin{bmatrix} C_{pp,t-1} & 0 \\ 0 & C_{\epsilon\epsilon,t} \end{bmatrix}. \tag{3.173}$$

By applying the unscented transformation on the dynamic model function in equation (3.132) we get the predicted mean $\bar{\boldsymbol{p}}_t$ and covariance $C_{\vec{p}\vec{p},t}$ as well the transformed sigma points $\vec{\boldsymbol{\chi}}^{[i]}$ after the prediction. Using the non-linear observation function $\boldsymbol{z} = \boldsymbol{f}(\boldsymbol{p})$ of the Gauss-Markov-model we can compute sigma points for the predicted observations $\vec{\boldsymbol{z}}^{[i]}$, their mean $\bar{\boldsymbol{z}}$ and the predicted covariance $C_{\vec{z}\vec{z}}$. We observe, that $\bar{\boldsymbol{z}}$ is similar to $\boldsymbol{f}(\vec{\boldsymbol{p}}_t)$ in equation (3.167) and $C_{\vec{z}\vec{z}}$ is similar to the predicted observation covariance $AC_{\vec{p}\vec{p}}A^{\mathsf{T}}$ in equation (3.163). According to the Kalman filter we can compute a gain matrix using the cross-correlation similar to $C_{\vec{p}\vec{p},t}A^{\mathsf{T}}$ by

$$C_{\chi\vec{z}} = \sum_i W^{[i]} (\vec{\boldsymbol{\chi}}^{[i]} - \bar{\boldsymbol{p}}_t)(\boldsymbol{z}^{[i]} - \bar{\boldsymbol{z}})^{\mathsf{T}} \tag{3.174}$$

and the additive observation covariance C_{zz} by

$$F = C_{\chi\vec{z}}(C_{zz} + C_{\vec{z}\vec{z}})^{-1}. \tag{3.175}$$

Then a final update can be derived by

$$\boldsymbol{p}_t = \vec{\boldsymbol{p}}_t + F(\boldsymbol{z} - \bar{\boldsymbol{z}}) \tag{3.176}$$

with its covariance using equation (3.150)

$$C_{pp,t} = C_{\vec{p}\vec{p},t} - F(C_{zz} + C_{\vec{z}\vec{z}})F^{\mathsf{T}}. \tag{3.177}$$

The *Unscented Kalman filter* above has two advantages. First, the Gaussian approximation of the transformed uncertainty through the non-linear transformation take higher order therms up to degree four of a Taylor expansion into account. Second, the computational complexity is equal to the Kalman filter but parallelizable. But there are also disadvantages. First, the update scheme does not ensure a positive definite covariance matrix $C_{pp,t}$. A solution of this disadvantage is the *Square root Kalman filter*, which ensure positive semi-definiteness. Second, the non-linear functions must be continuous. We will explain this last issue in chapter 4 in more detail.

3.4.4 Outlier detection and robustification of Kalman filtering

As we mentioned the update process is a weighted mean of the direct observation of the state obtained by the dynamic model and a set of noisy observations. Comparing it to the least square outlier detection in section 3.3.2 this mean process can be reformulated using the normal equations. Therefore the predicted state vector is a pseudo observation vector with full correlations, so that from the theory of outlier detection we can only test the whole pseudo observation as a group. Additionally, the unknowns in the state vector depends on groups of observations of the measurement model. Following the considerations in 3.3.2, it is not possible to distinguish between gross errors in the prediction and gross errors in the observations.

However, in some real applications the mean and covariance of the predicted state approximate the true state very well. If and only if we can neglect the influence of the prediction we can test the observations against gross errors using equation (3.111) in the Gauss-Markov-model (3.113) in the Gauss-Helmert-model respectively.

Therefore, the residuals for the recursive update using the Gauss-Markov-model have to be obtained at the estimated state using

$$\boldsymbol{v} = \boldsymbol{z} - \boldsymbol{f}(\widehat{\boldsymbol{p}}). \tag{3.178}$$

The contradictions for the recursive update in the Gauss-Helmert-model can be tested using equations (3.112) and (3.113) at the convergence point.

Similar to the reweighting method for the least square method the reweighting scheme is applicable to the recursive update using equation (3.116). A method to estimate the variance components of individual groups of observations can be found in Caspary (1998).

In this chapter we first summarized basic concepts in image geometry followed by a derivation of global least square methods. We introduced the notion of dynamic state systems as a general Bayes filter and derived similar to the global least square method, a new formulation of an implicit recursive estimation scheme. We showed that the classical least square recursive estimation algorithm is a specialization of our new derivation. Furthermore, we explained how outliers can be rejected and their influence can be reduced to achieve more robust estimation with respect to erroneous observations inside the global least square as well as in the recursive estimation scheme.

Chapter 4

Kalman filter based localization and mapping with a single camera

The goal of this chapter is to work out the details of a Kalman filter based system for the task of localization and mapping for a single camera only. We are especially focused on the state representation and initialization.

Full autonomous simultaneous localization and mapping using image based measurements is still a hard problem. A broad range of applications, especially in the field of robotics and surveillance, require fully autonomous, robust, accurate and real-time capable methods. The goal of such applications is to acquire detailed map information. This includes an adequate surface model and the determination of the full observer localization at every point in time.

In the following we will discuss the requirements of such systems using a single camera sensor only and we will focus on the problem of real-time capable recursive estimation of the environment. As motivated in the introduction the environment is focused on the task of acquiring elevation maps from an unmanned aerial vehicle (UAV).

4.1 Introduction

Unmanned aerial vehicles become more and more interesting as a tool for medium and large scale image acquisition. To determine a dense surface model of an observed scene the camera orientations of the acquired images in a fixed world coordinate system are necessary. Classical offline approaches are well-known in the photogrammetric community. In this case the images are acquired with large baselines, information about the projection center and camera orientation are known from available GPS and gyro measurements or are reconstructed automatically from image measurements. In the past two decades automatic image measurement and image to image matching techniques Mikolajczyk & Schmid (2005) have become available, which allow a fully automatic orientation process without any user interaction. Besides, the computer vision and robotic community invented techniques for short-range and small baseline image sequences. Driven by real-time requirements of the orientation determination of a camera setup, which is not limited in the number of cameras, several solutions especially for small local maps were proposed.

In all cases visual measurements and estimation techniques to determine metric informations about the camera orientation and the map are used. For small baseline image sequences the proposed task is called visual simultaneous localization and mapping (V-SLAM). In the following we will highlight the background of such a system requirement using an image sequence acquired by a single camera with an unmanned aerial vehicle. Additional sensors like GPS and INS can always be integrated as direct or relative observations to an arbitrary metric estimation technique. However, in some environments and caused by hardware restrictions this information is not available. From this motivation, we will ignore this measurement in the proposed models.

The task of simultaneous localization and mapping can be understood as a discrete time driven dynamic process. Based on the scheme of figure 4.1 we will explain the meaning of the needed components.

- Initialization module

  Initializing a real-time system is a problem of its own and different than the online update. It is also required if the system has failed and a reinitialization is necessary. From the nature of time dependent processes, there is a beginning at time t_0. At this time we may have information of a particular fragment of the environment. This prior information includes information about possible velocities and accelerations of the camera and distances to objects or full coordinates of control points in an arbitrary local or world coordinate system. The initialization module usually fixes the reference system including the initial camera orientation and the environment scale as well as their uncertainties.

- Metric estimation module

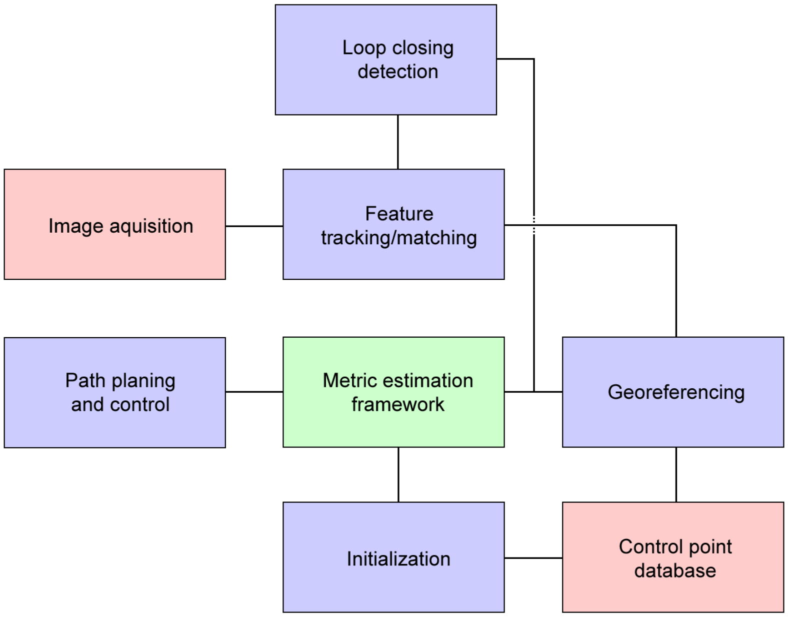


Figure 4.1: System components and connections for the task of simultaneous localization and mapping using a single camera

According to the dynamic process and the acquired observations the estimation framework provides the knowledge of current and if needed past camera orientations as well as map information. The module incorporates observations by updating the represented knowledge in a statistical manner. Map information in our context is represented by parameterized objects, for instance coordinates in a world reference system. In addition, knowledge about its absolute and relative uncertainty has to be represented and provided. In our case this module will be realized using a Kalman filter estimation framework.

- Feature tracking and/or matching module

  Features are defined as image observations which are associated with objects in the environment. Here these objects are observed by a single camera. The goal of this module is to automatically identify features and track them across the image sequence.

- Loop closing detection module

  In case the feature tracking fails or an object is revisited at a later point in time this module associates the feature with a previously observed object. Usually this will be solved using descriptors for features or feature groups.

- Georeferencing module

  Depending on the a-priori knowledge introduced by the initialization module a later georeferencing stage can be applied, if there are observations available to achieve absolute orientations in a fixed world coordinate system.

- Path planing and control module

  Depending on the task and the a-priori knowledge of the autonomous system this module generates control actions to achieve a complete mapping. In the case of mapping with unmanned aerial vehicles often the path is fixed a-priori using an approximate knowledge of the surface and of the achievable accuracy.

The aim of an implementation of the scheme above is to provide an autonomous localization and navigation using visual information only. In our opinion the core module is the metric estimation framework. The requirements of this framework are real-time capability, high accuracy and robustness. In this section we will therefore focus on the estimation framework, its initialization and required observations. To achieve real-time capable estimation of a set of metric parameters the Kalman filter framework will be our choice. We will derive and discuss different kinds of state representations as well as their initialization and dynamic evolution in time.

4.2 Feature extraction, tracking and matching

First we will clarify the meaning of *features*. In general images are only light intensity information in different directions. The image is a carrier of such information. Known from the information theory the information can be measured globally or locally by its entropy. But, we cannot decide by only one image whether its information is high or low. It depends on our task.

We give a small example: Assuming we already mapped a large green area like a golf field. We get an image of a homogeneous bright surface. The information inside the image can be low according to its entropy, but the information for the localization task can be high according to the identification of a sand strap.

From this consideration, in our opinion we have to distinguish between *landmarks* usable as *control points*, which are high level location based objects or information in a global map and *features* as pure image based metric objects usable as *tie points*. Features for instance are interest points, blobs[1] and line segments. Moreover, features can also be identifiable objects, for instance cars or windows. In our experiments we will only use interest points as features, but every kind of feature can be used for our task. A point based feature should fulfill certain criteria:

1. Invariance - The detection of a specific feature should be invariant with respect to geometric and radiometric distortions, for instance relative rotations or intensity changes.

2. Stability - The detection of a feature should be robust against noise in the observation.

3. Distinctness - The feature should be distinguishable from neighboring features in terms of local image information.

4. Infrequency - For the task of loop closing detection the surrounding local image information of a feature should be unique.

5. Interpretability - In case of object recognition tasks it is necessary that the feature or a feature group can be assigned to semantic objects.

In the past decade many point based interest operators were developed Förstner & Gülch (1987), Harris & Stephens (1988), Lindeberg (1998), Mikolajczyk & Schmid (2002), Lowe (2003). We will use interest points according to Förstner/Harris, which are suitable for matching. Matching here means to identify the same feature, which is associated to a location in the world space, in a following image. There are two situations possible. First the following image was taken with small disparities, then tracking algorithms can be used for matching. Second, a feature is revisited and has to be identified.

Tracking algorithms assume small disparities as well as small scale and rotation changes. The surrounding local image information of a feature will be used as a template. The assumption is, that the corresponding object point is locally planar. It has been shown by Förstner (1982), that the accuracy of template based matching using least square methods depends on the signal noise ratio and on window size around a feature point. Using a large window size, the planar assumption can be corrupted and therefore it is necessary to find a trade off between accuracy and local planarity of a feature. A well known tracking algorithm

[1] Blobs are homogeneous image areas.

for distinctive feature points was developed by Tomasi & Kanade (1991)[2], which combines interest points and least square matching. Tracking algorithms for line segments are also available, for instance Chiba & Knade (1998).

The identification of a revisited feature is much more complicated. The environment usually does not guarantee the uniqueness of a feature. Furthermore, the number of features can be huge. Hence, common techniques use information about the image neighborhood of a feature using descriptors Mikolajczyk & Schmid (2005) and relaxations to other features Herbert Bay & Gool (2005) as well as geometric information about their location in a world coordinate frame. The re-identification is closely related to the loop closing detection. In this thesis we do not focus on this aspect.

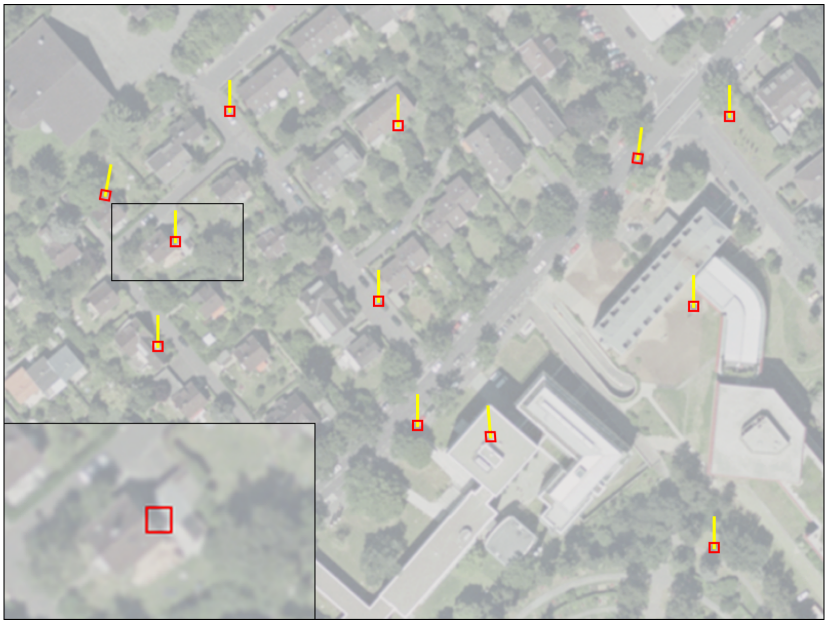


Figure 4.2: Example for point based feature tracking assuming small disparities in consecutive frames. The red box marks an extracted feature, the yellow line is the displacement vector of the feature to the consecutive image. In the lower left corner a detail of the image is enlarged.

In the following we use an implementation of a point based feature tracking of the *OPENCV*[3] library, which is a realization of the KLT-Tracker by Tomasi & Kanade (1991). To overcome the limitation of small disparities, the feature disparity will be obtained by template matching through a Gaussian scale space of the image.

[2] In the literature the reader can find this tracking algorithm searching for KLT-Tracker (*Kanade-Lucas-Thomasi*).

[3] http://www.opencv.org

4.3 State representation

According to the scheme of the dynamic Bayes filter in section 3.4 the state $\boldsymbol{p}_t$ contains the actual knowledge about the known world at time t. This knowledge is encoded as metric information in a set of variables in $\boldsymbol{p}_t$. Its uncertainties and their correlations using a Gaussian uncertainty approximation are encoded in the covariance matrix $\mathsf{C}_{pp,t}$. The state vector is partitioned into camera orientation and map information. In this section we will derive different kinds of camera state and map state representations, the non-linear dynamic state changes and the observation model. We will discuss the filter initialization and how new map parameters can be initialized.

Equivalent to the classical aerial photogrammetry we will first define the camera coordinate system as a right hand system with the negative Z-component in view direction. In case of a photogrammetric strip the typical flight direction is in X-direction, because most digital image sensors are not square and the width is in column direction and larger than the height of the sensor. We will define the measurement in the pixel systems as $\boldsymbol{x} = \{x, y\}$. Therefore,

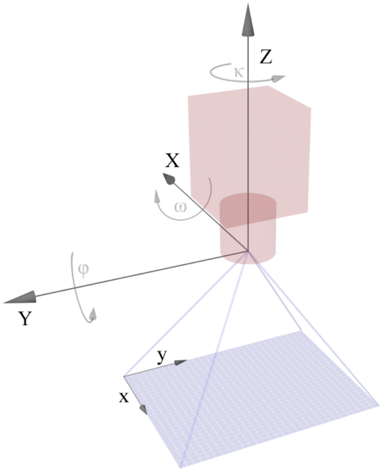


Figure 4.3: Camera coordinate system definition and image plane in negative Z direction.

the calibration matrix can be build in the following way

$$\mathsf{K} = \begin{bmatrix} c & 0 & x_h \\ 0 & c & y_h \\ 0 & 0 & 1 \end{bmatrix}, \tag{4.1}$$

with x_h and y_h as the principle point in row and column direction (c. f. figure 4.3) and $c < 0$

as the principle distance. To achieve the rotation between the image and the camera system we multiply K right hand side by a rotation matrix $R(\kappa = 180)$ around the Z-component.

4.3.1 Single camera state

The camera orientation in a 3-dimensional space is defined by using six parameters, its location and its angular orientation. As mentioned in section 3.1.2 an angular representation using Euler angles involves trigonometric functions and can fail in case of deviations close to the 180 degree singularity. For this reason the angular orientation will be represented using quaternions. A convenient effect is that the dynamic state change equation becomes linear. Therefore the actual state can be represented as follows:

$$\boldsymbol{p}_t = [\boldsymbol{T}_t, \mathbf{q}_t, \boldsymbol{X}_1, \ldots, \boldsymbol{X}_n, \ldots, \boldsymbol{X}_N]^{\mathsf{T}} \tag{4.2}$$

with $\boldsymbol{T}_t = \{X_t, Y_t, Z_t\}$ as the camera position and $\mathbf{q}_t$ as the angular camera orientation represented as quaternion. Additionally the state contains N object point coordinates. Following the state representation model of Davison (2003) the state also contains the velocity and angular velocity.

$$\boldsymbol{p}_t = \left[\underbrace{\boldsymbol{T}_t, \mathbf{q}_t, \dot{\boldsymbol{T}}_t, \dot{\boldsymbol{q}}_t}_{13}, \underbrace{\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n, \ldots, \boldsymbol{X}_N}_{3N}\right]^{\mathsf{T}} \tag{4.3}$$

We denote, that the angular velocity $\dot{\boldsymbol{q}}_t$ is usually small and below 180°. In this case we can represent the velocity as the vector part only as a three component vector, c. f. equation (3.36). As an extension also the acceleration can be added. The scheme of a single camera state is outlined in figure 4.4.

In the following, we will present a linear dynamic model and the error propagation using accelerations as a perturbation with mean zero. The derivation can be found in the appendix in section A.2. The prediction of the state according to a time difference Δt can then be computed as

$$\vec{\boldsymbol{p}}_{t+1} = \boldsymbol{h}(\boldsymbol{p}_t) = \begin{bmatrix} \boldsymbol{T}_t + \dot{\boldsymbol{T}}_t \Delta t + \frac{1}{2}\ddot{\boldsymbol{T}}_t \Delta t^2 \\ \Upsilon(\frac{1}{2}\ddot{\boldsymbol{q}}_t \Delta t^2)\Upsilon(\dot{\boldsymbol{q}}\Delta t)\mathbf{q}_t \\ \dot{\boldsymbol{T}}_t + \ddot{\boldsymbol{T}}_t \Delta t \\ \ddot{\boldsymbol{q}}_t \Delta t + \dot{\boldsymbol{q}} \\ \boldsymbol{X}_1 \\ \vdots \\ \boldsymbol{X}_N \end{bmatrix}, \tag{4.4}$$

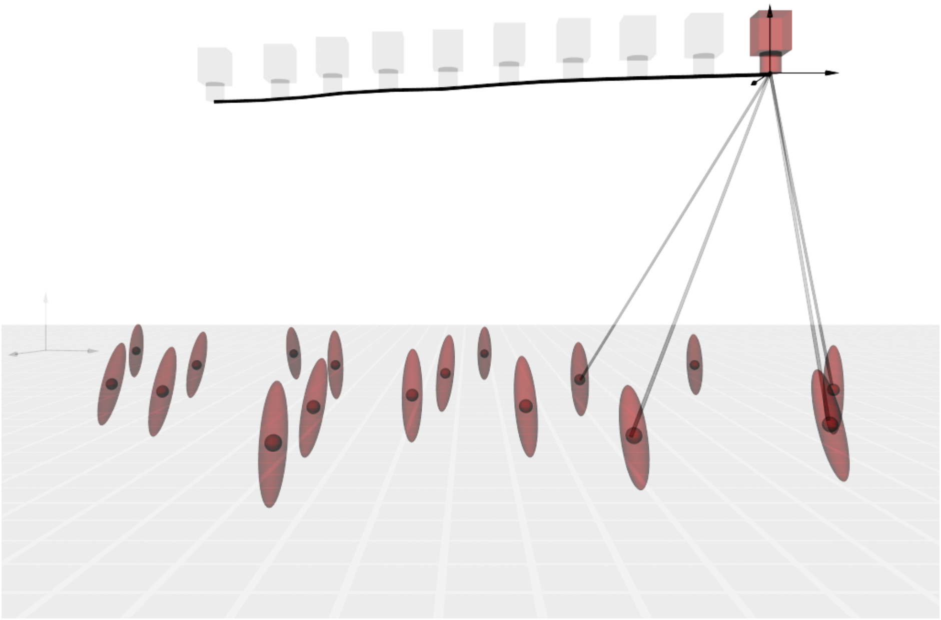


Figure 4.4: Camera state representation scheme using a single camera state

where $\dot{\boldsymbol{q}}$ and $\ddot{\boldsymbol{q}}$ are the angular velocity[4] and angular acceleration represented as the vector part of a quaternion only (c. f. equation (3.36)). Note, the expression $\Upsilon(\dot{\boldsymbol{q}})$ can be written as $\Upsilon([1, \dot{\boldsymbol{q}}^{\mathsf{T}}])$. The unknown accelerations $\ddot{\boldsymbol{T}}_t$ and $\ddot{\boldsymbol{q}}_t$ are assumed to be Gaussian distributed with mean zero. The prediction of the covariance $C_{\vec{p}\vec{p}}$ following equation (3.133) can then be expressed with the Jacobians H and W as

$$\underset{13+3N\times 13+3N}{H} = \begin{bmatrix} I_3 & 0 & \Delta t \boldsymbol{I} & 0 & 0 \\ 0 & \Upsilon(\dot{\boldsymbol{q}}\Delta t) & 0 & \breve{\Upsilon}(\mathbf{q}_t)[\mathbf{0}_3|\Delta t I_3]^{\mathsf{T}} & 0 \\ 0 & 0 & I_3 & 0 & 0 \\ 0 & 0 & 0 & I_3 & 0 \\ 0 & 0 & 0 & 0 & I_{3N} \end{bmatrix} \tag{4.5}$$

[4] The derivation of the angular velocity can be comprehended in section 3.1.2.

and for the system noise

$$\underset{13+3N\times 6}{\mathsf{W}} = \begin{bmatrix} \frac{1}{2}\Delta t^2 \mathsf{I}_3 & \mathit{0} \\ \mathit{0} & \breve{\Upsilon}(\Upsilon(\dot{\boldsymbol{q}}\Delta t)\mathbf{q}_t)[\mathbf{0}_3|\frac{1}{2}\Delta t^2 \mathsf{I}_3]^{\mathsf{T}} \\ \Delta t \mathsf{I}_3 & \mathit{0} \\ \mathit{0} & \Delta t \mathsf{I}_3 \\ \mathit{0} & \mathit{0} \end{bmatrix}. \tag{4.6}$$

We observe, that using quaternions and assuming small rotations, matrix multiplications without any trigonometric functions are necessary.

4.3.2 Sliding window camera state

It is well known from an entire bundle adjustment with all camera orientations, arbitrary camera parameters are correlated. Using the single camera state representation presented so far we neglect past camera orientations as well as their correlations. For this reason there is an leakage of information. The correlation between close camera locations is high and should not be neglected. Furthermore, the map will be updated by new observations, which will indirectly influence past camera parameters, which are useful for an off-line dense map computation. Therefore we propose a multiple camera state as outlined in figure 4.5. In the following we will show how the linear prediction model can be applied to a multi camera state representation. Because we have take into account the correlations between all the camera orientations and the estimated object points, the error propagation in the prediction step has to be performed using a state extension. We extend the state to the new camera position at time $t+1$ computed from the state at time $\boldsymbol{p}_t$. Let us assume we take k sets of camera

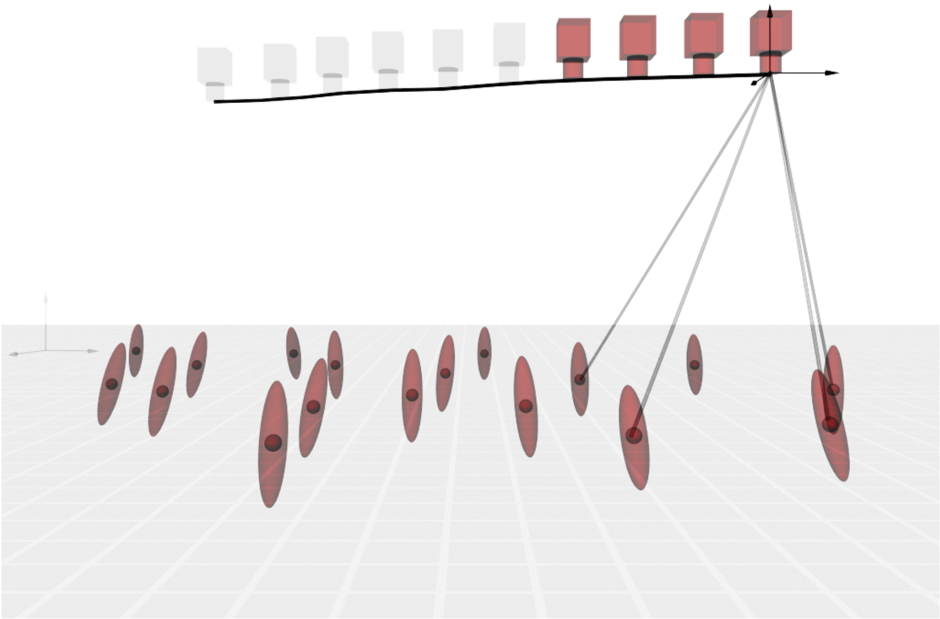


Figure 4.5: Camera state representation scheme using a multiple camera state

parameters in the state vector, then the state vector and the predicted state vector are

$$\boldsymbol{p}_t = \begin{bmatrix} \boldsymbol{T}_t \\ \mathbf{q}_t \\ \boldsymbol{T}_{t-1} \\ \mathbf{q}_{t-1} \\ \vdots \\ \boldsymbol{T}_{t-k} \\ \mathbf{q}_{t-k} \\ \boldsymbol{X}_1 \\ \vdots \\ \boldsymbol{X}_n \\ \vdots \\ \boldsymbol{X}_N \end{bmatrix} \qquad \vec{\boldsymbol{p}}_{t+1} = \boldsymbol{h}(\boldsymbol{p}_t) = \begin{bmatrix} \boldsymbol{T}_t + (\boldsymbol{T}_t - \boldsymbol{T}_{t-1})\Delta t + \frac{1}{2}\ddot{\boldsymbol{T}}_t \Delta t^2 \\ \Upsilon(\frac{1}{2}\ddot{\boldsymbol{q}}_t \Delta t^2)\Upsilon(U_{\Delta t}\Upsilon(\mathbf{q}_t)V\mathbf{q}_{t-1})\mathbf{q}_t \\ \boldsymbol{T}_t \\ \mathbf{q}_t \\ \boldsymbol{T}_{t-1} \\ \mathbf{q}_{t-1} \\ \vdots \\ \boldsymbol{T}_{t-k} \\ \mathbf{q}_{t-k} \\ \boldsymbol{X}_1 \\ \vdots \\ \boldsymbol{X}_n \\ \vdots \\ \boldsymbol{X}_N \end{bmatrix}, \qquad (4.7)$$

where $\Delta t = \frac{t_{t+1}-t_t}{t_t-t_{t-1}}$ is a time dependent ratio factor. The extended covariance matrix can be obtained using the error propagation for the prediction step using the Jacobian

$$\underset{7(k+2)+3N\times 7(k+1)+3N}{H} = \begin{bmatrix} I_3+\Delta t I_3 & 0 & -\Delta t I_3 & 0 & 0 \\ 0 & \begin{matrix}\Upsilon(U_{\Delta t}\Upsilon(\mathbf{q}_t)V\mathbf{q}_{t-1})+ \\ \breve{\Upsilon}(\mathbf{q}_t)U_{\Delta t}\breve{\Upsilon}(V\mathbf{q}_{t-1})\end{matrix} & 0 & \breve{\Upsilon}(\mathbf{q}_t)U_{\Delta t}\Upsilon(\mathbf{q}_t)V & 0 \\ I_3 & 0 & 0 & 0 & 0 \\ 0 & I_4 & 0 & 0 & 0 \\ 0 & 0 & I_3 & 0 & 0 \\ 0 & 0 & 0 & I_4 & 0 \\ 0 & 0 & 0 & 0 & I_{7(k-1)+3N} \end{bmatrix} \tag{4.8}$$

and for the system noise

$$\underset{7(k+2)+3N\times 6}{W} = \begin{bmatrix} \frac{1}{2}\Delta t^2 I_3 & 0 \\ 0 & \breve{\Upsilon}(\Upsilon(U_{\Delta t}\Upsilon(\mathbf{q}_t)V\mathbf{q}_{t-1})\mathbf{q}_t)[\mathbf{0}|\frac{1}{2}\Delta t^2 I_3] \\ 0 & 0 \end{bmatrix} \tag{4.9}$$

with U and V from equations (3.37) and (3.35). The derivation is outlined in the appendix in section A.2.

Up to now, the state is expanded by the new camera orientation. If intended, the camera parameter for the camera at time $t-k$ can be neglected. To do so, we delete the corresponding rows and columns of the covariance respectively of the state vector. For real implementations the memory reordering for the covariance matrix can be performed using an error propagation scheme with the Jacobian

$$\underset{7(k+1)+3N\times 7(k+2)+3N}{J} = \begin{bmatrix} I_{7(k+1)} & 0 & 0 \\ 0 & 0 & I_{3N} \end{bmatrix}. \tag{4.10}$$

Discussion: One can show, that the single camera representation including velocities is equivalent to a multi-camera state representation with $k=1$, by using error propagation one representation can be transformed to the other one and vice versa. In case of large system noise the dynamic state change model will not cause correlations between consecutive camera orientations. But correlations transfered to past camera orientations across the map state further on exist. The overhead of the computational complexity using a sliding window camera state is usually small. An implementation should decide which past camera orientations have to stay inside the state vector. This decision should be taken into account which past cameras are useful for off-line dense map computation and the degree of correlations.

4.3.3 Feature representation and update models

In this section we will describe the representation of the object point coordinates and formulate the corresponding update or observation model which can be used in the bundle adjustment and in the Kalman filter based solution.

Euclidean representation. In the classical case a point in the object space will be represented by its Euclidean coordinates $\boldsymbol{X}_n = \{X_n, Y_n, Z_n\}$. As shown in section 3.2.1 the observation model can be described by the colinear equation (3.59) or (3.58). Using equation (3.58) the projection matrix and therefore the projection of all visible points X_n in frame t can be computed as

$$\mathbf{x}_n = \mathsf{K}R^{\mathsf{T}}(\mathbf{q}_t)[I_3| - \boldsymbol{T}_t]\mathbf{X}_n \tag{4.11}$$

from the predicted state vector representation of the camera position and angular orientation. Note, that in the case of the bundle adjustment the over-representation of the camera orientation using the quaternion cause a rank deficiency. To avoid this rank deficiency in the bundle adjustment two solutions are available. First, we can formulate the constraint, that the length of the quaternion is equal to one. Second we can introduce a pseudo observation with a high weight for their length. In a Kalman filter based approach there is no inversion of a rank deficient matrix necessary. But, the free length of the quaternion can cause numerical instabilities. The quaternion length can grow up to large values, which cause numerical instability. In contrast, we do not know any method for a constraint on the state vector in a Kalman filter similar to the global least square method. Instead we can use a pseudo observation for the quaternion length in arbitrary time steps.

Inverse distance representation - The inverse distance representation was introduced by Montiel *et al.* (2006). Instead of a Euclidean representation of an object point, the point will be represented using polar coordinates with an inverse distance. It has been shown and quantified that for small parallax angles the distribution of the intersection in Euclidean space is not well modeled. This can be proven by a Monte-Carlo simulation shown in figure 4.7. The advantage of an inverse distance representation is a better approximation of the distribution of an object point. The goal of this parametrization is to reduce the effect of a bad Gaussian approximation of the object point location in case of small parallaxes between consecutive frames in an early life time of the object point. This representation uses six parameters to determine an object point. The first three components are the location of a reference point. This reference point X_r can be obtained by the camera location for the first visibility of the object point. The following three components are the polar coordinates relative to the orientation of the object coordinate system (c. f. figure 4.6). The distance will be represented by the inverse distance ρ, so that the distance is $d = 1/\rho$.

$$^{i}\boldsymbol{X}_n = \begin{bmatrix} X_r \\ Y_r \\ Z_r \\ \theta \\ \zeta \\ \rho \end{bmatrix}_n \tag{4.12}$$

A Euclidean object point $\boldsymbol{X}$ can be obtained from its inverse distance representation by

$$\boldsymbol{X} = \begin{bmatrix} X_r + \frac{1}{\rho}\sin\zeta\cos\theta \\ Y_r + \frac{1}{\rho}\sin\zeta\sin\theta \\ Z_r + \frac{1}{\rho}\cos\zeta \end{bmatrix}. \tag{4.13}$$

The measurement model can be performed using equation 4.13 and 4.11 in a chain.

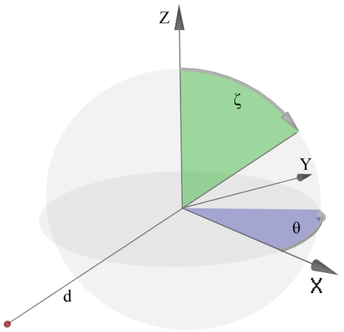


Figure 4.6: Polar coordinate definition for the inverse distance representation.

Given a 6 × 6 covariance $C_{^iX\,^iX}$ of a point $^{i}\boldsymbol{X}$ in inverse distance representation the covariance of a Euclidean point representation can be obtained using linear error propagation.

In section 4.5 we will explain, why the error propagation using the unscented transformation can fail. The inversion of the transformation (4.13) will be derived in section 4.3.4.3.

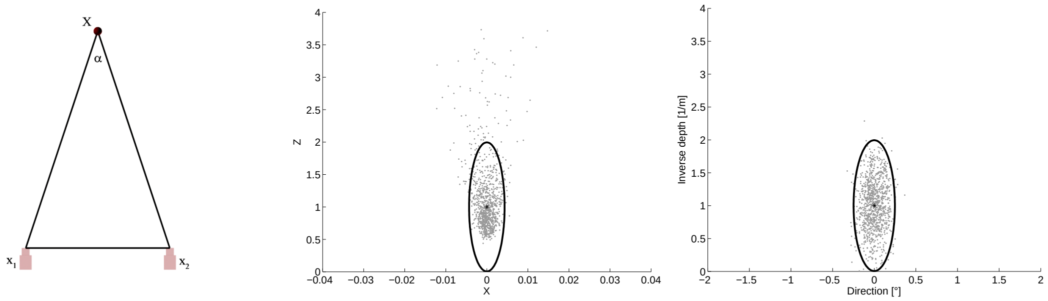


Figure 4.7: **Left:** Simulation scheme of a point reconstruction with small parallax angle (0.5°) and 1000 samples, **Middle:** Distribution in Euclidean space with linear covariance propagation. **Right:** Distribution in the inverse distance space with linear covariance propagation.

4.3.4 The initialization problem

When the discrete time process starts at time t_0 we have to initialize the parameter vector $\boldsymbol{p}$ and its covariance C_{pp} with reasonable initial values. This initialization fixes the gauge of the state space, which is defined by an orientation in the 3-dimensional space and a scale of the space. Because the camera is not stationary the gauge is encoded in the map coordinates themselves. The initialization depends on the availability of at least three control points visible and identifiable in the first image. In this case the control points will be introduced with their uncertainties as the initial map. On the other hand if there are no control points available, there is no information about a scale of the world space, but an arbitrary gauge for the orientation can be fixed on three arbitrary observed object points. Both methods will be explained in more detail in this section. Furthermore, at every time of the process new object points will be visible and have to be initialized. The initial coordinates appears as the linearization point for the non-linear observation model and have to be chosen correctly as much as possible. Their covariance should be infinite[5], because the initialization is a direct observation of the objects coordinate and is therefore information introduced in the stochastical representation.

[5] Covariance matrix entries cannot be infinite. The covariance of the new object point coordinate should be huge in contrast to the covariance of the camera orientation.

4.3.4.1 Control point initialization

The initialization of the filter can be performed using at least three control points. Usually these control points can be obtained by a-priori global or local map information, e.g. from an independent absolute measurement. Their coordinates, uncertainties as well as their correlations will be used to initialize the map coordinates inside the filter.

The initial camera parameter can be obtained using the observations of the control points in the first image. The advantage of a direct approach is that we do not need any approximate values a-priori. Using three points only few direct approaches are available to determine the distances of the camera projection center to the control points. A good overview can be found in Haralick *et al.* (1991), who analyzes the numerical stability of different approaches. These approaches result in up to four equivalent solutions. We need a fourth point to pick up the correct one.

The camera position $\boldsymbol{T}$ and the angular orientation R can be obtained solving the model

$$\lambda_i \boldsymbol{m}_i = R(\boldsymbol{X}_i - \boldsymbol{T}) \tag{4.14}$$

with $\boldsymbol{m}_i$ as the normalized directions to the control points $\boldsymbol{X}_i$ and λ_i as the distances to the control points already obtained by the direct solution. The algorithm for the determination of $\boldsymbol{T}$ and R can be found in McGlone *et al.* (2004) pages 786ff.

The covariance matrix for the camera parameter will be initialized with a huge uncertainty. In the following Kalman filter measurement update the camera parameter will be corrected.

4.3.4.2 Scale free bundle adjustment initialization

In some real environments no control points and no additional information about the visible world space are available. To make the Kalman filter operable and to fix the gauge we propose a free bundle adjustment initialization procedure in the early stage of the process. To do so, we have to decide which of the obtained frames we will use for the initialization.

Here we will derive a real-time capable algorithm to decide when the acquired images are adequate to initialize the estimation process, c. f. Beder & Steffen (2006). The contribution is to present a statistically motivated measure for the quality of a bundle adjustment initialization procedure.

We will now analyze the uncertainty of an object point from two measured corresponding image coordinates. Observe that we will neglect the correlation between a set of points. If a scene point $\mathbf{X}$ is observed by two projective cameras P_1 and P_2, the image coordinates $\mathbf{x}_1$ and $\mathbf{x}_2$ can be computed using equation (3.58) which can be rewritten using the skew-symmetric matrix reflecting the cross product as

$$\mathbf{0} = \mathbf{x} \times \mathsf{P}\mathbf{X} = \mathsf{S}(\mathbf{x})\mathsf{P}\mathbf{X} = -\mathsf{S}(\mathsf{P}\mathbf{X})\mathbf{x}. \tag{4.15}$$

Only two constrains are independent. Using (4.15) for two observations and choosing two arbitrary constrains indicated by $\mathsf{S}_2(\cdot)$, both expressions are linear in the object point as well as in the image point, so that follows

$$\underbrace{\begin{bmatrix} \mathsf{S}_2(\mathbf{x}_1)\mathsf{P}_1 \\ \mathsf{S}_2(\mathbf{x}_2)\mathsf{P}_2 \end{bmatrix}}_{\underset{4\times 4}{A}} \mathbf{X} = \mathbf{0} \tag{4.16}$$

and

$$\underbrace{\begin{bmatrix} -\mathsf{S}_2(\mathsf{P}_1\mathbf{X}) & \mathbf{0} \\ \mathbf{0} & -\mathsf{S}_2(\mathsf{P}_2\mathbf{X}) \end{bmatrix}}_{\underset{4\times 6}{B}} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} = \mathbf{0}. \tag{4.17}$$

Now the scene point coordinate and the two image point coordinates are assumed to be random variables with Gaussian distribution. We observe that, as all three quantities are homogeneous, the covariance matrices of their distributions are singular. In case the covariances of the image points are given by C_{11} and C_{22}, then it has been shown by Heuel (2004) according to our derivation of the normal equation system in section 3.3, that the covariance matrix C_{XX} of an object point is proportional to the upper left 4×4-submatrix of the inverse of

$$\mathsf{N} = \begin{bmatrix} A^\mathsf{T} \left[B \begin{bmatrix} C_{11} & \mathbf{0} \\ \mathbf{0} & C_{22} \end{bmatrix} B^\mathsf{T} \right]^{-1} A & \mathbf{X} \\ \mathbf{X}^\mathsf{T} & 0 \end{bmatrix}. \tag{4.18}$$

We have neglected the effect of the uncertainty of the projection matrices P here, because the relative orientation of two cameras is determined by many points, so that it is of superior precision compared to a single object point. In the second step we can compute the Euclidean covariance of $\boldsymbol{X}$ indicated by ${}^{e}C_{XX}$ using the error-propagation with the Jacobian

$${}^{e}J = \frac{\partial \boldsymbol{X}_0 / X_h}{\partial \boldsymbol{X}_0} = \frac{1}{X_h} \left[I_3 \mid -\frac{\boldsymbol{X}_0}{X_h} \right] \tag{4.19}$$

according to

$$^{e}\mathsf{C}_{XX} = {}^{e}\mathsf{J}\mathsf{C}_{XX}\,{}^{e}\mathsf{J}^{\mathsf{T}}. \tag{4.20}$$

The roundness as a measure for the reconstruction stability is directly related to the condition number of the 3d reconstruction of the object point. Hence, apply the singular value decomposition[6] to the covariance

$$^{e}\mathsf{C}_{XX} = \mathsf{G}\begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix}\mathsf{G}^{\mathsf{T}} \tag{4.21}$$

the roundness L is defined as the square root of the quotient of the smallest and the largest singular value

$$L = \sqrt{\frac{\lambda_3}{\lambda_1}}. \tag{4.22}$$

This measure lies between zero and one, is invariant to scale changes and only depends on the relative geometry of the camera pose and on the measured image point accuracy.

The algorithm for a scale free bundle adjustment initialization is outlined in 4.1.

4.3.4.3 Inverse distance feature initialization and reduction

As shown in 4.3.3 an object point can be represented using polar coordinates with respect to a reference system. Here we will show, how the object point can be initialized using the inverse distance parametrization and explain the introduction into the state vector and its covariance matrix. Furthermore we will describe the transformation from the inverse distance representation to the Euclidean representation.

The first time a new object point is visible the point can be introduced to the state vector. Because we will fix the reference polar coordinate system to the actual camera position, we have to take into consideration to its uncertainty. The direction to the object point can be determined using image observation.

Using the estimated camera orientation R_t at time t and the observed image point $\mathbf{x}$ in homogeneous representation we obtain the direction vector $\boldsymbol{d} = \{d_X, d_Y, d_Z\}$ to the object point by

$$\boldsymbol{d} = -\mathsf{R}_t\mathsf{K}^{-1}\mathbf{x}. \tag{4.23}$$

[6] Note, the value decomposition of a covariance matrix leads to a symmetric solution, where G is a rotation matrix.

Algorithm 4.1 Scale free bundle adjustment initialization algorithm

```
Fix the first image of the sequence and let its projection matrix be $\mathsf{P}_{t=1} = \mathsf{K}[\mathit{I}_3|\mathbf{0}]$
Extract the interest points $\mathbf{x}_{1,n}$ together with their covariance matrices $\mathit{C}_{11,n}$ from this image
repeat
    Get the next image at time $t = t + 1$
    Extract the interest points $\mathbf{x}_{t,n}$ together with their covariance matrices $\mathit{C}_{tt,n}$ from this new image
    Determine the point correspondences $\mathbf{x}_{1,n} \leftrightarrow \mathbf{x}_{t,n}$ and the relative orientation $\mathit{R}_t$, $\boldsymbol{T}_t$ to the first image of the sequence according to the algorithm proposed in Nistér (2004). The camera matrix for the current image is then $\mathsf{P}_t = \mathsf{K}\mathit{R}_t^{\mathsf{T}}[\mathit{I}_3| - \boldsymbol{T}_t]$ with $|\boldsymbol{T}_t| = 1$.
    Determine the scene point positions $\mathbf{X}_n$ for each found correspondence by forward intersection according to the linear system in equation (4.16) using the singular value decomposition as an linear equation solver.
    Determine the roundness $L_n$ (c. f. equation (4.22)) of each scene point $\mathbf{X}_n$'s confidence ellipsoid as outlined.
until the mean roundness is above a given threshold $\Theta_L$ for example 0.1.
Compute approximate values for the camera orientations $P_{2..t-1}$ by spatial resection using the scene points $\boldsymbol{X}_n$.
Perform a bundle adjustment with all images up to time $t$ by fixing the gauge on the first camera and the base line length to the camera at time $t$ with one.
Use the reconstructed point cloud and according camera orientations for the initialization of the state vector of the Kalman filter approach.
```

We observe, that the negative direction has to be used with respect to the view direction to the negative Z-component. Applying the polar transformation

$$\theta = \operatorname{atan2}\left(d_Y, d_X\right) \tag{4.24}$$

$$\zeta = \operatorname{atan2}\left(\sqrt{d_X^2 + d_Y^2}, d_Z\right) \tag{4.25}$$

to the direction vector and choosing a reasonable distance we get the inverse distance parameterization as ${}^i\boldsymbol{X} = \boldsymbol{f}(\boldsymbol{T}_t, \mathbf{q}_t, \boldsymbol{z}, \rho)$. The state vector then has to be extended by ${}^i\boldsymbol{X}$. The extended covariance matrix ${}^{x}C_{pp}$ has to be computed using error propagation with

$$ {}^{x}C_{pp} = {}^{x}J \begin{bmatrix} C_{pp} & 0 & 0 \\ 0 & C_{zz} & 0 \\ 0 & 0 & \sigma_{\rho}^{2} \end{bmatrix} {}^{x}J^{\mathsf{T}} \tag{4.26} $$

$$ {}^{x}J = \begin{bmatrix} I_7 & 0 & 0 \\ 0 & I & 0 \\ \left[\frac{\partial\, {}^{i}\boldsymbol{X}}{\partial \boldsymbol{T}_t} \Big| \frac{\partial\, {}^{i}\boldsymbol{X}}{\partial \mathbf{q}_t}\right] & 0 & \left[\frac{\partial\, {}^{i}\boldsymbol{X}}{\partial \boldsymbol{z}} \Big| \frac{\partial\, {}^{i}\boldsymbol{X}}{\partial \rho}\right] \end{bmatrix}. \tag{4.27} $$

The analytical derivation in equation (4.27) is complex and depends on a set of trigonometric functions. We get these derivatives using numerical differentiation. As we mentioned before, the inverse distance representation enlarges the state vector significantly. Therefore Gaussian approximation in the Euclidean space can be obtained, if the object point will be observed several times from different directions. In this case it is sufficient to change the representation of an object point from inverse representation to a Euclidean one.

Using the roundness measurement of equation (4.22) we can determine whether a Euclidean representation is sufficient for the uncertainty representation.

The transformation of an arbitrary point ${}^{i}\boldsymbol{X}_n$ inside the state vector can be computed using equation (4.13). The covariance again can be obtained using error propagation by

$$ {}^{e}C_{pp} = {}^{e}J C_{pp}\, {}^{e}J^{\mathsf{T}} \tag{4.28} $$

$$ {}^{e}J = \begin{bmatrix} I & 0 & 0 \\ 0 & \frac{\partial \boldsymbol{X}_n}{\partial\, {}^{i}\boldsymbol{X}_n} & 0 \\ 0 & 0 & I \end{bmatrix} \tag{4.29} $$

with

$$ \frac{\partial \boldsymbol{X}_n}{\partial\, {}^{i}\boldsymbol{X}_n} = \left[\begin{array}{c|ccc} & -\frac{1}{\rho}\sin\zeta\sin\theta & \frac{1}{\rho}\cos\zeta\cos\theta & -\frac{1}{\rho^2}\sin\zeta\cos\theta \\ I_3 & \frac{1}{\rho}\sin\zeta\cos\theta & \frac{1}{\rho}\cos\zeta\sin\theta & -\frac{1}{\rho^2}\sin\zeta\sin\theta \\ & 0 & -\frac{1}{\rho}\sin\zeta & -\frac{1}{\rho^2}\cos\zeta \end{array}\right]. \tag{4.30} $$

In our experiments in chapter 5 we will analyze the influence of the parameter reduction using different roundness measurement thresholds on the estimated camera orientation in a photogrammetric strip.

4.3.4.4 Stable feature initialization procedure

As already outlined in figure 4.7, due to the small parallaxes, the distribution of the intersection point cannot be approximated with a Gaussian well enough. As we will show in the experimental section this will lead to a bias in the recursive estimation. The problem can be circumvented by the inverse distance representation, but requires more parameters to represent object points and furthermore increases the update computation complexity as well as a complicated point management.

In Steffen & Förstner (2008) we introduce a new algorithm, which is able to deal with a single camera state and uses a Euclidean object point representation. This approach is able to reduce the bias problem significantly.

We suggest postponing the initialization until the geometry guarantees the 3D-point is stable. This requires to store for each 3d-point coordinate $\boldsymbol{X}_n$, which is observed in frame t for the first time, the orientation parameters $\boldsymbol{T}_t$, $\mathbf{q}_t$ and the measured image coordinate $\boldsymbol{x}_t$. As the initialization needs not to be perfect in terms of accuracy, we do not need to introduce this data into the Kalman filter at later stages $> t$.

Each new point is tracked leading to a sequence of image points $\{\boldsymbol{x}_t, t = t_k, t_k + 1, ...\}$ in the following frames. At each time t we select that pair of frames (t_o, t) with $t_o \in \{t_k, t_k{+}1, ...\}$ for determining the 3D-point where the length of the basis $|\boldsymbol{T}_{t_o} - \boldsymbol{T}_t|$ is largest, as we expect the determination of the 3D-point to be most stable. If the covariance matrix of the 3d-point is round enough (c. f. section 4.3.4.2), we initialize the 3D-point and include it into the Kalman filter. The roundness can be computed according to the proposed algorithm in section 4.3.4.2.

The covariance matrix can be determined assuming the camera orientation parameters to be error free. In our implementation the triangulation and the error propagation is performed using the unscented transformation Julier & Uhlmann (1997), c. f. section 3.4.3.

Therefore the precision is overestimated and, what is even more severe, the correlations between the new point and the state of the system are zero. Thus a blind inclusion of the point together with its covariance matrix, derived this way, would make this new point act like a control point. For this reason we need to couple the uncertainty of the parameters of the state vector and the uncertainty of the new point. The idea is to determine the new point

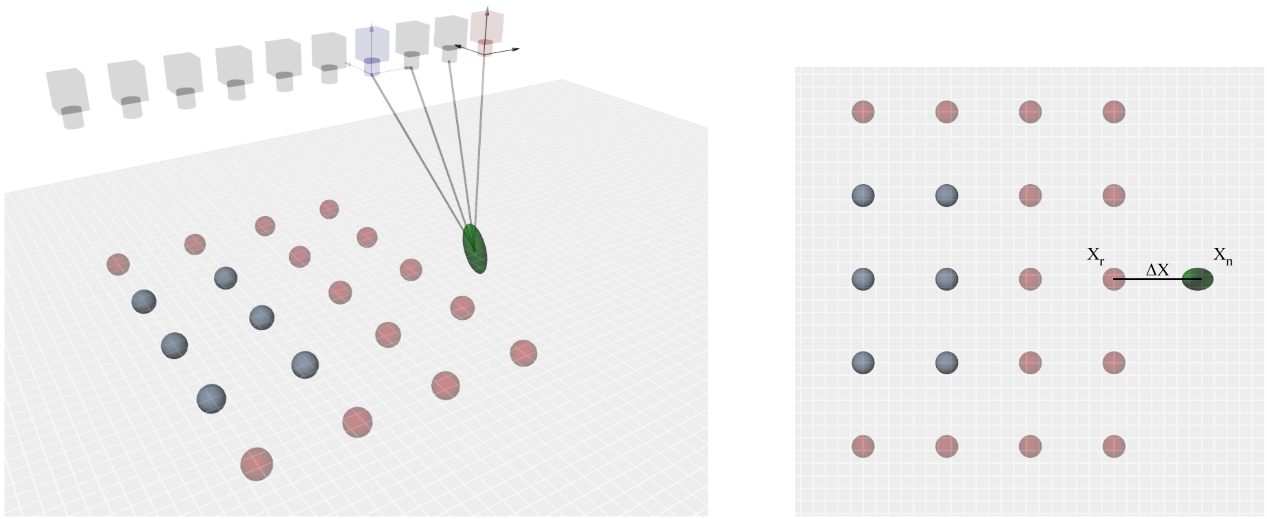


Figure 4.8: **Left:** 3d view of the intersection of a postponing initialization **Right:** Top view.

$\boldsymbol{X}_n$ via a point $\boldsymbol{X}_r$ which is close to the new point via:

$$\boldsymbol{X}_n = \boldsymbol{X}_r + \Delta\boldsymbol{X}_n \tag{4.31}$$

and perform error propagation assuming the given covariance matrix of $\boldsymbol{X}_r$ and its correlation with all parameters of the state vector. However, the coordinate difference $\Delta\boldsymbol{X}_n = \boldsymbol{X}_n - \boldsymbol{X}_r$ has covariance matrix which in the worst case is

$$C_{\Delta X_n \Delta X_n} = 2C_{X_n X_n}$$

and *is assumed to be independent* of $\boldsymbol{X}_r$. This is a realistic assumption, as the reference point would probably have a similar accuracy if it were triangulated from the same two images. An explanation of this assumption can be found in the Appendix in A.3. Thus we obtain the extension of the state vector in the following manner

$$\boldsymbol{p}^e = \begin{bmatrix} \boldsymbol{p}^* \\ \boldsymbol{X}_r \\ \boldsymbol{X}_n \end{bmatrix} = \underbrace{\begin{bmatrix} I & & \\ & I_3 & \\ & I_3 & I_3 \end{bmatrix}}_{J} \underbrace{\begin{bmatrix} \boldsymbol{p}^* \\ \boldsymbol{X}_r \\ \Delta\boldsymbol{X}_n \end{bmatrix}}_{\boldsymbol{p}^s} \tag{4.32}$$

where $\boldsymbol{p}^*$ is the state vector without point $\boldsymbol{X}_r$. The generating vector has the covariance matrix

$$C_{p^s p^s} = \begin{bmatrix} C_{pp^*} & C_{p^*r} & 0 \\ C_{rp^*} & C_{rr} & 0 \\ 0 & 0 & C_{\Delta X_n \Delta X_n} \end{bmatrix}. \tag{4.33}$$

The new covariance matrix of the state vector including the new landmark can now be calculated as follows

$$C_{p^e p^e} = J C_{p^s p^s} J^\mathsf{T} = \begin{bmatrix} C_{p^* p^*} & C_{p^* r} & C_{p^* r} \\ C_{r p^*} & C_{rr} & C_{rr} \\ C_{r p^*} & C_{rr} & C_{rr} + C_{\Delta X_n \Delta X_n} \end{bmatrix} . \tag{4.34}$$

This covariance matrix is guaranteed to be positive semi-definite and reflects the new situation realistically. Obviously, the linear transformation is sparse and except for the addition of the covariance matrices of the reference and the point difference only copies need to be realized, which is extremely efficient. The algorithm for initialization of new points thus can be summarized as follows:

Algorithm 4.2 *Stable Initialization Procedure* algorithm

```
Detecting that a new point $\boldsymbol{x}_n$ is observed at time $t_k$.
repeat
    Get the next observation at time $t = t + 1$
    Determine the frame at time $t_o$ with the longest base length
    $d = \text{argmax}_{t_o \in \{t_k..t\}} (|\boldsymbol{T}_t - \boldsymbol{T}_{t_o}|)$.
    Determine $\boldsymbol{X}_n$ and its covariance matrix using frame pair $(t_o, t)$ and its roundness $L_n$
until $L_n > \Theta_L$
Search for the point $\boldsymbol{X}_r$ closest to $\boldsymbol{X}_n$.
Include the new point into the state vector using (4.32) and (4.34).
```

The proposed algorithm is able to initialize new points in a stable manner considering the correlation structure.

4.4 Georeferencing

As we have shown before the initialization of the Kalman filter can be performed using control points or a scale free bundle adjustment procedure. The control points can be given in a global coordinate system as well as in a local reference system. In the following we will assume that the initialization was performed by control points in a local reference system or by a scale free bundle adjustment. In case we are able to detect at least three non-collinear control points in a global reference system, we can transform the state vectors and its covariance using a general *Helmert*-transformation. Using only three identical points a direct solution can be computed. If more points are available we will obtain the best results using a least square solution. In the following we will first derive the transformation using three points only. In a second step we use the estimated parameters to transform the state and its covariance matrix and additionally transform the gauge of the system with respect to the control points.

First, we assume three control points $\boldsymbol{X}_i^c$ corresponding to three object points $\boldsymbol{X}_i, \quad i =$ 1..3 inside the state vector are given. For all points the incidence equation

$$\mathbf{X}_i^c = \mathsf{M}_\lambda \mathsf{M}_R \mathsf{M}_{\Delta \boldsymbol{T}} \mathbf{X}_i \tag{4.35}$$

with M_T as a translation, M_R as a rotation and M_S as a scale matrix must hold. The transformation parameter λ, R and $\Delta \boldsymbol{T}$ can easily be derived using the direct solution for the Helmert-transformation by Horn *et al.* (1988).

Now we are able to transform the state coordinate system to the world coordinate system using the estimated parameters of the Helmert-transformation, also called the *K-transformation* according to Molenaar (1981). The transformation can be obtained by a non-linear function as

$$\boldsymbol{p}_h = \begin{bmatrix} \begin{bmatrix} \lambda R(\mathbf{q}_h)\boldsymbol{T} + \Delta \boldsymbol{T} \\ \mathbf{q}_h \mathbf{q} \end{bmatrix}_k \\ \left[\lambda R(\mathbf{q}_h) + \Delta \boldsymbol{T}\right]_i \end{bmatrix} \tag{4.36}$$

for all camera orientations k and all points i inside the state vector. The transformation of the covariance matrix can be performed using error propagation with the Jacobian

$$J = \begin{bmatrix} \lambda R(\mathbf{q}_h) & 0 & 0 \\ 0 & \Upsilon(\mathbf{q}_h) & 0 \\ 0 & 0 & \lambda R(\mathbf{q}_h) \end{bmatrix}. \tag{4.37}$$

Actually, the gauge is still defined by the initialization method of the state vector. If desired, the gauge can be transformed to the three reference points using the *S-Transformation*, c. f. Baarda (1973) and Dickscheid *et al.* (2008). The transformation can be applied to the covariance matrix in the following way: First create an stacked matrix A consistent of A_k for all camera parameter and A_i for all object points

$$A = \begin{bmatrix} A_{1..K} \\ A_{1..I} \end{bmatrix} \tag{4.38}$$

with

$$A_k = \begin{bmatrix} I_3 & -S(\boldsymbol{T}) & \boldsymbol{T} \\ 0 & \begin{bmatrix} -\frac{1}{2}\boldsymbol{q}^\mathsf{T} \\ \frac{1}{2}(qI_3 - S(\boldsymbol{q})) \end{bmatrix} & 0 \end{bmatrix}_k \tag{4.39}$$

$$A_i = \begin{bmatrix} I_3 & 0 & 0 \end{bmatrix}_i . \tag{4.40}$$

The Jacobian can then be computed by

$$J_S = I - A(A^\mathsf{T} W A)^{-1} A^\mathsf{T} W \tag{4.41}$$

with the gauge points selection matrix, where the diagonal elements according to the reference point coordinates are one. Applying the Jacobian to the covariance matrix, the gauge is transformed to the specified reference points.

The proposed geo-referencing algorithm is applicable using a sliding window state representation. Furthermore, it is assumed that the transformation parameters are non-stochastic and the control point uncertainty is zero.

4.5 On particle based Kalman filtering for simultaneous localization and mapping

In section 3.4.3 we introduced the unscented transformation and its enhancement to a particle based Kalman filter. In the literature the reader will find examples for successful applications of the *sigma point Kalman filter*, e. g. Sünderhauf *et al.* (2007), van der Merwe (2004). Nevertheless, we will show that these approaches will fail for the task of simultaneous localization and mapping in case of larger uncertainties of parameters in the state vector.

First we will illustrate these at the error propagation using the inverse distance representation and in a second example on the measurement update using a Euclidean point representation. The distance d of a point is represented in the inverse distance representation $\rho = \frac{1}{d}$. We assume the uncertainty representation is a Gaussian with mean μ_ρ and their standard deviation σ_ρ. A small value μ is equal to a large distance and it is necessary that

$$d = \lim_{+|\rho| \to 0} \frac{1}{\rho} = \infty. \tag{4.42}$$

On the other hand for negative values of μ_ρ it is necessary that

$$d = \lim_{-|\rho| \to 0} \frac{1}{\rho} = -\infty. \tag{4.43}$$

Obviously, there is a singularity at $\rho = 0$. Crossing this singularity, the distance turns from positive to negative infinity. As shown in figure 4.9 in case of large uncertainty in relation to the inverse distance the sigma point needed to compute the error propagation can cross this border, which results in a wrong error propagation. To illustrate the singularities we perform the transformation of the inverse distance to the Euclidean distance using the unscented transformation for the mean and standard deviation. The uncertainty in the inverse distance space is chosen as $\sigma_\rho = 1$. As shown in figure 4.10 the singularities appear between $-\sqrt{n+\kappa} < \mu_\rho < +\sqrt{n+\kappa}$, in this case $n = 1$, so $\kappa = 2$, so that the sigma point crossing the singularity at $\pm\sqrt{n+\kappa} = \pm 1.73$. As we have shown in the example before, singularities in the transformation result in some cases in an erroneous mean and variance reconstruction using the unscented transformation. Projecting an object point $\boldsymbol{X}$ to the image plane via an optical center $\boldsymbol{O}$ and computing mean and standard deviation using sigma points can fail

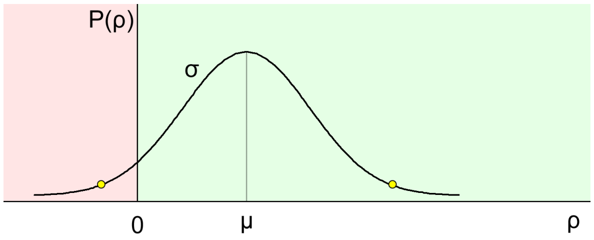


Figure 4.9: Sigma point crosses positive to negative infinity singularity

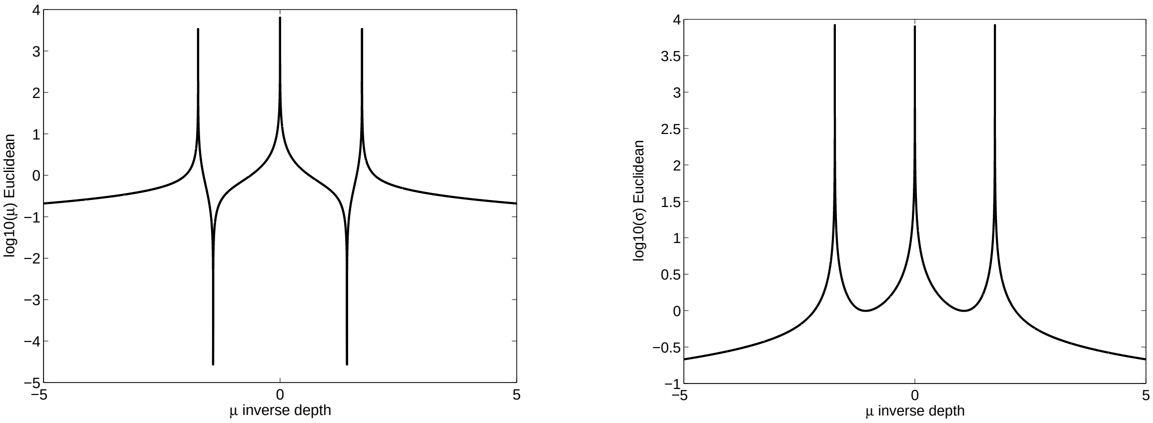


Figure 4.10: **Left:** Mean propagation using unscented transformation from inverse distance to Euclidean space **Right:** Variance propagation using unscented transformation from inverse distance to Euclidean space **General:** Gaussian distribution $\sigma = 1$ inside inverse space.

as shown in figure 4.11. This comes from a singularity induced at the plane parallel to the image plane thru the projection center. Sigma points can lie behind this plane in case of large object point uncertainties. Their projection does not represent the uncertainty of the projected image point $\boldsymbol{x}$ anymore.

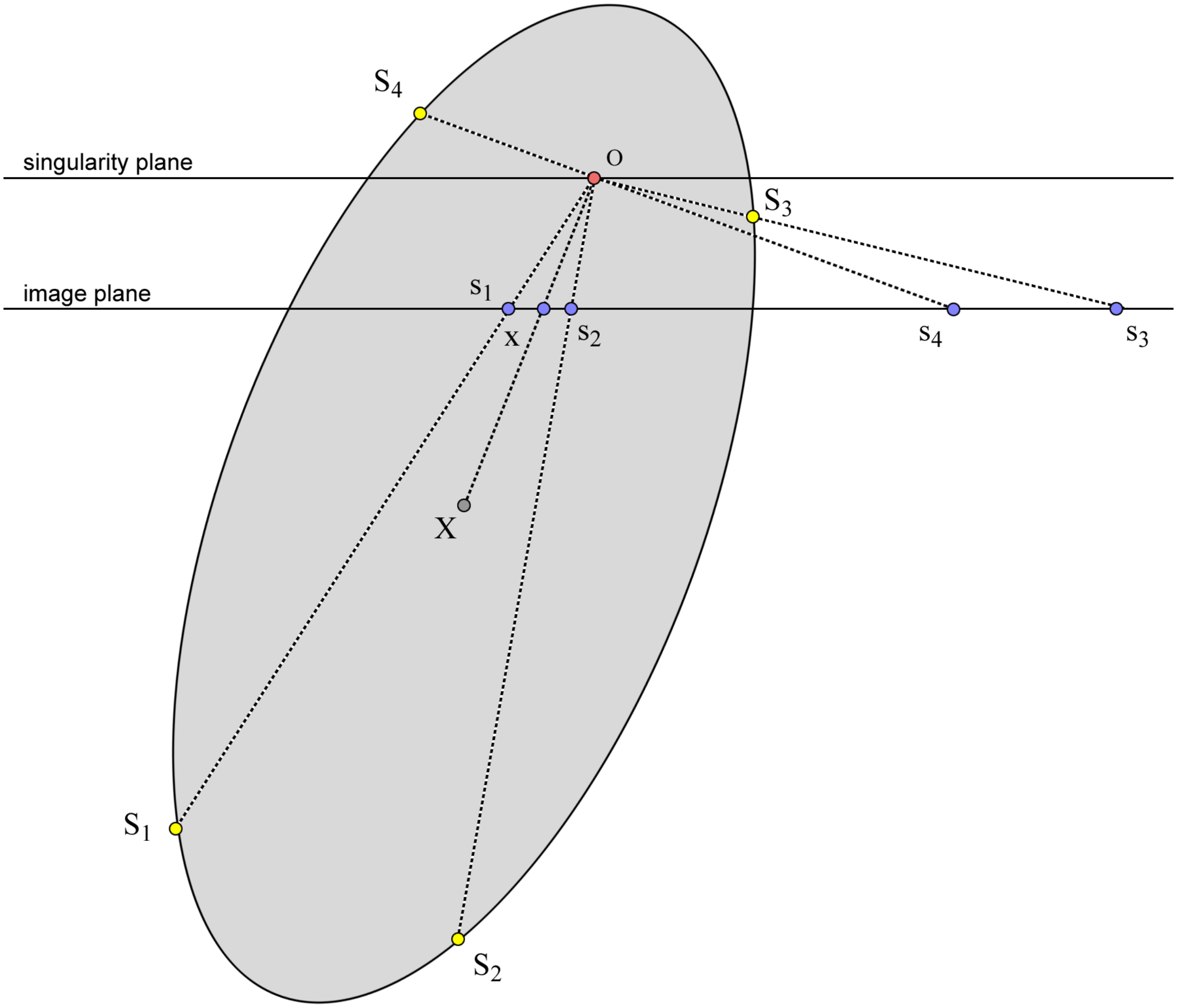


Figure 4.11: Example of sigma point projection in case of large object point uncertainty.

Discussion: As shown in two examples the unscented transformation as well as its extension to an particle based Kalman filter can fail in case of singularities of the nonlinear transformation. Someone could argue, that the sigma points could be scaled to smaller distances adjusting k. In some application this can be beneficial. But, in case of pure visual simultaneous localization and mapping a large set of parameters with different levels of uncertainties occur. In this case the Gaussian will not be represented well anymore. The loss of this advantage of the *sigma point Kalman filter* makes this approach uninteresting.

Nevertheless, using *GPS* and *INS* systems for absolute and relative observations with nearly constant uncertainty at every point in time, an implementation can prevent singularities in the prediction and measurement update model. The benefit was shown for example by Sünderhauf *et al.* (2007).

In this chapter we introduced the need of a system for visual simultaneous localization and mapping. To make a Kalman filter core module for estimating metric map and localization parameters operable, we preferred feature tracking and matching techniques. We introduced different camera orientation and point based map representations and derived their dynamic and measurement updates. We proposed a novel dynamic update, which will use only linear transformations for the state vector as well as for the covariance matrix. We pointed out, how the Kalman filter for a simultaneous localization and mapping task can be initialized and how new parameters in a variable state can be introduced. According to the initialization a geo-referencing approach was proposed. Besides, we commented the disadvantage of the unscented transformation and *sigma point Kalman* filter for this task.

Chapter 5

Evaluation of the proposed methods

In this chapter the applicability of Kalman filter based localization and mapping will be demonstrated. In the classical aerial-triangulation images are be taken with large baselines according to an image overlapping of approximately 60%. Therefore, an object point typically is visible in at least three images. Special high resolution cameras with a stable interior orientation will be used. These cameras are mounted on a motion compensation platform to prevent motion blur coming from longitudinal and angular velocity. In contrast we will use an image sequence taken from an unmanned aerial vehicle. The camera is a low resolution consumer video camera. Due to the limited speed of flight in our case we deal with an overlapping ratio of consecutive images of larger than 95%. Hence object points are visible multiple times with small disparities in consecutive images, also called *frames*.

First we will derive a general law to assess the theoretical accuracy of a photogrammetric strip. To neglect systematic errors arising from camera calibration errors, gross errors in the point based feature tracking and influences of the approximate values of the parameters the law will be determined using synthetic observation of a simulated environment. In a second step the influence of the representation and initialization of new object points in a Kalman filter will be analyzed. In an evaluation we will compare the different methods regarding accuracy and consistency. To proof the applicability of our point based feature tracking we will use two different kinds of rendered scenes. One scene consists of a planar surface mapped with real image data of a high altitude airborne image data provided by the company Vexcel. The second scene consists of a model of the *Basilika St. Peter* in Rome. The texture was generated using a fractal texture algorithm. For both scenes the interior and exterior orientation was extracted from the rendering software. Finally, we will run the algorithms on a full real data set.

5.1 On the theoretical accuracy of photogrammetric strips from image sequences

In the classical aero-photogrammetry a general law for the evolution of the triangulation accuracy of a strip flight has been found by Finsterwalder & Hofmann (1964). The proposed law specifies the accuracy of the camera location and angular orientation depending on the image number and the base length between consecutive frames up to a scale factor. The scale factor depends on the configuration of the photogrammetric strip, the camera parameters and the measurement accuracy. For every configuration the scale factor have to be determined individually. In this section we extend this law, so that we incorporate the flight configuration parameters as well as the camera parameters. This extended general law has the following challenge:

a) The goal of a general law is to calculate the achievable accuracy under specific parameter settings using a simple approximation function.

b) The function has to be as precise as possible for a specific interval for the given parameter.

First, we have to distinguished between a systematic errors and a stochastic errors. The error propagation laws by Finsterwalder & Hofmann (1964) summarized in table 5.1 are useful to assess the achievable accuracy of classical photogrammetric aerial-triangulation for very long strips. This law has been derived from the error propagation and has been simplified for very long strips. We can interpret the variance transition factors in that way, that smaller factor yield in an accuracy improvement of the estimated camera orientations. These laws are true under the condition that the gauge definition depends on the first frame and the covariance matrix has a significant band structure for the strip. However, there are only a small number of different aerial camera systems available and the configuration of flight strips is typically fixed to 60% longitudinal overlapping. For this reason only a small set of variance transition factors has to be analyzed for different configurations.

In the field of an unmanned aerial vehicle arbitrary cameras and flight configurations are possible. In this case the variance transition factors are functions of a large set of parameters,

Error component	Systematic	Stochastic
Lateral	$\Delta Y_n \approx \Delta\kappa n^2$	$\sigma^2_{Y_n} \approx \frac{B^2}{3} m^2_\kappa n^3 + \sigma^2_{Y_0}$
Longitudinal	$\Delta X_n \approx \Delta s n^2$	$\sigma^2_{X_n} \approx \frac{B^2}{3} m^2_s n^3 + \sigma^2_{X_0}$
Height	$\Delta Z_n \approx \Delta\phi n^2$	$\sigma^2_{Z_n} \approx \frac{B^2}{3} m^2_\phi n^3 + \sigma^2_{Z_0}$
Angular	$\Delta\omega_n \approx \Delta\omega n$ $\Delta\phi_n \approx \Delta\phi n$ $\Delta\kappa_n \approx \Delta\kappa n$	$\sigma^2_{\omega_n} \approx m^2_\omega n + \sigma^2_{\omega_0}$ $\sigma^2_{\phi_n} \approx m^2_\phi n + \sigma^2_{\phi_0}$ $\sigma^2_{\kappa_n} \approx m^2_\kappa n + \sigma^2_{\kappa_0}$

Table 5.1: **General:** The table shows the error transition law for the camera orientation parameters of a very long photogrammetric strip in case of systematic and stochastic errors. Parameter B is the base length between consecutive frames, n is the frame number of the strip. **Systematic:** The systematic transition error factors $\Delta s, \Delta\omega, \Delta\phi$ and $\Delta\kappa$ are constant for arbitrary chosen configurations and depend on systematic error. The highest influence to these factors are non-linear calibration errors of the camera. **Stochastic:** The variance transition factors $m^2_s, m^2_\omega, m^2_\phi$ and m^2_κ can be interpreted as a variance error between consecutive frames. The additive variances $\sigma^2_{X_0}, \sigma^2_{Y_0}, \sigma^2_{Z_0}, \sigma^2_{\omega_0}, \sigma^2_{\phi_0}$ and $\sigma^2_{\kappa_0}$ depend on the gauge definition of the first frame of the strip. Assuming an error-free camera orientation for the first frame the additives are zero. Using control points as a gauge definition the additives depend on the accuracy of the spatial resection.

c. f. table 5.2.

camera parameters		path parameter	
field of view	α	flight altitude	H_g
video rate	r	flight speed	v
camera resolution	w	features per image	N_X
tracking accuracy	σ_{xy}		

Table 5.2: Configuration parameters

In this work we will rearrange the law for the stochastical lateral, longitudinal and height error to make them independent from the angular errors. Furthermore, the base length B will be represented inside the variance transition factor. Following the definition of the stochastic error in table 5.3 we get six variance transition factors depending on a chosen flight configuration.

Error component	Stochastic
Lateral	$\sigma^2_{Y_n} \approx m^2_Y n^3 + \sigma^2_{Y_0}$
Longitudinal	$\sigma^2_{X_n} \approx m^2_X n^3 + \sigma^2_{X_0}$
Height	$\sigma^2_{Z_n} \approx m^2_Z n^3 + \sigma^2_{Z_0}$
Angular	$\sigma^2_{\omega_n} \approx m^2_\omega n + \sigma^2_{\omega 0}$ $\sigma^2_{\phi_n} \approx m^2_\phi n + \sigma^2_{\phi 0}$ $\sigma^2_{\kappa_n} \approx m^2_\kappa n + \sigma^2_{\kappa 0}$

Table 5.3: Simplified stochastical transition error model for photogrammetic strips.

These six variance transition factors are a function defined by

$$m^2_* = f_*(H_g, v, r, \sigma_{xy}, w, N_X, \alpha). \tag{5.1}$$

Some of these parameters can be combined. The *flight speed* v and *video rate* r determine the base length B between consecutive frames. The *tracking accuracy* σ_{xy} in pixel coordinates, the *camera resolution* w and the field of view α determine the normalized direction measurement accuracy. First, a normalized direction measurement accuracy can be derived using the principal distance

$$c = \frac{w}{2\tan(\frac{1}{2}\alpha_w)} \tag{5.2}$$

with w as the width of the camera and α_w as the field of view according to the width. Assuming square image pixels the normalized direction measurement accuracy σ_d can be computed from the image measurement accuracy σ_{xy} by

$$\sigma^2_d = \frac{\sigma^2_{xy}}{c^2}. \tag{5.3}$$

Using this normalized direction measurement accuracy the camera resolution and image measurement accuracy can be neglected from the general law. Because the field of view also determines geometric stability, we do not neglect this parameter.

Second, we will combine the flight speed and video rate yielding in the base length. Moreover, from the principle that the triangulation is independent of a scale factor we will neglect the flight altitude by using a normalized base length

$$b = \frac{v}{rH_g} \tag{5.4}$$

Again, the video rate r, flight speed v and flight altitude H_g can be neglected from the influence function by using the normalized base length b.

As a consequence the achievable accuracy of a photogrammetric strip depends on four parameters. In the following we will analyze in some experiments, how does the accuracy depends on these parameters.

5.1.1 Experimental setup

As we shown above the stochastical accuracy of a photogrammetric strip will be influenced by four parameters. To analyze their influence we here define four experiments. The parameter themeself are physically bounded. For example, a camera view angle larger than 180° is not reasonable. Furthermore the smallest base length can be zero, the largest base length is restricted by the overlapping, and so indirect of the maximal possible video rate and the maximum flight speed.

In table 5.4 we will define a standard configuration and different experiment configurations of our simulated environment.

Exp 1: In this experiment we will change the field of view angle between 40 and 120 degree. The lowest bound is a typical small angle camera objective, the upper bound is a wide angle objective.

Exp 2: This experiment varies the base length, depending on the video rate and the flight speed according to equation (5.4). The maximum base length is restricted by an overlapping ratio of the images of approximately 60% to guarantee, so that a subset of object points are visible in at least three images. This is neccessary to achieve the transition of the scale factor to consecutive frames in a bundle adjustment solution.

Exp 3: The fully automatic detection and tracking accuracy can be broken down to $\sigma_{xy} \approx 0.1[pel]$ using a sub-pixel estimation. Typically point based features on the original image resolution are unstable in case of image noise. Therefore it is suitable to track features at a reduced resolution which can by quantified by a change of the tracking accuracy.

Exp 4: The mean number of used object points in an image is proportional to the size of the unknown parameter. Here a trade of between computational complexity and achievable accuracy has to be found. Furthermore, the content of the observed scene limits the number of identifiable and trackable object points. It has been shown in our experiments

camera parameter	standard	Exp 1	Exp 2	Exp 3	Exp 4
camera $w \times h$ resolution	800×600				
field of view α	90°	40,50,60 70,80,90, 100,110,120			
principal point	center				
video rate r	$25Hz$				
tracking σ_{xy} accuracy	0.25 [pel]			0.1, 0.25, 0.5, 1, 2, 4, 8, 16	
features N_X per image	25				10,15,20,25, 30,35,40
path parameter					
strip length s	200 [m]				
flight altitude H_g	30 [m]				
flight speed v	5 [m/s]				
base line B	0.20 [m]		0.1, 0.2, 0.5, 1, 2, 4, 8		
image overlapping	99.7 [%]		99.8 → 82.2		

Table 5.4: Summery of the experimental setup with a standard configuration and its variations in four experiments.

that a marginal accuracy advantage can be achieved using more than 40 features per image.

In the following we will compute the theoretical accuracy of the camera orientations for a photogrammetric strip by any variation of the four defined experiments. The theoretical accuracy will be estimated by a bundle adjustment under the condition, that the observations are error free and the approximate values are equal to the true values. The variances of the estimated camera parameter of the resulting theoretical covariance are used as data points for a curve fitting using the simplified transition model in table 5.3. The variance transition factors are be computed by a least square solution.

5.1.2 The influence of the field of view

In the first experiment (*Exp 1*, c.f. table 5.4) the variance transition factors for different view angles of the camera are estimated. In figure 5.1 the square root of the transition factors for the six camera orientation components are visualized.

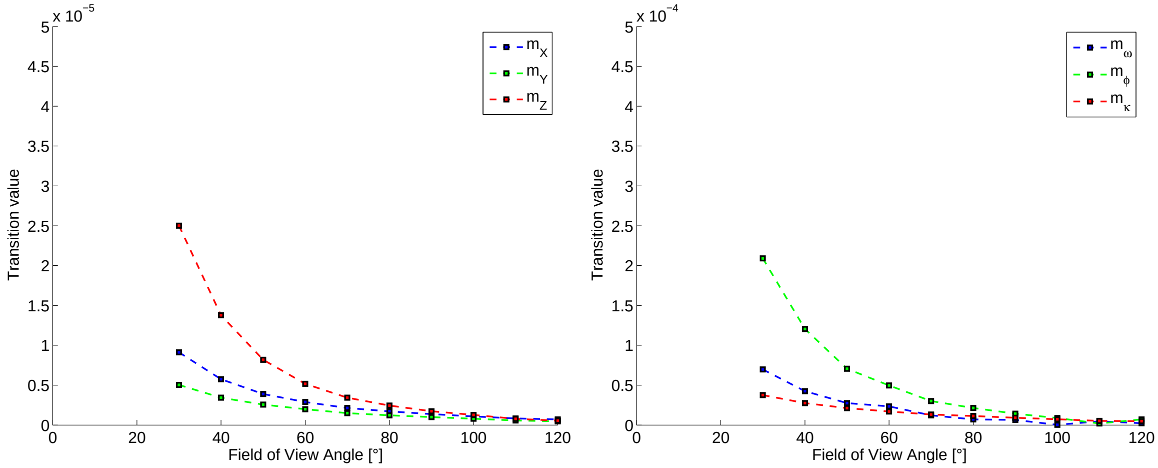


Figure 5.1: Influence of the principal distance (field of view) onto the accuracy transition factors. The cameras field of view varies between 30° and 120°. **Left:** Square root of the transition factors for the camera position, **Right:** Square root of the transition factors for the angular camera orientation.

We can observe that decreasing the field of view of the camera the uncertainty of the Z-component and latitude angular orientation of the camera is growing rapidly. A small field of view reduces the maximum intersection angles of a visible object point and therefore its accuracy in the Z-component. Due to the correlation, the accuracy of the camera location in the Z-component will increase by using a camera with a larger field of view. This behavior can also be observed at the angular orientation ϕ, which will be effected by a known high correlation to the Z-component. Because we do not change the resolution of the camera as well as the tracking accuracy, the direction measurement accuracy, which depends on the field of view, will decrease by increasing the field of view. Its effect to the variance transition factor will be superimposed by choosing a camera with a larger field of view.

5.1.3 The influence of the base length

In the second experiment (*Exp 2*, c.f. table 5.4) we vary the base length according to a change of the UAVs flight speed or the frame rate of the camera.

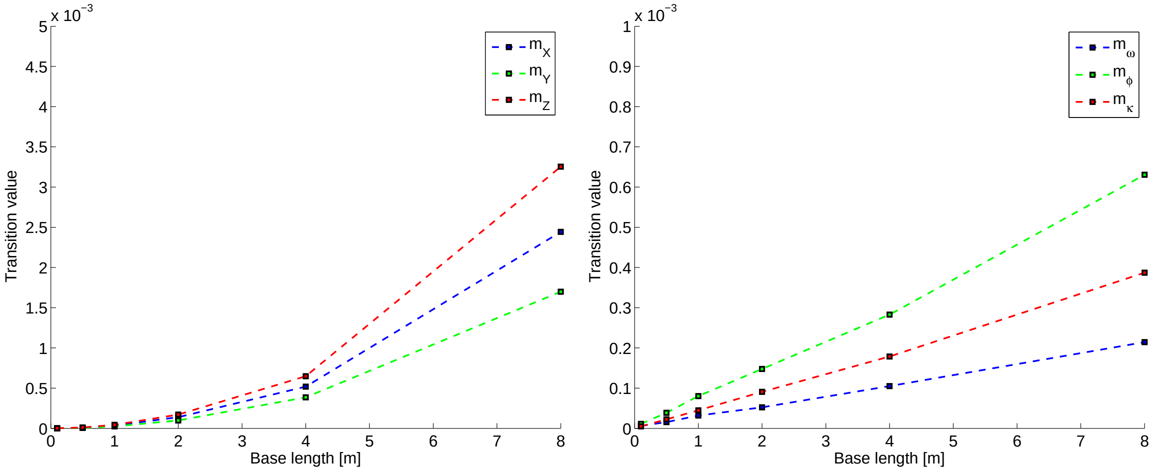


Figure 5.2: Influence of the chosen base length onto the accuracy transition factors. The base length varies between 0.1 [m] and 8 [m]. **Left:** Square root of the transition factors for the camera position, **Right:** Square root of the transition factors for the angular camera orientation.

In figure 5.2 the resulting variance transition factors are shown. We can observe, that the uncertainty of the camera location is growing for larger baselines in an exponential manner, the uncertainty of the angular orientation is growing linear. Decrease the base length, the number of observations to have a part of an individual object point will be increase.

The maximum intersection angle to determine an object point location does not change. From a theoretical point of view, the influence of the involved observations due to the accuracy of the camera orientation parameters depend on the geometric configuration, which will be reflected in the redundancy number of the observation.

5.1.4 The influence of measurement accuracy

In the third experiment (*Exp 3*, c. f. table 5.4) we will analyze the influence of the identification accuracy of features in the image onto the variance transition factors. The measurement of a feature can be reinterpreted as a direction measurement to an object point, which depends on the resolution of the image, the object identification accuracy and the field of view of the chosen camera. The resolution and identification accuracy depends linear by each other. For example, if we double the resolution of the camera the measurement accuracy has to be multiplied by a factor of two.

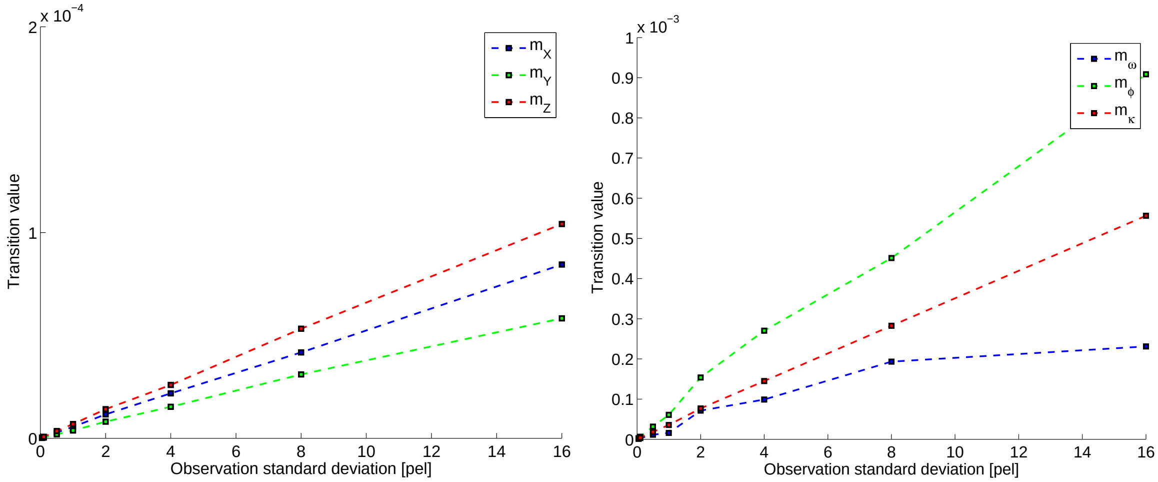


Figure 5.3: Influence of the measurement accuracy onto the accuracy transition factors. The feature measurement accuracy varies between 0.1 and 16 [pel]. **Left:** Square root of the transition factors for the camera position, **Right:** Square root of the transition factors for the angular camera orientation.

As expected, the influence of the measurement accuracy is linear to the square root of the transition factors shown in figure 5.3. We observe that the angular accuracy transition is unstable. We suppose, that this instability comes from the non-homogeneous arrangement and limited number of object points in the scene. Nonetheless, in average the influence effects in a linear manner.

5.1.5 The influence of map density

In the fourth experiment (*Exp 4*, c. f. table 5.4), we will analyze the influence of the map density e. g. the number of used object points visible in each image. We require that the observed object points are uniformly distributed in the bounded image plane and are therefore uniformly distributed on the observed planar surface. But, the uniform distribution cannot be guaranteed, because the number of object points visible in each image is small.

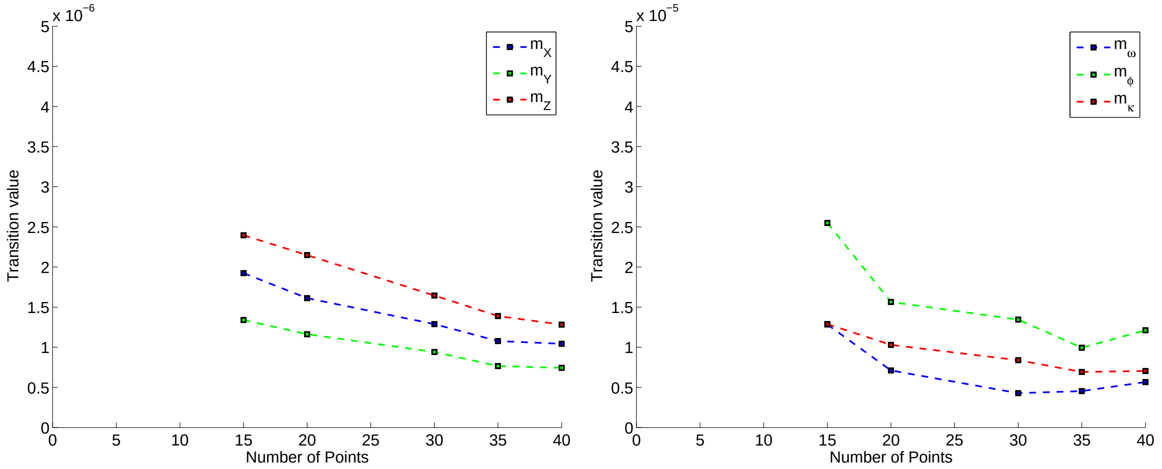


Figure 5.4: Influence of the map density onto the accuracy transition factors. The number of used feature in an image varies between 10 and 40. **Left:** Square root of the transition factors for the camera position, **Right:** Square root of the transition factors for the angular camera orientation.

In figure 5.4 the estimated square root variance transition factors with respect to the average number of visible object points are presented. From the nature of repeated normal distributed measurements we know, that the uncertainty of the average measurement will decrease by a factor of $\sqrt{N_z}$, with N_z as the number of repeated observations. As the distribution of the object points at the surface is supposed to be uniform, we can interpret the map density as a repeated measurement.

5.1.6 A general law to assess the theoretical accuracy of a photogrammetric strip

In the following we will find a general law to specify the achievable accuracy for a strip flight. At the beginning, we assume that the influence of the parameters is multiplicative with some unknown exponentials $i_{\{b,\alpha,N,d\}}$ and unknown factors k. Furthermore, the transition factors for the angular orientation components must be independent of the flight altitude H_g.

$$m^2_{\{X,Y,Z\}} = k_{\{X,Y,Z\}} \cdot b^{i_b} \cdot f(\alpha)^{i_\alpha} \cdot N_X^{i_N} \cdot \sigma_d^{i_d} \cdot H_g^2 \tag{5.5}$$

$$m^2_{\{\omega,\phi,\kappa\}} = k_{\{\omega,\phi,\kappa\}} \cdot b^{i_b} \cdot f(\alpha)^{i_\alpha} \cdot N_X^{i_N} \cdot \sigma_d^{i_d} \tag{5.6}$$

As we have shown in figures 5.3, the accuracy transition factor grows linearly with the expected tracking accuracy. Therefore the exponent should be two. As we mentioned before, doubling the number of observations will divide the uncertainty by the factor $\sqrt{2}$. Therefore the exponent of N_X can be set to minus one. The length of the base between consecutive images directly influences the number of observations of an object point. Furthermore, the maximum intersection angle will be divided in more peaces of intersections. The influence of every individual observation caused by a smaller baseline is not equal. In figure 5.2 we observe, that the accuracy transition factors for the angular orientation increase linearly with respect to the base length. Therefore we set the exponent to two. The influence of the base length to the location parameter seems to be square. Therefore we set the exponent to 4. We model the influence of the field of view angle α using a trigonometric function. The choice is motivated by the dependency of an angle. As we found out, a good approximation is achievable using different exponents for the six component.

$$f_X(\alpha) = \tan\left(\frac{\alpha}{2}\right)^{-5} \tag{5.7}$$

$$f_Y(\alpha) = \tan\left(\frac{\alpha}{2}\right)^{-4} \tag{5.8}$$

$$f_Z(\alpha) = \tan\left(\frac{\alpha}{2}\right)^{-6} \tag{5.9}$$

$$f_\omega(\alpha) = \tan\left(\frac{\alpha}{2}\right)^{-6} \tag{5.10}$$

$$f_\phi(\alpha) = \tan\left(\frac{\alpha}{2}\right)^{-6} \tag{5.11}$$

$$f_\kappa(\alpha) = \tan\left(\frac{\alpha}{2}\right)^{-4} \tag{5.12}$$

and we finally obtain the approximation functions

$$m^2_{\{X,Y,Z\}} = k_{\{X,Y,Z\}} \, \frac{b^4}{N_X} \, f_{\{X,Y,Z\}}(\alpha) \; \sigma_d^2 \; H_g^2 \tag{5.13}$$

$$m^2_{\{\omega,\phi,\kappa\}} = k_{\{\omega,\phi,\kappa\}} \, \frac{b^2}{N_X} \, f_{\{\omega,\phi,\kappa\}}(\alpha) \; \sigma_d^2 \tag{5.14}$$

Using the determined transition factors of the four experiments we estimate the constant factors $k_X, k_Y, k_Z, k_\omega, k_\phi$ and k_κ in a least square solution using equations (5.13) and (5.14)

Variance factor	value	σ
k_X	84	0.3
k_Y	40	0.3
k_Z	148	0.3
k_ω	23	6.8
k_ϕ	299	6.8
k_κ	112	6.8

Table 5.5: Estimated variance transition factors of the approximation function.

as the measurement model. This model holds for the condition, that a) the field of view for the camera is between 30° and 120°, b) the mean number of object points is between 10 and 40, c) a direction measurement accuracy between $\sigma_d = 0.00025$ and 0.04 (c. f. equation (5.3)) and d) a normalized base length between $b = 0.0001$ and 0.01 (c. f. equation (5.4)). The estimated variance transition factors of our model are presented in table 5.5. Because the particular factors depend on different functions relating to the field of view influence, the factors are not interpretable. As a degree of the reliability of our model we compute the maximum approximation error between our approximation function and the variance transition factors of the experiments. For the experiments with a variable field of view, direction measurement accuracy as well as the number of object points the approximation error is below 50%. For the case of a base line $B > 2m$ the maximum approximation error is 185%. A summery of the approximation error can be found in appendix A.4.

5.2 The influence of the initialization method of new object points in a Kalman filter based estimation

In the following we will analyze, how the Kalman filter based approaches performs on the same dataset using different initialization methods for new object points proposed in section 4.3. As we already mentioned, the initialization has a main influence on the consistency of the result due to the small disparities of feature observations in consecutive images, which is caused by a poorly Gaussian approximation of the object point determination by ray intersection. In all experiments, where the new object points will be introduced instantly, new object points are initialized at the first projection ray with a huge uncertainty. The distance will be assumed to be known using the ground truth distance obtainable as the Euclidean distance between the camera projection center and the object point location. The object point coordinate to be introduced can be computed using equation (3.63). As the dynamic motion model will be used to obtain approximate values for the camera orientation only, the error propagation will be computed with large system noise for the camera parameter transition. As a consequence the uncertainty of the predicted camera parameter is large and the correlations between the predicted camera parameters and previous camera parameters as well as the object points after the prediction step are equal to zero. Therefore, our dynamic model does not restrict the trajectory. We do so, because the camera trajectory in the bundle adjustment solution is also not restricted and therefore the results are comparable. The measurement update can be considered as a spatial resection to obtain the camera orientation. An update of the object point locations will be caused by the influence of the residuals between the measurements and the back-projected object points themselves.

We will run the Kalman filter using the initialization methods with

1. Euclidean representation
2. Euclidean representation with sliding window camera model
3. Inverse distance representation
4. Inverse distance representation with reparameterization to the Euclidean representation
5. Delayed initialization with sliding window camera model
6. Delayed initialization using *stable initialization procedure*

In order to evaluate the Kalman filter based approaches concerning their accuracy and consistency we will run all algorithms on a simple synthetic dataset, where ground truth data is available. The dataset was generated using the standard parameter defined in table 5.4 on page 102. The planar surface with object points were generated by sampling a uniform distribution in the xy-plane. The observations were obtained by a back-projection of the ground truth object point locations into the camera plane using the true camera orientation at time t. Finally, the observations were disturbed by Gaussian noise using an expected observation accuracy of $\sigma_{xy} = 0.25$ [pel]. The global coordinate system is fixed by the object points visible in the first image and assumed to be error free.

First, we will estimate the camera orientation parameter using a bundle adjustment as a high precision reference. To quantify the result we will use a consistency and a precision measure proposed by Dickscheid *et al.* (2008). The consistency measure c_c is defined by a weighted distance of two datasets using their covariances. In our case ground truth data for the camera parameter $\widetilde{\boldsymbol{p}}_n$ are available so that $\mathsf{C}_{\widetilde{p}_n\widetilde{p}_n} = \mathit{0}$ and therefore, the consistency measure reflects the consistency between the estimated camera parameter $\boldsymbol{p}_n$ and its covariances $\mathsf{C}_{p_np_n}$ for all camera orientations n. The consistency measure is defined by

$$c_c = \sqrt{\frac{\Omega}{6n-7}} \tag{5.15}$$

with

$$\Omega = (\boldsymbol{p}_n - \widetilde{\boldsymbol{p}}_n)(\mathsf{C}_{p_np_n} + \mathsf{C}_{\widetilde{p}_n\widetilde{p}_n})(\boldsymbol{p}_n - \widetilde{\boldsymbol{p}}_n)^{\mathsf{T}}. \tag{5.16}$$

We will call a result consistent, if its consistency measure is approximately $c_c \approx 1$.

The precision c_p is a measure to compare two covariances using a metric for covariance proposed by Förstner & Moonen (1999). The precision level is defined for a regular covariance matrix only and will reflect the relative precision of the covariance of a parameter vector to a covariance of superior precision. The precision measure can be obtained by

$$c_p = e^{\sqrt{\overline{\ln r^2}}} \tag{5.17}$$

with

$$\overline{\ln r^2} = \frac{\sum_{i=1}^{m} \ln^2 r_i}{m} \geq 1 \tag{5.18}$$

Here, r_i^2 are the generalized eigenvalues from $|C_{p_np_n} - r^2\ C_{\widetilde{p}_n\widetilde{p}_n}| = 0$. Due to the fact that we will use the precision value to compare some results to each other, an arbitrary high precision regular covariance matrix as a reference can be used. Our high precision reference covariance matrix is generated by multiplying the identity matrix by 10^{-10}. An overall indicator to benchmark a result is to compute a benchmark value by $c_b = c_c \cdot c_p$.

The results of the bundle adjustment of our synthetic dataset can be seen in figure 5.5 on page 112. On the left hand side the difference between the estimated camera location and the ground truth camera location and on the right hand side the difference to the angular camera orientation is visualized. In all plots the 2σ bounds of the parameters accuracy is plotted. We observe, that in the average the estimated parameters are inside the uncertainty bounds. The estimated variance factor was approximately $\sigma_0 \approx 1$. The consistency measure of the result including all correlations of the camera is $c_c = 1.3$, that indicate a consistent result. In the Kalman filter results we do not obtain the correlations between all estimated camera orientations, including previous camera orientation. Therefore, neglecting these correlations the resulting consistency measure is an approximation. In case of the bundle adjustment result we get $c_c = 0.9$, which is comparable to the consistency including all correlations.

In the first experiment we will perform the Kalman filter in its simplest form, where the new object points are represented in the Euclidean space and are initialized instantly. The estimated camera orientation parameters as well as their uncertainty are presented in figure 5.6 on page 113. We observe, that the obtained altitude (Z-component) and the latitude angular orientation (ϕ-component) are significantly inconsistent. The estimated camera orientation parameter are correlated to the estimated object points. As the object points are determined by a recursive intersection of the projection rays with small disparities and its probability density is non-Gaussian, a bias occurs in the distance to the object points. The average of the bias of all object point distances will be transfered to the camera orientation parameters via its correlations.

In a second experiment we will use a sliding window camera representation. Here, the state vector consists of twenty camera orientations, c. f. section 4.3.2. This experiment will analyze the influence of correlations to previous camera orientation parameters. Comparing the results in figure 5.6 on page 113 for the single camera state with the results in figure 5.7 on page 114 for the *sliding window* representation we observe only marginal differences. As we do not restrict the trajectory between consecutive camera orientations due to a large system noise, the correlations between the camera orientations will be transfered by the object points only. Therefore, the previous camera orientations are influenced marginally, if the object point locations slightly change effected by the new observations.

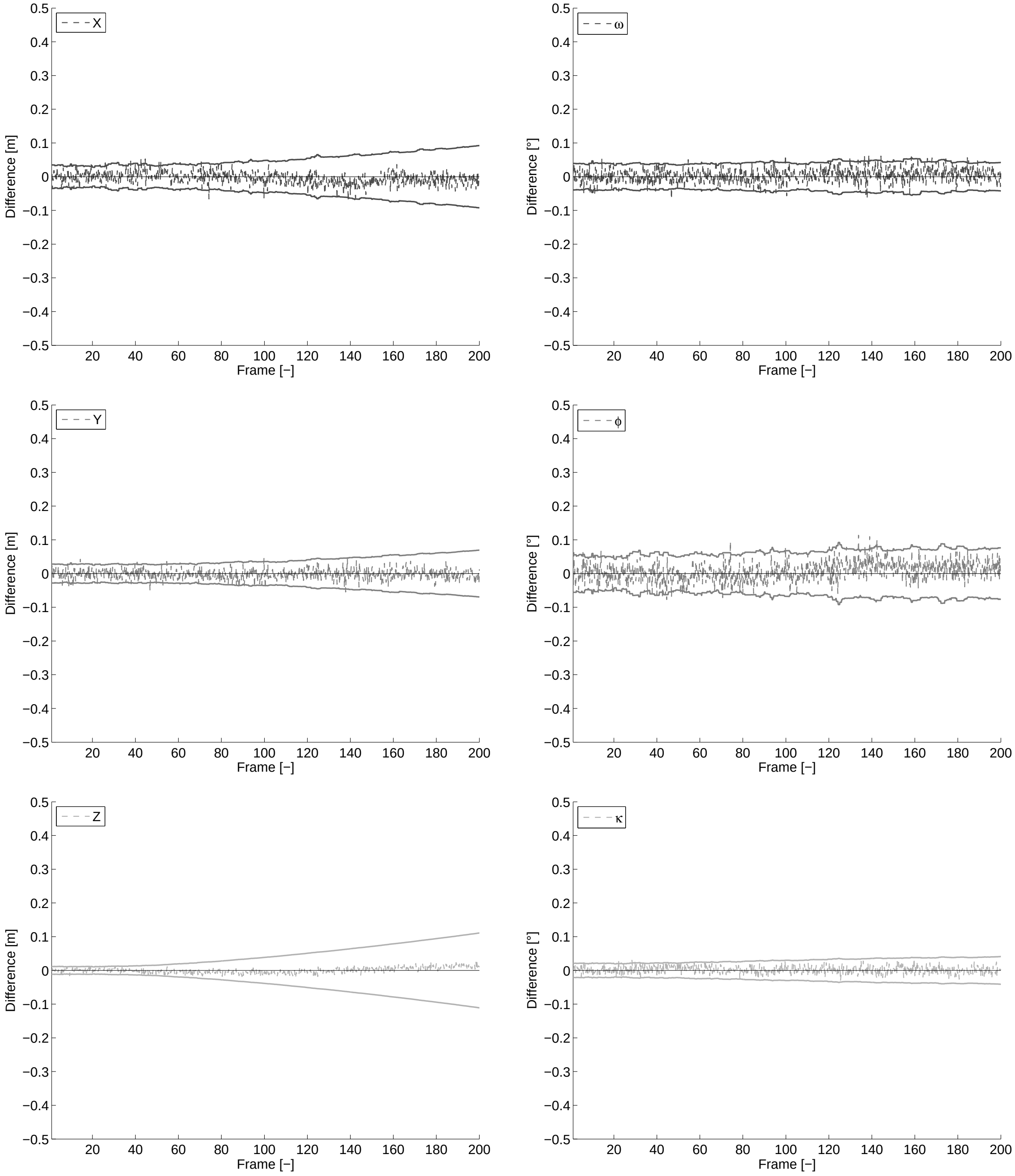


Figure 5.5: **Left:** Bundle adjustment results for the camera position parameters **Right:** Bundle adjustment result for the camera orientation parameters

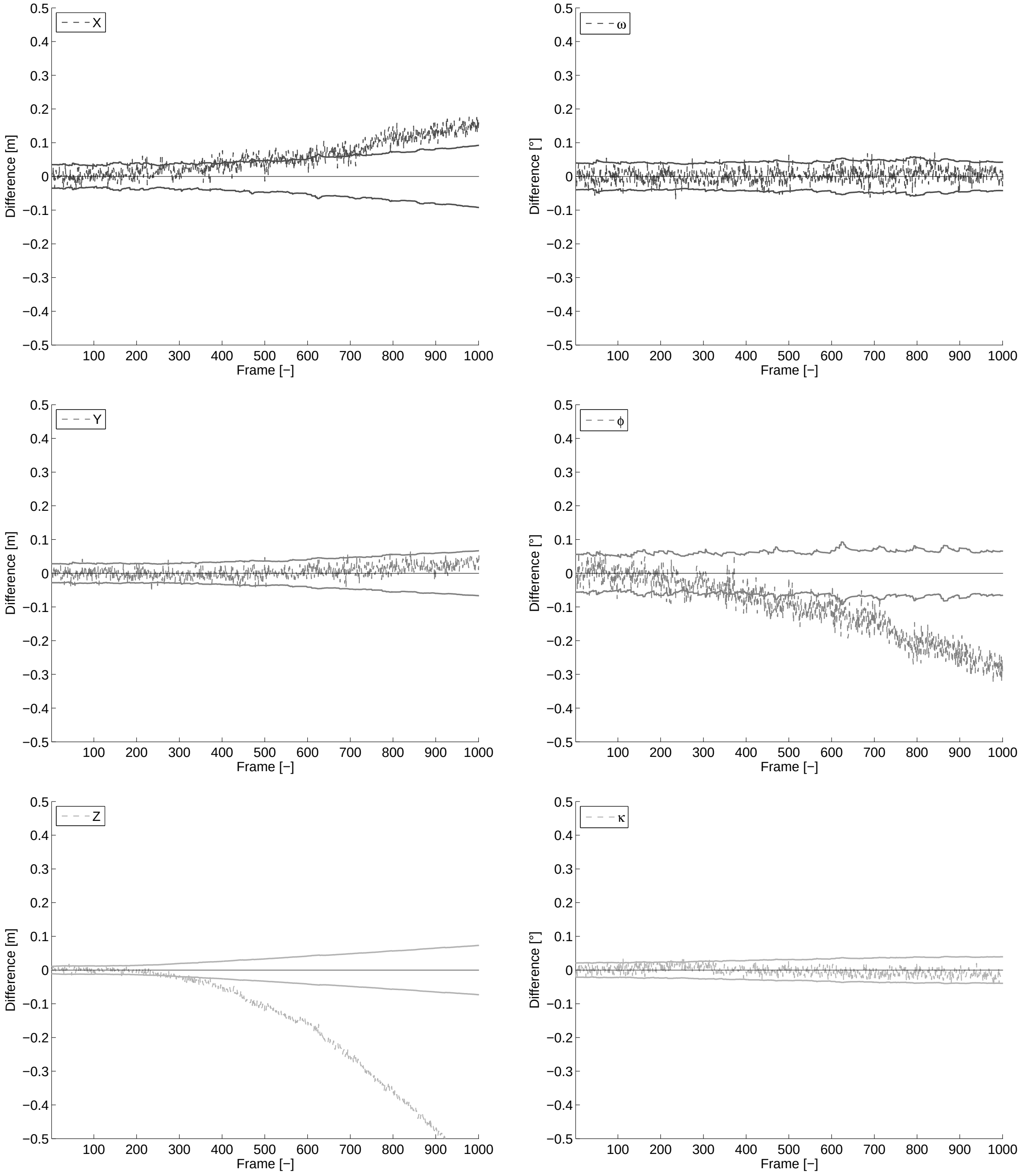


Figure 5.6: Initialization using *Euclidean representation* with ground truth distance. **Left:** Kalman filter result for the camera position parameters **Right:** Kalman filter result for the camera orientation parameters.

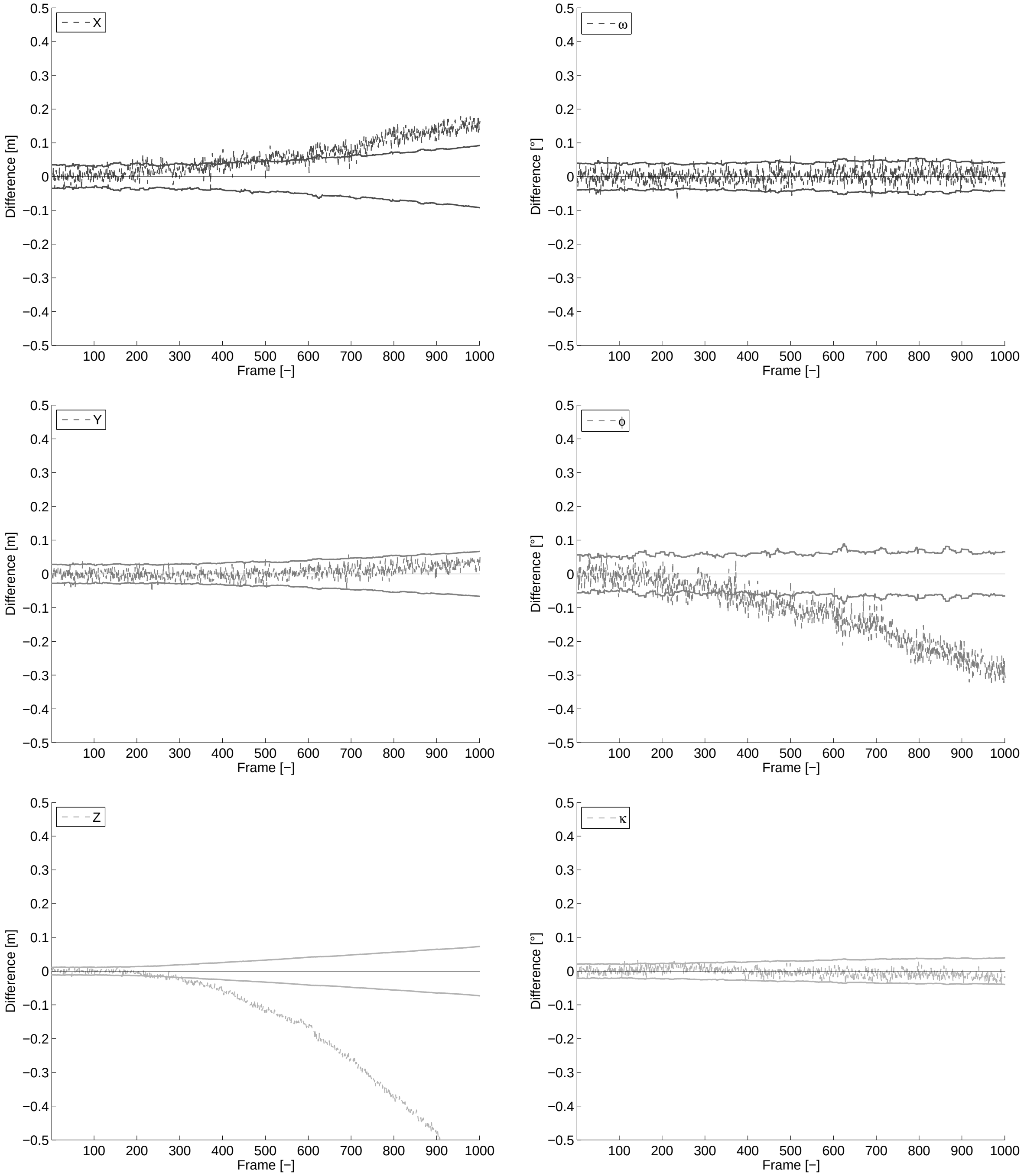


Figure 5.7: Initialization using ground truth distance and a sliding window camera representation (20). **Left:** Kalman filter result for the camera position parameters **Right:** Kalman filter result for the camera orientation parameters.

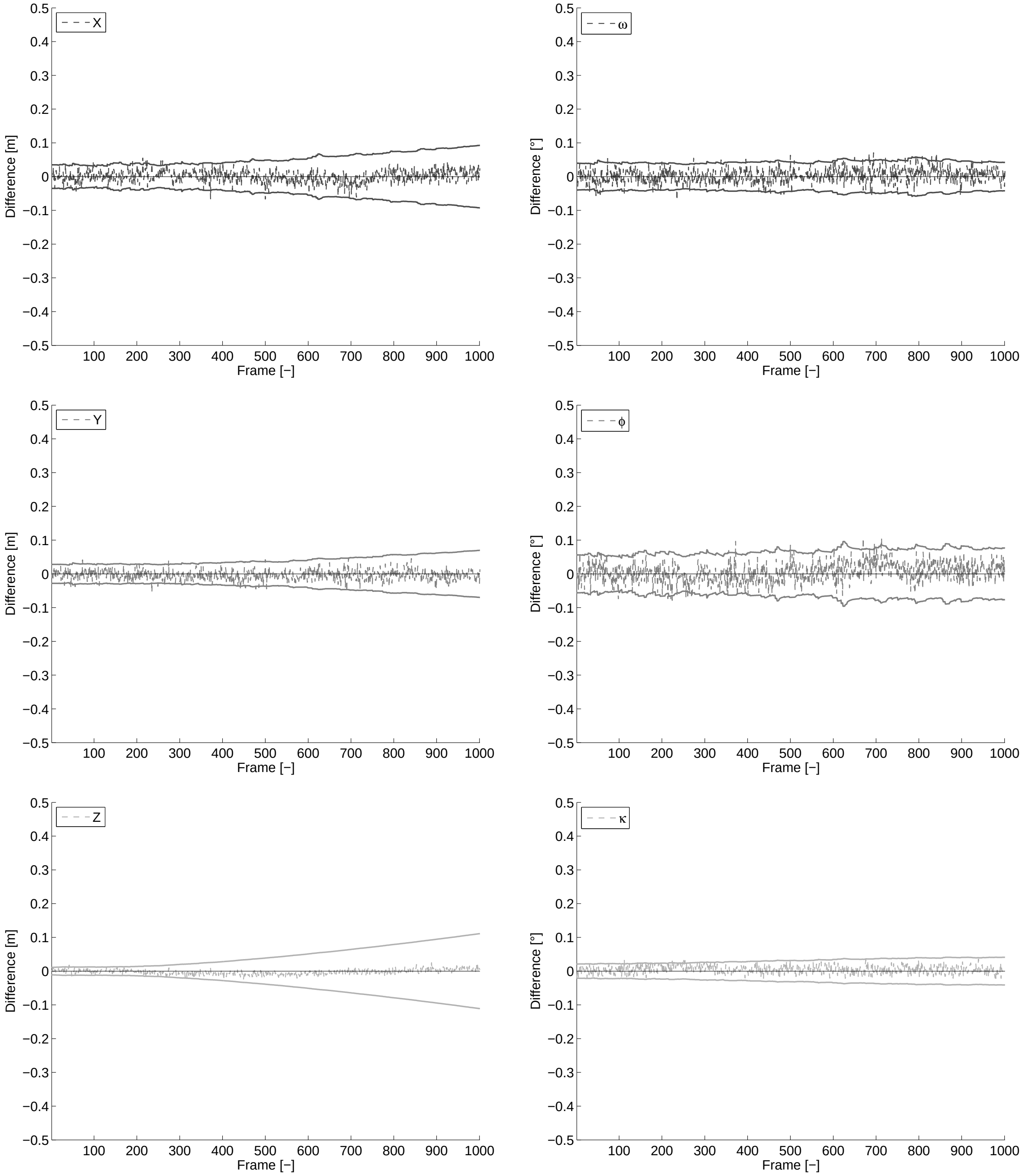


Figure 5.8: Initialization using *inverse distance representation* with ground truth distance. **Left:** Kalman filter result for the camera position parameters **Right:** Kalman filter result for the camera orientation parameters.

The third and forth experiment will be performed using the inverse distance parameterization for new object points. First, we will run the Kalman filter without any reparameterization of the object points to a Euclidean representation. Second, we will analyze the influence of a reparameterization of the object points depending on the roundness of the associated covariance matrix. The roundness reflect the quality of Gaussian distribution in the Euclidean space due to the small baseline intersection problem. In figure 5.8 on page 115 the resulting camera orientations for the experiment using the inverse distance is presented. We can observe, that the results are consistent according to its uncertainty. Obviously, the continuously introduced bias will be eliminated completely. As the main influence of the bias will be visible in the Z-component of the camera location we can see the influence of a reparameterization of the inverse distance to the Euclidean representation using different roundness thresholds in figure 5.9 on page 117. A reparameterization using a roundness threshold of $\Theta_L = 0.5$ seems to have no influence onto the camera orientation estimation. The bias grows with decreasing the roundness thresholds.

The experiments presented so far introduced an object point instantly assuming an arbitrary distance. In our case this distance was chosen knowing its ground truth distance. As we already mentioned, also a delayed initialization of new object points is possible. This technique requires the collection of observations for a limited time slot. Hence, a delayed initialization is able to predetermine the object points location and we do not need to assume an arbitrary distance. A second advantage is, that the method is able to previously rate the influence of the bias due to the intersection angle. First we will show the results using the sliding window state representation. An object point will be initialized only if the roundness measure of the covariance of the intersection is below a threshold. The object points roundness measure can be obtained using the estimated camera orientations and the collected image observation. As we can see in figure 5.10 on page 118 , introducing new object points by a delayed initialization, the estimated uncertainty of the parameter vector increases in contrast to an instant initialization. This is due to the fact, that the determination of the camera orientation parameters depends on the distribution of the object points on the surface and therefore on the accuracy of the spatial resection. The geometric configuration using a delayed initialization becomes weaker. The uncertainty of the spatial resection will increase, if new object points will be initialized later by increasing Θ_L. We can observe, that the delayed initialization cannot prevent the effect of the bias in our experiments. Using

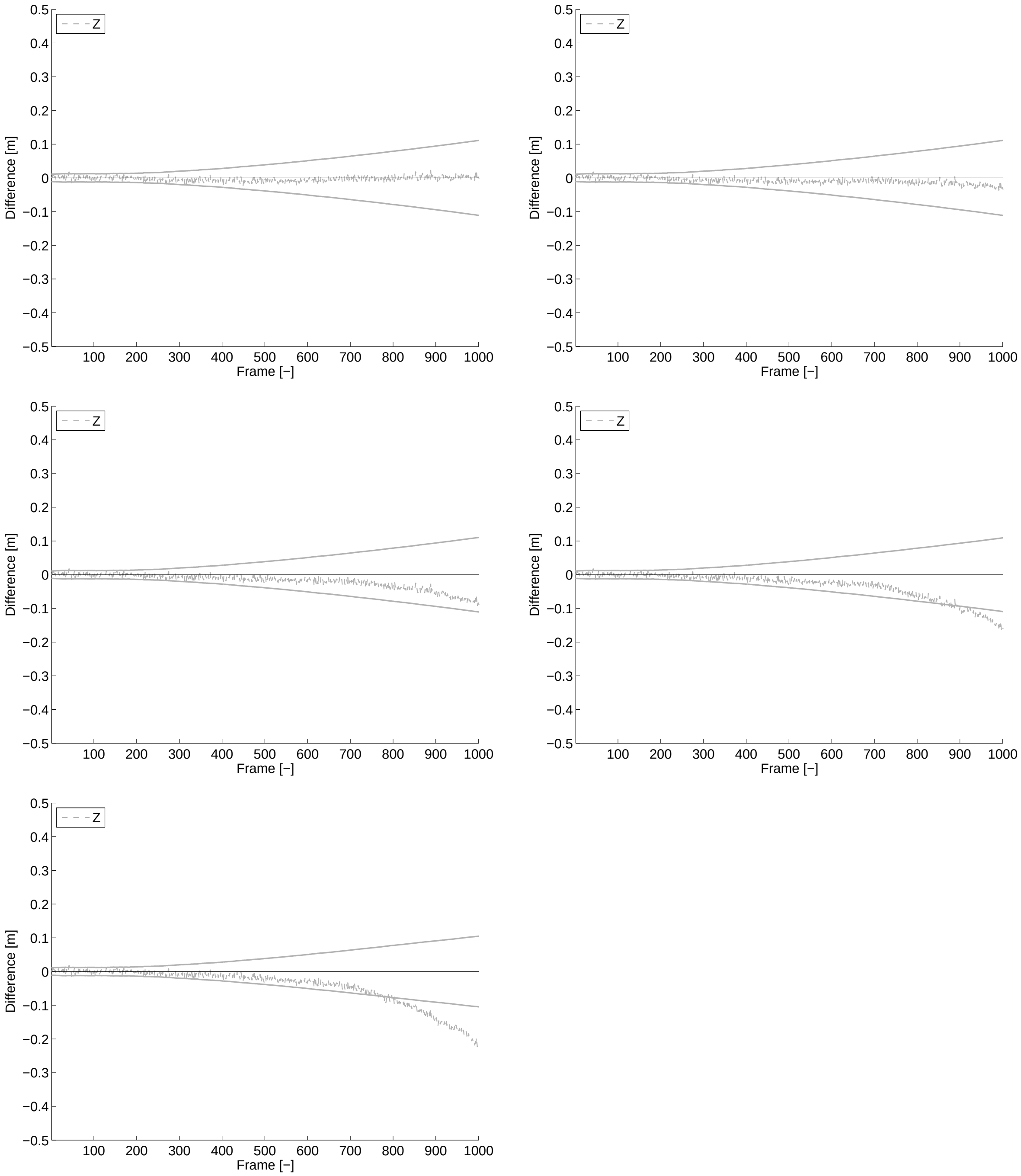


Figure 5.9: Initialization using *inverse distance representation* with ground truth distance and reparameterization with roundness test values of $\Theta_L = \{0.5, 0.25, 0.1, 0.05, 0.025\}$ (left to right row wise). Kalman filter result for the camera position parameters in Z directions.

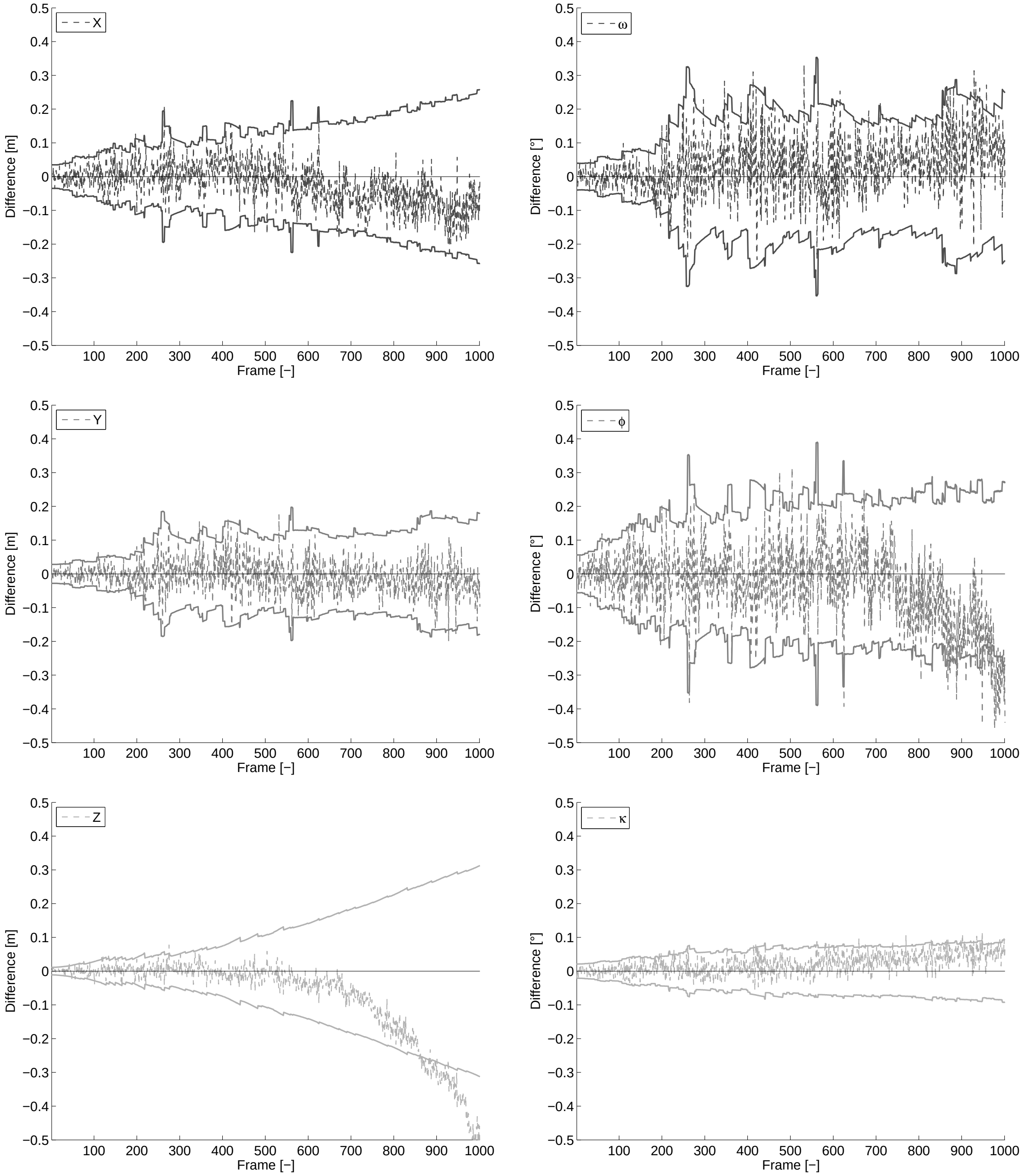


Figure 5.10: Initialization using ground truth distance and Sliding Window with a roundness test value of $\Theta_L = 0.05$. **Left:** Kalman filter result for the camera position parameters **Right:** Kalman filter result for the camera orientation parameters.

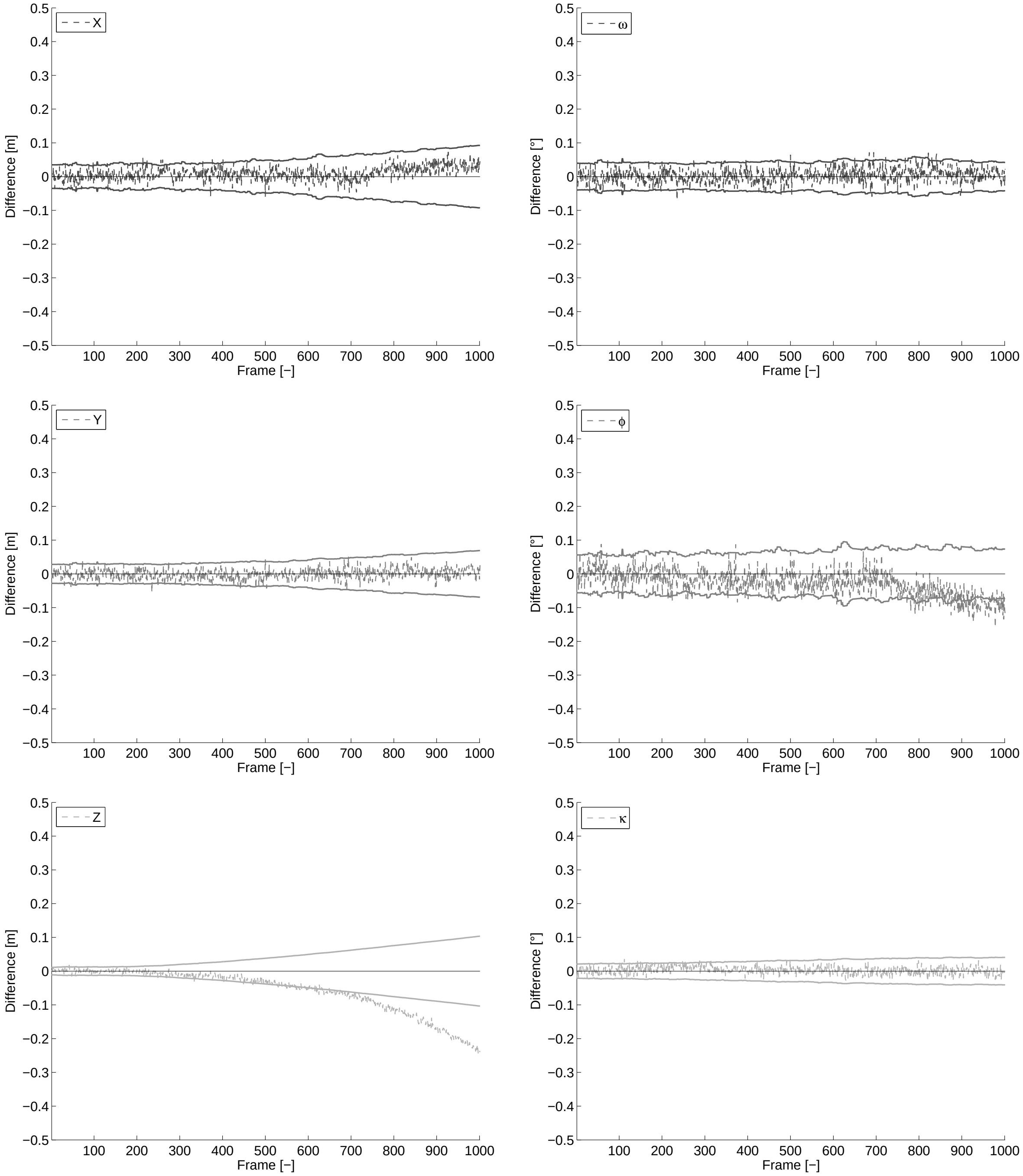


Figure 5.11: Initialization using ground truth distance and Sliding Window with a roundness test value of $\Theta_L = 0.01$. **Left:** Kalman filter result for the camera position parameters **Right:** Kalman filter result for the camera orientation parameters.

a smaller test value Θ_L, which will effect that new object points are introduced earlier, the spatial resection will be much more stable. The results can be seen in figure 5.11. Comparing to the first experiment (figure 5.6 on page 113), the effect of the bias is significantly reduced.

The sliding window and the inverse distance representation will enlarge the state vector significantly. In the following we will analyze the results using the *stable initialization procedure* introduced in section 4.3.4.4. Again, new object points will be introduced delayed. We will run the algorithm with a roundness test value of $\Theta_L = 0.05$ and $\Theta_L = 0.01$. The results are presented in figures 5.12 on page 121 and 5.2 on page 122. In contrast to the delayed sliding window initialization, we cannot observe an increase in the uncertainty of the estimated camera parameters. Obviously, the weak spatial resection will be compensated by the high inner accuracy of the reconstructed object points according to the transfer of the correlations between adjacent object points. If we decrease the roundness test value Θ_L, a new object point will be introduced earlier. The new object points are introduced with larger uncertainty. Therefore the effect of the bias increases. Hence, an implementation using the *sliding window initialization* or the *stable initialization method* should find an appropriate value for the roundness test value as a trade off between the influence of the bias and a stability of the spatial resection.

Discussion: The experiments presented so far were running on a synthetic dataset to analyze the expected accuracy of a Kalman filter based solution for the task of simultaneous localization and mapping in case of a photogrammetric strip. In table 5.6 and figure 5.14 on page 124 the computed consistency, precision and benchmark values for the experiments are summarized.

We observe, that the method to initialize new object points has a significant influence on the achievable consistency and accuracy of the results. A naive introduction of the object point using Euclidean coordinates assuming a realistic distance results in a drift of the camera orientation. This drift will be extremely effected by the altitude as well as in the latitude orientation of the strip. As we do not restrict the camera trajectory in the dynamic model the experiment does not show any improvement using the sliding window representation.

We analyzed the results of four major kinds of initialization methods to indicate a reduction of the effect on the bias. As we can see in figure 5.14 the inverse distance parameterization seems to get the best results. A reparameterization of the object points from the inverse dis-

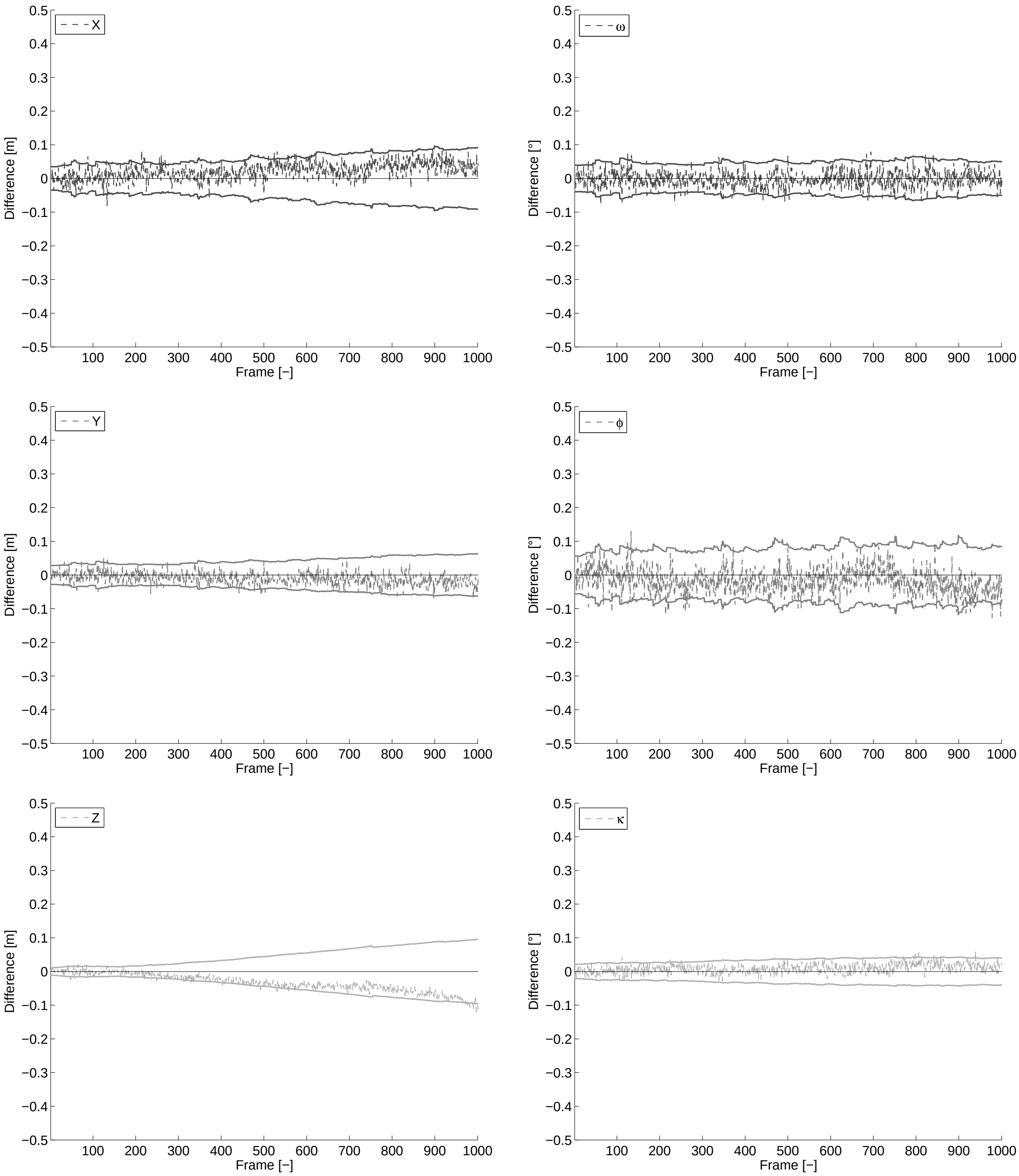


Figure 5.12: Initialization using the *stable initialization procedure* with a roundness test value of $\Theta_L = 0.05$. **Left:** Kalman filter result for the camera position parameters **Right:** Kalman filter result for the camera orientation parameters.

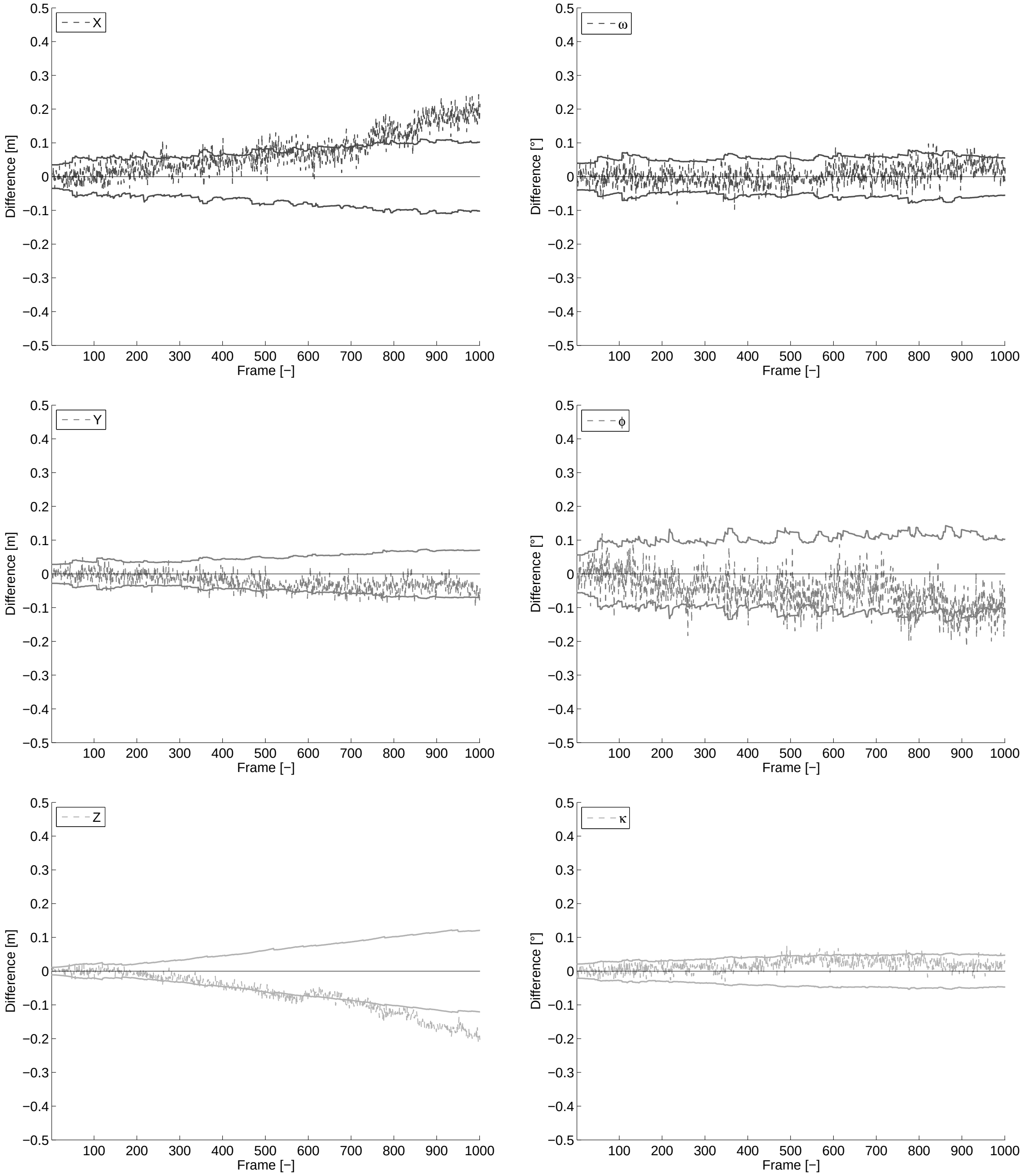


Figure 5.13: Initialization using the *stable initialization procedure* with a roundness test value of $\Theta_L = 0.01$. **Left:** Kalman filter result for the camera position parameters **Right:** Kalman filter result for the camera orientation parameters.

Method	consistency c_c	precision value c_p	benchmark c_b
Euclidean	6.77	106496.5	720981.3
Euclidean $k = 19$	7.06	102320.8	722384.8
Inverse distance	2.88	115726.2	333291.4
Inverse distance $\Theta_L = 0.025$	3.14	115093.3	361393.0
Inverse distance $\Theta_L = 0.05$	3.00	115699.9	347099.7
Inverse distance $\Theta_L = 0.1$	2.91	115699.9	336686.7
Inverse distance $\Theta_L = 0.25$	2.88	115769.4	333415.9
Inverse distance $\Theta_L = 0.5$	2.88	115728.8	333298.9
Sliding window $\Theta_L = 0.05$	5.22	572035.8	9860269.0
Sliding window $\Theta_L = 0.01$	3.34	115857.6	386964.4
Stable init. procedure $\Theta_L = 0.05$	3.55	135295.9	480300.4
Stable init. procedure $\Theta_L = 0.1$	4.49	186672.6	838160.0

Table 5.6: Consistency c_c, precision c_p and benchmark value c_b of the Kalman filter results. The first eight rows are the results for the instantly object point initialization methods followed by the two delayed initialization methods. The best results are be obtained by the inverse distance representation.

tance to the Euclidean parameterization using a roundness threshold below $\Theta_L = 0.25$ will cause an observable drift. The major problem using the proposed methods for a delayed initialization is to find a convenient threshold Θ_L so that the influence of the bias does not occur and the spatial resection will be as precise as possible. Moreover, these methods require a uniform distribution of the object point locations to guarantee a good geometric configuration for the spatial resection. We recommend to use the inverse distance parameterization using reparameterization to reduce the effect of the bias and to keep the state vector sufficient small.

Remark: In appendix A.5 we additionally visualize the differences between the ground truth object point coordinates and the estimated coordinates of the object points of the experiments.

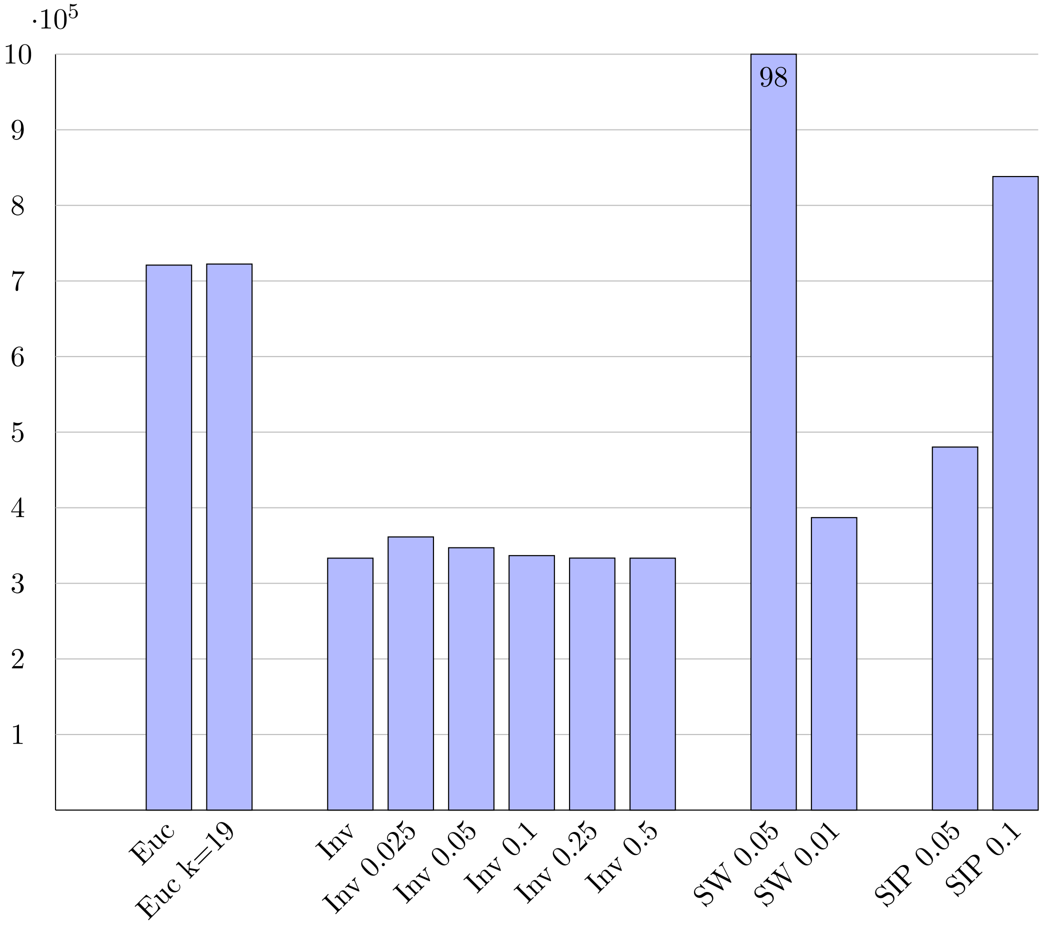


Figure 5.14: Benchmark summary of the Kalman filter results.

5.3 Evaluation on simulated image data

In this section we will show the capability of the proposed estimation framework on a more realistic scenario. A first synthetic image sequence was rendered using real image data of a high altitude airborne image mapped on a planar surface. According to the previous simulations and the following full real data experiment our environment is specified in table 5.7. To evaluate the accuracy of the point based tracking algorithm in a realistic manner, the image sequence will be disturbed by Gaussian image noise with a standard deviation of $\sigma_I = 5$ [gr] of the intensity. Using consumer cameras in video mode this is a realistic assumption about the image perturbation by noise. The mean number of features and a uniform sampling of the features in the image sequence can be adjusted by choosing a circular free space around a feature, where no new features can be initialized. However, the uniform sampling depends on the content of the image and therefore cannot be guaranteed. Some control points are

camera parameter	value
camera resolution	800×600
principal point	image center
frames per second	25
field of view	$\approx 60°$
expected tracking accuracy	0.25 [pel]
mean features per image	23
path parameter	
strip length	200 [m]
flight altitude	30 [m]
flight speed	5 [m/s]
base line	0.20 [m]

Table 5.7: Parameter of the experimental environment using a synthetic image sequence

visible in the first images of the sequence, where ground truth data is available (see figure 5.15).

In a first step we run the dataset using a bundle adjustment to indicate that the obtained observations of the tracking algorithm can be used for the trajectory reconstruction in a Kalman filter. Approximate values for the camera orientations will be obtained by the known ground truth camera orientation. Approximate values for the object points will be computed by the intersection of the first projection ray and the known surface plane. In figure 5.16 on page 127 the results for the estimated camera orientation are presented. The square root of the estimated variance factor of the bundle adjustment is a quality measure of the accuracy of the point based feature tracking and is $\widehat{\sigma}_0 = 0.36$ [1]. According to the assumed tracking accuracy of $\sigma_{xy} = 0.25$ [pel] the estimated observation standard deviation is $\widehat{\sigma}_{xy} \approx 0.09$ [pel]. The consistency of the result is $c_c = 42.6$, the precision measure is $c_p = 49694$. This indicates, that the result is not consistent. The result shows, that the accuracy of the point based tracking is much more precise, by a factor of three, as we expected. However, it seems that there are outliers in the observations due to apparently non-stationary object points. Therefore, we will run the dataset using a bundle adjustment including a reweighting method proposed in section 3.3.2. Here we will use the *Huber*-weighting function for a robustification using a tuning factor of $k = 3$. The results are presented in figure 5.17 on page 128. The

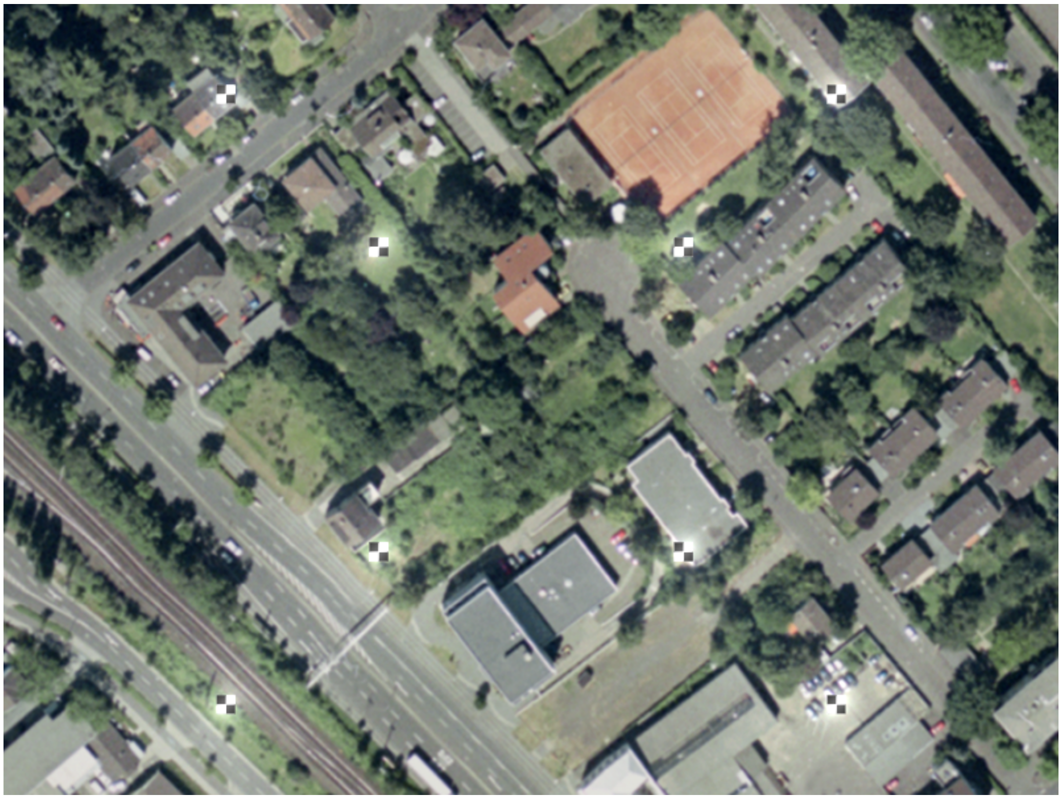


Figure 5.15: First image of the simulated image data sequence with signalized control points

square root of the estimated variance factor of the robust bundle adjustment was $\widehat{\sigma}_0 = 0.26$ [1] and therefore the estimated observation standard deviation is $\widehat{\sigma}_{xy} \approx 0.07$ [pel]. The overall number of observations in this experiment is $n = 46298$, where $n_w = 1375$ observations are identified as outliers and reweighted. The consistency and the precision measure are $c_c = 42.8$ and $c_p = 28529$. Comparing the results in the figures 5.16 with 5.17 on page 127 and 128, we observe that the robustification in the bundle adjustment solution is able to reduce the effect of the outliers, but does not lead to an overall consistent solution. Apparently, the used robustification method does not identify any outlier and so it cannot remove their influences completely.

Now, we will apply the dataset to the Kalman filter approach. In this experiment we will introduce new object points using the inverse distance parameterization with the same conditions as before. The tracking accuracy will be assumed to be $\sigma_{xy} = 0.25$ [pel]. In a second experiment a Kalman filter using a reweighting strategy to reduce the effect of outliers is used. Again, we will apply the *Huber*-weighting function with a tuning factor of $k = 3$. The results can be seen in figures 5.18 and 5.19 on page 130 and 131. In both experiments

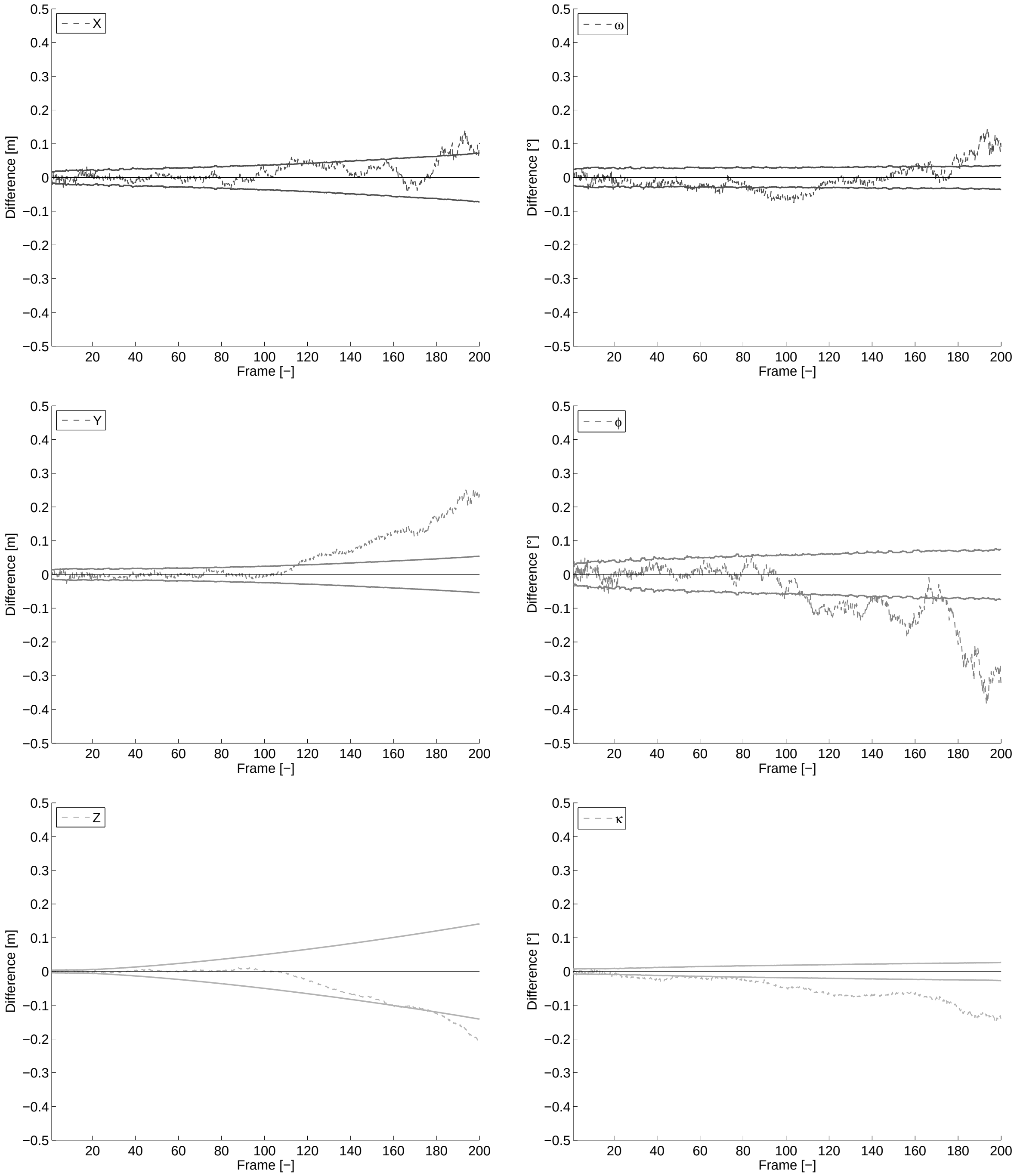


Figure 5.16: Bundle adjustment results for the estimated camera parameters with 2-sigma error bounds

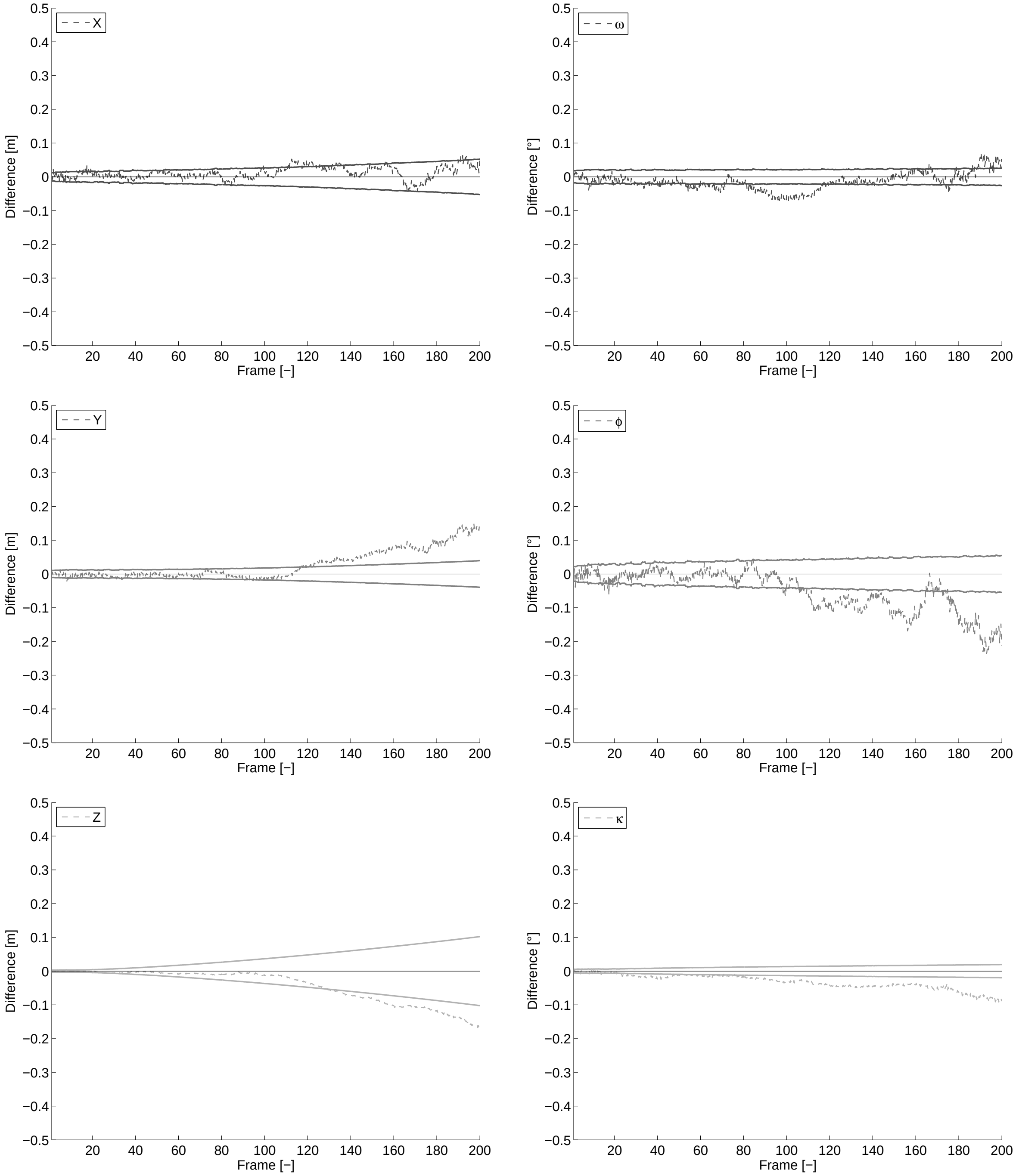


Figure 5.17: Robust bundle adjustment results for the estimated camera parameters with 2-sigma error bounds.

the consistency is approximately $c_c \approx 3.6$ and the precisions are equal. Obviously, in the case of the Kalman filter the robustification has no influence on the results for this dataset. The outliers caused by the tracking are not predominantly characterized by large errors in consecutive images. Rather the erroneous tracking leads to an apparent non-stationary motion of the corresponding object point. Because the reweighting method in the Kalman filter identifies outliers by analyzing the innovation in consecutive images and the influence of a small motion of an object point to the innovation is small, it seems that there is no reason to reweight an observation. In table 5.8 we summarized the benchmark value for the four experiments using the synthetic image dataset. A very interesting fact is, that the Kalman filter based solution performs better than the non-robust version of the bundle adjustment. We guess again, that the reason could be the apparent non-stationarity of the erroneous object points caused by a continuous drift of the tracked feature. In the Kalman filter a violation of the stationary assumption tends to a drift of the object point location. Therefore the residuals in the update step will be slightly decreased and its influence on the camera orientation will be reduced. We can observe that in case of a robust bundle adjustment the influence of the outliers are reduced and the result will outperform the Kalman filter.

experiment	consistency c_c	precision c_p	benchmark c_b
Bundle	42.8	49694	2116964
Bundle (robust)	42.6	28529	1221041
Kalman filter	3.6	427315	1529768
Kalman filter (robust)	3.6	427422	1532663

Table 5.8: Summary of the benchmarking for synthetic image sequence consist of a photogrammetric strip.

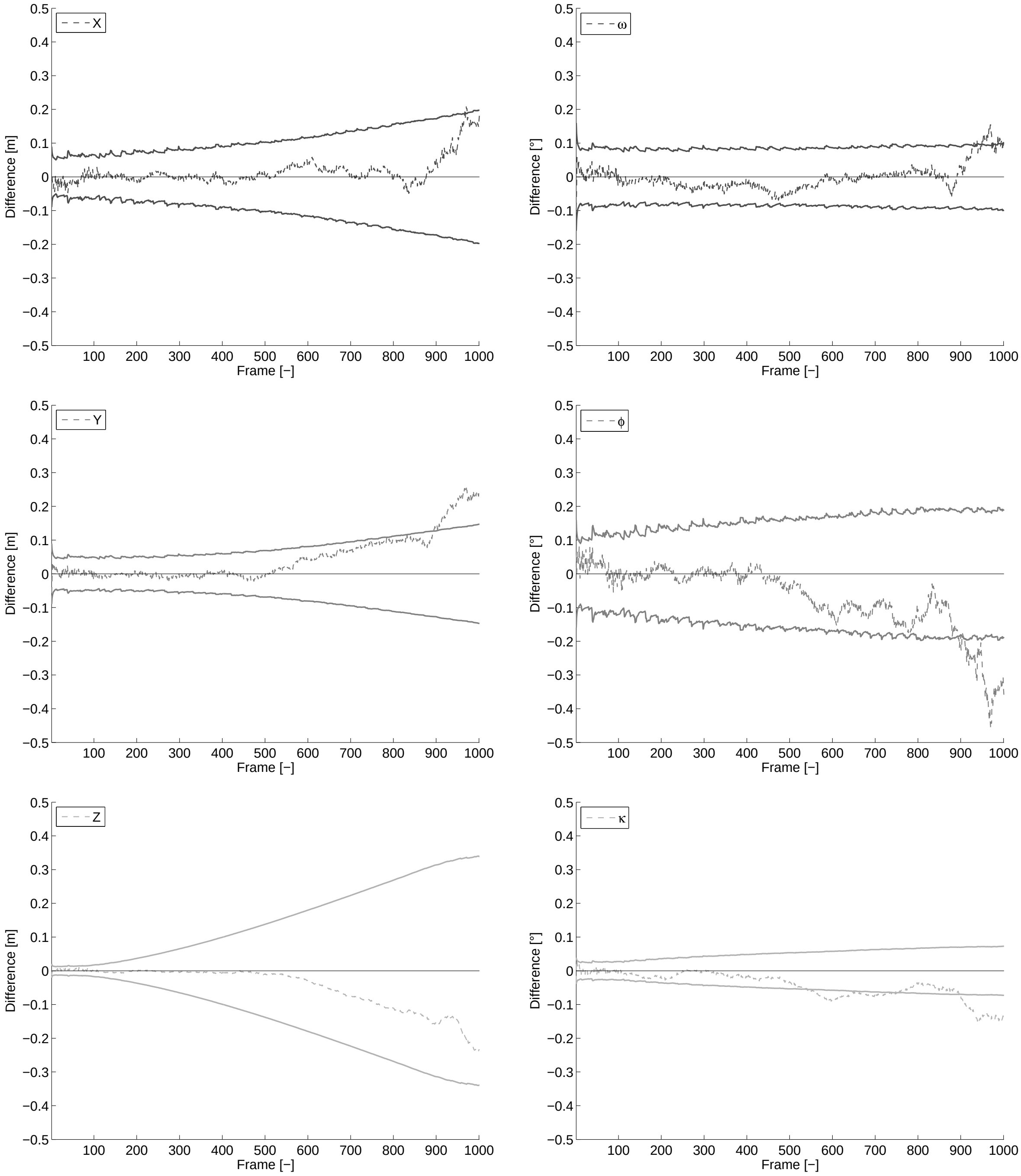


Figure 5.18: Kalman filter results for the estimated camera parameters with 2-sigma error bounds.

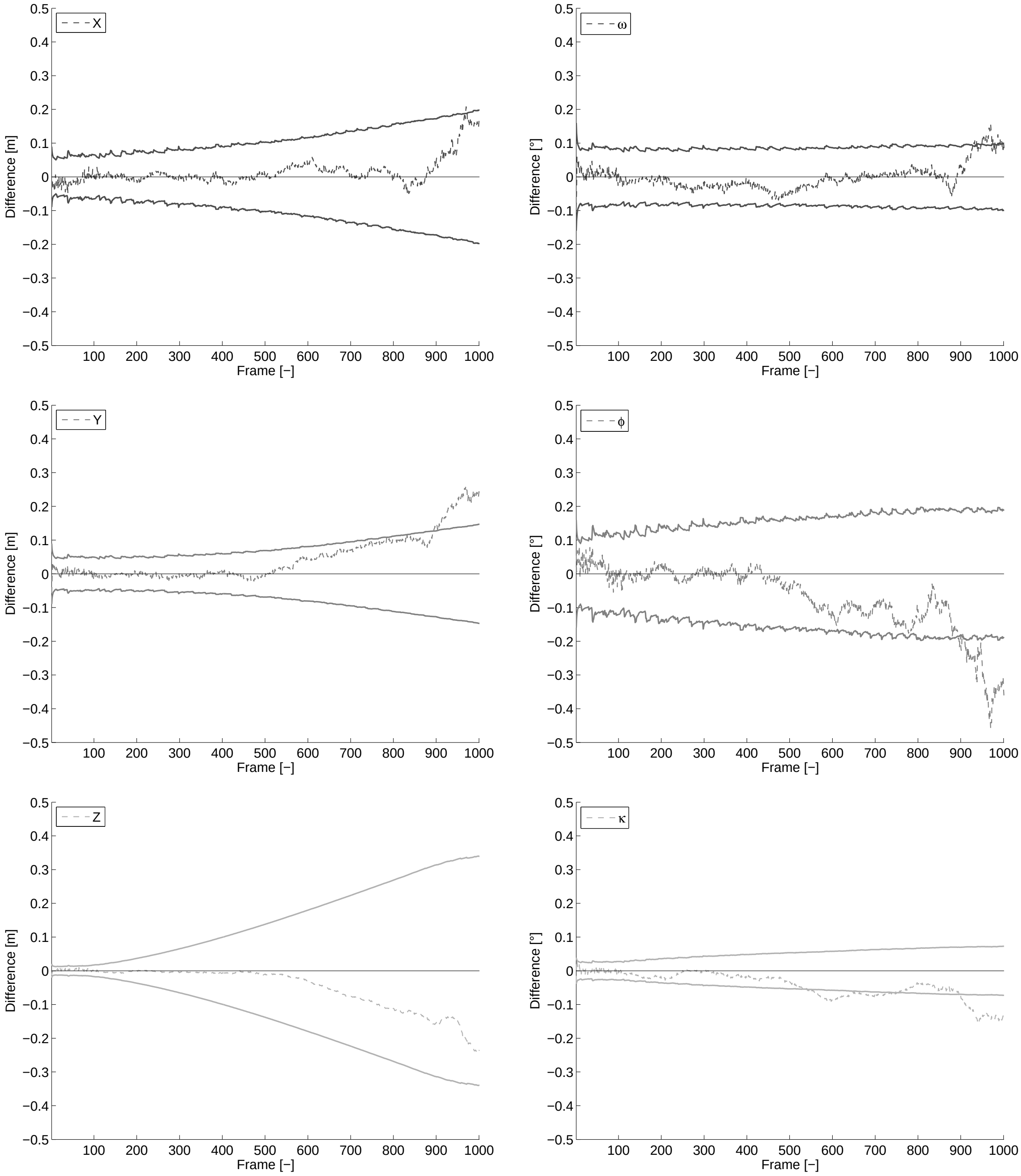


Figure 5.19: Robust Kalman filter results for the estimated camera parameters with 2-sigma error bounds.

The second synthetic scene consists of a model of the *Basilica St. Peter* in Rome. In this experiment the ground truth trajectory of the camera describes a circular trajectory with a radius of 80 m and a flight altitude of 30 m. Consecutive images are rendered every degree. The pitch angle of the camera is adjusted to $\phi = 15°$. In figure 5.20 few selected images can be seen. Four control points with its ground truth data on the front of the basilica will be used to introduce a reference coordinate system. In a first step we applied the tracking algorithm to this sequence. Here two major problems appear. First, after some time the tracking algorithm lose tracks of features. The reasons are occlusions and appearance changes due to perspective changes. This leads to short tracks. Second, the object points are apparently non-stationary due to the occlusions and specular surfaces. Therefore, we adjust the tracking algorithm in a way, that the mean number of simultaneous tracked image points are large ($N_X \approx 220$). In a second step we computed approximate values for every tracked object points by intersecting the projection rays using the ground truth camera orientations.

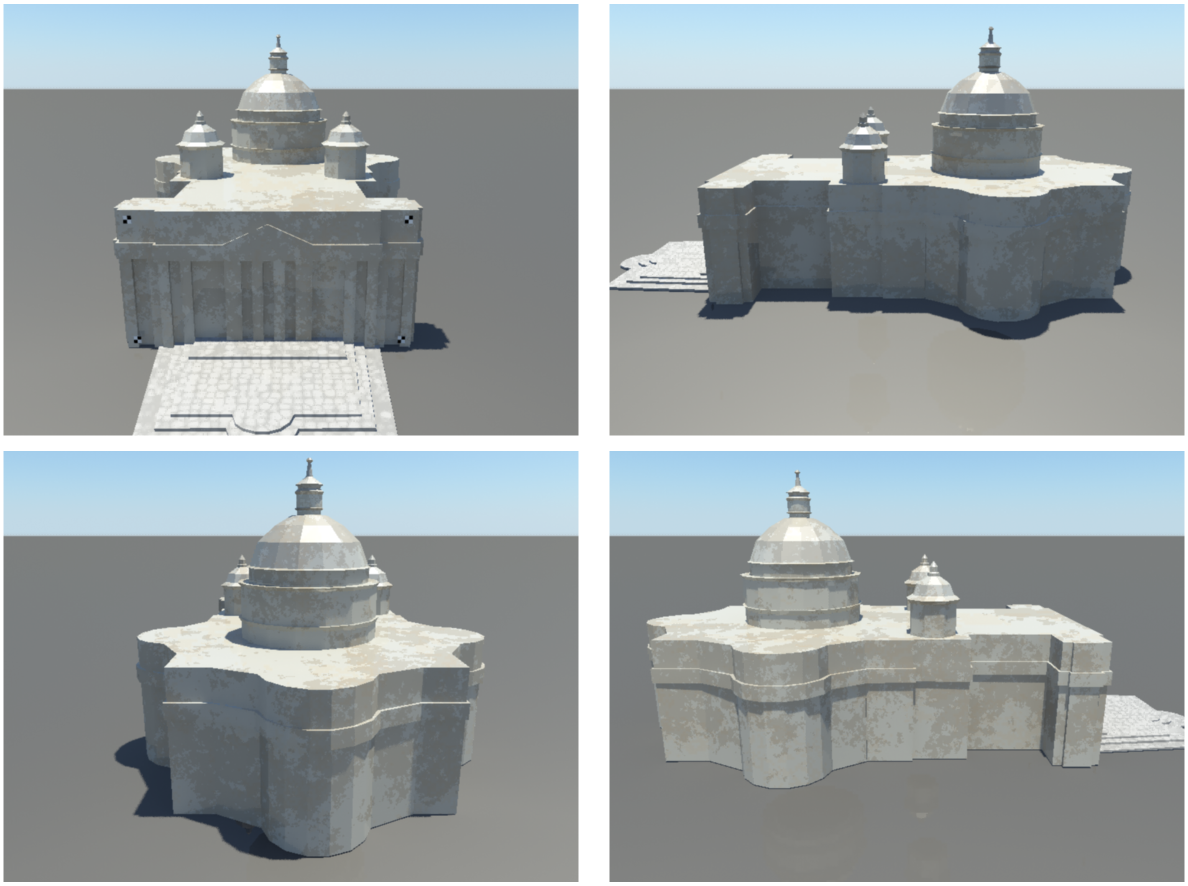


Figure 5.20: Images of the synthetic image sequence of the *Basilika St. Peter* in Rom.

In the first experiment we applied a robust bundle adjustment to the dataset, where the influence of outliers will be reduced by a reweighting method using the *Huber*-weighting function. The algorithm diverges due to the large number of outliers. Obviously, the reweighting method is not able to detect the huge number of outliers and to reduce their effects.

In a second experiment we applied a bundle adjustment to this synthetic dataset by eliminating outliers iteratively. Here, we estimate the variance components of all groups of observations to have a part of the determination of individual object points, c. f. section 3.3.2. In every iteration all groups with a variance component above a threshold of $\Theta_{\sigma_0} > 2$ will be identified. We eliminate the observations of the worst 10% groups and run the algorithm again. The resulting residuals between the ground truth and estimated camera trajectory are shown in figure 5.21. The computed square root of the variance factor of the remaining observations is $\widehat{\sigma}_0 = 0.63$ [1]. The dataset contains of $N = 2536$ object points. The outlier detection method classified 1166 object points as outliers. We can observe, that the result is not consistent. The estimated accuracy does not reflect the true uncertainty of the results. Also the outlier detection method based on the variance component cannot detect all apparently non-stationary object points in this sequence. A more robust implementation should detect outliers using a filter step prior to the adjustment. The filter can determine the relative orientation between two selected images using the coplanarity constraint. In case of a huge number of wrong correspondences a robust estimation can be achieved using the RANSAC-procedure following Fischler & Bolles (1981). Object points should be eliminated, where the coplanarity constraint does not hold. From the concept of the Kalman filter we typically are not able to fall back to previous observations. Observe, that these observations are already encoded in the state of the system and cannot be removed.

In a third experiment using this dataset we tried to estimate the trajectory by a Kalman filter based approach using the inverse distance representation. We observed an early divergence of the filter process. We suppose that the huge number of outliers in the observation cause numerical instabilities.

In the previous sections we analyzed the capability of the Kalman filter to the task of simultaneous localization and mapping from image sequences. The experiments were restricted to the case of classical photogrammetric strips. We first derived a general law to assess the stochastical accuracy of a photogrammetric strip. The law is generalized to get the accuracy under different kinds of flight configurations. Second, we compared the

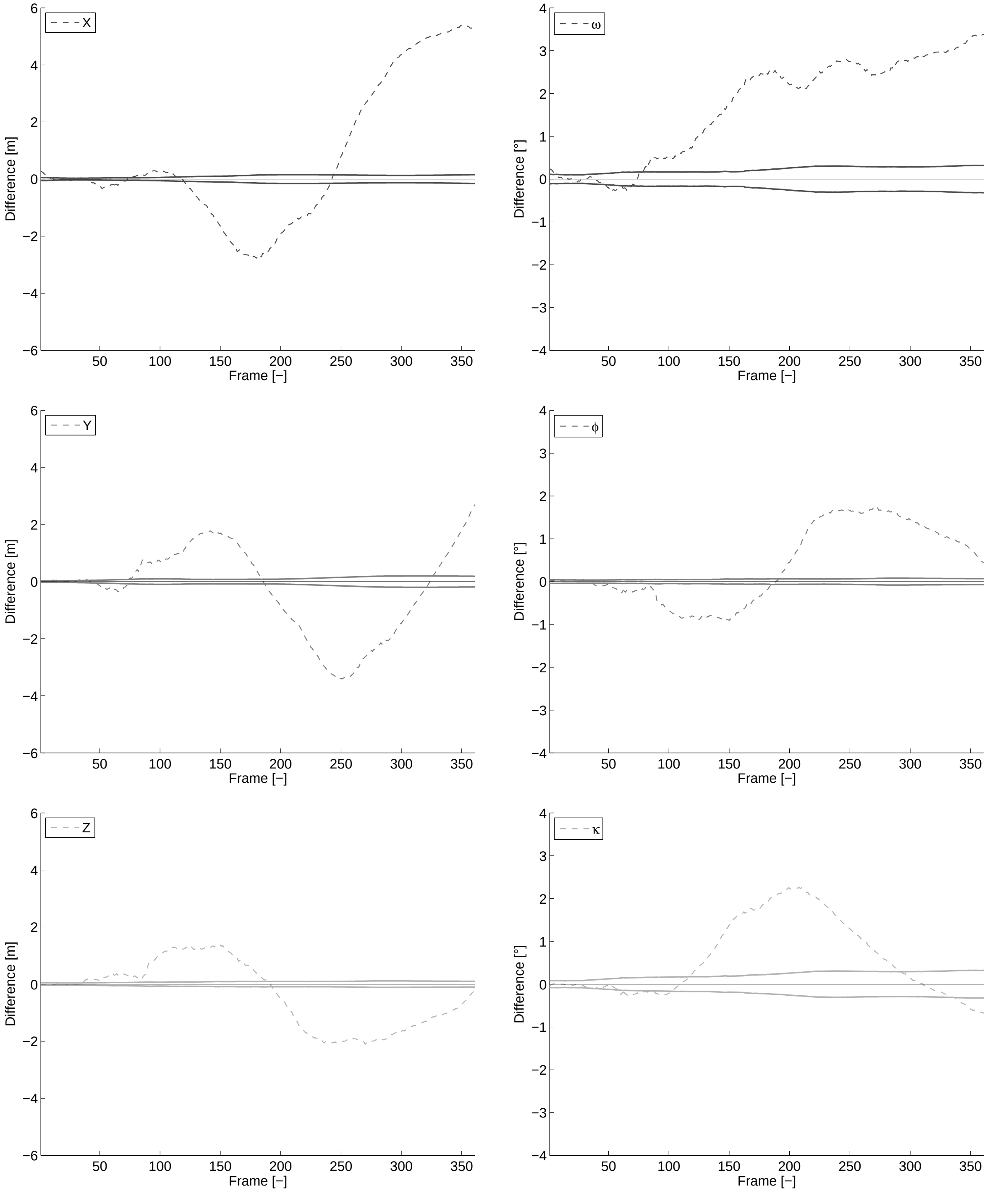


Figure 5.21: Robust bundle adjustment solution using outlier detection by variance component estimation on the *Basilika* dataset.

Kalman filter results their accuracy and precision using synthetic data. We showed that a bias occurs. However, the influence of the bias can be reduced using advanced initialization and representation strategies for new object points. Synthetic image sequences were used to evaluate the capability of the proposed methods using a real point based feature tracking.

5.4 Real experiment evaluation

In this section we will apply our Kalman filter approach to a real image sequence acquired by an unmanned aerial vehicle. In contrast to the experiments using simulated data we do not have control about systematic errors that occur due to camera calibration errors and unexpected errors in the feature tracking. First we will give an overview about the hardware system and the flight-test area. Thereafter, we will present our method to reduce systematic errors and get high accuracy reference data for the camera orientation. We will present the results of the Kalman filter based approach and evaluate the results.

5.4.1 Used hardware architecture

The real data experiment shown below is based on image sequences taken with the UAV from Microdrones GmbH. The drone shown in figure 5.22 is an electric powered quad-copter, which is manually controlled. Its empty weight is 585 g, it can carry up to approximately 200 g of payload and its diameter is approximately 70 cm. This drone is equipped with a Panasonic Lumix camera with a resolution of 848 x 480 pixels, a viewing angle of approximately 60° and a frame rate of 30 Hz in video mode. The camera can be tilted from 0° to 90° nadir angle. The battery allows a flying time up to approximately 30 minutes. The image sequence is stored on a flash card and compressed as a quick time movie. In addition, the drone carries an inertial system for flight stabilization and a code based GPS-receiver for automatic position hold. The drone's position, low resolution video data of the camera and status data of the drone will be transmitted to a base station for flight guidance of the operator. We point out that due to the low weight of the drone it is difficult to steer manually along a defined flight trajectory. Especially, the drone is highly sensitive to wind changes that cause high angular accelerations of the camera orientations. As a consequence the images taken by the camera can be unsharp.

We already mentioned that the proposed algorithms are developed for calibrated cameras

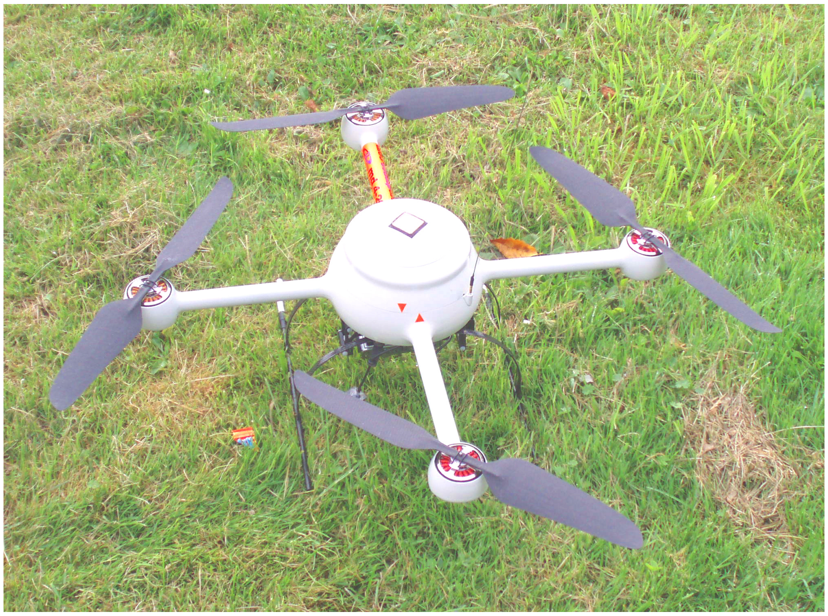


Figure 5.22: Used hardware. Drone MD 4-200 from Microdrones © equipped with a Panasonic Lumix video camera.

only. The used camera is a low cost consumer camera with variable focal length. In the video mode the focal length can be fixed to infinity. For a practical use of the whole system the calibration of the camera has to be done in a laboratory. First we will analyze the stability of independent camera calibrations. Therefore, we performed four independent calibrations in the laboratory on different days. The calibration is done with a radial symmetric model up to a quadratic term (c. f. McGlone *et al.* (2004) p. 297ff), where the principal distance c, scale difference for the principal distance m, principal point x_h and y_h, shear s and two coefficients A_1 and A_2 for radial symmetric polynomial distortion are estimated. Given the parameter for the calibration we compute a look-up table for every pixel to obtain a direction vector for observed object points in the camera coordinate system. The average of the four look-up tables is used as our final calibration. In the figures 5.23 the absolute differences between the average calibration to the four independent calibrations projected to the image space using an average principal distance are visualized. In table 5.9 the minimum and maximum of the error displacements between the average calibration and independent calibrations are summarized. We can observe that the calibration can be assumed to be stable up to an error of 0.6 pixel displacement. This non-linear error in the calibration will effect the geometric reconstruction as a systematic error in an unpredictable manner.

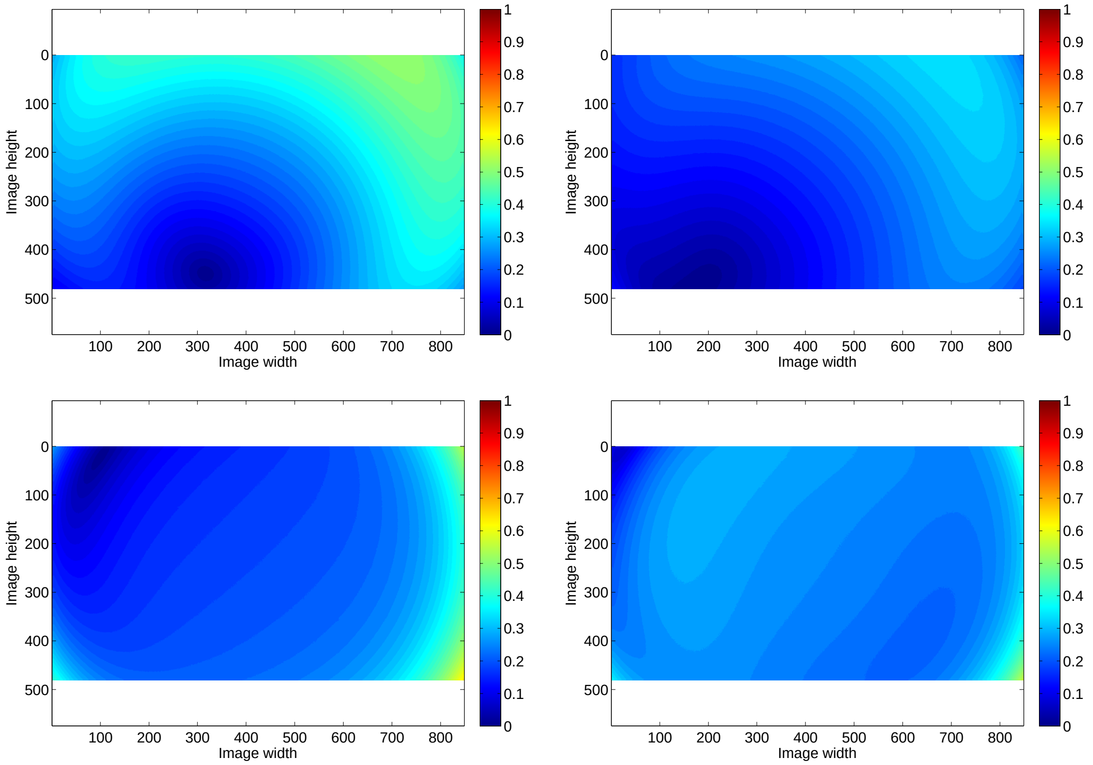


Figure 5.23: Absolute displacement error in pixel between independent camera calibrations using the average principal distance.

	width (min/max) [pel]	height (min/max) [pel]	absolut (min/max) [pel]
A	-0.40/0.21	-0.03/0.41	0.00/0.51
B	-0.05/0.29	-0.25/0.02	0.00/0.34
C	-0.24/0.53	-0.12/0.34	0.00/0.63
D	-0.44/0.09	-0.34/-0.02	0.06/0.55

Table 5.9: Minimum and maximum of the absolute displacement error in pixel between independent camera calibrations.

5.4.2 Flight-test area and reference data

Our real image sequence was acquired by the UAV in our flight-test area shown in figure 5.24, that is also used as an agriculture-test area. The test area consists of a grid of signalized control points. The grid is defined in a way, that following the predefined flight path at least four control points are visible in every image. Therefore, approximate values of the camera orientation can be computed using a spatial resection. The control point coordinates are

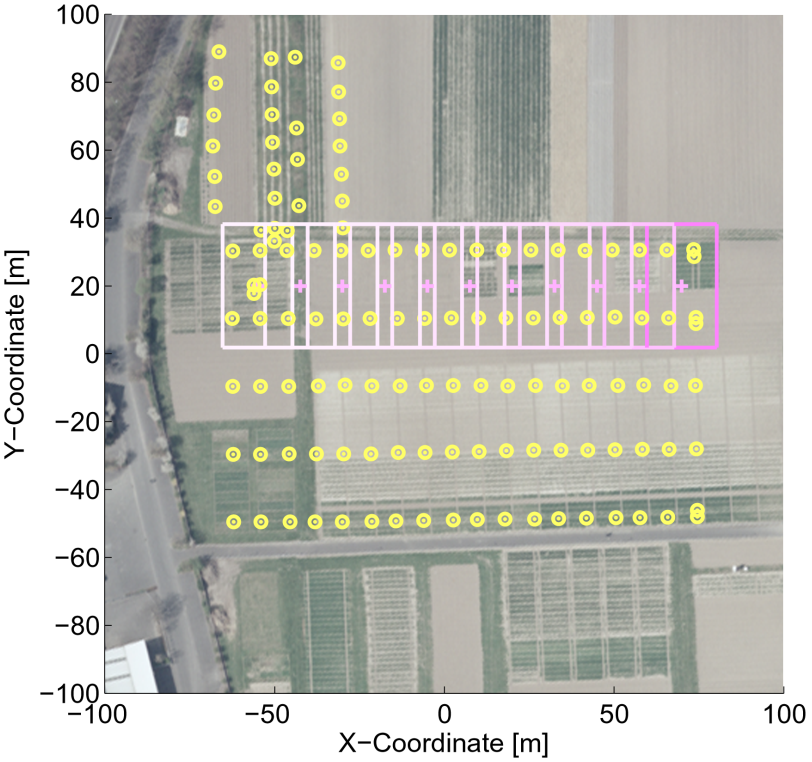


Figure 5.24: High altitude image of the flight-test area and control point field in the test area in a local coordinate system. Additionally the planed camera trajectory is visualized.

measured using a *Laser Total Station* with an overall accuracy of approximately $\sigma = 1$ cm in all coordinate components.

The overall acquired image sequence consists of multiple photogrammetric strips. A part of the image sequence consisting of 978 images is taken to perform the test of our algorithm. The average altitude over ground is approximately $H_g = 30$ m, the length of the strip is 130 m. Because of the light weight of the drone and the influence of the wind an overall vertical view cannot be guaranteed. The average flight speed is $v \approx 3\frac{m}{s}$ and the flight time is ≈ 40 seconds.

To evaluate the results of our Kalman filter approach, we perform a bundle adjustment incorporating all control points visible in the image sequence with known coordinates and all tracked features. The control points in the images are identified manually by an operator and are measured by a template matching with subpixel accuracy. We assume a measurement accuracy of all features in the images of $\sigma_{xy} = 0.25$ [pel]. Furthermore, all image measurements are corrected due to the nonlinear terms of the camera calibration. In average $N_X = 30$

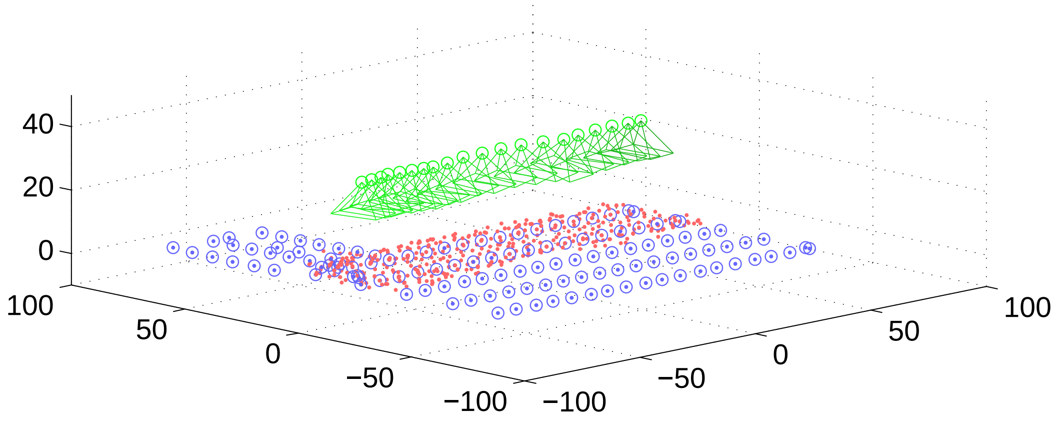


Figure 5.25: Reference trajectory of the camera orientation of the test image sequence. The shown camera trajectory consists of every 50th image of the sequence. In *red* all estimated object points and in *blue* all control points are visualized.

features are visible per image. The result of the bundle adjustment is shown in figure 5.25. The estimated square root of the variance factor is $\widehat{\sigma}_0 \approx 1.14$.

5.4.3 Kalman filter based results of a real image sequence

As we have shown in section 5.2 the best results of a Kalman filter based approach can be obtained by using the inverse distance initialization for new object points. Here we compare the results of the high accuracy bundle adjustment to the Kalman filter results using this initialization method. The filter is initialized introducing the control points of the first image with an accuracy of $\sigma = 1$ cm. We activate the proposed robustification method to reduce the effect of outliers. The initial camera orientation can be obtained by a spatial resection using the visible control points. The object points will be reparameterized from the inverse distance representation to the Euclidean representation using a roundness measurement threshold of $\Theta_L = 0.5$, c. f. section 4.3.4.3. This reduces the size of the state length significantly. As we have shown in figure 5.9 on page 117, this threshold can avoid the effect of the bias. Up to image number $n = 200$ initial control points are visible. Therefore, the resulting difference between the bundle adjustment and the Kalman filter based approach shown in figure 5.26 is small. Between image number $n = 220$ and 240 the camera orientation tilts with a huge angular acceleration and the image sequence shows blurring effects. Therefore, some feature tracks are lost and we observe a decrease of the accuracy in the resulting camera orientation. Between the image number $n = 390$ and 400 some tracked features are erroneous. This results

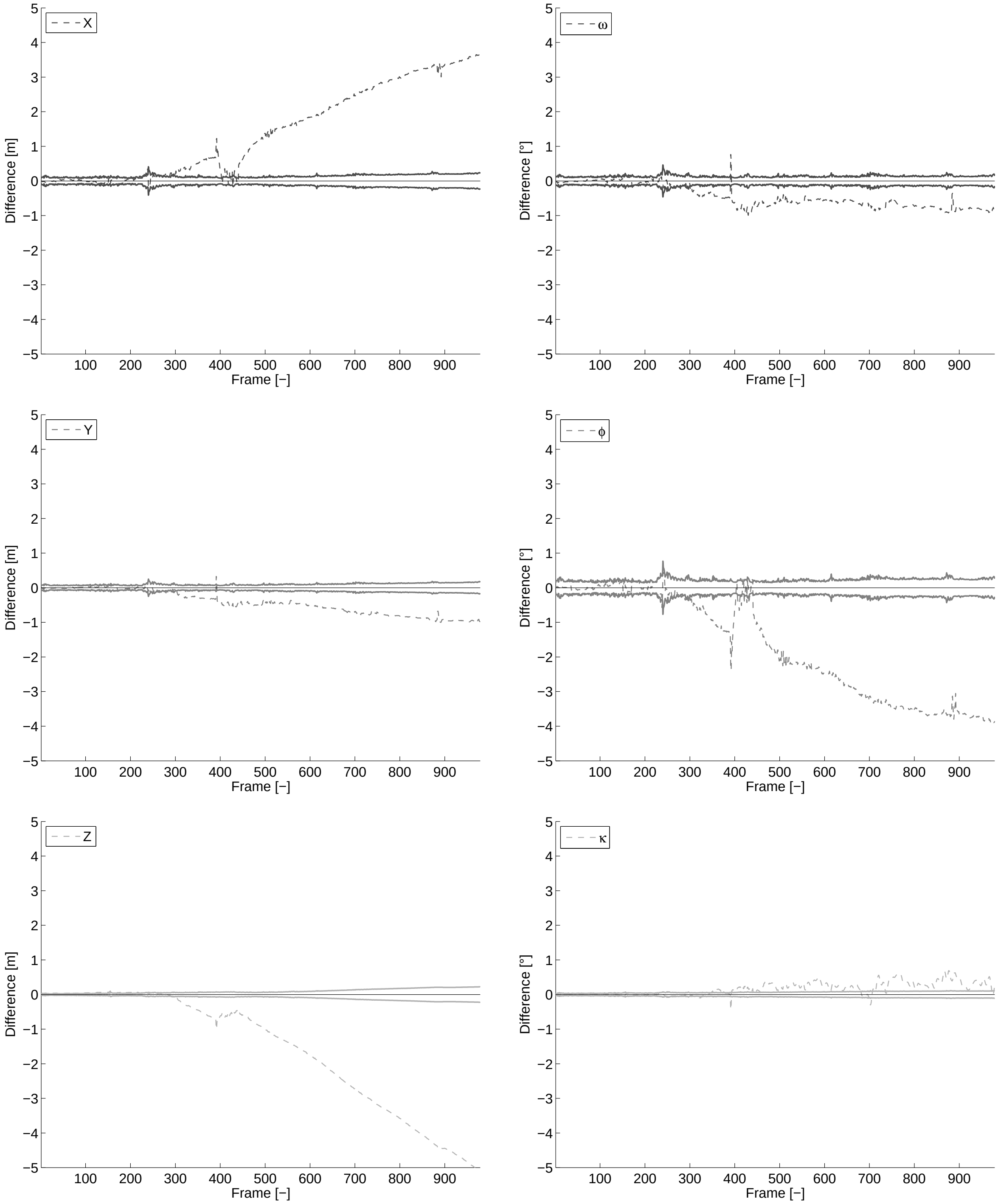


Figure 5.26: Difference between high accuracy bundle solution and Kalman filter based solution. Initialization using *inverse distance representation* with 30 m distance, reparameterization with a roundness test value of $\Theta_L = 0.5$. **Left:** Kalman filter result for the camera position parameters **Right:** Kalman filter result for the camera orientation parameters.

in an immediate change of the camera orientation. However, we can observe an apparently continuous drift in the estimated parameters. We suppose, this drift comes from the erroneous camera calibration and acts as a constant systematic error.

Discussion: In this section we presented our results for a real image sequence. In opposite to the experiments using a simulated environment additional influences on the results occur. First, we do not have ground truth data for the camera orientation. Instead, we used the results of a robust bundle adjustment incorporating high accuracy control points in all images. Nevertheless, the bundle adjustment is influenced by systematic errors of an imperfect camera calibration. We are not able to quantify this error. However, the bundle adjustment incorporate control point coordinates and the influence of the imperfect calibration can be assumed to be small. In the Kalman filter based approach, the information of the control points are neglected except for the initial images. We observed a continuous drift in the estimated camera orientation parameters. Second, gross errors in the feature tracks have a significant effect to the Kalman filter based estimate. Obviously, our robustification method is not able to reduce the effect of outliers completely. At present the algorithm does not have a prefiltering to identify gross errors in the feature tracking. This could be achieved by verifying the coplanarity constraint or the homography constraint[1] in consecutive images (c. f. section 3.2.2). Third, the real image sequence shows huge angular accelerations. Therefore, some feature tracks are lost in corresponding images. The associated object points recur in the following images. An advanced algorithm should reidentify these object points and continue the feature track. An alternative way is to stabilize the angular camera orientation by force to the nadir view by a hardware solution.

[1] The homography constraint can be used in case of small camera location chances or a planar object space.

Chapter 6

Conclusion and future work

In this thesis the theory and the applicability of a Kalman filter based framework has been presented that can be used to solve the task of simultaneous localization and mapping using image sequences only. It has been shown that the reconstruction of the trajectory of a single calibrated camera using point based features is feasible. Although, Kalman filter based solutions as an *online SLAM* approach need to consider additional issues as opposite to *full SLAM* approaches such as a bundle adjustment. These issues can be separated into two different aspects.

First, it has been shown that the continuous enlargement of the state requires an adequate initialization method for geometric entities. The initialization method has to be taken into account of the non-Gaussian distribution of the recovery of an object point coordinate for small disparities in consecutive frames. When neglecting this issue a continuous bias occurs. A comparison of the results using instant initialization methods and delayed initialization methods has been shown. It seems that the inverse distance parameterization of new object points is the best way to overcome the bias effect.

Second, compared to a bundle adjustment the detection of gross errors in the observations in a Kalman filter based method is more difficult. In case object points are non-stationary, their identification seems to be apparently impossible without having any access to previous observations of the object point.

In case of image sequences acquired by an unmanned aerial vehicle the approaches has a great potential shown by synthetic data. If gross errors can be excluded and systematic errors from imperfect camera calibration can be minimized the results are comparable with a bundle adjustment solution. The advantage compared to the bundle adjustment appears, when the number of object points is small and the number of images is large. If the application requires a state solution at any point of time the computational complexity can be significantly reduced using a Kalman filter based approach. However, in case of real image data Kalman filter based approaches are extremely sensitive to outliers.

This thesis consists of novel algorithms and methods close to the Kalman filter based simultaneous localization and mapping task. The first contribution of this work is a general derivation of the recursive update of the Kalman filter. This derivation is based on implicit measurement equations. Similar to the derivation of the least square methods based on the Gauss-Helmert-model (c. f. McGlone *et al.* (2004) p. 79ff) it has been shown that the classical iterative non-linear as well as the non-iterative and linear Kalman filter are specializations of our general Kalman filter. A significant improvement can be expected by using a Kalman filter based SLAM solution incorporating new geometric entities using this general framework.

Second, a new formulation of a linear prediction for the single camera state model and the sliding window camera state model has been derived. Here, the prediction of the state can be computed using matrix vector multiplications only. The angular orientation is defined by a quaternion representation, so that no singularities can occur. The method is limited to the case that the angular velocity is small.

The third contribution is a new method to initialize new object points in a delayed manner. The advantage of this method is that it is not necessary to keep previous camera orientations inside the state in contrast to other delayed initialization methods. However, it has been shown in the experiments that this method cannot maintain the consistency of the results.

Forth, a general law to assess the stochastical accuracy of the camera orientation of a photogrammetric strip has been derived. The law depends on four parameters to describe different kinds of flight configurations. The law can be considered as a generalization of the law introduced by Finsterwalder & Hofmann (1964) applied to image sequences. It has been shown that in image sequences with a small baseline to object distance ratio the law can predict the achievable accuracy of a photogrammetric strip up to an error of ±50%.

The final major contribution of this work is an analysis of the influence of different initialization methods for new object points to the Kalman filter based results. The effect of the bias is still a major challenge using the Kalman filter to solve the SLAM problem.

It has been shown that the analyzed methods have assets and drawbacks concerning computational complexity and consistency of the results.

Several main research paths in concatenation of this work are possible. In case the environment is large the Kalman filter seems to be sensitive to numerical inaccuracy of the covariance update of the state vector. Advanced methods such as a square root update or a covariance regularization satisfies numerical stability. Furthermore, it would be very important to integrate other geometric primitives and constraints between them. The practical problem to do this is that the observation models and constraints often yield in implicit equations. In my opinion the most promising approach to incorporate new geometric primitives for instance lines and planes and constraints between them would be to apply the novel *iterative implicit Kalman filter* proposed in this thesis. Also the integration of other sensor data such as GPS and INS is a very interesting topic. The main advantage is that the long term drift which occurs by systematic errors can be potentially compensated. Although, it must be studied how these observations have an effect on the inner accuracy of the reconstruction, especially on the problem of the continuous bias and on the numerical stability of the filter. Moreover, from a practical point of view the detection of erroneous observation, also known as the data association problem, would be very important. A major problem faced in this case is that without the access to past observations, which is usually desired to keep the memory usage small, the detection of non-stationary objects seems to be impossible. Using pre-filtering and rating strategies for the observations can possibly improve the robustness and accuracy of the presented framework. However, further studies are desired regarding to extended recursive update techniques, incorporate additional sensor data and robustification methods.

Appendix A

Appendix

A.1 Angular differentiation of circular motion using quaternion

First, lets assume an angle α is a function of time t depends on an initial angle α_0, an angular velocity $\dot{\alpha}$ and angular acceleration $\ddot{\alpha}$, so

$$\alpha_t = \alpha_0 + \dot{\alpha}t + \frac{1}{2}\ddot{\alpha}t^2. \tag{A.1}$$

Substitute α_t into the unit quaternion representation with angle-vector representation we get

$$\mathbf{q}(t) = \begin{bmatrix} \cos\alpha_t \\ \sin\alpha_t \boldsymbol{n} \end{bmatrix} = \begin{bmatrix} \cos\frac{\alpha_0+\dot{\alpha}t+\frac{1}{2}\ddot{\alpha}t^2}{2} \\ \sin\frac{\alpha_0+\dot{\alpha}t+\frac{1}{2}\ddot{\alpha}t^2}{2}\boldsymbol{n} \end{bmatrix} \tag{A.2}$$

as the quaternion as a function of time. Now lets approximate this quaternion by a Taylor expansion by

$$\mathbf{q}(t) = \mathbf{q}(t_0) + \frac{\partial \mathbf{q}_t}{\partial t}(t-t_0) + \frac{1}{2}\frac{\partial^2 \mathbf{q}_t}{\partial t}(t-t_0)^2 + ... \tag{A.3}$$

Evaluating equation (A.2) at $t_0 = 0$ and $\alpha_0 = 0$ we get the unit quaternion $\mathbf{q}(t_0) = [1,0,0,0]^\mathsf{T}$. To get the angular velocity we need to differentiate equation (A.2) with respect to time t, that

$$\dot{\mathbf{q}} = \frac{\partial \mathbf{q}}{\partial t} = \frac{\dot{\alpha}+\ddot{\alpha}t}{2}\begin{bmatrix} -\sin\frac{\alpha_0+\dot{\alpha}t+\frac{1}{2}\ddot{\alpha}t^2}{2} \\ \cos\frac{\alpha_0+\dot{\alpha}t+\frac{1}{2}\ddot{\alpha}t^2}{2}\boldsymbol{n} \end{bmatrix} \tag{A.4}$$

and for the angular acceleration

$$\ddot{\mathbf{q}} = \frac{\partial^2 \mathbf{q}}{\partial t} = \frac{\partial \dot{\mathbf{q}}}{\partial t} = \frac{\ddot{\alpha}}{2}\begin{bmatrix} -\sin\frac{\alpha_0+\dot{\alpha}t+\frac{1}{2}\ddot{\alpha}t^2}{2} \\ \cos\frac{\alpha_0+\dot{\alpha}t+\frac{1}{2}\ddot{\alpha}t^2}{2}\boldsymbol{n} \end{bmatrix} + \left(\frac{\dot{\alpha}+\ddot{\alpha}t}{2}\right)^2 \begin{bmatrix} -\cos\frac{\alpha_0+\dot{\alpha}t+\frac{1}{2}\ddot{\alpha}t^2}{2} \\ -\sin\frac{\alpha_0+\dot{\alpha}t+\frac{1}{2}\ddot{\alpha}t^2}{2}\boldsymbol{n} \end{bmatrix} \tag{A.5}$$

In case the angular velocity $\dot{\alpha}$ is small and $\alpha_0 = 0$ as well as $\ddot{\alpha} = 0$, then the quaternion velocity simplifies to

$$\dot{\mathbf{q}} \approx \begin{bmatrix} 0 \\ \frac{\dot{\alpha}}{2}\boldsymbol{n} \end{bmatrix} \tag{A.6}$$

In case the angular acceleration $\ddot{\alpha}$ is small and $\alpha_0 = 0$ as well $\dot{\alpha} = 0$, then the quaternion acceleration simplifies to

$$\begin{aligned} \ddot{\mathbf{q}} &\approx \frac{\ddot{\alpha}}{2} \begin{bmatrix} 0 \\ 1 \cdot \boldsymbol{n} \end{bmatrix} + \left(\frac{\ddot{\alpha}t}{2}\right)^2 \begin{bmatrix} -1 \\ 0 \cdot \boldsymbol{n} \end{bmatrix} && \text{(A.7)} \\ &\approx \begin{bmatrix} 0 \\ \frac{\ddot{\alpha}}{2}\boldsymbol{n} \end{bmatrix} && \text{(A.8)} \end{aligned}$$

Now we evaluate equation (A.3) up to the second order and we get

$$\mathbf{q}(t) \approx \mathbf{q}(t_0) + \dot{\mathbf{q}}t + \frac{1}{2}\ddot{\mathbf{q}}t^2 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{\dot{\alpha}}{2}\boldsymbol{n} \end{bmatrix} t + \frac{1}{2} \begin{bmatrix} 0 \\ \frac{\ddot{\alpha}}{2}\boldsymbol{n} \end{bmatrix} t^2. \tag{A.9}$$

From equation (A.9) we can see that it is suitable to representing a small angular velocity by the vector part only and equivalently a small angular acceleration. Furthermore, a perturbation of a small angular velocity by a small angular acceleration can be expressed by

$$\dot{\boldsymbol{q}} = \dot{\boldsymbol{q}}_0 + \ddot{\boldsymbol{q}}t = \frac{\dot{\alpha}_0}{2}\boldsymbol{n}_0 + \frac{\ddot{\alpha}}{2}\boldsymbol{n}t \tag{A.10}$$

A rotation represented by a quaternion depending on the time t and only on an angular velocity can be expressed by

$$\boldsymbol{q}(\dot{\boldsymbol{q}}, t) = \begin{bmatrix} 1 \\ \dot{\boldsymbol{q}}t \end{bmatrix} \tag{A.11}$$

equivalently for an angular acceleration

$$\boldsymbol{q}(\ddot{\boldsymbol{q}}, t) = \begin{bmatrix} 1 \\ \frac{1}{2}\ddot{\boldsymbol{q}}t^2 \end{bmatrix} \tag{A.12}$$

A.2 Linear dynamic model derivation

Let us assume that we have a uniformly accelerated linear motion in 3d space in the time interval Δt. Following classical physics this motion can be described by

$$\boldsymbol{T}_{t+1} = \boldsymbol{T}_t + \dot{\boldsymbol{T}}_t \Delta t + \frac{1}{2}\ddot{\boldsymbol{T}}_t \Delta t^2 \tag{A.13}$$

If we differentiate with respect to Δt we get the velocity

$$\dot{\boldsymbol{T}}_{t+1} = \dot{\boldsymbol{T}}_t + \ddot{\boldsymbol{T}}_t \Delta t \tag{A.14}$$

The other partial derivatives of equation (A.13) with respect to $\boldsymbol{T}_t, \dot{\boldsymbol{T}}_t$ and $\ddot{\boldsymbol{T}}_t$ are given by

$$\frac{\partial \boldsymbol{T}_{t+1}}{\partial \boldsymbol{T}_t} = I_3 \tag{A.15}$$

$$\frac{\partial \boldsymbol{T}_{t+1}}{\partial \dot{\boldsymbol{T}}_t} = \Delta t I_3 \tag{A.16}$$

$$\frac{\partial \boldsymbol{T}_{t+1}}{\partial \ddot{\boldsymbol{T}}_t} = \frac{1}{2}\Delta t^2 I_3 \tag{A.17}$$

The partial derivatives of equation (A.14) with respect to $\dot{\boldsymbol{T}}_t$ and $\ddot{\boldsymbol{T}}_t$ are given by

$$\frac{\partial \dot{\boldsymbol{T}}_{t+1}}{\partial \dot{\boldsymbol{T}}_t} = I_3 \tag{A.18}$$

$$\frac{\partial \dot{\boldsymbol{T}}_{t+1}}{\partial \ddot{\boldsymbol{T}}_t} = \Delta t I_3 \tag{A.19}$$

In some cases it is useful to replace the velocity by $\dot{\boldsymbol{T}}_t = \boldsymbol{T}_t - \boldsymbol{T}_{t-1}$ and equation (A.13) can be rewritten as

$$\boldsymbol{T}_{t+1} = \boldsymbol{T}_t + (\boldsymbol{T}_t - \boldsymbol{T}_{t-1})\Delta t + \frac{1}{2}\ddot{\boldsymbol{T}}_t \Delta t^2 \tag{A.20}$$

and therefore the partial derivatives with respect to $\boldsymbol{T}_t$ and $\boldsymbol{T}_{t-1}$ are given by

$$\frac{\partial \boldsymbol{T}_{t+1}}{\partial \boldsymbol{T}_t} = I_3 + \Delta t I_3 \tag{A.21}$$

$$\frac{\partial \boldsymbol{T}_{t+1}}{\partial \boldsymbol{T}_{t-1}} = -\Delta t I_3. \tag{A.22}$$

In case of uniformly accelerated linear circular motion in 3d space during the time interval Δt we can compute the partial derivatives in the same manner. To represent the rotation we will use quaternions. Similar to the linear motion model in equation (A.13) we can compute the linear circular motion as

$$\mathbf{q}_{t+1} = \mathbf{q}(\ddot{\boldsymbol{q}}_t)\mathbf{q}(\dot{\boldsymbol{q}}_t)\mathbf{q}_t. \tag{A.23}$$

We will assume that the angular velocity $\dot{\boldsymbol{q}}_t$ and angular acceleration $\ddot{\boldsymbol{q}}_t$ are small, so that it is sufficient to use the vector part only. in this case equation (3.36) can be used to concatenate two rotations. The rotation around a fixed axis in the time interval Δt given the angular velocity and angular acceleration can be computed using quaternion representation simply as

$$\mathbf{q}(\dot{\boldsymbol{q}}) = \begin{bmatrix} 1 \\ \dot{\boldsymbol{q}}_t \Delta t \end{bmatrix} \tag{A.24}$$

$$\mathbf{q}(\ddot{\boldsymbol{q}}) = \begin{bmatrix} 1 \\ \frac{1}{2}\ddot{\boldsymbol{q}}_t \Delta t^2 \end{bmatrix}. \tag{A.25}$$

Now using equation (3.31) the linear circular motion in equation (A.23) can be written as

$$\mathbf{q}_{t+1} = \Upsilon(\mathbf{q}(\ddot{\boldsymbol{q}}))\Upsilon(\mathbf{q}(\dot{\boldsymbol{q}}))\mathbf{q}_t. \tag{A.26}$$

Also note that using equation (3.31) this can be conveniently rewritten to extract the desired vector on the right hand side as follows

$$\mathbf{q}_{t+1} = \Upsilon(\frac{1}{2}\ddot{\boldsymbol{q}}_t \Delta t^2)\Upsilon(\dot{\boldsymbol{q}}_t \Delta t)\mathbf{q}_t \tag{A.27}$$

$$= \Upsilon(\frac{1}{2}\ddot{\boldsymbol{q}}_t \Delta t^2)\breve{\Upsilon}(\mathbf{q}_t)\mathbf{q}(\dot{\boldsymbol{q}}_t \Delta t) \tag{A.28}$$

$$= \breve{\Upsilon}(\Upsilon(\dot{\boldsymbol{q}}_t \Delta t)\mathbf{q}_t)\mathbf{q}(\frac{1}{2}\ddot{\boldsymbol{q}}_t \Delta t^2). \tag{A.29}$$

The partial derivatives of equation (A.29) w.r.t. $\mathbf{q}_t, \dot{\boldsymbol{q}}_t$ and $\ddot{\boldsymbol{q}}_t$ are therefore easily obtained as

$$\frac{\partial \mathbf{q}_{t+1}}{\partial \mathbf{q}_t} = \Upsilon(\frac{1}{2}\ddot{\boldsymbol{q}}_t \Delta t^2)\Upsilon(\dot{\boldsymbol{q}}_t \Delta t) \tag{A.30}$$

$$\frac{\partial \boldsymbol{q}_{t+1}}{\partial \dot{\boldsymbol{q}}_t} = \Upsilon(\frac{1}{2}\ddot{\boldsymbol{q}}_t \Delta t^2)\breve{\Upsilon}(\mathbf{q}_t)\left[\mathbf{0}_3 | \Delta t I_3\right] \tag{A.31}$$

$$\frac{\partial \boldsymbol{q}_{t+1}}{\partial \ddot{\boldsymbol{q}}_t} = \breve{\Upsilon}(\Upsilon(\dot{\boldsymbol{q}}_t \Delta t)\mathbf{q}_t)\left[\mathbf{0}_3 | \frac{1}{2}\Delta t^2 \ddot{\boldsymbol{q}}_t\right]. \tag{A.32}$$

Now we will assume a series of rotations with constant time differences Δt is given. The rotational difference between two time steps can be computed as

$$\Delta\mathbf{q} = \mathbf{q}_t \mathbf{q}_{t-1}^{-1}. \tag{A.33}$$

Assuming a linear circular motion, the rotation at time $t+1$ can be obtained according to equation (A.23) by

$$\begin{align}
\mathbf{q}_{t+1} &= \mathbf{q}(\frac{1}{2}\ddot{\boldsymbol{q}}_t \Delta t^2)\Delta\mathbf{q}\mathbf{q}_t \tag{A.34}\\
&= \mathbf{q}(\frac{1}{2}\ddot{\boldsymbol{q}}_t \Delta t^2)\mathbf{q}_t\mathbf{q}_{t-1}^{-1}\mathbf{q}_t \tag{A.35}\\
&= \Upsilon(\frac{1}{2}\ddot{\boldsymbol{q}}_t \Delta t^2)\Upsilon(\mathbf{q}_t)\Upsilon(\mathbf{q}_{t-1}^{-1})\mathbf{q}_t \tag{A.36}\\
&= \Upsilon(\frac{1}{2}\ddot{\boldsymbol{q}}_t \Delta t^2)\Upsilon(\mathbf{q}_t)\breve{\Upsilon}(\mathbf{q}_t)V\mathbf{q}_{t-1} \tag{A.37}\\
&= \Upsilon(\frac{1}{2}\ddot{\boldsymbol{q}}_t \Delta t^2)\breve{\Upsilon}(\breve{\Upsilon}(\mathbf{q}_t)V\mathbf{q}_{t-1})\mathbf{q}_t \tag{A.38}\\
&= \breve{\Upsilon}(\Upsilon(\mathbf{q}_t)\Upsilon(\mathbf{q}_{t-1}^{-1})\mathbf{q}_t)\mathbf{q}(\frac{1}{2}\ddot{\boldsymbol{q}}_t \Delta t^2) \tag{A.39}
\end{align}$$

The partial derivatives of $\mathbf{q}_{t+1}$ w.r.t. $\mathbf{q}_{t-1}, \ddot{\boldsymbol{q}}$ and $\mathbf{q}_t$ using the chain rule are then given by

$$\begin{align}
\frac{\partial\mathbf{q}_{t+1}}{\partial\mathbf{q}_{t-1}} &= \Upsilon(\frac{1}{2}\ddot{\boldsymbol{q}}_t \Delta t^2)\Upsilon(\mathbf{q}_t)\breve{\Upsilon}(\mathbf{q}_t)V \tag{A.40}\\
\frac{\partial\mathbf{q}_{t+1}}{\partial\ddot{\boldsymbol{q}}_t} &= \breve{\Upsilon}(\Upsilon(\mathbf{q}_t)\Upsilon(\mathbf{q}_{t-1}^{-1})\mathbf{q}_t)[\mathbf{0}|\frac{1}{2}\Delta t^2 I_3] \tag{A.41}\\
\frac{\partial\mathbf{q}_{t+1}}{\partial\mathbf{q}_t} &= \Upsilon(\frac{1}{2}\ddot{\boldsymbol{q}}_t \Delta t^2)(\Upsilon(\mathbf{q}_t)\Upsilon(V\mathbf{q}_{t-1}) + \breve{\Upsilon}(\breve{\Upsilon}(\mathbf{q}_t)V\mathbf{q}_{t-1})) \tag{A.42}
\end{align}$$

In some cases the time difference between consecutive images changes. In this case we have to deal with the time dependent factor denoted as Δt here. It is given by

$$\Delta t = \frac{t_{t+1} - t_t}{t_t - t_{t-1}}. \tag{A.43}$$

In case the rotation $\Delta\mathbf{q}$ is small we can use the concatenation approximation

$$\Delta\mathbf{q}_{\Delta t} = U_{\Delta t}(\mathbf{q}_t\mathbf{q}_{t-1}^{-1}) = U_{\Delta t}\Upsilon(\mathbf{q}_t)V\mathbf{q}_{t-1} \tag{A.44}$$

Now using this approximation the new rotation at time $t+1$ can be computed as

$$\begin{align}
\mathbf{q}_{t+1} &= \Upsilon(\frac{1}{2}\ddot{\boldsymbol{q}}_t \Delta t^2)\Upsilon(U_{\Delta t}\Upsilon(\mathbf{q}_t)V\mathbf{q}_{t-1})\mathbf{q}_t \tag{A.45}\\
&= \Upsilon(\frac{1}{2}\ddot{\boldsymbol{q}}_t \Delta t^2)\breve{\Upsilon}(\mathbf{q}_t)U_{\Delta t}\Upsilon(\mathbf{q}_t)V\mathbf{q}_{t-1} \tag{A.46}\\
&= \Upsilon(\frac{1}{2}\ddot{\boldsymbol{q}}_t \Delta t^2)\breve{\Upsilon}(\mathbf{q}_t)U_{\Delta t}\breve{\Upsilon}(V\mathbf{q}_{t-1})\mathbf{q}_t \tag{A.47}\\
&= \breve{\Upsilon}(\Upsilon(U_{\Delta t}\Upsilon(\mathbf{q}_t)V\mathbf{q}_{t-1})\mathbf{q}_t)\mathbf{q}(\frac{1}{2}\ddot{\boldsymbol{q}}_t \Delta t^2) \tag{A.48}
\end{align}$$

The derivative w.r.t $\mathbf{q}_{t-1}$ and $\ddot{\boldsymbol{q}}_t$ is therefore given by

$$\begin{align}
\frac{\partial\mathbf{q}_{t+1}}{\partial\boldsymbol{q}_{t-1}} &= \Upsilon(\frac{1}{2}\ddot{\boldsymbol{q}}_t \Delta t^2)\breve{\Upsilon}(\mathbf{q}_t)U_{\Delta t}\Upsilon(\mathbf{q}_t)V \tag{A.49}\\
\frac{\partial\mathbf{q}_{t+1}}{\partial\ddot{\boldsymbol{q}}_t} &= \breve{\Upsilon}(\Upsilon(U_{\Delta t}\Upsilon(\mathbf{q}_t)V\mathbf{q}_{t-1})\mathbf{q}_t)[\mathbf{0}_3|\frac{1}{2}\Delta t^2 I_3]. \tag{A.50}
\end{align}$$

The derivative w.r.t. $\mathbf{q}_t$ using the chain rule is given by

$$\frac{\partial \mathbf{q}_{t+1}}{\partial \boldsymbol{q}_t} = \Upsilon(\frac{1}{2}\ddot{\boldsymbol{q}}_t \Delta t^2)(\Upsilon(\mathsf{U}_{\Delta t}\Upsilon(\mathbf{q}_t)\mathsf{V}\mathbf{q}_{t-1}) + \breve{\Upsilon}(\mathbf{q}_t)\mathsf{U}_{\Delta t}\breve{\Upsilon}(\mathsf{V}\mathbf{q}_{t-1})). \tag{A.51}$$

A.3 Uncertainty transfer

Lets assume we have a reference point $\boldsymbol{X}_r$ and a new point X_n close to each other. The uncertainty of a Euclidean coordinate derived from camera observations can be expressed as a sum of two independent parts

$$\boldsymbol{d}_{X_r} = \boldsymbol{d}^v_{X_r} + \boldsymbol{d}^c_{X_r} \tag{A.52}$$

$$\boldsymbol{d}_{X_n} = \boldsymbol{d}^v_{X_n} + \boldsymbol{d}^c_{X_n} \tag{A.53}$$

with v as the uncertainty part of the intersection and c as the uncertainty part of the camera orientation. Furthermore we can assume, that the uncertainty of the new point $\boldsymbol{X}_n$ can be expressed as the uncertainty of the reference point adding an unknown additional relative uncertainty.

$$\boldsymbol{d}_{X_n} = \boldsymbol{d}^v_{X_r} + \boldsymbol{d}\Delta X \tag{A.54}$$

This $d\Delta X$ can be expressed using the difference of equation (A.52) and (A.53)

$$\boldsymbol{d}_{X_n} = \boldsymbol{d}_{X_r} + (\boldsymbol{d}^v_{X_n} + \boldsymbol{d}^c_{X_n} - (\boldsymbol{d}^v_{X_r} + \boldsymbol{d}^c_{X_r})) \tag{A.55}$$

Now, lets assume the influence part of the orientation for both points are equal, which is a sufficient assumption. In this case equation (A.55) simplifies to

$$\boldsymbol{d}_{X_n} = \boldsymbol{d}_{X_r} + \boldsymbol{d}^v_{X_n} - \boldsymbol{d}^v_{X_r} \tag{A.56}$$

Now lets rewrite in matrix expression

$$\begin{bmatrix} \boldsymbol{d}_{X_r} \\ \boldsymbol{d}\Delta X \end{bmatrix} = \underbrace{\begin{pmatrix} \mathsf{I} & \mathsf{I} & \mathsf{0} \\ -\mathsf{I} & \mathsf{0} & \mathsf{I} \end{pmatrix}}_{\mathsf{A}} \begin{bmatrix} \boldsymbol{d}^v_{X_r} \\ \boldsymbol{d}^c_{X_r} \\ \boldsymbol{d}^v_{X_n} \end{bmatrix} \tag{A.57}$$

and apply the transformation to a Gaussian uncertainty representation which leads to

$$\mathsf{C}_{X_r \Delta \boldsymbol{X}} = \mathsf{A} \begin{pmatrix} \mathsf{C}^v_{X_r} & \mathsf{0} & \mathsf{0} \\ \mathsf{0} & \mathsf{C}^c_{X_r} & \mathsf{0} \\ \mathsf{0} & \mathsf{0} & \mathsf{C}^v_{X_n} \end{pmatrix} \mathsf{A}^\mathsf{T} = \begin{pmatrix} \mathsf{C}^v_{X_r} + \mathsf{C}^c_{X_r} & -\mathsf{C}^v_{X_n} \\ -\mathsf{C}^v_{X_n} & 2\mathsf{C}^v_{X_n} \end{pmatrix} = \begin{pmatrix} \mathsf{C}_{X_r} & -\mathsf{C}^v_{X_n} \\ -\mathsf{C}^v_{X_n} & 2\mathsf{C}^v_{X_n} \end{pmatrix} \tag{A.58}$$

Note, that $\mathsf{C}^v_{X_n}$ should be smaller than C_{X_r} to guarantee positive semi-definiteness.

A.4 Accuracy analysis of the general law

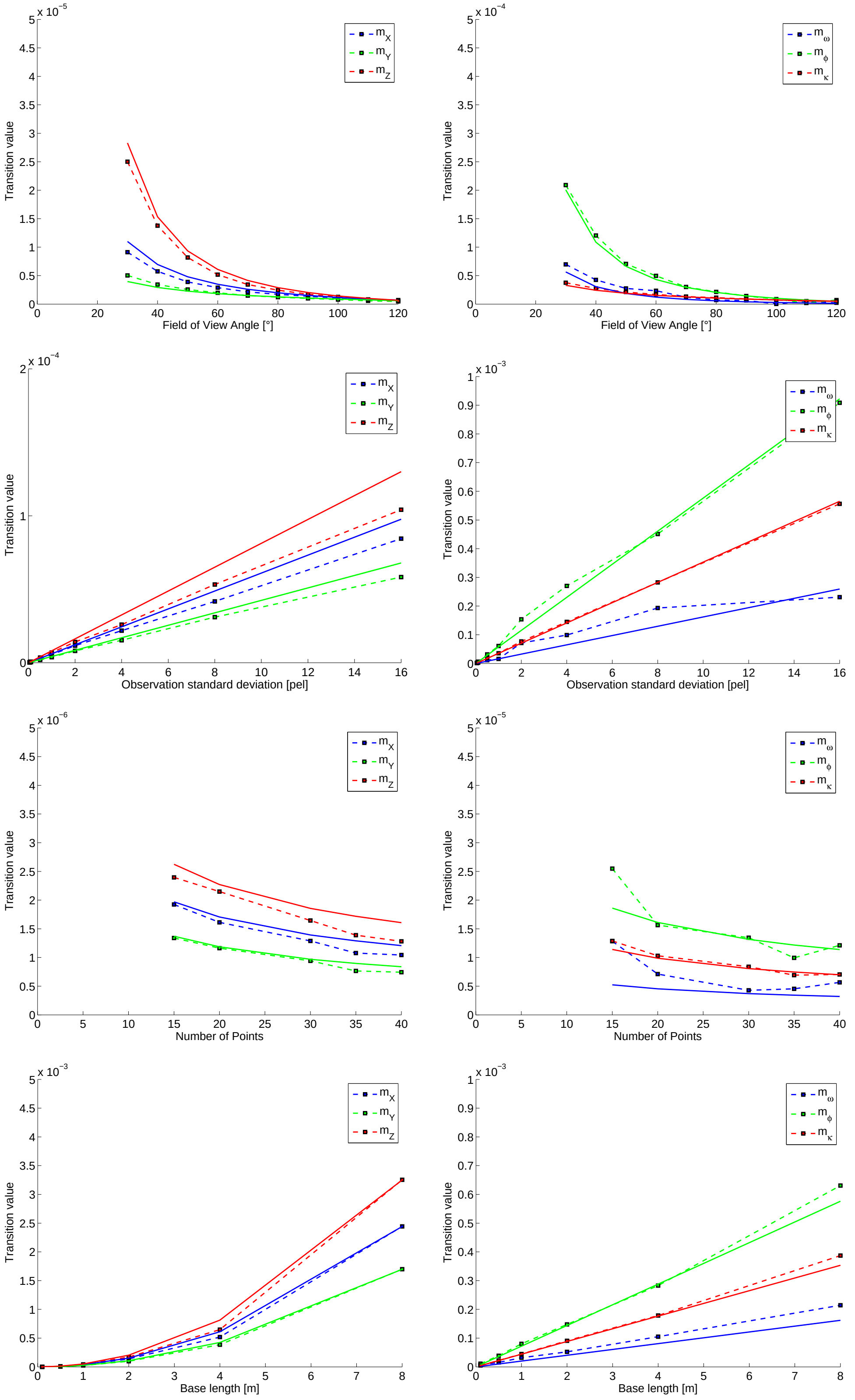


Figure A.1: Approximation error of the four simulated experiments.

	Negative approx. error						Positive approx. error					
Exp	σ_X	σ_Y	σ_Z	σ_ω	σ_ϕ	σ_κ	σ_X	σ_Y	σ_Z	σ_ω	σ_ϕ	σ_κ
Fov°												
30	39	27	10	22	11	14	33	21	25	22	41	36
40	40	28	9	25	16	14	31	14	40	17	37	51
50	30	26	11	30	15	14	29	22	34	27	40	45
60	36	34	10	41	24	12	27	1	39	1	9	28
70	26	20	4	25	22	9	24	14	38	17	30	35
80	23	11	6	23	28	13	32	25	33	31	44	29
90	16	11	7	24	25	8	26	19	28	21	38	24
100	12	14	12	15	17	8	25	19	20	21	31	20
110	8	5	7	3	12	5	23	25	29	34	39	28
120	26	18	9	28	36	11	7	21	24	20	14	17
σ_{xy}[**pel**]												
0.05	14	12	1	23	15	6	31	20	42	25	46	36
0.1	24	22	7	29	27	10	16	13	21	16	23	23
0.5	20	11	8	22	23	9	22	23	26	23	28	21
1	23	17	9	20	23	7	13	20	25	23	17	20
2	43	24	7	34	44	12	8	11	26	11	11	16
4	34	25	22	27	38	17	15	15	33	12	7	19
8	21	18	9	20	22	7	23	16	34	14	29	27
16	10	10	3	16	17	8	22	26	41	26	35	32
N_X[-]												
10	62	83	58	87	64	34	25	1	36	3	35	21
15	30	32	29	40	39	20	9	15	24	15	16	18
20	27	25	7	33	32	11	28	16	16	18	40	19
30	16	12	6	20	19	9	26	21	18	28	34	21
35	18	5	4	7	18	4	25	23	31	33	33	28
40	18	18	7	24	18	8	24	16	37	20	33	26
B[m]												
0.1	37	23	16	31	40	14	9	11	29	6	4	12
0.5	16	9	8	28	16	8	23	21	29	27	38	41
1	34	29	14	33	24	10	14	15	45	4	21	49
2	10	7	14	26	8	10	27	23	55	24	55	98
4	23	19	16	21	19	9	24	28	72	30	64	140
8	23	14	28	33	23	16	31	33	168	36	58	185
min‖max	62	83	58	87	64	34	33	33	168	36	64	185

Table A.1: Approximation error in percent.

A.5 Distortion field of the reconstructed object coordinates in the simulated environment

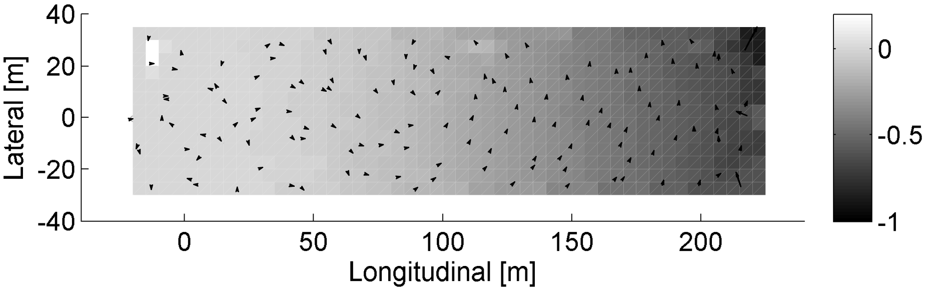


Figure A.2: Distortion field visualization for the initialization using *Euclidean representation*. Color coded distortion of the point field in height direction, arrow field distortion for lateral and longitudinal component

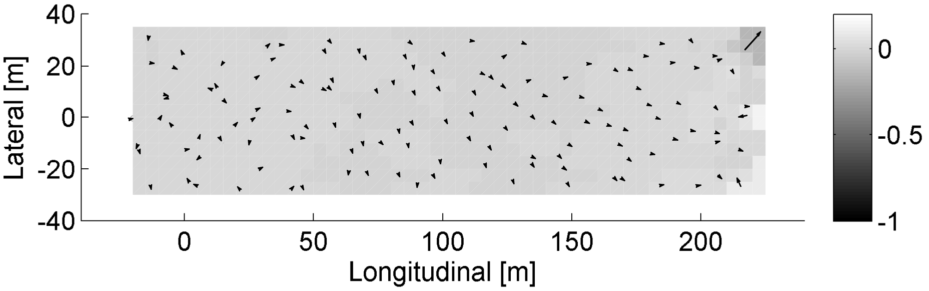


Figure A.3: Distortion field visualization for the initialization using *inverse distance representation*. Color coded distortion of the point field in height direction, arrow field distortion for lateral and longitudinal component

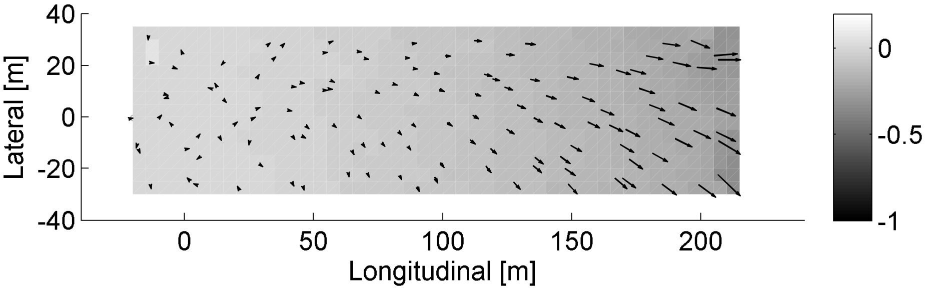


Figure A.4: Distortion field visualization for the initialization using *stable initialization procedure*. Color coded distortion of the point field in height direction, arrow field distortion for lateral and longitudinal component

List of Algorithms

List of Tables

List of Figures

Bibliography

Abraham, Steffen. 1999. *Kamera Kalibrierung und metrische Auswertung monokularer Bildfolgen.* Ph.D. thesis, Institute of Photogrammetry, University of Bonn.

Ackermann, Friedrich. 1965. *Fehlertheoretische Untersuchungen über die Genauigkeit photogrammetrischer Streifentriangulationen.* Ph.D. thesis, Deutsche Geodätische Kommission, Reihe C, Nr. 87, München.

Akbarzadeh, A., Frahm, J.-M., Mordohai, P., Clipp, B., Engels, C., Gallup, D., Merrell, P., Phelps, M., Sinha, S., Talton, B., Wang, L., Yang, Q., Stewéius, H., Yang, R., Welch, G., Towles, H., Nistér, D.,Pollefeys, M. 2006. Towards urban 3d reconstruction from video. *In: Third International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT).*

Baarda, Willem. 1967. *Statistical Concepts in Geodesy.* New Series, vol. 2. Delft: Netherlands Geodetic Commission.

Baarda, Willem. 1973. S-Transformations and Criterion Matrices. *Netherlands geodetic Commission*, **volume 5 of 1**.

Bailey, Tim, Nieto, Juan, Guivant, Jose, Stevens, Michael,Nebot, Eduardo. 2006. Consistency of the EKF-SLAM Algorithm. *In: International Conference on Intelligent Robots and Systems.*

Beder, Christian,Steffen, Richard. 2006. Determining an initial image pair for fixing the scale of a 3d reconstruction from an image sequence. *Pages 657–666 of:* Franke, K., Müller, K.-R., Nickolay, B.,Schäfer, R. (eds), *Pattern Recognition.* LNCS, no. 4174. Springer.

Beder, Christian,Steffen, Richard. 2008. Incremental estimation without specifying a-priori covariance matrices for the novel parameters. *VLMP Workshop on CVPR. Anchorage, USA.*

Benítez, Fernando Caballero, Cabañas, Luis Merino, Melero, Joaquin Ferruz,Baturone, Anibal Ollero. 2005. A Visual Odometer Without 3d Reconstruction for Aerial Vehicles. Applications to Building Inspection. *Pages 4684–4689 of: IEEE International Conference on Robotics and Automation.*

Besl, Paul J.,D.McKay, Neil. 1992. A Method for Registration of 3-D Shapes. *IEEE Trans. Pattern Anal. Mach. Intell.*, **14**(2), 239–256.

Bibby, Charles,Reid, Ian. 2007. Simultaneous Localisation and Mapping in Dynamic Environments (SLAMIDE) with Reversible Data Association. *Robotics: Sience and Systems.*

Blanco, Jose-Luis, Fernandez-Madrigal, Juan-Antonio,Gonzalez, Javier. 2007. A New Approach for Large-Scale Localization and Mapping: Hybrid Metric-Topological SLAM. *Pages 2061–2067 of: ICRA.*

Brennecke, Christian, Wolf, Oliver,Wagner, Bernardo. 2003. Using 3D Laser Range Data for SLAM in Outdoor Environments. *In: IEEE/RSJ International Conference on Intelligent Robots and Systems.*

Caspary, W., Wang J. 1998. Redundanzanteile und Varianzkomponenten im Kalman Filter. *ZfV 123*, 121–128.

Castellanos, Jose A., Neira, Jose,Tardos, Juan D. 2004. Limits to the Consistency of EKF-based SLAM. *5th symposium on Intelligent Autonomous Vehicles*, 21–26.

Chiba, Naoki,Knade, Takeo. 1998. A Tracker for Broken and Closely-Spaced Lines. *ISPRS International Archives of Photogrammetry and Remote Sensing*, **XXXII**, 676–683.

Choset, Howie,Nagatani, K. 2001. Topological simultaneous localization and mapping (SLAM): toward exact localization without explicit localization. *IEEE Transactions on Robotics and Automation*, **17**(1), 125 – 137.

Civera, Javier, Davison, Andrew J.,Montiel, J. M. M. 2007a. Dimensionless Monocular SLAM. *In: IBPRIA.*

Civera, Javier, Davison, Andrew J.,Montiel, J. M. M. 2007b. Inverse Depth to Depth Conversion for Monocular SLAM. *In: ICRA.*

Clemente, Laura, Davison, Andrew, Reid, Ian, Neira, José,Tardós, Juan Domingo. 2007 (June). Mapping Large Loops with a Single Hand-Held Camera. *In: Proc. Robotics: Science and Systems Conference.*

Davison, Andrew J. 2003. Real-Time Simultaneous Localisation and Mapping with a Single Camera. *In: Proc. International Conference on Computer Vision, Nice.*

Davison, Andrew J.,Kita, Nobuyuki. 2001. 3D Simultaneous Localisation and Map-Building Using Active Vision for a Robot Moving on Undulating Terrain. *In: CVPR.*

Davison, Andrew J.,Murray, David W. 2002. Simultaneous Localisation and Map-Building using Active Vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **24**(7), 865–880.

Dellaert, Frank. 2005 (June). Square Root SAM. *In: Proceedings of Robotics: Science and Systems.*

Dickscheid, Timo, Läbe, Thomas,Förstner, Wolfgang. 2008. Benchmarking Automatic Bundle Adjustment Results. *In: To appear in: 21st Congress of the International Society for Photogrammetry and Remote Sensing (ISPRS).*

Dissanayake, M. W. M. Gamini, Newman, Paul, Clark, Steven, Durrant-whyte, Hugh F.,Csorba, M. 2001. A solution to the simultaneous localization and map building (SLAM) problem. *IEEE Transactions on Robotics and Automation*, **17**, 229–241.

Eliazar, Austin I.,Parr, Ronald. 2003. DP-SLAM: Fast, Robust Simultaneous Localization and Mapping Without Predetermined Landmarks. *Pages 1135–1142 of: IJCAI.*

Eliazar, Austin I.,Parr, Ronald. 2004. DP-SLAM 2.0. *Pages 1314–1320 of: ICRA.*

Estrada, Carlos, Neira, Jose,Tardos, Juan D. 2005. Hierarchical SLAM: real-time accurate mapping of large environments. *IEEE Transactions on Robotics*, 588–596.

Everaerts, J., Lewyckyj, N.,Fransaer, D. 2004. PEGASUS: Design of a Stratospheric Long Endurance UAV System for Remote Sensing. *In: The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XXXV Part B2.

Fairfield, Nathaniel, Kantor, George A.,Wettergreen, David. 2006 (May). Towards Particle Filter SLAM with Three Dimensional Evidence Grids in a Flooded Subterranean Environment. *Pages 3575 – 3580 of: Proceedings of ICRA 2006.*

Faugeras, Olivier, Robert, Luc, Laveau, Stéphane, Csurka, Gabriella, Zeller, Cyril, Gauclin, Cyrille,Zoghlami, Imed. 1998. 3-D Reconstruction of Urban Scenes from Image Sequences. *Pages 229–309 of: Computer Vision and Image Understanding*, vol. 69-3.

Finsterwalder, Richard,Hofmann, Walter. 1964. *Photogrammetrie.* Walter de Gruyter, Berlin.

Fischler, M. A.,Bolles, R. C. 1981. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Pages 381–395 of: Comm. of the ACM*, vol. Vol 24.

Folkesson, John,Christensen, Hendrik. 2004. Graphical slam - a self-correction map. *In IEEE International Conference on Robotics and Automation.*

Förstner, Wolfgang. 1987. Reliability Analysis of Parameter Estimation in Linear Models with Applications to Mensuration Problems in Computer Vision. *Pages 273–310 of: CVGIP - Computer Vision, Graphics and Image Processing.*

Förstner, Wolfgang,Moonen, Boudewijn. 1999. A Metric for Covariance Matrices. *In: In: Krumm, F. / Schwarze, V. S. (Hg.): Quo vadis geodesia ...?, Festschrift for Erik W. Grafarend on the occasion of his 60th birthday. TR Dept. of Geodesy and Geoinformatics, Stuttgart University.*

Förstner, Wolfgang,Wrobel, Bernhard. 2004. Mathematical Concepts in Photogrammetry. *Pages 15–180 of:* J.C.McGlone, E.M.Mikhail,J.Bethel (eds), *Manual of Photogrammetry.* ASPRS.

Förstner, Wolfgang. 1982. On the Geometric Precision of Digital Correlation. *Pages 176–189 of: Proceedings of the ISPRS Symposium Mathematical Models, Accuray Aspects and Quality Control.*

Förstner, Wolfgang,Gülch, Eberhard. 1987. A Fast Operator for Detection and Precise Location of Distict Point, Corners and Centres of Circular Features. *Pages 281–305 of: Proceedings of the ISPRS Conference on Fast Processing of Photogrammetric Data.*

Grisetti, Giorgio, Stachniss, Cyrill,Burgard, Wolfram. 2006. Improved Techniques for Grid Mapping with Rao-Blackwellized Particle Filters. *IEEE Transactions on Robotics.*

Grün, Armin. 1982. An Optimum Algorithm for On-Line Triangulation. *Pages 131–151 of: International Society for Photogrammetry and Remote Sensing, Volume 24 - III/2.*

Guivant, José E., Nebot, Eduardo Mario, Nieto, Juan,Masson, Favio R. 2004. Navigation and Mapping in Large Unstructured Environments. *I. J. Robotic Res.*, **23**(4-5), 449–472.

Hampel, Frank R., Ronchetti, Elvezio M., Rousseeuw, Peter J.,Stahel, Werner A. 1986. *Robust Statistics - The Approach Based on Influence Functions.* Wiley.

Haralick, R. M., Lee, C., Ottenberg, K.,Nölle, M. 1991. Analysis and Solutions of the Three Point Perspective Pose Estimation Problem. *Pages 592–598 of: Proc. of the IEEE Comp. Society Conf. on "Computer Vision and Pattern Recognition".*

Harati, A., Gächter, S.,Siegwart, R. 2007. Fast Range Image Segmentation for Indoor 3D-SLAM. *In: Proceedings of the Sixth IFAC Symposium on Intelligent Autonomous Vehicles (IAV-6).*

Harris, Chris,Stephens, Mike. 1988. A Combined Corner and Edge Detector. *Pages 147–151 of: The Fourth Alvey Vision Conference.*

Hartley, Richard I.,Zisserman, Andrew. 2000. *Multiple View Geometry in Computer Vision.* Cambridge University Press, ISBN: 0521623049.

Herbert Bay, Vittorio Ferraris,Gool, Luc Van. 2005. Wide-baseline stereo matching with line segments. *Pages 329–336 vol. 1 of: Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1.

Heuel, Stephan. 2004. *Uncertain Projective Geometry - Statistical Reasoning for Polyhedral Object Reconstruction.* Springer.

HOLZ, D., LORKEN, C.,SURMANN, H. 2008 (30 2008-July 3). Continuous 3D sensing for navigation and SLAM in cluttered and dynamic environments. *Pages 1–7 of: 11th International Conference on Information Fusion.*

HORN, BERTHOLD K. P., HILDEN, H.M.,NEGAHDARIPOUR, SHARIAR. 1988. Closed-Form Solution of Absolute Orientation using Orthonormal Matrices. *Journal of the Optical Society A*, **5**(7), 1127–1135.

HOWARD, ANDREW,KITCHEN, LES. 1997. Fast Visual mapping for Mobile Robot Navigation. *ICIPS'97 IEEE International Conference on Intelligent Processing Systems, Beijing.*

J. A. CASTELLANOS, J. M. M. MONTIEL, J. NEIRA,TARDÓS, J. D. 1999. The SPmap: A Probabilistic Framework for Simultaneous Localization and Map Building. *IEEE Transactions on Robotics and Automation*, **15(5)**(October), pp. 948 – 952.

JULIER, S.,UHLMANN, J. 1996. *A General Method for Approximating Nonlinear Transformations of Probability Distributions.* Tech. rept. Oxford University.

JULIER, S.,UHLMANN, J. 1997. A new extension of the Kalman filter to nonlinear systems. *In: Int. Symp. Aerospace/Defense Sensing, Simul. and Controls,Orlando, FL.*

JULIER, SIMON J.,UHLMANN, JEFFREY K. 2007. Using covariance intersection for SLAM. *Robot. Auton. Syst.*, **55**(1), 3–20.

KALMAN, R.E. 1960. A new approach to linear filtering and prediction problems,. *Transactions of the ASME - Journal of Basic Engineering on Automatic Control*, **82(D)**, 35–45.

KANADE, TAKEO, AMIDI, OMEAD,KE, QIFA. 2004 (December). Real-Time and 3D Vision for Autonomous Small and Micro Air Vehicles. *In: 43rd IEEE Conference on Decision and Control (CDC 2004).*

KIM, JONG-HYUK,SUKKARIEH, SALAH. 2006. Robust Multi-loop Airborne SLAM in Unknown Wind Environments. *Pages 1536–1541 of: ICRA*. IEEE.

KOCH, K.-R. 1988. *Parameter estimation and hypothesis testing in linear models.* Springer.

KOCH, KARL-RUDOLF. 1997. *Parameterschätzung und Hypothesentests.* Dümmler.

KUIPERS, BENJAMIN, BROWNING, ROB, GRIBBLE, BILL, HEWETT, MIKE,REMOLINA, EMILIO. 2000. The Spatial Semantic Hierarchy. *Artificial Intelligence*, **119**, 191–233.

Lemaire, Thomas,Lacroix, Simon. 2007 (April). Monocular-vision based SLAM using line segments. *In: IEEE International Conference on Robotics and Automation.*

Li, Deren. 1987. *Theorie und Untersuchung der Trennbarkeit von groben Paßpunktfehlern und systematischen Bildfehlern bei der photogrammetrischen Punktbestimmung.* Ph.D. thesis, Deutsche Geodätische Kommission, Reihe C, Nr. 324, München.

Li, Ron, Di, Kaichang, Wang, Jue, Agarwal, Sanchit, Matthies, Larry, Howard, Andrew,Willson, Reg. 2005. Incremental Bundle Adjustment Techniques Using Networked Overhead and Ground Imagery for Long-Range Autonomous Mars Rover Localization. *Proceedings of the 8th International Symposium on Artificial Intelligence, Robotics and Automation (iSAIRAS), Munich, Germany.*

Liao, L., Fox, D., Hightower, J., Kautz, H.,Schulz, D. 2003. *Voronoi tracking: Location estimation using sparse and noisy sensor data.*

Lindeberg, Tony. 1998. Feature detection with automatic scale selection. *International Journal of Computer Vision*, **30**, 79–116.

Lowe, David. 2003. Distinctive image features from scale-invariant keypoints. *Pages 91–110 of: International Journal of Computer Vision*, vol. 20.

Lu, Feng,Milios, Evangelos. 1994. Robot Pose Estimation in Unknown Environments by Matching 2D Range Scans. *Journal of Intelligent and Robotic Systems*, **18**, 249–275.

Martinez-Cantin, Ruben,Castellanos, Jose A. 2005. Unscented SLAM for Large-Scale Outdoor Environments. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 328–333.

Marzorati, Daniele, Matteucci, Matteo, Migliore, Davide,Sorrenti, Domenico G. 2007. Integration of 3D Lines and Points in 6DoF Visual SLAM by Uncertain Projective Geometry. *Proceedings of ECMR.*

McGlone, J. Chris, Mikhail, Edward M., Bethel, James,Mullen, Roy (eds). 2004. *Manual of Photogrammetry.* American Society for Photogrammetry and Remote Sensing.

McLauchlan, Philip F.,Murray, David W. 1995. A Unifying Framework for Structure and Motion Recovery from Image Sequences. *Pages 314–320 of: Proceedings of the International Conference on Computer Vision.*

McLauchlan, Philip.F. 2000. A Batch/Recursive Algorithm for 3D Scene Reconstruction. *Pages II: 738–743 of: CVPR00.*

Mikhail, E. M.,Ackermann, F. 1976. *Observations and Least Squares.* University Press of America.

Mikolajczyk, Krystian,Schmid, Cordelia. 2002. An Affine Invariant Interest Point Detector. *Pages 128–142 of: ECCV '02: Proceedings of the 7th European Conference on Computer Vision-Part I.* London, UK: Springer-Verlag.

Mikolajczyk, Krystian,Schmid, Cordelia. 2005. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **27**(10), 1615–1630.

Molenaar, M. 1981. A further inquiry into the theory of S-transformation and criterion matrices. *Publications on geodesy, Vol. 7(1), Netherlands geodetic Commision, Delft.*

Montemerlo, M.,Thrun, S. 2003. Simultaneous localization and mapping with unknown data association using FastSLAM. *In: In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA).*

Montemerlo, M., Thrun, S., Koller, D.,Wegbreit, B. 2002. FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem. *Pages 593–598 of: Proceedings of the 18th National Conference on Artificial Intelligence (AAAI).*

Montiel, J.M.M., Civera, Javier,Davison, Andrew J. 2006. Unified Inverse Depth Parametrization for Monocular SLAM. *In: Robotics Science and Systems.*

Mouragnon, E., Lhuillier, Maxime, Dhome, Michel, Dekeyser, Fabien,Sayd, Patrick. 2006. Monocular Vision Based SLAM for Mobile Robots. *Pages 1027–1031 of: ICPR (3).* IEEE Computer Society.

Nüchter, Andreas. 2007. Parallelization of Scan Matching for Robotic 3D Mapping. *In: In Proceedings of the 3rd European Conference on Mobile Robots (ECMR '07), Freiburg.*

Nüchter, Andreas, Lingemann, Kai,Hertzberg, Joachim. 2006. Extracting drivable surfaces in outdoor 6d slam. *In: In Proc. of the 37nd Int. Symp. on Robotics (ISR 06.*

Niemeier, Wolfgang. 2001. *Augleichungsrechnung.* deGruyter-Verlag.

Nieto, Juan, Bailey, Tim,Nebot, Eduardo Mario. 2005. Scan-SLAM: Combining EKF-SLAM and Scan Correlation. *Pages 167–178 of: FSR.*

Nistér, David. 2004. An Efficient Solution to the Five-Point Relative Pose Problem. *IEEE Trans. Pattern Anal. Mach. Intell.*, **26**(6), 756–777.

Nistér, David. 2004. An Efficient Solution to the Five-Point Relative Pose Problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **26**(6), 756–777.

Ollero, Anibal, Ferruz, Joaquin, Caballero, Fernando, Hurtado, Sebastián,Merino, Luis. 2004. Motion Compensation and Object Detection for Autonomous Helicopter Visual Navigation in the Comets System. *In: IEEE International Conference on Robotics & Automation.*

P. M. Newman, J. J. Leonard,Rikoski, R. J. 2003. Towards Constant-Time SLAM on an Autonomous Underwater Vehicle Using Synthetic Aperture Sonar. *In: Proceedings of the Eleventh International Symposium on Robotics Research, Sienna, Italy.*

Paz, L.M., Jensfelt, P., Tardós, J.D.,Neira, J. 2007 (apr). EKF SLAM updates in O(n) with Divide and Conquer SLAM. *In: Proc. of the IEEE International Conference on Robotics and Automation (ICRA'07).*

Pfaff, Patrick, Triebel, Rudolph,Burgard, Wolfram. 2007. An Efficient Extension to Elevation Maps for Outdoor Terrain Mapping and Loop Closing. *Int. J. Rob. Res.*, **26**(2), 217–230.

Pupilli, Mark,Calway, Andrew. 2005. Real-Time Camera Tracking Using a Particle Filter. *Pages 519–528 of: Proceedings of the British Machine Vision Conference.* BMVA Press.

Pupilli, Mark,Calway, Andrew. 2006 (June). Real-time Visual SLAM with Resilience to Erratic Motion. *In: IEEE Computer Vision and Pattern Recognition.*

Ribas, D., Neira, J., Ridao, P.,Tardos, J.D. 2006 (October). SLAM using an imaging sonar for partially structured environments. *In: IEEE/RSJ International Conference on Intelligent Robots and Systems.*

Ribas, David, Ridao, Pere, Tardós, Juan Domingo,Neira, José. 2008. Underwater SLAM in man-made structured environments. *J. Field Robot.*, **25**(11-12), 898–921.

Rodríguez-Losada, Diego, Matía, Fernando, Jiménez, Agustín,Galán, Ramón. 2006. Consistency Improvement for SLAM - EKF for Indoor Environments. *ICRA*, 418–423.

Rusinkiewicz, Szymon,Levoy, Marc. 2001 (jun). Efficient Variants of the ICP Algorithm. *In: Third International Conference on 3D Digital Imaging and Modeling (3DIM).*

Schlaile, Christian, Meister, Oliver, Wendel, Jan,Trommer, Gert F. 2006. Vision based navigation of a four rotor helicopter in unprepared indoor environments. *European Micro Air Vehicle Conference and Flight Competition 2006.*

Se, S., Barfoot, T.,Jasiobedzki, P. 2005 (August). Visual Motion Estimation and Terrain Modeling for Planetary Rovers. *In: 'i-SAIRAS 2005' - The 8th International Symposium on Artificial Intelligence, Robotics and Automation in Space.* ESA Special Publication, vol. 603.

Sibley, G., Sukhatme, G.,Matthies, L. 2006 (August). The Iterated Sigma Point Kalman Filter with Applications to Long Range Stereo. *In: Proceedings of Robotics: Science and Systems.*

Sibley, Gabe. 2006. *Sliding Window Filters for SLAM.* Tech. rept. University of Southern California.

Simon, Dan. 2006. *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches.* 1. auflage edn. Wiley & Sons.

Sünderhauf, Nico, Konolige, Kurt, Lacroix, Simon,PeterProtzel. 2005. Visual Odometry using Sparse Bundle Adjustment on an Autonomous Outdoor Vehicle. *Tagungsband Autonome Mobile Systeme*, 157–163.

Sünderhauf, Nico, Lange, Sven,Protzel, Peter. 2007. Using the Unscented Kalman Filter for Mono-SLAM with Inverse Depth Parametrization for Autonomous Airship Control. *Proc. of IEEE International Workshop on Safety Security and Rescue Robotics, SSRR, Rome.*

Solà, Joan, Monin, Andre, Devy, Michel,Lemaire, Thomas. 2005. Undelayed initialization in bearing only slam. *IEEE International Conference on Intelligent Robots and Systems.*

Spero, D. J.,Jarvis, R. A. 2005 (15 Jul.). *A new solution to the simultaneous localisation and map building (SLAM) problem.* Tech. rept. MECSE-27-2005. Department of Electrical and Computer Systems Engineering, Monash University, Melbourne, Australia.

Stefanovic, P. 1978. *Blunder and least square.* Publications of the International Institute for Aerial Survey and Earth Science (ITC).

Steffen, Richard,Beder, Christian. 2007. Recursive Estimation with Implicit Constraints. *Pages 194–203 of:* Hamprecht, F.A., Schnörr, C.,Jähne, B. (eds), *Proceedings of the DAGM 2007.* LNCS, no. 4713. Springer.

Steffen, Richard,Förstner, Wolfgang. 2008. On Visual Real Time Mapping for Unmanned Aerial Vehicles. *In: 21st Congress of the International Society for Photogrammetry and Remote Sensing (ISPRS).*

Sturm, Peter, Ramalingam, Srikumar,Lodha, Suresh. 2006. On Calibration, Structure from Motion and Multi-View Geometry for Generic Camera Models. *In:* Daniilidis, Kostas,Klette, Reinhard (eds), *Imaging Beyond the Pinhole Camera.* Computational Imaging and Vision, vol. 33. Springer.

Tardós, Juan D., Neira, José, Newman, Paul M.,Leonard, John J. 2002. Robust mapping and localization in indoor environments using sonar data. *Int. J. Robotics Research*, **21**, 311–330.

Thrun, S. 2002. Robotic Mapping: A Survey. *In:* Lakemeyer, G.,Nebel, B. (eds), *Exploring Artificial Intelligence in the New Millenium.* Morgan Kaufmann. to appear.

Thrun, S.,Montemerlo, M. 2005. The GraphSLAM Algorithm With Applications to Large-Scale Mapping of Urban Structures. *International Journal on Robotics Research*, **25**(5/6), 403–430.

Thrun, Sebastian, Burgard, Wolfram,Fox, Dieter. 2005. *Probabilistic robotics.* MIT Press.

Tomasi, Carlo,Kanade, Takeo. 1991 (April). *Detection and Tracking of Point Features.* Tech. rept. CMU-CS-91-132. Carnegie Mellon University.

Trawny, Nikolas,Roumeliotis, Stergios I. 2006 (May 15 – May 19). A Unified Framework for Nearby and Distant Landmarks in Bearing-Only SLAM. *Pages 1923–1929 of: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, vol. 1.

Triggs, B., McLauchlan, P.F., Hartley, R.I.,Fitzgibbon, A.W. 1999. Bundle Adjustment - A Modern Synthesis. *Pages 298–372 of: Vision Algorithms: Theory and Practice*. Springer-Verlag.

Täubig, Holger,Schröter, Christof. 2004. Simultaneous Localisation and Mapping (SLAM) mit Hierarchischen Partikelfiltern. *In: 3rd Workshop an SelfOrganization af Adaptive Behavior.*

van der Merwe, Rudolph. 2004 (April). *Sigma-Point Kalman Filters for Probabilistic Inference in Dynamic State-Space Models*. Ph.D. thesis, OGI School of Science & Engineering, Oregon Health & Science University, Portland, OR, USA.

Yu, Chunhe,Zhang, Danping. 2006 (16-20). A New 3D Map Reconstruction Based Mobile Robot Navigation. *Pages – of: 8th International Conference on Signal Processing*, vol. 4.

Zask, R.,Dailey, M.N. 2009. Rapid 3D visualization of indoor scenes using 3D occupancy grid isosurfaces. *In: In Proceedings of ECTI-CON.*