INFORMATION FILTERING WITH SUBMAPS FOR INERTIAL AIDED VISUAL ODOMETRY

M. Kleinert[a,*] U. Stilla[b]

[a] Department Scene Analysis, Fraunhofer IOSB, Gutleuthausstr. 1, 76275 Ettlingen, Germany - markus.kleinert@iosb.fraunhofer.de
[b] Photogrammetry and Remote Sensing, Technische Universität München, Arcisstr. 21, 80333 München, Germany - stilla@tum.de

KEY WORDS: Indoor Positioning, Inertial Aided Visual Odometry, Bundle Adjustment, Submapping

ABSTRACT:

This work is concerned with the fusion of inertial measurements (accelerations and angular velocities) with imagery data (feature points extracted in a video stream) in a recursive bundle adjustment framework for indoor position and attitude estimation. Recursive processing is achieved by a combination of local submaps and the Schur complement. The Schur complement is used to reduce the problem size at regular intervals while retaining the information provided by past measurements. Local submaps provide a way to propagate the gauge constraints and thereby to alleviate the detrimental effects of linearization errors in the prior. Though the presented technique is not real-time capable in its current implementation, it can be employed to process arbitrarily long trajectories.
The presented system is evaluated by comparing the estimated trajectory of the system with a reference trajectory of a prism attached to the system, which was recorded by a total station.

1. INTRODUCTION

1.1 Importance for autonomous indoor localization

For first responders or special forces operating in unknown environments, positioning is a crucial capability. Besides position, users are often interested in their heading, i.e. the direction they are facing. Both can in principle be computed from global navigation satellite system (GNSS) signals, but recovering the heading requires that the user is moving.

However, in indoor scenarios or urban canyons, where GNSS signals cannot be received or are severely distorted due to multipath effects, alternative methods to determine position are required. In contrast to position, heading can be obtained by magnetometer measurements almost everywhere, but the local magnetic field may be disturbed by reinforced concrete inside or near building walls.

Thus, indoor positioning systems for firefighters have been developed using radio beacons which have to be placed around the site of operation before a mission starts (McCroskey et al., 2010). Such systems typically combine position estimates obtained by trilateration with relative motion estimates. In the context of pedestrian navigation, such relative motion measurements are often obtained by inertial sensors placed on the foot where the foot's stand still phase can be exploited to obtain accurate motion estimates even with inertial sensors of relatively low quality. But in this case the question how estimates of the foot's motion can be fused with measurements of devices attached to the torso needs to be addressed.

One possibility that does not rely on external infrastructure is to fuse measurements from a camera and an inertial measurement unit (IMU), to estimate one's position as well as attitude relative to a starting point. The combination of visual and inertial measurements is attractive because of the complementary error characteristics of these sensors. Compared to systems relying on foot-mounted inertial sensors, this setup allows to put all sensors in a single housing fixed to the torso of a person and it does not rely on a special motion pattern.

*Corresponding author.

1.2 Related Work

The task of estimating the pose (position and attitude) of a moving platform occurs frequently in robotics and photogrammetry. The main sensor of interest in classical photogrammetry is a single camera and the dominant approach to estimate a camera's pose from a given image sequence is to solve a non-linear optimization problem over camera poses and the positions of observed landmarks called bundle adjustment (Triggs et al., 2000). However, such a batch processing approach quickly becomes infeasible for problems involving a large number of cameras and landmarks. This has triggered attempts to employ alternative estimation techniques such as Kalman filtering (Jones and Soatto, 2011). A drawback of Kalman filtering approaches is that they require accurate initial estimates of landmark positions relative to the sensor due to their inherent susceptibility to linearization errors. Unfortunately, such initial estimates are difficult to obtain with a single camera. To alleviate this problem, (Beder and Steffen, 2008) proposed a delayed-state Kalman filtering approach, where a sliding window of poses is kept in the filter's state and the introduction of new landmarks in the state can be handled in an optimal way by iterative optimization involving all observations from poses in the sliding window. On the other hand, a lot of research is focused on modifying the bundle adjustment approach in a way that it can be applied in online applications. The theoretically sound way to achieve this is to reduce the number of estimated states by marginalizing old landmarks and poses from the state vector using the Schur complement. This results in a Bayesian filter working on the information form representation of the estimated state, which is given by the information matrix and information vector. A detailed discussion of this information filtering approach can be found in (Sibley et al., 2010). Therein a potential drawback of marginalization was already anticipated: Linearization errors may accumulate in the prior and affect later estimation. A detailed analysis of the effect of linearization errors on estimator performance, especially consistency, has been conducted in (Dong-Si and Mourikis, 2011). They show that linearization errors render directions of state space observable which are not observable by theory due to the symmetry of the problem. It is also shown that fixing the linearization points for states which are part of the prior remedies this effect.

Motivated by the observation that the detrimental effect of linearization discovered by (Dong-Si and Mourikis, 2011) is closely related to the problem of gauge fixing in estimation problems, the approach presented in this work tackles this problem by applying the local submapping procedure presented in (Piniés and Tardós, 2008) in the context of information filtering. Local submaps provide a way for consistent gauge definition and thus may provide a way to control the observable subspace. However, a formal proof of this statement has to be deferred to future work.

1.2.1 The main contribution of this paper is presented in Sec. 2.6, where it is shown how a new local reference coordinate system is set up before marginalizing old states.

2. INFORMATION FILTERING WITH SUBMAPS

2.1 Coordinate systems and notation

Several coordinate systems are used in the following presentation of the system model. All coordinate systems are assumed to be right-handed. The purpose of the algorithm presented here is to estimate the system's trajectory and a sparse map of point feature locations relative to an established frame of reference, which is henceforth called navigation frame $\{\mathrm{n}\}$. Its z-axis points in the direction of local gravity, but its origin and rotation about the z-axis may be chosen arbitrarily, reflecting the freedom in selecting the gauge constraints. If some guess of initial position and heading is available, the free parameters can be adjusted accordingly. Each local submap is build up relative to its own frame of reference $\{\mathrm{s_i}\}$, where i refers to the number of the local map. The submap index is omitted wherever confusion is not possible. Furthermore, the sensor system's frame of reference (body frame) is denoted $\{\mathrm{b}\}$. It is assumed that the rigid transformations between all sensors are fixed and known, possibly from a calibration procedure. As a result, all sensor readings can be written w.r.t. the body frame.

The rigid body transformation between two frames, $\{\mathrm{a}\}$ and $\{\mathrm{b}\}$, is described by a pair consisting of a rotation matrix (direction cosine matrix) C_b^a and a translation vector ${}^a\boldsymbol{p}_b$. Since C_b^a transforms coordinates written in the basis of $\{\mathrm{b}\}$ to the basis $\{\mathrm{a}\}$ and ${}^a\boldsymbol{p}_b$ is the position of $\{\mathrm{b}\}$ in $\{\mathrm{a}\}$'s coordinates, the pair T_b^a=(C_b^a,${}^a\boldsymbol{p}_b$), maps points from $\{\mathrm{b}\}$ to $\{\mathrm{a}\}$.

For the refinement of initial estimates, the error state notation is used (Farrell and Barth, 1999). Estimates are marked by a hat $\hat{(\cdot)}$, measurements by a bar $\bar{(\cdot)}$, and errors by a tilde $\tilde{(\cdot)}$. For most state variables an additive error model can be applied: $\tilde{(\cdot)} = (\cdot) - \hat{(\cdot)}$. However, attitude error is represented by a rotation vector Ψ_b^a, which is an element of the Lie algebra $\mathfrak{so}(3)$ belonging to the group of rotation matrices $\mathrm{SO}(3)$.

Attitude errors are corrected by left-multiplication:

$$C_b^a = C(\Psi_b^a)\hat{C}_b^a \tag{1}$$

The relationship between a rotation vector and the corresponding rotation matrix is $C(\Psi) = \exp(\Psi) \approx I + \lfloor\Psi\rfloor_\times$. This can also be stated as $\Psi = \mathrm{vec}(\log(C(\Psi)))$. Rotation vectors can be mapped between frames just like ordinary vectors using the adjoint map Ad_g. These facts and more background material can be found in (Murray et al., 1994). Whereas attitude is represented by rotation matrices here, it is represented by quaternions in the implementation.

To handle heterogeneous states, which may be composed of rotation matrices and vectors, the different entities are combined to a tuple. The tuple is then corrected by applying the corrections to its elements individually, i.e. using Eq. 1 for rotations and addition for vectors. The operators '$\oplus$', '$\ominus$' are used to mark this operation on the tupel's elements: $\boldsymbol{t} = \hat{\boldsymbol{t}} \oplus \tilde{\boldsymbol{t}}$. Note, that the error $\tilde{\boldsymbol{t}}$ belongs to a vector space.

2.2 Camera measurement model

A camera can be regarded as a bearings measuring device. Thus it is assumed that a camera projection model $\pi(.)$ is available, which allows to calculate the projection of a 3D-point onto the image plane and its inverse. The projection of landmark number j onto image plane i is given by:

$$\bar{\boldsymbol{z}}_{ij} = \pi_i({}^b\boldsymbol{X}_j) + \boldsymbol{v}_{ij} \tag{2}$$

Where $\boldsymbol{v}_{ij}$ is an error term, which is usually assumed to arise from a zero-mean white Gaussian noise process with covariance Σ_{cam}.

2.3 IMU measurement model

IMUs measure angular velocity $\boldsymbol{\omega}$ and specific force $\boldsymbol{a}$ relative to their own reference frame. This work adopts the common assumption that the measurement noise can be described by the combination of a slow-varying bias and additive zero mean white noise:

$${}^b\bar{\boldsymbol{a}} = {}^b\boldsymbol{a} + \boldsymbol{b}_a + \boldsymbol{n}_a \tag{3}$$
$${}^b\bar{\boldsymbol{\omega}} = {}^b\boldsymbol{\omega} + \boldsymbol{b}_g + \boldsymbol{n}_g \tag{4}$$

Here, $\boldsymbol{b}_a$ and $\boldsymbol{b}_g$ contain accelerometer and gyroscope bias and $\boldsymbol{n}_a$, $\boldsymbol{n}_g$ are the corresponding noise terms.

Integrating the inertial measurements yields an estimate of the sensor system's motion during the integration interval. For this purpose the inertial mechanization equations are implemented w.r.t. the strapdown frame, as suggested by (Lupton, 2010). As a consequence, the error of the rigid body transformation relating the $\{\mathrm{n}\}$ to the $\{\mathrm{s}\}$ frame does not affect the projection of a landmark onto the image plane, because the quantities in Eq. 2 only depend on the system's position w.r.t. $\{\mathrm{s}\}$. Additionally, introducing an intermediate $\{\mathrm{s}\}$-frame facilitates the enforcement of the conditional independence properties for the local submaps as detailed in Sec. 2.6. The inertial mechanization equations for one timestep can be stated as follows:

$${}^s\boldsymbol{a} \leftarrow C_b^s\left({}^b\bar{\boldsymbol{a}} - \boldsymbol{b}_a - \boldsymbol{n}_a\right) + C_n^{sn}\boldsymbol{g} \tag{5}$$
$${}^s\boldsymbol{v} \leftarrow {}^s\boldsymbol{v} + {}^s\boldsymbol{a}\tau \tag{6}$$
$${}^s\boldsymbol{p}_b \leftarrow {}^s\boldsymbol{p}_b + {}^s\boldsymbol{v}\tau + \frac{1}{2}{}^s\boldsymbol{a}\tau^2 \tag{7}$$
$$C_b^s \leftarrow C_b^s C\left(\left[{}^b\bar{\boldsymbol{\omega}} - \boldsymbol{b}_g - \boldsymbol{n}_g\right]\tau\right) \tag{8}$$
$$\boldsymbol{b}_a \leftarrow \boldsymbol{b}_a + \boldsymbol{n}_{b_a} \tag{9}$$
$$\boldsymbol{b}_g \leftarrow \boldsymbol{b}_g + \boldsymbol{n}_{b_g} \tag{10}$$

Where the quantities on the left hand side refer to the point in time $t + \tau$ while quantities on the right hand side are given at time t.

In the above equations, τ is the timespan between two samples and ${}^n\boldsymbol{g} = [0, 0, 9.81]^{\mathrm{T}}$ is the vector of gravitational acceleration. Note, that at least the attitude (roll and pitch angles) of the $\{\mathrm{n}\}$ frame relative to the $\{\mathrm{s}\}$ frame needs to be known to compensate gravitational acceleration. By setting the noise terms to zero and replacing all quantities by their estimated counterparts, the mechanization equations for estimated quantities follow from Eqs. 5-10.

Combining the state variables into a tupel $\boldsymbol{s}_t$ and writing the above equations as a single state transition function f gives:

$$\begin{aligned} \boldsymbol{s}_{t+\tau} &= f(\boldsymbol{s}_t, \boldsymbol{u}, \boldsymbol{n}) && (11)\\ &\approx f(\hat{\boldsymbol{s}}_t, \boldsymbol{u}, \boldsymbol{0}) \oplus (\Phi \tilde{\boldsymbol{s}}_t + G\boldsymbol{n}) && (12) \end{aligned}$$

$$\begin{aligned} \tilde{\boldsymbol{s}}_{t+\tau} &= \boldsymbol{s}_{t+\tau} \ominus f(\hat{\boldsymbol{s}}_t, \boldsymbol{u}, \boldsymbol{0}) && (13)\\ &= \Phi \tilde{\boldsymbol{s}}_t + G\boldsymbol{n} && (14) \end{aligned}$$

Here, $\boldsymbol{u}$ contains all inertial measurements, $\boldsymbol{n}$ contains all the noise terms, and Φ, G are f's derivatives w.r.t. $\boldsymbol{s}$ and $\boldsymbol{n}$. When calculating the Jacobians, the states related to attitude require special attention. Explicitly writing Eqs. 5 and 8 in terms of incremental rotations yields:

$$\begin{aligned} {}^s\boldsymbol{a} &\leftarrow C(\Psi_b^s)\hat{C}_b^s \left({}^b\bar{\boldsymbol{a}} - \boldsymbol{b}_a - \boldsymbol{n}_a\right) + \dots \\ &\quad \hat{C}_n^s C(-\Psi_s^n)\,{}^n\boldsymbol{g} && (15)\\ &\approx (I + \lfloor \Psi_b^s \rfloor_\times)\hat{C}_b^s \left({}^b\bar{\boldsymbol{a}} - \hat{\boldsymbol{b}}_a\right) + \dots \\ &\quad \hat{C}_n^s (I - \lfloor \Psi_s^n \rfloor_\times)\,{}^n\boldsymbol{g} && (16)\\ C(\Psi_b^s)\hat{C}_b^s &\leftarrow C(\Psi_b^s)\hat{C}_b^s C\left(\left[{}^b\bar{\boldsymbol{\omega}} - \boldsymbol{b}_g - \boldsymbol{n}_g\right]\tau\right) && (17) \end{aligned}$$

Using the facts about rotation vectors and matrices presented in Sec. 2.1, the state transition function for Ψ_b^s is obtained from Eq. 17:

$$\Psi_b^s \leftarrow \Psi_b^s + C_b^s\left(\left[-\tilde{\boldsymbol{b}}_g - \boldsymbol{n}_g\right]\tau\right) \quad (18)$$

Likewise, Eq. 16 yields the derivatives of ${}^s\boldsymbol{a}$ w.r.t. Ψ_b^s, Ψ_s^n:

$$\begin{aligned} \frac{\partial {}^s\boldsymbol{a}}{\partial \Psi_b^s} &= -\left\lfloor \hat{C}_b^s\left({}^b\bar{\boldsymbol{a}} - \hat{\boldsymbol{b}}_a\right)\right\rfloor_\times && (19)\\ \frac{\partial {}^s\boldsymbol{a}}{\partial \Psi_s^n} &= \hat{C}_n^s \lfloor {}^n\boldsymbol{g} \rfloor_\times && (20) \end{aligned}$$

Thereby, the entries of Φ are obtained by standard calculus from Eqs. 5-10 and 18-20.

Concatenating the non-linear state transition functions (Eq. 11) yields the state transition function $f' = f_{k+m:m}$ for several measurements between measurement number m and $k+m$. The Jacobians Φ, G provide a linear error propagation model between successive inertial measurements. To propagate the error over several inertial measurements between exteroceptive sensor readings, a cumulative state transition matrix Φ' and covariance Σ'_{imu} are computed as follows:

$$\begin{aligned} \Phi' &= \Phi_{k+m:m} = \prod_{i=m}^{k+m-1} \Phi_{i+1:i} && (21)\\ \Sigma'_{imu} &= \sum_{i=m}^{k+m-1} \Phi_{i:m} G Q G^{\mathrm{T}} \Phi_{i:m}^{\mathrm{T}} && (22) \end{aligned}$$

By the chain rule Φ' is the Jacobian of f'. Eqs. 21-22 provide a linearized constraint for successive pose and velocity estimates between exteroceptive sensor readings, which can be used within a bundle adjustment framework.

2.4 Inference

Graphical models have become popular tools to formalize optimization problems. There are two types of graphical models which are commonly used: Dynamic Bayesian networks (DBNs) and factor graphs (Bishop, 2006). While DBNs are useful to examine the stochastic independence properties between variables in a model, factor graphs relate directly to the Gauss-Newton algorithm in the case that the distribution of state variables is jointly Gaussian. Thus, factor graphs can facilitate the implementation and description of optimization problems by providing a formal framework, which directly translates to a class hierarchy in object-oriented programming languages. In what follows, a factor graph formulation is used in the presentation of the estimation procedure, especially the marginalization of older states.

A factor graph consists of vertices, which represent state variables, and edges representing relationships or constraints between them. Typically there are different kinds of edges connecting different kinds of state variables. The relationship between vertices associated with an edge is expressed by an objective function that often depends on measured values. The strength of a constraint is determined by a weight matrix, typically the inverse of a measurement covariance. The following types of constraint edges are of interest in this work:

Landmark measurement: Each landmark observation gives rise to a constraint according to the model described in Sec. 2.2. The constraint function is

$$h_{cam}(V_{lm,ij}) = \bar{\boldsymbol{z}}_{ij} - \pi_i({}^b\boldsymbol{X}_j) \quad (23)$$

with weight matrix $\Lambda_{cam} = \Sigma_{cam}^{-1}$. $V_{lm,ij}$ is the set of connected vertices. The number of vertices in $V_{lm,ij}$ depends on the employed parameterization and measurement model. For instance, $V_{lm,ij}$ may contain a vertex containing the camera's calibration or a landmark anchor vertex.

Motion constraint: Motion constraints can be obtained by integrating inertial measurements as described in Sec. 2.3. The constraint can be stated as:

$$h_{imu}(V_{motion,k+m:m}) = \Pi\left(\hat{\boldsymbol{s}}_{k+m} \ominus f(\hat{\boldsymbol{s}}_m, \boldsymbol{u}, \boldsymbol{0})\right) \quad (24)$$

The associated weight matrix is $\Lambda_{imu} = \Sigma'^{-1}_{imu}$. In Eq. 24, Π projects the error to the states corresponding to the sensor's pose and velocity. The dimension of the error vector is therefore nine. Hence, bias and global pose are not part of the projected error vector, but the projected constraint error depends on these states nonetheless. The vertices connected by a motion edge are: $V_{motion,i:j} = \{v_{T_{b_i}^s}, v_{\boldsymbol{v}_i}, v_{T_{b_j}^s}, v_{\boldsymbol{v}_j}, v_{T_s^n}, v_{\boldsymbol{b}_{b_{a,g}}}\}$, where the trailing subscripts indicate which state variables belong to a vertex.

Equality Constraint: Equality constraints between vertices are used to model slow varying random walk processes, like biases:

$$h_{eq}(\{v_{\boldsymbol{a}}, v_{\boldsymbol{b}}\}) = \boldsymbol{a} \ominus \boldsymbol{b} \tag{25}$$

The corresponding weight matrix depends on the random walk parameters.

Transformation constraint: These are used when a new submap is created to link transformed coordinates of state variables to their estimates in the preceding submap. Depending on the type of transformed vertices, there may be different types of transformation constraints. For velocity vertices the following constraint is used:

$$h_{trans,vel}(\{v_{^a\boldsymbol{v}}, v_{b_{\boldsymbol{v}}}, v_{T_b^a}\}) = {}^a\boldsymbol{v} - C_b^{a\,b}\boldsymbol{v} \tag{26}$$

The weight matrix is a design parameter that can also be used to model process noise.

Here, inference refers to the process of estimating the state of a system based on available measurements. The sets of all constraint edges and all vertices belonging to the graph are denoted by E and V, respectively. Each edge $e \in E$ connects a set of vertices denoted $V(e)$. At the beginning of an inference step the current state dimension is calculated and each vertex is assigned an index in the state vector $\boldsymbol{x}$, which is formed by concatenating the state of all relevant vertices. Then, an empty information matrix Ω and information vector $\boldsymbol{\xi}$ are created. Let H_e be the Jacobian of Edge e, w.r.t. its vertices. A normal equations system is built up based on the constraints defined by all edges:

$$\Omega \leftarrow \Omega + \sum_{e \in E} H_e^{\mathrm{T}} \Lambda_e H_e \tag{27}$$

$$\boldsymbol{\xi} \leftarrow \boldsymbol{\xi} + \sum_{e \in E} H_e^{\mathrm{T}} \Lambda_e \boldsymbol{\epsilon}_e \tag{28}$$

In Eq. 28, $\boldsymbol{\epsilon}_e$ is the error associated with an edge and Λ_e its weight as described above. Some vertices can be fixed, for instance to enforce gauge constraints. In this case the corresponding rows and columns are deleted from Ω and $\boldsymbol{\xi}$.

Solving the normal equations yields a vector of improvements $\tilde{\boldsymbol{x}}_+$, which are applied to correct the current state estimate:

$$\hat{\boldsymbol{x}}_{i+1} = \hat{\boldsymbol{x}}_i \oplus \tilde{\boldsymbol{x}}_+ \tag{29}$$

In the implementation, the inference step is performed using the Levenberg-Marquardt algorithm based on the description in (Lourakis and Argyros, 2005).

2.5 Landmark parameterization

This work makes use of the feature bundle parameterization for landmarks (Pietzsch, 2008) in combination with the negative log parameterization (Parsley and Julier, 2008) for landmark depth. For this purpose all landmarks are assigned to an anchor pose, which is usually the pose of the sensor frame they were observed in first. Anchors are created by cloning the pose vertex of the associated sensor pose and adding an equality constraint edge between the anchor and the associated sensor pose. Thus, they can be altered even when their associated pose has been marginalized. Only the anchor's pose and the negative logarithm of the depth of its associated landmarks are treated as free parameters during estimation. This reflects the point of view that the first observation of a feature determines its direction and all following observations are noisy measurements of the directions to the same point.

2.6 Marginalization of old states

To reduce problem size and thus processing time, this work combines conditionally independent local maps (Piniés and Tardós, 2008) and marginalization via the Schur complement (Dong-Si and Mourikis, 2011). Pose and velocity vertices are added to the graph for each detected keyframe until a maximum number of pose vertices is reached. In this case a new submap is created. The last $n/2$ pose and velocity vertices remain in the graph, where n is the number of sensor pose vertices present in the graph when marginalization is started. This enables further refinement of the associated estimates when new measurements are added. Additionally, all landmark vertices connected to the remaining poses via measurement edges, the bias vertex, and the global pose vertex remain in the graph. The set of remaining vertices is henceforth denoted V_{rem} and the set of vertices to marginalize, which belong to the last submap, is V_{marg}. The set of edges connecting at least one vertex in V_{marg} is denoted E_{conn} and V_{conn} are all vertices connected by at least one Edge in E_{conn}. V_{border} is the set of remaining vertices which are connected to vertices in V_{marg} via an edge in E_{conn}.

The vertices remaining in the graph are transformed to a new coordinate system whose origin is the first pose vertex present in the new submap. To this end, the vertices in V_{border} are cloned. The resulting set of new vertices is then V_{new}. A coordinate transformation is then applied to each vertex in this set to transform it to the origin of the new map and a new edge is created and added to E_{new}, the set of new edges. This edge generally connects the new vertex, its template in V_{border}, and the pose vertex that determines the transformation and becomes the first pose vertex in the new map. Note however, that different types of vertices in V_{border} are affected by this coordinate transformation in different ways and are thus connected to their counterparts in V_{new} by different kinds of edges. E.g., the bias vertices are not affected at all by a change of coordinates. Hence, the bias vertices for the new submap are connected to the preceding bias vertices via an equality constraint whose uncertainty is determined by the bias random walk parameters. Furthermore, since the new first pose vertex is the new map's origin, its coordinates in the new map are fixed and it is not connected to any vertex in V_{border} at all. For those remaining vertices which are not connected to the previous submap, it is sufficient to transform the coordinates to the new origin. New edges do not have to be inserted in this case. Figure 1 illustrates the structure of the network after performing the coordinate transformation and adding new vertices as a DBN. It can be seen that the new layer of transformed border vertices renders the remaining vertices conditionally independent from the vertices in the previous submap. Hence, it constitutes a sufficient statistic for the remaining vertices.

Next, a set of transformed vertices (V_{trans}) is defined containing all vertices which were transformed to the new submap's origin by the procedure described above. This set of vertices constitutes the new submap. By contrast, the border vertices are added to the set of vertices to marginalize and the set of connected edges is redefined to take this into account. The redefined sets are henceforth denoted by a prime. The preparations on the network prior to calculating the Schur complement can be summed up by the following sequence of operations:

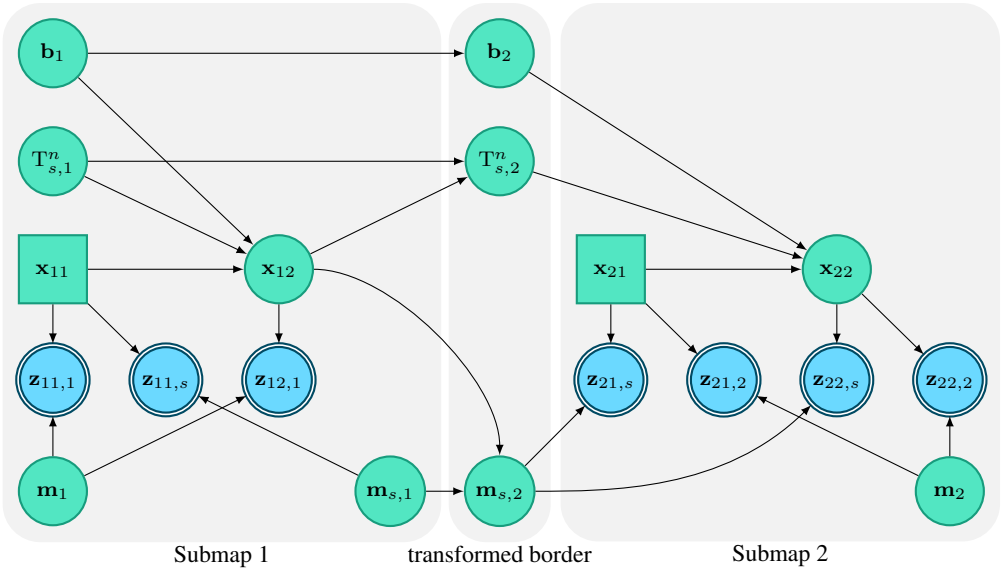

Figure 1. Dependencies between nodes in the Bayesian network corresponding to the SLAM problem before performing marginalization with the Schur complement. Green nodes with a single boundary are state variables. Rectangular green nodes represent states which are held fixed during optimization to impose the gauge constraints. Blue nodes with a double boundary represent measurements (landmark observations). Inertial measurements and velocities are not shown in this simplified network.

$$V_{marg} = V \setminus V_{rem} \tag{30}$$
$$E_{conn} = \{e \in E | \exists v_i \in V_{marg} : v_i \in V(e)\} \tag{31}$$
$$V_{conn} = \{v \in V | \exists e_i \in E_{conn} : v \in V(e_i)\} \tag{32}$$
$$V_{border} = V_{rem} \cap V_{conn} \tag{33}$$
$$V_{new} = \mathrm{cloneAndTransform}(V_{border}) \tag{34}$$
$$E_{new} = \mathrm{createTransformationEdges}(V_{border}, V_{new}) \tag{35}$$
$$\forall v \in V_{rem} \setminus V_{border} : \mathrm{applyTransform}(v) \tag{36}$$
$$V_{trans} = V_{new} \cup (V_{rem} \setminus V_{border}) \tag{37}$$
$$V' = V \cup V_{new} \tag{38}$$
$$E' = E \cup E_{new} \tag{39}$$
$$V'_{marg} = V' \setminus V_{trans} \tag{40}$$
$$E'_{conn} = \{e \in E' | \exists v_i \in V'_{marg} : v_i \in V(e)\} \tag{41}$$
$$V'_{conn} = \{v \in V' | \exists e_i \in E'_{conn} : v \in V(e_i)\} \tag{42}$$
$$V'_{rem} = V'_{conn} \setminus V'_{marg} \tag{43}$$

Here, the function $\mathrm{cloneAndTransform}(V)$ clones each vertex in V and transforms its coordinates, $\mathrm{createTransformationEdges}(V_1, V_2)$ creates constraint edges between the vertices in V_1 and V_2, and $\mathrm{applyTransform}(v)$ applies the coordinate transformation to vertex v. The applied coordinate transformation must not change the internal geometry of the network. Therefore, it is related to a S-transformation, which can be used to change between gauges during computations (Triggs et al., 2000).

Let $\boldsymbol{x}$ denote the state vector obtained by stacking the states of vertices in V'_{rem} and $\boldsymbol{y}$ the state vector associated with V'_{marg}. First, a normal equations system is build up as described by Eqs. 27 and 28, but using only edges in E'_{conn}. Then the Schur complement is used to marginalize the states in $\boldsymbol{y}$, resulting in a new information matrix and -vector for the remaining states $\boldsymbol{x}$:

$$\Omega'_{\boldsymbol{x},\boldsymbol{x}} = \Omega_{\boldsymbol{x},\boldsymbol{x}} - \Omega_{\boldsymbol{x},\boldsymbol{y}} \Omega^{-1}_{\boldsymbol{y},\boldsymbol{y}} \Omega_{\boldsymbol{y},\boldsymbol{x}} \tag{44}$$

$$\boldsymbol{\xi}'_x = \boldsymbol{\xi}_x - \Omega_{\boldsymbol{x},\boldsymbol{y}} \Omega^{-1}_{\boldsymbol{y},\boldsymbol{y}} \boldsymbol{\xi}_x \tag{45}$$

Next, a new edge, e_{prior}, is created to hold the prior information which is represented by $\Omega'_{\boldsymbol{x},\boldsymbol{x}}$ and $\boldsymbol{\xi}'_x$. To this end, all vertices belonging to $\boldsymbol{x}$ are cloned and the cloned vertices are added to $V(e_{prior})$. Furthermore, the weight matrix for e_{prior} is set to $\Omega'_{\boldsymbol{x},\boldsymbol{x}}$. To calculate the error associated with this edge, the vertices in $V(e_{prior})$ are stacked to a tuple $\boldsymbol{x}_{prior}$. Then the edge error is calculated by:

$$h_{prior} = \boldsymbol{x}_{prior} \ominus \boldsymbol{x} \tag{46}$$

The error is initially zero because $\boldsymbol{x}$ and $\boldsymbol{x}_{prior}$ are equal, but when the estimate of $\boldsymbol{x}$ is adapted due to new measurements, deviations from $\boldsymbol{x}_{prior}$ are penalized according to the weights in $\Omega'_{\boldsymbol{x},\boldsymbol{x}}$. Finally, the marginalized vertices and all edges connecting to them can be removed from the graph since the corresponding information is now represented by the prior edge:

$$E = (E' \setminus E'_{conn}) \cup \{e_{prior}\} \tag{47}$$
$$V = V' \setminus V'_{marg} \tag{48}$$

3. EXPERIMENTAL RESULTS

3.1 Simulation experiments

3.1.1 Scenario Experiments with real data demonstrate the applicability of an approach under similar conditions. When evaluating real datasets, the results are probably affected by synchronization errors, systematic feature matching errors, inaccuracies of the employed sensor model, and sensor-to-sensor calibration errors. Thus, it is difficult to draw conclusions about the performance of the employed sensor data fusion algorithm alone based on the evaluation of real datasets.

One possibility to evaluate the sensor data fusion algorithm is to compare the estimation error for several trajectories with the theoretical lower bound for accuracy. This was done to assess the

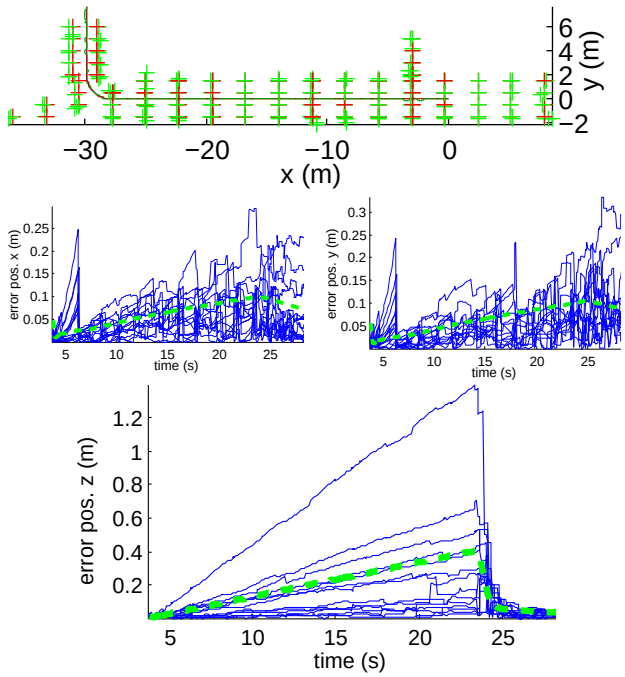

Figure 2. Evaluation of a simulated walk through a hallway. Top: Reference trajectory (dark red), estimated trajectory (dark green), landmark reference positions (red) and estimated landmark positions (green) for one simulation run. Thereunder: Position error in the x-, y- and z-direction for 13 simulation runs (blue) with the square root of the Cramér-Rao lower bound (dashed green).

performance of different approaches to structure and motion estimation from two views in (Weng et al., 1993) using the Cramér-Rao lower bound (CRLB). The CRLB is a lower bound for the variance of unbiased estimators which can be stated as follows (Bar-Shalom et al., 2001):

$$\Sigma \quad \geq \quad \check{\Omega}^{-1} \tag{49}$$

Where $\check{\Omega}$ is the Fisher information matrix, i.e., the information matrix built up in Eq. 27 with the Jacobians H_e calculated at the true values and Σ is the covariance matrix for the estimation error. Eq. 49 is valid for Gaussian, zero-mean measurement noise.

Since the calculation of the CRLB requires knowledge of the true state values, its application is essentially limited to Monte Carlo simulations. In this case it is also possible to simulate zero-mean, normally distributed measurement noise, hence satisfying another prerequisite to the application of Eq. 49. It can not be assumed that the estimation process described in Sec. 2. is unbiased. However, as stated in (Weng et al., 1993) the CRLB can still be regarded as a lower bound for the mean squared error.

A walk through a hallway was simulated in order to compare the proposed method to the CRLB. The walk starts in the middle of a hallway that is approximately 3.5 m wide and 3 m high. After following the main hallway for approximately 30 m it turns to the right into a smaller corridor that is approximately 2.5 m wide. The trajectory of the sensor system's origin was specified by a C^2-spline. The control points for this spline were chosen so as to resemble the typical up and down pattern of a walking person. White Gaussian noise was added to these control points to make the motion less regular. Another spline was used to determine the viewing direction for each point in time. The second derivative of the former spline can be calculated analytically and was used to generate acceleration measurements. Likewise, gyroscope measurements were generated by calculating the rotation vector pertaining to the incremental rotations between sampling points. The true acceleration and angular rate values obtained this way were distorted by artificial white Gaussian noise and constant offsets to match the sensor error model described in Sec. 2.3.

Image measurements were generated according to the model described in Sec. 2.2 using a fisheye projection model and the true values for landmark location and camera pose. To this end a backward-looking camera was assumed.

The CRLB was calculated once for each keyframe by building up the graphical model as described in Sec. 2.4 using all measurements available up to this point in time and setting the state variables contained in each vertex to their true values. Then the Jacobians were calculated and the system matrix was built up according to Eq. 27. This matrix was inverted to obtain the CRLB for the point in time corresponding to the keyframe. Note that marginalization was not performed in the computation of the CRLB.

3.1.2 Results In order to illustrate the spread of estimation errors, the errors pertaining to position estimates in each direction are shown in Figure 2 together with the square root of the calculated CRLB for 13 Monte Carlo runs using the scenario described in the previous section. The limitation to position errors in this investigation is justified by the fact that they depend on the remaining motion parameters through integration. Therefore, it can not be expected to achieve good position estimates when the estimates for attitude or velocity are severely distorted. When interpreting the error plots in Figure 2 it has to be considered that the optimization of the whole graph was only performed when the reprojection error exceeded a threshold at irregular time intervals. In the meantime the estimation error could grow arbitrarily. This explains the ragged appearance of the error curves.

For an estimator that attains the CRLB it can be expected that approximately 70% of all error plots lie below the square root of the CRLB. This is because the CRLB is the lower bound on the error covariance and for a normally distributed variable approximately 70% of all realizations lie in the one sigma interval. Therefore, approximately four out of the 13 error plots shown in Figure 2 are expected to exceed the square root of the CRLB, if the estimator is efficient.

A visual inspection of the plots shows that the trend in the error curves generally follows the square root of the CRLB. Furthermore, the number of error curves exceeding the square root of the CRLB is slightly higher than the expected number of four. This indicates that the proposed method does not make full use of the information available. However, the deviation from the CRLB appears to be acceptable in this example.

After approximately 24 sec., a drop in the position error plots and the square root of the CRLB can be observed. This corresponds to the point in time when the trajectory bends to the right into the next corridor, which goes along with a notable rotation about the yaw axis. A possible explanation for this behavior is that the roll- and pitch angles become separable from the acceleration biases by this motion. If the error in z-direction (height) is correlated with the roll- and pitch errors, this would also cause a correction of the estimated height.

3.2 Real-data experiments

3.2.1 Experimental setup The submapping approach presented in this paper was tested on datasets recorded inside a university building. The experimental site features wide and long

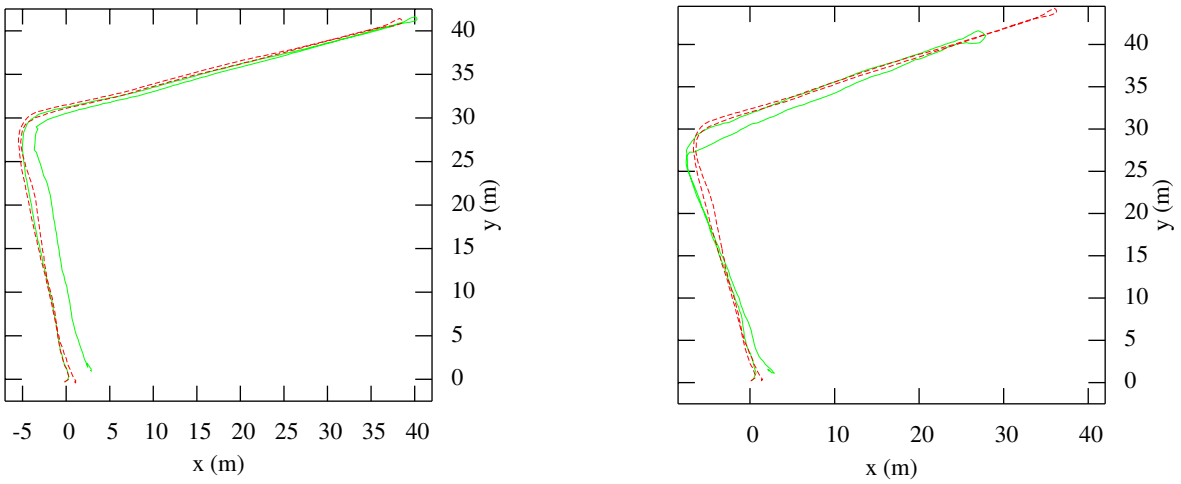

Figure 3. Experimental results for one run with the camera looking forward (left), and a run with the camera looking backwards (right). The reference trajectories recorded by the total station are shown as dashed red lines, the estimated trajectories are shown as green lines.

hallways and varying illumination conditions. In the experiments the sensor system was attached to the torso of a person walking through a hallway. For each test run, a reference trajectory was recorded by tracking the position of a prism with a total station while the system was recording data. The prism was attached to a rod which in turn was mounted on a plastic plate that was rigidly attached to the sensor system. The leverarm between the prism and the camera was calibrated prior to the experiments while being in standstill. For this purpose, a mirror-based calibration procedure similar to the one presented in (Hesch et al., 2009) was developed, which allows to obtain an estimate of the leverarm without the necessity to resort to additional sensors. Moreover, visual markers whose position were measured by the total station were placed such that they were in the camera's field of view at startup. In combination with the leverarm, this allows to transfer the reference trajectory to the frame of reference the estimates are calculated in. The employed sensor system comprises an XSens MTi-G-700 IMU which triggered an industrial camera at approximately 28 Hz to obtain synchronized video data. The camera was equipped with a Fisheye-lens to facilitate the tracking of features in indoor scenarios.

To obtain a quantitative measure for the similarity of estimated trajectories to their associated reference trajectories, the trajectories are downsampled to polygonal curves with an equal and fixed number of segments. Here, 250 segments were used. Then, the Fréchet distance between the downsampled polygonal curves is computed using a publicly available implementation of the algorithm described in (Alt and Godau, 1995). The Fréchet distance can be imagined as the minimum length of a rope needed to connect two curves while moving along them without going backwards. Thus, it provides a parameterization-independent measure of the resemblance of polygonal curves. If the samples are evenly spaced we expect the error introduced by sampling to be below 0.5 m as long as the overall length of the trajectory is less then 125 m.

3.2.2 Results Figure 3 shows the results obtained for two walks under the conditions described in the previous section. Both experiments were conducted in the same hallway, but while the camera was pointing in walking direction during the first experiment (left figure), it was mounted on the pedestrian's back during the second experiment (right figure). By visual inspection the estimated trajectory seems to match the reference trajectory well in the first experiment, but there is a significant error for the second experiment. This is also reflected by the calculated Fréchet distances of 2.3 m for the forward-looking and 8.4 m for the backward-looking configuration, respectively.

Due to the flexibility of the plastic plate, the rod holding the prism was able to swing. This resulted in a deviation of the prism's position from the equilibrium position in the order of a few centimeters. However, it is assumed that this effect can be neglected compared to the estimation error, which is in the order of some meters.

The large deviation between the estimated and the reference trajectory observed for the run with the backward-looking camera shown on the right side in Figure 3 raises questions about the presence of systematic, unmodeled errors. The simulation results presented in Sec. 3.1.2 suggest that the backward-looking configuration itself is not the cause of those errors.

At startup the walls with observed features were further away from the camera when it was looking backwards than in the first experiment with a forwad-looking camera. Hence, the prior for the depth of landmarks observed at startup described the true depth distribution better for the forward-looking configuration. However, an investigation of initial depth prior edges for the backward-looking configuration showed that their energy (i.e. the normalized sum of squared residuals) is generally small compared to the energy associated with measurement edges. Moreover, prior edges for landmark depth with high energy are removed from the graph. Thus, prior edges should not contribute spurious information.

Image features are extracted and tracked using the algorithms described in (Förstner and Gülch, 1987) which provide a measure that is used to prune false matches between successive frames. In addition, the epipolar constraint is enforced for all pairs of matching features between successive frames using a RANSAC-based algorithm. However, as illustrated in Figure 4 these steps can not prevent a gradual drift of feature locations over time. Therefore, the maximal track length was limited to 25 frames. A comparison of the distribution of landmark measurement edge energies during one simulation run and the real-data experiment with the backwards-looking camera indicates that the noise distribution can not be described by white Gaussian noise for the real-data experiment, cf. Figure 5. Based on this observation, it is expected that gradual feature track drift is the most probable cause for the error observed in the second experiment.

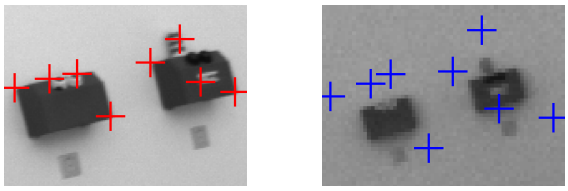

Figure 4. Observed drift of feature tracks over time. Left image: Red crosses mark detected features. Right image: Blue crosses mark the tracked features after 160 images (camera facing backwards).

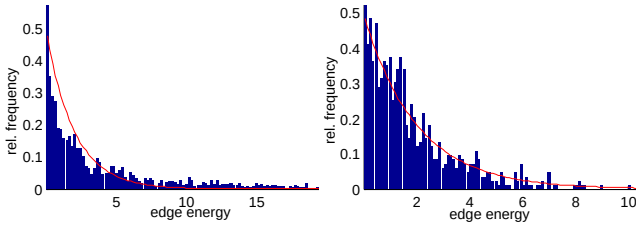

Figure 5. Relative frequency of landmark measurement edge energy at one point in time for the real dataset with the camera facing backwards (left) and for one simulation run (right). The corresponding χ^2 pdf is drawn in red assuming a stdv. of 0.33 pixel for real measurement noise and 1 pixel stdv. for simulated measurement noise. Under the white Gaussian noise assumption the χ^2 pdf should provide a upper bound for the distribution of edge energies.

4. CONCLUSIONS AND FUTURE WORK

This work presents a local submapping approach to the inertial-aided visual odometry problem which allows to relinearize over past poses in an information filter framework. The key idea is to establish a consistent gauge based on local submaps. However, the quality of the trajectories estimated by the current approach does not seem to justify the excessive processing time due to repeated relinearization and inversion of densely populated normal equations. A possible application of the presented algorithm would be to use it as a reference for simpler algorithms in situations where accurate reference data is not available.

Future work should concentrate on improving the condition number of the system matrix built up during the inference step. For this purpose it might be of interest to consider alternative gauge specifications. As a step towards real-time capability it would be beneficial to obtain a sparse approximation for the prior information matrix, for instance by applying a sparsification step as it is done in SEIFs (Thrun et al., 2005).

ACKNOWLEDGEMENTS

The authors would like to thank Mr. Zachary Danziger for publicly providing Matlab code to calculate the Fréchet distance between two polygonal curves.

REFERENCES

Alt, H. and Godau, M., 1995. Computing the fréchet distance between two polygonal curves. International Journal of Computational Geometry & Applications 5(01n02), pp. 75–91.

Bar-Shalom, Y., Li, X. R. and Kirubarajan, T., 2001. Estimation with Applications to Tracking and Navigation. John Wiley & Sons, Inc.

Beder, C. and Steffen, R., 2008. Incremetal estimation without specifying a-priori covariance matrices for the novel parameters. In: CVPR Workshop on Visual Localization for Mobile Platforms (VLMP).

Bishop, C. M., 2006. Pattern Recognition and Machine Learning. Springer Science+Business Media, LLC.

Dong-Si, T.-C. and Mourikis, A. I., 2011. Motion tracking with fixed-lag smoothing: Algorithm and consistency analysis. In: Robotics and Automation (ICRA), 2011 IEEE International Conference on, IEEE, pp. 5655–5662.

Farrell, J. and Barth, M., 1999. The Global Positioning System & Inertial Navigation. McGraw-Hill.

Förstner, W. and Gülch, E., 1987. A fast operator for detection and precise location of distinct points, corners and centres of circular features. In: Proceedings of the ISPRS Conference on Fast Processing of Photogrammetric Data, pp. 281–305.

Hesch, J. A., Mourikis, A. I. and Roumeliotis, S. I., 2009. Mirror-based extrinsic camera calibration. In: Algorithmic Foundation of Robotics VIII, Springer, pp. 285–299.

Jones, E. S. and Soatto, S., 2011. Visual-inertial navigation, mapping and localization: A scalable real-time causal approach. The International Journal of Robotics Research 30(4), pp. 407–430.

Lourakis, M. I. and Argyros, A. A., 2005. Is levenberg-marquardt the most efficient optimization algorithm for implementing bundle adjustment? In: International Conference on Computer Vision, 2005, Vol. 2, IEEE, pp. 1526–1531.

Lupton, T., 2010. Inertial SLAM with Delayed Initialisation. PhD thesis, School of Aerospace, Mechanical and Mechatronic Engineering, The University of Sydney.

McCroskey, R., Samanant, P., Hawkinson, W., Huseth, S. and Hartman, R., 2010. Glanser - an emergency responder locator system for indoor and gps-denied applications. In: 23rd International Technical Meeting of the Satellite Division of The Institute of Navigation, Portland, OR, September 21-24, 2010.

Murray, R. M., Li, Z. and Sastry, S. S., 1994. A Mathematical Introduction to Robotic Manipulation. CRC Press.

Parsley, M. P. and Julier, S. J., 2008. Avoiding negative depth in inverse depth bearing-only slam. In: IEEE/RSJ International Conference on Intelligent Robots and Systems.

Pietzsch, T., 2008. Efficient feature parameterisation for visual slam using inverse depth bundles. In: Proceedings of BMVC.

Piniés, P. and Tardós, J. D., 2008. Large scale slam building conditionally independent local maps: Application to monocular vision. IEEE Transactions on Robotics pp. 1–13.

Sibley, G., Matthies, L. and Sukhatme, G., 2010. Sliding window filter with application to planetary landing. Journal of Field Robotics 27(5), pp. 587–608.

Thrun, S., Burgard, W. and Fox, D., 2005. Probabilistic Robotics. The MIT Press.

Triggs, B., McLauchlan, P., Hartley, R. and Fitzgibbon, A., 2000. Bundle adjustment - a modern synthesis. In: B. Triggs, A. Zisserman and R. Szeliski (eds), Vision Algorithms: Theory and Practice, Lecture Notes in Computer Science, Vol. 1883, Springer Berlin / Heidelberg, pp. 153–177.

Weng, J., Ahuja, N. and Huang, T. S., 1993. Optimal motion and structure estimation. IEEE Transactions on Pattern Analysis and Machine Intelligence 15(9), pp. 864–884.