Motion-based Online Calibration for Non-overlapping Camera Views

Frank Pagel and Dieter Willersinn
Fraunhofer Institute of Optronics, System Technologies and Image Exploitation (IOSB)
Autonomous Systems and Machine Vision
76131 Karlsruhe, Germany
{frank.pagel, dieter.willersinn}@iosb.fraunhofer.de

***Abstract*— The use of multiple cameras in vehicles becomes more and more attractive as hardware prices decrease rapidly. Multiple camera sensors can be used to cover the whole environment of a vehicle and for 3D scene reconstruction using stereo or structure from motion techniques. To be able to use all sensor informations in a common coordinate frame, it is necessary to know the relative positions and orientations of the cameras. Calibration procedures (offline or online) determine these parameters using point correspondences between the camera images. However, camera configurations to monitor the entire area around a vehicle often have non-overlapping fields of view due to cost reasons. In that case, common techniques based on corresponding image points are no more applicable. This contribution outlines a concept to perform an online calibration of multiple cameras on a mobile platform with non-overlapping fields of view. We use the motion of the cameras and local image features to define constraints that allow for the calculation of the calibration parameters.**

I. Introduction

More and more cameras are mounted on mobile robots to improve their situation awareness. Cameras are also in use in modern advanced driver assistant systems to provide additional environmental information or to give warnings in critical situations to the driver. Structure from motion techniques (SFM) are widely used to reconstruct the three-dimensional structure of the surrounding scene. This contribution focuses on a multiocular camera configuration of a moving platform with sparsely or non-overlapping fields of view (FOV) as e. g. shown in fig. 1. Such camera constellations may be in use when a large portion of the surrounding scene shall be covered but only a limited number of cameras is available mainly to cost reasons.

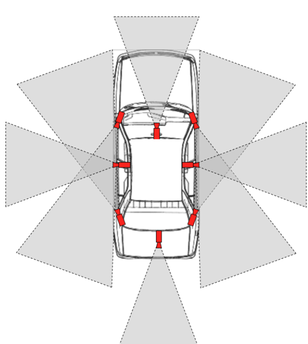

Fig. 1. Example of a multiple camera set with only sparsely overlapping and completely non-overlapping FOV.

3D scene reconstruction with multiple cameras is a growing field of research [3], [7]. Considering modern SFM techniques, single cameras can also be used for egomotion estimation [5], [8]. Most important for all multiocular reconstruction tasks is the knowledge of the camera parameters, that are the intrinsic (focal length, principal point and lens distortion) and the extrinsic (rotation and translation) parameters. Extrinsic parameters describe the geometric relationship between the cameras. For example, if a fusion of the cameras' 3D data is required (e. g. acquired from a SFM-approach) one might need to know the extrinsic parameters of the camera rig. Common calibration techniques fail because of the non-overlapping FOV. The cameras do not see the same scene and hence no corresponding image features can be used. This paper adresses the online calibration without using any pattern or known scene structure.

The next Section II gives a short overview of the current state of the art that is of interest for the proposed solution. Section III introduces the geometric model and basic algorithmic techniques. The motion estimation process is presented in Section IV whereas Section V deals with the calibration process. Section VI proposes an iterative refinement procedure. Some remarks are made in VII and Section VIII gives a conclusion and an outlook to future work.

II. Related Work and Proposed Solution

There exist several approaches in literature for calibrating rigid multi-camera networks with non-overlapping FOV with mirrors [5] or automatically [9]. However, such approaches are hardly transferable to mobile camera rigs.

Lamprecht et al. [6] use well-known patterns from the scene, e. g. traffic signs, to determine the extrinsic parameters of two non-overlapping cameras in a vehicle. The drawback of this approach is that the localization of the pattern must be very precise to perform a precise calibration. Once a pattern is detected in a camera it must be stored and redetected in the other cameras where the object may occur projectively distorted or from a completely different view.

A purely motion-based approach was proposed by Esquivel et al. [4]. They only use the trajectories of the single cameras to determine the extrinsic configuration. Similar to [6] this approach needs a robust egomotion estimation and could be used offline as well as online. Unfortunately, the algorithm strongly depends on the rotational motion and

hence the quality of the calibration suffers from the almost planar motion of regular ground vehicles.
Dang et al. [3] propose an approach to calibrate a stereo camera rig online by using a robust Kalman filter and constraints based on image features. Although they assume overlapping FOV their approach is quite related to our contribution.

Our goal is to determine the extrinsic calibration parameters of a set of multiple cameras on a moving platform. As the proposed calibration procedure is based on the cameras' motion, there is also need for an accurate motion estimation. Each camera is embedded into a module that serves as a calculation unit and hence manages the motion and calibration states. Furthermore we are interested in a global state estimation. *Global* in this context means, that the states of all other modules are considered when a single module's state is estimated. In contrast, when a module's state is calculated only based on the local sensor data, it is called *local*. All local estimations in this contribution are performed with an extended Kalman filter.
A global overall optimization for all calibration and motion parameters is likely to fail because of the large dimension of the resulting state vector. For N cameras there are N motion vectors and $N(N-1)/2$ extrinsic transformation parameters to estimate. As a Eucliden transformation can be described with three rotational and three translational parameters, such a global model would result in a $6 \cdot (N + N(N-1)/2)$-dimensional state vector.
The concept of propagating and merging spatial transformation parameters and its uncertainties was already used by Smith and Cheeseman [11]. This concept can be transferred to our calibration purposes. Instead of calculating the global state of a whole camera rig with a single Kalman filter that uses all sensor measurements simultaneously, the local state of each module is calculated first. Then, by using the initial extrinsic calibration and motion parameters, the local motions and extrinsic parameters as well as the corresponding errors can be propagated for the other modules. Afterwards the (local) propagations and uncertainties can be fused to get a global estimation. Such an approach is much more effective in practice because of its scalabilty and the lower computational cost per module. Our approach also ensures that the communication bandwidth between the modules is kept low.
The approach of local optimization, propagation and global fusion can be applied for both motion and extrinsic parameter estimation. Therefore a module's calibration process is devided into a motion and a calibration step.
The whole algorithm can be outlined as follows:

1) Local motion estimation
2) Local motion propagation (which is the local guess of the other motions)
3) Global motion estimation (which is the fusion of the propagated local motions)
4) Local estimation of the $N-1$ transformations describing the position and orientation of the other modules
5) Local calibration propagation (which is the local guess of all remaining extrinsics)
6) Global fusion of the local calibration guesses

As one process depends on the results of the other process (e. g. the global motion estimation needs the extrinsic parameters and the calibration depends on the cameras' motions), the whole process can be repeated iteratively.

III. Estimation, Propagation and Fusion

The calibration of a multi-camera rig on a vehicle is a complex task. Computational efficiency and scalability must be considered when developing such a system. Due to the complexity of the geometric model, the calibration of multiple cameras contains a lot of redundancy. On the other hand, this redundancy may help to increase the robustness of the results. Therefor knowledge about the quality of the current parameters as well as a concept for fusing the redundant information is necessary. Hence, we perform three steps for each motion and extrinsic parameter estimation: Local estimation of the state parameters based on the local sensor data, modelling the global system for each local module by propagating the locally computed state and uncertainty, and finally, fusing the redundant parameters to yield an unified, global model.
The first step is done using an extended Kalman filter (Section III-B). The second step is needed to make a local guess of the global model. Hence, for further processing and because of the complexity and dependencies of the geometric model it is necessary to consider the uncertainties of all variables. The task is to determine the uncertainty of a function $f(x_1, ..., x_n)$ given the covariances of the argument variables $x_1, ..., x_n$. This is done by Gaussian error propagation (Section III-C). Due to efficiency, the third step is realized by the decentralized Kalman filter concept (Section III-D).
For simplicity, we assume the cameras all to be intrinsically calibrated, which means that focal length, principal point as well as lens distortion parameters are known.

A. Geometric Model

In this Section we shortly present the parameters that are necessary to describe the complete geometric structure of a moving camera rig. Both the motion of a single camera and the relative position of two cameras can be considered as a Euclidean transformation. The transformation between two cameras $\mathbf{c}_i$ and $\mathbf{c}_j$ at time t is given by the transformation matrix

$$T_{ij} = \begin{pmatrix} \mathbf{R}_{ij} & \mathbf{t}_{ij} \\ \mathbf{0}^T & 1 \end{pmatrix}_{4\times4}.$$

$\mathbf{R}(\alpha, \beta, \gamma) \in \mathbb{R}^{3\times3}$ is a rotation matrix with $\mathbf{R}^T\mathbf{R} = \mathbf{R}\mathbf{R}^T = \mathbf{I}$ and $\mathbf{t} \in \mathbb{R}^3$ is a translation vector. Hence, the inverse is given by

$$T_{ij}^{-1} = T_{ji} = \begin{pmatrix} \mathbf{R}_{ij}^T & -\mathbf{R}^T\mathbf{t}_{ij} \\ \mathbf{0}^T & 1 \end{pmatrix}_{4\times4}.$$

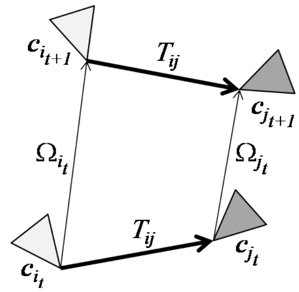

Fig. 2. Basic geometric constellation for a 2-camera rig. T is the extrinsic calibration matrix, Ω is the camera motion.

The motion of camera $\mathbf{c}_i$ between two time steps t and $t+1$ is given by

$$\Omega_{i_t} = \begin{pmatrix} \mathbf{W}_{i_t} & \mathbf{v}_{i_t} \\ \mathbf{0}^T & 1 \end{pmatrix}_{4\times 4},$$

with rotation matrix $\mathbf{W}(\omega_x, \omega_y, \omega_z) \in \mathbb{R}^{3\times 3}$ and translation vector $\mathbf{v} \in \mathbb{R}^3$. These relations are shown in Fig. 2.

B. Extended Kalman Filter

In this paper we propose the well known extended Kalman filter (EKF) for the local estimation of the cameras' egomotion as well as the calibration parameters. Given an observation at time t and the system state of the previous time step, the EKF calculates the current state by minimizing a constraint function

$$\mathbf{h}(\mathbf{x}, \mathbf{z}) = \begin{bmatrix} \vdots \\ \mathbf{h}_i(\mathbf{x}, \mathbf{z}_i) \\ \vdots \end{bmatrix} = \mathbf{0} \tag{1}$$

with state $\mathbf{x}$ and measurement $\mathbf{z} = ..., \mathbf{z}_i, ...$ For our intention, the EKF is adequate for two reasons: First, especially for the egomotion, the Kalman filter is easily extendable by additional predictions, e. g. odometric informations or algorithmic precalculations like a fundamental matrix factorization. So, in principle any motion estimation sensor or algorithm can be integrated into the filter. Second, unlike raw sensor data or time independent motion calculations, the Kalman filter provides an information respective an uncertainty measure in the form of a covariance matrix. The inverse covariance matrix $\mathbf{P}_t^{+^{-1}}$ encodes the accuracy of $\hat{\mathbf{x}}_t^+$ and hence measures the information of the current estimation. That is why $\mathbf{P}_t^{+^{-1}}$ is also called the *information matrix*. A detailed description of the basic equations and assumptions of the EKF can be found at [10].

C. Error Propagation

In this Section we describe and derive the basic principle of Gaussian error propagation (GEP).
Given a linear function $f : \mathbb{R}^n \longrightarrow \mathbb{R}^m$ with

$$f(\mathbf{x}) = \mathbf{a} + \mathbf{A}\mathbf{x} = \mathbf{y}. \tag{2}$$

The observation $\vec{\xi}$ and the resulting function value $\vec{\eta}$ both follow a normal distribution $\vec{\xi} \sim N(\mathbf{x}_0, \mathbf{\Sigma_x})$ and $\vec{\eta} \sim N(\mathbf{y}_0, \mathbf{\Sigma_y})$ respectively around the true values $\mathbf{x}_0$ and $\mathbf{y}_0$. It follows for the observed values $\vec{\eta}$ and $\vec{\xi}$

$$\vec{\eta} = \mathbf{a} + \mathbf{A}\vec{\xi} \tag{3}$$

and

$$\mathbf{y}_0 = \vec{\eta} + \vec{\epsilon} = \mathbf{a} + \mathbf{A}\mathbf{x}_0 = \mathbf{a} + \mathbf{A}(\vec{\xi} + \vec{\rho}). \tag{4}$$

Subtracting $\vec{\eta}$ yields

$$\vec{\epsilon} = \mathbf{A}\vec{\rho}. \tag{5}$$

As the resulting errors are also normally distributed with $\vec{\rho} \sim N(\mathbf{0}, \mathbf{\Sigma_x})$ and $\vec{\epsilon} \sim N(\mathbf{0}, \mathbf{\Sigma_y})$, it follows due to the linearity characteristics of the normal distribution

$$\mathbf{\Sigma_y} = \mathbf{A}\mathbf{\Sigma_x}\mathbf{A}^T. \tag{6}$$

Now, consider a nonlinear function $g : \mathbb{R}^{n\times p} \longrightarrow \mathbb{R}^m$ with

$$\mathbf{y} = g(\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_p) \tag{7}$$

and independent variables $\mathbf{x}_1, ..., \mathbf{x}_p$. May g be continously differentiable around the observed values $\vec{\xi}_1, ..., \vec{\xi}_p$. Then we can linearize g with a Taylor expansion:

$$\begin{aligned} \mathbf{y}_0 &= \vec{\eta} + \vec{\epsilon} = g(\vec{\xi}_1 + \vec{\rho}_1, ..., \vec{\xi}_p + \vec{\rho}_p) \\ &= g(\vec{\xi}_1, ..., \vec{\xi}_p) \\ &\quad + \tfrac{\partial g}{\partial \mathbf{x}_1}(\vec{\xi}_1, ..., \vec{\xi}_p) \cdot \vec{\rho}_1 \\ &\quad + ... \\ &\quad + \tfrac{\partial g}{\partial \mathbf{x}_p}(\vec{\xi}_1, ..., \vec{\xi}_p) \cdot \vec{\rho}_p \\ &\quad + G \\ &= g(\vec{\xi}_1, ..., \vec{\xi}_p) + \mathbf{J}_1\vec{\rho}_1 + ... + \mathbf{J}_p\vec{\rho}_p + G \end{aligned} \tag{8}$$

where $\mathbf{J}_i$ are the Jacobian matrices and G are the higher order terms. As we assume the errors $\vec{\rho}_i$ to be very small, we break the taylor expansion after the first order term. Subtracting $\vec{\eta}$ leads to

$$\vec{\epsilon} = \sum_{i=1}^{p} \mathbf{J}_i \vec{\rho}_i. \tag{9}$$

Analogue to (6) we get the covariance of $\mathbf{y}$ Finally, we get the covariance of $\mathbf{y}$

$$\mathbf{\Sigma_y} = \sum_{i=1}^{p} \mathbf{J}_i \mathbf{\Sigma_{x_i}} \mathbf{J}_i^T. \tag{10}$$

D. Fusion

We use the approach of the decentralized Kalman filter (DKF) as described in [1]. Hence, we assume a hierarchical structure where each sensor is connected to a single module and hence to a local Kalman filter (Fig. 3). The local results are then fused in a common fusion center. For a better reading, we omit the time subscripts t indicating the time reference.
The fused covariance and state are given by

$$(\mathbf{P}_F^+)^{-1} = (\mathbf{P}_F^-)^{-1} + \sum_{i=1}^{N} \left[(\mathbf{P}_i^+)^{-1} - (\mathbf{P}_i^-)^{-1} \right] \tag{11}$$

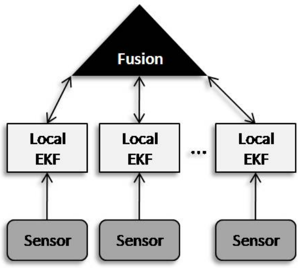

Fig. 3. Hierarchical fusion of several local Kalman filters with seperate sensor data.

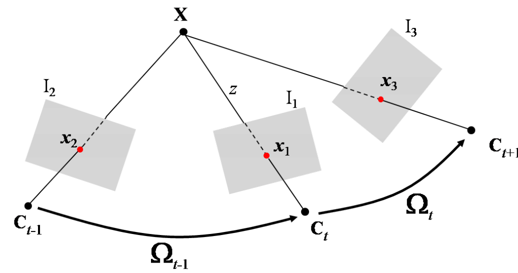

Fig. 4. Motion scheme for the estimation of $\mathbf{\Omega}_t$. Three consecutive frames with corresponding image points and a scene point $\mathbf{X}$ with depth z.

and

$$
\begin{aligned}
(\mathbf{P}_F^+)^{-1}\hat{\mathbf{x}}_F^+ = \quad & (\mathbf{P}_F^-)^{-1}\hat{\mathbf{x}}_F^- \\
& + \sum_{i=1}^N \left[(\mathbf{P}_i^+)^{-1}\hat{\mathbf{x}}_i^+ - (\mathbf{P}_i^-)^{-1}\hat{\mathbf{x}}_i^- \right].
\end{aligned} \tag{12}
$$

The index F labels the fused results. As (11) and (12) use the inverse of the covariance matrices as a measure of the information content of the estimation, this approach is also called *information matrix approach*.
Alternatively, a centralized Kalman filter could be used instead of the DKF. Centralized means that one single Kalman filter uses all sensor data at once. However, this is not desirable for practical reasons because the sensor data would have to be transmitted through a sensor network. Especially in real-time applications the communication bandwidth may become a bottleneck with an increasing number of sensors.

IV. Camera Motion

A. Local Motion Estimation

Motion or egomotion estimation purely based on camera data is also known as *visual odometry*. Here, a modified version of the motion estimation approach of Pagel [8] is used. The motion parameters of a single camera are estimated via a robust iterated extended Kalman filter (RIEKF) as proposed by Dang et al. [3]. As in [3], the motion parameters Ω_i of module M_i are determined by minimizing the epipolar constraint, the trifocal constraint and the projection error within the function

$$h_{mot}(\Omega_i, \mathbf{z}) \tag{13}$$

with respect to Ω_i and measurement $\mathbf{z} = (..., \mathbf{z}_i, ...)^T$. A single measurement is given by $\mathbf{z}_i = (\mathbf{v}, z)^T$, where $\mathbf{v}$ is an optical flow triple $\mathbf{v} = (\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3)^T$ and z is the depth of the respective scene point. z can be calculated by triangulating $\mathbf{x}_1$ and $\mathbf{x}_2$ using the known previous motion Ω_{t-1} (see Fig. 4). The robust preprocessing step detects measurement outliers in a RANSAC-like procedure. The RIEKF also refines the measurement during the iteration process.

The Kalman filter here has two big advantages: First, it can be easily extendended with other sensor sources (e. g. odometry data) or motion estimation approaches (e. g. fundamental matrix factorization) by adjoining them to the prediction step. And second and most important, the Kalman filter provides an uncertainty of the current estimation in form of a covariance matrix that is necessary for the propagation and fusion step.

B. Motion Propagation

After each camera module has estimated its own egomotion with the EKF and the sensor data (Fig. 5a), we can now determine the motion parameters of all other modules by using the calibration parameters (Fig. 5b). Again, we drop the time subscripts t. The motion estimation of module M_j based on the motion parameters of module M_i is given by

$$
\begin{aligned}
\breve{\mathbf{x}}(\hat{\Omega}_{j_i}) &= \breve{\mathbf{x}}(T_{ij}^{-1}\Omega_{i_i}T_{ij}) \\
&= \varphi\big(\breve{\mathbf{x}}(\Omega_{i_i}), \breve{\mathbf{x}}(T_{ij})\big).
\end{aligned} \tag{14}
$$

where $\breve{\mathbf{x}}(T)$ is the 6×1 state vector (3 translational and 3 rotational parameters) of the transformation matrix T. To highlight the circumstance, that the motion Ω_j is in fact a local estimation from module M_i, we add an additional subscript: Ω_{j_i}.
For a better reading we define

$$
\begin{aligned}
\breve{\mathbf{x}}(\Omega_{j_i}) &=: \mathbf{m}_{j_i} \\
\breve{\mathbf{x}}(T_{ij_k}) &=: \mathbf{c}_{ij_k} \\
Cov(\mathbf{m}_{j_i}) &=: \Gamma_{j_i} \\
Cov(\mathbf{c}_{ij_k}) &=: \Lambda_{ij_k}
\end{aligned} \tag{15}
$$

Unluckily, we have to assume that the calibration parameters are noisy or uncertain and hence the estimation $\hat{\Omega}_{j_i}$ is likely to differ from the real Ω_j. To be able to say how reliable $\hat{\Omega}_{j_i}$ actually is, we propagate the error and determine the covariance according to (10)

$$
\begin{aligned}
\Gamma_{j_i} &= \left(\frac{\partial\varphi}{\partial\mathbf{m}}\Big|_{\mathbf{m}_{i_i}}\right)^2 \Gamma_{i_i} + \left(\frac{\partial\varphi}{\partial\mathbf{c}}\Big|_{\mathbf{c}_{ij}}\right)^2 \Lambda_{ij} \\
&= \mathbf{J}_{\mathbf{m}_{i_i}} \Gamma_{i_i} \mathbf{J}_{\mathbf{m}_{i_i}}^T + \mathbf{J}_{\mathbf{c}_{ij}} \Lambda_{ij} \mathbf{J}_{\mathbf{c}_{ij}}^T.
\end{aligned} \tag{16}
$$

So, each module can calculate a local estimation of the global model by considering uncertainties of the calibration and egomotion estimations (Fig. 5c).

C. Global Motion Fusion

From the local propagation we have an estimation of each camera motion from each of the N modules (Fig. 5d). These N estimations per motion are now fused to one. Therefore

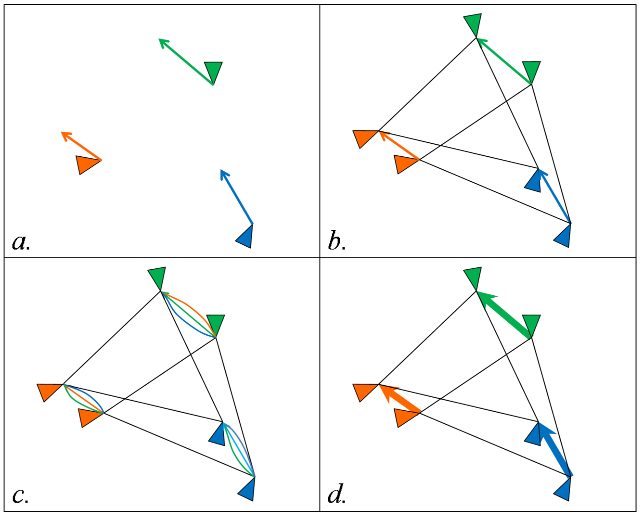

Fig. 5. *a.* Each modul performs a local motion estimation with a Kalman filter using the local sensor data. *b.* Extrinsic calibration parameters are taken into account. *c.* Each modul determines the motions of the other modules based on the extrinsic parameters via state and error propagation. Hence, there are N guesses for each of the N motions. *d.* The N local guesses of each module's motion are fused to a global estimation using the DKF approach.

we use the DKF approach. If we use the same predictions from the fusion center for all modules, (11) and (12) can be reformulated as

$$(\Gamma_j^+)^{-1} = \sum_{i=1}^{N} (\Gamma_{j_i}^+)^{-1} - (N-1) \cdot (\Gamma_j^-)^{-1} \qquad (17)$$

and

$$(\Gamma_j^+)^{-1}\hat{\mathbf{m}}_j^+ = \sum_{i=1}^{N} (\Gamma_{j_i}^+)^{-1}\hat{\mathbf{m}}_{j_i}^+ - (N-1) \cdot (\Gamma_j^-)^{-1}\hat{\mathbf{m}}_j^- \qquad (18)$$

D. Consistency of the Egomotion

To decide whether an estimated motion fits the global rig and whether it should be used for further calibration each motion can be evaluated with respect to the global rig and hence under a global point of view. In other words, we would like to know whether a module's motion is consistent with the other modules' motions depending on the extrinsic parameters.
Let us assume that each module knows all the motion parameters of the other modules calculated so far. Similar to (14) and (16), the motion Ω_i can be reformulated as

$$\hat{\Omega}_{i_j} = T_{ij}\Omega_j T_{ij}^{-1} \qquad (19)$$

with covariance Γ_{i_j} according to (10) by using the egomotion and the extrinsic calibration parameters of module M_j as well as its uncertainties. Using the definitions from (15), we can now calculate the Mahalanobis distance

$$d_{ij} = \sqrt{(\hat{\mathbf{m}}_{i_j} - \mathbf{m}_i)^T \Gamma_{i_j}^{-1} (\hat{\mathbf{m}}_{i_j} - \mathbf{m}_i)} \qquad (20)$$

This distance measure explicitly considers the uncertainty of the extrinsic parameters. If, for example, the extrinsic parameters strongly differ from the real values, but have a big variance (as is the case in the initialization phase), the Euclidean distance $\sqrt{(\hat{\mathbf{m}}_{i_j} - \mathbf{m}_i)^2}$ would be very big. However, as Γ_{i_j} is also big, d_{ij} remains small. It remains also small, if the motion Ω_j is very uncertain.
The final measure for the motion of a module M_i can then be formulated as

$$d_{M_i} = \frac{1}{N-1} \sum_{i \neq j}^{N} d_{ij}. \qquad (21)$$

V. Calibration

A. Local Parameter Estimation

At this point each module has knowledge about each module's motion parameters $\hat{\mathbf{m}}_i^+$ and hence Ω_i as well as the covariances $\Gamma_{j_F}^+ =: \Gamma_j$. Using the (robust) data $\mathbf{z}$ from the motion estimation step, we can minimize the epipolar constraint and the projection error of module M_i with a Kalman filter by replacing Ω_i in eqn. (13) with $T_{ij}\Omega_j T_{ij}^{-1}$. For module M_i, we can reformulate the constraint functions in terms of T_{ij_i} as

$$h_{cal}(T_{ij_i}, \mathbf{z}, \Omega_j) \stackrel{!}{=} 0. \qquad (22)$$

T_{ij_i} is hence the estimated extrinsic transformation between modules M_i and M_j, calculated by module M_i. The motion parameters Ω_j can be handled as additional measurements with known noise Γ_j.
For this Kalman filter step, each module uses only its own measurements. This is reasonable because as only state parameters and covariances instead of measurement data are exchanged between the modules, the communication bandwidth remains small. So, each module M_i can estimate the transformations $T_{ij_i}, j = 1, ..., N$ (Fig. 6a). Hence, $N-1$ optimizations or $N-1$ Kalman filters must be run respectively. Overall, there are $N(N-1)/2$ transformations to estimate (considering that $T_{jk} = T_{kj}^{-1}$).

B. Extrinsic Parameter Propagation and Global Fusion

At this point, after the $N-1$ Kalman filter runs, each module M_i has $N-1$ estimations of the calibration parameters $T_{ij_i}, j = 1, ..., N$. These $N-1$ transformations are sufficient to calculate the remaining transformations as follows:

$$\forall_{j,k \neq i} : \hat{T}_{jk_i} = T_{ij_i}^{-1} T_{ik_i} = T_{ji_i} T_{ik_i} \qquad (23)$$

The covariance Λ_{jk_i} can be propagated according to (10). These are $(N-1)(N-2)/2$ additional calculations. As a result, each module has a local guess for each of the $N(N-1)/2$ extrinsic parameters between the cameras. This means on the other hand that there are N guesses for each extrinsic camera transformation.

To merge these N estimates we can again proceed just like in the fusion step in Section III-D and IV-C:

$$(\Lambda_{ij}^+)^{-1} = \sum_{k=1}^{N} (\Lambda_{ij_k}^+)^{-1} - (N-1) \cdot (\Lambda_{ij}^-)^{-1} \qquad (24)$$

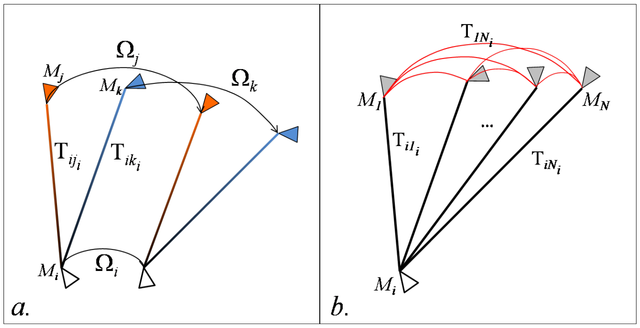

Fig. 6. *a.* Local estimations of the extrinsic parameters between module M_i and $M_{j\neq i}$ using a Kalman filter and the local sensor data. *b.* Determination of the remaining calibration parameters by state and error propagation based on the locally estimated calibration data.

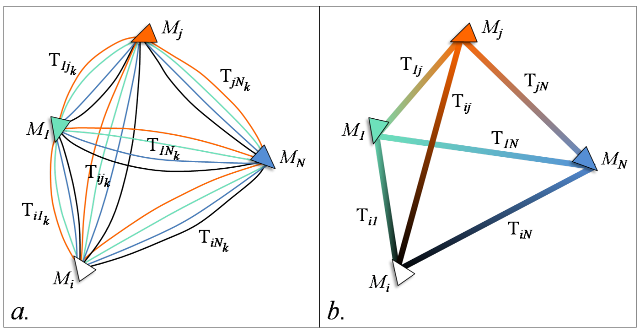

Fig. 7. *a.* After propagation, there exist $k = 1, ..., N$ local guesses for each of the $N(N-1)/2$ extrinsic transformations. *b.* The N local guesses of each transformation are fused to a global transformation using the DKF approach.

$$(\Lambda_{ij}^{+})^{-1}\hat{\mathbf{c}}_{ij}^{+} = \sum_{k=1}^{N}(\Lambda_{ij_k}^{+})^{-1}\hat{\mathbf{c}}_{ij_k}^{+} - (N-1)\cdot(\Lambda_{ij}^{-})^{-1}\hat{\mathbf{c}}_{ij}^{-}. \quad (25)$$

We now have N motion estimations for each camera and $N(N-1)/2$ estimations for all extrinsic camera transformations. Furthermore, we have a covariance matrix for every motion and calibration state vector. So finally, we have all parameters to describe to global geometric model of the moving camera rig.

VI. Iterative Refinement

The proposed algorithm so far is kind of a chicken and egg problem: Without the calibration parameters only local motion estimations are possible (that may differ from the global motion). And without knowledge of the cameras' motion, no calibration can be done at all.

At least local motion estimations are possible that can be evaluated. Hence, it can be decided, whether a module's motion should be used for the calibration or not. If the calibration parameters are unknown, the covariance matrix is modelled via a diagonal matrix with infinitely big entries. Then, after the calibration process, we assume the calibration error to be smaller than before. So we could also expect a smaller error of the global motion parameters after repeating the motion estimation process with the new calibration parameters. And with better motion results also better calibration results could be expected.

The procedure works as follows: At the beginning of iteration l, initialize the parameters of each module

$$\Omega_j'^{-}(l) := \Omega_j^{+}(l-1) \quad \text{and} \quad T_{jk}^{-}(l) := T_{jk}^{+}(l-1) \quad (26)$$

for $j, k = 1, ..., N$. Then, for the motion propagation of module M_i, set

$$\Omega_i(l) := \Omega_i^{-}(l). \quad (27)$$

According to (14), the local motion guesses made by module M_i of all the other modules $M_{j=1,...,N}, j \neq i$ can be determined as in Section IV-B

$$\hat{\mathbf{m}}_{j_i} = \varphi(\mathbf{m}_i', \mathbf{c}_{ij}^{-}) \quad (28)$$

and the covariances according to (10). The fused states $\Omega_i^{+}(l)$ and the respective covariances can be calculated as in Section IV-C with (18) and (17).

To determine the local extrinsic parameters $T_{ij_i}(l)$ for module M_i, based on the updated motion parameters, again state and covariane are propagated according to the product

$$T_{ij_i}(l) := \Omega_i^{+}(l)\cdot T_{ij}^{-}(l)\cdot \Omega_j^{+^{-1}}(l). \quad (29)$$

From now on, we can proceed with the propagation and fusion as described in Section V-B and l can be increased $l := l + 1$.

This refinement can be repeated until a fixed number of iterations is reached or the parameters do not change significantly. Notice that for the proposed refinement no Kalman filter optimization like in Section IV-A and V-A is necessary. This is because the Kalman filter based motion estimation does not depend on the extrinsic parameters (it only depends on the local measurements and hence has been optimized in the first run). And the $N-1$ Kalman filters for the local calibration estimations are replaced by a simpler state and error propagation. The basic scheme of the refinement process is shown in Fig. 8.

VII. Remarks

The DKF-steps could also be centralized as the data and the calculation in each module is exactly the same. As a drawback, the communication bandwidth would be increased, because the input and output data of the fusion step must be transferred between the central fusion unit and the modules.

From [4] we know that for an estimation of the translational parameters a rotational motion is necessary. Especially the determination of the longitudinal translation is difficult as it needs a pitch or roll rotation. Furthermore, in [2] was shown that a constant rotational motion (e. g. a circle ride) does not provide enough constraints for a unique estimation.

Unluckily, the usage of image features for minimizing the epipolar constraint and the projection error with the Kalman filter does not overcome this problem. Solutions that handle this drawback will be the focus of future research.

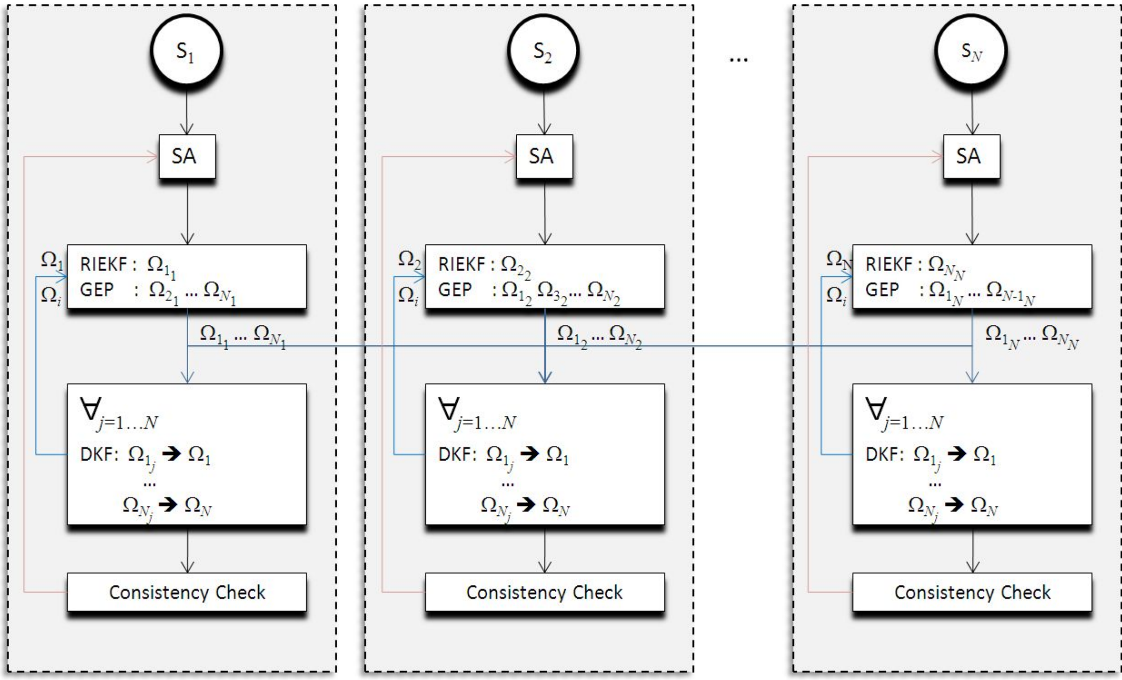

Fig. 9. Simplified scheme for motion estimation of N modules. S_i are the modules' sensors. *SA* is a situation analysis instance that evaluates the results from the motion's consistency check (Section IV-D). RIEKF and GEP (Section IV-A and IV-B)calculate local motion estimations and the DKF calculates global motion parameters by fusing the local guesses (Section IV-C).

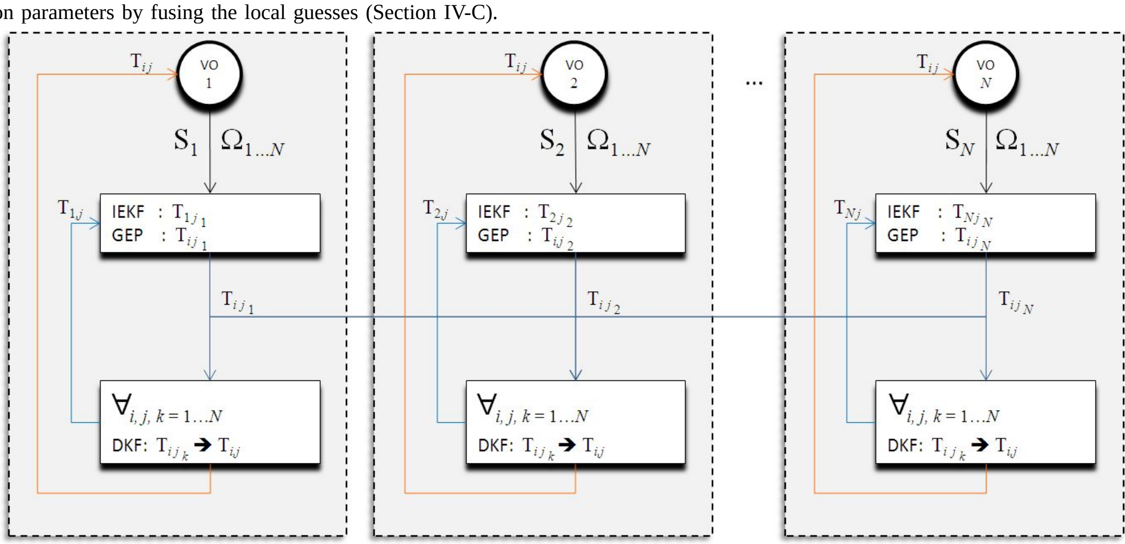

Fig. 10. Simplified scheme for extrinsic calibration of N modules. *VO* is the visual odometry from fig. 9. The IEKF and GEP (Section V-A and V-B)calculate local estimations of the calibration parameters whereas the fusion via the DKF (Section V-B) calculates the global parameters.

VIII. Conclusion and Future Work

We presented a novel approach for calibrating multiple non-overlapping cameras in a vehicle. As the geometric structure of the whole system and hence the estimation task is quite complex, our approach accounts for the uncertainties of the calibration and motion parameters. Furthermore, an iterative refinement approach was presented that allows an improvement of the parameters without requiring an additional local optimization procedure (e. g. a Kalman filter update). As one could see, the communication rate is very low as only state parameters must be transferred between the modules instead of the complete or preprocessed sensor data. Fig. 9 and 10 show a simplified architecture for a modular execution of the optimization, propagation and fusion steps. These two figures summarize Section IV and V.

In the near future we are going to implement a demo version of the proposed approach and are going to test it with simulated data. Especially robustness and convergence capabilities are of special interest. It is also planned to run the whole system in real-time on a mobile robot. Further work will yield on the integration of the ground floor estimation to overcome the restrictions described in Section VII.

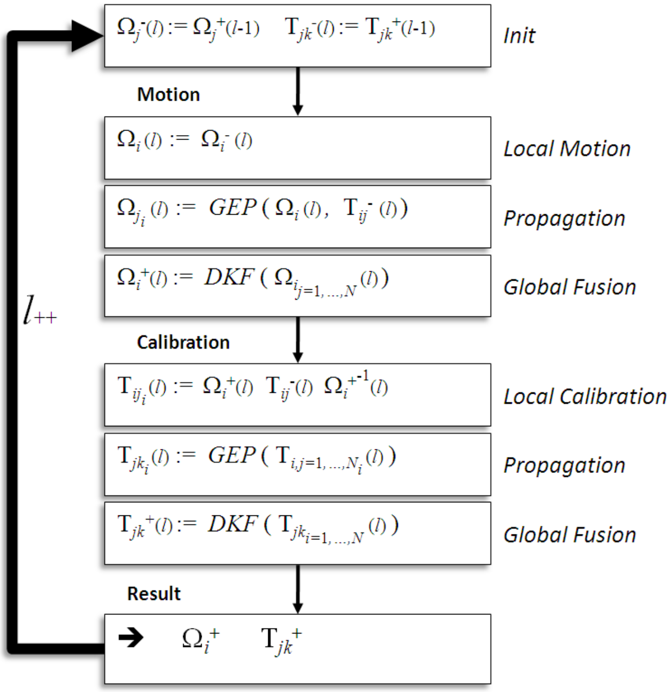

Fig. 8. Overview of the iterative refinement procedure.

References

[1] C.-Y. Chong, "Hierarchical Estimation", *in Proceedings of MIT/ONR Workshop on C3*, 1979.

[2] B. Clipp, J.H. Kim, J.M. Frahm, M. Pollefeys, R.I. Hartley, "Robust 6DOF Motion Estimation for Non-Overlapping, Multi-Camera Systems", *in Proceedings of the 2008 IEEE Workshop on Applications of Computer Vision*, 2008.

[3] T. Dang, C. Hoffmann and C. Stiller, "Continuous stereo self-calibration by camera parameter tracking", *IEEE Transactions on Image Processing*, 2009.

[4] S. Esquivel, F. Woelk, R. Koch, "Calibration of a Multi-camera Rig from Non-overlapping Views", *DAGM-Symposium*, 2007.

[5] R.K. Kumar, A. Ilie, J.M. Frahm, M. Pollefeys, "Simple calibration of non-overlapping cameras with a mirror", *in Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, 2008.

[6] B. Lamprecht, S. Rass, S. Fuchs, K. Kyamakya, "Extrinsic Camera Calibration for an On-Board Two Camera System Without Overlapping Field of View", *in Proceedings of the IEEE Intelligent Transportation Systems Conference*, 2007.

[7] A. F. Mordohai, A. Akbarzadeh, J. M. Frahm, P. Mordohai, C. Engels, D. Gallup, P. Merrell, M. Phelps, S. Sinha, B. Talton, L. Wang, Q. Yang, H. Stewenius, R. Yang, G. Welch, H. Towles, D. Nister, M. Pollefeys, "Towards Urban 3D Reconstruction From Video", *in 3DPVT*, 2006.

[8] F. Pagel, "Robust Monocular Egomotion Estimation Based on an IEKF", *IEEE Canadian Conference on Computer and Robot Vision*, 2009.

[9] A. Rahimi, B. Dunagan, T. Darrell, "Simultaneous calibration and tracking with a network of non-overlapping sensor", *in Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, 2004.

[10] D. Simon, "Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches", *Wiley-Interscience*, 2006.

[11] R. C. Smith,P. Cheeseman, "On the Representation and Estimation of Spatial Uncertaint", *in The International Journal of Robotics Research, vol. 5 no. 4*, Dec. 1986.