Extrinsic self-calibration of multiple cameras with non-overlapping views in vehicles

Frank Pagel

Fraunhofer Institute of Optronics, System Technologies and Image Exploitation IOSB
Dpt. of Video Exploitation Systems VID, Fraunhoferstr. 1, 76131 Karlsruhe, Germany

ABSTRACT

Due to decreasing sensor prices and increasing processing performance, the use of multiple cameras in vehicles becomes an attractive possibility for environment perception. This contribution focuses on non-overlapping multi-camera configurations on a mobile platform and their purely vision-based self-calibration as well as their restrictions. The usage of corresponding features between the cameras is very difficult to realize and likely to fail due to different appearances in different views and motion-dependent time delays. Instead, the hand-eye calibration (HEC) technique based on visual odometry is considered to solve this problem by exploiting the camera motions. For that purpose, this contribution presents an approach to continuously calibrate cameras by making use of the so-called motion adjustment (MA) and an IEKF. Visual odometry in driving vehicles often struggles with estimating the relative magnitudes of the translational motion, which is crucial for the HEC. So, MA simultaneously estimates the extrinsic parameters up to scale as well as the relative motion magnitudes. Furthermore, the estimation process is embedded into a global fusion framework to benefit from the redundant information resulting from multiple cameras in order to yield more robust results. This paper presents results with simulated and real data.

Keywords: Extrinsic Calibration, Multiple Cameras, Hand-Eye Calibration, Visual Odometry, Multi-Camera Fusion, Error Propagation, Motion-based Calibration, Non-overlapping Fields of View

1. INTRODUCTION

Over the last years multi-camera applications in vehicles have become more and more popular as the costs of the sensor production decreased drastically. Multiple sensors can be used to cover a wider field of view (FOV) for scene recognition and reconstruction tasks.[1] However, in many cases it is not possible to cover the full environment or to guarantee overlapping fields of view due to restrictions in design, energy consumption or physical capacity.
To be able to use the full range of possibilities that go along with such a sensor setup, the relative adjustment of the sensors must be known. The extrinsic parameters between each camera can be described as a Euclidean transformation. Then, the information that is extracted from the single cameras can be merged and referenced in a common coordinate system. This might be very useful for reconstruction, object detection or attention guidance tasks.

3D scene reconstruction with multiple cameras is a growing field of research.[2] Structure from Motion (SFM) and stereo algorithms for 3D scene reconstruction are addressed in many scientific publications (e. g.,[3,4,5]). Both, SFM and stereo algorithms are techniques for monocular or multiocular 3D scene reconstruction. Considering modern SFM techniques, single cameras can also be used for egomotion estimation[6,7] as well as for dense reconstruction.[5] Important for all multiocular reconstruction tasks is the knowledge of the camera parameters, specifically 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 that might be needed for a fusion of 3D data (e. g., acquired from a SFM-approach).
Common calibration techniques like for stereo cameras[8] fail because of the non-overlapping FOV. The cameras do not see the same scene and hence no corresponding image features can be used. So, the challenge is to recover the extrinsic parameters from a set of cameras on a moving rig with non-overlapping FOV. This paper addresses the calibration without using any pattern or known scene structure.
Although pattern-based calibration methods for non-overlapping cameras in vehicles exist, such methods are difficult to apply. In[9] traffic signs are used as calibration patterns, seen by different cameras at different times. A matching between cameras at different times causes different object appearences and requires storage and time management respectively.

Further author information: frank.pagel@iosb.fraunhofer.de, Phone: +49 (0)721 6091 518

Furthermore, for general motions, surroundings and camera adjustments, correspondences between frames cannot be guaranteed at any time (imagine, for example, a left and a right looking camera in a vehicle). The authors of[10] and[11] calibrated a multi-camera rig on a mobile robot by performing a bundle adjustment between several views and different robot positions. However, they performed their calibration in a single room and hence in a restricted and controlled area, to guarantee corresponding objects between the frames at different platform positions.
Another solution comes from the robotics community, called Hand-Eye Calibration (HEC)[12][13]. Given a robot arm and a camera (eye) at its end (hand), the goal is to determine the transformation between the coordinate systems of the eye and the hand. Hand-Eye Calibration is based on the knowledge of the position of the robot arm as well as the camera's position. In our case, instead of a robot arm and a camera, we have to deal with two cameras mounted on a vehicle and we try to determine the extrinsic parameters between them based on the camera motions. Such a solution is presented in[14] for a hand-held binocular camera rig. In,[15] a planar approach is proposed for extrinsic calibration of a two camera system with non-overlapping views and fixed camera height in a parking scenario at low speeds.

The difficulties in calibrating $\mathfrak{N} >= 2$ cameras continuously are manifold. Non-overlapping FOV mean that we cannot use common features in different views for calibrating the cameras. Instead, we calculate the motion of the cameras and use the Hand-Eye Calibration approach.[13]
By using the HEC approach, in practice we suffer from two circumstances: First, in a vehicle the rotational part of the motion transformations is rather small. The motion estimation is vision-based and therefore needs correponding features between the frames, so that we only use continuous motion estimations. And second, it appears to be difficult to estimate the absolute magnitude of the motion's translation (see Fig. 3). But this magnitude, especially the difference between the velocities of two camera modules, is crucial for the calibration. The basic assumption behind the Hand-Eye Calibration is that the distance between two cameras stays the same, no matter how the rig moves. Without the knowledge of the relative translational motion of the cameras the constraint is mostly useless.
The approach that handles these difficulties is the so-called Motion Adjustment (MA).[16] Motion adjustment simultaneously estimates the extrinsic parameters as well as the motion scales and is designed of being used for both offline and online calibration tasks.
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. Therefore, 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 sensor data of each single sensor, modelling the global system for each local module by propagating the locally computed state and uncertainty, and finally, fusing the redundant parameters to yield a unified, global model. For that purpose, each camera in the system is thought of as a processing unit which we call here a module $\mathcal{M}$. Each module $\mathcal{M}_i$ can communicate via a bus system with all other modules $\mathcal{M}_j$.
Based on SFM, HEC and MA, we can estimate the extrinsic position and orientation under certain circumstances - given the motion of each single camera. So, a robust motion estimation is the basis for this calibration approach. Hence, the solution of the calibration problem is basically devided into the problem of visual odometry and based on that the determination of the extrinsic parameters. Visual odometry is calculated via the so called sparse bundle adjustment (SBA)[6] for each camera based on corresponding image features in consecutive frames. To estimate the parameters continuously we embedded the HEC model into an Iterated Extended Kalman Filter (IEKF) which merges and filters parameter vectors. Furthermore, the IEKF calculates the uncertainties of the transformation parameters. These covariance matrices can be used for merging redundant information on a higher level.[17] By estimating the extrinsic parameters continuously the system is able to perform a self calibration.

This paper is structured as follows: Section 2 presents the basic techniques used for the motion-based Hand-Eye Calibration. Furthermore, the approach of motion adjustment for scaled motion is presented. Section 3 explains the propagation and fusion framework for arbitrary multi-camera setups and implementation details. In Section 4, experiments and results with simulated and real data are shown. Finally, Section 5 concludes and gives an outlook to further research.

2. EXTRINSIC SELF-CALIBRATION WITH NON-OVERLAPING FOV

2.1 Motion-based Hand-Eye Calibration

This Section shortly introduces 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

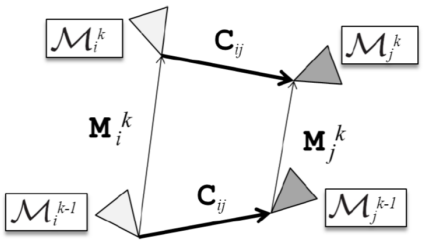

Figure 1. Basic geometric constellation for a 2-camera rig. $\mathtt{C}$ is the extrinsic calibration matrix, $\mathtt{M}$ is the camera motion.

transformation. The transformation between two camera modules $\mathcal{M}_i$ and $\mathcal{M}_j$ at time t is given by the transformation matrix

$$\mathtt{C}_{ij} = \begin{bmatrix} \mathbf{R}_{ij} & \mathbf{t}_{ij} \\ \mathbf{0}^T & 1 \end{bmatrix}_{4\times4}. \tag{1}$$

$\mathbf{R}(r_x, r_y, r_z) \in \mathbb{R}^{3\times3}$ is a rotation matrix with $\mathbf{R}^T\mathbf{R} = \mathbf{R}\mathbf{R}^T = \mathbf{I}$ and $\mathbf{t} = [t_x, t_y, t_z]^T$ is a translation vector. Hence, the inverse is given by $\mathtt{C}_{ij}^{-1} = \mathtt{C}_{ji}$. The Euler angles are represented by the vector $\mathbf{r} = [r_x, r_y, r_z]^T$. $\mathtt{C}$ is the extrinsic transformation with its parameters $\mathbf{t}$ and $\mathbf{r}$. The XZ-plane is considered as the ground plane and Y the longitudinal axis of the world coordinate system.
The motion of camera $\mathcal{M}_i$ between two time steps t and $t+1$ is given by

$$\mathtt{M}_{i_t} = \begin{bmatrix} \mathbf{\Omega}_{i_t} & \mathbf{v}_{i_t} \\ \mathbf{0}^T & 1 \end{bmatrix}_{4\times4}, \tag{2}$$

with rotation matrix $\mathbf{\Omega}(\omega_x, \omega_y, \omega_z) \in \mathbb{R}^{3\times3}$, translation vector $\mathbf{v} = [v_x, v_y, v_z]^T$ and the rotation parameters $\boldsymbol{\omega} = [\omega_x, \omega_y, \omega_z]^T$.

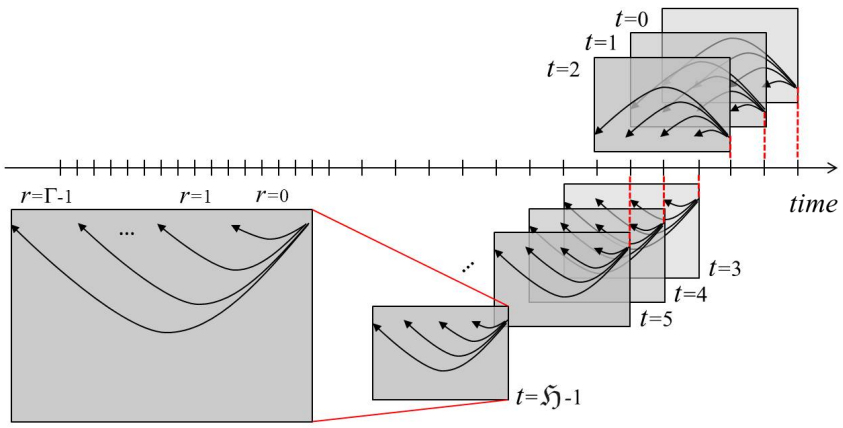

Figure 2. Bundle adjustment is calculated for a motion block of size Γ, motion adjustment is calculated over history of ϑ motion blocks. The motion history will be needed for the Motion Adjustment (Section 2.2)

The extrinsic parameters $\mathbf{R}$ and $\mathbf{t}$ between $\mathcal{M}_i$ and $\mathcal{M}_j$ can be expressed with respect to the cameras' motions (Fig. 1)

$$\mathtt{M}_i = \mathtt{C}_{ij}^{-1}\mathtt{M}_j\mathtt{C}_{ij} \tag{3}$$

which leads us with Eqn. 1 and 2 to the basic Hand-Eye Calibration equation

$$[\mathbf{I} - \mathbf{\Omega}_i]\mathbf{t}_{ij} + \mathbf{R}_{ij}\mathbf{v}_j - \mathbf{v}_i = \mathbf{0} \tag{4}$$

in dependence of the extrinsic translation $\mathbf{t}_{ij}$ and rotation $\mathbf{R}_{ij}$, which could be solved with any non-linear optimization method.
Fundamental for performing a motion-based hand-eye calibration is the motion estimation of the cameras. Consider, for

example, a vehicle driving around the corner with two cameras mounted, one on the left and one on the right side. Then surely the distance covered by the outer camera is longer than the distance of the inner camera. This is also the reason why odometric data from the vehicle is only little useful for hand-eye calibrations, as it does not cover the differences of the cameras' motion scales.
In our case, this is done using a standard bundle adjustment approach[6] by minimizing the projection error of a sparse set of points. significant image points are detected and tracked using a standard KLT-tracker.[18] The transformations as well as the scene coordinates are estimated within a block of Γ consecutive frames (Fig. 2). In our case, the intrinsic parameters are assumed to be known.

2.2 Motion Adjustment

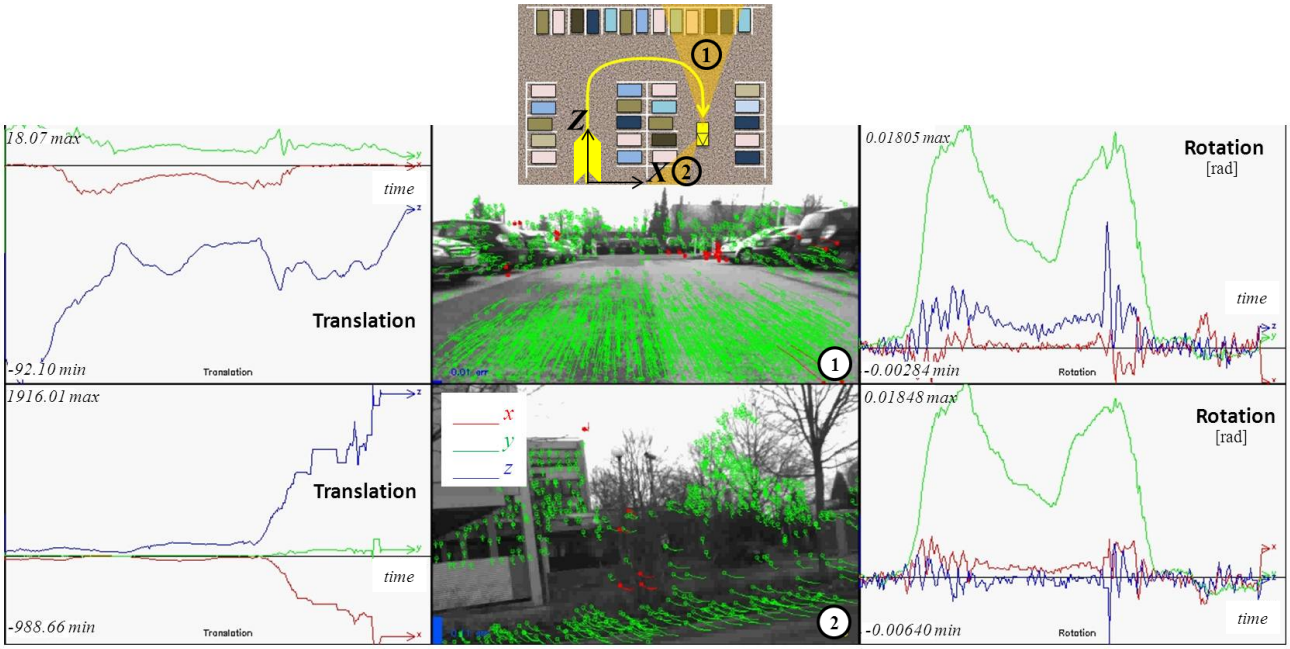

Figure 3. Estimated motion transformation parameters from our test sequence on a parking lot. Center top: Sketch of the scene and the vehicle-mounted cameras. The vehicle performs of a long U-turn. Center: Frames from the two cameras with optical flow vectors. Left: Continously estimated translation parameters t_x, t_y, t_z. Right: Continously estimated rotation parameters r_x, r_y, r_z. The translation is initially estimated only up to scale and for the rest of the sequence relative to the previous time step, and hence unitless.

Despite the fact that the motion parameters were predicted temporally (relative to the previous $\Gamma - 1$ estimations), the estimation of the camera modules' velocities can become unreliable over time as seen in Fig.3. However, the estimation of the translational motion is not entirely useless. As can be seen in Fig. 4, the scaled translation vectors reflect the car's motion adequatly.

The simple idea of motion adjustment is not only estimating the extrinsic parameters but also the scale factors of the normalized motion translations simultaneously. This leads us to the extended hand-eye calibration equation

$$[\mathbf{I} - \mathbf{\Omega}_{i_t}]\mathbf{t}_{ij} + \kappa_t \mathbf{R}_{ij}\mathbf{v}^s_{j_t} - \mathbf{v}^s_{i_t} = \mathbf{0} \tag{5}$$

with $\kappa_t \in \mathbb{R}$, $\mathbf{v}^s = \frac{\mathbf{v}}{\|\mathbf{v}\|}$ and $t = 1, ..., \mathfrak{H}$. Eq. 5 can then be minimized with respect to $\mathbf{R}_{ij}, \mathbf{t}_{ij}$ and κ_t with standard non-linear optimization techniques. Due to the scaled translational motion, the extrinsic translation can also only be estimated up to scale (Fig. 5).
It is well known and can easily be seen in Eq. 5 that rotational motion $\mathbf{\Omega}_{i_t} \neq \mathbf{I}$ is necessary to be able to estimate $\mathbf{t}_{ij}$.[13] Assuming $\mathbf{R}_{ij}$ to be known, we still need at least two different motions to solve Eq. 5 for $\mathbf{t}_{ij}$ and κ_t. Hence κ_t is estimated for a motion block of size $\Gamma > 1$ (Fig. 2). Within such a block the relative translational scales must be preserved. Fortunately, this is the case with a motion estimation via bundle adjustment with a consecutive block of frames. Furthermore, to preserve full rank of the linear equation system in Eq. 5, the rotation matrix $\mathbf{\Omega}$ needs to vary within a

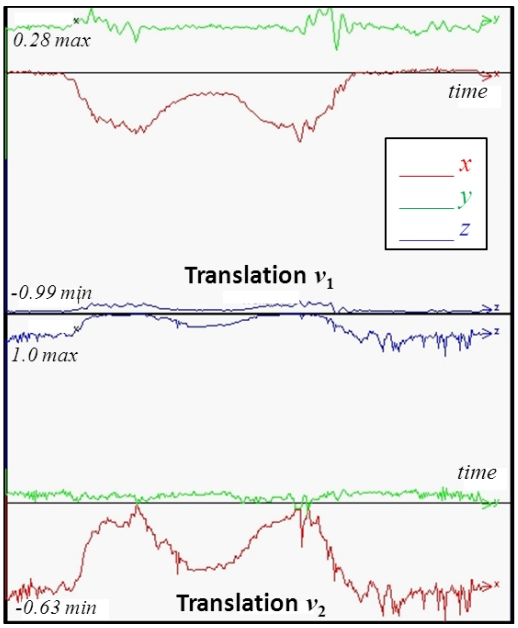

Figure 4. Scaled translation values of the motion parameters from Fig. 3 with $||\mathbf{v}_t|| = 1$.

motion block Γ, which means that moving in circles is an insufficient motion.
The problem of scaled motion adjustment is equivalent to the bundle adjustment used for simultaneous motion and structure estimation.[6] So, motion adjustment can be implemented quite efficiently. Because of the relation to the solving of sparse bundle adjustment, this approach is called sparse motion adjustment (SMA).[16]

As could be shown in,[16] the observability of the translational parameters using MA depends on the motion of the rig and the position of the cameras. Fig. 6 shows two examples of the resulting error heat maps for two differently scaled motions (but both with the same rotational motion). Surprisingly, the extrinsic position of the two cameras could not be calculated equally well at all positions around the center. We could observe that in cases with bad estimation results the scale factors κ_t could not be determined correctly.
Although such "bad" cases cannot be specified $a\ priori$, they can be detected by analyzing the result's covariance matrix. A principal component analysis (PCA) of the covariance matrix reveals the remaining uncertainty in the parameter space (see[16] for further details).

3. SYSTEM DESIGN

3.1 Multi-Camera Fusion

In this approach, both, the motion estimation via SBA as well as the extrinsic parameter estimation via SMA are bedded into an Iterated Extended Kalman Filter. This filtering fulfills two purposes. First, the parameter states can be tracked continously, which is useful for an online-calibration. And second, we get the uncertainty of the motion and calibration parameters in terms of covariance matrices. These covariance matrices can be further used for a propagation and fusion scheme that allows the holistic consideration of all modules' *local* estimations. This works as follows:
After each camera module has estimated its own egomotion (Fig. 7a), we can now determine the motion parameters of all other modules by using the calibration parameters and their covariance matrices (Fig. 7b). This is done by state and error propagation.[17] Consider a nonlinear function $g : \mathbb{R}^{n\times p} \longrightarrow \mathbb{R}^m$ with $\mathbf{y} = g(\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_p)$ and independent variables $\mathbf{x}_1, ..., \mathbf{x}_p$. Then the covariance of $\mathbf{y}$ can be approximated

$$\mathbf{\Sigma_y} = \sum_{i=1}^{p} \mathbf{J}_i \mathbf{\Sigma_{x_i}} \mathbf{J}_i^T. \tag{6}$$

Notice that even in the initial case when the extrinsic parameters are still unknown, the uncertainty can be set extremely high so that only the local estimations fall into account. Finally, each module can calculate a local estimation of the

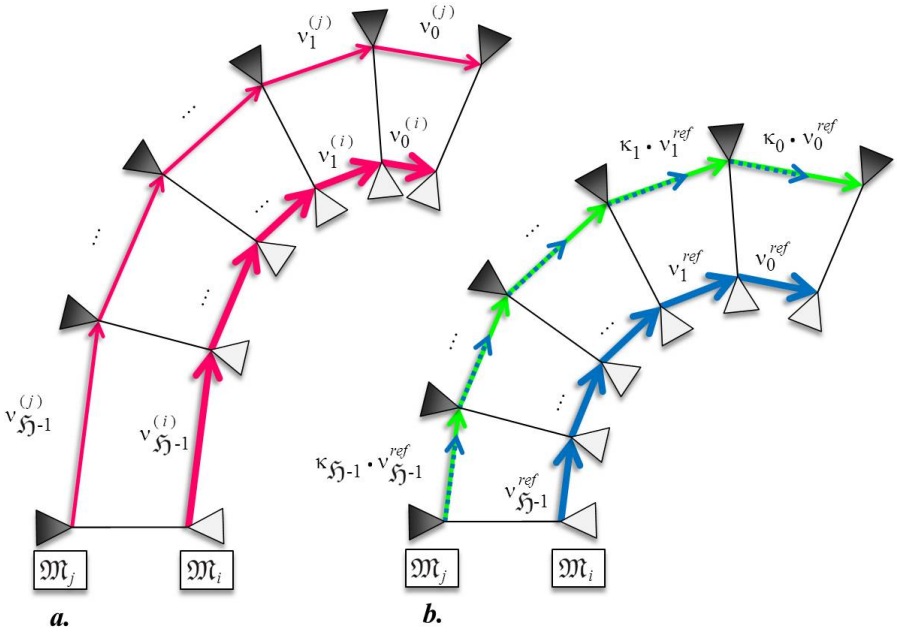

Figure 5. Motion adjustment scheme. *a.* Original trajectory (red) of two camera modules $\mathcal{M}_i, \mathcal{M}_j$ with unscaled velocity magnitudes $\nu_t^{\{i,j\}}$. *b.* First, the translation vectors are scaled to a reference magnitude ν_t^{ref} (blue/blue dotted), e. g., $\nu_t^{ref} = 1$. Afterwards the translation of $\mathcal{M}_j$ is rescaled to $\kappa \cdot \nu^{ref}$ (green) using the scales from the motion adjustment. The scales κ_t guarantee consistent motions with respect to the relative position $\mathbf{t}_{ij}$ and orientation $\mathbf{R}_{ij}$ of the cameras.

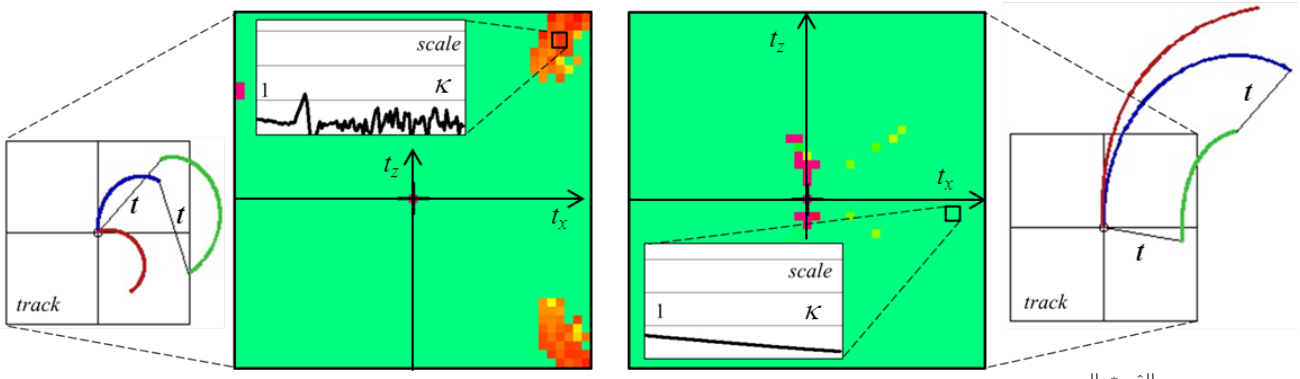

Figure 6. Error maps in dependence of the camera positions for differently scaled motions in a 2D-plane. $E = \frac{\|\mathbf{t}^* - \hat{\mathbf{t}}\|}{\|\mathbf{t}^*\|}$ with $\mathbf{t}^*$ is the ground truth value of the camera position and $\hat{\mathbf{t}}$ is the estimated translation. The reference camera $\mathcal{M}_1$ is always in the center, the position of the second camera $\mathcal{M}_2$ is within a square around the reference camera. Red means $E > 1$, green is $E = 0$. The blue track is the trajectory of $\mathcal{M}_1$ (each motion transformation is scaled to $||\mathbf{v}_{t,0}|| = \nu_{ref}$). The red track is the trajectory of the motion of $\mathcal{M}_2$ with relative scales $\hat{\kappa}_t = 1$ and $\hat{t}_x = \hat{t}_z = 0$. The blue and the red motions are the input data for the motion adjustment computation. The green track is the ground truth of the resulting trajectory of $\mathcal{M}_2$ with the correct relative scales κ_t^* and $\hat{t}_x = t_x^*, \hat{t}_z = t_z^*$. Left: Error map for a motion with small scale ($\nu_{ref} = 0.5$) and a detail view of the estimated motion scales with a resulting big error E. Right: Error map for a motion with big scale ($\nu_{ref} = 1.5$) and a detail view of the estimated motion scales with a small error E. The rotional motion in both heatmaps is the same with linearly increasing values of $\omega_y \in [0.01, 0.02]$ and $\mathfrak{H} = 100, \Gamma = 6$.

global model by considering uncertainties of the calibration and egomotion estimations (Fig. 7c). The propagation step is described in detail in Pagel et al.[17]

From the local propagation we have an estimation of each camera motion from each of the $\mathfrak{N}$ modules (Fig. 7d). These $\mathfrak{N}$ motion estimations per camera module are now merged to one. The fusion of the states $\mathbf{x}$ and covariances $\mathbf{\Sigma}$ can be done

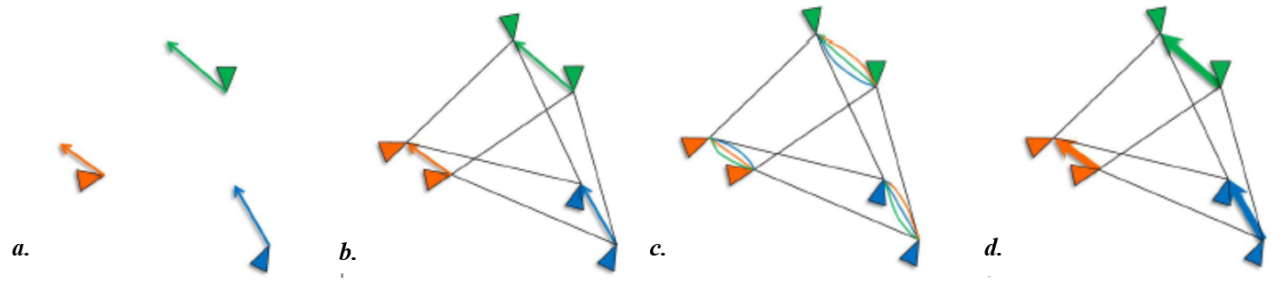

Figure 7. *a.* Each module performs a local motion estimation with a Kalman filter using the local sensor data and SBA. *b.* Extrinsic calibration parameters and their uncertainties are taken into account. *c.* Each modul $\mathcal{M}_i$ determines the motions of the other modules $\mathcal{M}_j$ based on the extrinsic parameters via state and error propagation. Hence, there are $\mathfrak{N}$ guesses for each of the $\mathfrak{N}$ motions by each module $modi$. *d.* The $\mathfrak{N}$ local guesses of each module's motion are merged to a global estimation.

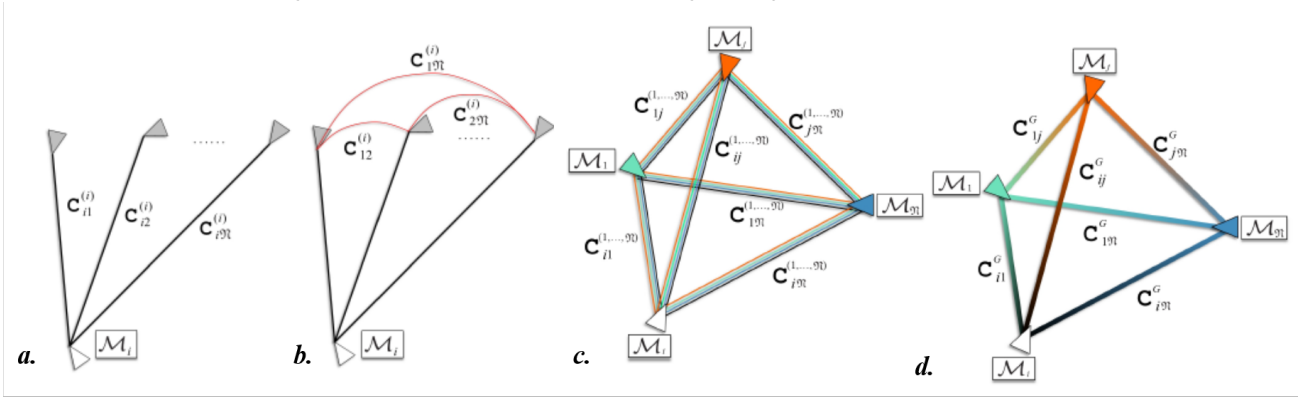

Figure 8. *a.* Local estimations of the extrinsic parameters between module $\mathcal{M}_i$ and $\mathcal{M}_{j\neq i}$ using the local motion parameters and an IEKF. *b.* Determination of the remaining calibration parameters in the camera network by state and error propagation based on the locally estimated calibration data. *c.* After propagation, there exist $k = 1, ..., \mathfrak{N}$ local guesses for each of the $\mathfrak{N}(\mathfrak{N}-1)/2$ extrinsic transformations by each module $\mathcal{M}_i$. *d.* The $\mathfrak{N}$ local guesses of each transformation are merged to a global transformation.

in a pairwise manner following the approach of Smith & Cheeseman:[19]

$$\mathbf{x}^G = \mathbf{x}_1 + \mathbf{\Sigma}_1 \cdot (\mathbf{\Sigma}_1 + \mathbf{\Sigma}_2)^{-1} \cdot (\mathbf{x}_2 - \mathbf{x}_1) \quad \text{and} \quad \mathbf{\Sigma}^G = \mathbf{\Sigma}_1 - \mathbf{\Sigma}_1 \cdot (\mathbf{\Sigma}_1 + \mathbf{\Sigma}_2)^{-1} \cdot \mathbf{\Sigma}_1 \tag{7}$$

where G indicates the merged ($= global$) parameters and uncertainties, respectively.
At this point, after the $\mathfrak{N}-1$ Kalman filter runs, each module $\mathcal{M}_i$ has $\mathfrak{N}-1$ estimations of the calibration parameters $\mathtt{C}_{ij}$. These $\mathfrak{N}-1$ transformations are sufficient to calculate the remaining transformations as follows:

$$\hat{\mathtt{C}}_{jl} = \mathtt{C}_{ij}^{-1}\mathtt{C}_{il} = \mathtt{C}_{ji}\mathtt{C}_{il} \tag{8}$$

These are $(\mathfrak{N}-1)(\mathfrak{N}-2)/2$ additional calculations. As a result, each module has a local guess for each of the $\mathfrak{N}(\mathfrak{N}-1)/2$ extrinsic parameters between the cameras. This means on the other hand that there are $\mathfrak{N}$ guesses for each extrinsic camera transformation (Fig. 8c). The covariances are propagated according to.[17] To merge these $\mathfrak{N}$ estimates per camera pair we can again proceed with Eq. 7 (Fig. 8d).

Here is a summary of the algorithm:

1. Track image features in each camera module $\mathcal{M}_i$ $(i = 1, ..., \mathfrak{N})$
2. Calculate SBA for each camera module $\mathcal{M}_i$ (with block size Γ, see Fig. 2)
3. Propagate $\mathfrak{N}$ local camera motion (block)s to all other modules
4. Merge all propagated motions and get (scaled) global motions

5. Calculate $\mathfrak{N} - 1$ local SMA procedures and get (scaled) local extrincics
6. Propagate local extrinsics to all other modules
7. Merge all propagated extrinsics and get (scaled) global motions

3.2 Implementation Details

As is shown in[13] there are some restrictions of the motion-based HEC approach according to the kind of motion. For example, as can be seen from eq. 4, if there is no rotational motion ($\mathbf{\Omega}_i = \mathbf{I}$), the extrinsic translation $\mathbf{t}_{ij}$ disappears from the equation and hence cannot be calculated. Also, we need more than one motion to build up an overdetermined linear equation system, which leads to a history of motions. It can be shown[13] that a purely circular driving under static conditions and hence fixed rotational parameters leads to an underdetermined equation system. Also, when the vehicle performs a purely planar motion the relative translational, longitudinal parameter t_y cannot be estimated anymore as there is no rotational motion around a horizontal axis.
In order to satisfy these restrictions this implementation performs an additional step where the ground plane is estimated explicitly. Once the transformation into the ground plane is known (which is obviously the same for all camera modules), motion transformations can be transformed into this 2D plane where a 2D-SMA is calculated by each module (see Fig. 9). Finally, the extrinsic parameters (transformations) can be back-transformed from the 2D-plane to the 3D space. The transformtion into the ground plane is based on the calculated 3D points from the Bundle Adjustment used for the motion estimation of each camera. A RANSAC procedure can be used to extract points from the ground plane. More details can be found in.[20]

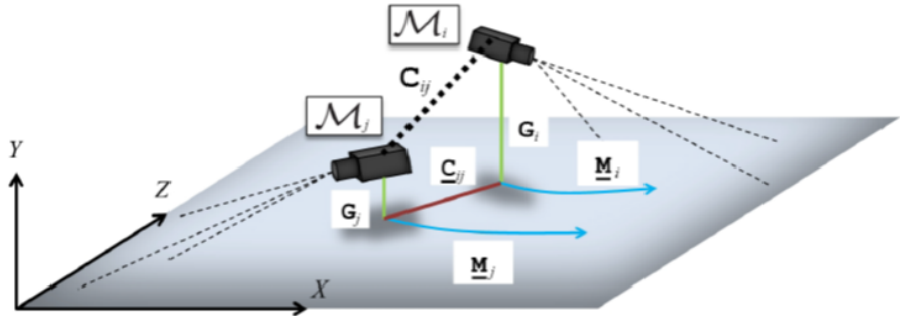

Figure 9. Using the ground plane to reduce the HEC problem from the 3D-space to the 2D-plane.

In the implementation we considered the circumstance that different kind of motions can be used to estimate different parameters. So, for example, in[16] straight (non-rotational) motion were used to initialize the cameras' orientations. Table 1 gives an overview of the different phases that were used to initialize the parameters of the system.

Motion Phase	Calculation of ...	Used for ...
1. Straight motion	the init. local motion estimations (pure translation) in the beginning	initializing motion estimation (SBA)
2. curve (rotational motion)	the rotation axes of all cameras[16]	calculating the transformation matrices in order to transform the motion transformations into the ground plane[20] and selecting points from the ground[20]
3. Straight motion	the transformations into the ground plane;[20] the relative camera orientations[16]	Motion Adjustment (MA/SMA); merging transformations[20]
4. curve (rotational motion)	Motion Adjustment (MA/SMA) state filtering (IEKF)	estimating the extrinsic calibration parameters; propagation and fusion

Table 1. Motion phases of the calibration system. The first three phases are for initializing the system. From phase 4 on the state parameters (motion and extrinsics) can be estimated continously with the IEKF.

4. EXPERIMENTS AND RESULTS

4.1 Simulations

In order to demonstrate the general capability of this global, motion-based calibration approach a two-camera rig was simulated. The platform performed a helix-like motion (linearly increasing steering angle, constant velocity) (Fig. 10). 3D-points were equally distributed in space. Their projections (which are the basis for the motion estimation and hence the calibration) were normally distributed with 0.5 pixel variance. Additionally, 10% outliers were added. The setting parameters used were $\mathfrak{H} = 100, \Gamma = 5$ and $\mathfrak{P} = 300$ points for the KLT-tracker. Fig. 11 and show the results. The extrinsic parameters converge towards the ground truth values.

4.2 Real Data Experiments

In order to test the calibration approach with real data, a 3-camera rig was implemented on a test vehicle. The cameras were calibrated (intrinsically as well as extrinsically) with a standard pattern-based approach,[21][22] (Fig. 12). These calibration parameters serve as ground truth for the evaluation and are shown in Table 2. The reference calibration had a remaining average projection error of 1.7 pixels. The three cameras are labeled $\mathcal{M}_1, \mathcal{M}_2$ and $\mathcal{M}_3$. As no absolute reference like odometry or GPS was available, the extrinsic distances between the cameras are all scaled relative to the translation vector $\mathbf{t}_{12}$ between $\{\mathcal{M}_1, \mathcal{M}_2\}$. The longitudinal parameter t_y was neglected (see[20] for discussion).

pair	t_x	t_y	t_z	$[r_x] =$ rad	$[r_y] =$ rad	$[r_z] =$ rad
$1 \rightarrow 2$	-0,39	0,16	-0,91	-0,339	-2,612	0,074
$1 \rightarrow 3$	0,71	0,22	-1,19	-0,448	2,441	0,178

Table 2. Reference extrinsic calibration parameters. Translation vectors are scaled relative to $\mathbf{t}_{12}$ with $\|\mathbf{t}_{12}\| = 1$ and hence have no unit.

As all cameras were coplanar, the parameter t_y was fixed during the estimation process, which means only $r_x,\ r_y,\ r_z$ and the (relative) extrinsic parameters $t_x,\ t_z$ were estimated during the tests. Fig. 13 shows the test-trajectory of the vehicle. The setting parameters were $\mathfrak{H} = 100,\ \Gamma = 5,\ \mathfrak{P} = 300$ with the intitial state parameters $\mathbf{t}_{ij} = \mathbf{r}_{ij} = \mathbf{0}$. The covariance matrices were initialized with very big values.

Fig. 14 and 15 show the estimated and merged (=global) motion parameters of the three cameras as well as the estimated extrinsic parameters. It is clearly visible that the calibration process starts after the second straight motion part according to the motion phases in Table 1. Fig. 16 shows the capability of the propagation and fusion approach.

Fig. 17 shows an example application of the resulting calibration parameters. Once the extrinsic parameters as well as the ground plane (from the reference calibration) are known, one can transform the camera images into the ground plane and generate a bird's eye view.[16]

Experiments with real data were performed on 7 sequences. Overall, compared to the reference calibration, a mean error of all estimated orientation angles of 0.8° and a mean translational error of 10.8% remained. The relatively high translational error is due to the dependency between camera position and motion (see Section 2.2 or[16]). Here, odometric data could be a very valuable sensor information to merge with the image information in order to yield more accurate results.

5. CONCLUSION

This paper presented a completely vision-based approach for self-calibrating a camera rig with non-overlapping FOV using onloy the motion of the camera. For a purely vision-based motion estimation, which estimates the motion of each camera only up to scale, the concept of motion adjustment was introduced. Furthermore, to use the information of multiple sensors in a multi-camera rig, a global fusion system was shown and demonstrated in simulated and real data with a reference calibration. The experiments yielded very good results in the orientation and respectable results in the scaled translational parameters. Also, the capability of the global fusion scheme could be shown.
It turned out, that without further sensor information or restrictions, no absolute translational calibration parameters could be estimated. So, future research focuses on the multi sensor fusion with vehicle data including odometry and GPS data.

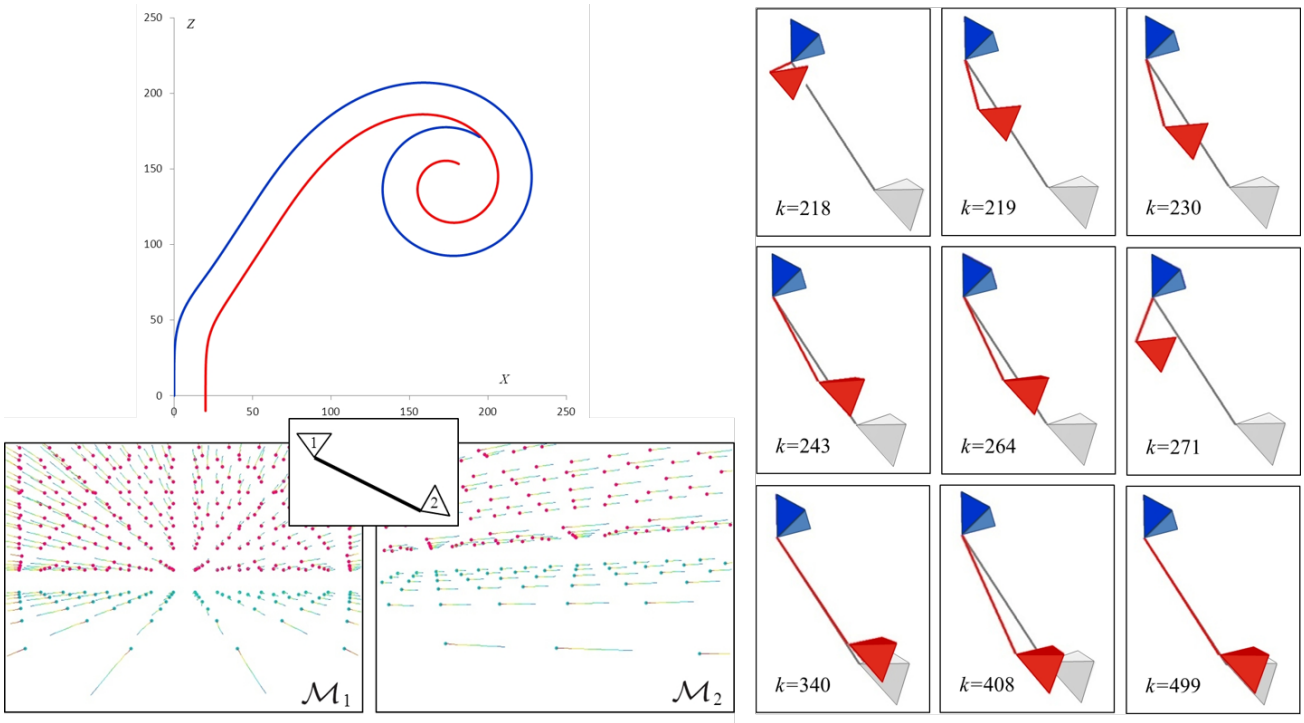

Figure 10. Left, Bottom: Simulated flow vectors of two camera modules. Left, Top: Helix-like motion of the two cameras. The first curve initializes the rotation axes, then the camera orientations are intialized. Right: Perspective representation of the results from Fig. 11 (ground truth = grey.

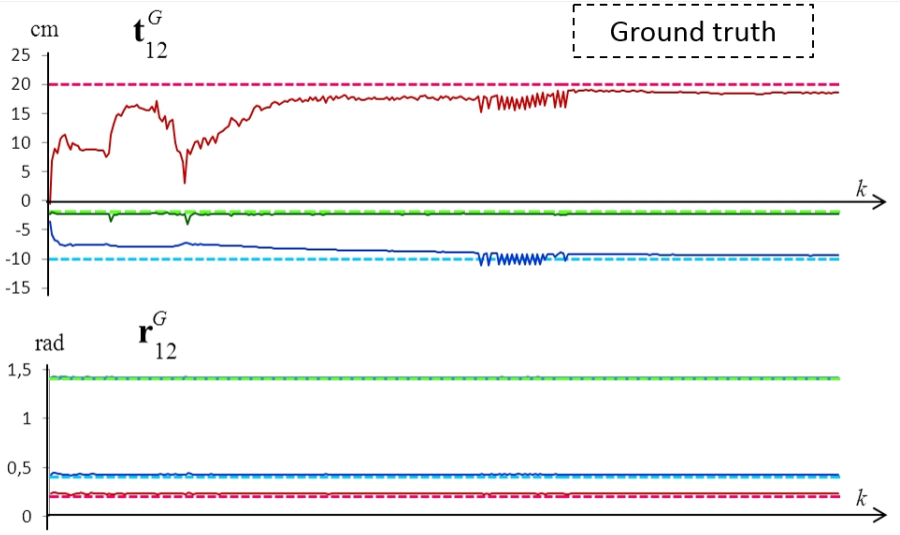

Figure 11. Result of the Helix-Simulation: Estimated extrinsic parameters (translation $\mathbf{t}$ and euler angles $\mathbf{r}$) of modules $\mathcal{M}_1$ and $\mathcal{M}_2$.

REFERENCES

[1] Clipp, B., Kim, J., Frahm, J., Pollefeys, M., and Hartley, R., "Robust 6dof motion estimation for non-overlapping, multi-camera systems," *Proceedings of the IEEE Workshop on Applications of Computer Vision* , 1–8 (2008).

[2] Mordohai, A., Akbarzadeh, A., Frahm, J., Mordohai, P., Engels, C., Gallup, D., Merrell, P., Phelps, M., Sinha, S., Talton, B., Wang, L., Yang, Q., Stewenius, H., Yang, R., Welch, G., Towles, H., Nister, D., and Pollefeys, M., "Towards urban 3d reconstruction from video," *3DPVT* , 1–8 (2006).

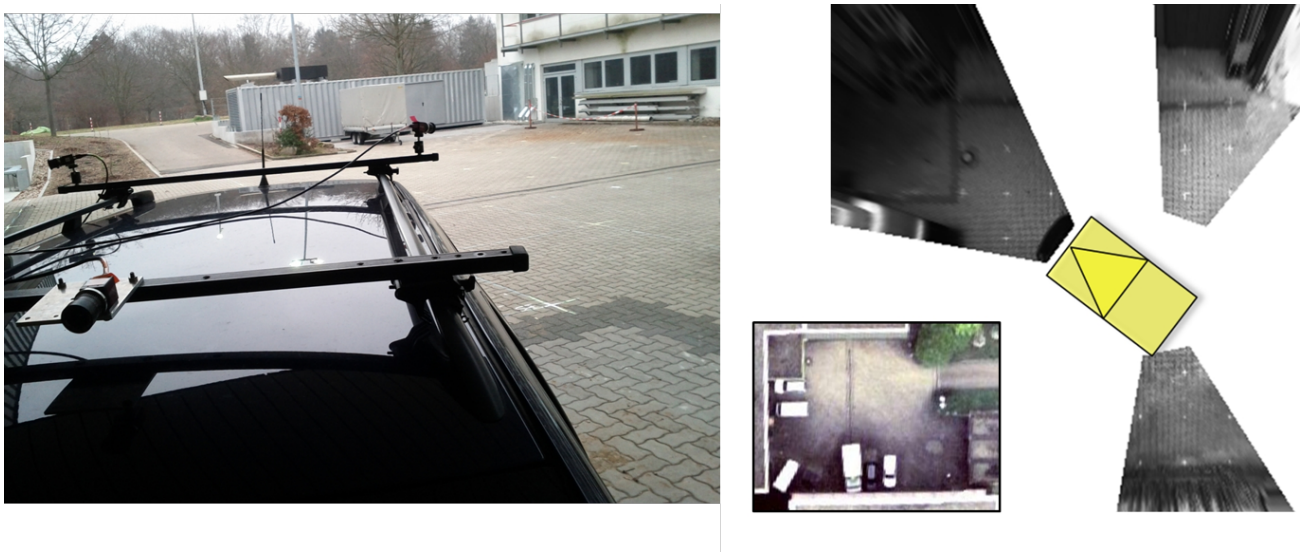

Figure 12. Left: Vehicle with 3-camera rig. Center, Bottom: Top view of the calibration site. Right: Bird's eye view after the pattern-based offline calibration (the markings are still visible on the ground)

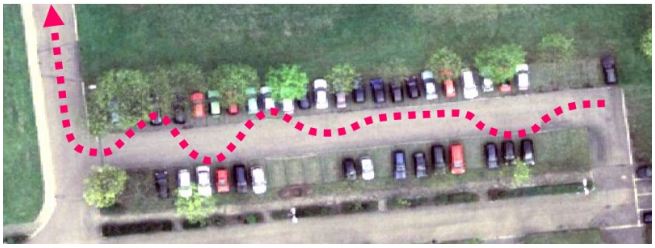

Figure 13. Test ride according to the motion scheme from Table 1.

[3] Nister, D., Naroditsky, O., and Bergen, J., "Visual odometry," *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition* , 652–659 (2004).

[4] Davison, A., Reid, I., Molton, N., and Stasse, O., "Monoslam: Real-time single camera slam," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **29**(6), 1052–1067 (2007).

[5] Newcombe, R. and Davison, A., "Live dense reconstruction with a single moving camera," *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition* , 1498–1505 (2010).

[6] Lourakis, M. and Argyros, A., "Sba: A software package for generic sparse bundle adjustment," *ACM Transactions on Mathematical Software* **36**, 1–30 (2009).

[7] Pagel, F., "Robust monocular egomotion estimation based on an iekf," *Proceedings of the Canadian Conference on Computer and Robot Vision, Kelowna, B.C, Kanada* , 213–220 (2009).

[8] Dang, T., Hoffmann, C., and Stiller, C., "Continuous stereo self-calibration by camera parameter tracking," *IEEE Transactions on Image Processing* **18**, 1536–1550 (2009).

[9] Lamprecht, B., Rass, S., Fuchs, S., and Kyamakya, K., "Extrinsic camera calibration for an on-board two camera system without overlapping field of view," *Proceedings of the IEEE Intelligent Transportation Systems Conference* , 265–270 (2007).

[10] Carrera, G., Angeli, A., and Davison, A., "Slam-based automatic extrinsic calibration of a multi-camera rig," *Proceedings of the IEEE International Conference on Robotics and Automation* , 2652–2659 (2011).

[11] Lebraly, P., Royer, E., Ait-Aider, O., Deymier, C., and Dhome, M., "Fast calibration of embedded non-overlapping cameras," *Proceedings of the IEEE International Conference on Robotics and Automation* , 221–227 (2011).

[12] Horaud, R. and Dornaika, F., "Hand-eye calibration," *International Journal of Robotic Research* **14**(3), 195–210 (1995).

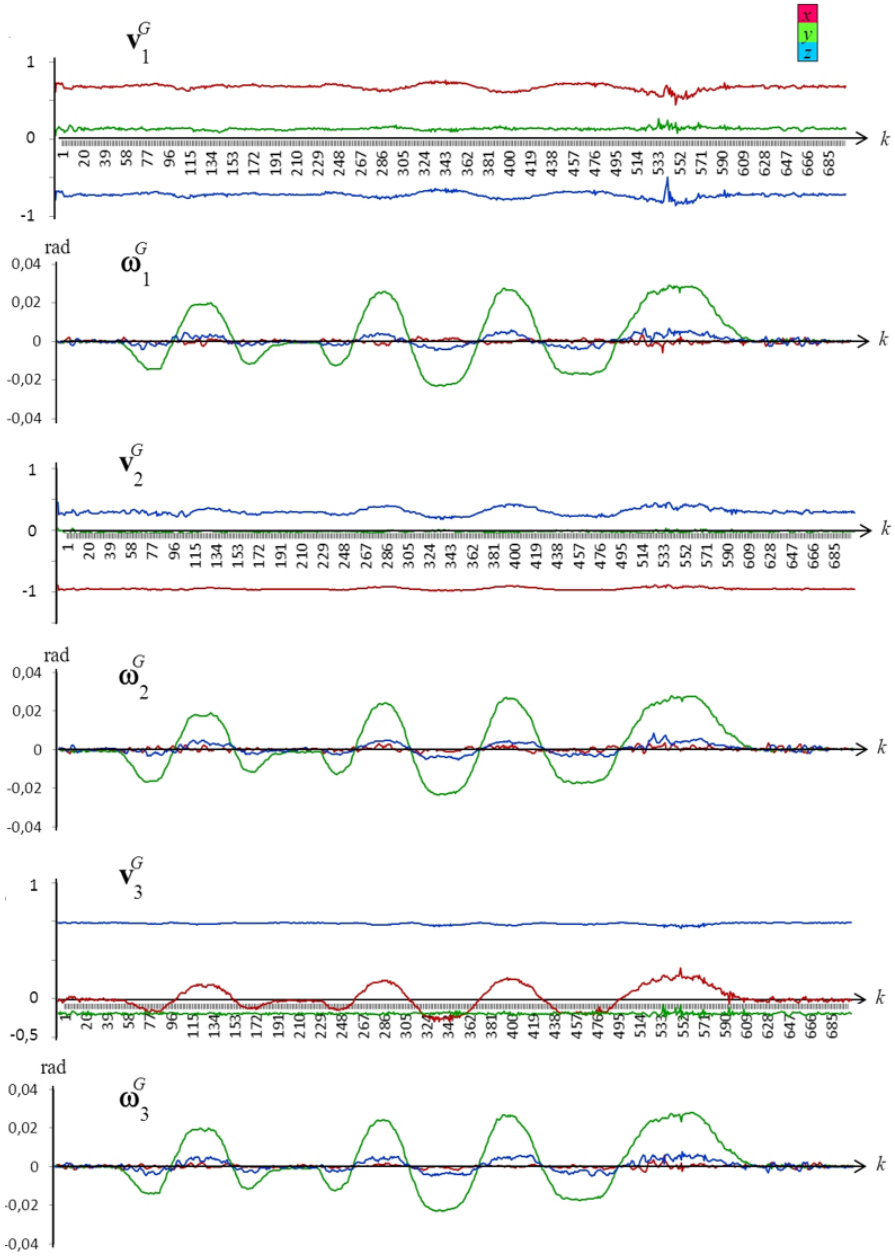

Figure 14. Estimated (global) motions of the three camera modules $\mathcal{M}_1, \mathcal{M}_2, \mathcal{M}_3$ with scaled translation vectors so that $\left\|\mathbf{v}^G\right\| = 1$. The rotational parameters are very similar to each other (which they are supposed to be) and they reflect the trajectory shown in Fig. 13.

[13] Tsai, R. and Lenz, R., "A new technique for fully autonomous and efficient 3d robotics hand/eye calibration," *IEEE Transactions on Robotics and Automation* **5**(3), 345–358 (1989).

[14] Esquivel, S., Wölk, F., and Koch, R., "Calibration of a multi-camera rig from non-overlapping views," *Proceedings of the DAGM-Symposium* (2007).

[15] Ruland, T., Loose, H., Pajdla, T., and Krüger, L., "Hand-eye autocalibration of camera positions on vehicles," *Proceedings of the IEEE Intelligent Transportation Systems Conference* , 367–372 (2010).

[16] Pagel, F., "Motion adjustment for extrinsic calibration of cameras with non-overlapping views," *Proceedings of the Canadian Conference on Computer and Robot Vision, Toronto, Ontario, Kanada* , 94–100 (2012).

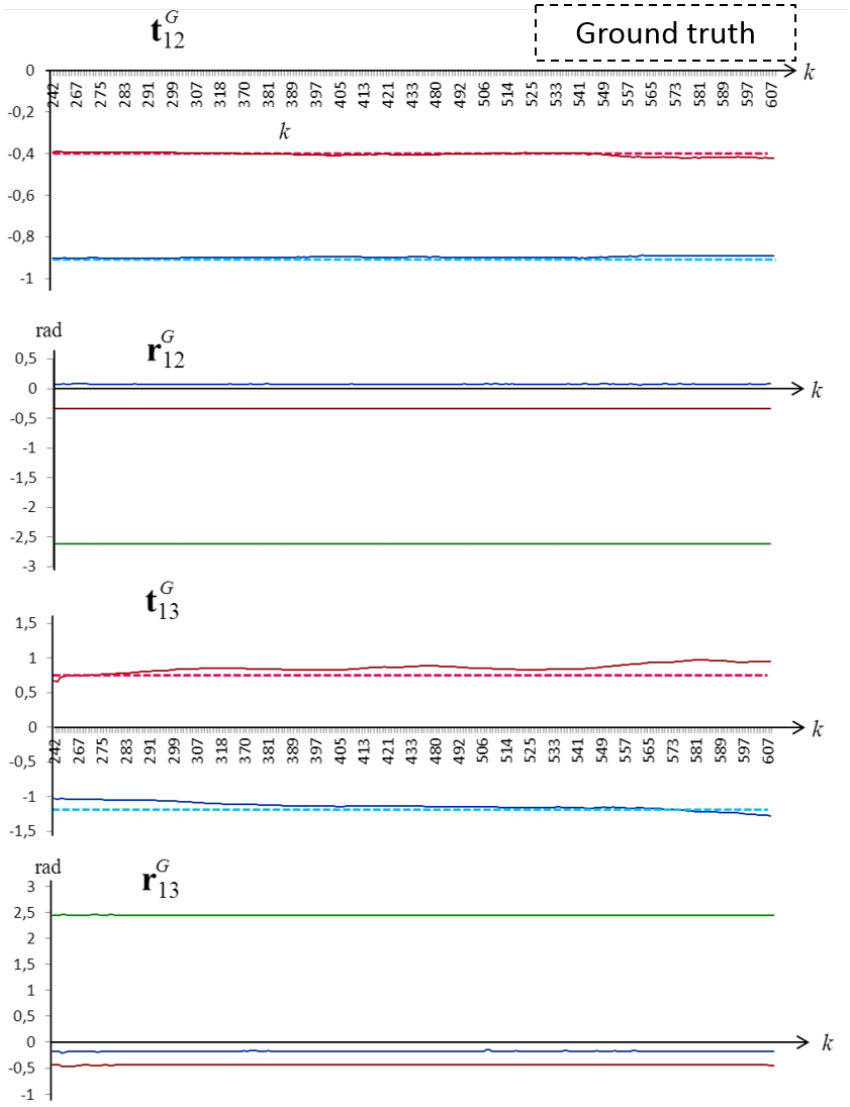

Figure 15. Estimated extrinsic parameters between modules $\mathcal{M}_1, \mathcal{M}_2$ and $\mathcal{M}_1, \mathcal{M}_3$. The longitudinal parameter t_y was neglected. The translation vector $\mathbf{t}_{13}^G$ is scaled relative to $\mathbf{t}_{12}^G$ with $\left\|\mathbf{t}_{12}^G\right\| = 1$. The orientation angles $\mathbf{r}_{12}^G$ and $\mathbf{r}_{13}^G$ could be estimated very robustly, also $\mathbf{t}_{12}^G$. $\mathbf{t}_{13}^G$ is slightly differing.

[17] Pagel, F. and Willersinn, D., "Motion-based online calibration for non-overlapping camera views," *Proceedings of the IEEE Intelligent Transportation Systems Conference, Madeira, Spanien* , 843–848 (2010).

[18] Tomasi, C. and Kanade, T., "Detection and tracking of point features," *Tech. Report CMU-CS-91-132, Carnegie Mellon University* (1991).

[19] Smith, R. and Cheeseman, P., "On the representation and estimation of spatial uncertainty," *International Journal of Robotics Research* **5**(4) (1986).

[20] Pagel, F. and Willersinn, D., "Extrinsic camera calibration in vehicles with explicit ground estimation," *Proceedings of the International Workshop on Intelligent Transportation, Hamburg* (2011).

[21] Zhang, Z., "A flexible new technique for camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22**(11), 1330–1334 (2000).

[22] Pagel, F., "Calibration of non-overlapping cameras in vehicles," *IEEE Intelligent Vehicles Symposium* , 1178–1183 (2010).

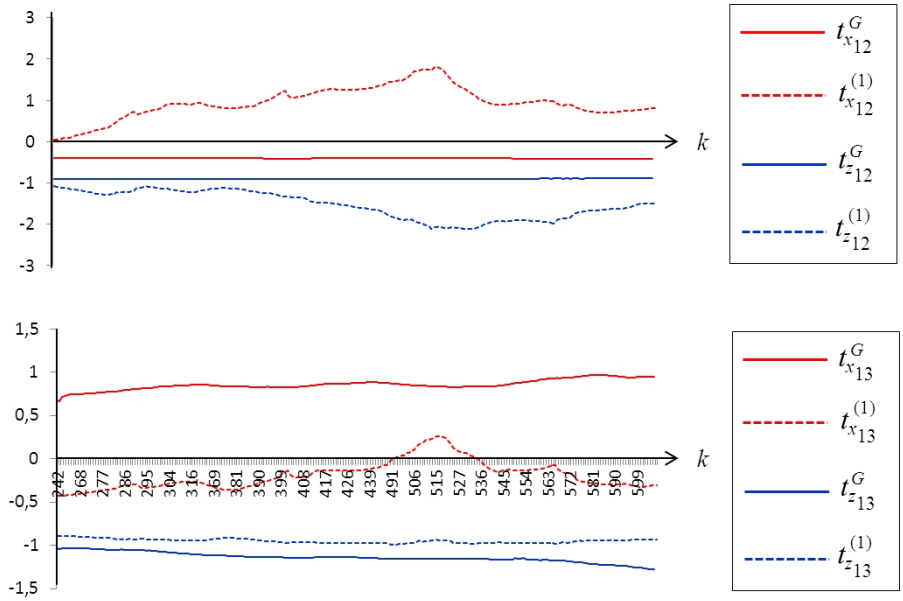

Figure 16. Comparison of the estimated extrinsic parameters $\mathbf{t}_{12}^{G}$ und $\mathbf{t}_{13}^{G}$ with (thin line) and without the global propagation and fusion approach (dotted line = average of all local estimations). A comparison with Fig. 15 shows, that the global estimation is far closer to the real parameters than the local estimations. Furthermore, the non-merged parameter estimations are much more fluctuating.

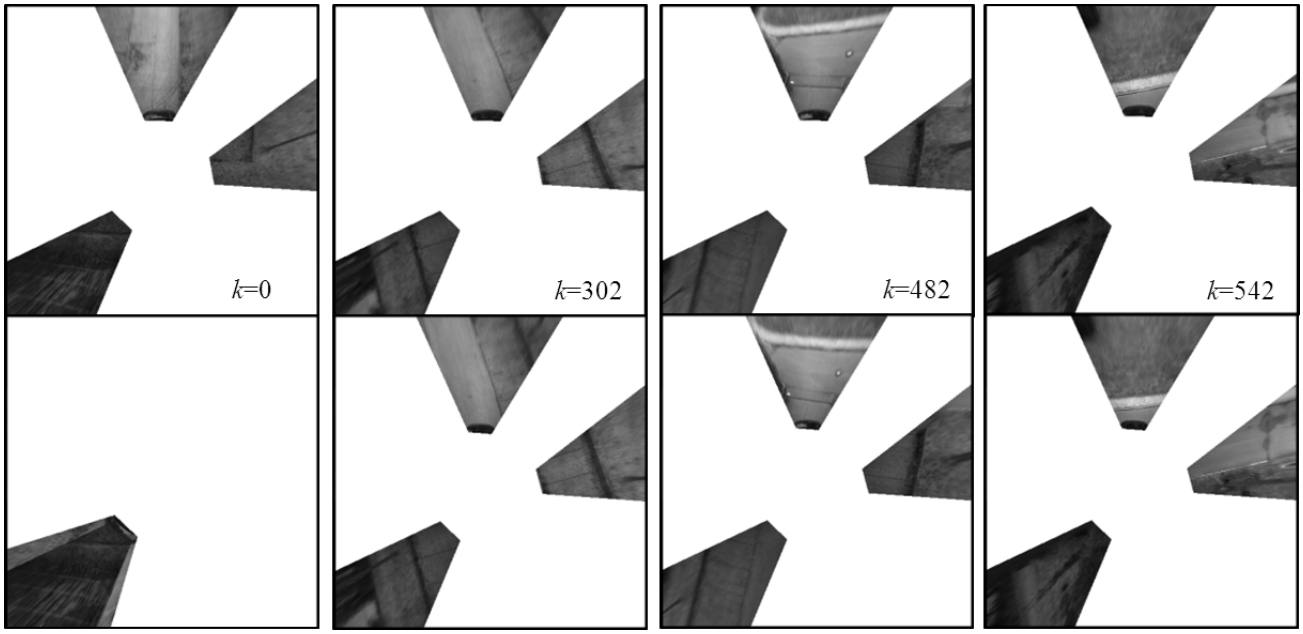

Figure 17. Calculated bird's eye view with the extrinsic parameters from the reference calibration (top) and the continuously estimated extrinsic parameters, following the global, motion-based approach (bottom). The lower left image shows the initial camera configuration (= 0). Ideally, the upper and lower row look exaclty the same. Comparing the positions and orientations of the cameras' FOV allows a qualitative evaluation of the resulting parameters.