Real-Time 6D Stereo Visual Odometry with Non-Overlapping Fields of View

Tim Kazik[1], Laurent Kneip[1], Janosch Nikolic[1], Marc Pollefeys[2] and Roland Siegwart[1]

[1]Autonomous Systems Lab
ETH Zurich, Switzerland

[2]Computer Vision and Geometry Group
ETH Zurich, Switzerland

Abstract

In this paper, we present a framework for 6D absolute scale motion and structure estimation of a multi-camera system in challenging indoor environments. It operates in real-time and employs information from two cameras with non-overlapping fields of view. Monocular Visual Odometry supplying up-to-scale 6D motion information is carried out in each of the cameras, and the metric scale is recovered via a linear solution by imposing the known static transformation between both sensors. The redundancy in the motion estimates is finally exploited by a statistical fusion to an optimal 6D metric result. The proposed technique is robust to outliers and able to continuously deliver a reasonable measurement of the scale factor. The quality of the framework is demonstrated by a concise evaluation on indoor datasets, including a comparison to accurate ground truth data provided by an external motion tracking system.

1. Introduction

During the last decade, the computer vision community investigated vision based motion estimation to a great extent. Nistér *et al*. [20] proposed Visual Odometry (VO) frameworks for both monocular and stereo setups. The stereo configuration not only renders the motion estimation more robust, but also allows inference of metric scale. The field of view (FOV) in a classical stereo setup is however limited as both cameras are required to observe the same scene. We introduce a method which relaxes this constraint, and allows the cameras to perceive different scenes while still operating in absolute scale. The extended field of view is especially beneficial in poorly textured environments. Our approach is backed by automotive vision system designers who increasingly propose almost non-overlapping settings with cameras placed in the side mirrors of the car.

The proposed method operates in real-time and employs information from two cameras with non-overlapping fields of view. The motion of the rig is estimated by first performing monocular VO in each camera individually. The two motion estimates are then used to derive the absolute scale by enforcing the known rigid relative placement of the two cameras. The estimation of the scale is robustified by applying a RANSAC [6] scheme to a windowed buffer of several recent frames, which removes degenerate constraints for scale estimation. In a final step, we fuse the two motion estimates to an optimal motion of the entire multi-camera system by taking also the uncertainties of the individual pose computations into account. Fig. 1 shows a trajectory and scene reconstructed by the presented algorithm.

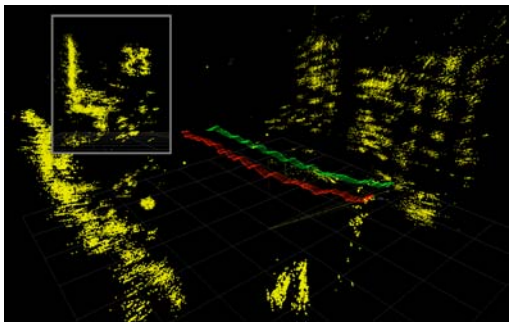

Figure 1. Map and trajectory generated by our VO system. The right (red) and left (green) camera face opposite directions, and the small box shows a side view on the reconstruction of the right scene (vertical wall with office desks in front).

The remainder of this paper is organized as follows: Next, we discuss previous work on related approaches for motion estimation and present a brief overview of our system. Section 2 then introduces our approach for metric scale estimation, while section 3 focuses on the fusion of two individual monocular VO estimates. Section 4 finally outlines the conducted experiments, before concluding the work in section 5.

1.1. Related Work

It is a well-known fact that a single approximatively central camera can reconstruct a given scene only up to scale [10]. In order to estimate the motion in metric scale, supplementary information is required typically given by an ad-

978-1-4673-1228-8/12/$31.00 ©2012 IEEE

ditional sensor such as a range or inertial measurement unit, e.g. [21].

In this work, we focus on metric motion estimation by means of an additional camera. Rigs consisting of multiple cameras not only enhance constraints in pose estimation problems, but also allow a larger FOV, which in turn improves robustness. Various configurations have been presented over the years: The classical stereo setup [20], [16] with a large common FOV exploits the known baseline between the cameras in order to derive the metric scale. In [5], the authors relax the constraint of having a large common FOV and present a system with only minimal overlap in the FOV. Lately, approaches have been proposed which do not require the cameras to see the same scene at all.

Metric motion estimation by means of multiple non-overlapping cameras can be approached in two ways. The cameras are treated either individually, or as one single camera using the Generalized Camera Model (GCM) [8]. Successful solutions have been presented for both approaches:

Multiple Individual Cameras: Clipp *et al*. [4] introduce a method for full 6 DOF motion estimation by performing monocular VO in one camera, and determining the scale by a single point correspondence extracted from the second camera. Although applying a RANSAC scheme, the scale can only be computed at certain instances and shows a significant variance. In [14], the authors show how to transform the motion estimation into a triangulation problem, which is subsequently solved using Second-Order Cone Programming. Kim *et al*. [13] suggest to solve the triangulation by means of a Linear Programming based Branch and Bound algorithm. Although computation time is reduced, the method is still not real-time compliant.

Multiple Cameras Described by the GCM: The authors of [18] and [13] propose approaches to motion estimation using the linear seventeen-point algorithm [22]. Degenerate cases as shown in [12] are taken into account. Mouragnon *et al*. [19] present the first real-time capable VO system using the GCM. The work contains results using perspective, stereo and catadioptric cameras, and experiments on more complex multi-camera systems are left as future work. The minimal solution to the relative pose problem using the GCM is presented in [25], where only 6 corresponding image rays are required. It can be adapted to the non-overlapping stereo case, however leads to 64 solutions for the relative transformation and thus represents a computationally inefficient approach.

Both approaches employ joint information from multiple cameras. The fact that we compute visual odometry with relative scale propagation in each camera individually has the advantage of giving the information from both cameras equal importance and potentially overcoming short sequences where one of the cameras fails. Moreover, it also allows an efficient distributed computation.

1.2. System Overview

The presented motion estimation pipeline is summarized as a flowchart in Fig. 2. The transformations of the left (L) and right (R) camera are denoted by the euclidean transformation matrices $\mathbf{T}$ (expressing translation and rotation), the scales of the monocular VOs are indicated by λ and μ, and a fused quantity is denoted by an asterisk. The scale estimation and fusion modules are explained in the following.

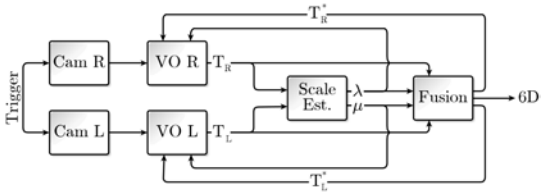

Figure 2. System overview

2. Metric Scale Estimation

The fact that both cameras are rigidly mounted on the same object can be exploited in order to infer metric scale. We first show how the metric scale is derived from a single rig displacement, then improve the condition of the problem by considering multiple constraints inside a windowed buffer of multiple frames, and finally apply a RANSAC scheme in order to enhance robustness.

2.1. Rig Constraint

The constraint expressing the static coupling of the cameras originates from 'hand-eye' calibration [11] and links the two cameras at two different time steps t_1 and t_2. It requires the concatenation of the transformations of right c.p. (camera position) at t_1 to right c.p. at t_2 ($\mathbf{T}_{\mathrm{R2R1}}$) and right c.p. at t_2 to left c.p. at t_2 ($\mathbf{T}_{\mathrm{LR}}$) to be equal to the concatenation of the transformations of right c.p. at t_1 to left c.p. at t_1 ($\mathbf{T}_{\mathrm{LR}}$) and left c.p. at t_1 to left c.p. at t_2 ($\mathbf{T}_{\mathrm{L2L1}}$). Note that the static transformation of right c.p. to left c.p. is assumed to be known from a calibration process, and therefore does not need subscripts indicating time indices. Fig. 3 illustrates these four transformations and the above mentioned equivalence.

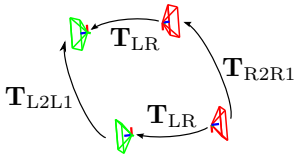

Figure 3. Rig constraint linking the euclidean transformations.

The above constraint translates into the algebraic expression

$$\mathbf{T}_{\mathrm{L2L1}}\,\mathbf{T}_{\mathrm{LR}} \;=\; \mathbf{T}_{\mathrm{LR}}\,\mathbf{T}_{\mathrm{R2R1}}. \qquad (1)$$

Expanding the euclidean transformations and introducing the unknown scale factors λ and μ for the right and left camera respectively, we obtain

$$\begin{bmatrix} \mathbf{R}_{\mathrm{L2L1}} & \mu\ {}_{\mathrm{L2}}\mathbf{t}_{\mathrm{L2L1}} \\ \mathbf{0}_{1\mathrm{x}3} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R}_{\mathrm{LR}} & {}_{\mathrm{L}}\mathbf{t}_{\mathrm{LR}} \\ \mathbf{0}_{1\mathrm{x}3} & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{\mathrm{LR}} & {}_{\mathrm{L}}\mathbf{t}_{\mathrm{LR}} \\ \mathbf{0}_{1\mathrm{x}3} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R}_{\mathrm{R2R1}} & \lambda\ {}_{\mathrm{R2}}\mathbf{t}_{\mathrm{R2R1}} \\ \mathbf{0}_{1\mathrm{x}3} & 1 \end{bmatrix}. \tag{2}$$

Inspection of (2) reveals that the rotational and translational parts can be decoupled, which leads to

$$\begin{aligned} \mathbf{R}_{\mathrm{L2L1}}\ \mathbf{R}_{\mathrm{LR}} &= \mathbf{R}_{\mathrm{LR}}\ \mathbf{R}_{\mathrm{R2R1}} \\ \mathbf{R}_{\mathrm{L2L1}}\ {}_{\mathrm{L}}\mathbf{t}_{\mathrm{LR}} + \mu\ {}_{\mathrm{L2}}\mathbf{t}_{\mathrm{L2L1}} &= \mathbf{R}_{\mathrm{LR}}\ \lambda\ {}_{\mathrm{R2}}\mathbf{t}_{\mathrm{R2R1}} + {}_{\mathrm{L}}\mathbf{t}_{\mathrm{LR}}. \end{aligned} \tag{3}$$

Solving the translational part of the above expression for the scales yields

$$\underbrace{\begin{bmatrix} \mathbf{R}_{\mathrm{LR}}\ {}_{\mathrm{R2}}\mathbf{t}_{\mathrm{R2R1}} & -{}_{\mathrm{L2}}\mathbf{t}_{\mathrm{L2L1}} \end{bmatrix}}_{:=\mathbf{A}_{\mathrm{i}}} \underbrace{\begin{bmatrix} \lambda \\ \mu \end{bmatrix}}_{:=\mathbf{x}_{\mathrm{i}}} = \underbrace{(\mathbf{R}_{\mathrm{L2L1}} - \mathbf{I}_{3\mathrm{x}3})\ {}_{\mathrm{L}}\mathbf{t}_{\mathrm{LR}}}_{:=\mathbf{b}_{\mathrm{i}}}. \tag{4}$$

With the above expression, the absolute scale of each monocular VO can be derived by means of a linear least squares (LS) scheme using the relative transformation of the cameras between two different time steps only.

2.2. Multi-Frame Window

In order to improve the condition of the LS problem in (4), we compute the scales not only on two consecutive images, but over several recent frames. Each of the monocular VOs is performing internal relative scale propagation, and hence the scales are drifting only slowly. We thus make the assumption that the scales are locally constant. We then consider multiple constraints inside a sliding window taking the last N poses into account, as illustrated in Fig. 4. Within this window, the rigid transformation constraint needs to be constantly fulfilled for same scale factor values.

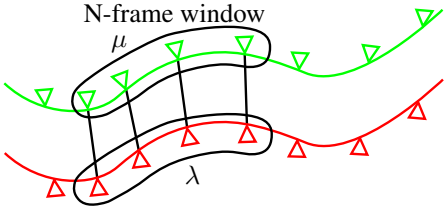

Figure 4. Multi-frame buffer spanning the last N camera poses.

Stacking multiple incremental constraints together leads to the augmented LS problem

$$\underbrace{\begin{bmatrix} \mathbf{R}_{\mathrm{LR}}\ {}^{1}_{\mathrm{R2}}\mathbf{t}_{\mathrm{R2R1}} & -{}^{1}_{\mathrm{L2}}\mathbf{t}_{\mathrm{L2L1}} \\ \mathbf{R}_{\mathrm{LR}}\ {}^{2}_{\mathrm{R2}}\mathbf{t}_{\mathrm{R2R1}} & -{}^{2}_{\mathrm{L2}}\mathbf{t}_{\mathrm{L2L1}} \\ \cdots & \cdots \\ \mathbf{R}_{\mathrm{LR}}\ {}^{N}_{\mathrm{R2}}\mathbf{t}_{\mathrm{R2R1}} & -{}^{N}_{\mathrm{L2}}\mathbf{t}_{\mathrm{L2L1}} \end{bmatrix}}_{:=\mathbf{A}_{\mathrm{LS}}} \underbrace{\begin{bmatrix} \lambda \\ \mu \end{bmatrix}}_{:=\mathbf{x}_{\mathrm{LS}}} = \underbrace{\begin{bmatrix} ({}^{1}\mathbf{R}_{\mathrm{L2L1}} - \mathbf{I}_{3\mathrm{x}3})\ {}_{\mathrm{L}}\mathbf{t}_{\mathrm{LR}} \\ ({}^{2}\mathbf{R}_{\mathrm{L2L1}} - \mathbf{I}_{3\mathrm{x}3})\ {}_{\mathrm{L}}\mathbf{t}_{\mathrm{LR}} \\ \cdots \\ ({}^{N}\mathbf{R}_{\mathrm{L2L1}} - \mathbf{I}_{3\mathrm{x}3})\ {}_{\mathrm{L}}\mathbf{t}_{\mathrm{LR}} \end{bmatrix}}_{:=\mathbf{b}_{\mathrm{LS}}}, \tag{5}$$

which is solved for the scales using the Moore-Penrose pseudoinverse

$$\mathbf{x}_{\mathrm{LS}} = \mathbf{A}_{\mathrm{LS}}^{\dagger}\mathbf{b}_{\mathrm{LS}}. \tag{6}$$

2.3. Robust Estimation of Scale

There may be constraints between two stereo-frames which are ill-conditioned for computing the scales using (4). Either the motion-estimates of the individual VOs are inaccurate at a particular instant, or the motion which is performed by the rig is degenerate for scale estimation. In such a situation, we need to be able to exclude the corresponding constraints from the LS computation since they would lead to erroneous results. We therefore adopt a RANSAC scheme for removing outlier-constraints. Our implementation consists of the following three subfunctions:

- *Fitting*-function: Computes all free parameters of the model using a randomly picked minimal set of data samples. In our case, this means picking two stereo-frames from the windowed buffer and computing a scale hypothesis using (4). We thus only need one sample, i.e. a single constraint between two stereo-frames.
- *Is-Degenerate*-function: Checks if a randomly picked single constraint is degenerate before fitting the model. Degenerate cases of (4) are examined in [1], and occur if both cameras experience exclusive translation or translation combined with rotation around baseline, or if the system undergoes a specific planar motion (e.g. Ackermann motion). Degenerate constraints are not considered for hypothesizing scale values.
- *Distance*-function: This function tests all other constraints against the fitted model. A constraint i is regarded as an inlier if the scales hypothesis sufficiently satisfies the corresponding equation, e.g. (4). The inlier condition thus equals to $\left|\frac{||\mathbf{A}_{\mathrm{i}}\mathbf{x}_{\mathrm{hyp}}||}{||\mathbf{b}_{\mathrm{i}}||} - 1\right| < \mathrm{t}_{\mathrm{DIST}}$. During the experiments, the distance threshold $\mathrm{t}_{\mathrm{DIST}}$ was set to 0.3.

The RANSAC step finds the inlier constraints, which are then subsequently used to compute the scales by means of the LS scheme in (5). Note: If it happens that there are no inliers at all, the scales are set to one, which corresponds to scale propagation. Also note that the degeneracy depends only on the motion and not on the orientation of the cameras. From a geometric point of view, both cameras always represent omnidirectional bearing sensors separated by a certain distance—independently of the physical setup.

As illustrated in Fig. 2, the computed scales are fed back to the monocular VOs in order to rescale all poses and world points therein.

3. Fusion of Two Odometries

The previous section showed how to deterministically derive metric scales from two monocular VOs. Yet, we have not fully exploited the redundancy given by the motion information of two individual cameras. The rig constraint at t_2 is not necessarily fulfilled. This section addresses this is-

sue by fusing the two motion estimates into optimal camera positions at t_2 that satisfy the rigid rig constraint.

The fusion of the two motion estimates is tackled in four steps. First, the individual motion estimates are expressed in a common reference frame. Secondly, we derive the corresponding motion covariance matrices. Thirdly, the two estimates are fused in the common coordinate system, and, finally, the fused motion is transformed back into the individual camera frames.

3.1. Expressing Motions in a Common Frame

As indicated in Fig. 5, the additional reference frame S, named rig reference frame, is introduced as to similarly transform the individual motion estimates to a common frame via known extrinsic calibration parameters. The subscript before the transformation matrix $\mathbf{T}_{\mathrm{S2S1}}$ indicates the camera from which it is obtained. Rotational and translational part are decoupled, and for the sake of the subsequent fusion, rotations are written in terms of quaternions. The transformations in S thus result to

$$\begin{aligned} {}^{\mathrm{C}}\overline{\mathbf{q}}_{\mathrm{S2S1}} &= \overline{\mathbf{q}}_{\mathrm{CS}}^{-1} \otimes \overline{\mathbf{q}}_{\mathrm{C2C1}} \otimes \overline{\mathbf{q}}_{\mathrm{CS}} \\ {}^{\mathrm{C}}_{\mathrm{S1}}\mathbf{t}_{\mathrm{S1S2}} &= {}_{\mathrm{S}}\mathbf{t}_{\mathrm{SC}} + \mathbf{R}_{\mathrm{CS}}^{\mathrm{T}}\, {}_{\mathrm{R1}}\mathbf{t}_{\mathrm{C1C2}} - \mathbf{R}_{\mathrm{CS}}^{\mathrm{T}}\, \mathbf{R}_{\mathrm{C2C1}}^{\mathrm{T}}\, \mathbf{R}_{\mathrm{CS}}\, {}_{\mathrm{S}}\mathbf{t}_{\mathrm{SC}}. \end{aligned} \tag{7}$$

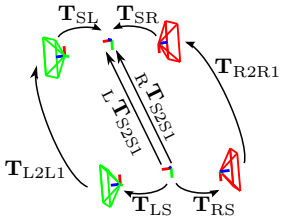

Figure 5. Transforming camera motions to rig reference system.

The above equations are applied $\forall\, \mathrm{C} \in \{R, L\}$.

3.2. Derivation of Covariances

In order to have a measure of the uncertainty of the estimated incremental motions of the cameras, we now derive their covariances. The covariances of the transformations from the rig frame S to the cameras are assumed to be known from the calibration process. For the covariances of the VO motion estimates themselves, a similar approach as in [29] is applied. Under the assumption that the 3D points are fixed, the covariance of the six motion parameters is derived from the reprojection error

$$E_C = \sum_{i=1..N} \left|\left| {}^{C}\mathcal{P}\left(\mathbf{T}\, \mathbf{X}_i\right) - {}^{C}\mathbf{x}_i \right|\right|_2, \tag{8}$$

where C stands for the right or left camera, respectively, $\mathcal{P}$ for the projection function of the camera, $\mathbf{X}$ for a 3D world point, and $\mathbf{x}$ for a 2D image point. During the Bundle-Adjustment (BA) in the monocular VOs, which is executed whenever a new keyframe is triggered, this reprojection error is minimized by means of a nonlinear least squares solver (Levenberg-Maquardt). Triggs *et al.* [28] showed that the covariance is then given up to scale by the inverse of the Hessian evaluated at the minimum of the cost function, i.e. the reprojection error. In analogy to data-fitting problems, it is common to approximate the Hessian by $\mathbf{J}^{\mathrm{T}}\mathbf{J}$ with $\mathbf{J}$ being the Jacobian [7]. The covariance of the motion parameters $\mathbf{\Theta} = \begin{bmatrix} t_x & t_y & t_z & \alpha & \beta & \gamma \end{bmatrix}^{\mathrm{T}}$ then reads

$$\mathbf{\Sigma}_{\mathbf{\Theta}} = \mathbf{\Sigma}_{\mathbf{T}_{\mathbf{t},\mathbf{rpy}}} = \begin{bmatrix} \mathbf{\Sigma}_{\mathbf{t},\mathbf{t}} & \mathbf{\Sigma}_{\mathbf{t},\mathbf{rpy}} \\ \mathbf{\Sigma}_{\mathbf{rpy},\mathbf{t}} & \mathbf{\Sigma}_{\mathbf{rpy},\mathbf{rpy}} \end{bmatrix} = \hat{\sigma}^2 (\mathbf{J}^{\mathrm{T}}\mathbf{J})^{-1}, \tag{9}$$

where $\hat{\sigma}^2$ is the estimated variance of the residual and $\mathbf{t}$ and $\mathbf{rpy}$ represent the translation and Euler angles, respectively.

For the fusion however, the covariances need to be expressed in terms of quaternions. The conversion from roll, pitch, yaw covariance representation to quaternion covariance representation is achieved by the Jacobian $\mathbf{H}$ introduced in [27]. Furthermore, we need to apply the scale factors to the translational part of the covariance matrices. Thus, the final covariance derived from (9) results to

$$\mathbf{\Sigma}_{\mathbf{T}_{\mathbf{t},\boldsymbol{\theta}}} = \begin{bmatrix} \mathbf{\Sigma}_{\mathbf{t},\mathbf{t}} & \mathbf{\Sigma}_{\mathbf{t},\boldsymbol{\theta}} \\ \mathbf{\Sigma}_{\boldsymbol{\theta},\mathbf{t}} & \mathbf{\Sigma}_{\boldsymbol{\theta},\boldsymbol{\theta}} \end{bmatrix} = \begin{bmatrix} \lambda\mathbf{\Sigma}_{\mathbf{t},\mathbf{t}}\lambda & \lambda\mathbf{\Sigma}_{\mathbf{t},\mathbf{rpy}}\mathbf{H}^{\mathrm{T}} \\ \mathbf{H}\mathbf{\Sigma}_{\mathbf{rpy},\mathbf{t}}\lambda & \mathbf{H}\mathbf{\Sigma}_{\mathbf{rpy},\mathbf{rpy}}\mathbf{H}^{\mathrm{T}} \end{bmatrix}, \tag{10}$$

where $\boldsymbol{\theta}$ represents the quaternion axis and $\mathbf{rpy}$ represents the Roll-Pitch-Yaw-angles. The Jacobian $\mathbf{H}$ in case of Tait-Bryan angles is given by

$$\mathbf{H} = \left[\, \mathbf{e}_x,\ \ \mathbf{R}_x(\alpha)\, \mathbf{e}_y,\ \ \mathbf{R}_x(\alpha)\mathbf{R}_y(\beta)\, \mathbf{e}_z \,\right], \tag{11}$$

with unit vectors $\mathbf{e}_i$ and rotation matrices $\mathbf{R}_i$.

The total covariance of the rig motion is approximated by the sum of the individual covariance matrices expressed in the rig reference system. It finally results to

$$\begin{aligned} {}_{\mathrm{C}}\mathbf{\Sigma}_{{}_{\mathrm{S1}}\mathbf{t}_{\mathrm{S1S2}},\boldsymbol{\theta}_{\mathrm{S2S1}}} &= \mathbf{\Sigma}_{{}_{\mathrm{S}}\mathbf{t}_{\mathrm{SC}},\boldsymbol{\theta}_{\mathrm{CS}}} + \mathcal{R}_{\mathrm{CS}}^{\mathrm{T}}\mathbf{\Sigma}_{{}_{\mathrm{C1}}\mathbf{t}_{\mathrm{C1C2}},\boldsymbol{\theta}_{\mathrm{C2C1}}}\mathcal{R}_{\mathrm{CS}} + \\ &\quad \mathcal{R}_{\mathrm{CS}}^{\mathrm{T}}\mathcal{R}_{\mathrm{C2C1}}^{\mathrm{T}}\mathcal{R}_{\mathrm{CS}}\mathbf{\Sigma}_{{}_{\mathrm{S}}\mathbf{t}_{\mathrm{SR}},\boldsymbol{\theta}_{\mathrm{CS}}}\mathcal{R}_{\mathrm{CS}}^{\mathrm{T}}\mathcal{R}_{\mathrm{C2C1}}\mathcal{R}_{\mathrm{CS}}, \end{aligned} \tag{12}$$

where $\mathcal{R}_{\mathrm{BI}}$ consists of rotation matrices along its diagonal following

$$\mathcal{R}_{\mathrm{BI}} := \begin{bmatrix} \mathbf{R}_{\mathrm{BI}} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_{\mathrm{BI}} \end{bmatrix}. \tag{13}$$

3.3. Fusion

The two motion estimates for the rig center are fused in a similar way as suggested by [26]. They estimate the state of an orientation sensor by means of an Extended Kalman Filter. In the update step, the prediction of the model is fused with the observed measurement of the sensor. We fuse the two motion estimates of the rig center in a similar way. This is accomplished by carrying out the following steps:

1. Residual We compute the difference in translation and rotation of the two motion estimates

$$\mathbf{r} = \begin{bmatrix} \mathbf{\Delta t} \\ \boldsymbol{\delta\theta} \end{bmatrix} \tag{14}$$

with

$$\begin{aligned} \mathbf{\Delta t} &= {}^{\mathrm{L}}_{\mathrm{S1}}\mathbf{t}_{\mathrm{S1S2}} - {}^{\mathrm{R}}_{\mathrm{S1}}\mathbf{t}_{\mathrm{S1S2}} \\ \begin{bmatrix} 1 \\ \frac{1}{2}\boldsymbol{\delta\theta} \end{bmatrix} &\approx {}^{\mathrm{L}}\overline{\mathbf{q}}_{\mathrm{S2S1}} \otimes {}^{\mathrm{R}}\overline{\mathbf{q}}_{\mathrm{S2S1}}^{-1}, \end{aligned} \tag{15}$$

where we make use of the small rotation approximation for the quaternion.

2. Weight Matrix The weight matrix is given by

$$\mathbf{F} = {}_{\mathrm{R}}\boldsymbol{\Sigma}_{\mathrm{S1}\mathbf{t}_{\mathrm{S1S2}},\boldsymbol{\theta}_{\mathrm{S2S1}}} \left({}_{\mathrm{R}}\boldsymbol{\Sigma}_{\mathrm{S1}\mathbf{t}_{\mathrm{S1S2}},\boldsymbol{\theta}_{\mathrm{S2S1}}} + {}_{\mathrm{L}}\boldsymbol{\Sigma}_{\mathrm{S1}\mathbf{t}_{\mathrm{S1S2}},\boldsymbol{\theta}_{\mathrm{S2S1}}} \right)^{-1}. \tag{16}$$

3. Correction The weight matrix is multiplied with the residual in order to get the correction terms

$$\begin{bmatrix} \boldsymbol{\Delta}\hat{\mathbf{t}} \\ \boldsymbol{\delta}\hat{\boldsymbol{\theta}} \end{bmatrix} = \mathbf{F}\,\mathbf{r}. \tag{17}$$

4. Fusion In the final fusion step, the corrected translation is computed additively, while the quaternion is obtained by multiplication. We obtain

$$\begin{aligned} {}_{\mathrm{S}}\mathbf{t}^{*}_{\mathrm{S1S2}} &= {}^{\mathrm{R}}_{\mathrm{S}}\mathbf{t}_{\mathrm{S1S2}} + \boldsymbol{\Delta}\hat{\mathbf{t}} \\ \overline{\mathbf{q}}^{*}_{\mathrm{S2S1}} &= \boldsymbol{\delta}\hat{\overline{\mathbf{q}}} \otimes {}^{\mathrm{R}}\overline{\mathbf{q}}_{\mathrm{S2S1}}, \end{aligned} \tag{18}$$

with

$$\boldsymbol{\delta}\hat{\overline{\mathbf{q}}} = \begin{bmatrix} \alpha \\ \frac{1}{2}\boldsymbol{\delta}\hat{\boldsymbol{\theta}} \end{bmatrix}. \tag{19}$$

The scaling factor α is needed in order to enforce that the length of the quaternion remains equal to one.

3.4. Transforming Motion back to Camera Frames

In this step, the optimal translations and rotations of the cameras are inferred from the fused motion of the rig center S. By doing so, it is ensured that the rig constraint at t_2 is still fulfilled. The final poses result to

$$\begin{aligned} \overline{\mathbf{q}}^{*}_{\mathrm{C2C1}} &= \overline{\mathbf{q}}_{\mathrm{CS}} \otimes \overline{\mathbf{q}}^{*}_{\mathrm{S2S1}} \otimes \overline{\mathbf{q}}^{-1}_{\mathrm{CS}} \\ {}_{\mathrm{C1}}\mathbf{t}^{*}_{\mathrm{C1C2}} &= -\mathbf{R}_{\mathrm{CS}}\,{}_{\mathrm{S}}\mathbf{t}_{\mathrm{SC}} + \mathbf{R}_{\mathrm{CS}}\,{}_{\mathrm{S1}}\mathbf{t}^{*}_{\mathrm{S1S2}} + \mathbf{R}_{\mathrm{CS}}\,\mathbf{R}^{\mathrm{T}}_{\mathrm{S2S1}}\,{}_{\mathrm{S}}\mathbf{t}_{\mathrm{SC}}, \end{aligned} \tag{20}$$

with $\mathrm{C} \in \{R, L\}$. After the optimal motion of the cameras has been derived, the corresponding translations and rotations are fed back to the monocular VOs as illustrated in Fig. 2. This is equivalent to a shift of the individual vision reference frames.

4. Experiments

The algorithm described in the previous sections is thoroughly tested on real data. The setup for the experiments as well as the achieved results are presented in the following.

4.1. The Camera Rig

The stereo rig used during the experiments consists of two global shutter CMOS imagers with 150° FOV each. As shown in Fig. 6, the cameras are facing opposite directions and their optical axes are aligned. Pless *et al.* [22] showed that this is the preferred configuration for two cameras by composing the Fisher information matrix. The rig additionally carries a camera triggering unit and markers for the external Vicon motion capture system.

A crucial point for the 6D motion estimation proposed in the previous sections is accurate knowledge about the extrinsic calibration of the cameras, i.e. the transformation

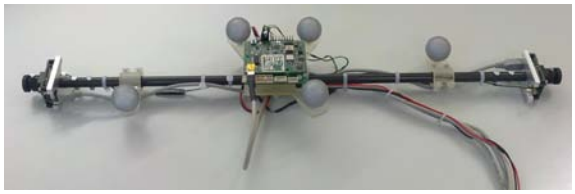

Figure 6. The stereo rig used during the experiments.

from the right to the left camera. While [17] uses a mirror to calibrate non-overlapping cameras, we arrange a calibration method which makes use of the external tracking system. The rig is immobile and the cameras capture images of a moving checkerboard of which the position is tracked by the tracking system. The position of the cameras can be reconstructed by concatenating three consecutive transformations. The first transformation is from the tracking system to the tracking markers on the checkerboard, the second from the markers to the checkerboard pattern and the third from the pattern to the camera. Whereas the first transformation is supplied by the tracking system, the second is known by construction and the third is obtained by the toolbox from [24]. The chains of transformations read

$$\begin{aligned} {}^{i}\mathbf{T}_{\mathrm{RV}} &= {}^{i}\mathbf{T}_{\mathrm{RP}} \cdot \mathbf{T}_{\mathrm{PM}} \cdot {}^{i}\mathbf{T}_{\mathrm{MV}} \\ {}^{i}\mathbf{T}_{\mathrm{LV}} &= {}^{i}\mathbf{T}_{\mathrm{LP}} \cdot \mathbf{T}_{\mathrm{PM}} \cdot {}^{i}\mathbf{T}_{\mathrm{MV}}, \end{aligned} \tag{21}$$

where 'V' stands for Vicon reference frame, 'M' for markers on the checkerboard, 'P' for the checkerboard pattern and 'R' or 'L' for the right or left camera, respectively. The superscript 'i' represents the time index (the transformation from the markers to the pattern is constant). The relative displacement from the right to the left camera is computed using the poses averaged over all i (typically 20):

$$\mathbf{T}_{\mathrm{LR}} = \bar{\mathbf{T}}_{\mathrm{LV}} \cdot \bar{\mathbf{T}}^{-1}_{\mathrm{RV}}. \tag{22}$$

Fig. 7 shows the result of the calibration. Note that our calibration process also recovers the transformation covariances. Also note that the central rig coordinate system is simply defined to be aligned with the markers on the rig.

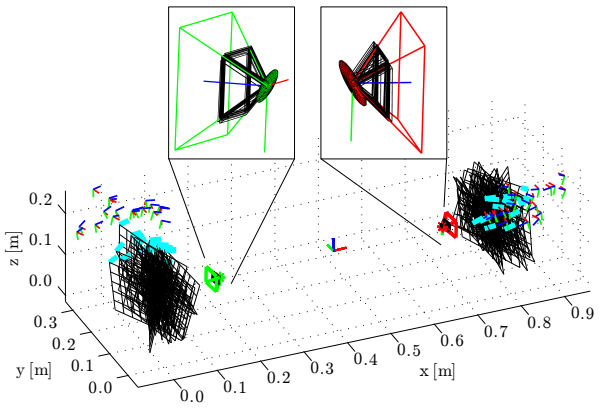

Figure 7. Extrinsic Calibration: Camera and checkerboard poses. Translational part of covariance matrix is indicated by 3D-ellipses.

4.2. Circular Trajectory

In order to assess the performance of the proposed system, we tested it on two datasets and compared the reconstructed trajectories to the ground truth supplied by the external tracking system. In the first dataset, the rig experiences a circular motion with reasonable amount of rotation.

Three different configurations of the algorithm were tested: In the first configuration, denoted by $(\bar{\mathbf{S}}, \bar{\mathbf{F}})$, monocular VO was carried out in each camera without subsequent scale estimation or fusion step. The scales have only been initialized using ground truth. In the second setup, indicated by $(\mathbf{S}, \bar{\mathbf{F}})$, scale estimation was conducted, but no fusion took place. In the third configuration, both the scale estimation and the fusion step were active $(\mathbf{S}, \mathbf{F})$. The outputs of these three setups can be seen in Fig. 8.

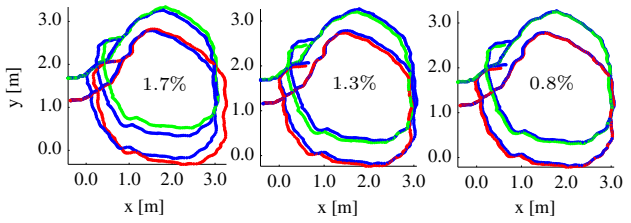

Figure 8. Dataset 'Circle' with different algorithm configurations: $(\bar{\mathbf{S}}, \bar{\mathbf{F}})$, $(\mathbf{S}, \bar{\mathbf{F}})$ and $(\mathbf{S}, \mathbf{F})$ with drifts. The ground truth is colored in blue, the motion estimates are red (right) and green (left).

The motion estimate becomes increasingly accurate from left to right. It is important to note, that in the left and middle configurations, each camera suffers from an individual drift as they do not fuse the estimates in a common reference frame. The setup $(\mathbf{S}, \mathbf{F})$ exhibits the least drift with only 0.8%.

Furthermore, common performance criteria are the ratio of the norm of the incremental translations (estimated vs. ground truth) and the relative translation vector error. The quotient of the estimated translation and the ground truth translation should stick to one for a sound motion estimation system. The second quality factor takes also the direction of the translations into account, and should ideally be zero. Both ratios are computed at keyframe instances only and illustrated in Fig. 9, where the superscript VO stands for the estimated and GT for the ground truth quantity.

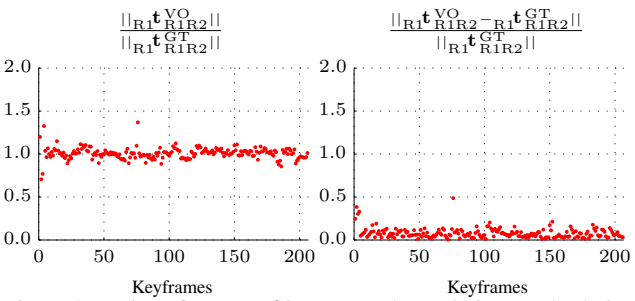

Figure 9. Ratios of norms of incremental translations and relative translation vector errors at keyframe instances.

Table 1 opposes our results to those obtained in [4] in terms of the above mentioned criteria. The comparison shows that the means of our method are relatively close to one or zero respectively, with reasonably small standard deviations. Note that our method continuously delivers scale results whereas the method in [4] has been applied to an outdoor dataset captured on a ground vehicle, attempting to estimate challenging Ackermann-like motion. Therefore, the compared approach was able to deliver meaningful scale results only at particular instants.

Method	Ratio of Norms	Vector Error
Our system	1.005 ± 0.071	0.079 ± 0.061
Approach by [4]	0.90 ± 0.28	0.23 ± 0.19

Table 1. Results of our system contrasted to results by [4].

Next, we examine the relation between the accuracy of the scale estimation, the number of inliers in the RANSAC scheme, and the degeneracy of the rig motion. The covariance of the estimated scales is derived from the LS-problem (6). The motion degeneracy is estimated by the scalar product of the translation and rotation axes. It indicates whether the camera undergoes planar motion or motion with almost no rotation or translation. The three quantities are indicated in Fig. 10.

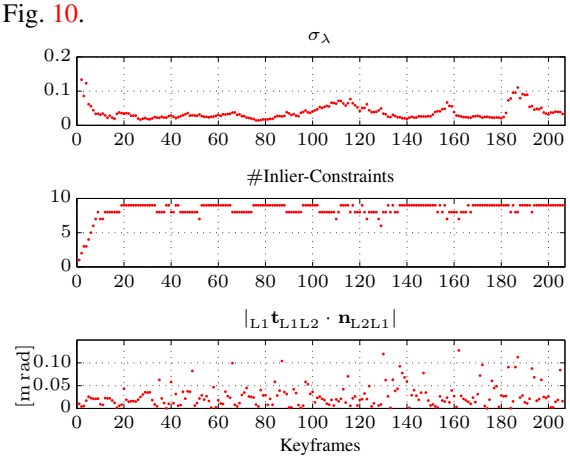

Figure 10. Relation of scale estimation accuracy, number of inliers in the RANSAC scheme and motion degeneracy.

The influence of the multi-frame buffer is visible at the beginning of the dataset, where only few frames are buffered and the scale estimation in turn has a large standard deviation. This dataset does not contain much degenerate motion for scale estimation.

4.3. Straight Trajectory

In this dataset, the rig experienced a straight motion with only little rotation. As in the previous circular motion dataset, the algorithm is tested in three different configurations, which are shown in Fig. 11.

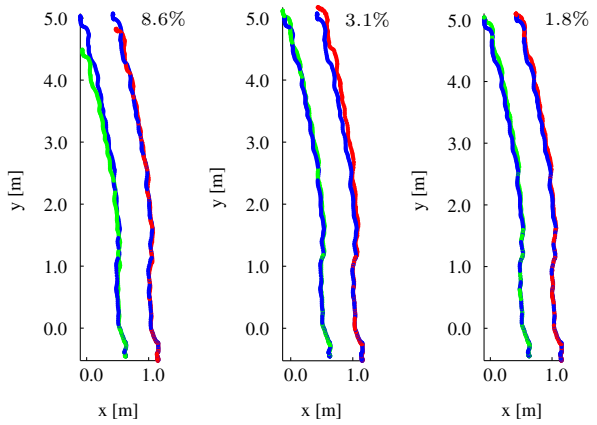

Figure 11. Dataset 'Straight' with the configurations: $(\bar{\mathbf{S}}, \bar{\mathbf{F}})$, $(\mathbf{S}, \bar{\mathbf{F}})$ and $(\mathbf{S}, \mathbf{F})$ with corresponding drifts (left to right).

Again, a steady improvement of accuracy from left to right is observed. Note that there is clearly more drift present than in the circular dataset. However, the setup with scale estimation and fusion still performs best with only 1.8%.

The standard deviation of the scale estimation, number of inliers in the RANSAC scheme and motion degeneracy of the rig are shown in Fig. 12.

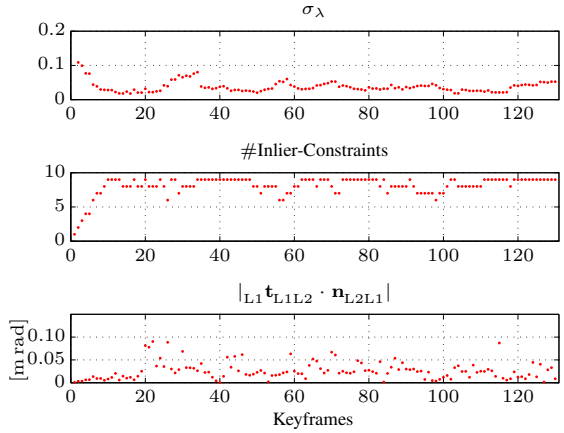

Figure 12. Scale estimation accuracy, number of inliers in the RANSAC scheme, and motion degeneracy of the rig.

One can observe that there are less inliers in the RANSAC scheme and that the standard deviation for the scale estimation is slightly higher compared to the circular dataset, which is due to the smaller amount of rotation between keyframes. The effect of the buffer length on the variance of the scale estimation can again be observed at the beginning of the dataset.

4.4. Real-Time Implementation

Our non-overlapping stereo system runs in real-time and employs the single camera VO framework from [15] on each camera individually, augmented by a windowed BA in order to increase robustness and accuracy of the estimated motion and structure. Moreover, it employs an enhanced version of the rotation-variant BRIEF [3] descriptor. It takes prior inter-frame in-plane rotation information into account in order to accordingly rotate the descriptor extraction pattern, and thus increase the robustness of feature matching even in case of rotation. The computational time remains similar and the descriptor similarity can still be efficiently evaluated by means of the Hamming distance. Note that, even though this VO framework employs information from an additional IMU in order to recover relative rotation priors in between successive frames, we still do not make use of the IMU for scale estimation here. Hence, the individual motion estimates could just as well be delivered by a vision-only based monocular odometry.

The implementation is integrated into the *Robot Operating System* (ROS) [23] and makes use of the OpenCV [2] and Eigen [9] libraries. The entire odometry system runs at a frame rate of 15 Hz on a 32 bit Intel Core i7 2.8 GHz machine with 4 GB RAM. The average computation times for double pose estimation between two frames (multi-threaded execution), scale estimation, and the fusion step are indicated in table 2. The time required for the scale estimation and the optimal fusion step is negligible in comparison to the runtime needed for the two monocular VOs. Given the fact that the proposed method is of modular nature, any monocular VO with pose covariance output can in principle be employed.

Step	Runtime [ms]
2x Monocular VO	69.5
Scale Estimation	0.1
Optimal Fusion	0.07

Table 2. Runtime analysis.

5. Conclusion

In this paper, we have presented a novel method for estimating metric 6DOF motion of a stereo rig with non-overlapping fields of view. To the best of our knowledge, it is the first metric VO system of this type that is demonstrated in real-time and employs information from two cameras which do not share their FOV. The proposed technique sustainably determines metric scale from rigid transformation constraints, and fuses cues from individual monocular VOs for an optimal result. Degenerate motion constraints which render the scale unobservable are robustly detected and excluded. Comprehensive performance evaluation by means of a comparison to accurate ground truth motion information underlines the accuracy and robustness of our metric motion estimates, and our results are better than those reported in the literature.

Acknowledgments

The research leading to these results has received funding from the European Community's Seventh Framework Programme

(FP7/2007-2013) under grant agreement n. 231855 (sFly) and from the Swiss National Science Foundation under grant agreement n. 200020-135050.

References

[1] N. Andreff, R. Horaud, and B. Espiau. Robot Hand-Eye Calibration Using Structure-from-Motion. *The International Journal of Robotics Research*, 20(3):228–248, 2001. 3

[2] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000. 7

[3] M. Calonder, V. Lepetit, C. Strecha, and P. Fua. BRIEF: Binary Robust Independent Elementary Features. In *Proceedings of the 11th European Conference on Computer Vision: Part IV*, ECCV'10, pages 778–792, Berlin, Heidelberg, 2010. Springer-Verlag. 7

[4] B. Clipp, J.-H. Kim, J.-M. Frahm, M. Pollefeys, and R. Hartley. Robust 6DOF Motion Estimation for Non-Overlapping, Multi-Camera Systems. In *Proceedings of the 2008 IEEE Workshop on Applications of Computer Vision*, pages 1–8, Washington, DC, USA, 2008. IEEE Computer Society. 2, 6

[5] B. Clipp, C. Zach, J. M. Frahm, and M. Pollefeys. A new Minimal Solution to the Relative Pose of a Calibrated Stereo Camera with Small Field of View Overlap. In *2009 IEEE 12th International Conference on Computer Vision*, pages 1725–1732. IEEE, October 2009. 2

[6] M. A. Fischler and R. C. Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM*, 24(6):381–395, 1981. 1

[7] P. E. Gill and W. Murray. Algorithms for the Solution of the Nonlinear Least-Squares Problem. *SIAM Journal on Numerical Analysis*, 15(5):977–992, October 1978. 4

[8] M. Grossberg and S. Nayar. A General Imaging Model and a Method for Finding its Parameters. In *IEEE International Conference on Computer Vision (ICCV)*, volume 2, pages 108–115, Jul 2001. 2

[9] G. Guennebaud, B. Jacob, et al. Eigen v3. http://eigen.tuxfamily.org, 2010. 7

[10] R. I. Hartley and A. Zisserman. *Multiple View Veometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004. 1

[11] R. Horaud and F. Dornaika. Hand-Eye Calibration. *The International Journal of Robotics Research*, 14(3):195–210, June 1995. 2

[12] J. Kim and T. Kanade. Degeneracy of the Linear Seventeen-Point Algorithm for Generalized Essential Matrix. *Journal of Mathematical Imaging and Vision*, 37:40–48, 2010. 2

[13] J. Kim, H. Li, and R. Hartley. Motion Estimation for Nonoverlapping Multicamera Rigs: Linear Algebraic and L_∞ Geometric Solutions. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(6):1044–1059, 2010. 2

[14] J.-H. Kim, R. Hartley, J.-M. Frahm, and M. Pollefeys. Visual Odometry for Non-Overlapping Views Using Second-Order Cone Programming. In *Proceedings of the 8th Asian conference on Computer vision - Volume Part II*, ACCV'07, pages 353–362, Berlin, Heidelberg, 2007. Springer-Verlag. 2

[15] L. Kneip, M. Chli, and R. Siegwart. Robust Real-Time Visual Odometry with a Single Camera and an IMU. In *Proc. of The British Machine Vision Conference (BMVC)*, Dundee, Scotland, August 2011. 7

[16] K. Konolige, M. Agrawal, and J. Solà. Large Scale Visual Odometry for Rough Terrain. In *Proc. International Symposium on Research in Robotics (ISRR)*, pages 201–212, November 2007. 2

[17] R. K. Kumar, A. Ilie, J. Frahm, and M. Pollefeys. Simple Calibration of Non-Overlapping Cameras with a Mirror. In *IEEE Conference on Computer Vision and Pattern Recognition, 2008. CVPR 2008*, pages 1–7. IEEE, June 2008. 5

[18] H. Li, R. Hartley, and J. Kim. A Linear Approach to Motion Estimation using Generalized Camera Models. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008. 2

[19] E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, and P. Sayd. Generic and Real-Time Structure from Motion Using Local Bundle Adjustment. *Image and Vision Computing*, 27(8):1178–1193, July 2009. 2

[20] D. Nistér, O. Naroditsky, and J. Bergen. Visual Odometry. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 1:652–659, 2004. 1, 2

[21] G. Nützi, S. Weiss, D. Scaramuzza, and R. Siegwart. Fusion of IMU and Vision for Absolute Scale Estimation in Monocular SLAM. *Journal of Intelligent & Robotic Systems*, 61:287–299, November 2010. 2

[22] R. Pless. Using Many Cameras as One. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 2:587–593, 2003. 2, 5

[23] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng. ROS: An Open-Source Robot Operating System. In *ICRA Workshop on Open Source Software*, 2009. 7

[24] D. Scaramuzza, A. Martinelli, and R. Siegwart. A Toolbox for Easily Calibrating Omnidirectional Cameras. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5695–5701. IEEE, October 2006. 5

[25] H. D. Stewénius and D. Nistér. Solutions to Minimal Generalized Relative Pose Problems. In *In Workshop on Omnidirectional Vision*, Beijing China, October 2005. 2

[26] N. Trawny and S. I. Roumeliotis. Indirect Kalman Filter for 3D Attitude Estimation. Technical 2005-002, University of Minnesota, March 2005. 4

[27] N. Trawny and S. I. Roumeliotis. Jacobian for Conversion from Euler Angles to Quaternions. Technical 2005-004, University of Minnesota, November 2005. 4

[28] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon. Bundle Adjustment – A Modern Synthesis. In *Proceedings of the International Workshop on Vision Algorithms: Theory and Practice*, ICCV '99, pages 298–372, London, UK, 2000. Springer-Verlag. 4

[29] R. Voigt, J. Nikolic, C. Hürzeler, S. Weiss, L. Kneip, and R. Siegwart. Robust Embedded Egomotion Estimation. *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, September 2011. 4