On Scale Initialization in Non-overlapping Multi-perspective Visual Odometry

Yifu Wang(✉) and Laurent Kneip

Australian National University, Canberra, Australia
u5434194@anu.edu.au, laurent.kneip@anu.edu.au

Abstract. Multi-perspective camera systems pointing into all directions represent an increasingly interesting solution for visual localization and mapping. They combine the benefits of omni-directional measurements with a sufficient baseline for producing measurements in metric scale. However, the observability of metric scale suffers from degenerate cases if the cameras do not share any overlap in their field of view. This problem is of particular importance in many relevant practical applications, and it impacts most heavily on the difficulty of bootstrapping the structure-from-motion process. The present paper introduces a complete real-time pipeline for visual odometry with non-overlapping, multi-perspective camera systems, and in particular presents a solution to the scale initialization problem. We evaluate our method on both simulated and real data, thus proving robust initialization capacity as well as best-in-class performance regarding the overall motion estimation accuracy.

1 Introduction

Over the past decade, automated real-time visual localization and mapping has often been proclaimed as a mature computer vision technology. However, it is only with the emerge of novel, billion-dollar industries such as autonomous driving, robotics, and mixed reality consumer products that this technology gets now put to a serious test. While single camera solutions [4,10,19,20,22,28] are certainly the most interesting from a more scientific point of view, they are also challenged by many potential bottlenecks such as a limited field of view, moderate sampling rates, and a low ability to deal with texture-poor environments or agile motion. In addition to fast sensors such as inertial measurement units, the engineering standpoint therefore envisages the use of stereo [17], depth [21,25,29,30], or even light-field cameras that simplify or robustify the solution of the structure-from-motion problem by providing direct 3D depth-of-scene measurements.

The present paper is focusing on yet another type of sensor system that aims at combining benefits from different directions, namely *Multi-Perspective Cameras (MPCs)*. If pointing the cameras into different, opposite directions, the flow fields caused by translational and rotational motion become very distinctive [15], meaning that MPC solutions are strong at avoiding motion degeneracies. Furthermore, omni-directional observation of the environment makes failures due to texture-poor situations much more unlikely. In contrast to regular

M. Liu et al. (Eds.): ICVS 2017, LNCS 10528, pp. 144–157, 2017.
DOI: 10.1007/978-3-319-68345-4_13

omni-directional cameras, MPCs maintain the advantage of not introducing any significant lens distortions in the perceived visual information. Just like plain monocular cameras, MPCs also remain kinetic depth sensors. This means that they have no inherent limitations like stereo or depth cameras, which have limited range, or—in the latter case—cannot be used outdoors. As a final benefit, MPC systems are able to produce measurements in metric scale even if there is no internal overlap in the cameras' field of view.

MPCs are becoming increasingly important from an economic point of view. Looking at the most recent designs from the automotive or the consumer electronics industry, it is not uncommon to find a large number of affordable visual onboard sensors looking into various directions to provide complete capturing of the surrounding environment. An example of the fields of view of a modern car's visual sensors is shown in Fig. 1. The drawback with many such arrangements, however, is that the sensors do not share any significant overlap in their field of view. We call those camera arrays *non-overlapping MPCs*.

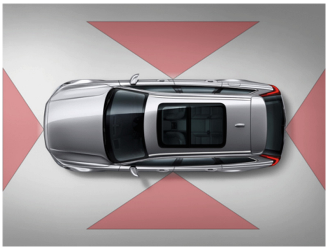


Fig. 1. Example fields of view of a multi-perspective camera mounted on a modern car.

The proper handling of non-overlapping MPCs requires the solution of two fundamental problems:

- As discussed in [3], non-overlapping MPCs are easily affected by motion degeneracies that cause scale unobservabilities, such as straight or Ackermann motion. This is a severe problem especially in automotive applications or in general during the bootstrapping phase, where no scale information can be propagated from prior processing.
- In order to truly benefit from the omni-directional measurements of MPCs, the measurements need to be processed jointly in each step of the computation. This is challenging as classical formulations of space resectioning and bundle adjustment all rely on a simple perspective camera model.

The present paper notably provides solutions to these two problems. The paper is organized as follows. Section 2 introduces further related work. Section 3 then provides an overview of our complete non-overlapping MPC motion estimation pipeline as well as the joint bootstrapping and global optimization modules. Section 4 finally presents the promising results we have obtained on both simulated and real data.

2 Motivation and Further Background

The motion estimation problem with MPCs can be approached in two fundamentally different ways. The first one consists of a loosely-coupled scheme where the information in each camera is used to solve individual monocular structure-from-motion problems, and the results from every camera are then fused in a subsequent pose averaging module. Kazik et al. [9] apply this solution strategy to a stereo camera rig with two cameras pointing into opposite directions. The inherent difficulty of this approach results from the scale invariance of the individual monocular structure-from-motion results. Individual visual scales first have to be resolved through an application of the hand-eye calibration constraint [8] before the individual pose results can be fused. Furthermore, the fact that the measurements of each camera are processed independently means that the benefit of having omni-directional measurements remains effectively unexploited during the geometric computations.

The second solution strategy assumes that the frames captured by each camera are synchronized, and hence can be bundled in a multi-frame measurement that contains one image of each camera from the same instant in time. Relying on the idea of *Using many cameras as one* [24], the fundamental problems of structure from motion can now be solved jointly for the entire MPC system, rather than for each camera individually. The measurements captured by the entire MPC can notably be described using a generalized camera, a model that envisages the description of measured image points via spatial rays that intersect with the corresponding camera's center, all expressed in a common frame for the entire MPC. By relying on the generalized camera model, the problems of joint absolute and relative camera pose estimation for the entire MPC rig have been successfully solved [12–14,18,23,24,27]. An excellent summary of the state-of-the-art in generalized camera pose computation is provided by the OpenGV library [11], a relatively complete collection of algorithms for solving related problems.

Despite the fact that closed-form solutions for the underlying algebraic geometry problems of generalized absolute and relative camera pose computation have already been presented, a full end-to-end pipeline for visual odometry with a non-overlapping MPC system that relies exclusively on the generalized camera paradigm remains an open problem. The problem mostly lies in the bootstrapping phase. As explained in [3], the relative pose for a multi-camera system can only be computed if the motion does not suffer from the degenerate case of Ackermann-like motion (which includes the case of purely straight motion). Unfortunately, in a visual odometry scenario, the images often originate from a smooth trajectory with only moderate dynamics, hence causing the motion between two sufficiently close frames to be almost always very close to the degenerate case. Kneip and Li [13] claim that the rotation can still be found, but we confirmed through our experiments that even the quality of the relative rotation is not sufficiently good to reliably bootstrap MPC visual odometry. A robust initialization procedure, as well as a complete, real-time end-to-end pipeline, notably, are the main contributions of this work.

3 Joint Motion Estimation with Non-overlapping Multi-perspective Cameras

This section outlines our complete MPC motion estimation pipeline. We start with an overview of the entire framework, explaining the state machine and resulting sequence of operations especially during the initialization procedure. We then look at two important sub-problems of the initialization, namely the robust retrieval of absolute orientations for the first frames of a sequence, as well as a joint linear recovery of the corresponding relative translations and 3D points. We conclude with an insight into the final bundle adjustment back-end that is entered once the initialization is completed.

3.1 Notations and Prior Assumptions

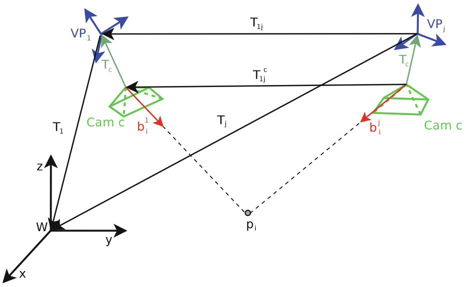


Fig. 2. Notations used throughout this paper (best viewed in color). Please see text for detailed explanations.

The MPC frames of a video sequence are denoted by VP_j, where $j = \{1, \cdots, m\}$. Their poses are expressed by transformation matrices $\mathbf{T}_j = \begin{bmatrix} \mathbf{R}_j & \mathbf{t}_j \\ \mathbf{0} & 1 \end{bmatrix}$ such that $\mathbf{T}_j\mathbf{x}$ transforms $\mathbf{x}$ from the MPC to the world frame (denoted W). Let us now assume that our MPC has k cameras. This leads to the definition of transformation matrices $\mathbf{T}_c = \begin{bmatrix} \mathbf{R}_c & \mathbf{t}_c \\ \mathbf{0} & 1 \end{bmatrix}$, where $c \in \{1, \cdots, k\}$. They permit the transformation of points from the respective camera frame c to the MPC frame. Assuming that the MPC rig is static, these transformations are constant and determined through a prior extrinsic calibration process. We also define the relative transformation $\mathbf{T}_{1j} = \begin{bmatrix} \mathbf{R}_{1j} & \mathbf{t}_{1j} \\ \mathbf{0} & 1 \end{bmatrix}$ that allows us to transform points from VP_j back to VP_1. We furthermore assume that—given that we are in a visual odometry scenario and that the cameras have no overlap in their fields of view—the cameras do not share any point observations. We therefore can associate each one of our points $\mathbf{p}_i, i \in \{1, \cdots, n\}$ to one specific camera within the rig, denoted by the index c_i. To conclude, we also assume that the intrinsic camera parameters are known, which is why we can always transform 2D points into spatial unit vectors pointing from the individual camera centers to the respective world points. We denote these measurements $\mathbf{b}_i^j$, meaning the measurement of point $\mathbf{p}_i$ (with camera c_i) in the MPC frame VP_j. Our derivations furthermore utilize the transformation $\mathbf{T}_{1j}^c$, which permits the direct transformation of points from the camera frame c in VP_j to the camera frame c in VP_1. All variables are indicated in Fig. 2.

3.2 Framework Overview

A flowchart of our proposed method detailing all steps including the initialization procedure is illustrated in Fig. 3. After the definition of a first (multi-perspective)

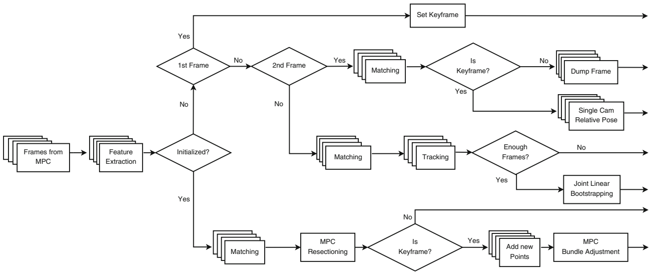


Fig. 3. Overview of the proposed visual localization and mapping pipeline for MPC systems. The flowchart in particular outlines the detailed idea behind the initialization procedure.

keyframe[1], the algorithm keeps matching inter-camera correspondences between the first and subsequent MPC frames until the average of the median frame-to-frame disparity for each camera surpasses a predefined threshold (verified in the decision nodes "Is Keyframe?"). Once this happens, we add a second keyframe and compute all $\mathbf{T}_{12}^{c}$ using classical single camera calibrated relative pose computation [26]. We furthermore triangulate an individual point cloud for every camera in the MPC array. Subsequent frames from the individual cameras are then aligned with respect to these maps using classical single camera calibrated absolute pose computation [16]. Once enough frames are collected, the initialization is completed by the joint, linear MPC pose initialization module outlined in Sects. 3.3 and 3.4. Note that individual single camera tracking is only performed in order to eliminate outlier measurements and obtain prior knowledge about relative rotations. It bypasses the weakness of methods such as [13] of not being able to deliver robust generalized relative pose results in most practically relevant cases. The actual final initialization step and all subsequent modules then perform joint MPC measurement processing.

After the initialization is completed, the frames of each new MPC pose are matched individually to the frames of the most recent MPC keyframe, but the alignment is solved jointly using generalized camera absolute pose computation [12]. We keep checking the local distinctiveness of every MPC frame by evaluating the frame-to-frame disparities in the above outlined manner, and add new keyframes everytime the threshold is surpassed. To conclude, we add new 3D points everytime a new keyframe is added, and perform generalized windowed bundle adjustment to jointly optimize over several recent MPC poses and the 3D landmark positions. This back-end optimization procedure is outlined in Sect. 3.5.

[1] *Keyframes* are simply frames that are retained in a buffer of frames due to sufficient local distinctiveness [10].

3.3 Initial Estimation of Relative Rotations

The very first part of our computation executes visual odometry in each camera individually. In order to make use of the relative orientations, we propose to first eliminate the redundancy in the information. This is done by first combining the computed orientations with the camera-to-MPC transformations $\mathbf{T}_c$ in order to obtain relative orientation estimates for the entire MPC rig. We now have k samples for the MPC frame-to-frame orientations in the frame buffer. We apply L_1 rotation averaging based on the Weiszfeld algorithm as outlined in [6] in order to obtain an accurate, unique representation.

3.4 Joint Linear Bootstrapping

The computation steps until here provide sets of inlier inter-camera correspondences and reasonable relative rotations between subsequent MPC frames. The missing variables towards a successful bootstrapping of the computation are given by MPC positions and point depths. Translations and point depths can also be taken from the prior individual visual odometry computations [9], but they may be unreliable, and—more importantly—have different unknown visual scale factors that would first have to be resolved.

We propose a new solution to this problem which solves for all scaled variables (i.e. positions and point depths) through one joint, closed-form, linear initialization procedure. What we are exploiting here is the known fact that structure from motion can be formulated as a linear problem once the relative rotations are subtracted from the computation (although results will not minimize a geometrically meaningful error anymore).

Let us assume that we have two MPC view-points VP_1 and VP_j. We start by formulating the hand-eye calibration constraint for a camera c inside the MPC.

$$\begin{cases} \mathbf{t}_c = \mathbf{t}_{1j} + \mathbf{R}_{1j} \cdot \mathbf{t}_c + \mathbf{R}_{1j} \cdot \mathbf{R}_c \cdot \mathbf{t}_{j1}^c \\ \mathbf{R}_c = \mathbf{R}_{1j} \cdot \mathbf{R}_c \cdot \mathbf{R}_{j1}^c \end{cases} \tag{1}$$

Let us now assume that there is one observed world point $\mathbf{p}_i$ giving rise to the measurements $\mathbf{b}_i^1$ and $\mathbf{b}_i^j$ inside the camera. The latter now has the index c_i. The point inside the first camera is simply given as $\lambda_i \cdot \mathbf{b}_i^1$, where λ_i denotes the depth of $\mathbf{p}_i$ seen from camera c_i in VP_1. We now apply $\mathbf{T}_{j1}^{c_i}$ and transform this point into camera c_i of VP_j. In here, the point obviously needs to align with the direction $\mathbf{b}_i^j$, which leads us to the constraint

$$(\mathbf{R}_{j1}^{c_i} \cdot \lambda_i \cdot \mathbf{b}_i^1 + \mathbf{t}_{j1}^{c_i}) \times \mathbf{b}_i^j = 0. \tag{2}$$

By replacing (1) in (2), we finally arrive at

$$(\mathbf{R}_{c_i}^T \cdot \mathbf{R}_{1j}^T \cdot \mathbf{R}_{c_i} \cdot \lambda_i \cdot \mathbf{b}_i^1) \times \mathbf{b}_i^j - (\mathbf{R}_{c_i}^T \cdot \mathbf{R}_{1j}^T \cdot \mathbf{t}_{1j}) \times \mathbf{b}_i^j = -\mathbf{R}_{c_i}^T \cdot \mathbf{R}_{1j}^T (\mathbf{t}_{c_i} - \mathbf{R}_{1j} \cdot \mathbf{t}_{c_i}) \times \mathbf{b}_i^j. \tag{3}$$

Let us now assume that we have n points and m MPC frames. The unknowns are hence given by λ_i, where $i \in \{1, \cdots, n\}$, and $\mathbf{t}_{1j}$, where $j \in \{2, \cdots, m\}$. We only use fully observed points, meaning that each point $\mathbf{p}_i$ is observed by camera c_i in each MPC frame, thus generating the measurement sequence $\left\{\mathbf{b}_i^1, \cdots, \mathbf{b}_i^m\right\}$. All pair-wise constraints in the form of (3) can now be grouped in one large linear problem $\mathbf{Ax} = \mathbf{b}$, where

$$\mathbf{A} = \left[\begin{array}{cccccc} (\mathbf{R}_{c_1}^T\mathbf{R}_{12}^T\mathbf{R}_{c_1}\mathbf{b}_1^1)\times\mathbf{b}_1^2 & & & [\mathbf{b}_1^2]_\times\mathbf{R}_{c_1}^T\mathbf{R}_{12}^T & & \\ & \cdots & & \cdots & & \\ & & (\mathbf{R}_{c_n}^T\mathbf{R}_{12}^T\mathbf{R}_{c_n}\mathbf{b}_n^1)\times\mathbf{b}_n^2 & [\mathbf{b}_n^2]_\times\mathbf{R}_{c_n}^T\mathbf{R}_{12}^T & & \\ \cdots & \cdots & \cdots & & \cdots & \\ (\mathbf{R}_{c_1}^T\mathbf{R}_{1m}^T\mathbf{R}_{c_1}\mathbf{b}_1^1)\times\mathbf{b}_1^m & & & & & [\mathbf{b}_1^m]_\times\mathbf{R}_{c_1}^T\mathbf{R}_{1m}^T \\ & \cdots & & & & \cdots \\ & & (\mathbf{R}_{c_n}^T\mathbf{R}_{1m}^T\mathbf{R}_{c_n}\mathbf{b}_n^1)\times\mathbf{b}_n^m & & & [\mathbf{b}_n^m]_\times\mathbf{R}_{c_n}^T\mathbf{R}_{1m}^T \end{array}\right] \tag{4}$$

$$\mathbf{x} = \begin{bmatrix} \lambda_1 \\ \cdots \\ \lambda_n \\ \mathbf{t}_{12} \\ \cdots \\ \mathbf{t}_{1m} \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} -\mathbf{R}_{c_1}^T\mathbf{R}_{12}^T(\mathbf{t}_{c_1} - \mathbf{R}_{12}\mathbf{t}_{c_1})\times\mathbf{b}_1^2 \\ \cdots \\ -\mathbf{R}_{c_n}^T\mathbf{R}_{12}^T(\mathbf{t}_{c_n} - \mathbf{R}_{12}\mathbf{t}_{c_n})\times\mathbf{b}_n^2 \\ \cdots \\ -\mathbf{R}_{c_1}^T\mathbf{R}_{1m}^T(\mathbf{t}_{c_1} - \mathbf{R}_{1m}\mathbf{t}_{c_1})\times\mathbf{b}_1^m \\ \cdots \\ -\mathbf{R}_{c_n}^T\mathbf{R}_{1m}^T(\mathbf{t}_{c_n} - \mathbf{R}_{1m}\mathbf{t}_{c_n})\times\mathbf{b}_n^m \end{bmatrix} \tag{5}$$

$\mathbf{A}$ and $\mathbf{b}$ can be computed from the known extrinsics, inlier measurements, and relative rotations, whereas $\mathbf{x}$ contains all unknowns.

The non-homogeneous linear problem $\mathbf{Ax} = \mathbf{b}$ could be solved by a standard technique such as QR decomposition, thus resulting in $\mathbf{x} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b}$. However, in order to improve the efficiency, we utilize the Schur-complement trick and exploit the sparsity pattern of the matrix. Matrix $\mathbf{A^TA}$ is divided into four smaller sub-blocks $\mathbf{A}^T\mathbf{A} = \begin{bmatrix}\mathbf{P} & \mathbf{Q} \\ \mathbf{R} & \mathbf{S}\end{bmatrix}$, and our two vectors $\mathbf{x}$ and $\mathbf{b}$ are decomposed accordingly thus resulting in $\mathbf{x} = [\mathbf{x}_1^T, \mathbf{x}_2^T]^T$ and $\mathbf{A^Tb} = [\left(\mathbf{A^Tb}_1\right)^T, \left(\mathbf{A^Tb}_2\right)^T]^T$. Substituted into the original equation $\mathbf{A}^T\mathbf{Ax} = \mathbf{A}^T\mathbf{b}$, and after variable elimination, we obtain

$$\begin{cases} \mathbf{Px}_1 = \mathbf{A^Tb}_1 - \mathbf{Qx}_2 \\ (\mathbf{S} - \mathbf{RP}^{-1}\mathbf{Q})\mathbf{x}_2 = \mathbf{A^Tb}_2 - \mathbf{RP}^{-1}\mathbf{A^Tb}_1 \end{cases} \tag{6}$$

This form permits us to first solve for $\mathbf{x}_2$ individually, a much smaller problem due to the relatively small number of MPC frames. $\mathbf{x}_1$ is subsequently retrieved by simple variable back-substitution.

3.5 Multi-perspective Windowed Bundle Adjustment

After bootstrapping, we can continuously use multi-perspective absolute camera pose computation [23] in order to align subsequent MPC frames with respect to the local point cloud. Furthermore, we keep buffering keyframes each time the average frame-to-frame disparity exceeds a given threshold. This in fact already constitutes a complete procedure for MPC visual odometry. In order to improve the accuracy of the solution, we add a windowed bundle adjustment back-end to our pipeline [7]. The goal of windowed bundle adjustment (BA) is to optimize 3D point positions and estimated MPC poses over all correspondences observed in a certain number of most recent keyframes. The key idea here is that points are generally observed in more than just two keyframes. By minimizing the reprojection error of every point into every observation frame, we implicitly take multi-view constraints into account, thus improving the final accuracy of both structure and camera poses. The computation is restricted to a bounded window of keyframes not to compromise computational efficiency. This form of non-linear optimization is also known as *sliding window bundle adjustment*.

Let us define the set $\mathcal{J}_i = \{j_1, \cdots, j_k\}$ as the set of MPC keyframe indices for which camera c_i observes the point $\mathbf{p}_i$. Let us furthermore assume that the size of the optimization window is s, and the set of points is already limited to points that have at least two observations within the s most recent keyframes. The objective of windowed bundle adjustment can now be formulated as

$$\left\{\hat{\mathbf{T}}_{m-s+1}, \cdots, \hat{\mathbf{T}}_m, \hat{\mathbf{p}}_1, \cdots, \hat{\mathbf{p}}_n\right\} =$$
$$\underset{\mathbf{T}_{m-s+1}, \cdots, \mathbf{T}_m, \mathbf{p}_1, \cdots, \mathbf{p}_n}{\operatorname{argmin}} \sum_{i=1}^{n} \sum_{j \in \mathcal{J}_i} \|\pi_{c_i}(\tilde{\mathbf{b}}_i^j) - \pi_{c_i}(\mathbf{T}_{c_i}^{-1}\mathbf{T}_j^{-1}\tilde{\mathbf{p}}_i)\|^2. \tag{7}$$

where

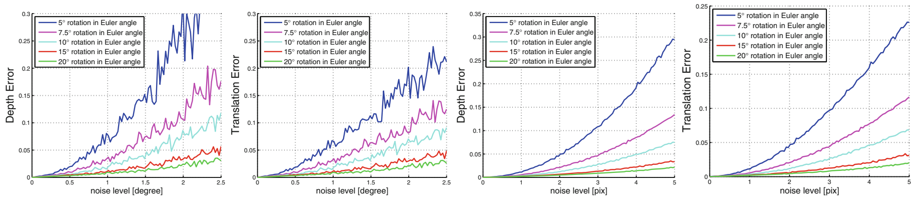
- $\mathbf{T}_j$ is parametrized minimally as a function of 6 variables.
- $j \in \{m - s + 1, \cdots, m\}$.
- π_{c_i} is the known (precalibrated) camera-to-world function of camera c_i. It transforms 3D points in homogeneous form into 2D Euclidean points.
- $\tilde{\mathbf{x}} =$ takes the homogeneous form of $\mathbf{x}$ by appending a 1.
- $\pi_{c_i}(\tilde{\mathbf{b}}_i^j)$ is the original, measured image location of the spatial direction $\mathbf{b}_i^j$.

4 Experimental Results

We test our algorithm on both simulated and real data. The simulation experiments analyze the noise resilience of our linear bootstrapping algorithm. The real data experiment then evaluates the performance of the complete pipeline by comparing the obtained results against ground truth data collected with an external motion tracking system, as well as a loosely-coupled alternative.

4.1 Results on Simulated Data

We perform experiments on synthetic data to analyze the performance of our linear MPC pose initialization module in the presence of varying levels of noise. In all our simulation experiments, we simply use 2 cameras pointing into opposite directions, and generate 10 random points in front of each camera. We furthermore generate 10 homogeneously distributed camera poses generating near fronto-parallel motion for both cameras. To conclude, we add an oscillating rotation about the main direction of motion. The maximum amplitude of this rotation is set to either $5°, 7.5°, 10°, 15°$ or $20°$, which creates an increasing distance



(a) (b) (c) (d)

Fig. 4. Benchmark of our linear bootstrapping algorithm showing relative translation and 3D point depths error for different levels of noise in the relative rotation and 2D landmark observations. The experiment is repeated for different "out-of-plane" dynamics, which causes significant differences in the scale observability of the problem. Note that each value in the figures is averaged over 1000 random experiments.

to the degenerate case of Ackerman motion. We perform two separate experiments in which we add noise to either the relative rotations or the 2D bearing vector measurements pointing from the camera centers to the landmarks. The error for each noise level is averaged over 1000 random experiments.

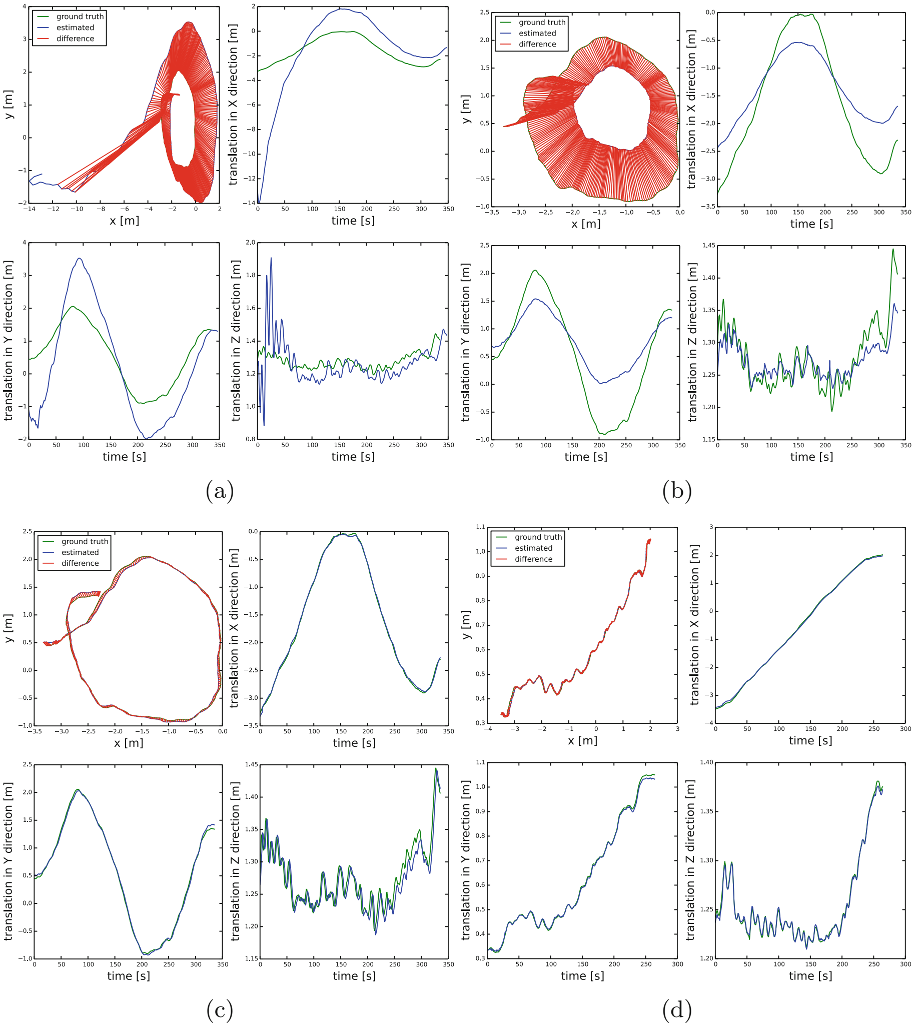
In our first experiment, we add noise to the relative rotations by multiplying them with another random rotation matrix that is derived from uniformly sampled Euler angles with a maximum value reaching from zero to 2.5°. The reported errors are the relative depth error of the 3D points $\frac{\|\lambda_{est}\|}{\|\lambda_{true}\|}$, and the relative translation magnitude error $\frac{\|\mathbf{t}_{est}\|}{\|\mathbf{t}_{true}\|}$. The errors are indicated in Figs. 4(a) and (b), respectively.

In our second experiment, we simulate noise on the bearing vectors by adding a random angular deviation θ_{rand} such that $\tan\theta_{\text{rand}} < \frac{\sigma}{f}$, where f is a virtual focal length of 500 pixels, and σ is a virtual maximum pixel noise level reaching from 0 to 5 pixels. We analyze the same errors and the results are reported in Figs. 4(c) and (d), respectively.

As can be concluded from the results, a reasonable amount of noise in both the point observations as well as the relative rotations can be tolerated. However, the correct functionality of the linear solver depends critically on the observability of the metric scale. Limiting the maximum amplitude of the out-of-plane rotation to a low angle (e.g. below 5°) can quickly compromise the stability of the solver and cause very large errors. In practice, this means that accurate results can only be expected if we add sufficiently many frames with sufficiently rich dynamics to our solver.

4.2 Results on Real Data

We have been given access to the data already used in [9], which allows us to compare our method against accurate ground truth measurements obtained by an external tracking device, a loosely-coupled alternative [9], and a more traditional approach from the literature [3]. The data consists of two different sequences captured with a synchronized, non-overlapping stereo rig that contains two cameras facing opposite directions. For further details about the hardware including intrinsic parameter values as well as the extrinsic calibration procedure, the reader is kindly referred to [9]. The sequences are henceforth referred to as the *circular* and *straight* motion sequences. In the circular motion sequence, the rig moves with significant out-of-plane rotation along a large loop. In the straight motion sequence, the rig simply moves forward with significantly reduced out-of-plane rotation. Both *circular* and *straight* datasets run at 10FPS. All experiments are conducted on a regular desktop computer with 8 GB RAM and



(a)

(b)

(c)

(d)

Fig. 5. Bottom row: results obtained on the circular and straight motion sequences from [9]. Top row: results obtained for different algorithm configurations that do not fully exploit the modules presented in this paper.

an Intel Core i7 2.8 GHz CPU. Our C++ implementation runs in real-time, and uses OpenCV [2], Eigen [5], OpenGV [11] and the Ceres Solver library [1]. op In order to assess the impact of our proposed linear bootstrapping and generalized sliding window bundle adjustment modules, we analyze three different algorithm configurations on the circular motion sequence. In our first test—indicated in Fig. 5(a)—we do not use our proposed initialization procedure, but simply rely on the method presented in [13] to bootstrap the algorithm from a pair of sufficiently separated frames in the beginning of the sequence. We tested numerous entry points, but the algorithm consistently fails to produce a good initial relative translation, thus resulting in severely distorted trajectories. In our second test—indicated in Fig. 5(b)—we rely on our linear bootstrapping algorithm to initialize the structure-from-motion process, but still do not activate windowed bundle adjustment. The obtained results are already much better, but still relatively far away from ground truth. It seems that our linear solver is able to produce meaningful initial values, but—due to the ill-posed nature of the problem—still has some error and further error is accumulated throughout the sequence. In our final test—indicated in Fig. 5(c)—we then also activate the sliding window bundle adjustment, thus leading to high-quality results with very little drift away from ground truth. Once a sufficiently close initialization point is given, the non-linear optimization module is consistently able to compensate remaining scale and orientation errors. Finally, Fig. 5(d) shows that the algorithm is also able to successfully process the more challenging straight motion sequence.

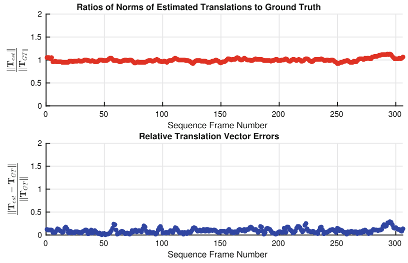


Fig. 6. Ratios of norms of estimated translations to ground truth and relative translation vector errors

Table 1. Performance comparison against [3,9]

Method	Ratio of norms
Approach by [3]	1.005 ± 0.071
Approach by [9]	0.90 ± 0.28
Our method	0.996 ± 0.038
Method	Vector error
Approach by [3]	0.079 ± 0.061
Approach by [9]	0.23 ± 0.19
Our method	0.092 ± 0.049

Similar to [9] we also calculated the ratio of the norms of the estimated and the ground truth translations as well as the relative translation vector error. The results are indicated in Fig. 6. Table 1 furthermore compares our result against the results obtained in [3,9]. It can be observed that our method operates closest to the ideal ratio of 1 with smallest standard deviation with respect to the ratio of norms of the estimated and ground truth translations. Looking at the relative translation vector error ratio, our result is very close to the one obtained in [9], and again achieves smaller standard deviation. The better standard deviation makes us believe that part of the reason for the slightly worse mean may be biases originating from an imperfect alignment with ground truth.

As a final test, we consider it important to verify the performance of our linear bootstrapping algorithm on real data. Rather than applying it just in the very beginning of the dataset, we therefore test if the initialization method can work for arbitrary starting positions across the entire circular motion sequence. The test result of the ratio of norms of estimated and ground truth translations is indicated in Fig. 7. The mean value of the ratio equals to 0.956 and the standard deviation is 0.075. We can conclude that, at least on this sequence, the linear initialization module performs consistently well.

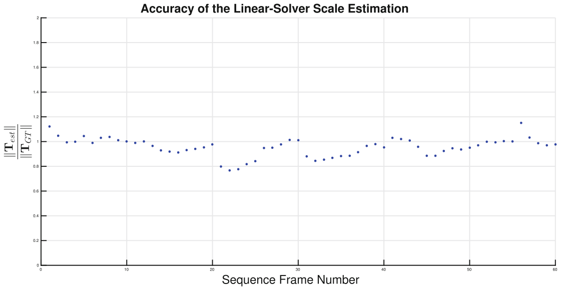


Fig. 7. Accuracy of the linear bootstrapping technique for various starting points across the entire circular motion sequence.

5 Discussion

This paper introduces a complete pipeline for motion estimation with non-overlapping multi-perspective camera systems. The main novelty lies in the fact that nearly all processing stages including bootstrapping, pose tracking, and mapping use the measurements from all cameras simultaneously. The approach is compared against a loosely coupled alternative, thus proving that the joint exploitation of the omni-directional measurements leads to superior motion estimation accuracy.

While our result represents an unprecedented integration of the paradigm of *Using many cameras as one* into a full end-to-end real-time visual odometry pipeline, there still remains space for further improvements. For example, one remaining problem is that the success of our approach still depends on sufficiently good relative rotations estimated from each camera individually at the very beginning. Future research therefore consists of pushing generalized relative pose methods towards a robust recovery of relative rotations even in the case of motion degeneracies. A further point consists of parameterizing poses with similarity transformations, which would simplify drift compensation in the case of extended periods of scale unobservability.

References

1. Agarwal, S., Mierle, K., et al.: Ceres solver. http://ceres-solver.org
2. Bradski, G.: The OpenCV library. Dr. Dobb's J. Softw. Tools (2000)
3. Clipp, B., Kim, J.H., Frahm, J.M., Pollefeys, M., Hartley, R.: Robust 6DOF motion estimation for non-overlapping, multi-camera systems. In: Proceedings of the IEEE Workshop on Applications of Computer Vision, Washington, DC, USA, pp. 1–8 (2008)
4. Engel, J., Schöps, T., Cremers, D.: LSD-SLAM: large-scale direct monocular SLAM. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8690, pp. 834–849. Springer, Cham (2014). doi:10.1007/978-3-319-10605-2_54
5. Guennebaud, G., Jacob, B., et al.: Eigen v3 (2010). http://eigen.tuxfamily.org
6. Hartley, R., Trumpf, J., Yuchao, D., Li, H.: Rotation averaging. Int. J. Comput. Vis. (IJCV) **103**(3), 267–305 (2013)
7. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision, 2nd edn. Cambridge University Press, New York (2004)
8. Horaud, R., Dornaika, F.: Hand-eye calibration. Int. J. Robot. Res. (IJRR) **14**(3), 195–210 (1995)
9. Kazik, T., Kneip, L., Nikolic, J., Pollefeys, M., Siegwart, R.: Real-time 6D stereo visual odometry with non-overlapping fields of view. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Providence, USA (2012)
10. Klein, G., Murray, D.: Parallel tracking and mapping for small AR workspaces. In: Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR), Nara, Japan (2007)
11. Kneip, L., Furgale, P.: OpenGV: a unified and generalized approach to real-time calibrated geometric vision. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Hongkong (2014)
12. Kneip, L., Furgale, P., Siegwart, R.: Using multi-camera systems in robotics: efficient solutions to the NPnP problem. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Karlsruhe, Germany (2013)
13. Kneip, L., Li, H.: Efficient computation of relative pose for multi-camera systems. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, USA (2014)
14. Kneip, L., Li, H., Seo, Y.: UPnP: an optimal $O(n)$ solution to the absolute pose problem with universal applicability. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8689, pp. 127–142. Springer, Cham (2014). doi:10.1007/978-3-319-10590-1_9
15. Kneip, L., Lynen, S.: Direct optimization of frame-to-frame rotation. In: Proceedings of the International Conference on Computer Vision (ICCV), Sydney, Australia (2013)
16. Kneip, L., Scaramuzza, D., Siegwart, R.: A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Colorado Springs, USA (2011)
17. Konolige, K., Agrawal, M., Solà, J.: Large scale visual odometry for rough terrain. In: Proceedings of the International Symposium on Robotics Research (ISRR), Hiroshima, Japan (2007)

18. Li, H., Hartley, R., Kim, J.H.: A linear approach to motion estimation using generalized camera models. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Anchorage, Alaska, USA, pp. 1–8 (2008)
19. Mur-Artal, R., Montiel, J.M.M., Tardós, J.D.: ORB-SLAM: a versatile and accurate monocular SLAM system. IEEE Trans. Robot. (T-RO) **31**(5), 1147–1163 (2015)
20. Newcombe, R., Lovegrove, S., Davison, A.: DTAM: dense tracking and mapping in real-time. In: Proceedings of the International Conference on Computer Vision (ICCV), Barcelona, Spain (2011)
21. Newcombe, R.A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A.J., Kohli, P., Shotton, J., Hodges, S., Fitzgibbon, A.: Kinectfusion: real-time dense surface mapping and tracking. In: Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR) (2011)
22. Nistér, D., Naroditsky, O., Bergen, J.: Visual odometry. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Washington, DC, USA, pp. 652–659 (2004)
23. Nistér, D., Stewénius, H.: A minimal solution to the generalized 3-point pose problem. J. Math. Imaging Vis. (JMIV) **27**(1), 67–79 (2006)
24. Pless, R.: Using many cameras as one. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Madison, WI, USA, pp. 587–593 (2003)
25. Steinbrücker, F., Sturm, J., Cremers, D.: Real-time visual odometry from dense RGB-D images. In: IEEE International Conference on Computer Vision Workshops (ICCV Workshops) (2011)
26. Stewénius, H., Engels, C., Nistér, D.: Recent developments on direct relative orientation. ISPRS J. Photogramm. Remote Sens. **60**(4), 284–294 (2006)
27. Stewénius, H., Nistér, D.: Solutions to minimal generalized relative pose problems. In: Workshop on Omnidirectional Vision (ICCV), Beijing, China (2005)
28. Strasdat, H., Davison, A.: Scale drift-aware large scale monocular SLAM. In: Proceedings of Robotics: Science and Systems (RSS), Zaragoza, Spain (2010)
29. Tykkälä, T., Audras, C., Comport, A.I.: Direct iterative closest point for real-time visual odometry. In: IEEE International Conference on Computer Vision Workshops (ICCV Workshops) (2011)
30. Whelan, T., Johannsson, H., Kaess, M., Leonard, J.J., McDonald, J.: Robust real-time visual odometry for dense RGB-D mapping. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA) (2013)