IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, Korea, October 2016.

Efficient Multi-Camera Visual-Inertial SLAM for Micro Aerial Vehicles

Sebastian Houben, Jan Quenzel, Nicola Krombach, and Sven Behnke

***Abstract*— Visual SLAM is an area of vivid research and bears countless applications for moving robots. In particular, micro aerial vehicles benefit from visual sensors due to their low weight. Their motion is, however, often faster and more complex than that of ground-based robots which is why systems with multiple cameras are currently evaluated and deployed. This, in turn, drives the computational demand for visual SLAM algorithms.**

We present an extension of the recently introduced monocular ORB-SLAM for multiple cameras alongside an inertial measurement unit (IMU). Our main contributions are: Embedding the multi-camera setup into the underlying graph SLAM approach that defines the upcoming sparse optimization problems on several adjusted subgraphs, integration of an IMU filter that supports visual tracking, and enhancements of the original algorithm in local map estimation and keyframe creation. The SLAM system is evaluated on a public stereo SLAM dataset for flying robots and on a new dataset with three mounted cameras.

The main advantages of the proposed method are its restricted computational load, high positional accuracy, and low number of parameters.

I. Introduction

Highly accurate simultaneous localization and mapping (SLAM) is one of the most important capabilities for micro air vehicles (MAV), in particular during autonomous operation. Compared to established environment-perceiving sensors, like laser scanners, cameras bear the advantage that they are lightweight, can detect most solid materials, and yield readings with comparably high frequency. Hence, they are essential in navigation, motion planning, and obstacle avoidance. However, in typical applications the camera field of view is limited. Close obstacles occlude large parts of the scene, structureless surfaces often lack visual cues, and repetitive textures complicate finding correspondences. Using multiple cameras with completely different viewing directions can help in mitigating these effects. Furthermore, these system can handle rapid motions when one camera is directed towards the axis of rotation (cf. Fig. 1). Timely processing a single video stream comes, however, already at the burden of high computational cost. This is even more critical if several cameras are deployed.

In this work, we present a multi-camera visual 6 DOF-SLAM that is able to integrate inertial measurements. The relative poses (extrinsics) among the cameras are assumed to be static and calibrated beforehand as are camera intrinsics and lens distortion.

All authors are with the Autonomous Intelligent Systems Group, Computer Science Institute VI, University of Bonn, 53113 Bonn, Germany {houben, quenzel, behnke}@ais.uni-bonn.de

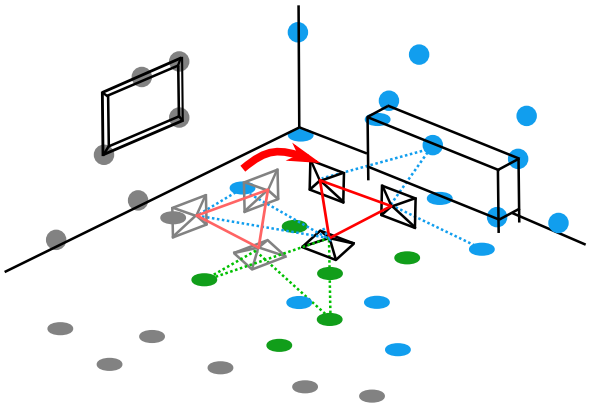


Fig. 1: The problem at hand: A (flying) robot equipped with multiple cameras traverses a scene orienting itself at distinct feature points that are tracked over time and between cameras. Observing points in multiple directions facilitates self-localization and motion estimation during complex movements. The bottom camera can track this turning manoeuvre while the stereo camera cannot.

We base our work on the release of Mur-Artal et al. [1] who introduced ORB-SLAM: a monocular SLAM approach based on ORB features [2]. It builds a map representation that is a covisibility pose graph by inserting relative pose constraints between frames that are covisible, i.e., have observed the same part of the scene. Loop closing is performed via a Bag-of-Words representation for every keyframe based on ORB descriptors.

We extend ORB-SLAM by applying the covisibility graph concept to multiple cameras, integrating IMU readings, and adapting the concept of local mapping to also account for large scenes with high interconnectivity between covisible frames. In order to restrict the computational effort, we present a new criterion that avoids creating keyframes from all cameras at every possible instance but still exploits the valuable known rigid relative poses between them. Very recently, Mur-Artal et al. have released an open-source version of their algorithm that was extended by a stereo variant. It requires, however, a pixel-wise stereo transform as pre-processing step and makes strong use of epipolar constraints between known camera poses. Our work focuses on possibly non-overlapping multi-camera systems and handles them independently.

Our field of application is directed to autonomous flying robots which is the main focus of the experiments presented

in this paper. We evaluate the algorithm on a public dataset combining a stereo camera pair with an IMU and a newly acquired dataset with three cameras (a stereo pair and a monocular camera facing downward) and an IMU.

Our contribution is an accurate self-localization, re-localization, and mapping interface with multiple cameras at moderate computational cost suitable for onboard execution. The reconstruction allows for navigation and motion planning in safe distance to obstacles. The entire algorithm has very few critical parameters, apart from the camera calibration, and can be deployed out-of-the-box in most cases.

II. Related Work

Visual Odometry (VO) and visual SLAM methods can be roughly categorized into dense, semi-dense, and feature-based approaches. Dense methods like DTAM [3] estimate for each pixel a depth value and often rely on massive hardware parallelization via GPUs to obtain real-time capabilities. The prerequisite of a GPU typically prohibits the usage of these algorithms in many MAV applications where very strict limitations on payload exist.

Instead of estimating the ego-motion from the whole image, semi dense approaches often use regions with high-intensity gradients like edges, corners, and texture. The most prominent semi-dense method is LSD-SLAM [4], which aligns images by minimizing the photometric error between the current frame and the last keyframe. The high-gradient regions are then used to estimate the inverse depth of the scene and graph optimization is utilized on keyframe poses to reduce drift over time and allow for loop closures.

Feature-based approaches typically extract robust and distinctive features [1] sparsely distributed over the whole image. These are matched between frames using feature descriptors, before the ego-motion is estimated within a RANSAC-scheme to discard outliers [5], [6]. Instead, SVO [7] uses a combination of feature-based and semi-dense approaches by first using direct alignment of previously seen feature-patches towards the current image and then finding corresponding features, before continuing to use features only. Likewise, Krombach et al. presented a feature-based approach facilitating semi-dense reconstruction with a variant of stereo LSD-SLAM [8].

Since monocular SLAM can only create the map up to scale, many researchers either use a second camera with known baseline (Stereo-LSD-SLAM [9], ORB-SLAM2 [1]) or an Inertial Measurement Unit [10], [11] to obtain the correct scale. If possible, the use of a calibrated stereo camera is most often preferred, due to the noise characteristics of an IMU [12]. Nevertheless, the IMU provides crucial information about the ego-motion and research has recently focused on tightly-coupling the IMU into the VO/SLAM system. Tanskanen et al. [11] used patches instead of points and integrated the minimization of the photometric error into an EKF, that allows to directly include the IMU measurements.

A more standard approach is taken by Leutenegger et al. [13] in OKVIS. The IMU measurements are incorporated in a probabilistic way into the non-linear optimization, linking consecutive keyframes and allowing to apply keyframe-marginalization. The reference implementation also supports multiple cameras and can improve the extrinsic calibration.

In contrast, Forster et al. [14] preintegrate all inertial measurements between consecutive keyframes to obtain a single constraint, while considering the rotations manifold structure.

Apart from OKVIS [13] the work of Kaess et al. [15] is one video-based approach that is able to incorporate multi-camera setups in real-time by restricting local bundle adjustment to the three most recent frames.

Within the categorization laid out in this paragraph, the proposed ORB-SLAM extension is a multi-camera keyframe and (ORB-)feature-based SLAM algorithm that aggregates IMU readings to a motion prior which is in turn used to accelerate the tracking and quickly retrieve the relevant local map.

III. Method

The original version of ORB-SLAM is a monocular feature-based SLAM approach that uses interest points from a multi-scale FAST or alternatively Harris corner detector to achieve a matching between two consecutive camera frames. Apart from the position of a keypoint, its ORB descriptor, scale, and dominant orientation are taken into account. A RANSAC scheme is used to estimate a pose or, in degenerate cases, a homography. After a coarse relative pose to the previous frame has been established, the algorithm tries to match the current observations against the local map that consists of triangulated points which have been observed before. Depending on the amount of unmatched points, a new keyframe is added and passed along to the mapping thread that asynchronously builds and refines the observed scene as well as the relative keyframe poses. Likewise, every keyframe is asynchronously searched for possible loop closures via a similarity score based on a bag-of-words approach over the feature descriptors and a subsequent relative pose estimation.

Multi-scale ORB features provide several useful properties that ORB-SLAM builds upon. For one, they have been shown to yield fast, stable, robust, and repeatable interest points that are mostly unaffected by changes in perspective, rotation, and blur which renders them ideal for map tracking and matching between frames. The orientation allows to define a global constraint on every matched frame as the relative rotation should approximately be the same for every interest point. In this particular setting, ORB features are also extracted from different image scales which represents another important cue for matching, tracking, and localization accuracy when the distance to the corresponding map point is known and, thus, the change in apparent size can easily be computed. As a side effect, the ORB descriptors can straightforwardly be used as complete-frame similarity score for the purpose of loop closing.

Aside from the use of the eponymous ORB features, a central characteristic of ORB-SLAM is the graph-like data structure that manages the observed scene and allows fast determination of currently relevant parts of the map. The covisibility graph is a sub-graph of this structure that encompasses all keyframes that have been created so far. Here, two keyframes are adjacent if they are covisible, i.e., they observe a common part of the scene. Furthermore, each keyframe node is adjacent to all map points it observes. No further correspondence is stored. In particular, it is impossible to reconstruct the order in which the keyframes were created from the graph structure alone as two consecutive keyframes may not be covisible if the scene in between their poses is already known. Although the keyframes and map points as nodes of the graph carry geometric information, the covisiblity graph manages common visual features which allows a simple and elegant formulation of many problems in visual SLAM. In fact, it avoids probabilistic representations that are oftentimes hard to parametrize and more costly to compute [16].

Apart from this near-topologic map interpretation within the covisibility subgraph, the pose graph tracks the variable correspondences and, thus, the sparsity of several underlying optimization problems posed by visual SLAM: During local bundle adjustment, the poses of the keyframes and the positions of the map points are optimized. During loop closing, only the relative poses of the covisibility graph spanning tree combined with the loop detections and strong covisibility edges, coined *essential graph*, are taken into account. All optimization problems are solved with the Levenberg-Marquardt implementation of the *g2o* library [17].

We denote sets and matrices with capital letters and vectors with bold letters. Let $\mathcal{F} = (T_\mathcal{F}, K_\mathcal{F})$ be a camera frame (or keyframe) with pose matrix $T_\mathcal{F} = T_{cw}$, mapping from world to camera coordinates and corresponding camera matrix $K_\mathcal{F}$. The world is defined by the first recorded frame. $\mathcal{M}$ is a map point with position $\mathbf{p}_\mathcal{M} = (p_x, p_y, p_z)^T$ in world coordinates. The projection of $\mathcal{M}$ into $\mathcal{F}$ is, hence, given as $\pi_\mathcal{F}(\mathbf{p}_\mathcal{M})$ with

$$\pi_\mathcal{F} : \mathbf{p}_w \to \mathbf{p}_c, \mathbf{p}_c = \frac{1}{p'_z}\begin{pmatrix} p'_x \\ p'_y \end{pmatrix}, \mathbf{p}' = K_\mathcal{F} T_\mathcal{F} \begin{pmatrix} \mathbf{p}_w \\ 1 \end{pmatrix}. \quad (1)$$

The nodes within the pose graph are connected by two types of edges: $\mathcal{P}$ and $\mathcal{C}$. The projection edge $\mathcal{P} = ((\mathcal{F}_\mathcal{P}, \mathcal{M}_\mathcal{P}), \mathbf{x}_\mathcal{P})$ denotes the connection between frame $\mathcal{F}$ and an observed map point $\mathcal{M}$ where $\mathbf{x}_\mathcal{P}$ is the corresponding feature point of $\mathcal{M}_\mathcal{P}$ in image coordinates. $\mathcal{C} = ((\mathcal{F}_\mathcal{C}, \mathcal{F}'_\mathcal{C}), T_\mathcal{C}, w_\mathcal{C})$ defines the weighted covisibility edges where $w_\mathcal{C}$ is the number of covisible map points and $T_\mathcal{C} = T_\mathcal{F} T_{\mathcal{F}'}^{-1}$ is the relative transform between the corresponding poses of frame $\mathcal{F}$ and $\mathcal{F}'$.

A. Multi-camera Integration

In order to introduce multiple cameras to the SLAM algorithm, we add an element c that denotes the camera the keyframe was created by: $\mathcal{F} = (T_\mathcal{F}, K_\mathcal{F}, c_\mathcal{F})$. We assume that the cameras that are used in the SLAM approach are triggered to expose simultaneously and that their relative pose $T_{cc'}$ is known. Even though it is possible to compute overlap, the proposed procedure does not handle overlapping cameras differently apart from the initialization where matching key points between stereo frames is much more reliable than monocular initialization and initially yields an accurate map scale. In case of a static arrangement, the cameras must, hence, be calibrated extrinsically.

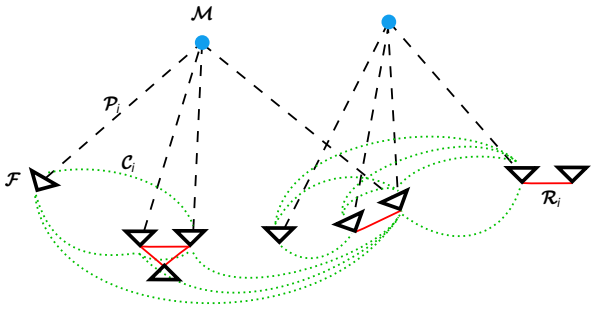


Fig. 2: The adapted pose graph with three kinds of correspondences. Solid (red) lines indicate a rigid correspondence $\mathcal{R}$ between frames from two different cameras that represent the constraint from the calibrated relative pose. Dashed (black) lines $\mathcal{P}$ connect an observed map point and the correspondent keyframe and carry the frame pose, the map point position, and the image coordinates of the corresponding key point in the frame. The dotted (green) lines denote covisibility relations $\mathcal{C}$ that carry relative poses between incident keyframes. Please note that the keyframe criterion does not necessarily add keyframes for all cameras at the same time. Thus, the size of the cliques formed by rigid edges varies.

We extend the pose graph by a new edge type that we call *rigid edge* (cf. Fig. 2). Keyframes $\mathcal{F}, \mathcal{F}'$ of different cameras from the same timestep are fully connected by rigid edges $\mathcal{R} = ((\mathcal{F}, \mathcal{F}'), T_\mathcal{R})$ that carry the respective relative pose $T_\mathcal{R} = T_{cc'}$ that was extrinsically calibrated beforehand. It represents a new strong constraint during all graph optimization steps and is used during determination of the local map (cf. Sec. III-D).

1) Tracking: After a set of newly acquired camera frames Ω has been matched against the most recent local map—establishing corresponding projection edges—a nonlinear optimization is performed to minimize the reprojection error $e_\mathcal{P}$ and refine the robot pose:

$$\begin{aligned} &\underset{T_\mathcal{F}, \mathcal{F} \in \Omega}{\arg\min} \ \left\{ e_\mathcal{P}(\Omega) + \lambda \, e_{T_\mathcal{R}}(\Omega) \right\}, \\ &e_\mathcal{P}(A) = \sum_{\mathcal{F} \in A} \sum_{\mathcal{M} \in N_\mathcal{P}(\mathcal{F})} \rho\left(\pi_\mathcal{F}(\mathbf{p}_\mathcal{M}) - \mathbf{x}_\mathcal{M}\right), \\ &e_{T_B}(A) = \sum_{\mathcal{F} \in A} \sum_{\mathcal{F}' \in N_\mathcal{B}(\mathcal{F})} d\left(T_B, T_\mathcal{F} T_{\mathcal{F}'}^{-1}\right), \end{aligned} \quad (2)$$

where $\rho(\cdot)$ denotes the robust Huber-norm and $N_\mathcal{P}(\mathcal{F}), N_\mathcal{R}(\mathcal{F})$ are the neighbouring map points of $\mathcal{F}$ connected by a projection edge and the neighbouring frames of $\mathcal{F}$ connected by a rigid edge, respectively. $d(\cdot,\cdot)$

is a measure of difference between two relative poses and the scalar parameter λ defines a trade-off between both terms.

2) Local Bundle Adjustment: After a set of keyframes Ω from different cameras has been created (cf. Sec. III-B), a local bundle adjustment is performed to refine the poses of all covisible keyframes and the positions of all observed map points:

$$\begin{aligned} &\underset{\substack{\mathbf{p}_{\mathcal{M}}, T_{\mathcal{F}} \\ \mathcal{M} \in N_{\mathcal{P}}(\mathcal{F}) \\ \mathcal{F} \in (N_{\mathcal{C}}(\Omega) \cup N_{\mathcal{R}}(\Omega))}}{\arg\min} \left\{ e_{\mathcal{P}}\left(N_{\mathcal{C}}\left(\Omega\right)\right) + \lambda_1 \; e_{T_{\mathcal{R}}}\left(\Omega\right) + \lambda_2 \; e_{\mathcal{M}}\left(\Omega\right) \right\}, \\ &e_{\mathcal{M}}\left(A\right) = e_{\mathcal{P}}\left(N_{\mathcal{C}}(N_{\mathcal{C}}(A)) \setminus N_{\mathcal{C}}(A)\right). \end{aligned} \tag{3}$$

Please note that the last sum imposes a correspondence between the local and the non-local map.

3) Loop Closing: When a loop closure between two keyframes $\mathcal{F}_L, \mathcal{F}'_L$ was found, the pose graph is optimized by only considering the essential graph, i.e., the spanning tree of the covisibility graph with additional strong covisibility edges. We denote its edges, including the loop edge itself, by $\mathcal{E} \ni (\mathcal{F}_L, \mathcal{F}'_L)$ and its set of keyframe nodes by $\Omega \supset \{\mathcal{F}_L, \mathcal{F}'_L\}$. The optimization problem that takes the newly defined rigid edges into account is

$$\underset{T_{\mathcal{F}}, \forall \mathcal{F} \in \Omega}{\arg\min} \left\{ e_{\mathcal{T}_{\mathcal{C}}}\left(\mathcal{E}\right) + e_{\mathcal{T}_{\mathcal{R}}}\left(\mathcal{R}\right) \right\}. \tag{4}$$

B. Keyframe Creation

In monocular ORB-SLAM, a new keyframe is created if and when a given percentage μ of the computed features cannot be assigned to the local map and is, hence, considered novel. This threshold is derived from the corresponding matching ratio τ of the most covisible keyframe in order to adapt to regions of the scene with varying quality in keypoint matching. Since the number of involved keyframes is the main factor for computational complexity in all optimization stages, the local mapping thread contains a culling criterion that may remove keyframes afterwards when their contribution to the map is considered too low. We found the reliance on only the most covisible keyframe to be instable at times. A weighted average over all covisible keyframes with the number of shared map point observations provided a smoother thresholding:

$$\tau_{\mathcal{F}} := \sum_{\mathcal{F}' \in N_{\mathcal{C}}(\mathcal{F})} w_{\mathcal{F}'} \mu_{\mathcal{F}'}, \tag{5}$$

where $\mu_{\mathcal{F}}$ is the ratio of matched features in keyframe $\mathcal{F}$.

With more than one camera, two conflicting objectives need to be balanced. On the one hand, the number of keyframes should still be as low as possible as it governs the size of all optimization problems, on the other hand, the accurately known relative pose between two or more isochronically created keyframes is a very valuable constraint during pose graph optimization due to the static robot configuration which can be calibrated very precisely.

In order to coordinate the keyframe creation events, we propose a scheme where the matching ratio $\mu_{\mathcal{F}_i}$ of each camera frame $\mathcal{F}_i$ is compared to two thresholds $m_{low}\tau_{\mathcal{F}_i}$ and $m_{high}\tau_{\mathcal{F}_i}$ with $m_{low} < m_{high}$. If the ratio of at least one frame drops below m_{low}, all frames with $\mu_{\mathcal{F}_i} < m_{high}$ are used to create a keyframe, i.e., the creation of keyframes is preponed for camera frames with a medium matching ratio $m_{low} < \mu_{\mathcal{F}_i} < m_{high}$.

C. IMU Integration

Apart from control purposes of a robot, the usage of an IMU exhibits a number of benefits, especially in visually challenging scenarios including, e.g., high differences in lighting within one or between consecutive images, motion blur due to fast movements, or repetitive surface textures yielding many local minima for triangulation. For example, if the scene features are far away, the triangulation can estimate the orientation quite well while the translation remains imprecise. The IMU can then improve the position accuracy and prevent tracking loss. In a multi-camera setup, the IMU provides viable information to bridge contradicting feature matchings or frames missing due to insufficient exposure or rapidly changing lighting conditions.

Instead of using one filter for both the attitude and the position, we combine two existing filters. The attitude is estimated by the quaternion based complementary filter of [18]. Given the current attitude $R_{bw} \in \mathbb{R}^{3\times3}$, a fixed covariance filter is employed to estimate the current position $\mathbf{t}_{bw} \in \mathbb{R}^3$ [19]. We chose this approach for its speed and capabilities to estimate the gyroscope and accelerator noise, as well as straightforward parameter choice. Currently, we only use five parameters for the position filter and two for the attitude filter—in contrast to standard Kalman filter approaches which need accurate covariance matrices that are often hard to obtain.

During typical outdoor applications, we can rely on the magnetometer as an absolute reference. Indoors, the magnetometer is, however, often useless due to interference by metal in walls or objects. For flight in proximity to walls and obstacles, we use the orientation of our SLAM system to provide a reference w.r.t the first frame. Obviously, this reference underlies drift but is much more precise than the accumulated gyroscope drift. This is especially important under frequent and large rotations.

The combined filter provides the current 3D pose $T_{bw} = (R_{bw}, \mathbf{t}_{bw})$ w.r.t. the first frame with gravity pointing downward in direction of the negative z-axis of a right handed coordinate system with the x- and y-axis pointing forward and leftwards, respectively. On arrival of the next frame, our system tries to match the current features against the local map that was determined for the previous frame. For this purpose, the IMU yields a prior on the current motion estimate which can speed up feature-matching. Given the coarse pose and the extrinsic calibration between body and camera

$$T_{\mathcal{F}} = T_{c_{\mathcal{F}}b} \cdot T_{bw}, \forall \mathcal{F} \in \Omega, \tag{6}$$

the local map points are reprojected onto the current image plane and matched with the extracted feature points. If too few matches were found, we perform a full frame-to-frame

matching, before we proceed with the remainder of the visual SLAM pipeline.

D. Local Mapping

It is essential for successful self-localization to be able to access currently relevant parts of the map fast and reliably. The covisibility graph as defined in Sec. III allows to do this seamlessly without knowledge of the current pose or a sensor model that encompasses a range characteristic. In the original ORB-SLAM, one starts with a coarse pose estimate by tracking features from the previous camera frame. This allows to backproject the map points from the previous local map estimate for matching them with the keypoints of the current frame. The observations of all covisible keyframes and their closure, i.e., the set of the adjacent keyframes in the covisibility graph, form the local map.

As detailed below, we refine this procedure that is implemented in the original ORB-SLAM as it quickly becomes intractable with multiple cameras since the number of involved keyframes grows fast. Furthermore, we found that the number of directly covisible keyframes may already be high in scenes recorded from far and near distance and that a large number of keyframes from the covisible closure may be irrelevant for the currently observed part of the scene. To address these issues, we add another criterion that controls when to include a keyframe to the local map. At least one of the observed map points must fulfill the following conditions:

- It must lie in the frustrum of the current frame which can be checked by comparing the reprojection of the map point against the image borders.
- Given the distance of the respective map point and the scale of its ORB feature, the apparent scale when projected into the current camera frame can be computed. This scale must lie between the minimum and maximum observable scale of the keypoints of that frame.
- The angle between the viewing direction of the map point and the average viewing direction of all other keyframes that are adjacent must fall below a given threshold.

If these criteria are satisfied, the regarded keyframe does indeed observe a currently relevant part of the scene. The computational cost to check the conditions is lightweight since intermediate results, like the reprojected coordinates and the apparent scale, must be computed anyway and can be stored and re-used later on.

Starting from the directly covisible keyframes, we initiate a breadth-first-search on covisibility edges and continue to add adjacent keyframes with at least one map point that fulfils the above stated conditions. The search stops if the respective keyframe does not contain any relevant scene points.

IV. Experiments

All experiments were performed on an Intel Core i7-4710MQ CPU @ 2.50GHz desktop PC with 16 GB RAM. It should be noted that the algorithm does not make use of GPU implementations. The video stream was asynchronously replayed at original speed to simulate realistic runtime constraints.

A. Stereo and IMU

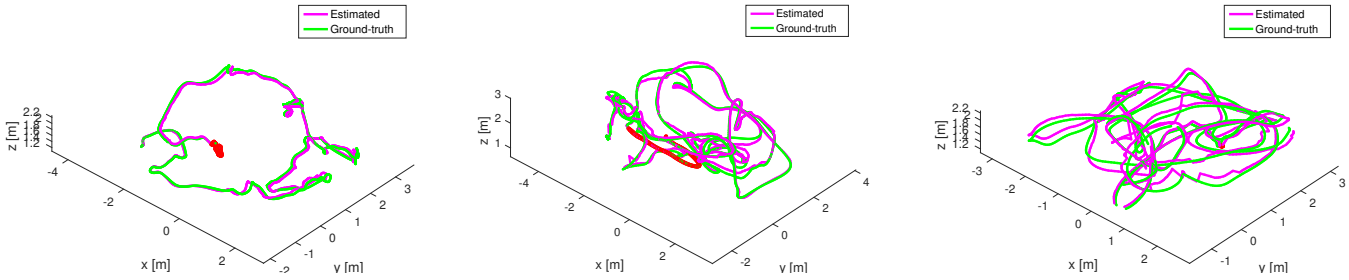
We use the publicly available dataset from ETH Zürich [20] obtained from dynamic flights in a small cluttered room. The sequences are recorded with an Asctec Firefly hex-rotor copter. The MAV pose is tracked with a Vicon 6D motion capture system at $100\,\mathrm{Hz}$. Depending on the density of the local map, the main tracking thread performed at an average frame rate of 15–$20\,\mathrm{Hz}$, where $20\,\mathrm{Hz}$ is also the rate of the cameras. Table I presents the results of our method in comparison to those of state-of-the-art algorithms that we adopt partly from the paper by Krombach et al. [8]. It contains the absolute trajectory error (ATE) between the estimated and the reference trajectory. In all cases both trajectories were aligned by a rigid transform that minimizes their distance. For monocular SLAM algorithms, an additional scale was estimated. For the evaluation the published pose estimate was recorded, thus, allowing to use only the sequence information up to that time.

Our adapted version of ORB-SLAM is able to self-localize the copter with a level of accuracy in the range of established SLAM and VO methods, clearly outperforming LIBVISO 2, LSD-SLAM, Mono-ORB-SLAM, and S-PTAM. Unlike the approach from this paper where IMU information is used as a prior, OKVIS computes a tight coupling between image and IMU information which performs very well on the test sequences. ORB-SLAM2 makes direct use of the stereo pair in order to obtain a depth estimate, while our approach allows for an arbitrary setup that contains the stereo sensor as special case. However, the aptitude of both methods is similar. Sequence *V1 03* contains large motion blur and changing lighting conditions which is handled very well with the fast and robust ORB features. Sequence *V2 03* is even more challenging. Six of seven methods, including ours, lose their track during the sequence.

B. Monocular, Stereo, and IMU

We use an AscTec Neo copter with a visual-inertial stereo camera pair facing front and a monocular wide-angle camera facing downward for inner-building flights in a room of size $8\times10\,\mathrm{m}$ and $4\,\mathrm{m}$ in height. Ground truth is obtained via a Vicon 6D motion capture system at a rate of $100\,\mathrm{Hz}$. Both the extrinsic and the intrinsic calibration of the three cameras where obtained with an extended version of the Kalibr calibration toolbox [22] that is able to estimate the extrinsics of non-overlapping cameras when observing a calibration pattern, which was projected to a flat wall surface.

The IMU yielded an accurate pose that allowed to match the map points from the previous frame in $33\,\%$ of all cases and to match its entire local map in $95\,\%$ of the cases which is only done when matching with the previous frame fails. Thus, using the IMU prior is beneficial as it avoids the extensive RANSAC-based feature matching between consecutive frames in many cases and without any drop in tracking accuracy.



(a) Sequence V02 01 (easy) (b) Sequence V01 02 (medium) (c) Sequence V02 02 (medium)

Fig. 3: Comparison of several trajectories from our approach with the given ground truth.

TABLE I: ATE results on EuRoC / ASL dataset

EuRoC Dataset	Ours	LIBVISO2 [5]	LSD-SLAM [4]∗	ORB-SLAM [1]	ORB-SLAM 2 [1]	OKVIS [13]	S-PTAM [21]
V1 (easy)	0.11 (0.10)	0.26 (0.24)	0.19 (0.10)	0.08 (0.08)	0.09 (0.09)	0.08 (0.08)	0.20 (0.17)
V1 (medium)	0.14 (0.10)	0.17 (0.15)	0.98 (0.92)	0.73 (0.67)	0.17 (0.15)	0.61 (0.48)	0.58 (0.50)
V1 (difficult)	0.36 (0.24)	0.24 (0.21)	X	X	X	0.14 (0.13)	X
V2 (easy)	0.11 (0.10)	0.53 (0.51)	0.45 (0.41)	0.16 (0.15)	0.09 (0.08)	0.10 (0.10)	1.88 (1.56)
V2 (medium)	0.27 (0.23)	0.92 (0.75)	0.51 (0.48)	X	0.22 (0.21)	0.18 (0.17)	X
V2 (difficult)	X	X	X	X	X	0.24 (0.23)	X
Mean	0.20 (0.15)	0.42 (0.37)	0.53 (0.48)	0.32 (0.30)	0.14 (0.13)	0.22 (0.20)	0.89 (0.74)

∗ numbers taken from [8]

TABLE II: Performance on own dataset with three cameras and a stereo camera pair only.

Dataset	ATE (median) mono + stereo	ATE (median) stereo
Spiraling flight facing outward	0.08 (0.07)	0.14 (0.14)
Flight into scaffolding and fast manoeuvring	0.08 (0.07)	0.07 (0.07)
Chasing the author	0.02 (0.02)	0.03 (0.02)
Aisle of scaffolding covered by tarpaulin	0.11 (0.08)	0.13 (0.09)
Aisle of mattresses (loop)	0.09 (0.05)	0.10 (0.07)
Gate from scaffolding	0.08 (0.08)	0.06 (0.06)
Aisle of mattresses (double loop)	0.09 (0.06)	0.09 (0.08)
Mean	0.08 (0.06)	0.09 (0.08)

We measured the ATE with respect to the Vicon system in several piloted flights with reasonably fast manoeuvring. Table II shows the results for a stereo and a three-camera system (stereo and mono). The sequences included flying into and through a scaffolding and chasing the first author of this paper. The presence of moving objects results in map points that are incorrectly assumed to be static, but which are then recognized as outliers and removed from the map.

As shown in Tab. II, both setups allowed for an accurate trajectory estimation at a frame rate of 12–18 Hz due to the load of two and three cameras, respectively. The multi-camera setup yields slightly more precise trajectories as there are situations where the estimation benefits from multiple cameras, albeit seldomly. Figure 3 shows some example trajectories with their respective ground truth from the public ETH dataset, Fig. 4 inserts them into an occupancy map with the map points estimated by our own method and LSD-SLAM for reference.

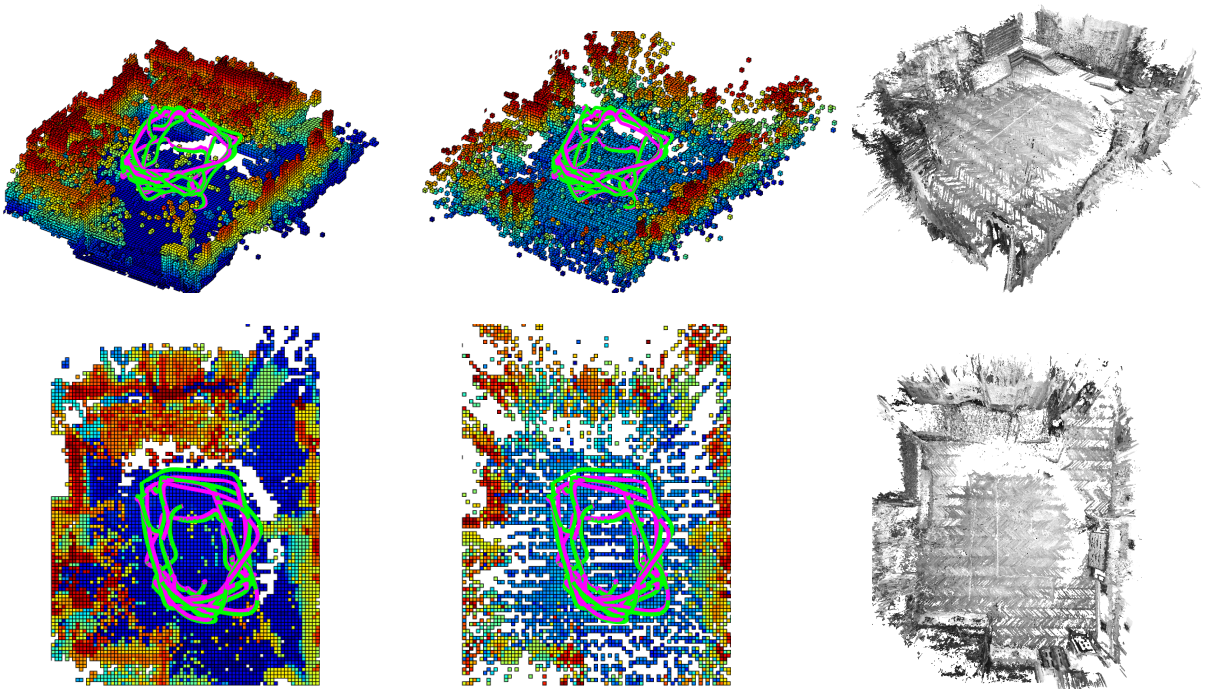
V. Conclusions

We have demonstrated how an adaptation of the recently introduced ORB-SLAM algorithm can be used to yield an accurate and fast simultaneous localization and mapping for micro aerial vehicles. An inspection of the constructed map reveals that it is sufficient for path planning with moderate safety margin, but falls short when operating near obstacles.

The algorithm shows the necessary robustness to handle very dynamic flight manoeuvres. Adapting the problem size to the current flight situation by dynamically changing the number of key points per frame yields potential for improvement. Preliminary results on the copter CPU show that this load shedding approach is able to provide a pose estimation at 15–20 Hz, however, further tests in real-life flying scenarios will have to be performed. Secondly, we will address the shortcomings of the mapping approach by feeding the computed poses and visual cues into a semi-dense scene reconstruction framework [4].

VI. Acknowledgments

This work was supported by grants BE 2556/7 and BE 2556/8 of the German Research Foundation (DFG) and 608849 of the European Union's 7th FP.



(a) Semi-dense direct reconstruction based on [8]

(b) Ours

(c) Intensity value overlay

Fig. 4: Result for our first dataset showing the occupancy grid map by a semi-dense direct method (a) and the proposed feature-based method (b). The color encodes the height over the ground plane. The green curve denotes the Vicon ground truth, the magenta curve the estimated trajectory. Figure (c) shows the reconstruction from (a) overlaid with intensity values.

REFERENCES

[1] R. Mur-Artal, J. Montiel, and J. D. Tardos, "ORB-SLAM: A versatile and accurate monocular SLAM system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[2] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *IEEE International Conference on Computer Vision (ICCV)*, 2011, pp. 2564–2571.

[3] R. Newcombe, S. Lovegrove, and A. Davison, "DTAM: Dense tracking and mapping in real-time," in *IEEE International Conference on Computer Vision*, 2011, pp. 2320–2327.

[4] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," in *European Conference on Computer Vision*, 2014.

[5] A. Geiger, J. Ziegler, and C. Stiller, "StereoScan: Dense 3D reconstruction in real-time," in *Intelligent Vehicles Symposium*, 2011.

[6] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *Proc. Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'07)*, 2007.

[7] C. Forster, M. Pizzoli, and D. Scaramuzza, "SVO: Fast semi-direct monocular visual odometry," in *IEEE International Conference on Robotics and Automation*, 2014, pp. 15–22.

[8] N. Krombach, D. Droeschel, and S. Behnke, "Combining feature-based and direct methods for semi-dense real-time stereo visual odometry," in *International Conference on Intelligent Autonomous Systems*, 2016.

[9] J. Engel, J. Stueckler, and D. Cremers, "Large-scale direct SLAM with stereo Cameras," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2015.

[10] S. Shen, N. Michael, and V. Kumar, "Tightly-coupled monocular visual-inertial fusion for autonomous flight of rotorcraft MAVs," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 5303–5310.

[11] P. Tanskanen, T. Naegeli, M. Pollefeys, and O. Hilliges, "Semi-direct EKF-based monocular visual-inertial odometry," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2015, pp. 6073–6078.

[12] S. Shen, Y. Mulgaonkar, N. Michael, and V. Kumar, "Vision-based state estimation for autonomous rotorcraft MAVs in complex environments," in *IEEE International Conference on Robotics and Automation*, 2013, pp. 1758–1764.

[13] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visualinertial odometry using nonlinear optimization," *The International Journal of Robotics Research*, vol. 34, no. 3, pp. 314–334, 2015.

[14] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "IMU preintegration on manifold for efficient visual-inertial maximum-a-posteriori estimation," in *Robotics: Science and Systems XI*, 2015.

[15] M. Kaess and F. Dellaert, "Visual slam with a multi-camera rig," 2006.

[16] C. Mei, G. Sibley, and P. Newman, "Closing loops without places," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010, pp. 3738–3744.

[17] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g2o: A general framework for graph optimization," in *IEEE International Conference on Robotics and Automation*, 2011, pp. 3607–3613.

[18] R. Valenti, I. Dryanovski, and J. Xiao, "Keeping a good attitude: A quaternion-based orientation filter for IMUs and MARGs," *Sensors*, vol. 15, no. 8, pp. 19 302–19 330, 2015.

[19] L. Meier, P. Tanskanen, F. Fraundorfer, and M. Pollefeys, "Pixhawk: A system for autonomous flight using onboard computer vision," in *IEEE International Conference on Robotics and Automation*, 2011, pp. 2992–2997.

[20] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart, "The EuRoC micro aerial vehicle datasets," *International Journal of Robotics Research*, 2016.

[21] T. Pire, T. Fischer, J. Civera, P. De Cristoforis, and J. J. Berlles, "Stereo parallel tracking and mapping for robot localization," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2015, pp. 1373–1378.

[22] P. Furgale, J. Rehder, and R. Siegwart, "Unified temporal and spatial calibration for multi-sensor systems," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 1280–1286.