ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial and Multi-Map SLAM

Carlos Campos*, Richard Elvira*, Juan J. Gómez Rodríguez, José M.M. Montiel and Juan D. Tardós

Abstract—This paper presents ORB-SLAM3, the first system able to perform visual, visual-inertial and multi-map SLAM with monocular, stereo and RGB-D cameras, using pin-hole and fisheye lens models.

The first main novelty is a feature-based tightly-integrated visual-inertial SLAM system that fully relies on Maximum-a-Posteriori (MAP) estimation, even during the IMU initialization phase. The result is a system that operates robustly in real time, in small and large, indoor and outdoor environments, and is 2 to 5 times more accurate than previous approaches.

The second main novelty is a multiple map system that relies on a new place recognition method with improved recall. Thanks to it, ORB-SLAM3 is able to survive to long periods of poor visual information: when it gets lost, it starts a new map that will be seamlessly merged with previous maps when revisiting mapped areas. Compared with visual odometry systems that only use information from the last few seconds, ORB-SLAM3 is the first system able to reuse in all the algorithm stages all previous information. This allows to include in bundle adjustment co-visible keyframes, that provide high parallax observations boosting accuracy, even if they are widely separated in time or if they come from a previous mapping session.

Our experiments show that, in all sensor configurations, ORB-SLAM3 is as robust as the best systems available in the literature, and significantly more accurate. Notably, our stereo-inertial SLAM achieves an average accuracy of 3.6 cm on the EuRoC drone and 9 mm under quick hand-held motions in the room of TUM-VI dataset, a setting representative of AR/VR scenarios. For the benefit of the community we make public the source code.

I. INTRODUCTION

Intense research on Visual Simultaneous Localization and Mapping systems (SLAM) and Visual Odometry (VO), using cameras alone or in combination with inertial sensors, has produced during the last two decades excellent systems, with increasing accuracy and robustness. Modern systems rely on Maximum a Posteriori (MAP) estimation, which in the case of visual sensors corresponds to Bundle Adjustment (BA), either geometric BA that minimizes feature reprojection error, in feature-based methods, or photometric BA that minimizes the photometric error of a set of selected pixels, in direct methods.

With the recent emergence of VO systems that integrate loop-closing techniques, the frontier between VO and SLAM is more diffuse. The goal of Visual SLAM is to use the sensors on-board a mobile agent to build a map of the environment and compute in real-time the pose of the agent in that map. In contrast, VO systems put their focus on computing the agent’s ego-motion, not on building a map. The big advantage of a SLAM map is that it allows matching and using in BA previous observations performing three types of data association (extending the terminology used in [1]):

- **Short-term data association**, matching map elements obtained during the last few seconds. This is the only data association type used by most VO systems, that forget environment elements once they get out of view, resulting in continuous estimation drift even when the system moves in the same area.
- **Mid-term data association**, matching map elements that are close to the camera whose accumulated drift is still small. These can be matched and used in BA in the same way than short-term observations and allow to reach zero drift when the systems moves in mapped areas. They are the key of the better accuracy obtained by our system compared against VO systems with loop detection.
- **Long-term data association**, matching observations with elements in previously visited areas using a place recognition technique, regardless of the accumulated drift (loop detection) or even if the tracking was lost (relocation). Long term matchings allow to reset the drift and to correct the loop using pose-graph (PG) optimization, or more accurately, using BA. This is the key of SLAM accuracy in medium and large loopy environments.

In this work we build on ORB-SLAM [2, 3] and ORB-SLAM Visual-Inertial [4], the first visual and visual-inertial systems able to take full profit of short-term, mid-term and long-term data association, reaching zero drift in mapped areas. Here we go one step further providing **multi-map data association**, which allows us to match and use in BA map elements coming from previous mapping sessions, achieving the true goal of a SLAM system: building a map than can be used later to provide accurate localization.

This is essentially a system paper, whose most important contribution is the ORB-SLAM3 library itself [5], the most complete and accurate visual, visual-inertial and multi-map SLAM system to date (see table 1). The main novelties of ORB-SLAM3 are:

- **A monocular and stereo visual-inertial SLAM system** that fully relies on Maximum-a-Posteriori (MAP) estimation, even during the IMU (Inertial Measurement Unit) initialization phase. The initialization method proposed was previously presented in [6]. Here we add its integration with ORB-SLAM visual-inertial [4], the extension

* Both authors contributed equally to this work.

The authors are with the Instituto de Investigación en Ingeniería de Aragón (I3A), Universidad de Zaragoza, María de Luna 1, 50018 Zaragoza, Spain. E-mail: {campos, richard, jjgomez, josemari, tardos}@unizar.es.

This work was supported in part by the Spanish government under grants PGC2018-096367-B-I00 and DPI2017-91104-EXP, and by Aragón government under grant DGA_T45-17R.
to stereo-inertial SLAM, and a thorough evaluation in public datasets. Our results show that the monocular and stereo visual-inertial systems are extremely robust and significantly more accurate than other visual-inertial approaches, even in sequences without loops.

- **High-recall place recognition.** Many recent visual SLAM and VO systems [2], [52] solve place recognition using the DBoW2 bag of words library [9]. DBoW2 requires temporal consistency, matching three consecutive keyframes to the same area, before checking geometric consistency, boosting precision at the expense of recall. As a result, the system is too slow at closing loops and reusing previous maps. We propose a novel place recognition algorithm, in which candidate keyframes are first checked for geometrical consistency, and then for local consistency with three covisible keyframes, that in most occasions are already in the map. This strategy increases recall and densifies data association improving map accuracy, at the expense of a slightly higher computational cost.

- **ORB-SLAM Atlas.** The first complete multi-map SLAM system able to handle visual and visual-inertial systems, in monocular and stereo configurations. The Atlas can represent a set of disconnected maps, and apply to them all the mapping operations smoothly: place recognition, camera relocalization, loop closure and accurate seamless map merging. This allows to automatically use and combine maps built at different times, performing incremental multi-session SLAM. A preliminary version of ORB-SLAM Atlas for visual sensors was presented in [10]. Here we add the new place recognition system, the visual-inertial multi-map system and its evaluation on public datasets.

- **An abstract camera representation** making the SLAM code agnostic of the camera model used, and allowing to add new models by providing their projection, unprojection and Jacobian functions. We provide the implementations of pinhole [11] and fisheye [12] models.

All these novelties, together with a few code improvements make ORB-SLAM3 the new reference visual and visual-inertial open-source SLAM library, being as robust as the best systems available in the literature, and significantly more accurate, as shown by our experimental results in section VII. We also provide comparisons between monocular, stereo, monocular-inertial and stereo-inertial SLAM results that can be of interest for practitioners.

II. RELATED WORK

Table I presents a summary of the most representative visual and visual-inertial systems, showing the main techniques used for estimation and data association. The qualitative accuracy and robustness ratings included in the table are based, for modern systems, on the comparisons reported in section VII and for classical systems, on previous comparisons in the literature [2], [52].

A. Visual SLAM

Monocular SLAM was first solved in MonoSLAM [13], [14], [53] using an Extended Kalman Filter (EKF) and Shi-Tomasi points that were tracked in subsequent images doing a guided search by correlation. Mid-term data association was significantly improved using techniques that guarantee that the feature matches used are consistent, achieving hand-held visual SLAM [54], [55].

In contrast, keyframe-based approaches estimate the map using only a few selected frames, discarding the information coming from intermediate frames. This allows to perform the more costly, but more accurate, BA optimization at keyframe rate. The most representative system was PTAM [16], that split camera tracking and mapping in two parallel threads. Keyframe-based techniques are more accurate than filtering for the same computational cost [56], becoming the gold standard in visual SLAM and VO. Large scale monocular SLAM was achieved in [57] using sliding-window BA, and in [58] using a double-window optimization and a covisibility graph.

Building on these ideas, ORB-SLAM [2], [3] uses ORB features, whose descriptor provides short-term and mid-term data association, builds a covisibility graph to limit the complexity of tracking and mapping, and performs loop-closing and relocalization using the bag-of-words library DBoW2 [9], achieving long-term data association. To date is the only visual SLAM system integrating the three types of data association, which we believe is the key of its excellent accuracy. In this work we improve its robustness in pure visual SLAM with the new Atlas system that starts a new map when tracking is lost, and its accuracy in loopy scenarios with the new high-recall place recognition method.

Direct methods do not extract features, but use directly the pixel intensities in the images, and estimate motion and structure by minimizing a photometric error. LSD-SLAM [20] was able to build large scale semi-dense maps using high gradient pixels. However, map estimation was reduced to pose graph, achieving lower accuracy than PTAM and ORB-SLAM [2]. The hybrid system SVO [23], [24] extracts FAST features, uses a direct method to track features, and any pixel with nonzero intensity gradient, from frame to frame, and optimizes camera trajectory and 3D structure using reprojection error. SVO is extremely efficient, but being a pure VO method, it only performs short-term data association, which limits its accuracy. Direct Sparse Odometry DSO [27] is able to compute accurate camera poses in situations where point detectors perform poorly, enhancing robustness in low textured areas or against blurred images. It introduces local photometric BA that simultaneously optimizes a window of 7 recent keyframes and the inverse depth of the points. Extensions of this work include stereo [29], loop-closing using features and DBoW2 [59], [60], and visual-inertial odometry [46]. Direct Sparse Mapping DSM [31] introduces the idea of map reusing in direct methods, showing the importance of mid-term data association. In all cases, the lack of integration of short, mid, and long-term data association results in lower accuracy than our proposal (see section VII).
Table I: Summary of the most representative visual (top) and visual-inertial (bottom) systems, in chronological order.

| SLAM or VO | Pixels used | Data association | Estimation | Relocation | Loop closing | Multi Maps | Stereo | Mono IMU | Stereo IMU | Fish-eye | Accuracy | Robustness | Open source |
|------------|-------------|------------------|------------|------------|-------------|-----------|--------|---------|-----------|----------|---------|-----------|------------|
| Mono-SLAM  | SLAM        | Shi Tomasi       | Correlation| EKF        | -           | -         | -      | -       | -         | -        | Fair    | Fair      | [13], [14] |
| PTAM       | SLAM        | FAST             | Pyramid SSD| BA         | Thumbnail   | -         | -      | ✓       | -         | -        | Very Good| Fair      | [16]–[18]  |
| LSD-SLAM   | SLAM        | Edgelets         | Direct     | PG         | -           | ✓         | ✓      | -       | -         | -        | Good    | Good      | 20, [21]  |
| SVO        | VO          | FAST+ Hi. grad.  | Direct     | Local BA   | -           | ✓         | ✓      | ✓       | -         | -        | Very Good| Very Good | [23], [24] |
| ORB-SLAM2  | SLAM        | ORB              | Descriptor | Local BA   | DBoW2       | ✓         | ✓      | ✓       | -         | -        | Exc.    | Very Good | [3], [3]  |
| DSO        | VO          | High grad.       | Direct     | Local BA   | ✓           | ✓         | ✓      | ✓       | -         | -        | Good    | Very Good | [27], [29] |
| DSM        | SLAM        | High grad.       | Direct     | Local BA   | -           | -         | -      | -       | -         | -        | Very Good| Very Good | [31]       |
| MSCKF      | VO          | Shi Tomasi       | Cross correlation| EKF        | -           | -         | ✓      | ✓       | ✓         | -        | Fair    | Very Good | [33]–[36]  |
| OKVIS      | VO          | BRISK            | Descriptor | Local BA   | -           | -         | -      | ✓       | ✓         | ✓        | Good    | Very Good | [38], [39] |
| ROVIO      | VO          | Shi Tomasi       | Direct     | EKF        | -           | -         | -      | ✓       | ✓         | ✓        | Good    | Very Good | [41], [42] |
| ORBSLAM-VI | SLAM        | ORB              | Descriptor | Local BA   | DBoW2       | ✓         | ✓      | ✓       | ✓         | ✓        | Very Good| Good      | [4]       |
| VINS-Fusion| VO          | Shi Tomasi       | KLT        | Local BA   | DBoW2       | ✓         | ✓      | ✓       | ✓         | ✓        | Very Good| Exc.      | [7], [44] |
| VI-DSO     | VO          | High grad.       | Direct     | Local BA   | -           | -         | -      | -       | ✓         | -        | Very Good| Exc.      | [46]       |
| BASALT     | VO          | FAST             | KLT (LSSD) | Local BA   | -           | ✓         | ✓      | ✓       | ✓         | ✓        | Very Good| Exc.      | [47]       |
| Kimera     | VO          | Shi Tomasi       | KLT        | Local BA   | DBoW2       | ✓         | ✓      | ✓       | ✓         | ✓        | Exc.    | Exc.      | [48]       |
| ORBSLAM3 (ours) | SLAM | ORB              | Descriptor | Local BA | DBoW2       | ✓         | ✓      | ✓       | ✓         | ✓        | Exc.    | Exc.      | [5]       |

1 Last source code provided by a different author. Original software is available at [50].
2 Source code available only for the first version, SVO 2.0 is not open source.
3 MSCKF is patented [51]. Only a re-implementation by a different author is available as open source.

B. Visual-Inertial SLAM

The combination of visual and inertial sensors provide robustness to poor texture, motion blur and occlusions, and in the case of monocular systems, make scale observable. Research in tightly coupled approaches can be traced back to MSCKF [33] where the EKF quadratic cost in the number of features is avoided by feature marginalization. The initial system was perfected in [34] and extended to stereo in [35], [36]. The first tightly coupled visual odometry system based on keyframes and bundle adjustment was OKVIS [38], [39] that is also able to use monocular and stereo vision. While these systems rely on features, ROVIO [41], [42] feeds an EKF with photometric error using direct data association.

ORB-SLAM-VI [4] presented for the first time a visual-inertial SLAM system able to reuse a map with short-term, mid-term and long-term data associations, using them in an accurate local visual-inertial BA based on IMU preintegration [61], [62]. However, its IMU initialization technique was too slow, taking 15 seconds, which harmed robustness and accuracy. Faster initialization techniques were proposed in [63], [64], based on a closed-form solution to jointly retrieve scale, gravity, accelerometer bias and initial velocity, as well as visual features depth. Crucially, they ignore IMU noise properties, and minimize the 3D error of points in space, and not their reprojection errors, that is the gold standard in feature-based computer vision. Our previous work [65] shows that this results in large unpredictable errors.

VINS-Mono [7] is a very accurate and robust monocular-inertial odometry system, with loop-closing using DBoW2 and 4 DoF pose-graph optimization, and map-merging. Feature tracking is performed with Lucas-Kanade tracker, being slightly more robust than descriptor matching. In VINS-Fusion [44] it has been extended to stereo and stereo-inertial. Kimera [8] is an novel outstanding metric-semantic mapping system, but its metric part consists in stereo-inertial odometry plus loop closing with DBoW2 and pose-graph optimization, achieving similar accuracy to VINS-Fusion. The recent BASALT [47] is a stereo visual-inertial odometry system that extracts non-linear factors from visual-inertial odometry to use them in BA, and closes loops matching ORB features, achieving very good to excellent accuracy.

Recently, VI-DSO [66] extends DSO to visual-inertial odometry. They propose a bundle adjustment which combines inertial observations with the photometric error in selected high gradient pixels, what renders very good accuracy. As the
information in high gradient pixels is successfully exploited, the robustness in scene regions with poor texture is also boosted. Their initialization method relies on visual-inertial BA and takes 20-30 seconds to converge within 1% scale error.

In this work we build on ORB-SLAM-VI and extend it to stereo-inertial SLAM. We propose a novel fast initialization method based on Maximum-a-Posteriori (MAP) estimation that properly takes into account visual and inertial sensor uncertainties, and estimates the true scale with 5% error in 2 seconds, converging to 1% scale error in 15 seconds. All other systems discussed above are visual-inertial odometry methods, some of them extended with loop closing, and lack the capability of using mid-term data associations. We believe this, together with our fast and precise initialization, is the key of the better accuracy consistently obtained by our system, even in sequences without loops.

C. Multi-Map SLAM

The idea of adding robustness to track losses during exploration by means of map creation and fusion was first proposed in [67] within a filtering approach. One of the first keyframe-based multi-map systems was [68], but the map initialization was manual, and the system was not able to merge or relate the different sub-maps. Multi-map capability has been researched as a component of collaborative mapping systems, with several mapping agents and a central server that only receives information [69] or with bidirectional information flow as in C2TAM [70]. MOARSLAM [71] proposes a robust stateless client-server architecture for collaborative multiple-device SLAM, but the main focus is the software architecture, not reporting accuracy results.

More recently, CCM-SLAM [72], [73] proposes a distributed multi-map system for multiple drones with bidirectional information flow, built on top of ORB-SLAM. Their focus is on overcoming the challenges of limited bandwidth and distributed processing, while ours is on accuracy and robustness, achieving significantly better results on the EuRoC dataset. ORB-SLAMMM [74] also proposes a multi-map extension of ORB-SLAM2, but keeps sub-maps as separated entities, while we perform seamless map merging, building a more accurate global map.

VINS-Mono [7] is a visual odometry system with loop closing and multi-map capabilities that rely on the place recognition library DBoW2 [9]. Our experiments show that ORB-SLAM3 is 2.6 times more accurate than VINS-Mono in monocular-inertial single-session operation on the EuRoC dataset, and thanks to our ability to use mid-term data association. Our Atlas system also builds on DBoW2, but proposes a novel higher-recall place recognition technique, and performs more detailed and accurate map merging using local BA, increasing the advantage to 3.2 times better accuracy than VINS-Mono in multi-session operation on EuRoC.

III. SYSTEM OVERVIEW

ORB-SLAM3 is built on ORB-SLAM2 [3] and ORB-SLAM-VI [4]. It is a full multi-map and multi-session system able to work in pure visual or visual-inertial modes with monocular, stereo or RGB-D sensors, using pin-hole and fisheye camera models. Figure 1 shows the main system components, that are parallel to those of ORB-SLAM2 with some significant novelties, that are summarized next:

- **Atlas** is a multi-map representation composed of a set of disconnected maps. There is an active map where the tracking thread localizes the incoming frames, and is continuously optimized and grown with new keyframes by the local mapping thread. We refer to the other maps in the Atlas as the non-active maps. The system builds a unique DBoW2 database of keyframes that is used for relocalization, loop closing and map merging.

- **Tracking thread** processes sensor information and computes the pose of the current frame with respect to the active map in real-time, minimizing the reprojection error of the matched map features. It also decides whether the current frame becomes a keyframe. In visual-inertial mode, the body velocity and IMU biases are estimated by including the inertial residuals in the optimization. When tracking is lost, the tracking thread tries to relocate the current frame in all the Atlas’ maps. If relocated, tracking is resumed, switching the active map if needed. Otherwise, after a certain time, the active map is stored as non-active, and a new active map is initialized from scratch.

- **Local mapping thread** detects common regions between the active map and the whole Atlas at keyframe rate. Additionally, in the inertial case, the IMU parameters are initialized and refined by the mapping thread using our novel MAP-estimation technique.

- **Loop and map merging thread** detects common regions between the active map and the whole Atlas at keyframe rate. If the common area belongs to the active map, it
performs loop correction; if it belongs to a different map, both maps are seamlessly merged into a single one, that becomes the active map. After a loop correction, a full BA is launched in an independent thread to further refine the map without affecting real-time performance.

IV. Camera Model

ORB-SLAM assumed in all system components a pin-hole camera model. Our goal is to abstract the camera model from the whole SLAM pipeline by extracting all properties and functions related to the camera model (projection and unprojection functions, Jacobian, etc) to separate modules. This allows our system to use any camera model by providing the corresponding camera module. In ORB-SLAM3 library, apart from the pinhole model, we provide the Kannala-Brandt fisheye model. However, camera model abstraction raises some difficulties that need to be addressed, and are discussed next.

A. Relocalization

A robust SLAM system needs the capability of relocating the camera when tracking fails. ORB-SLAM solves the relocalization problem by setting a Perspective-n-Points solver based on the pNP algorithm, which assumes a calibrated pinhole camera along all its formulation. To follow up with our approach, we need a PnP algorithm that works independently of the camera model used. For that reason, we have adopted Maximum Likelihood Perspective-n-Point algorithm (MLPNP) that is completely decoupled from the camera model as it uses projective rays as input. The camera model just needs to provide an unprojection function passing from pixels to projection rays, to be able to use relocalization.

B. Non-rectified Stereo SLAM

Most stereo SLAM systems assume that stereo frames are rectified, i.e. both images are transformed to pin-hole projections using the same focal length, with image planes co-planar, and are aligned with horizontal epipolar lines, such that a feature in one image can be easily matched by looking at the same row in the other image. However the assumption of rectified stereo images is very restrictive and, in many applications, is not suitable nor feasible. For example, rectifying a divergent stereo pair, or a stereo fisheye camera would require severe image cropping, loosing the advantages of a large field of view: faster mapping of the environment and better robustness to occlusions.

For that reason, our system does not rely on image rectification, considering the stereo rig as two monocular cameras having:

1) a constant relative SE(3) transformation between them, and
2) optionally, a common image region that observes the same portion of the scene.

These constrains allow us to effectively estimate the scale of the map by introducing that information when triangulating new landmarks and in the Bundle Adjustment optimization. Following up with this idea, our SLAM pipeline estimates a 6 DoF rigid body pose, whose reference system can be located in one of the cameras or in the IMU sensor, and represent the cameras with respect to the rigid body pose.

If both cameras have an overlapping area in which we have stereo observations, we can triangulate true scale landmarks the first time they are seen. The rest of both images still has a lot of relevant information for the SLAM pipeline and it is used as monocular information. Features first seen in these areas are triangulated from multiple views, as in the monocular case.

V. Visual-Inertial SLAM

ORB-SLAM-VI was the first true visual-inertial SLAM system capable of map reusing. However, it was limited to pinhole monocular cameras, and its initialization was too slow, failing in some challenging scenarios. In this work, we build on ORB-SLAM-VI providing a fast an accurate IMU initialization technique, and an open-source SLAM library capable of monocular-inertial and stereo-inertial SLAM, with pinhole and fisheye cameras.

A. Fundamentals

While in pure visual SLAM, the estimated state only includes the current camera pose, in visual-inertial SLAM, additional variables need to be computed. These are the body pose \( T_i = [R_i, p_i] \in SE(3) \) and velocity \( v_i \) in the world frame, and the gyroscope and accelerometer biases, \( b^g_i \) and \( b^a_i \), which are assumed to evolve according to a Brownian motion. This leads to the state vector:

\[
S_i = \{ T_i, v_i, b^g_i, b^a_i \}
\]  

For visual-inertial SLAM, we preintegrate IMU measurements between consecutive visual frames, \( i \) and \( i+1 \), following the theory developed in [61], and formulated on manifolds in [62]. We obtain preintegrated rotation, velocity and position measurements, denoted as \( \Delta R_{i,i+1} \), \( \Delta v_{i,i+1} \), and \( \Delta p_{i,i+1} \), as well an information matrix \( \Sigma_{i,i+1} \) for the whole measurement vector. Given these preintegrated terms and states \( S_i \) and \( S_{i+1} \), we adopt the definition of inertial residual \( r_{i,i+1} \) from [62]:

\[
\begin{align*}
  r_{\Delta R_{i,i+1}} &= [r_{\Delta \Delta R_{i,i+1}}, r_{\Delta \Delta \Delta R_{i,i+1}}] \\
  r_{\Delta R_{i,i+1}} &= \log \left( R^T_{i,i+1} R_{i,i+1} \right) \\
  r_{\Delta R_{i,i+1}} &= R^T_{i,i+1} \left( v_{i+1} - v_i - g \Delta v_{i,i+1} \right) - \Delta v_{i,i+1} \\
  r_{\Delta p_{i,i+1}} &= R^T_{i,i+1} \left( p_{i+1} - p_i - \frac{1}{2} g \Delta t^2 \right) - \Delta p_{i,i+1}
\end{align*}
\]  

Together with inertial residuals, we also use reprojection errors \( r_{ij} \) between frame \( i \) and 3D point \( j \) at position \( x_j \):

\[
r_{ij} = x_j - \Pi \left( T_{CB} T_{i}^{-1} \odot x_j \right)
\]  

where \( \Pi : \mathbb{R}^3 \rightarrow \mathbb{R}^n \) is the projection function for the corresponding camera model, \( u_{ij} \) is the observation of point \( j \) at image \( i \), having an information matrix \( \Sigma_{ij} \). \( T_{CB} \in SE(3) \) stands for the rigid transformation from body-IMU to camera (left or right), known from calibration. \( \odot \) is the transformation operation of \( SE(3) \) group over \( \mathbb{R}^3 \) elements.
Combining inertial and visual residual terms, visual-inertial SLAM can be posed as a keyframe-based minimization problem \[ L_k = \sum_{i=1}^{k} \left( \sum_{j=0}^{l} \rho_{ij} \left( \| r_{ij} \|_{\Sigma_{ij}} \right) \right) \] where \( K^j \) is the set of keyframes observing 3D point \( j \). This optimization may be outlined as the factor-graph shown in figure 2a. Note that for reprojection error we use a robust Huber kernel \( \rho_{ij} \) to reduce the influence of spurious matchings, while for inertial residuals it is not needed, since miss-associations do not exist. This optimization needs to be adapted for efficiency during tracking and mapping, but more importantly, it requires good initial seeds to converge to accurate solutions.

**B. IMU Initialization**

The goal of this step is to obtain good initial values for the inertial variables: body velocities, gravity direction, and IMU biases. Some systems like VI-DSO \[46\] try to solve from scratch visual-inertial BA, sidestepping a specific initialization process, obtaining slow convergence for inertial parameters (up to 30 seconds).

In this work we propose a fast and accurate initialization method based on three key insights:

- Pure monocular SLAM can provide very accurate initial maps \[2\], whose main problem is that scale is unknown. Solving first the vision-only problem will enhance IMU initialization.
- As shown in \[77\], scale converges much faster when it is explicitly represented as an optimization variable, instead of using the implicit representation of BA.
- Ignoring sensor uncertainties during IMU initialization produces large unpredictable errors \[65\].

So, taking properly into account sensor uncertainties, we state the IMU initialization as a MAP estimation problem, split in three steps:

1) **Vision-only MAP Estimation**: We initialize pure monocular SLAM \[2\] and run it during 2 seconds, inserting keyframes at 4Hz. After this period, we have an up-to-scale map composed of \( k = 10 \) camera poses and hundreds of points, that is optimized using Visual-Only BA (figure 2b). These poses are transformed to body reference, obtaining the trajectory \( \bar{T}_{0:k} = [\bar{R}, \bar{p}]_{0:k} \) where the bar denotes up-to-scale variables.

2) **Inertial-only MAP Estimation**: In this step we aim to obtain the optimal estimation of the inertial variables, in the sense of MAP estimation, using only \( \bar{T}_{0:k} \) and inertial measurements between these keyframes. These inertial variables may be stacked in the inertial-only state vector:

\[
\mathcal{Y}_k = \{s, R_{wg}, b, \bar{v}_{0:k}\}
\]

where \( s \in \mathbb{R}^+ \) is the scale factor of the vision-only solution; \( R_{wg} \in SO(3) \) is the gravity direction, represented with two angles, such that gravity vector in world reference is \( g = R_{wg} g_b \), with \( g_b = (0,0,G)^T \) being \( G \) the magnitude of gravity; \( b = (b^a, b^g) \in \mathbb{R}^6 \) are the accelerometer and gyroscope biases assumed to be constant during initialization; and \( \bar{v}_{0:k} \in \mathbb{R}^3 \) is the up-to-scale body velocities from first to last keyframe, initially estimated from \( \bar{T}_{0:k} \). At this point, we are only considering the set of inertial measurements \( \mathcal{I}_{0:k} = \{I_0, \ldots, I_{k-1}\} \). Thus, we can state a MAP estimation problem, where the posterior distribution to be maximized is:

\[
p(\mathcal{Y}_k|\mathcal{I}_{0:k}) \propto p(\mathcal{I}_{0:k}|\mathcal{Y}_k)p(\mathcal{Y}_k)
\]

where \( p(\mathcal{I}_{0:k}|\mathcal{Y}_k) \) stands for likelihood and \( p(\mathcal{Y}_k) \) for prior. Considering independence of measurements, the inertial-only MAP estimation problem can be written as:

\[
\mathcal{Y}^*_k = \arg\max_{\mathcal{Y}_k} \left( p(\mathcal{Y}_k) \prod_{i=1}^{k} p(I_{i-1},|s, g_{dir}, b, \bar{v}_{i-1}, \bar{v}_i) \right)
\]

Taking negative logarithm and assuming Gaussian error for IMU preintegration and prior distribution, this finally results in the optimization problem:

\[
\mathcal{Y}^*_k = \arg\min_{\mathcal{Y}_k} \left( \| r_p \|_{\Sigma_p}^2 + \sum_{i=1}^{k} \| r_{X_{i-1},i} \|_{\Sigma_{X_{i-1},i}}^2 \right)
\]

This optimization, represented in figure 2c, differs from equation 4 in not including visual residuals, but a prior residual \( r_p \) that we use to impose that IMU biases should
be close to zero, with a covariance given by the IMU characteristics. As we are optimizing in a manifold we need to define a retraction \[ [62] \] to update the gravity direction estimation during the optimization:
\[
\mathbf{P}_{wg}^{\text{new}} = \mathbf{P}_{wg}^{\text{old}} \exp(\delta \alpha_{g}, \delta \beta_{g}, 0)
\]
being \( \exp(\cdot) \) the exponential map from \( \text{so}(3) \) to \( \text{SO}(3) \). To guarantee that scale factor remains positive during optimization we define its update as:
\[
s_{\text{new}} = s_{\text{old}} \exp(\delta s)
\]

Once the inertial-only optimization is finished, the frame poses and velocities and the 3D map points are scaled with the estimated scale and rotated to align the \( z \) axis with the estimated gravity direction. Biases are updated and IMU preintegration is repeated, aiming to reduce future linearization errors.

3) \textbf{Visual-Inertial MAP Estimation}: Once we have a good estimation for inertial and visual parameters, we can perform a joint visual-inertial optimization for further refining the solution. This optimization may be represented as figure \[ \text{2a} \] but having common biases for all keyframes and including the same prior information than in the inertial-only step.

Our exhaustive initialization experiments on the EuRoC dataset [6] show that this initialization is very efficient, achieving 5% scale error with 2 seconds trajectories. To improve the initial estimation, visual-inertial BA is performed 5 and 15 seconds after initialization, converging to 1% scale error as shown in section \[ \text{VII} \]. After these BAs, we say that the map is \textit{mature}, meaning that scale, IMU parameters and gravity directions are already accurately estimated.

Our initialization is much more accurate than joint initialization methods that solve a set of algebraic equations [63]–[65], and much faster than the initialization used in ORB-SLAM-VI [4] that needed 15 seconds to get the first scale estimation, or that used in VI-DSO [66], that starts with a huge scale error and requires 20-30 seconds to converge to 1% error.

We have easily extended our monocular-inertial initialization to stereo-inertial by fixing the scale factor to one and taking it out from the inertial-only optimization variables, enhancing its convergence.

\textbf{C. Tracking and Mapping}

For tracking and mapping we adopt the schemes proposed in [4]. Tracking solves a simplified visual-inertial optimization where only the states of the last two frames are optimized, while map points remain fixed.

For mapping, trying to solve the whole optimization from equation [4] would be intractable for large maps. We use as optimizable variables a sliding window of keyframes and their points, including covisible keyframes but keeping them fixed.

In some specific cases, when slow motion does not provide good observability of the inertial parameters, initialization may fail to converge to accurate solutions in just 15 seconds. To get robustness against this situation, we propose a novel scale refinement technique, based on a modified inertial-only optimization, where all inserted keyframes are included but scale and gravity direction are the only parameters to be estimated (figure 2d). Notice that in that case, the assumption of constant biases would not be correct. Instead, we use the values estimated for each frame, and we fix them. This optimization, which is very computationally efficient, is performed in the Local Mapping thread every ten seconds, until the map has more than 100 keyframes or more than 75 seconds have passed since initialization.

\textbf{D. Robustness to tracking loss}

In pure visual SLAM or VO systems, temporal camera occlusion and fast motions result in losing track of visual elements, getting the system lost. ORB-SLAM pioneered the use of fast relocation techniques based on bag-of-words place recognition, but they proved insufficient to solve difficult sequences in the EuRoC dataset [3]. Our visual-inertial system enters into \textit{visually lost} state when less than 15 point maps are tracked, and achieves robustness in two stages:

- \textit{Short-term lost}: the current body state is estimated from IMU readings, and map points are projected in the estimated camera pose and searched for matches within a large image window. The resulting matches are included in visual-inertial optimization. In most cases this allows to recover visual tracking. Otherwise, after 5 seconds, we pass to the next stage.
- \textit{Long-term lost}: A new visual-inertial map is initialized as explained above, and it becomes the active map.

\textbf{VI. Map Merging and Loop Closing}

Short-term and mid-term data-associations between a frame and the active map are routinely found by the tracking and mapping threads by projecting map points into the estimated camera pose and searching for matches in an image window of just a few pixels. To achieve long-term data association for relocation and loop detection, ORB-SLAM uses the DBoW2 bag-of-words place recognition system [9], [78]. This method has been also adopted by most recent VO and SLAM systems that implement loop closures (Table 1).

Unlike tracking, place recognition does not start from an initial guess for camera pose. Instead, DBoW2 builds a database of keyframes with their bag-of-words vectors, and given a query image is able to efficiently provide the most similar keyframes according to their bag-of-words. Using only the first candidate, the raw DBoW2 queries achieve precision and recall in the order of 50-80% [9]. To avoid false positives that would corrupt the map, DBoW2 implements temporal and geometric consistency checks moving the working point to 100% precision and 30-40% recall [9], [78]. Crucially, the temporal consistency check delays place recognition at least during 3 keyframes. When trying to use it in our Atlas system, we found that this delay and the low recall results too often in duplicated areas in the same or in different maps.

In this work we propose a new place recognition algorithm with improved recall for long-term and multi-map data association. Whenever the mapping thread creates a new keyframe,
place recognition is launched trying to detect matches with any of the keyframes already in the Atlas. If the matching keyframe found belongs to the active map, a loop closure is performed. Otherwise, it is a multi-map data association, then, the active and the matching maps are merged. As a second novelty in our approach, once the relative pose between the new keyframe and the matching map is estimated, we define a local window with the matching keyframe and its neighbours in the covisibility graph. In this window we intensively search for mid-term data associations, improving the accuracy of loop closing and map merging. These two novelities explain the better accuracy obtained by ORB-SLAM3 compared with ORB-SLAM2 in the EuRoC experiments. The details of the different operations are explained next.

A. Place Recognition

To achieve high recall, for every new active keyframe we query the DBoW2 database for several similar keyframes in the Atlas. To achieve 100% precision, each of these candidates goes through several steps of geometric verification. The elementary operation of all the geometrical verification steps consists in checking whether there is an ORB keypoint inside an image window whose descriptor matches the ORB descriptor of a map point, using a threshold for the Hamming distance ratio to the second-closest match [79]. The steps of our place recognition algorithm are:

1) **DBoW2 candidate keyframes.** We query the Atlas DBoW2 database with the active keyframe $K_a$ to retrieve the three most similar keyframes, excluding keyframes covisible with $K_a$. We refer to each matching candidate for place recognition as $K_m$.

2) **Local window.** Per each $K_m$ we define a local window that includes $K_m$, its best covisible keyframes, and the map points observed by all of them. The DBoW2 direct index provides a set of putative matches between keypoints in $K_a$ and in the local window keyframes. Per each of these 2D-2D match we have also available the 3D-3D match between their corresponding map points.

3) **3D aligning transformation.** We compute using RANSAC the transformation $T_{am}$ that better aligns the map points in $K_m$ local window with those of $K_a$. In pure monocular, or in monocular-inertial when the map is still not mature, we compute $T_{am} \in \text{Sim}(3)$, otherwise $T_{am} \in \text{SE}(3)$. In both cases we use Horn algorithm [80] using a minimal set of three 3D-3D matches to find each hypothesis for $T_{am}$. The putative matches that, after transforming the map point in $K_a$ by $T_{am}$, achieve a reprojection error in $K_a$ below a threshold, give a positive vote to the hypothesis. The hypothesis with more votes is selected, provided the number is over a threshold.

4) **Guided matching refinement.** All the map points in the local window are transformed with $T_{am}$ to find more matches with the keypoints in $K_a$. The search is also reversed, finding matches for $K_a$ map points in all the keyframes of the local window. Using all the matchings found, $T_{am}$ is refined by non-linear optimization, where the goal function is the bidirectional reprojection error, using Huber influence function to provide robustness to spurious matches. If the number of inliers after the optimization is over a threshold, a second iteration of guided matching and non-linear refinement is launched, using a smaller image search window.

5) **Verification in three covisible keyframes.** To avoid false positives, DBoW2 waited for place recognition to fire in three consecutive keyframes, delaying or missing place recognition. Our crucial insight is that, most of the time, the information required for verification is already in the map. To verify place recognition, we search in the active part of the map two keyframes covisible with $K_a$ where the number of matches with points in the local window is over a threshold. If they are not found, the validation is further tried with the new incoming keyframes, without requiring the bag-of-words to fire again. The validation continues until three keyframes verify $T_{am}$, or two consecutive new keyframes fail to verify it.

6) **VI Gravity direction verification.** In the visual-inertial case, if the active map is mature, we have estimated $T_{am} \in \text{SE}(3)$. We further check whether the pitch and roll angles are below a threshold to definitively accept the place recognition hypothesis.

B. Visual Map Merging

When a successful place recognition produces multi-map data association between keyframe $K_a$ in the active map $M_a$, and a matching keyframe $K_m$ from a different map stored in the Atlas $M_m$, with an aligning transformation $T_{am}$, we launch a map merging operation. In the process, special care must be taken to ensure that the information in $M_m$ can be promptly reused by the tracking thread to avoid map duplication. For this, we propose to bring the $M_a$ map into $M_m$ reference. As $M_a$ may contain many elements and merging them might take a long time, merging is split in two steps. First, the merge is performed in a welding window defined by the neighbours of $K_a$ and $K_m$, and in a second stage, the correction is propagated to the rest of the merged map by a pose-graph optimization. The detailed steps of the merging algorithm are:

1) **Welding window assembly.** The welding window includes $K_a$ and its covisible keyframes, $K_m$ and its covisible keyframes, and all the map point observed by them. Before their inclusion in the welding window, the keyframes and map points belonging to $M_a$ are transformed by $T_{ma}$ to align them with respect to $M_m$.

2) **Merging maps.** Maps $M_a$ and $M_m$ are fused together to become the new active map. To remove duplicated points, matches are actively searched for $M_a$ points in the $M_m$ keyframes. For each match, the point from $M_a$ is removed, and the point in $M_m$ is kept accumulating all the observations of the removed point. The covisibility and essential graphs [2] are updated by the addition of
edges connecting keyframes from $M_m$ and $M_a$ thanks to the new mid-term point associations found.

3) **Welding bundle adjustment.** A local BA is performed optimizing all the keyframes from $M_a$ and $M_m$ in the welding window (Fig. 3a). To fix gauge freedoms, the keyframes that are covisible with those in $M_m$ are kept fixed. Once the optimization finishes, all the keyframes included in the welding area can be used for camera tracking, achieving fast and accurate reuse of map $M_m$.

4) **Pose-graph optimization.** A pose-graph optimization is performed using the essential graph of the whole merged map, keeping fixed the keyframes in the welding area. This optimization propagates corrections from the welding window to the rest of the map.

C. **Visual-Inertial Map Merging**

The visual-inertial merging algorithm follows similar steps than the pure visual case. Steps 1) and 3) are modified to better exploit the inertial information:

1) **VI welding window assembly:** If the active map is mature, we apply the available $T_{ma} \in \text{SE}(3)$ to map $M_a$ before its inclusion in the welding window. If the active map is not mature, we align $M_a$ using the available $T_{ma} \in \text{Sim}(3)$.

2) **VI welding bundle adjustment:** Poses, velocities and biases of the active keyframe $K_a$ and its 5 last temporal keyframes are included as optimizable. These variables are related by IMU preintegration terms. For map $M_m$, we proceed similarly, including poses, velocities and biases of $K_m$ and its 5 temporal neighbours, as shown in Figure 3b. For $M_m$, the keyframe immediately before the local window is included but fixed, while for $M_a$ the similar keyframe is included but its pose remains optimizable. All points seen by all these keyframes, and keyframes poses observing these points are also optimized. All keyframes and points are related by means of reprojection error.

D. **Loop Closing**

Loop closing correction algorithm is analogous to map merging, but in a situation where both keyframes matched by place recognition belong to the active map. A welding window is assembled from the matched keyframes, and point duplicates are detected and fused creating new links in the covisibility and essential graphs. The next step is a pose-graph optimization to propagate the loop correction to the rest of the map. The final step is a global BA to find the MAP estimate after considering the loop closure mid-term and long-term matches. In the visual inertial case, the global BA is only performed if the number of keyframes is below a threshold to avoid a huge computational cost.

VII. **EXPERIMENTAL RESULTS**

The evaluation of the whole system is split in:

- Single session experiments in EuRoC [81]: each of the 11 sequences is processed to produce a map, with

the four sensor configurations: Monocular, Monocular-Inertial, Stereo and Stereo-Inertial.
- Performance of monocular and stereo visual-inertial SLAM with fisheye cameras, in the challenging TUM VI Benchmark [82].
- Multi-session experiments in both datasets.

As usual in the field, we measure accuracy with RMS ATE [83], aligning the estimated trajectory with ground truth using a $\text{Sim}(3)$ transformation in the pure monocular case, and a $\text{SE}(3)$ transformation in the rest of sensor configurations. All experiments have been run in an Intel Core i7-7700 CPU, at 3.6GHz, with 32 GB memory.

A. **Single-session SLAM on EuRoC**

Table II compares the performance of ORB-SLAM3 using its four sensor configurations with the most relevant systems in the state-of-the-art. Our reported values are the median after 10 executions. As shown in the table, ORB-SLAM3 achieves in all sensor configurations more accurate result than the best systems available in the literature, in most cases by a wide margin.
In monocular-inertial configuration, ORB-SLAM3 more than doubles the accuracy of VI-DSO and VINS-Mono, showing again the advantages of mid-term and long-term data association. Compared with ORB-SLAM VI, our novel fast IMU initialization allows ORB-SLAM3 to calibrate the inertial sensor in a few seconds and use it from the very beginning, being able to complete all EuRoC sequences, and obtaining better accuracy.

In stereo-inertial configuration, ORB-SLAM3 is vastly superior to OKVIS, VINS-Fusion and Kimera. It’s accuracy is only approached by the recent Basalt that, being a native stereo-inertial system, was not able to complete sequence V203, where some frames from one of the cameras are missing.

To summarize performance, we have presented the median of 10 executions for each sensor configuration. For a robust system, the median represents accurately the behavior of the system. But a non-robust system will show high variance in its results. This can be analyzed using figure 4 that shows with colors the error obtained in each of the 10 executions. Comparison with the figures for DSO, ROVIO and VI-DSO published in [66] confirms the superiority of our method.

In pure visual configurations, the multi-map system adds in pure visual configurations, the multi-map system adds in pure visual configurations, the multi-map system adds in pure visual configurations, the multi-map system adds in pure visual configurations, the multi-map system adds
that could not be solved by ORB-SLAM2 and are successfully solved by our system in most executions. As expected, stereo is more robust than monocular thanks to its faster feature initialization, with the additional advantage that the real scale is estimated.

However, the big leap in robustness is obtained by our novel visual-inertial SLAM system, both in monocular and stereo configurations. The stereo-inertial system comes with a very slight advantage, particularly in the most challenging V203 sequence.

We can conclude that inertial integration not only boosts accuracy, reducing the median ATE error compared to pure visual solutions, but it also endows the system with excellent robustness, having a much more stable performance.

B. Visual-Inertial SLAM on TUM-VI Benchmark

The TUM-VI dataset [82] consists of 28 sequences in 6 different environments, recorded using a hand-held fisheye stereo-inertial rig. Ground-truth for the trajectory is only available at the beginning and at the end of the sequences. Many sequences in the dataset do not contain loops. Even if the starting and ending point are in the same room, point of view directions are opposite and place recognition cannot detect any common region. The idea is to measure the accumulated drift along the whole trajectory. In fact, the RMS ATE error computed after GT alignment is about half the accumulated drift.

We extract 1500 ORB points per image in monocular-inertial setup, and 1000 points per image in stereo-inertial, after applying CLAHE equalization to address under and over exposure found in the dataset. For outdoor sequences, our system struggles with very far points coming from the cloudy sky, that is very visible in fisheye cameras. These points may have slow motion that can corrupt the system. For preventing this, we discard points further than 20 meters from the current camera pose, only for outdoors sequences. A more sophisticated solution would be to use an image segmentation algorithm to detect and discard the sky.

The results obtained are compared with the most relevant systems in the literature in table III that clearly shows the superiority of ORB-SLAM3 both in monocular-inertial and stereo-inertial. The closest systems are VINS-Mono and BASALT, that are essentially visual-inertial odometry systems with loop closures, and miss mid-term data associations.

Analyzing more in detail the performance of our system, it gets lowest error in small and medium indoor environments, room and corridor sequences, with errors below 10 cm for most of them. In these trajectories, the system is continuously revisiting and reusing previously mapped regions, which is one of the main strengths of ORB-SLAM3. Also, tracked points are typically closer than 5 m, which makes easier to estimate inertial parameters, preventing them from diverging.

In magistrale indoors sequences, that are up to 900 m long, most tracked points are relatively close, and ORB-SLAM3 obtains errors around 1 m except in one sequence that goes close to 5 m. In contrast, in some long outdoors sequences, the scarcity of close visual features may cause divergence of the inertial parameters, notably scale and accelerometer bias, which leads to errors in the order of 10 to 70 meters. Even though, ORB-SLAM3 is the best performing system in the outdoor sequences.

This dataset also contains three really challenging slides sequences, where the user descends though a dark tubular slide with almost total lack of visual features. In these situation, a pure visual system would be lost, but our visual-inertial system is able to process the whole sequence with competitive error, even if no loop-closures can be detected. Interestingly, VINS-Mono and BASALT, that track features using Lukas-Kanade, obtain in some of these sequences better accuracy than ORB-SLAM3, that matches ORB descriptors.

Finally, the room sequences can be representative of typical AR/VR applications, where the user moves with a hand-held or head-mounted device in a small environment. Table III shows that ORB-SLAM3 is significantly more accurate that competing approaches. The results obtained using our four sensor configurations are compared in table IV.
Table V: Multi-session RMS ATE (m) on the EuRoC dataset. Values of CCM-SLAM and VINS are copied verbatim from their papers. Our values are the median of 5 executions, aligning the trajectories with the processed GT.

| Room       | Machine Hall | Vicon 1 | Vicon 2 |
|------------|--------------|---------|---------|
| Sequences  | MH01-03      | MH01-05 | V101-103| V201-203|
| ORB-SLAM3 Mono ATE | 0.030 | 0.058 | 0.058 | 0.284 |
| CCM-SLAM Mono ATE | 0.077 | - | - | - |
| ORB-SLAM3 Stereo ATE | 0.028 | 0.040 | 0.027 | 0.163 |
| ORB-SLAM3 Mono-Inertial ATE | 0.037 | 0.065 | 0.040 | 0.048 |
| VINS Mono-Inertial Scale error | 0.4 | 0.3 | 1.4 | 0.9 |
| ORB-SLAM3 Stereo-Inertial ATE | 0.041 | 0.047 | 0.051 | 0.046 |
| Stereo-Inertial Scale error | 0.6 | 0.3 | 0.6 | 0.8 |

accuracy of pure monocular compared with stereo is only apparent: the monocular solution is up-to-scale and is aligned with ground-truth with 7DoF, while stereo provides the true scale, and is aligned with 6DoF. Using monocular-inertial, we further reduce the average RMSE ATE error below 2 cm, also obtaining the true scale. Finally, our stereo-inertial SLAM brings error below 1 cm, making it an excellent choice for AR/VR applications.

C. Multi-session SLAM

EuRoC dataset contains several sessions for each of its three environments: 5 in Machine Hall, 3 in Vicon1 and 3 in Vicon2. To test the multi-session performance of ORB-SLAM3, we process sequentially all the sessions corresponding to each environment. Each trajectory in the same environment has ground truth with the same world reference, which allows to perform a single global alignment to compute ATE.

The first sequence in each room provides an initial map. Processing the following sequences starts with the creation of a new active map, that is quickly merged with the map of the previous sessions, and from that point on, ORB-SLAM3 profits from reusing the previous map.

Table V reports the global multi-session RMS ATE for the four sensor configurations in the three rooms. Is is interesting to compare these multi-session performances with the single-session results reported in Table II. For example, in the pure monocular case, multi-session Vicon 1 achieves a global error that is smaller than the mean error of the single-session maps, and significantly smaller than the error of the single-session V103. Multi-session Vicon 2 can process V203 sequence that failed in single-session, thanks to the exploitation of the previous map. In the difficult Machine Hall sequences, MH04 and MH05, the multi-session error is smaller than single-session errors.

The table also compares with the two only published multi-session results in EuRoC dataset: CCM-SLAM [73] reporting pure monocular results in MH01-MH03, and VINS-Mono [7] in the five Machine Hall sequences, using monocular-inertial. In both cases ORB-SLAM3 more than doubles the accuracy of competing methods. In the case of VINS-Mono, ORB-SLAM3 obtains 2.6 better accuracy in single-session, and the advantage goes up 3.2 times in multi-session, showing the superiority of our map merging operations.

We have also performed some multi-session experiments on the TUM-VI dataset. Figure 5 shows the result after processing several sequences inside the TUM building in this case, the small room sequence provides loop closures that were missing in the longer sequences, bringing all errors to centimeter level. Although ground truth is not available outside the room, comparing the figure with the figures published in [84] clearly shows our point: our multi-session SLAM system obtains far better accuracy that existing visual-inertial odometry systems. This is further exemplified in Figure 6. Although ORB-SLAM3 ranks higher in stereo inertial single-

\[1\] Videos of this and other experiments can be found at https://www.youtube.com/channel/UCXVf-LXG6t95Z4tVaY1U8Q0Q
session processing of outdoors1, there is still a significant drift (≈ 60 m). In contrast, if outdoors1 is processed after magistrale2 in a multi-session manner, this drift is significantly reduced, and the final map is much more accurate.

VIII. CONCLUSIONS

Building on [2]–[4], we have presented ORB-SLAM3, the most complete open-source library for visual, visual-inertial and multi-session SLAM, with monocular, stereo, RGB-D, pin-hole and fisheye cameras. Our main contributions, apart from the integrated library itself, are the fast and accurate IMU initialization technique, and the multi-session map-merging functions, that rely on a new place recognition technique with improved recall, and make ORB-SLAM3 highly suitable for long-term and large-scale SLAM in real applications.

Our experimental results show that ORB-SLAM3 is the first visual and visual-inertial system capable of effectively exploiting short-term, mid-term, long-term and multi-map data associations, reaching an accuracy level that is beyond the reach of existing systems. Our results also suggest that, regarding accuracy, the capability of using all these types of data association overpowers other choices such as using direct methods instead of features, or performing keyframe marginalization for local BA, instead of assuming an outer set of static keyframes as we do.

Regarding robustness, direct methods can be more robust in low-texture environments, but are limited to short-term [27] and mid-term [31] data association. On the other hand, matching feature descriptors successfully solves long-term and multi-map data association, but seems to be less robust for tracking than Lucas-Kanade, that uses photometric information. An interesting line of research could be developing photometric techniques adequate for the four data association problems. We are currently exploring this idea for map building from endoscope images inside the human body.

About the four different sensor configurations, there is no question, stereo-inertial SLAM provides the most robust and accurate solution. Furthermore, the inertial sensor allows to estimate pose at IMU rate, which is orders of magnitude higher than frame rate, being a key feature for some use cases. For applications where a stereo camera is undesirable because of its higher bulk, cost, or processing requirements, you can use monocular-inertial without missing much in terms of robustness and accuracy. Only keep in mind that pure rotations during exploration would not allow to estimate depth.

In applications with slow motions, or without roll and pitch rotations, such as a car in a flat area, IMU sensors can be difficult to initialize. In those cases, if possible, use stereo SLAM. Otherwise, recent advances on depth estimation from a single image with CNNs offer good promise for reliable and true-scale monocular SLAM [85], at least in the same type of environments where the CNN has been trained.

REFERENCES

[1] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, “Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age,” IEEE Transactions on Robotics, vol. 32, no. 6, pp. 1309–1332, 2016.
[2] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, “ORB-SLAM: a versatile and accurate monocular SLAM system,” IEEE Transactions on Robotics, vol. 31, no. 5, pp. 1147–1163, 2015.
[3] R. Mur-Artal and J. D. Tardós, “ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras,” IEEE Transactions on Robotics, vol. 33, no. 5, pp. 1255–1262, 2017.
[4] ——, “Visual-inertial monocular SLAM with map reuse,” IEEE Robotics and Automation Letters, vol. 2, no. 2, pp. 796–803, 2017.
[5] C. Campos, R. Elvira, J. J. Gómez-Rodríguez, J. M. M. Montiel, and J. D. Tardós, “ORB-SLAM3: An accurate open-source library for visual, visual-inertial and multi-map SLAM,” https://github.com/ uzslamlab/ORB_SLAM3 2020.
[6] C. Campos, J. M. M. Montiel, and J. D. Tardós, “Inertial-only optimization for visual-inertial initialization,” in IEEE International Conference on Robotics and Automation (ICRA), 2020.
[7] T. Qin, P. Li, and S. Shen, “VINS-Mono: A robust and versatile monocular visual-inertial state estimator,” IEEE Transactions on Robotics, vol. 34, no. 4, pp. 1004–1020, 2018.
[8] A. Rosinol, M. Abate, Y. Chang, and L. Carlone, “Kimera: an open-source library for real-time metric-semantic localization and mapping,” in IEEE Intl. Conf. on Robotics and Automation (ICRA), 2020.
[9] G. Gálvez-López and J. D. Tardós, “Bags of binary words for fast place recognition in image sequences,” IEEE Transactions on Robotics, vol. 28, no. 5, pp. 1188–1197, October 2012.
[10] R. Elvira, J. D. Tardós, and J. M. M. Montiel, “ORB-SLAM-atlas: a robust and accurate multi-map system,” in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2019.
[11] R. Tsai, “A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf TV cameras and lenses,” IEEE Journal on Robotics and Automation, vol. 3, no. 4, pp. 323–344, 1987.
[12] J. Kannala and S. S. Brandt, “A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, no. 8, pp. 1335–1340, 2006.
[13] A. J. Davison, “Real-time simultaneous localisation and mapping with a single camera,” in Proc. IEEE Int. Conf. Computer Vision (ICCV), Oct 2003, pp. 1403–1410, vol. 2.
A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, “MonoSLAM: Real-time single camera SLAM,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 29, no. 6, pp. 1052–1067, 2007.

H. Kim, “SceneLib - MonoSLAM open-source library,” https://github.com/hannemann/SceneLib

G. Klein and D. Murray, “Parallel tracking and mapping for small AR workspaces,” in IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR), Nara, Japan, 2007, pp. 225–234.

—, “Improving the agility of keyframe-based SLAM,” in European Conference on Computer Vision (ECCV), 2008, pp. 802–815.

—, “Parallel tracking and mapping on a camera phone,” in 2009 8th IEEE International Symposium on Mixed and Augmented Reality, Oct 2009, pp. 83–88.

“PTAM-GPL,” https://github.com/Oxford-PTAM/PTAM-GPL, 2013.

J. Engel, T. Schöps, and D. Cremers, “LSD-SLAM: Large-scale direct monocular SLAM,” in European Conference on Computer Vision (ECCV), 2014, pp. 834–849.

M. Li and A. I. Mourikis, “High-precision, consistent ekf-based visual-inertial odometry using dynamic marginalization,” in IEEE Robotics and Automation Letters, vol. 5, no. 2, April 2020.

V. Usenko and N. Demmel, “BASALT,” https://github.com/VladyslavUsenko/basalt, 2019.

X. Gao, R. Wang, N. Demmel, and D. Cremers, “Direct sparse visual-inertial odometry using dynamic marginalization,” in IEEE International Conference on Robotics and Automation (ICRA), 2018, pp. 2510–2517.

J. Civera, A. J. Davison, and J. M. M. Montiel, “Inverse depth parametrization for monocular SLAM,” IEEE Transactions on Robotics, vol. 24, no. 5, pp. 932–945, 2008.

L. Clemente, A. J. Davison, I. D. Reid, J. Neira, and J. D. Tardós, “Mapping large loops with a single hand-held camera,” in Proc. Robotics: Science and Systems, Atlanta, GA, USA, June 2007.

J. Civera, O. G. Grasa, A. J. Davison, and J. M. M. Montiel, “1-point RANSAC for extended Kalman filtering: Application to real-time structure from motion and visual odometry,” Journal of field robotics, vol. 27, no. 5, pp. 609–631, 2010.

H. Strasdat, J. M. M. Montiel, and A. J. Davison, “Visual SLAM: Why filter?” Image and Vision Computing, vol. 30, no. 2, pp. 65–77, 2012.

J. Civera, “Scale drift-aware large scale monocular SLAM,” in Robotics: Science and Systems (RSS), 2010.

H. Strasdat, A. J. Davison, J. M. M. Montiel, and K. Konolige, “Double window optimisation for constant time visual SLAM,” in IEEE International Conference on Computer Vision (ICCV), 2011, pp. 2352–2359.

X. Gao, R. Wang, N. Demmel, and D. Cremers, “LDSO: Direct sparse odometry with loop closure,” in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2018, pp. 2198–2204.

S. H. Lee and J. Civera, “Loosely-coupled semi-direct monocular SLAM,” IEEE Robotics and Automation Letters, vol. 4, no. 2, pp. 399–406, 2018.

T. Lupton and S. Sukkarieh, “Visual-inertial-aided navigation for high-dynamic motion in built environments without initial conditions,” IEEE Transactions on Robotics, vol. 2, no. 1, pp. 61–76, 2012.

S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, “Keyframe-based visual–inertial odometry using nonlinear optimization,” Proceedings of Robotics Science and Systems (RSS), 2013.

S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, “Keyframe-based visual–inertial odometry using nonlinear optimization,” The International Journal of Robotics Research, vol. 34, no. 3, pp. 314–325, 2015.

S. Leutenegger, A. Forster, P. Furgale, P. Gohl, and S. Lynen, “OKVIS: Open keyframe-based visual-inertial SLAM (ROS version),” https://github.com/ethz-asl/okvis_ros, 2016.
[67] E. Eade and T. Drummond, “Unified loop closing and recovery for real time monocular SLAM,” in Proc. 19th British Machine Vision Conference (BMVC), Leeds, UK, September 2008.

[68] R. Castle, G. Klein, and D. W. Murray, “Video-rate localization in multiple maps for wearable augmented reality,” in 12th IEEE International Symposium on Wearable Computers, Sept 2008, pp. 15–22.

[69] C. Forster, S. Lynen, L. Kneip, and D. Scaramuzza, “Collaborative monocular SLAM with multiple micro aerial vehicles,” in IEEE/RSJ International Conference on Intelligent Robots and Systems, 2013, pp. 3962–3970.

[70] L. Riazuelo, J. Civera, and J. M. M. Montiel, “C2TAM: A cloud framework for cooperative tracking and mapping,” Robotics and Autonomous Systems, vol. 62, no. 4, pp. 401–413, 2014.

[71] J. G. Morrison, D. Gálvez-López, and G. Sibley, “MOARSLAM: Multiple operator augmented RSLAM,” in Distributed autonomous robotic systems. Springer, 2016, pp. 119–132.

[72] P. Schmuck and M. Chili, “Multi-UAV collaborative monocular SLAM,” in IEEE International Conference on Robotics and Automation (ICRA), 2017, pp. 3863–3870.

[73] ——, “CCM-SLAM: Robust and efficient centralized collaborative monocular simultaneous localization and mapping for robotic teams,” Journal of Field Robotics, vol. 36, no. 4, pp. 763–781, 2019.

[74] H. A. Daoud, A. Q. M. Sabri, C. K. Luo, and A. M. Mansoor, “SLAMM: Visual monocular SLAM with continuous mapping using multiple maps,” PloS one, vol. 13, no. 4, 2018.

[75] V. Lepetit, F. Moreno-Noguer, and P. Fua, “EPnP: An accurate O(n) solution to the PnP problem,” International journal of computer vision, vol. 81, no. 2, p. 155, 2009.

[76] S. Urban, J. Leitloff, and S. Hinz, “MLPhP - A Real-Time Maximum Likelihood Solution to the Perspective-n-Point Problem,” arXiv preprint arXiv:1607.08312, 2016.

[77] H. Strasdat, J. M. M. Montiel, and A. J. Davison, “Scale drift-aware large scale monocular SLAM,” Robotics: Science and Systems VI, vol. 2, 2010.

[78] R. Mur-Artal and J. D. Tardós, “Fast relocalisation and loop closing in keyframe-based SLAM,” in Proc. IEEE Int. Conf. Robotics and Automation (ICRA). IEEE, 2014, pp. 846–853.

[79] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” International journal of computer vision, vol. 60, no. 2, pp. 91–110, 2004.

[80] B. K. Horn, “Closed-form solution of absolute orientation using unit quaternions,” JOSA A, vol. 4, no. 4, pp. 629–642, 1987.

[81] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart, “The EuRoC micro aerial vehicle datasets,” The International Journal of Robotics Research, vol. 35, no. 10, pp. 1157–1163, 2016.

[82] D. Schubert, T. Goll, N. Demmel, V. Usenko, J. Stückler, and D. Cremers, “The TUM VI benchmark for evaluating visual-inertial odometry,” in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2018, pp. 1680–1687.

[83] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, “A benchmark for the evaluation of RGB-D SLAM systems,” in Proc. of the International Conference on Intelligent Robot Systems (IROS), Oct. 2012.

[84] D. Schubert, T. Goll, N. Demmel, V. Usenko, J. Stückler, and D. Cremers, “The TUM VI benchmark for evaluating visual-inertial odometry,” arXiv preprint arXiv:1804.06120v3, March 2020.

[85] N. Yang, L. v. Stumberg, R. Wang, and D. Cremers, “D3VO: Deep depth, deep pose and deep uncertainty for monocular visual odometry,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 1281–1292.