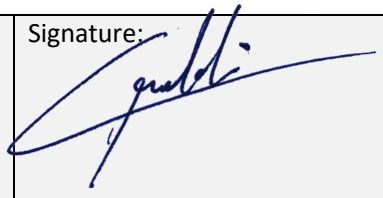
Autonomous Robotic Inspection and Maintenance on Ship Hulls and Storage Tanks

Deliverable report – D4.1 Localisation

Context		
Deliverable title	Localisation	
Lead beneficiary	UNI-KLU	
Author(s)	Stephan WEISS (UNI-KLU), Alberto ORTIZ (UIB), Cédric PRADALIER (CNRS)	
Work Package	WP04	
Deliverable due date	31st March 2022 (M27)	
Document status		
Version No.	1	
Type	REPORT	
Dissemination level	**Public**	
Last modified	04 April 2022	
Status	RELEASED	
Date approved	04 April 2022	
Approved by Coordinator	Prof. Cédric Pradalier (CNRS)	Signature: 
Declaration	Any work or result described therein is genuinely a result of the BugWright2 project. Any other source will be properly referenced where and when relevant.	

This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No. 871260

TABLE OF CONTENTS

LIST OF FIGURES

HISTORY OF CHANGES

Date	Written by	Description of change	Approver	Version No.
23.09.2021	UIB	Starting document	/	v0.1
20.01.2022	UNI-KLU	Pre-filling document	/	v0.2
21.03.2022	UNI-KLU	Feature complete draft	/	v.03
23.03.2022	UNI-KLU	Extending experiments sections	/	v.04
25.03.2022	CNRS	Proofreading	CNRS	V1.0

ABBREVIATIONS

UAV	Unmanned Aerial Vehicle
AUV	Autonomous Underwater Vehicle
MAV	Micro Aerial Vehicle
GNSS	Global Navigation Satellite Systems
UWB	Ultra-Wide Band
USBL	Ultra-Short Baseline acoustic positioning system
DVL	Doppler velocity sensor
VINS	Visual-Inertial Navigation System
LiDAR	Light Detection And Ranging

I. Introduction

This document aims at describing the localisation approaches developed for the different robotic platforms involved in the BUGWRIGHT2 framework: MAVs, AUVs and above- and underwater crawlers. Due to the particularities of both the platforms and the operating scenarios, this report is organised with one section per platform and operating area, where the localisation method is described followed by the respective performance assessments.

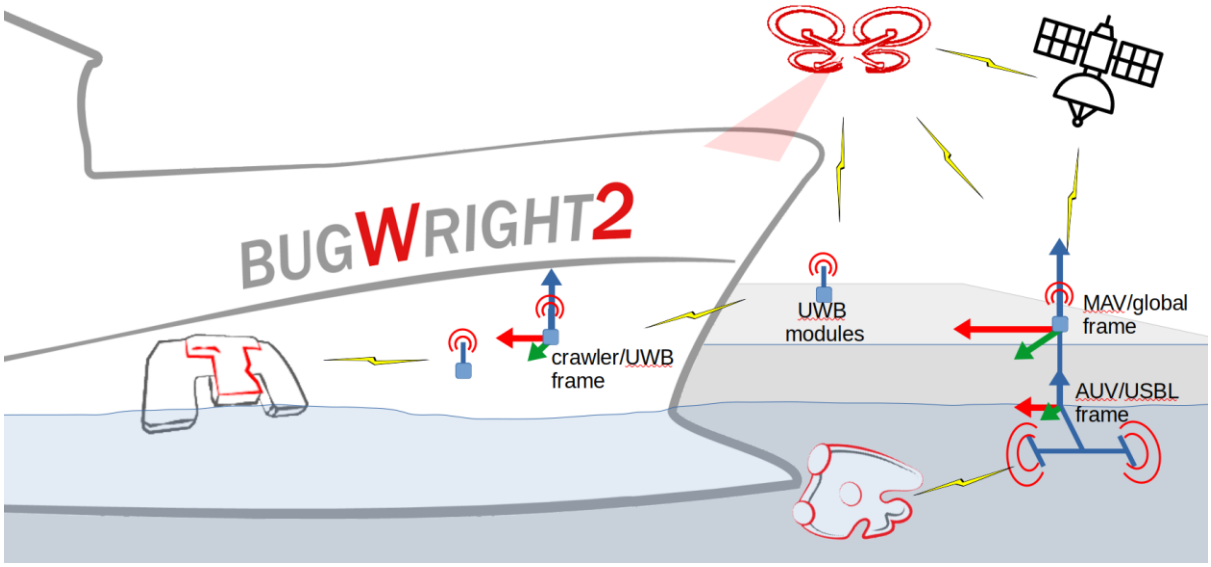


Figure 1: Schematic depiction of the different robot platforms, their operating scenarios and their reference frames.

The overall goal of the robot localisation in this project is to have all mobile systems working in a common reference frame in which, at a later stage, swarm algorithms, data acquisition, data visualisation, and user interaction can be carried out. A natural choice of the global reference system is the one of the global navigation satellite systems (GNSS). That being said, several operating scenarios in BUGWRIGHT2 impede the use of direct GNSS signals. Thus, the reference frame alignment across different domains and platforms requires special attention. A network of (meshed) distance measuring ultra-wide band (UWB) modules is considered suitable sensors to span a GNSS independent reference frame across the different aerial platforms and operation domains. Thus, research and development efforts were specifically focusing on the use, initialisation, characterisation, and alignment of such sensors. Figure 1 depicts schematically the relevant robotic systems, sensors, and reference frames which are used to align the navigation of all platforms in the GNSS (or UWB) reference frame. The interplay between those elements can be summarized as follows.

For the MAVs, their (sporadic) GNSS reception is used to align their inertial aided vision, depth, and UWB based navigation systems with the global frame. Their localisation result is further used to globally reference UWB anchor-positions on the pier and on the ship during the operation process initialisation phase in an autonomous fashion (see section II.1.ii.d).This removes the requirement of the end-users to manually measure the placed UWB anchors in the environment and on the ship hull. Once the UWB anchors are initialised their readings add to the localisation of the MAV. If no GNSS signal is available, the MAVs initialise the UWB modules to a common origin against which the remaining sensors and frames are aligned following the above described procedure.

The (potentially) globally referenced UWB anchors are further used as reference for the magnetic crawlers. Inherently this leads to a crawler navigation in the GNSS frame. Wheel odometry, inclinometer, plate based information, and UWB readings are then fused to obtain localisation information (see Section I.IV.2). For the underwater crawler, UWB measurements will not be available. Instead USBL measurements are incorporated.

For the AUVs, their reference frame is anchored in the USBL. In turn, the USBL has either GNSS reception or/and is incorporated in the overall UWB reference frame and has a defined heading such that this offset can be propagated to the AUVs, rendering their navigation aligned with the overall GNSS/UWB reference frame. Since the USBL signal is often interrupted and inaccurate (similar to the GNSS reception issues on the MAVs) the AUVs feature additional sensors for localisation encompassing a Doppler velocity sensor (DVL), depth sensor, and IMU (see Section III).

Multi-robot interaction and information will be included in the future in WP6 (mainly Task 6.3) to further improve the localisation and task execution performance through collaborative structures. Merging the localisation of the single robots with multi-robot tasks and missions will be further elaborated in WP6.

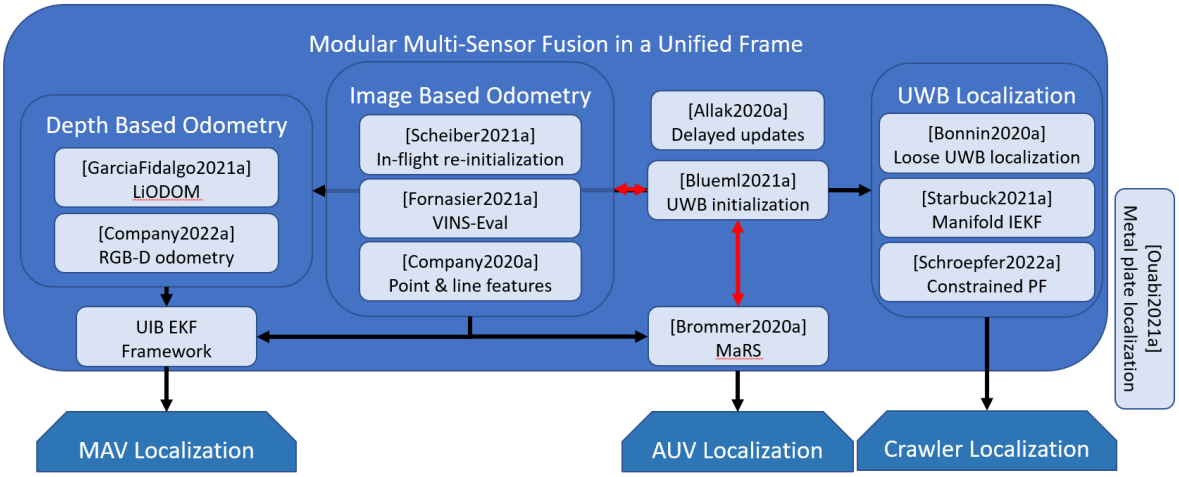


Figure 2: Overview of the contributions to Task 4.1. Localisation elements connecting scientific papers and challenges in the project to be addressed for localisation of the MAV, AUV, and crawler robots.

The most complex localisation framework is required for the MAV to ensure uninterrupted and precise information for high-rate control. Therefore, three main types of environmental information is fused: geometric structure (depth based), visual information (image based), and data from additional infrastructure (GNSS/UWB).

A framework (VINS-Eval) ensures rigorous statistical testing of the otherwise complex to assess image-based methods. Also, specific to the image-based methods, since metric information (depth) is not directly available, is an in-flight re-initialisation including other sensing modalities. The inclusion of salient information (i.e., lines) usually dominant in man environments further adds robustness to the image-based methods. The image, depth, and UWB/GNSS information is fused in the overall UIB cascaded EKF framework for MAV control. Different depth-based methods (RGB-D and Lidar) provide additional redundancy. A subset (MaRS) of the MAV localisation cascade is re-used with different platform specific sensors for the AUV localisation.

For the crawlers, the infrastructure-based information (UWB) is first required to be aligned with the other vehicle's reference frames (frame alignment and initialisation marked with red arrows). Then, UWB readings and wheel odometry are constrained to the crawler motion on the hull, seen as a 2D manifold embedded in 3D. Different manifold-constrained methods were tested to assess estimator complexity, versatility, consistency, and robustness. Acoustic guided waves-based localisation is included in this deliverable for completeness; however, further details are presented in D3.1.

II. Localisation approach for the aerial platforms

In the following, we describe and evaluate the localisation approaches on motion estimation and localisation developed and integrated into the two BUGWRIGHT2 MAVs. Due to the unavailability of the final MAV platform (DJI M100) as well as due to Covid and the resulting limitations in travel, synchronisation, and integration possibilities, the partners working on the approaches for the MAV localisation split the module testing elements to two different platforms. At UIB the Matrice 100 was used for UWB- and depth-based localisation techniques while UNI-KLU developed and tested the platform-agnostic UWB and image-based approaches TWINS Science UAV. Initial sensors and platform specifications are listed in Deliverable 1.2 "Software and Hardware Modification Specification" Section 3.

Providing uninterrupted localisation information to the aerial platforms is arguably the most challenging part of the robot localisation tasks in the project. Since the MAVs are inherently unstable platform, a high-precision and continuous provision of the localisation is crucial to maintain the platform airborne. In contrast, crawlers and even the AUVs can stop their actuators for algorithm and sensor re-initialisation without endangering the mission, platform, or their surroundings.

1. Image-based motion estimation and localisation

i. Sensor Modalities

To ensure uninterrupted localisation data to the MAVs, a multi-sensor approach was chosen. UNI-KLU focused on using image, UWB range sensing, magnetometer, and GNSS data for the localisation estimation with the former two sensors being the main information providers. Magnetometer and GNSS, if available in sufficient quality, are used to align the vision and UWB frames to a global coordinate system. The platform including the sensors used is depicted in Figure 3. Additional sensors like barometer or laser rangefinder are included for safety purposes in a resilient control scheme (see deliverable 5.1).

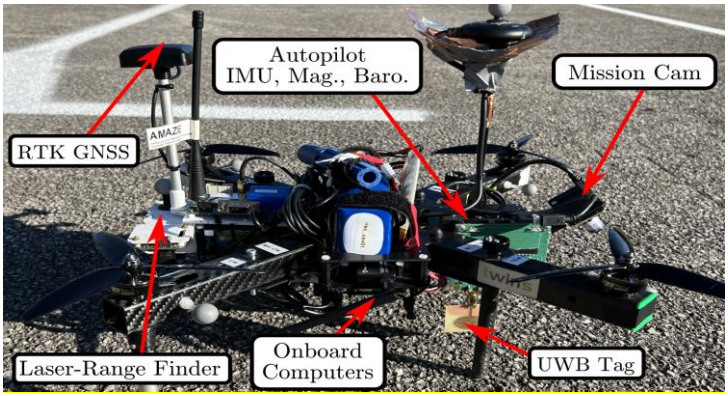


Figure 3: TWINS Science MAV platform used by UNI-KLU for image based multi-sensor localisation

Sensor	Specifications	Rate	Usage
PX4 10DoF IMU	3DoF gyro, accelerometer, and magnetometer, plus barometer	200Hz	Main modelling (propagation) sensor
Matrixvision MLC-200wG	WVGA monochrome	20Hz	Main navigation sensor
Lidar-lite point laser	10m range	10Hz	For contingency measures
RTK GNSS receiver	Up to cm-precision, 16cm and worse with bad/no RTK fix	7Hz	For global navigation frame alignment
Decawave UWB	50m range, decimetre precision	10Hz (varying with number of anchors)	For navigation frame alignment across robots
OdroidXU4 / PI4 compute board	ARM CPU (Exynos5422 / Broadcom BCM2711)		On-board computing

While initial intrinsic and extrinsic calibration is performed, all these states (including camera intrinsic and time delay) are estimated online in the estimator framework.

ii. Localisation Method Description

In connection with the surrogate platform TWINS Science, a series of fundamental platform-independent localisation elements were developed and tested. In a subsequent step, these elements were integrated in the DJI M100 platform which is targeted to be the overall demonstration platform for MAV localisation and also on the AUV platform for improve state estimation (see Section III). The localisation elements are:

1. Improved place recognition and loop closure for vision-based approaches using points and lines in man-made structure. [Company2020a]
2. Simulation framework to test, validate, and compare different state estimation approaches (with a focus on image-based methods) with statistical relevance [Fornasier2021a]
3. Mid-air image-based initialisation to mitigate vision based failures in-flight while bridging such sensor failures with redundant sensor information and control strategies (cf. deliverable 5.1) [Scheiber2021a]

4. UWB frame initialisation for local frame alignment across different robot platform types and for simpler alignment with GNSS if available. [Bluemel2021a]

5. Modular multi-modal sensor fusion framework with fast and scalable sensor handling despite signal delay and rate differences. [Brommer2020a, Allak2022a]

Improved place recognition and loop closure (adapted from [Company2020a]):

Visual SLAM approaches typically depend on loop closure detection to correct the in consistencies that may arise during the map and camera trajectory calculations, typically making use of point features for detecting and closing the existing loops. However, in low-textured scenarios as e.g., on the ship hull and on the deck along containers, it is difficult to find enough point features and, hence, the performance of these solutions drops drastically. An alternative for human-made scenarios, due to their structural regularity, is the use of geometrical cues such as straight segments, frequently present within these environments.

Under this context, UIB introduced with [Company2020a] an appearance-based loop closure detection method that integrates lines and points (see Figure 12). Adopting the idea of incremental Bag-of-Binary-Words (BoW) schemes, separate BoW models for each feature are built and used to retrieve previously seen images using a late fusion strategy. Additionally, a simple but effective mechanism, based on the concept of island, groups similar images close in time to reduce the image candidate search effort. A final step validates geometrically the loop candidates by incorporating the detected lines by means of a process comprising a line feature matching stage, followed by a robust spatial verification stage, now combining both lines and points. The approach compares well with several state-of-the-art solutions for a number of datasets involving different environmental conditions.

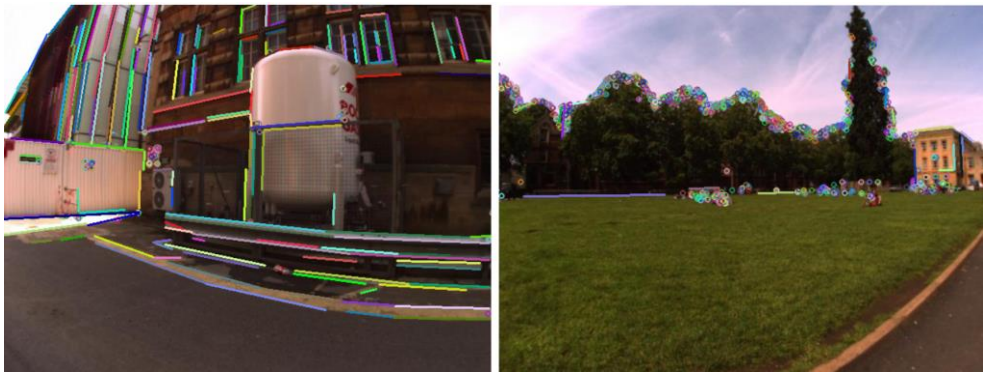


Figure 4: (left) a human-made environment including a high number of lines and a low number of points (right). An outdoor environment presenting the opposite situation.

Simulation framework (adapted from [Fornasier2021a]):

In the research community, there exist several different Visual-Inertial Navigation System (VINS) algorithms to localise mobile robots in a 3D environment. Comparison methods, rigor, depth, and repeatability of comparisons have a large spread and an unbiased evaluation framework to evaluate the best methods for the project did not exist. Further, with existing simulators and photo-realistic frameworks that could be extended for image-based performance analysis, the user is not able to easily test the sensitivity of the algorithms under examination with respect to specific environmental conditions and sensor specifications.

Due to the high complexity of image-based localisation methods, tests often include unwillingly many polluting effects falsifying the analysis and interpretations.

In addition, edge cases and corresponding failure modes often remain undiscovered due to the limited breadth of the test sequences. In this project, such edge cases are, however, of utmost importance to detect, in order to enable a later transition to industry and their applications. The unified evaluation framework developed in [Fornasier2021a] allows, in a fully automated fashion, a reproducible analysis of different VINS methods with respect to specific environmental and sensor parameters. The analyses per parameter are done over a multitude of test sets to obtain both statistically valid results and an average over other, potentially polluting effects with respect to the one parameter under test to mitigate biased interpretations. The automated performance results per method over all tested parameters are then summarized in unified radar charts (see Figure 21 as an example) for a fair comparison across authors and institutions.

Considering the output of this analysis and the required compute power, OpenVINS was chosen as base method for the image-based approach in this project. The open-sourced VINSEval framework is made available via https://github.com/aau-cns/vins_eval. A demonstration video of VINSEval is made available on https://youtu.be/KuA3nibxWok.

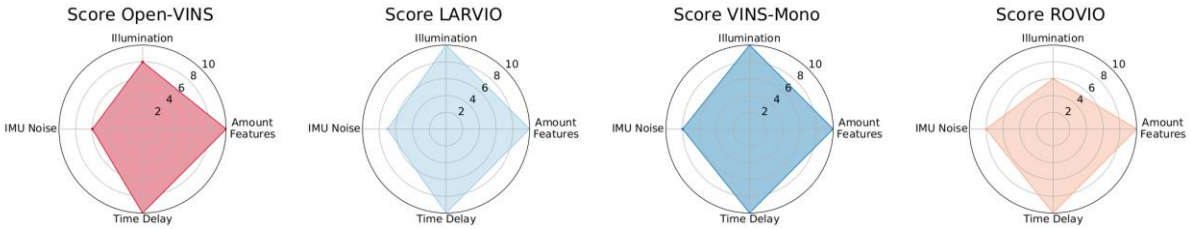


Figure 13: Robustness overall score and Breaking Point (BP) of the VINS algorithms.

Robustness overall score and Breaking Point (BP) of the VINS algorithms under examination with increasing difficulty levels for each of the considered environmental and/or sensor parameters. The BP per parameter is visually defined as the level next to the corner of the polygon.

Mid-air image-based initialisation (adapted from [Scheiber2021a]):

Even though the selected OpenVINS approach and several other state-of-the-art VINS show a remarkable robustness also in unprepared environments, the approaches fail at a rate that is still intolerable for industrial applications. A failure can have dramatic consequences. To prevent this, VINS must be able to re-initialise in mid-air, either during a free fall or on a constant velocity trajectory after attitude control has been re-established.

However, for both of these trajectory behaviours typically occurring after VINS failure, the visual scale cannot be observed because of the absence of acceleration change. [Scheiber2021a] proposes to use a small and lightweight laser-range finder (LRF) and a scene facet model to initialise vision-based navigation at the right scale under any motion condition and over any scene structure. This new range constraint is integrated into a visual-inertial bundle-adjustment initialiser. The approach is evaluated in simulation, including robustness to various parameters, and we demonstrated on real data how this approach can address mid-air state estimation failure in real-time. Sophisticated failure detection and mitigation strategies for the control behaviour are detailed in deliverable 5.1.

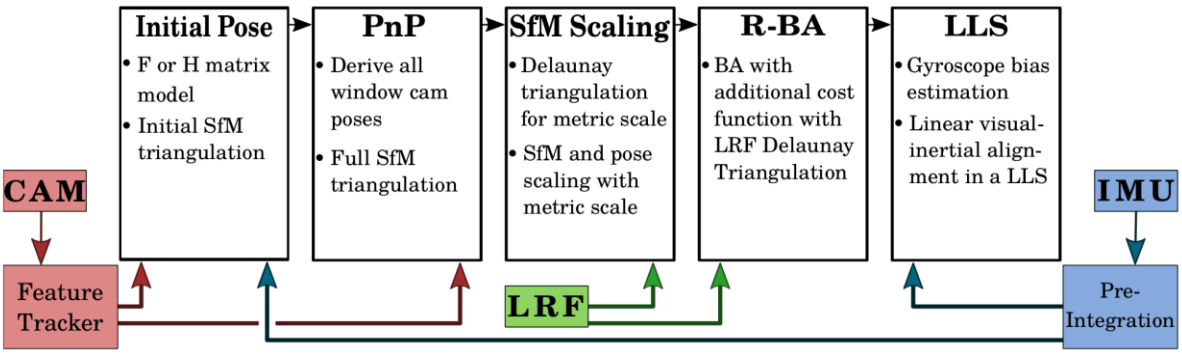


Figure 22: Proposed mid-air initialisation concept for ad-hoc VINS initialisation

Figure 30 shows the proposed ad-hoc initialisation concept: camera images are used to derive the initial camera poses using the Fundamental or Homography matrix method. Then, the scene structure from motion (SfM) is derived using a perspective-n-point (PnP) approach. After, this structure is scaled metrically with the range measurements received by the LRF. To reduce the impact of measurement noise a range-visual bundle-adjustment (R-BA) is performed. Finally, the range-visual poses are aligned with the pre-integrated IMU measurements, to derive the globally aligned states.

With the above modules, the core frame of image-based navigation and (re-)initialisation is available. The following modules extend this core capability towards using additional sensor modalities for increased resilience and towards unifying all robots in a common (local) UWB and, if available, GNSS frame.

UWB frame initialisation (adapted from [Blueml2021a]):

For UWB based navigation an accurate initialisation of the anchors in a reference coordinate system is crucial for precise subsequent UWB-inertial based or multi-sensor pose estimation. In [Blueml2021a] a strategy is developed based on information theory to initialise such UWB anchors using raw distance measurements from tag to anchor(s) and aerial vehicle poses. The vehicle poses can originate either from GNSS signals or image-based navigation. In the former case, the UWB mesh and all robots navigating in it are globally aligned. In the latter case, the mesh is referenced against an arbitrarily chosen origin and heading (gravity aligned). This still allows to coordinate all robots in a unified frame through UWB mesh coordinates. As soon as GNSS information is injected by one of the robots, the entire mesh can be, if necessary, aligned globally. The initialisation process includes a linear distance-dependent bias term and an offset in order to achieve unprecedented accuracy in the 3D position estimates of the anchors (error reduction by a factor of about 3.5 compared to state-of-the-art approaches) without the need of prior

knowledge. After an initial coarse position triangulation of the anchors using random vehicle positions, a bounding volume is created in the vicinity of the roughly estimated anchor position. In this volume, points are calculated which provide the maximal triangulation related information based on the Fisher Information Matrix (FIM). Using these information theoretic optimal points, a fine triangulation is done including bias term estimation. The approach is evaluated in simulations with realistic sensor noise as well as with real world experiments. A closed loop controlled flight is also performed using the UWB anchor positions based on this initialisation strategy (see Figure 42).

This initialisation approach is integrated as an extended version of the OpenVINS method described and evaluated above, such that the MAV can seamlessly initialise and use UWB and image data for resilient navigation and control (see deliverable 5.1). The estimated overall pose including uncertainty can further be forwarded to an overarching modular estimation module (e.g. MaRS described below or to UIB as described in Section II.3). Figure 38 depicts the schematic connections of the used sensors with the UWB initialisation module and our extended version of the VIO framework OpenVINS.

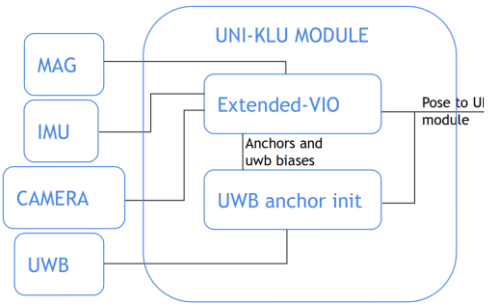


Figure 31: Schematic presentation of the extended OpenVINS image based localisation method merged with the UWB based navigation and initialisation.

Figure 52 shows a closed loop initialisation process where the MAV initially only navigates using camera images (VIO) and automatically initialises the UWB anchors in the same navigation frame. Then, once the UWB initial position is deemed to be sufficiently accurate based on statistical verification, their information is included in the multi-modal image-UWB-inertial based localisation.

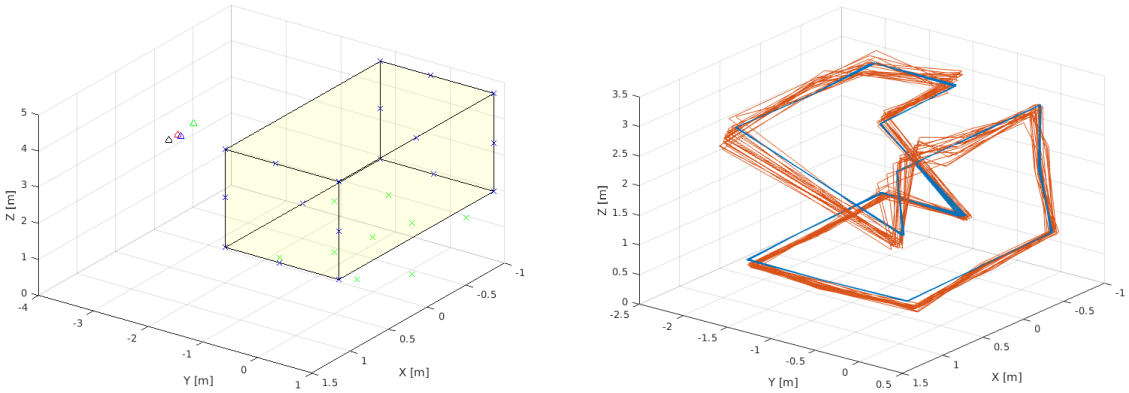


Figure 39: Proposed initialisation procedure (left). (Right) trajectory flown with the MAV

Figure 43: The proposed initialisation procedure (left) first using random triangulation points (green x) for coarse anchor initialisation (green triangle) and subsequently for the FIM optimisation to find optimal triangulation points (blue x) within a volume for position refinement (blue triangle). Also, the consideration of bias terms has an important positive performance impact (blue versus black triangle). Ground truth is

the red triangle. Right is the trajectory flown with the MAV fusing the UWB anchors initialised with the proposed approach and inertial data (red: estimated, blue: ground).

Modular multi-modal sensor fusion framework (adapted from Allak2022a and [Brommer2020a]):

In order to allow a unified localisation platform, a Modular and Robust Sensor-fusion (MaRS) Framework was developed [Brommer2020a]. This framework not only serves as unifying estimator on the UNI-KLU MAV for localisation and control testing and verification (see also deliverable 5.1) but was also integrated on the AUV by NTNU for improved localisation and more versatile sensor usage. State-of-the-art recursive sensor filtering frameworks allow the fusion of multiple sensors only tailored to a specific problem but do not allow a dynamic and efficient introduction of additional sensors during runtime: an important feature to enable long-term missions in dynamic environments and to render a localisation approach versatile for unified localisation across heterogeneous platforms. In contrast, the developed MaRS is a modular sensor-fusion framework that enables the addition and removal of sensors at runtime. These sensors could be not a priori known to the system. The framework handles the complexity of system and sensor initialization, measurement updates, and switching of asynchronous multi-rate sensor information with sensor self-calibration in a truly modular and generic design.

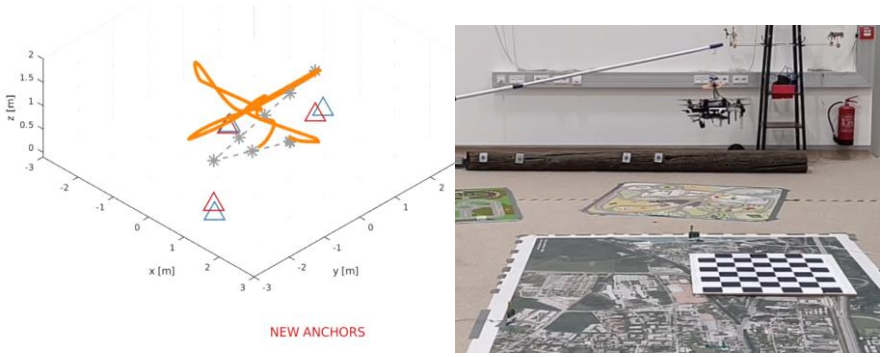


Figure 44: Closed loop controlled image based flight for a UWB initialisation test referencing the UWB anchors in the visual navigation frame (no GNSS).

In addition, the framework can handle delayed measurements, out-of-sequence updates, and can monitor sensor health. The introduced true modularity is based on covariance segmentation to allow the isolated (i.e., modular) processing of propagation and updates on a per-sensor basis. MaRS can maintain crucial properties of the overall state covariance. Naive implementation of such a modularization would invalidate the covariance matrix. The framework acts as a central localisation unit on the UNI-KLU MAV and the NTNU AUV. In [Brommer2020a] it is thoroughly evaluated for a precision landing scenario switching between combinations of GNSS, barometer, and vision measurements. Tests are performed in simulation and in real-world scenarios to show the validity of the introduced method. The MaRS technology is patented in Austria (AT-523734) and open sourced on GitHub (https://github.com/aau-cns/mars_lib).

Figure 9 depicts a scenario, where MaRS was used for a camera supported take-off and only then GPS and barometer were added as additional navigation sensors. During the main mission, the camera support was stopped and only re-activated (with self-calibration by MaRS) for the landing procedure. Orange is the flown trajectory, grey stars the dynamically determined waypoints, blue initial and red refined anchor pose estimates.

MaRS (and any other sensor fusion system merging multiple potentially delayed sensor signals through a statistical approach) is challenging for resource constrained platforms. For statistical consistency, it is required to keep an appropriate history, apply the correcting signal at the given time stamp in the past, and re-apply all information received until the present time. This re-calculation becomes impractical (the bottleneck being the re-propagation of the covariance matrices for estimator consistency) for platforms with multiple sensors/states and low compute power such as the MAVs and AUVs in the project (the tethered crawlers may provide certain off-board computation with fast and reliable data connection). In [Allak2022a] an approach for consistent covariance pre-integration was developed to allow delayed sensor signals to be incorporated in a statistically consistent fashion with very low complexity. Insights from the

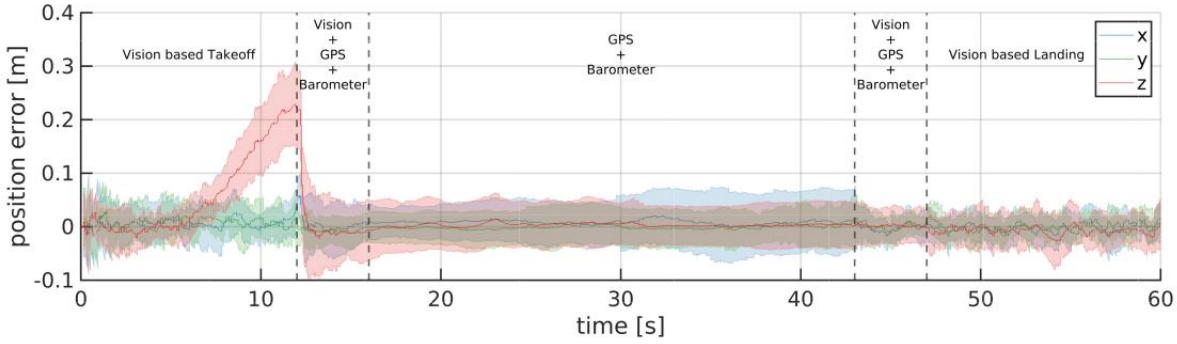


Figure 53: State error for the position and orientation of the core state in MaRS in an adaptive multi-modal cross-domain flight.

scattering theory were used to mimic the re-calculation process as a medium through which we can propagate waves (covariance information in this case) in single operation steps.

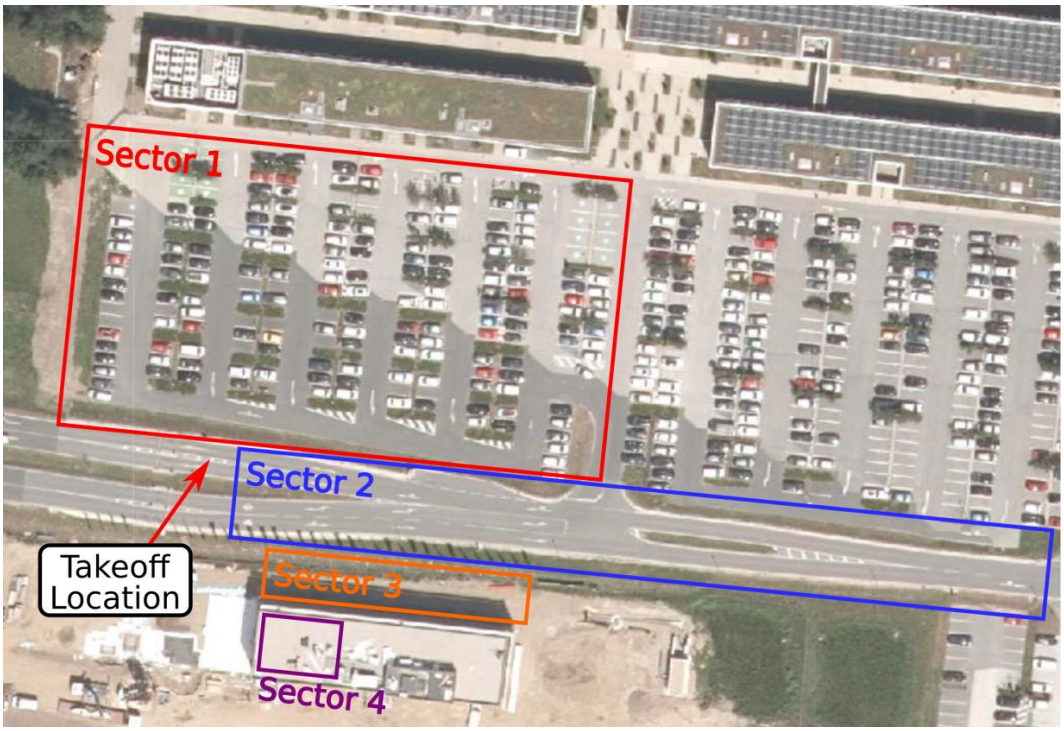
The modularity and self-calibration capability of MaRS directly allows to globally reference the navigation frame if GNSS signals are available. If no GNSS signal is available MaRS synchronises all sensor modalities to the initially observed, fixed navigation frame. In this project this is the UWB reference frame (since the camera navigation frame is drifting in position and yaw). Figure 10 is a state error for the position and orientation of the core state in MaRS in an adaptive multi-modal cross-domain flight. This scenario was performed with 20 datasets to gain a statistically significant result for the truly modular approach. The initial increase of the error in z-position is caused by vision drift due to the takeoff maneuver.

iii. Experiments and Integration

The above described building blocks consisting of the extended OpenVINS framework, UWB initialisation method and inclusion in the overarching MaRS localisation module has been extensively tested and prepared for the following efforts on multi-agent extensions in the remainder of the project. Apart of the inclusion in a resilient autonomy and control module (see deliverable 5.1) a large data set was collected to test indoor, outdoor and transition missions. The set contains the following data:

Sensor	Type	Rate [Hz]	Description
High Rate IMU	LSM9DS1		
IMU		900	Rigidly attached
Magnetometer		20	
Pixhawk			
Inertial Measurement Unit (IMU)	ICM20689	200	Internaly dampened
IMU	BMI055	200	(Disabled)
Global Navigation Satellite System (GNSS)		5	
Magnetometer	UST8310	80	
Barometer	MS5611	20	
Motorspeeds		100	
RealSense T256			
IMU	BMI055	200	
6DoF Odometry	V-SLAM	200	
Cameras			
Nav Camera	IDS UI-3270LE-M-GL	20	CMOS Mono, 2056x1542, global-shutter; Lens BM4018S118C, FoV(D=126°, H=101°), 3MP, aperture 1.8
Stereo Cam	RealSense T256	30	848x800, global-shutter, 64mm baseline, 163°FoV
External Sensors			
2× Real Time Kinematic (RTK) GNSS	UBLOX C94-M8P	8	Coordinates and velocity
Laser Range Finder (LRF)	Garmin Lidar Lite v3	30	40m range,1cm resolution
3×Ultra-wideband (UWB)	Decawave TREK1000	?TBD?	With additional vehicle marker
125× Fiducial Marker	ArUCO		Rate is the same as Nav Cam
Motion Capture	Optitrack	300	37 camera dronehall setup (TBD accuracy)
Pulse Tachometer	Wachendorff PT99		RPM ground-truth

The dataset was captured at the UNI-KLU campus (sectors 1-3) and within the UNI-KLU drone hall (sector 4).



Due to Covid constraints, no real flights on a real ship could be performed. That said, a sophisticated virtualisation framework for the drones was created. Using the above described VINS-Eval [Fornasier2021a] and virtual ship modelled therein, real flying drones in the UNI-KLU drone hall were "equipped with AR goggles": The real drone was using the real IMU for a real flight in the real drone hall. However, the images for the camera based navigation were not obtained through the real camera. Instead, the tracking system in the drone hall provided the current pose of the MAV to the virtual, coordinate aligned ship and VINS-Eval created online the corresponding image (see Figure 80). This proved to be a powerful tool not only to bridge covid related integration issues but also to emulate a variety of different environments and conditions therein -- leveraging all the benefits of VINSEval combined with real drone flights.

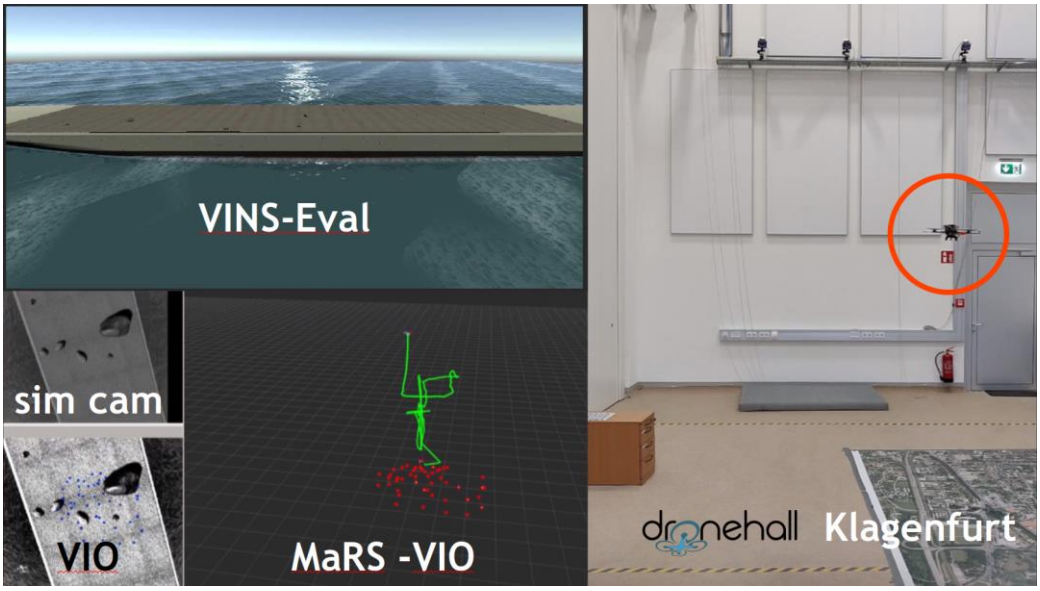


Figure 73: Closed loop AR flight test in the UNI-KLU drone hall. Right: real flight of the drone in the drone hall. Left top: simulated ship in VINS-Eval. Left bottom-left: simulated camera image based on the real drone position transformed to the simulated ship and image used by the VIO on-board the real drone. Left bottom-right: Estimated trajectory by the state-estimator using real IMU and simulated camera data.

As a synchronised effort, apart from merging the above described framework with the UIB framework on the M100 demonstration platform, comparable flight test as shown in Figure 80 was performed using a simulated mock-up that is currently being built for real tests (see Figure 88, the camera based approach handles well the fairly homogeneous texture and challenging geometry).

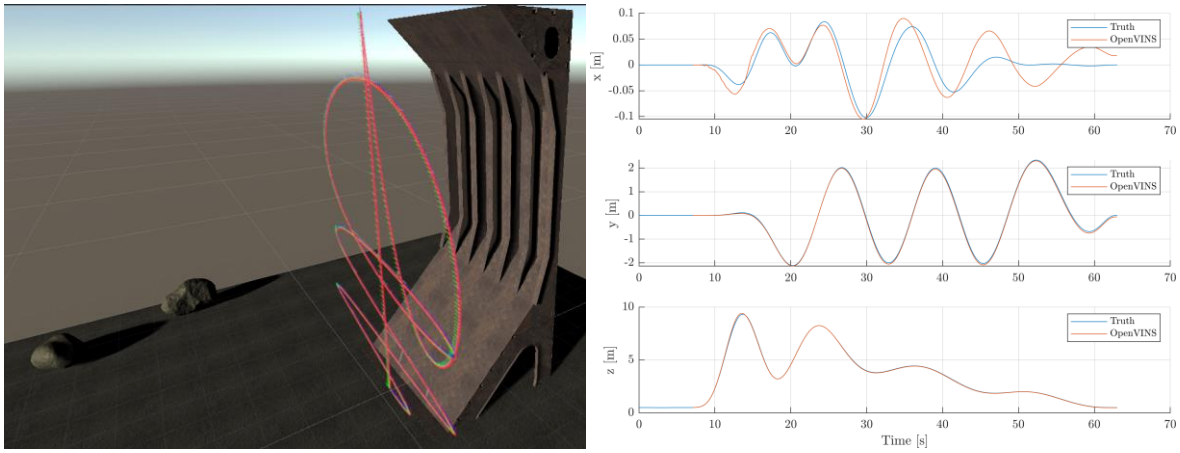


Figure 81: Similar setup as depicted in Figure 80 but with simulated and artificially textured mock-up system currently built in the project for initial real tests.

2. Laser-based motion estimation and localisation

i. Sensor Modalities

The previously described multi-sensor approach focusing on fusing UWB, GNSS, and images can tackle a variety of real-world situations. However, GNSS shadowing, UWB multipath issues, and visually homogeneous textures quickly deteriorate the performance of the overall localisation when only taking these sensor modalities into account. Thus, UIB focused on depth-based state estimation to include the geometrical aspects of the environment. Formulating this element in a loosely coupled framework allows to not only add further sensors in a stand-alone fashion but to seamlessly include the previously described camera and UWB focused approach. This architecture also allowed UIB and UNI-KLU to develop the different modules of the overall MAV state estimation framework largely independent without the need (and the possibility due to Covid) of many physical integration meetings. For the depth-based part of the MAV localisation, the following table summarizes the sensors, rates, and their usage.

Sensor	Specifications	Rate	Usage
DJI 6DoF IMU	3DoF gyro, accelerometer, and magnetometer, plus barometer	50Hz	Main modelling (propagation) sensor
Ouster Laser OS1 scanner	120m range, 1.3M points per second	20Hz revolution	Main navigation sensor
Depth camera Realsense D435i/D455	3m (6m for the D455) range, HD resolution	30Hz (90Hz for the D455)	Main navigation sensor
RTK GNSS receiver	Up to cm-precision, 16cm and worse with bad, no RTK fix	7Hz	For global navigation frame alignment
Decawave UWB	50m range, decimetre precision	10Hz (varying with number of anchors)	For navigation frame alignment across robots

The output of the localisation framework is used as input in the DJI SDK setting the current (and desired) position.

ii. Localisation Method Description

Specifically for the MAV localisation on the demonstration platform DJI M100, two depth-based concepts were pursued: depth-based state estimation using depth cameras and using laser scanners. Both methods were extended with GNSS and UWB support to best merge with the previously described image-based methods. The elements on depth-based localisation consists of:

1. RGB-D camera-based state estimation using environmental properties for improved localisation, place recognition, and loop closure. [Company2022a]
2. Lidar based state estimation using adaptive local mapping. [GarciaFidalgo2021a]
3. Extension to use initialise d UWB anchors in a pose computation approach that is resilient to multi-path issues due to improved triangulation techniques. [Bonnin2020a]

RGB-D camera-based state estimation (adapted from [Company2022a]):

Visual odometry algorithms tend to degrade when facing low-textured scenes —from e.g., human-made environments such as ship hulls or containers on cargo decks—, where it is often difficult to find a sufficient number of point features. Alternative geometrical visual cues, such as lines, which can often be found within these scenarios, can become particularly useful. Moreover, these scenarios typically present structural regularities, such as parallelism or orthogonality, and hold the Manhattan World assumption. Under these premises, [Company2022a] introduces an RGB-D-based visual odometry approach that combines both point and line features and leverages, if possible, those structural regularities and the Manhattan axes of the scene. Within this approach, these structural constraints are initially used to estimate accurately the 3D position of the extracted lines. These constraints are also combined next with the estimated Manhattan axes and the reprojection errors of points and lines to refine the camera pose by means of local map optimisation. Such a combination enables to operate even in the absence of the aforementioned constraints, allowing the method to work for a wider variety of scenarios. Furthermore, a novel multi-view Manhattan axes estimation procedure that mainly relies online features is developed. The approach dubbed MSC-VO is assessed using several public datasets, outperforming other state-of-the-art solutions, and comparing favourably even with some SLAM methods.

Lidar based state estimation (adapted from [GarciaFidalgo2021a]):

Light Detection and Ranging (LiDAR) technology is known as a robust alternative for self-localisation and mapping. These approaches typically state ego-motion estimation as a non-linear optimisation problem dependent on the correspondences established between the current point cloud and a map, whatever its scope, local or global. [GarciaFidalgo2021a] proposes a novel LiDAR-only odometry and mapping approach (dubbed LiODOM) for pose estimation and map-building, based on minimizing a loss function derived from a set of weighted point-to-line correspondences with a local map abstracted from the set of available point clouds. It places a particular emphasis on map representation given its relevance for quick data association.

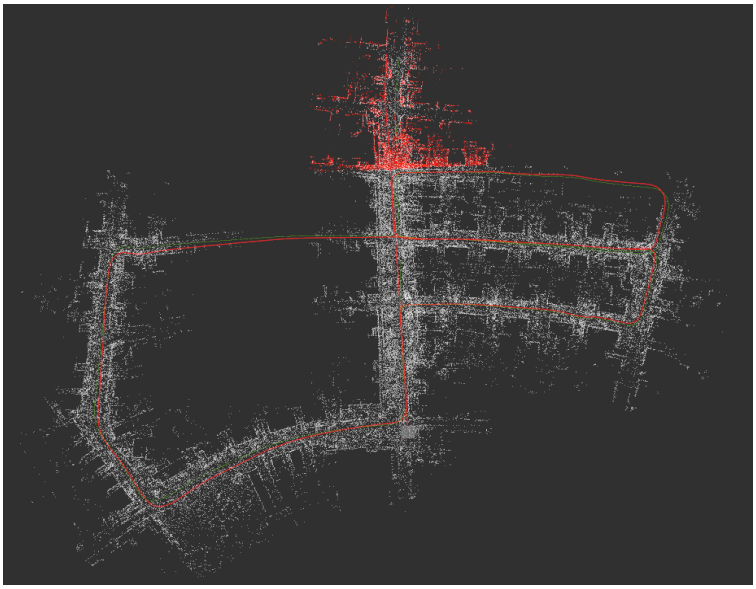


Figure 89: Example of map produced by LiODOM (KITTI 05 sequence), comprising an un-optimised global map generated during navigation (in white) and a local map (in red) that is retrieved according to the position of the vehicle, to be used for next pose estimation.

To efficiently represent the environment, a data structure that, combined with a hashing scheme, allows for fast access to any section of the map is developed. LiODOM is validated by means of a set of experiments on public datasets (see Figure 96), for which it compares favourably against other solutions. Its performance on-board an aerial platform is also reported in the corresponding publication [GarciaFidalgo2021a].

UWB based localisation [adapted from Bonnin2020a]:

Following the self-initialisation approach presented above in [Blueml2021a] UWB anchors can readily be used for localisation. Compared to [Blueml2021a] where such information is used rather rudimentarily in a tightly coupled approach, the triangulation based approach in [Bonnin2020a] show some resiliency against multi-path issues. This is particularly important since the project's environments can comprise metallic structures or other elements which can negatively affect the signal transmission and hence the accuracy of UWB-based position estimations. Regarding this fact, [Bonnin2020a] proposes a novel method based on point-to-sphere ICP (Iterative Closest Point) to determine the 3D position of a UWB tag. In order to improve the results in noise-prone environments, the method first selects the anchors' subset which provides the position estimate with least uncertainty (i.e., largest agreement) in the approach. Furthermore, a previous stage to filter the anchor-tag distances is used as input of the ICP stage. Also, the addition of a final step based on non-linear Kalman Filtering to improve the position estimates is considered. Performance results for several configurations of our approach are reported in the experimental results in the publication [Bonnin2020a], including a comparison with the performance of other position-estimation algorithms based on trilateration. The experimental evaluation under laboratory conditions and inside the cargo hold of a vessel (i.e., a noise-prone scenario, see Figure 104) proves the good performance of the ICP-based algorithm, as well as the effects induced by the prior and posterior filtering stages.

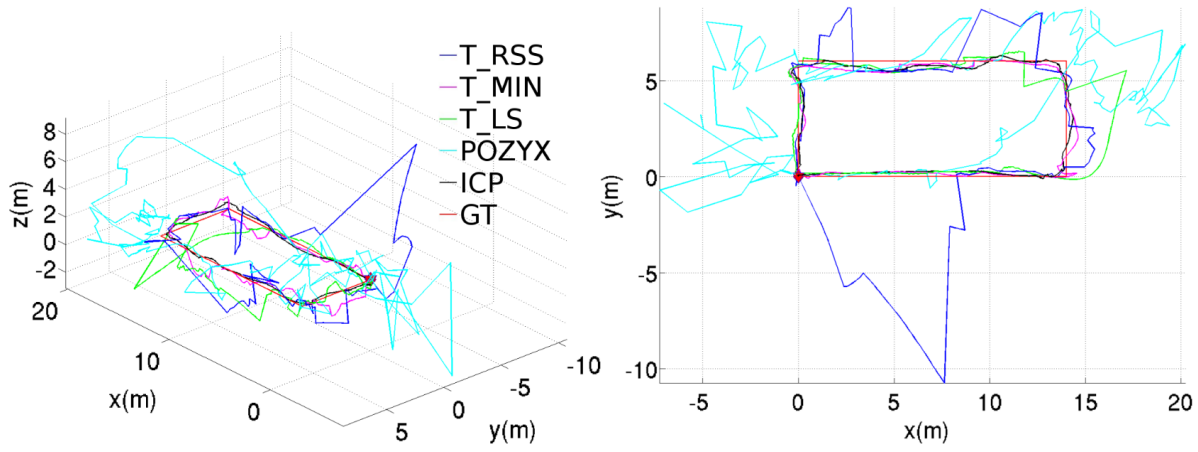


Figure 97: Position estimations provided by the different methods for a rectangular trajectory performed inside the vessel hold: ground truth (GT), proposed ICP based method (ICP), off-the-shelve triangulation method (POSZYX) and other variants (T_*). Clearly the proposed method strongly improves the estimation in such difficult scenario making it a strong candidate for the project's estimation framework.

iii. Experiments and Integration

The above-described building blocks on depth based localisation were rigorously tested on several public datasets as mentioned above and also integrated on the DJI platform to perform closed loop flight tests. Several test runs were performed at the UIB campus and using a Ro-Ro vessel data set gathered with the M100 MAV with the real sensor modalities. Additional details can be found in the corresponding publications. Figure 112 and Figure 120 depict the experiments on the UIB campus regarding the LiDAR based approaches.

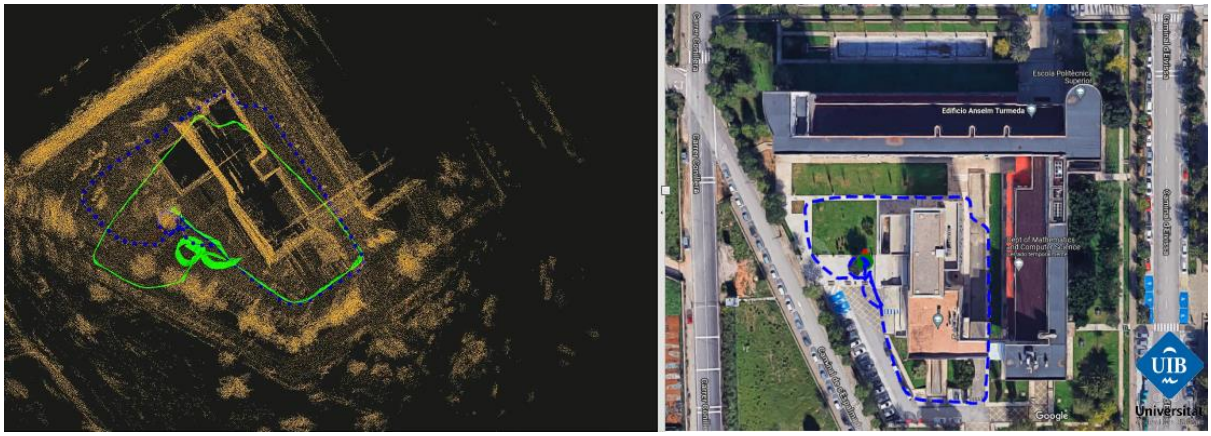


Figure 105: Testing of the multi-sensor state estimation localisation & mapping functionalities of the depth-oriented state estimation at the UIB campus: 3D laser-based odometer LiODOM and 3D laser-based SLAM = LiODOM + loop closure detection.

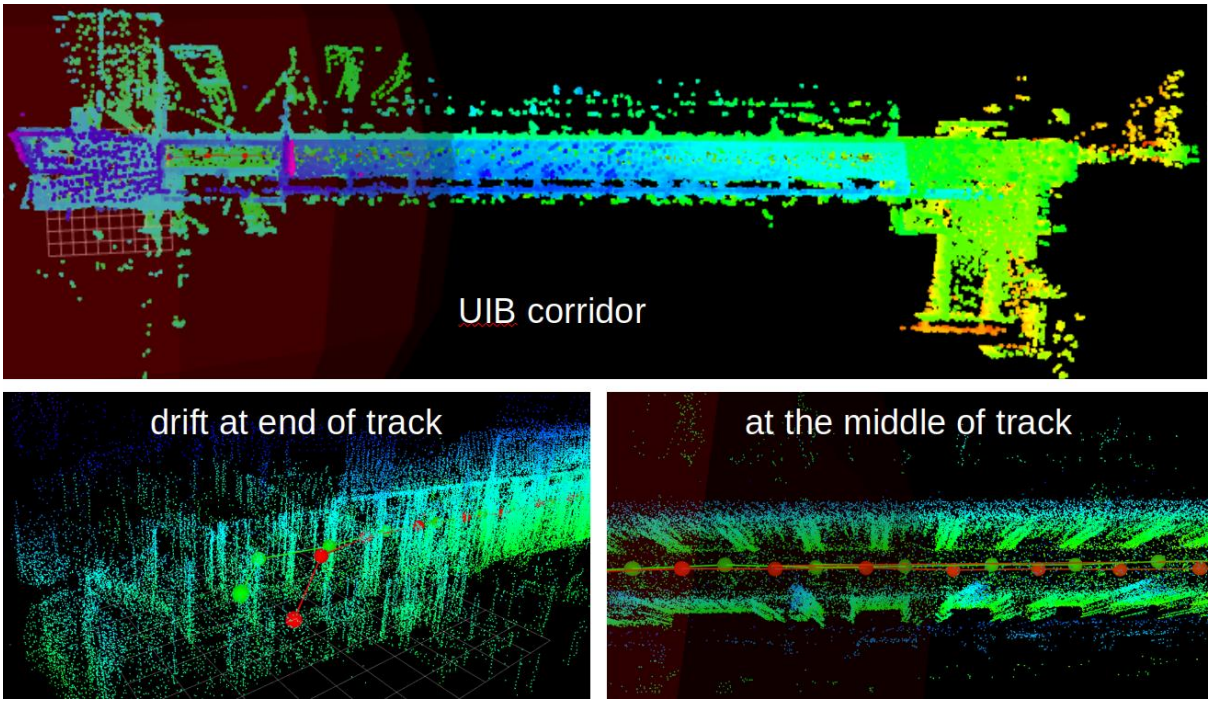


Figure 113: LiODOM-based SLAM test in a UIB corridor. LiODOM is running onboard the MAV while loop closing and global map optimization is running on the base station.

The RGB-D approach (MSC-VO based SLAM) was tested on the Ro-Ro vessel dataset (see Figure 128). Besides the 6DoF trajectory, a point cloud was aggregated, and a mesh was generated.

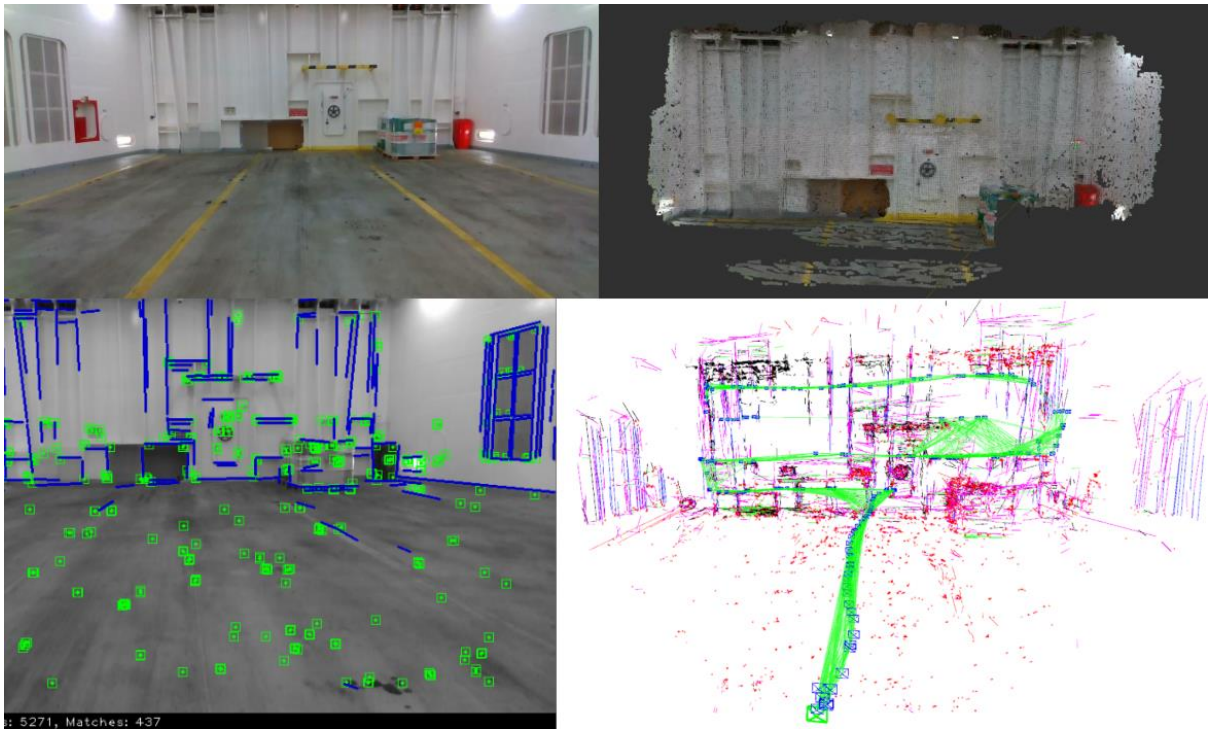


Figure 121: Experiment for the RGB-D based MSC-VO depth-based state estimation on the Ro-Ro vessel dataset. Top left: scene image. Top right: aggregated point cloud for subsequent scene meshing. Bottom left: presented MSC-VO approach using points and lines. Bottom right: estimated scene features and 6DoF pose.

Besides the tests on the vessel data set for the UWB based localisation mentioned previously, the approach was also implemented on a ground robot for live tests in a project relevant setup with UWB anchors placed vertically on a wall. Different estimator types were implemented (EKF, IEKF, AEKF, etc.) and compared against each other. Also, different anchor placements to test the influence of the mesh geometry to the localisation precision. A sample test setup is shown in Figure 136.

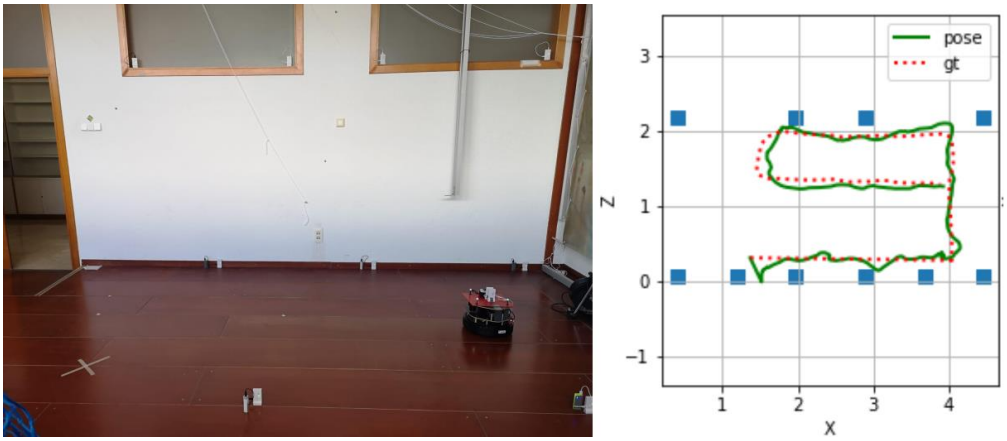


Figure 129: Sample test setup for the UWB based localisation method at the UIB laboratory. 10 anchors were placed on a vertical plane and at different ranges in x and z direction, but same direction at 0m and 2.46m.
The RMSE for this setup and an EKF implementation (see right plot) is about 9cm.

3. Combination of approaches

After separate development, evaluation, and testing the image (led by UNI-KLU) and depth based (led by UIB) approaches were merged together. This was done using UIB's loosely coupled overall cascaded EKF based estimator framework (see schematic in Figure 152): IMU propagation and depth based odometry information (LiDAR, RGB-D odometry, and altimeter) are fused in a local EKF module. A global EKF module fuses GNSS data (if available) and globally referenced UWB modules as well as SLAM information. Similarly to the local EKF module, UNI-KLU's image based framework feeds into the global EKF as an external, virtual sensor. The output is a resilient platform pose for control. Several tests were already performed with data gathered with the real M100 platform and sensors as well as with the hardware in the loop with simulated data. Figure 144 shows the results of these tests. Real live tests in front of the real mock-up are planned in the April'22 and June'22 integration weeks. This merged framework marks the milestone of MAV localisation for a single platform in the project's relevant settings in view of platform type, sensors, and, to some extent modulo real-world mock-up, environment.

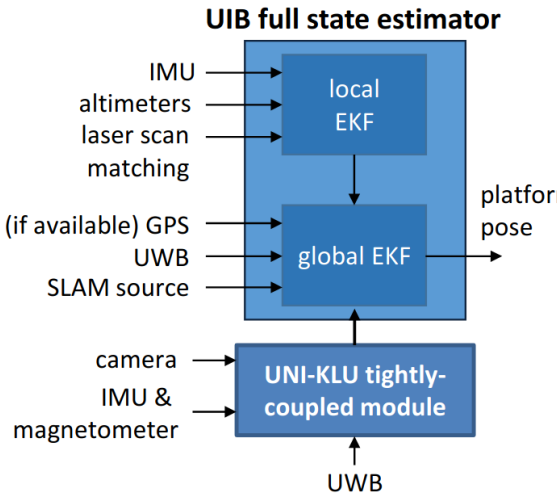


Figure 145: UIB's cascaded EKF based estimator framework using a local EKF for IMU propagation, altimeter reading, and LiDAR odometry. A global EKF module includes GNSS signals, referenced UWB anchors, and SLAM information. Additionally, it inputs the UNI-KLU's image based localisation and UWB initialization module as virtual sensor.

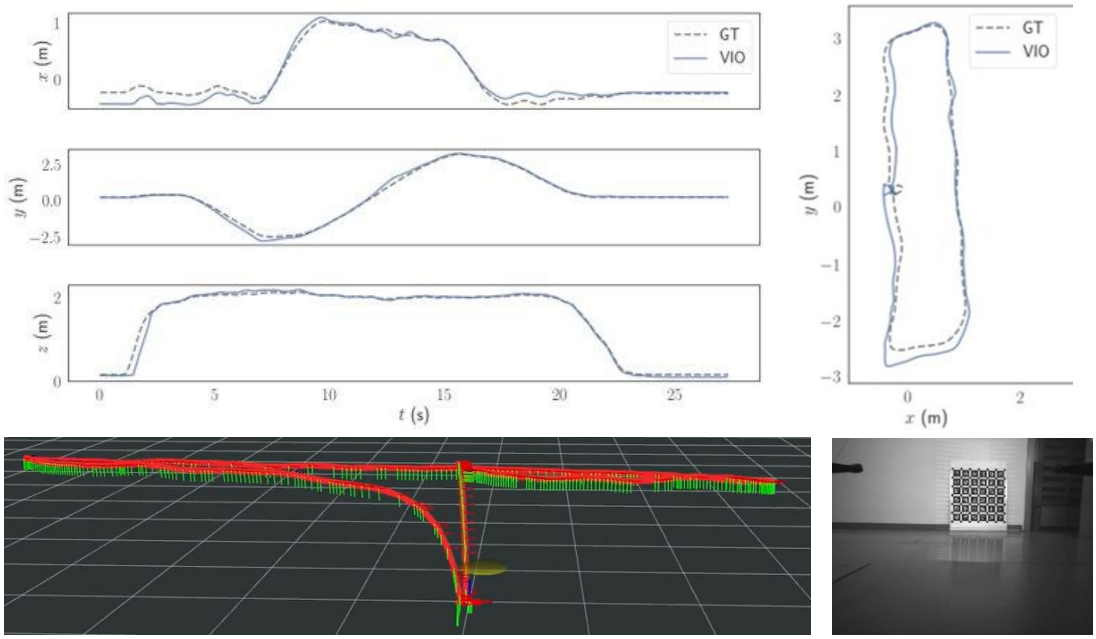


Figure 137: Trajectory flown using the UNI-KLU image based localisation in the UIB cascaded EKF framework. Top left: Position in [x, y, z]. Top right: top down view of the trajectory. Bottom left: trajectory visualized in RViz. Bottom right: live image of the challenging (reflections and low texture) environment.

III. Localisation approach for the autonomous underwater vehicle

In the following, we describe and evaluate the localisation approaches on motion estimation and localisation developed and integrated into the BUGWRIGHT2 AUV. Following the state estimation methods in full 3D space for the MAV, the AUV can directly adapt the core methods. This applies particularly to the modular multi-sensor fusion method MaRS developed in [Brommer2020a] described in Section II.1.ii.e. MaRS is highly modular and versatile such that the sensing modalities from the MAV were easily replaced by the ones of the AUV. Moreover, similarly to the GNSS signal for the MAV, the USBL (with its GNSS sensor and defined heading) provides means to globally align the AUV navigation frame whenever a USBL signal is available. Even though those signals are very sparse and noisy (in a way similar to often distorted GNSS signals), MaRS will include the provided information in a statistically best possible fashion for AUV navigation frame alignment. Other sensors as well as the fusion results are detailed in the following sections.

1. Sensor Modalities

The AUV setup features the two IMUs with an integrated inclinometer algorithm. This algorithm averages the readings of the two IMUs and provides an absolute roll and pitch and a relative (to the start pose) yaw reading. Both IMUs also provide 3DoF gyro, accelerometer, and magnetometer readings. In addition, a doppler velocity sensor DVL is mounted that provides 3DoF body velocity and the distance to ground. The sensor also features its own IMU with an inclinometer and an algorithm that integrates the body velocity rotated into the inertial frame to a 3D position relative to its start position. The mentioned USBL device features a GNSS receiver and allows a fix heading correction such that its Cartesian readings can be aligned with a local GNSS frame. Through this information and using MaRS that can adequately incorporate this information in a self-calibrating fashion, the UAV can be globally referenced in the same frame as the remaining robotic platforms of the project. Figure 160 depict the sensor setup on the AUV and the table below lists the sensors.

Figure 153: AUV setup with the sensors and their frames as used in MaRS.

Sensor	Specifications	Rate	Usage
2x IMU MPU-9250	3DoF gyro, accelerometer, and magnetometer	200Hz	Main modelling sensor (averaged propagation)
SeaTrac USBL X115/X110	1km range	10Hz	Global referencing
WaterLinked DVL A50	Body velocity, distance to ground (50m), attitude, and integrated position	8Hz	Attitude and body velocity used for IMU integration correction. Integrated position as verification.
MS5837-30BA Pressure sensor	Up to 30 bar	10Hz	Depth below water surface
Rasberry PI 3 equivalent	System on Chip		On-board computing

2. Localisation Method Description

The navigation method is identical to the MaRS approach presented in Section II.1.ii.e based on [Brommer2020a]. For the AUV, MaRS is initialised with the local position. Initial roll and pitch are determined using the averaged gravity vector from the IMU. The heading is initialised using the magnetometer. Note that this can be disturbed and has a declination with respect to the GNSS absolute heading. However, this initialisation is still well in the convergence basin such that the AUV's global attitude quickly converges with first movements and USBL readings upon mission start. The same is true for the initial roll and pitch which may be polluted by IMU biases and initially not correspond to the true values. During operation, IMU based pose propagation is executed. The integration errors are corrected by the DVL body velocity measurements, the DVL attitude measurement, and the depth sensor pressure readings. Sporadically, if MaRS deems the USBL readings as trustworthy, 3D USBL readings are incorporated. With time and motion, the latter ensures global alignment of the AUV with the remaining robots. This is the important benefit of using MaRS instead of directly using the integrated position by the DVL manufacturer. In the latter case, the (statistically correct) inclusion of global information or other additional sensing modalities would be increasingly complex with more additional sensors. MaRS modularly handles sensor modality additions in a statistically correct fashion, such that statistical tests can be performed on the quality of any sensor signal.

3. Experiments and Integration

Several tests have been conducted in a fjord area and in a pool for ground truthing. Generally, ground truthing for the AUV is a difficult task. Even though USBL was used in the pool area and in the fjord datasets, this type of sensor has a low rate and is, even under well behaving conditions, very noisy. For the estimation verification, in addition to the USBL data, the integrated DVL velocity as a relative position was used. This is an output the manufacturer provides and is deemed, apart from the inherent drift due to integration, as reliable in a relative reference frame during short mission periods.

For the experiments, MaRS was implemented in the embedded CPU on the AUV and ran live during the data gathering process. This allows a verification on the run-time performance of MaRS as well as post processing using the gathered raw and verification data. Figure 168 depicts a run in the fjord using USBL, pressure sensor, DVL body velocity, and DVL attitude in MaRS compared to the raw USBL measurements and the integrated DVL velocity. For the short run, the drift of the integrated DVL velocity is negligible and shows, thus, similar performance to MaRS. That being said, the integrated velocity was manually aligned with the global reference frame in a post processing step while MaRS performed this calibration online automatically. It is also clearly visible that the USBL information is highly noisy. Nevertheless, MaRS is able to leverage the best of all information providing a smooth, accurate, yet globally aligned pose estimation of the AUV.

With this framework, we are now able to localise a single AUV robot in the global reference frame along all other robots in the project. This can be leveraged in upcoming steps regarding multi-agent methods.

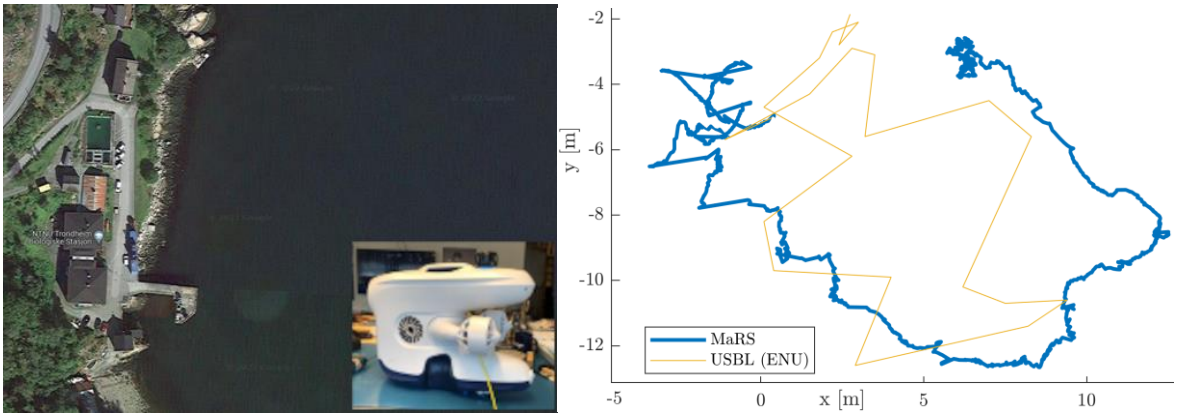


Figure 161: AUV localisation test in the fjord using USBL, DVL velocity and attitude, and IMU readings. Even though in seemingly good conditions (without interferences from harbour structures and ships), the USBL (orange) readings are highly noise. However, MaRS (blue) is able to leverage the velocity information from DVL for a much smoother pose estimate while using the sparse, noise prone USBL readings for maintaining global reference in position and heading.

IV. Localisation approach for the crawlers

In the following, we describe and evaluate the localisation approaches on motion estimation and localisation developed and integrated into the two BUGWRIGHT2 crawler types. This task is strongly connected to the Task 3.1 on simultaneous localisation and geometry inference on a metal plate. And the corresponding deliverable 3.1. Thus, we focus on the localisation elements not including the metal plate geometry inference and only give a brief overview of those efforts. While the metal plate localisation methods are valid for the above- and below-water platforms, other localisation methods are only valid for one or the other. In the following sections we focus in particular on UWB and LiDAR based localisation for the above-water platforms while USBL based localisation methods are used for the below-water.

1. Sensor Modalities

Apart from the transducer used in WP3 (see deliverable D3.1), several sensors are used to perceive the environment for path planning, obstacle avoidance, and localisation. The table below summarises the different sensors. Instead of using the IMU as a core propagation sensor as it is done for the MAV and the AUV, the crawlers use the wheel odometry as primary motion model. The IMU, due to the low excitation during the crawler motion only provides minimal information and may best be used as an inclinometer. For local positioning 3D point clouds from a LiDAR, stereo or IFM sensor is used. Global positioning is achieved through UWB anchors.

Sensor	Specifications	Rate	Usage
Wheel encoder	mm-precision	50Hz	Main modelling (propagation sensor)
MPU-9250 9DoF IMU	3DoF gyro, accelerometer, and magnetometer	100Hz	Inclinometer
Ouster Laser OS1-16 scanner	120m range, 1.3M points per second	20Hz revolution	Former navigation sensor for testing
IFM IP69 3D camera O3D305	8m range	20Hz	Former navigation sensor for testing
Robosense bPearl IP69 laser 3D	30m range, 1.1M points per second	20Hz revolution	Former navigation sensor for testing
LiVOX MID-70 3D laser	90m range	100k points per second	Main navigation sensor
Depth camera Realsense D435i	3m range, HD resolution	30Hz	Images for colouring the point clouds
Decawave UWB	15m range, decimetre precision	<10Hz (varying with number of anchors)	For navigation frame alignment across robots

2. Localisation Method Description

Providing accurate and consistent localisation information to the crawlers is probably the most challenging compared to the MAV and AUV. Whereas the challenging task for the MAV is to provide uninterrupted localisation, the crawlers move very slow, can easily rest upon non-nominal state detection, and have very accurate wheel odometry making this sensor a viable source of information. Exteroceptive sensors improving this information are, however, complex to integrate due to the reflective surface and the challenging perspective the crawlers have thereof (very close to the structure). Furthermore, the surface may be curved and thus occlude global positioning information e.g., from UWB anchors mounted on the surface. Thus, the efforts on providing accurate crawler localisation focus on the following aspects:

1. Manifold constrained consistent estimation using UWB anchors and IMU propagation in an invariant Kalman filter formulation. [Starbuck2021a]
2. Mesh-constrained particle filter-based estimation using UWB anchors. IMU, and the crawler's odometry. [Schroepfer2022a]
3. FastSLAM based localisation on a metal plate leveraging reflections of ultrasonic guided waves in a particle filter setup. [Ouabi2021a]

Manifold constrained invariant Kalman filter (adapted from [Starbuck2021a]):

The developed Manifold Invariant Extended Kalman Filter is a novel approach for better consistency and accuracy in state estimation on manifolds such as tanks and ship hulls. The robustness of this filter allows for techniques with high noise potential like ultra-wideband localisation to be used for a wider variety of applications like autonomous metal structure inspection. The filter is derived, and its performance is evaluated by testing it on two different manifolds: a cylindrical one and a bivariate b-spline representation of a real vessel surface, showing its flexibility to being used on different types of surfaces. Its comparison with a standard EKF that uses virtual, noise-free measurements as manifold constraints proves that it outperforms standard approaches in consistency and accuracy (see Figure 176). Further, an experiment using the real magnetic crawler robot on the curved metal surface with ultra-wideband localisation shows that the proposed approach is viable in the real-world application of autonomous metal structure inspection.

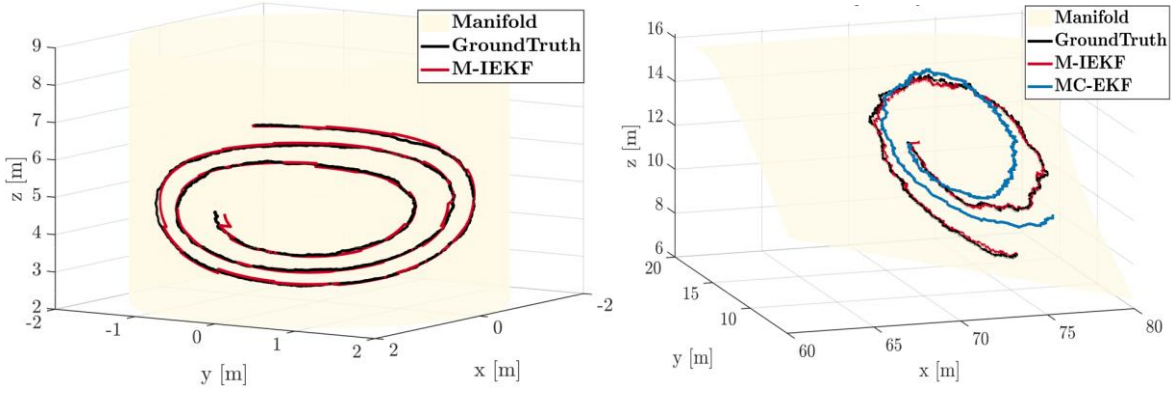


Figure 169: Manifold constrained invariant extended Kalman filter (M-IEKF) tested in simulation on a cylindrical shape and on a metal surface compared to a naively constrained EKF (M-EKF).

Mesh-constrained particle filter based estimation (adapted from [Schroepfer2022a]):

The invariant (and equivariant) extended Kalman filter requires certain geometries of the group on which the state vector and measurements are defined. This geometry is not directly satisfied when taking raw distance measurements from UWB into account nor when taking IMU acceleration or gyro biases into the state vector. Thus, for more versatile use, a mesh constrained estimator was developed using a particle filter approach. This field-tested mesh constrained particle filter for mobile robots is capable of estimating poses with 6DoF in real-time due to low particle count requirements. In this filter, particles are constrained by a mesh surface approximating the surface the robot is travelling on. By constraining the particles, the dimensions of the effective work space the robot is operating in is reduced. In other words, the robot is effectively lying on a manifold (locally) with 3DoF embedded in SE(3). Importantly, by reducing this effective workspace, significantly improved accuracy is achieved with low particle density when compared to a dense standard particle filter. This particle reduction also allows to represent particles with 6DoF in real-time on a mobile robot embedded computer. Further, by constraining particles to the mesh and avoiding the use of an Extended Kalman Filter, high levels of robustness to lost or dropped anchor measurements can be demonstrated.

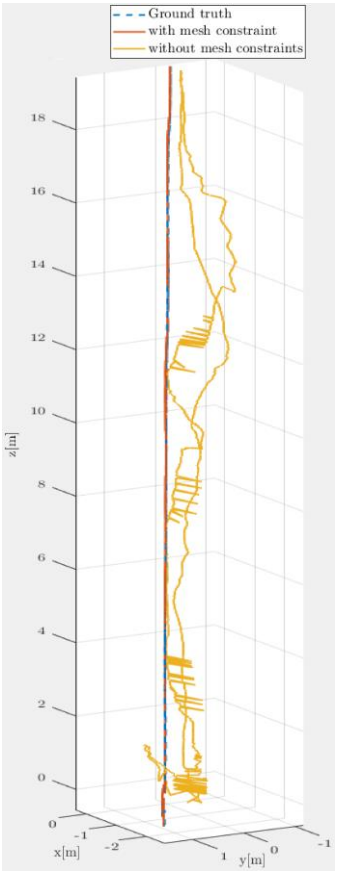


Figure 177: Crawler localisation on a 3D mesh of a metal tank using UWB, IMU, and crawler odometry. The chosen approach is a mesh constrained particle filter.

FastSLAM based localisation on a metal plate (adapted from [Ouabi2021a]):

This localisation method is more detailed described in deliverable 3.1 and mentioned here for completeness. A FastSLAM approach for a robotic system inspecting structures made of large metal plates has been developed. By taking advantage of the reflections of ultrasonic guided waves on the plate boundaries, it is possible to recover, with enough precision, both the plate shape and the robot trajectory. Contrary to previous work, this approach considers the dispersive nature of guided waves in metal plates. This is leveraged to construct beam-forming maps from which we solve the mapping problem through plate edges estimation for every particle, in a FastSLAM fashion. It is demonstrated, with real acoustic measurements obtained on different metal plates, that such a framework achieves more accurate results, while the complexity of the algorithm is sensibly reduced (see Figure 192).

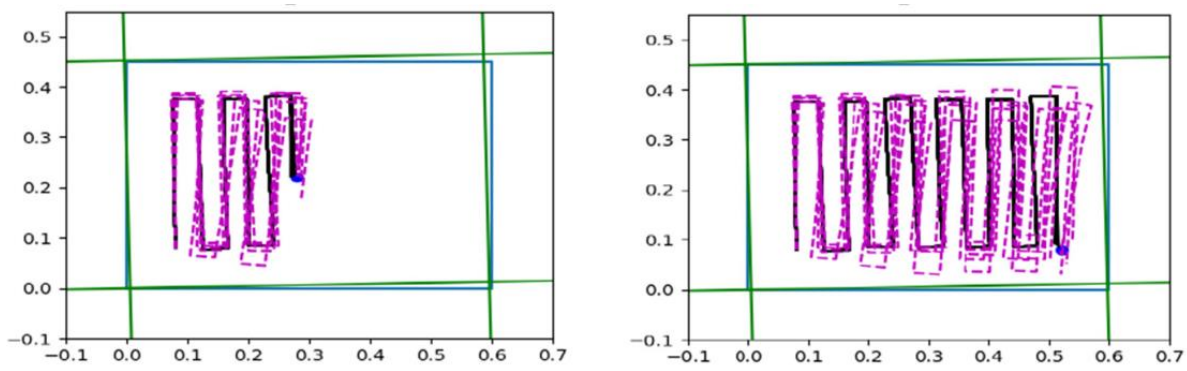


Figure 185: Trajectories estimated by all the particles (black lines), dead-reckoning trajectories (dash magenta lines) and map retrieved by the most likely particle (green lines) during Steps 50 and 108 for a lawn-mower path on a plate 1. The true outline of the plate and true sensor positions correspond to the blue rectangle and blue dot respectively.

3. Experiments and Integration

Several experiments were conducted on mock-up setups using a bended metallic plate up to a real-world test on a large metallic tank. The crawler localisation is mainly based on UWB in these experiments. Thus, and since the UWB anchors can be initialised in a reference frame common to all robotic platforms (see Section **Error! Reference source not found.**), the crawlers inherently operate in the same unified reference f rame. Additional frame alignment refinement can be investigated in the upcoming efforts in the project on multi-agent methods.

For the above-mentioned manifold constrained localisation, a mock-up with a bended metal plate was set up and tested with a real crawler and UWB measurements. Ground truth was obtained by an external laser scanner (see Figure 200). The developed invariant manifold constrained approach was tested against a naively constrained regular extended Kalman filter (see Figure 210).

Figure 193: Magnetic crawler robot (green arrow) on a curved metal surface with ultra-wideband localisation (red circles) and laser (yellow arrow) to track the robot for ground truth.

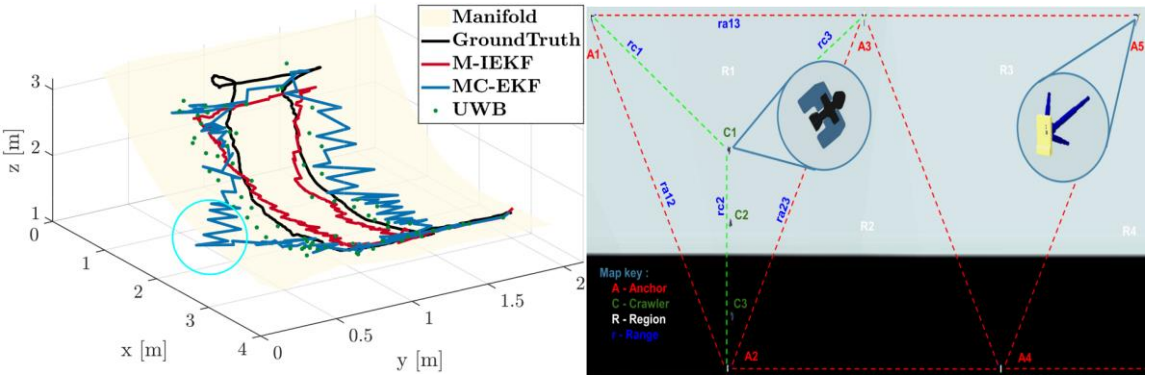


Figure 209: Localisation result on the mock-up plate showing the comparison of the naively constrained extended Kalman filter (MC-EKF) and the novel approach based on manifold mapping and invariant formulation (M-IEKF). Right: Schematic representation of the UWB anchor distribution and the measurements.

A larger integration and testing effort was done on a real tank structure. Ground truth was set up via laser scanner and several UWB anchors were placed to enable the IMU, odometry, and UWB based localisation aid of the above-described particle filter method (see Figure 208).

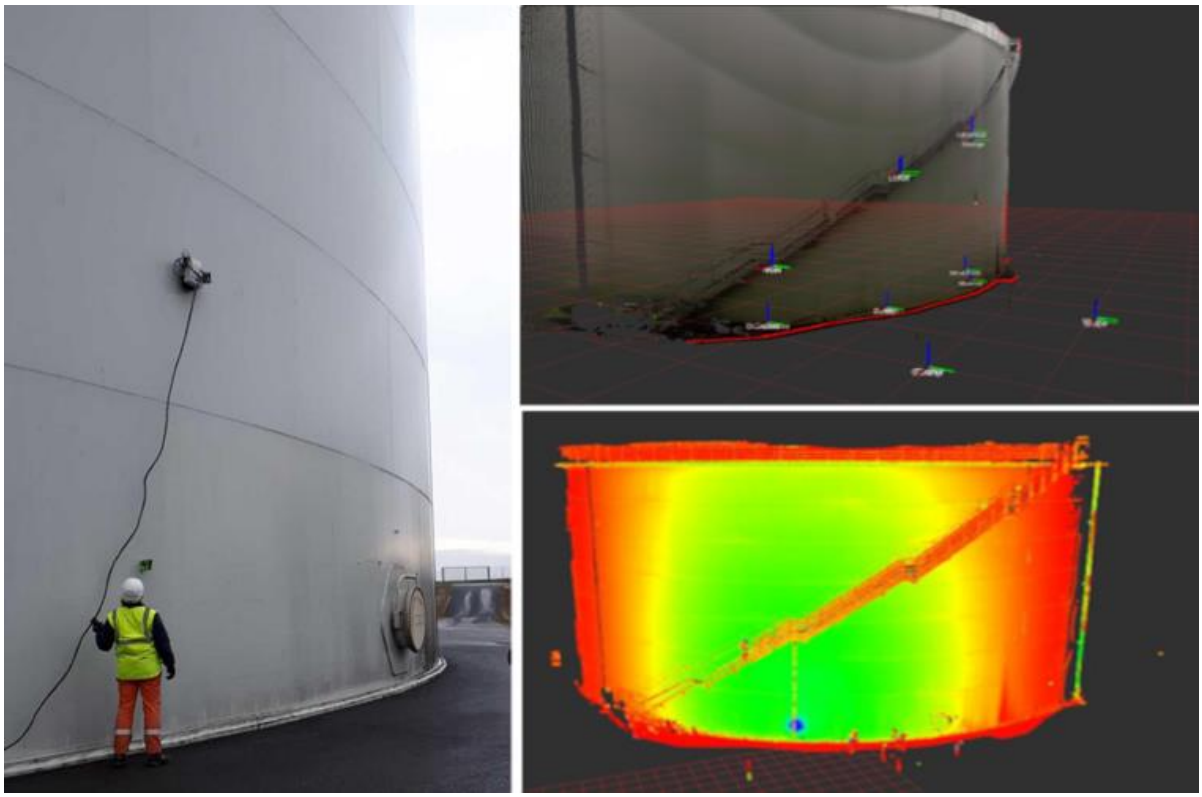


Figure 201: Crawler on a real metal tank in a real environment (left). The crawler was joystick-guided along the wall while the particle filter based estimation approach estimated its 6DoF (top right as red line). The ground truth as well as the mesh was captured by a laser scanner (bottom right).

Using the UWB (or USBL for the underwater crawler) inherently allows the crawlers to estimate their pose in the same reference frame as the remainder of the robots. Thus, the developed approaches enable a solid single-robot localisation that can further be used in the upcoming multi-robot tasks in the project.

V. Conclusions

Overall, the consortium reached a state where the robotic platforms can be localised in a common reference frame as single agents. Starting from the MAV with probably the highest requirements on localisation quality both in precision but also in robustness, the consortium developed image based, depth based, and UWB based methods for precise MAV localisation via a cascaded EKF framework. Within this effort, a UWB anchor initialisation has also been developed to initialise their positions in the respective reference frame that serves as a common coordinate system across all robot platforms in the project. The depth and image-based approaches for the MAV have been successfully merged and demonstrated with real platforms and data.

In particular the modular multi-sensor fusion approach used in the image-based approach for the MAV could directly be extended to act as the localisation backbone for the AUV. Using USBL, DVL, and pressure sensors for the IMU integration correction (instead of GNSS, camera, and UWB as for the MAV) the framework could directly be re-used to estimate the MAV pose in the UWB/USBL reference frame.

For the crawlers, due to their constraints on the manifold, different, manifold-constrained estimators were developed. An invariant extended Kalman filter approach proved to be highly consistent yet limited in application with complex sensors models. A particle-filter-based approach proved then to be accurate and very light in cost with good performance on a large real-world experiment on a tank structure. Again, relying on UWB (or USBL underwater) measurements, and with the MAV based initialisation routine for the UWBs, the crawlers can be localised in a unified navigation frame.

These results mark an important milestone rendering the global localisation of all robot platforms possible. This will be leveraged in upcoming effort in multi-agent aspects. We expect that multi-agent information will continue to improve the localisation accuracy of the single robot through collaborative state estimation methods.

Annexes

References

[Allak2022a] Eren Allak, Axel Barrau, Roland Jung, and Stephan Weiss: Centralized-Equivalent Pairwise Estimation with Asynchronous Communication Constraints for two Robots. Submitted to IROS 2022.

[Blueml2021a] Julian Blueml, Alessandro Fornasier, Stephan Weiss: Bias Compensated UWB Anchor Initialisation using Information-Theoretic Supported Triangulation Points. Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), IEEE, Xi'an, 2021.

[Bonnin2020a] Francisco Bonnin-Pascual, and Alberto Ortiz: UWB-Based Self-Localisation Strategies: An ICP-Based Novel Method and a Comparative Assessment for Noisy Ranges-Prone Environments Sensors, vol. 20, no. 19, article nr. 5613, 2020.

[Brommer2020a] Christian Brommer, Roland Jung, Jan Steinbrener, and Stephan Weiss: MaRS: A Modular and Robust Sensor-Fusion Framework. In IEEE Robotics and Automation Letters (RA-L), November 2020.

[Company2020a] Joan P. Company-Corcoles, Emilio Garcia-Fidalgo, and Alberto Ortiz: LiPo-LCD: Combining Lines and Points for Appearance-based Loop Closure Detection, British Machine Vision Conference, 2020.

[Company2022a] Joan P. Company-Corcoles, Emilio Garcia-Fidalgo, and Alberto Ortiz: MSC-VO: Exploiting Manhattan and Structural Constraints for Visual Odometry, IEEE Robotics and Automation Letters, vol. 7 no. 2 pp. 2803 – 2810, 2022.

[Fornasier2021a] Alessandro Fornasier, Martin Scheiber, Alexander Hardt-Stremayr, Roland Jung and Stephan Weiss: VINSEval: Evaluation Framework for Unified Testing of Consistency and Robustness of Visual-Inertial Navigation System Algorithms. Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), IEEE, Xi'an, 2021.

[GarciaFidalgo2021a] Emilio Garcia-Fidalgo, Joan P. Company-Corcoles, Francisco Bonnin-Pascual, and Alberto Ortiz: LiODOM: Adaptive Local Mapping for Robust LiDAR-Only Odometry, under review

[Ouabi2021a] O. -L. Ouabi, P. Pomarede, M. Geist, N. F. Declercq and C. Pradalier, "A FastSLAM Approach Integrating Beamforming Maps for Ultrasound-Based Robotic Inspection of Metal Structures", in IEEE Robotics and Automation Letters, vol. 6, no. 2, pp. 2908-2913, April 2021.

[Scheiber2021a] Martin Scheiber, Jeff Delaune, Stephan Weiss, and Roland Brockers: Mid-Air Range-Visual-Inertial Estimator Initialisation for Micro Air Vehicles. Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), IEEE, Xi'an, 2021.

[Schroepfer2022a] Pete Schroepfer, Georges Chahine, Cédric Pradalier: A Mesh Constrained Particle Filer for Real-Time Accurate State Estimation and 6DoF For Mobile Robots. Submitted to IROS 2022.

[Starbuck2021a] Bryan Starbuck, Alessandro Fornasier, Stephan Weiss and Cédric Pradalier: Consistent State Estimation on Manifolds for Autonomous Metal Structure Inspection. Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), IEEE, Xi'an, 2021.

Centralized-Equivalent Pairwise Estimation with Asynchronous Communication Constraints for two Robots

Eren Allak[1], Axel Barrau[2], Roland Jung[3], and Stephan Weiss[1]

***Abstract*— Collaboratively estimating the state of two robots under communication constraints is challenging regarding computational complexity and statistical optimality. Previous work only achieves practical solutions by either disregarding parts of the measurements or imposing a communication overhead, being non-optimal or not entirely distributed, respectively.**

In this work, we present a *centralized-equivalent* but *distributed* approach for pairwise state estimation where two agents only communicate when they meet. Our approach utilizes elements from the wave scattering theory for efficiently and consistently summarizing (pre-compute) past estimator information (i.e., state evolution and uncertainty) between encounters of two agents. This summarized information is then used in a joint correction step taking all past information of each agent statistically correct into account.

This novel approach enables us to distribute the pre-computations of both state evolution and uncertainties on the agents and reconstruct the centralized-equivalent system estimate with very few computations once the agents meet again while still applying all measurements from both agents on both estimates upon encounter. We compare our approach on a real-world dataset against a state of the art collaborative state estimation approach.

I. Introduction

Pairwise estimating states between two agents is, especially in the field of autonomously navigating systems, key to achieve precise and robust localization in challenging environments. Features like sensor sharing (e.g., propagating global information from a GNSS reception on one agent to the other) and the resulting redundancy or instantaneous capturing of a dynamic scene via shared pose information and the resulting variable baseline-stereo setup [1][2] are only some of the benefits that directly result from an accurate pairwise state estimation across two agents. As a specific real-world example, the variable and ad-hoc baseline formation finds application in e.g., landslide or avalanche monitoring, where two aerial agents can form a flexible, sufficiently large baseline on-demand and use their cameras for joint photogrammetric reconstruction of the dynamic events.

Strictly speaking, only a centralized fusion of all the information of both agents at any time a reading is processed would adequately consider cross-correlations of states on and between the state maximizing consistency and accuracy of the overall system's state estimate. This is generally computationally not tractable for small mobile systems with limited computing power. Current approaches either approximate inter-agent correlations or assume them to be unknown. Other approaches that maintain correlations between agents, on the other hand, need to communicate when global pose information becomes available to keep the belief equivalent to the centralized fusion.

Our approach for a centralized-equivalent decoupling scheme utilizing elements of the scattering theory efficiently solves estimation problems with pairwise communication constraints. We show that this theory, typically used in physics, can also be used for pairwise state estimation, requiring communication only on meet-ups, i.e., when one agent is able to sense the other and locate itself relatively to it.

The analogy between waves traveling through media and estimation problems was partially covered in the previous work [3]. However, the focus was only on interpreting measurements as sections of a scattering medium, leading to remarkably fast covariance pre-integration by concatenating those sections into one medium. A key aspect of this work is to complete the previous theory by looking at the waves that travel through this medium, leading to reusable pre-computations of the mean values of the state variables. While the covariances and closed-loop transfer functions are described by the scattering matrix describing the medium, it turns out that the estimate means are the waves traveling through this medium forward in time. Moreover, the wave traveling backward in time is at least as important as the forward wave, since it carries information from future measurements to estimates of past states. This process in estimation is also known as *smoothing* and is related to the *adjoint variable*, which is, in fact, the backward wave and will be further described in III-B.

We show in this work that both, the covariances and the means can be computed with just a few steps using Scattering Theory. All the advantages like changing of initial conditions and concatenating measurements also carry over to the mean computations. The main limitation is that scattering theory was developed for linear systems. Therefore, we present methods to apply it on non-linear systems that paves the way for efficient distributed estimation in a multitude of realistic estimation problems.

Our contributions are:

- Extending previous work on linear systems [4] to cover efficient mean and covariance pre-computations for non-

[1]Eren Allak and Stephan Weiss are with the Department of Smart Systems Technologies in the Control of Networked Systems Group, Universität Klagenfurt, 9020 Klagenfurt, Austria {eren.allak, stephan.weiss}@ieee.org

[2]Axel Barrau is with Safran Tech, Groupe Safran, 78772 Magny Les Hameaux CEDEX, France axel.barrau@safrangroup.com

[3]Roland Jung is with the Karl Popper School on Networked Autonomous Aerial Vehicles, University of Klagenfurt, Austria (e-mail: roland.jung@ieee.org)

linear systems by the use of scattering theory (III-C).
- Centralized-equivalent estimates under asynchronous communication constraints for pairwise distributed state estimation on computationally constrained vehicles (IV).
- Comparison of the proposed method to a centralized implementation using real data (V).

II. Related Work

Before introducing the related work, the terms central and centralized-equivalent are explained. Centralized estimation refers to an estimation approach, where all measurements of each agent are processed in one entity leading to the statistically best possible beliefs. This approach requires constant communication between the central entity and all the agents. If the communication overhead is intractable, the next best estimation scheme is the centralized-equivalent estimation. Compared to the centralized version, the communication is reduced in some way. However, once the agents can communicate, a belief can be computed equivalent to the centralized version, i.e., a centralized-equivalent belief is achieved. In the past decades, different filter-based approaches for collective multiagent localization have been presented. Previous approaches can be roughly classified as (i) centralized-equivalent (e.g., [5], [6]), (ii) approximated (e.g., [7], [8]), (iii) covariance intersection based methods for unknown correlations (e.g., [9], [10]), (iv) optimizing correlations (e.g., [11]), and (v) graph-based methods (e.g., [12]).

The general challenge in all these approaches remains to *decouple* (statistically) the individual agents to relax the communication constraints, while at the same time *maintain* and account for coupling/cross-correlations between agents to achieve statistically optimal and consistent estimates. Current approaches apply different decoupling strategies at the cost of estimator consistency.

In i) [6], Kia et al. proposed a centralized-equivalent decoupled approach based on passing messages with correction terms after joint or global observations to the rest of the agents in a network.

The approximated decoupled filter approaches (ii) are a reasonable choice for real-world applications in terms of scalability regarding the number of involved agents, communication constraints, and accuracy with respect to centralized equivalent approaches, while not being consistent. Luft et al.'s approach presented in [7] requires communication only when agents meet ($\mathcal{O}(1)$) and the maintenance effort for the interdependencies scales with $\mathcal{O}(N)$ for N agents.

At high sensor rates, as it is the case for systems using an IMU as propagation sensor (in aided inertial systems the rate is typically between 100—1kHz), the maintenance effort was identified as a limitation for large swarms. Therefore, Jung and Weiss proposed in [8] the use of common correction buffers, allowing the maintenance cost to scale with $\mathcal{O}(1)$ with increasing number of known agents.

In contrast to these desired properties of [7], [8], one major disadvantage remains: directly or indirectly correlated agents that are not participating in the current observation between two other agents do inherit the information of this observation. Meaning that their beliefs experience no correction despite their (theoretical) coupling via cross correlation terms. The loss of accuracy due to this approximation is often favored over the reduced computational complexity in practice.

As for the works [9], [10] of iii), unknown correlations are only an issue if the inter-agent correlation terms are not maintained although the agents interacted in the past. Otherwise, inter-agent correlations can be assumed to be zero if they never met. Similarly in iv) [11], the inter-agent correlations are not maintained and must be inferred via optimization and consistency considerations. They can therefore not be completely recovered.

In our previous work, we used Scattering Theory (ST) [13], [14], [4] to perform covariance pre-integration in a single-agent multi-sensor setup [15] and further developed our findings [3] for single agent invariant filtering approaches [16], [17], [18] to enable statistically consistent covariance pre-integration.

This work achieves centralized-equivalent accuracy and consistency for a pair of agents in contrast to ii), iii) and iv), while still needing to communicate less than i) (we do not need to communicate global information immediately). Our approach takes past *private* observations of the other agent into account upon a *joint* observation. A measurement concerning only the local state of an agent, it is called *private*, and if it also concerns other agents' states, it is called a *joint* measurement.

We achieve centralized-equivalence in two steps by employing efficient pre-computations through the use of the scattering theory distributed on each agent as they move and measure independently. We require communication between the two agents to exchange pre-computations only at meet-ups when joint updates are performed (like, e.g., [7], [8]). In doing so, we are not concerned with one of the major disadvantages for centralized-equivalent approaches as those require either extensive bookkeeping or information distribution across the entire swarm of agents, e.g., [5], [6].

The rest of the paper is organized as follows: We first develop the necessary tools for a single agent in Sec. III. The covariance pre-computations derived in the form of *scattering matrices* are discussed in III-A, as they are needed for the mean pre-computations. In III-B, we derive *source vectors* used as pre-computation elements for state mean values. The novel extension to non-linear systems is shown in III-C. Then, in Sec. IV, we bring the elements of the single agents together to a pairwise estimation approach for the mean (IV-A) and covariance (IV-B) computation. The approach is evaluated on a dataset for differential wheel robots in Sec. V and in Sec. VI we draw the conclusions.

III. Single Agent Pre-Computations with Scattering Theory

The core aspect of this paper is to consider all private observations two agents may have had between their previous and current encounter in a statistically correct fashion. The

method should be equivalent to a fully centralized approach, but with reduced compute and communication requirements. The centralized version updates all beliefs of all agents whenever an agent receives an observation. Our approach is to continuously process all observations of the agents separately (distributed) as they move to pre-computation terms, and then exchange them with the other agent when they meet. Using these exchanged pre-computations and then applying the joint measurement is statistically equivalent to a centralized approach. For our EKF setup, the above involves precomputations for the covariance and the mean.

A. Covariance Pre-Computations as Scattering Matrices

A non-linear system with additive white Gaussian noise $\mathbf{n}_u, \mathbf{n}_y$, state $\mathbf{x}$ and measurement $\mathbf{y}$ in discrete time is given by

$$\mathbf{x}_{i+1} = f(\mathbf{x}_i, \mathbf{u}_i) + \mathbf{n}_{u,i}, \qquad \mathbf{n}_u \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_u), \tag{1}$$

$$\mathbf{y}_i = h(\mathbf{x}_i) + \mathbf{n}_{y,i}, \qquad \mathbf{n}_y \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_y). \tag{2}$$

A quick recapitulation of the previous work on covariance pre-integration [15] will introduce the basic concepts in scattering theory for covariance pre-computations, which will be relevant for this work. We show how many measurements are combined into one element, such that all measurements can be later applied in one step, for example to update with delayed measurements or to perform pairwise estimation. The star product [14] Eq. (4) is used to combine measurements for propagation and updates by their respective generators Eq. (5) and Eq. (6) with $\mathbf{F}$ and $\mathbf{H}$ being the Jacobian of the state dynamics and measurements, respectively. This leads to a scattering matrix Eq. (7), i.e., single agent covariance pre-computations, enabling the computation of covariances $\mathbf{P}$ considering all measurements for a given initial covariance, shown in Eq. (8). The subscripts t and m indicate *time propagation* and *measurement*.

$$\mathcal{S} = \mathcal{S}_1 \star \mathcal{S}_2 = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \star \begin{bmatrix} A & B \\ C & D \end{bmatrix} \tag{3}$$

$$= \begin{bmatrix} A(I-bC)^{-1}a & B + Ab(I-Cb)^{-1}D \\ c + dC(I-bC)^{-1}a & d(I-Cb)^{-1}D \end{bmatrix} \tag{4}$$

$$M_{t,i} = \begin{bmatrix} \mathbf{F}_i & \mathbf{\Sigma}_u \\ \mathbf{0} & \mathbf{F}_i^T \end{bmatrix} \quad \mathbf{F}_i = \frac{\partial f(\mathbf{x}, u)}{\partial \mathbf{x}}\Big|_{\mathbf{x}=\hat{\mathbf{x}}_i, u=u_i} \tag{5}$$

$$M_{m,i} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{H}_i^T \mathbf{\Sigma}_y^{-1} \mathbf{H}_i & \mathbf{I} \end{bmatrix} \quad \mathbf{H}_i = \frac{\partial h(\mathbf{x})}{\partial \mathbf{x}}\Big|_{\mathbf{x}=\hat{\mathbf{x}}_i} \tag{6}$$

$$\mathcal{S}^0_{i,N} = M_i \star M_{i+1} \star \ldots \star M_{N-1} \tag{7}$$

$$\mathcal{S}_{i,N} = \begin{bmatrix} \mathbf{I} & \mathbf{P}_{i,0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \star \mathcal{S}^0_{i,N} = \begin{bmatrix} \mathbf{\Phi}_{N,i} & \boldsymbol{\mathcal{P}}_{N,i} \\ -\boldsymbol{\mathcal{O}}_{N,i} & \mathbf{\Phi}^T_{N,i} \end{bmatrix} \tag{8}$$

$\mathcal{S}_{i,N}$ (later used for smoothed estimates) differs from $\mathcal{S}^0_{i,N}$ in that the initial conditions are already applied. The first equation Eq. (9) describes the closed loop transfer function that is required mainly to compute cross-covariances from innovations to states. Eq. (10) is the covariance of the estimation error after all measurements are applied. Eq. (11) is called the observability Gramian and is also at the same time the covariance of the adjoint variable, that is important to smoothing and will be introduced in Sec. III-B.

$$\mathbf{\Phi}_{N,i} = \mathbf{\Phi}_p(N,i) = \mathbf{F}_{p,N-1}\mathbf{F}_{p,N-2}\ldots\mathbf{F}_{p,i} \tag{9}$$

$$\boldsymbol{\mathcal{P}}_{N,i} = \mathbf{P}_{N|N-1} \tag{10}$$

$$\boldsymbol{\mathcal{O}}_{N,i} = \sum_{j=i}^{N-1} \mathbf{\Phi}_p^T(j,i)\mathbf{H}_j^T\mathbf{R}_{e,j}^{-1}\mathbf{H}_j\mathbf{\Phi}_p(j,i) \tag{11}$$

$$\mathbf{\Phi}_p(i,i) = \mathbf{I} \qquad \mathbf{F}_{p,i} = \mathbf{F}_i - \mathbf{K}_{p,i}\mathbf{H}_i \tag{12}$$

$$\mathbf{K}_{p,i} = \mathbf{F}_i\mathbf{P}_i\mathbf{H}_i^T\mathbf{R}_{e,i}^{-1} \qquad \mathbf{R}_{e,i} = \mathbf{H}_i\mathbf{P}_i\mathbf{H}_i^T + \mathbf{\Sigma}_y \tag{13}$$

B. Mean Pre-Computations as Source Vectors

The scattering theory has two integral components: the scattering medium and the waves that travel through the medium. In this section, the waves are connected to their counterparts in state estimation as the *estimate means* $\hat{\mathbf{x}}$ for the forward-in-time wave and the *adjoint variable* $\boldsymbol{\lambda}$ for the backward-in-time wave. The adjoint variable is required in the context of *smoothing* as described in the following. So far, the scattering medium was only characterized by the scattering matrix, which was enough for covariance pre-integration, but for the mean pre-computations (i.e., the waves) also the *source vectors* of the scattering medium are required. The adjoint variable is a part of the source vector, and thus can be pre-computed. Also, the following derivations are a preparation for the derivations in IV, since our approach is based on smoothing and uses adjoint variables as well.

Given a linear system with additive white noise Eq. (14), that is not related to Eq. (1-2), also given initial conditions $\{\mathbf{x}_0, \mathbf{P}_0\}$ and all measurements $\mathbf{y}_0 \ldots \mathbf{y}_N$ from time 0 to N, there are three different linear least mean squared (l.l.m.s.) estimates for the state $\mathbf{x}_i$ at time i: the *filtered* estimate $\hat{\mathbf{x}}_{i|i}$ which is considering $\mathbf{y}_0 \ldots \mathbf{y}_i$, the *predicted* estimate $\hat{\mathbf{x}}_{i|i-1}$ which is considering $\mathbf{y}_0 \ldots \mathbf{y}_{i-1}$, and the *smoothed* estimate $\hat{\mathbf{x}}_{i|N}$ which is considering all measurements $\mathbf{y}_0 \ldots \mathbf{y}_N$.

$$\mathbf{x}_{i+1} = \mathbf{F}_i\mathbf{x}_i + \mathbf{B}_i\mathbf{u}_i + \mathbf{n}_{u,i} \qquad \mathbf{y}_i = \mathbf{H}_i\mathbf{x}_i + \mathbf{n}_{y,i} \tag{14}$$

The estimates $\hat{\mathbf{x}}_{i|i-1}$ and $\hat{\mathbf{x}}_{i|i}$ can be computed with a Kalman Filter, that is also providing the innovations $\mathbf{e}_i = \mathbf{y}_i - \mathbf{H}_i\hat{\mathbf{x}}_{i|i-1}$ during the filtering process. These innovations can now be used to define the smoothed estimate and the adjoint variable $\boldsymbol{\lambda}_{i|N}$ by the innovations approach:

$$\hat{\mathbf{x}}_{i|N} = \sum_{j=0}^{N} \langle \mathbf{x}_i, \mathbf{e}_j \rangle \langle \mathbf{e}_j, \mathbf{e}_j \rangle^{-1} \mathbf{e}_j \tag{15}$$

$$= \hat{\mathbf{x}}_{i|i-1} + \sum_{j=i}^{N} \langle \mathbf{x}_i, \mathbf{e}_j \rangle \langle \mathbf{e}_j, \mathbf{e}_j \rangle^{-1} \mathbf{e}_j$$

$$= \hat{\mathbf{x}}_{i|i-1} + \mathbf{P}_{i|i-1} \sum_{j=i}^{N} \mathbf{\Phi}_p(j,i)^T \mathbf{H}_j^T \mathbf{R}_{e,j}^{-1} \mathbf{e}_j$$

$$= \hat{\mathbf{x}}_{i|i-1} + \mathbf{P}_{i|i-1}\boldsymbol{\lambda}_{i|N} \tag{16}$$

The covariance is denoted $\langle \cdot, \cdot \rangle$ and $\langle \mathbf{x}_i, \mathbf{e}_j \rangle = \mathbf{P}_{i|i-1}\mathbf{\Phi}_p(j,i)^T\mathbf{H}_j^T$ is just presented without derivation.

When measurements build up the scattering medium, they are attached one by one to the medium with their generator and the star product, as in defined Eq. (4-7). In the process,

every measurement also defines the source vectors of the scattering sections. In Eq. (17) the source vector for a propagation measurement $m_{t,i}$ and for an update measurement $m_{m,i}$ are shown. To combine source vectors of scattering sections $\{\mathcal{S}_1, \mathbf{s}_1\}$ and $\{\mathcal{S}_2, \mathbf{s}_2\}$ to one medium the dot-sum is used, as defined in Eq. (18). The scattering matrices are defined as in Eq. (3).

$$m_{t,i} = \begin{bmatrix} \mathbf{B}_i \mathbf{u}_i \\ \mathbf{0} \end{bmatrix} \qquad m_{m,i} = \begin{bmatrix} \mathbf{0} \\ \mathbf{H}_i^T \boldsymbol{\Sigma}_{y,i}^{-1} \mathbf{y}_i \end{bmatrix} \tag{17}$$

$$\begin{aligned} \mathbf{s} = \mathbf{s}_1 \bullet \mathbf{s}_2 &= \begin{bmatrix} r^+ \\ r^- \end{bmatrix} \bullet \begin{bmatrix} R^+ \\ R^- \end{bmatrix} \\ &= \begin{bmatrix} R^+ \\ r^- \end{bmatrix} + \left(\begin{bmatrix} I & b \\ 0 & d \end{bmatrix} \star \begin{bmatrix} A & 0 \\ C & I \end{bmatrix} \right) * \begin{bmatrix} r^+ \\ R^- \end{bmatrix} \\ &= \begin{bmatrix} R^+ \\ r^- \end{bmatrix} + \begin{bmatrix} A(I-bC)^{-1}(r^+ + bR^-) \\ d(I-Cb)^{-1}(R^- + Cr^+) \end{bmatrix} \end{aligned} \tag{18}$$

Finally, many measurements (i.e., their source vectors) $m_i \dots m_{N-1}$ are combined, as in Eq. (19). These are the single agent mean pre-computations for linear systems. After adding the initial conditions, as in Eq. (20), the resulting source vector $\mathbf{s}_{N,i}$ solves two estimation problems simultaneously: The estimation at time N as $\hat{\mathbf{x}}_{N|N-1}$ and the adjoint variable for smoothing at time i as $\boldsymbol{\lambda}_{i|N}$, given all measurements $y_i \dots y_N$ and initial conditions $\{\mathbf{x}_{i,0}, \mathbf{P}_{i,0}\}$.

$$\mathbf{s}^0_{N,i} = m_i \bullet m_{i+1} \dots \bullet m_{N-1} \tag{19}$$

$$\mathbf{s}^b = \begin{bmatrix} \mathbf{x}_{i,0} \\ \mathbf{0} \end{bmatrix} \qquad \mathbf{s}_{N,i} = \mathbf{s}^b \bullet \mathbf{s}^0_{N,i} = \begin{bmatrix} \hat{\mathbf{x}}_{N|N-1} \\ \boldsymbol{\lambda}_{i|N} \end{bmatrix} \tag{20}$$

C. Extension to Non-Linear Systems

The Extended Kalman Filter (EKF) is a special case of the linearized Kalman Filter, where the linearization points are taken as the last state estimates. On the same linearized system that the EKF is applied, also the mean computations of the scattering theory can be applied. This only requires certain pre-computations (Eq. (7) and Eq. (19)) to be done, while the EKF is applied to the measurements for the first time. This results in one step re-computations of the EKF means and adjoint variable for new initial conditions (Eq. (25)). In the following derivations, all measurements are processed once with an EKF and therefore all linearization points $\hat{\mathbf{x}}^{\text{lin}}_{i|i}$ and $\hat{\mathbf{x}}^{\text{lin}}_{i|i-1}$ are available. Linearizing is done at a propagated or filtered estimate, $\hat{\mathbf{x}}^{\text{lin}}_i = \hat{\mathbf{x}}^{\text{lin}}_{i|i-1}$ or $\hat{\mathbf{x}}^{\text{lin}}_i = \hat{\mathbf{x}}^{\text{lin}}_{i|i}$, respectively. The linear system for the EKF at the linearization point $\hat{\mathbf{x}}^{\text{lin}}_i$ at time i is described by Eq. (21-22):

$$\begin{aligned} f(\mathbf{x}_i, \mathbf{u}_i) &\approx f\left(\hat{\mathbf{x}}^{\text{lin}}_i, \mathbf{u}_i\right) + \left.\frac{\partial f(\mathbf{x}, \cdot)}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}^{\text{lin}}_i} (\mathbf{x}_i - \hat{\mathbf{x}}^{\text{lin}}_i) \\ \mathbf{x}_{i+1} &\approx \hat{\mathbf{x}}^{\text{lin}}_{i+1} + \mathbf{F}_i \Delta\mathbf{x}_i \\ \Delta\mathbf{x}_{i+1} = \mathbf{x}_{i+1} - \hat{\mathbf{x}}^{\text{lin}}_{i+1} &\approx \mathbf{F}_i \Delta\mathbf{x}_i \end{aligned} \tag{21}$$

$$\begin{aligned} h(\mathbf{x}_i, \cdot) &\approx h\left(\hat{\mathbf{x}}^{\text{lin}}_i\right) + \left.\frac{\partial h(\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}^{\text{lin}}_i} (\mathbf{x}_i - \hat{\mathbf{x}}^{\text{lin}}_i) \\ \mathbf{y}_i &\approx h\left(\hat{\mathbf{x}}^{\text{lin}}_i\right) + \mathbf{H}_i \Delta\mathbf{x}_i \\ \mathbf{y}_i - h\left(\hat{\mathbf{x}}^{\text{lin}}_i\right) &\approx \mathbf{H}_i \Delta\mathbf{x}_i \end{aligned} \tag{22}$$

An update to the linearization point is then described as

$$\begin{aligned} \hat{\mathbf{x}}^{\text{lin}}_{i|i} &= \hat{\mathbf{x}}^{\text{lin}}_{i|i-1} + \mathbf{K}_i(\mathbf{y}_i - h(\hat{\mathbf{x}}^{\text{lin}}_{i|i-1})) = \hat{\mathbf{x}}^{\text{lin}}_{i|i-1} + \delta\mathbf{x}_i \\ \mathbf{x}_i - \Delta\mathbf{x}_{i|i} &= \mathbf{x}_i - \Delta\mathbf{x}_{i|i-1} + \delta\mathbf{x}_i \\ \Delta\mathbf{x}_{i|i} &= \Delta\mathbf{x}_{i|i-1} - \delta\mathbf{x}_i \end{aligned} \tag{23}$$

On the linearized system described by Eq. (21-23) the mean computations of the scattering theory, i.e. Eq. (19), can now be applied with slight adaptations to the update measurement source vector $m_{m,i}$:

$$m_{m,i} = \begin{bmatrix} -\delta\mathbf{x}_i \\ \mathbf{H}_i^T \boldsymbol{\Sigma}_{y,i}^{-1}(\mathbf{y}_i - h(\hat{\mathbf{x}}^{\text{lin}}_i)) \end{bmatrix} \tag{24}$$

Given new initial conditions $\{\mathbf{x}^{\text{new}}_{i,0}, \mathbf{P}^{\text{new}}_{i,0}\}$ with $\Delta\mathbf{x}_{i,0} = \mathbf{x}^{\text{new}}_{i,0} - \mathbf{x}_{i,0}$ the smoothed estimates as well as EKF mean estimates can be computed in one step:

$$\mathbf{s}^b = \begin{bmatrix} \Delta\mathbf{x}_{i,0} \\ \mathbf{0} \end{bmatrix} \qquad \mathbf{s}_{N,i} = \mathbf{s}^b \bullet \mathbf{s}^0_{N,i} = \begin{bmatrix} \Delta\mathbf{x}_N \\ \Delta\boldsymbol{\lambda}_{i|N} \end{bmatrix} \tag{25}$$

$$\hat{\mathbf{x}}^{\text{new}}_{N|N} = \hat{\mathbf{x}}^{\text{lin}}_{N|N} + \Delta\mathbf{x}_N \qquad \text{(for N as update)} \tag{26}$$

$$\hat{\mathbf{x}}^{\text{new}}_{N|N-1} = \hat{\mathbf{x}}^{\text{lin}}_{N|N-1} + \Delta\mathbf{x}_N \qquad \text{(for N as propagation)} \tag{27}$$

$$\hat{\mathbf{x}}^{\text{new}}_{i|N} = \hat{\mathbf{x}}^{\text{new}}_{i,0} + \mathbf{P}^{\text{new}}_{i,0} \Delta\boldsymbol{\lambda}_{i|N} \tag{28}$$

IV. Centralized-Equivalent Pairwise State Estimation with Scattering Theory

In the previous section, we described how the two agents generate pre-computations for means and covariances described by Eq.(19) and Eq. (7), respectively. Assume agents A and B are initially correlated at time i and then every agent performs a standard EKF to update its state, assume $\mathbf{x}_A$, with private measurements, say $\mathbf{y}^A_i \dots \mathbf{y}^A_N$, but at the same time also builds up the scattering matrix $\mathcal{S}_A$ and the source vectors $\mathbf{s}_A$ from all its private measurements. Once agent A meets agent B again, they share $\{\mathcal{S}_A, \mathbf{s}_A\}$ and $\{\mathcal{S}_B, \mathbf{s}_B\}$ with each other and update their own state in just two steps with the private measurements of the other agent as changes of their own initial conditions.

A. Centralized-Equivalent Mean Computations

Incorporating all information of agent A to agent B is done by smoothing agent B's state at the initial time i with all of agent A's private measurements to get $\hat{\mathbf{x}}^B_{i|N(A)} = \hat{\mathbf{x}}^B(i, \mathbf{y}^A_i \dots \mathbf{y}^A_N)$ as a first step. Then all of B's own private measurements $\mathbf{y}^B_i \dots \mathbf{y}^B_N$ are applied on top of that changed initial condition as a second step. For better legibility and understanding, we present the following derivations using the regular state notation. For the error-state notation in non-linear systems, the matrices are replaced by their corresponding Jacobians according to Eq. (21-22). The effect of this piecewise linearized representation (linearized at each EKF step) has minimal impact on the performance (c.f. results on real data in Sec. V), yet allows the use of our proposed scattering theory repertoire for fast (re-)computations. Deriving $\hat{\mathbf{x}}^B_{i|N(A)}$ and $\mathbf{P}^B_{i|N(A)}$, by applying the innovations $\mathbf{e}$ of A on a joint state vector $\mathbf{z}_i$:

$$\mathbf{z}_i = \begin{bmatrix}\mathbf{x}_i^A \\ \mathbf{x}_i^B\end{bmatrix} \qquad \mathbf{e}_j = \mathbf{H}_{A,j}\tilde{\mathbf{x}}_j^A + \mathbf{n}_{y,j}^A$$

$$\begin{bmatrix}\hat{\mathbf{x}}_{i|N(A)}^A \\ \hat{\mathbf{x}}_{i|N(A)}^B\end{bmatrix} = \begin{bmatrix}\hat{\mathbf{x}}_i^A \\ \hat{\mathbf{x}}_i^B\end{bmatrix} + \sum_{j=i}^{N}\langle \mathbf{z}_i, \mathbf{e}_j\rangle\langle \mathbf{e}_j, \mathbf{e}_j\rangle^{-1}\mathbf{e}_j \tag{29}$$

Computing the covariances from states to innovations, and noting that $\hat{\mathbf{x}}_i \perp \tilde{\mathbf{x}}_i$ (so $\langle \hat{\mathbf{x}}_i, \tilde{\mathbf{x}}_i\rangle = 0$) and $\mathbf{n}_{y,j} \perp \{\tilde{\mathbf{x}}_i, \hat{\mathbf{x}}_i\}$ for $j > i$ by definition:

$$\langle \mathbf{z}_i, \mathbf{e}_j\rangle = \langle \begin{bmatrix}\mathbf{x}_i^A \\ \mathbf{x}_i^B\end{bmatrix}, \mathbf{e}_j\rangle \tag{30}$$

$$\begin{aligned}\langle \mathbf{x}_i^A, \mathbf{e}_j\rangle &= \langle \mathbf{x}_i^A, \mathbf{H}_{A,j}\tilde{\mathbf{x}}_j^A + \mathbf{n}_{y,j}^A\rangle \\ &= \langle \hat{\mathbf{x}}_i^A + \tilde{\mathbf{x}}_i^A, \mathbf{H}_{A,j}\tilde{\mathbf{x}}_j^A + \mathbf{n}_{y,j}^A\rangle \\ &= \langle \hat{\mathbf{x}}_i^A, \mathbf{H}_{A,j}\tilde{\mathbf{x}}_j^A\rangle + \langle \hat{\mathbf{x}}_i^A, \mathbf{n}_{y,j}^A\rangle + \dots \\ &\quad \langle \tilde{\mathbf{x}}_i^A, \mathbf{H}_{A,j}\tilde{\mathbf{x}}_j^A\rangle + \langle \tilde{\mathbf{x}}_i^A, \mathbf{n}_{y,j}^A\rangle \\ &= 0 + 0 + \langle \tilde{\mathbf{x}}_i^A, \mathbf{H}_{A,j}\tilde{\mathbf{x}}_j^A\rangle + 0\end{aligned} \tag{31}$$

$$\begin{aligned}&= \langle \tilde{\mathbf{x}}_i^A, \mathbf{H}_{A,j}\mathbf{\Phi}_{p,A}(j,i)\tilde{\mathbf{x}}_i^A\rangle \\ &= \langle \tilde{\mathbf{x}}_i^A, \tilde{\mathbf{x}}_i^A\rangle\mathbf{\Phi}_{p,A}(j,i)^T\mathbf{H}_{A,j}^T \\ &= \mathbf{P}_{A,i}\mathbf{\Phi}_{p,A}(j,i)^T\mathbf{H}_{A,j}^T\end{aligned} \tag{32}$$

$$\langle \mathbf{x}_i^B, \mathbf{e}_j\rangle = \langle \tilde{\mathbf{x}}_i^B, \tilde{\mathbf{x}}_i^A\rangle\mathbf{\Phi}_{p,A}(j,i)^T\mathbf{H}_{A,j}^T \tag{33}$$

$$= \mathbf{P}_{BA,i}\mathbf{\Phi}_{p,A}(j,i)^T\mathbf{H}_{A,j}^T \tag{34}$$

Now the smoothed estimate is:

$$\begin{aligned}\hat{\mathbf{x}}_{i|N(A)}^B &= \hat{\mathbf{x}}_i^B + \sum_{j=i}^{N}\langle \mathbf{x}_i^B, \mathbf{e}_j\rangle\langle \mathbf{e}_j, \mathbf{e}_j\rangle^{-1}\mathbf{e}_j \\ &= \hat{\mathbf{x}}_i^B + \mathbf{P}_{BA,i}\sum_{j=i}^{N}\mathbf{\Phi}_{p,A}(j,i)^T\mathbf{H}_{A,j}^T\langle \mathbf{e}_j, \mathbf{e}_j\rangle^{-1}\mathbf{e}_j \\ &= \hat{\mathbf{x}}_i^B + \mathbf{P}_{BA,i}\boldsymbol{\lambda}_{i|N}^A\end{aligned} \tag{35}$$

Given the initial covariance $\mathbf{P}_{BA,i}$ at start time i, it can be seen in Eq. (35) that the adjoint variable of A, which was computed by A as part of $\mathbf{s}_A$, can be directly used to smooth B with A's private measurements. This is a strong contrast to previous distributed approaches, that assume/simplify that private measurements do not change the state of other agents [7], [8]. The corresponding covariance $\mathbf{P}_{i|N(A)}^B$ of the smoothed estimate error is then:

$$\begin{aligned}\tilde{\mathbf{x}}_{i|N(A)}^B &= \mathbf{x}_i^B - \hat{\mathbf{x}}_{i|N(A)}^B \\ &= \mathbf{x}_i^B - (\hat{\mathbf{x}}_i^B + \mathbf{P}_{BA,i}\boldsymbol{\lambda}_{i|N}^A) \\ &= \tilde{\mathbf{x}}_i^B - \mathbf{P}_{BA,i}\boldsymbol{\lambda}_{i|N}^A\end{aligned} \tag{36}$$

$$\langle \tilde{\mathbf{x}}_{i|N(A)}^B, \tilde{\mathbf{x}}_{i|N(A)}^B\rangle = \mathbf{P}_{B,i} + \mathbf{P}_{BA,i}\langle \boldsymbol{\lambda}_{i|N}^A, \boldsymbol{\lambda}_{i|N}^A\rangle\mathbf{P}_{BA,i}^T$$

$$\boldsymbol{\lambda}_{i|N}^A = \sum_{j=i}^{N}\mathbf{\Phi}_{p,A}(j,i)^T\mathbf{H}_{A,j}^T\langle \mathbf{e}_j, \mathbf{e}_j\rangle^{-1}\mathbf{e}_j \tag{37}$$

$$\begin{aligned}\langle \boldsymbol{\lambda}_{i|N}^A, \boldsymbol{\lambda}_{i|N}^A\rangle &= \sum_{j=i}^{N}\mathbf{\Phi}_{p,A}(j,i)^T\mathbf{H}_{A,j}^T\langle \mathbf{e}_j, \mathbf{e}_j\rangle^{-1}\cdot\dots \\ &\quad \langle \mathbf{e}_j, \mathbf{e}_j\rangle\langle \mathbf{e}_j, \mathbf{e}_j\rangle^{-1,T}\mathbf{H}_{A,j}\mathbf{\Phi}_{p,A}(j,i) \\ &= \sum_{j=i}^{N}\mathbf{\Phi}_{p,A}(j,i)^T\mathbf{H}_{A,j}^T\langle \mathbf{e}_j, \mathbf{e}_j\rangle^{-1}\mathbf{H}_{A,j}\mathbf{\Phi}_{p,A}(j,i) \\ &= \boldsymbol{\mathcal{O}}_{A,i|N}\end{aligned} \tag{38}$$

$$\mathbf{P}_{i|N(A)}^B = \mathbf{P}_{B,i} + \mathbf{P}_{BA,i}\boldsymbol{\mathcal{O}}_{A,i|N}\mathbf{P}_{BA,i}^T \tag{39}$$

$\boldsymbol{\mathcal{O}}_{A,i|N}$ is the observability gramian from the scattering matrix $\mathcal{S}_A$ with already included initial conditions of A that is passed from A to B. For the covariance, state-of-the-art distributed algorithms assume that there is no change for the passive agents, but the private measurements of the active agent do affect the mean and the covariance c.f. Eq. (35) and Eq. (39). Our approach takes these changes into account with minimal compute and communication requirements.

Now the private measurements of B can be applied on the smoothed initial estimate, leading to a final estimate $\{\hat{\mathbf{x}}_{N|N(A,B)}^B, \mathbf{P}_{N|N(A,B)}^B\}$ at time N that is equivalent to the final estimate of a joint centralized system, although all computations were done in a distributed fashion with a single encounter (i.e., information exchange) of the two agents. Note that $\{\mathcal{S}_{i,N}^{0,B}, \mathbf{s}_{i,N}^{0,B}\}$ were used, which do not include the initial conditions of B, since they are replaced by the smoothed initial estimate. The dots are entries that don't need to be computed and can be omitted.

$$\begin{bmatrix}\mathbf{I} & \mathbf{P}_{i|N(A)}^B \\ \mathbf{0} & \mathbf{I}\end{bmatrix} \star \mathcal{S}_{i,N}^{0,B} = \begin{bmatrix}\cdot & \mathbf{P}_{N|N(A,B)}^B \\ \cdot & \cdot\end{bmatrix} \tag{40}$$

$$\begin{bmatrix}\hat{\mathbf{x}}_{i|N(A)}^B \\ \mathbf{0}\end{bmatrix} \bullet \mathbf{s}_{i,N}^{0,B} = \begin{bmatrix}\hat{\mathbf{x}}_{N|N(A,B)}^B \\ \cdot\end{bmatrix} \tag{41}$$

B. Centralized-Equivalent Covariance Computations

The computation of the covariance of the smoothed initial state and the final state of the passive agent B in Eq. (39) and Eq. (40) was derived explicitly to show the contributions of the active agent's private measurements on the passive states. But there is a more direct way, again, using scattering theory, computing the *complete* joint covariance of A and B having processed all private measurements at the final time N.

To find the cross covariances, we can convert a smoothing problem to a filtering problem by extending the state and fixing the time. For a fixed k and $k < i$ we have the extended state and system with initial conditions at time k:

$$\mathbf{z}_{i,k} = \begin{bmatrix} \mathbf{x}_i \\ \mathbf{x}_k \end{bmatrix} \qquad \mathbf{z}_{i+1} = \boldsymbol{\mathcal{F}}_i \mathbf{z}_i + \boldsymbol{\mathcal{B}}_i \mathbf{u}_i + \boldsymbol{\mathcal{G}}_i \mathbf{n}_{u,i}$$

$$\mathbf{y}_i = \boldsymbol{\mathcal{H}}_i \mathbf{z}_i + \mathbf{n}_{y,i}$$

$$\boldsymbol{\mathcal{F}}_i = \begin{bmatrix} \mathbf{F}_i & 0 \\ 0 & I \end{bmatrix} \quad \boldsymbol{\mathcal{G}}_i = \begin{bmatrix} \mathbf{G}_i \\ 0 \end{bmatrix} \quad \boldsymbol{\mathcal{H}}_i = \begin{bmatrix} \mathbf{H}_i & 0 \end{bmatrix}$$

$$\hat{\mathbf{z}}_{k,i} = \begin{bmatrix} \hat{\mathbf{x}}_k \\ \hat{\mathbf{x}}_k \end{bmatrix} \quad \boldsymbol{\mathcal{P}}_{z,k} = \begin{bmatrix} P_k & P_k \\ P_k & P_k \end{bmatrix} = \bar{\boldsymbol{P}}_k$$

Noting that $\hat{\mathbf{x}}_{k|i}$ is the smoothed estimate of $\mathbf{x}_k$, applying all measurements $\mathbf{y}_k \dots \mathbf{y}_i$ gives the estimates and covariances:

$$\hat{\mathbf{z}}_{i,k} = \begin{bmatrix} \hat{\mathbf{x}}_{i|i} \\ \hat{\mathbf{x}}_{k|i} \end{bmatrix} \qquad \boldsymbol{\mathcal{P}}_{z,i} = \begin{bmatrix} P_i & P_{12,ik} \\ P_{21,ik} & P_{k|i} \end{bmatrix}$$

$\boldsymbol{\mathcal{P}}_{z,i}$ can be computed directly with scattering matrices, because scattering matrices and the joint covariances with interchanged columns satisfy the same Riccati equations [14], [4]. If the columns of $\boldsymbol{\mathcal{P}}_{z,i}$ are interchanged, we get the scattering matrix of the single state system with the initial condition of $\bar{\boldsymbol{P}}_k$(i.e., $\bar{\boldsymbol{P}}_k \star \mathcal{S}^0_{k,N}$):

$$\boldsymbol{\mathcal{J}} = \begin{bmatrix} 0 & I \\ I & 0 \end{bmatrix} \qquad \boldsymbol{\mathcal{P}}_{z,N} = (\bar{\boldsymbol{P}}_k \star \mathcal{S}^0_{k,N}) \boldsymbol{\mathcal{J}}$$

$$\begin{bmatrix} P_{12,Nk} & P_N \\ P_{k|N} & P_{21,Nk} \end{bmatrix} = \bar{\boldsymbol{P}}_k \star \begin{bmatrix} \boldsymbol{\Phi}^0_{k,N} & \boldsymbol{\mathcal{P}}^0_{k,N} \\ -\boldsymbol{\mathcal{O}}^0_{k,N} & \boldsymbol{\Phi}^{0,T}_{k,N} \end{bmatrix}$$

Given agents A and B with initial covariance $\bar{\mathbf{P}}_k$ and generated $\mathcal{S}^{A,0}_{k,N}$ and $\mathcal{S}^{B,0}_{k,N}$ by their private measurements, we can compute a centralized-equivalent covariance before the joint update as:

$$\mathcal{S}_A = (\bar{\mathbf{P}}_k \boldsymbol{\mathcal{J}}) \star \mathcal{S}^{A,0}_{k,N} \qquad \boldsymbol{\mathcal{P}}_{k|N(A)} = \mathcal{S}_A \boldsymbol{\mathcal{J}}$$

$$\boldsymbol{\mathcal{P}}_{k|N(A),\mathrm{p}} = \boldsymbol{\mathcal{J}} \boldsymbol{\mathcal{P}}_{k|N(A)} \boldsymbol{\mathcal{J}}$$

$$\mathcal{S}_{B,p} = (\boldsymbol{\mathcal{P}}_{k|N(A),\mathrm{p}} \boldsymbol{\mathcal{J}}) \star \mathcal{S}^{B,0}_{k,N} \qquad \boldsymbol{\mathcal{P}}_{k|N(A,B),\mathrm{p}} = \mathcal{S}_{B,p} \boldsymbol{\mathcal{J}}$$

$$\boldsymbol{\mathcal{P}}_{k|N(A,B)} = \boldsymbol{\mathcal{J}} \boldsymbol{\mathcal{P}}_{k|N(A,B),\mathrm{p}} \boldsymbol{\mathcal{J}} \tag{42}$$

Permutations, i.e. $\boldsymbol{\mathcal{P}}_{\dots,p}$, for the input covariances are necessary, because agent A and B change the role of active and passive, and then the order in the joint system is also interchanged. As a last step, the joint measurement can now be applied by both agents as a standard EKF update on the joint system with centralized-equivalent covariance $\boldsymbol{\mathcal{P}}_{k|N(A,B)}$.

V. Results

We have used UTIAS Multi-Robot Cooperative Localization and Mapping Dataset [19] to evaluate our approach on real data. Differential drive robots move indoors, logging odometry data and range-bearing measurements of known landmarks and other agents when they meet. We considered robots 1 and 2 from the first dataset with their trajectories shown in Fig. 1 and Fig. 2. The sampling rate of the odometry was 25 Hz, and the trajectory duration was 375 sec. While landmark measurements updated the state, range-bearing measurements of other agents can jointly update the state of both agents.

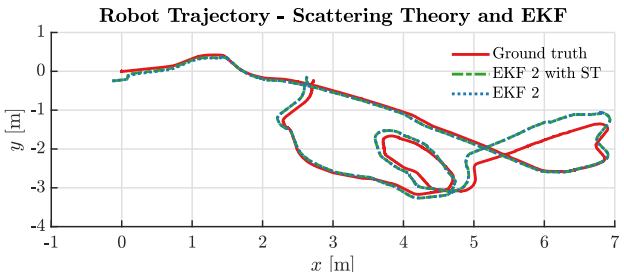


Fig. 1. The trajectory of the first robot is shown in red, the slow EKF computations are shown in blue (EKF 2) and the proposed fast EKF computations are shown in green (EKF 2 with ST). When changing the initial conditions and recomputing the EKF estimates, we achieve the same estimates (i.e., path overlap), indicating that the proposed method can replace computationally intensive re-computations while leading to the same results.

A. Centralized-Equivalent Pairwise Estimation with Ground Robots and Range-Bearing Measurements

We use the proposed methods to perform centralized-equivalent estimation for two robots with asynchronous pairwise communication constraints and compare the results to a fully centralized implementation. Robot one and robot two from the first dataset estimate their state with odometry and known landmark measurements while building up scattering matrices and source vectors. When they meet, they exchange these pre-computed elements and can reproduce a centralized-equivalent joint system update, as if they were connected and exchanging information during the whole time. Fig. 3 shows the estimation error against ground truth, while Fig. 4 shows the error between the estimation methods. There is an order of magnitude lower difference between the two approaches compared to the error of the estimations with respect to ground truth. However, our approach only needs sporadic communication between the robots compared to the fully centralized EKF implementation. There is a maximal error of 2.25 cm and for the heading 0.7 degrees between the two methods (due to space limitations the heading error plot is not depicted).

And finally, in Fig. 5, we compare our approach against Luft et al. [7] in terms of the joint system belief. Their method has the same communication constraints but different distributed covariance pre-computations than our approach. They make certain approximations, and the resulting joint belief is therefore not centralized-equivalent anymore during joint updates. The employed Kullback-Leibler (KL) divergence quantifies the difference between two probability distributions, and should therefore be close to zero if the beliefs are identical. The KL divergence between the joint system and our proposed method (shown in light blue) is close to zero overall, while the values are an order of magnitude higher for the method of Luft et al. [7](shown in light red). This indicates that the proposed method provides indeed centralized-equivalent beliefs.

B. Computation Times

The presented computation times correspond to the same experiment as in the previous section. To describe our

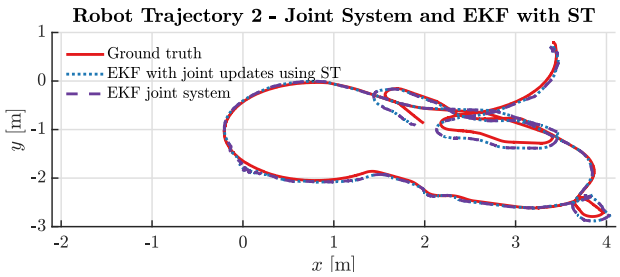


Fig. 2. The trajectory of the second robot (red), the joint estimates (purple) and the proposed centralized-equivalent approach (blue). The estimation behavior is the same, although our approach is restricted in communication, indicating that the proposed method can replace computationally intensive re-computations while leading to the same results.

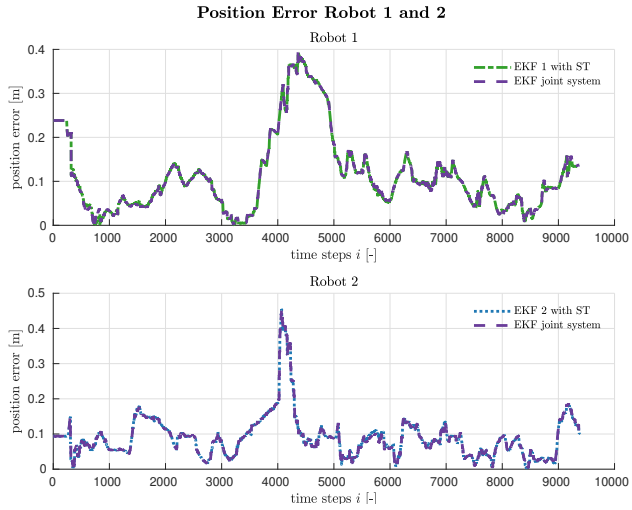


Fig. 3. The estimation error for the position of the joint system and the presented approach is shown. The joint system (purple) and the estimates of the agents (green and blue for robot 1 and 2, respectively). The estimation errors are very close for both approaches, showing that the presented work can achieve the same estimation performance, while performing computations efficiently and only when the agents meet.

approach's computational efficiency, we need to compare it to the computations required to process all measurements of both agents in a joint system at the moment they meet again, which is done in Fig. 6. The first plot shows the computation times for propagation of one agent (0.15 ms) in red and the overhead in each propagation step (0.1 ms) to build the scattering matrices for our approach in blue. The second plot shows how fast our approach computes the joint covariances for joint updates. The longer the agents did not see each other, the more measurements are processed, and therefore more processing time is necessary (maximum of 0.069 s at $t = 166$). On the other hand, if the measurements would be all processed by a joint system only once the agents meet *and not while they are moving*, then the computation takes longer, as shown in the third plot in red (maximum of 1.0 s at $t = 166$). Note that the agents can not communicate until they meet, i.e., can not process the other agent's measurements while moving. The relative computation time of our approach compared to the joint system computation is shown in the last plot, especially when the agents do not meet for long

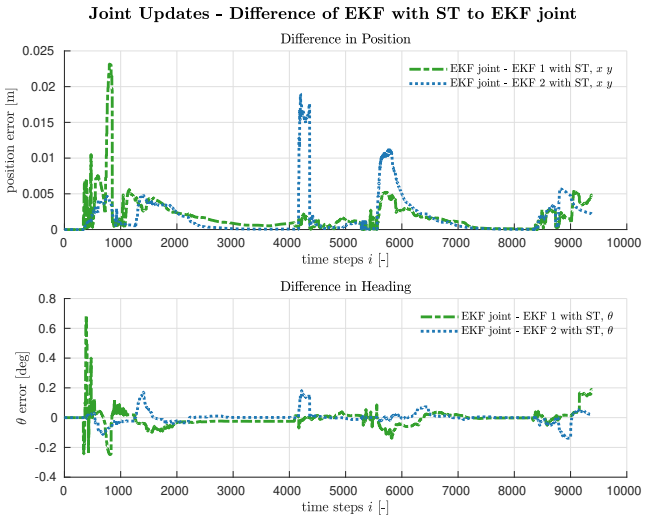


Fig. 4. The plots show the difference in the estimates of the centralized estimator and the two agents (agent 1 in green, agent 2 in blue) performing joint updates with the presented centralized-equivalent approach. In the top plot the difference in x and y is computed as a norm, and in the bottom plot the difference in heading is shown. The error introduced by the presented approach is about an order of magnitude lower than the estimation error itself, comparing the spike at 2.25 cm against a maximum error of 40 cm in Fig. 3.

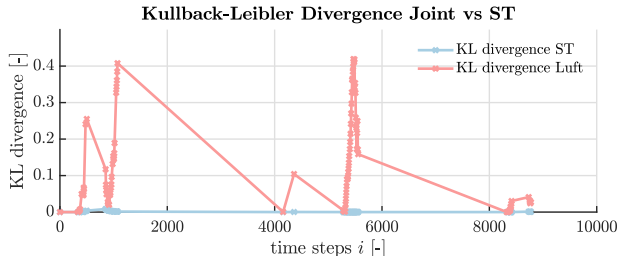


Fig. 5. The symmetric Kullback-Leibler Divergence (KL) quantifies the difference of two beliefs, in our case considering normal distributions. We plotted the KL divergence between the belief of the joint system and the belief of the two agents when they meet and perform a joint update using scattering theory in light blue. As a comparison the KL divergence of an approximating approach of Luft et al. [7], which has the same communication constraints, is also shown in light red. While the difference for our approach is overall very low, the approximations of Luft et al. lead to an order of magnitude higher values, indicating that the proposed method is indeed centralized-equivalent.

times our approach becomes more efficient (6.8% at $t = 166$ compared to the joint system).

VI. Conclusions

We presented a distributed but centralized-equivalent state estimation approach for two robots that have asynchronous pairwise communications constraints. The approach is based on the scattering theory and a fruitful analogy of waves traveling trough media was made. In this analogy we first derived the necessary and novel methods for distributed mean pre-computations on non-linear systems and then applied it to pairwise estimation. The combination of many measurements and the ability to change initial conditions in one step enabled us to smooth the state of agents with the measurements of other agents only when they meet, not requiring any

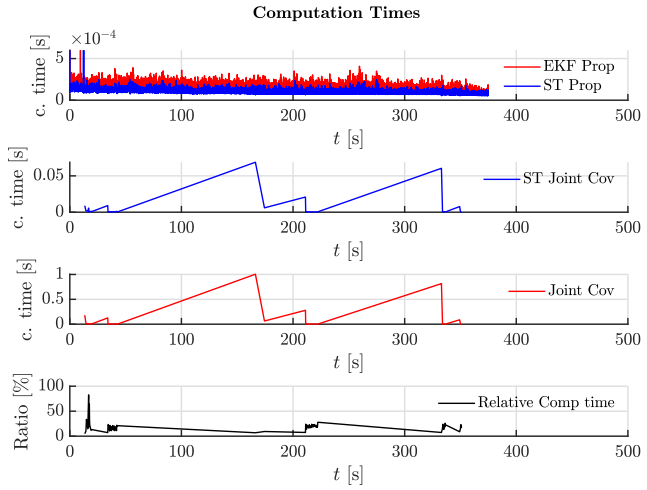


Fig. 6. We show the computational efficiency of our approach compared to a joint system computation. Agents can only exchange measurements when they meet and therefore all measurements would need to processed either with our faster approach (see second plot) or by forming a joint system and reprocessing all measurements on meetup with a joint system (see third plot). Our approach induces an overhead (see blue line in first plot), but reduces the overall computation time on meetup drastically (see relative comparison last plot).

constant communication channel to be open yet being able to reconstruct all statistical information from observations the other agent had since the previous meeting. Our novelty is that we extended the previous work on Collaborative State Estimation with constrained and pairwise communication to be statistically truly centralized-equivalent for two robots. Furthermore, we showed that the benefits of pairwise updates are maintained while requiring only very few computations, because measurements can be readily applied with the help of scattering matrices and source vectors. We evaluated our algorithm on real data and showed that the difference to the estimation in a centralized system is an order of magnitude smaller than the actual estimation error of both systems.

References

[1] M. Karrer, M. Agarwal, M. Kamel, R. Siegwart, and M. Chli, "Collaborative 6dof relative pose estimation for two uavs with overlapping fields of view," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 6688–6693.

[2] D. Gallup, J.-M. Frahm, P. Mordohai, and M. Pollefeys, "Variable baseline/resolution stereo," in *2008 IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8.

[3] E. Allak, R. Jung, and S. Weiss, "Covariance pre-integration for delayed measurements in multi-sensor fusion," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 6642–6649.

[4] G. Verghese, B. Friedlander, and T. Kailath, "Scattering theory and linear least-squares estimation, part iii: The estimates," *IEEE Transactions on Automatic Control*, vol. 25, no. 4, pp. 794–802, 1980.

[5] S. I. Roumeliotis and G. A. Bekey, "Distributed multirobot localization," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 781–795, 2002, zSCC: 0000819.

[6] S. S. Kia, S. F. Rounds, and S. Martínez, "A centralized-equivalent decentralized implementation of Extended Kalman Filters for cooperative localization," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sep. 2014, pp. 3761–3766, iSSN: 2153-0866.

[7] L. Luft, T. Schubert, S. I. Roumeliotis, and W. Burgard, "Recursive decentralized localization for multi-robot systems with asynchronous pairwise communication," *The International Journal of Robotics Research*, p. 0278364918760698, Mar. 2018. [Online]. Available: https://doi.org/10.1177/0278364918760698

[8] R. Jung and S. M. Weiss, "Scalable Recursive Distributed Collaborative State Estimation for Aided Inertial Navigation," in -, 2021 IEEE International Conference on Robotics and Automation (ICRA), 2021, pp. –. [Online]. Available: https://www.aau.at/wp-content/uploads/2021/02/RJung-CSE-ICRA21.pdf

[9] L. C. Carrillo-Arce, E. D. Nerurkar, J. L. Gordillo, and S. I. Roumeliotis, "Decentralized multi-robot cooperative localization using covariance intersection," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nov. 2013, pp. 1412–1417, iSSN: 2153-0866.

[10] T. R. Wanasinghe, G. K. I. Mann, and R. G. Gosine, "Decentralized Cooperative Localization for Heterogeneous Multi-robot System Using Split Covariance Intersection Filter," in *2014 Canadian Conference on Computer and Robot Vision*, May 2014, pp. 167–174.

[11] J. Zhu and S. S. Kia, "Cooperative Localization Under Limited Connectivity," *IEEE Transactions on Robotics*, vol. 35, no. 6, pp. 1523–1530, Dec. 2019, conference Name: IEEE Transactions on Robotics.

[12] V. Indelman, P. Gurfil, E. Rivlin, and H. Rotstein, "Graph-based distributed cooperative navigation," in *2011 IEEE International Conference on Robotics and Automation*, May 2011, pp. 4786–4791, iSSN: 1050-4729.

[13] L. Ljung, T. Kailath, and B. Friedlander, "Scattering theory and linear least squares estimation—part i: Continuous-time problems," *Proceedings of the IEEE*, vol. 64, no. 1, pp. 131–139, 1976.

[14] B. Friedlander, T. Kailath, and L. Ljung, "Scattering theory and linear least squares estimation—ii. discrete-time problems," *Journal of the Franklin Institute*, vol. 301, no. 1-2, pp. 71–82, 1976.

[15] E. Allak, R. Jung, and S. Weiss, "Covariance Pre-Integration for Delayed Measurements in Multi-Sensor Fusion," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Nov. 2019, pp. 6642–6649, iSSN: 2153-0866.

[16] A. Barrau and S. Bonnabel, "The invariant extended kalman filter as a stable observer," *IEEE Transactions on Automatic Control*, vol. 62, no. 4, pp. 1797–1812, April 2017.

[17] A. Barrau and S. Bonnabel, "Linear observed systems on groups," *Systems & Control Letters*, vol. 129, pp. 36–42, 2019.

[18] ——, "Invariant filtering for pose ekf-slam aided by an imu," in *2015 54th IEEE Conference on Decision and Control (CDC)*. IEEE, 2015, pp. 2133–2138.

[19] K. Y. Leung, Y. Halpern, T. D. Barfoot, and H. H. Liu, "The utias multi-robot cooperative localization and mapping dataset," *The International Journal of Robotics Research*, vol. 30, no. 8, pp. 969–974, 2011.

Bias Compensated UWB Anchor Initialization using Information-Theoretic Supported Triangulation Points

Julian Blueml[1], Alessandro Fornasier[1], Stephan Weiss[1]

***Abstract*— For Ultra-Wide-Band (UWB) based navigation, an accurate initialization of the anchors in a reference coordinate system is crucial for precise subsequent UWB-inertial based pose estimation. This paper presents a strategy based on information theory to initialize such UWB anchors using raw distance measurements from tag to anchor(s) and aerial vehicle poses. We include a linear distance-dependent bias term and an offset in our estimation process in order to achieve unprecedented accuracy in the 3D position estimates of the anchors (error reduction by a factor of about 3.5 compared to current approaches) without the need of prior knowledge. After an initial coarse position triangulation of the anchors using random vehicle positions, a bounding volume is created in the vicinity of the roughly estimated anchor position. In this volume, we calculate points which provide the maximal triangulation related information based on the Fisher Information Theory. Using these information theoretic optimal points, a fine triangulation is done including bias term estimation. We evaluate our approach in simulations with realistic sensor noise as well as with real world experiments. We also fly an aerial vehicle with UWB-inertial based closed loop control demonstrating that precise anchor initialization does improve navigation precision. Our initialization approach is compared to state-of-the-art as well as to an initialization without the simultaneous bias estimation.**

I. Introduction

For UAV localization, often a Global Navigation Satellite System (GNSS) is used. But in areas where there is no GNSS signal available, e.g. forest or indoor locations, some other form of localization provider needs to be available. This localization provider can for example be a set of UWB modules. UWB is a communication technique which operates in the RF (radio frequency) spectrum and as the name implies, it operates on a large band of frequency. This results in much more precise and less error prone distance measurement than other e.g. ultrasonic based systems. The position of a mobile robot can be calculated in a similar fashion as it is done in GNSS systems. The position can be computed through trilateration using at least three UWB modules which are configured to be senders (also called anchors). Similarly to the GNSS satellites for an accurate estimation of the mobile robot, the positions of these anchors have to be known as accurately as possible. Often, this is measured manually but this can be very time consuming and inaccurate, especially in wide areas and with low-quality beacons, in buildings with a large number of rooms, or in areas where the anchors are hard to reach. Even though the location measurement is accurate, biases in the signals may introduce inaccuracies in the trilateration process.

[1]The authors are with the Control of Networked Systems Group, University of Klagenfurt, Austria j1blueml@edu.aau.at, {firstname.lastname}@ieee.org

This work is supported by the EU-H2020 project BUGWRIGHT2 (GA 871260) and the BMVIT project SCAMPI (GA 878661)

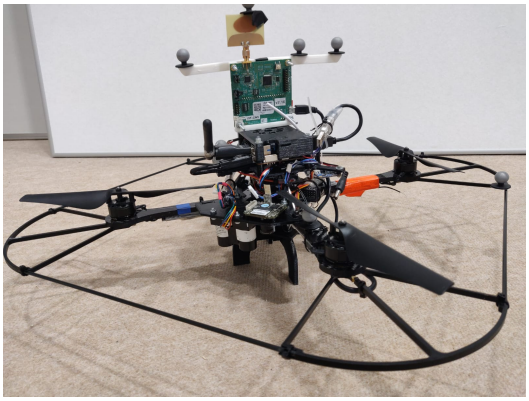


Fig. 1: UAV used for real world experiments with computation board and UWB node.

Thus, the here presented approach not only focuses on the precise anchor initialization without prior knowledge, but also on the estimation of bias terms in the raw signal. The goal is to place the anchors randomly in a room and the mobile robot, in our case a UAV, initializes the anchor positions automatically. In addition it calculates a linear bias model with a constant offset term for the distance dependent error of the UWB modules. The proposed initialization is a two stage process. First, the UAV navigates to some random points in space and records at each point a measurement to the UWB anchor which's position should be initialized. After sufficient points have been reached an initial guess of the position is performed. We leverage and extend the approach presented in [1] with a modified least squares approach to include the bias terms. The calculated position and corresponding covariance matrix is used to calculate an appropriate boundary volume which is used to construct optimal points using the Fisher Information Matrix (FIM). At these optimal points, the available information of the range-related UWB measurements is maximised to archive best trilateration results. With the information obtained by the mobile robot at these optimal points a final estimation of the position of the UWB anchor including the linear distance dependend bias and constant offset is calculated using the same modified least squares algorithm used for the trilateration from the random points.

II. Related work

In GPS denied environments e.g. inside buildings, range sensors are a popular choice for localization tasks. In [2]

several different indoor positioning systems and their algorithms are examined. They found out that systems using infrared, ultrasonic sound or UWB signals have the best accuracy but infrared and ultrasonic sound suffer in non line of sight situations. With these signals the error increases while with UWB signals the accuracy stays approximately the same even in non line of sight conditions. The authors of [3] propose a UWB-IMU pose estimation system. The system assumes known, fix UWB anchor positions and is reliable under multipath effects and non line of sight conditions. Ledergerber et al. [4] presented a localization system using UWB transceivers with known positions for robot localization. The system is also able to handle multiple robots simultaneously.

There is a large body of work in the area of calibrating (or initializing) positioning systems. The position of the anchors have to be known as exact as possible to reduce the localization error. Usually calibration is done manually by measuring the exact position of the UWB anchors but since this is an error prone and time-consuming procedure and also not suitable in some scenarios we want to avoid it whenever possible. Hol et al. [5] proposed a calibration method for UWB receivers for indoor positioning. First multiple UWB receivers are placed to stationary places. The same number of transmitters are placed near to the receivers. They acquire a dataset for this configuration. On this dataset a nonlinear optimization is performed. Then a transmitter is moved around the receivers and another dataset was recorded. A second nonlinear optimization was done on the second dataset with the positions obtained from the first optimization as initial values.

Another approach to perform anchor initialization is described in [6]. The goal of this paper is to provide a initialization method for dynamic anchor setups. Range only measurements are performed between mobile tag and fied anchors. The proposed approach is apparently very robust against multipath propagation because a RANSAC based outlier rejection is used before the position candidate is further refined by an Unscented Kalman Filter (UKF).

Another way to auto calibrate UWB anchors is to use range information from a receiver and estimate the position of the anchors. Therefore, the range-related information of the anchors is maximized. For maximizing information a popular tool is the FIM or its inverse which corresponds to the Cramer-Rao Lower Bound (CRLB). Cardinali et al. [7] used the Cramer-Rao Lower Bound on different UWB signals to obtain the ranging accuracy of these signals. The authors of [8] proposed an algorithm for optimal sensor placement in 2D. By maximizing the FIM the optimal sensor positions can be obtained in order to get the position of the signal transmitters.

In our work, we extend the approach of [8] to 3D and flip the problem set to determine the optimal positions of the moving module to gather most information for the triangulation of the fixed module(s).

In [6], the authors provide an initialization method for dynamic anchor setups using only the range measurements from the UWB modules. The authors apply a cascade containing an outlier removal step through RANSAC with a subsequent filtering process based on an Unscented Kalman Filter (UKF). The double use of the same information in the RANSAC and UKF step may lead to inconsistencies. In addition, the selected positions for triangulation are on a fix grid pattern and not chosen based on their information content.

With respect to the state of the art, we improve the initialization of the anchors' position in 3D and include signal bias terms to additionally improve subsequent state estimators on mobile systems using the UWB anchors as positioning system. In particular our contributions are as follows:

- the extension from 2D to 3D space and flipping of FIM/CRLB based optimal sensor placement methods [8] for range sensing modules .
- FIM/CRLB definition for the problem set with extended covariance models including distance dependency, bias terms, and correlation between measurement positions.
- the extension to initialize several UWB anchors in real-time with low computational complexity and improved models including distance dependent bias and offset terms without any prior knowledge.
- a detailed evaluation based on verified simulations and realistic real experiments including a comparison (and improvement) to a state of the art approach.
- an evaluation of the effect of the anchor initialization-precision on the navigation precision when three UWB anchors are used for on-board real-time UWB-inertial positioning control of a UAV.

III. UWB ANCHOR INITIALIZATION PROCESS

A. Coarse initial position computation

Over the entire initialization process to compute the UWB anchor positions in a 3D reference frame, we assume the mobile robot, in our case a UAV, can estimate its own pose in the 3D reference frame through other sensor modalities (e.g. vision based, with GNSS signals, laser, etc). In our real world examples, we use an Optitrack motion capture system.

To calculate the information content of a UAV position for best UWB anchor initialization based on the FIM, at least a rough estimate of the UWB anchor needs to be available. For this coarse initialization, we fly the UAV to random positions while gathering range measurements from the UWB node on the vehicle to the anchor we want to initialize. We extend the approach presented in [1] such that we can formulate a linear least squares as shown in the following even with our additional states including the distance dependent bias and constant offset. The distance from the node on the UAV to the anchor can be expressed as:

$$z^2 = (p-q)^2 = (p^2 - 2pq + q^2) \tag{1}$$

$$d_p^2 = p_x^2 + p_y^2 + p_z^2 \ , \ d_q^2 = q_x^2 + q_y^2 + q_z^2 \tag{2}$$

where $p = [p_x,\, p_y,\, p_z]^T$ describes the node position in the global frame, $q = [q_x,\, q_y,\, q_z]^T$ describe the position of the

anchor, d_q the distance from the anchor to the origin of the global frame. z is the distance between node and anchor and d_p the distance from the node position to the origin of the world frame. Assuming known node (i.e. UAV) positions and no biases as done in [1], for each distance measurement between node and anchor we can then formulate a modified least squares problem as

$$\begin{bmatrix} 2p_{x(t_1)} & -2p_{y(t_1)} & -2p_{z(t_1)} & 1 \\ -2p_{x(t_2)} & -2p_{y(t_2)} & -2p_{z(t_2)} & 1 \\ \vdots & & \vdots & \\ -2p_{x(t_n)} & -2p_{y(t_n)} & -2p_{z(t_n)} & 1 \end{bmatrix} \begin{bmatrix} q_x \\ q_y \\ q_z \\ d_q^2 \end{bmatrix} = \begin{bmatrix} z_{(t_1)}^2 - d_{p(t_1)}^2 \\ z_{(t_2)}^2 - d_{p(t_2)}^2 \\ \vdots \\ z_{(t_3)}^2 - d_{p(t_3)}^2 \end{bmatrix} \tag{3}$$

which is a set of linear equations in the form of $Ax = b$ where the rows of A are a measurement at time t_i. This can be solved for the anchor position q. Although UWB sensors are said to be fairly robust against multi-path issues, they show in practice a non-negligible distance dependent bias and constant offset depending on the manufacturer. To increase the accuracy of the triangulation results, we extend the above distance model of Eq.(1) with a distance dependent bias β and a constant offset γ to better reflect the actually measured distance z_m

$$z_m = \beta z + \gamma \tag{4}$$

Following the idea in [1], we design two additional auxiliary elements β^2 and γ, and modify the previous distance term d_q^2 in Eq.(3) to include the new bias terms

$$\begin{bmatrix} -2p_{x(t_1)} & \dots & -2p_{x(t_n)} \\ -2p_{y(t_1} & \dots & -2p_{y(t_n} \\ -2p_{z(t_1)} & \dots & -2p_{z(t_n)} \\ d_{p(t_1)}^2 & \dots & d_{p(t_n)}^2 \\ 2z_{(t_1)} & \dots & 2z_{(t_n)} \\ 1 & \dots & 1 \end{bmatrix}^T \begin{bmatrix} \beta^2 q_x \\ \beta^2 q_y \\ \beta^2 q_z \\ \beta^2 \\ \gamma \\ \beta^2(d_q^2 - \gamma^2) \end{bmatrix} = \begin{bmatrix} z_{t_1}^2 \\ z_{t_2}^2 \\ \vdots \\ z_{t_n}^2 \end{bmatrix} \tag{5}$$

solving this linear set of equation in the form of $Ax = b$ allows then to solve for the anchor position q and the two bias terms β and γ. The entries of Eq.(5) are based on the randomly chosen UAV positions. In practice, this system of equations is usually not well posed yielding poor solutions. Nevertheless, the coarse direction and distance can be inferred as an initial guess to apply our information theoretic approach for optimal UAV position selection in a refinement step as detailed below.

B. FIM based optimal points calculation

The goal is to find the optimal positions where the UAV (i.e. the UWB node) has to be placed in a limited volume to best triangulate a fix UWB anchor in the global coordinate frame. Until this anchor is triangulated, we assume the UAV position is known within a bounded volume (e.g. through fusion of IMU and a visual fiducial in the volume where the fiducial is in the field of view, a traking system, an area where GNSS signals are available, etc). In order to find the optimal sensor placement, the corresponding Cramer-Rao Lower Bound (CRLB) or FIM is considered [9]. The CRLB expresses a lower bound on the variance of estimators of a deterministic parameter. By achieving this bound the unbiased estimator is said to be (fully) efficient. The FIM on the other hand captures the amount of information from the obtained measured data of an unknown parameter which gets estimated. Under the regularity conditions the variance of any unbiased estimator is at least as high as the inverse of the FIM and the following inequality holds:

$$Cov\{\hat{\theta}\} \geq FIM(\theta)^{-1} = CRLB(\theta) \tag{6}$$

where θ is the variable of the estimation problem and where

$$Cov\{\hat{\theta}\} = E\left\{(\hat{\theta} - \theta)(\hat{\theta} - \theta)^T\right\} \tag{7}$$

$Cov\{\hat{\theta}\}$ corresponds to the covariance matrix of the estimated parameters. In the following, $FIM(\theta)$ (abbreviated as FIM) is defined as

$$FIM(\theta) = E\left\{(\nabla_\theta log\ p_\theta(z))(\nabla_\theta log\ p_\theta(z))^T\right\} \tag{8}$$

where $\nabla_\theta log\ p_\theta(z)$ denotes the gradient of the log-likelihood function with respect to the unknown parameter θ. By selecting a proper estimator the minimization of the CRLB or the maximization of the FIM leads to a decrease of the uncertainty when estimating the parameter.

1) Fisher Information Matrix for UWB anchor initialization: Let $\mathcal{I}$ denote the global reference frame and let $q = [q_x, q_y, q_z]^T$ be the position of the UWB anchor which's position needs to be refined in $\mathcal{I}$. Furthermore, let the position of the UWB node mounted on the UAV, assuming no or known offset between IMU and mounted UWB node, in $\mathcal{I}$ be $p_i = [p_{ix}, p_{iy}, p_{iz}]^T$ with $i = 1, 2, \dots, n$ the $i-th$ position of the UAV where a measurment was taken. The distance between the UWB anchor and the $i-th$ position of the UWB node on the UAV is then given by $d_i = ||q - p_i||$, where $||\cdot||$ denotes the euclidean norm. The, now noisy, measurement model from Eq. (4) is then given by

$$z_{m_i} = \beta(||q - p_i|| + \omega_i) + \gamma = \beta(d_i + \omega_i) + \gamma, i = 1 \dots n \tag{9}$$

where z_{m_i} is the $i-th$ distance measurement and ω_i as distance dependent additive noise. Usually it is assumed that the measurement noise is additive zero mean white Gaussian noise with $\omega_i \sim \mathcal{N}(0, C_i(d_i))$ and $C_i = \sigma^2(I + d_i)^2$, where I is the identity matrix (i.e. all noise sources are independent). In vector notation we have $\mathbf{z}_m = [z_{m_1}, z_{m_2}, \dots, z_{m_n}]^T$ which corresponds to the vector containing the distance measurements, the vector of the actual ranges is $\mathbf{d} = [d_1, d_2, \dots, d_n]^T$ and the corresponding measurement noise vector is $\boldsymbol{\omega} = [\omega_1, \omega_2, \dots, \omega_n]^T$. In order to obtain the Fisher Information Matrix we have to calculate

$$FIM(\theta) = E\left\{(\nabla_q log\ p_q(\mathbf{z}_m))(\nabla_q log\ p_q(\mathbf{z}_m))^T\right\} \tag{10}$$

where $p_q(z)$ is the likelihood function for the target positioning problem which is given by

$$p_q(z_m)) = \frac{1}{(2\pi)^{\frac{n}{2}}|C|^{\frac{1}{2}}} exp\left\{-\frac{1}{2}(\mathbf{z}_m - \mathbf{d})^T C^{-1}(\mathbf{z}_m - \mathbf{d})\right\} \tag{11}$$

For general Gaussian noise there is also a general expression of the Fisher Information Matrix [10]. For the estimation of

the UWB module this expression is given by Eq. (12).

$$FIM(q)_{kl} = \frac{\partial \mathbf{z}_m(q)}{\partial q_k}^T C(q)^{-1} \frac{\partial \mathbf{z}_m(q)}{\partial q_l} + \frac{1}{2} tr \left\{ C^{-1}(q) \frac{\partial C(q)}{\partial q_k} C^{-1}(q) \frac{\partial C(q)}{\partial q_l} \right\} \quad (12)$$

with the indices k and l representing the three coordinate axis x, y, z respectively. Note that with our extension to use a distant dependent bias term, each covariance matrix C_i per measurement is dependent on the anchor position q. Thus, the second term in Eq. 12 needs to be considered as non-zero term.

2) Optimality criteria: There are several optimality criteria for the Fisher Information Matrix to maximize the gathered information. Some of them are described in [11]:

- D-optimum design: the determinant of the FIM gets maximized: $\left(arg \max_{\theta \in \Re^n} |FIM(\theta)| \right)$.
- A-optimum design: the trace of the inverse of the FIM gets minimized: $\left(arg \min_{\theta \in \Re^n} tr \left(FIM(\theta)^{-1} \right) \right)$.
- E-optimum design: the smallest eigenvalue of the FIM gets maximized: $\left(arg \max \min_{\theta \in \Re^n} eigv \left(FIM(\theta) \right) \right)$.

For this paper the D-optimum design is chosen. It minimizes the volume of the multi-dimensional uncertainty ellipsoid for the parameters to be estimated for a given model. The A-optimum design minimizes the trace of the CRLB which results in minimizing the average variance of the estimates. The E-optimum design maximizes the smallest eigenvalue of the Fisher Information Matrix which means that the length of the largest axis of the uncertainty ellipsoid gets minimized. The main advantage of D-optimum design is that it is scale invariant in the parameters and it is also invariant to linear transformations. A-optimum design and E-optimum design are not invariant to these transformations. The disadvantage of D-optimum design is that if no global optimum is found the obtained D-optimum design can be erroneous. This is due to the fact that the uncertainty ellipsoid can get minimized in one dimension while in the other dimension we do not have information at all. In other words, the uncertainty ellipsoid is very small in on direction while it is very large in the other direction. Due to the computational constraints we have on the UAV and the benefit of the D-optimum of not requiring to compute a matrix inverse, it is, however, still our favorite choice; the E-optimum design needs to compute the eigenvalues of the FIM and the A-optimum design needs to inverse the FIM.

Under certain assumptions, the maximization of the FIM determinant could be solved analytically. As an example [12] assumes that the measurement points are only on a circle and the source is in the middle of the circle. This gives an optimal sensor placement when the sensors are placed in $2\pi\ i/n;\ i = 1, 2, \ldots, n$ angles around the source on the circle. With this approach the number of sensors placed around the source can be arbitrary. In [13] this approach gets extended to 3D. Again assumptions are made in order to get an analytical solution. The sensors are now placed on a sphere and the source is placed in the middle of it. This sphere gets intersected with a hyperboloid. The sensors are then placed on the intersection area. Since we do not want to make any assumption on the position of the range module and the measuring point e.g. we want to place the measurement point freely in a certain area and the source can be placed anywhere in a certain location, we calculate the maximum of the FIM determinant numerically using the previous coarse initialization of the anchor as a rough estimate of q. For simulation purposes the Global Optimization Toolbox of MATLAB is used.

As a toy example to demonstrate the functioning of our approach, in Fig. 2, we assume that the UAV is only allowed to move in a volume of $1 \times 1 \times 1m$ and we would like to achieve best UWB anchor-position initialization by only flying the UAV to five positions. Furthermore, we assume distance dependent covariance matrix. The true location of the UWB anchor to be estimated is set to $[1.5, 1, 0]^T m$. As it is intuitive, the optimal positions for the UAV to fly to within the allowed volume are at the corners of the cube closest to the anchor.

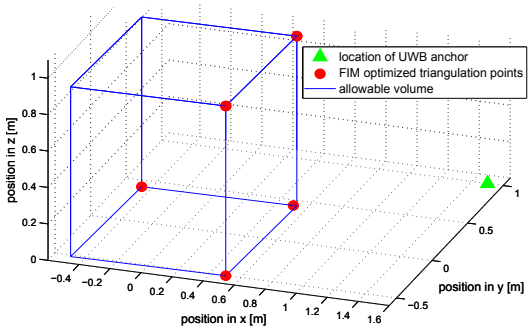


Fig. 2: Optimal measurement points for 5 measurements and 1 UWB anchor at $[1.5, 1, 0]^T$ for a distance dependent covariance matrix.

This toy example also highlights the low sensitivity of the selection of the optimal points in the volume with respect to the UWB anchor position: Already a rough initial direction and distance of the anchor with respect to the volume is suffcient to converge to the depicted result in Fig. 2. Or in other words, to make the optimal points be placed at different locations than depicted in Fig. 2, the true anchor position needs to drastically change. Furthermore, on distance independent covariance matrices, [14] proposed to transform the FIM to spherical coordinates to emphasize that the FIM depends on the *angle* between the range vectors. Adding the distance dependent element essentially adds the requirement "closer is better" – again without the need of very precise initial position information of the anchor. This low sensitivity of the optimal point placement with respect to the anchor point is in favor our our coarse initialization still being sufficiently accurate to generate informative points in a volume for the subsequent anchor-position refinement.

C. Distance dependent and position correlated covariance

The above toy example included the constraint that no two positions are allowed to be selected at the same locations. Consider again the above mentioned distance dependent covariance matrix $C_i = \sigma^2(I + d_i)^2$. One can see that it depends explicitly on the distance between the anchor and the measurement points (i.e. on $d_i = ||q - p_i||$). When the determinant of the FIM gets maximized, all positions of the measurement points tend to collapse over the range module since the distance dependent measurement error gets reduced as much as possible. This means that we have to define constraints for the optimization algorithm. In reality, and given the requirement of a base-line for later trilateration of the anchor position through use of the UAV positions, the measurement points are more correlated the closer they are to each other. This has to be considered in the covariance matrix for the FIM. For the correlated covariance matrix the squared exponential covariance is used. It is defined as follows per element:

$$C_{c_{i,j}} = \sigma^2 \, exp\left(-\frac{(p_i - p_j)^2}{2l^2}\right), \tag{13}$$

where l is the length-scale. The length-scale indicates the smoothness of the function. Large length-scale values characterize slow changing functions while small values characterize functions which can change quickly.
By combining the distance depended covariance matrix and the correlated covariance matrix one obtains

$$C = \sigma^2 \left(I + \delta\left(\mathbf{d}\right)\right)^2 + C_c \tag{14}$$

D. Refined anchor positioning and bias calculation

Using this definition of the covariance matrix in the proposed D-optimum FIM optimizer, we take the UAV positions correlation into account and can ensure well spaced measurement points in the defined volume. Once the optimal positions are defined in the volume we re-solve Eq. 5 for the refinement of the anchor position and at the same time for the bias terms. The anchor position and bias terms are later used in the closed loop tightly coupled UWB-inertial based control of the UAV.

IV. Results

A. Simulation results

Using the process described in Section III we simulated UWB range measurements to different locations using our distant dependent bias model from Eq. (9) with $\beta = 0.0049$ and $\gamma = 0.0951$. These values result from static tests with real hardware. For the standard deviation of the added noise, we did a sweep from σ starting at $0.02m$ to $0.2m$ in $0.02m$ steps. Each σ step consists of 200 individual simulation runs in order to obtain statistically relevant results. Fig. 3 shows the results. We noticed that our bias compensation significantly improved the results: for e.g. $\sigma = 0.1m$ the mean initialization error without bias consideration was $0.29m$ whereas it dropped to $0.13m$ using our model including the bias terms. Similarly, the error dropped from $0.58m$ to $0.23m$ for $\sigma = 0.2m$ with increasing improvements at higher noise values.

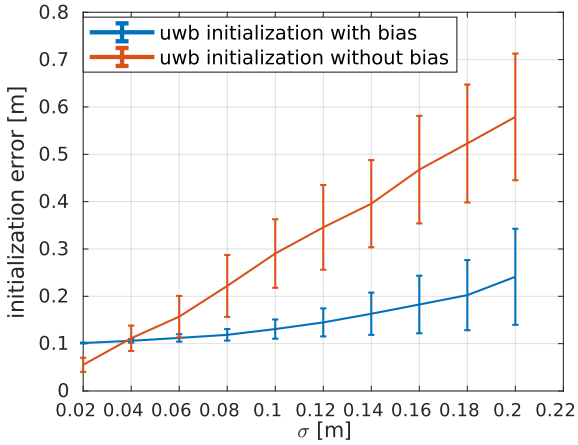


Fig. 3: Error statistics for our proposed bias compensated UWB anchor position initialization in Eq. (5) versus the one proposed in [1] without bias compensation in Eq. (3).

In Fig. 4, we show the complete initialization procedure showing the true position of the anchor (red triangle), the randomly selected initial triangulation points (green "x") with the coarse initial anchor estimation resulting from using these positions in Eq. (5) (green triangle), the subsequently selected volume within which information theoretic optimal triangulation positions are chosen (blue "x"), and the refined anchor position estimation based on these optimal position using again Eq. (5) (blue triangle). As a comparison and demonstration of the effect of taking our suggested bias compensation into account, the figure also shows the triangulated anchor position using the optimal points but Eq. (3) without modelling the bias (black triangle).

B. Real world results

We further performed a series of real experiments to demonstrate the use of our approach with real hardware and even for subsequent UWB-inertial closed loop control of a UAV. For all real experiments, we use an Asctec Hummingbird quadrotor (Fig. 1) equipped with a flight computer (Odroid XU4) and a UWB module (DecaWave TREK1000). Furthermore, three UWB modules (DecaWave TREK1000) are placed arbitrarily in the environment. The UWB distance measurements have a standard deviation of $0.09m$. We use an Optitrack motion capture system to obtain the UAV position for all our process steps. We compare our real world results to the ones reported in [1] where the authors move a UAV on random trajectories to add range measurements whenever they improve the condition number of the matrix in Eq. 3 consisting of previous measurements. New measurements are added up to a maximum number of measurements or until a certain quality of the matrix' condition number is reached.

In a first experiment, we performed 120 initializations as reported in [1]. The mean initialization distance error using our bias compensated method in Eq. 5 is $0.0984m \pm 0.0401m$. Not using the bias compensation but with our

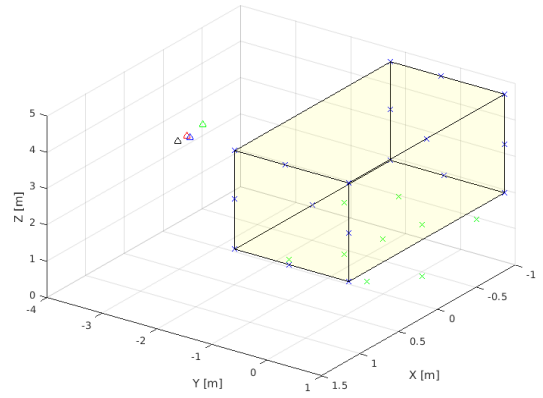


Fig. 4: Our proposed initialization procedure first using random triangulation points (green x) for coarse anchor initialization (green triangle) and subsequently for the FIM optimization to find optimal triangulation points (blue x) within a volume for position refinement (blue triangle). Also, the consideration of bias terms has an important positive performance impact (blue versus black triangle). Ground truth is the red triangle.

suggested method on FIM based triangulation position optimization we achieve a mean initialization distance error of $0.1417m \pm 0.0344m$. In contrast, the random approach based on the matrix condition number without considering biases in [1] reports an error of $0.3444m \pm 0.1326m$ (over 40 runs). Our approach shows an improvement by a factor of nearly 3.5. Fig. 5 shows the initialization results of our approach with bias consideration.

Fig. 5: Error statistics over 120 runs of UWB initialization

Additionally two more experiments where performed, a hovering test and an trajectory tracking test using a tightly coupled UWB-inertial EKF based on the anchor position initialized by our proposed method. Ground truth is optained by our Optitrack system. The mean tracking error for the trajectory following was $0.19m$ with a standard deviation of $0.0997m$ while flying 20 times a mission with 18 waypoints (Fig. 6). In the hovering test, the UAV was sent to the height of 1m and was hovering there for 60 seconds. We used five different pose estimators on the UAV for closed loop control: i) Optitrack as a reference (ref), ii) UWB measurements with correctly initialized anchor positions but without a bias model (u-gt), iii) UWB measurements with estimated anchor positions using our FIM optimization but without a bias model (u-est), iv) UWB measurements with estimated anchor positions using our FIM optimization and proposed bias model (u-bias), v) UWB measurements with

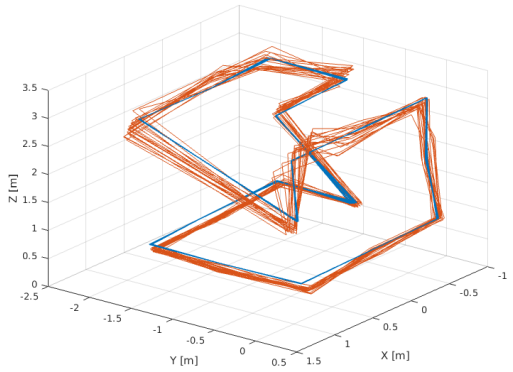


Fig. 6: Flying 20 times through 18 waypoints using a tightly coupled UWB-inertial EKF based on the anchor initializations of our proposed method. Ground trugh (blue) is obtained from an Optitrack system.

estimated anchor positions using the approach in [1]. Tab. I shows the RMSE for all setups. For the method proposed in [1] and with our best tuning knowledge applied, we still got to an RMSE of $0.713m$. Unfortunately, the authors in [1] did not report the performance purely navigating based on UWB-inertial estimation in their work. Interestingly, all other UWB based setups show similar performance despite the improved UWB positioning and bias compensation through our method. With an RMSE of over 1cm even with Optitrack measurement, we assume that this is due to the low controller performance of the UAV shadowing estimation accuracy.

	ref	u-gt	u-est	u-bias	[1]
RMSE [m]	0.012	0.025	0.029	0.028	0.713

TABLE I: Results of the hovering experiment

V. Conclusion

In this paper, we addressed the problem of accurate UWB anchor initialization without prior knowledge using the FIM for information-optimized triangulation-position selection and using a distance dependent bias model for the UWB measurements to improve the final triangulation accuracy. Our approach is based on two steps where we first use randomized triangulation points for a coarse anchor initialization and bias estimation. These values serve then for a FIM based optimization to generate optimal triangulation points used in a refinement step for anchor position and measurement biases. The result has a 3.5 times lower position error compared to state of the art and reaches an anchor initialization accuracy of $9.8cm$. The proposed approach can be applied sequentially or as a lump-sum optimization to multiple anchors to use their initialized positions for subsequent UAV flight based on on-board, real-time UWB-inertial state estimation. We showed real flight following a trajectory with an RMSE of 19cm and a hover performance of under 3cm RMSE greatly superseding previous approaches.

REFERENCES

[1] K. Hausman, S. Weiss, R. Brockers, L. Matthies, and G. S. Sukhatme. Self-calibrating multi-sensor fusion with probabilistic measurement validation for seamless sensor switching on a uav. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4289–4296, May 2016.

[2] Klaithem Al Nuaimi and Hesham Kamel. A survey of indoor positioning systems and algorithms. In *Innovations in information technology (IIT), 2011 international conference on*, pages 185–190. IEEE, 2011.

[3] J. D. Hol, F. Dijkstra, H. Luinge, and T. B. Schon. Tightly coupled uwb/imu pose estimation. In *2009 IEEE International Conference on Ultra-Wideband*, pages 688–692, Sept 2009.

[4] A. Ledergerber, M. Hamer, and R. D'Andrea. A robot self-localization system using one-way ultra-wideband communication. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3131–3137, Sept 2015.

[5] Jeroen D Hol, Thomas B Schön, and Fredrik Gustafsson. Ultra-wideband calibration for indoor positioning. In *Ultra-Wideband (ICUWB), 2010 IEEE International Conference on*, volume 2, pages 1–4. IEEE, 2010.

[6] Byungjae Park and Sejin Lee. Robust range-only beacon mapping in multipath environments. *ETRI Journal*, 42(1):108–117, 2020.

[7] Roberta Cardinali, Luca De Nardis, Maria Gabriella Di Benedetto, and Pierfrancesco Lombarde. Uwb ranging accuracy in high- and low-data-rate applications. *IEEE Transactions on Microwave Theory and Techniques*, 54(4):1865–1875, 4 2006.

[8] David Moreno-Salinas, Antonio M Pascoal, and Joaquin Aranda. Optimal sensor placement for multiple target positioning with range-only measurements in two-dimensional scenarios. *Sensors*, 13(8):10674–10710, 2013.

[9] Harry L Van Trees and Kristine L Bell. *Detection estimation and modulation theory, pt. I*. Wiley, 2013.

[10] Steven M. Kay. *Fundamentals of Statistical Signal Processing: Estimation Theory*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1993.

[11] Dariusz Ucinski. *Optimal measurement methods for distributed parameter system identification*. CRC Press, 2004.

[12] David Moreno-Salinas, Antonio M Pascoal, and Joaquin Aranda. Optimal sensor placement for multiple target positioning with range-only measurements in two-dimensional scenarios. *Sensors*, 13(8):10674–10710, 2013.

[13] David Moreno-Salinas, Antonio Pascoal, and Joaquin Aranda. Optimal sensor placement for acoustic underwater target positioning with range-only measurements. *IEEE Journal of Oceanic Engineering*, 41(3):620–643, 2016.

[14] Bruno Ferreira, Aníbal Matos, and Nuno Cruz. Optimal positioning of autonomous marine vehicles for underwater acoustic source localization using toa measurements. In *Underwater Technology Symposium (UT), 2013 IEEE International*, pages 1–7. IEEE, 2013.

sensors

MDPI

Article

UWB-Based Self-Localization Strategies: A Novel ICP-Based Method and a Comparative Assessment for Noisy-Ranges-Prone Environments

Francisco Bonnin-Pascual and Alberto Ortiz *

Department of Mathematics and Computer Science, University of the Balearic Islands, 07122 Palma de Mallorca, Spain; xisco.bonnin@uib.es

* Correspondence: alberto.ortiz@uib.es

Received: 31 July 2020; Accepted: 25 September 2020; Published: 1 October 2020

Abstract: Ultra-Wide-Band (UWB) positioning systems are now a real option to estimate the position of generic agents (e.g., robots) within indoor/GPS-denied environments. However, these environments can comprise metallic structures or other elements which can negatively affect the signal transmission and hence the accuracy of UWB-based position estimations. Regarding this fact, this paper proposes a novel method based on point-to-sphere ICP (Iterative Closest Point) to determine the 3D position of a UWB tag. In order to improve the results in noise-prone environments, our method first selects the anchors' subset which provides the position estimate with least uncertainty (i.e., largest agreement) in our approach. Furthermore, we propose a previous stage to filter the anchor-tag distances used as input of the ICP stage. We also consider the addition of a final step based on non-linear Kalman Filtering to improve the position estimates. Performance results for several configurations of our approach are reported in the experimental results section, including a comparison with the performance of other position-estimation algorithms based on trilateration. The experimental evaluation under laboratory conditions and inside the cargo hold of a vessel (i.e., a noise-prone scenario) proves the good performance of the ICP-based algorithm, as well as the effects induced by the prior and posterior filtering stages.

Keywords: UWB positioning system; point-to-sphere ICP; range filtering; ferromagnetic interference

1. Introduction

Position estimation in GPS-denied environments is of great interest in a large variety of applications, including indoor mobile robotics. Generally speaking, the so-called Indoor Positioning Systems (IPSs)—that is, systems that continuously and in real-time determine the position of an object in an indoor environment—can be applied in these cases [1]. From the technological point of view, IPSs comprise Radio Frequency Identification (RFID)-, Infrared (IR)-, Ultrasound (US)-, ZigBee-, Wireless Local Area Network (WLAN)-, and Ultra-Wide-Band (UWB)-based approaches, to name but a few. It is well-known that each of these technologies has its own pros and cons. By way of example, RFID localization systems do not require Line-of-Sight (LOS) to operate, which is critical for IR-based devices, but the coverage of the former is smaller in comparison with other technologies; IR and US signals do not penetrate solid walls, while ZigBee and WLAN signals do; ZigBee, however, is vulnerable to a wide range of signal types using the same frequency; while the performance of WLAN-based systems can be affected by changes in the strength map of the operating area. Likewise, UWB systems allow high-accuracy positioning, but can be affected by the presence of metallic materials. These are only a selection of the considerations to be made; the reader is referred to [2] for a more detailed overview and a comparison of IPSs.

In this paper, we focus on UWB positioning systems, with the goal of estimating the position of one or more devices, generally named *tags*, which are moving through an environment where a set of devices/beacons named *anchors* have previously been placed. UWB-based IPSs typically measure the distance from the tag to each of the anchors (e.g., four for 3D pose estimation), and combine them to obtain the position estimate. Unfortunately, the quality of the measured ranges may be affected by noise, which consequently propagates through the calculations and affects the reliability of the position estimates [3]. Indeed, some manufacturers warn about unpredictable effects on range measurements because of the presence of metallic materials in the surroundings of the operation area, and hence they recommend ensuring a minimum distance (above 20 cm) between the anchors' antennas and any metallic element (see, e.g., www.pozyx.io/technology/where-to-place-the-anchors).

In this work, we propose a new method that has exhibited good performance in these noise-prone environments. The main novelty of this method is that it makes use of the well-known Iterative Closest Point (ICP) algorithm to estimate the position of the tag. Toward this end, we modified ICP, which is typically used to find the translation and rotation between two point clouds, to compute the position of the UWB tag through the computation of point-to-sphere correspondences. To the best of our knowledge, this new method is the first ICP-like algorithm that produces position estimates from data provided by a UWB-based localization system.

For performance evaluation purposes, we compared the aforementioned ICP-based method with other UWB-based position estimation approaches based on trilateration, focusing on the assessment of their noise-tolerance capabilities. Moreover, we also evaluated the effect of attaching pre- and post-processing filters to each of the methods involved in the comparison: on the one side, the pre-processing stage filters the tag-to-anchor range measurements on an anchor-by-anchor basis, while on the other side, the post-processing stage filters the raw position estimates resulting from the calculations. Finally, the algorithm's performance is assessed under laboratory conditions and—as already mentioned—within a particularly noise-prone environment, such as a cargo hold of a large-tonnage vessel. Once more, to the best of our knowledge, this is the first time that a UWB-based IPS has been evaluated inside a ship, and the obtained results are reported, which can be regarded as a secondary contribution.

The rest of the paper is organized as follows: Section 2 reviews main approaches in UWB-based position estimation methods; next, Section 3 overviews our methodology regarding UWB-based position estimation and details the pre- and post-processing stages; Section 4 describes the novel ICP-based method to estimate the UWB tag position; Section 5 overviews the different methods chosen for the comparative assessment, which is actually performed in Section 6, where we evaluate the performance of all the configurations considered, both under laboratory conditions and within a real, noise-prone environment; finally, Section 7 draws some conclusions about the new method, as well as about the experimental results reported.

2. UWB-Based Position Estimation

Among the different currently available possibilities, UWB technology has emerged as one of the leading core technologies for IPS development thanks to (1) the resilience of UWB ultra-short pulses to frequency-dependent absorption, (2) a relatively low cost and easy deployment, and (3) the ultimate accuracy which can be achieved. It is well known that one of the key points is the measurement of distances between the tag(s) and the anchors. In this regard, a rough classification of UWB-based position estimation methods can be stated according to the base estimation technique that is adopted [4]:

- **Time of Arrival (TOA)**. Algorithms in this category estimate the position of the tag computing the intersection between the circumferences (or spheres in 3D) centred at each anchor, whose radius is the estimated distance from the tag to the corresponding anchor. A survey reviewing several TOA methods can be found in [5]. In [6], the authors evaluate different TOA-based algorithms in

a realistic indoor environment. As a real application example, a UWB system based on TOA is used in [7] for personnel localization inside a coal mine.

- **Time Difference of Arrival (TDOA)**.This category comprises algorithms which estimate the position of the tag considering the difference between the reception times in each anchor given a signal sent by the tag. These methods require some synchronization mechanism between the different devices, as well as significant bandwidth in comparison with other methods. In [8], the authors propose a TDOA method to operate in complex environments, specially under non-line-of-sight (NLOS) conditions. This method makes use of an Extended Kalman Filter (EKF) as a post-processing stage. Another practical example is [9], which describes a real-time positioning system intended for disaster aid missions.
- **Angle of Arrival (AOA)**.Methods in this category estimate the position of the tag using the direction of propagation of the signals sent by multiple sources (i.e., the anchors). The location is found from the intersection of the angle line for each signal source. The algorithms based on AOA have a higher complexity and their accuracy may decrease when the distance increases. Among the large number of AOA-based approaches that can be found in the literature, we can mention [10], which makes use of a KF and relies on a linear quadratic frequency domain invariant beamforming strategy, and [11], which presents a cooperative positioning method that makes use of all the sensor nodes instead of using only the anchors.
- **Received Signal Strength (RSS)**.These methods employ the signal strength as an estimator of the distance. Among the many RSS-based algorithms, we can differentiate two main strategies. On the one hand, approaches based on trilateration, where the distance estimates are used to guess the position of the tag using the same methods employed by TOA methods (see for example [12,13]). On the other hand, a strategy based on RSS fingerprinting, where a dataset needs to be generated during a previous learning stage for collecting RSS data throughout the environment. This dataset is later used to compare with the RSS online measurements to estimate the location (see for example [14]).
- **Hybrid algorithms**.Hybrid techniques aim is to increase the precision of the position estimates by means of the combination of two or more of the aforementioned strategies. These methods are typically more complex and of higher and more intensive computational cost. By way of example, [15] reports on an EKF based on a TDOA/RSS algorithm to localize a UWB tag inside underground mines under NLOS conditions, while [16] evaluates several TDOA algorithms and concludes that a combination of them improves the accuracy of position estimates.

For a complete survey of UWB-based positioning algorithms, the reader is referred to [4,17,18].

3. General Overview and Methodology

The point of departure of our method is the availability of a regularly updated set of anchor-tag ranges, so that any beacon-based positioning system able to supply these data is susceptible to adopt our method for position estimation. This requirement is usually satisfied by UWB-based IPS vendors (see, e.g., Pozyx (www.pozyx.io) and Decawave (www.decawave.com) TOA-based solutions).

Regarding the position estimation procedure itself, we organize it as a process involving the following tasks (which are not sequenced in this order):

(a) Estimation of the position of the tag given a set of ranges to the anchors;
(b) Selection of the best subset of anchors to obtain the most accurate position estimation for the estimation method;
(c) Pre-filter (denoise) the available ranges; and
(d) Post-process (filter) the estimated positions.

Our aim in this work is to address and assess the four blocks. Regarding block (a), in this paper, we propose a novel method based on a particularization of the ICP algorithm. To properly feed this

block, which is detailed in Section 4, we select first the most suitable collection of anchors presumably leading to the best position estimate, addressing thus block (b) of the previous list.

Regarding (c), the idea behind introducing a previous stage is to improve the data used as input by the position estimation block [19]. As mentioned before, the estimated distances to the anchors can be affected by external disturbances because of the presence of metallic elements in the environment. For this reason, we consider the addition of a pre-processing stage to filter the distances to the anchors, so that only good distance estimates are used within the position estimation block. The rationale behind this is to prevent the position estimation process to provide results when the distances to the anchors are not of sufficient quality.

The pre-filtering process that we adopt comprises two stages: A peak filter (PF) and a moving average filter(MAF) [20,21]. On the one hand, the peak filter removes all those values whose absolute difference with the previous value is above a certain threshold. These values are considered as peaks and are discarded, while the surviving values are considered as valid measurements.

On the other hand, the moving average filter supplies smoothed distance estimates by computing the mean of the N last consecutive valid ranges received. Since the ranges are required to be consecutive, when a peak is detected, the moving average filter does not provide output values until N new consecutive valid measures are available again.

Finally, as for (d), we adopt a post-processing stage consisting in a Kalman Filter-based strategy for position estimates [22]. This block implements an EKF which combines position estimates with motion data supplied by an Inertial Measurement Unit (IMU). More precisely, we make use of the IMU orientation and linear accelerations.

As for the experimental methodology, in the experimental results section, we will consider different configurations for the previous organization: without pre- and post- filters, with only one of them, or with both, and at the same time using different position estimation blocks (a), being one of them the new ICP-based method described in Section 4 and being the others each one of the different estimation strategies based on trilateration which are reviewed in Section 5.

4. Point-to-Sphere ICP for UWB-Based Position Estimation

In this section, we propose a novel method for estimating the position of a UWB tag by means of a modified version of the well-known ICP algorithm. To the best of the authors' knowledge, the ICP algorithm, which is widely used for computing the roto-translation between two point clouds (e.g., provided by a LiDAR or from a depth camera), has never been used with data provided by a UWB positioning system. Nevertheless, the method described in this section can also be used with data provided by other systems based on the distances measured from a moving device to a set of beacons situated at known locations.

Our method modifies the ICP standard algorithm by computing a point-to-sphere correspondence between the 3D position of each anchor and the sphere defined by the distance to the anchor (i.e., the sphere radius) and the previous known location of the tag (i.e., the sphere center). More formally, let $\mathcal{A} = \{a_1, ..., a_N\}$ be the collection of $N \geq 4$ anchors located at known positions $\{l_1, ..., l_N\}$, at let us consider a moving tag situated at distances $\{r_1, ..., r_N\}$ from the anchors. We next consider the set of spheres $\mathcal{S} = \{s_1, ..., s_N\}$ with radius $\{r_1, ..., r_N\}$ centred at the last known location of the tag t. Then, the point-to-sphere ICP algorithm estimates the 3D translation of the spheres (and therefore the tag) necessary to allow for the anchors $a_i \in \mathcal{A}$ to lie on the surface of the corresponding spheres $s_i \in \mathcal{S}$.

The point-to-sphere ICP algorithm proceeds similarly to the point-to-line version of ICP [23]. At each iteration j, the algorithm computes, for each anchor $a_i \in \mathcal{A}$, the point c_{ij} in the surface of the corresponding sphere $s_i \in \mathcal{S}$ which is closer to the anchor location l_i. Being (t_x, t_y, t_z) the coordinates of the last known position of the tag, and (l_{ix}, l_{iy}, l_{iz}) the coordinates of the anchor a_i, the coordinates of the point c_{ij} can be computed as:

$$\begin{aligned} c_{ij,x} &= t_x + r_i \cos\alpha \cos\beta\,, \\ c_{ij,y} &= t_y + r_i \sin\alpha \cos\beta\,, \\ c_{ij,z} &= t_z + r_i \sin\beta\,, \end{aligned} \tag{1}$$

where

$$\begin{aligned} \alpha &= \tan^{-1} \frac{l_{iy} - t_y}{l_{ix} - t_x}\,, \\ \beta &= \tan^{-1} \frac{l_{iz} - t_z}{\sqrt{(l_{ix} - t_x)^2 + (l_{iy} - t_y)^2}}\,. \end{aligned} \tag{2}$$

Once all the correspondences c_{ij} have been obtained for all anchors $i \in \{1, 2, \ldots, N\}$, we define the set of points $\mathcal{C}_j = \{c_{1j}, ..., c_{Nj}\}$ for the current iteration j. This set of points $\mathcal{C}_j$ is used next to estimate the translation of the tag by means of least squares point-to-point distance minimization, by which the optimum translation can be proved to be the average of distances between the anchors l_i and the respective closest points c_{ij}. In the following iteration, the algorithm computes a new set of points $\mathcal{C}_{j+1}$, which is then used to update the translation estimate (notice that this algorithm only computes a translation, while the point-to-line ICP algorithm computes a full roto-translation). To make all this easier to understand, Figure 1 illustrates graphically the point-to-sphere correspondence process by means of the 2D version (i.e., the point-to-circumference correspondence process).

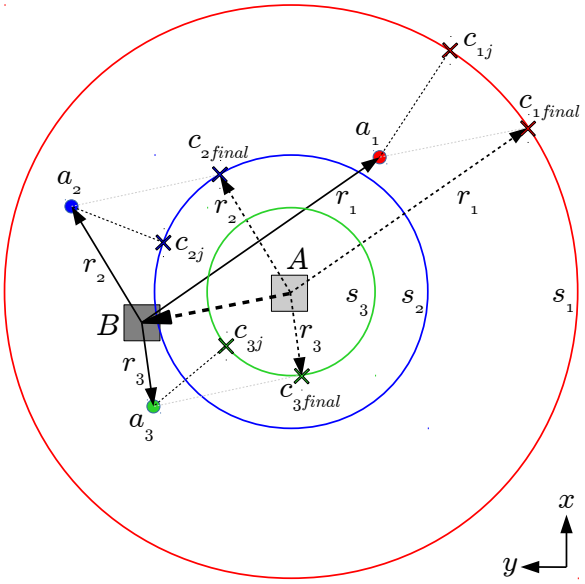


Figure 1. Example of point-to-circumference correspondences (2D case of point-to-sphere Iterative Closest Point (ICP)). The tag moves from point A to point B, while the anchors a_1, a_2 and a_3 are static. c_{1j}, c_{2j} and c_{3j} are the points over the three circumferences which are closer to the corresponding anchor in the current iteration j. $c_{1,\text{final}}$, $c_{2,\text{final}}$ and $c_{3,\text{final}}$ are the final correspondences after ICP convergence.

The ICP loop iterates until the update in the estimated translation is below a certain threshold, that is, convergence is achieved, or a maximum number of iterations is reached. Considering the typically reduced number of anchors used in UWB positioning, together with the fact that ICP can start from the previous estimate, the point-to-sphere ICP algorithm usually converges in a few iterations—around 50, and typically less than 200 irrespective of the starting position employed (e.g., $t_0 = (0, 0, 0)$). A description in pseudo-code of the point-to-sphere ICP algorithm can be found in Algorithm 1.

Algorithm 1 Point-to-sphere ICP algorithm to estimate the position of the UWB tag

procedure POINTTOSPHEREICP($\mathcal{L}, \mathcal{R}, t, \delta_{min}, max\_iter$)
  $\mathcal{L} = \{l_1, ..., l_N\}$: anchors' 3D positions
  $\mathcal{R} = \{r_1, ..., r_N\}$: distances from the tag to the anchors
  t: starting estimate of the tag position, such as the last estimate or $(0, 0, 0)$ the very first time
  δ_{min}: smallest position update to iterate once more
  max_iter: maximum number of iterations to stop ICP
  $\delta \leftarrow \infty$, $num\_iter \leftarrow 0$
  while $(\delta > \delta_{min})$ **and** $(num\_iter < max\_iter)$ **do**
    $\mathcal{C} \leftarrow$ **getClosestPoints**$(\mathcal{L}, \mathcal{R}, t)$ ▷ closest points obtained from Equation (1) and (2)
    $\mathcal{U} \leftarrow \mathcal{L} - \mathcal{C}$ ▷ set of 3D translations required for each sphere
    $mean\_update \leftarrow$ **average**$(\mathcal{U})$ ▷ average update for each axis (from closed-form
    ▷ solution of the underlying least-squares problem)
    $t \leftarrow t + mean\_update$ ▷ update the 3D position of the tag
    $\delta \leftarrow$ **norm**$(mean\_update)$ ▷ L2 norm of the update vector
    $num\_iter \leftarrow num\_iter + 1$
  end while
  return t ▷ return the updated 3D position of the tag
end procedure

Further, for higher robustness of point-to-sphere ICP, we enhance Algorithm 1 adopting a RANSAC-like estimation strategy [24]. That is to say, we consider random sets of $m \in \{4, \ldots, N\}$ anchors/ranges, apply Algorithm 1 to these minimum sets and determine the number of inliers among the full set of N available anchors/ranges. For inlier definition, we use the final point-to-sphere distance resulting for each anchor/range after ICP:

$$d_{\text{point-to-sphere, } i} = \|c_{i,\text{final}} - l_i\|_2 \tag{3}$$

that is, an anchor/range a_i/r_i is an inlier if $d_{\text{point-to-sphere, } i} < \tau_{inl}$, for a given threshold τ_{inl}. Finally, a set of anchors/ranges is considered the best set if it gives rise to the highest number of inliers, or, in case of tie, the sum of point-to-sphere distances is the lowest. Once the set of best anchors/ranges is available, we find the updated position applying Algorithm 1 to the corresponding set of inliers. Notice that, if the number of anchors is low, one can consider all possible combinations instead of a lower amount, as done by the original formulation of RANSAC.

To finish, it is worth mentioning that our method is also able to operate when, sporadically, less than four ranges are available because the remaining anchors are too distant, due to the presence of obstructing obstacles, because of punctual electromagnetic interference, etc. Under these conditions, Algorithm 1 can employ the available ranges to estimate the position of the tag, although at the expense of a higher error, which will depend on the number of available anchors and their locations. This makes it possible to operate in highly dynamic environments where other UWB positioning methods can not be used. However, although this is possible, we cancel the estimation process when not enough inliers can be found (at least m), and the method waits for the next set of ranges, in line with the idea of only supplying reliable position estimates.

Figure 2 depicts graphically the ICP-based algorithm, including the pre- and post-filtering stages which would also be attached to the approaches described in Section 5.

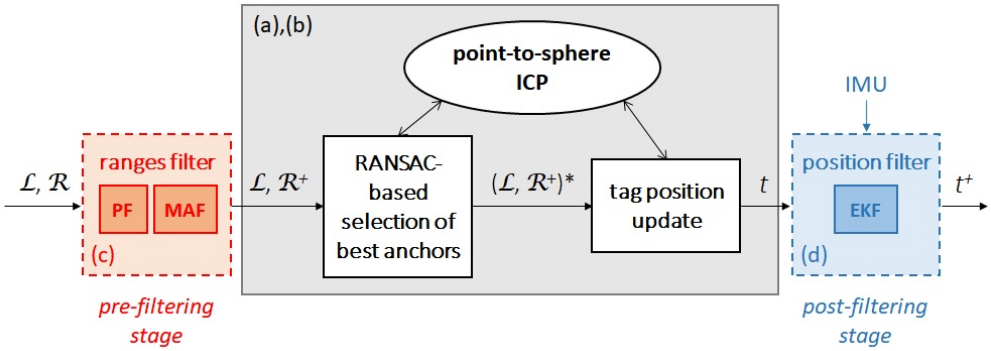


Figure 2. Block diagram of the full version of the ICP-based position estimation algorithm: $\mathcal{L}$ is the set of anchor locations, $\mathcal{R}$ is the set of ranges, $\mathcal{R}^+$ is the set of filtered ranges, $(\mathcal{L}, \mathcal{R}^+)^*$ denotes the best set of anchors/ranges, t is the tag position, and t^+ is the filtered tag position. The dashed boxes used for the pre- and post-filtering stages denote that they can be removed. The gray box refers to the section that would be replaced by any of the methods overviewed in Section 5. (**a**–**d**) as defined in Section 3.

5. Alternative Strategies

As alternative strategies to compare with point-to-sphere ICP, we consider three position estimation strategies based on trilateration. Trilateration can be described as a geometric method to find the location of a point based on the geometry of spheres, circles, or triangles. In the three-dimensional case, this method requires the location of at least three known points (e.g., the anchors) and the distances from all of them to the position to be determined (e.g., the UWB tag).

To solve for the position of the tag, the intersection of the spheres involved has to be found, using the distance between the tag and the corresponding anchor as the respective sphere radius. For a better understanding, see Figure 3, which shows this intersection for the 2D case. For the case of three anchors, the 3D position of the tag $t = (t_x, t_y, t_z)$ can be computed from the equations of the three spheres:

$$\begin{aligned} {r_1}^2 &= (t_x - l_{1x})^2 + (t_y - l_{1y})^2 + (t_z - l_{1z})^2, \\ {r_2}^2 &= (t_x - l_{2x})^2 + (t_y - l_{2y})^2 + (t_z - l_{2z})^2, \\ {r_3}^2 &= (t_x - l_{3x})^2 + (t_y - l_{3y})^2 + (t_z - l_{3z})^2, \end{aligned} \tag{4}$$

where $l_i = (l_{ix}, l_{iy}, l_{iz})$ is the location of anchor i, and r_i is the distance between anchor i and the tag.

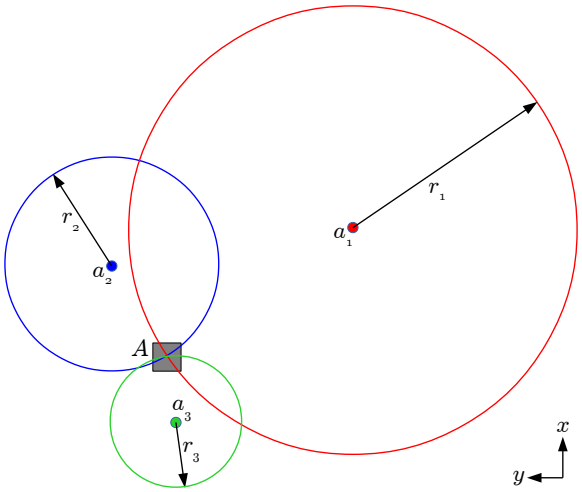


Figure 3. Trilateration example for the 2D case. The intersection of the three circumferences with radius r_1, r_2 and r_3, respectively centred at the anchors (a_1, l_1), (a_2, l_2) and (a_3, l_3), is used to compute the position of the tag situated at point A.

As already mentioned, for the comparative assessment with the point-to-sphere ICP, we consider three alternative strategies regarding the anchor selection. They all are detailed in the following sections.

5.1. RSS-Based Method

This method makes use of the RSS indicator to select the four anchors with highest values. Once the spheres are selected and ordered by this indicator, the method proceeds to compute the position of the tag from Equation (4) using the first three spheres, while the calculations for the fourth sphere are saved if they are not necessary. Three different situations may occur when the intersection between three spheres is considered:

1. the three spheres intersect in a single point (ideal case),
2. the circumference resulting from the intersection between the two first spheres does not intersect with the third sphere, and
3. the circumference resulting from the intersection between the two first spheres intersects with the third sphere at two points.

These three cases are depicted in Figure 4. In the second case, the intersection between the three spheres is accepted when the distance between the circumference resulting from the intersection of the first two spheres and the third sphere is below a certain threshold. Otherwise, the algorithm does not provide solution for the given anchors and it waits for the next distance measurements. In the third case, the algorithm selects the intersection point which is closest to the surface of the fourth sphere. Hence, this algorithm requires at least four anchors/spheres to proceed.

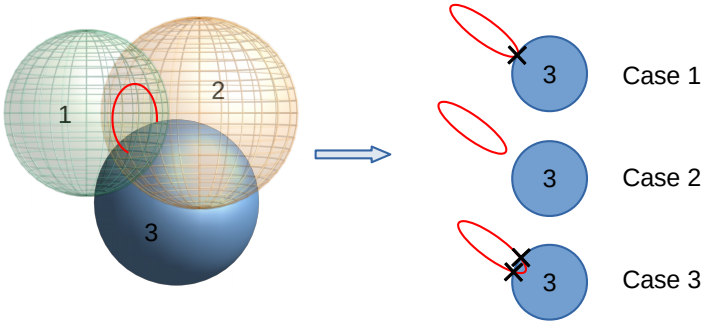


Figure 4. Three different cases for the intersection of three spheres.

To solve Equation (4) for the selected anchors, the anchors' coordinates are transformed to an auxiliary coordinate frame centred at the location of the first anchor (i.e., the anchor with the highest RSS value) with the x-axis pointing to the second anchor, and so that the XY plane is defined with the third anchor. After this reference frame change, the first case (i.e., the three spheres intersecting in a single point) takes place when

$$r_1{}^2 - x^2 - y^2 = 0, \tag{5}$$

where (x, y) is the intersection point in the auxiliary coordinate frame. The second case occurs when

$$r_1{}^2 - x^2 - y^2 < 0, \tag{6}$$

and the third case takes place when

$$r_1{}^2 - x^2 - y^2 > 0. \tag{7}$$

Notice that the intersection point (if any) is always located in the XY plane of the auxiliary coordinate frame, and thus its z coordinate is always 0. The coordinates of the UWB tag in this reference system can next be obtained by means of:

$$\begin{aligned} t_x &= \frac{{r_1}^2 - {r_2}^2 + {l_{2x}}^2}{2\,l_{2x}}, \\ t_y &= \frac{{r_1}^2 - {r_3}^2 + {l_{3x}}^2 + {l_{3y}}^2}{2\,l_{3y}} - \frac{l_{3x}\,t_x}{l_{3y}}, \\ t_z &= \sqrt{{r_1}^2 - {t_x}^2 - {t_y}^2}, \end{aligned} \tag{8}$$

where r_1 to r_3 are the distances to the four anchors chosen and l_2, l_3 and l_4 are the positions of the second, third and fourth anchors in the auxiliary reference frame. Then, if $t_z \neq 0$, two situations may occur:

- ${r_1}^2 - {t_x}^2 - {t_y}^2 < 0$ (case 2 above), that is, there is no intersection between the spheres, and
- ${r_1}^2 - {t_x}^2 - {t_y}^2 > 0$ (case 3 above). In this case, we compute the Euclidean distance from the tag to the fourth anchor considering the positive and negative solutions for t_z, and we select the solution which leads to the shortest distance.

Finally, the estimated position of the tag must be transformed back to the original reference system used by the UWB device.

5.2. Minimum Discrepancy-Based Method

In this case, we apply the same steps as the RSS-based method to compute the position of the tag, although we select the four anchors in a different way. Indeed, this method tries all the combinations of four anchors, among all the available anchors, and selects the one which leads to the minimum trilateration discrepancy. Considering four specific anchors, the trilateration discrepancy is computed as the mean of the differences between the measured anchor-tag distances and the distances computed from the estimated tag position (estimated using these four anchors) to the position of each one of these anchors. In other words, the optimum subset $\mathcal{B} \subset \mathcal{A}$ is such that $|\mathcal{B}| = 4$ and:

$$\mathcal{B} = \underset{\mathcal{B}' \subset \mathcal{A}}{\arg\min} \sum_{i=1}^{N} \frac{\left| r_i - \left(\sqrt{(t_{\mathcal{B}'x} - l_{ix})^2 + (t_{\mathcal{B}'y} - l_{iy})^2 + (t_{\mathcal{B}'z} - l_{iz})^2} \right) \right|}{N}, \tag{9}$$

where $t_{\mathcal{B}x}$, $t_{\mathcal{B}y}$ and $t_{\mathcal{B}z}$ are the coordinates of the tag position estimated using the subset of anchors $\mathcal{B}$ and as described in Section 5.1.

5.3. Least Squares-Based Method

Unlike the previous methods, this method makes use of all the available anchors to estimate the position of the UWB tag. This is performed through a least squares formulation which can be explained starting from the following equations corresponding to the N spheres:

$$\begin{aligned} {r_1}^2 &= (x - x_1)^2 + (y - y_1)^2 + (z - z_1)^2, \\ &\vdots \\ {r_N}^2 &= (x - x_N)^2 + (y - y_N)^2 + (z - z_N)^2. \end{aligned} \tag{10}$$

The subtraction of the last equation from the preceding ones, gives rise to the $N-1$ following equations:

$$\begin{aligned} 2(x_N - x_1)x + 2(y_N - y_1)y + 2(z_N - z_1)z = \\ {r_1}^2 - {r_N}^2 - {x_1}^2 - {y_1}^2 - {z_1}^2 + {x_N}^2 + {y_N}^2 + {z_N}^2, \\ \vdots \\ 2(x_N - x_{N-1})x + 2(y_N - y_{N-1})y + 2(z_N - z_{N-1})z = \\ {r_{N-1}}^2 - {r_N}^2 - {x_{N-1}}^2 - {y_{N-1}}^2 - {z_{N-1}}^2 + {x_N}^2 + {y_N}^2 + {z_N}^2. \end{aligned} \tag{11}$$

Using matrix notation, we can express the previous system of equations as:

$$Ap = b \tag{12}$$

where

$$A = \begin{bmatrix} 2(x_N - x_1) & 2(y_N - y_1) & 2(z_N - z_1) \\ \vdots & \vdots & \vdots \\ 2(x_N - x_{N-1}) & 2(y_N - y_{N-1}) & 2(z_N - z_{N-1}) \end{bmatrix},$$

$$p = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

and

$$b = \begin{bmatrix} {r_1}^2 - {r_N}^2 - {x_1}^2 - {y_1}^2 - {z_1}^2 + {x_N}^2 + {y_N}^2 + {z_N}^2 \\ \vdots \\ {r_{N-1}}^2 - {r_N}^2 - {x_{N-1}}^2 - {y_{N-1}}^2 - {z_{N-1}}^2 + {x_N}^2 + {y_N}^2 + {z_N}^2 \end{bmatrix}$$

Finally, we obtain a standard least squares problem:

$$\min_p (Ap - b)^T (Ap - b) \tag{13}$$

from which we can obtain a closed-form solution in terms of the pseudo-inverse of matrix A:

$$p = A^+ b = (A^T A)^{-1} A^T b \tag{14}$$

Using this method, the solution p minimizes the root mean square error, what provides better results in case of inaccurate distance measurements. Similarly to the previous methods, in this case we also proceed with the calculations only when at least four anchors are available.

6. Comparative Evaluation

In this section, we report on the performance evaluation of the point-to-sphere ICP algorithm in comparison to the trilateration-based methods reviewed in Section 5. Besides, we evaluate the effect of introducing the pre- and post-processing stages explained in Section 3 for all position estimation approaches described.

For this evaluation, we have used the Pozyx Creator UWB kit (www.pozyx.io/products-and-services/creator), for a total of eight anchors, which have been placed on the walls surrounding the testing area. Further, two different environments have been considered: inside a laboratory and in a noise-prone environment. In the laboratory experiments, we employ $\delta_{min} = 0.05$ (m) and *max_iter* = 200 (both from Algorithm 1), while $\delta_{min} = 0.03$ (m) and *max_iter* = 300 for the noise-prone environment experiments. In all cases, an anchor/range is considered an inlier within RANSAC according to $\tau_{inl} = 0.5$ (m).

These environments, and the experiments carried out, are detailed in the following sections.

6.1. Laboratory Experiments

The laboratory experiments have been carried out within a 10 m × 5 m × 5 m (L × W × H) volume inside the Aerial Robotics Lab, at the University of the Balearic Islands. The eight anchors have been placed on the walls and floor of the laboratory, at heights ranging from 0 to 4 m. This laboratory is equipped with a motion tracking system which is able to provide very accurate motion estimation, and thus can be used as ground truth data for the UWB tag position during the evaluation. For performance assessment purposes, we considered three different trajectories:

- Trajectory 1—a rectangular trajectory of 5 × 2 m, performed at a constant height;
- Trajectory 2—a figure-eight-like trajectory of 5 × 2 m, performed at a constant height; and
- Trajectory 3—a rectangular trajectory of 5 × 2 m changing the height of the tag, where the height was 2.5 m for the two longer transects and 1.5 m for the two shorter transects.

These datasets have been generated by following the intended trajectories and manually holding the UWB tag with motion tracking markers attached to it. For further insight, Table 1 reports on the amount of noise in the tag-anchor ranges as supplied by the UWB kit for the eight anchors and the three different motion paths followed during the laboratory experiments. Toward this end, we determined the discrepancy between the ranges measured by the anchors and the true ranges calculated by means of the available ground truth motion data. The table shows, on an anchor-by-anchor basis, the average discrepancy and the corresponding standard deviation (as statistical measures of the ranges' noise) and the maximum discrepancy (to illustrate worst cases), all for each anchor independently in order to account for favourable/non-favourable anchor placement during the experiments. As can be observed, the average error was up to around 10 cm, while the worst errors reached several meters.

Table 1. Discrepancy between true tag-anchor ranges and measured ranges as supplied by the UWB kit involved in the laboratory experiments. All values are in meters.

Anchor	Trajectory 1			Trajectory 2			Trajectory 3		
	Mean	Std. Dev.	Max.	Mean	Std. Dev.	Max.	Mean	Std. Dev.	Max.
0	0.135	0.508	4.795	0.072	0.061	0.364	0.135	0.391	4.166
1	0.176	0.559	6.502	0.089	0.090	0.396	0.147	0.415	3.994
2	0.064	0.045	0.218	0.075	0.054	0.225	0.118	0.536	6.681
3	0.076	0.067	0.450	0.106	0.076	0.457	0.115	0.256	3.173
4	0.043	0.025	0.111	0.054	0.044	0.231	0.118	0.452	5.636
5	0.060	0.032	0.138	0.092	0.117	0.685	0.050	0.034	0.162
6	0.095	0.059	0.294	0.093	0.073	0.377	0.152	0.663	6.628
7	0.088	0.304	3.483	0.140	0.462	3.693	0.091	0.086	0.376

In the following sections, we make use of the notation described next to refer to the different methods and data:

- The trilateration methods are denoted as *T_RSS*, *T_MIN* and *T_LS*, for, respectively, the RSS-based method, the minimum discrepancy-based method and the least squares-based method;
- The point-to-sphere ICP-based method is referred to as *ICP*;
- The position estimates provided by the Pozyx kit itself are denoted as *POZYX*; and
- The ground truth data supplied by the motion tracking system is labelled as *GT*.

During the laboratory experiments, the position estimation methods are evaluated for three different configurations: (1) standalone configuration (i.e., without pre- and post- filtering), (2) adding the pre-filtering stage, and (3) incorporating both the pre-filtering and the post-filtering stages.

6.1.1. Results Using the Standalone Configuration

Figure 5 shows the position estimation results obtained with the different methods, when these are used in standalone configuration. These results correspond to the rectangular trajectory, while the results provided through Figures 6 and 7 correspond to, respectively, the figure-eight-like trajectory and the rectangular trajectory with changes in height. As can be seen in the three figures, in the laboratory, most of the methods present similar performances. Nevertheless, the T_RSS method leads to considerably noisier position estimates (in the figure, these are provided separately for a better visualization of the position estimates resulting from the rest of methods).

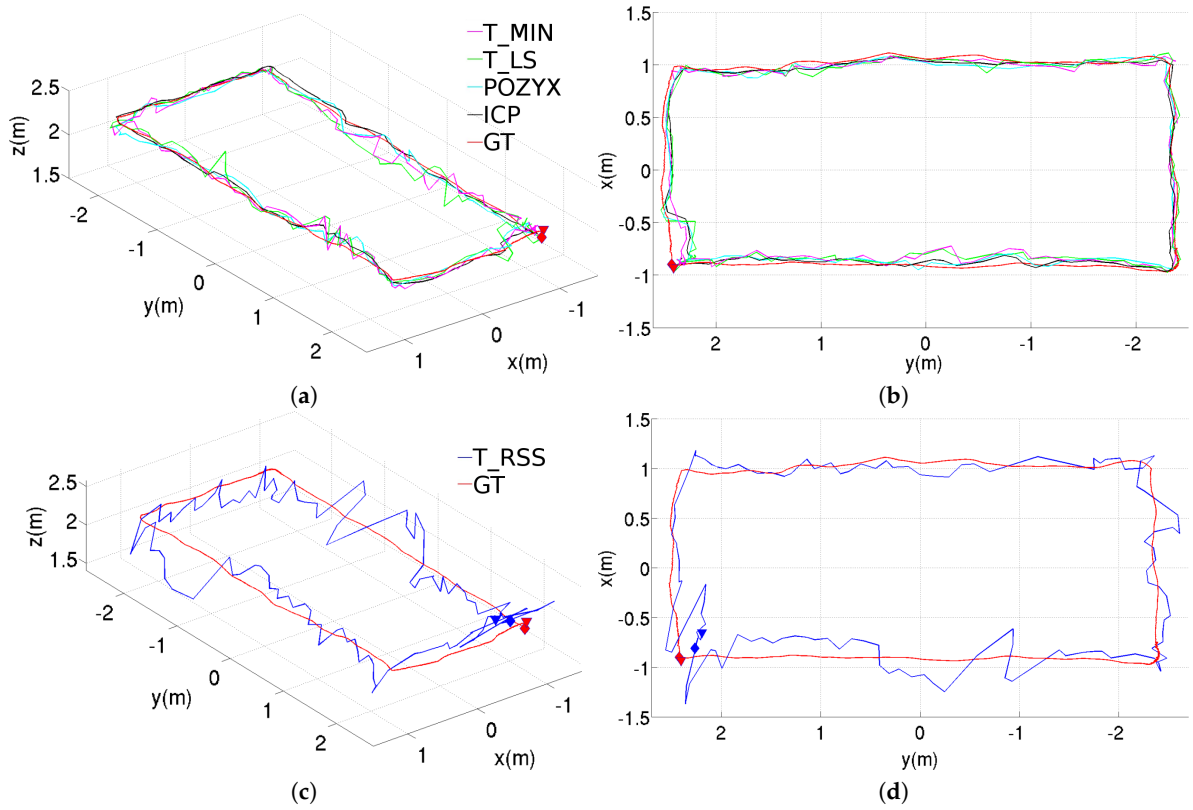


Figure 5. Position estimations provided by the different methods for the rectangular trajectory using the standalone configuration, results for the T_RSS method are shown separately to facilitate the comparison: (**a**) perspective and (**b**) top views for T_MIN, T_LS, POZYX and ICP, (**c**) perspective and (**d**) top views for T_RSS.

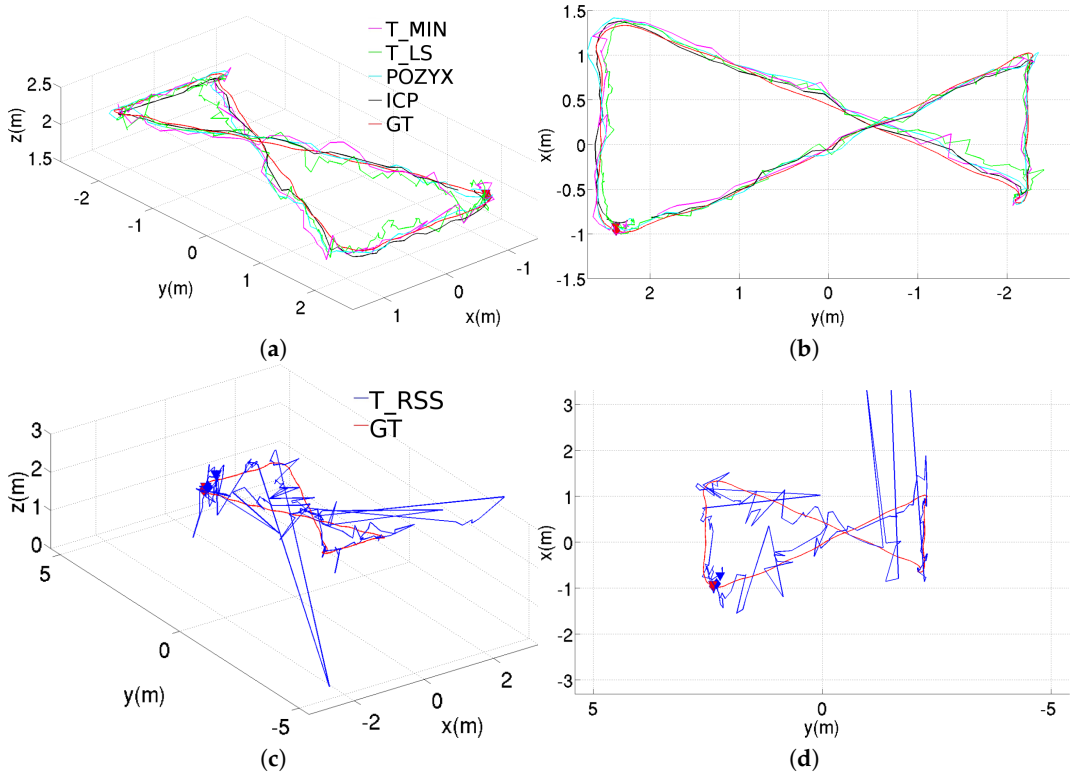


Figure 6. Position estimations provided by the different methods for the figure-eight-like trajectory using the standalone configuration, results for the T_RSS method are shown separately to facilitate the comparison: (**a**) perspective and (**b**) top views for T_MIN, T_LS, POZYX and ICP, (**c**) perspective and (**d**) top views for T_RSS.

Table 2 quantitatively compares all methods for the rectangular trajectory inside the laboratory. The table reports on different metrics about the difference between position estimates and the GT data supplied by the motion tracking system, namely the mean, the standard deviation, the Root-Mean-Square Error (RMSE), the median, and the 90th, 95th, and 98th percentiles of the error [25]. Referring to the results obtained for the standalone configuration of each method, Table 2 shows that the performance of the ICP-based method is comparable to those of POZYX and T_MIN. It is worth noting that ICP leads to the lowest standard deviation and the lowest errors at 95th and 98th percentiles. Among the trilateration-based methods, T_MIN gives rise to the best results, followed by T_LS and, finally, T_RSS. This indicates that the selection of the anchors plays an important role: on the one hand, the subset of anchors which minimizes the trilateration error (used by T_MIN) seems to lead to better performance than considering all the anchors (as in T_LS), probably because the distance to some of the anchors is incorrectly estimated, possibly due to interferences from, for example, metallic elements in the walls and the floor. On the other hand, RSS does not seem to be the best indicator for selecting the anchors.

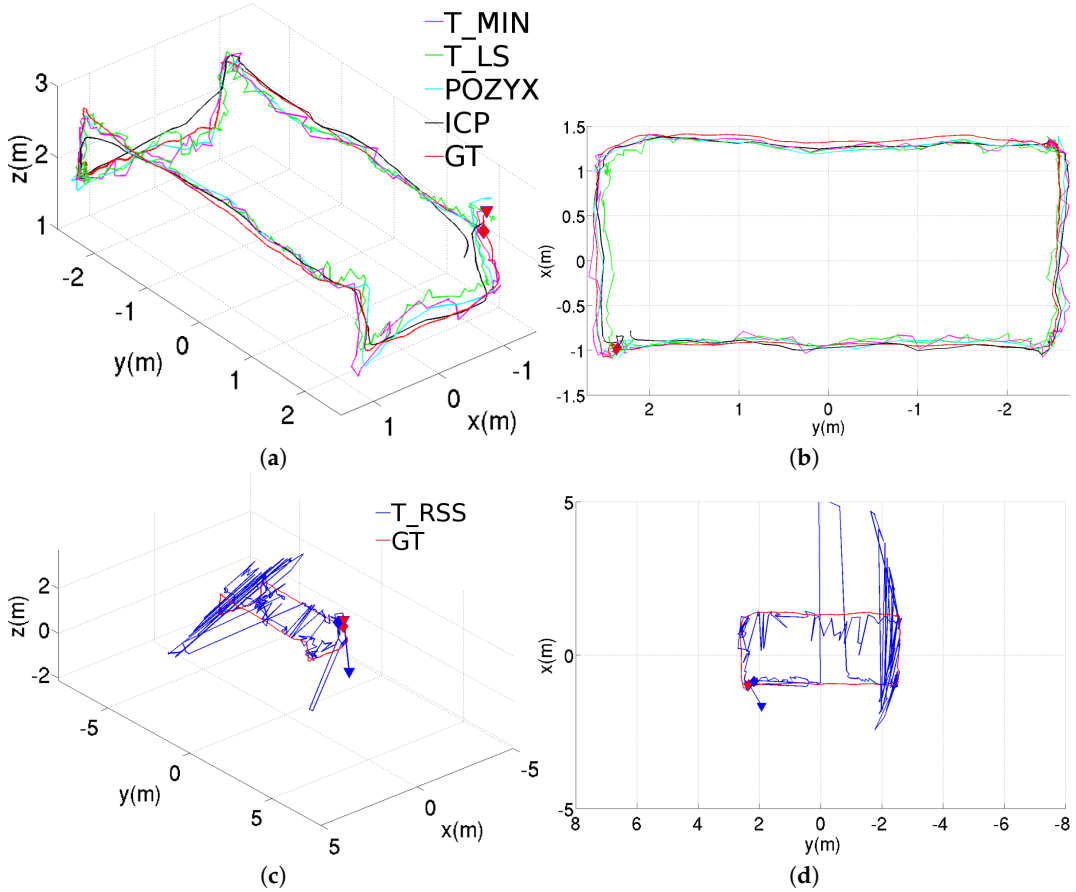


Figure 7. Position estimations provided by the different methods for the rectangular trajectory with changes in height using the standalone configuration, results for the T_RSS method are shown separately to facilitate the comparison: (**a**) perspective and (**b**) top views for T_MIN, T_LS, POZYX and ICP, (**c**) perspective and (**d**) top views for T_RSS.

Similarly, Table 3 shows performance data for the figure-eight-like trajectory. When considering the standalone configurations, we can observe that the ICP-based method leads to the best values for all the metrics considered. Regarding the trilateration-based methods, the performance presented by these methods agree with what we have observed for the previous experiment, what reinforces our hypothesis about the importance of the anchors selection.

Finally, Table 4 reports on the third kind of trajectory, where the rectangular path is followed at different heights. Again, regarding the standalone configurations, the results of the ICP-based method are better than those for the three trilateration-based methods considered in this study.

6.1.2. Results after Adding the Pre-Filtering Stage

Figure 8 shows the estimated trajectories corresponding for the same three paths, but incorporating the pre-filtering stage to filter the anchor-tag distances. As can be observed, all the position estimates provided by the different methods now look smoother, being T_RSS the method which is more favored by the addition of this stage.

Regarding the numerical values of Tables 2–4, we can observe that the use of the pre-filtering stage leads, in general, to lower values of the different metrics for all the methods considered.

Table 2. Performance data for the rectangular trajectory. Values in red denote the three best values for each metric. All values are in meters.

Method	Configuration	Mean	Std. Dev.	RMSE	Median	90th per.	95th per.	98th per.
POZYX	—	0.113	0.053	0.125	0.105	0.203	0.213	0.221
T_RSS	standalone	0.231	0.120	0.260	0.234	0.372	0.479	0.553
	pre-filter	0.191	0.112	0.221	0.191	0.283	0.356	0.425
	pre- & post-filter	0.182	0.087	0.202	0.187	0.288	0.326	0.371
T_MIN	standalone	0.117	0.056	0.129	0.116	0.179	0.207	0.249
	pre-filter	0.118	0.052	0.129	0.119	0.190	0.206	0.229
	pre- & post-filter	0.119	0.048	0.128	0.120	0.183	0.203	0.231
T_LS	standalone	0.126	0.069	0.144	0.117	0.228	0.262	0.305
	pre-filter	0.124	0.048	0.133	0.118	0.193	0.235	0.243
	pre- & post-filter	0.132	0.049	0.141	0.131	0.199	0.246	0.249
ICP	standalone	0.121	0.045	0.129	0.125	0.180	0.189	0.195
	pre-filter	0.121	0.034	0.125	0.121	0.162	0.165	0.170
	pre- & post-filter	0.123	0.039	0.129	0.124	0.172	0.176	0.180

Table 3. Performance data for the figure-eight-like trajectory. Values in red denote the three best values for each metric. All values are in meters.

Method	Configuration	Mean	Std. Dev.	RMSE	Median	90th per.	95th per.	98th per.
POZYX	—	0.110	0.044	0.119	0.114	0.166	0.181	0.191
T_RSS	standalone	0.501	0.830	0.969	0.281	0.827	1.239	4.012
	pre-filter	0.236	0.100	0.256	0.234	0.371	0.389	0.464
	pre- & post-filter	0.239	0.112	0.264	0.240	0.395	0.418	0.442
T_MIN	standalone	0.118	0.068	0.136	0.112	0.213	0.251	0.287
	pre-filter	0.111	0.044	0.119	0.109	0.173	0.183	0.193
	pre- & post-filter	0.112	0.050	0.122	0.111	0.180	0.197	0.211
T_LS	standalone	0.125	0.076	0.147	0.111	0.235	0.255	0.315
	pre-filter	0.122	0.066	0.138	0.100	0.226	0.237	0.243
	pre- & post-filter	0.119	0.070	0.139	0.102	0.217	0.259	0.279
ICP	standalone	0.081	0.043	0.092	0.075	0.139	0.162	0.189
	pre-filter	0.078	0.046	0.090	0.066	0.142	0.176	0.181
	pre- & post-filter	0.090	0.050	0.103	0.091	0.165	0.194	0.200

Table 4. Performance data for the rectangular trajectory with changes in height. Values in red denote the three best values for each metric. All values are in meters.

Method	Configuration	Mean	Std. Dev.	RMSE	Median	90th per.	95th per.	98th per.
POZYX	—	0.103	0.060	0.119	0.100	0.180	0.211	0.250
T_RSS	standalone	0.641	0.990	1.179	0.265	1.880	2.511	3.642
	pre-filter	0.238	0.241	0.339	0.187	0.389	0.499	1.288
	pre- & post-filter	0.196	0.126	0.233	0.176	0.334	0.401	0.548
T_MIN	standalone	0.120	0.080	0.145	0.102	0.212	0.269	0.344
	pre-filter	0.116	0.081	0.142	0.102	0.216	0.258	0.354
	pre- & post-filter	0.129	0.083	0.153	0.112	0.247	0.300	0.327
T_LS	standalone	0.125	0.081	0.149	0.114	0.227	0.290	0.355
	pre-filter	0.116	0.079	0.141	0.105	0.208	0.251	0.352
	pre- & post-filter	0.133	0.084	0.157	0.118	0.259	0.299	0.342
ICP	standalone	0.117	0.064	0.134	0.110	0.185	0.211	0.266
	pre-filter	0.109	0.055	0.122	0.107	0.188	0.198	0.208
	pre- & post-filter	0.126	0.064	0.142	0.116	0.213	0.229	0.255

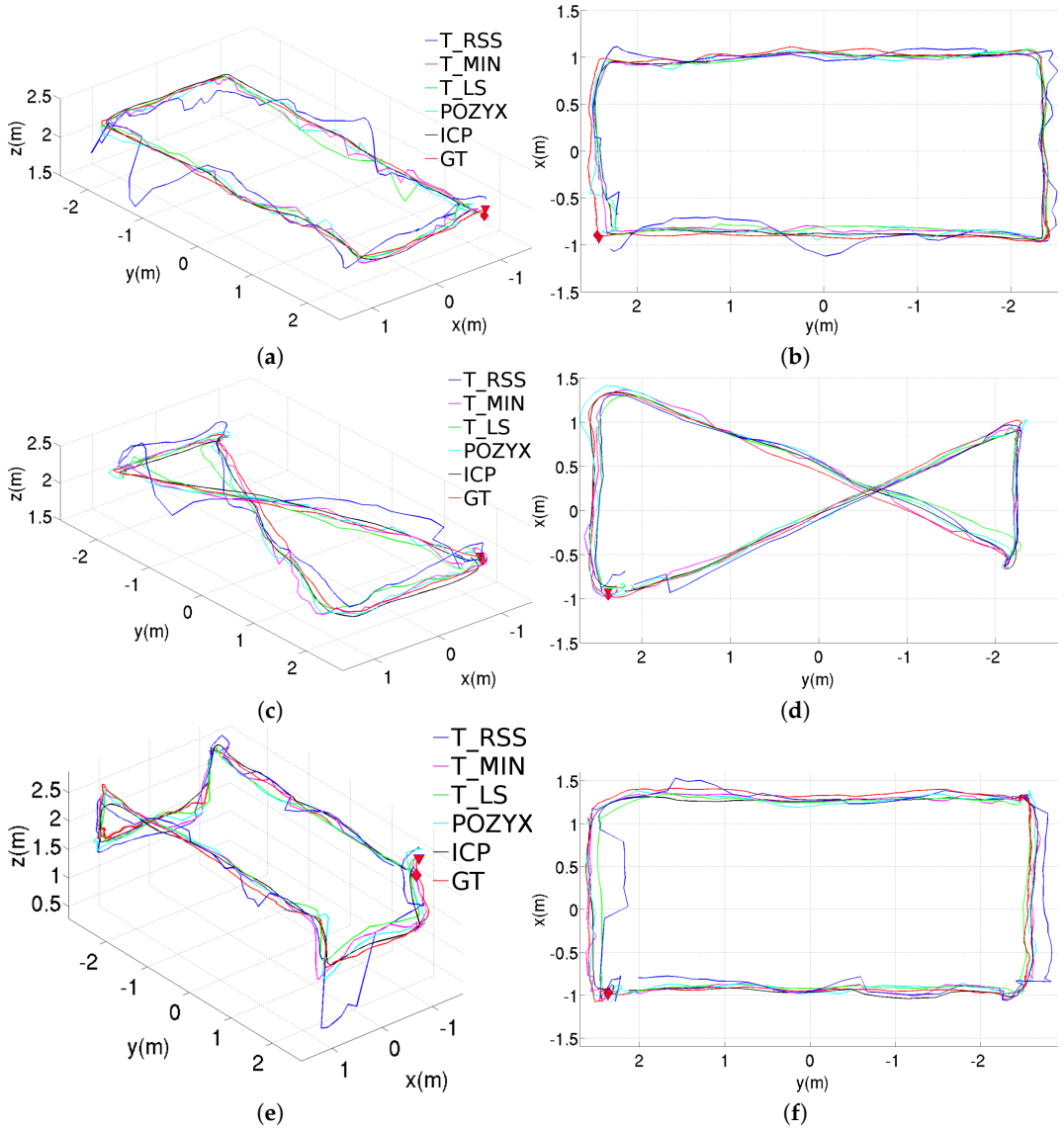


Figure 8. Position estimations provided by the different methods for the three trajectories performed inside the laboratory, results obtained using the pre-filtering stage: (**a**) perspective and (**b**) top views for the rectangular trajectory; (**c**) perspective and (**d**) top views for the figure-eight-like trajectory; (**e**) perspective and (**f**) top views for the rectangular trajectory with changes in height.

6.1.3. Results after Adding the Pre- and Post-Filtering Stages

Figure 9 plots the estimated trajectories for the tree experiments carried out in the laboratory for the full configurations. In comparison with the trajectories plotted in Figure 8, the addition of the post-filtering stage leads to smoother trajectories, as expected.

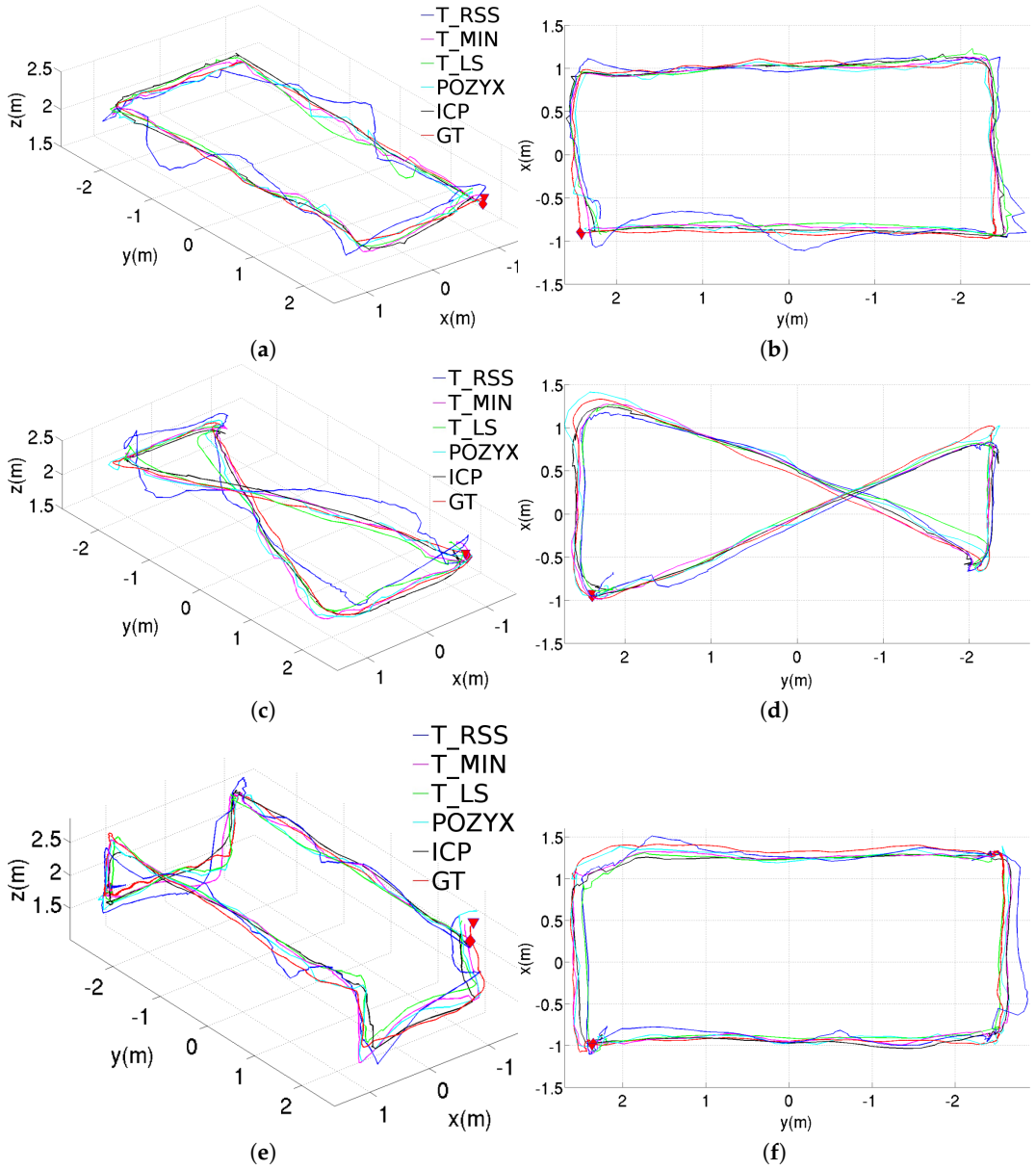


Figure 9. Position estimations provided by the different methods for the three trajectories performed inside the laboratory, results obtained using both the pre- and post-filtering stages: (**a**) perspective and (**b**) top views for the rectangular trajectory; (**c**) perspective and (**d**) top views for the figure-eight-like trajectory; (**e**) perspective and (**f**) top views for the rectangular trajectory with changes in height.

Looking at the numerical performance data shown in Tables 2–4, one can observe that the results obtained after adding the EKF at the output of the pipeline are similar to the ones obtained using only the pre-filtering stage, or even slightly worse in some cases (probably due to the inherent delay introduced by this kind of filters). In any case, the performance of the post-filtering stage is expected to be more notorious in a non-UWB-favorable environment, where the incorporation of data from other sensors (e.g., an IMU) can really benefit position estimators with regard to using purely UWB-based methods.

6.2. Experiments in a Noise-Prone Environment

In this section, we report on some field experiments which have been carried out in one of the cargo holds of a Ro-Ro type vessel (typically intended for transporting cars, trucks, etc.). As a merchant

ship, the cargo holds consist in metallic boxes, so that this kind of environment can be considered as a noise-prone scenario for a UWB positioning system.

Figure 10 plots the results obtained from the different UWB methods during a rectangular trajectory and a figure-eight-like trajectory, both performed inside one of the cargo holds of the aforementioned ship. All methods have been configured to make use of both the pre- and post-filtering stages. The performance exhibited in general for the other configurations (i.e., standalone and using only the pre-filtering stage) can be reported to be of low quality, in accordance to such noisy environment. In all plots, we make use of the trajectories labelled as GT in Figure 10 as reference trajectories for qualitative comparison, since, inside the cargo hold, there was no way to have access to accurate positioning data such as the ones provided by the motion tracking system of our laboratory. These reference trajectories were manually planned by means of a measuring tape and tracked during the experiments using reference lines painted on the floor. In the same way as for the laboratory experiments, the position estimates supplied by the manufacturer's software are also shown in Figure 10 and labelled as POZYX.

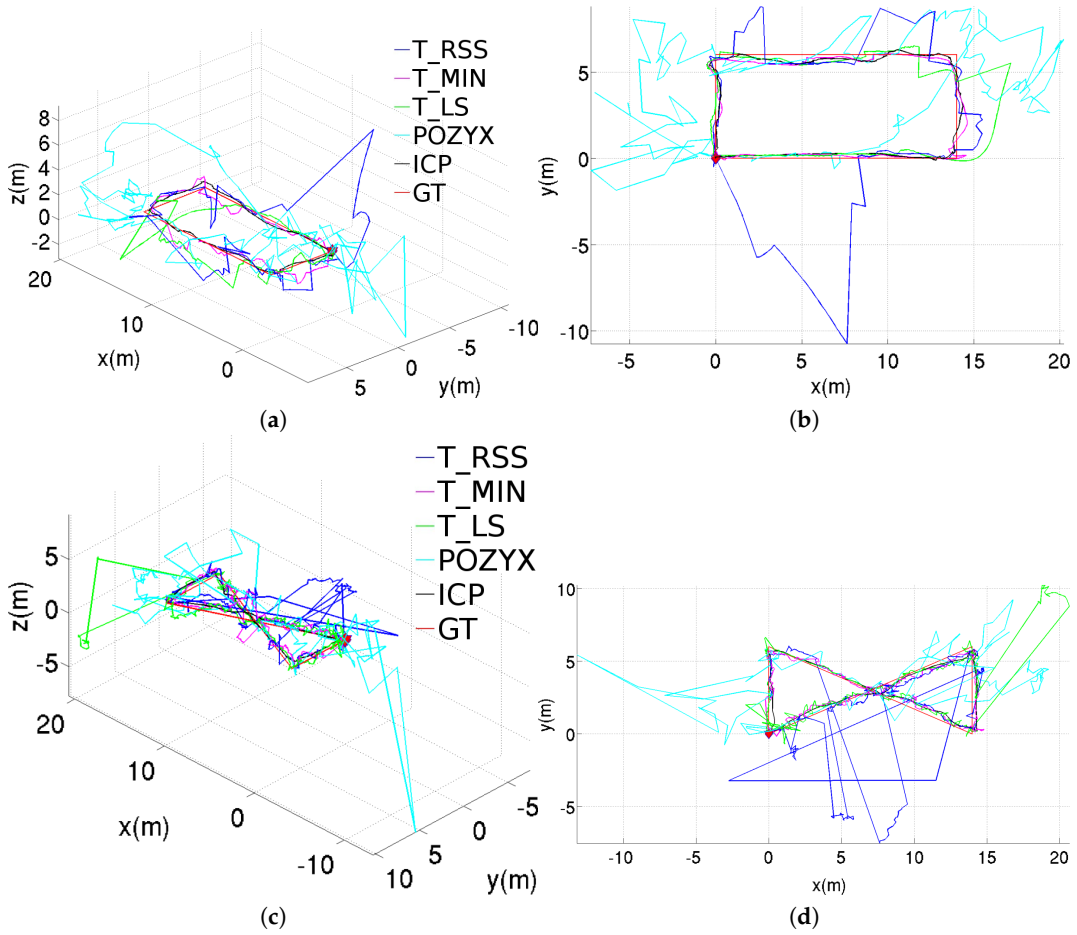


Figure 10. Position estimations provided by the different methods for the two trajectories performed inside the vessel hold, results obtained using both the pre- and post-filtering stages: (**a**) perspective and (**b**) top views for the rectangular trajectory; (**c**) perspective and (**d**) top views for the figure-eight-like trajectory.

As can be observed, the only methods which are able to adhere to the ground truth are T_MIN and ICP, being the latter the method which behaves better. The good performance of these methods is partially due to the good selection of the subset of anchors. The T_LS method is able to follow the trajectory most part of the time, but at certain points suffers from some large deviations due to the use of anchors whose range has been poorly estimated. As happened for the laboratory experiments, T_RSS

gives rise to the worse results, which in this case cannot be sufficiently improved after incorporating the pre- and post-filtering stages. A special mention is made to the quality of the POZYX estimates, which are severely affected by the metallic environment, as already warned by the manufacturer.

7. Conclusions and Future Work

In this work, we have presented a novel method for UWB-based position estimation by means of point-to-sphere ICP. The method has been described and its performance has been compared with alternative position estimators based on trilateration. During the development of the proposed method, and the subsequent comparative evaluation, one of our concerns has been the quality of the anchor-tag distance estimations and thus to establish an adequate anchor selection process. Following with this, we have also considered as part of the performance evaluation the effect of incorporating a pre-processing stage that filters and improves the quality of the range estimates, which are in turn used as input for the position estimation method. Similarly, we have also evaluated the incorporation of a post-processing stage that filters the position estimates by means of non-linear Kalman filtering.

We have reported results for laboratory and field experiments, showing the good performance of the point-to-sphere ICP-based method, which outperforms the alternative position estimation methods considered in the paper. The results also allows us to confirm the importance of the anchors selection step: among the trilateration-based methods, T_MIN has led to the best performance in all experiments, since this method selects the subset of anchors which minimizes the trilateration error. A similar idea is implemented within the point-to-sphere ICP-based method, where RANSAC is used to choose the subset of anchors which provides the lowest global error.

The results of the experiments using the pre-filtering stage indicate that this step is useful to improve the range estimates that are subsequently used by all the methods evaluated. On the other side, the post-filtering stage based on an EKF has proved useful when the UWB devices are operating within a noisy environment, where data provided by other sensors can contribute to obtain more accurate position estimates.

Regarding the computational cost of the ICP-based approach, we have observed that, for the configurations we have considered, convergence is attained after a few iterations—around 50 if the previous estimate is used, and less than 200 irrespectively of the starting estimate—, what, in a standard computer, means execution times of the order of milliseconds, a time-lapse comparable to the computation time of the other methods involved in the comparison. Increasing the number of anchors will make the computational cost increase as well, although normally the bottleneck is rather on the time needed to collect the ranges from the different anchors instead of on the calculations.

Like any other UWB-based positioning method, the point-to-sphere ICP-based method can be affected by poor positioning of the anchors, what in turn can result in an ill-conditioned problem. Since our method is based on ICP, it may need some additional iterations to converge when the anchors are not properly situated. In this case, the update in the position estimate between iterations can be rather small, so that, depending on the stopping conditions used in the ICP loop, the algorithm might decide that convergence conditions are met and stop prematurely, giving rise to inaccuracies in the position estimates.

As for future work, we plan to improve the estimation of the tag's height by tolerating better the lack of variation in the anchors heights, by means of the incorporation of additional sensors into the data fusion step. In particular, we are concerned with the use of the point-to-sphere ICP-based method for estimating the 3D position of a Micro-Aerial Vehicle (MAV) and, hence, the enhancements in height estimation can greatly contribute to improving the performance of the full system as a whole. The integration of the point-to-sphere ICP into a SLAM solution which is currently under development is another item which will be part of future–though relatively immediate–work.

Author Contributions: Conceptualization, F.B.-P. and A.O.; methodology, F.B.-P. and A.O.; software and investigation, F.B.-P.; formal analysis and validation, F.B.-P. and A.O.; writing—original draft preparation, F.B.-P.; writing—review and editing, A.O.; supervision, A.O.; project administration, A.O.; funding acquisition, A.O. All authors have read and agreed to the published version of the manuscript.

Funding: This work is partially supported by EU-H2020 projects BUGWRIGHT2 (GA 871260) and ROBINS (GA 779776), PGC2018-095709-B-C21 (MCIU/AEI/FEDER, UE), and PROCOE/4/2017 (Govern Balear, 50% P.O. FEDER 2014-2020 Illes Balears). This publication reflects only the authors views and the European Union is not liable for any use that may be made of the information contained therein.

Acknowledgments: We would like to thank Marc Pozo for his contributions to the work here presented.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Gu, Y.; Lo, A.; Niemegeers, I. A Survey of Indoor Positioning Systems for Wireless Personal Networks. *IEEE Commun. Surv. Tutor.* **2009**, *11*, 13–32. [CrossRef]
2. Al-Ammar, M.A.; Alhadhrami, S.; Al-Salman, A.; Alarifi, A.; Al-Khalifa, H.S.; Alnafessah, A.; Alsaleh, M. Comparative Survey of Indoor Positioning Technologies, Techniques, and Algorithms. In Proceedings of the International Conference on Cyberworlds, Santander, Spain, 6–8 October 2014; pp. 245–252.
3. Liu, H.; Darabi, H.; Banerjee, P.; Liu, J. Survey of Wireless Indoor Positioning Techniques and Systems. *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.* **2007**, *37*, 1067–1080. [CrossRef]
4. Alarifi, A.; Al-Salman, A.; Alsaleh, M.; Alnafessah, A.; Al-Hadhrami, S.; Al-Ammar, M.A.; Al-Khalifa, H.S. Ultra Wideband Indoor Positioning Technologies: Analysis and Recent Advances. *Sensors* **2016**, *16*, 707. [CrossRef] [PubMed]
5. Guvenc, I.; Chong, C. A Survey on TOA Based Wireless Localization and NLOS Mitigation Techniques. *IEEE Commun. Surv. Tutor.* **2009**, *11*, 107–124. [CrossRef]
6. Chóliz, J.; Eguizabal, M.; Hernandez-Solana, A.; Valdovinos, A. Comparison of Algorithms for UWB Indoor Location and Tracking Systems. In Proceedings of the Vehicular Technology Conference, Yokohama, Japan, 15–18 May 2011.
7. Cheng, G. Accurate TOA-Based UWB Localization System in Coal Mine Based on WSN. *Phys. Procedia* **2012**, *24*, 534–540. [CrossRef]
8. García, E.; Poudereux, P.; Hernández, A.; Ureña, J.; Gualda, D. A Robust UWB Indoor Positioning System for Highly Complex Environments. In Proceedings of the IEEE International Conference on Industrial Technology, Seville, Spain, 17–19 March 2015; pp. 3386–3391.
9. Liu, J.; Wang, Q.; Xiong, J.; Huang, W.; Peng, H. Indoor and Outdoor Coperative Real-Time Positioning System. *J. Theor. Appl. Inf. Technol.* **2013**, *48*, 1066–1073.
10. Subramanian, A. UWB Linear Quadratic Frequency Domain Frequency Invariant Beamforming and Angle of Arrival Estimation. In Proceedings of the Vehicular Technology Conference, Dublin, Ireland, 22–25 April 2007; pp. 614–618.
11. Xu, J.; Ma, M.; Law, C.L. AOA Cooperative Position Localization. In Proceedings of the IEEE Global Telecommunications Conference, New Orleans, LA, USA, 30 November–4 December 2008.
12. Gigl, T.; Janssen, G.; Dizdarevic, V.; Witrisal, K.; Irahhauten, Z. Analysis of a UWB Indoor Positioning System Based on Received Signal Strength. In Proceedings of the Workshop on Positioning, Navigation and Communication, Hannover, Germany, 22–22 March 2007; pp. 97–101.
13. Leitinger, E.; Fröhle, M.; Meissner, P.; Witrisal, K. Multipath-Assisted Maximum-Likelihood Indoor Positioning using UWB Signals. In Proceedings of the IEEE International Conference on Communications Workshops, Sydney, Australia, 10–14 June 2014; pp. 170–175.
14. Kodippili, N.; Dias, D. Integration of Fingerprinting and Trilateration Techniques for Improved Indoor Localization. In Proceedings of the Wireless and Optical Communications Networks, Colombo, Sri Lanka, 6–8 September 2010; pp. 1–6.
15. Zhu, D.; Yi, K. EKF Localization based on TDOA/RSS in Underground Mines using UWB Ranging. In Proceedings of the IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC), Xi'an, China, 14–16 September 2011; pp. 1–4.
16. Zwirello, L.; Schipper, T.; Harter, M.; Zwick, T. UWB Localization System for Indoor Applications: Concept, Realization and Analysis. *J. Electr. Comput. Eng.* **2012**, *2012*, 849638. [CrossRef]

17. Shi, G.; Ming, Y. Survey of Indoor Positioning Systems Based on Ultra-Wideband (UWB) Technology. In *Wireless Communications, Networking and Applications*; Zeng, Q.A., Ed.; Springer: New Delhi, India, 2016; pp. 1269–1278.
18. Yassin, A.; Nasser, Y.; Awad, M.; Al-Dubai, A.; Liu, R.; Yuen, C.; Raulefs, R.; Aboutanios, E. Recent Advances in Indoor Localization: A Survey on Theoretical Approaches and Applications. *IEEE Commun. Surv. Tutor.* **2017**, *19*, 1327–1346. [CrossRef]
19. Saraiva Campos, R.; Lovisolo, L. *RF Positioning: Fundamentals, Applications, and Tools*; Artech House Publishers: Norwood, MA, USA, 2015.
20. Antoniou, A. *Digital Signal Processing: Signals, Systems, and Filters*; McGraw-Hill Education: New York City, NY, USA, 2006.
21. O'Haver, T. *A Pragmatic Introduction to Signal Processing with Applications in Scientific Measurement*; Department of Chemistry and Biochemistry, The University of Maryland at College Park: College Park, MD, USA, 2020. Available online: https://terpconnect.umd.edu/~toh/spectrum/IntroToSignalProcessing2020.pdf (accessed on 30 September 2020).
22. Dardari, D.; Closas, P.; Djuric, P.M. Indoor Tracking: Theory, Methods, and Technologies. *IEEE Trans. Veh. Technol.* **2015**, *64*, 1263–1278. [CrossRef]
23. Censi, A. An ICP Variant using a Point-to-Line Metric. In Proceedings of the IEEE International Conference on Robotics and Automation, Pasadena, CA, USA, 19–23 May 2008.
24. Fischler, M.A.; Bolles, R.C. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Commun. ACM* **1981**, *24*, 381–395. [CrossRef]
25. Potortì, F.; Park, S.; Jiménez-Ruiz, A.; Barsocchi, P.; Girolami, M.; Crivello, A.; Lee, S.; Lim, J.; Torres-Sospedra, J.; Seco, F.; et al. Comparing the Performance of Indoor Localization Systems through the EvAAL Framework. *Sensors* **2017**, *17*, 2327. [CrossRef] [PubMed]

MaRS: A Modular and Robust Sensor-Fusion Framework

Christian Brommer[1], Roland Jung[2], Jan Steinbrener[1], and Stephan Weiss[1]

***Abstract*—State-of-the-art recursive sensor filtering frameworks allow the fusion of multiple sensors tailored to a specific problem but do not allow a dynamic and efficient introduction of additional sensors during runtime - an important feature to enable long-term missions in dynamic environments. This paper presents a robust, modular sensor-fusion framework that enables the addition and removal of sensors at runtime. These sensors could be not *a priori* known to the system. The framework handles the complexity of system and sensor initialization, measurement updates, and switching of asynchronous multi-rate sensor information with sensor self-calibration in a truly modular and generic design. In addition, the framework can handle delayed measurements, out-of-sequence updates, and can monitor sensor health. The introduced *true-modularity* is based on covariance segmentation to allow the isolated (i.e., modular) processing of propagation and updates on a per-sensor basis. We show how crucial properties of the overall state covariance can be maintained as naive implementation of such a modularization would invalidate the covariance matrix. We evaluate our framework for a precision landing scenario switching between combinations of GNSS, barometer, and vision measurements. Tests are performed in simulation and in real-world scenarios to show the validity of the introduced method. The presented framework will be open-sourced and made available online to the community.**

***Index Terms*—Sensor Fusion, State-Estimation, Modularity, Autonomous Navigation**

I. Introduction

STATE-Estimation is an essential part of robotics and engineering. The accurate knowledge of the location of a robotic platform in the world is crucial for navigation, control, and manipulation. Dedicated estimators are repeatedly developed, and most existing approaches are tailored to accomplish a specific task on specific hardware under specific conditions, limiting re-usability if the scenario, sensor suite, or the platform changes. Current open-source and state-of-the-art Extended Kalman Filter (EKF) frameworks start to address this issue, but they are designed to handle a setup of sensors that is pre-defined during the compilation time or start-up phase of the filter. The reference frames of additional sensors are also often pre-defined and are not dynamically adapted to the current situation. Such frameworks do not allow the initialization of sensors during runtime, especially if the sensor definition is not *a priori* known to the system. This limits their application to static hardware configurations and does not support novel applications with modular platforms that can be extended during runtime with sensor modules not known to the core framework (e.g., connectable snake robots or humanoids with exchangeable end-effectors).

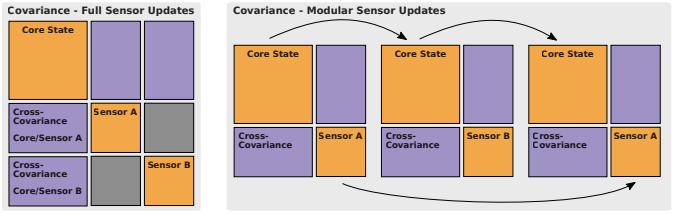


Fig. 1. True modularity. Left: Structure of a state-of-the-art multi-sensor fusion approach. The covariance matrix is always fully updated. Right: Modular segmentation of the update and propagation process. Only the core and currently active sensor state covariance is used to perform the update.

A major challenge is that additional sensors require additional calibration states because they are rarely aligned with the robot's estimated body frame, nor are they intrinsically calibrated. Therefore, the number of calibration-states increases with the number of sensors. An increasing number of states requires more operations to perform the estimation (e.g., for propagation and updates in filter-based estimates). The processing time of a naive estimator increases *cubically* $\mathcal{O}(n^3)$ with the number of sensors n due to matrix multiplications. This effect is even worse for delayed and out-of-sequence measurements in a multi-sensor system because delayed signals trigger numerous re-computation steps should the estimator remain credible. Hardware synchronization can mitigate this issue, but it may not always be possible (particularly with dynamic sensor rates). While non-recursive filter formulations (e.g., graph optimization-based) have been shown to be able to initialize previously unknown sensors during runtime, their computational load makes them ill-suited for execution on resource-constrained platforms such as Unmanned

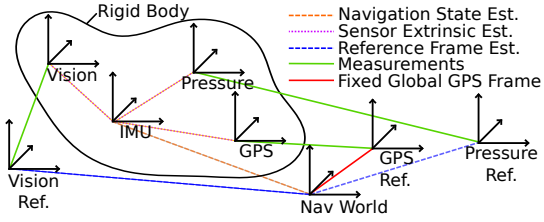


Fig. 2. Estimated state variables with self-calibration are shown as dotted lines; measurements and one fixed global reference frame (e.g. Global Navigation Satellite System (GNSS)) are shown as continuous lines.

Manuscript received: August, 11, 2020; Revised October, 10, 2020; Accepted September, 28, 2020.

This paper was recommended for publication by Editor Dan Popa upon evaluation of the Associate Editor and Reviewers' comments. The research leading to these results was supported by the ARL within the BAA W911NF-12-R-0011 under grant agreement W911NF-16-2-0112, the University of Klagenfurt within the doctoral school KPK-NAV, and the European Commission under grant agreement 871260 - BugWright2.

[1] Control of Networked Systems Group, University of Klagenfurt, Austria. E-Mail: { christian.brommer, jan.steinbrener, stephan.weiss }@ieee.org

[2]Karl Popper School on Networked Autonomous Aerial Vehicles, University of Klagenfurt, Austria. E-Mail: roland.jung@ieee.org

Post-print version, accepted September/2020, DOI follows ASAP ©IEEE.

Aerial Vehicles (UAVs). Here, we present a recursive, unified, and modular multi-sensor-fusion framework with support for efficient, multiple asynchronous updates resulting in constant complexity independent of the number of sensors. It further provides generalized interfaces that allow an easy exchange of components between projects and collaborators. The contributions of this work are the following:

- The design and implementation of a truly modular multi-sensor fusion framework as a recursive filter with the capability of *on-line* addition of previously undefined sensors with delay and asynchronous measurements. These sensors will be self-initialized and self-calibrated based on their extrinsic states, which are added to the system.
- A novel approach for correct covariance segmentation, which preserves the properties of a covariance matrix throughout the isolated processing of individually joined segments. This renders the framework both consistent and computationally tractable on constrained platforms: the complexity depends only linearly on the number of sensors and the propagation step constant/independent of the number of sensors.
- Statistically relevant tests in simulations and verification of the proposed framework with real data.

On-line sensor addition is achieved by decoupling the navigation states (e.g., position, velocity, and attitude for a mobile system) from calibration states of individual sensors (e.g., the transformation between sensor and agent body frame). This allows the introduction of a *sensor-update-module* during runtime (e.g., through independently launchable nodelets in a Robot Operating System (ROS) environment). Our design also accounts for offsets between global and local references maintaining smooth state evolution upon inclusion of a new reference frame. To maintain consistency despite this modularization, we introduce a covariance segmentation approach, which retains the filter's credibility, to correctly propagate isolated covariance components by maintaining fundamental properties of covariance matrices. This is crucial for the correct application of propagation and update steps. A naive approach would result in ill-conditioned covariances. The presented approach reduces the complexity of updates and renders the processing time of the propagation as well as the update phase constant and independent of the number of sensors.

The approach provides continuous self-calibration (see Fig. 2) while keeping maximum flexibility at a low computational cost. We validate the proper propagation of information (i.e., observability properties), the credibility of the overall approach, and the performance with a statistically relevant number of simulations. We illustrate the feasibility of our approach for computationally-constrained platforms with real-world experiments and an UAV. The experiments are performed with Inertial Measurement Unit (IMU) driven dynamics; however, different dynamic formulations are possible, and the framework can, of course, be deployed on other platforms, not limiting the contribution to UAVs.

II. Related Work

State estimation with pre-defined sensor suites and complementing calibration states, including self-calibration and delay compensation with multi-sensor rates, have been studied thoroughly in literature. The Single Sensor Fusion framework (SSF) presented by [1] covers the topics of online self-calibration and accurate handling of sensor delays (out-of-sequence updates). An extended version of SSF was used by [2] in a multi-sensor setup for long-duration autonomy. The Multi-Sensor Fusion framework (MSF) has been introduced by [3], and [4] has presented similar work that details relative and absolute sensor updates using local vision updates and global position information as an example. While both frameworks, SSF, and MSF can accommodate sensor outages, the work of [5] extended the MSF framework and studied the topic of online sensor initialization and switching based on sensor availability and health metrics.

[6] introduced a method for the handling of delayed measurements, designed for computationally-constrained embedded systems. [7] presented a generalized extended Kalman filter implementation based on ROS. This framework defines its sensor structure during startup but does not allow modification of the setup during runtime. Sensor measurements are assumed to be expressed in the robot's origin. The framework does not introduce sensor calibration states and does not perform online self-calibration. It neither estimates Gyroscopic biases for the IMU, and the process noise of the system is tuned by hand. The method presented here goes further and allows the incorporation and removal of sensors that are not *a priori* known to the system during runtime by decoupling the core states from the sensor states. This allows a decentralized yet tightly-coupled processing of sensor information.

The work of [8] describes the state-of-the-art centralized and decentralized sensor-fusion for driver-assistance systems and discusses the current challenges of this approach. In short, centralized approaches allow tightly-coupled estimation but require high communication bandwidth. Existing approaches are also hard to extend and require extensive workload for the implementation of new sensor instances. State-of-the-art decentralized systems make use of loosely-coupled sensor integration, which has the disadvantage of inconsistencies because of inadequate handling of the sensor and core states cross-covariances. The focus of the presented work relies specifically on modularity and consistency/credibility; however, it will also benefit the development of decentralized systems (e.g., swarms). It allows tightly-coupled sensor-fusion with reduced bandwidth between system components because the states of a sensor instance can be stored and processed locally. It further simplifies the development and extension of systems by minimizing the workload for integrating new sensors and allows online retrofitting.

The work of [9] and [10] studied the modularization of multi-sensor fusion and presented a vector graph-based method which employs a real-time batch optimization process. Both authors' work focuses on the optimal and the minimal selection as a subset of the given sensor suit and covers observability for sensor selection. The authors perform plug and play experiments by abstracting the sensor to avoid the direct use of physical measurements. It is important to note that the use of vector graph-based methods is limited in terms of scalability, especially in combination with computationally-

constrained resources. The presented work is different because it focuses on a truly modular approach with a recursive filtering technique. The presented approach minimizes the size of covariance matrices, reducing the number of mathematical operations and increasing performance/scalability. To the best of our knowledge, no truly modular recursive filter approach, as presented in this paper, has been reported in the literature.

III. Method

A. Truly Modular Sensor Fusion

Recursive filters such as EKFs require all states and covariances during the update and propagation phase. A typical setup of a filter for estimating a system, defines core states that describe the essential variables of a platform that are necessary for propagation and to perform controls. We use the core state definition established by [1] and shown by Equation (1). The essential core states are the translation from the world to the IMU/body frame ${}_W\mathbf{p}_{WI} \equiv \mathbf{p}_{WI}$ expressed in the world frame, velocity $\mathbf{v}_{WI}$, the orientation of the IMU in the world frame $\mathbf{q}_{WI}$, gyroscopic bias $\mathbf{b}_\omega$ and accelerometer bias $\mathbf{b}_a$, with ${}_C\mathbf{p}_{AB} = \mathbf{R}_{(\mathbf{q}_{CA})}{}_A\mathbf{p}_{AB}$ and $\mathbf{R}_{(\mathbf{q}_{CA})} \equiv \mathbf{R}_{CA}$.

$$\mathbf{X}_\mathsf{C} = \left[\mathbf{p}_{WI}^\mathsf{T}, \mathbf{v}_{WI}^\mathsf{T}, \mathbf{q}_{WI}^\mathsf{T}, \mathbf{b}_\omega^\mathsf{T}, \mathbf{b}_a^\mathsf{T}\right]^\mathsf{T} \tag{1}$$

Generally, for mobile systems, the state and covariance can be propagated by an IMU driven and time-dependent dynamic model. The following differential equations govern the state-dynamics, with $\mathbf{\Omega}(\boldsymbol{\omega})$ being the right side Quaternion multiplication matrix for $\boldsymbol{\omega}$, gravity expressed in the world frame $\mathbf{g}$, and $\mathbf{n}_{\mathbf{b}_a}$, $\mathbf{n}_{\mathbf{b}_\omega}$ being zero mean white Gaussian noise of the accelerometer and gyroscope measurements.

$$\dot{\mathbf{p}}_{WI} = \mathbf{v}_{WI} \tag{2}$$

$$\dot{\mathbf{v}}_{WI} = \mathbf{R}_{(\mathbf{q}_{WI})}(\mathbf{a}_m - \mathbf{b}_a - \mathbf{n}_a) - \mathbf{g} \tag{3}$$

$$\dot{\mathbf{q}}_{WI} = \frac{1}{2}\mathbf{\Omega}(\boldsymbol{\omega}_m - \mathbf{b}_w - \mathbf{n}_w)\mathbf{q}_{WI} \tag{4}$$

$$\dot{\mathbf{b}}_\omega = \mathbf{n}_{\mathbf{b}_\omega},\ \dot{\mathbf{b}}_a = \mathbf{n}_{\mathbf{b}_a}. \tag{5}$$

If the system provides additional sensors, they are likely not aligned with the center of the platform. The extrinsics of individual sensors can be implemented as calibration states and may be estimated online. Given a system with e.g., two additional sensors S_1 and S_2 the core state can be augmented with their extrinsic calibration states accordingly

$$\mathbf{X} = \left[\mathbf{X}_\mathsf{C};\ \mathbf{X}_{\mathsf{S}_1};\ \mathbf{X}_{\mathsf{S}_2}\right]. \tag{6}$$

The observation of additional sensors introduces cross-correlations between the core and sensor states, resulting in cross-covariances in the covariance matrix $\mathbf{P}$. The joint covariance matrix after sensor observations is

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_\mathsf{C} & \mathbf{P}_{\mathsf{CS}_1} & \mathbf{P}_{\mathsf{CS}_2} \\ \mathbf{P}_{\mathsf{S}_1\mathsf{C}} & \mathbf{P}_{\mathsf{S}_1} & 0 \\ \mathbf{P}_{\mathsf{S}_2\mathsf{C}} & 0 & \mathbf{P}_{\mathsf{S}_2} \end{bmatrix}, \tag{7}$$

with $\mathbf{P}_{\mathsf{CS}_2} = (\mathbf{P}_{\mathsf{S}_2\mathsf{C}})^\mathsf{T}$ and sensors S_1 and S_2 assumed to be independent to each other.

Starting from this structure, we propose a segmentation approach that isolates the core and sensor covariance components. Performing the propagation on the isolated core states reduces the size of the covariance matrix, minimizing computational effort. A possible filtering routine with covariance segmentation for a two-sensor scenario is shown by Figure 3.

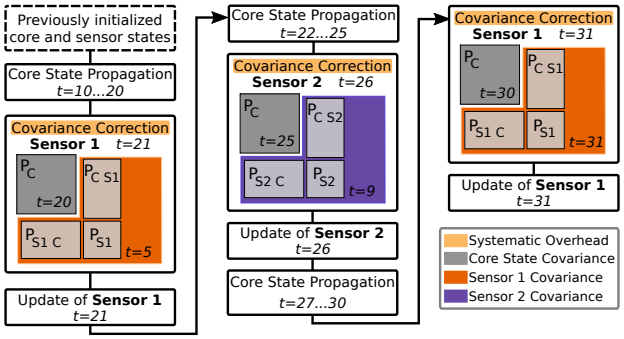


Fig. 3. The figure shows a representative sequence for the truly modular filter process with two sensors. The covariance correction element performs two steps, first the generation of the state-transition blocks for the propagation of the sensor covariance, and second, the Eigenvalue covariance correction. Please note that the sensor covariance and cross-covariance (orange and purple boxes) are stored at the time of their update and do not evolve over time until their next update. Sensor measurements occur at t = 21, 26, and 31s.

The scenario describes the filtering procedure for a time section between t=0 s and 31 s. The filter is initialized at t=0 s, two sensors have been added and initialized during runtime, S_1 at t=5 s, and S_2 at t=9 s. The covariance handling is performed as follows: The core covariance and states are propagated separately in the time t=10 s and 20 s. Sensors S_1 and S_2 have not been part of this propagation. S_1 provides a measurement at t=21 s; at this point, the latest sensor covariance $\mathbf{P}_{\mathsf{S}1}$ and cross-covariance with the core states $\mathbf{P}_{\mathsf{CS}1}$ at t=5 s (orange segment) is joined with the latest core state covariance $\mathbf{P}_\mathsf{C}$ at t=20 s, and the update for t=21 s is performed. The sensor covariance $\mathbf{P}_{\mathsf{S}1}$ and sensor/core cross-covariance $\mathbf{P}_{\mathsf{CS}1}$ is separated from the core covariance $\mathbf{P}_\mathsf{C}$ afterwards and stored until the next update of S_1 at t=31 s.

The core state is propagated until the measurement update of S_2 at t=26 s, and the latest covariance segments $\mathbf{P}_\mathsf{C}$ of the core (t=25 s) and S_2 (t=9 s) are used for the update at t=26 s. At this point, the process continues with the same procedures: the propagation of the isolated core covariance $\mathbf{P}_\mathsf{C}$ and the update of individual sensor states with the core. One important aspect is that measurement updates of one sensor are separated from the state of any other sensor (see Eq. (7)). This is one of the key components of the introduced covariance segmentation that allows true modularity. The routine shows that any sensor can be added or removed without interfering with other sensor covariances. We call this *truly modular* since only the minimal representation of the current state, and covariance segments are joined for a particular update or propagation.

B. Consistent Truly Modular Covariance Estimation

The described covariance segmentation introduces two problems. Due to this approach, two or more sensors are never updated in the same step, and thus, no cross-covariance terms between sensors are generated. We make the design choice that the cross-covariance between any sensor states are zero (see Eq. (7)). Thus, all additional auxiliary states of individual sensors are independent, but the covariance of an individual sensor and its cross-covariance with the core state is maintained (see Fig. 1). An intuitive physical example can be

given by using a 3DoF magnetometer and 3DoF GNSS sensor. The rotational calibration of the magnetometer with respect to the IMU and the translation of the GNSS with respect to the IMU do not have a physical relation. Although these cross-covariance do exist from an analytical point of view; they are negligible as the experiments in Section IV-A validate. Thus, negligible losses in accuracy allow vast performance improvements given the gained recursive modularity.

The second problem is the validity of the covariance matrix for the joined covariance segments. The covariance segments were calculated for different points in time and do not include the same amount of sample data, which leads to non-positive-semidefiniteness. Thus, we propose pre-update routines to reintroduce the information that was not handled during the propagation and individual update phases. We then select the closest valid (positive-semidefinite) covariance matrix from this augmented matrix.

1) Propagation: The information that each isolated sensor component was missing during the propagation phase can be fetched and propagated forward consistently to the current update step. In [11] it is shown that the cross-covariances can be independently propagated using the *state-transition matrix series*. The state-transition matrix series $\mathbf{\Phi}(m,n)$ between two time instances $\mathrm{t}(m)$ and $\mathrm{t}(n)$ is defined as

$$\mathbf{\Phi}(m,n) = \mathbf{\Phi}_n \mathbf{\Phi}_{n-1} \ \dots \ \mathbf{\Phi}_m \text{ with } \mathrm{t}(m) < \mathrm{t}(n), \quad (8)$$

with $\mathbf{\Phi}_k$ as the discrete state-transition matrix $\mathbf{\Phi}_{k|k-1}$ that encodes the state dynamic, evaluated based on the system input, and integrated for the propagation step $\delta\mathrm{t} = \mathrm{t}(k) - \mathrm{t}(k-1)$. The corresponding cross-covariance $\mathbf{P}_{\mathsf{CS}}$ between core C and sensor S can be propagated from the time instance $\mathrm{t}(m)$ until $\mathrm{t}(n)$ with

$$\mathbf{P}_{\mathsf{CS},n(-)} = \mathbf{\Phi}_{\mathsf{C}}(m,n)\mathbf{P}_{\mathsf{CS},m}\mathbf{\Phi}_{\mathsf{S}}(m,n)^{\mathsf{T}}, \quad (9)$$

$\mathbf{\Phi}_{\mathsf{C}}(m,n)$ being the state-transition matrix series of the core and $\mathbf{\Phi}_{\mathsf{S}}(m,n)$ for the sensor state. Storing a history of state-transition matrices, allows the generation of a state-transition matrix series to propagate sensor covariance and cross-covariance between core and sensor states. The sensor covariance $\mathbf{P}_{\mathsf{CS}}$ inherits the information that was not introduced while the core $\mathbf{P}_{\mathsf{C}}$ was propagated in isolation. This *on-demand* information inheritance allows to only compute the core states at each propagation step, keeping this step at constant complexity independent of the number of sensors, but requires a computational spike for the pre-update step.

2) Updates: [11] also showed that indirect observations affect core and sensor states because of cross-correlations between the core and individual sensor states. This means that a sensor observation, e.g., provided by S_1, results in an update and correction of states correlated with the core state e.g., those of S_2. Due to this, sensor covariances can usually not be removed and reintroduced directly. Considering Figure 3, the removal of a previously introduced sensor S_1 at t=21 s and its reintroduction at t=31 s after other measurement updates have been performed (S_2 at t=26 s), invalidates the covariance matrix, which becomes non-positive semidefinite and is called a pseudo covariance matrix. This non-continuous evolution of the segmented covariance matrix can be corrected by enforcing the required properties of covariance and correlation matrices, respectively. Covariance matrices are symmetric and positive-semidefinite $\mathbf{P} \in S_+^n$, which ensures that its correlations are coherent, but it is not guaranteed that the combination of covariance segments, as described above, satisfies this property. As an example: Given the covariance matrix $\mathbf{P}$ in Equation (10): Let $\mathbf{P}_{AB}$ and $\mathbf{P}_{BC}$ be a positive cross-covariance between the states. Due to this relation, $\mathbf{P}_{AC}$ needs to represent a positive correlation as well.

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_A & \mathbf{P}_{AB} & \mathbf{P}_{AC} \\ \mathbf{P}_{BA} & \mathbf{P}_B & \mathbf{P}_{BC} \\ \mathbf{P}_{CA} & \mathbf{P}_{CB} & \mathbf{P}_C \end{bmatrix} \in S_+^n \quad (10)$$

A covariance correction step needs to be applied to accommodate this issue. [12] and [13] discuss a variety of methods to estimate the nearest positive-semidefinite covariance matrix of a given pseudo covariance matrix. The more interesting approaches are the Eigenvalue method and the Scaling/Hypersphere decomposition with angular parametrization, which have not been applied to the field of state-estimation to our knowledge.

The scaling method uses an optimization process to minimize the Frobenius distance (lower caps are the scalar elements of a matrix) $\mathbf{A} = \sum_i^n \sum_j^n (p_{i,j} - \tilde{p}_{i,j})^2$ with respect to a given covariance matrix where $\mathbf{P}$ is the true covariance matrix and $\tilde{\mathbf{P}}$ is the closest approximation. The Eigenvalue method approximates a positive-semidefinite matrix by correcting negative Eigenvalues. [14] proves that the Eigenvalue method also minimizes the Frobenius norm. Due to its deterministic nature and the lower complexity, the Eigenvalue method is the preferred choice for the presented real-time estimation problem. To perform the Eigenvalue correction, the first step is to decompose the covariance matrix $\mathbf{P} = \mathbf{DED}^T$ that needs to be adapted. $\mathbf{E}$ is a diagonal matrix with Eigenvalues, and $\mathbf{D}$ are the Eigenvectors. If the covariance matrix is non-positive-semidefinite, then a subset of the Eigenvalues $\mathbf{E}_{(<0)}$ is negative. These can be corrected by performing the:

- Absolute Eigenvalue correction (ABS), to preserve the dimension that is spanned by the Eigenvectors.
- Zero Eigenvalue correction (Zero), performing the minimal change required to gain a positive determinant, and
- Delta Eigenvalue correction (Delta), which sets the negative Eigenvalues to a positive empirical parameter.

The covariance matrix is then constructed based on the corrected Eigenvalues and can be used to update the recursive filter. The three methods are applied for the framework and evaluated in Section IV-A.

C. Implementation

The framework is structured in logical blocks (see Fig. 4) that represent the system design. Each component is self-contained with clearly defined interfaces for exchangeability. The core logic handles the organizational part of the framework and is the bridge between the buffer (see Fig. 5) and all sensor components. Its high-level logic determines if measurements are still valuable to the system or rejected (e.g. if the measurement was delayed and is older than the latest buffered

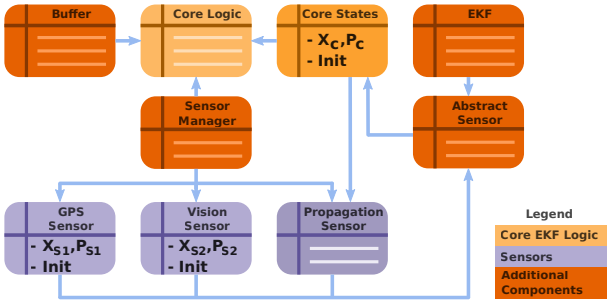


Fig. 4. Modular system design.

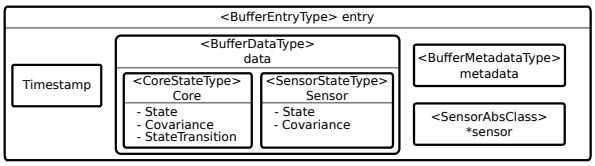


Fig. 5. Data structure of a single buffer entry.

entry). The core logic redirects dynamics measurements (e.g., from an IMU) to the core state module, which propagates the core state vector and its covariance. If measurements are associated with any other sensor instance (e.g., GNSS), then the core logic requests the latest state entry from the buffer and all state-transition matrices starting at the previous update of this sensor until the current step. The information is used to generate the state-transition matrix series described in Section III-B, which is used to propagate the cross-covariance terms of the core and sensor states. The propagated sensor cross-covariance is used to build a covariance matrix with the corresponding sensor and core covariance. The resulting covariance matrix is corrected with the Eigenvalue method and is passed to the sensor instance, which performs the update. The sensor instance can handle the reduced covariance matrix and state-prior (latest core and sensor state) as it is done for the classical approach. This also allows the use of statistical tests (e.g. χ^2 test) within the sensor's update step. The sensor instance returns the updated states and covariance segments of the core and the sensor to the core logic, which stores it in the buffer. The presented method renders the core agnostic to the sensor definition, which allows the arbitrary addition of sensors. Each sensor instance is self-contained, performs its own updates, and applies the corrections to its states. The same holds if a new sensor instance is added during runtime. A sensor module also handles its initialization based on the current core state, provided by the core logic. The framework is programmed in Matlab for fast prototyping, and implemented in C++ for high-performance applications. The C++ framework has minimal dependencies and only relies on the Eigen library. A ROS package that uses the API of the C++ library is also provided.

IV. Experiments

A. Validity and Observability

Due to the assumptions made in Section III-B, the approach needs to be evaluated in terms of performance and characteristics in simulation and the real-world. The tests of this section have three objectives:

1) The evaluation of the three Eigenvalue correction methods (ABS, Zero, Delta = 0.05).
2) An experimental analysis that unobservable vision states become observable by introducing a global pose sensor (i.e., that correct/consistent information is propagated despite the simplifications).
3) The validation of the overall modular approach.

The setup is as follows: we use the same simulated ground-truth trajectory to generate 20 independent Monte-Carlo datasets, which allow a statistically significant number of repetitions. The trajectory has a duration of 15 minutes with continuously varying velocity, accelerations, and randomly introduced smooth orientations. Each sequence has the same trajectory and the same Gaussian noise characteristics for sensor and IMU measurements. The datasets provide 200 Hz IMU measurements for propagation, 6DoF loosely-coupled vision pose (10 Hz), and 6DoF pose sensor measurements (50 Hz). The validity of the filter and the underlying modular approach is quantified by the:

- Average Normalized Estimation Error Squared (ANEES) described by [15] to determine the filter characteristics in terms of consistency and credibility,
- State error plots for time-dependent coherence, and
- Root-Mean-Square Error (RMSE) w.r.t. ground-truth, comparing the classical filter and our modular approach.

One dataset was processed with the classical full filter approach to establish a baseline for the best-case scenario (similar to the framework introduced by [3]). Each of the 20 datasets was processed with the modular filter definition using the three different Eigenvalue correction methods. The individual result of each state, from the modular approach, was used to generate an RMSE with respect to the full filter scenario. The mean of the individual core state RMSE for the different Eigenvalue correction methods are shown in Figure 6. The results show that the absolute Eigenvalue correction method performs best for all states except for velocity, where the zero method performs slightly better on two out of the three dimensions.

The same test was performed for the observability validation with introduced random state initializations for each sequence. The calibration of the vision-world reference frame for the vision sensor is unobservable but can be rendered observable by introducing a global pose sensor. Since we are using the segmentation of the covariance matrix, the vision and pose sensor are never jointly present in the covariance matrix in the same update step. Thus, we need to validate that the usual flow of information from the pose sensor to the vision sensor, which contributes to the observability of the vision-world reference frame due to cross-covariances, can be recovered from the core

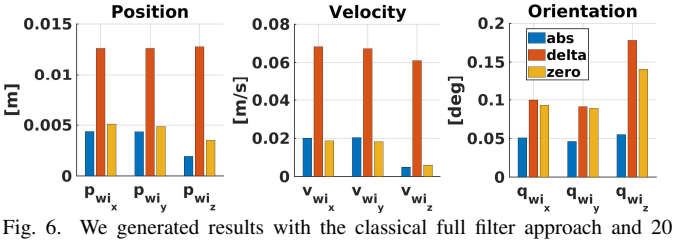


Fig. 6. We generated results with the classical full filter approach and 20 datasets each, with the three Eigenvalue correction methods of the modular approach. The graph shows the mean of the RMSE between the results of the classical and the modular filter for the essential core states.

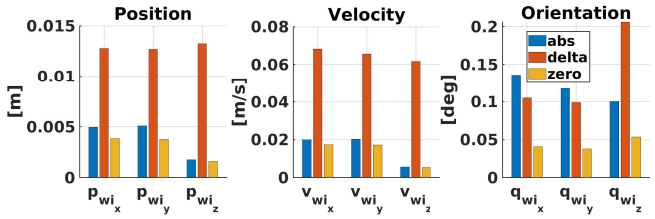


Fig. 7. We applied randomly generated and minorly wrong initializations of the states that are unobservable without using other sensor modalities to prove that the method preserves observability properties when using multiple sensors. The initial covariance encloses the error of the initialization by 3σ to allow for correct convergence.

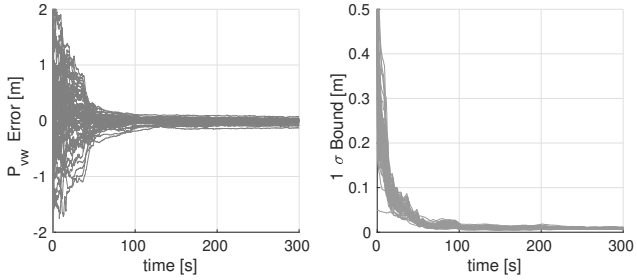


Fig. 8. The vision state $\mathbf{p}_{VW}$ is unobservable without additional global information. The plot shows the convergence of the state error (left) and standard deviation (right) using the modular approach with the absolute Eigenvalue method, 20 Monte-Carlo independent datasets, and varying errors on the initialization of $\mathbf{p}_{VW}$. The result further proves that observability properties are preserved with the modular approach.

states despite the covariance segmentation for the modular approach. This is not inherently given as we explicitly decouple the covariances of the sensors. If this information flow is not maintained, our approach would not be adequate for practical usage. For testing purposes, the state initializations for the vision sensor are altered for each dataset, and the covariance is adapted such that the introduced error is enclosed by a 3σ bound.

The RMSE of the core state is expected to be significantly higher if the vision states do not converge. Figures 6 and 7 as well as Table II confirm that the Eigenvalue correction to a small delta value ($\Delta = 0.05$) shows the least accurate performance, and motivate the usage of the absolute Eigenvalue method, which was therefore used for the remaining experiments and in Section IV. The low RMSE for the described scenario, shown by Figure 7, and the correct convergence of the state error and covariance in Figure 8, using the absolute Eigenvalue method, confirm that the modular approach preserves observability properties.

The next step is the validation of the overall filter credibility. We are using a sensor setup with two pose sensors for this test. The set of 20 datasets is processed with the modular and full filter setup, and the NEES for each run is used to generate the ANEES. Figure 9 shows the ANEES for the full and the modular approach with their corresponding mean. It also shows the 3σ upper-bound of the ANEES based on the number of states and datasets. Both ANEES results are below the upper 3σ ANEES bound, and the individual mean of the ANEES is shown in Table II.

The error plot for this scenario is not shown because the state errors are small and well presented by the RMSE in Table I. Although the mean error is slightly higher due to our approximation, the credibility is still given. The same

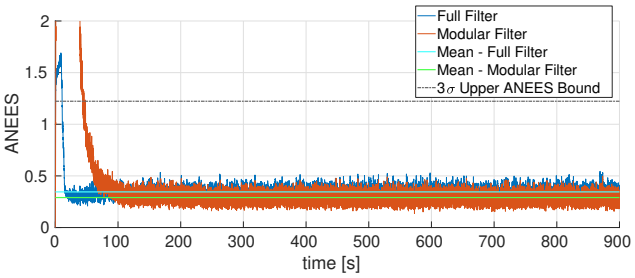


Fig. 9. ANEES for the core states of the full (blue) and modular (red) filter implementation using the absolute Eigenvalue method. We used 20 Monte-Carlo independent datasets to generate a statistically significant characterization of both filter methods. The plot shows the mean for both ANEES after the individual filter method converged. The upper 3σ bound of the the ANEES represented by the dashed line, is based on the number of core states and datasets.

evaluation is done for real-world in-flight data with additional environmental effects such as vibrations of the rotors that affect IMU readings in Section IV-C.

TABLE I
STATE ERRORS FOR THE FULL AND MODULAR FILTER DEFINITION

	$\mathbf{p}_{WI}$ [cm]			$\mathbf{q}_{WI}$ [degree]		
	x	y	z	roll	pitch	yaw
Full μ	0.84	0.83	1.27	0.361	0.375	1.707
Full σ	0.08	0.12	0.48	0.040	0.082	0.332
Modular μ	1.51	1.60	1.42	0.509	0.478	1.154
Modular σ	0.29	0.46	0.40	0.125	0.084	0.741

TABLE II
SUMMARY OF THE MEAN FOR THE ANEES RESULTS

States	Full Filter	Modular Abs	Modular Zero
Nav. Core	0.35	0.3	0.3
Pose Sensor	0.9	0.6	13

B. Performance

The performance of the modular filter is another essential aspect. This section presents timing profiles for a standard scenario and a scenario that forces the framework to repropagate states because of a delayed sensor measurement. Timing profiles are generated with three complete runs for each data point. Figure 10 shows the processing time of the update and propagation step for a series of 1-10 pose sensors. Each sensor introduces a 6DoF calibration state for translation and orientation. The core error state is defined with 15 states, derived from Section III-C and [1]. Thus, the figure shows the timings for 'one sensor' + 'core state' (21 States) and 'ten sensors' + 'core states' (75 states). Considering the case with 75 states: The corresponding covariance matrix has 5625 elements, which are processed by the classical approach for each update and propagation step. The benefit of the modular version is that it only processes the core state during the propagation phase and the core state with one additional sensor ($21 \cdot 21 = 441$) during any update phase.

The evaluation confirms that the propagation (Fig. 10, right) for the modular approach is independent of the number of additional sensors while the processing time of the classical implementation increases with the number of additional states. The processing time of the update (Fig. 10, left) for the classical approach grows exponentially while the modular approach grows linearly. The modular version is more efficient in terms of the total processing time (update + propagation phase), starting at a scenario with three additional sensors (see

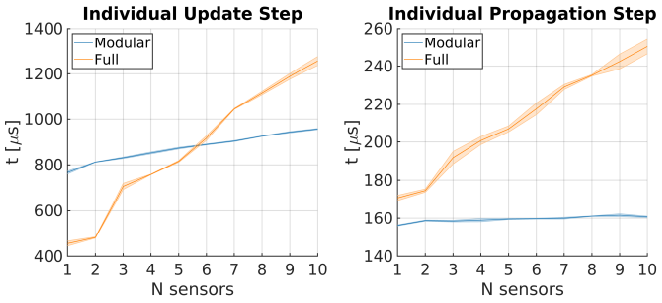


Fig. 10. Timing profile for the classical full and modular EKF approach. The time includes the buffer access and communication of data between the filter instances. This is done to provide a fair comparison since the modular approach introduces a slight overhead with respect to the classical approach. Overall, the modular version still outperforms the classical approach.

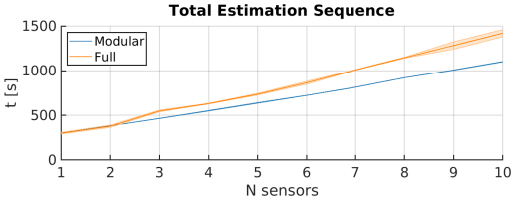


Fig. 11. Total running time comparing the full and modular method over a range of active sensors.

Fig. 11). The reason is a slight overhead due to the covariance correction, being the state transition block generation and the Eigenvalue correction, shown by the flow chart in Figure 3. The overall efficiency of the whole system is higher for the modular approach due to the decreased sensor update, and core propagation time.

C. Vision Aided Landing Scenario with Sensor Switching

This section presents experiments with a realistic flight scenario that is performed in simulation (see Fig. 13) and the real-world (see Fig. 16). The setup uses a GNSS sensor that provides position and velocity measurements at 5 Hz with a position standard deviation of $\sigma_{\mathbf{p}_g} = [0.85\ 0.85\ 2.16]^\mathsf{T}$ according to [16] and $\sigma_{\mathbf{v}_g} = 0.15\,\mathrm{m/s^2}$ for the velocity measurement as suggested by [17]. The setup also includes a loosely-coupled vision sensor in the form of a RealSense T265 with $\sigma_{\mathbf{p}_v} = 0.05\,\mathrm{m}$ for the position and $\sigma_{\mathbf{R}_v} = 1^\circ$ for the orientation measurement. The sensor suite further includes an NXP MPXH6115A integrated pressure sensor with $\sigma_{\mathbf{p}_p} = 0.15\,\mathrm{m}$. Sensor delays are not intentionally introduced. Figure 12 shows the flight profile and phases in which the sensors are switched with the same self-calibration states that are shown by Figure 2. Sensor states are initialized based on the current core state, and the covariance is initialized to enclose the possible error by a 3σ bound. The experiment is performed with 0.5 m/s velocity for all translations. The vehicle performs a vision based takeoff until an altitude of 3 m is reached (segment ①). The GNSS and barometric sensors are initialized in ②. Since these two sensor instances are not *a priori* known to the system, this event represents the addition of new sensors to the system. The vision sensor is deactivated at some point after the start of the horizontal translation. The vehicle performs a 3 m translation in the x-direction, holds at ④, translates 1 m in the y-direction, and returns to the takeoff location ③. Back in ②, the vision sensor is initialized to the current location, and after a short overlapping period,

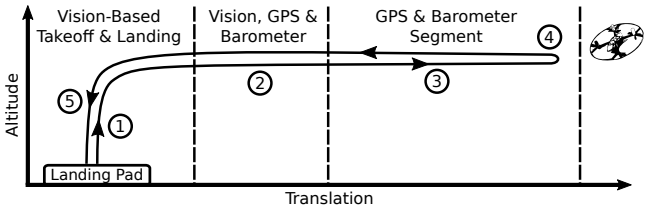


Fig. 12. Experiment THL flight profile with sensor switching cues.

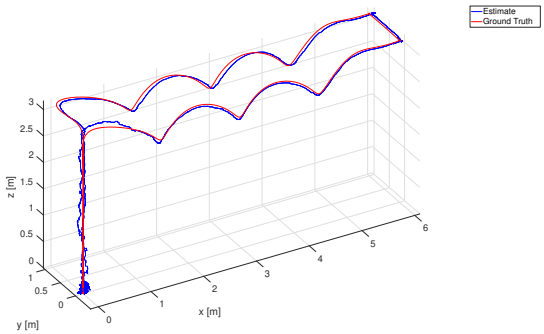


Fig. 13. 3D position estimate (blue) of the simulated sensor switching scenario and overlayed ground-truth (red). RMSE are shown by Table III. The curvy path allows for improved yaw estimation using GNSS and pressure sensors.

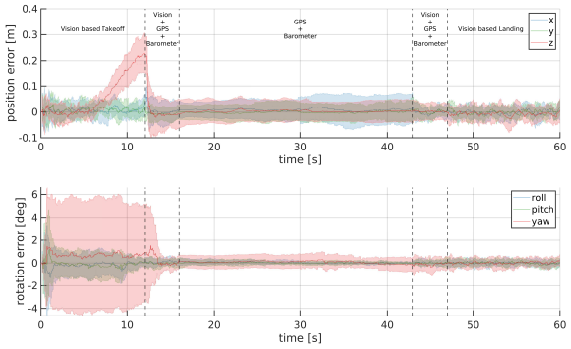


Fig. 14. State error for the position and orientation of the core state. This scenario was performed with 20 datasets to gain a statistically significant result for the truly modular approach. The initial increase of the error in z-position is caused by vision drift due to the takeoff maneuver, which also causes the increased initial covariance of the rotation in yaw.

the GNSS and barometric sensors are deactivated. The vehicle performs a vision-based landing at ⑤.

The real-world experiment is performed in a motion capturing room that provides 6DoF ground-truth for the vehicle's pose. The real vision and pressure measurements are used, and the GNSS position and velocity measurements are generated based on the ground-truth, with normal distributed noise, according to the characteristics mentioned before. The simulated and real-world scenarios do not provide synchronized measurements, and the datasets have high acceleration sections to render the bias of the core state observable. The presented modular state estimation framework performs self-initialization and self-calibration of the individual sensor reference frames and extrinsics based on the current state and sensor measurement.

The results of the simulation (see Fig. 14, Fig. 15, and Table III) further confirm the validity of the approach. They show that the covariance converges quickly after the absolute measurement is introduced and is consistent but underconfident throughout the experiment shown by the ANEES plot. The low RMSE in Table III also confirms the validity of the approach. The results of the real-world scenario (see Fig. 16

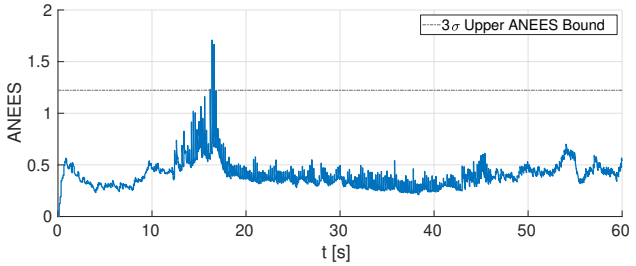


Fig. 15. ANEES for the simulated sensor switching scenario.

TABLE III
RMSE FOR THE SIMULATED AND REAL-WORLD THL SCENARIO.

	$\mathbf{p}_{WI}$ [cm]			$\mathbf{q}_{WI}$ [deg]		
	x	y	z	roll	pitch	yaw
Simulated						
Modular μ	4.03	3.27	6.23	0.62	0.65	1.94
Modular σ	1.17	0.61	1.25	0.156	0.199	1.083
Real-World						
Modular μ	15.23	12.65	15.06	2.59	1.97	2.20
Modular σ	14.24	11.54	13.03	2.51	1.73	1.49

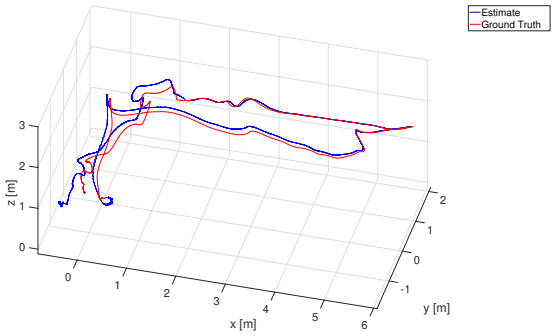


Fig. 16. 3D estimate of real-world data with sensor switching (blue) and overlayed ground-truth (red).

and Table III) illustrate that similar results are obtained with real data.

V. CONCLUSIONS

We introduce a novel truly modular multi-sensor fusion approach based on state covariance segmentation, which allows for the addition and removal of sensors at runtime with a significant gain of performance. Naive separation of covariance elements and successive propagation and update steps invalidates the fundamental properties of a covariance matrix. The introduced approach preserves these properties and ensures a consistent filter process. Extensive experiments in simulation and real-world prove that the true modularity approach is credible based on statistically significant ANEES analysis. Furthermore, the modular approach preserves observability, performs self-calibration, and self-initialization. This was shown throughout a vision based takeoff, transition, and landing scenario with four different sensor measurement updates. The scenario showed that the filter remains stable, consistent, and accurate throughout the presented sensor switching scenario with four sensor switching cues and two self-initialization procedures. All scenarios have been performed with asynchronous sensor measurements both in simulation and in a real flight. The modular approach outperforms the classical approach due to faster sensor updates, which improves general scalability for implementations that use this approach.

It was also shown that the propagation phase of the modular approach is constant and invariant to the number of sensors, while the processing time of the classical approach grows exponentially. This is especially interesting if a system uses sensors that introduce measurement delays because the modular approach requires significantly less time to perform a re-propagation step.

REFERENCES

[1] S. Weiss and R. Y. Siegwart, "Real-time metric state estimation for modular vision-inertial systems," *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 231855, pp. 4531–4537, 2011.

[2] C. Brommer, D. Malyuta, D. Hentzen, and R. Brockers, "Long-duration autonomy for small rotorcraft UAS including recharging," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, Oct. 2018, pp. 7252–7258.

[3] S. Lynen, M. W. Achtelik, S. Weiss, M. Chli, and R. Siegwart, "A robust and modular multi-sensor fusion approach applied to MAV navigation," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, Nov. 2013.

[4] S. Shen, Y. Mulgaonkar, N. Michael, and V. Kumar, "Multi-sensor fusion for robust autonomous flight in indoor and outdoor environments with a rotorcraft MAV," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 4974–4981, 2014.

[5] K. Hausman, S. Weiss, R. Brockers, L. Matthies, and G. S. Sukhatme, "Self-calibrating multi-sensor fusion with probabilistic measurement validation for seamless sensor switching on a UAV," *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2016-June, pp. 4289–4296, 2016.

[6] C. Tessier, C. Cariou, C. Debain, F. Chausse, R. Chapuis, and C. Rousset, "A real-time, multi-sensor architecture for fusion of delayed observations: application to vehicle localization," in *2006 IEEE Intelligent Transportation Systems Conference*. IEEE, 2006. [Online]. Available: https://doi.org/10.1109/itsc.2006.1707405

[7] T. Moore and D. Stouch, "A generalized extended kalman filter implementation for the robot operating system," in *Proceedings of the 13th International Conference on Intelligent Autonomous Systems (IAS-13)*. Springer, Jul. 2014.

[8] M. Darms and H. Winner, "A modular system architecture for sensor data processing of ADAS applications," *IEEE Intelligent Vehicles Symposium, Proceedings*, vol. 2005, pp. 729–734, 2005.

[9] D. A. Cucci and M. Matteucci, "On the Development of a Generic Multi-Sensor Fusion Framework for Robust Odometry Estimation," *Journal of Software Engineering for Robotics*, vol. 5, no. May, pp. 48–62, 2014.

[10] H.-P. Chiu, X. S. Zhou, L. Carlone, F. Dellaert, S. Samarasekera, and R. Kumar, "Constrained optimal selection for multi-sensor robot navigation using plug-and-play factor graphs," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, May 2014.

[11] S. I. Roumeliotis and G. A. Bekey, "Distributed multirobot localization," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 781–795, 2002.

[12] P. J. Rousseeuw and G. Molenberghs, "Transformation of non positive semidefinite correlation matrices," *Communications in Statistics - Theory and Methods*, vol. 22, no. 4, pp. 965–984, Jan. 1993. [Online]. Available: https://doi.org/10.1080/03610928308831068

[13] R. Rebonato and P. Jaeckel, "The most general methodology to create a valid correlation matrix for risk management and option pricing purposes," *SSRN Electronic Journal*, 2011. [Online]. Available: https://doi.org/10.2139/ssrn.1969689

[14] N. J. Higham, "Computing a nearest symmetric positive semidefinite matrix," *Linear Algebra and its Applications*, vol. 103, pp. 103–118, May 1988. [Online]. Available: https://doi.org/10.1016/0024-3795(88)90223-6

[15] X. R. Li, Z. Zhao, and X. B. Li, "Evaluation of Estimation Algorithms: Credibility Tests," *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans*, vol. 42, no. 1, pp. 147–163, 2012.

[16] W. J. Hughes, "Global positioning system (gps) standard positioning service (sps) performance analysis report," *Tech. Cntr. NSTB/WAAS T and E Team*, no. 87, 2014.

[17] L. Serrano, D. Kim, R. B. Langley, K. Itani, and M. Ueno, "A gps velocity sensor: how accurate can it be?–a first look," in *ION NTM*, vol. 2004, 2004, pp. 875–885.

LiPo-LCD: Combining Lines and Points for Appearance-based Loop Closure Detection

Joan P. Company-Corcoles
joanpep.company@uib.es
Emilio Garcia-Fidalgo
emilio.garcia@uib.es
Alberto Ortiz
alberto.ortiz@uib.es

Department of Mathematics and Computer Science, University of the Balearic Islands, and
IDISBA (Institut d'Investigacio Sanitaria de les Illes Balears),
Palma de Mallorca, Spain

Abstract

Visual SLAM approaches typically depend on loop closure detection to correct the inconsistencies that may arise during the map and camera trajectory calculations, typically making use of point features for detecting and closing the existing loops. In low-textured scenarios, however, it is difficult to find enough point features and, hence, the performance of these solutions drops drastically. An alternative for human-made scenarios, due to their structural regularity, is the use of geometrical cues such as straight segments, frequently present within these environments. Under this context, in this paper we introduce LiPo-LCD, a novel appearance-based loop closure detection method that integrates lines and points. Adopting the idea of incremental Bag-of-Binary-Words schemes, we build separate BoW models for each feature, and use them to retrieve previously seen images using a late fusion strategy. Additionally, a simple but effective mechanism, based on the concept of island, groups similar images close in time to reduce the image candidate search effort. A final step validates geometrically the loop candidates by incorporating the detected lines by means of a process comprising a line feature matching stage, followed by a robust spatial verification stage, now combining both lines and points. As it is reported in the paper, LiPo-LCD compares well with several state-of-the-art solutions for a number of datasets involving different environmental conditions.

1 Introduction

Simultaneous Localization and Mapping (SLAM) is a fundamental task in autonomous mobile robotics. Regardless of the sensor used to perceive the environment, unavoidable noise sources always interfere, leading to errors in the map and the robot's pose calculations, resulting in inconsistent representations. To overcome this problem, SLAM systems usually rely on *loop closure detection* (LCD) methods to recognize previously seen places. These detections provide additional constraints that can be used to correct the accumulated drift. When cameras are involved, these methods are referred to as *appearance-based* loop closure detection approaches [1, 5, 11, 12, 13, 20, 26].

It is well known that many visual SLAM solutions rely on point features because of their wider applicability in general terms [23, 28]. Human-made environments, however, can lack

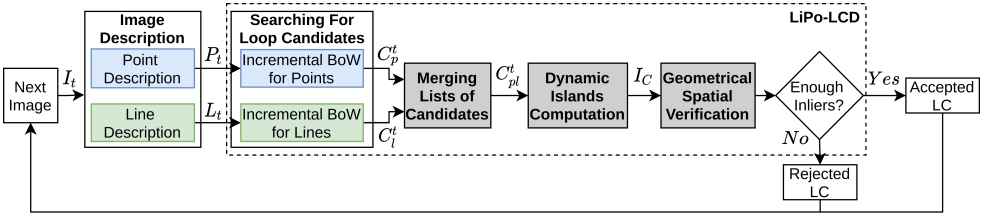


Figure 1: General overview of the proposed loop closure detection system.

texture and thus give rise to a low number of detected features. Nevertheless, precisely because of their nature, these environments usually exhibit structural regularities that can be described using richer features such as lines, which can be more robust and less sensitive to illumination changes. Several solutions can be found in the literature describing approaches combining both kinds of features, points and lines [31, 40]. However, despite their success, most of them rely exclusively on feature points during the LCD stage, discarding information about lines that may be useful to improve the association performance for textureless environments. Other approaches opt for using holistic image representations [4, 26, 36], which can be faster to compute but less tolerant to visual changes, while, lately, solutions based on Convolutional Neural Networks (CNNs) [2, 8, 37] have shown to exhibit enhanced robustness and general performance, although they are still disengaged from real-time SLAM problems [13, 39]. This is because they tend to require significant computational resources, e.g. on-board GPU, which makes them not suitable for mobile robotics in all cases.

The Bag of Words (BoW) model [29, 34], in combination with an inverted file, is arguably the most used indexing scheme for appearance-based loop closure detection [14, 25]. Depending on how the visual vocabulary is generated, BoW-based solutions can be classified into off-line and on-line approaches. Off-line solutions generate the visual dictionary during a training phase [11, 12, 27], what can be high time-consuming, while the general application of the resulting vocabulary becomes highly dependent on the diversity of the training set. As an alternative, there are approaches that propose to generate the dictionary on-line [1, 13, 20, 21, 38, 39]. Moreover, binary descriptors [12, 13, 27] have emerged recently as an alternative to real-valued descriptors for BoW models [1, 11, 21], since they offer advantages in terms of computational time and memory requirements. Additionally, similarity calculations can be performed using the Hamming distance or an of its variations, what can be efficiently implemented in modern processors.

Under this context, in this paper, we introduce *Lines and Points Loop Closure Detection* (LiPo-LCD), a novel appearance-based loop closure detection approach which combines points and lines. For a start, both features are described using binary descriptors. Next, an incremental BoW scheme is used for feature indexing. Lines and points are maintained separately into two incremental visual vocabularies and employed in parallel to obtain loop closure candidates efficiently. To combine the information provided by the two vocabularies, we propose a late fusion method based on a ranked voting system. Finally, to discard false positives, we improve the typical spatial verification step integrating lines into the procedure through: (1) a line matching strategy which includes structural information to achieve a higher number of matching candidates; and (2) a line representation tolerant to occlusions, which is combined with points into an epipolarity analysis step. A set of experiments validating LiPo-LCD and characterizing its performance against several state-of-the-art solutions is

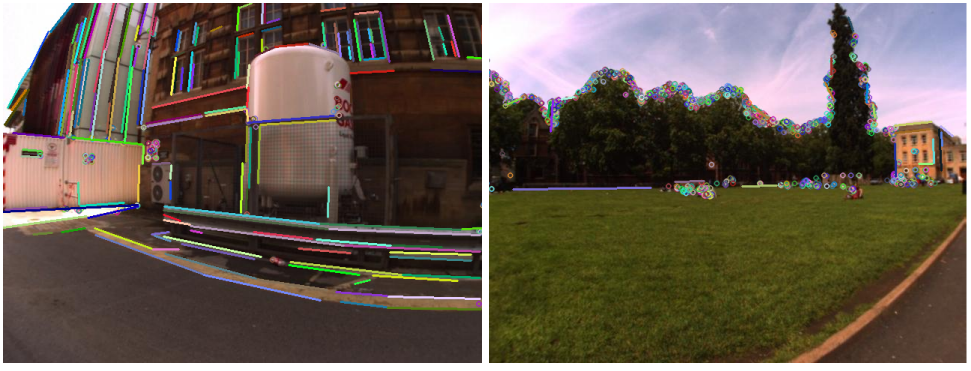


Figure 2: (left) A human-made environment including a high number of lines and a low number of points. (right) An outdoor environment presenting the opposite situation.

reported at the end of the paper.

Our approach follows a dual scheme to combine points and lines, such as the solutions proposed by [17, 45]. Nonetheless, LiPo-LCD takes advantage of an incremental BoW strategy and incorporates lines into the spatial verification procedure that does not require map information, increasing its ability to be adapted to the operating environment, requiring only a monocular camera, and improving the performance in several datasets, as shown later.

2 Overview of the Loop-Closure Detection Approach

Figure 1 illustrates the approach proposed for loop closure detection. As can be observed, incremental visual vocabularies, along with the corresponding inverted files, are maintained independently for each visual feature. When a new image is sampled, a set of line and point binary descriptors is computed and used to (1) update the corresponding visual vocabulary and (2) obtain a list of the most similar images from each vocabulary. Next, the two lists are fused using a ranked voting procedure to obtain a final set of loop-closing candidates. To avoid adjacent images from competing with each other as loop candidates, we group images close in time using the concept of *dynamic island* [13]. Among the resulting islands, the one best corresponding with the query image is selected, while its representative image is geometrically assessed against the query to accept/reject the loop. The details about the aforementioned processes can be found next.

2.1 Image Description

As stated previously, LiPo-LCD describes images using lines and points. The rationale behind this approach is that the combination of multiple, complementary description techniques is a way leading to improving the performance and robustness of the loop closing method [19]. In our solution, the image I_t at time t is described by $\phi(I_t) = \{P_t, L_t\}$, being P_t a set of local keypoint descriptors and L_t a set of line descriptors, both deriving from I_t. These two descriptions complement each other to make image representation more robust: while some environments may be described more distinctively using lines than points, i.e. textureless scenes, others lacking structure will benefit from keypoints, and the net result is a joint descriptor of a wider scope. Figure 2 illustrates this issue for two environments.

2.1.1 Point Description

Given the above-mentioned advantages about binary descriptors, in this work, we have opted for detecting and describing points using ORB [32]. Although the proposed strategy can be used with any other binary descriptor, we employ ORB because of its robustness to rotation, scale and illumination changes [28]. The m ORB descriptors found at image I_t define the point descriptor as $P_t = \{p_0^t, p_1^t, \ldots, p_{m-1}^t\}$.

2.1.2 Line Description

Lines are found using the Line Segment Detector (LSD) [18]. LSD is a linear-time line segment detector that provides high-precision results and subpixel accuracy without parameter tuning. On the one hand, detected lines are described using a binary form of the Line Band Descriptor (LBD) [43]. In the original implementation, a rectangular region centred on each line is considered. Such region is divided into a set of bands B_i, from which a descriptor BD_i is computed contrasting B_i with its neighbouring bands. On the other hand, the binary descriptor is finally obtained considering 32 possible pairs of band descriptors BD_i within the support region. Each pair is compared bit by bit, generating an 8-bit string per pair. A final 256-bit descriptor is generated concatenating the resulting strings for all pairs. The set of n LBD binary descriptors for image I_t defines the line descriptor $L_t = \{l_0^t, l_1^t, \ldots, l_{n-1}^t\}$.

2.2 Searching for Loop Closure Candidates

To index and retrieve loop closure candidates, we rely on the *OBIndex2* approach [13], a hierarchical tree structure to manage an increasing number of binary descriptors in an efficient way. This structure can then be used as an incremental BoW scheme and combined with an inverted file for fast image retrieval. The reader is referred to [13] for further detail.

Given that LiPo-LCD describes all visual features using binary descriptors, we maintain two instances of *OBIndex2*, one for points and one for lines. Each instance builds an incremental visual vocabulary along with an index of images for each feature. Given an image I_t, a parallel search is performed on each index to retrieve the most similar images of points and lines. As a result, two lists are obtained: (1) the m most similar images using points $C_p^t = \{I_{p_0}^t, \ldots, I_{p_{m-1}}^t\}$ and (2) the n most similar images using lines $C_l^t = \{I_{l_0}^t, \ldots, I_{l_{n-1}}^t\}$. Each list is sorted by, respectively, their associated scores $s_p^t(I_t, I_j^t)$ and $s_l^t(I_t, I_j^t)$, which measure the similarity between the query image I_t and the image I_j. Since the range of these scores varies depending on the distribution of visual words for each vocabulary, they are mapped onto the range [0,1] using *min-max normalization* as follows:

$$\tilde{s}_k^t\left(I_t, I_j^t\right) = \frac{s_k^t\left(I_t, I_j^t\right) - s_k^t\left(I_t, I_{min}^t\right)}{s_k^t\left(I_t, I_{max}^t\right) - s_k^t\left(I_t, I_{min}^t\right)}, \tag{1}$$

where $s_k^t\left(I_t, I_{min}^t\right)$ and $s_k^t\left(I_t, I_{max}^t\right)$ respectively corresponds to the minimum and the maximum scores of an image candidate list, being $k \in \{p, l\}$. Images whose normalized score $\tilde{s}_k^t$ is lower than a threshold are discarded to limit the maximum number of candidates. Additionally, the current image descriptors are used to update the visual vocabularies appropriately.

2.3 Merging Lists of Candidates

The two resulting lists C_p^t and C_l^t provide loop closure candidates from each individual perspective. Thus, the next step is to combine both lists to obtain an overall overview of possible candidates but considering lines and points altogether. In this regard, the literature comprises multiple techniques to combine multimodal information for image retrieval [7]. These can generally be categorized into two schemes, namely *early* and *late fusion*: while the former combines all features into a single representation before being processed, the latter works at the decision level, combining the outputs produced by different retrieval systems. In our proposal, given the heterogeneity of the features to combine, we rely on a late fusion approach that employs a ranked voting system based on the Borda count [33] to merge lists of candidates. This is a simple data fusion form based on democratic election strategies: first, a set of voters rank a list of fixed candidates on the basis of their preferences; scores are next given to each candidate in inverse proportion to their ranking; finally, once all votes have been emitted, the candidate with the highest number of votes wins. In LiPo-LCD, two independent voters, one for each visual vocabulary, emit an different-size ordered list of candidates C_k^t. The number of candidates c to vote for is set as the minimum length of the two lists. Next, top-c images on each list C_k^t are ranked with a score b_k as:

$$b_k(I_i^t) = (c-i)\,\tilde{s}_k^t\left(I_t, I_i^t\right), \tag{2}$$

where i denotes the order of the image I_i in the list C_k^t and $\tilde{s}_k^t\left(I_t, I_i^t\right)$ is the normalized score of the image in that list. For each image that appears in both lists, a combined Borda score β is computed as the geometric mean of the individual scores:

$$\beta(I_i^t) = \sqrt{b_p(I_i^t)\, b_l(I_i^t)}\,. \tag{3}$$

We employ the geometric mean instead of the arithmetic mean to reduce the influence of false positives in one of the lists. An integrated image list C_{pl}^t results next by sorting the scores $\beta(I_i^t)$ of all the retrieved images. This list merges information from the two visual vocabularies, independently of the number of features detected in the current environment. Finally, to deal with the fact that some environments mostly exhibit one type of feature, images that only appear in one list are also incorporated into C_{pl}^t, although penalized.

2.4 Dynamic Islands Computation

In pursuit of selecting a final loop closure candidate, in this stage we verify the temporal consistency of the images retrieved in C_{pl}^t. To this end, we rely on the concept of *dynamic islands* used by iBoW-LCD [13]. This method permits to avoid images competing among them as loop candidates when they come from the same area of the environment. A dynamic island Υ_n^m groups the images whose timestamps range from m to n. Initially, a set of islands Γ_t for the current image I_t is computed considering images in the list C_{pl}^t sequentially: every image $I_i \in C_{pl}^t$ is either associated to an existing island Υ_n^m if the image timestamp lies in the $[m,n]$ interval or else is used to create a new island. After processing all images in C_{pl}^t, a global score g is computed for each island as:

$$g(\Upsilon_n^m) = \frac{\sum_{i=m}^{n} \beta(I_i^t)}{n-m+1}\,. \tag{4}$$

Unlike [13], where only points are considered, in LiPo-LCD, score g is the average of the Borda scores of the images belonging to the island, integrating both points and lines. Finally, the resulting set of islands Γ_t is sorted in descending order according to g. Next step is to select one of the resulting islands, denoted by $\Upsilon^*(t)$, to determine which area of the environment is the one most likely closing a loop with I_t. iBoW-LCD makes use of the concept of *priority islands*, defined as the islands in Γ_t that overlap in time with the island selected at time $t-1$, $\Upsilon^*(t-1)$. This is inspired by the fact that consecutive images should close loops with areas of the environments where previous images also closed a loop. iBoW-LCD selects, as a final island, the priority island with the highest score g, if any. However, this approach is just based on the appearance of the images and, therefore, due to perceptual aliasing, it might produce incorrect island associations in some human-made environments. For this reason, LiPo-LCD proposes a simple but effective modification of the original approach that only retains an island for the next time step if the final selected loop candidate satisfies the spatial verification procedure explained in Section 2.5. Once the best island $\Upsilon^*(t)$ has been determined, the image I_c with the highest Borda score β of $\Upsilon^*(t)$ is selected as its representative and evaluated in the next verification stage.

2.5 Spatial Verification

Although the BoW scheme is a good starting point to find loop closure candidates, to finish, we perform a final geometric check to take into account the spatial arrangement of the image features and avoid perceptual aliasing. This final step comprises an epipolarity analysis between the current image I_t and the loop candidate I_c on the basis of the number of inliers that support the roto-translation of the camera (after computing the fundamental matrix F using RANSAC). If the number of inliers is not high enough, the loop hypothesis is rejected.

The epipolarity analysis is typically carried out using a putative set of point matchings. However, as stated along this paper, point features might not be helpful because of the nature of the environment, and hence integrating lines into the geometric check can be useful, apart from the fact that straight segments can tolerate partial occlusions. To this end, LiPo-LCD makes use of (1) a novel line feature matching approach and (2) incorporates these line matchings, together with point matchings, into the geometric check. To match points, we make use of the available ORB descriptors, the Hamming distance and the Nearest Neighbour Distance Ratio (NNDR) [24].

2.5.1 Line Feature Matching

Although NNDR is normally useful to discard false matchings between keypoints, it performs poor in respect to line descriptors matching, especially in human-made environments where line descriptors tend to be affected by perceptual aliasing [43]. To enhance line matching performance, the authors of [43] combine structural and appearance information in a relational graph. Despite their good results, their approach requires a high amount of memory and does not escalate well with the number of lines. In this work, we propose a much simpler but effective method to combine structural and appearance information for line feature matching. First, for each line descriptor l_i^t in the current image I_t, we retrieve an ordered list of the most similar line descriptors of the candidate image I_c. Next, to deal with camera rotations, we compute a global rotation θ_g between the two frames as explained in [43]. θ_g

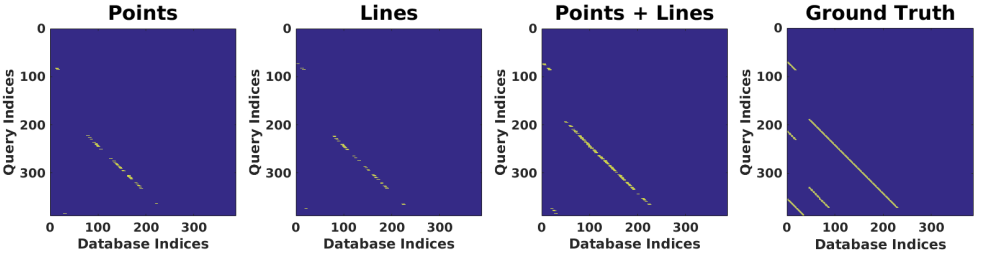


Figure 3: Loop closure detections found in the L6I dataset using different visual features (Points, Lines, Points + Lines), and the associated ground truth. White dots represent a loop closure detected.

is next used to compute the relative orientation α_i^j between each pair of lines as:

$$\alpha_i^j = |\theta_i^t - \theta_j^c + \theta_g|, \tag{5}$$

being θ_i^t the orientation of the line on the current image and θ_j^c the orientation of their corresponding line in the list. For each list, all line matchings with high values of α_i^j are discarded, and, as a result, a filtered list of line candidate matchings is obtained. To generate the final set of line matchings, we choose the two most similar surviving nearest neighbours from each list and apply the NNDR test.

2.5.2 Epipolar Geometry Analysis Combining Points and Lines

Works described in [6, 30] compute the fundamental matrix F from homographies estimated from line segment matchings across images, provided these segments lie in at least two different planes. LiPo-LCD makes use of a simpler but effective approach that avoids this constraint. On the one hand, differently to other representations that can be found in the literature [22, 41, 44], in this work, line segments are represented by their endpoints. On the other hand, endpoints are first matched between matching lines and next regarded as additional point correspondences for F computation. To associate segment endpoints (taking into account that a starting point of a line might correspond to the end point of the line in the other image), we select that pair that minimizes the rotation between lines using lines orientation and the global rotation θ_g, as computed in Eq. 5. We consider a candidate line matching as an inlier if at least one endpoint pair supports the geometric model.

3 Experimental Results

In this section, we evaluate the performance of LiPo-LCD using several public datasets. LiPo-LCD is also compared against some state-of-the-art solutions. All experiments were performed on an Intel Core i7-9750H (2.60 GHz) processor with 16 GB RAM.

3.1 Methodology

Precision-recall metrics are used to evaluate the system. Given that false detections can be critical if LiPo-LCD is used in a real SLAM solution, we are especially interested in observing the maximum recall that can be achieved at 100% precision. OBIndex2 and iBoW-LCD

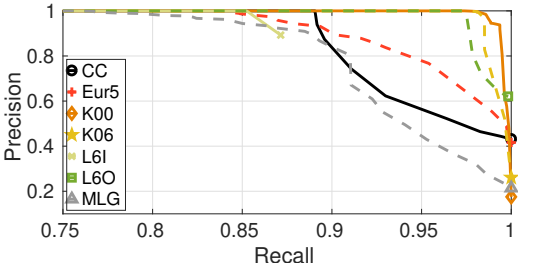


Figure 4: P-R curves for each dataset. P is 1.0 for all R values lower than 0.75.

	CC	EuR5	K00	K06	L6I	L6O	MLG
NNDR	8.54	8.91	15.06	12.35	7.43	3.47	4.24
Proposed	18.15	19.21	25.37	22.51	10.14	8.45	11.34

Table 1: Average number of line inliers after the epipolar geometric analysis using NNDR and the proposed line feature matching method.

were configured as explained in [13]. The rest of the approaches shown in this section were executed using the default parameters proposed by their original authors. The following datasets were considered to validate LiPo-LCD: CityCentre [10] (CC), EuRoC Machine Hall 05 [9] (EuR5), KITTI 00 [16] (K00), KITTI 06 [16] (K06), Lip6Indoor [1] (L6I), Lip6Outdoor [1] (L6O) and Malaga 2009 Parking 6L [8] (MLG). These datasets encompass a wide range of environments including, for instance, urban and indoor scenarios, which are usually rich in lines, or outdoor scenarios, where points predominate over lines. For each dataset, we use the ground truth provided by the original authors except for the KITTI sequences, where we employ the one provided by [4], and the EuR5 and MLG datasets, where we use the files provided by [39].

3.2 General Performance

First, we validate the combination of points and lines proposed in this work. To this end, Fig. 3 shows the loop closures detected by LiPo-LCD using points, lines and both features, as well as the ground truth for the L6I dataset, whose images are poor in feature points. As can be observed, system performance increases when points and lines are used together as visual features. To measure the global performance of the system, Fig. 4 shows precision-recall curves for LiPo-LCD and for each dataset. As can be observed, high recall rates are

	FE	VU	SC	SV
Points	18.05	183.16	146.73	-
Lines	17.60	23.76	18.13	-
Parallel	19.05	196.58	159.01	15.09

Table 2: Average response time (ms) per image, calculated for each part of the pipeline. These times were computed over the K00 dataset. FE: Feature Extraction; VU: Vocabulary Update; SC: Search for Candidates; SV: Spatial Verification.

	CC	EuR5	K00	K06	L6I	L6O	MLG
Bampis [5]	71.14	n.a.	96.53	n.a.	52.22	58.32	87.56
Gálvez-López [12]	31.61	n.a.	n.a.	n.a.	n.a.	n.a.	74.75
Mur-Artal [27]	43.03	n.a.	n.a.	n.a.	n.a.	n.a.	81.51
Cummins [10]	38.77	n.a.	49.2	55.34	n.a.	n.a.	68.52
Stumm [35]	38.00	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.
Gomez-Ojeda [17]	n.a.	1.61	75.93	56.94	n.a.	n.a.	n.a.
Tsintotas [39]	n.a.	**83.7**	97.5	n.a.	n.a.	50.0	85.0
Tsintotas [38]	n.a.	69.2	93.2	n.a.	n.a.	n.a.	**87.9**
Angeli [1]	n.a.	n.a.	n.a.	n.a.	36.86	23.59	n.a.
Zhang [42]	41.2	n.a.	n.a.	n.a.	n.a.	n.a.	82.6
Gehrig [15]	n.a.	71.0	93.1	n.a.	n.a.	n.a.	n.a.
Khan [20]	38.92	n.a.	n.a.	n.a.	41.74	25.58	78.13
Garcia-Fidalgo [13]	88.25	n.a.	76.50	95.53	83.18	85.24	n.a.
LiPo-LCD	**89.30**	81.94	**97.80**	**97.38**	**85.24**	**97.31**	75.73

Table 3: Maximum recall at 100% precision for several off-line approaches (top), on-line approaches (middle) and the proposed solution (bottom). Winners are indicated in bold face.

always achieved while maintaining the precision at 100%. Moreover, LiPo-LCD exhibits very stable behaviour in all cases.

Next, we evaluate our novel line feature matching strategy. For that purpose, we compute the average number of line inliers on each dataset using either a classical NNDR approach for lines and our approach. Results are shown in Table 1. As can be seen, the proposed line matching technique achieves a higher number of inliers in all datasets, even in sequences with severe appearance changes.

Finally, we evaluate the performance of LiPo-LCD in terms of computational times. The results obtained can be found in Table 2. We show results for K00 since it is the largest dataset considered in this work. We measure the average execution time in milliseconds for each stage of the pipeline, not taking into account times for merging lists of candidates and island selection, since they are negligible. The average response time of the whole system per image turns out to be 389.79 ms using a parallel implementation. As can be observed, feature extraction steps are very fast in all cases. The vocabulary update and the search for candidates steps are slower for points, due to the number of features to handle on each case. The spatial verification stage is always performed using points and lines together, and, hence, times for each feature separately are not available.

3.3 Comparison with Other Solutions

In this last section, LiPo-LCD is compared with other solutions. Table 3 shows the maximum recall achieved at 100% precision for all approaches. The results reported come from the original works, except for [17], which was executed by ourselves using the vocabularies and the default parameters provided by their authors. Results not available are indicated by *n.a.* As can be observed, LiPo-LCD achieves, in most cases, a higher recall than the other solutions. This is particularly interesting regarding the L6I dataset, where the combination of points and lines allows us to increase the performance in a low-textured scenario. It is also worth mentioning that LiPo-LCD outperforms [17], which is perhaps the most similar solution to ours.

4 Conclusions

In this work, we have described LiPo-LCD, an appearance-based loop closure detection method that combines points and lines. This combination allows us to detect loops in environments poor of feature points. Moreover, points and lines are described using binary descriptors for execution time reduction. To obtain loop closure candidates from both visual clues, we rely on a dual incremental BoW scheme. A late fusion method for merging both lists of candidates, based on the Borda count, is also proposed. The loop candidate hypothesis is finally validated by means of a geometrical check, which involves both points and lines. LiPo-LCD compares favourably with several state-of-the-art methods under different environmental conditions.

Acknowledgements

This work is partially supported by EU-H2020 projects BUGWRIGHT2 (GA 871260) and ROBINS (GA 779776), and by projects PGC2018-095709-B-C21 (MCIU/AEI/FEDER, UE), and PROCOE/4/2017 (Govern Balear, 50% P.O. FEDER 2014-2020 Illes Balears). This publication reflects only the authors views and the European Union is not liable for any use that may be made of the information contained therein.

References

[1] Adrien Angeli, David Filliat, Stéphane Doncieux, and Jean-Arcady Meyer. A fast and incremental method for loop-closure detection using bags of visual words. *IEEE Transactions on Robotics*, 24(5):1027–1037, 2008. ISSN 15523098.

[2] R. Arandjelović, P. Gronat, A. Torii, T. Pajdla, and J. Sivic. NetVLAD: CNN architecture for weakly supervised place recognition. In *International Conference on Computer Vision and Pattern Recognition*, 2016.

[3] R. Arroyo, P. F. Alcantarilla, L. M. Bergasa, and E. Romera. Fusion and binarization of CNN features for robust topological localization across seasons. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4656–4663, 2016.

[4] Roberto Arroyo, Pablo. F Alcantarilla, Luis. M. Bergasa, J. Javier Yebes, and Sebastian Bronte. Fast and effective visual place recognition using binary codes and disparity information. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3089–3094, 2014.

[5] Loukas Bampis, Angelos Amanatiadis, and Antonios Gasteratos. Fast loop-closure detection using visual-word-vectors from image sequences. *International Journal of Robotics Research*, 37(1):62–82, 2018.

[6] H. Bay, V. Ferraris, and L. Van Gool. Wide-baseline stereo matching with line segments. In *International Conference on Computer Vision and Pattern Recognition*, volume 1, pages 329–336 vol. 1, 2005.

[7] N. Bhowmik, R. González V., V. Gouet-Brunet, H. Pedrini, and G. Bloch. Efficient fusion of multidimensional descriptors for image retrieval. In *IEEE International Conference on Image Processing*, pages 5766–5770, 2014.

[8] Jose-Luis Blanco, Francisco-Angel Moreno, and Javier Gonzalez. A collection of outdoor robotic datasets with centimeter-accuracy ground truth. *Autonomous Robots*, 27 (4):327, 2009.

[9] Michael Burri, Janosch Nikolic, Pascal Gohl, Thomas Schneider, Joern Rehder, Sammy Omari, Markus W Achtelik, and Roland Siegwart. The EuRoC micro aerial vehicle datasets. *International Journal of Robotics Research*, 35(10):1157–1163, 2016.

[10] Mark Cummins and Paul Newman. FAB-MAP: probabilistic localization and mapping in the space of appearance. *International Journal of Robotics Research*, 27(6):647–665, 2008.

[11] Mark Cummins and Paul Newman. Appearance-only SLAM at large scale with FAB-MAP 2.0. *International Journal of Robotics Research*, 30(9):1100–1123, 2011.

[12] D. Galvez-López and J. D. Tardos. Bags of binary words for fast place recognition in image sequences. *IEEE Transactions on Robotics*, 28(5):1188–1197, 2012.

[13] E. Garcia-Fidalgo and A. Ortiz. iBoW-LCD: an appearance-based loop-closure detection approach using incremental bags of binary words. *IEEE Robotics and Automation Letters*, 3(4):3051–3057, 2018.

[14] Emilio Garcia-Fidalgo and Alberto Ortiz. Vision-based topological mapping and localization methods: A survey. *Robotics and Autonomous Systems*, 64:1 – 20, 2015. ISSN 0921-8890.

[15] Mathias Gehrig, Elena Stumm, Timo Hinzmann, and Roland Siegwart. Visual place recognition with probabilistic voting. In *IEEE International Conference on Robotics and Automation*, pages 3192–3199, 2017.

[16] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? The KITTI vision benchmark suite. In *International Conference on Computer Vision and Pattern Recognition*, pages 3354–3361, 2012.

[17] Ruben Gomez-Ojeda, David Zuñiga-Noël, Francisco-Angel Moreno, Davide Scaramuzza, and Javier Gonzalez-Jimenez. PL-SLAM: a stereo SLAM system through the combination of points and line segments. *arXiv preprint arXiv:1705.09479*, 2017.

[18] R. Grompone von Gioi, J. Jakubowicz, J. Morel, and G. Randall. LSD: A fast line segment detector with a false detection control. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(4):722–732, 2010. ISSN 1939-3539.

[19] Stephen Hausler and Michael Milford. Hierarchical multi-process fusion for visual place recognition. *arXiv preprint arXiv:2002.03895*, 2020.

[20] Sheraz Khan and Dirk Wollherr. IBuILD: Incremental bag of binary words for appearance based loop closure detection. In *IEEE International Conference on Robotics and Automation*, pages 5441–5447, 2015.

[21] M. Labbé and F. Michaud. Appearance-based loop closure detection for online large-scale and long-term operation. *IEEE Transactions on Robotics*, 29(3):734–745, 2013.

[22] K. Li, J. Yao, M. Xia, and L. Li. Joint point and line segment matching on wide-baseline stereo images. In *IEEE Winter Conference on Applications of Computer Vision*, pages 1–9, 2016.

[23] H. Lim, J. Lim, and H. J. Kim. Real-time 6-DOF monocular visual SLAM in a large-scale environment. In *IEEE International Conference on Robotics and Automation*, pages 1532–1539, 2014.

[24] David G Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[25] S. Lowry, N. Sünderhauf, P. Newman, J. J. Leonard, D. Cox, P. Corke, and M. J. Milford. Visual place recognition: A survey. *IEEE Transactions on Robotics*, 32(1): 1–19, 2016.

[26] M. J. Milford and G. F. Wyeth. SeqSLAM: visual route-based navigation for sunny summer days and stormy winter nights. In *IEEE International Conference on Robotics and Automation*, pages 1643–1649, 2012.

[27] R. Mur-Artal and J. D. Tardós. Fast relocalisation and loop closing in keyframe-based SLAM. In *IEEE International Conference on Robotics and Automation*, pages 846–853, 2014.

[28] 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*, 33(5):1255–1262, 2017.

[29] David Nister and Henrik Stewenius. Scalable recognition with a vocabulary tree. In *International Conference on Computer Vision and Pattern Recognition*, volume 2, pages 2161–2168, 2006. ISBN 0769525970.

[30] Oscar A. Pellejero, Carlos Sagüés, and J. Jesús Guerrero. Automatic computation of the fundamental matrix from matched lines. In *Current Topics in Artificial Intelligence*, pages 197–206. Springer, 2004. ISBN 978-3-540-25945-9.

[31] A. Pumarola, A. Vakhitov, A. Agudo, A. Sanfeliu, and F. Moreno-Noguer. PL-SLAM: real-time monocular visual SLAM with points and lines. In *IEEE International Conference on Robotics and Automation*, pages 4503–4508, 2017.

[32] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. ORB: An efficient alternative to SIFT or SURF. In *International Conference on Computer Vision*, pages 2564–2571, 2011.

[33] Seyoon Jeong, Kyuheon Kim, Byungtae Chun, Jaeyeon Lee, and Y. J. Bae. An effective method for combining multiple features of image retrieval. In *IEEE Region 10 Conference. TENCON*, volume 2, pages 982–985 vol.2, 1999.

[34] Sivic and Zisserman. Video google: a text retrieval approach to object matching in videos. In *International Conference on Computer Vision*, pages 1470–1477 vol.2, 2003.

[35] Elena S. Stumm, Christopher Mei, and Simon Lacroix. Building location models for visual place recognition. *International Journal of Robotics Research*, 35(4):334–356, 2016.

[36] N. Sünderhauf and P. Protzel. BRIEF-Gist - closing the loop by simple means. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1234–1241, 2011.

[37] N. Sünderhauf, S. Shirazi, F. Dayoub, B. Upcroft, and M. Milford. On the performance of ConvNet features for place recognition. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4297–4304, 2015.

[38] K. A. Tsintotas, L. Bampis, and A. Gasteratos. Assigning visual words to places for loop closure detection. In *IEEE International Conference on Robotics and Automation*, pages 5979–5985, 2018.

[39] K. A. Tsintotas, L. Bampis, and A. Gasteratos. Probabilistic appearance-based place recognition through bag of tracked words. *IEEE Robotics and Automation Letters*, 4 (2):1737–1744, 2019. ISSN 2377-3774.

[40] Fukai Zhang, Ting Rui, Chengsong Yang, and Jianjun Shi. Lap-SLAM: A line-assisted point-based monocular VSLAM. *Electronics*, 8(2):243, 2019.

[41] G. Zhang and I. H. Suh. Building a partial 3d line-based map using a monocular SLAM. In *IEEE International Conference on Robotics and Automation*, pages 1497–1502, 2011.

[42] Guangcong Zhang, Mason J Lilly, and Patricio A Vela. Learning binary features online from motion dynamics for incremental loop-closure detection and place recognition. In *IEEE International Conference on Robotics and Automation*, pages 765–772, 2016.

[43] Lilian Zhang and Reinhard Koch. An efficient and robust line segment matching approach based on LBD descriptor and pairwise geometric consistency. *Journal of Visual Communication and Image Representation*, 24(7):794 – 805, 2013. ISSN 1047-3203.

[44] H. Zhou, D. Zou, L. Pei, R. Ying, P. Liu, and W. Yu. StructSLAM: visual SLAM with building structure lines. *IEEE Transactions on Vehicular Technology*, 64(4):1364–1375, 2015.

[45] X. Zuo, X. Xie, Y. Liu, and G. Huang. Robust visual SLAM with point and line features. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1775–1782, 2017.

MSC-VO: Exploiting Manhattan and Structural Constraints for Visual Odometry

Joan P. Company-Corcoles, Emilio Garcia-Fidalgo, and Alberto Ortiz, *Member, IEEE*

***Abstract*—Visual odometry algorithms tend to degrade when facing low-textured scenes —from e.g. human-made environments—, where it is often difficult to find a sufficient number of point features. Alternative geometrical visual cues, such as lines, which can often be found within these scenarios, can become particularly useful. Moreover, these scenarios typically present structural regularities, such as parallelism or orthogonality, and hold the Manhattan World assumption. Under these premises, in this work, we introduce MSC-VO, an RGB-D -based visual odometry approach that combines both point and line features and leverages, if exist, those structural regularities and the Manhattan axes of the scene. Within our approach, these structural constraints are initially used to estimate accurately the 3D position of the extracted lines. These constraints are also combined next with the estimated Manhattan axes and the reprojection errors of points and lines to refine the camera pose by means of local map optimization. Such a combination enables our approach to operate even in the absence of the aforementioned constraints, allowing the method to work for a wider variety of scenarios. Furthermore, we propose a novel multi-view Manhattan axes estimation procedure that mainly relies on line features. MSC-VO is assessed using several public datasets, outperforming other state-of-the-art solutions, and comparing favourably even with some SLAM methods.**

***Index Terms*—Localization, mapping, SLAM.**

I. Introduction

VISUAL Odometry (VO) is the process of estimating the trajectory of a camera within an environment by analysing the sequence of images captured. VO is a key part of a more sophisticated family of methods known as Visual Simultaneous Localization and Mapping (V-SLAM), which typically combine VO with a loop closure detection approach to perform both tasks at the same time. When a previously seen place is revisited, the accumulated drift produced by VO can be alleviated incorporating new constraints into the optimization stage. However, this strategy does not completely remove the camera pose error, so that the overall performance of any SLAM system gets determined by the VO accuracy [1].

Manuscript received September 9, 2021; accepted January 1, 2022. Date of publication January 13, 2022; date of current version February 2, 2022. This work is partially supported by EU-H2020 projects BUGWRIGHT2 (GA 871260) and ROBINS (GA 779776), and by project PGC2018-095709-B-C21 (funded by MCIU/AEI/10.13039/501100011033 and FEDER "Una manera de hacer Europa"). This publication reflects only the authors views and the European Union is not liable for any use that may be made of the information contained therein. *(Corresponding author: Joan P. Company-Corcoles.)*

The authors are with the Department of Mathematics and Computer Science, University of the Balearic Islands, 07122 Palma, Spain, and also with the IDISBA, Institut d'Investigacio Sanitaria de les Illes Balears, 07120 Palma de Mallorca, Spain (e-mail: joanpep.company@uib.es; emilio.garcia@uib.es; alberto.ortiz@uib.es).

Digital Object Identifier 10.1109/LRA.2022.3142900

Many VO and SLAM systems rely on point features because of their wider applicability in general terms [2]. However, in low-textured scenarios, their performance decrease due to the low number of points detected [3]. In this regard, the combination of point and line features has been demonstrated to reduce the number of tracking failures in these environments [3]–[5]. A complementary technique is to take profit of the structural constraints typically present in these scenarios, such as parallelism and/or orthogonality, through a pose-graph optimization strategy [6]. Another well-known strategy, which can be used to reduce the rotation drift in human-made environments, is to adopt the Manhattan World (MW) assumption [7]. This hypothesis assumes a Cartesian coordinate system for the environment and that most part of the geometrical entities present in the scene align to one of its axes, named as Manhattan Axes (MA). This assumption is fundamentally used during the tracking stage [8]–[12]. Nonetheless, these methods do not usually take into account that some indoor environments are not strictly adhering to this assumption, leading to degradation in accuracy or even to tracking failures [13].

Based on the above, this work exploits the benefits of point and line features used in combination with structural constraints and MA alignment to propose a new RGB-D VO framework named as MSC-VO from *Manhattan and Structural Constraints - Visual Odometry*. As already said, the proposed method relies on point and line features, mostly because of their low detection times. Additionally, to address the inaccuracies in depth estimation which result from occlusions, depth discontinuities and RGB-D noise, which is even more notorious for lines than for points, we propose a two-step procedure that can be briefly stated as (1) for each line detected in the image plane, we estimate its 3D line endpoints using a robust fitting procedure, and (2) we next refine the estimated endpoints using the scene structural regularities. Moreover, our approach proposes a novel local map optimization stage which combines point and line reprojection errors along with structural regularities and MA alignment, resulting into more precise local trajectory estimations. Unlike other approaches, where the MW constraints are used during the tracking stage, our solution incorporates the MW assumption during local map optimization, which allows us not to slow down the tracking, which typically requires real-time operation to perform properly. Finally, we propose a novel multi-view MA initialization procedure. A first illustration of the performance of MSC-VO can be found in Fig. 1.

In brief, the most important contributions of this work are:

1) A robust RGB-D VO framework for low-textured environments, which can improve the pose accuracy when

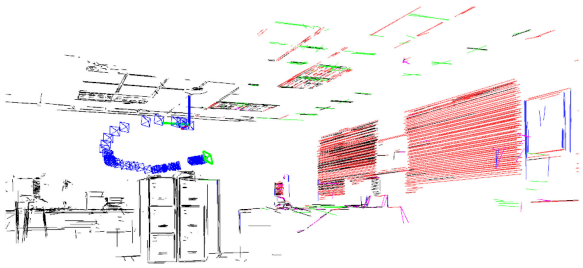


Fig. 1. Example of local map generated by MSC-VO. For a better understanding, only line features are shown. The map corresponds to a human-made environment, which, as expected, is rich in line features. Furthermore, parallel and orthogonal relations between lines are highly present due to the design of these environments. The Manhattan axis line associations are shown using red, green and blue colours, while non-associated lines are labelled in purple- Those lines not included in the covisibility graph are shown in black.

structural regularities and MA alignment are present in the scene. Otherwise, our solution remains operational, as will be shown in the experimental results section.

2) A 3D line endpoint computation method based on the structural information present in the scene.
3) An accurate and efficient 3D local map optimization strategy, which combines reprojection errors with structural constraints and MA alignment.
4) A novel MA initialization procedure that refines the estimation of the traditionally employed Mean Shift algorithm by using multiple frame observations in a multi-graph non-linear least squares formulation.
5) An extensive evaluation of the proposed approach on several public datasets and a comparison with other VO and SLAM state-of-the-art methods.
6) As an additional contribution, the source code MSC-VO is available online for the community.[1]

The rest of the paper is organized as follows: Section II overviews most relevant related works in the field; the proposed framework is introduced in Section III; Section IV reports on the results obtained; and, finally, Section V concludes the paper and suggests some future research lines.

II. Related Work

VO and Visual SLAM algorithms can be roughly classified into two main categories: feature-based and direct methods [1]. Among them, feature-based approaches are typically more robust to illumination changes than direct methods. Despite their impressive results on well-textured scenarios [2], their performance decreases when dealing with low-textured environments [4]. Due to this reason, some authors have opted for the combination of points with other geometric entities, e.g. lines [3]–[5], planes [6], or both [13].

Assuming a MW in human-made environments has demonstrated to be very effective to reduce the rotational drift [8]–[12]. Generally, this premise is taken into account during the tracking stage, being usually decoupled the rotation and the translation parts. Different strategies have been proposed to estimate and track the MA. For example, Zhou *et al.* [8] propose a single Mean Shift iteration that tracks the dominant MA for each frame by using a set of normal vectors. The translational part is computed through three simple 1D density alignments. In [9], the translation estimation is improved through a Kanade-Lucas-Tomasi (KLT) feature tracker. However, these two approaches require the existence of multiple orthogonal planes per frame. To solve this issue, Kim *et al.* [10] combine line and plane features within a Mean Shift-based approach. In addition, they propose to use the reprojection error from the tracked points in the estimation of the translation. In a more recent work, they add an orthogonal plane detection and tracking method [11]. Another solution to improve the tracking accuracy is presented in [12], where the authors introduce the concept of plane orientation relevance to track the MA. More recently, other features are employed in [14], which combines vanishing directions of 3D lines and plane normal vectors to track the MA. In this regard, [13], [14] report that the use of planar features increases the accuracy of the tracking, and, additionally, contributes positively to the estimation of the MA. However, plane detection usually relies on depth estimation, which can fail in some scenarios due to the range limitations and noise of RGB-D cameras [13]; contrarily, line features can be detected directly from the available images. Besides, planes and lines detection require similar computational times if the number of planes is not high; otherwise, the complexity of the underlying processes leads to larger running times for planes. Additionally, to detect and track the MA robustly, these methods typically combine planes with other features, such as lines. Consequently with the aforementioned, our pipeline combines points and lines.

Moreover, the accuracy of the estimated MA determines the correctness of the system during its operation. To reduce these inaccuracies, Li *et al.* [15] describe a method that refines the reference MA by tracking it on each frame, and, thus, obtains multiple reference MA, which are later fused by Kalman Filtering. Following this idea, we propose to refine the position of 3D lines during MA estimation by using a graph-based non-linear error function that includes multiple views of the lines. However, unlike [15], we estimate the MA only once and they remain fixed along the whole sequence.

Local map optimization is usually performed in the back-end to reduce the errors produced during the tracking stage. In this regard, some approaches refine the pose of some previous frames after tracking the MA. For instance, in [16], the authors propose a line-based local optimization method to refine only the translation. However, the rotation is still computed using the decoupled tracking strategy. Moreover, other approaches [6], [14] perform this local optimization by combining point and plane features in conjunction with structural constraints, which have been shown to achieve better results than the decoupled scheme [14].

There exist indoor environments that do not strictly conform to the MW assumption. In these cases, the performance of approaches purely based on it degrades, even leading to tracking failures. To overcome this issue, Zhang *et al.* [6] propose using parallel and perpendicular constraints as an alternative to the MW assumption. Despite its advantages, this method can not reduce the long-term rotation error as the MW assumption does. Another solution is proposed in [13], where the authors use either a decoupled or a non-decoupled tracking strategy depending on whether the scene meets the MW assumption. These strategies permit these works to not only focus on a specific environment.

[1] [Online]. Available: http://github.com/joanpepcompany/MSC-VO

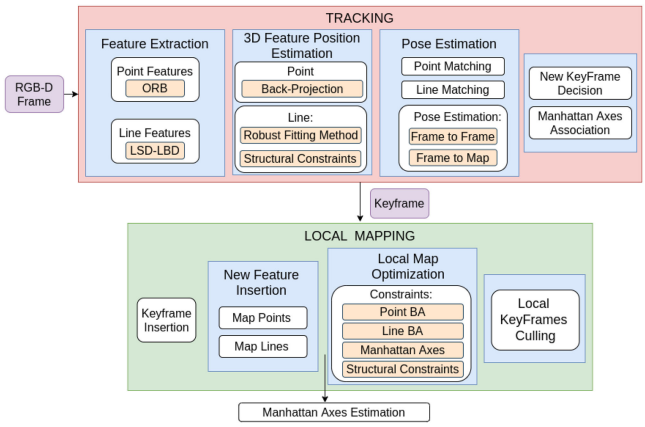


Fig. 2. Overview of MSC-VO.

The related works reviewed above suggest the use of the MW assumption to increase the localization accuracy of VO and SLAM methods. However, using this assumption as a primary source in the tracking procedure can lead to failures in some scenes where the MW assumption is not satisfied, what can restrict those solutions for certain specific environments. As a solution, we propose the incorporation of the MA in local map optimizations. Additionally, we take inspiration from [14], which reports the structural constraints as beneficial for the pose refinement process. To this end, we propose a novel local map optimization approach that combines the point and line reprojection error, the MA alignment and the structural constraints of the scene. Allowing that, the punctual dissatisfaction of some of these constraints does not affect the overall performance. As a result, our method leads to higher localization accuracy and allows working in a wider range of scenarios.

III. MSC-VO Overview

MSC-VO is built on top of the tracking and local mapping components of ORB-SLAM2 [2]. Therefore, it comprises two threads running in parallel, as it is illustrated in Fig. 2. Further details on MSC-VO can be found next.

A. Tracking

The tracking thread is in charge of estimating the position of every frame captured. Additionally, this module decides whether a new keyframe needs to be created. It also associates each new map line with one of the MA, if possible.

1) Feature Extraction: Every frame I_t coming from the RGB-D sensor at time t consists of a colour image I_t^c and a depth image I_t^d. Point and line features are extracted from I_t^c. Points are detected and described using ORB [17], while lines are detected using the Line Segment Detector (LSD) [18] and described using the binary form of the Line Band Descriptor (LBD) [19]. In the following, the location of a point i in image coordinates is denoted as p_i, while each line segment j detected in the image plane is represented by a start point s_j and an end point e_j. Additionally, the normalized line l_j is expressed as:

$$l_j = \frac{e_j - s_j}{\|e_j - s_j\|}. \tag{1}$$

2) 3D Feature Position Estimation: Once points and lines have been detected and described, their 3D positions in camera coordinates are obtained. A point p_i is backprojected using as depth the value corresponding to its 2D position in I_t^d. The resulting 3D position in camera coordinates is denoted as P_i^c. Since lines are more affected than points by depth discontinuities and occlusions, this simple procedure can end up with inaccurate 3D lines. To reduce this effect, we propose a robust two-step method to compute the 3D line endpoints.

First, for every line segment j, we calculate an initial 3D position for its endpoints, denoted by $\{S_j^c, E_j^c\}$, by backprojecting a subset of the points that conforms the line in the image and, next, performing a robust fitting step as in [14]. The 3D normalized line L_j^c is computed similarly to (1). Next, we employ the structural constraints of the scene to refine each detected line. We start by associating parallel and perpendicular lines. To this end, for every possible pair of lines (L_m^c, L_n^c) detected in the current image, we compute the cosine of the angle between the two direction vectors by means of the dot product:

$$\cos\left(L_m^c, L_n^c\right) = \frac{L_m^c \cdot L_n^c}{\|L_m^c\| \|L_n^c\|}. \tag{2}$$

We choose only those pairs (L_m^c, L_n^c) whose cosine value is close to 0 or 1 representing, respectively, perpendicular or parallel lines. The selected pairs are employed to refine their line endpoints by means of non-linear optimization. We use the Levenberg–Marquardt algorithm implemented in g2o [20] to this end. Formally, we define the orientation discrepancy d between lines L_m^c and L_n^c as:

$$d(L_m^c, L_n^c) = \left|\cos\left(L_m^c, L_n^c\right)\right|. \tag{3}$$

Let us denote $\mathbb{L}_\perp$ and $\mathbb{L}_\parallel$ as the sets of, respectively, valid perpendicular and valid parallel line pairs. Given a pair $(L_m^c, L_n^c) \in \mathbb{L}_\perp$, the error term $\mathbf{E}_{m,n}^\perp$ is defined as:

$$\mathbf{E}_{m,n}^\perp = d(L_m^c, L_n^c) \cdot \omega_n^{-1}, \tag{4}$$

where ω_n weights the error term in accordance to the line response returned by the LSD algorithm for segment n. Similarly, for another pair $(L_m^c, L_n^c) \in \mathbb{L}_\parallel$, the error term $\mathbf{E}_{m,n}^\parallel$ is defined as follows:

$$\mathbf{E}_{m,n}^\parallel = \sqrt{1 - d^2(L_m^c, L_n^c)} \cdot \omega_n^{-1}. \tag{5}$$

where $d(\cdot,\cdot) \in [0,1]$.

We define $\mathbf{L}$ as the set of variables to be optimized, which includes those lines that have at least one structural association either on $\mathbb{L}_\perp$ or $\mathbb{L}_\parallel$. We then compute the optimal line end points of $\mathbf{L}$ by minimizing the following cost function:

$$\mathbf{L} = \underset{\mathbf{L}}{\operatorname{argmin}} \left(\sum_{(i,j)\in\mathbb{L}_\perp} \rho\left(\mathbf{E}_{i,j}^\perp\right) + \sum_{(k,o)\in\mathbb{L}_\parallel} \rho\left(\mathbf{E}_{k,o}^\parallel\right) \right), \tag{6}$$

where ρ is the Huber loss function to reduce the influence of outliers. Fig. 3 summarizes the notation of points and lines regarding frame coordinates, and the two error terms defined in this section. As it will be shown in Section IV, using the outlined procedure, the 3D lines estimation accuracy improves, benefiting the whole system.

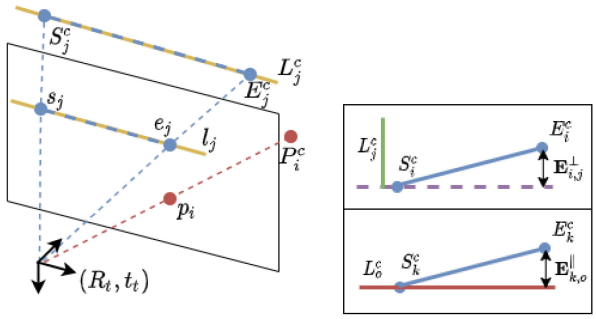


Fig. 3. The left drawing illustrates the notation used for 2D and 3D features, while the right drawings illustrate the line endpoints error terms. S_j^c and E_j^c are the line endpoints to optimize. The right-top drawing shows the error term $\mathbf{E}_{i,j}^{\perp}$ as the cosine of the angle between the normalized line defined by S_i^c and E_i^c and a perpendicular line, shown in green, for a perpendicular association L_j^c. The right-bottom drawing illustrates the parallel error term $\mathbf{E}_{k,o}^{\parallel}$, calculated as the sine of the angle between the normalized line defined by S_k^c and E_k^c and a parallel association L_o^c. (Both cases assume $\omega = 1$.).

3) Pose Estimation: Once features are extracted, an optimization procedure is carried out to estimate the current camera orientation $\mathbf{R}_t \in SO(3)$ and translation $\mathbf{t}_t \in \mathbb{R}^3$. Initially, map points and lines observed in the previous frame are projected to the current frame, assuming a constant velocity motion model. Next, two sets of 2D-3D correspondences, one for points as in [2] and one for lines as in [5], are computed. These associations are then employed to optimize the current camera pose, minimizing the following cost function:

$$\{\mathbf{R_t}, \mathbf{t_t}\} = \underset{\mathbf{R_t}, \mathbf{t_t}}{\operatorname{argmin}} \left(\sum_{i \in \mathbb{P}} \rho\left(\mathbf{E}_i^p\right) + \sum_{j \in \mathbb{V}} \rho\left(\mathbf{E}_j^l\right) \right), \quad (7)$$

where $\mathbb{P}$ and $\mathbb{V}$ are, respectively, the sets of all point and line matches. The error term for the observation of a map point i is defined as:

$$\mathbf{E}_i^p = \| p_i - \pi(R_t P_i^w + t_t) \|^2 \cdot \omega_i^{-1}, \quad (8)$$

where $P_i^w \in \mathbb{R}^3$ is the point in world coordinates corresponding to $p_i \in \mathbb{R}^2$ and ω_i weights the error term in accordance to the response of the ORB detector. The projection function π transforms a 3D point P_i^c in camera coordinates into the image plane using the camera calibration parameters [21]. On the other side, the error term for an observed map line j in the current frame is defined as:

$$\mathbf{E}_j^l = \left\| n_j \cdot \pi(R_t S_j^w + t_t),\ n_j \cdot \pi(R_t E_j^w + t_t) \right\|^2 \cdot \omega_j^{-1}, \quad (9)$$

where $L_j^w = \{S_j^w, E_j^w\}$ is the map line in world coordinates that matches the 2D segment l_j with normal vector n_j. Once the camera pose has been estimated, we project the local map into the current frame to obtain more correspondences, as performed in [2]. The pose is optimized again with the resulting matches.

4) Keyframe Insertion: Once the camera pose has been estimated, the current frame is evaluated to decide whether it should be considered as a new keyframe. We use a similar policy as ORB-SLAM2 [2], but incorporating line correspondences. Unlike ORB-SLAM2, we do not use the condition of a minimum number of features tracked. The rationale behind this idea is that the proposed method is focused on low-textured environments, where typically the number of features tracked per frame can change drastically between scenes. Therefore, it is not possible to fix a reasonable threshold. Instead, we propose to use the ratio between the current frame features that are being tracked in the map, and the sum of these features with the ones that could be potentially created. Once a new keyframe is generated, points and lines are included in the local map and redundant features are culled, as performed in [2]. For each new map line, we search for parallel or perpendicular line correspondences in the local map. Additionally, each line is also associated to an MA, if possible, as explained in the next section.

5) Manhattan Axes Association: Given $\mathcal{M} = \{\mathrm{MA}_0, \mathrm{MA}_1, \mathrm{MA}_2\}$ as the set of Manhattan Axes, when a new keyframe is inserted, each new map line j is associated to axis $M_j \in \mathcal{M}$ whenever possible. To this end, we compare every line L_j^w with each of the three axes: if the value of expression in (3) gets close enough to 1 for axis MA_k, the line is considered as parallel to MA_k, and they are matched, i.e. $M_j = \mathrm{MA}_k$. These associations are used during local map optimization to reduce the camera rotation drift. Notice that, given the combination of structural constraints and this MA alignment, our approach is able to operate even if these axes are not available. The procedure to estimate these MA is explained in Section III-B2.

B. Local Mapping

Whenever a keyframe is inserted, the local mapping thread refines recent keyframe poses and landmarks by a multi-graph optimization process. Furthermore, this thread also estimates the reference MA, if required. Finally, redundant keyframes are culled using the strategy introduced in [2]. Further details can be found next.

1) Local Map Optimization: Once keyframe k is generated, the local optimization procedure refines its pose along with the poses of a set of connected keyframes $\mathcal{K}_c$ obtained from a covisibility graph [2] and all the map points $\mathcal{P}$ and lines $\mathcal{L}$ seen by those keyframes. Other keyframes that observe these points and lines but are not connected to k, denoted by $\mathcal{K}_f$, are included in the optimization, but their poses remain fixed. We denote $\mathbb{P}_k$ and $\mathbb{V}_k$ as the sets of matches between, respectively, points and lines in $\mathcal{P}$ and $\mathcal{L}$ and features in keyframe k. To introduce the structural constraints of the scene into the optimization, we define $\mathbb{L}_{\perp}^k$ and $\mathbb{L}_{\parallel}^k$ as the sets of perpendicular and parallel pairs of lines in $\mathcal{L}$, respectively, co-observed in keyframe k. Finally, we denote as $\mathbb{M}$ the set of map lines that are associated to a MA and that are seen by any keyframe in $\mathcal{K}_c$. Defining $\Gamma = \{P_i^w, L_j^w, R_l, t_l, | i \in \mathcal{P}, j \in \mathcal{L}, l \in \mathcal{K}_c\}$ as the set of variables to be estimated, the optimization problem is defined as:

$$\begin{aligned} \mathbf{\Gamma} = \underset{\mathbf{\Gamma}}{\operatorname{argmin}} \Bigg[& \sum_{k \in \{\mathcal{K}_c \cup \mathcal{K}_f\}} \left(\sum_{i \in \mathbb{P}_k} \rho\left(\mathbf{E}_i^p\right) + \sum_{j \in \mathbb{V}_k} \rho\left(\mathbf{E}_j^l\right) \right) \\ & + \sum_{z \in \mathcal{K}_c} \left(\sum_{(i,j) \in \mathbb{L}_{\perp}^z} \rho\left(\mathbf{E}_{i,j}^{\perp}\right) + \sum_{(i,j) \in \mathbb{L}_{\parallel}^z} \rho\left(\mathbf{E}_{i,j}^{\parallel}\right) \right) \\ & + \sum_{j \in \mathbb{M}} \rho\left(\mathbf{E}_{j,M_j}^{\parallel}\right) \Bigg] \end{aligned} \quad (10)$$

where $\mathbf{E}^{\perp}_{i,j}$, $\mathbf{E}^{\parallel}_{i,j}$, $\mathbf{E}^{p}_{i}$ and $\mathbf{E}^{l}_{j}$ were respectively defined in (4), (5), (8) and (9), and the MA alignment error $\mathbf{E}^{\parallel}_{j,M_j}$ is the error term corresponding to a map line j and its associated Manhattan axis $M_j \in \mathcal{M}$, calculated using (5).

2) Manhattan Axes Estimation: As already said, the Manhattan Axes comprise a set of three orthogonal directions, in world coordinates, which represent the main scene directions. These directions remain fixed over time and, therefore, the MA extraction procedure is performed only once during the whole sequence. Their estimation should be very accurate to prevent misalignments during optimization steps. In this respect, this work proposes a coarse-to-fine MA estimation strategy, where the estimation at the coarsest level is obtained extending the work by Kim *et al.* [10]. The estimated MA are then refined by considering multiple line observations along different keyframes.

For a start, a first estimation of the MA is computed from the first keyframe once it is available using the Mean Shift-based method proposed in [10]. In this first stage, the only features involved are the line direction vectors and the surface normal vectors for a selection of points defined over a grid. The normal vectors are calculated using a modified version of the approach proposed in [22], which is based on integral images to speed up calculations. This procedure is repeated for the next keyframes until valid, though typically noisy, MA are obtained.

Once the local map comprises a sufficient number of keyframes, being denoted by $\mathcal{K}_M$, a non-linear optimization procedure is performed in a second MA refinement stage, using hence the inaccurate MA computed in the first stage as initial guess. Given $\mathcal{M}$ as the set of MA, and defining $\mathbb{V}_k^{\mathrm{MA}_i}$ as the set of map lines associated to the Manhattan axis MA_i observed in keyframe k, the optimization problem can be stated as follows:

$$\mathcal{M} = \underset{\mathcal{M}}{\operatorname{argmin}} \sum_{k \in \mathcal{K}_M} \left(\sum_{j \in \mathbb{V}_k^{\mathrm{MA}_0}} \rho(\mathbf{E}_j^{\mathrm{MA}_0}) + \sum_{j \in \mathbb{V}_k^{\mathrm{MA}_1}} \rho(\mathbf{E}_j^{\mathrm{MA}_1}) + \sum_{j \in \mathbb{V}_k^{\mathrm{MA}_2}} \rho(\mathbf{E}_j^{\mathrm{MA}_2}) \right), \tag{11}$$

where the error term of a line j associated to the axis $M_j \in \mathcal{M}$ is given by:

$$\mathbf{E}_j^{M_j} = \mathbf{E}^{\parallel}_{j,M_j} + \mathbf{E}^{\perp}_{j,M_{j'}} + \mathbf{E}^{\perp}_{j,M_{j''}}, \tag{12}$$

being $M_{j'}$ and $M_{j''}$ the two other MA non-associated to line j. These two last terms enforce the orthogonality among the finally resulting axes. We reduce further the orthogonality error of the MA by means of Singular Value Decomposition (SVD), as also performed in [9], [10], [13].

IV. Experimental Results

To demonstrate the performance of MSC-VO, we conduct various experiments in both synthetic and real image sequences. Additionally, we compare its localization accuracy with some state-of-the-art VO and visual SLAM systems by means of the following datasets:

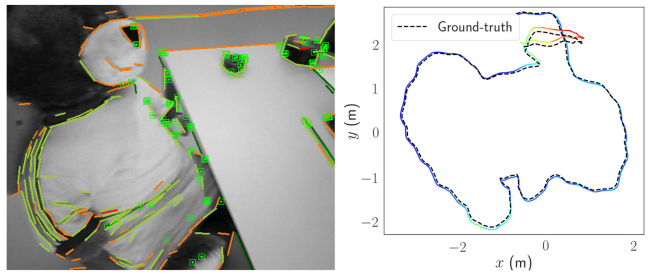


Fig. 4. (left) The MA maybe absent in a scene, e.g. a frame of the *fr3-longoffice* sequence. (right) Trajectory estimated by MSC-VO for this sequence, where no tracking failures are observed.

1) *ICL-NUIM [23]:* This is a synthetic dataset which comprises two main scenes, the living room and the office, coined in our experiments as lr and of, respectively. Furthermore, this is an indoor dataset with large structured areas, where the MW assumption and the structural constraints are highly present. Additionally, this dataset involves some low-textured challenging elements such as floors, ceilings and walls.
2) *TUM RGB-D benchmark [24]:* This is also an indoor dataset that contains several sequences with different structure, illumination and texture conditions. Unlike ICL-NUIM, this is a noisy dataset since a real RGB-D sensor was used.
3) *TAMU RGB-D [25]:* This dataset contains several indoor sequences, among which we employ *Corridor-A* and *Entry-Hall* to validate the final trajectory error (the travel distances are, respectively, 82 m and 54 m).

Regarding the MSC-VO parameters, we have used the default values provided by ORB-SLAM2 authors for the common parts, whereas the remaining parameters have been set experimentally from a single dataset, and they have been kept unaltered for the rest of sequences.

To evaluate the overall performance of MSC-VO, for the ICL-NUIM dataset and the TUM RGB-D benchmark, we use the Root-Mean-Square Error (RMSE) of the Absolute Trajectory Error (ATE) expressed in meters, as computed by the RGB-D TUM benchmark tools [24]. Regarding the TAMU RGB-D dataset, we provide the Trajectory Endpoint Drift (TED) [25], computed as the Euclidean distance between the starting and end points of the path. All the experiments have been performed on an Intel Core i7-9750H @ 2.60GHz / 16 GB RAM, without GPU parallelization.

A. General Performance

For a start, Fig. 4 illustrates the fact that the MA may be absent in a scene, leading to tracking failures for some solutions. In the case of MSC-VO, the fact of involving the MA only in local map optimizations can prevent these failures from occurring. In Fig. 4 (left), we show a frame of the *fr3-longoffice* sequence, for which the MW assumption is not very appropriate. In the image, green, red and blue colours denote the correspondences of a line with a single Manhattan axis, whereas yellow is for 3D lines that do not correspond to any axis and orange is for lines

TABLE I
RMSE OF THE ATE OF MSC-VO (IN METERS)

Sequence	PL-VO	PL-VO-Depth	MSC-VO-OR	MSC-VO
lr-kt0	0.051	0.024	0.012	**0.006**
lr-kt1	0.064	0.048	0.013	**0.010**
lr-kt2	0.054	0.030	0.010	**0.009**
lr-kt3	0.061	0.057	0.040	**0.038**
of-kt0	0.047	0.032	0.030	**0.028**
of-kt1	0.056	0.053	0.025	**0.017**
of-kt2	0.040	0.039	0.019	**0.014**
of-kt3	0.042	0.038	0.031	**0.010**
fr1-xyz	0.015	0.013	0.012	**0.010**
fr1-desk	0.023	0.022	0.024	**0.019**
fr2-xyz	0.011	0.009	0.006	**0.005**
fr2-desk	0.121	0.060	**0.023**	**0.023**
large-cabinet	0.173	0.152	0.131	**0.120**
fr3-longoffice	0.108	0.096	0.034	**0.022**

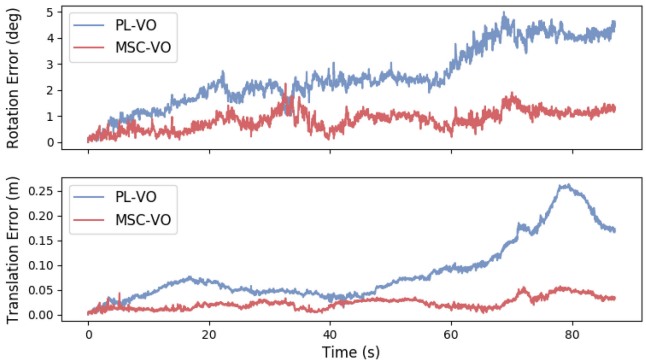


Fig. 5. Rotation and translation error over time for PL-VO and MSC-VO on the *fr3-longoffice* dataset.

TABLE II
TED ON THE TAMU RGB-D DATASET (IN METERS)

Sequence	PL-VO	PL-VO-Depth	MSC-VO-OR	MSC-VO
Corridor-A	2.76	2.29	1.38	**0.91**
Entry-Hall	1.89	1.70	1.26	**1.07**

whose 3D position has not been estimated. Fig. 4 (right) shows that MSC-VO can estimate the whole trajectory.

Next, we compare several versions of MSC-VO to show the effect of the different contributions: *PL-VO* is the part of MSC-VO that just combines point and line features; *PL-VO-Depth* combines PL-VO with the proposed 3D line endpoint estimation method; *MSC-VO-OR* corresponds to a modified version of the proposed solution, where, if a line is associated with an MA and, at the same time, it includes structural constraints, only the MA constraints are considered during the optimization (10); finally, the last case is the full version of MSC-VO. Estimation performance results for multiple sequences can be found in Table I. Moreover, Fig. 5 illustrates the rotation and translation error over time for PL-VO and MSC-VO on the *fr3-longoffice* dataset. Taking PL-VO as the baseline, MSC-VO reduces on average 76.5% and 80% the rotation and translation errors for this dataset.

Table II reports on the TED for each version of MSC-VO for the TAMU RGB-D dataset to assess its performance in long sequences. It is noticeable that each variation of our approach helps to reduce the accumulated drift along the trajectory.

TABLE III
MEAN EXECUTION TIMES (TUM RGB-D BENCHMARK)

Mean Executxion Time (ms)				
Tracking			Local Mapping	
Feat. Extrac. and 3D Pose Estimation	Camera Pose Estimation	Total (Hz)	Local Map Optimization	MA Estimation
23.2	29.1	18	152.6	206.6

TABLE IV
COMPARISON WITH OTHER APPROACHES (TED IN METERS)

Sequence	MSC-VO	ManhSLAM [13]	ORB-SLAM2 [2]
Corridor-A	0.91	**0.51**	3.17
Entry-Hall	**1.07**	1.52	2.18

On the other side, Fig. 6 shows local maps from the same cases as above for the *fr3-longoffice* sequence. The first and second plots result from, respectively, PL-VO and PL-VO-Depth. In the former case, noise from lines depth calculation affects the local map and, consequently, also the pose estimation accuracy. In the second case, this noise is of a lower magnitude, but pose inaccuracies are still observed. The third plot results from MSC-VO with the best local map and the highest localization accuracy. These results show that the local map optimization procedure not only improves the camera pose accuracy, but also refines the map lines. As a result, the misalignment that affects the PL-VO-Depth case is notably reduced. To conclude, the fourth plot shows the trajectories from each approach together with the ground truth, for a further understanding of the pose accuracy achievable on each case.

To finish, average running times for the main stages of MSC-VO can be found in Table III. The averages result from three different sequences of the TUM RBG-D benchmark. As expected, adding line features into point based VO or SLAM methods improves the accuracy and the robustness, though at the expense of increasing the computational complexity [4]. In more detail regarding our solution: (1) the robust fitting method used for 3D line pose estimation increases the low times required to extract line features and adds execution time to the feature extraction stage over other solutions; (2) regarding MA estimation, its execution time is high due to 180.4 ms that are required by the coarsest estimation step, although it needs to be computed only once (in scenarios where the MW assumption holds); and (3) despite local map optimizations require more time than other, more traditional methods based on local bundle adjustment, it can still be fast enough, as they run in a parallel thread. As a general comment, the final frame rate achieved is around 18 Hz.

B. Comparison With Other Solutions

Table V compares MSC-VO regarding localization accuracy with other state-of-the-art approaches, for which the results reported in the original works are reproduced. Best performances are indicated in bold, whereas the second best is shown in bold blue, *n.a.* refers to a not-available value, and × reports a tracking failure. The left side of the table reports on solutions based on the MA assumption that do not perform any global map optimization or loop closure detection (LCD), while the right

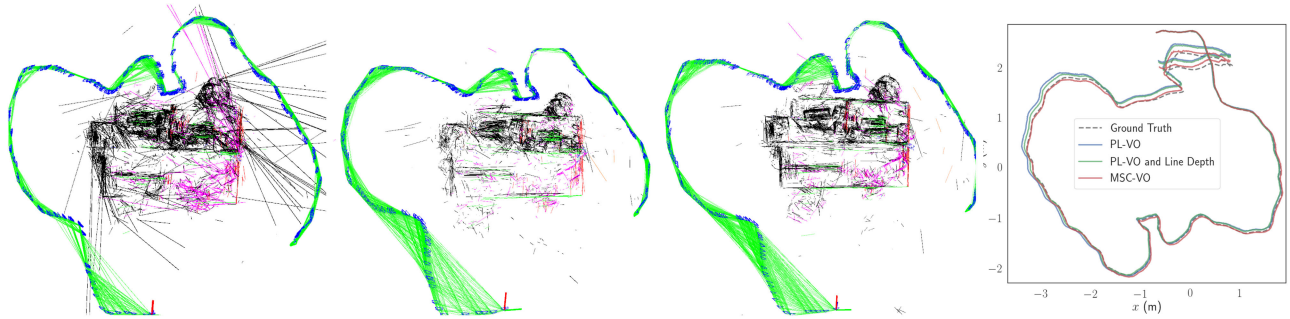


Fig. 6. (left) Local maps for the *fr3-longoffice* sequence and different versions of MSC-VO: 1st – only using points and lines (PL-VO), 2nd – PL-VO using the proposed line depth extraction procedure (PL-VO-Depth), 3rd – full MSC-VO. (right) 2D trajectories for PL-VO, PL-VO-Depth and MSC-VO, respectively shown in blue, green and red, and the ground truth in dashed grey.

TABLE V
RMSE OF THE ATE FOR MSC-VO AND OTHER STATE-OF-THE-ART APPROACHES (IN METERS)

	Without Global Optimization nor LCD						With Global Optimization and/or LCD			
Sequence	MSC-VO	OPVO [9]	LPVO [10]	MWO [8]	SReg [14]	ManhSLAM [13]	ORB-SLAM2 [2]	PS-SLAM [6]	L-SLAM [11]	InfiniTAM [26]
lr-kt0	**0.006**	×	0.015	×	**0.006**	**0.007**	0.025	0.016	0.012	×
lr-kt1	0.010	0.04	0.039	0.32	0.015	0.011	**0.008**	0.018	0.027	**0.006**
lr-kt2	**0.009**	0.06	0.034	0.11	0.020	0.015	0.023	0.017	0.053	**0.013**
lr-kt3	0.038	0.10	0.102	0.40	**0.012**	**0.011**	0.021	0.025	0.143	×
of-kt0	0.028	0.06	0.061	0.31	0.041	**0.025**	0.037	0.032	**0.020**	0.042
of-kt1	0.017	0.05	0.052	1.10	0.020	**0.013**	0.029	0.019	**0.015**	0.025
of-kt2	**0.014**	×	0.039	×	**0.011**	0.015	0.039	0.026	0.026	×
of-kt3	**0.010**	0.04	0.030	1.38	0.014	0.013	0.065	0.012	**0.011**	**0.010**
fr1-xyz	**0.010**	n.a.	n.a.	n.a.	×	**0.010**	**0.010**	**0.010**	n.a.	n.a.
fr1-desk	**0.019**	n.a.	n.a.	n.a.	×	0.027	**0.022**	0.026	n.a.	n.a.
fr2-xyz	**0.005**	n.a.	n.a.	n.a.	×	**0.008**	0.009	0.009	n.a.	n.a.
fr2-desk	**0.023**	n.a.	n.a.	n.a.	×	0.037	0.040	**0.025**	n.a.	n.a.
snot-far	0.077	0.13	0.075	0.47	**0.022**	0.040	×	**0.020**	0.141	0.037
snot-near	×	0.16	0.080	0.95	0.025	0.023	×	**0.013**	0.066	**0.022**
large-cabinet	0.120	0.51	0.279	0.83	**0.071**	0.083	0.124	**0.079**	0.140	0.512
fr3-longoffice	**0.022**	×	0.19	×	n.a.	0.049	**0.028**	n.a.	n.a.	n.a.

× and n.a. respectively stand for *tracking failure* and *not available* value. The best result for each sequence is shown in bold orange and the second best in bold blue.

side of the table is for solutions that benefit from those stages. As can be observed from the ICL-NUIM dataset, the proposed method, which only uses point and lines, achieves competitive results in contrast to other methods that rely on points, lines and planes, such as [13], [14]. Conversely, from the *fr1* and *fr2* sequences, we observe that methods relying on planes are not able to correctly estimate the MA. This is due to the fact that these methods fail to find or track orthogonal planes along the sequence. Contrarily, our approach can estimate the MA on these scenarios, except for the *fr2-desk sequence*, although the structural constraints are fully applicable in this sequence, allowing our approach to remain operational and outperform the rest of solutions. MSC-VO produces a tracking failure in the *snot-near* sequence. We do not observe this behaviour in works relying on planar features, due to the continuous presence of orthogonal planes in the sequence. It is noteworthy that MSC-VO compares favourably with more sophisticated solutions (right side of the table) even without global map optimization or LCD stages.

Finally, Table IV compares the performance of MSC-VO with other solutions in long sequences. On the one hand, we have observed that in the *Corridor-A* sequence most part of the error is due to tracking failures: in these cases, the proposed local map optimization can not fix the problem since no lines are detected in the axis where the errors take place. Despite this is not a common situation, we consider that planes can help to avoid this behaviour due to the continuous detection of the floor. However, it is important to remark that this dataset contains noisy depth data, which highly affects plane detection, and, therefore, the MA assumption does not hold for all frames. As an example, [13] tracks the pose using the MA assumption in, respectively, 15.1% and 12.5% of the frames of *Corridor-A* and *Entry-Hall*. However, MSC-VO uses the MA assumption in all the frames that at least contain one single line associated to an MA, which represents 100% of the frames in both sequences.

V. CONCLUSION AND FUTURE WORK

In this work, we have described MSC-VO, a VO that improves camera pose estimation accuracy in human-made environments. This is achieved by a combined point and line VO approach that leverages the structural regularities of the environment as well as the satisfaction of the MW assumption. On the one side, the structural constraints are used to improve line depth extraction and MA estimation. On the other side, these structural constraints are combined with point and line reprojection errors together with the MW assumption for local map optimization. All these contributions have been shown to increase the accuracy of 3D map lines position estimation and the computed trajectory for MSC-VO. Furthermore, contrary to other state-of-the-art works that use the MW in the tracking stage, our pipeline is designed to deal with the absence of the MA, allowing us to work in a wider range of environments.

Regarding future work, we plan to integrate MSC-VO with an incremental loop closure detection strategy. We are also intent to make use of the structural constraints and the MA alignment for global map optimization.

REFERENCES

[1] N. Yang, R. Wang, and D. Cremers, "Feature-based or direct: An evaluation of monocular visual odometry," 2017, *arXiv:1705.04300*.

[2] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras," *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1255–1262, Oct. 2017.

[3] J. P. Company-Corcoles, E. Garcia-Fidalgo, and A. Ortiz, "LiPo-LCD: Combining lines and points for appearance-based loop closure detection," in *Proc. Brit. Mach. Vis. Conf.*, 2020, pp. 1–13.

[4] A. Pumarola, A. Vakhitov, A. Agudo, A. Sanfeliu, and F. Moreno-Noguer, "PL-SLAM: Real-time monocular visual SLAM with points and lines," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2017, pp. 4503–4508.

[5] R. Gomez-Ojeda, D. Zuñiga-Noël, F.-A. Moreno, D. Scaramuzza, and J. Gonzalez-Jimenez, "PL-SLAM: A stereo SLAM system through the combination of points and line segments," 2017, *arXiv:1705.09479*.

[6] X. Zhang, W. Wang, X. Qi, Z. Liao, and R. Wei, "Point-plane SLAM using supposed planes for indoor environments," *Sensors*, vol. 19, no. 17, 2019, Art. no. 3795.

[7] J. Coughlan and A. Yuille, "Manhattan world: Compass direction from a single image by bayesian inference," in *Proc. Int. Conf. Comput. Vis.*, 1999, pp. 941–947.

[8] Y. Zhou, L. Kneip, C. Rodriguez, and H. Li, "Divide and conquer: Efficient density-based tracking of 3D sensors in Manhattan worlds," in *Proc. Asian Conf. Comput. Vis.*, 2016, pp. 3–19.

[9] P. Kim, B. Coltin, and H. J. Kim, "Visual odometry with drift-free rotation estimation using indoor scene regularities," in *Proc. Brit. Mach. Vis. Conf.*, 2017, pp. 62.1–62.12.

[10] P. Kim, B. Coltin, and H. J. Kim, "Low-drift visual odometry in structured environments by decoupling rotational and translational motion," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2018, pp. 7247–7253.

[11] P. Kim, B. Coltin, and H. J. Kim, "Linear RGB-D SLAM for planar environments," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 333–348.

[12] L. Wang and Z. Wu, "RGB-D SLAM with manhattan frame estimation using orientation relevance," *Sensors*, vol. 19, no. 5, 2019, Art. no. 1050.

[13] R. Yunus, Y. Li, and F. Tombari, "ManhattanSLAM: Robust planar tracking and mapping leveraging mixture of manhattan frames," 2021, *arXiv:2103.15068*.

[14] Y. Li, R. Yunus, N. Brasch, N. Navab, and F. Tombari, "RGB-D SLAM with structural regularities," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 11581–11587.

[15] H. Li, Y. Xing, J. Zhao, J.-C. Bazin, Z. Liu, and Y.-H. Liu, "Leveraging structural regularity of atlanta world for monocular SLAM," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2019, pp. 2412–2418.

[16] H. Li, J. Yao, J. Bazin, X. Lu, Y. Xing, and K. Liu, "A monocular SLAM system leveraging structural regularity in Manhattan world," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2018, pp. 2518–2525.

[17] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Proc. Int. Conf. Comput. Vis.*, 2011, pp. 2564–2571.

[18] R. Grompone von Gioi, J. Jakubowicz, J. Morel, and G. Randall, "LSD: A fast line segment detector with a false detection control," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 4, pp. 722–732, Apr. 2010.

[19] L. Zhang and R. Koch, "An efficient and robust line segment matching approach based on LBD descriptor and pairwise geometric consistency," *J. Vis. Commun. Image Representation*, vol. 24, no. 7, pp. 794–805, 2013.

[20] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "G^2o: A general framework for graph optimization," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2011, pp. 3607–3613.

[21] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge, MA, USA: Cambridge Univ. Press, 2003, doi: 10.1017/CBO9780511811685.

[22] D. Holz, S. Holzer, R. B. Rusu, and S. Behnke, "Real-time plane segmentation using RGB-D cameras," in *RoboCup 2011: Robot Soccer World Cup*, T. Röfer, N. M. Mayer, J. Savage, and U. Saranlı, Eds. Berlin, Germany: Springer, 2012, pp. 306–317.

[23] A. Handa, T. Whelan, J. McDonald, and A. Davison, "A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2014, pp. 1524–1531.

[24] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *Proc. IEEE Int. Conf. Intell. Robots Syst.*, 2012, pp. 573–580.

[25] Y. Lu and D. Song, "Robust RGB-D odometry using point and line features," in *Proc. Int. Conf. Comput. Vis.*, 2015, pp. 3934–3942.

[26] V. A. Prisacariu *et al.*, "InfiniTAM v3: A framework for large-scale 3D reconstruction with loop closure," 2017, *arXiv:1708.00783*.

VINSEval: Evaluation Framework for Unified Testing of Consistency and Robustness of Visual-Inertial Navigation System Algorithms

Alessandro Fornasier[1], Martin Scheiber[1], Alexander Hardt-Stremayr[1], Roland Jung[1] and Stephan Weiss[1]

***Abstract*— The research community presented significant advances in many different Visual-Inertial Navigation System (VINS) algorithms to localize mobile robots or hand-held devices in a 3D environment. While authors of the algorithms often do compare to, at that time, existing competing approaches, their comparison methods, rigor, depth, and repeatability at later points in time have a large spread. Further, with existing simulators and photo-realistic frameworks, the user is not able to easily test the sensitivity of the algorithm under examination with respect to specific environmental conditions and sensor specifications. Rather, tests often include unwillingly many polluting effects falsifying the analysis and interpretations. In addition, edge cases and corresponding failure modes often remain undiscovered due to the limited breadth of the test sequences. Our unified evaluation framework allows, in a fully automated fashion, a reproducible analysis of different VINS methods with respect to specific environmental and sensor parameters. The analyses per parameter are done over a multitude of test sets to obtain both statistically valid results and an average over other, potentially polluting effects with respect to the one parameter under test to mitigate biased interpretations. The automated performance results per method over all tested parameters are then summarized in unified *radar charts* for a fair comparison across authors and institutions.**

SOFTWARE & VIDEO

The open-sourced VINSEval framework is made available via https://github.com/aau-cns/vins_eval. A demonstration video of VINSEval is made available on https://youtu.be/KuA3nibxWok.

I. INTRODUCTION

Data-driven algorithms for autonomous robotics gained significant attention over the last years, enabling a paradigm shift in state estimation for mobile robotic applications. This trend allowed the robotic research community to design and develop *Visual-Inertial SLAM (VI-SLAM)*, *Visual-Inertial Odometry (VIO)* algorithms, or, in general, *Visual-Inertial Navigation System (VINS)* algorithms able to reach high performance in terms of accuracy and efficiency. To do so, simulation and synthetic data have been one of the fundamental tools for engineers and researchers during the design and development of such algorithms. They allow fast prototyping, safe, and inexpensive testing without dealing with real-world experiments and hardware issues in the early development stages. Further, simulations provide high repeatability of data and the precise control of various parameters. Despite the progress made to let state-of-the-art estimation algorithms reach high performance in terms of accuracy with respect to commonly used error metrics (i.e., Absolute Trajectory Error (ATE), and Relative Trajectory Error (RTE)), current VINS solutions still lack in robustness and consistency. In addition, as more and more such methods are presented by the community, the lack of a unified comparison and benchmarking tool starts to become an important issue.

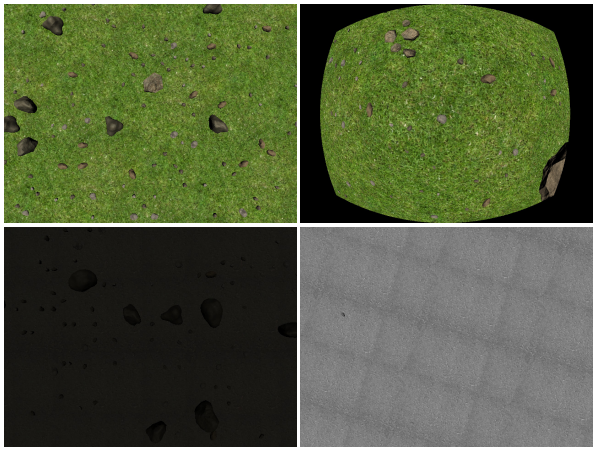


Fig. 1. Views of different rendered scenes. Top row, left to right: a scene with a grass ground texture and stones providing informative visual features, with no camera distortion and with fisheye lens distortion. Bottom row, left to right: a scene with a cement self-similar ground texture and no informative visual features, with low illumination, and with optimal illumination.

Motivated by the shortcomings in robustness and consistency in VINS methods, and in particular the upcoming era of research dubbed *robust-perception-age* [1], in this paper, we present *VINSEval*: a fully automated photo-realistic visual and inertial data generation, simulation, and estimator evaluation framework for fast VINS development, improvement, and unified comparison. VINSEval has two core capabilities: (i) For researchers to speed up the prototyping and development of consistent and robust VINS algorithms through the capability of generating setups and data with very specific parameters and parameter changes, and (ii) for both researchers and end-user engineers to evaluate and compare the performance of VINS algorithms in terms of consistency and robustness in a unified, fully automated fashion over a large set of parameter sweeps. VINSEval is not only general in that sense, but it also allows the generation of very specific edge-case scenarios where VINS can be tested in. Each scenario has unique visual characteristics and requirements, and we believe providing a framework for understanding the constraints therein is critical when assessing performance and guiding subsequent development. State-of-the-art VINS benchmarking approaches tend to ignore differences between individual scenarios leading to solutions that only partially address end-user needs. They rather focus on system-level

[1]Control of Networked Systems, University of Klagenfurt, Austria {firstname.lastname}@ieee.org

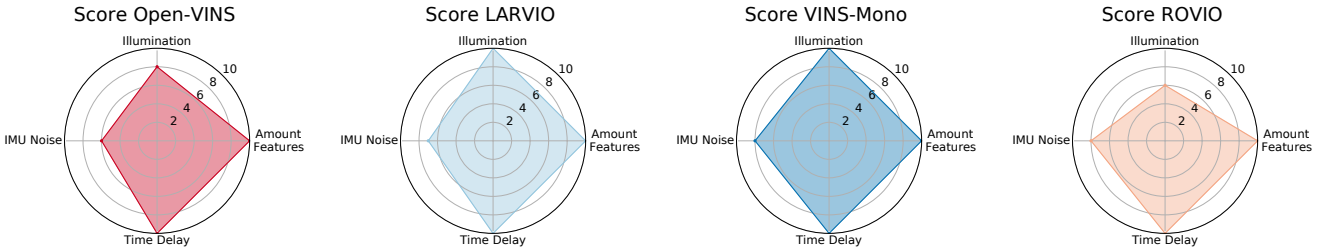


Fig. 2. Robustness overall score and Breaking Point (BP) of the VINS algorithms under examination with increasing difficulty levels for each of the considered environmental and/or sensor parameters. The BP per parameter is visually defined as the level next to the the corner of the polygon.

performance and accuracy but overlook that none of these error metrics, when uncontextualized, indicate how well a VINS algorithm could perform on a given specific scenario.

The multiple key contributions of the presented work are: ***Unified statistical evaluation framework***: To the best of our knowledge, VINSEval is the first work that provides a framework to evaluate with statistical relevance the consistency and robustness of VINS algorithms in a fully automated fashion over a multitude of parameters and parameter ranges. ***Sample evaluation***: We demonstrate how VINSEval can statistically compare the consistency and robustness of four state-of-the-art algorithms when applying parameter sweeps over (i) amount of features seen, (ii) illumination conditions, (iii) IMU noise values, and (iv) sensor time delay. The results are automatically summarized in radar-charts in Fig. 2 for quick information access with minimal user effort. ***Extensibility***: VINSEval is an easily extendable framework. This is true for the photo-realistic scene, different parameter sweeps, and different evaluation parameters that can further be included. All such extensions are directly included in the fully automated evaluation process enabling VINSEval as a useful tool for VINS evaluation in various different scenarios. ***Flexibility and modularity***: All the modules of VINSEval are modular and flexible. Indeed the data provided as input to the core of VINSEval can be either synthetically generated or recorded from a real platform. The rendering module then allows automated changes of the rendered scene and flexibility to manipulate rendering parameters, UAV parameters, and sensors noise specifications online.

II. Related Work

With particular regard to UAVs, Hector Quadrotor [2] and RotorS [3] are Gazebo simulators that allow the user to simulate different types of multi-rotor UAVs with specific sensors such as IMU, LIDAR and camera. These environments do not provide photo-realistically rendered camera images – an issue addressed by AirSim [4]. This work proposes a software-in-the-loop simulation with popular flight controllers such as PX4 and ArduPilot and hardware-in-loop for physically and visually realistic simulations. Recently [5] and [6] published their work FlightGoggles and Flightmare, respectively, which are both ROS-based open-source photo-realistic simulation framework for MAVs. They mainly differ from AirSim by having fewer rigid structures and an integrated physics engine for dynamics simulation. InteriorNet [7] proposes an end-to-end pipeline for an RGB-D-inertial benchmark in large-scale interior scene understanding and mapping. The trajectories, the scenes, and rendering parameters have a high level of customizability. However, the simulator lacks flexibility as it is limited to a fixed set of indoor scenes and CAD models of indoor furniture. The authors in [8], [9], [10] presented SlamBench (currently at version 3) which is a dataset- and sensor-agnostic framework for qualitative, quantitative, and easily reproducible evaluation for accuracy and computation time of SLAM systems with plug-and-play algorithm support. SlamBench incorporates a wide range of error metrics, datasets, and evaluation tools for different SLAM algorithms. However, its flexibility is limited since it does not provide a way to generate individual data for a specific scenario. Its focus is on the evaluation of computational complexity and estimation accuracy, not on robustness and consistency. Regarding robustness, the authors in [11] proposed a characterization of state-of-the-art SLAM benchmarks and methods by comparisons of the performance of different SLAM algorithms. They use publicly available datasets, at both real-time speed and slo-mo playback, clustering the results into four classes denoted *fail*, *low*, *medium*, and *high*. Furthermore, the authors in [12] proposed firstly new datasets for wheeled robots, including different locations, day-night shifts, moving objects, and poor illumination, and second a new metric for robustness evaluation based on a judgment of ”correctness” through an empirically chosen threshold on the ATE. Like the previously cited SlamBench, the last works' main weakness is the limited flexibility, controllability, and scalability of the data without automated procedures, limiting the possible usage for statistically relevant large scale tests on robustness and consistency.

III. Framework Architecture

The core of the VINSEval framework architecture, shown in Fig. 3, is organized as a *Robot Operating System (ROS)* package and is composed by two fundamental software modules: the *Data Generation Module* (cyan block) and the *Estimators Evaluation Module* (red block). The former takes as an input a given generated trajectory file containing timestamped ground-truth poses, velocity, and acceleration measurements that could be either noisy (e.g., recorded from a real platform equipped with an IMU and a motion capture system) or noise-free (e.g., synthetically generated). The input data is then processed and used to produce ROS bagfiles of sensor data for the other module, whose primary

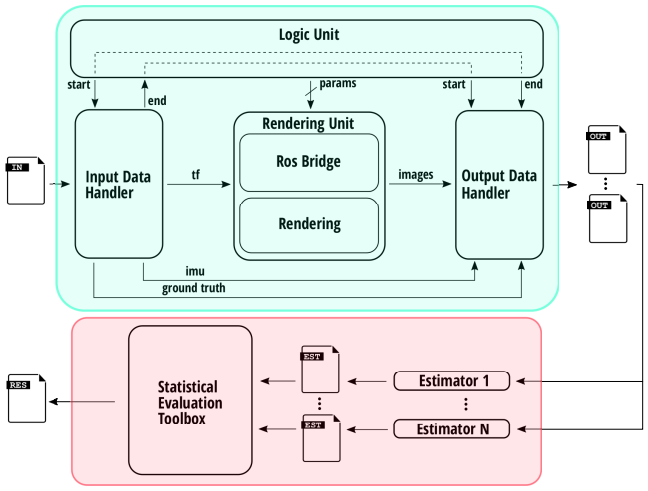


Fig. 3. Framework architecture overview: the full pipeline is composed by two main blocks, in cyan is the data generation block that allows to generate unlimited amount of data containing ground-truth, noisy IMU, and photo-realistic images given a set of trajectories while in red is the estimator evaluation block, that allows to perform consistency and robustness evaluation of estimators with the generated data.

objective is to run different VINS algorithms and provide a statistical evaluation of robustness and consistency.

A. Data Generation Module

The data generation module is divided into four submodules. The *logic unit* directs all the other submodules by providing them with control signals and parameters. Upon the logic unit's start signal, the *input data handler* takes the provided trajectory file as input and parses it. If the data contain noise-free IMU measurements, the input data handler adds noise and biases, for which statistics are provided as parameters following the inertial sensor model described in [13]. This data is then published as ROS messages to be used by subsequent submodules. The *rendering unit* is derived from the photo-realistic simulation framework FlightGoggles [5] with our extended capabilities such as the introduction of a fisheye distortion model for the camera lens, a variable time delay on the image header timestamp, and rendering parameters that allow changes of the visual attributes (e.g., illumination, objects distribution, etc. . . See Sec. V) in the scene. The rendering of photo-realistic camera images can be set to either asynchronous, such that the camera images are rendered in real-time at a frame rate dependent on the machine's performance, or synchronous, such that the camera images are synchronized with the given trajectory poses. Note that this modularization of the input data handler and the rendering unit allows for proprioceptive data from real systems (i.e., robot motion) to be used for a virtual rendering of precisely controlled exteroceptive measurements (i.e., camera image). The last submodule of the data generation module is the *output data handler*, which manages to start and stop recording the data into bagfiles when triggered by control signals from the logic unit. It automatically applies a realignment of the bagfiles substituting the wall time with the header time.

B. Estimators Evaluation Module

The statistical evaluation toolbox module of VINSEval consists of different submodules such as (i) the conversion of estimated trajectories to CSV files, (ii) finding associations between two trajectories based on their timestamps, (iii) spatial alignment tools to align the estimated trajectory with the true trajectory supporting different alignment types as in [14], (iv) absolute trajectory error evaluation based on associated and aligned trajectories, and finally (v) the computation of the Normalized Estimation Error Squared (NEES) and Average NEES (ANEES). On top of these, the estimator evaluation module supports an automated evaluation of different scenarios and multiple experiments and automated report generation.

IV. Error Metrics and Estimators Evaluation

Before we detail our approach on defining the different parameters to test for and their sweeping range, we first define what we understand under consistency, credibility, and robustness and explain the associated error metrics.

A. Consistency and Credibility

Estimators such as least-squares and Kalman filters provide assessments in terms of their error covariance matrix or Mean Squared Error Matrix (MSE) and the estimated state. The estimation error $\epsilon_j = \hat{\mathbf{x}}_j - \mathbf{x}_j \in \mathbb{R}^k$ is the difference between estimated and actual true value. The NEES is a commonly used metric that normalizes the scalar magnitudes of the estimation error ϵ_j based on the error covariance $\mathbf{P}_j$

$$\|\epsilon_j\|^2_{\mathbf{P}_j^{-1}} = \epsilon_j^T \mathbf{P}_j^{-1} \epsilon_j \in [0, \infty]. \tag{1}$$

The NEES is assumed to be χ^2 distributed with k degrees of freedom and a mean of k. Therefore, a chi-square significance test can be performed to judge if an estimator violates a certain credibility threshold [15]. A too low or too high NEES, depending on k, indicates under- and overconfidence, respectively. As the ground-truth is needed, the NEES is typically computed offline using M Monte Carlo simulations and then averaged over the M runs and normalized with respect to the state dimension k resulting in the ANEES:

$$\text{ANEES} = \frac{1}{kM} \sum_{m=1}^{M} \|\epsilon_{j_m}\|^2_{\mathbf{P}_{j_m}^{-1}}. \tag{2}$$

For our evaluation, we propose to compute the $\overline{\text{NEES}}$, the mean of the NEES, over the time span of each trajectory with D time steps, and then the $\overline{\text{ANEES}}$ as follows:

$$\overline{\text{ANEES}} = \frac{1}{kM} \sum_{m=1}^{M} \frac{1}{D} \sum_{j=0}^{D} \|\epsilon_{j_m}\|^2_{\mathbf{P}_{j_m}^{-1}}. \tag{3}$$

Computing the $\overline{\text{NEES}}$ reduces the significance of sporadic spikes that occur typically at the initialization phase until the filter starts to converge. Based on the $\overline{\text{ANEES}}$ and a credibility threshold, e.g., a probability interval of 99 %, we classify estimators to be credible or not. If the credibility threshold is reasonably high and violated, we assume the filter to be inconsistent.

B. Robustness

"*Robustness is the ability to withstand or overcome adverse conditions*." – [from online dictionary]. In the context of VI-SLAM and VIO, we can say that a robust estimator is resistant to deviations from the assumptions of optimal conditions. Hence if the assumptions are only approximately met, the estimator still has a reasonable performance. Contrary to estimator credibility, finding a metric to judge the robustness of a given estimator is particularly difficult. Here, we adopt a simple metric based on the Root Mean Square Error (RMSE) to define the so-called *Breaking Point (BP)*. Consider a given visual attribute (e.g., the illumination in the scene) that is changed L times from the optimal condition with an increasing amount of changes towards a bad condition. The BP is the point along the scale of change at which a given estimator breaks. Thus, for each attribute value change, the average RMSE of the ATE is computed. The RMSE is then compared to an empirical threshold to distinguish whether the estimator has broken or not.

V. Environment and parameter setup

As mentioned in Sec. III the rendering module inherits all the capabilities of FlightGoggles [5] and thus the various other types of exteroceptive sensors (other than camera and IMU) such as RGB-D cameras, IR beacon sensors and time-of-flight range sensors for which intrinsic, extrinsic parameters and noise specification can be easily adjusted. Moreover we added options for variable sensor time delays and online parameter adaptations in the scene and the system. We also extended the default pinhole camera model with a realistic fisheye lens based on the *atan model* [16]. To improve runtime efficiency the undistortion of each output pixel is calculated at startup and saved in a lookup table, given the diagonal distortion parameter s as described in [16]. However, the undistorted pixel values are most likely non integer values. Therefore the average color value is calculated at runtime with Eq. (4). $C\big(\mathbf{p}_d(j,i)\big)$ is the color value of the distorted integer pixel $\mathbf{p}_d(j,i)$, $\mathbf{p}_u(j,i)$ its corresponding undistorted non-integer pixel value. $C\big(\underline{\mathbf{p}}_u\big)$ is the lower-left, $C\big(\overline{\mathbf{p}}_u\big)$ the upper-left, $C\big(\overline{\mathbf{p}}_{u+1}\big)$ the upper-right, and $C\big(\underline{\mathbf{p}}_{u+1}\big)$ the lower-right surrounding undistorted pixel color values. δx and δy are the differences between the lower-left (integer) undistorted pixel $\underline{\mathbf{p}_u}$ and the calculated undistorted pixel $\mathbf{p}_u(j,i)$, in x- and y-axis respectively.

$$\begin{aligned} C(\mathbf{p}_d(j,i)) = \delta x \cdot \Big(\delta y \cdot C\big(\underline{\mathbf{p}}_u\big) + \big(1-\delta y\big) \cdot C\big(\overline{\mathbf{p}}_u\big)\Big) \\ +\big(1-\delta x\big) \cdot \Big(\delta y \cdot C\big(\underline{\mathbf{p}}_{u+1}\big) + \big(1-\delta y\big) \cdot C\big(\overline{\mathbf{p}}_{u+1}\big)\Big) \end{aligned} \tag{4}$$

Further, this framework provides an RGBA color to grayscale conversion based on the methods described in [17]. This work showed that the method used to convert colored images can greatly impact the result. Although all methods are implemented in VINSEval, we opted to use the *Luminance* method in the presented sample evaluation, as it maps the human eye brightness perception most closely [18].

A. Estimator Parameter Setup

Although highly customizable, we suggest here a specific set of parameters and environment settings to use in the proposed VINSEval framework to generate data and evaluate different open source state-of-the-art VINS algorithms. We generate UAVs feasible trajectories and noise-free IMU measurements at 200Hz. Trajectories are generated with a minimum snap trajectory generation approach, as described by [19]. The considered VINS algorithms are: **LARVIO** [20] and **OpenVins** [21] which are both filter-based VIO algorithms leveraging the Multi-State Constraint Kalman Filter (MSCKF) sliding window formulation. Both filters allow online camera-imu calibration, zero velocity update, different landmark parametrizations and first estimate jacobian formulation aiming to improve the filter consistency. **ROVIO** [22], [23], a fully robocentric and direct filter based VIO algorithm which makes use of the pixel intensity errors of image patches, aiming to achieve high level of robustness. **Vins-Mono** [24], an optimization-based sliding window formulation VIO algorithm aiming to provide high accuracy. All the algorithms used in our experiments have been tuned to get the best results in a randomly selected subset of the whole data used. The extrinsic and intrinsic parameters of the camera, as well as the distortion coefficient, have been set to the correct value provided by the rendering unit of VINSEval, and the online calibration of such parameters was turned off. When there is a time delay between the camera and the IMU we turn on the online estimation of such time delay providing the correct value as an initial guess, on the estimators that allow that. Moreover, we provide all estimators with the correct IMU noise statistics. Regarding the feature tracker, we similarly tuned every feature-based algorithm to achieve best results for all involved algorithms.

B. Experiments Setup

In our experiments we have considered mainly four attributes which are particularly relevant in real-world situations: (i) Changes in amount of informative visual features (ii) Changes in illumination (iii) Changes in time delay between the camera and the IMU (iv) Changes of the IMU noise and noise statistics. For each of the considered attributes a, we have defined $L = 10$ different "difficulty" levels for which increasing levels produce a more complicated scenario for an estimator. For every single level $l \in [1, L]$ we run $M = 20$ different simulations where we dynamically change all the other environmental conditions and sensor specifications, including the attributes that are not evaluated and other parameters such as object placement distribution or UAV trajectory while keeping all of them in the range of what we defined to be "optimal". These parameter swaps provide randomness to the evaluation and average over-polluting effects, leaving only the change effects in the single attribute under consideration. Thus, sweeping over one attribute generates 200 test runs per VINS algorithm.

For a given attribute a, the following subsections describe how the level l has been mapped to a change of the considered attribute, and how "optimal" values are defined.

C. Changes in amount of informative visual features

For this attribute, we evaluate the former cited algorithms' performance when the amount of informative features changes. We introduce an *information-density* parameter D, determining the overall amount and placement of recognizable features within the scene compared to either self-similar or featureless ground. A value of 1 corresponds to the approximation of the ideal, informative-rich scene, while 0 will not place any objects. Values in between will decrease the placement probability of objects linearly, with a multiplier based on position-dependent Perlin noise. A value of 0.5 would place half as many objects compared to the ideal scene, with higher object densities around Perlin-based clusters. The attribute level l influences the generated scene twofold: a linear multiplier of the object placement density between the maximum at the easiest and $5\,\%$ at the most challenging level as well as decreasing clustering with growing difficulty.

D. Changes in illumination

For this attribute, the illumination of the virtual scene changes over the different values of the level l, reducing the illumination intensity I for a fixed window of time during the UAV trajectory, from its optimal value to lower values as the level l increases. The effect of decreasing the illumination in a real-world scenario using a camera set with auto-exposure triggers a chain reaction, which increases the camera's exposure time with the consequence of increasing the amount of motion blur that the images will have. However, in these experiments, we are simulating a camera with fixed exposure time and without any simulated motion blur being applied. Thus we aim to evaluate the estimators' performance against abrupt changes of the illumination intensity on the scene. We consider the optimal value to be $I = 1$, corresponding to the attribute level $l = 1$, which emulates a sunlight condition on a clear day. The mapping between the attribute level l and the illumination intensity I has been defined through a second-order function, as follows:

$$I = \alpha\,(l-1)^2 + \beta\,(l-1) + 1 \tag{5}$$

With empirical values $\alpha = 0.0137$ and $\beta = -0.23$ to achieve a fairly dark environment at the most challenging level.

E. Changes in time delay between the camera and the IMU

Let us consider the scenario for which camera images are captured synchronously with the IMU measurements. For time delay between the camera and IMU, we consider the delayed image's timestamp when the image is available to the estimator (e.g., USB delay). Thus, in this scenario, we aim to evaluate how estimators manage such a delay. For a given attribute level l, the images header timestamp is defined:

$$t_{cam} = t_{imu} + k\big(l\,(l-1)\,\big) \tag{6}$$

With $k = \frac{5}{3000}$ heuristically chosen, leading to a maximum time delay of $150\,\mathrm{ms}$, for $l = 10$ and to no delay for $l = 1$.

F. Changes of the IMU noise and noise statistics

The last attributes we considered within this evaluation are the accelerometer and gyroscope noise densities and random walk. The value changes range from a simulated high grade IMU down to a very low-performing MEMS IMU. For a given attribute level l the IMU noise statistics are thus changed according to:

$$\sigma_* = \varphi\left(10^{\psi(l-1)}\right) l\sigma_{*_{opt}} \tag{7}$$

Where σ_* indicates the continuous-time IMU noise densities and random walks scaled concerning the optimum value $\sigma_{*_{opt}}$ by a scale factor. The value $\varphi = 2$ and $\psi = \frac{2}{9}$ has been chosen empirically leading to the min. and max. values reported in Tab. I. In this case, the optimal (or better, realistic) value has been chosen to correspond to an attribute level $l = 5$, in order to have IMU noise statistics in the same order of magnitude of the majority of the MEMS IMU used nowadays on UAVs, and to avoid cases of having an accurate estimation even if all tracked features are lost.

TABLE I
MINIMUM AND MAXIMUM VALUES OF THE ACCELEROMETER AND GYROSCOPE NOISE DENSITIES AND RANDOM WALKS CONSIDERED

	$\sigma_{*_{min}}$	$\sigma_{*_{max}}$	
σ_a	$1.0e^{-4}$	$2.0e^{-1}$	$m/s^2\sqrt{Hz}$
σ_g	$1.0e^{-5}$	$2.0e^{-2}$	$rad/s\sqrt{Hz}$
σ_{ba}	$1.0e^{-5}$	$2.0e^{-2}$	$m/s^3\sqrt{Hz}$
σ_{bg}	$1.0e^{-6}$	$2.0e^{-3}$	$rad/s^2\sqrt{Hz}$

VI. SAMPLE EVALUATION PROCESS

For each of the previously described attributes we have generated $L \times M = 200$ different bagfiles of raw data, each one containing $40\,\mathrm{Hz}$ rendered VGA camera images and $200\,\mathrm{Hz}$ IMU data for a UAV trajectory of $100\,\mathrm{s}$, resulting in a total of 800 different bagfiles and about $1\,\mathrm{TB}$ of data. With the synchronous rendering option, the data can be generated faster than real-time, meaning more than $22\,\mathrm{h}$ of data has been generated in $8\,\mathrm{h}$[1]. The generated raw data is then used in the estimator evaluation module (cf. Sec. III-B), which feeds the data to the mentioned VINS estimators and starts a, in our case 3-day long, batch run. Each estimated trajectory corresponding to a specific attribute a, level l, run m, and estimator e is first aligned with the corresponding ground-truth trajectory along the unobservable dimensions (i.e. position and yaw). Then the ARMSE and the ANEES over the time span of the trajectory are computed for position and orientation. This automatically generates a report at very high detail level. As last step of the estimator evaluation, the $10\,\%$ of the data, corresponding to the worst runs in terms of normalized sum of the single error metrics, accounting for processor-load hick-ups in the OS, have been removed. Then a final summary report, shown in Fig. 2, 4, 5, is produced by computing, for each attribute a and level l, the ARMSE and ANEES over the M runs for each estimator.

[1]We run VINSEval on a high-performance simulation PC, however, data generation in synchronous mode results faster than real-time even in a mid-performance laptop.

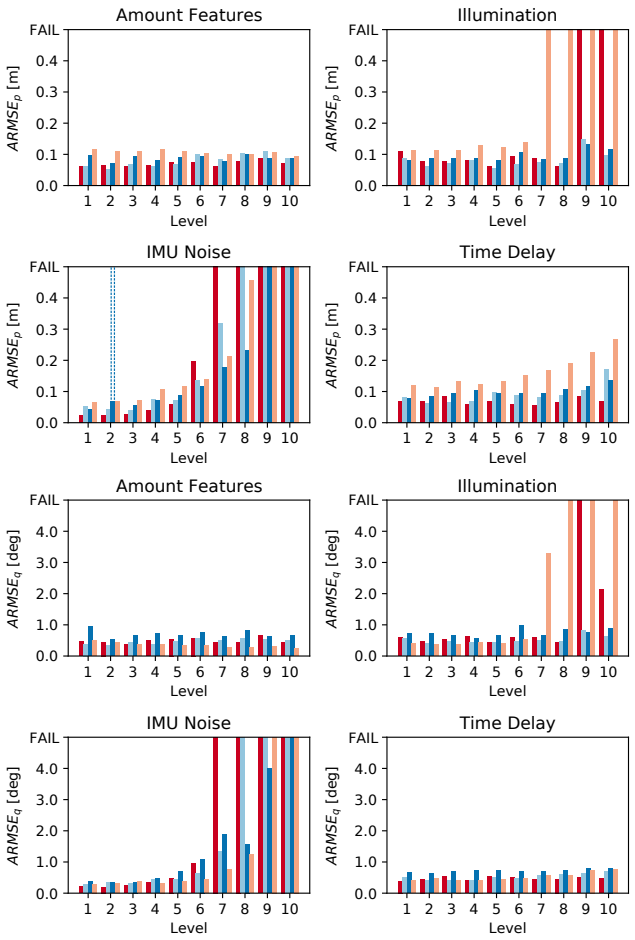


Fig. 4. Performance Comparison: Position and orientation ARMSE. Open-Vins [21] in red, LARVIO [20] in cyan, Vins-Mono [24] in blue and ROVIO [22], [23] in light red. We can notice in dashed blue that Vins-Mono, when tested with increasing IMU noise, is failing at attribute level $l = 1$ but not on further levels. Our investigation lead to the conclusion that very low IMU noise values cause numerical issue and then lead to a failure. We tackled the problem by tuning Vins-Mono with a falsely increased IMU noise. A strength of our framework is precisely to reveal such edge cases.

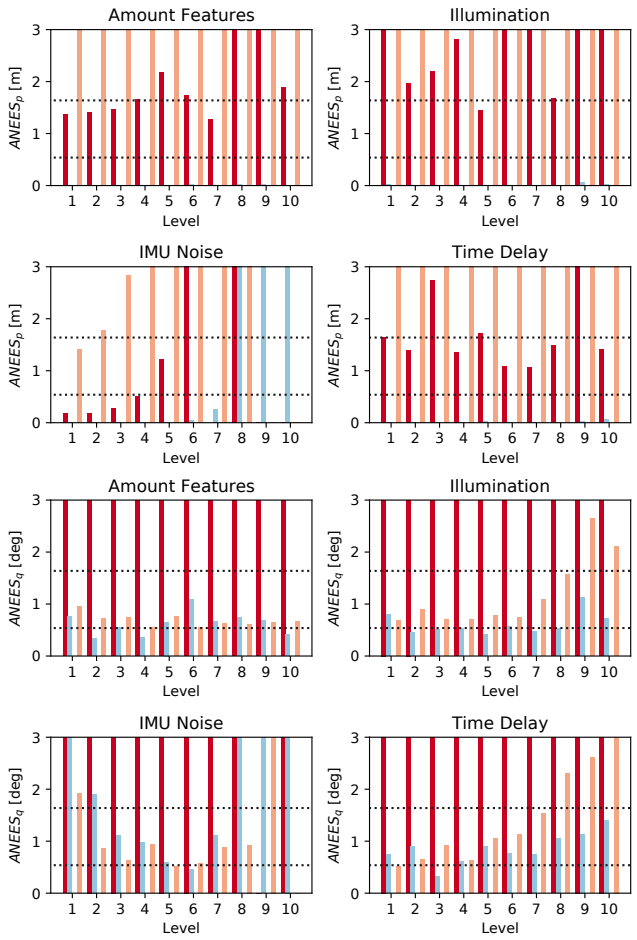


Fig. 5. Performance Comparison: Position and orientation $\overline{\text{ANEES}}$. Open-Vins [21] in red, LARVIO [20] in cyan and ROVIO [22], [23] in light red. Dashed are the confidence bounds. Despite our best tuning efforts, we were not able to reproduce the $\overline{\text{ANEES}}$ for Open-Vins reported by the authors; too little details on their method is given in [21]

As described in Sec. IV-B, we made use of the position and orientation ARMSE for which we defined a threshold to judge the failure of a an estimator. In particular, for a given attribute a, a level l and an estimator e we define the following binary score: ${}^{a}_{l}\mathcal{F}_e = True\ if\ ARMSE_* > T_{H_*};\ False\ if\ ARMSE_* < T_{H_*}$. Where the symbols $*$ stand either for position or orientation and T_{H_*} is the chosen threshold. With about $70\,\mathrm{m}$ trajectories, T_{H_*} are heuristically set to be $0.5\,\mathrm{m}$ for position and $5°$ for orientation. The first occurrence of ${}^{a}_{l}\mathcal{F}_e = True$ for increasing values of l, will define the BP per attribute a, per estimator e.

VII. Conclusion

In this paper we presented VINSEval, a unified framework for statistical relevant evaluation of consistency and robustness of VINS algorithms with fully automated scoreboard generation over a set of selectable attributes. We showed the ability of effective evaluation given by the flexibility on parameter selection, the mitigation of polluting effects through multiple runs with randomization in dimensions not under test, and the inherent detection of edge-cases through the wide test span in an automated fashion. We will open-source VINSEval making it a usable and extendable tool for the community towards unified estimator evaluation.

As a sample VINSEval demonstration, we let a Breaking Point score, in Fig. 2, to be generated to show how robust and consistent current state-of-the-art algortihms are. All tested algorithms generally exhibit low ARMSE when challenged with increased imu-camera time delay, decreasing illumination and amount of informative features showing the ability to compensate for dark scenes and correctly detect and track self-similar features on the background. However, all the algorithms show high sensitivity to IMU noise statistics, with a tendency to fail with a low-preforming MEMS IMU. Particularly interesting, Fig. 4 shows the edge-case of numerical errors encountered in Vins-Mono [24] when having very low IMU noise values. Regarding credibility/consistency results, Fig. 5, show that none of the considered algorithms can be labeled as credible due to its under- or overconfidence and that still much research is required towards this direction.

Acknowledgement

This work is supported by the EU-H2020 project BUG-WRIGHT2 (GA 871260); ARL under grant agreement W911NF-16-2-0112; University of Klagenfurt (KPK-NAV).

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, 12 2016.

[2] J. Meyer, A. Sendobry, S. Kohlbrecher, U. Klingauf, and O. Von Stryk, "Comprehensive simulation of quadrotor UAVs using ROS and Gazebo," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 7628 LNAI, 2012, pp. 400–411.

[3] F. Furrer, M. Burri, M. Achtelik, and R. Siegwart, "RotorS—A modular gazebo MAV simulator framework," in *Studies in Computational Intelligence*, A. Koubaa, Ed. Cham: Springer International Publishing, 2016, vol. 625, ch. RotorS—A, pp. 595–625. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-26054-9_23

[4] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "AirSim: High-Fidelity Visual and Physical Simulation for Autonomous Vehicles," in *Field and Service Robotics*, 2018, pp. 621–635. [Online]. Available: https://arxiv.org/abs/1705.05065

[5] W. Guerra, E. Tal, V. Murali, G. Ryou, and S. Karaman, "FlightGoggles: Photorealistic Sensor Simulation for Perception-driven Robotics using Photogrammetry and Virtual Reality," in *IEEE International Conference on Intelligent Robots and Systems*. IEEE, 11 2019, pp. 6941–6948. [Online]. Available: https://doi.org/10.1109/iros40897.2019.8968116

[6] Y. Song, S. Naji, E. Kaufmann, A. Loquercio, and D. Scaramuzza, "Flightmare: A Flexible Quadrotor Simulator," *arXiv preprint arXiv:2009.00563*, 2020. [Online]. Available: http://arxiv.org/abs/2009.00563

[7] W. Li, S. Saeedi, J. McCormac, R. Clark, D. Tzoumanikas, Q. Ye, Y. Huang, R. Tang, and S. Leutenegger, "Interiornet: Mega-scale multi-sensor photo-realistic indoor scenes dataset," in *British Machine Vision Conference 2018, BMVC 2018*, 2019.

[8] L. Nardi, B. Bodin, M. Z. Zia, J. Mawer, A. Nisbet, P. H. Kelly, A. J. Davison, M. Luján, M. F. O'Boyle, G. Riley, N. Topham, and S. Furber, "Introducing SLAMBench, a performance and accuracy benchmarking methodology for SLAM," in *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2015-June, no. June, 5 2015, pp. 5783–5790.

[9] B. Bodin, H. Wagstaff, S. Saecdi, L. Nardi, E. Vespa, J. Mawer, A. Nisbet, M. Lujan, S. Furber, A. J. Davison, P. H. Kelly, and M. F. O'Boyle, "SLAMBench2: Multi-Objective Head-to-Head Benchmarking for Visual SLAM," in *Proceedings - IEEE International Conference on Robotics and Automation*, 5 2018, pp. 3637–3644.

[10] M. Bujanca, P. Gafton, S. Saeedi, A. Nisbet, B. Bodin, M. F. Oaboyle, A. J. Davison, P. H. Kelly, G. Riley, B. Lennox, M. Lujan, and S. Furber, "SLAMBench 3.0: Systematic automated reproducible evaluation of slam systems for robot vision challenges and scene understanding," in *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2019-May. IEEE, 2019, pp. 6351–6358.

[11] W. Ye, Y. Zhao, and P. A. Vela, "Characterizing SLAM Benchmarks and Methods for the Robust Perception Age," 5 2019. [Online]. Available: http://arxiv.org/abs/1905.07808

[12] X. Shi, D. Li, P. Zhao, Q. Tian, Y. Tian, Q. Long, C. Zhu, J. Song, F. Qiao, L. Song, Y. Guo, Z. Wang, Y. Zhang, B. Qin, W. Yang, F. Wang, R. H. M. Chan, and Q. She, "Are We Ready for Service Robots? The OpenLORIS-Scene Datasets for Lifelong SLAM," pp. 3139–3145, 11 2019. [Online]. Available: http://arxiv.org/abs/1911.05603

[13] S. Weiss and R. Siegwart, "Real-time metric state estimation for modular vision-inertial systems," *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 231855, pp. 4531–4537, 2011.

[14] Z. Zhang and D. Scaramuzza, "A Tutorial on Quantitative Trajectory Evaluation for Visual(-Inertial) Odometry," in *IEEE International Conference on Intelligent Robots and Systems*. IEEE, 2018, pp. 7244–7251.

[15] X. R. Li, Z. Zhao, and X. B. Li, "Evaluation of Estimation Algorithms: Credibility Tests," *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans*, vol. 42, no. 1, pp. 147–163, 2012.

[16] F. Devernay and O. Faugeras, "Straight lines have to be straight," *Machine Vision and Applications*, vol. 13, no. 1, pp. 14–24, 2001. [Online]. Available: http://link.springer.com/10.1007/PL00013269

[17] C. Kanan and G. W. Cottrell, "Color-to-Grayscale : Does the Method Matter in Image Recognition ?" *PLoS ONE*, vol. 7, no. 1, 2012.

[18] W. K. Pratt, "Digital Image Processing, 4th Edition," *Journal of Electronic Imaging*, 2007.

[19] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *Proceedings - IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 2520–2525.

[20] X. QIU, H. ZHANG, and W. FU, "Lightweight hybrid visual-inertial odometry with closed-form zero velocity update," *Chinese Journal of Aeronautics*, 2020.

[21] P. Geneva, K. Eckenhoff, W. Lee, Y. Yang, and G. Huang, "OpenVINS: A Research Platform for Visual-Inertial Estimation," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 4666–4672.

[22] M. Bloesch, S. Omari, M. Hutter, and R. Siegwart, "Robust visual inertial odometry using a direct EKF-based approach," in *IEEE International Conference on Intelligent Robots and Systems*, vol. 2015-Decem. IEEE, 2015, pp. 298–304.

[23] M. Bloesch, M. Burri, S. Omari, M. Hutter, and R. Siegwart, "Iterated extended Kalman filter based visual-inertial odometry using direct photometric feedback," *International Journal of Robotics Research*, vol. 36, no. 10, pp. 1053–1072, 2017.

[24] 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.

LiODOM: Adaptive Local Mapping for Robust LiDAR-Only Odometry

Emilio Garcia-Fidalgo, Joan P. Company-Corcoles, Francisco Bonnin-Pascual and Alberto Ortiz

***Abstract*— In the last decades, Light Detection And Ranging (LiDAR) technology has been extensively explored as a robust alternative for self-localization and mapping. These approaches typically state ego-motion estimation as a non-linear optimization problem dependent on the correspondences established between the current point cloud and a map, whatever its scope, local or global. This paper proposes LiODOM, a novel LiDAR-only ODOmetry and Mapping approach for pose estimation and map-building, based on minimizing a loss function derived from a set of weighted point-to-line correspondences with a local map abstracted from the set of available point clouds. Furthermore, this work places a particular emphasis on map representation given its relevance for quick data association. To efficiently represent the environment, we propose a data structure that combined with a hashing scheme allows for fast access to any section of the map. LiODOM is validated by means of a set of experiments on public datasets, for which it compares favourably against other solutions. Its performance on-board an aerial platform is also reported.**

I. Introduction

Self-localization and mapping, either performed simultaneously or in a sequential fashion, are crucial abilities for a mobile robot to be useful in relevant applications, irrespective of whether the robot operates fully autonomously or in a semi-autonomous way. As stated many years ago, odometry estimation is a fundamental piece within this framework. A plethora of sensing devices have been adopted throughout the years, comprising tachometers/wheel encoders, inertial and heading sensors, time of flight sensors, and motion estimation devices, to name but a few. Among all of them, laser scanners and, for a few years now, cameras have turned out to be the sensors of choice. The latter have been extensively used [1], [2] due to the rich perception of the surrounding world encoded in images. Vision-based estimation is however sensitive to lighting conditions, have a limited horizontal field of view and require additional calculations to acquire depth and shape perception. In contrast, 3D laser scanners provide a 360-degree overview of the platform surroundings, supply reliable range estimations and, especially motivated by the development of self-driving cars, recently have become an affordable choice for pose estimation and mapping.

LiDAR odometry is typically stated as an optimization problem that is solved using the Iterative Closed Point (ICP) algorithm [3] or any of its variants. For this to happen in a satisfactory, fast and accurate way, a set of reliable correspondences between the current point cloud and a map must be found. A KD-tree is a popular choice to represent the whole map [4], although the resulting performance degrades as the number of points to be managed increases, what makes necessary a filtering step to screen most relevant points. An alternative is to build a local map using a sliding window [5], [6], although this might discard useful associations that could be found if the search was performed over a global map.

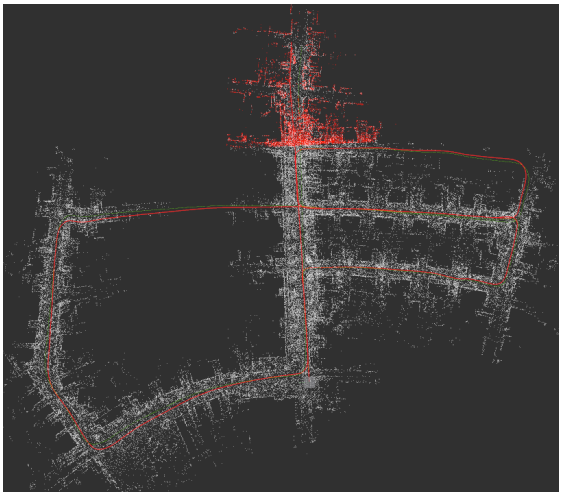


Fig. 1. Example of map produced by LiODOM (KITTI 05 sequence), comprising an unoptimized global map generated during navigation (in white) and a local map (in red) that is retrieved according to the position of the vehicle, to be used for next pose estimation.

This paper proposes LiODOM, a novel LiDAR odometry and mapping approach that is able to estimate the pose without additional sensors, e.g. IMU and/or GPS, unlike other recent approaches [5], [6]. Our approach is formulated as a non-linear optimization problem based on a set of point-to-line constraints, weighted according to the distance from each point to the sensor center. Furthermore, we propose an efficient data structure, based on a hashing scheme, to represent the map. As a result, a local map can be retrieved according to the pose of the robot in an effective way, and point cloud correspondences against the local map can be efficiently established. This *adaptive* solution naturally allows us to find correspondences between the current point cloud and revisited places (contrary to just using a sliding window). Figure 1 illustrates the performance of LiODOM.

In brief, the main contributions of this work are:

- A LiDAR-only odometry framework, inspired by the principles of LOAM [4]. It is based on an optimization

This work is partially supported by EU-H2020 projects BUGWRIGHT2 (GA 871260) and ROBINS (GA 779776), and by project PGC2018-095709-B-C21 (funded by MCIU/AEI/10.13039/501100011033 and FEDER "Una manera de hacer Europa"). This publication reflects only the authors views and the European Union is not liable for any use that may be made of the information contained therein.

All authors are with the Department of Mathematics and Computer Science (University of the Balearic Islands) and IDISBA (Institut d'Investigacio Sanitaria de les Illes Balears), Palma de Mallorca, Spain. {emilio.garcia, joanpep.company, xisco.bonnin, alberto.ortiz}@uib.es.

problem supported by weighted point-to-line factors computed from the correspondences with the local map.
- A fast and efficient hash-based data structure for mapping, speeding up searches and gracefully updating on large-scale maps.
- An extensive evaluation of the proposed approach on several public datasets, including a comparison with other state-of-the-art methods. Its performance on-board an aerial platform is also reported.
- As an additional contribution, we make available to the community the source code[1] of our approach.

The rest of the paper is organized as follows: Section II overviews most relevant works in the field; the proposed framework is introduced in Sections III, IV and V; Section VI reports on the results obtained; to finish, Section VII concludes the paper and suggests future research lines.

II. Related Work

Most recent approaches carry out LiDAR-based odometry in combination with an IMU for higher accuracy. These solutions are typycally regarded as loosely- and tightly-coupled methods [6], [7]. Loosely-coupled methods estimate the state from each sensor separately. Arguably the most well-known method that falls into this category is LiDAR Odometry and Mapping (LOAM) [4], where *edges* and *surfaces* are detected and registered to a map through point-to-line and point-to-plane constraints within an optimization framework. In LOAM, an IMU can be optionally used to de-skew the input point cloud and provide a prior motion estimate. LOAM extensions proposed to be used specifically on ground vehicles or with solid-state LiDARs can be found in, respectively, [8] and [9]. More recently, a lightweight LOAM version named F-LOAM [10] has been proposed. This is probably the closest work to our solution. In this respect, LiODOM introduces a simpler but efficient pose optimization scheme and a novel mapping approach, resulting into more robust estimations, as shown in Section VI. Another option in this class is to fuse sensor data by means of an Extended Kalman Filter (EKF) [11], [12].

Tightly-coupled methods fuse sensor data jointly, either through optimization [5], [6] or filtering [7], [13]. In this regard, Ye et al. [5] introduces LIOM, a tightly-coupled odometry and mapping approach which jointly minimizes LiDAR and IMU observations in a sliding window. Despite its good performance, it is computationally expensive, making difficult its use in practical situations. In a more recent work [7], the same authors opt for an iterated Error-State Kalman Filter (ESKF), resulting into a faster solution. A recent work [6] introduces LIO-SAM as a new tightly-coupled method. In LIO-SAM, LiDAR-inertial odometry is stated as a factor graph, allowing to easily incorporate any type of observation as a constraint, such as loop closures, GPS or IMUs. Unlike the approaches surveyed so far, we tackle the problem of pose estimation using solely a LiDAR.

[1] http://github.com/emiliofidalgo/liodom

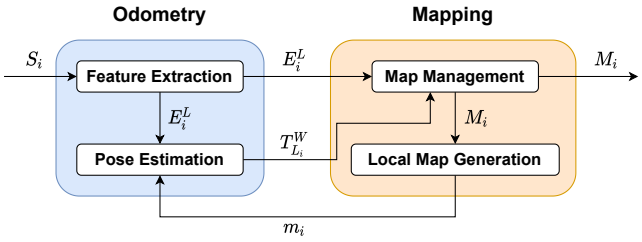


Fig. 2. Overview of LiODOM.

As mentioned above, establishing a set of correspondences between the input scan and a map is of prime importance for efficient pose estimation. Some authors have opted for indexing the points of a global map using a tree-based approach [4], although usually these solutions do not scale well. In our work, the global map is devised as a disjoint partition of the 3D space, and, inspired by other approaches [14], [15], it is indexed using a hashing scheme. An alternative for fast data association is building a local map from an sliding window [5], [6], instead of matching directly to a global map, but this option tends to discard useful correspondences. In this matter, our approach also introduces an adaptive local map mechanism, which can be seen as an alternative to the classical local mapping paradigm.

III. System Overview

For a start, we define a *sweep* as a set of 360-degree 2D scans. A sweep received at time i is denoted as S_i. We also define two coordinate systems: (1) L, the *LiDAR* coordinate system, which is a frame attached to the geometric center of the sensor; and (2) W, the world coordinate system, which coincides with L at the beginning. We denote by $T_B^A \in SE(3)$ the transformation that maps a point $p^B \in \mathbb{R}^3$ expressed in B to a point $p^A \in \mathbb{R}^3$ expressed in A. The rotation matrix and the translation vector of T_B^A are respectively denoted by $R_B^A \in SO(3)$ and $t_B^A \in \mathbb{R}^3$.

Figure 2 illustrates LiODOM. As in other works [4], [5], our solution consists of two main components, odometry and mapping, which run concurrently: while the odometry module (Sec. IV) computes a set of features, i.e. LOAM edges, from S_i and estimates the current pose of the LiDAR $T_{L_i}^W$, the mapping module (Sec. V) registers the resulting edges to a global map and generates an adaptive local map to be employed in the subsequent pose estimation step.

IV. LiDAR Odometry

The LiDAR odometry module is, in turn, structured in two synchronized execution threads to decouple feature extraction from pose estimation. Both are described next.

A. Feature Extraction

For a start, each sweep S_i is divided into its different scans, discarding at the same time those points whose range do not fall within a certain interval $[r_{\text{min}}, r_{\text{max}}]$, which has to be configured accordingly to the sensor operating and noise characteristics. Each pre-processed scan is next considered,

Algorithm 1 LiDAR Odometry

Input: S_i, $T^W_{L_{i-1}}$, $T^W_{L_{i-2}}$, m_{i-1}
Output: E^L_i, $T^W_{L_i}$

1: $E^L_i \leftarrow$ set of edges from S_i
2: $T^W_{L_i} \leftarrow \widehat{T}^W_{L_i} \equiv$ initial transformation estimate [Eq. (6)]
3: **for** n iterations **do**
4: **for** $p^{L_i}_j \in E^L_i$ **do**
5: $p^W_j \leftarrow T^W_{L_i} p^{L_i}_j$ [Eq. (1)]
6: $N(p^W_j) \leftarrow 5$ NN of p^W_j in m_{i-1}
7: **if** $N(p^W_j)$ is a line **then**
8: Compute $d_e(p^W_j, l(p^W_j))$ [Eq. (2)]
9: Compute residual $\varrho_e(p^W_j, l(p^W_j))$ [Eq. (3)]
10: Add residual $\varrho_e(p^W_j, l(p^W_j))$
11: to the optimization problem
12: Optimize pose $T^W_{L_i}$ [Eq. (5)]

selecting a number of key points to reduce the computational requirements. In this regard, points on sharp edges or on locally planar surfaces are the choice in most cases, e.g. LOAM features, given their utility for the intended purpose and their simpler computation [4]. In this work, we have observed that using only edges is a good trade-off between accuracy and efficiency. To select the best features, a local smoothness measure c is calculated for each point, as in [4]. Moreover, to distribute edges throughout the environment, a scan is further divided into equally-sized sectors, and a maximum number of edges is set for every sector. Unlike [4], we split each scan into 8 sectors and choose a maximum of 10 edges per sector after sorting them in decreasing order of curvature c. Furthermore, the selection applies non-maxima suppression, i.e. a point is chosen as an edge if none of its neighbours has been already selected. The result of this procedure is a set of edges E^L_i chosen from sweep S_i.

B. Pose Optimization

Let us consider the transformation $T^W_{L_i}$ from the LiDAR at time i to the world. Then, every point $p^{L_i}_j \in E^L_i$ projects into the world frame W as:

$$p^W_j = T^W_{L_i} p^{L_i}_j = R^W_{L_i} p^{L_i}_j + t^W_{L_i}\,, \tag{1}$$

being $R^W_{L_i}$ and $t^W_{L_i}$ the respective rotation matrix and translation vector of $T^W_{L_i}$. We denote the set of transformed edges as E^W_i. Subsequently, a set of point-to-line correspondences between E^W_i and a local map are computed for pose estimation. We have opted for this solution rather than using, for instance, a global map, because it turns out to be more computationally stable as more frames are processed. In LiODOM, that local map is not built after pose estimation in a sequential way as in [5], [6], but it is built concurrently with pose estimation by the mapping module (see Section V).

Let us assume now the existence of a local map m_{i-1} at time $i-1$, which is a subset of the global map M_{i-1}. This map m_{i-1} contains the points in M_{i-1} closest to the LiDAR according to the latest pose estimate $T^W_{L_{i-1}}$. For each point $p^W_j \in E^W_i$, we obtain the k nearest points in m_{i-1}, where $k = 5$ in this work. Let us denote this set as $N(p^W_j)$ and the n-th nearest neighbour of p^W_j as $N_n(p^W_j)$. We next assess whether points in $N(p^W_j)$ are aligned by analyzing their scatter matrix [9]. If the largest eigenvalue of this matrix is, at least, three times the second largest eigenvalue, we consider that a valid point-to-line correspondence can be established between p^W_j and the line $l(p^W_j)$ resulting from $N_1(p^W_j)$ and $N_2(p^W_j)$. We then calculate the point-to-line distance d_e as

$$d_e(p^W_j, l(p^W_j)) = \frac{\left\| \left(p^W_j - N_1(p^W_j)\right) \times N_{12} \right\|}{\| N_{12} \|}\,, \tag{2}$$

with $N_{12} = N_1(p^W_j) - N_2(p^W_j)$.

LiODOM, as an odometer, optimizes only the current pose of the LiDAR $T^W_{L_i}$. Within the optimization framework, each correspondence provides a constraint between $T^W_{L_i}$ and the local map m_{i-1}, whose residual ϱ_e is computed as:

$$\varrho_e(p^W_j, l(p^W_j)) = \omega_j \, d_e(p^W_j, l(p^W_j))\,, \tag{3}$$

where ω_j is a weighting term computed as:

$$\omega_j = 1 - \frac{r_j - r_{\min}}{r_{\max} - r_{\min}}\,, \tag{4}$$

being r_j the range returned by the LiDAR for edge $p^{L_i}_j$. The rationale behind this factor is that LiDARs tend to decrease their accuracy at longer distances and, thus, we give more importance to correspondences established at closer distances. We then compute the optimal transformation $T^W_{L_i}$ as the minimizer of the loss function $J(\widetilde{T}^W_{L_i}, \Upsilon)$:

$$J(\widetilde{T}^W_{L_i}, \Upsilon) = \frac{1}{2} \sum_{j \in \Upsilon} \rho\left(\left\| \varrho_e \left(\widetilde{T}^W_{L_i} p^{L_i}_j, l\left(\widetilde{T}^W_{L_i} p^{L_i}_j \right) \right) \right\|^2 \right)$$
$$T^W_{L_i} = \min_{\widetilde{T}^W_{L_i}} J(\widetilde{T}^W_{L_i}, \Upsilon) \tag{5}$$

where Υ is the set of correspondences established between E^W_i and the local map m_{i-1}, and ρ is a Huber loss function to reduce the influence of outliers. The system of non-linear equations is solved by means of the Levenberg-Marquardt algorithm using the Ceres Solver library [16], using transformation $\widehat{T}^W_{L_i}$ as initial guess:

$$\widehat{T}^W_{L_i} = T^W_{L_{i-1}} \widehat{T}^{L_{i-1}}_{L_i}$$
$$= T^W_{L_{i-1}} T^{L_{i-2}}_{L_{i-1}} = T^W_{L_{i-1}} \left(T^W_{L_{i-2}} \right)^{-1} T^W_{L_{i-1}}\,, \tag{6}$$

i.e. we assume the same motion as for the previously estimated pose. Although LiODOM deals only with LiDAR data, it is clear that any additional motion estimate, e.g. from an IMU, can be incorporated at this point.

The full LiDAR odometry procedure is stated algorithmically in Alg. 1. In our experiments, 1 or 2 refining iterations are enough, i.e. $n = 2$ or 3 at line 3 of Alg. 1.

V. LiDAR Mapping

The registration of the extracted edges E^L_i on the global map M_i is performed by the mapping module using the last optimized pose $T^W_{L_i}$. This module also generates the corresponding local map m_i as described next.

Algorithm 2 LiDAR Mapping

Input: E_i^L, $T_{L_i}^W$
Output: M_i, m_i

for $p_j^{L_i} \in E_i^L$ **do**
  $p_j^W \leftarrow T_{L_i}^W p_j^{L_i}$ [Eq. (1)]
  $C_q \leftarrow$ cell where p_j^W should be [Eq. (8)]
  if $H(C_q) \notin \mathbb{H}$ **then**
    Create new cell C_n using C_q coordinates
    Add p_j^W to C_n
    Update M_i adding C_n to $\mathbb{C}$
    Update M_i adding $H(C_q)$ to $\mathbb{H}$
  else
    Retrieve cell C_q using $H(C_q)$
    Update M_i adding p_j^W to C_q
    if C_q has more points than τ **then**
      Update M_i filtering C_q using a 3D voxel grid
$C_{L_i} \leftarrow$ cell where the LiDAR should be [Eq. (8)]
$m_i \leftarrow \emptyset$
for $C_i \in$ Neighbours of C_{L_i} in M_i **do**
  $m_i \leftarrow m_i \cup C_i$

A. Map Representation

Given the high frequency at which the map must be accessed, the type of data structure chosen to represent 3D space becomes crucial for fast operation. A single KD-tree has been typically used to this end [4]. However, this option presents several drawbacks, such as, on the one hand, the full tree tends to change as points are added or deleted to/from the tree, and, on the other hand, the KD-tree performance decreases as more points need to be managed [4]. To overcome these issues, in LiODOM we introduce an efficient hashing data structure for representing the map taking inspiration from other recent works [14], [15]. To be more specific, the 3D space is partitioned into a set of disjoint cuboids of a fixed size that we name *cells*. A cell C_j is represented by its geometric center, denoted by (c_{jx}, c_{jy}, c_{jz}), and includes all 3D points whose coordinates fall into its limits. We define a map at time i as $M_i = \{\mathbb{H}_i, \mathbb{C}_i\}$, where $\mathbb{H}_i$ is a hash table and $\mathbb{C}_i$ is the set of existing cells up to time i. The table $\mathbb{H}_i$ allows us to rapidly access to a specific cell C_j using a hashing function of its coordinates, defined by:

$$H(C_j) = (c_{jx} \oplus (c_{jy} << 1)) \oplus (c_{jz} << 2)\,, \tag{7}$$

where $\oplus$ and $<<$ are, respectively, the bitwise XOR and the left shift operators. This function has been selected in order to minimize, as much as possible, hash collisions.

B. Map Updates

In LiODOM, map updates are performed once per sweep, being the set of edges E_i^L, extracted from S_i, and the last optimized transformation $T_{L_i}^W$ the input data. Unlike other approaches [4], where the raw point cloud is used for mapping, in our approach, the map is built using directly the edges to speed up the mapping procedure, resulting into more sparse maps. Initially, every point $p_j^{L_i} \in E_i^L$ is transformed to world coordinates using $T_{L_i}^W$ and (1). Next, for each point $p_j^W = (x, y, z)$, we compute the geometric center of the cell C_q in which the point should be stored as:

$$\begin{bmatrix} c_{qx} \\ c_{qy} \\ c_{qz} \end{bmatrix} = \begin{bmatrix} \lfloor x/s_{xy} \rfloor \; s_{xy} + \frac{1}{2}\, s_{xy} \\ \lfloor y/s_{xy} \rfloor \; s_{xy} + \frac{1}{2}\, s_{xy} \\ \lfloor z/s_z \rfloor \; s_z + \frac{1}{2}\, s_z \end{bmatrix}, \tag{8}$$

where s_{xy} and s_z are the metric cell sizes for the corresponding dimension. We next check if the cell C_q is already in the map by querying the hash table $\mathbb{H}$ using the key $H(C_q)$. If this is the case, the point is added to the existing cell. Otherwise, a new cell C_n is created with point p_j^W as seed, to be added next to $\mathbb{C}$ and indexed on $\mathbb{H}$ by $H(C_n)$. Finally, modified cells exceeding a certain number of points are filtered using a 3D voxel grid. Note that our data structure allows us to rapidly update just the required areas of the environment, avoiding the update of the whole map on each iteration. This fact contributes to speed up the mapping procedure, as will be shown in the experiments.

C. Adaptive Local Map Computation

Lastly, the mapping module generates a local map m_i, which contains the points of M_i within a certain range from the current LiDAR pose. Assuming a moderate motion between two consecutive sweeps, these points are enough to find correspondences for the next pose estimation step. To build the local map, we first retrieve the cell C_{L_i} where the LiDAR is located at that moment using its current position $T_{L_i}^W$ and (8). Next, assuming a 3D grid arranged over M_i, neighbouring cells of C_{L_i} up to a certain distance are further retrieved from M_i, and their corresponding points are merged to form the local map m_i. This operation results to be very fast due to the proposed hashing structure.

On the other side, we refer to this local map as *adaptive* since it always covers a specific area of the environment, contrary to a local map built by aggregation of a sliding window [5], [6]. Besides, it provides us with correspondences with revisited areas of the environment in a natural way. Additionally, the availability of m_i avoids us to search for correspondences against the whole map, as done by other solutions [4]. Finally, to avoid reduced amounts of points from unexplored areas, we always add the last three sweeps to m_i. The complete mapping procedure is outlined in Alg. 2.

VI. Experimental Results

In this section, we report on the results of several experiments conducted to evaluate LiODOM, including a comparison with other solutions. A laptop featuring an Intel Core i7-10750H @2.6Ghz, 16 GB RAM has been used in all cases.

A. Methodology

We validate our approach using the KITTI odometry benchmark [17]. This dataset consists of 22 sequences collected using a Velodyne HDL-64E sensor. Eleven of these sequences include GPS poses that can be used as

ground truth. The average translational (%) and rotational (deg/100m) errors are adopted in the following as main performance metrics. We additionally consider the Absolute Trajectory Error (ATE), although it rather focuses on the global consistency of the whole trajectory and thus it is more appropriate for SLAM systems.

To further validate LiODOM, we compare it with other pure LiDAR-based odometry and also with SLAM solutions, namely F-LOAM [10], ISC-LOAM [18] and LeGO-LOAM [8]. We are aware of the existence of recent fusion-based [5], [6] or even semantic-aided [19] solutions. They are not considered in this evaluation since, in contrast to our method, they imply additional complexities, such as synchronization and calibration procedures or increasing computational resources.

B. Odometry Performance

Table II summarizes the results obtained in terms of translational and rotational errors. Results for F-LOAM were obtained by ourselves using its open source implementation, while results for ISC-LOAM and LeGo-LOAM are directly reported from, respectively, [19] and [20]. As can be obseved, LiODOM achieves competitive results in all sequences in terms of translation error. This can be observed even in sequences comprising loop closures, such as K05, K06 and K07, where our approach achieves the second best results, sometimes very close to complete SLAM solutions like ISC-LOAM. We obtain, on average, 1.038% drift in translation, outperforming the other solutions in this matter. Regarding rotation error, again our solution leads to the lowest errors in most of the sequences. On average, the rotational error of LiODOM is 0.296% deg / 100m, which represents again the lowest average error.

Table II reports on the ATE for the KITTI sequences that contain loop closures. Again, results for F-LOAM were obtained by ourselves, while results for ISC-LOAM and LeGo-LOAM are reported from, respectively, [19] and [21]. LiODOM again achieves competitive results in all sequences despite it is actually a pure odometry system and, therefore, does not take any advantage from global map optimization nor from loop closures. On average, the ATE for LiODOM results to be 3.535 m, which represents the second best performance among the different methods considered. By way of illustration, Fig. 3 shows the resulting trajectory estimates from our approach and from F-LOAM for several KITTI sequences.

To finish, we choose the largest dataset considered in this work (K02) to analyze the computational complexity of LiODOM. Average response times for every odometry stage can be found in Fig. 4. These are the stages that should operate online. The capabilities of the mapping module, which is executed as a standalone procedure, are evaluated in the next section. As can be noticed, feature extraction and pose estimation takes respectively 10.36 ms and 75.15 ms on average, resulting into an overall response time of 85.51 ms per frame. The latter also includes the search for correspondences within the local map. This response time means a frame rate of around 11 Hz, which represents a good trade-off between computational requirements and precision, in contrast to other solutions [10].

C. Mapping Performance

In this section, we report on an experiment intended to assess the efficiency of the mapping approach adopted in LiODOM. In this experiment, we measure the times required to update the global map and to build the local maps using our hashing-based data structure and a KD-tree. The K05 sequence was chosen in this case for computational reasons. The results are shown in Fig. 5. As can be observed, the time required to update the global map by our approach remains roughly constant along the whole sequence. Contrarily, the running times for the KD-tree approach grow as more frames are processed, which can lead to an impractical operation. This behaviour can be attributed to the fact that, unlike our approach, the whole tree needs to be rebuilt on each update. The differences are less evident as for the times required to build the local maps, where both approaches are very fast, although our approach seems to perform slightly better.

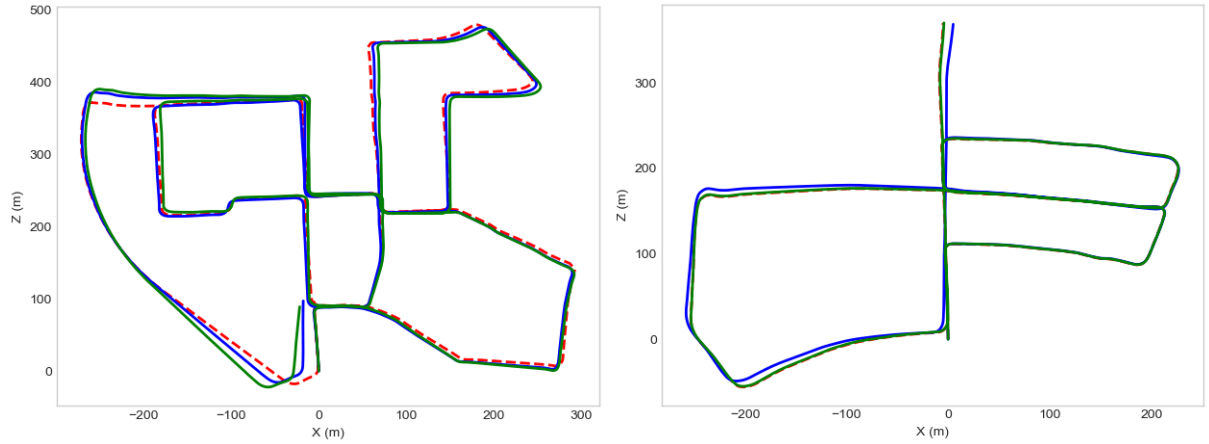
D. Experiments on-board an Aerial Platform

Finally, we also report on some experiments conducted on-board an aerial platform intended for visual inspection tasks [22]. This platform has been recently fitted with an Ouster OS1-64 3D laser scanner that feeds LiODOM. The experiments have been carried out inside the Aerial Robotics laboratory of the University of the Balearic Islands, which is equipped with an OptiTrack Motion Capture system (MOCAP) that supplied ground truth data during the tests. Table III shows the ATE for five different experiments. As can be noticed, the ATE values range from 17 to 30 cm, indicating that position estimates closely resemble the ground truth. On the other side, by way of example, Fig. 6 compares graphically the position estimates from LiODOM with the ground truth during one of these experiments; a perspective view is shown in Fig. 7. In this case, X and Y estimates mostly coincide with the ground truth, while, as also happens for other LiDAR-based odometry frameworks [8], [10], [18], some drift can be appreciated at the end regarding the Z-axis estimates.

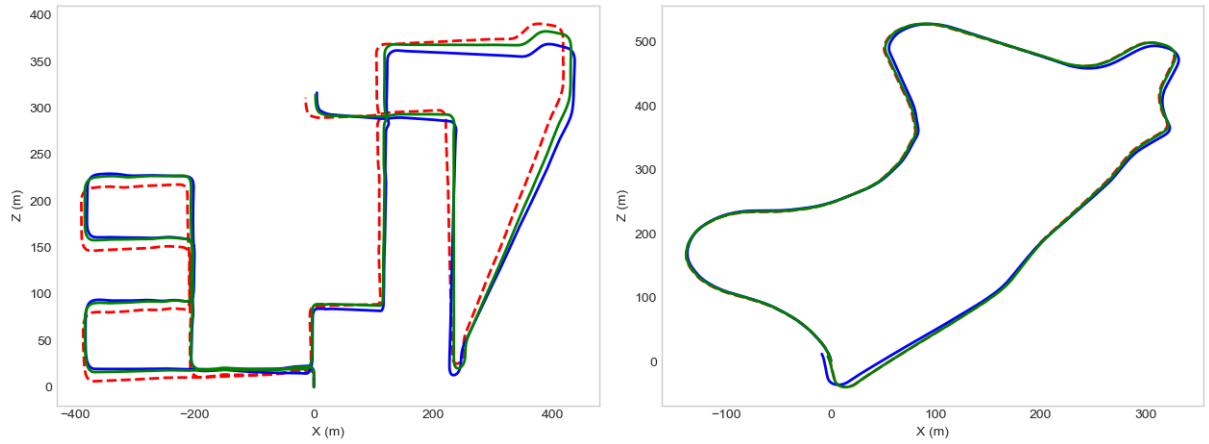
Within this robotic system, developed under the Supervised Autonomy paradigm, LiODOM is expected to supply not only pose estimates, but also velocity estimates, which constitute the basis for platform control in this case. Table III reports on the velocity estimation results for the same five experiments as above, in the form of Root Mean Square Errors (RMSE) separately for each axis. The reported values indicate a very high accuracy in the estimation of X and Y velocities, and a slightly larger error for the Z axis. To finish, Fig. 8 compares graphically the vehicle velocities estimated by LiODOM with the values provided by the MOCAP for one of these experiments. As also observed for the position estimates, the X and Y velocity estimates coincide almost perfectly with the ground truth, while the Z-axis estimates are less accurate at certain moments.

TABLE I
AVERAGE TRANSLATIONAL AND ROTATIONAL ERRORS FOR THE KITTI ODOMETRY BENCHMARK.
BEST RESULTS ARE SHOWN IN BOLD RED AND SECOND BEST IN BLUE.

	Translational Error (%)				Rotational Error (deg/100m)			
	F-LOAM	**ISC-LOAM**	**LeGO**	**Ours**	**F-LOAM**	**ISC-LOAM**	**LeGO**	**Ours**
K00	0.861	1.020	2.170	**0.857**	0.349	0.420	1.050	**0.348**
K01	1.309	2.920	13.400	**1.301**	**0.128**	0.630	1.020	0.129
K02	0.952	1.670	2.170	**0.947**	0.310	0.540	1.010	**0.309**
K03	1.267	**1.150**	2.340	1.262	0.227	0.720	1.180	**0.226**
K04	1.417	1.500	**1.270**	1.411	0.010	0.560	1.010	**0.009**
K05	0.835	**0.810**	1.280	0.834	0.360	0.370	0.740	**0.359**
K06	0.835	**0.760**	1.060	0.834	0.332	0.410	0.630	**0.331**
K07	0.883	**0.560**	1.120	0.881	0.617	**0.430**	0.810	0.614
K08	0.869	1.200	1.990	**0.864**	0.332	0.500	0.940	**0.331**
K09	1.033	1.400	1.970	**1.029**	**0.317**	0.590	0.980	0.318
K10	1.203	1.870	2.210	**1.196**	**0.287**	0.620	0.920	0.288
Average	1.042	1.351	2.816	**1.038**	0.297	0.526	0.935	**0.296**



K00 K05



K08 K09

Fig. 3. Examples of trajectories estimated for some sequences of the KITTI odometry benchmark. The ground truth is shown as a red dashed line, while F-LOAM and LiODOM estimates are respectively shown as blue and green lines.

VII. CONCLUSIONS AND FUTURE WORK

This paper proposes LiODOM, a novel LiDAR-only odometry and mapping approach. Our solution fundamentally consists of two parts working concurrently: (1) the odometry module, which is in charge of extracting a set of edges from the input sweep and estimating the current pose of the LiDAR; and (2) the mapping module, which builds and maintains a global map of the environment, and also generates a local map employed for pose estimation. Pose

TABLE II

ABSOLUTE TRAJECTORY ERROR (M) FOR THE KITTI DATASET. BEST RESULTS ARE SHOWN IN BOLD RED AND SECOND BEST IN BLUE.

	F-LOAM	ISC-LOAM	LeGO	Ours
K00	5.137	**1.600**	6.300	7.135
K02	9.294	**4.770**	14.700	9.754
K05	2.546	2.490	2.800	**0.322**
K06	0.934	1.030	**0.800**	0.956
K07	**0.498**	0.560	0.700	1.518
K08	4.344	4.880	**3.500**	4.592
K09	2.144	2.310	2.100	**0.470**
Average	3.557	**2.520**	4.414	3.535

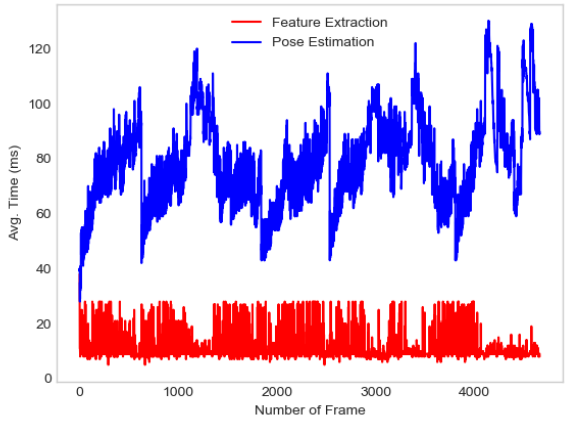


Fig. 4. Average response time for each odometry estimation stage.

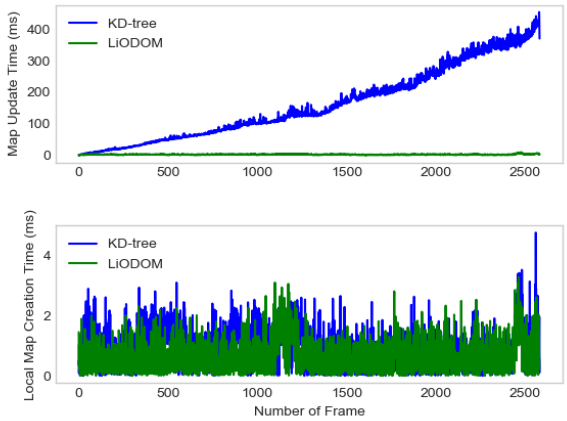


Fig. 5. Performance of the LiODOM mapping structure vs. a KD-tree.

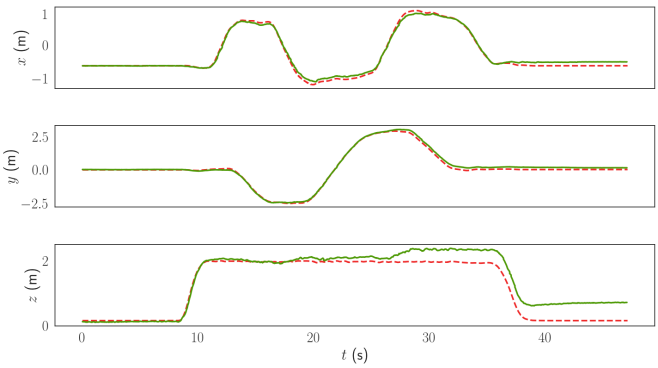


Fig. 6. Position estimates for one of the experiments running LiODOM on-board an aerial platform. LiODOM estimates are shown in green, while the ground truth is shown as a red dashed plot.

estimation is conceived as an optimization problem which involves a set of weighted point-to-line constraints between the current sweep and a local map. We have also described a data structure based on a hashing scheme which allows us to rapidly get access to any part of the map and manage it in an efficient way. Furthermore, this structure is also employed to obtain an adaptive local map, used to facilitate data association. Our experiments show that LiODOM compares favourably against other state-of-the-art approaches, and that it can be used for both position and velocity estimation.

Despite its good performance, LiODOM is an odometer and unavoidably drifts. Therefore, we will consider extending the ideas proposed in this paper to develop a complete SLAM / 3D reconstruction system, incorporating other motion estimation sensors into a fusion scheme for enhanced performance.

TABLE III

ATE AND VELOCITY RMSE FOR THE EXPERIMENTS ON-BOARD A UAV

Experiment	ATE	Velocity RMSE (x / y / z)
1	0.177	0.039 / 0.038 / 0.092
2	0.306	0.046 / 0.040 / 0.105
3	0.173	0.034 / 0.041 / 0.066
4	0.244	0.063 / 0.048 / 0.080
5	0.272	0.040 / 0.048 / 0.062

** Values in m and m/s.

REFERENCES

[1] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1255–1262, 2017.

[2] M. Ferrera, A. Eudes, J. Moras, M. Sanfourche, and G. Le Besnerais, "OV^{2}SLAM: A Fully Online and Versatile Visual SLAM for Real-Time Applications," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1399–1406, 2021.

[3] P. Besl and N. D. McKay, "A Method for Registration of 3-D Shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 239–256, 1992.

[4] J. Zhang and S. Singh, "LOAM: Lidar Odometry and Mapping in Real-time," in *Robot.: Sci. Syst.*, vol. 2, no. 9, 2014.

[5] H. Ye, Y. Chen, and M. Liu, "Tightly Coupled 3D Lidar Inertial Odometry and Mapping," in *IEEE Int. Conf. Robot. Autom.*, 2019, pp. 3144–3150.

[6] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and R. Daniela, "LIO-SAM: Tightly-coupled Lidar Inertial Odometry via Smoothing and Mapping," in *IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 5135–5142.

[7] C. Qin, H. Ye, C. E. Pranata, J. Han, S. Zhang, and M. Liu, "LINS: A Lidar-Inertial State Estimator for Robust and Efficient Navigation," in *IEEE Int. Conf. Robot. Autom.*, 2020, pp. 8899–8906.

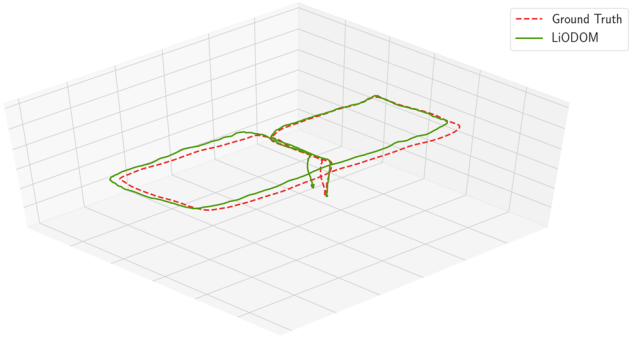


Fig. 7. Perspective view of the trajectory estimated by LiODOM for one of the experiments running LiODOM on-board an aerial platform.

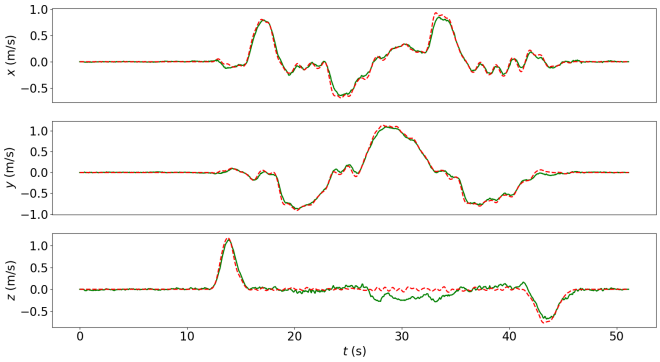


Fig. 8. Velocity estimates for one of the experiments running LiODOM on-board an aerial platform. LiODOM estimates are shown in green, while the ground truth is shown as a red dashed plot.

[8] T. Shan and B. Englot, "LeGO-LOAM: Lightweight and Ground-Optimized Lidar Odometry and Mapping on Variable Terrain," in *IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 4758–4765.

[9] J. Lin and F. Zhang, "Loam Livox: A Fast, Robust, High-Precision LiDAR Odometry and Mapping Package for LiDARs of Small FoV," in *IEEE Int. Conf. Robot. Autom.*, 2020, pp. 3126–3131.

[10] H. Wang, C. Wang, C. Chen, and L. Xie, "F-LOAM: Fast LiDAR Odometry and Mapping," in *IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020.

[11] S. Yang, X. Zhu, X. Nian, L. Feng, X. Qu, and T. Ma, "A Robust Pose Graph Approach for City Scale LiDAR Mapping," in *IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 1175–1182.

[12] M. Demir and K. Fujimura, "Robust Localization with Low-Mounted Multiple LiDARs in Urban Environments," in *IEEE Intell. Transp. Syst.*, 2019, pp. 3288–3293.

[13] W. Xu and F. Zhang, "FAST-LIO: A Fast, Robust LiDAR-Inertial Odometry Package by Tightly-Coupled Iterated Kalman Filter," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3317–3324, 2021.

[14] J. Lin and F. Zhang, "A Fast, Complete, Point Cloud based Loop Closure for LiDAR Odometry and Mapping," *arXiv e-prints: abs/1909.11811*, 2019.

[15] S. Zhao, H. Zhang, P. Wang, L. Nogueira, and S. A. Scherer, "Super Odometry: IMU-centric LiDAR-Visual-Inertial Estimator for Challenging Environments," *arXiv e-prints: abs/2104.14938*, 2021.

[16] S. Agarwal, K. Mierle, and Others, "Ceres solver," http://ceres-solver.org.

[17] A. Geiger, P. Lenz, and R. Urtasun, "Are We Ready for Autonomous Driving? The KITTI Vision Benchmark Suite," in *IEEE Conf. Comput. Vision Pattern Recog.*, 2012.

[18] H. Wang, C. Wang, and L. Xie, "Intensity Scan Context: Coding Intensity and Geometry Relations for Loop Closure Detection," in *IEEE Int. Conf. Robot. Autom.*, 2020, pp. 2095–2101.

[19] L. Li, X. Kong, X. Zhao, W. Li, F. Wen, H. Zhang, and Y. Liu, "SA-LOAM: Semantic-aided LiDAR SLAM with Loop Closure," in *IEEE Int. Conf. Robot. Autom.*, 2021.

[20] X. Zheng and J. Zhu, "Efficient LiDAR Odometry for Autonomous Driving," *arXiv e-prints: abs/2104.10879*, 2021.

[21] M. Yokozuka, K. Koide, S. Oishi, and A. Banno, "LiTAMIN2: Ultra Light LiDAR-based SLAM using Geometric Approximation applied with KL-Divergence," *arXiv e-prints: abs/2103.00784*, 2021. [Online]. Available: https://arxiv.org/abs/2103.00784

[22] F. Bonnin-Pascual, E. Garcia-Fidalgo, J. P. Company-Corcoles, and A. Ortiz, "MUSSOL: A Micro-Uas to Survey Ship Cargo hOLds," *Remote Sens.*, vol. 13, no. 3419, 2021.

A FastSLAM Approach Integrating Beamforming Maps for Ultrasound-Based Robotic Inspection of Metal Structures

Othmane-Latif Ouabi, *Graduate Student Member, IEEE*, Pascal Pomarede, Matthieu Geist, Nico F. Declercq, and Cédric Pradalier, *Member, IEEE*

***Abstract*—We present a novel FastSLAM approach for a robotic system inspecting structures made of large metal plates. By taking advantage of the reflections of ultrasonic guided waves on the plate boundaries, it is possible to recover, with enough precision, both the plate shape and the robot trajectory. Contrary to our previous work, this approach takes into account the dispersive nature of guided waves in metal plates. This is leveraged to construct beamforming maps from which we solve the mapping problem through plate edges estimation for every particle, in a FastSLAM fashion. It will be demonstrated, with real acoustic measurements obtained on different metal plates, that such a framework achieves more accurate results, while the complexity of the algorithm is sensibly reduced.**

***Index Terms*—Industrial robots, range sensing, SLAM.**

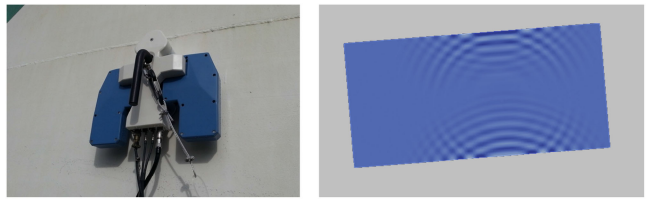


Fig. 1. (Left) A magnetic crawler carrying out an inspection task on a metal structure. (Right) Guided waves reflected by the edges of a plate in a simulation environment. We aim to enable on-plate localization and mapping with a high precision for magnetic crawlers equipped with acoustic transducers, and relying on such ultrasonic reflections.

I. Introduction

In this work[1], we describe a new FastSLAM approach [1] to achieve Simultaneous Localization and Mapping (SLAM) for a robotic system relying on Ultrasonic Guided Waves (UGWs) to support inspection tasks on large metal structures such as storage tanks or ship hulls. In Structural Health Monitoring (SHM), acoustic tomography techniques can be used for defect detection and characterization, but they rely on the accurate prior knowledge of the positions of the sensors which are integrated into the structure [2], [3]. To deploy similar methods on a robotic platform, recovering the robot position with respect to the individual metal plates may be beneficial, as it could lead, in combination with external localization systems, to precise localization of the mobile unit, and thus, to accurate inspection results.

On metal plates, guided waves are often generated by applying piezo-electric transducers in contact with the surface. These waves propagate radially around the emitter through the plate material, and potentially over large distances. When encountering the plate edges, these waves are reflected perpendicularly, and a receiver can sense the reflections. In this setup, the resulting acoustic data carry essential information on the source position and the plate geometry.

In this work, we consider a mobile unit equipped with acoustic transducers for both emission and reception, and moving on a metal surface. We leverage the sensing of the ultrasonic reflections to estimate both the plate shape and the robot trajectory. The principle of this approach is illustrated in Fig. 1. In the robotic field, this problem is known as Simultaneous Localization and Mapping (SLAM).

One of the significant challenges arises from the dispersive nature of UGWs [4]. It means that the propagation velocity is a function of the wave frequency, resulting in a waveform deformation when the propagation distance increases. Besides, propagation in metal plates is highly reverberant. These characteristics account for the relative complexity of acoustic data and call for specific processing methods to achieve on-plate localization and mapping with high accuracy. On the robotic aspect, recent works consider the similar problem of room shape reconstruction from acoustic echoes [5], [6]. As the sound velocity in the air is constant, the determination, from the measurements, of the first-order reflections is not a significant issue. However, identifying several echoes from guided wave data is

Manuscript received October 14, 2020; accepted February 15, 2021. Date of publication February 26, 2021; date of current version March 18, 2021. This letter was recommended for publication by Associate Editor M. Kaess and Editor S. Behnke upon evaluation of the reviewers' comments. *(Corresponding author: Othmane-Latif Ouabi.)*

Othmane-Latif Ouabi, Pascal Pomarede, and Cédric Pradalier are with the International Research Lab. Georgia Tech-CNRS in Metz, Metz 57070, France (e-mail: oouabi@georgiatech-metz.fr; Pascal.POMAREDE@ensam.eu; cedric.pradalier@gmail.com).

Matthieu Geist is with the Google Research, Paris, France (e-mail: mfgeist@google.com).

Nico F. Declercq is with the International Research Lab. Georgia Tech-CNRS in Metz, Metz 57070, France and also with the Georgia Institute of Technology, Atlanta, GA 30332-0250 USA (e-mail: declercq@gatech.edu).

This letter has supplementary downloadable material available at https://doi.org/10.1109/LRA.2021.3062600, provided by the authors.

Digital Object Identifier 10.1109/LRA.2021.3062600

[1]This work is part of the BugWright2 project. This project is supported by the European Commission under Grant 871260 - BugWright2.

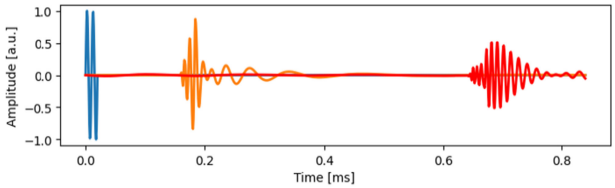


Fig. 2. Illustration of wave dispersion in plates with simulated data. The excitation signal is in blue, the signal propagated after 0.5 meters in orange, and the signal propagated after 2 meters in red.

more difficult due to the wave dispersion and the wave packets overlapping.

In our previous work [7], ultrasonic measurements on metal plates have proven to yield sufficient information to provide both localization and mapping capabilities on metal plates. However, the dispersive nature of the waves was not taken into account and the relative complexity of the algorithm may jeopardize its robustness and accuracy. In this paper, we present an alternative method to solve the SLAM problem from ultrasonic measurements. First, a wave propagation model is introduced and is leveraged to detect acoustic reflections. From them, we build beamforming maps [8] which are subsequently integrated into a FastSLAM framework to solve the mapping problem. Our approach achieves more accurate results than our previous method on real data, with less algorithmic complexity.

In summary, our contributions are the introduction of propagation models and the integration of beamforming maps in FastSLAM to achieve on-plate Simultaneous Localization and Mapping with high accuracy for robotic inspection.

II. Related Work

On the one hand, standard methods to inspect large metal structures consist in deploying a mobile robot to perform point-by-point thickness measurements with an acoustic probe, but the entire surface cannot be inspected in a reasonable amount of time due to the limited surface of the transducer. On the other hand, UGWs have been successfully used by SHM systems to inspect large structures such as pipelines or rails [4], [9], but the transducers are integrated into the structure and their position is known accurately. Hence, outside of the authors' works, UGWs-based techniques have not been deployed on a robotic system, nor have guided waves proven to yield accurate localization capabilities which are critical for such methods to work.

Moreover, UGWs propagation is dispersive, which means that the longer the distance a wave packet travels in a metal plate, the more it deforms. Fig. 2 illustrates this phenomenon. It shows that the shape of the signal is significantly different after propagating over two meters. In SHM, the chosen frequency range generally lies in a dispersion-limited bandwidth, but for our case-study, waves might propagate over much larger distances. Hence, wave dispersion may still have some effects on the signals, and shall not be neglected. In the literature, the use of propagation models in the context of localization and mapping on metal structures has not been thoroughly investigated. This work aims to answer this need.

In typical guided wave data, there are numerous echoes due to the multiple reflections on the plate edges and their number increases exponentially with the observation time. In addition, the wave packets overlap because of the wave dispersion. The consequence is that it is very challenging to recover individual wave-packets from the mixture data [10]. Therefore, most of the recent SHM techniques still rely only on the incident wave packet [3], [4], [11], [12]. For on-plate localization and mapping purposes, however, the retrieval of multiple echoes is essential, as they all provide range-only information to the edges. In the echo detection literature, time-delay estimation techniques have been successfully applied to ultrasound waves in the air [13], [14] but in a non-dispersive context. In [7], we used $\mathcal{L}_1$-regularized least squares to retrieve the multiple echoes without taking into account wave dispersion. Here, we rely on a wave propagation model to determine, through correlation with acoustic data, the likelihood of a reflection over a full range of distances to the transducers. In this new setup, the resolution of the difficult echo association problem is no longer required.

Recently, there have been attempts to infer a plate geometry from guided waves data [15]. Yet, non-dispersive propagation models are used, and the sensors are integrated into the structure. In robotics, the most similar problem is room shape reconstruction from acoustic echoes [5], [6]. However, they rely on sound waves propagating in the air without dispersion and do not consider the association problem to determinate the wall from which each echo originates. In [7], we rely on the most likely echo-line association but the overall algorithm is rather complicated due to the map management, and its robustness is limited. Here, from the likelihoods of reflection, we build beamforming maps to estimate the plate shape and limit ourselves to rectangular geometries (which are to be expected in our application). Then, these elements are integrated into a FastSLAM algorithm to achieve localization and mapping simultaneously.

In summary, we present a new method that efficiently integrates wave propagation models from the guided waves theory and beamforming maps in a FastSLAM algorithm to achieve more accurate on-plate localization and mapping results with less algorithmic complexity comparing to our previous method. The results obtained with experimental acoustic data from different metal plates support our claim.

III. Method

In this work, we are considering a mobile unit equipped with a co-localized emitter/receiver pair of transducers and moving on a metal surface. At the i^{th} scanning position, the emitter sends a pulse $s(t)$ to excite guided waves in the plate material, and the receiver collects the acoustic response $z_i(t)$ which contains the ultrasonic echoes. We intend to use these data and the robot odometry to recover accurately both the plate shape and the robot trajectory.

A. Measurement Model

Acoustic measurements essentially consist in a succession of the reflections of the excitation wave on the plate boundaries. As the small-sized corrosion patches we aim to detect with robotic inspection may not act as reflectors, their potential effect is neglected for the SLAM problem. Under the assumption that the material is isotropic, the propagation linear, and the reflections on the edges are orthogonal, a standard measurement model

to reverberation is the image source model [16]. It relies on the fact that each reflection from the plate boundaries can be considered as a signal originating from a fictional source, which is deduced from the real source position and the reverberant media geometry. In metal plates, the image source model can be leveraged to account for first order as well as higher order reflections, resulting in the following measurements:

$$z_i(t) = \sum_{\mathbf{x} \in \mathcal{I}(\mathbf{x}_i)} g(\mathbf{x}, \mathbf{x}_i, t) * s(t)$$

where $\mathbf{x}_i = [x_i, y_i]$ is the position of the robot during time step i, $\mathcal{I}(\mathbf{x}_i)$ the set of the image sources positions when the real source is in $\mathbf{x}_i$, $g(\mathbf{x}, \mathbf{x}_i, t)$ the acoustic response of the plate to an impulse being generated in $\mathbf{x}$ and received in $\mathbf{x}_i$, and $*$ denotes the convolution operation. In a non-dispersive media, the impulse response is simply given by $g(\mathbf{x}, \mathbf{x}_i, t) = \delta(t - \frac{||\mathbf{x}-\mathbf{x}_i||}{c})$, where δ denotes the Dirac distribution, and c is the constant propagation velocity. It results in waves propagating at a constant speed and without distortion. In a dispersive media like metal plates, a well-suited model of the propagation is given by the solutions of the Helmholtz equation [17]. For an ideal isotropic media, the impulse response is only a function of the propagation distance r between the (fictional) source and the receiver. Moreover, it is usually reduced, in the Fourier domain, to:

$$\hat{g}(r, \omega) \approx e^{-jk(\omega)r} / \sqrt{k(\omega) r}. \tag{1}$$

where $k(\omega)$ is the wavenumber of the major acoustic mode, and its non-linear dependency with respect to the pulsation ω is the typical characteristic of dispersive propagation. More details on how to determine this relation given prior information on the plate material can be found in the literature [4].

B. Correlation-Based Echo Detection

With the aim to retrieve the distances of the robot to the edges from data $z_i(t)$, we use the designed propagation model to estimate the likelihood that an orthogonal reflection occurred at a distance r. First, we consider the signal that would only contain such a reflection: $\hat{z}(r, t) = \hat{g}(2r, t) * s(t)$. Next, we build the correlation signal to assess the likelihood that this pattern is present within the measurement:

$$z_i'(r) = \frac{\langle z_i(t), \hat{z}(r, t) \rangle}{\sqrt{\langle z_i(t), z_i(t) \rangle \langle \hat{z}(r, t), \hat{z}(r, t) \rangle}} \tag{2}$$

where $\langle ., . \rangle$ denotes the scalar product in the domain of continuous signals: $\langle u(t), v(t) \rangle = \int_{-\infty}^{+\infty} u(\tau) v(\tau) d\tau$. As the resulting signal z_i' presents oscillations consistent with the wave spatial periodicity, it is more convenient to only work with its envelope that we will call $z_i(r)$ for simplicity (which shall not be mistaken with the temporal signal $z_i(t)$):

$$z_i(r) = |z_i'(r) + j\mathcal{H}(z_i')(r)| \tag{3}$$

where $\mathcal{H}$ denotes the Hilbert transform operator. Hence, the resulting signal z_i takes its values only between 0 and 1, and a higher value at r translates into a high likelihood that a reflection occurred at such a distance. In summary, by looking at the local maxima of $z_i(r)$, one can deduce the most likely reflections. Besides, it is noteworthy that a single measurement cannot provide enough information to determine an edge without ambiguity, as all the lines tangent to the circle with radius r and centered at the sensors position may equally account for the correlation measurement.

C. Map Estimation Via Beamforming

Similarly to our previous work, the map is represented by a set of lines: $\mathbf{M} = \{r_l, \theta_l\}_{l=1...4}$ where the parameters (r_l, θ_l) define the line equation in the 2D plane with:

$$x \cdot \cos\theta_l + y \cdot \sin\theta_l - r_l = 0$$

in a non-mobile frame with respect to the plate. Moreover, as we limit our case-study to rectangular shapes, the possible maps possess only four lines forming a rectangle altogether.

Let's assume a hypothetical robot trajectory $\{x_i, y_i\}_{i=1...T}$. We aim at estimating the map $\mathbf{M}$, which means establishing the probability density function $p(\mathbf{M}|x_{1..T}, y_{1..T}, z_{1...T})$. A first solution would consist in assessing, for each map in the 8-D domain, the correlation between the observations and the predicted data based on the image source model. However, such an approach would be far too cumbersome for a real-time application. Instead, we rely on a beamforming map. Such a map attributes, to every line parameters (r, θ), the likelihood of the line existence given the observations with:

$$\mathcal{L}_T(r, \theta) = \sum_{i=1}^{T} z_i(|x_i \cdot \cos\theta + y_i \cdot \sin\theta - r|).$$

where $d_i(r, \theta) = |x_i \cdot \cos\theta + y_i \cdot \sin\theta - r|$ is the distance between the robot during time-step i and the hypothetical line being considered. In the equation, all the correlation values add up constructively along all the observations if an edge is indeed present. Also, it can be noted that only first-order reflections are taken into account, as we reason on individual lines. One may consider that higher order reflections are less likely to account for high correlation amplitudes because of wave scattering after each additional reflection which causes loss of energy to the wave packet. Finally, to retrieve the most plausible map, we solve the following optimization problem:

$$\hat{\mathbf{M}} = \arg\max_{\mathbf{M}} \mathcal{L}_T(\mathbf{M}) = \arg\max_{\mathbf{M}} \sum_{l=1}^{4} \mathcal{L}_T(r_l, \theta_l)$$

where $\mathbf{M}$ is restricted to be a rectangle. It can be solved efficiently by taking that constraint into account. First, one can determine the most likely line:

$$(r_1, \theta_1) = \arg\max_{r, \theta} \mathcal{L}_T(r, \theta).$$

Next, it is possible to rely on the assumption that the retrieved line provides the most reliable estimation of the plate orientation w.r.t. the robot. Therefore, the determination of the other lines for $l = 2, 3, 4$ reduces to solving simple and independent one-dimensional optimization problems:

$$\theta_l = \theta_1 + \frac{\pi(l-1)}{2}\ ; \quad r_l = \arg\max_{r} \mathcal{L}_T(r, \theta_l).$$

D. Particle Evaluation and FastSLAM Algorithm

FastSLAM is a common approach to approximate Bayesian filters in the context of a SLAM problem. It relies on a particle

Algorithm 1: FastSLAM($\mathcal{P}_{T-1}, \mathbf{u}_{T-1}, z_T(r)$).

Data: Particle set $\mathcal{P}_{T-1}$, odometry data $\mathbf{u}_{T-1}$ and correlation measurement $z_T(r)$.
Result: Particle set $\mathcal{P}_T$ for the current time step T.
if *T=0* **then**
  Initialize the particle filter with $\mathcal{P}_0 = \{[x_0, y_0, \alpha_0], \text{null-fuction}\}_{n=1\ldots N}$
else
  for $n = 1\ldots N$ **do**
    $\mathbf{X}_T^{(n)} \sim p\left(\mathbf{X}_T | \mathbf{X}_{T-1}^{(n)}, \mathbf{u}_{T-1}\right)$;
    $\mathcal{L}_T^{(n)}(r,\theta) = \mathcal{L}_{T-1}^{(n)}(r,\theta) + z_T(|x_T^{(n)}\cos\theta + y_T^{(n)}\sin\theta - r|)$;
    $\mathbf{M}_T^{(n)} = \arg\max_{\mathbf{M}} \mathcal{L}_T^{(n)}(\mathbf{M})$;
    $w_T^{(n)} \propto \exp\left\{\beta \sum_{(r_l,\theta_l)\in\mathbf{M}_T^{(n)}} z_T\left(d_T^{(n)}(r_l,\theta_l)\right)\right\}$
  end
  Construct $\mathcal{P}_T$ by sampling each particle proportionally to their respective weight.
end
return $\mathcal{P}_T$.

filter in the localization space, where each particle holds a hypothesis on the map which is inferred from the particle trajectory and the measurements. During time step T, a set with N particles has the following form:

$$\mathcal{P}_T = \left\{\mathbf{X}_T^{(n)} = \{x_i^{(n)}, y_i^{(n)}, \alpha_i^{(n)}\}_{i=1\ldots T}, \mathcal{L}_T^{(n)}\right\}_{n=1\ldots N}$$

where $\mathbf{X}_T^{(n)}$ represents the n-th particle belief on the robot trajectory augmented with its heading over time steps $i = 1\ldots T$, and $\mathcal{L}_T^{(n)}$ its beamforming map which depends on the trajectory. Moreover, each particle is provided with a weight indicating how the particle belief accounts for the measurements. To define it, we rely on the current correlation measurement and assess the likelihoods of the map edges retrieved from $\mathcal{L}_T^{(n)}$ and the current robot position belief:

$$w_T^{(n)} = \eta \cdot \exp\left\{\beta \sum_{(r_l,\theta_l)\in\mathbf{M}_T^{(n)}} z_T\left(d_T^{(n)}(r_l,\theta_l)\right)\right\} \quad (4)$$

where η is the normalization factor and β a positive parameter. It enables to fix the confidence in the correlation measurements and shall be tuned so that the resulting weight distribution is consistent with the motion and observation noises. The weights are used to sample, with replacement, the particles after each time step. Besides, one may note that we are not considering, in (4), the uncertainty on the lines retrieval from the beamforming maps for simplicity. Altogether, the implementation of FastSLAM is given in Algorithm 1.

IV. Results

In this part, we test our FastSLAM approach on experimental data. We detail the experimental setup and show the results in terms of localization and mapping accuracy.

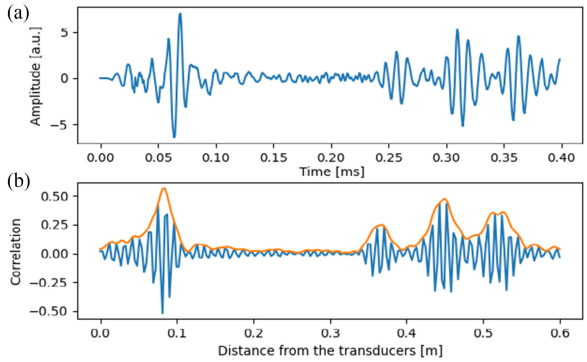


Fig. 3. Illustration of the echo detection principle based on correlation with a propagation model. a) represents the acoustic measurement. b) shows the correlation signal (blue) and its envelope (orange).

A. Experimental Setup

In order to assess the efficiency of our procedure, we use an emitter-receiver pair of transducers on two different metal plates. The first plate has dimensions 600 × 450 × 6mm, is in aluminium, and has small artificial holes on it. The second plate has dimensions 1700 × 1000 × 6 mm and is in steel. The acoustic data for the plate 1 have been already presented in [7] and will serve as a way to demonstrate the improvement of the procedure. The acquisition process is globally the same to collect the data on the second plate: the transducer pair is moved by hand on the vertices of a regular grid. At every position, 10 measurements of the ultrasonic response are averaged to improve the signal quality. This operation is not critical in a laboratory environment, but it may be necessary in outdoor conditions, where the inspection robot shall operate, to alleviate the effect of external disturbances. The acquisition positions are also carefully recorded. In total, 108 measurements are collected on the plate 1, while this number increases to 117 for plate 2. We use two tonebursts of a sinusoidal wave at 100 kHz as the excitation. Moreover, the direct incident signal is smoothly removed from the data as it does not correspond to a reflection on an edge.

For each plate, we determine a wave propagation model as in eq. (1) and use $N = 20$ particles. To simulate a sweep of a plate by a robotic crawler, a sequence of measurements is selected from the database and is presented to the SLAM framework, with the theoretic displacement between grid cells used as odometry. Also, we add Gaussian noise on the odometry data: $\bar{\Delta r} \sim \mathcal{N}(\Delta r, (10^{-2}\Delta r + \Delta r_0)^2)$ and $\bar{\Delta\theta} \sim \mathcal{N}(\Delta\theta, (10^{-2}\Delta\theta + \Delta\theta_0)^2)$ with $\Delta r_0 = 10^{-3}$ m and $\Delta\theta_0 = 10^{-2}$ rad to simulate odometry uncertainty which may be limited due to the robot magnetic adherence and embedded accelerometers used to provide precise heading on a nearly-vertical structure, in a realistic scenario [7].

B. Echo Detection

First, we illustrate the echo-detection principle. We show, in Fig. 3.a), the measured acoustic signal for a position corresponding to 8 cm to the edges, in a corner of plate 1. On b), we show the resulting correlation signal computed using eq. (2) and its envelope calculated with eq. (3), yielding the signal which is

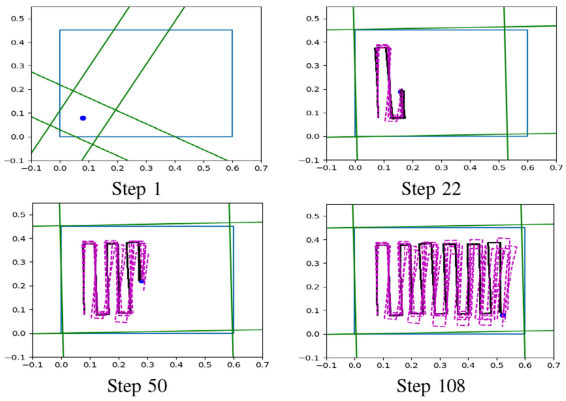


Fig. 4. Trajectories estimated by all the particles (black lines), dead-reckoning trajectories (dash magenta lines) and map retrieved by the most likely particle (green lines) during Steps 1, 22, 50 and 108 for a lawn-mower path on plate 1 (zoom for details). The true outline of the plate and true sensor positions correspond to the blue rectangle and blue dot respectively.

fed to the FastSLAM algorithm. It can be seen on b) that we manage to retrieve, from the local maxima, all the distances where first-order reflections occurred which are 8, 37 and 52 cm. The echo detected at nearly 45 cm corresponds to a higher-order reflection, but still has an amplitude that is comparable to that of the first-order wave packets. The existence of such a reflection is not assumed by the algorithm. Hence, we will determine a posteriori if their presence has a detrimental effect on the results.

C. Localization and Mapping Results

We run our FastSLAM algorithm using the data of plate 1, and simulate a lawn-mower path. Although the results are generated off-line, our method can run online on a real robotic platform. Indeed, as the beamforming maps of size $Z \times Z$ are updated incrementally, the complexity of one FastSLAM iteration with N particles is $\mathcal{O}(N \times Z^2)$, which leads to a computational time of a few tens of milliseconds per iteration in our setup, with $Z = 300$ and $N = 20$.

In Fig. 4, we show the particles' belief on the sensors trajectory during measurement steps 1, 22, 50 and 108. We also represent the map retrieved by the particle with the highest weight and several dead-reckoning trajectories obtained using noisy odometry input only. During Step 1, the map is not correctly estimated. As only one measurement has been integrated, the distance to the closest edge can be recovered but, the orientation is essentially random. Rapidly, the three closer edges are recovered as shown during Step 22. However, the right edge is not yet well estimated as it is further away. During Step 50, the plate shape is fully recovered, and during the final step, both the estimation of the plate shape and trajectory are accurate. In contrast, the dead-reckoning trajectories present noticeable drift. This illustrates that, by relying on the acoustic data, the proposed approach can appropriately compensate for moderate odometry noise.

Fig. 5 depicts the beamforming map for the most likely particle during the final step. We can see that the intensity peaks due to the edges are clearly visible, and our optimization method does not face difficulty to retrieve them.

To compare our new FastSLAM approach with the previous one, we show, in Fig. 6, the average localization and line parameters estimation errors calculated over 100 runs of each

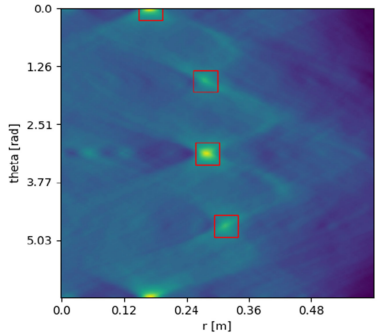


Fig. 5. Beamforming map for the particle with the highest weight during the final step. The rectangles indicate the edges retrieved with our method.

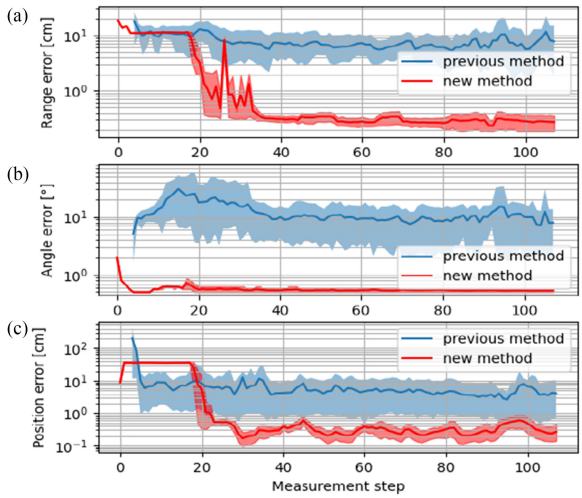


Fig. 6. Localization and mapping results over 100 repetitions of a lawn-mower path on plate 1 for the previous and the new method. a) Average estimation errors on the range parameter of the lines. b) Average estimation errors on the angle parameter. c) Average localization errors in the estimated plate frame. The 10% and 90% quantiles correspond to the upper and lower bounds of the coloured areas. The scales along the y-axis are logarithmic.

algorithm, and using the same acoustic data on plate 1. We simulated 100 repetitions of the lawn-mower path for the sensors trajectory. In the figure, we represent the 10% and 90% quantiles with the aim to measure the repeatability of each approach. It can be observed that, with our new method, only a few tens of measurement steps are necessary to recover, in average, the range parameters of the lines with a precision of a few millimeters, and the plate orientation with a precision better than one degree. The localization result is also very precise as, after a quick convergence, the position errors remain in the order of a few millimeters despite the defects on the plate. Besides, the estimation is not subject to randomness as the 10% and 90% quantiles remain close to the average results. In comparison, our previous method demonstrates poorer results. Indeed, not only are the estimation errors higher, but also the variation of precision can be relatively significant between two runs. Altogether, the results illustrate the improvement of localization and mapping that is achieved by our new method.

With the aim to assess the results for a larger plate, we run our algorithm with the measurements obtained on plate 2, and simulate again a lawn-mower path. The results obtained over 100 runs are provided in Fig. 7. On this plate, the echo

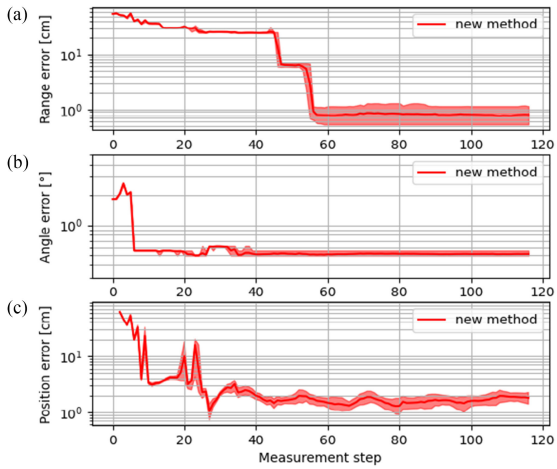


Fig. 7. Localization and mapping results over 100 repetitions of a lawn-mower path on plate 2 for the new method. a) Average estimation errors on the range parameter of the lines. b) Average estimation error on the angle parameter. c) Average localization errors in the estimated plate frame. The 10% and 90% quantiles correspond to the upper and lower bounds of the coloured areas. The scales along the y-axis are logarithmic.

Scenario	Range error [mm]	Angle error [degree]
Scenario 1	3.007 ± 0.098	0.234 ± 0.0004
Scenario 2	10.766 ± 22.921	0.206 ± 0.134

Fig. 8. Average estimation errors and standard deviations on the lines parameters obtained during the last measurement step for the two scenarios in consideration. The errors are evaluated using 100 repetitions.

detection employed by our previous method is not efficient, as it does not consider the wave dispersion effect, whereas the propagation distances are larger. This induces large misdetection rates and poor results. Hence, we display only the results of our new approach. Despite the slower convergence caused by the larger surface, and the slightly higher localization error, our method still provides precise estimates of the trajectory and plate geometry. This result indicates that our approach still works on surfaces sufficiently large to be used for realistic applications. The underlying prerequisites are a wave propagation model and filter parameters that conveniently fit the acoustic measurements and on-the-field noisy conditions. Naturally, one may also expect longer convergence times when the plate surface increases, as the echo detection is expected to be efficient mostly for short ranges as shown in Fig. 7.a).

As a final evaluation, we determine the average mapping errors and standard deviations over 100 runs obtained during the final step for a lawn-mower path (Scenario 1) and a random walk (Scenario 2) on plate 1. Fig. 8 presents the results. It can be noticed that the overall results are relatively poorer for the random walk. This illustrates that the estimation accuracy also strongly depends on the robot path which shall be optimized for optimal reconstruction.

V. Conclusion

We have designed a new FastSLAM approach to achieve Simultaneous Localization and Mapping on metal plates by relying on ultrasonic guided waves. Comparing to our previous work, this method relies on wave propagation models and beamforming maps. Experiments carried on an undamaged and a damaged metal plate in a laboratory environment demonstrate that this new approach achieves better results in terms of accuracy and robustness with less algorithmic complexity. In future works, this method shall be adapted and tested in more realistic scenarios. Indeed, on a large metal structure in outdoor environments, more complex and noisy signals are expected due, for example, to inferior surface quality, to the presence of anti-fouling coating on the plates, to more complex plate geometries, or due to wave scattering caused by the welds which fix the different plates altogether. Furthermore, adaptive techniques shall be investigated to adjust the propagation model and filter parameters which may no longer be assumed known a priori. Also, a real robotic platform shall be used, and active-sensing strategies shall be investigated to recover the plate geometry efficiently.

References

[1] M. Montemerlo, S. Thrun, D. Koller, B. Wegbreit, "Fastslam: A factored solution to the simultaneous localization and mapping problem," in *Proc. 18th Nat. Conf. Artif. Intelli.*, 2002, pp. 593–598.
[2] W. Cailly and H. Walaszek, "Three dimensional ultrasonic imaging of mechanical components by inversion," in *7th edition of the International Symposium on AirCraft Materials*, 2018. Available: https://www.researchgate.net/publication/331985263_Three_dimensional_ultrasonic_imaging_of_mechanical_components_by_inversion_7_th_INTERNATIONAL_SYMPOSIUM_ON_AIRCRAFT_MATERIALS-Compiegne_France_2_Contents
[3] P. Huthwaite and F. Simonetti, "High-resolution guided wave tomography," Wave Motion, 2013.
[4] Z. Su and L. Ye, *Identification of Damage Using Lamb Waves: From Fundamentals to Applications*, 01 2009, vol. 48.
[5] M. Krekovic, I. Dokmanic, and M. Vetterli, "Echoslam: Simultaneous localization and mapping with acoustic echoes," in *Proc. IEEE Int. Conf. Acoustics, Speech Signal Process.*, 2016, pp. 11–15.
[6] F. Peng, T. Wang, and B. Chen, "Room shape reconstruction with a single mobile acoustic sensor," in *Proc. IEEE Glob. Conf. Signal Inf. Process.*, 2015, pp. 1116–1120.
[7] C. Pradalier, O.-L. Ouabi, P. Pomarede, and J. Steckel, "*On-Plate localization and mapping for an inspection robot using ultrasonic guided waves: A proof of concept*," in *Proc. Int. Conf. Intelli. Robot. Syst.*, 2020, pp. 5045–5050.
[8] B. D. Van Veen and K. M. Buckley, "Beamforming: A versatile approach to spatial filtering," *IEEE ASSP Mag.*, vol. 5, no. 2, pp. 4–24, Apr. 1988.
[9] P. Cawley and D. Alleyne, "The use of lamb waves for the long range inspection of large structures," *Ultrasonics*, 1996.
[10] Z. Su, L. Ye, and Y. Lu, "Guided lamb waves for identification of damage in composite structures: A review," *J. Sound Vib.*, vol. 295, no. 3, pp. 753–780, 2006.
[11] M. Zhao, W. Zhou, Y. Huang, and H. Li, "Sparse bayesian learning approach for propagation distance recognition and damage localization in plate-like structures using guided waves," *Struct. Health Monit.*, vol. 20, no. 1, Art. no. 1475921720902277.
[12] Q. Jianxi, F. Li, S. Abbas, and Y. Zhu, "A baseline-free damage detection approach based on distance compensation of guided waves," *J. Low Freq. Noise, Vib. Act. Control*, 2018.
[13] J. Steckel and H. Peremans, "Sparse decomposition of in-air sonar images for object localization," in *Proc. IEEE Sensors*, pp. 1356–1359, 2014.
[14] B. Fontaine and H. Peremans, "Determining biosonar images using sparse representations." *The J. Acoust. Soc. Amer.*, vol. 125, no. 5, pp. 3052–9, May 2009.
[15] E. Hong and C. Schaal, "Reverse engineering stiffened plates using guided wave-based nondestructive testing methods ," in *Health Monit. Struct. Biol. Syst. XII, T. Kundu*, Ed., *International Society for Optics and Photonics. SPIE*, 2018.
[16] H. Kuttruff, *Room Acoustics*, fourth edition, 2000.
[17] N. Quaegebeur, P. Masson, D. Langlois Demers, and P. Micheau, "Dispersion-based imaging for structural health monitoring using sparse and compact arrays," *Smart Mater. Structures*, 2011.

Mid-Air Range-Visual-Inertial Estimator Initialization for Micro Air Vehicles

Martin Scheiber[1], Jeff Delaune[2], Stephan Weiss[1], and Roland Brockers[2]

***Abstract*— Monocular Visual-Inertial Odometry (VIO) has become ubiquitous for navigation of autonomous Micro Air Vehicles (MAVs). Yet, state-of-the-art VIO is still very failure-prone, which can have dramatic consequences. To prevent this, VIO must be able to re-initialize in mid-air, either during a free fall or on a constant velocity trajectory after attitude control has been re-established. However, for both of these trajectories, the visual scale cannot be observed with VIO batch initializers because of the absence of acceleration change. We propose to use a small and lightweight laser-range finder (LRF) and a scene facet model to initialize vision-based navigation at the right scale under any motion condition and over any scene structure. This new range constraint is integrated into a visual-inertial bundle-adjustment initializer. We evaluate our approach in simulation, including robustness to various parameters, and demonstrate on real data how this approach can address mid-air state estimation failure in real-time.**

I. Introduction

Autonomous, safe, and robust navigation is crucial for a micro air vehicle (MAV). In-flight pose estimation must provide accurate and robust poses for flight controllers to perform ever more complex maneuvers. Many different approaches exist, ranging from multi-sensor to minimal-sensor set state estimation. Although these approaches differ, their common ground is the need for an initial state.

Especially minimum sensor suite approaches, i.e., visual-inertial odometry (VIO) algorithms, are constrained on their estimator initialization. Most state-of-the-art VIO rely on a specific scenario or motion to start their estimator correctly. However, this limits the level of MAV autonomy since the scenario or motion might be unknown when (re-)initializing. Particularly, fully-autonomous systems should be able to initialize in all airborne scenarios, which are

(a) excitation motion,
(b) constant velocity motion, including hovering (no motion), and
(c) free-fall motion.

Excitation motions are perfect for initialization, and nearly all state-of-the-art VIO algorithms rely on excitation in their initialization technique. Similarly, filter-based algorithms can cover hover or static initialization. These scenarios also refer to the most common initialization motion, especially when performing manual or velocity-control-based takeoff. Nevertheless, constant velocity and free-fall initialization can occur in mid-air deployment or mid-air recovery scenarios. However, traditional VIO frameworks cannot handle these initialization trajectories. Hence, additional environment information is needed to provide a full state visual-inertial initialization for motion (b) or (c), removing the autonomy of such approaches.

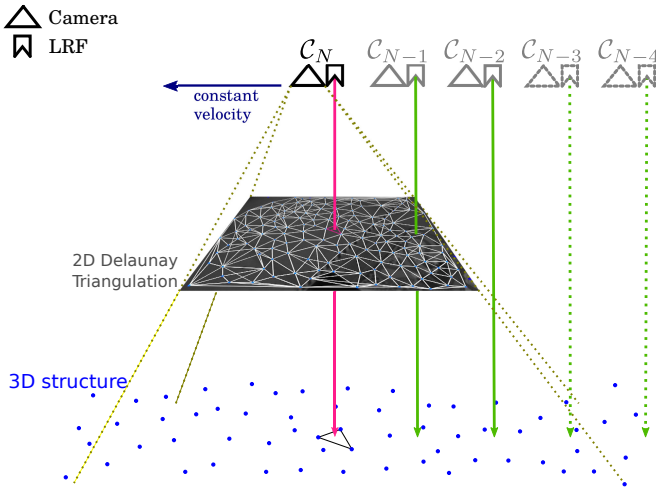


Fig. 1. Illustration of the proposed mid-air initialization algorithm for constant velocity flights. The 3D-structured scene is captured by a downward looking camera. With its generated images, features can be triangulated and divided into subgroups of triangles. Then a laser-range finder (LRF) can be used to metrically scale the Delaunay triangulated structure and camera poses in a non-linear optimization. Further, in combination with an IMU, the full MAV navigation states can be recovered in a linear way, providing a full onboard initialization. Please note that the environment is not assumed to be planar (i.e., it can be structured).

Therefore, this work aims to provide an initialization algorithm that is

- **Motion independent**: Our proposed framework can initialize in any non-zero motion, regardless of being excitation, constant velocity (as illustrated in Fig. 1), or free-fall motions.
- **Computationally fast**: Analysis of our proposed approach showed it is able to run in real-time onboard an embedded platform to provide fast initialization under time-limited motions (e.g., free-fall).
- **Free of prior knowledge**: Typically, initializers take advantage of prior knowledge, e.g., planar ground, height, level attitude, or similar. Our proposed approach works without any prior information on the motion or environment.

This work is structured as follows: Sec. II will examine state-of-the-art initialization techniques, their limitations in the in-flight reference scenarios, and why range measurement can lift these limitations. Sec. III presents our initialization

[1]These authors are with Faculty of Intelligent System Technologies, Group Control of Networked Systems, Universität Klagenfurt, Klagenfurt, Austria {martin.scheiber, stephan.weiss}@ieee.org
[2]These authors are with the Jet Propulsion Laboratory, California Institute of Technology, Pasadena, California, USA {jeff.h.delaune, roland.brockers}@jpl.nasa.gov

algorithm, that can initialize in any mid-air scenario. Sec. IV takes a closer look at the influence of noisy measurements on our proposed algorithm, and Sec. V presents and discusses results on real-world constant flight experiment, as depicted in Fig. 2.

II. Related Work

A. Visual-Inertial Odometry

State-of-the-art visual-inertial state estimation frameworks comprise many different methods and algorithms. Nevertheless, such frameworks are usually grouped into two main algorithm categories: filter and optimization-based [1].

Filter approaches typically represented with a variant of extended Kalman filter (EKF) [2]. Filter based visual-inertial estimators are able to quickly propagate the state and its covariance and provide information needed for flight control using high-frequency information from the inertial measurement unit (IMU). With the IMU typically modeled as input for the system dynamics and therefore generating growing uncertainties over time, a camera sensor can provide a pose update to correct eventual drift and to decrease the uncertainty. Filter approaches shine by their ability to efficiently retain past information through marginalization implicitly in the error covariance matrix, allowing estimations without the need for time-consuming iterative optimizations. Filter-based frameworks can be divided into tightly and loosely coupled estimators [3]. Loosely coupled estimators [4], [5] perform the visual pose calculation independently from the state update and include a metric scale in their state definition. In comparison, tightly coupled systems use the include the tracked features directly in their dynamics to update and correct the state [6], [7], [8], [9], [10].

Non-linear optimization-based algorithms iteratively perform a least-square approach to converge to a state estimate [3]. The most commonly used optimization is the bundle-adjustment (BA) that minimizes the re-projection error of tracked features. The BA can be used for vison-only systems such as ORB-SLAM [11], SVO [12], or fused with inertial measurements as the *Robust and Versatile Monocular Visual-Inertial State Estimator* (VINS-Mono) [13] or *Open Keyframe-based Visual-Inertial SLAM* (OKVIS) [14] showed. Their advantage is that they can approach with sufficient iterations they can achieve better estimation quality. However, they require translation to triangulate a map. Further, they are computationally more costly since they optimize over past measurements. This problem has been mitigated in recent years as onboard processing power has increased, and through marginalization of past information.

As both nonlinear filtering and optimization approaches find the local optima, they are dependent on an accurate initialization of the state vector in the vicinity of the global optima.

B. Initialization

Robustness and performance of both filter- and optimization-based algorithms depend on the quality of the initialization routine. The former require an initial pose and velocity state estimates, which can be zero motion

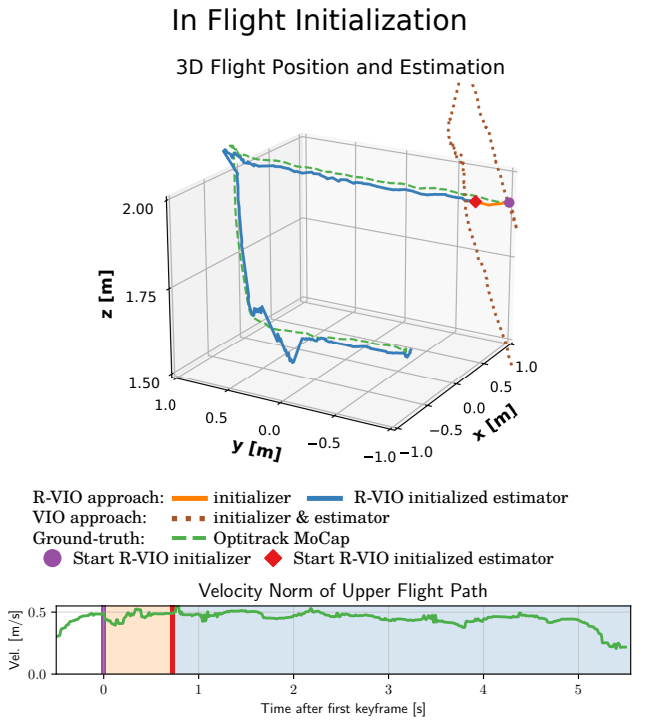


Fig. 2. Real-world experiment for initializing the proposed framework under a constant velocity flight. The initialization is triggered at (pink point), and the next 10 image frames (i.e., 0.33 s) are taken for the initialization window (orange estimates). After computing the initial navigation states (after approx. 0.75 s), the estimator VINS-Mono is initialized (red point) and continues with a visual-inertial navigation (blue estimates). The norm of the velocity throughout the initialization phase, computed with the position derivatives from the motion capturing system, are shown in the lower plot.

(assuming MAV starting on the ground before take-off). This estimate has to be relatively close to the actual value in order for the filter to converge. On the other hand, optimization-based approaches need an initial map and visual scale.

As an example of mid-air self-initialization without particular excitation motions, several studies have been presented that address the throw-and-go (TnG) problem under specific assumptions: [15] used height assumption to provide an initial estimate to their filter-based estimator, whereas [16] required an attitude estimation before the fall, flat ground surface, and horizontal translation to triangulate the initial structure and derive the metric scale for their optimization-based estimator. Further, in our previous work [17], we managed to initialize in a free-fall by aligning the magnitude of visual acceleration to the magnitude of gravity.

These free-fall initialization approaches are limited to that exact scenario and prior knowledge or assumptions and cannot be applied to horizontal motion at constant velocities. Nevertheless, IMU-pre-integration [18] can provide an opportunity to unify the mid-air self-initialization approaches in one framework and remove pre-initialization assumptions. E.g., methods with visual-inertial optimization in initialization, such as VINS-Mono, OKVIS, or OrbSLAM3 [19], rely on the IMU pre-integration to generalize their initialization algorithm to all visual-inertial observable motions.

We selected VINS-Mono as state-of-the-art algorithm to compare our approach against because of both maturity

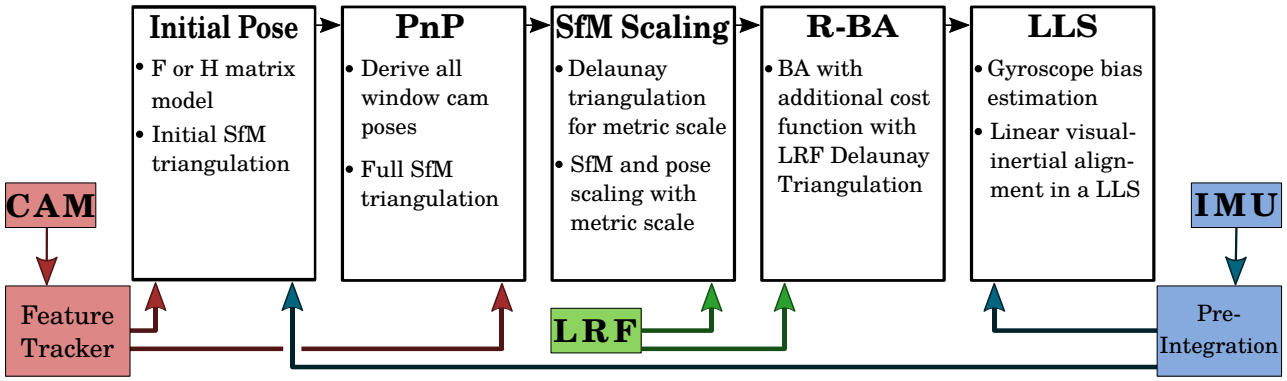


Fig. 3. The proposed Range-Visual-Inertial initialization framework. Images are used to derive the initial camera poses using the Fundamental or Homography matrix method (Sec. III-A). First the scene structure from motion (SfM) is derived using a perspective-n-point (PnP) approach (Sec. III-B). Second, this structure is scaled metrically with the range measurements received by the LRF (Sec. III-C). Then, to reduce the impact of measurement noise a range-visual bundle-adjustment (R-BA) is performed (Sec. III-D). Finally, the range-visual poses are aligned with the pre-integrated IMU measurements, to derive the globally aligned states (Sec. III-E).

and repeated good initialization performance in different scenarios. Taking a closer look on VINS-Mono's four step initialization algorithm [20], this approach first estimates the initial pose and structure using camera trigonometry, given a initialization window of N keyframes. Then a perspective-n-point (PnP) is performed to derive all other keyframe camera poses in the window and triangulate all remaining matches to form a complete structure. This structure and camera poses are then used in a visual BA to minimize measurement noise and triangulation errors, and improve the estimated poses. Given the first keyframe set as visual camera coordinate frame $\mathcal{C}$, and given the body (or IMU) coordinate frames $k = \mathcal{B}_k$ for each image at time t_k, all initialization window position and rotations, ${}^{\mathcal{C}}\mathbf{p}_k$ and ${}^{\mathcal{C}}\mathbf{R}_k$, are derived in the BA. At the last step, VINS-Mono performs a linear least-square (LLS) to linearly align these visual with the inertial IMU measurements. The latter are pre-integrated to derive the frame-to-frame position and velocity, ${}^{k}\hat{\boldsymbol{\alpha}}_{k+1}$ and ${}^{k}\hat{\boldsymbol{\beta}}_{k+1}$, respectively. Equ. (1) describes the LLS that solves for the remaining state vector ${}^{k}\hat{\mathbf{x}}_{k+N} = \left[{}^{k}\hat{\mathbf{v}}_{k+1}^{\mathsf{T}}, \quad \dots, \quad {}^{k+N-1}\hat{\mathbf{v}}_{k+N}^{\mathsf{T}}, \quad {}^{\mathcal{C}}\hat{\mathbf{g}}^{\mathsf{T}}, \quad \lambda\right]^{\mathsf{T}}$ containing the camera velocities expressed in the body frame, gravity vector expressed in the initial camera frame ${}^{\mathcal{C}}\hat{\mathbf{g}}$, and metric scale λ. Further, δt_k is the frame-to-frame time difference, $\Delta^{\mathcal{C}}\mathbf{p}_k = {}^{\mathcal{C}}\mathbf{p}_{k+1} - {}^{\mathcal{C}}\mathbf{p}_k$ the frame-to-frame position difference from the BA, and ${}^{k}\mathbf{R}_{k+1}$ the body frame-to-frame rotation derived from IMU pre-integration.

$$ {}^{k}\hat{\mathbf{x}}_{k+N} = \left({}^{k}\mathbf{H}_{k+N}^{\mathsf{T}}\,{}^{k}\mathbf{H}_{k+N}\right)^{-1} \cdot {}^{k}\mathbf{H}_{k+N}^{\mathsf{T}} \cdot {}^{k}\mathbf{z}_{k+N} \tag{1} $$

with the frame-to-frame measurement matrix and vector

$$ {}^{k}\mathbf{z}_{k+1} = \begin{bmatrix} {}^{k}\hat{\boldsymbol{\alpha}}_{k+1} - {}^{\mathcal{B}}\mathbf{p}_{\mathcal{C}} + {}^{k}\mathbf{R}_{k+1}\,{}^{\mathcal{B}}\mathbf{p}_{\mathcal{C}} \\ {}^{k}\hat{\boldsymbol{\beta}}_{k+1} \end{bmatrix} \tag{2} $$

$$ {}^{k}\mathbf{H}_{k+1} = \begin{bmatrix} -\mathbf{I}_3\,\delta t_k & \mathbf{0}_3 & \frac{1}{2}\,{}^{k}\mathbf{R}_{\mathcal{C}}\,\delta t_k^2 & {}^{k}\mathbf{R}_{\mathcal{C}}\,\Delta^{\mathcal{C}}\mathbf{p}_k \\ -\mathbf{I}_3 & {}^{k}\mathbf{R}_{k+1} & {}^{k}\mathbf{R}_{\mathcal{C}}\,\delta t_k & \mathbf{0}_3 \end{bmatrix} \tag{3} $$

However, this final step already shows the sensor limitations of this visual-inertial algorithm using a IMU pre-integration and visual optimization method. First, one can show [21] that under constant velocity motions, the Grammian of the measurement matrix ${}^{k}\mathbf{H}_{k+N}$ is 0. Hence the matrix ${}^{k}\mathbf{H}_{k+N}^{\mathsf{T}}\,{}^{k}\mathbf{H}_{k+N}$ is singular and the LLS not solvable [22]. Similarly, in a free-fall motion, this linear formulation yields to the measurement vector ${}^{k}\mathbf{z}_{k+N}$ being $\mathbf{0}$. As a result, the estimation of the LLS Equ. (1) can only yield a state estimate of ${}^{k}\hat{\mathbf{x}}_{k+N} = \mathbf{0}$, which differs from the ground truth. Hence, in our work's two given reference scenarios, the visual-inertial approach cannot yield a correct initialization. This also corresponds to previous work performed on visual-inertial closed-form solution [23] and visual-inertial navigation system (VINS) [24] unobservability analysis. For this reason, and to the best of our knowledge, there are no previous works attempting to initialize a VINS system in a constant velocity flight. Therefore, in the next section, we will present a range-visual-inertial approach that keeps this computationally efficient structure and can mitigate the visual-inertial unobservable motions.

III. Range-Visual-Inertial Initialization

Given VIO unobservability issues discussed in the previous section, we present a new algorithm extending the visual-inertial initialization with a range sensor. In previous work [25], we already showed the improvements range measurements can bring to a visual-inertial filter framework. With our current approach, we extend the VINS-Mono with the ranged facet constraints. Therefore, we keep the general structure of VINS-Mono' initialization algorithm and extend it with the additional range sensor, which accounts for the new scene distance information, to a five-step algorithm as shown in Fig. 3.

A. Keyframe Selection and Initial Structure

The keyframes are selected based on a baseline criterion of [26]. If the baseline after accounting for rotation between the current image and the last keyframe exceeds a threshold th_b, the current frame is selected as the next keyframe. Further, feature tracking takes place on a frame-to-frame basis with consistent tracks developed as new frames appear.

Initially, a structure from motion (SfM) is created using the newest and oldest keyframes that exceed a baseline threshold th_b. This threshold is needed to account for hover-like motions. Then using the pose recovery criterion provided by

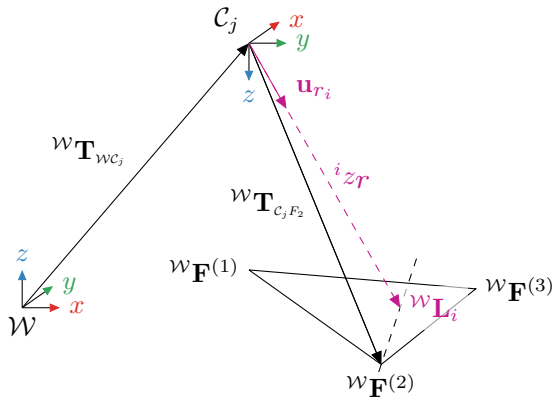


Fig. 4. The plane spanned by a Delaunay triangle which the LRF measurement intersects $\{{}^{\mathcal{W}}\mathbf{F}^{(1)}, {}^{\mathcal{W}}\mathbf{F}^{(2)}, {}^{\mathcal{W}}\mathbf{F}^{(3)}\}$ is used to to derive the estimated range ${}^{i}\hat{z}_{\boldsymbol{r}}$ from the SfM. This estimate is then compared to the LRF distance measurement ${}^{i}\breve{z}_{\boldsymbol{r}}$ to derive the metric scale for the structure and camera poses.

OrbSlam [11], the initial transformation is derived using the Fundamental or Homography matrix in the 5-point or DLT algorithm, respectively. This provides more flexibility for initialization scenarios, as it accounts for planar or structured environments. This differentiation is especially needed for downward-looking cameras, since their field-of-view more likely covers only the ground plane when flying at a low altitude.

B. Full Structure and Camera Poses

The other $N-2$ camera poses are derived using a PnP approach. First, all transforms from the initial camera pose $\mathcal{C}_k$ to all other camera poses $\mathcal{C}_j, 0 < j < N-1, j \neq k$ are derived in a *forward*-PnP. Further, any missing feature matches are triangulated. To decrease the transform calculation error between camera frames with a large baseline, a similar *backward*-PnP is performed. As a next step the newest camera pose $\mathcal{C}_k$ and all other camera poses $\mathcal{C}_j, N-1 > j > 0, j \neq k$ are used. Again all previously untriangulated feature matches between two image frames are triangulated. This vice-versa PnP is chosen for two reasons: First, this increases the number of triangulated features in the structure, which increases the amount of information available in the later bundle-adjustment stage. Second, the image overlap between the initial keyframe k and any other keyframe cannot be guaranteed. This approach tries to mitigate this issue by using the newest frame N for the transform calculation.

C. Structure Scaling

Camera only triangulation suffers from scale ambiguity. Therefore, an additional sensor is needed to scale the resulting structure of the previous step metrically. In most scenarios, the onboard IMU provides sufficient information to do so. However, in the given reference scenarios, an IMU will not yield enough metric scale information. Therefore, an additional sensor, the laser-range finder (LRF), is added to the system to provide single distance measurements at the camera rate. This range is then used to scale the structure initially.

This scaling approach follows the one proposed by Ref. [26], which models the surface structure and the range estimate as a function of the current states and measurement. However, at this point in the initialization, no state estimates are available. Therefore, only the raw, scalar distance measurements ${}^{i}\breve{z}_{\boldsymbol{r}}$ are used.

$$\begin{aligned} {}^{i}\hat{z}_{\boldsymbol{r}} &= {}^{i}\hat{z}_{\boldsymbol{r}} \cdot \frac{\mathbf{u}_{r_i}^{\mathsf{T}} \cdot \mathbf{n}}{\mathbf{u}_{r_i}^{\mathsf{T}} \cdot \mathbf{n}} \\ &= \frac{\left({}^{\mathcal{W}}\mathbf{p}_{\mathcal{C}F_2} - {}^{\mathcal{W}}\mathbf{p}_{\mathcal{C}_i}\right)^{\mathsf{T}} \cdot \mathbf{n}}{\mathbf{u}_{r_i}^{\mathsf{T}} \cdot \mathbf{n}} \end{aligned} \tag{4}$$

with

$$\mathbf{n} = \left({}^{\mathcal{W}}\mathbf{p}_{\mathcal{C}F_1} - {}^{\mathcal{W}}\mathbf{p}_{\mathcal{C}F_2}\right) \times \left({}^{\mathcal{W}}\mathbf{p}_{\mathcal{C}F_3} - {}^{\mathcal{W}}\mathbf{p}_{\mathcal{C}F_2}\right) \tag{5}$$

All tracked features from the initial triangulation frames are grouped in triangles using the Delaunay triangulation [27]. The triple of features in which the range measurement falls is selected, and its range is derived in camera frame using Equ. (4), with a visual representation shown in Fig. 4. This approach assumes a local flatness of the plane spanned by the selected triangle, an assumption that holds given enough tracked features.

Then in Equ. (6) the derived plane depth is compared to the range measurement to derive the metric scale s. This scale is then used to scale the camera poses and resulting structure metrically.

$$s = \frac{{}^{i}\hat{z}_{\boldsymbol{r}}}{{}^{i}\breve{z}_{\boldsymbol{r}}} \tag{6}$$

Please note that this derived scale is subject to the range sensor's measurement noise, feature tracker implementation, and violation of the triangle plane real-world flatness. Hence the derived scale might be error-prone. Consequently, the next step performs a range-visual optimization to minimize this initial scale error.

Further, one could argue that this scaling step can be performed before the PnP. However we chose to do this after the PnP for two reasons: First, the initial structure (A) is error prone and is minorly optimized through the PnP (B). Secondly, simulation analysis showed that scaling the structure before the R-BA (D) yields best initialization results overall.

D. Range-Visual Bundle-Adjustment

All sensors used in the above steps are subject to measurement noise. Therefore, we perform a range-visual bundle-adjustment (R-BA) optimization to reduce noise-induced measurement errors. The R-BA extends the standard bundle-adjustment with an additional term in the cost function for the LRF measurement. This addition is necessary, as the initial range measurement used for the structure scaling might be noisy and thus slightly wrong. However, adding the additional cost to the optimization reduces the impact of the assumed Gaussian white noise on the range measurement.

${}^{i}\mathbf{P}$ is the i-th image projection matrix used to project the j-th 3D-feature $\mathbf{F}^{(j)}$ onto the image plane. It is selected based on the criterion discussed in Sec. III-A. ${}^{i}\mathbf{f}^{(j)}$ is the

corresponding normalized pixel measurement in the i-th image. With this, the cost function to be minimized becomes

$$\arg\min_{{}^{i}\mathbf{P},\mathbf{F}^{(j)}} \sum_{i=0}^{N} \left(\left| {}^{i}\breve{z}_{\boldsymbol{r}} - {}^{i}\hat{z}_{\boldsymbol{r}} \right| + \sum_{j=0}^{M} d\left({}^{i}\mathbf{P}\mathbf{F}^{(j)}, {}^{i}\mathbf{f}^{(j)} \right) \right) . \quad (7)$$

E. Bias Estimation and Inertial Alignment

The IMU bias estimation from VINS-Mono is kept, which estimates the gyroscope bias using the IMU pre-integration first presented in Ref. [18]. Further, the initial acceleration bias ${}^{\mathcal{W}}\hat{\mathbf{b}}_{a} = \mathbf{0}_{\mathrm{d}}\,\mathrm{m\,s^{-2}}$ is used. Several state-of-the-art visual-inertial estimators have shown that they can handle an initial zero acceleration bias estimate and converge to the ground truth [5], [13].

The remaining initial states including only the camera frame velocities and the gravity direction, are estimated in a LLS estimation using the metrically scaled camera poses from the previous step. The frame to frame measurement matrix and vector for these remaining states are

$${}^{k}\mathbf{z}_{k+1} = \begin{bmatrix} {}^{k}\hat{\boldsymbol{\alpha}}_{k+1} - {}^{\mathcal{B}}\mathbf{p}_{\mathcal{C}} + {}^{k}\mathbf{R}_{k+1}\,{}^{\mathcal{B}}\mathbf{p}_{\mathcal{C}} - {}^{k}\mathbf{R}_{\mathcal{C}}\,\Delta^{\mathcal{C}}\mathbf{p}_{k} \\ {}^{k}\boldsymbol{\beta}_{k+1} \end{bmatrix} \quad (8)$$

$${}^{k}\mathbf{H}_{k+1} = \begin{bmatrix} -\mathbf{I}_3\,\delta t_k & \mathbf{0}_3 & \frac{1}{2}\,{}^{k}\mathbf{R}_{\mathcal{C}}\,\delta t_k^2 \\ -\mathbf{I}_3 & {}^{k}\mathbf{R}_{k+1} & {}^{k}\mathbf{R}_{\mathcal{C}}\,\delta t_k \end{bmatrix} \quad (9)$$

In contrast to the VINS-Mono formulation (see Eqs. (2)-(3)) the new full measurement matrix ${}^{k}\mathbf{H}_{k+N} \in \mathbb{R}^{4N\times(6N+3)}$ matrix only needs three camera poses to become invertible and the states therefore observable. Further, regardless of the scenario, the measurement vector is guaranteed to be non-zero, eliminating the possibility of the trivial solution in constant-velocity or free-fall scenarios.

IV. Simulation Tests

Initially, we investigate the performance of the proposed algorithm under the influence of standard measurement noise. Therefore, we generated range, feature, and inertial data in a point-based simulation under a constant velocity motion with ${}^{\mathcal{W}}\mathbf{v}_0 = \begin{bmatrix}1 & 1 & 0\end{bmatrix}^{\mathsf{T}}\mathrm{m\,s^{-1}}$. All sensor noises are assumed to be white Gaussian, and are set to values representative of the sensors listed in Sec. V. We then evaluated the initialization algorithm on 100 independent Monte-Carlo runs.

The results of this Monte-Carlo simulation are displayed in Fig. 5. This figure shows the mean and standard deviation of the norm of the error in attitude, position, and velocity throughout the window. As can be seen, for all three states, the error norm is low. Especially the small estimated velocity error shows that this approach can be used to initialize a visual-inertial estimator near the optimal solution.

Furthermore, we performed various sensitivity tests with simulated data on different parameters such as (i) feature tracking pixel noise, (ii) distance measurement noise, (iii) number of keyframes in the initialization window, (iv) number of tracked features and required baseline for keyframe selection, and (v) planar and structured environments. From these tests we concluded that our algorithm performs as expected independently of the environment, with 10 keyframes in the initialization window, and with $100-200$ tracked features. The authors refer to [21] for a more detailed simulation analysis and to [28] for a stress test of the facet assumption. Further, this evaluations showed that the optimization can mitigate measurement noise if its standard deviation is below $3\,\mathrm{px}$ for the features and $10\,\%$ of the flight height for the range measurement.

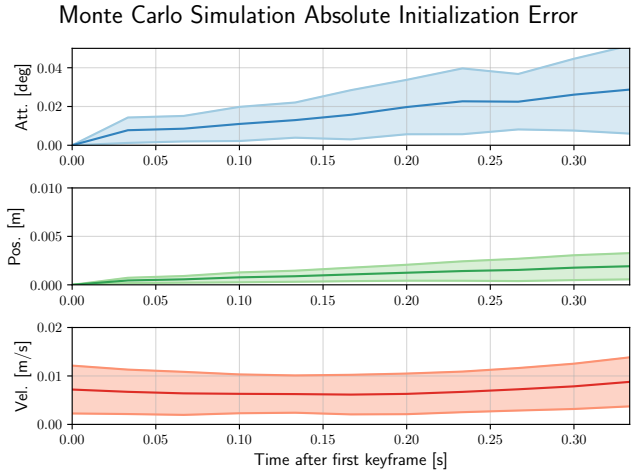


Fig. 5. Monte-Carlo evaluation of the proposed algorithm with 100 independently simulated data runs with constant velocity flight of ${}^{\mathcal{W}}\mathbf{v} = \begin{bmatrix}1 & 1 & 0\end{bmatrix}^{\mathsf{T}}\mathrm{m/s}$ and a flight height of $1\,\mathrm{m}$. This result shows the mean and standard deviation boundaries (1σ boundary) of the error for each keyframe in the initialization window. For all runs the window was set to 10 keyframes at an image rate of $30\,\mathrm{Hz}$. The position and velocity errors throughout the initialization period is low enough to initialize a visual-inertial estimator.

V. Experiments

A. Experimental Setup

The experiments were carried out on an AscTec Hummingbird quadrocopter. Sensors included the internal IMU of the Hummingbird, a Matrixvision Bluefox mvBlueFox-MLCw camera with $640\,\mathrm{px}\times 480\,\mathrm{px}$ resolution, and a Garmin Lidar Range v3. Ground truth for all flights was recorded with an Optitrack motion capture system. The algorithm was implemented in C++, as an extension of the open-source version of VINS-Mono using the Ceres Solver [29] for the R-BA. It ran on OdroidXU4 under Ubuntu 18.04 and ROS melodic in SkiffOS [30].

In our test, the MAV was commanded to a constant velocity flight of $0.5\,\mathrm{m\,s^{-1}}$ using the Optitrack pose as reference input for the flight controller. Although inertial attitude control would be more representative of an actual mid-air re-initialization scenario, attitude and velocity control with motion capture was deemed safer to avoid a crash in the limited lab space. The constant velocity is representative of a MAV applying constant thrust and controlled to a level attitude through an IMU after a VIO failure. The initialization algorithm was started on board in real time using a window of 10 image frames with corresponding LRF measurements. The initial state estimate was then used to start the VIO navigation framework VINS-Mono. Once initialized, the reference input of the controller was switched from motion capture to VINS-Mono to demonstrate mid-air recovery and stable follow-up flight. Further, the experiments were carried out in an cluttered environment with small

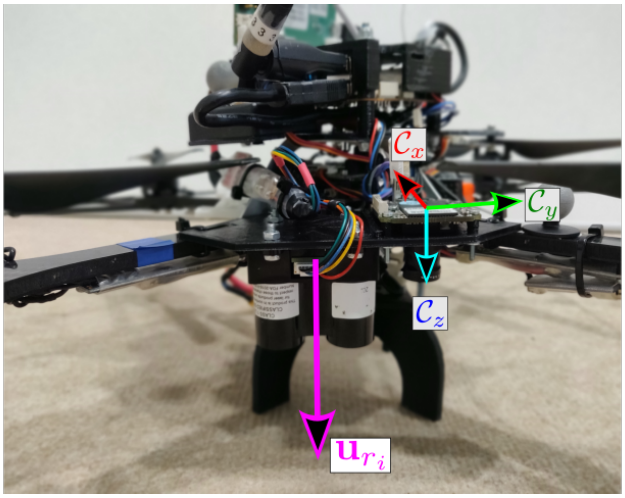


Fig. 6. For the experiments a AscTec Hummingbird quadrocopter equipped with an OdroidXU4 for onboard computations was used. The visual data (images) were provided by and Matrixvision Bluefox mvBlueFox-MLCw camera (coordinate system) mounted next to a Garmin Lidar Range v3 (pink range arrow) for single range measurements.

objects lying on a plane with a maximum height difference of $10\,\%$ of the flight height.

B. Results

The trajectory ground truth of this experiment is presented in Fig. 2. It demonstrates that our framework can initialize in a constant velocity flight condition, which would be unobservable for any VIO approach. Further, our approach can also initialize the full state of the optimization-based estimator VINS-Mono at metric scale, and then safely use it for the MAV control input. This figure further shows that our framework is accurate enough to initialize an estimator and fast enough to run onboard an embedded MAV system. For this experiment, the computation time was measured to be approximately $0.75\,\mathrm{s}$, including a data acquisition time of $0.33\,\mathrm{s}$ on the OdroidXU4 embedded computer.

Furthermore, as shown in Fig. 7, the position, velocity, and attitude error norms throughout the initialization period are low enough to initialize visual-inertial estimators. The mean and standard deviation of the error in the initialization window within this experiment is calculated to be $2.524 \pm 0.799°$ in attitude, $0.0070 \pm 0.0051\,\mathrm{m}$ in position, and $0.0794 \pm 0.0038\,\mathrm{m\,s^{-1}}$ in velocity.

We then tried to start VINS-Mono with its original initialization approach offline. Out-of-the-box VINS-Mono does not initialize in the given scenario since insufficient accelerations are present for VIO. For comparison purposes, we disabled all excitation checks in VINS-Mono and tried to initialize it under the constant velocity motion. The outcome of this test is shown in Fig. 7 (dashed lines).

In comparison to our approach, the visual-inertial initialization algorithm of VINS-Mono results in larger initial errors. Especially the unobservable metric scale in VINS-Mono's problem formulation renders it degenerate, as expected and analyzed in Sec. II-B. Subsequently, the visual-initially derived initial state led VINS-Mono to diverge as shown in Fig. 2.

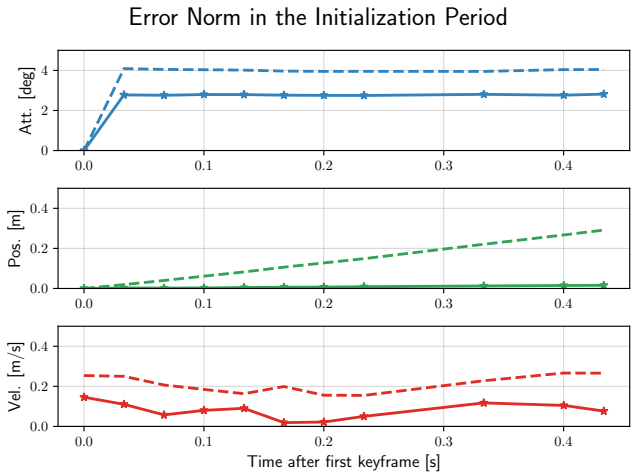


Fig. 7. Our approach's attitude, position, and velocity error norms (solid lines) of the initialization period in a real-world experiment with constant velocity flight shown in Fig. 2. In comparison, VINS-Mono's initialization state error norms are presented (dashed lines). As can be seen, our framework outperforms the visual-inertial only initialization for all three states.

VI. Conclusion

Visual-inertial odometry cannot observe the metric scale in the absence of acceleration change. This VIO limitation is even more problematic in the event of mid-air re-initialization, where either constant velocity (zero acceleration) or free-fall trajectories (constant acceleration) are expected, and other navigation states are completely unknown (unlike e.g., before take-off on the ground). We tackled this issue through a novel range-visual-inertial MAV initialization algorithm that can function even in the absence of excitation, and without prior environment nor state knowledge. As a core element of our approach, we leverage the distance measurement of a laser range finder which is tightly integrated into the visual-inertial system for robust metric system initialization in arbitrary situations. With the only requirement of local flatness (i.e., planar terrain in between three visual features) our approach is applicable in a large variety of, even to some extent cluttered, environments.

We analyzed our proposed approach in a Monte-Carlos simulation environment, which showed it to be robust against standard sensor noise values. We demonstrated our approach in real-time with closed-loop control onboard an MAV and compared it to the start-of-the-art VINS-Mono initialization algorithm. Future work includes outlier identification and rejection of the facet triangulation and full integration in an in-flight fault-detection and recovery framework.

Acknowledgment

Part of this work has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement 871260. Part of the research described in this paper was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration (80NMOOI8D0004).

REFERENCES

[1] H. Strasdat, J. M. Montiel, and A. J. Davison, "Real-time Monocular SLAM: Why Filter?" *Proceedings - IEEE International Conference on Robotics and Automation 2010 (ICRA10)*, pp. 2657–2664, 2010.

[2] G. Welch and G. Bishop, "An Introduction to the Kalman Filter," *In Practice*, vol. 8, 2006.

[3] D. Donavanik, A. Hardt-Stremayr, G. Gremillion, S. Weiss, and W. Nothwang, "Multi-sensor fusion techniques for state estimation of micro air vehicles," in *Micro- and Nanotechnology Sensors, Systems, and Applications VIII*. Baltimore, MA: SPIE, 2016.

[4] S. Lynen, M. W. Achtelik, S. Weiss, M. Chli, and R. Siegwart, "A robust and modular multi-sensor fusion approach applied to MAV navigation," in *IEEE International Conference on Intelligent Robots and Systems*, 2013.

[5] S. Weiss, M. W. Achtelik, S. Lynen, M. Chli, and R. Siegwart, "Real-time onboard visual-inertial state estimation and self-calibration of MAVs in unknown environments," in *Proceedings - IEEE International Conference on Robotics and Automation*, 2012.

[6] A. I. Mourikis and S. I. Roumeliotis, "A multi-state constraint Kalman filter for vision-aided inertial navigation," in *Proceedings - 2007 IEEE International Conference on Robotics and Automation (ICRA)*. Rome, Italy: IEEE, 2007, pp. 3565–3572.

[7] M. Li and A. I. Mourikis, "High-precision, consistent EKF-based visual-inertial odometry," *The International Journal of Robotics Research*, vol. 32, no. 6, pp. 690–711, 2013. [Online]. Available: http://ijr.sagepub.com/cgi/doi/10.1177/0278364913481251

[8] J. Delaune, R. Hewitt, L. Lytle, C. Sorice, R. Thakker, and L. Matthies, "Thermal-Inertial Odometry for Autonomous Flight Throughout the Night," *IEEE International Conference on Intelligent Robots and Systems (IROS) 2019*, pp. 1122–1128, 2019.

[9] P. Geneva, K. Eckenhoff, W. Lee, Y. Yang, and G. Huang, "OpenVINS: A Research Platform for Visual-Inertial Estimation," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 4666–4672.

[10] M. Bloesch, M. Burri, S. Omari, M. Hutter, and R. Siegwart, "Iterated extended Kalman filter based visual-inertial odometry using direct photometric feedback," *International Journal of Robotics Research*, vol. 36, no. 10, pp. 1053–1072, 2017.

[11] R. Mur-Artal, J. M. 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, 10 2015.

[12] C. Forster, M. Pizzoli, and D. Scaramuzza, "SVO : Fast Semi-Direct Monocular Visual Odometry," in *IEEE International Conference on Robotics and Automation*, 2014.

[13] 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. 1–17, 2018.

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

[15] S. Weiss, R. Brockers, S. Albrektsen, and L. Matthies, "Inertial Optical Flow for Throw-And-Go Micro Air Vehicles," in *Proceedings - 2015 IEEE Winter Conference on Applications of Computer Vision, WACV 2015*, 2015, pp. 262–269.

[16] M. Faessler, F. Fontana, C. Forster, and D. Scaramuzza, "Automatic Re-Initialization and Failure Recovery for Aggressive Flight with a Monocular Vision-Based Quadrotor," in *Proceedings - IEEE International Conference on Robotics and Automation (ICRA)*. Seattle, WA: IEEE, 2015, pp. 1722–1729.

[17] M. Scheiber, J. Delaune, R. Brockers, and S. Weiss, "Visual-Inertial On-Board Throw-and-Go Initialization for Micro Air Vehicles," in *Proceedings - IEEE International Conference on Intelligent Robots and Systems (IROS) 2019*. Macau, China: IEEE, 2019, pp. 6899–6905.

[18] 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*, 2015.

[19] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial and Multi-Map SLAM," 0. [Online]. Available: http://arxiv.org/abs/2007.11898

[20] T. Qin and S. Shen, "Robust initialization of monocular visual-inertial estimation on aerial robots," in *IEEE International Conference on Intelligent Robots and Systems*, Vancouver, BA, 2017.

[21] M. Scheiber, "Range-Visual-Inertial Initialization For Micro Aerial Vehicles," Master's thesis, Universität Klagenfurt, Klagenfurt, Austria, 8 2020. [Online]. Available: https://bit.ly/37Vw4jj

[22] F. R. Gantmacher and K. A. Hirsch, *The Theory of Matrices*. New York, NY, USA: Chealsea Publishing Company, 1959, vol. 1.

[23] A. Martinelli, "Closed-Form Solution of Visual-Inertial Structure from Motion," *International Journal of Computer Vision*, vol. 106, no. 2, pp. 138–152, 2014.

[24] K. J. Wu and S. I. Roumeliotis, "Unobservable Directions of VINS Under Special Motions," Multiple Autonomous Robotic Systems Laboratory (Mars Lab), Minneapolis, MN, USA, Tech. Rep. 2, 2016.

[25] J. Delaune, R. Brockers, D. S. Bayard, H. Dor, R. Hewitt, J. Sawoniewicz, G. Kubiak, T. Tzanetos, L. Matthies, and J. B. Balaram, "Extended Navigation Capabilities for a Future Mars Science Helicopter Concept," in *2020 IEEE Aerospace Conference*. Big Sky, MT, USA: IEEE, 2020, pp. 1–10.

[26] J. Delaune, D. S. Bayard, and R. Brockers, "xVIO: A Range-Visual-Inertial Odometry Framework," 2020. [Online]. Available: http://arxiv.org/abs/2010.06677

[27] D. T. Lee and B. J. Schachter, "Two algorithms for constructing a Delaunay triangulation," *International Journal of Computer & Information Sciences*, vol. 9, no. 3, p. 219–242, 1980.

[28] J. Delaune, D. Bayard, and R. Brockers, "Range-visual-inertial odometry: Scale observability without excitation," *IEEE Robotics and Automation Letters*, 2021.

[29] S. Agarwal, K. Mierle, and Others, "Ceres Solver." [Online]. Available: http://ceres-solver.org

[30] C. Stewart, "SkiffOS: Minimal Cross-compiled Linux for Embedded Containers," Mar. 2021. [Online]. Available: https://doi.org/10.5281/zenodo.4628881

6DoF State Estimation with a Mesh Constrained Particle Filter For Wheeled Robots

Pete Schroepfer[1,3], Georges Chahine[1,3], Cédric Pradalier[1,2]

***Abstract*— In this work, we present a highly accurate *Mesh Constrained Particle Filter* (MCPF) for wheeled robots. We demonstrate that the MCPF is capable of estimating poses with 6DoF in real-time on an embedded computer due to low particle count requirements. To achieve this, MCPF's transition function constrains particle movement to a mesh surface approximating the surface the robot is traveling on. By constraining the particles, we reduce the dimensions of the effective work space the robot is operating in. In other words, the robot is effectively lying on a manifold (locally) with 3DoF embedded in SE(3). Importantly, by reducing this effective work space, significantly improved accuracy is also achieved while maintaining low particle density when compared to a dense SPF. In addition to showing improved accuracy and real-time performance, we demonstrate that the MCPF provides high levels of robustness to lost or dropped anchor measurements. Moreover, this approach has been tested on the walls of real-world storage tanks using a magnetic-wheeled crawler in the field.**

Fig. 1. Cralwer Robot Magnetically Attached to a Storage Tank in Bazancourt, France.

I. INTRODUCTION

In mobile robotics, wheeled robots are fairly ubiquitous. There are vehicles, wheeled humanoid robots, home cleaning robots, inspection robots, and forest exploring robots, to name a few. These robots all tend to share a common trait insofar as they traditionally move on a smooth or relatively smooth surface. They are often, also, equipped with wheel encoders to provide odometry and an IMU or compass to provide heading information.

Similarly, for most of the tasks these robots perform, quality localization is tightly coupled with their measured utility. For example, autonomous inspection robots (like the one discussed in this paper) are tasked with autonomously scanning large metal surfaces such as ships or storage tanks. In doing so, they must identify and map hazards such as thinning or decaying metal walls. However, if the robot has a significant localization error, then as the error grows, the usefulness of the hazard map diminishes, as does the usefulness of the robot.

Many of these robots also operate in GPS-denied areas. GPS denied areas occur wherever there is occlusion blocking a receiver from receiving satellite signals. This may include, among other places, being indoors or under a shelter, next to tall buildings or structures, in forests, in mines, in tunnels, or, in our particular case, attached to a giant metal storage tank or ship [1].

In GPS-Denied areas, one popular sensor used as an alternative to GPS are Ultra-Wide Band (UWB) sensors. Generally, by using either two-way ranging (TWR) or time-of-flight (TOF), an UWB system can provide range measurements with an accuracy of around $\pm$ 10cm with varying ranges from 30 meters to 100 meters depending on the types of filter and antenna used [2].

Given that wheeled robots with similar components are becoming more commonplace, and localization for these robots is often inextricably linked to their usefulness, it behooves the robotics community to explore and report on as many highly accurate and robust localization options it can discover. This is even more true if the presented option could potentially serve as an enhancing component within other similar systems.

In this paper, this is exactly what we aim to present. With the Mesh Constrained Particle Filter (MCPF) presented in this paper, we show how we can leverage the fact that wheeled robots move on a surface to enhance our particle filter's transition function. Below, we demonstrate that: (1) by constraining the movement of the particles to a mesh during the transition function the MCPF is able to achieve high accuracy localization with 6DoF operating in real-time on an embedded system within the crawler; (2) the mesh constraint is essential to performance as removing the mesh constraint greatly degrades accuracy even with extremely high particle

*This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No. 871260. Corresponding author: cedric.pradalier@georgiatech-metz.fr
[1]CNRS IRL2958 GT-CNRS, Metz, France
[2]GeorgiaTech Lorraine, Metz, France
[3]Georgia Institute of Technology, Atlanta, USA

density; and (3) the use of the MCPF provides robustness against lost or dropped range measurements.

II. Related Work

The concept of the MCPF for wheeled robots is rooted in a body of combined works from both non-mobile and mobile robotics. In both subfields, there is support for the concept of leveraging a surface approximation in a Recursive Bayesian Filter. While in mobile robotics, there is strong support for using a PF over an EKF when using range based measurements such as UWB to correct global position.

A similar implementation to the MCPF is the Manifold Particle Filter (MPF) used for localizing a robot arm and underactuated hand [3]. Like the MCPF, the MPF leverages a surface approximation to limit the possible poses of the robot [3]. This, in turn, improves the filter's accuracy while simultaneously lowering the required number of particles to achieve this [3].

While the MPF is similar, one primary difference is that the manifold is not constraining the motion model. In our case, we have a motion model in SE(2) that we apply to particles in SE(3). Here, within the transition function, the particle motion is constrained to a surface approximation because, unlike a mechanical arm, the robot cannot leave the surface. Furthermore, due to the physical attitude constraints enforced by the surface on the wheeled robot, we are able to leverage information about the normal vector of the manifold to constrain the robot's pose.

In wheeled robotics, the Invariant Extended Kalman Filter (IEKF) also uses surface approximations to increase accuracy by projecting the estimated pose to the surface of a manifold as "measurement" update [4][1]. The Manifold Invariant Extended Kalman Filter (M-IEKF) then offered an improvement to the IEFK by both shifting the manifold constraint to the transition function, and mapping the motion model in SE(2) directly to the manifold in SE(3) eliminating any error caused by projection [5]. However, the M-IEKF implemented in [5] was implemented with a position measurement model, assuming a successful trilateration of the UWB anchors at every step. As will be discussed later, this assumption is not valid in our settings. Because of the specific structure of the state representation, integrating non-linear range measurements in the M-IEKF is a significant challenge that was not addressed in this paper.

Additionally, a significant difference between the situation presented in [5] and the current paper stems from the increased scale, in terms of size and complexity, of the experiment surface. While UWB is generally very accurate, when line-of-site (LOS) is not available it requires walls or structures to bounce signals off to ensure the tag can receive measurements. As mentioned above, our experiment was conducted outdoors in an open space with very few objects or walls for signals to bounce off of. As a result, the tag often received only between 0 and 2 messages, as seen in Figure 5. Based on [6] it is unclear whether an EKF could remain accurate with sporadically dropped or long periods of dropped range measurements. This too was not explored in [5] where the experiment was performed on a smaller scale with an inwardly curved plate where LOS for four anchors measurements could be guaranteed.

[1]Note that "Measurement" here is not a real measurement by a sensor, but instead just a state correction performed after motion is applied

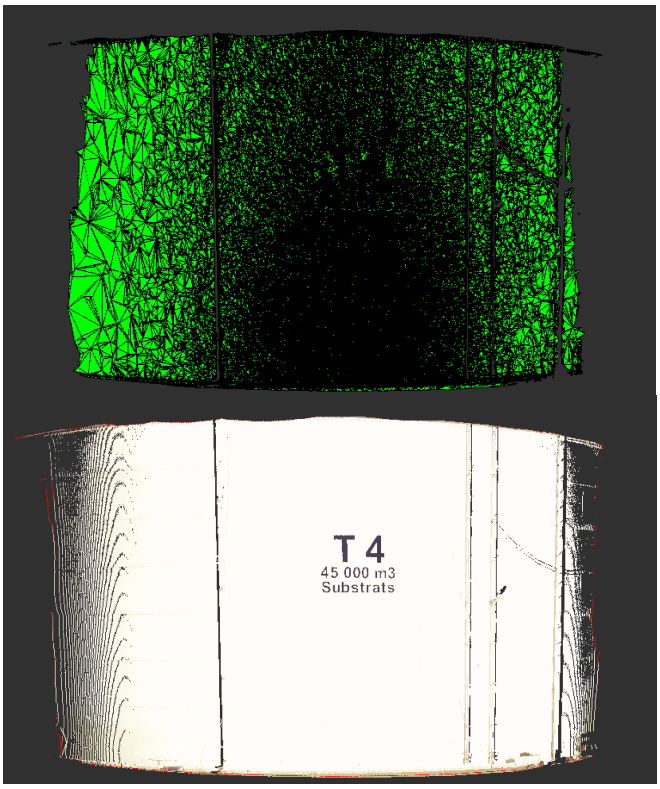


Fig. 2. caption.

In addition to the issue of scale, the choice of a particle filter when using UWB measurement is also supported in the current literature with respect to mobile robotics. A potential weakness of the EKF comes from the assumption that measurement noise is Gaussian and that pose probability distributions are not multimodal[7], [8], [9]. As range sensors can result in multimodal distributions and often exhibit non-Gaussian noise [7], [9], using an EKF may create a discrepancy between the model and reality, resulting in an increased estimation error [10].

Lastly, for particle filters using in mobile robotics, to the best of the authors' knowledge, surface approximation has rarely been integrated in such a filter, if at all to constrain the pose estimate. The closest integration found was in a case where a mesh had been integrated within a LiDAR model as part of range based correction step [11]. However, in this case, the mesh in only used to support the observation model but not used to constrain the motion model.

III. Mesh Constrained Particle Filter

In this section, we first provide the basic theoretical assumptions underlying the key components of the MCPF, namely, the initialization and translation functions. We then

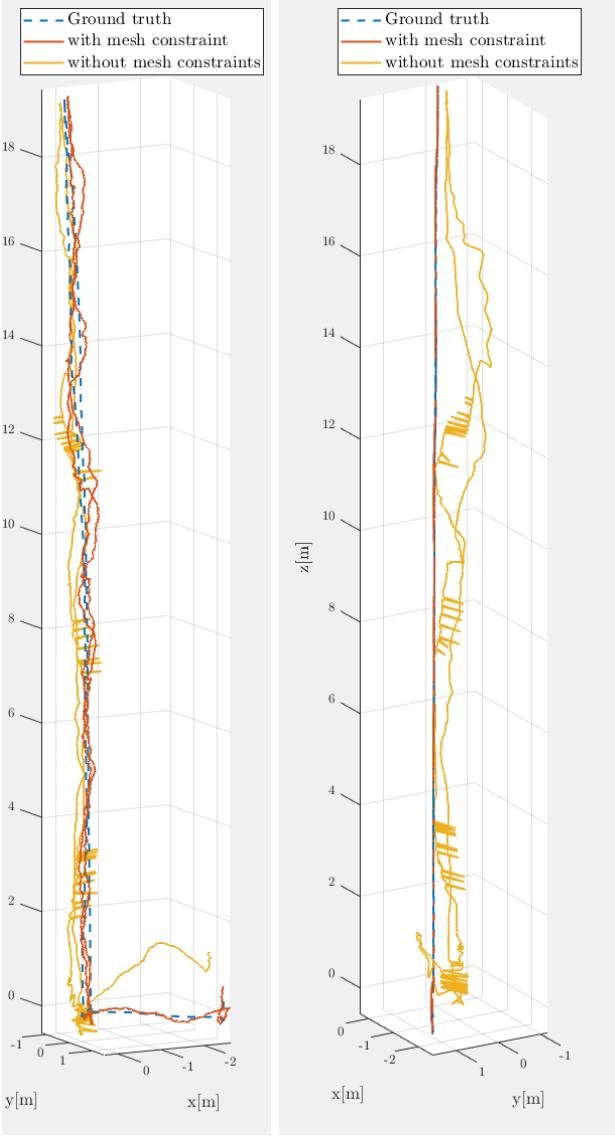


Fig. 3. 3D path comparison of the 200 particle MCPF compared to a 20000 SPF and the ground truth (both a front and side view)

discuss, at an abstract level, the methodology of these two components.[2] Finally, we show how, at the implementation level, we are able to instantiate the MCPF.

A. Assumptions

We are considering a robot R with a starting position at r and a state (x, y, z, q) where q is a quaternion representing orientation. R is moving on a surface that can be defined mathematically as a smooth manifold $M \subset \mathbb{R}^3$. As R is moving on M and is a rigid body wheeled robot, R's orientation will be constrained by its location on M and the normal vector at that location. We further assume there exists some approximation of M, $aM \subset \mathbb{R}^3$, such that each point $a_i \in aM$ is a good approximation of a point $m_i \in M$. Lastly, we assume that during initialization we know some point $k_i \in \mathbb{R}^3$ in a neighborhood of the real robot position r.

[2]Importantly, the main contribution of the MCPF comes from the surface approximation constraint in the transition function. The correction or measurement steps and resampling use conventional models and algorithms for UWB and an IMU. As such, we do not address these elements of the MCPF as part of the methodology.

B. Methodology of Key Components

1) Initialization: For initialization, the goal is to create a normally distributed set of particles attached to the mesh, close to the point r. Additionally, the set of particles should have a uniform distribution of orientations, with each particle's pose being consistent with the normal vector of M at that particle's respective position.

To this end, we create an initial sphere centered around k_i. Within this sphere, N particles are created following a normal distribution centered on k_i. The orientations, by contrast, follow a uniform distribution because we do not assume knowledge of an approximate initial orientation.

Once the sphere of particles is created, each particle $P_i \in SE(3)$ is projected to the closest point on aM using a general projection function $f_{\text{proj}}(P_i)$. This results in a set of particles P_n attached to the surface of, aM, normally distributed with a center close to r.

Once each particle P_i is attached to the surface of mA, we further constrain the orientation of P_i to be consistent with a robot driving on M by leveraging the normal vector of M at P_i's position. Here we let the attitude of P_i be expressed as a rotation matrix $A \in SO(3)$ not yet subject to mesh constraints and $B \in SO(3)$ as a rotation matrix subject to the mesh constraints. We then express A and B in their 3D vector basis form: $A = [A_x, A_y, A_z]$, and $B = [B_x, B_y, B_z]$. Let n be the normal vector to the mesh. A particle on the local mesh plane will have its attitude constrained such that:

$$B_x = \frac{A_y \times n}{\|A_y \times n\|} \tag{1}$$

$$B_y = \frac{n \times B_x}{\|n \times B_x\|} \tag{2}$$

$$B_z = n \tag{3}$$

$B = [B_x, B_y, B_z]$ is therefore a family of orthonormal unit vectors, hence the 3D basis of a rotation matrix that describes the orientation of P_i as constrained by aM, with B_x pointing forward, B_y sideways, and n pointing outwards.

2) Transition: The transition step is completed in a manner similar to the process stated in the initialization section. First we apply standard motion by determining the displacement $\delta O = O_{\text{t-1}}^{-1} O_{\text{t}}$ where O_t is a transformation matrix representing the odometry readings at time t. Because we consider a wheeled robot with a 2D odometry model, we have $\delta O \in SE(2)$. We expand it to $SE(3)$ and calculate the new state of each P_i as $P_i = P_i\, \delta O$.

Finally, each particle P_i is projected using $f_{\text{proj}}(P_i)$ onto aM and, likewise, each particle's orientation is further constrained to aM using eq. 1 to 3.

C. Implementation

In this section, we will give a lower level account of how the key components of the MCPF were implemented in our system, in addition to how use utilized the sensor measurements. As a general matter, the implementation of the MCPF was done in C++ using CGAL (5.x) for handling the mesh and mesh projections.

1) Initialization: To create the initial known position, k_i we perform a trilateration using the range measurements from four UWB anchors and then publish these results. The MCPF can then listen for the results during initialization. Note that the trilateration was used primarily for convenience, as this could be done by manually measuring or some other estimation method.

To create aM, we generate a point cloud of the surface the robot will be moving on. We then processes this point cloud to create a mesh. The mesh data structure is then provided to the MCPF via a service call. By using a service call, this implementation allows other nodes to share the same mesh.[3]

2) Measurements: For the range measurements, we are using a standard UWB sensor setup where a tag receives range measurements from a set of anchors. When we receive a range measurement, we calculate the Euclidean distance between the known position of the anchor and P_i and then compare that to the range measurement to determine the particle likelihood.

For the orientation measurement, we extract the accelerometer data from the IMU and compare it with a predicted gravity vector using the particle orientation. We then use the geodesic distance between the two normalized vectors in $S(2)$ to determine the particle likelihood.

IV. Field Experiment

A. Experiment Description

In our case, the MCPF was field tested on a large storage container at a factory in Bazancourt, France. The storage containers were approximately 20 meters high and 20 meters in diameter. The robot is an Altiscan magnetic crawler manufactured by Roboplanet[4], with large magnetic wheels and employing a differential drive system (the tank and crawler can be seen in Fig. 1). The magnetic wheels are strong enough to attach the robot to metal objects at least 1mm thick. The CPU used by the crawler (which also ran the particle filter in real-time) was an Intel®Atom®x5-E3940. The sensors used for the particle filter were an ICM-20948 IMU and Decawave's MDECK1001 UWB Development Kit with a default configuration. Importantly, the MDECK1001 tag can only receive a maximum of the 4 closest anchors at any one time.

To create the mesh used by the MCPF, we took a high density laser scan of the storage tank with a Leica MS-60 total station (Leica). The scan was then processed, converted into a mesh data structure, and shared as a mesh message though a service call. The robot was then placed on the bottom of the storage tank where it could be tracked by the total station to create ground truth.

Two anchors were placed at the top of the storage tank, while 7 anchors were paced at various points around the base of the tank. The crawler then performed a mission scaling the tank, returning to the bottom, rotating, and then finally moving to the right. The movement of the crawler is mostly autonomous except for asking the user for input when an object, such as the safety rope, is observed by the crawler and considered a potential obstacle.

During the initial testing, the MCPF was able to run in real-time during the experiment and all sensor data was recorded in a ROS bag file. For validation purposes, we used this bag file without the recording of the particle filter to provide a statistical performance analysis of the particle filter, the results of which are reported below.

Lastly, to measure positional ground truth, the total station was used in tracking mode, whereby it could track the motion of a 3D Prism attached to the handle of the crawler. Motion tracking is done with millimeter precision but with some timing uncertainty reducing the precision to the centimer scale.

TABLE I
Average RMSE Comparison

Particles	Mesh	NoMesh
200	0.0856	0.3694
500	0.0780	0.3389
20000	–	0.2532

B. Evaluation

In this section, we evaluate the performance of the MCPF with respect to three metrics, namely the: (i) accuracy of the MCPF as compared to an identical filter without the mesh constrained transition function, which for convenience, we will call the standard particle filter (SPF); (ii) computational performance cost of adding the mesh constraint when compared to increasing particle sizes of the SPF; and finally (iii) robustness of the MCPF with respect to dropped or lost signals.

1) Position Accuracy: The positional accuracy of the MCPF is greatly improved when compared to the SPF. Figure 3 shows the crawler's estimated path from both the MCPF with 200 particles and the SPF with 20,000 particles. As can be seen in this image, the SPF estimates the path of the crawler as if it were oscillating on and off the mesh, making lateral movements as it moves up the wall. By contrast, the MCPF not only stays attached to the wall (as would be expected), large lateral movements are constrained as well.

The dramatic increase in accuracy of the MCPF is also evidence by the data in Table I. Here, the MCPF had an RMSE of 0.0856 and 0.0780 with a particle density of 200 and 500, respectively. By contrast, the SPF had an RMSE of 0.3694 and 0.3389 with particle densities of 200 and 500

[3]As a practical matter, using a mesh here instead of a mathematical model allows some expanded functionality through mesh tools that have been developed for ROS as well as visualization though RVIZ mesh plugins.

[4]http://www.roboplanet.fr/en/

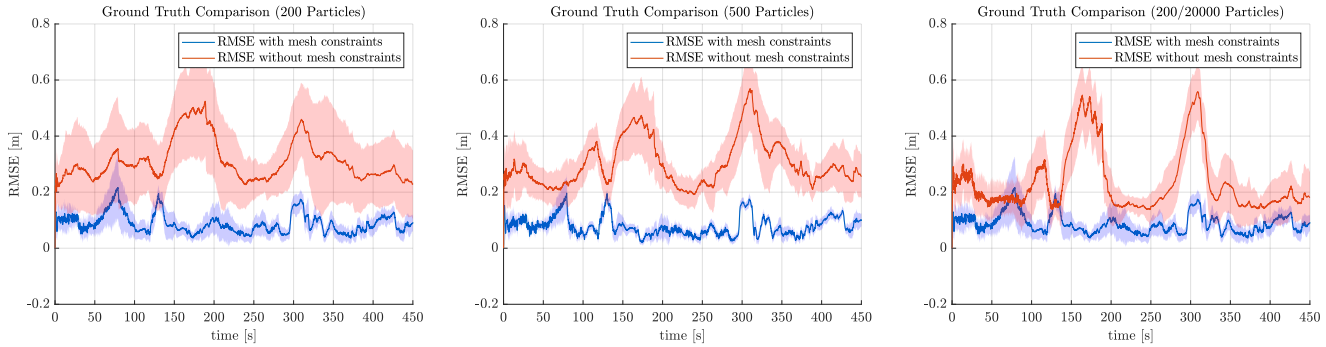


Fig. 4. Comparison of the RMS translation error for 200, 500 and 20000 particles, against ground truth obtained from the Leica MS60 Total Station. Shaded areas represent the standard deviation of the error. The inclusion of mesh constraints dramatically improves position estimates, as well as the standard deviation of the error.

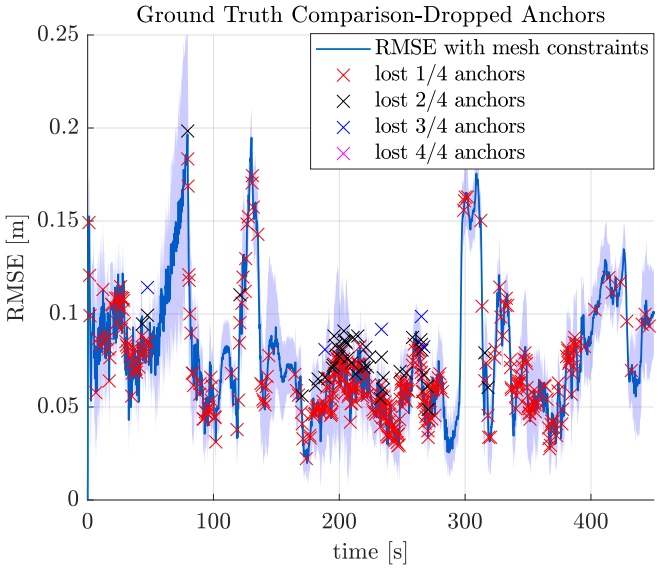


Fig. 5. Dropped anchor measurements during the field test overlaid on the RMSE graph for the 200 Particle MCPF. Figure 4 with 200 particles

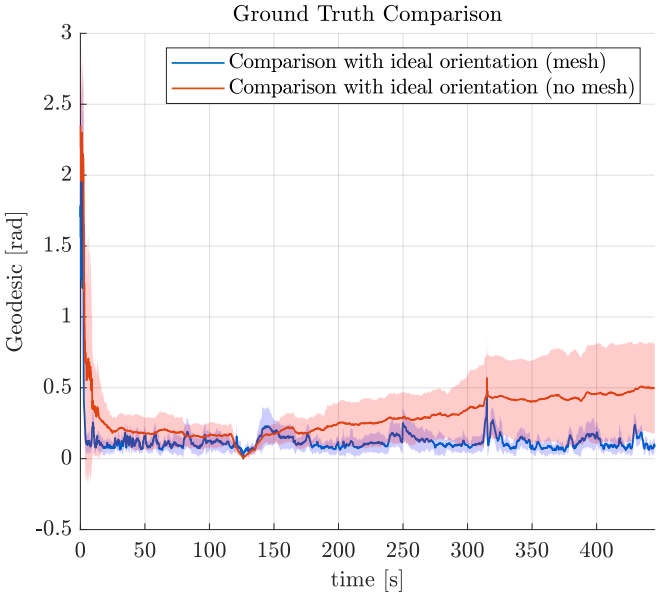


Fig. 6. Average residual geodesic over 20 runs, after comparing the robot attitude to that of an ideal orientation *i.e.,* when the robot is moving vertically on a straight line (lower is better). Shaded areas represent the standard deviations of the respective residuals.

respectively. Moreover, even when the particle density was increased to 20,000 RMSE of the SPF was only able to get as low as 0.2532 (which is still four times higher than the error of the MCPF with only 200 particles). This juxtaposition of accuracy of the MCPF and the SPF not only demonstrates the improvement of the MCPF, but shows how central adding the mesh constraint was to this improved accuracy, as that is the only key difference between the two cases.

2) Orientation Accuracy: Although a total station is capable of providing millimeter-accurate position estimates, a single station is incapable of providing attitude estimates. To that end, an experiment was performed where the robot was driven vertically, on a straight line. The robot is therefore expected to maintain its initial orientation throughout the trajectory, which we will call the "ideal orientation". By comparing the current attitude to the ideal orientation, Figure 6 shows how the inclusion of mesh constraints significantly improves attitude estimation. Still in the same figure, it is shown that the standard deviation is significantly smaller after adding mesh constraints. In numbers, the average geodesic residual for 20 runs with mesh constraints is 0.2574 rad. with an average standard deviation of $\pm$0.0701 rad.; whereas 20 runs without mesh constraints yielded a residual of 0.4263 rad, with an average standard deviation of $\pm$0.2073 rad..

3) performance: The accuracy benefits from the MCPF strongly outweigh the performance cost of adding the mesh projection into the transition function. The average execution time for the MCPF with 200 particles was 0.004281 ns while the average execution time for the MCPF with 500 particles was 0.009769 ns. As can be seen in Table I, the SPF has similar performance time at around 1500 and 4000 particles respectively. This means that, while the execution time of the SPF, below 1500 particles, outperforms the MCPF, the SPF, even when running 20,000 particles, cannot provide similar accuracy (see Table I.

By implication this also means, if a robot is capable of running a particle filter with 1500 particles in real-time, then that same robot could likely run the MCPF in real-time while

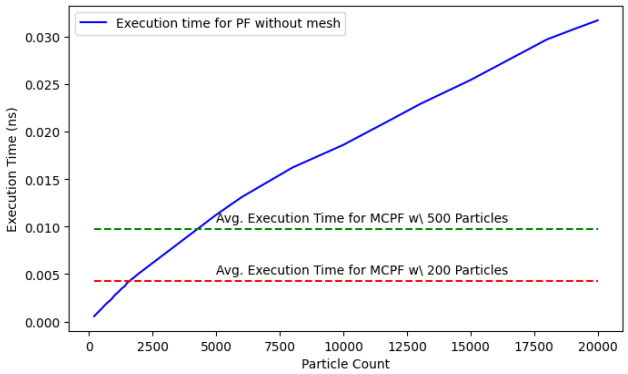


Fig. 7. Here you can see the execution time growth for a conventional particle filter with 6DoF (i.e. the MCPF with the mesh constraint removed) as particles increase. Layered on top is average execution time of the MCPF with 200 and 500 particles.

benefiting from improved accuracy and stability.[5]

4) Robustness to Dropped Measurements: Based on the Figure 4 and Figure 5, the MCPF appears to be fairly robust against both sporadically dropped anchor measurements and periods of low anchor measurements. As can be seen in Figure 5, the UWB used outdoors in this setting suffered from a significant number of dropped or missing range measurements during the field test. During the beginning and the end of the crawler's mission, the lost anchors appear to be dropped sporadically. Whereas around the middle (starting around 180s) there is a consistent state of dropped or missing measurements (in the middle this is likely due to the crawler being at the top of the tank and out of range from many of the anchors placed on the ground. Regardless, it can be seen in Figure 5 and Figure 4 that there are no major spikes in the RMSE during either the tail-ends or the middle of the crawler's mission. The consistent level of the RMSE for the MCPF despite both sporadically missing and consistently missing measurements strongly implies it is robust to these situations.

V. Conclusions

As can be seen from the results of the experiment discussed above, the addition of a mesh constraint to the particle filter greatly enhances the accuracy of the particle filter. By examining an identical particle filter with the mesh constraint removed and finding that even with 20,000 particles the MCPF still benefits from significantly higher accuracy with only 200 particles, it seems likely the mesh constraint is the primary cause of the increased performance.

Further, we have showed the MCPF is also extremely robust to dropped or periods of low measurement counts. As noted above, we observed both sporadic and prolonged periods of dropped anchor measurements. The RMSE during these periods barely varied. As such, the dropped or low anchor periods likely had little effect on the accuracy of the MCPF.

As a final note, it is important to recognize that the most significant change to the presented particle filter was only within the transition function. As such, this update to the transition function could be leveraged by several other instantiations of the particle filter. While we were not able to test this formally in this experiment, this could serve as a tool that could be plugged into other projects to potentially provide significant enhancements. This idea is even further bolstered by the similarity between what was done here and what was done in the case of the robot arm discussed above[3]. As such, further exploration of the portability of this transition function enhancement could also warrant further investigation.

[5]Indeed, as mentioned in section IV.A we were able to run the MCPF in real-time on an embedded computer that was simultaneously running all the other required applications necessary for the robot to operate autonomously and control its components.

Acknowledgment

This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No. 871260.

References

[1] J. Siva and C. Poellabauer, *Robot and Drone Localization in GPS-Denied Areas*. Cham: Springer International Publishing, 2019, pp. 597–631. [Online]. Available: https://doi.org/10.1007/978-3-319-92384-0_17

[2] A. R. Jiménez and F. Seco, "Comparing Decawave and Bespoon UWB location systems: Indoor/outdoor performance analysis," *2016 International Conference on Indoor Positioning and Indoor Navigation, IPIN 2016*, no. October, pp. 4–7, 2016.

[3] M. C. Koval, M. Klingensmith, S. S. Srinivasa, N. S. Pollard, and M. Kaess, "The manifold particle filter for state estimation on high-dimensional implicit manifolds," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 4673–4680, 2017.

[4] A. Barrau and S. Bonnabel, "The invariant extended Kalman filter as a stable observer," *IEEE Transactions on Automatic Control*, vol. 62, no. 4, pp. 1797–1812, 2017.

[5] B. Starbuck, A. Fornasier, S. Weiss, and C. Pradalier, "Consistent State Estimation on Manifolds for Autonomous Metal Structure Inspection," no. Icra, pp. 10 250–10 256, 2021.

[6] S. Pfeiffer, C. D. Wagter, and G. C. Croon, "A Computationally Efficient Moving Horizon Estimator for Ultra-Wideband Localization on Small Quadrotors," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 6725–6732, 2021.

[7] J. González, J. L. Blanco, C. Galindo, A. Ortiz-de Galisteo, J. A. Fernández-Madrigal, F. A. Moreno, and J. L. Martínez, "Mobile robot localization based on Ultra-Wide-Band ranging: A particle filter approach," *Robotics and Autonomous Systems*, vol. 57, no. 5, pp. 496–507, 2009.

[8] P. Vernaza and D. D. Lee, "Rao-blackwellized particle filtering for 6-DOF estimation of attitude and position via GPS and inertial sensors," *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2006, no. May, pp. 1571–1578, 2006.

[9] D. Knobloch, "Practical challenges of particle filter based UWB localization in vehicular environments," *2017 International Conference on Indoor Positioning and Indoor Navigation, IPIN 2017*, vol. 2017-January, pp. 1–5, 2017.

[10] P. M. Blok, K. van Boheemen, F. K. van Evert, J. IJsselmuiden, and G. H. Kim, "Robot navigation in orchards with localization based on Particle filter and Kalman filter," *Computers and Electronics in Agriculture*, vol. 157, no. October 2018, pp. 261–269, 2019.

[11] X. Chen, I. Vizzo, T. Labe, J. Behley, and C. Stachniss, "Range Image-based LiDAR Localization for Autonomous Vehicles," no. Mcl, pp. 5802–5808, 2021.

Consistent State Estimation on Manifolds for Autonomous Metal Structure Inspection

Bryan Starbuck[1†], Alessandro Fornasier[2†], Stephan Weiss[2], and Cédric Pradalier[1]

***Abstract*— This work presents the *Manifold Invariant Extended Kalman Filter*, a novel approach for better consistency and accuracy in state estimation on manifolds. The robustness of this filter allows for techniques with high noise potential like ultra-wideband localization to be used for a wider variety of applications like autonomous metal structure inspection. The filter is derived and its performance is evaluated by testing it on two different manifolds: a cylindrical one and a bivariate b-spline representation of a real vessel surface, showing its flexibility to being used on different types of surfaces. Its comparison with a standard EKF that uses virtual, noise-free measurements as manifold constraints proves that it outperforms standard approaches in consistency and accuracy. Further, an experiment using a real magnetic crawler robot on a curved metal surface with ultra-wideband localization shows that the proposed approach is viable in the real world application of autonomous metal structure inspection.**

I. INTRODUCTION

Routine inspection of large metal structures is of the utmost importance in avoiding environmental catastrophe and maintaining safety standards. Small differential-drive robots with magnetic wheels are being deployed on vessels and cargo ship hulls to ensure that these standards are met, but as of yet, the task is being completed via manual operation. Given the expansive dimensions of these structures, completing this task autonomously would be preferable, but with such high stakes, having the best localization accuracy and consistency is paramount. Even though classical methods for state estimation exist, they do not consider the fact that the robot is a planar robot moving on a curved surface. Thus, they tend to estimate the six-dimensional state, enforcing constraints on all known degrees of freedom, affecting the consistency of the approach. Therefore, motivated by the recent development of the consistent Invariant Extended Kalman Filter (IEKF) [1] [2] [3] [4], in this work, we propose a *Manifold Invariant Extended Kalman Filter*, a novel approach to consistent state estimation on manifolds with application to ship hull inspection.

A metal structure, e.g. a ship hull, can be thought of as a smooth surface embedded in three-dimensional Euclidean space, which can be viewed as a two-dimensional, differentiable, Riemannian manifold. This allows us to select a chart from the maximum atlas and hence define a chart map, a continuous, invertible, bijective map, that maps each point of the considered manifold to a two-dimensional Euclidean space. The full state of the planar robot moving on the manifold is six-dimensional, including position and orientation within a three-dimensional euclidean space, however, its "planarity" gives only three degrees of freedom: a two-dimensional position and the heading angle. Therefore, by applying a consistent IEKF on the product space $\mathbb{R}^2 \times SO(2)$, the localization problem can be solved entirely on the chart.

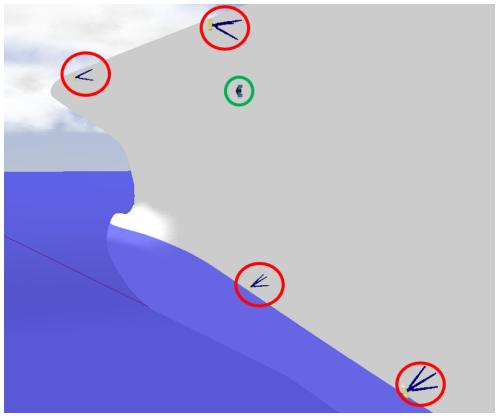


Fig. 1. Simulation of a Magnetic Crawler robot (green circle) on a Ship Hull with an Ultra-wideband Localization Grid (red circles).

Being able to evaluate the surface and it's derivatives at any point is necessary to create a basis for the tangent space in order to recover the full orientation of the robot from the minimal state estimated on the chart. Given that there are no equations for generic metal structures like a ship hull, and that the equation of a surface must be known to apply the proposed methodology, a bivariate b-spline representation of the surface was recognized as a sufficient substitute. This can be obtained by extracting the vertices of the surface from its CAD model for interpolation, or by taking a laser scan of the structure and interpolating the resulting point cloud.

The propagation model of the magnetic crawler robot is based on its odometry measurements [5], but even with high precision wheel encoders, this is only reliable for predicting the robot's state within a plane that is tangent to the surface. The measurement model for state localization is given by modelling ultra-wideband (UWB) range measurements with a trilateration framework [6], which in ideal conditions can accurately update the robot's position within $\pm 5\,\mathrm{cm}$, but with high noise potential from wave deflection off the metal surface, this is not a safe assumption to make [7]. Therefore, to account for the inherent drifting from the surface that the

This work was supported by the EU-H2020 project BUGWRIGHT2 (GA 871260)

[1]Bryan Starbuck and Cédric Pradalier are with Georgia Tech Lorraine - CNRS UMI 2958 bstarbuck3@gatech.edu, cedric.pradalier@georgiatech-metz.fr

[2]Alessandro Fornasier and Stephan Weiss are with the Control of Networked Systems Group, University of Klagenfurt, Austria alessandro.fornasier, stephan.weiss@ieee.org

[†]These authors contributed equally

robot's state will experience, classical approaches tend to solve the localization problem by forcing constraints within the Extended Kalman Filter (EKF) framework [8]. Imposing two virtual, zero noise measurements as constraints such that the first constraint maps the state of the robot to the surface, and the second one maintains collinearity between the vertical axis of the robot and the normal to the surface resulting in a full estimation of the robot's position and orientation, but sacrificing the filter's consistency. With the loss of consistency, the loss of accuracy and robustness follows. To validate the benefit and versatility of the proposed approach compared to classical approaches, simulations were carried out specifically on cylindrical and curved surfaces, simulating respectively a cylindrical vessel and a ship hull. Moreover, An experiment with a real magnetic crawler robot on a curved metal plate has been performed to show the feasibility of the methodology in real-world scenarios.

II. Related Work

Strategies for metal structure inspection can take on many forms, but in every case, fundamental questions must be investigated, such as: *Which sensors should be used for mapping and localization?* and, *Which filtering technique will produce the best results?* For bridge inspection, unmanned aerial vehicles (UAVs) equipped with lidar for mapping and visual, inertial odometry systems for localization collect data from the bridge to be processed for structural analysis [9]. For ship hull inspection, autonomous underwater vehicles (AUVs) equipped with cameras for mapping and sonar systems for localization similarly complete the task [10]. However, it should be noted that the inspection of metal structures and vessels is not solely confined to airborn inspection or to below the waterline. In fact, large cargo ships can protrude up to and exceeding fifty meters above water level especially when unloaded. Therefore, to complete the inspection most efficiently and in it's entirety, utilizing a combination of UAVs, AUVs, and differential-drive, magnetic-wheeled crawler robots could be quite advantageous.

The crawlers hold primary responsibility for inspecting the portion of the ship hull that protrudes from the water, and high accuracy localization is fundamental to this being accomplished autonomously. There are various sensors that come to mind as candidates for correcting the position of the robot such as RTK-GPS, Wifi, and UWB. RTK-GPS is too unreliable given that clear line of sight to satellites is always required, and Wifi is also unreliable because it is too sensitive to interference. UWB which is based on the time of flight of wave transmission resulting in a range measurement is proven to be a reliable method of localizing multiple moving targets [11]. The major factor which highlights UWB as a more robust method for this application is that it has high bandwidth meaning that the waves experience less interference while reliably transmitting small packets of data at a distance generally up to 30 meters [12]. Although UWB is generally used for indoor object tracking, given that more specialized filters are being developed to enhance its robustness, it is becoming increasingly feasible to experiment with outdoor applications like metal structure inspection. It therefore follows that a grid of UWB beacons for a robot

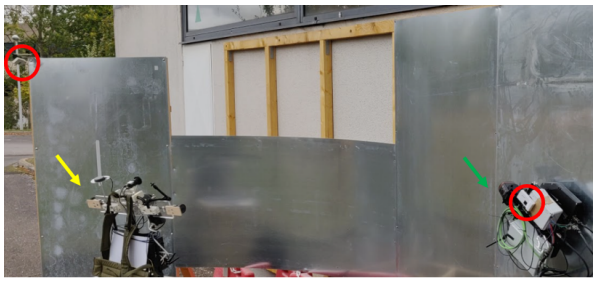


Fig. 2. Magnetic Crawler Robot (green arrow) on a Curved Metal Surface with Ultra-wideband Localization (red circles) and laser (yellow arrow) to track the robot for ground truth.

to localize with respect to could be temporarily installed on the side of a ship hull. Fig. 1 shows a simulation of a ship hull with a magnetic crawler robot and four UWB beacons in place to form a localization grid.

Two main factors to consider when developing a filter for a problem like this are its accuracy and consistency. It can be difficult to maintain accuracy when using UWB for metal structure inspection due to high noise from wave deflection off the metal surface. This error causes a prolonged time of flight resulting in over-exaggerated range measurements. Some propose including methods of detecting these divergences by analysing the noise distribution to decide if a measurement is usable [13]. Others suggest loosely or tightly coupled filters to resolve the problem [14]. A loosely coupled, two step update of orientation correction followed by position correction can give good results, although it is said that a tightly coupled measurement model, where position and orientation are corrected at the same time can better overcome large positioning errors [15]. Even when using tightly coupled EKFs to achieve higher accuracy, there is still the likelihood of inconsistency in this case due to the aforementioned problem related to the robot's planarity being expressed with six degrees of freedom. This can cause the covariance of the robot's state to become disproportionately small resulting in an overconfidence in the propagation and eventually a divergence to an incorrect solution [16]. As Manifold filters solve this problem, they have proven themselves to be more consistent, and more accurate on average, than other filters [17]. The Invariant filter formulation [1] [2] [3] is proven to solve the aforementioned problems by ensuring the *Log-Linear property of the error*, that is, the independence of the error dynamics from the state estimate. We employed the Invariant filter formulation within a manifold-based space showing that our *Manifold Invariant Extended Kalman Filter (M-IEKF)* results in greater consistency and improved accuracy.

III. Theory

In this section, a general understanding of differential geometry, manifolds, and bivariate b-spline surface representations is introduced.

A. Manifolds

An n-dimensional manifold $\mathcal{M}$ is a topological space $(\mathcal{M}, \Theta)$ with the property that each point $p \in \mathcal{M}$ has a neighborhood that is homeomorphic to the Euclidean space

$\mathbb{R}^n$. Thus, if $\forall\, p \in \mathcal{M}, \exists\, \mathcal{U} \in \Theta \mid \sigma : \mathcal{U} \mapsto \sigma(\mathcal{U}) \subset \mathbb{R}^n$ for which the following conditions hold:

$$\sigma \text{ is invertible, thus } \exists\, \sigma^{-1} : \sigma(\mathcal{U}) \mapsto \mathcal{U} \tag{1}$$

$$\sigma \text{ is continuous} \tag{2}$$

$$\sigma^{-1} \text{ is continuous} \tag{3}$$

Then $(\mathcal{U}, \sigma)$ is called a chart at $(\mathcal{M}, \Theta)$ and $\sigma : \mathcal{U} \mapsto \sigma(\mathcal{U}) \subset \mathbb{R}^n$ is called a chart map.

Although there are different classifications of manifolds, differentiable manifolds are of primary focus along this work, because this type of manifold allows a globally differentiable tangent space, shown in Fig. 3, to be defined using calculus. For each point $p \in \mathcal{M}$, the tangent space $T_p\mathcal{M}$ is the space formed by the collection of all tangent vector velocities that a curve $\gamma(t)$ passing through p may have. More formal definitions and a more detailed introduction of the tangent space can be found in [18].

B. Surfaces

Considered smooth surfaces embedded in $\mathbb{R}^3$, which in practice would cover almost all encountered vessel surfaces, are 2-Dimensional parallelizable manifolds $\mathcal{M} = \{(x, y, z) \in \mathbb{R}^3 \mid \phi(x, y, z) = 0\}$, where $\phi : \mathbb{R}^3 \to \mathbb{R}$ is a scalar function that imposes a constraint that defines the shape of the surface. A manifold is called parallelizable if there exists a smooth vector field $\{B_1, B_2\}$, such that for every point $p \in \mathcal{M}$, the tangent vectors $\{B_1(p), B_2(p)\}$ provide a basis of the tangent space $T_p\mathcal{M}$ at p. Within these surfaces being considered are explicit surfaces, where one of its variables can be solved for given the constraint imposed by $\phi(x, y, z) = 0$ (e.g. a paraboloid), and implicit surfaces which are described by an implicit equation $\phi(x, y, z)$, where one of its variables cannot be solved for (e.g. a cylinder). However, any given surface embedded in $\mathbb{R}^3$ can always be written in its implicit form $\phi(x, y, z) = 0$, where the zeros of the constraint are the points $p \in \mathcal{M}$ of the surface. Therefore, the basis of the tangent space $T_p\mathcal{M}$ at p, and thus the manifold parallelization, can be defined as follows:

$$\mathbf{V}_1(p) = \begin{bmatrix} 1 & 0 & \mathrm{D}_x\phi(x, y, z) \end{bmatrix}^T \tag{4}$$

$$\mathbf{V}_2(p) = \begin{bmatrix} 0 & 1 & \mathrm{D}_y\phi(x, y, z) \end{bmatrix}^T \tag{5}$$

Although this way of defining the parallelization is perfectly valid, it is not the only admissible one, and, as shown in Fig. 3, one can also choose a parallelization which forms an orthonormal basis of the tangent space $T_p\mathcal{M}$ at p:

$$\mathbf{L}_1(p) = \mathrm{D}_x\phi(x, y, z)\, \mathbf{V}_2(p) - \mathrm{D}_y\phi(x, y, z)\, \mathbf{V}_1(p) \tag{6}$$

$$\mathbf{L}_2(p) = \mathrm{D}_x\phi(x, y, z)\, \mathbf{V}_1(p) + \mathrm{D}_y\phi(x, y, z)\, \mathbf{V}_2(p) \tag{7}$$

$$\mathbf{B}_1(p) = \frac{L_1(p)}{\|L_1(p)\|} \quad \mathbf{B}_2(p) = \frac{L_2(p)}{\|L_2(p)\|} \tag{8}$$

Then, the normal vector to the tangent space $T_p\mathcal{M}$ can be computed at p as follows:

$$\mathbf{L}_3(p) = \begin{bmatrix} \mathrm{D}_x\phi(x, y, z) & \mathrm{D}_y\phi(x, y, z) & -1 \end{bmatrix}^T \tag{9}$$

$$\mathbf{N}(p) = \frac{\mathbf{L}_3(p)}{\|\mathbf{L}_3(p)\|} \tag{10}$$

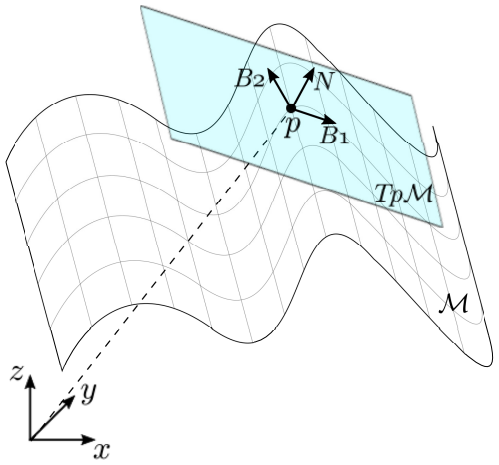


Fig. 3. Illustration of a manifold $\mathcal{M}$, the tangent space $T_p\mathcal{M}$ at $p \in \mathcal{M}$ and its basis vectors $\{B_1(p), B_2(p), N(p)\}$.

Furthermore, for any given surface, or in other words, for any considered manifold, we can choose a chart and hence a continuous, differentiable, and invertible chart map $\sigma : \mathcal{M} \to \mathbb{R}^2$ which maps points from the manifold to a euclidean space of a dimension equal to $dim(\mathcal{M})$.

C. Spline Interpolation

Bivariate b-splines, which are piecewise polynomial functions can fit a variety of complex shapes while maintaining continuity in their derivatives. This surface representation can be evaluated at any point, and being a polynomial, the derivatives are easily obtained, making it sufficient to create a basis for the tangent space so that the manifold properties and constraints can be applied in the state estimation. The surface is defined as follows:

$$f(x, y) = \sum_{i=1}^{k} \sum_{j=1}^{l} B_{xi} B_{yj} c_{ij} \tag{11}$$

The coefficients c_{ij} are determined from the vertices being interpolated. The b-splines B_x and B_y are determined from their endpoints, known as knots, in each respective dimension, for each piecewise polynomial. Then, the coefficients are multiplied by the tensor product of the b-splines resulting in a surface [19].

IV. METHODOLOGY

In this section, the general problem of state estimation for a wheeled robot moving on a smooth surface and a detailed description of the adopted methodology to solve this problem is introduced, followed by the experimental procedure that was carried out. This includes the process of charting the manifold and applying an M-IEKF to a minimal state represented on the product space between the chosen chart and $SO(2)$, or directly on $SE(2)$.

The key to implementing the methodology is to first find a chart that covers the whole manifold being considered, and hence find a continuous, differentiable chart map $\sigma(p)$, and its inverse $\sigma^{-1}(u, v)$.

Let us first consider the easiest case of an explicit, smooth surface described by an explicit function where one of the variables involved is solved for. For example, $z = f(x, y)$.

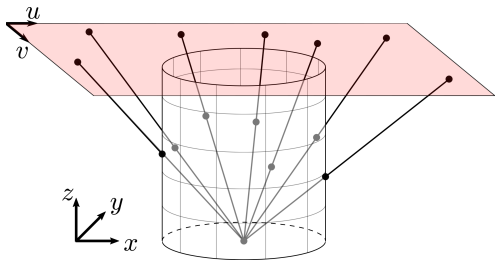


Fig. 4. Illustration of stereographic projection leveraged to define a continuous, differentiable and invertible chart map on the cylinder.

In this case, for every $p \in \mathcal{M}$, the chart map and its inverse are simply determined as follows:

$$\sigma(p) = \sigma(x, y, z) = \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} \tag{12}$$

$$\sigma^{-1}(u, v) = p = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} u \\ v \\ f(u, v) \end{bmatrix} \tag{13}$$

In the most difficult case of implicit, smooth surfaces, a chart, and hence a chart map covering the whole manifold needs to be defined without having a simple and predefined recipe to apply. Consider a cylinder of radius R and height h as a possible manifold to cover with a chart. As a first solution, mapping every point of the cylinder to a plane by unwrapping the cylinder seems logical, however, this solution will result in discontinuities at the border of the map at 2π. Instead, the stereographic projection can be leveraged to find a continuous, differentiable chart map, shown in Fig. 4, and defined as follows:

$$\sigma(p) = \sigma(x, y, z) = \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \frac{xh}{\exp(z)} \\ \frac{yh}{\exp(z)} \end{bmatrix} \tag{14}$$

$$\sigma^{-1}(u, v) = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \frac{Ru}{\sqrt{u^2+v^2}} \\ \frac{Rv}{\sqrt{u^2+v^2}} \\ \log\left(\frac{Rh}{\sqrt{u^2+v^2}}\right) \end{bmatrix} \tag{15}$$

Once a chart covering the manifold has been found, an IEKF is applied on a space which is partially defined by the chosen chart and then lifted back to all the estimated results on the manifold. In order to do so, first, a mapping that allows us to map a velocity vector (or displacement vector) $[\Delta b1\ \Delta b2]^T$ on the tangent space $T_p\mathcal{M}$ at p to a velocity vector (or displacement vector) $[\Delta x\ \Delta y\ \Delta z]^T$ on $\mathbb{R}^3$ must be found. Then, the chosen chart map must be used to project the robot position to the chart. In general, if a wheeled robot is moving on a manifold and $p_k = \{x_k, y_k, z_k\} \in \mathcal{M}$ is the position of the robot at a given time step k, and $[\Delta b_{1k}\ \Delta b_{2k}]^T \in T_p\mathcal{M}$ is the linear displacement vector in the tangent space, then we can compute:

$$\begin{bmatrix} \Delta x_k \\ \Delta y_k \\ \Delta z_k \end{bmatrix} = \begin{bmatrix} B_1(p_k) & B_2(p_k) & N(p_k) \end{bmatrix} \begin{bmatrix} \Delta b_{1k} \\ \Delta b_{2k} \\ 0 \end{bmatrix} \tag{16}$$

The robot position projected on the manifold can then be easily computed through the chosen chart map as follows:

$$\begin{bmatrix} u_k \\ v_k \end{bmatrix} = \sigma(x_k + \Delta x_k, y_k + \Delta y_k, z_k + \Delta z_k) \tag{17}$$

It is important to note that if the velocity vector (or dispacement vector) $[\Delta b1\ \Delta b2]^T$ on the tangent space $T_p\mathcal{M}$ at p is affected by gaussian noise, the linearity of the mapping in Eq. (16) will allow its gaussianity to be preserved.

If a minimal state representation given by $\mathcal{X} = (\mathbf{t}, \mathbf{R}(\theta)) = (u, v, \mathbf{R}(\theta)) \in \mathbb{R}^2 \times SO(2)$ on the product space between the chosen chart and $SO(2)$, where the rotation defined by $SO(2)$ is the rotation of the robot about its own vertical axis, thus its heading, then an IEKF can be designed following algorithm 1.

In the case of the ship hull simulation, the same methodology is applied to a bivariate b-spline representation of the surface. The vertices are extracted from the CAD model of the ship and interpolated. In the case of the real metal plate experiment, the vertices are taken from a laser scan of the surface before the experiment is carried out, and the point cloud is interpolated. Fig. 2 shows the magnetic crawler robot attached to the curved metal surface that was used, with a UWB beacon attached to it (a tag), and another in the corner (an anchor). Only one anchor is shown, but In total there were four. The laser was also used during the experiment to track the robot for ground truth. The robot collects four tag-to-anchor ranges at a time and uses trilateration to compute its position as a measurement in the update function of the M-IEKF algorithm.

V. EXPERIMENTS

A. Evaluation

In this section, the performance of the M-IEKF is evaluated first by testing it on a cylindrical manifold to show its ability to work with any surface that is a parallelizable manifold and to simulate the case of a cylindrical vessel. The M-IEKF is then compared to a standard filter (MC-EKF) that uses two virtual, zero noise measurements to keep the state constrained on the curved surface. Moreover, as a proof of concept for metal structure inspection, we have tested the M-IEKF on a simulated ship hull showing that the proposed methodology can handle the case of a priori not-known surface obtained by bivariate b-spline intepolation from known points on the surface. Finally, the real world viability of the M-IEKF in metal structure inspection is shown with an experiement employing a magnetic wheeled crawler robot on a curved metal surface. In this last experiment, the triangulated position of the robot was availabe via UWB measurements.

For the two simulated tests, a Monte-Carlo simulation of $N = 100$ trials was run. We computed the *Root Mean Squared Error (RMSE)* in position and orientation, furthermore, the *Average Normalized Estimation Error Squared (ANEES)* were computed for each time step, averaged over the N trials, and compared between the two filters. The aforementioned metrics are defined as follows:

Algorithm 1: IEKF on the product space $\mathbb{R}^2 \times SO(2)$

Input: $\hat{\mathcal{X}}^+_{k-1}, \hat{\mathbf{P}}^+_{k-1}, \boldsymbol{\Delta}\mathbf{b}_k, \Delta\theta_k, \mathbf{y}_k$

Propagation

// Map estimate onto $\mathcal{M}$

$\hat{p}^+_{k-1} = \sigma^{-1}\left(\hat{\mathbf{t}}^+_{k-1}\right)$

// Robot rotation $\mathbf{C}_{k-1}$ in $\mathbb{R}^3$

$\mathbf{B}_{k-1} = \left[B_1\left(p_{k-1}\right) \quad B_2\left(p_{k-1}\right) \quad N\left(p_{k-1}\right)\right]$

$$\mathbf{C}_{k-1} = \mathbf{B}_{k-1}\begin{bmatrix}\mathbf{R}\left(\theta_{k-1}\right) & \mathbf{0} \\ \mathbf{0}^T & 1\end{bmatrix}$$

// Map deltas from $T_p\mathcal{M}$ to $\mathbb{R}^3$

$\boldsymbol{\Delta}\mathbf{p}_k = \mathbf{C}_{k-1}\boldsymbol{\Delta}\mathbf{b}_k$

// Projection onto the chart

$\hat{\mathbf{t}}^-_k = \sigma\left(\hat{p}^+_{k-1} + \boldsymbol{\Delta}\mathbf{p}_k\right)$

// IEKF rotation propagation

$\mathbf{R}\left(\hat{\theta}^-_k\right) = \mathbf{R}\left(\hat{\theta}^+_{k-1}\right) Exp\left(\Delta\theta_k\right)$

// Jacobians

$$\mathbf{F}_k = \left[\begin{array}{c|c} \left.\frac{\partial\sigma\left(\sigma^{-1}\left(\mathbf{t}_{k-1}\right)\right)}{\partial \mathbf{t}_{k-1}}\right|_{\hat{\mathbf{t}}^+_{k-1}} & \\ \hline & 1 \end{array}\right]$$

$$\mathbf{G}_k = \left[\begin{array}{c|c} \left.\frac{\partial\sigma(p_k)}{\partial(p_k)}\right|_{\hat{p}^+_{k-1}+\boldsymbol{\Delta}\mathbf{p}_k} \mathbf{C}_{k-1} & \\ \hline & 1 \end{array}\right]$$

// Covariance propagation

$$\hat{\mathbf{P}}^-_k = \mathbf{F}_k\hat{\mathbf{P}}^+_{k-1}\mathbf{F}^T_k + \mathbf{G}_k\left[\begin{array}{c|c} \boldsymbol{\Sigma}_{\boldsymbol{\Delta}\mathbf{b}_k} & \\ \hline & \sigma^2_{\Delta\theta_k} \end{array}\right]\mathbf{G}^T_k$$

End

Update

// Residual

$\mathbf{r}_k = h\left(\hat{\mathcal{X}}^-_k\right) - \mathbf{y}_k$

// Jacobian

$\mathbf{H}_k = \left.\frac{\partial h(\mathcal{X})}{\partial\mathcal{X}}\right|_{\hat{\mathcal{X}}^+_k}$

// Kalman gain

$\mathbf{K}_k = \hat{\mathbf{P}}^-_k\mathbf{H}^T_k\left(\mathbf{H}_k\hat{\mathbf{P}}^-_k\mathbf{H}^T_k + \boldsymbol{\Sigma}_{\mathbf{y}_k}\right)^{-1}$

// IEKF Update

$\hat{\mathbf{t}}^+_k = \hat{\mathbf{t}}_k + \delta_{\mathbf{t}_k}$

$\mathbf{R}\left(\hat{\theta}^+_k\right) = Exp\left(\delta_{\theta_k}\right)\mathbf{R}\left(\hat{\theta}_k\right)$

$\hat{\mathbf{P}}^+_k = \left(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k\right)\hat{\mathbf{P}}^-_k$

End

Output: $\hat{\mathcal{X}}_k, \hat{\mathbf{P}}_k$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}\mathbf{e}^2_{i_k}}{N}} \tag{18}$$

$$ANEES = \frac{1}{Nm}\sum_{i=1}^{N}\mathbf{e}^T_{i_k}\mathbf{P}^{-1}_{i_k}\mathbf{e}_{i_k} \tag{19}$$

where $\mathbf{e}_{i_k}$ and $\mathbf{P}_{i_k}$ are respectively the estimation error and the error covariance for the i-th run at a given time step k.

The RMSE gives an indication of how far the estimate varies from the ground truth on average, whereas the ANEES, which is normalized by the covariance of the filter

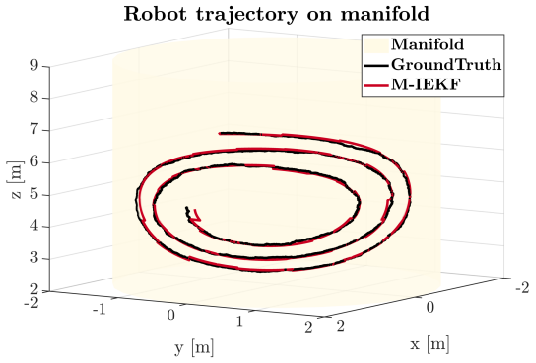


Fig. 5. Ground-truth (in black) and estimated trajectory (in red) of the M-IEKF on a cylindrical surface. Note the wrong initialization of the filter.

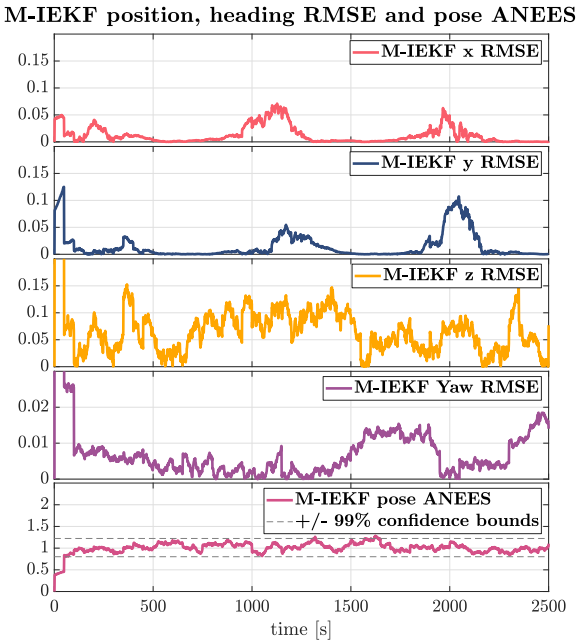


Fig. 6. M-IEKF full state RMSE and ANEES averaged over 100 runs corresponding to the estimation problem on the cylindrical surface.

at each time step, gives a standard for whether a filter is a credible estimator. The closer to 1 an estimator is within the probability interval, the more credible it is, and therefore the more consistency the filter has [20] [21].

B. Results

Fig. 5 and 6 show the trajectory and the error metrics respectively for the M-IEKF during the cylinder manifold simulation. The trajectory plot shows that the state estimate follows closely with the ground truth which is also corroborated by the error metrics. The RMSE for the heading is mostly below $0.01\,\mathrm{rad}$, and the RMSE for its position are predominantly below $10\,\mathrm{cm}$ in each dimension giving a good indication that the filter can perform accurately. Furthermore, the ANEES is almost completely confined to the probability interval, and it is centered about 1 indicating that the filter is credible and consistent. To further evaluate the filter, Fig. 7 and 9 show the trajectory and the error metrics respectively for the M-IEKF and the MC-EKF during the ship hull simulation. The trajectory plot shows that the

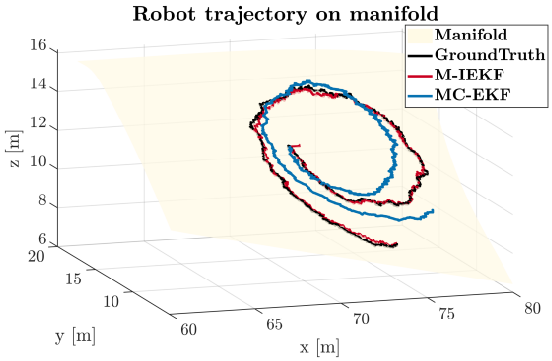


Fig. 7. Ground-truth (in black) and estimated trajectory of the M-IEKF and the MC-EKF (respectively in red and blue) on a b-spline interpolated surface corresponding to the curved surface of a ship hull.

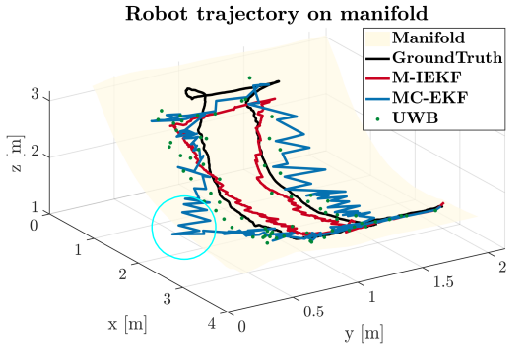


Fig. 8. Ground-truth of the magnetic crawler robot (in black) and estimated trajectory of the M-IEKF and the MC-EKF (respectively in red and blue). Dots (in green) correspond to the position measurements from the UWB trilateration. Note the cyan circle showing the failure of the MC-EKF on providing an estimate that is not attached to the surface.

state estimate of the M-IEKF follows closely with the ground truth like it did in the cylinder experiment, whereas the MC-EKF clearly starts diverging. The error metrics show that the M-IEKF still performs consistently and accurately, but with a little bit more error in comparison with the error in the cylinder simulation which was expected considering that its state is being estimated on an interpolated surface this time. By contrast, the MC-EKF shows significantly higher error in the RMSE for its position up to $50\,\mathrm{cm}$ in some instances in the x direction, and the ANEES plot clearly shows that it goes outside of the probability interval and is therefore not consistent. Fig. 8 shows the trajectory from the real experiment on the curved metal plate for each filter along with the UWB measurements, and Fig. 10 shows the position RMSE for each filter. The M-IEKF follows quite closely to the ground truth, only having noticeable error when there is a high concentration of erroneous UWB measurements due to the metal surface deflection which can be seen near time step 625. The MC-EKF does not follow closely to the ground truth as expected with errors up to $80\,\mathrm{cm}$. The results back up the fact that the M-IEKF is consistent and more accurate than standard approaches like the MC-EKF, allowing further extensions like the inclusion of a measurement update rejection test, making it a viable option for consistent and robust metal structure inspection with ultra-wideband localization.

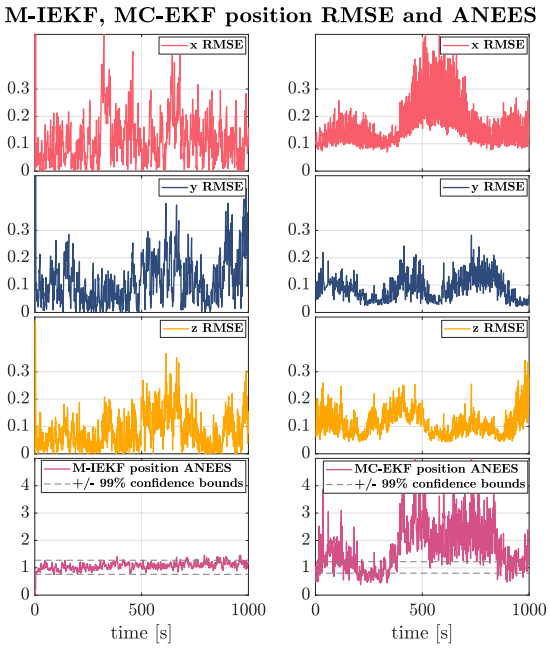


Fig. 9. Comparison between MC-EKF and M-IEK in terms of position RMSE and ANEES corresponding to the case of b-spline interpolated surface.

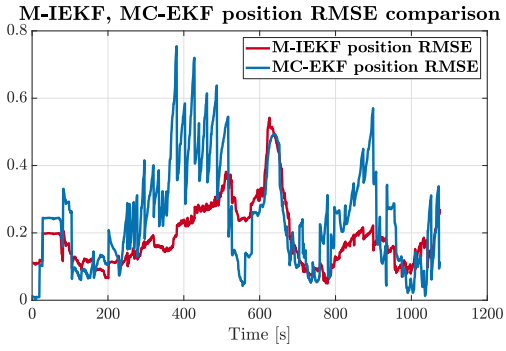


Fig. 10. Position RMSE of the M-IEKF and MC-EKF (in red and blue respectively) corresponding to the real magnetic crawler robot experiment.

VI. CONCLUSION

The *Manifold Invariant Extended Kalman Filter* is a novel approach for consistent state estimation on manifolds. It combines manifold state representation and invariance to achieve greater consistency and accuracy. We proved that the proposed M-IEKF is applicable to a wide range of vessel surfaces encountered in real world applications. Further, we showed results validating that the M-IEKF outperforms classical approaches when using real robot wheel odometry and UWB measurements. Therefore, the M-IEKF makes metal structure inspection with ultra-wideband localization viable.

REFERENCES

[1] A. Barrau and S. Bonnabel, "The invariant extended kalman filter as a stable observer," *IEEE Transactions on Automatic Control*, vol. 62, no. 4, pp. 1797–1812, 2016.

[2] ——, "An ekf-slam algorithm with consistency properties," *arXiv preprint arXiv:1510.06263*, 2015.

[3] ——, "Invariant kalman filtering," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 1, no. 1, pp. 237–257, 2018. [Online]. Available: https://doi.org/10.1146/annurev-control-060117-105010

[4] E. Allak, A. Fornasier, and S. Weiss, "Consistent covariance pre-integration for invariant filters with delayed measurements," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020.

[5] S. Thrun, "Probabilistic robotics," *Communications of the ACM*, vol. 45, no. 3, pp. 52–57, 2002.

[6] M. Mirbach, "A simple surface estimation algorithm for uwb pulse radars based on trilateration," in *2011 IEEE International Conference on Ultra-Wideband (ICUWB)*. IEEE, 2011, pp. 273–277.

[7] D. Gao, A. Li, and J. Fu, "Analysis of positioning performance of uwb system in metal nlos environment," in *2018 Chinese Automation Congress (CAC)*. IEEE, 2018, pp. 600–604.

[8] A. J. Trevor, J. G. Rogers, C. Nieto, and H. I. Christensen, "Applying domain knowledge to slam using virtual measurements," in *2010 IEEE International Conference on Robotics and Automation*. IEEE, 2010, pp. 5389–5394.

[9] S. Jung, S. Song, S. Kim, J. Park, J. Her, K. Roh, and H. Myung, "Toward autonomous bridge inspection: A framework and experimental results," in *2019 16th International Conference on Ubiquitous Robots (UR)*. IEEE, 2019, pp. 208–211.

[10] M. S. B. M. Soberi and M. Z. B. Zakaria, "Autonomous ship hull inspection by omnidirectional path and view," in *2016 IEEE/OES Autonomous Underwater Vehicles (AUV)*. IEEE, 2016, pp. 38–43.

[11] S. Lan, C. Yang, B. Liu, J. Qiu, and A. Denisov, "Indoor real-time multiple moving targets detection and tracking using uwb antenna arrays," in *2015 International Symposium on Antennas and Propagation (ISAP)*. IEEE, 2015, pp. 1–4.

[12] R. Zetik, O. Hirsch, and R. Thoma, "Kalman filter based tracking of moving persons using uwb sensors," in *2009 IEEE MTT-S International Microwave Workshop on Wireless Sensing, Local Positioning, and RFID*. IEEE, 2009, pp. 1–4.

[13] L. Cheng, H. Chang, K. Wang, and Z. Wu, "Real time indoor positioning system for smart grid based on uwb and artificial intelligence techniques," in *2020 IEEE Conference on Technologies for Sustainability (SusTech)*. IEEE, 2020, pp. 1–7.

[14] J. Clemens and K. Schill, "Extended kalman filter with manifold state representation for navigating a maneuverable melting probe," in *2016 19Th International Conference On Information Fusion (FUSION)*. IEEE, 2016, pp. 1789–1796.

[15] H. Benzerrouk and A. Nebylov, "Robust imu/uwb integration for indoor pedestrian navigation," in *2018 25th Saint Petersburg International Conference on Integrated Navigation Systems (ICINS)*. IEEE, 2018, pp. 1–5.

[16] B. B. Ready, "Filtering techniques for pose estimation with applications to unmanned air vehicles," 2012.

[17] M. Brossard, A. Barrau, and S. Bonnabel, "A code for unscented kalman filtering on manifolds (ukf-m)," *arXiv preprint arXiv:2002.00878*, 2020.

[18] P.-A. Absil, R. Mahony, and R. Sepulchre, *Optimization algorithms on matrix manifolds*. Princeton University Press, 2009.

[19] C.-J. Li and R.-H. Wang, "Bivariate cubic spline space and bivariate cubic nurbs surfaces," in *Geometric Modeling and Processing, 2004. Proceedings*. IEEE, 2004, pp. 115–123.

[20] X. R. Li, Z. Zhao, and V. P. Jilkov, "Practical measures and test for credibility of an estimator," in *Proc. Workshop on Estimation, Tracking, and Fusion—A Tribute to Yaakov Bar-Shalom*. Citeseer, 2001, pp. 481–495.

[21] X. R. Li, Z. Zhao, and X. B. Li, "Evaluation of Estimation Algorithms: Credibility Tests," *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans*, vol. 42, no. 1, pp. 147–163, 2012.