COLLABORATIVE VISUAL-INERTIAL LOCALIZATION OF TEAMS WITH FLOORPLAN EXTRACTION

Sándor Gazdag[*†], *Dániel Pásztornicky*[*], *Zsolt Jankó*[*], *Tamás Szirányi*[*†], *András L. Majdik*[*†]

[*] Institute for Computer Science and Control (SZTAKI)
Machine Perception Research Laboratory
H-1111, Budapest, Kende u. 13.-17.
[†] Budapest University of Technology and Economics,
Department of Material Handling and Logistics Systems
H-1111, Budapest, Bertalan Lajos u. 7.

ABSTRACT

This paper showcases a real-world example of a system that achieves collaborative localization and mapping of multiple agents within a building. The proposed system processes the odometry and 3D point cloud data collected by the agents moving around the building to automatically generate the building's floorplan on which the agent trajectories are overlaid. The wearable hardware consists of a low-cost passive integrated sensor that includes both a camera and an IMU (Inertial Measurement Unit) and an embedded compute unit. The system's capabilities are shown through real-world experiments.

Index Terms— collaborative localization, floorplan extraction, visual-inertial localization, human-interpretable map, wearable hardware

1. INTRODUCTION

This paper presents a case study of a wearable system that can accomplish the cooperative positioning and mapping of multiple agents within a building where GNSS (Global Navigation Satellite System) signals are unavailable, by utilizing low-cost passive multimodal sensors, e.g., camera and IMU (Inertial Measurement Unit) to compute the ego-motion of the users. Also, the floorplan of the building is automatically computed by processing the 3D point cloud cooperatively gathered by the users moving around the establishment (warehouse, office, factory, etc.).

The motivation of this work is to provide a tool for hybrid human and robot teams to co-localize in environments where the GNSS has no coverage indoors [1] or has large errors between buildings [2]. Such a system is useful in warehouses where hybrid teams perform work cooperatively, or for example in disaster relief situations where the knowledge of the relative position of rescuers and helping ground and air vehicles is important [3]. Also, this technology can be applied in multi-user virtual or mixed-reality scenarios. Embedded hardware offers the possibility to integrate these capabilities into ever smaller wearable devices, in contrast to active sensors like LiDAR that require significant amounts of power which can prevent their usage in compact systems.

Indoor localization approaches are often based on already existing infrastructure with known transmitter positions like Wi-Fi, UWB (Ultra Wide Band), RFID (Radio Frequency IDentification), Bluetooth, or light (either visible or infrared) [4]. This infrastructure constrains the applicability of such systems. The state-of-the-art commercial solutions for localizing a device without outside infrastructure include the two

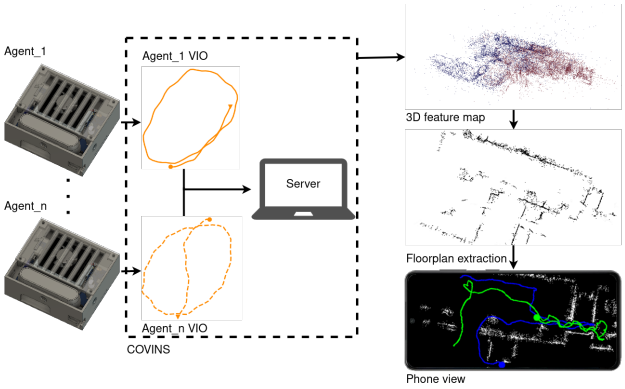

Fig. 1. Visual abstract of the proposed system: individual agents process their VIO (Visual-Inertial Odometry) locally and transfer these to the server which produces a consistent map, extracts the floorplan of the place, and sends it together with agent trajectories to a mobile device for viewing.

This work was supported by the European Union project RRF-2.3.1-21-2022-00004 within the framework of the Artificial Intelligence National Laboratory, by the Hungarian Scientific Research Fund (No. NKFIH OTKA KH-126513), and by the Force Modernization and Transformation Command of the Hungarian Defence Forces. The research reported in this paper is part of project no. SZTAKI-NVA-01, implemented with the support provided by the Ministry of Innovation and Technology of Hungary from the National Research, Development and Innovation Fund, financed under the TKP2021 funding scheme.

biggest smartphone operating system manufacturers Google[1] and Apple[2]. Both of these companies provide developer kits to develop Augmented Reality (AR) and Virtual Reality (VR) applications that can be leveraged to solve the cooperative localization problem with smartphones. However, these solutions are optimized for augmented reality applications often specially confined to a room-scale experience.

There are also different open source approaches for solving the multi-agent localization problem[5], but these solutions lack an easily interpretable map as seen in the top right of Fig. 1. Our approach is built on top of COVINS [6] algorithm, a state-of-the-art cooperative SLAM (Simultaneous Localization And Mapping) system. We extend the original COVINS algorithm with a floorplan extraction step to create a human-interpretable, top-view map of the place. The accuracy of the system is similar to the accuracy of Google ARCore in a room as shown in our experiments and it can work reliably in larger areas like an entire building floor.

The main components of the showcased system are on the visual abstract in Fig. 1. To summarize, the contributions of this paper are as follows:

- We proposed an algorithm that computes and shares a human-interpretable floorplan-like map of the environment in real-time with the locations of the users and their respective trajectories which can be viewed on a phone or any other device.
- We developed a compact wearable hardware prototype for human or robotic agents using embedded hardware that was tested in real-world experiments[3].

2. RELATED WORK

Real-time localization in environments without reliable GNSS signals is a widely researched topic. In [4] the state-of-the-art indoor localization methods are surveyed. In the paper different device-based, monitor-based, and proximity detection methods are compared based on localization techniques, such as angle of arrival (AoA), time of flight (ToF), return time of flight (RTOF), and received signal strength (RSS), and on technologies, such as Wi-Fi, RFID, UWB, and Bluetooth. A similar compilation is presented in [7], where the state-of-the-art relative localization methods for robot swarms are surveyed.

SpotFi [8] is an indoor localization system using commercial Wi-Fi chips without special hardware or firmware. The algorithm can achieve a median accuracy of 40 cm by using super-resolution algorithms that can accurately compute the angle of arrival of the signal from the localized device. The localization is solved with multiple Wi-Fi access points with known positions.

In [9] a system using UWB to accurately localize robots is proposed. It uses 4 ceiling-mounted reference nodes and the time-difference-of-arrival technique to achieve a localization RMS error of 15 cm. There are also commercial systems using UWB nodes available such as Ubisense[4]. Common in all these localization techniques is that they rely on previously installed infrastructure as reference nodes for the localization.

Both Google's ARCore and Apple's ARKit are software developer kits to create AR applications. These provide motion tracking functions that rely on the phones' sensors and also functions such as Google's Cloud Anchor for building multi-user VR experiences. These are closed-source solutions mostly for smartphones. They do not provide extensive mapping capabilities and also perform most of the processing in the cloud which is an important constraint in many applications.

SLAM is the research field in robotics for localization in previously unknown environments. Collaborative or multi-agent SLAM builds a consistent map and localizes multiple agents in it. In [5] and [10] the state of collaborative SLAMs is surveyed. At this point there are multiple open-source solutions for centralized collaborative SLAM, meaning the agents communicate with a server that performs the global optimization of the map. Visual collaborative SLAM algorithms include [11] and [12].

CORB2I-SLAM [13] and COVINS [6] both propose centralized visual-inertial collaborative SLAM algorithms built on either ORB-SLAM2 [14] in the case of CORB2I-SLAM or ORB-SLAM3[15] in the case of COVINS. Both of the methods perform map building and localization with passive sensors, however, the resulting maps lack easy interpretability for humans.

3. ARCHITECTURE

The proposed approach is built on top of COVINS[6] which provides client-server communication and maintains the coherent global mapping between the agents, while every agent computes its ego-motion using the ORB-SLAM3[15] algorithm. The performance of COVINS is detailed in the original paper [6] where it was shown to run with 12 agents and also with fast-moving drones in real-time. Inherently these performance properties apply also to the system presented in this paper.

The client side of the algorithm is installed on NVIDIA Jetson Xavier NX developer boards. The server performs global optimizations when a place recognition is accepted which can either be a loop closure event or a map fusion event as detailed in [6]. This ensures the global consistency of the map and the different agent trajectories inside.

3.1. Floorplan extraction

In the proposed method the global optimized map from COVINS is further processed to achieve an easy-to-interpret

[1] ARCore: `https://developers.google.com/ar`

[2] ARKit: `https://developer.apple.com/arkit`

[3] Video demonstration of the proposed system: `https://youtu.be/RrW0zypa7nA`

[4] `https://ubisense.com/`

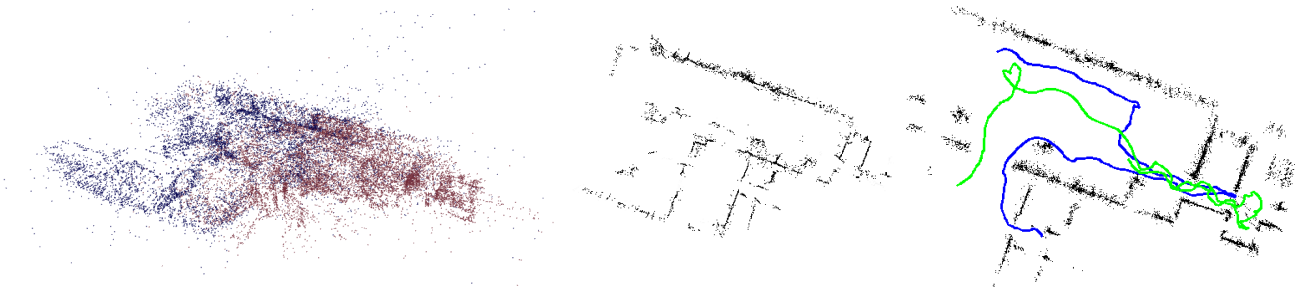

Fig. 2. Steps of the floorplan extraction. Left to right: original 3D feature map; extracted floorplan; zoomed-in map with trajectories.

overview map with floorplan and the trajectories of the different agents. The steps of the floorplan extraction are shown in Fig. 2. The input of the process is the sparse combined 3D feature map of the different agents, seen on the left of Fig. 2 where the yellow and blue colors denote the two agents which recorded the points.

First, a statistical outlier removal is applied to the point cloud to remove the outliers which can be seen all around the first image of Fig. 2. The mean and standard deviation of the filter were set empirically.

After the outlier removal, the floor and ceiling planes are removed so the outline of the walls will become apparent. Various methods, such as plane fitting with RANdom SAmple Consensus (RANSAC) were tested but by using them other horizontal planes, such as tables were also found. Thus we decided to follow another approach: the floor and ceiling planes were extracted by a distribution cut. The points are grouped by their respective height into a histogram with a set bin size and the algorithm searches for local maxima in this height histogram. The floor plane should be the first and the ceiling plane should be the last local maximum. A bin is considered to be a local maximum if it has the maximum number of points in the immediate surrounding bins and more than the average number per bin. Then the top and bottom of the point cloud are simply cut along horizontal planes. These planes are over the average height of the floor bin and under the average height of the ceiling bin by the standard deviation in that bin. Our approach assumes a single floor scenario but the distribution cut can be generalized to multiple floors.

Next, another statistical outlier filter is applied to delete the remaining outliers. The resulting floorplan can be seen in the second image of Fig. 2.

Finally, an image is created by the orthographic projection of the remaining points to the horizontal plane, the last few seconds (this is a viewing parameter) of the trajectories are plotted on top and the region of interest around the trajectories is cropped as seen in the right image of Fig. 2. This is saved and published to a simple website using an Apache web server and can be viewed on a phone or any other device.

3.2. Hardware

The goal was to create a small device for the agents that people can wear and can also be mounted on robots. A lot of preparation, work, and hardware testing went into the final system so a short list and considerations are presented here.

3.2.1. Client

The Auvidea JNX30D developer board for NVIDIA Jetson Xavier NX was chosen as the compute module for the client because it has considerable hardware resources while requiring low power. It can flawlessly run the ORB-SLAM3 client and has a GPU that can later be used for better environment understanding using CNNs. We power it using a 20 000 mAh Xiaomi power bank with 50W maximal output. It has a USB 3.0 port for the camera and another 2.0 port that can be soldered on for the Wi-Fi adapter.

The Intel RealSense D435i was chosen as the camera and IMU for the client. Although only one IR camera is used from this active stereo camera, the integration of the IMU sensor, the low cost, and the readily available ROS package make it the best candidate for our design choice. The IR camera has the additional advantage of working great in lowlight scenarios. The IMU error and noise parameters were characterized using the Allan Variance method[5] and the cameras and IMU extrinsic parameters were calibrated using Kalibr [16].

The parts of the client system are secured and encapsulated in a 3D printed box as in the left of Fig. 1.

3.2.2. Server and network

We use an ASUS laptop with AMD Ryzen 9 5900hx and 64 GB RAM as the server which is more than sufficient for 3 agent scenarios.

For the network, an ASUS Wireless AC2400 Router was used which provides a pretty good range in a small package with easy to configure interface. TODO about 80 meters or a building floor inside our office space.

4. RESULTS

Two main tests were carried out in the real world. These tests were recorded in the basement of our institution. For details,

[5] https://github.com/ori-drs/allan_variance_ros

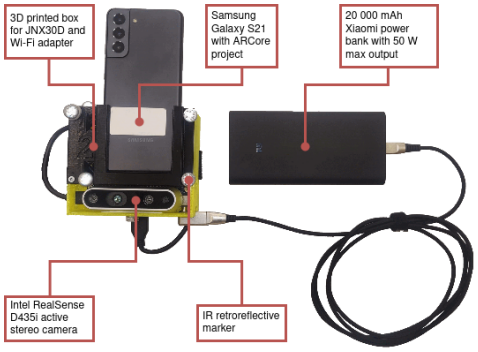

Fig. 3. Image of the different hardware components used in the trajectory comparison experiment.

kindly check the video attachment here: `https://youtu.be/RrW0zypa7nA`.

In the first experiment, the agents initialized at the same place and traversed the basement along different trajectories. Note that the trajectories on the COVINS map update constantly when a global optimization is performed, and the 3D map is becoming ever more cluttered but the mobile map is easy to interpret throughout.

In the second experiment, the robustness of the place recognition pipeline was tested by starting all agents without scene overlap between their camera views. Note that right after all the maps are fused the mobile map appears. Also, the map and the trajectories are very similar in quality to the first experiment.

The system performed well and robustly in both cases and the floorplan is much easier to interpret than the original COVINS map.

4.1. Trajectory accuracy

4.1.1. Solution with ARCore

We built a unity application for Android to compare our approach based on COVINS. The application uses Google's ARCore developer kit and specifically the Cloud Anchor concept. This is a method to provide shared AR experiences for multiple users. In our case, we used it to get the relative device locations and benchmark COVINS against the trajectories.

Cloud Anchors are local feature clouds around a user-specified 3D point, collected by one phone and then shared online through the Cloud Anchor API with other phones. In our case, we place a Cloud Anchor by one of the phones and then resolve it by the others to have a common coordinate system. After that, the 6DoF pose of each phone in this common coordinate system is known, and an overview map is created with the anchor and the phone poses.

This works great in one room but each user defines its location in relation to the anchor, and no inter-user matches and loop closures are used for global optimization which significantly worsens the trajectory further away from the anchor.

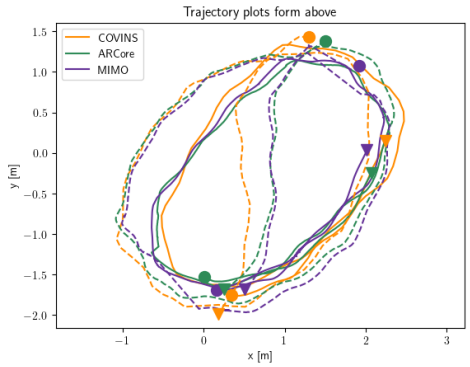

Fig. 4. Trajectory overview of the two solutions and GT. Note that there are no significant differences in terms of localization error between the COVINS and ARCore based solutions.

4.1.2. Trajectory comparision

The front of the 3D-printed client box was updated to hold a phone as seen in Fig. 3. With this change, a qualitative comparison could be carried out by running the ARCore and COVINS methods at the same time on the same rigid body.

Ground Truth (GT) trajectories were also recorded with the SZTAKI MIcro aerial vehicle and MOtion capture (MIMO) system[17]. The MIMO arena uses an OptiTrack motion capture system to provide sub-millimeter accurate, 240 Hz tracking data for bodies with IR retroreflective markers.

Two markered-up agents traversed the MIMO arena along two different trajectories while recording both COVINS, ARCore, and GT data. The recordings were synchronized by saving each trajectory pose with UNIX timestamps. This lacks the synchronization accuracy for quantitative measurements, however, it shows that the proposed method performs similarly to ARCore in small spaces as seen in 4.

The COVINS and ARCore data is recorded in an arbitrary coordinate system while the GT data was recorded in the coordinate system of the arena. The trajectories were aligned by calculating the rigid body transformation (SE3) between them. For each COVINS and ARCore trajectory position, a corresponding GT trajectory position was determined, choosing the closest measurement in time and the transformation was calculated by least square fitting of the point sets[18].

The aligned trajectories are plotted in 2D from above in Fig. 4. There are 6 trajectories all starting from a filled circle and terminating in a filled triangle. The trajectory of the first agent is denoted with solid lines while the second agent is denoted with dashed lines. The colors represent the recording method as seen in the legend. Note, that both of the calculated trajectories are similar in shape and similarly close to the GT.

In conclusion, our method based on COVINS can achieve similar accuracy to ARCore in small spaces. As demonstrated in the previous subsection, it can also robustly localize teams in much larger environments that contain multiple rooms with easily interpretable floorplan extraction.

5. REFERENCES

[1] Marko Modsching and R. Kramer, "Field trial on gps accuracy in a medium size city: The influence of built-up," 2006.

[2] Jinyong Jeong, Younggun Cho, Young-Sik Shin, Hyunchul Roh, and Ayoung Kim, "Complex urban dataset with multi-level sensors from highly diverse urban environments," *The International Journal of Robotics Research*, vol. 38, no. 6, 2019.

[3] Chang Liu and Tamas Sziranyi, "Road condition detection and emergency rescue recognition using on-board UAV in the wildness," *Remote Sensing*, vol. 14, pp. 4355, 2022.

[4] Faheem Zafari, Athanasios Gkelias, and Kin K. Leung, "A survey of indoor localization systems and technologies," *IEEE Communications Surveys Tutorials*, vol. 21, no. 3, pp. 2568–2599, 2019.

[5] Danping Zou, Ping Tan, and Wenxian Yu, "Collaborative visual slam for multiple agents:a brief survey," *Virtual Reality Intelligent Hardware*, vol. 1, no. 5, pp. 461–482, 2019, 3D Vision.

[6] Patrik Schmuck, Thomas Ziegler, Marco Karrer, Jonathan Perraudin, and Margarita Chli, "Covins: Visual-inertial slam for centralized collaboration," in *2021 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, 2021, pp. 171–176.

[7] Siyuan Chen, Dong Yin, and Yifeng Niu, "A survey of robot swarmsrsquo; relative localization method," *Sensors*, vol. 22, no. 12, 2022.

[8] Manikanta Kotaru, Kiran Joshi, Dinesh Bharadia, and Sachin Katti, "Spotfi: Decimeter level localization using wifi," *SIGCOMM Comput. Commun. Rev.*, vol. 45, no. 4, pp. 269–282, aug 2015.

[9] Sivanand Krishnan, Pankaj Sharma, Zhang Guoping, and Ong Hwee Woon, "A uwb based localization system for indoor robot navigation," in *2007 IEEE International Conference on Ultra-Wideband*, 2007, pp. 77–82.

[10] Pierre-Yves Lajoie, Benjamin Ramtoula, Fang Wu, and Giovanni Beltrame, "Towards collaborative simultaneous localization and mapping: a survey of the current research landscape," *Field Robotics*, vol. 2, no. 1, pp. 971–1000, mar 2022.

[11] Robert Castle, Georg Klein, and David W. Murray, "Video-rate localization in multiple maps for wearable augmented reality," in *2008 12th IEEE International Symposium on Wearable Computers*, 2008, pp. 15–22.

[12] Patrik Schmuck and Margarita Chli, "CCM-SLAM: Robust and efficient centralized collaborative monocular simultaneous localization and mapping for robotic teams," in *Journal of Field Robotics (JFR)*, 2018.

[13] Arindam Saha, Bibhas Chandra Dhara, Saiyed Umer, Ahmad Ali AlZubi, Jazem Mutared Alanazi, and Kulakov Yurii, "Corb2i-slam: An adaptive collaborative visual-inertial slam for multiple robots," *Electronics*, vol. 11, no. 18, 2022.

[14] Raúl Mur-Artal and Juan D. Tardós, "ORB-SLAM2: an open-source SLAM system for monocular, stereo and RGB-D cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[15] Carlos Campos, Richard Elvira, Juan J. Gomez Rodriguez, Jose M. M. Montiel, and Juan D. Tardos, "ORB-SLAM3: An accurate open-source library for visual, visual–inertial, and multimap SLAM," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, dec 2021.

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

[17] Sandor Gazdag, Albert Kiskaroly, Tamas Sziranyi, and Andras L. Majdik, "Autonomous racing of micro air vehicles and their visual tracking within the micro aerial vehicle and motion capture (mimo) arena," in *ISR Europe 2022; 54th International Symposium on Robotics*, 2022, pp. 1–8.

[18] K. S. Arun, T. S. Huang, and S. D. Blostein, "Least-squares fitting of two 3-d point sets," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-9, no. 5, pp. 698–700, 1987.