Collaborative Navigation for Flying and Walking Robots

Conference Paper

Author(s):
Fankhauser, Péter; Bloesch, Michael; Krüsi, Philipp; Diethelm, Remo; Wermelinger, Martin; Schneider, Thomas; Dymczyk, Marcin; Hutter, Marco; Siegwart, Roland

Publication date:
2016

Permanent link:
https://doi.org/10.3929/ethz-a-010687710

Originally published in:
https://doi.org/10.1109/IROS.2016.7759443

Collaborative Navigation for Flying and Walking Robots

Péter Fankhauser[1,2], Michael Bloesch[1], Philipp Krüsi[1], Remo Diethelm[1], Martin Wermelinger[2],
Thomas Schneider[1], Marcin Dymczyk[1], Marco Hutter[2], Roland Siegwart[1]

Abstract— **Flying and walking robots can use their complementary features in terms of viewpoint and payload capability to the best in a heterogeneous team. To this end, we present our online collaborative navigation framework for unknown and challenging terrain. The method leverages the flying robot's onboard monocular camera to create both a map of visual features for simultaneous localization and mapping and a dense representation of the environment as an elevation map. This shared knowledge from the flying platform enables the walking robot to localize itself against the global map, and plan a global path to the goal by interpreting the elevation map in terms of traversability. While following the planned path, the absolute pose corrections are fused with the legged state estimation and the elevation map is continuously updated with distance measurements from an onboard laser range sensor. This allows the legged robot to safely navigate towards the goal while taking into account any changes in the environment. In this setup, our approach is independent of external localization, relative observations between the robots, and does not require an initial guess about the pose of the robots. The presented methods are fully integrated and we demonstrate their capabilities in an experiment with a hexacopter and a quadrupedal robot.**

I. Introduction

Flying and ground vehicles, such as walking robots, possess complementary beneficial properties for autonomous navigation tasks. Flying robots can quickly cover large areas and have a favorable viewpoint for situational assessment. On the other hand, a ground vehicle has longer autonomy, can carry substantial payload, and can actively interact with the environment. Working with a heterogeneous team of flying and ground vehicles opens new possibilities to combine these unique features. In this paper, we present our work on a fully integrated approach on collaborative navigation for flying and walking robots in unknown and challenging terrain. We are interested in particular in legged robots, because of their superior capability to traverse challenging terrain which can be found for example in search and rescue scenarios. Our approach is to have the (faster) flying robot explore and map the environment (see Fig. 1), while the (slower) walking robot utilizes the shared data to localize itself and plan a global path to the goal position.

In our work, we introduce a range of methods which interact as follows: A flying robot (e.g. a hexacopter or similar) is guided to fly over an unknown area of interest in order to collect visual data with a monocular camera. The visual data is used to create a global map of visual features (landmarks) and simultaneously localize itself against it (SLAM). Furthermore, the collected monocular image stream is used to create depth images with help of a semi-dense reconstruction pipeline. These depth-images are used to generate a probabilistic elevation map representation of the terrain. The information about the terrain profile is leveraged by the walking robot to plan a global path to a desired goal position. This is achieved by interpreting the elevation map as a traversability map, judging the easiness/safety to walk over a certain region. While following the planned path with a walking gait, the pose of the robot relative to the global landmark map is tracked with an onboard camera. The absolute pose references and the high-rate data from joint states and Inertial Measurement Unit (IMU) are fused in the legged state estimator providing accurate, fast, and drift-free pose estimations for the locomotion controller. As the legged robot moves through the environment, the original elevation and traversability maps are continuously updated with help of a laser range sensor. The walking robot performs continuous replanning of the path to account for the new data acquired in the map. This allows the walking robot to navigate through a previously unknown environment while preventing collision with unforeseen obstacles.

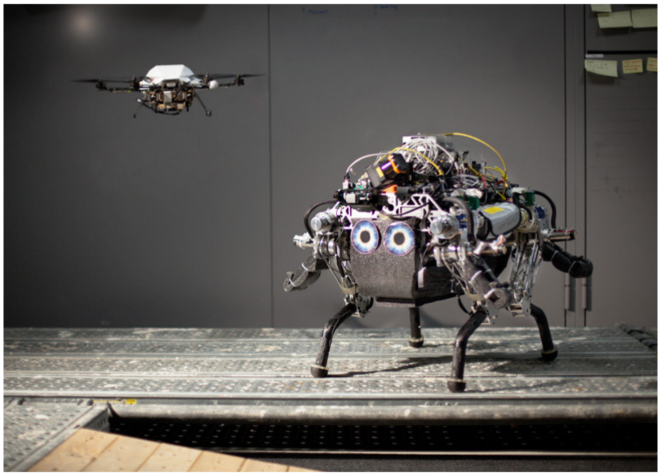

Fig. 1: A heterogeneous team consisting of a flying and a legged robot is used to navigate in rough terrain: `https://youtu.be/9PprNdIKRaw`.

To the best of our knowledge, the only work that focuses on a collaboration of a legged and a flying robot for navigation was discussed in [1]. However, it was limited to the application of receiving an alternative camera viewpoint by visually servoing a quadrotor with attached markers.

Prior work in collaboration between aerial and ground robots often addressed the navigation problem in large-scale outdoor environments [2, 3, 4, 5, 6]. These approaches rely on the (at least partial) availability of satellite-based localization, which precludes their application in indoor environments and GPS-denied outdoor areas such as dense

[1] Autonomous Systems Lab (ASL), ETH Zurich (pfankhauser@ethz.ch)

[2] Robotic Systems Lab (RSL), ETH Zurich

The research leading to these results has received funding from Google's project Tango. This work was supported in part by the Swiss National Science Foundation (SNF) through project 200021_149427 / 1, the National Centre of Competence in Research Robotics, and the armasuisse S+T UGV research program.

forests or narrow urban canyons. Alternatively, several approaches have been proposed for GPS-independent *relative* localization of aerial and ground robots. A flying robot is tracked using a sensor mounted on a ground robot [7], or vice versa [8, 9], while mapping is accomplished by only one of the agents. The two robots are required to stay close to each other at all times, since the tracked agent can only be localized as long as it is in the field-of-view of the other robot. In a navigation scenario, this prevents the aerial robot from exploring areas further away from the ground robot, which is essential for efficient global guidance. Moreover, tracking-based systems may not easily be scalable to groups of more than two robots.

Our localization and mapping system is independent of both external localization (such as GPS) and relative observations between the robots. In contrast, the robots build and localize within a single map of visual features using on-board cameras. Hence, the system is readily extendable to groups of more than two robots. The approaches of Michael et al. [10] and Forster et al. [11] are conceptually similar to ours, yet based on geometric registration, which is known to be sensitive to initial guesses. We experimentally demonstrate that our feature-based visual localization and mapping framework enables reliable operation of both aerial and ground robots using a single map — despite the strongly varying viewpoints of the different agents, and without requiring prior information about the relative poses of the robots.

Moreover, unlike [10] and [11], we integrate localization and mapping with a complete system for autonomous navigation of the ground robot, comprising collaborative terrain mapping using both cameras and laser scanners (multi-modal mapping), as well as traversability estimation, path planning, and motion control. To this end, we also integrate the high-level localization and mapping framework into our high-bandwidth but drift-affected Extended Kalman Filter-based (EKF) state estimation fusing kinematics and inertial measurements. Together, this provides the control and planning framework with an accurate, drift-free, global state estimation, with sufficient bandwidth such that the legged robot can stabilize itself and precisely navigate through the environment.

In this work, we contribute with the design and implementation of a navigation framework for ground robots in rough terrain that is enhanced by the help of flying robots. The framework is designed for a collaborative navigation in the sense that data is actively shared by the robots in order to increase the planning breadth and thus enhance the capability to navigate in previously unknown terrain. The proposed methods comprise all aspects of navigation such as localization, mapping, map interpretation, path planning, state estimation, and control in a tightly coupled manner. To this end, we present an experimental evaluation showcasing the advantages of our approach.

The remainder of the paper is structured as follows. Section II introduces the different methods applied in the navigation framework and describes their interrelation. Section III presents the experimental setup and evaluation of the proposed framework. In Section IV, we extend the sequential deployment of the flying and walking robot with a concurrent operation of the platforms. Section V concludes our work.

II. Methods

In the following, we describe the methods that we have developed for the collaborative navigation of flying and walking robots. Although our approach can be used for multiple flying and walking robots in the same mission, we will assume that the tasks involve one robot of each type (see Fig. 1).

For the walking robot, we work with *StarlETH* [12], an electrically actuated quadrupedal robot of the length of ~1 m. All joints are driven by series elastic actuators (SEA) enabling the execution of various gaits that are robust against variations of the ground. The flying robot is an *Asctec Firefly Hexacopter*. Both machines can operate self-autonomously as they have onboard batteries and computers. Each robot has a network for internal data transfer and the robots can share information through a dedicated WLAN network.

For global localization and mapping, the flying and walking robots are equipped with a VI-Sensor [13] containing two global-shutter, wide-VGA 1/3 inch cameras in a front-parallel stereo configuration. For this work, the stereo configuration is not made use of (although could readily be used) because a single camera is sufficient and the adaptive/larger baseline can provide better accuracy. The mounted lenses have a diagonal field of view of 120°. The intrinsics and extrinsics of the sensor are factory calibrated for a standard pinhole camera model and a radial-tangential distortion model. StarlETH is additionally equipped with a front-facing rotating Hokuyo laser range sensor.

Fig. 2 provides an overview of the system and the workflow of the involved components.

A. Localization and Mapping

For achieving a proper collaboration between flying and legged robots, it is essential to localize both robots within a common global frame of reference. To this end, we propose the use of a framework based on [14]. The idea is to provide a statistically consistent interface between visual-inertial odometry and a mapping backend. Both parts rely on the extraction of visual point features and corresponding descriptors.

The mapping backend merges the visual-inertial data from multiple independent robot trajectories into a single global map representation. Each trajectory is composed of a graph of visual-inertial keyframes and a set of 3D landmarks. The map merging process establishes landmark correspondences between the different trajetories [15] based on visual appearance-matching. Subsequently, it performs a joint visual-inertial bundle-adjustment in order to achieve a globally consistent map (global localization map). Finally, landmarks with low information for localization are pruned from the map [16] (compression step) to obtain a reduced set of landmarks which can be used by the online localization.

The visual-inertial localization is based on a non-linear fixed-lag smoother. It is able to track its own set of *SLAM landmarks* and simultaneously integrate global localization information by tracking the *map landmarks* provided by the backend. This is enabled by co-estimating the relative transformation between the global frame of reference, in which the map landmarks are represented, and the odometry

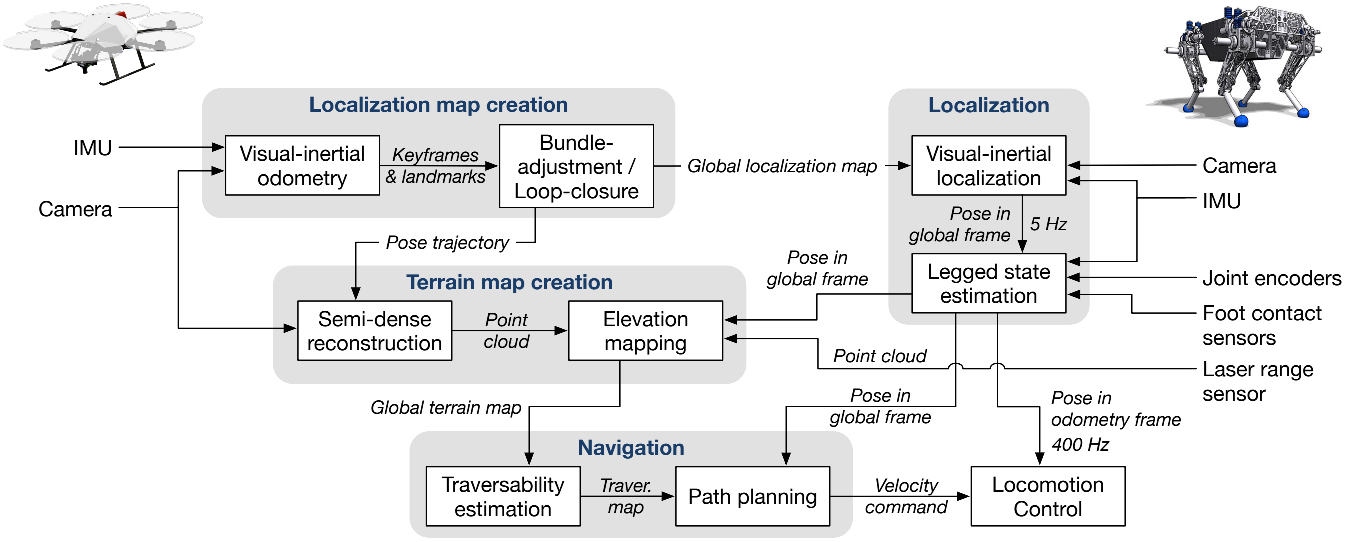

Fig. 2: Overview of the collaborative navigation framework: The hexacopter's onboard IMU and camera are used to localize, construct a global sparse map of landmarks (global localization map) and to provide a semi-dense reconstruction of the surroundings. The global localization map is used by the legged robot to localize itself and is fused in the state estimation with data from IMU and leg kinematics. A joint terrain map is created from visual data (flying robot) and laser data (legged robot). Finally, control and navigation of the legged robot rely on the state estimation and traversability map to move through previously unseen rough terrain.

frame of reference for local SLAM landmarks. Only a fixed-number of keyframes is kept within the optimization window to keep the computational complexity tractable. Local SLAM landmarks get marginalized once the corresponding keyframe leaves the temporal window. This approach enables accurate visual-inertial localization with respect to a global reference frame. At each localization step, the SLAM landmarks are queried in the database of the global map landmarks to establish the feature matches. This means that *no* initial guess about the pose of the walking robot is required.

B. Legged State Estimation

Locomotion with force-controlled legged robots requires permanent motion stabilization using an on-board feedback controller which computes corrective joint torques based on the current estimation of the robot's pose and velocity (e.g. [12]). The discussed localization and mapping framework provides an accurate pose estimation (see [14] for an evaluation), but the obtained bandwidth (5 Hz) is insufficient for the feedback controller for stabilizing the system.

In order to overcome the above shortcoming, an EKF-based state estimator is implemented on the legged robot as presented in [17]. This filter fuses the measurements from the dedicated on-board IMU with data it receives from the joint encoders at a frequency of 400 Hz in order to guarantee a high bandwidth feedback. While the local egomotion can be estimated accurately, this filter is prone to drift in the position and the yaw angle (due to the IMU roll and pitch are fully observable) [17].

To overcome this limitation, we extend the existing on-board legged state estimation by enabling the incorporation of external 6 Degree of Freedom (DoF) pose measurements between an arbitrary inertial coordinate frame (not necessarily gravity aligned) and a body-fixed coordinate frame. The motivation is twofold. First, the additional pose measurements improve the quality of the legged state estimator by limiting the drift of the unobservable states and by facilitating the estimation of online calibration parameters like IMU biases. Second, it provides the controller and planner processes with a unified, high accuracy, high bandwidth, gravity aligned estimate of the robot pose and its derivatives.

Directly fusing the external 6 DoF pose measurements with the internally estimated pose can lead to inconsistencies, especially if the frame of reference of the external pose is not properly aligned with gravity. Thus, we decouple the system by introducing two inertial frames, one for the 'internal' fusion of kinematics and IMU which is always aligned with gravity, and one for the reference for the 'external' pose measurements. The filter state is then augmented with the transformation between both inertial frames, which is modeled as a random walk process with small covariance parameter. This so-called *loose* coupling of both frameworks yields a state estimation which exhibits a high accuracy and drift-free localization with respect to the global map while ensuring a sufficiently high bandwidth. All derivatives are always represented w.r.t. the body frame (robo-centric formulation) and taken w.r.t. the 'internal' odometry frame in order to avoid discontinuities.

C. Elevation Mapping

We choose an elevation map representation to approximate the geometry of the terrain. In this model, a two-dimensional regular grid covers the mission area and stores for each cell the height of the terrain at that position. Although different terrain representations exists, elevation maps have the advantage that they can accurately capture the terrain profile while allowing for simple and efficient data handling.

The elevation map is generated by populating the grid cells with range measurements and fusing new measurements with existing data in the map. For the entire process, one elevation map is maintained and updated with data from the different robots. The range measurements originate either from estimating the image depth from the camera motion of the flying robot, or directly from the distance measurements from the rotating laser range sensor on the walking robot. In both cases, we are interested in the terrain profile below the robot and remove all distance measurements which are above the robot's current base.

For the flying robot, we employed a semi-dense reconstruction pipeline. Assuming an accurate estimation of the camera trajectory, the pipeline performs a patch warping and alignment between selected camera frames and thereby estimates the corresponding depth. Given a frame of interest, a matching frame is heuristically selected in order to allow a good estimation of the disparity. The baseline of the selected camera frames is adapted to the mean scene distance which improves depth estimation for varying flying altitudes of the robot.

For all data generated by the flying and the walking robot, we use the elevation mapping method presented in [18] to probabilistically fuse multiple measurements that fall into the same cell. To this end, we extend the elevation map representation with a height variance information for each cell. The noise of the sensor measurements is approximated with the measurement errors models from [19] for the laser range sensor and an error model based on [20] for the depth estimation from the camera. With these models, the sensor noise in direction of the measurement and lateral direction is expressed as a quadratic function of the measurement distance. With the knowledge of the current pose of the robot, the sensor noise is propagated to the corresponding elevation measurement noise. A recursive filter fuses the received height measurement to estimate the height and variance for each cell. We use the Mahalanobis distance to reject measurements that fall below the highest elevation measurement. This is important to correctly reflect the height of objects such as boxes and tables, where multiple heights fall into the same cell. Note that because in this work the robots are able to localize themselves accurately and drift-free, we do not have to propagate the robot pose uncertainty into the elevation map in contrast to [18].

D. Traversability Estimation

Finding a safe path for the walking robot through the environment requires to judge the suitability of the terrain for safe locomotion. To this end, we interpret the generated elevation map in terms of *traversability*, a local measure expressing the safety to pass the terrain at different locations. Similar to [21], we use the three local terrain characteristics *slope*, *roughness*, and *step height* to estimate the traversability for each cell. The slope value reflects the local surface normal in a small radius around the cell. The roughness is computed as the standard deviation of the height values of the neighboring cells. The difference between the maximal and minimal height of the neighboring cells is captured in the step height value. These terrain characteristics are normalized with a constant for the critical value and summed up to the final traversability value as weighted mean. The traversability is expressed in the interval $[0, 1]$, where a value of 1 indicates full traversability (flat and smooth terrain) and a value of 0 means that the terrain is locally untraversable. If one of the terrain characteristics exceeds its critical value, the traversability is set to 0.

In difference to the method presented in [21], we do not directly assess the traversability for the region of the robot's size, but we choose the radii for the different traversability characteristics as small as reasonably possible depending on the grid resolution [22]. This results in a more precise representation of the environment which enables us to mark features such as stairs or small gaps as traversable.

E. Navigation Planning

We employ a sampling-based RRT* planner to find the shortest valid path from the walking robot's start pose to the goal as presented in [22]. The planner searches for a trajectory of the x- and y-position of the robot's footprint. Since a legged robot works omni-directionally, we ignore the yaw-orientation of the robot in the planning process and adapt the yaw-angle such that the robot walks forward towards the next node on the path. To guarantee a rotation-independent traversability of the terrain, we approximate the footprint conservatively with a circumscribed circle. When checking the connectivity between two nodes in the planning process, we check if all cells within the convex hull of the two circles of the footprint are traversable. We define the path cost between two nodes as the weighted sum of the *length cost* and the *traversability cost*. The length cost is computed as the Euclidean distance between the two nodes. The traversability cost is given by the inverse of the sum of the traversability values in the convex hull of the two circular footprints normalized with the path length. The weighting parameters in the summation of the total costs determine how much short paths over safe paths are preferred or vice-verca.

Since the map is permanently updated, the planner is automatically restarted on a regular basis to update the plan from the current position. In the recurring planning steps, new information in the updated map is taken into account which makes sure that previously unmapped obstacles are avoided and potential shortcuts are taken.

F. Path Execution

The robot receives walking commands by iterating through the nodes of the planned path as the robots walks. A PID controller determines the desired velocity from the error between the current and the next desired pose. The desired velocity contains the translational velocities in x- and y-direction and the yaw rotational velocity. For long distances, we prefer to walk forward and orient the robot towards the goal pose while for short distances, the controller is also allowed to step sideways. The commands are executed by the walking robot by placing its feet in a static walking gait corresponding to the desired velocity [23].

As the navigation planning updates the path on a regular basis, the planning time might cause the new path to start behind the current robot location. To tackle this issue, a simple algorithm iteratively removes the first pose on the path as long as the angle between its connections to the second pose and current robot pose is smaller than 90°.

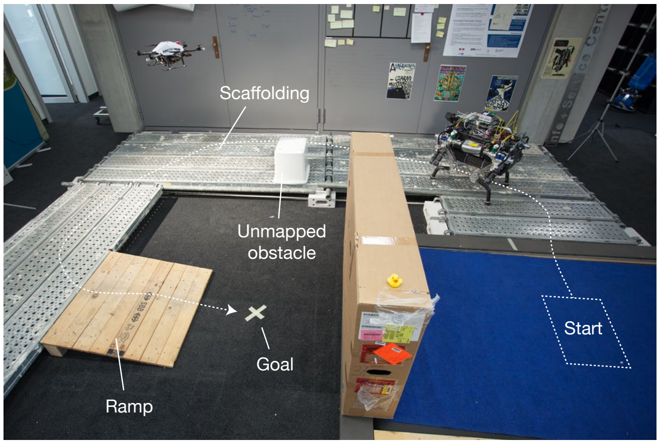

Fig. 3: Overview of the mission in which the walking robot is asked to autonomously find a path from the start region to the goal. In an exploration flight, the hexacopter covers the area to provide global maps of the environment in which the quadruped robot can plan a path and localize against. During execution a previously unmapped obstacle is added, forcing the walking robot to adapt its plan.

III. Experimental Evaluation

We have successfully verified and analyzed our proposed approach with multiple experiments involving the entire algorithmic pipeline. In the following, we describe our results for the mission shown in Fig. 3.[1] The walking robot is deployed in a start area from which the direct line of sight to the desired goal position is blocked by a big obstacle. The only feasible path to the goal consists of a scaffolding with narrow passages (90 cm width) and a ramp which leads down to the floor. During the execution of the mission with the walking robot, a previously unmapped obstacle is put on to the pathway and forces the walking robot to maneuver around it.

A. Localization and Mapping

For mapping the environment with the flying robot, we let the UAV fly multiple rounds over our experimental area to detect and track the visual landmarks (see Fig. 4a). The captured data are transmitted to the walking robot. With its higher computational capacity (Intel Core i7 4500U dual-core 1.8 GHz and 8 Gb RAM), the generation of the global localization map takes several seconds. After the exploration with the flying robot from the air, the walking robot is able to localize itself within the global map. We have tested different start poses of the legged robot without an initial guess for the localization. The proposed method was able to reliably localize even though the landmarks are observed from a different view point (see Fig. 4b) than originally recorded.

Once the robot starts walking, the localization against the global map continuously tracks the pose of the robot (localization runs in real-time with $\sim$20 ms processing time per query). Figure 5 shows the successfully matched features of the walking robot as it moves through the global map. The pose correction updates to the legged state estimator helped to safely track the path on the scaffolding without causing any instabilities of the walking controller.

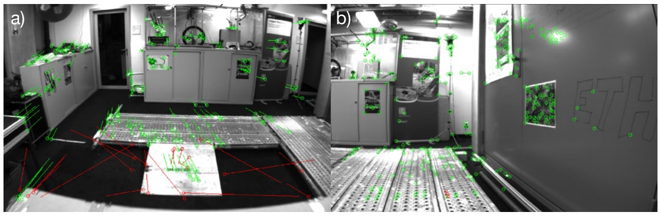

Fig. 4: The tracked features are shown for the flying robot (a) and the legged robot (b). Despite the change in view points both frames are successfully registered to a common global model.

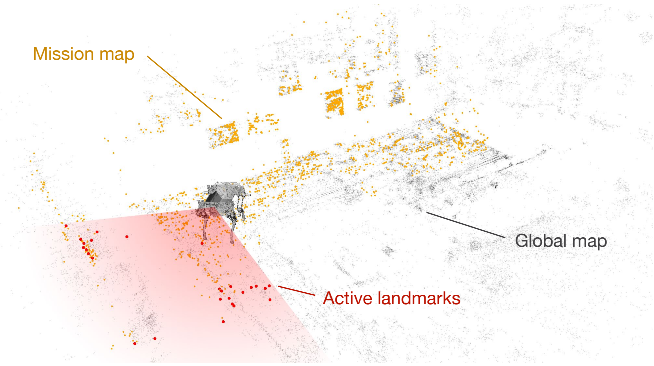

Fig. 5: Visualization of the landmarks as the walking robot executes the mission and localizes itself with the mission map against the global map from the flying robot.

B. Elevation Mapping

Similar to the localization map, the data from the UAV are used to generate dense vision depth estimates which are fused in an elevation map on the computer of the walking robot (processing in approx. real-time). Figure 6a shows a snapshot of the elevation map creation during the exploration phase of the flying robot. The dense reconstruction from the camera of the flying robot delivers accurate depth measurements at a frame rate of 5 Hz. This enables to generate an elevation map with a grid size of 4 cm with only few holes. After the exploration flight, the entire area is well covered as elevation map (see Fig. 6b).

As the legged robot moves through the environment, the elevation map is continuously updated from the rotating laser range sensor as shown in Fig. 6c. The moved obstacle is successfully captured by the map updates (*A* in Fig. 6). Errors in the pose estimation of the walking robot can cause an offset between the previous map from the flying robot and new data from the walking robot. While this has a bigger effect on map regions further away from the walking robot (such as *B* in Fig. 6), these offsets diminish for close regions and the important area in front of the robot is represented with more detail in comparison to the original map (*C* in Fig. 6).

C. Traversability Estimation and Navigation Planning

Figure 7 shows the interpreted elevation map (after the full mission) as traversability map. We can observe that all safe areas including the ramp are correctly marked as traversable while the borders of the scaffolding are correctly interpreted as dangerous. Thanks to the high resolution of the map and the locally bounded interpretation of the traversability, the

[1] A video is available at `https://youtu.be/9PprNdIKRaw`.

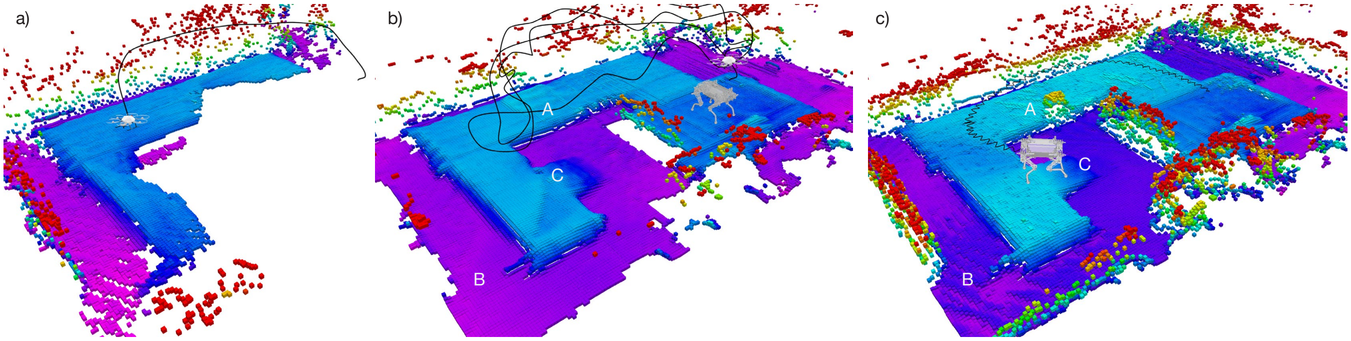

Fig. 6: Three snapshots of the estimated elevation map. a) The elevation map is built incrementally from the dense depth estimates of the flying robot's camera. b) Map after the flying robot has observed the full scene. This map is used for an initial navigation plan for the legged robot. c) Final elevation map where data from the laser range sensor on the walking robot is fused with the original map. Point *A* marks the point where an obstacle has been added after the initial mapping of the flying robot.

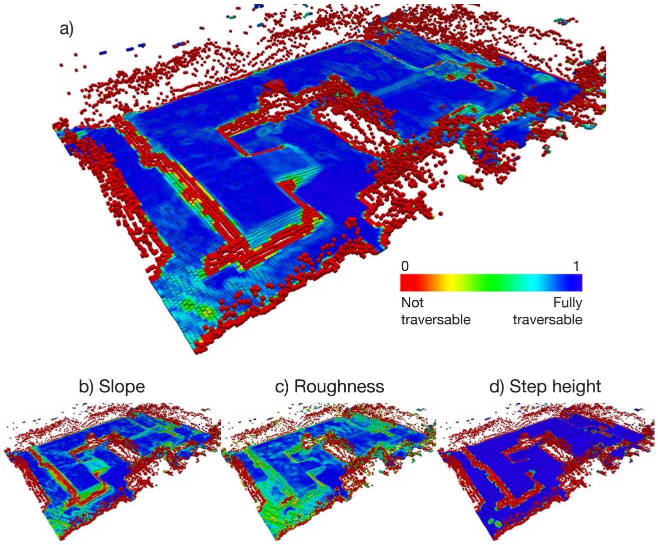

Fig. 7: The traversability of the terrain evaluates whether a certain region is safe to walk over for the legged robot. The total traversability (a) consists of a combination of the local characteristics taking slope (b), roughness (c), and step height (d) into account.

navigation planner can successfully find a feasible path even in narrow regions.

The first plan based on the map from the exploration of the flying robot (no updates from the walking robot yet) is shown in Fig. 8a. The planner is able to find a short path while maintaining a minimal distance to untraversable regions. As the robot starts tracking the planned path, the planner continuously replans the path to take into account the refined map and a changing environment. Figure 8b shows how the updated plan successfully guides the robot around the previously unmapped obstacle. Furthermore, we can observe that our navigation controller approach smoothly guides the robot along the path without causing irregularities because of plan updates.

IV. Concurrent Operation of Walking and Flying Robots

In the experimental evaluation of Section III, we have let the flying robot first fully explore the environment before sharing the data with the walking robot (sequential operation). In the following, we describe the main differences that have to be considered for an operation of the platforms at the same time (concurrent operation).

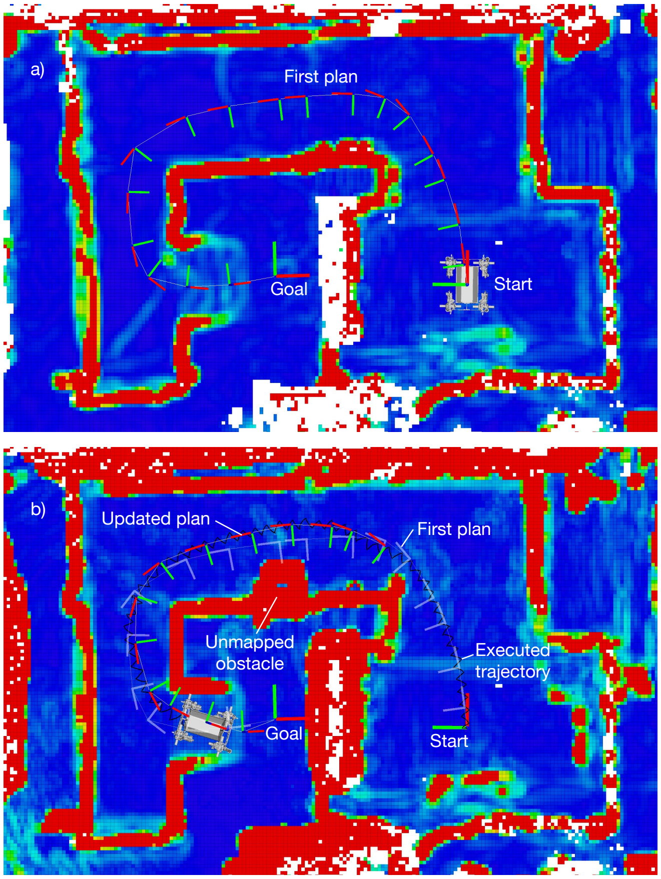

Fig. 8: Top-view on the traversability map and the planned paths for the walking robot. a) Initial path based on the traversability map acquired from the flying robot. b) Updated path to account for the additional obstacle.

We enable a simultaneous exploration and navigation of the robots by updating the shared map data between the robots in a batch-based approach. After the flying robot has transferred the first batch of the (incomplete) map, the walking robot starts to localize against the global localization map and uses the provided terrain data to plan its path and start execution. As the operation continues, new map

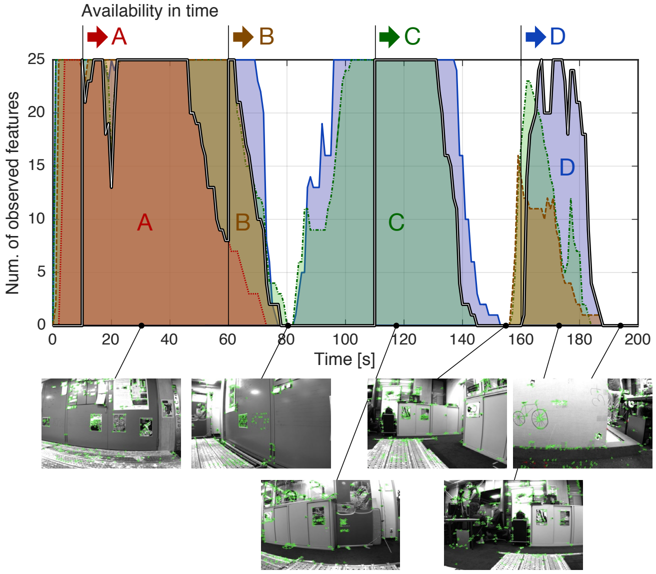

Fig. 9: Number of observed features over time for the legged robot's camera within the global localization map. The colored areas represent the theoretical number of observable features given the localization map at 4 different stages (A–D) of the flying robot's exploration trajectory. After 10 s map A is provided to the robot and the number of observed features (bold black-white) coincides with the observable features of map A. The coverage of the map gets increasingly better as more data is included (A→D).

data is provided by the flying robot and is merged with the existing data to extend the range in which the legged robot can operate within the global frame.

A. Localization and Mapping

As the flying robot operates, the environment is explored on an increasing scale. To maintain global consistency, we apply bundle-adjustment (see Section II-A) on the trajectory which was previously included in the last batch and the newly acquired trajectory data. We maintain a computationally scalable map by pruning features with the least visual information.

As the global map grows with each new batch, the legged robot is able to localize in more areas. Figure 9 shows an example of how the walking robot is able to localize with increased certainty as the map coverage and quality increases with each batch update A–D. We quantify the global localization quality as the number of feature correspondences between the global map and the features from the current view of the walking robot (vertical axis of Fig. 9). The lower limit for the number of features for a (long-term) reliable localization is 4, and for performance reasons we threshold the max. number at 25. Although some viewpoints of the walking robot contain large areas with very little visual features (e.g. at time 80 s and 155 s), the localization can still be performed reliably.

B. Legged State Estimation

With each update of the global map from the flying robot, the localization of the walking robot can vary. Figure 10 shows a top-down view on the trajectories for the localization with the different map batches A–D. For each update A–D, the resulting trajectory for the walking robot transits to the latest (and more accurate) localization. This means that at each update A–D the perceived robot's position can 'jump' instantaneously. We decouple these discontinuities from the control framework with the loosely coupled approach described in Section II-B.

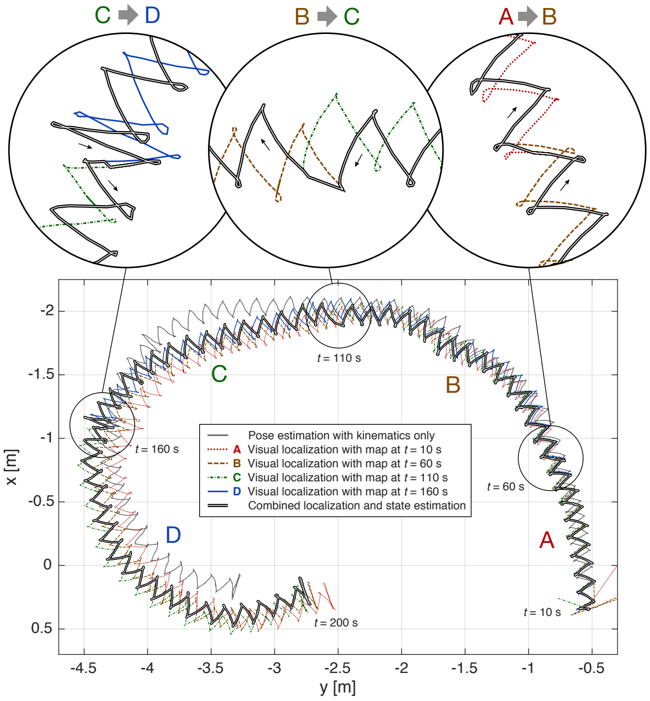

Fig. 10: Top-down view on the localization trajectories of the legged robot for the different map updates A→D. The resulting trajectory for the walking robot transits to the latest trajectory at each map update.

C. Mapping and Navigation Planning

The walking robot can utilize the map updates from the flying robot to increase the range and precision of the path planning. At each map update, empty cells in the global terrain map are filled in where new data is available, while existing cells are not changed. Where no data is yet provided from the flying robot and is not yet visible by the onboard sensors, our planner assumes a traversability with value 0.1. Figure 11 shows snapshots of the concurrent mapping and navigation planning process of the two platforms. The terrain and the walking robot's position changes slightly with each localization and terrain map update from the flying robot. These changes are taken into account in our navigation planning method (Section II-E) which adapts the path locally to changes in the environment and replans the global path on a regular basis.

V. Conclusion

We have presented the design and methods for a collaborative localization and mapping framework for a heterogeneous team of flying and ground robots. The methods applied to the flying and ground robot tightly integrate with each other and allow the ground robot to navigate with significantly increased performance than without the help of the flying robot. We have experimentally validated our approach with a full integration with a hexacopter and a quadrupedal robot,

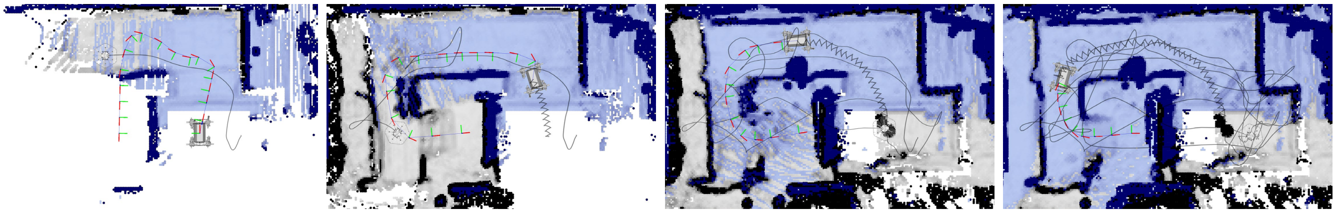

Fig. 11: The terrain map and associated motion plans for different execution stages (A→D). From left to right the coverage of the semi-dense visual map is increasing (grey) as the hexacopter flies over the area (smooth black line). Simultaneously, the legged robot adds its own perception data (laser, blue) to the elevation map. The estimated traversability (brightness of the terrain) as well as the planned (colored coordinate frames) and executed path (zigzag black line) of the legged robot are depicted.

in which the hexacopter enables the quadrupedal robot to navigate in previously unknown, rough terrain in an environment of ∼9×5 m. Our implementation demonstrates a successful functional interaction of many complex navigation tasks, which is rarely demonstrated in other work. We have additionally shown that our framework naturally extends to concurrent operation of the robots.

In current work, the goal is to extend the presented localization method to a larger team of robots (e.g. [24]), but reliable wireless communication is still a major issue. In the future, we will evaluate our approach in more extended and complex environments and adapt our strategy where necessary. Furthermore, we are working on extending the rough terrain capabilities of our legged robot by taking foothold selection and whole body climbing maneuvers into account. Eventually, the legged robot could also be equipped with a landing platform, where the flying robot could rest and charge until its action is required.

References

[1] G. Heppner, A. Roennau, and R. Dillman, "Enhancing Sensor Capabilities of Walking Robots Through Cooperative Exploration with Aerial Robots," *Journal of Automation, Mobile Robotics & Intelligent Systems*, vol. 7, pp. 5–11, 2013.

[2] T. Stentz, A. Kelly, H. Herman, P. Rander, O. Amidi, and R. Mandelbaum, "Integrated Air / Ground Vehicle System for Semi- Autonomous Off-Road Navigation," in *UVSI Symp. Unmanned Systems*, 2002.

[3] B. Grocholsky, S. Bayraktar, V. Kumar, and G. Pappas, "UAV and UGV Collaboration for Active Ground Feature Search and Localization," no. September, pp. 1–8, 2004.

[4] A. Kelly, A. Stentz, O. Amidi, M. Bode, D. Bradley, A. Diaz-Calderon, M. Happold, H. Herman, R. Mandelbaum, T. Pilarski, P. Rander, S. Thayer, N. Vallidis, and R. Warner, "Toward Reliable Off Road Autonomous Vehicles Operating in Challenging Environments," *The International Journal of Robotics Research*, vol. 25, no. 5-6, pp. 449–483, 2006.

[5] M. A. Hsie, A. Cowley, J. F. Keller, L. Chaimowicz, B. Grocholsky, V. Kumar, and C. J. Taylor, "Adaptive Teams of Autonomous Aerial and Ground Robots for Situational Awareness," *Journal of Field Robotics*, vol. 24, no. 11, pp. 991–1014, 2007.

[6] T. A. Vidal-Calleja, C. Berger, J. Solà, and S. Lacroix, "Large scale multiple robot visual mapping with heterogeneous landmarks in semi-structured terrain," *Robotics and Autonomous Systems*, vol. 59, no. 9, pp. 654–674, 2011.

[7] P. Rudol, M. Wzorek, G. Conte, and P. Doherty, "Micro Unmanned Aerial Vehicle Visual Servoing for Cooperative Indoor Exploration," in *IEEE Aerospace Conference*, 2008.

[8] E. Mueggler, M. Faessler, F. Fontana, and D. Scaramuzza, "Aerial-guided Navigation of a Ground Robot among Movable Obstacles," in *IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, 2014.

[9] E. H. C. Harik, F. Guérin, F. Guinand, J.-F. Brethé, and H. Pelvillain, "UAV-UGV Cooperation For Objects Transportation In An Industrial Area," in *IEEE International Conference on Industrial Technology (ICIT)*, 2015.

[10] N. Michael, S. Shen, K. Mohta, Y. Mulgaonkar, V. Kumar, K. Nagatani, Y. Okada, S. Kiribayashi, K. Otake, K. Yoshida, K. Ohno, E. Takeuchi, and S. Tadokoro, "Collaborative Mapping of an Earthquake-Damaged Building via Ground and Aerial Robots," *Journal of Field Robotics*, 2012.

[11] C. Forster, M. Pizzoli, and D. Scaramuzza, "Air-Ground Localization and Map Augmentation Using Monocular Dense Reconstruction," in *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2013.

[12] M. Hutter, C. Gehring, M. Hoepflinger, M. Bloesch, and R. Siegwart, "Towards combining Speed, Efficiency, Versatility and Robustness in an Autonomous Quadruped," *IEEE Transactions on Robotics*, vol. 30, no. 6, pp. 1427–1440, 2014.

[13] J. Nikolic, J. Rehder, M. Burri, P. Gohl, S. Leutenegger, P. T. Furgale, and R. Siegwart, "A Synchronized Visual-Inertial Sensor System with FPGA Pre-Processing for Accurate Real-Time SLAM," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2014.

[14] S. Lynen, T. Sattler, M. Bosse, J. Hesch, M. Pollefeys, and R. Siegwart, "Get out of my lab: Large-scale, real-time visual-inertial localization," in *Robotics: Science and Systems*, 2015.

[15] S. Lynen, M. Bosse, P. Furgale, and R. Siegwart, "Placeless place-recognition," in *International Conference on 3D Vision*, 2014.

[16] M. T. Dymczyk, S. Lynen, M. Bosse, and R. Siegwart, "Keep it brief: Scalable creation of compressed localization maps," in *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2015.

[17] M. Bloesch, C. Gehring, P. Fankhauser, M. Hutter, M. A. Hoepflinger, and R. Siegwart, "State Estimation for Legged Robots on Unstable and Slippery Terrain," in *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2013.

[18] P. Fankhauser, M. Bloesch, C. Gehring, M. Hutter, and R. Siegwart, "Robot-Centric Elevation Mapping with Uncertainty Estimates," in *International Conference on Climbing and Walking Robots (CLAWAR)*, 2014.

[19] F. Pomerleau, A. Breitenmoser, M. Liu, F. Colas, and R. Siegwart, "Noise Characterization of Depth Sensors for Surface Inspections," in *IEEE International Conference on Applied Robotics for the Power Industry (CARPI)*, 2012.

[20] T. Lemaire, C. Berger, I.-K. Jung, and S. Lacroix, "Vision-Based SLAM: Stereo and Monocular Approaches," *International Journal of Computer Vision*, vol. 74, pp. 343–364, Feb. 2007.

[21] A. Stelzer, H. Hirschmuller, and M. Gorner, "Stereo-vision-based navigation of a six-legged walking robot in unknown rough terrain," *The International Journal of Robotics Research*, vol. 31, pp. 381–402, Feb. 2012.

[22] M. Wermelinger, P. Fankhauser, R. Diethelm, P. Krüsi, R. Siegwart, and M. Hutter, "Navigation Planning for Legged Robots in Challenging Terrain," in *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2016.

[23] C. Gehring, S. Coros, M. Hutter, M. Bloesch, M. A. Hoepflinger, and R. Siegwart, "Control of Dynamic Gaits for a Quadrupedal Robot," *IEEE International Conference on Robotics and Automation (ICRA)*, 2013.

[24] T. Cieslewski, S. Lynen, M. Dymczyk, S. Magnenat, and R. Siegwart, "Map API - scalable decentralized map building for robots," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6241–6247, 2015.