

Bird's Eye View: Cooperative Exploration by UGV and UAV

Shannon Hood, Kelly Benson, Patrick Hamod, Daniel Madison, Jason M. O'Kane, and Ioannis Rekleitis

Abstract—This paper proposes a solution to the problem of cooperative exploration using an Unmanned Ground Vehicle (UGV) and an Unmanned Aerial Vehicle (UAV). More specifically, the UGV navigates through the free space, and the UAV provides enhanced situational awareness via its higher vantage point. The motivating application is search and rescue in a damaged building. A camera atop the UGV is used to track a fiducial tag on the underside of the UAV, allowing the UAV to maintain a fixed pose relative to the UGV. Furthermore, the UAV uses its front facing camera to provide a birds-eye-view to the remote operator, allowing for observation beyond obstacles that obscure the UGV's sensors. The proposed approach has been tested using a TurtleBot 2 equipped with a Hokuyo laser range finder and a Parrot Bebop 2. Experimental results demonstrate the feasibility of this approach. This work is based on several open source packages and the generated code is available on-line.

I. INTRODUCTION

With the proliferation of unmanned air vehicles (UAVs) a new viewpoint, a birds-eye-view, has become available in many domains, allowing structure inspection, environmental monitoring, virtual tourism, and search and rescue. The majority of aerial vehicles are severely limited in the sensing payload they can carry due to weight constraints. As such, simple tasks such as obstacle avoidance and localization in an unknown environment become rather challenging. On the other hand, ground robots, while extremely capable in navigating safely inside an unknown environment, are restricted to a limited field of view. Deploying a robot team comprised of a ground and an aerial vehicle enables an enhanced field of view while maintaining safe navigation and localization. Recent advances in robotics have made collaboration among a heterogeneous team of robots possible [1]. At the same time, many interesting problems associated with multi-robot collaborative exploration are being discussed, such as coordination, communication protocols, bandwidth constraints, etc. [2], [3]. One motivating application is search and rescue in a damaged building, where the UGV navigates on the ground while the UAV maintains a fixed pose above the UGV employing a robot-tracker sensor consisting of a camera and fiducial marker. The robot team could search and map dangerous areas, monitored by a remote operator at a safe distance.

The goal of this paper is to present a collaborative exploration approach utilizing a UGV and a UAV. The UGV, a Kobuki TurtleBot 2 equipped with a Hokuyo laser range finder, is responsible for guiding the UAV through the corridors of a building. The UAV, a Parrot Bebop 2 equipped with a forward facing camera, provides enhanced situational awareness to a remote expert. The UAV does not



Fig. 1. A Parrot Bebop 2 hovering over a Kobuki TurtleBot 2.

process all data and commands on-board; instead, utilizing a WiFi connection, the sensor data are transmitted to a ground station where all the processing takes place. Currently we utilize two computers: one on the UGV and a separate one as a ground control station (GCS). A unique fiducial tag placed on the bottom of the UAV is tracked by a wide angle camera on top of the UGV, and the relative pose between the two vehicles is estimated.

Using cooperative localization, the pose of the UAV is calculated using the TurtleBot's pose and the relative pose of the two robots. This calculated pose is compared to the pose provided by ORB-SLAM2 [4], a monocular SLAM (Simultaneous Localization and Mapping) package run using data provided by the UAV's camera. These two localization techniques improve the accuracy of UAV's estimated pose, allowing for easier path planning and tracking of the tag.

Related work is discussed next. Section III outlines the proposed methodology. Experimental results and factors af-

fecting the performance are discussed in Section IV. The paper concludes with a discussion of lessons learned and future directions.

II. RELATED WORK

Cooperative robotics has been explored as early as the 90's, such as when Cao et al. [5] discussed possible future applications and the open problem of collaboration between autonomous robots. Yuta and Premvuti [6] discussed how to organize multiple autonomous robots. Several previous experiments have examined the best software for tag tracking, autonomous communication between robots, or have used similar setups involving both a ground and an aerial based robot. Collaboration between UGVs and UAVs has been proposed for a variety of fields. For example, in the detection and fighting of wildfires [7], in measuring the amount of nitrogen in the soil for agricultural applications [8], for surveillance [9], and for detecting and disposing of mines [10]. UAVs have also been used to monitor traffic [11].

A. Robot Collaboration

Chaimowicz and Kumar [12] discussed ground robot and aerial robot localization in their 2007 paper. Their research focused on environments with limited GPS access, and on using GPS in conjunction with other sensors. Similar concerns also drove the work of Frietsch et al. [13]. However, unlike these two previous works, the approach proposed by this paper assumes no GPS access, instead relying on cooperative localization [14] between the two vehicles as well as pose estimates calculated from the laser scan and cameras.

Another area of interest is collaboration between sea-based robots and aerial robots; see Murphy et al. [15] and Shkurti et al. [1]. In Murphy's experiment, the robots were used to survey storm damage from Hurricane Wilma. It was also "the first known use of unmanned sea surface vehicles (USVs) for emergency response" [15]. In their paper titled "Multi-domain monitoring of marine environments using a heterogeneous robot team", Shkurti et al. used underwater vehicles, airboats, and aerial vehicles to autonomously collect footage of underwater scenes.

MacArthur and Crane [16] investigated using a UAV in the state estimation of a UGV. In their paper, they discuss camera calibration and extracting pose from camera data, as well as the coordinate transforms needed. Unlike with this project, GPS was integrated with pose information. MacArthur and Crane found that localization of a UGV using a passive UAV was successful in both simulated and experimental runs.

Cooperation between UAVs and UGVs for target detection and formation holding was proposed by Tanner [17]. Duan and Liu [18] discuss an overview of multiple vehicles collaborating with a focus on military applications. Papachristos and Tzes [19] proposed a tethered solution to address the limited battery time for longer operations. Autonomous landing on a UGV charging station was proposed by Rezelj and Skocaj [20].

B. Tag Tracking

Other projects on similar topics have focused more on landing the drone on a target rather than following the target for cooperative exploration with a ground vehicle. One example is the research by Minhua and Jiangtao, which involved landing an AR.Drone 2.0 on an augmented reality tag that designated a landing pad [21]. Minhua and Jiangtao tried two different methods of tracking the landing pad. The first was a QR code, which they found gave the UAV more information about the landing area, but at the cost of more processing time [21]. This caused delays and inaccuracy while tracking the QR code. The second method they tried was tracking augmented reality tags through a package called `ar_track_alvar`, which they found processed the tag faster. This faster computation is possible because `ar_track_alvar` is initialized with the size of the tag to help it better determine the position of the tag relative to the frame given. In this experiment, Minhua and Jiantao modified the front camera of the drone to point towards the ground for better tag tracking because the front camera has a higher resolution [21]. They also experimented with using `ar_track_alvar` in conjunction with `tum_ardrone`, a wrapper for PTAM (Parallel Tracking and Mapping) and PID controller for the drone, when they were landing the AR.Drone on a tag. They found that using `ar_track_alvar` to find the tag and `tum_ardrone` for the PID controller to move the UAV into position of the tag landing zone worked fairly well [21].



Fig. 2. An example of a QR code is shown first, while two fiducial tags generated by `ar_track_alvar` are shown after.

Another example of previous work related to autonomous flight is the experiment by Rezelj and Skocaj focused on landing a modified AR.Drone on a TurtleBot with an augmented reality tag, so that the drone could charge and then take off again by making use of the ground vehicle's longer battery life [20].

Although a fiducial tag will be used as a marker for the UGV in this paper, it is important to point out that other methods of tracking are available, such as tracking a colored object or a 3-D moving target, such as in Chakrabarty's experiment with an AR.Drone [22].

Lawrence and Turchina [23] evaluated different approaches to fiducial tag tracking. In their experiments on pose estimating software packages, they found that `ar_track_alvar` was the best package for estimating the pose of a tag from the frame of a camera. Packages they tried include: `ar_sys`, `visp_auto_tracker`, and `ar_track_alvar`. However, not all of them work with ROS and some require a wrapper. It is important to note that

the camera they used to test the packages had a resolution of 640x480, while our tracking camera has a resolution of 1920x1080. The higher resolution of the Bird's Eye View camera allows for better tag tracking, since the tag can move farther distances and still remain within view of the camera. The next tested package, `ar_sys`, gave a range of 30cm to 120cm at 30Hz. The next package, `visp_auto_tracker`, had a range of 60cm to 300cm at 30Hz. One major drawback of this package was that if the UAV lost sight of the tag, it needed to re-establish at the minimum distance to continue tracking. Thus with this package, any time the tag was lost the UAV would have to maneuver to a precise location to reinitialize the tracking. Lastly, Lawrence and Turchina tested `ar_track_alvar`, which ran from 30cm to 300cm at 10Hz. This package had the best range for establishing tracking and is what is used in this approach. One problem of tag tracking software, in addition to range limitations and reinitializing the package when the tag is lost, is blurring of the tag due to motion. This makes detecting the tag more difficult, and is discussed in detail in "A motion blur resilient fiducial for quad-copter imaging" by Meghshyam Prasad et al. [24].

One problem with `ar_track_alvar` is determining what to do if the tag moves out of view of the camera. If the tag is not visible, the package has no way of tracking the UGV. Huang et al. proposed a solution to this in their paper "An Object Following Method Based on Computational Geometry and PTAM for UAV in Unknown Environments" [25]. Their solution was to use a Kalman filter in order to predict the velocity of the tag, so the tag can be found and tracking reinitialized.

C. Monocular SLAM

Monocular Simultaneous Localization and Mapping relies on computer vision techniques to calculate the pose of the robot and map the environment. In this work, monocular SLAM is performed on the UAV's front camera for state estimation. Real-time monocular SLAM was once thought impossible due to high computational costs; one of the first implementations was in the Parallel Tracking and Mapping (PTAM) package. Now, there exist a great variety of monocular SLAM packages, including LSD-SLAM [26], SVO [27], and RatSLAM [28]. Many open-source vision-based SLAM packages were compared in the 2016 paper by Quattrini Li et al. [29]. ORB-SLAM [4] was found to be one of the better performing packages, and is used in this project.

PTAM was originally made by Klein and Murray in 2007 [30]. Using the information of the UAV's camera it is able to find "features," or the edges of detected obstacles, in order to estimate distance traveled as well as scale of distances of objects from the UAV. PTAM has also been used in the collaboration of multiple UAVs. Bazen et al.'s 2016 experiment [31] involved coordinating a fleet of quad-rotors using `ardrone_autonomy` and `tum_ardrone`. They found that, while the completed missions were more accurate when PTAM was used, the missions were successful less frequently. They theorized this was due to delay "between

acquisition of information by the drones and their actual displacement based on this information" [31].

Another later monocular SLAM package, ORB-SLAM [4], improves upon the work by Klein and Murray. ORB-SLAM uses ORB features, which are binary features that are invariant to rotation and scale. Unlike PTAM, ORB-SLAM is able to initialize with no user input, which is necessary for autonomous robot systems. ORB-SLAM allows for more accurate mapping by performing loop closure, where loops are detected and the ends merged to remove drift over time. Additionally, the number of keyframes used by ORB-SLAM when matching features increases depending on the complexity of the image, unlike in PTAM where keyframes increase over time. This is done by "culling" keyframes that are no longer used, and allows for ORB-SLAM to run for longer periods of time.

D. Cooperative Localization

Cooperative localization (CL) deals with estimating the pose of teams of mobile robots using sensor data. This is done in order to provide enhanced localization compared to the robots' localization capabilities without cooperation. In order for this to be done, a transformation matrix between two robots must be calculated. Cooperative localization was first introduced by Kurazume et al. [32] in 1994. At the time it was known as "cooperative positioning". In this experiment, robots were separated into two different teams: one team moving, while the other team remains a stationary landmark for the other team. The teams then switch roles until all robots reach their target position.

The term "cooperative localization" first appeared in 1998 in works by Bison et al. and Rekleitis et al. [33], [14]. Since its conception, CL has been applied to many different problem types in both 2- and 3-D, including vision-based [34], [35], sonar-based [36], [37], and range-based [38], [39]. Two types of cooperative localization techniques especially relevant to this paper are those dealing with vision-based and 3-D problems. In 2009, Bahr et al. used cooperative localization for the localization of autonomous underwater vehicles. The goal of this work was to "create a fully mobile network of Autonomous Underwater Vehicles (AUVs) that perform acoustic ranging and data exchange with one another to achieve cooperative positioning for extended duration missions over large areas" [40].

Early work was done on vision-based CL by Jennings, Murray, and Little [41]. In this work, two stereo vision-based mobile robots explore and map their environment. One robot finds landmarks in the environment that the second robot then uses to localize itself relative to the first robot's reference frame. This allows the robots to collaborate on tasks using a common local reference frame without a complete map of the environment. In 2011, Chang et al. [42] worked with humanoid robots in the RoboCup environment to localize and track moving objects using vision-based cooperative localization. First, they estimated the robot pose by modeling the uncertainty of motion commands and measurements. Then when other robots were within the field of view, state

estimates were refined using the estimated pose and distances calculated based on the image.

III. APPROACH

In this section we describe the necessary parts of the Bird's Eye View system and how they must act together in order to achieve the goals set out above. The robots must be able to communicate with each other and with a Ground Control Station. One of the robots must detect the fiducial tag and extract the coordinate transformation between the UAV and the UGV. The UAV has to be able to hover above the UGV while tag-tracking is maintained. The UGV must autonomously explore an indoor environment, and the UAV must know the pose of the UGV and move accordingly while maintaining tag tracking.

In the development of the UAV/UGV system, several open source ROS packages were used: `bebop_autonomy`, as a driver for the Bebop and joystick; `usb_cam`, a driver for the Logitech HD Pro Webcam C910; and ORB-SLAM 2 [4], a monocular SLAM package.



Fig. 3. Shows the network connections between the ground Control Station (GCS), the TurtleBot, and the Bebop. Connections are shown in blue.

A. Robot Team Network

Communication between the UAV, UGV, and a Ground Control Station is enabled by a network connecting the 3 systems. The Bird's Eye View project requires two PCs: a Ground Control Station (GCS) and a TurtleBot PC. The TurtleBot PC is connected to the TurtleBot and the Hokuyo laser scanner via USB. A portable router and power source (for the router) are also placed on the TurtleBot. The specific router used in this project is the TP-Link 150Mbps Wireless-N Nano Pocket Router. The TurtleBot PC was connected to this router using an Ethernet cable. The Ground Control Station was then connected wirelessly via WiFi to the router. These two connections allow the TurtleBot PC and GCS to communicate with each other. Since the TurtleBot PC can be controlled wirelessly using the GCS, the GCS can now be used to control both the Bebop and the TurtleBot. The GCS

must be kept within range of the portable router in order for both robots to operate correctly. However, the TurtleBot PC will always be within range of the router. All components of the Bird's Eye View network are shown in the diagram below; see Fig. 3.

B. Movement of the UGV

Two different motion strategies were used for the UGV. First, in order to challenge the tracking capabilities of the UAV, the UGV is placed inside an enclosed area and a random walk is executed. The UGV moves forward until it reaches near an obstacle, then performs an in-place rotation of 180 degrees plus a small random value, then moves again in a straight line. The UAV was able to follow the UGV through out these movements, even though in place rotations are more challenging. Second, the UGV uses a center-of-the corridor following algorithm employing a PID controller to navigate hallways. This algorithm compares the distances read by the laser scanner 60 degrees off the center of the UGV. As the UGV moves forward, it also turns slightly favoring the side with the shorter distance. This is the core behavior of the generalized Voronoi graph algorithm [43].

Communication between the two robots improves the UGV's movement algorithm. The UGV does not start moving until the UAV is able to see the fiducial tag. This allows for a much simpler initialization process. Furthermore, should the UAV lose sight of the tag for more than two seconds, the UGV will stop moving and wait for the UGV to find the tag again. Should the UGV be unable to find the tag again, it is possible for an operator to step in and move the UAV back into position using the joystick controller.

C. Tag Detection & Movement of the UAV

When first launched by a trained operator, the UAV must be moved into position so the fiducial tag is above the TurtleBot's camera using a Logitech F710 joystick and the `bebop_autonomy` package. The joystick allows for precise control in moving the UAV to a position where it is able to recognize the tag. The joystick also provides a safety feature to control the UAV should loss of control occur during the flight. Once in position, the UAV then autonomously follows the UGV utilizing the fiducial tag. Early work employed the Parrot AR.Drone 2.0 and the tag was mounted on the UGV; see Fig. 4. This early approach was abandoned due to the low camera quality of the AR.Drone which resulted in frequent loss of tracking.

The Bird's Eye View system uses tags generated by the `ar_track_alvar` ROS package. When a tag is within view of the UGV's upward-facing camera, the estimated pose of the tag is published to the `ar_pose_marker` ROS topic; see Fig. 5. The drone then moves so that the tag remains directly over the camera. This is done by transforming the tag's pose to x, y, z, and yaw values relative to the UAV's frame of reference. To maintain tracking, the tag must remain within the camera's field of view. A 4.6 cm tag was found to be most visible from a height of 0.65 to 0.95 meters. As a result, a constant height value of 0.8 meters is added



Fig. 4. Early approach using a Parrot AR.Drone 2.0 and the tag on the UGV.

to the negative z dimension. Next, the x - and y -velocities of the tag are calculated using the current state of the tag, the previous state, and the time difference between the two states. The current pose of the tag and the tag velocities are used to calculate the required speed of the UAV in order to maintain sight of the tag. The UAV's velocities are calculated as follows:

$$UAV.angular.z = tag.yaw * c$$

$$UAV.linear.z = tag.z * c$$

$$UAV.linear.x = tag.x * d + tag.vx$$

$$UAV.linear.y = tag.y * d + tag.vy$$

In these equations, c and d are constant scaling factors. These factors are necessary because the `bebop_autonomy` driver expects velocities between -1 and 1 inclusive. Best results were obtained using $c = 0.4$ and $d = 0.07$.

D. Cooperative Localization

The Bird's Eye View package performs the cooperative localization of the UAV using the pose of the UGV and the pose of the tag. First, the odometry data is transformed from the base.link frame to the map frame. This transformation converts the local pose at the rotational center of the robot to the fixed world frame, which does not drift over time. The camera is located approximately 0.45 meters above the TurtleBot's base, which is taken into account when



Fig. 5. Camera feeds: In the top image, the Bebop 2 flies over the UGV with a tag. The bottom image shows the UAV's front camera view.

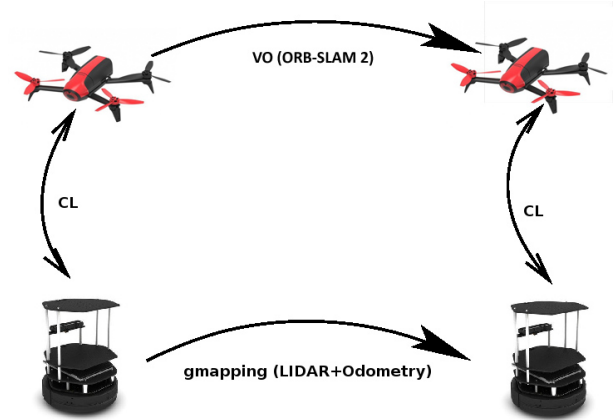


Fig. 6. Shows the relationships between the different odometry and localization methods used.

calculating the UAV's pose. Next, the relationship between the fiducial tag's position and the TurtleBot's camera is obtained using the data provided by `ar_track_alvar`. This pose is added to the UGV's pose to determine the UAV's current position in the world frame. Since the UAV's and UGV's poses are both with respect to the world frame, they are easily plotted in `rviz`¹ alongside the laser scan data. Figure 6 shows how the different odometry and SLAM data relate to one another.

¹wiki.ros.org/rviz

E. UAV Localization Using ORB-SLAM

In addition to being calculated through cooperative localization, the UAV's pose is also estimated using the ORB-SLAM 2 package. ORB-SLAM detects features in the UAV's camera feed and uses the change in location of these features to determine the pose of the camera. These features can be seen in Fig. 5; new features are added every time a new keyframe is taken. Because most monocular vision based SLAM algorithms, like ORB-SLAM, do not have scale information, either from ground odometry, from stereo, or from inertial data, it is impossible to calculate the scale of the estimated poses. As a result, ORB-SLAM uses an arbitrary scale. In order to compare the ORB-SLAM pose estimates to the poses calculated using cooperative localization, the scale of the ORB-SLAM data must be adjusted to match. The data is translated to the start position in the world frame and then multiplied by a scaling factor. This scale factor is computed using the distance between the current pose of the UAV and the pose when the tag first enters the camera's field of view. The ratio between the distance as estimated by ORB-SLAM and by cooperative localization is used as a scale factor. Since the scale factor remains close to constant, the scale factor found using the cooperative localization pose can be used even when tag tracking is no longer maintained.

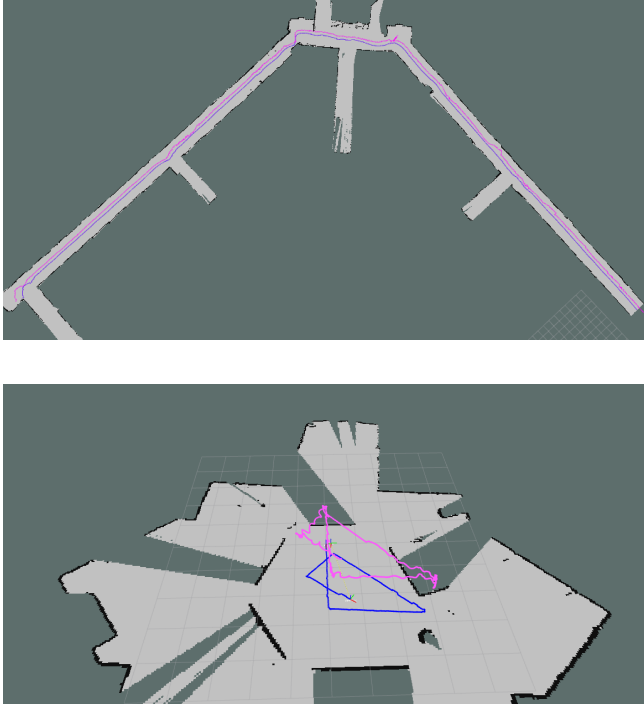


Fig. 7. The top figure shows the poses of the two robots navigating a corridor (blue UGV, pink UAV), while the bottom shows a random walk in an enclosed space.

F. Rejected Approach

Originally this project was attempted using an AR.Drone 2.0 rather than a Bebop. The ROS package `tum_ardrone`

was used as a PID controller and SLAM (Simultaneous Localization and Mapping) package. In that iteration, the tag was placed on the UGV while the UAV used its bottom-facing camera to track the tag's motion; see Fig. 4. However, the Parrot AR.drone UAV was unable to use the front and bottom cameras simultaneously. To deal with this issue, we attempted a workaround using a camera switching node to toggle between the front and bottom camera's approximately once a second. The goal was to switch between cameras continuously at a rate that allowed the UAV to track the UGV while still collecting data about the environment with the front camera. However, this approach did not succeed. PTAM was not able to accurately estimate the UAV's pose with such a limited number of camera frames, resulting in large amounts of drift. Additionally, the bottom-facing camera on the AR.Drone had a lower resolution, meaning if the UGV moved at a rate greater than 0.2 meters per second, the UGV could move out of view of the UAV in the time it took the cameras to switch, which would lead to a loss of tracking. Furthermore, the switching delay caused corrupted images every so often. In addition to difficulty in tracking the tag, the Parrot AR.Drone 2.0 was very unstable in position keeping and it easily drifted away from the target.

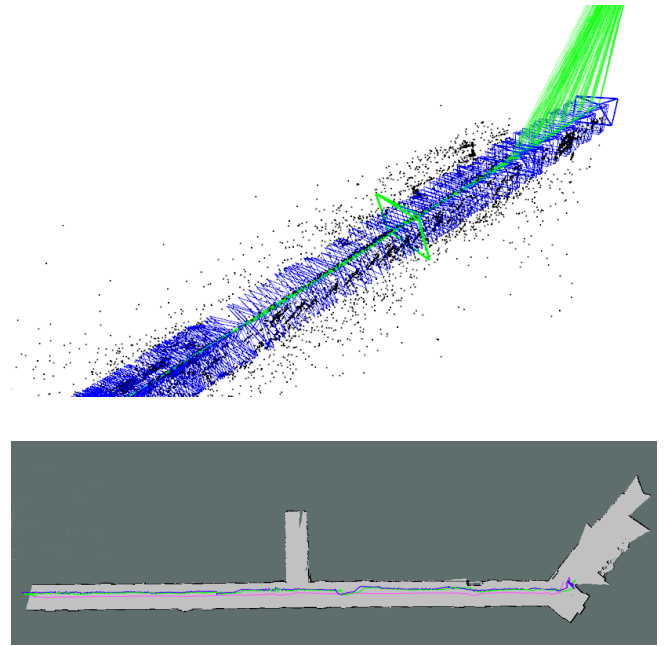


Fig. 8. Center-of-the-corridor ORB-SLAM experiments: (a) The top image shows a map of features and keyframes generated by ORB-SLAM. (b) In the bottom image, the UGV trajectory is shown in blue, the UAV trajectory as calculated by CL in pink, and the UAV trajectory according to ORB-SLAM in green.

IV. RESULTS

The Bird's Eye View project can be evaluated by comparing the poses of the UAV and UGV to determine how well the UAV succeeds in following the UGV. The pose of both the UAV and UGV are visualized in rviz, and the pose provided by cooperative localization is compared to the pose calculated by ORB-SLAM. Finally, the video from the TurtleBot's camera is examined to determine if any issues in tag detection or UGV following are present.

Figure 7 shows a long trial run down the 3rd floor of Swearingen Engineering Center. This experiment ran for 800 seconds. At a speed of .15 meters/second, the UAV and UGV covered 120 meters. For increased distance, the speed of the UGV could be increased. However, if the speed is too high, the UAV will be unable to follow the tag. This is because at high speeds the UAV will move out of view of the UGV's camera. During this experiment, the operator intervened one time when the UAV flew directly under an A/C vent and was blown off course. The length of this experiment shows that the UAV is able to follow the UGV for long distances without drifting off course. As can be seen in Fig. 7, the UAV very closely followed the UGV as it navigated the hallway. The tag remained within view as the robots rotated, and tracking was maintained. The tag was seen 89% of the time when run at a frequency of 30 Hz.

Figure 8(a) shows the ORB-SLAM resulting feature points and keyframes used in calculating the pose of the UAV. This pose is given in a world reference frame. However, since ORB-SLAM does not have any knowledge of the scale of the system, it is not the same world coordinate frame used by the UGV and cooperative localization. In order to be compared directly to those poses, the ORB-SLAM data must be scaled accordingly. Figure 8(b) shows the ORB-SLAM data scaled and plotted next to the CL estimate and UGV's trajectory. This scaled pose is used as an estimate for the UAV's world frame pose when the tag is not visible by the system. Using the UGV's global pose and ORB-SLAM, the UAV attempts to find the UGV autonomously if tracking is lost.

Figure 9 presents the evolution of the position of the UGV and the UAV based on different estimation approaches. The three figures show the values of the X-axis (along the direction of travel); the Y-axis (around 1 m drift of the UAV); and the Z-axis. In each figure, the top plot shows the odometry values for the UGV; the middle plot shows the estimate of the UAV's pose based on CL; and the bottom plot contains the estimates from ORB-SLAM scaled using the information from CL. The longest motion was along the X-axis; as expected the Z-axis results are constant for the UGV as it operates on flat terrain, the UAV though, maintained a nominal 1.1 m height, with small variations.

In all of the experimental data collected, the pose of the UAV and UGV are very similar, excluding the difference in height above the ground. The UGV successfully navigates the corridors of the Swearingen Engineering building, staying in the center of the hallway, and the resulting maps accurately show the robots' paths through the hallways. The system

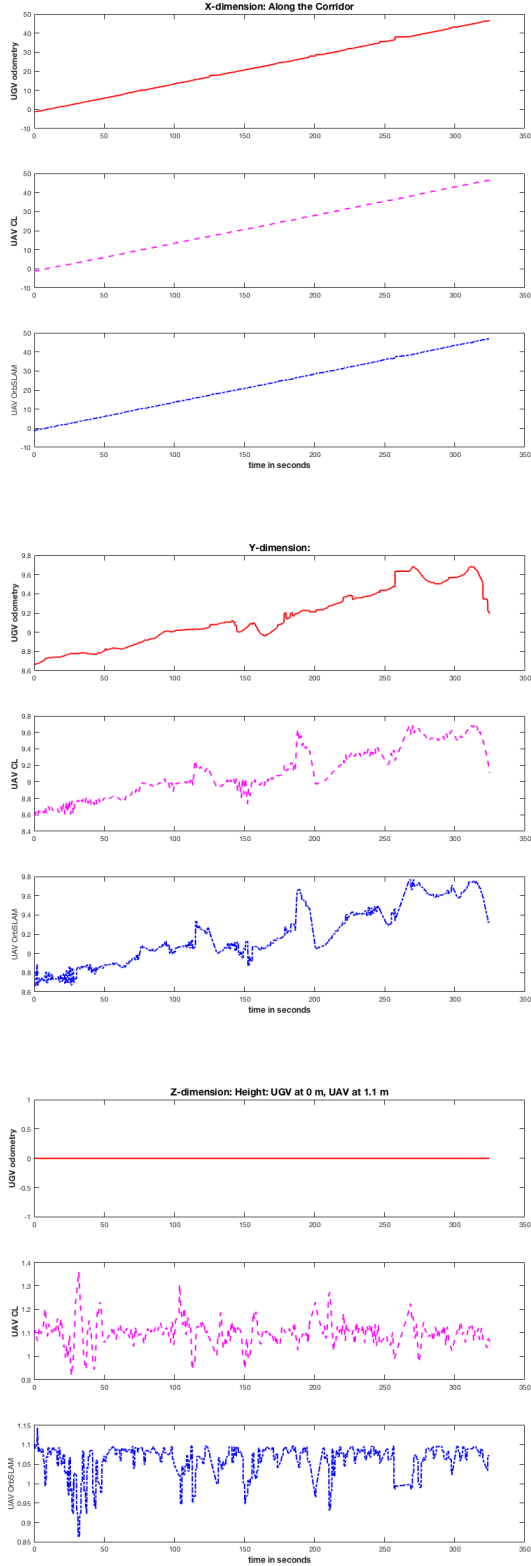


Fig. 9. Pose Estimation for the UGV and UAV. The X, Y, and Z estimates from the experiment depicted in Fig. 8 are plotted. In each image, the top plot presents the odometric estimate of the UGV; the middle plot contains the estimated pose of the UAV by combining the odometric value of the UGV and the CL transformation; and the bottom plot presents the estimate based on the ORB-SLAM package, scaled according to the CL value.

is able to successfully recover from a loss of tag tracking using the ORB-SLAM estimated pose. As all of these objectives are met, one can say the two robots are able to cooperatively and autonomously explore and map an unfamiliar environment.

V. CONCLUSION

This paper aims to determine the abilities of a UAV and UGV team to autonomously search, map, and explore an indoor environment. The two vehicles remain in contact with each other visually and over WiFi. Cooperative localization of the UAV using the UGV's pose and relative pose between the two is combined with the UAV's pose provided by ORB-SLAM for more accurate localization. The framework developed in this paper is targeted towards search and rescue operations.

The results show that the UGV is able to autonomously navigate the environment and that the UAV is able to follow the UGV using vision-based techniques for long periods of time. The pose of the UAV is calculated both using cooperative localization and the monocular SLAM package ORB-SLAM 2 for enhanced localization. Upon a loss of tag tracking, the ORB-SLAM estimate is used to attempt recovery of the system. These poses are compared to the TurtleBot's pose as calculated from the laser data to show the viability of this approach. All code used in this paper is available online².

Future work will consider expanding the ORB-SLAM output to perform 3-D mapping of an area by fusing the data from the UAV's front camera and the UGV's laser scanner. Currently the maps provided by the laser rangefinder and ORB-SLAM are kept separate, but will be combined for increased accuracy.

Further extensions of this work will exchange the UGV for an Autonomous Surface Vehicle (ASV), like the one in Fig. 10, to assist in environmental monitoring. Having a base to land will extend the range of operations for UAVs and provide additional information.



Fig. 10. Autonomous Surface Vehicle (ASV) based on the Mokai Eclipse platform.

ACKNOWLEDGMENT

The authors would like to thank the University of South Carolina for its generous support. We acknowledge support for this work from a Google Faculty Research Award and the National Science Foundation grants (NSF 1513203, 1637876, 1526862).

REFERENCES

- [1] F. Shkurti, A. Xu, M. Meghjani, J. C. G. Higuera, y. Girdhar, P. Giguere, B. B. Dey, J. Li, A. Kalmbach, C. Prahacs, K. Turgeon, I. Rekleitis, and G. Dudek, "Multi-domain monitoring of marine environments using a heterogeneous robot team," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Portugal, Oct. 2012, pp. 1447–1753.
- [2] D. Liu, X. Li, and W. Liu, "The research about collaboration techniques for aerial and ground mobile robots," *Proceedings of Science*, 2015.
- [3] S. L. Waslander, "Unmanned aerial and ground vehicle teams: Recent work and open problems," in *Autonomous control systems and vehicles*. Springer, 2013, pp. 21–36.
- [4] R. Mur-Artal, J. Montiel, and J. D. Tardós, "Orb-slam: a versatile and accurate monocular slam system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.
- [5] Y. U. Cao, A. S. Fukunaga, and A. Kahng, "Cooperative mobile robotics: Antecedents and directions," *Autonomous robots*, vol. 4, no. 1, pp. 7–27, 1997.
- [6] S. Yuta and S. Premvuti, "Coordinating autonomous and centralized decision making to achieve cooperative behaviors between multiple mobile robots," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 3. IEEE, 1992, pp. 1566–1574.
- [7] C. Phan and H. H. T. Liu, "A cooperative UAV/UGV platform for wildfire detection and fighting," in *System Simulation and Scientific Computing*, Oct 2008, pp. 494–498, 7th International Conference Asia Simulation Conference.
- [8] P. Tokekar, J. Vander Hook, D. Mulla, and V. Isler, "Sensor planning for a symbiotic uav and ugv system for precision agriculture," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 5321–5326.
- [9] B. Grocholsky, J. Keller, V. Kumar, and G. Pappas, "Cooperative air and ground surveillance," *IEEE Robotics & Automation Magazine*, vol. 13, no. 3, pp. 16–25, 2006.
- [10] D. M. E. MacArthur and C. Crane, "Use of cooperative unmanned air and ground vehicles for detection and disposal of mines," in *Proc. SPIE*, vol. 5999, 2005, p. 9, intelligent Systems in Design and Manufacturing VI. [Online]. Available: <http://dx.doi.org/10.1117/12.631314>
- [11] K. Kanistras, G. Martins, M. J. Rutherford, and K. P. Valavanis, "A survey of unmanned aerial vehicles (UAVs) for traffic monitoring," in *2013 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2013, pp. 2643–2666.
- [12] L. Chaimowicz and V. Kumar, "Aerial shepherds: Coordination among uavs and swarms of robots," in *Distributed Autonomous Robotic Systems 6*. Springer, 2007, pp. 243–252.
- [13] N. Frietsch, O. Meister, C. Schlaile, and G. Trommer, "Teaming of an ugv with a vtol-uav in urban environments," in *2008 IEEE/ION Position, Location and Navigation Symposium*. IEEE, 2008, pp. 1278–1285.
- [14] I. Rekleitis, G. Dudek, and E. E. Milios, "On multiagent exploration," in *Proceedings of Vision Interface 1998*, Vancouver, Canada, June 1998, pp. 455–461.
- [15] R. R. Murphy, E. Steimle, C. Griffin, C. Cullins, M. Hall, and K. Pratt, "Cooperative use of unmanned sea surface and micro aerial vehicles at hurricane wilma," *Journal of Field Robotics*, vol. 25, no. 3, pp. 164–180, 2008.
- [16] D. MacArthur and C. Crane, "Unmanned ground vehicle state estimation using an unmanned air vehicle," in *2007 International Symposium on Computational Intelligence in Robotics and Automation*, June 2007, pp. 473–478.
- [17] H. G. Tanner, "Switched uav-ugv cooperation scheme for target detection," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*, April 2007, pp. 3457–3462.

²afrl.cse.sc.edu/afrl/resources/code/birdseyeview

- [18] H. Duan and S. Liu, "Unmanned air/ground vehicles heterogeneous cooperative techniques: Current status and prospects," *Science China Technological Sciences*, vol. 53, no. 5, pp. 1349–1355, 2010.
- [19] C. Papachristos and A. Tzes, "The power-tethered uav-ugv team: A collaborative strategy for navigation in partially-mapped environments," in *22nd Mediterranean Conference of Control and Automation (MED)*. IEEE, 2014, pp. 1153–1158.
- [20] A. Rezelj and D. Skocaj, "Autonomous charging of a quadcopter on a mobile platform," in *Austrian Robotics Workshop*, 2015, pp. 65–66.
- [21] C. Minhua and G. Jiangtao, "Mobile robotics lab spring 2015 automatic landing on a moving target," 2015, https://robotics.shanghaitech.edu.cn/sites/default/files/files/robot_land_project2_report.pdf.
- [22] A. Chakrabarty, R. Morris, X. Bouysseunouse, and R. Hunt, "Autonomous indoor object tracking with the parrot ar. drone," in *International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2016, pp. 25–30.
- [23] A. Lawrence and A. Turchina, "Tag Tracking in ROS," 2014, <https://github.com/ablarry91/ros-tag-tracking>.
- [24] M. G. Prasad, S. Chandran, and M. S. Brown, "A motion blur resilient fiducial for quadcopter imaging," in *2015 IEEE Winter Conference on Applications of Computer Vision*. IEEE, 2015, pp. 254–261.
- [25] Y. Huang, M. Lva, D. Xiong, S. Yangb, and H. Lu, "An Object Following Method Based on Computational Geometry and PTAM for UAV in Unknown Environments," *IEEE International Conference on Information and Automation*, 2016.
- [26] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," in *European Conference on Computer Vision*. Springer, 2014, pp. 834–849.
- [27] C. Forster, M. Pizzoli, and D. Scaramuzza, "Svo: Fast semi-direct monocular visual odometry," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 15–22.
- [28] M. J. Milford, G. F. Wyeth, and D. Prasser, "RatSLAM: a hippocampal model for simultaneous localization and mapping," in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 1. IEEE, 2004, pp. 403–408.
- [29] A. Quattrini Li, A. Coskun, S. M. Doherty, S. Ghasemlou, A. S. Jagtap, M. Modasshir, S. Rahman, A. Singh, M. Xanthidis, J. M. O'Kane, and I. Rekleitis, "Experimental comparison of open source vision based state estimation algorithms," in *International Symposium on Experimental Robotics (ISER)*, Tokyo, Japan, 2016.
- [30] G. Klein and D. Murray, "Parallel tracking and mapping on a camera phone," in *8th IEEE International Symposium on Mixed and Augmented Reality, ISMAR*. IEEE, 2009, pp. 83–86.
- [31] M. Bazin, I. Bouguir, D. Combe, V. Germain, and G. Lassade, "Parallel tracking and mapping of a fleet of quad-rotor," *International Journal of Engineering and Technology*, vol. 9, no. 3, 2017.
- [32] R. Kurazume, S. Nagata, and S. Hirose, "Cooperative positioning with multiple robots," in *IEEE International Conference on Robotics and Automation*, vol. 2, 1994, pp. 1250–1257.
- [33] P. Bison, G. Chemello, C. Sossai, and G. Trainito, "Cooperative Localization using Possibilistic Sensor Fusion," in *IFAC Symposium on Intelligent Autonomous Vehicles*, Madrid, 1998, pp. 621–626.
- [34] L. Merino, F. Caballero, and P. Forssen, *Advances in Unmanned Aerial Vehicles*. Springer, 2007, ch. Single and multi-UAV relative position estimation based on natural landmarks, pp. 267–307.
- [35] D. Fox, W. Burgard, H. Kruppa, and S. Thrun, "Efficient multi-robot localization based on monte carlo approximation," *International Symposium of Robotics Research*, pp. 153–160, 1999.
- [36] F. Rivard and J. Bisson, "Ultrasonic relative positioning for multi-robot systems," *IEEE International Conference on Robotics and Automation*, pp. 323–328, 2008.
- [37] I. M. Rekleitis, G. Dudek, and E. Milios, "Probabilistic Cooperative Localization in Practice," in *IEEE International Conference on Robotics and Automation*, 2003, pp. 1907–1912.
- [38] I. Rekleitis, G. Dudek, and E. Milios, "Multi-Robot Exploration of an Unknown Environment, Efficiently Reducing the Odometry Error," in *International Joint Conference on A.I. (see also CIM Tech. Report TR-CIM-97-01)*, 1997.
- [39] M. Gilioli and K. Schilling, "Autonomous cooperative localization of mobile robots based on ranging systems," *SPIE Unmanned Ground Vehicle Technology*, vol. 5083, 2003.
- [40] A. Bahr, J. Leonard, and M. Fallon, "Cooperative localization for autonomous underwater vehicles," *Robotics Research*, vol. 28, no. 6, pp. 714–728, 2009.
- [41] C. Jennings, D. Murray, and J. Little, "Cooperative robot localization with vision-based mapping," in *IEEE International Conference on Robotics and Automation*, 1999, pp. 2659–2665.
- [42] C. Chang, S. Wang, and C. Wang, "Vision-based cooperative simultaneous localization and tracking," *IEEE International Conference on Robotics and Automation*, 2011.
- [43] H. Choset and J. Burdick, "Sensor based planning. I. The generalized Voronoi graph," in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 2. IEEE, 1995, pp. 1649–1655.