Localization of a Ground Robot by Aerial Robots for GPS-Deprived Control with Temporal Logic Constraints

Eric Cristofalo[1(✉)], Kevin Leahy[2], Cristian-Ioan Vasile[3], Eduardo Montijano[4], Mac Schwager[1], and Calin Belta[2]

[1] Stanford University, Stanford, CA, USA
ecristof@stanford.edu
[2] Boston University, Boston, MA, USA
[3] Massachusetts Institute of Technology, Cambridge, MA, USA
[4] Centro Universitario de la Defensa (CUD), Zaragoza, Spain

Abstract. In this work, we present a novel vision-based solution for operating a vehicle under Gaussian Distribution Temporal Logic (GDTL) constraints without global positioning infrastructure. We first present the mapping component that builds a high-resolution map of the environment by flying a team of two aerial vehicles in formation with sensor information provided by their onboard cameras. The control policy for the ground robot is synthesized under temporal and uncertainty constraints given the semantically labeled map. Finally, the ground robot executes the control policy given pose estimates from a dedicated aerial robot that tracks and localizes the ground robot. The proposed method is validated using a two-wheeled ground robot and a quadrotor with a camera for ten successful experimental trials.

Keywords: Vision-based localization · Temporal logic planning · Air-ground localization · Heterogeneous robot systems

1 Introduction

In this paper, we propose a solution to the following problem: localize and control a ground robot under temporal logic (TL) specifications in an environment with no global positioning infrastructure. Robots operating in the real world typically require accurate pose estimates to compute effective control actions, but in many cases, such as dense urban environments [1], GPS may be unavailable or unreliable. Furthermore, it is advantageous to consider an aerial robot for on-the-fly tracking of the ground robot because it can aid in terms of localization as well as obstacle avoidance, leaving the ground robot dedicated to other tasks. In this work, we present a vision-based, GPS-denied solution to this problem and demonstrate it experimentally with a sensor-deprived ground robot that performs a persistent monitoring task specified by TL, while being localized by a camera-equipped autonomous aerial vehicle (quadrotor). The solution is split into three major components: map building in unknown environments,

D. Kulić et al. (eds.), *2016 International Symposium on Experimental Robotics*, Springer Proceedings in Advanced Robotics 1, DOI 10.1007/978-3-319-50115-4_46

control synthesis under TL constraints, and localization during the mission. We use vision-based formation control to build the map from multiple aerial vehicles because we obtain a high fidelity mosaic map image without requiring SLAM or other complex mapping algorithms. Our algorithm synthesizes the ground robot's control policy based on a labeled version of the map and a TL specification. Finally, the ground robot executes the control policy while an aerial robot provides pose measurements.

Consider a robot that must perform the following task in an outdoor disaster site: "Periodically collect soil samples from the forest, then the beach. Deliver samples to researchers. Go to a charging station after tasks are complete. Always avoid known obstacles and restricted zones. Ensure that the uncertainty (measured by variance) of the robot's pose is always below $1\,\mathrm{m}^2$." Such a task may be specified using Gaussian distribution TL (GDTL) [2], a specification language that incorporates the robot's desired position as well as uncertainty. Unfortunately, the initial position of the robot is completely unknown and common methods to synthesize a control policy for the robot, even while operating under observation noise, will not be sufficient. Our solution alternatively requires the deployment of a small network of quadrotors with cameras to first map the space, prior to computing a control policy. Human operators then label the resulting map to capture the properties expressed in the specification. This process is known as *grounding*. Afterwards, our algorithm generates a feedback control policy to satisfy the temporal and uncertainty constraints encoded in the specification. With a map image and ground robot control policy, one quadrotor tracks and monitors the ground robot, providing it with pose information that it uses to execute the mission.

This work also considers the cooperation between ground and air vehicles and leverages their heterogeneous capabilities to jointly carry out a mission. While other research exists for cooperation among mixed teams of ground and air vehicles, existing research assumes the presence of GPS on either the ground vehicles [3] or on the aerial vehicles [1,4]. We, on the other hand, assume the robots are working in an environment with no external positioning framework whatsoever. Other work that has focused on planning without GPS, such as [5], uses the visual capabilities of an aerial vehicle to enhance a map built by a ground vehicle. In this work, we assume the map is built by a team of aerial vehicles using their high vantage point so that the ground vehicle can perform a specific task based on the resulting map. Further, unlike these works, in our work, the mission to be carried out is specified using GDTL, allowing for the encoding of much more complex missions, including specifying the uncertainty of the ground vehicle's localization.

Map building and localization in unknown environments could be formulated as in SLAM [6], where a robot uses its onboard sensor data—perhaps only vision [7]—to refine an estimation of its pose while building a map of the environment. Unfortunately, these algorithms are typically computationally demanding and require one or more sensing technologies which may not be feasible to include on a ground robot due to cost, weight, or hardware limitations. Using vision-based solutions from aerial cameras, on the other hand, allows for accurate pose

estimation in complicated environments while only employing cheap, readily-available RGB cameras. For example, homography-based visual servoing methods provide accurate localization with only the use of camera data [8]. In this work, we make use of homography-based consensus control methods [9] for the aerial vehicles to build a mosaic map, and monitor the ground robot with a Position-Based Visual Servoing (PBVS) control method designed to keep the robot in the field of view at all times while guaranteeing sufficient overlap with the map.

2 Technical Approach

We propose an end-to-end framework (see Fig. 1) that includes a specialized, two-wheeled ground robot and a team of aerial robots, i.e., N quadrotors, each equipped with a downward facing camera and an altimeter. The team of quadrotors are first responsible for building the map of the unknown environment using their onboard camera images. Then the ground robot operates under the computed optimal control policy with the measurements provided by a single quadrotor tracking it from above. The entire framework is divided into three sequential phases that include the following:

1. Generate a mosaic map image of the unknown environment using purely vision and homography-based formation control [9] with multiple quadrotors.
2. Label the generated map and define the mission specification (to be completed by human operator) and then automatically synthesize a satisfying control policy for ground robot using GDTL-Feedback Information RoadMaps, or GDTL-FIRM [2].
3. Simultaneously track and localize the ground robot with a single aerial vehicle using a homography-based pose estimation and position-based visual servoing control method.

2.1 Inter-image Homography

Map building and ground robot pose estimation rely on the inter-image homography, $\mathbf{H}_{ij} \in \mathbb{R}^{3\times3}$, which defines the linear transformation between co-planar

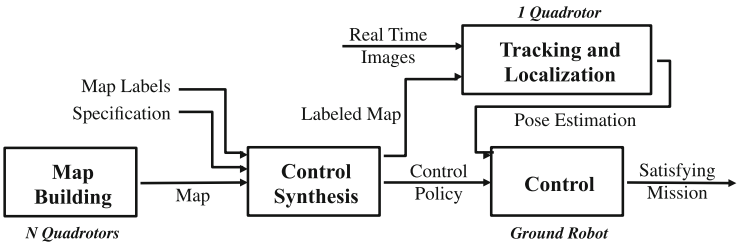


Fig. 1. The proposed framework includes three major components: (1) mapping in unknown environments, (2) control synthesis, and (3) online tracking and localization of a ground robot.

three-dimensional (3D) points described in two different coordinate frames, i.e., $\mathbf{P}_i = \mathbf{H}_{ij}\mathbf{P}_j$, where $\mathbf{P}_i \in \mathbb{R}^3$ and $\mathbf{P}_j \in \mathbb{R}^3$. The perspective projection of these 3D points yields the measured image features, $\mathbf{p}_i \in \mathbb{R}^2$ and $\mathbf{p}_j \in \mathbb{R}^2$, that are given by the cameras i and j, respectively. These two image features are related by the following homography, $\mathbf{p}_i = \tilde{\mathbf{H}}_{ij}\mathbf{p}_j$, where $\tilde{\mathbf{H}}_{ij} = \mathbf{K}\mathbf{H}_{ij}\mathbf{K}^{-1}$ is estimated using standard least squares estimation [10] with at least four matched image feature points, and $\mathbf{K}$ is the known calibration matrix of the identical cameras. In this work, we assume that all quadrotors are flying at a sufficiently high altitude to justify the co-planar requirements of points on the ground. Further, we assume that the cameras are always parallel to the ground – as with a hovering quadrotor. In this case, the *rectified* homography describes the transformation between two parallel, calibrated camera poses,

$$\mathbf{H}^r_{ij} = \begin{bmatrix} \cos(\psi_{ij}) & -\sin(\psi_{ij}) & -\frac{x_{ij}}{z_j} \\ \sin(\psi_{ij}) & \cos(\psi_{ij}) & -\frac{y_{ij}}{z_j} \\ 0 & 0 & 1 - \frac{z_{ij}}{z_j} \end{bmatrix}, \tag{1}$$

where $[x_{ij}, y_{ij}, z_{ij}, \psi_{ij}]^T \in \mathbb{R}^4$ is the estimated parallel pose of camera j in the frame of camera i. In practice, we guarantee the parallel camera assumption by removing the roll and pitch effect of a translating quadrotor from the acquired image, i.e., $\mathbf{H}^r_{ij} = \mathbf{R}_{\theta_i}\mathbf{R}_{\phi_i}\mathbf{K}^{-1}\tilde{\mathbf{H}}_{ij}\mathbf{K}\mathbf{R}^T_{\phi_j}\mathbf{R}^T_{\theta_j}$, given the roll, ϕ, and pitch, θ, of each quadrotor. We extract the relative position from the last column of $\mathbf{H}^r_{ij}$, given the altitude of the cameras provided by the altimeter, and the relative orientation from the upper 2×2 block of $\mathbf{H}^r_{ij}$.

2.2 Homography-Based Formation Control

Homography-based formation control [9] drives the team of quadrotors that generates the high fidelity mosaic map image, which is a composite image of the quadrotors' onboard images while in formation. The consensus-based kinematic control laws that drive the formation of quadrotors to their desired relative pose, $[x^*_{i,j}, y^*_{i,j}, \psi^*_{i,j}]^T$, are functions of the computed rectified homography from Eq. (1), i.e.,

$$w_{z_i} = K_w \sum_{j \in \mathcal{N}_i} \left(\arctan\left[\frac{[\mathbf{H}^r_{ij}]_{21}}{[\mathbf{H}^r_{ij}]_{11}} \right] - \psi^*_{ij} \right), \tag{2}$$

$$\begin{bmatrix} v_{x_i} \\ v_{y_i} \end{bmatrix} = K_v \sum_{j \in \mathcal{N}_i} \left(\begin{bmatrix} [\mathbf{H}^r_{ij}]_{13} \\ [\mathbf{H}^r_{ij}]_{23} \end{bmatrix} - \begin{bmatrix} x^*_{ij} \\ y^*_{ij} \end{bmatrix} \right), \tag{3}$$

$$v_{z_i} = K_v \sum_{j \in \mathcal{N}_i} \left(1 - [\mathbf{H}^r_{ij}]_{33} \right), \tag{4}$$

where $[v_{x_i}, v_{y_i}, v_{z_i}]^T$ is the translational velocity control and w_{z_i} is the rotational velocity control about the z-axis of the quadrotor, i.e., its yaw. Note that the element in row a and column b of $\mathbf{H}^r_{ij}$ is denoted by $[\mathbf{H}^r_{ij}]_{ab}$. The relative yaw does not affect z_{ij}, therefore, the relative altitude can be controlled towards zero using $[\mathbf{H}^r_{ij}]_{33}$. The team produces the mosiac map of the environment when

the quadrotors reach the chosen formation that yields sufficient image overlap for accurate pose estimation and large enough field of view to cover the region of interest in the environment. It is worth noting that this component of our solution framework could be omitted if given a high resolution map, such as a satellite image.

2.3 GDTL-FIRM

The GDTL-FIRM algorithm computes the optimal control action for the ground robot under a GDTL specification given that the previously computed map has been labeled and the specification has been provided. We assume that a human operator labels the map built by the aerial vehicles. Alternately, this labeling could be accomplished automatically by a segmentation and classification algorithm. We utilize the work of [2] to synthesize the control polices for the ground robot with temporal and uncertainty constraints. In brief, the sampling-based algorithm generates a transition system in the belief space and uses local feedback controllers to break the curse of history associated with belief space planning. The algorithm is based on Feedback Information RoadMaps (FIRMs), where points are sampled directly in the state space and feedback controllers are used to stabilize the system about nodes in the roadmap, thus inducing a roadmap in the belief space. A product Markov Decision Process (MDP) between the transition system and the Rabin automaton encoding the GDTL task specification is used to compute the switching control policies. Finally, the MDP is queried for the existence of satisfying control policies of high enough probability.

2.4 Robot Tracking and Localization

The ground robot executes its mission in the environment by traversing the transition system generated in the previous phase while employing an Extended Kalman Filter (EKF) to estimate its position with measurements provided by the dedicated aerial vehicle. A localization marker on the ground robot includes two distinctly colored patches that aid in estimating its planar position and orientation in the environment frame. During localization, the quadrotor first localizes the centroid of each patch in the quadrotor's image frame as two image features, $(\mathbf{p}_1^q, \mathbf{p}_2^q)$. The quadrotor simultaneously calculates the rectified homography between the quadrotor's image frame (q) and the mosaic map image frame (m), i.e., $\mathbf{H}_{qm}^r$, to estimate the relative pose between the quadrotor and the map. The quadrotor projects the robot's pose in the image frame $(\mathbf{p}_1^q, \mathbf{p}_2^q)$ to the map frame $(\mathbf{p}_1^m, \mathbf{p}_2^m)$ using $\mathbf{H}_{qm}^r$. The quadrotor finally computes the ground robot's final pose in the environment frame (e), given by (x, y, θ), by linearly interpolating $(\mathbf{p}_1^m, \mathbf{p}_2^m)$ with the dimensions of the map image – in pixels – and the known dimensions of the environment – measured in meters. The centroid of the projected features yields the position, (x, y), while the orientation, θ, is calculated using the line that connects the two projected features.

Meanwhile, a 2D kinematic PBVS controller maneuvers the aerial robot to track the ground robot while simultaneously keeping sufficient overlap with the

mosaic map image for an accurate homography estimation. Recall that the field-of-view of the individual cameras is not sufficient to view the entire environment, hence the requirement for the composite map image. Homography-based control drives the quadrotor into a desired position above the environment that is defined by the estimated position of the ground robot, (x, y). The quadrotor's position is further constrained to a rectangle, $R = [x_{min}, x_{max}] \times [y_{min}, y_{max}]$, where the boundaries of R affect the amount of desired overlap with the mosaic image. For example, setting the boundaries equal to the dimensions of the environment will drive the quadrotor directly over the ground robot, thus degrading the homography estimate when hovering near the environment's edges. Conversely, setting the boundaries equal to zero would keep the quadrotor coincident with the mosaic image frame and will lose coverage when the ground robot is near the edge of the environment. The ideal boundary values for a downward facing camera allows the camera to move just far enough to see the entire environment, i.e.,

$$\begin{bmatrix} x_{min} \\ y_{min} \end{bmatrix} = -\begin{bmatrix} x_{max} \\ y_{max} \end{bmatrix} = \begin{bmatrix} \frac{w_e - {}^e w_q}{2} \\ \frac{h_e - {}^e h_q}{2} \end{bmatrix}, \tag{5}$$

where (w_e, h_e) are the width and height of the environment in meters, $({}^e w_q, {}^e h_q)$ are the dimensions of the quadrotor's image frame, (w_q, h_q), after being projected into the environment frame. This projection is computed as,

$$\begin{bmatrix} {}^e w_q \\ {}^e h_q \\ A \end{bmatrix} = A\mathbf{K}^{-1} \begin{bmatrix} w_q \\ h_q \\ 1 \end{bmatrix}, \tag{6}$$

given the camera's altitude, A, and camera calibration matrix, $\mathbf{K}$. The ideal rectangle size for our camera (640×360) at the desired experiment altitude of 1.8 m is approximately 0.85×1.45 m. Unfortunately, our camera is not downward-facing, therefore we expand R to 0.85×2.0 m to ensure proper coverage. Finally, we introduce an optional offset, $\mathbf{x}_{offset}$, that measures the center of mosaic map image's virtual position in space with respect to the quadrotor's frame. We use an offset 0.75 m in the positive x-direction of the local quadrotor frame (see Fig. 2) to account for the forward-facing camera.

The final controller is similar to the homography-based formation controller in Sect. 2.2. In fact, the yaw controller of Eq. (2) and the altitude controller of Eq. (4) remain the same with a desired relative pose equal to zero. The planar control vector is calculated as the following,

$$\begin{bmatrix} v_x \\ v_y \end{bmatrix} = K_v \left(\begin{bmatrix} [\mathbf{H}^r_{qm}]_{13} \\ [\mathbf{H}^r_{qm}]_{23} \end{bmatrix} - \begin{bmatrix} linint(x, (0, w_e), (x_{min}, x_{max})) - x_{offset} \\ linint(y, (0, h_e), (y_{min}, y_{max})) - y_{offset} \end{bmatrix} \right), \tag{7}$$

where $linint(\cdot)$ is the linear interpolation function that transforms the ground robot's environmental position into the quadrotor's desired position within R.

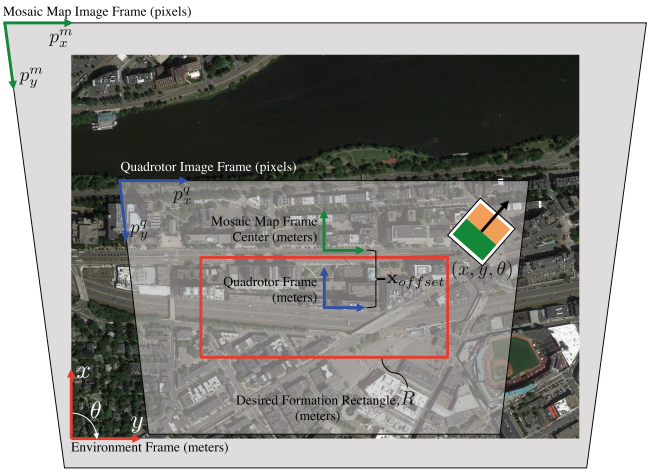


Fig. 2. Coordinate frame definitions for the PBVS controller from Eq. (7) include the: environment frame, mosaic map image frame, quadrotor image frame, mosaic map frame center, and quadrotor frame. The quadrotor estimates the ground robot's pose (x, y, θ) by transforming the pose in the quadrotor image frame to the environment frame. The quadrotor manuevers within R based on the ground robots's pose in the environment frame. The quadrotor local frame and mosaic map frame center are defined with the same orientation as the environment frame.

3 Results and Experiments

We validate all three phases of this framework by executing a complete mission experiment with a heterogeneous team of autonomous robots. The phases are completed in the order specified in Sect. 2 due to the dependence on the results from previous phases. We first detail our map building results with a mosaic map that is generated using the homography-based formation control and two quadrotors with cameras that do not have access to GPS. GDTL-FIRM synthesizes the control policy for a ground robot with nonlinear *unicycle* dynamics in the environment for a GDTL specification over belief states associated with the measurement of the robot's position. Finally, a quadrotor successfully tracks and localizes the ground robot while it completes the previously defined mission.

3.1 Experimental Setup

We perform experiments in the Boston University Robotics Laboratory. We use a map of Boston University's campus, located in Boston, MA, USA, that includes parts of Charles River, Massachusetts Turnpile, Fenway Stadium, and BU Central campus. We utilize the real landmarks in this map to formulate our specification. This map is chosen because it has sufficient detail and texture to allow for adequate feature matching (e.g., white buildings at the bottom of the map) as well other minimal feature regions (e.g., the Charles River). The physical map

is printed on a 12×16 ft^2 vinyl banner. We utilize an Optitrack motion capture system[1] for obtaining ground truth measurements.

The ground robot is a two-wheeled DrRobot X80Pro[2] with no onboard sensing. We fit the ground robot with an identifying marker composed of two uniquely colored patches in the YUV color space for planar position and orientation localization (see Fig. 5). Parrot Bebop quadrotors[3] are the aerial vehicles used for map building, and later, tracking. The Bebop is an off-the-shelf quadrotor platform with a suite of sensors that include an Inertial Measurement Unit (IMU), a downward-facing pinhole camera for optical flow stabilization, an ultrasonic sensor for altitude measurements, and a 180° wide-angle 14 megapixel forward-facing camera. The large forward-facing camera produces a 640×360 pixel stabilized video feed that can be 'steered' within the field-of-view of the wide-angle lens to produce a 'virtual camera' video feed. We position the virtual camera at the maximum angle of θ_{bebop} measured about the y-axis of the quadrotor (see Fig. 3(a)), where $\theta_{bebop} \approx 50°$, and rectify the image for this angle.

The Robot Operating System (ROS) [11] handles all communication on a local area network via Wi-Fi. We control the quadrotors from a base station computer running the ROS *Bebop Autonomy* package [12] which incorporates Parrot's open-source SDK. The computer also acquires and processes image frames from the quadrotors' real-time video stream via the OpenCV libraries [13]. Independent ROS nodes handle the individual quadrotors for the formation flight, demonstrating the distributed control. Independent ROS nodes also handle the quadrotor and ground robot control during the tracking phase. In this phase, separate quadrotor nodes handle the image processing for robot localization, pose estimation via homography, and the control. The ground robot node executes the local control and EKF estimation of the ground robot given its pose estimate and nonlinear dynamics. All vision computations are performed on an Ubuntu 14.04 machine with an Intel Core i7 CPU at 2.4 GHz and 8GB RAM.

3.2 Formation Control and Map Generation

We utilize a team of two quadrotors to reach a desired formation where, $y^*_{1,2} = -y^*_{2,1} = 1.2$ m, and all other desired relative poses are set to zero (see Fig. 3(a)). This formation is carefully chosen because it ensures the pair of aerial cameras have enough overlap for accurate relative pose estimation while guaranteeing a complete view of the environment. All quadrotors are flown to a desired height of 1.8 m. The quadrotors reach the desired formation (Fig. 3(c)) from the initial conditions (Fig. 3(b)) in approximately 15 s. From this point, the user has the ability to control one vehicle in the formation to fine tune the result of the online mosaic map, which is displayed at approximately 2.5 Hz. In this experiment, the operator maneuvers quadrotor 1 until the left edge of the map is

[1] Natural Point Optitrack: https://www.optitrack.com.
[2] DrRobot X80Pro: http://www.drrobot.com/products_item.asp?itemNumber=x80pro.
[3] Parrot Bebop: http://www.parrot.com/products/bebop-drone/.

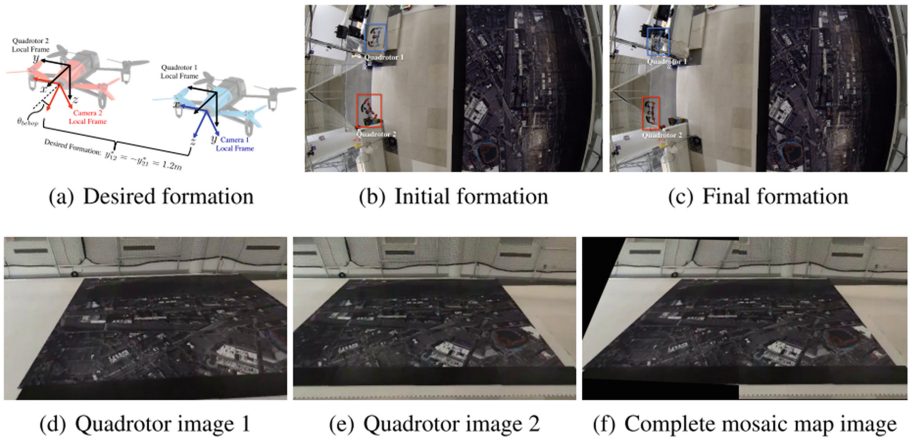


Fig. 3. Final mosaic map result using the homography-based formation control method. Note that quadrotor and camera coordinate systems are only labeled once in (a) for clarity.

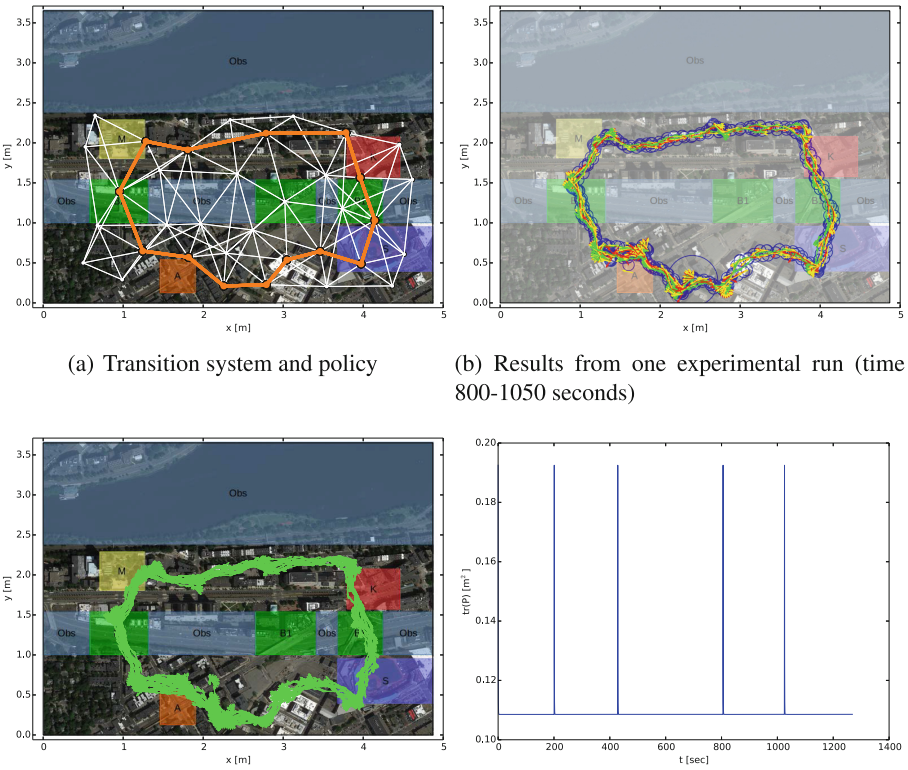
completely visible and then releases it to autonomous control again. Meanwhile, the formation control law in Sect. 2.2 controls quadrotor 2. The onboard images at the final desired formation (Figs. 3(d)–(e)) were used to generate the final mosaic map image shown in Fig. 3(f).

3.3 GDTL-FIRM

The specification for the ground robot is encoded with GDTL and is given as the following: "Always avoid all obstacles, i.e., Charles river and Massachusetts Turnpike. Always eventually visit Kenmore Square, Marsh Plaza, Audubon Circle, and Fenway Stadium. From Kenmore Square or Marsh Plaza, Bridge2 (St Mary's St) can not be used to visit Audubon Circle or Fenway Stadium. From Audubon Circle or Fenway Stadium, Bridge1 (Beacon Ave or Brookline Ave) can not be used to visit Kenmore Square or Marsh Plaza. Always keep uncertainty about the robot's pose below $0.9\,\mathrm{m}^2$, and on bridges, the uncertainty must be below $0.6\,\mathrm{m}^2$, where uncertainty is measured as the trace of the estimation pose covariance matrix." Fig. 4(a) shows the resulting transition system and control policy, computed by the algorithm from [2]. The transition system has 35 nodes and 226 edges while the product automaton has 316 nodes and 3274 edges. The algorithm executed in approximately 62.24 s.

3.4 Pose Estimation and Mission Execution

The ground robot executes the mission using the previous control policy and quadrotor for localization. Initially, the quadrotor takes off from a position where the camera's field of view is facing towards the ground robot. The homography-based localization and quadrotor control (Sect. 2.4) begin once the ground



(a) Transition system and policy

(b) Results from one experimental run (time 800-1050 seconds)

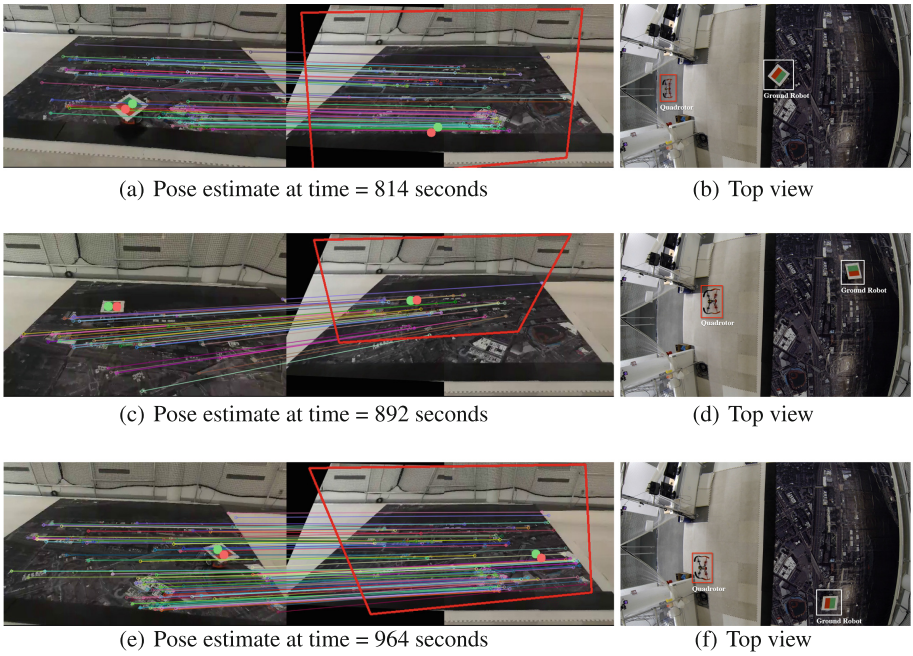
(c) Ground truth for ten laps

(d) Covariance for ten laps

Fig. 4. FIRM-GDTL results plotted over the ground truth environment image. (a) shows the transition system in white and the policy in orange. (b) shows the ground truth in green, the measurement in yellow, the estimated pose in red, and the covariance ellipses in blue. (c) shows the ground truth in green for all runs. (d) shows the covariance for all runs. The spikes in covariance indicate the beginning of a new run after a quadrotor battery had been replaced. We initialize the covariance to an arbitrarily large value at time step 0 that drastically decreases with the first pose measurement from the quadrotor at time step 1.

robot's marker has been detected. The ground robot localization estimates update at approximately 3.5 Hz. We show an example of the robot tracking and pose estimation for three time steps in Fig. 5. It is clear that the control method tracks the ground robot during its route with enough image resolution to detect the robot's patches and also maintains the required overlap with the mosaic map image.

Figure 5 also illustrates the final pose estimation in the mosaic map frame. It is important to note that the ground robot sits 0.2 m above the map, therefore projecting the image features of the ground robot's marker directly into the map frame would add significant error to the final estimation. The image features



(a) Pose estimate at time = 814 seconds (b) Top view

(c) Pose estimate at time = 892 seconds (d) Top view

(e) Pose estimate at time = 964 seconds (f) Top view

Fig. 5. Pose estimation results of live tracking and localization are shown in (a, c, e) with onboard images (left) and the mosaic map image (right). The corresponding top views of the experiment is shown in (b, d, f), respectively. The image matches and pose estimations are drawn for visualization purposes.

are instead offset to the map plane before projecting the features to the mosaic map to satisfy the homography's planar assumption. We determine this offset by measuring the pose estimation error at the extremes of the map and interpolating for the correction as a function of the estimated pose.

We ran the mission five times due to the limitations of the quadrotor battery, yielding ten complete laps of the environment and four partial laps, all of which satisfied the GDTL specification. We show an example run of 2.5 laps in Fig. 4(b) that displays the ground robot's ground truth pose, estimated pose, measured pose, and uncertainty. We check for satisfaction by inspecting the ground truth of all experimental runs to ensure the robot has reached each region appropriately while avoiding obstacles (Fig. 4(c)). Moreover, the covariance of the robot's estimate for all experimental runs is safely below the minimum 0.6 requirement, thus satisfying the specification (Fig. 4(d)).

4 Conclusion

The main experimental insight gained from this work is how to feasibly break the dependence on external positioning information while controlling robots under

TL specifications. Specifically, we are interested in studying the satisfaction of GDTL specifications by (ground) robots operating under uncertainty. Encoding specifications with GDTL is advantageous because it defines performance goals for the uncertainty of the system, allowing us to complete high-level missions under noisy measurements. This work also gives insight into the formulation of a mobile vision-based sensing method for control under TL specifications.

Another technical insight stems from the effects of using off-the-shelf equipment in this framework since airborne cameras are a cheap, light weight sensor solution that allow for high fidelity 3D pose estimation. We show that inexpensive and widely available ground and aerial robots can be used to perform complex missions with TL and uncertainty constraints, therefore adding flexibility in future applications. Moreover, we consider a simple dynamic sensor that is far more reconfigurable than a fixed-camera network alternative.

The experimental setup for vision-based control with aerial vehicles also provided valuable experimental insight. The lighting conditions of the flying space proved to be critical and had to be carefully modified to reduce glare from the reflective vinyl banner material. The oblique angle of the quadrotor's onboard camera also complicated the control strategies since we could not rely on the standard down-ward facing camera assumptions. Lastly, this vision-based technique does encounter pose estimation innacuracies when the quadrotor cameras have very poor resolution compared to the map. Further, the entire pipeline depends on the success of feature matching that encounters problems at drastic resolution differences. However, these experiments show that our framework is well suited for remote outdoor scenarios where aerial vehicles or satellite imagery could serve as the map and only camera-outfitted aerial vehicles are required for localization.

Acknowledgements. E. Cristofalo was supported in part by the 2015 National Defense Science and Engineering Graduate (NDSEG) fellowship. This work was also supported by US grants NSF CNS-1330008, NSF IIS-1350904, NSF NRI-1426907, NSF CMMI-1400167, ONR N00014-12-1-1000, and Spanish projects DPI2015-69376-R (MINECO/FEDER) and SIRENA (CUD2013-05). We are grateful for this support.

References

1. Hsieh, M.A., Cowley, A., Keller, J.F., Chaimowicz, L., Grocholsky, B., Kumar, V., Taylor, C.J., Endo, Y., Arkin, R.C., Jung, B., et al.: Adaptive teams of autonomous aerial and ground robots for situational awareness. J. Field Robot. **24**(11–12), 991–1014 (2007)
2. Vasile, C.I., Leahy, K., Cristofalo, E., Jones, A., Schwager, M., Belta, C.: Control in belief space with temporal logic specifications. In: Proceedings of the 2016 Conference on Decision and Control (CDC). IEEE (2016, to appear)
3. Vaughan, R.T., Sukhatme, G.S., Mesa-Martinez, F.J., Montgomery, J.F.: Fly spy: lightweight localization and target tracking for cooperating air and ground robots. In: Distributed Autonomous Robotic Systems 4, pp. 315–324. Springer, Japan (2000)

4. Grocholsky, B., Keller, J., Kumar, V., Pappas, G.: Cooperative air and ground surveillance. Robot. Autom. Mag. **13**(3), 16–25 (2006)
5. Forster, C., Pizzoli, M., Scaramuzza, D.: Air-ground localization and map augmentation using monocular dense reconstruction. In: Proceedings of the 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 3971–3978. IEEE (2013)
6. Thrun, S., Leonard, J.J.: Simultaneous localization and mapping. In: Springer Handbook of Robotics, pp. 871–889. Springer, Heidelberg (2008)
7. Newcombe, R.A., Lovegrove, S.J., Davison, A.J.: DTAM: dense tracking and mapping in real-time. In: Proceedings of the 2011 International Conference on Computer Vision (ICCV), pp. 2320–2327. IEEE (2011)
8. Benhimane, S., Malis, E.: Homography-based 2d visual servoing. In: Proceedings of the 2006 International Conference on Robotics and Automation (ICRA), pp. 2397–2402. IEEE (2006)
9. Montijano, E., Cristofalo, E., Zhou, D., Schwager, M., Sagues, C.: Vision-based distributed formation control without an external positioning system. Trans. Robot. **32**(1), 339–351 (2016)
10. Ma, Y., Soatto, S., Kosecka, J., Sastry, S.S.: An Invitation to 3-d Vision: From Images to Geometric Models, vol. 26. Springer Science & Business Media, New York (2012)
11. Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., Ng, A.Y.: ROS: an open-source robot operating system. In: ICRA Workshop on Open Source Software, vol. 3, p. 5 (2009)
12. Monajjemi, M.: Bebop autonomy (2015). https://github.com/AutonomyLab/bebop_autonomy
13. Bradski, G., et al.: The opencv library. Doctor Dobbs J. **25**(11), 120–126 (2000)