2018 IEEE International Conference on Robotics and Automation (ICRA)
May 21-25, 2018, Brisbane, Australia

Vision based Collaborative Path Planning for Micro Aerial Vehicles

Sai Vemprala and Srikanth Saripalli[1]

***Abstract*— In this paper, we present a collaborative path-planning framework for a group of micro aerial vehicles that are capable of localizing through vision. Each of the micro aerial vehicles is assumed to be equipped with a forward facing monocular camera. The vehicles initially use their captured images to build 3D maps through common features; and subsequently track these features to localize through 3D-2D correspondences. The planning algorithm, while connecting start locations to provided goal locations, also aims to reduce the localization uncertainty of the vehicles in the group. To achieve this, we develop a two-step planning framework: the first step attempts to build an improved map of the environment by solving the next-best-view problem for multiple cameras. We express this as a black-box optimization problem and solve it using the Covariance Matrix Adaption evolution strategy (CMA-ES). Once an improved map is available, the second stage of the planning framework performs belief space planning for the vehicles individually using the rapidly exploring random belief tree (RRBT) algorithm. Through the RRBT approach, the planner generates paths that ensure feature visibility while attempting to optimize path cost and reduce localization uncertainty. We validate our approach using experiments conducted in a high visual-fidelity aerial vehicle simulator, Microsoft AirSim.**

I. INTRODUCTION

Micro aerial vehicles (MAV) are being widely used in robotic applications due to advantages of size, weight, agility of flight as well as ease of prototyping. As the applications for MAVs grow complex, accurate localization becomes necessary, in which regard, vision sensors have demonstrated good potential. The decreasing size and ubiquity of cameras on most MAV platforms make vision-only localization a good candidate for MAV pose estimation. Furthermore, vision localization techniques that involve feature based methods are suitable for groups of MAVs. Feature data provides a simple way to transmit and receive sensing information, and lets multiple MAVs collaborate with each other to localize more accurately. Collaboration can also reduce computational load on individual MAVs.

Given localization information, generating trajectories for a vehicle from a start pose to a goal pose constitutes the problem of path planning. As sensor measurements, localization and motion all contain uncertainty, the planning algorithm needs to account for these uncertainties to ensure safe navigation. For example, in vision based applications, highly textured or well-lit areas offer higher fidelity of measurements than others. Hence, incorporating knowledge of this possible uncertainty during future states into the planning framework is important to compute robust plans.

[1]Sai Vemprala and Srikanth Saripalli are with the Department of Mechanical Engineering, Texas A&M University, College Station, Texas, USA svemprala@tamu.edu, ssaripalli@tamu.edu

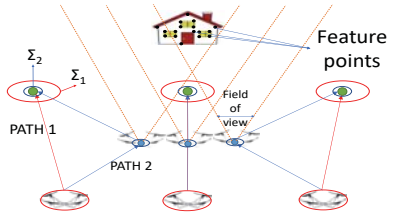

Fig. 1: Example of covariance minimization for multiple vehicles observing the same scene using vision. All vehicles start off with an initial covariance (red ellipses). Path 1 (red)indicates a direct, low travel cost path with minimal effect on uncertainty whereas path 2 (blue) minimizes final uncertainty by obtaining better relative measurements, making it the desired result.

Finally, for systems that collaborate in order to localize, a certain degree of collaboration at the planning level is also necessary.

In this paper, we seek to implement a framework to perform uncertainty aware path planning for multiple MAVs that collaborate between each other for vision based localization. We consider the MAVs to be equipped with forward facing cameras trying to localize by observing a global feature map, thus navigating fully in 3D space. This is built upon our previous work on collaborative localization [1], where we present an approach for localization of multiple aerial vehicles with respect to each other through feature detection and matching. Our vision based localization framework has two main stages: in the first stage, overlapping features between MAVs are utilized to build a 3D map of the surroundings. Once a 3D map is available, the MAVs individually perform pose estimation using 3D-2D correspondences. Similar to this, our planning approach also uses a two-fold strategy: in the first step, given partial knowledge of the surroundings, the algorithm computes viewpoints for each vehicle such that images captured from these locations would result in significant map improvement, and in the second step, the planner computes paths for each MAV from its position to a specified goal, while ensuring feature visibility and proximity to feature rich areas, thus minimizing the uncertainty of the MAVs.

II. RELATED WORK

Path planning for robots is a well-studied problem in the literature. Recently, in order to incorporate uncertainty estimation and reduction into the planning stage, several approaches have considered 'belief space' planning in the context of sampling based planning algorithms. The belief road map (BRM) algorithm, presented in [2], builds a roadmap of samples along with simulated measurements along candidate paths. In [3], belief space planning is per-

formed assuming maximum likelihood observations which are then used to perform trajectory optimization. The rapidly exploring random belief tree algorithm [4], which we employ in this paper, combines the a rapidly exploring random graph and belief information to obtain paths that seek to minimize uncertainty of the vehicle. Other belief space planning methods include direct trajectory optimization such as the one in [5] which calculates locally optimal control policies from an initial nominal path, incorporating motion uncertainty in a roadmap framework (FIRM) [6], and a localization-aware version of the RRBT algorithm [7]. In [8], the author presents a generalized framework for multi-robot belief space planning in unknown spaces by modeling future inferences and expressing indirect multi-robot constraints.

Recent work has also investigated belief space planning in the challenging domain of aerial vehicles. In [9], the authors use RRBTs for uncertainty aware MAV navigation, where the aerial vehicle is equipped with a downward facing camera. In [10], a perception aware planning approach is discussed for aerial vehicles, where dense, direct methods are used and photometric information of a scene is used as a metric. Next-best-view approaches have been developed in the literature for model improvement through surface texture appearance optimization [11], for multiple collaborative sensors to perform dense reconstruction [12] and more recently, Bircher et al [13] presented a receding horizon next best view planner that can perform 3D exploration for a micro aerial vehicle equipped with a stereo camera.

In this paper, we present an approach that is a combination of both next-best-view planning and belief space planning, thus attempting to perform both mapping and navigation in a way that reduces localization uncertainty. In vision based localization, the accuracy of estimation depends both on initial maps generated as well as effective tracking of salient features during flight: hence an approach that can optimize both the reconstruction and navigation has the potential to result in reduced overall uncertainty.

III. SYSTEM DESCRIPTION

In this section, we briefly describe the collaborative localization framework for MAVs. This idea has been discussed in our previous work [1] [14], where we consider MAVs equipped with forward facing monocular cameras as our target platforms and present a framework that is capable of collaborative pose estimation through a combination of single camera pose estimation (using 3D-2D correspondences) and multi-camera pose estimation (using overlapping fields of view). The system is initialized with an estimate of the scale of the environment. This is used to build a map that contains a possibly partial representation of the surroundings through feature matching and sequential triangulation; which is shared between all vehicles, acting as a 'global map'. Periodically, the MAVs use each other's images and overlapping feature information (keypoint locations and descriptors) to update the environment and the scale.

The first stage of our localization system can be seen illustrated in figure 2. At every time step, MAVs capture images

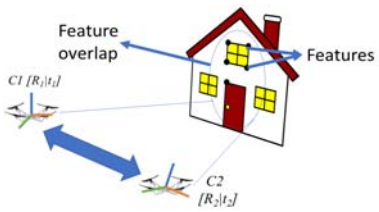

Fig. 2: Collaborative localization between two MAVs with cameras. Feature overlap (dotted circle) is essential to compute 6DoF poses of the cameras relative to each other as well as triangulation to generate 3D maps which are then shared by both MAVs.

and detect salient features in each image. This information is used to match correspondences with other vehicles or to track features from a globally available map, and eventually to obtain poses for each vehicle through the perspective N point algorithm (PNP). During localization, the uncertainty of a MAV's pose depends on two main factors: the accuracy of the map being reconstructed as well as the uncertainty coming from the PNP algorithm, both of which, in turn, depend on factors such as scene geometry and number of visible features.

IV. COLLABORATIVE PATH PLANNING

We decouple the path planning problem into two steps. The first step involves using multiple vehicles and computing a set of viewpoint poses for at least two of them, in order to generate a potentially improved map of the environment. The second step is to utilize this new map as the source of localization information, and perform belief space planning for individual vehicles, with an aim to reduce pose uncertainty as they navigate to their goals.

A. Next best view planning for multiple vehicles

Finding the best placement of a vision sensor in order to improve an existing representation of a scene is termed as the 'next best view' (NBV) problem [15]. Depending on the context, NBV can be applied for various types of vision sensors such as monocular cameras, stereo cameras and so on. In our application, we aim to solve the NBV problem for multiple vehicles (cameras). Given n number of vehicles, and a set of starting positions from which partial knowledge of a scene is available, we express the next-best-view problem as the problem of calculating sensor poses x_i for all $i \in [1, n]$ such that certain parameters relating to 3D reconstruction are optimized. We express this as the following minimization problem for a set of vehicles over a space of poses:

$$\min_{x_i} f(C, X) \tag{1}$$

where C is a cost function that encodes various parameters relating to 3D reconstruction and X is a set of 3D points representing the scene. For the sake of simplicity, we select candidate poses for each MAV/sensor only in 4DoF: i.e., 3D position (x, y, z) and the yaw angle ψ. Hence, a vehicle's pose is represented as:

$$c_i = [p_i, \psi_i] | p_i \in R^3, \psi_i \in SO(3) \tag{2}$$

1) Heuristics for optimization: We recall here that the reconstruction phase involves image captures from each MAV at a specific location in 3D space and isolating common features between all of them. Each MAV pose x_i represented as above has a 3D-2D projection associated with it arising from the image captured from that pose. When an initial set of 3D points X is known, projections of X on an image plane can be computed from any pose in 3D space. Given this information, we attempt to create a set of heuristics for any arbitrary pose, which when combined, describe the 'quality' of an image captured from that pose, and subsequently the quality of 3D reconstruction or localization. We list the heuristics below:

1) *Visibility*: In the next best view planner, the visibility factor calculates the inverse of the ratio of visible features from an MAV to the number of actual features that are part of the global map. When applied to a group of MAVs, this factor ensures that the features are visible at least from all the members together, while still attempting to maximize the number of features visible from each MAV.
2) *Span*: This factor calculates the inverse of the ratio of area occupied by features to the total pixel area of the image. For either PNP or relative pose computation, large evenly distributed regions of salient features result in accurate pose computation, whereas all features located in one corner of the image or in a line would result in inaccuracy or degenerate cases. Through this factor, poses that result in features spanning large areas of the image plane are favored as opposed to viewpoints from farther away. Maximizing this factor also increases the possibility of bringing new features into view.
3) *Overlap*: This factor calculates the inverse of the ratio between number of features that are common between the fields of view of two MAVs to the total number of features in the global map. Overlap between all the images is rewarded the most, in cases where that is not possible because of other factors, at least a sufficient amount of pairwise overlap is prioritized. Minimizing this factor maximizes the overlap.
4) *Baseline*: According to stereo geometry, the baseline of a pair of cameras is proportional to the amount of depth error. Hence, the camera baselines must be scaled in a way that takes into account their distance to the observed scene geometry (depth). This factor calculates the deviation of the baseline to scene distance ratio from a desired value.
5) *Vergence angle*: For a given binocular view, the vergence angle defines the angle between the projected rays from the cameras at their intersection point. A high vergence angle creates higher overlap, but could cause inaccuracies with feature matching as disparity is relative to vergence angle. Minimization of this parameter translates to favoring a lower vergence angle where possible.
6) *Collisions and occlusion*: Configurations that can potentially result in collisions between the vehicles or the scene (distance between vehicles below a certain threshold) and occlusion (one vehicle blocking the view of another) are penalized.

Finally, we define our optimization function as a weighted combination of the above metrics, where a minimum value indicates the best set of viewpoints. The weights were set equally in our implementation, but can be adjusted to make the algorithm prioritize one factor highly over the others. This combined optimization function takes a set of candidate poses for all vehicles being used as its input and outputs a value of the objective function that encodes all the heuristics.

Yet, for a given sampling space where each sample is a combination of candidate poses for multiple vehicles, the value of the function depends heavily on scene geometry and is sensitive to small changes in pose parameters: which makes it hard to estimate an algebraic model of the function. Given this constraint, we consider it as a black-box optimization problem and use a derivative-free method to solve the problem.

2) CMA-ES optimization: To minimize the NBV objective function, we utilize the CMA-ES (Covariance Matrix Adaptation Evolution Strategy) algorithm [16]. CMA-ES is an evolutionary algorithm that is suited for high dimensional black box optimization problems. CMA-ES works by sampling a random distribution of λ possible solution at every iteration, which form its population.

$$x_k = x_{fit} + \nu; \forall k \in [1, \lambda] \tag{3}$$

x_{fit} is the best solution obtained in the previous iteration and ν is a random variable from a multivariate normal distribution with covariance C. The algorithm determines the strongest sample by evaluating the function at all these samples; and once these values are available, then mean of the distribution is updated such that the likelihood of strong samples is maximized. On the other hand, the covariance matrix of this distribution is also updated at every iteration, such that the direction of the sampling is biased towards stronger solutions. We initialize the algorithm with the initial poses of the MAVs acting as the guess for the first iteration.

B. Belief space planning

As the MAVs in this context rely solely upon vision based localization, the plans to move from any start to goal locations need to consider the constraints placed due to monocular vision and the uncertainty arising from the localization. We recall here that this localization is performed using the PNP algorithm and 3D-2D correspondences with the reconstructed map. In order to generate plans such that localization does not fail during trajectory execution, the planner should attempt to minimize the localization uncertainty.

In other words, this is equivalent to trying to maximize the belief of the vehicle in a partially known 3D world, and hence we utilize the sampling-based approach known

Algorithm 1 Multi-MAV planning framework

procedure COMPUTENBV(x, X, C)
  set λ
  initialize $m \leftarrow x$, σ, $C \leftarrow I$
  while $\neg stop$ **do**
    for $i \in [1, \lambda]$ **do**
      $x_i \leftarrow \mathcal{N}(m, \sigma^2 C)$
      $f_i \leftarrow computeHeuristics(x_i, X)$
    $x_{1\ldots\lambda} \leftarrow x_{f(1)\ldots f(\lambda)}$
    $C \leftarrow updateCovariance()$
    $\sigma \leftarrow updateStepSize()$
  return x
procedure RRBT(v_{start}, v_{goal}, X)
  init $G.V, G.E$
  $v_{start}.P \leftarrow 0, v_{start}.c \leftarrow 0, v_{start}.\Sigma \leftarrow \Sigma_{init}$
  while $N_{iter} < ITER_{MAX}$ **do**
    $v_{rand} \leftarrow$ SAMPLE()
    $v_{nearest} \leftarrow$ NEAREST($G.V, v_{rand}$)
    $v_{new} \leftarrow$ STEER($v_{nearest}, v_{rand}, \epsilon$)
    $[v, s] \leftarrow computeHeuristics(v_{new}, X)$
    if $v == 0$ **then** goto 16
    $e_{new} \leftarrow$ CONNECT(v_{near}, v_{new})
    if v_{new}.PROPAGATE($v_{near}.J, v_{near}.\Sigma, e_{near}$) **then**
      $v_{new}.P = v_{near}$
      $G.V \leftarrow G.V \cup v_{new}$
      $Q \leftarrow Q \cup v_{new}$
      while $Q \neq \emptyset$ **do**
        $u \leftarrow pop(Q)$
        for $u_n \in$ NEAR(u) **do**
          $J(u) \leftarrow$PROPAGATE($u.J, e_u$)
          if $J(u) < J(u_n)$ **then**
            $G.E \leftarrow G.E \setminus \{u_n.P, u_n\}$
            $u_n.P = u$
            $G.E \leftarrow G.E \cup$ CONNECT(u, u_n)
            $Q \leftarrow Q \cup u_n$
  $v_{neighbor} \leftarrow$ NEAREST(v_{goal})
  $p = v_{neighbor}.P$
  while $p \neq 0$ **do**
    $Path \leftarrow Path \cup p$
    $v_{neighbor} = p$
    $p = v_{neighbor}.P$
  return $Path$

as rapidly exploring random belief trees (RRBT) as our planning algorithm with some modifications in the objective function. The foundation of the RRBT [4] is to combine graph construction and uncertainty estimation. The RRBT operates on a graph containing a set of vertices $G.V$ and edges $G.E$. Along with the state, the vertices contain properties describing the uncertainty, which is contained in a set of belief nodes $v.N$. In our adaptation of the RRBT, each belief node contains an estimated covariance Σ and a path cost c. At every iteration, we sample in the space of position and yaw. Using a steering function, a vertex v_{new} is created from the nearest neighbor through the edge edge e_{new}. Once sampled, the algorithm computes the image projection of the available 3D map from this pose. The sampling of the algorithm is biased by discarding any poses that result in less than minimum number of features being visible.

1) Belief propagation: The RRBT incorporates belief knowledge and updates the belief while traversing across edges using a propagation step similar to a Kalman filter's update step. We recall that the sampling step already ensures a minimum number of visible features, hence vision based measurements will always be possible: the priority now is to evaluate the quality of this possible measurement. For our application, we simulate the measurement as the pose itself, but we scale the measurement noise covariance according to the span and visibility heuristics computed after sampling (as the other heuristics in subsection A relate mostly to multiple vehicles). Using these heuristics to vary the measurement noise covariance indicates to the planner that locations resulting in high feature visibility and proximity to existing feature-rich areas have a low covariance and are hence favorable. Based on these metrics, the planner propagates an estimated value of state covariance to the new node and creates a new belief node at v_{new}. Then, algorithm investigates whether moving from this vertex to any nearby vertices would improve the existing beliefs at that vertex, known as the belief domination check.

2) Belief domination: To help the planner drive the vehicle to regions that can improve the belief, the RRBT algorithm examines whether a belief node n_1 is better than n_2. We note that n_1 and n_2 are beliefs at the same vertex, but resulting from different paths taken. In the original implementation, a belief node n_1 is considered better than n_2 if all the properties of n_1 such as cost, covariance etc. are better than those of n_2. For our approach, we use a modified comparison technique that makes use of a weighted cost function, which can be expressed for the belief node n_j at vertex v_i as:

$$J_{v_i.n_j} = w_c(v_i.n_j.c) + w_\Sigma(Tr(v_i.n_j.\Sigma)) \tag{4}$$

The effect of the estimated covariance on the quality of a node is represented through the trace of the matrix, and the path cost is represented as n_c, which is the combination of Euclidean distances over all edges that connect the start state to the node $v_i.n_j$. The parameters w_c and w_Σ are considered to be tuning parameters. If the value of this objective function at a belief node n_1 is higher than another belief node n_2 at the same vertex, the edge responsible for n_2 is pruned, thus making the path creating n_1 the preferred one. If this happens, the vertex that was modified is added to a queue, and the belief domination check repeats for its neighbors. This queue ensures that the discovery of areas affording good measurements and higher reduction in uncertainty is updated recursively through the entire existing graph, rewiring the whole graph.

The objective function in equation 4 is also responsible for a cost vs uncertainty reduction trade-off. Setting w_Σ to zero would cause the algorithm to perform similar to

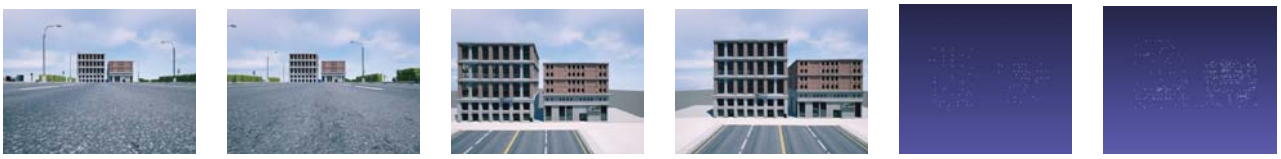

Fig. 3: Change in viewpoints after applying the next-best-view planner to a pair of 2 MAVs. The MAVs start from locations that are at a significant amount of distance from the texture-rich part of the scene (left two images) but the NBV planner commands them to move closer for a better map (middle two). Comparison of point clouds from the initial and final locations shown in the two rightmost images.

an RRT*, choosing paths solely based on minimizing the travel distance. Increasing the value of w_Σ would make the algorithm prefer covariance minimization over path cost.

V. IMPLEMENTATION AND RESULTS

In this section, we present results that validate our planning approach. We use the recently released UAV simulator Microsoft AirSim [17] as our test platform. AirSim is built on top of Unreal Engine, a photorealistic game engine with cutting edge graphics features such as high resolution textures, realistic lighting and shadows - which enables testing of vision based algorithms in close-to-real-life settings. We instantiate several quadrotors within AirSim while AirSim simulates the dynamics, provides low level autopilot features and a front camera stream for each vehicle. Images at 640x480 resolution were recorded at approximately 5 Hz from the MAVs. As our localization algorithm is not implemented in real-time yet, we perform our experiments on pre-recorded image data. For most of these results, we focus on sparsely populated environments and vehicles that are initialized at a significant distance from existing features: generally challenging scenarios for vision based localization.

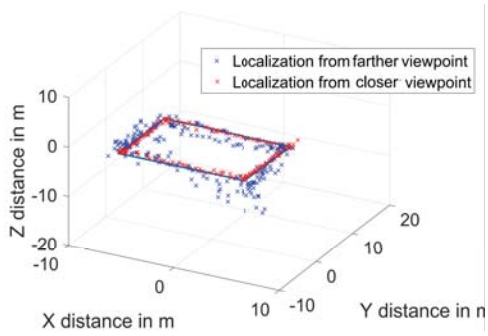

Fig. 4: Performance of localization against two different maps. Localization exhibits significant error, drift with the map that was built from far away (blue markers), whereas it follows ground truth closely with the map from closer viewpoint (red markers). Ground truth shown in solid blue.

A. Next Best View Planning

In order to test the next best view planner, we simulated a sparse environment in Unreal Engine and a pair of MAVs that are initially located 75 m from the scene. From their initial positions, the MAVs perform feature matching to build a sparse map containing 83 3D points. Using this map as the initial knowledge, we ran our CMA-ES based next best view planner in order to build an improved map. The CMA-ES algorithm obtained a new pair of positions, that, in accordance to the metrics in the optimization function, should result in a better reconstruction through overlap and possible observation of newer features. When reconstructed from the new viewpoints, the MAVs created a denser map with 207 points. We show the original and updated images and point clouds in figure 3. To evaluate whether the new map improved the localization, we executed a square-shaped trajectory using one of the MAVs. Images from this trajectory were localized against both maps, and the comparison can be seen in figure 4. The map generated from the closer viewpoint matched the ground truth better with a mean squared error (MSE) of approximately 23 cm, whereas the map from the farther viewpoint exhibited a significant amount of drift and inaccuracy.

As CMA-ES is an evolutionary strategy, it contains a certain degree of randomness which can change the result slightly between runs. In order to test its repeatability, we used the images from the previous trajectory and localized them with maps generated using the viewpoints resulting from 10 different runs of the NBV planner. In table 1, we present a comparison of the maps in terms of localization accuracy. The averaged MSE compared to the ground truth was under 25 cm for all the maps, showing that the algorithm was consistent.

TABLE I: Comparison of localization accuracy with maps from different runs of the NBV planner

Index	1	2	3	4	5
MSE (cm)	17.2	18.8	11.4	14.2	22.2
# pts	217	248	271	200	199
Index	6	7	8	9	10
MSE(cm)	20.9	19.1	24.3	21.2	19.4
# pts	230	232	197	212	257

We then used the NBV planner to improve an environment using five MAVs. After an initial map was built, we marked a point of interest where the MAVs would have to navigate to, in order to let the algorithm refine the map in that region. In this case, the algorithm resulted in a solution with three MAVs assigned to the region of interest, while letting the other two remain close to the starting positions, in order to keep the visibility metric satisfied. The initial and updated maps can be seen in figure 7.

In our third test, we tested the NBV planner in an environment that contained a set of buildings located along a curved road (figure 5). In this experiment, we used two MAVs,

Fig. 5: Results of NBV planner applied to a bigger environment. The original environment is shown in the middle. The view on the left shows the initial map obtained by the vehicles at the start, and on the right, the final map after executing the NBV planner can be seen. The NBV planner was given rough 'points of interest' which were used to find scenes in the proximity; and the partially known representations were used to generate better viewpoints.

Fig. 6: Images captured initially (top) vs captured from the viewpoints generated from the NBV planner (bottom). Results shown for three buildings present near the regions of interest.

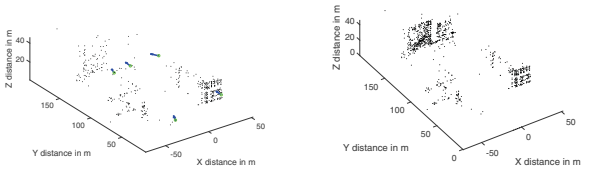

Fig. 7: Result of the NBV planner in an environment with five MAVs. Planner uses an initial representation to generate viewpoint poses (left), which results in a denser map (right).

provided region of interest estimates near each building: areas where the MAVs would have to navigate to. As this would require an improved reconstruction of the environment in those regions, the NBV planner computed poses that, when viewed from, maximize feature discovery and visibility from both MAVs. In figure 6, we show the process of the algorithm with images before and after the 'next-best-view' planning for each region of interest.

B. Perception aware planning for individual vehicles

Next, we tested the RRBT planner with some sample maps by specifying start and goal locations. In figures 8 and 9, we show two examples of how the RRBT planner generated paths and heading angles such that feature visibility was not impaired. Within figure 8, we show two sub-cases that demonstrate the effect of the weights on the cost and covariance reduction parts of the objective function.

By executing the RRBT generated trajectories in AirSim, we also evaluated the actual pose covariance from the localization algorithm; and how it is affected by the RRBT generated plans. As an example, we show the confidence of two paths from the vision based localization algorithm. The

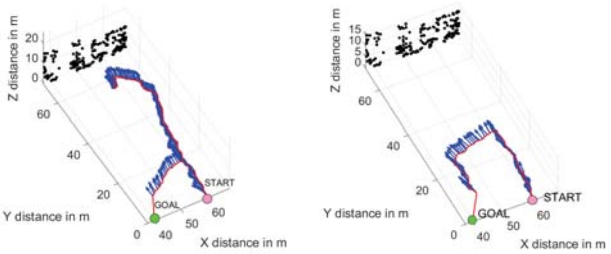

Fig. 8: Path plans generated from the RRBT for a sample map. 3D points from the global map are shown in black, and the plan from start to goal is drawn in red, with blue arrows representing the heading of the vehicle. Figure on the left shows the result of a high w_Σ, which prioritizes the best observations at the expense of longer trajectories. Figure on the right shows a plan generated with a balanced combination of w_c and w_Σ.

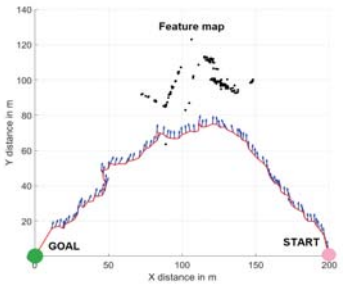

Fig. 9: Another example of the RRBT algorithm driving the MAV close to the feature map (dots in black), before navigating towards the goal. It can be noticed from the heading (blue arrows) that the planner ensures the MAV faces the features at every step.

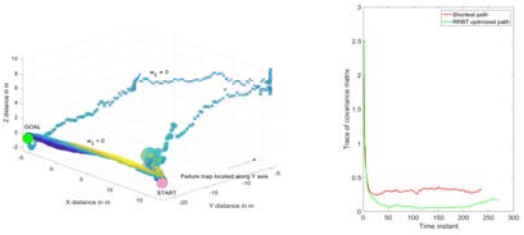

Fig. 10: Comparison of pose covariances for RRBT optimized path vs the shortest path for a sample test. RRBT optimized path results in a lower covariance at the final goal.

first path was generated with w_Σ set to zero, and hence is close to the shortest path with no improvement coming from the belief space. The second path was generated with a higher weight on the covariance parameter, and hence results in a path that moves the MAV closer to the features and seeks to reduce uncertainty. When these paths were executed within AirSim: as expected, we observed that path 1 resulted in a higher covariance throughout and at goal compared to path 2. Comparison of the paths with covariance ellipsoids can be seen in figure 10 (left), and the trace of the covariance matrices during the path execution is shown in figure 10 (right).

VI. DISCUSSION AND CONCLUSIONS

In this paper, we described a path planning framework geared towards groups of micro aerial vehicles that localize collaboratively through vision. Our planning framework contains two stages, both driven by vision heuristics that estimate the quality of images obtained from a given viewpoint. The first stage performs next-best-view planning using an evolutionary strategy (CMA-ES) in order to generate refined and better 3D reconstructions of the surrounding environment. The second stage contains a modified version of a rapidly exploring random belief tree that estimates how the heuristics, and thereby, image quality could affect the system covariance; and thereby generates path plans that ensure feature visibility and subsequently improve vehicle confidence. We demonstrated through experiments in photo-realistic simulation environments; that the next-best-view planner results in better maps for the MAVs, and also that the updated maps directly translate to better localization, thus validating the effectiveness of the heuristics. We also presented sample results from the RRBT based planner to demonstrate how it tries to favor feature-rich environments in order to reduce localization uncertainty.

Our primary aim for the future is to implement both the collaborative localization and planning frameworks in real time. In a real time implementation, we envision the information shared between the vehicles to be the feature locations and descriptors, which could be communicated over Wi-Fi to build maps. The NBV planner would act in a centralized high level fashion, generating waypoints for all the vehicles, whereas the RRBT planner acts on individual vehicles, generating trajectories to move them to NBV waypoints, or from the waypoints to their final goals. We also plan on evaluating the accuracy of the planners with different and more complex environments and how many reconstructions are required for robust navigation: complex or dynamic scenes could require more updates of the map. Furthermore, we plan on testing both the algorithms extensively with various kinds of environments in order to test their robustness as they are not deterministic algorithms; as well as to benchmark performance metrics such as running time. We also plan on integrating collision avoidance with information coming from multiple vehicles using an occupancy grid or similar techniques.

REFERENCES

[1] Sai Vemprala and Srikanth Saripalli. Vision based collaborative localization for multirotor vehicles. In *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, pages 1653–1658. IEEE, 2016.

[2] Sam Prentice and Nicholas Roy. The belief roadmap: Efficient planning in linear pomdps by factoring the covariance. In *Robotics Research*, pages 293–305. Springer, 2010.

[3] Robert Platt Jr, Russ Tedrake, Leslie Kaelbling, and Tomas Lozano-Perez. Belief space planning assuming maximum likelihood observations. 2010.

[4] Adam Bry and Nicholas Roy. Rapidly-exploring random belief trees for motion planning under uncertainty. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 723–730. IEEE, 2011.

[5] Jur Van Den Berg, Sachin Patil, and Ron Alterovitz. Motion planning under uncertainty using iterative local optimization in belief space. *The International Journal of Robotics Research*, 31(11):1263–1278, 2012.

[6] Ali-Akbar Agha-Mohammadi, Suman Chakravorty, and Nancy M Amato. Firm: Sampling-based feedback motion-planning under motion uncertainty and imperfect measurements. *The International Journal of Robotics Research*, 33(2):268–304, 2014.

[7] Vinay Pilania and Kamal Gupta. A localization aware sampling strategy for motion planning under uncertainty. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 6093–6099. IEEE, 2015.

[8] Vadim Indelman. Towards cooperative multi-robot belief space planning in unknown environments. In *Proc. of the Intl. Symp. of Robotics Research (ISRR)*, 2015.

[9] Markus W Achtelik, Simon Lynen, Stephan Weiss, Margarita Chli, and Roland Siegwart. Motion-and uncertainty-aware path planning for micro aerial vehicles. *Journal of Field Robotics*, 31(4):676–698, 2014.

[10] Gabriele Costante, Christian Forster, Jeffrey Delmerico, Paolo Valigi, and Davide Scaramuzza. Perception-aware path planning. *arXiv preprint arXiv:1605.04151*, 2016.

[11] Enrique Dunn and Jan-Michael Frahm. Next best view planning for active model improvement.

[12] Oscar Mendez, SJ Hadfield, Nicolas Pugeault, and Richard Bowden. Next-best stereo: extending next best view optimisation for collaborative sensors. *Proceedings of BMVC 2016*, 2016.

[13] Andreas Bircher, Mina Kamel, Kostas Alexis, Helen Oleynikova, and Roland Siegwart. Receding horizon" next-best-view" planner for 3d exploration. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pages 1462–1468. IEEE, 2016.

[14] Sai Vemprala and Srikanth Saripalli. Vision based collaborative localization for swarms of aerial vehicles. In *Proceedings of the American Helicopter Society 73rd Annual Forum and Technology Display*. AHS, 2017.

[15] Cl Connolly. The determination of next best views. In *Robotics and Automation. Proceedings. 1985 IEEE International Conference on*, volume 2, pages 432–435. IEEE, 1985.

[16] Nikolaus Hansen and Andreas Ostermeier. Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation. In *Evolutionary Computation, 1996., Proceedings of IEEE International Conference on*, pages 312–317. IEEE, 1996.

[17] Shital Shah, Debadeepta Dey, Chris Lovett, and Ashish Kapoor. Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In *Field and Service Robotics*, 2017.