2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
Madrid, Spain, October 1-5, 2018

Motion Planning for a UAV with a Straight or Kinked Tether

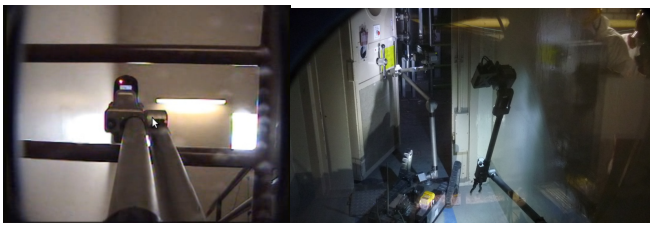
Xuesu Xiao[1], Jan Dufek[1], Mohamed Suhail[2] and Robin Murphy[1]

***Abstract*— This paper develops and compares two motion planning algorithms for a tethered UAV with and without the possibility of the tether contacting the confined and cluttered environment. Tethered aerial vehicles have been studied due to their advantages such as power duration, stability, and safety. However, the disadvantages brought in by the extra tether have not been well investigated by the robotic locomotion community, especially when the tethered agent is locomoting in a non-free space occupied with obstacles. In this work, we propose two motion planning frameworks that (1) reduce the reachable configuration space by taking into account the tether and (2) deliberately plan (and relax) the contact point(s) of the tether with the environment and enable an equivalent reachable configuration space as the non-tethered counterpart would have. Both methods are tested on a physical robot, Fotokite Pro. With our approaches, tethered aerial vehicles could find their applications in confined and cluttered environments with obstacles as opposed to ideal free space, while still maintaining the advantages from the usage of a tether. The motion planning strategies are particularly suitable for marsupial heterogeneous robotic teams, such as visual servoing/assisting for another mobile, tele-operated primary robot.**

I. INTRODUCTION

Tethered Unmanned Aerial Vehicles (UAVs) have been extensively investigated in the literature due to their alternative advantages when being compared to their non-tethered counterparts, such as stability [1], agility [2], and extended flight duration [3]. They also provide extra safety margins and are in accordance with certain authority requirements [4], [5]. Due to such advantages, tethered UAVs have also been paired with Unmanned Ground Vehicles (UGVs) [6] or Unmanned Surface Vehicles (USVs) [7], [8] to increase the primary robot's or the human operator's situational awareness. In particular, a tethered UAV was used to replace tele-operated visual assistant to provide primary robot's operator with better third-person visual feedback and thus improved situational awareness (Fig. 1).

In remote confined and cluttered environments, teleoperation of a robot with first person view from robot's onboard camera is difficult due to the perception limitations, such as the lack of depth perception (Fig. 1a). Using a separate tele-operated robot can partially solve this problem by providing more informative additional view points. For example, Fig. 1b shows two iRobot PackBots being used to conduct radiation surveys and read dials inside the plant facility, where



(a) First Person View of Robot's Onboard Camera

(b) Third Person View from Assistive Robot Helping Primary Robot Open a Door

(c) Heterogeneous Marsupial Robot Team with Autonomous Visual Assistant Using a Tethered UAV

Fig. 1. Teleoperation solution in a remote environment is improved from first person view by onboard camera (a), through a separate manually controlled ground robot for third person view (b), to an autonomous flying aerial visual assistant operating with a tether (c) [6].

the second PackBot provides camera views of the first robot in order to manipulate door handles, valves, and sensors faster. However, this approach introduces other problems including extra 1-2 human operators for the second robot and difficulty in communication between the two operator crews. Therefore, researchers started looking into replacing the second robot and its operating crew with an autonomous agent, especially UAVs due to their superior mobility to cover larger spaces (Fig. 1c) [6]. Moreover, adding a tether to the UAV can provide extra benefits to the visual assisting approach, including matching battery duration with the primary ground robot, reliable localization with light computational overhead, and easy retrieval in the case of malfunction or accident. This requires the autonomous tethered UAV to be able to plan an executable path and navigate through the cluttered environments, while being connected by a tether.

However, planning the motion of a tethered UAV has not been deeply researched, especially in a non-free environment with obstacles. This paper investigates how to generate executable motion plans so that the tethered UAV can navigate through cluttered environments with obstacles. Two motion

[1]Xuesu Xiao, Jan Dufek, and Robin Murphy are with the Department of Computer Science and Engineering, Texas A&M University, College Station, TX 77843 {xiaoxuesu, dufek, robin.r.murphy}@tamu.edu

[2]Mohamed Suhail is with the Department of Visualization, Texas A&M University, College Station, Texas 77843 mohamedsuhail@tamu.edu

planners are proposed to allow collision-free motion of the visual assistant as if it were tetherless, while still utilizing the advantages of being tethered to its primary robot.

This paper is organized as follows: Sec. II reviews related work regarding motion planning for tethered robots. Sec. III presents two motion planning algorithms: (1) reachable space reduction via ray casting and (2) contact point(s) planning and relaxation, along with their motion executor. Sec. IV gives results from exploratory trials using a physical robot. Discussion on results is presented in Sec. V. Sec. VI concludes the paper.

II. RELATED WORK

Tether planning is not deeply investigated by the Unmanned Aerial Vehicle (UAV) community. [3] presented a tethered UAV platform with the purpose of power-over-tether considerations with a specific tether lengthening and retraction method. However, other than power considerations the tethered UAV was not treated differently as a tetherless agent. [2] used a tether to achieve high-speed steady flight on a UAV. Novel tether management system and dynamics have been propsed in [9], [10]. All these researches focused on hardwares and controls, not tether planning.

For Unmanned Ground Vehicles (UGVs) and Unmanned Marine Vehicles (UMVs), tether has been widely used, but there is also very limited reported literature on planning with tethers. Tethers were found in robotic missions since tethers can mitigate the constraints imposed by onboard batteries, provide more reliable communication, enable transportation of gases, fluids, and other materials, and serve as a failsafe to retrieve the robot in case of malfunction [11], [12]. Tether systems were also designed to provide extra support or to rappel Unmanned Ground Vehicles (UGVs) [11], [13]. Most Remotely Operated Vehicles (ROVs) for underwater exploration are also tethered [14]. Both tethered UGVs and UMVs didn't address the tether specifically and assumed that the movement of the tether happens in completely free space and wouldn't cause any problems.

One important advantage brought in by a tether to UAVs is an alternative localization and navigation methodology, which would impact implementation of a tether planner. In lieu of traditional UAV sensory inputs including GPS [15], Inertia Measurement Units (IMUs) [16]–[18], laser range finder [16], [17], stereo [17], [18] and RGB-D cameras [19], the new approach comes with a tether-based UAV controller: tether length encoder, tether azimuth and elevation angle computed by fusing onboard IMU and tether angle sensor [1]. The localization of the UAV could then be realized by measuring the tether length and angles with respect to the global frame. Work in [1] is most relevant to this research, since they dealt with localization and stabilization of the UAV with a taut tether. This laid the ground work for this paper since our low level UAV position and attitude controller is based on a taut tether. However, the UAV in [1] was assumed to be hovering in free space without any obstacles. The tether was taut and straight, not in contact with the environment. This worked well in outdoor environments and free indoor spaces. But in more realistic scenarios it is not reasonable to always assume a free-flight zone, where the tether never touches the environment. Tether contact has been researched in the sense of tether entanglement prevention in multi-robot systems [20]. But researchers have not paid sufficient attention to the interaction of tether with physical environments, especially in 3-D cluttered spaces.

This paper looks into how to generate executable motion plans for a tethered UAV in 3-D cluttered environments, for purposes such as visual assisting of a primary ground robot. Two approaches are proposed: (1) using a ray-casting method to reduce the reachable space of the UAV, assuming a straight and non-contact tether and (2) integrating tether contact points into the planning space so the UAV could work with a kinked tether.

III. APPROACH

Two motion planners and a motion executor are discussed in this section. It is assumed that a map is precomputed in the form of 3-D occupancy grid. The static and complete map is occupied with obstacles. The goal is to generate an executable path for a tethered UAV from point *start* to point *goal* in the free space. The path is then interpreted and translated to tether-based motion commands. Reachable space reduction via ray casting assumes the tether does not contact the obstacles during the whole flight. Contact point(s) planning and relaxation allows the tether to touch the obstacles and form a kink, and therefore the UAV can reach any free space as if it were tetherless. After the offline motion planners generate waypoints with contacts in the given 3-D map, the online motion executor issues real-time motion commands based on the motion plan and tether localization feedback.

A. Reachable Space Reduction via Ray Casting

In contrast to conventional UAVs, tethered UAVs have to maintain connection with its ground station via a tether. If the tether needs to remain taut and straight, no contacts are allowed with the environments. This constraint reduces the reachable space. Obstacles cannot locate between the UAV and its tether reel, since otherwise the tether in between would touch the obstacles. Based on this idea, this planner uses a ray casting approach from tether reel to obstacles in order to identify spaces in the configuration space, which are feasible for the UAV alone, but not with a tether. For voxels on the ray, those between reel center and obstacles are still open, while those beyond obstacles are blocked.

After reachable space reduction, remaining space is completely free even with respect to the tether. Any path planning algorithm which works in 3-D could be applied between any two points in the reduced space. Here, Probabilistic Road Map (PRM) [21] is used to plan an executable path in the reduced space.

Ray casting has a complexity proportional to the number of obstacles $\mathcal{O}(o)$. Assuming n is the number of vertices in the road map, the complexity of PRM is $\mathcal{O}(n^2)$, including populating free space, running local planner, and final

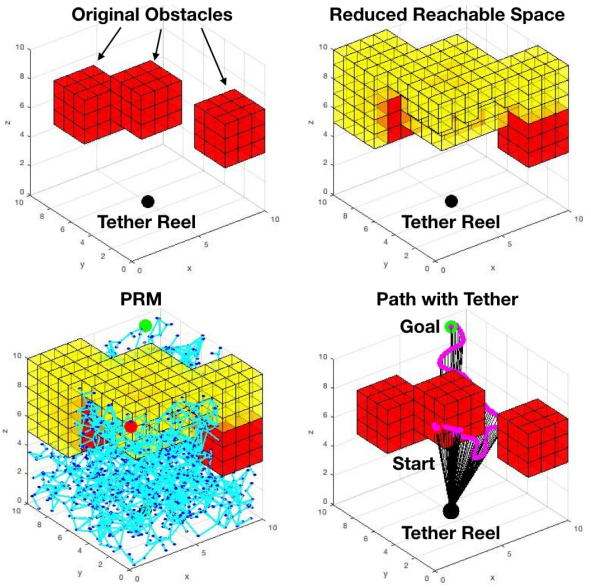


Fig. 2. Reachable space is reduced by ray casting from tether reel to original obstacles (in red). Yellow voxels are non-reachable space due to tether. UAV path is planned using PRM (cyan) in the reduced reachable space. The UAV has to go beneath the obstacles since a direct straight path above all obstacles is not allowed by the existence of tether.

search to find a path. The complete algorithm pipeline with complexity $\mathcal{O}(o + n^2)$ is illustrated in Fig. 2.

B. Contact Point(s) Planning and Relaxation

As we can see in Fig. 2, the free space is largely reduced by ray casting, leaving only a subset of the original free space reachable with a tether. A different algorithm with tether contact point(s) planning is presented here. It automatically plans the tether contact point(s) when the robot locates in the originally free but actually occupied spaces after reduction (yellow voxels in Fig. 2). It also has the capability to relax the contact point(s) when the UAV returns to the post-reduction free space. It assumes that once a contact point is formed, it doesn't move unless being relaxed. The algorithm is outlined in Algorithm 1.

The algorithm starts with inflating the original *map* with the radius of the robot, generating PRM in free spaces, and query a smooth path from *start* to *goal*. We use a stack (*CP_stack*) to keep track of all current active tether contact points with the environment. Line 5 initializes contact points of all waypoints along the path to be the original tether reel center (*tether_origin*). *relax_flag* indicates if contact point relaxation is necessary. The rest of the algorithm plans the contact point for each individual waypoint. Line 9 to 21 determines if it is necessary to relax the current contact point. It first checks whether the robot is located at a waypoint directly reachable from the last contact point (line 11). If true (*collision_flag* == 0), it is possibly necessary to relax the current contact point (*CP*), depending on if obstacle is confined within the triangle formed by the waypoint, last and current contact points. If false (*collision_flag* == 1), relaxation is not necessary. The actual relaxation is

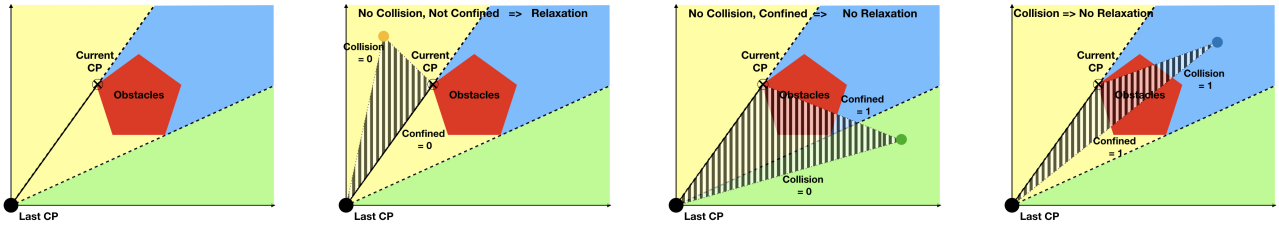
Algorithm 1 Contact Point(s) Planning and Relaxation

Input: *map*, *start*, *goal*, *tether_origin*
Output: executable path: waypoints with contact points

```
1:  Inflate original map
2:  Generate PRM in free space
3:  Query and smooth path from start to goal
4:  Initialize CP_stack with tether_origin
5:  Attach tether_origin to all waypoints WPs on path
6:  relax_flag = 0
7:  for every WP on path do
8:      curent_contact = CP_stack top CP
9:      if (CP_stack has more than one CPs) then
10:         last_contact = second CP from CP_stack top
11:         collision_flag = CheckCollision (last_contact, WP,
            map)
12:         if collision_flag == 0 then
13:             if ObstacleConfined (curent_contact, last_contact,
                WP, map) then
14:                 relax_flag = 0
15:             else
16:                 relax_flag = 1 // contact relaxation
17:             end if
18:         else if collision_flag == 1 then
19:             relax_flag = 0
20:         end if
21:     end if
22:     if relax_flag == 1 then
23:         pop CP_stack
24:         attach new CP_stack top to all following WPs
25:         relax_flag = 0
26:     else if relax_flag = 0 then
27:         if CheckCollision (current_contact, WP, map) then
28:             push new CP to CP_stack // contact planning
29:             attach new CP_stack top to all following WPs
30:         end if
31:     end if
32: end for
33: return all WPs along with their CPs
```

implemented in line 22 to 25. The current *CP* is popped from *CP_stack*, and the last contact point is assigned to all subsequent waypoints. If relaxation is not necessary, line 27 checks if it's necessary to form a new contact point. If yes, the new *CP* is pushed into *CP_stack* and all following waypoints are assigned the new contact point.

CheckCollision (point A, point B, map) draws a line between *point A* and *Point B* and see if any points on the line intersect with any obstacles in *map*. If there is no collision, *ObstacleConfined (point A, point B, point C, map)* further checks if any obstacle in *map* is confined in the triangle formed by *point A*, *point B*, and *point C*. A 2-D illustration of the tether relaxation pipeline is shown in Fig. 3. The 3-D version works on the projection onto x-y, y-z, and x-z planes. To be 3-dimensionally confined, obstacle needs to be 2-dimensionally confined in all three projection planes.



(a) Original Configuration Space with Current and Last Contact Points

(b) Current Contact Point Relaxed due to No Collision and Obstacles Not Being Confined

(c) Current Contact Point Not Relaxed due to No Collision and Obstacles Being Confined

(d) Current Contact Point Not Relaxed due to Collision

Fig. 3. 2-D Representation of the Tether Relaxation Scheme: based on *CheckCollision* between last contact point and current waypoint with the map, *ObstacleConfined* checks if any obstacles are confined within the triangle formed by waypoint, last and current contact points.

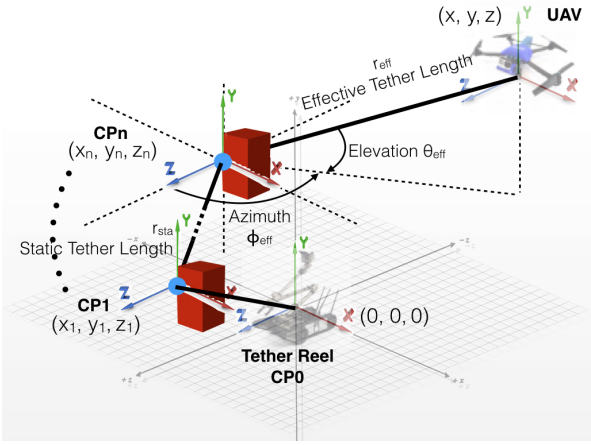


Fig. 4. Motion Executor Interpretation: Tether contact points are saved in a stack, where the latest contact point locates at the top. Tether is divided into several straight line segments, whose lengths are saved and associated with each contact point. Positional control is based on the planned relative coordinates of the UAV with respect to the last contact point.

Both *CheckCollision* and *ObstacleConfined* have a complexity proportional to the number of obstacles $\mathcal{O}(o)$. The complexity for PRM is still $\mathcal{O}(n^2)$. Assuming the executable path consists of p waypoints, the whole algorithm's complexity is $\mathcal{O}(n^2 + po)$.

The result of the algorithm is an executable path composed of 3-D waypoints along with corresponding 3-D contact points. If the tether is not touching the environment, contact point is treated as the tether reel center. So the motion planner outputs a 6-D waypoints and contact points file.

C. Motion Executor

The offline 6-dimensional motion plan is parsed by the online motion executor. The UAV is commanded to reach every single waypoint along the path. In this work, we treated the UAV as a mass point and thus only consider positional movement. The vehicle position control uses *tether length* r, *elevation* θ (vertical angle of tether), and *azimuth* ϕ (horizontal angle of tether). The position of the vehicle could be represented in polar coordinate system (Fig. 4). Given a certain x, y, and z, r, θ, and ϕ could be easily derived:

$$\begin{cases} r = & \sqrt{x^2+y^2+z^2} \\ \theta = & arcsin(\frac{y}{\sqrt{x^2+y^2+z^2}}) \\ \phi = & atan2(\frac{x}{z}) \end{cases} \tag{1}$$

However, since in this work multiple contact points are allowed, all three control parameters, r, θ, and ϕ, are not relative to the tether reel (origin), but to the last contact point CP_n. Here, we use a stack to store all contact points. Whenever the motion executor reaches a new contact point, it pushes it into the stack. It also saves the current static tether length (r_{sta}) from the reel to this contact point. It is termed as static since this portion of the tether remains static based on our assumption that formed contact points don't move unless being relaxed. Whenever a contact point is relaxed, the motion executor pops it from the stack and reduces the static tether length by the corresponding segment length. So we have:

$$r_{sta} = \sum_{0}^{n-1} \sqrt{(x_{i+1}-x_i)^2 + (y_{i+1}-y_i)^2 + (z_{i+1}-z_i)^2} \tag{2}$$

Since we have all our controls with respect to the last formed contact point (top of stack), we have effective values relative to this point:

$$\begin{cases} r_{eff} = & \sqrt{(x-x_n)^2 + (y-y_n)^2 + (z-z_n)^2} \\ \theta_{eff} = & arcsin(\frac{y-y_n}{\sqrt{(x-x_n)^2+(y-y_n)^2+(z-z_n)^2}}) \\ \phi_{eff} = & atan2(\frac{x-x_n}{z-z_n}) \end{cases} \tag{3}$$

So the desired controls are:

$$\begin{cases} r = & r_{eff} + r_{sta} \\ \theta = & \theta_{eff} \\ \phi = & \phi_{eff} \end{cases} \tag{4}$$

The desired values of r, θ, and ϕ are regulated by a PID controller based on the sensory feedback from the UAV (tether angle sensors and reel encoder). An acceptance radius R_{acc} is defined so that whenever the UAV reaches a ball with radius R_{acc} around the desired waypoint, this waypoint is treated as reached and the executor moves on to the next waypoint.

The motion executor doesn't need to discriminate between two different motion planners. The waypoint file from the ray casting approach could also be 6-dimensional, with all contact points to be the tether reel, namely the origin of the global coordinate system. This also applies to the non-contact path segment(s) from the contact planning approach.

IV. Exploratory Trials

The purpose of the exploratory trials is proof of concept of our two motion planning algorithms: reachable space reduction by ray casting and contact point(s) planning and relaxation. By running experiments on physical robots, we wanted to show that our motion planners can navigate the UAV between two points in the corresponding free space of each planner. The trial completion was determined based on the UAV's onboard localization. By running our two motion planning algorithms, we also wanted to demonstrate different reachability sets achievable by the two planners. It was computed as a percentage of reachable spaces in the whole map by offline computation based on the obstacles in the map. Finally, navigation accuracy in terms of cross track error was presented by comparison between planned paths and executed paths. The latter was captured by a ground truth motion capture (MoCap) system.

Our exploratory trials were conducted in a motion capture studio to capture motion ground truth. The studio is equipped with 12 OptiTrack Flex 13 cameras running at 120 Hz. The 1280×1024 high resolution cameras with a 56° Field of View provide less than 0.3mm positional error and cover the whole 3.3×3.3×2.97m space. The high number of cameras guarantee that the UAV could be captured even if the markers were blocked by the obstacles from some cameras. We used obstacles made of cardboard, which formed a 0.33×0.33×0.297m vertical shaft and located in the middle of the experimental environment. The choice of cardboard was to guarantee safe tether contact. This configuration of obstacles blocked most direct passages between different regions in the map, and was particularly difficult for a tethered UAV to navigate through. Fotokite Pro was used as our tethered UAV. The online motion executor executed the offline motion plan from the two algorithms. During the physical tests, the acceptance radius R_{acc} was set to 0.4m. This is the best localization accuracy achievable by Fotokite's sensory feedback measured by experiments. Fig. 5 shows the tethered UAV flying in the MoCap studio.

In order to validate our motion planning algorithms, we conducted three sets of experiments on the tethered UAV:

- Moving in free space after reachable space reduction using ray casting (Fig. 6a)
- Returning to free space by relaxing previously formed contact point (Fig. 6b)
- Entering non-reachable space with a straight tether by planning two contact points (Fig. 6c)

Since the two different motion planners are dealing with different configuration spaces, i.e. reduced and original reachable spaces, we cannot replicate the same navigation task (same *start* and *goal*) for both of them. For the first set of experiments, we manually chose pairs of *start* and *goal* in the reduced reachable space. The obvious direct paths between the pairs were not executable due to the tether. The ray casting motion planner needed to come up with an alternative path to circumvent the obstacles to remain a straight tether. For the second set of experiments, we manually chose a tuple of *(start, middle point, goal)* in the original free space. The *middle point* located at a position where one contact point was necessary to reach. The *Goal* located at a position where no contact was necessary. So the robot had to form and then relax the contact point to reach the final target during the flight. For the third set of experiments, we manually chose a tuple of *(start, middle point, goal)*. The *middle point* located at a position where one contact point was necessary to reach. The *Goal* located at a position where two contacts were necessary. So the robot had to form two contact points in a row to reach the final target during the flight.

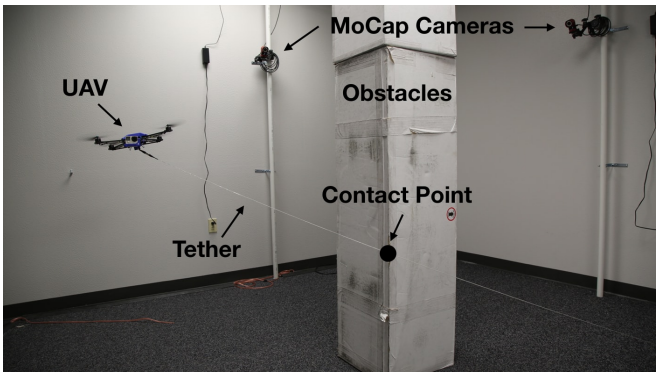


Fig. 5. UAV flying with one tether contact point in the MoCap studio

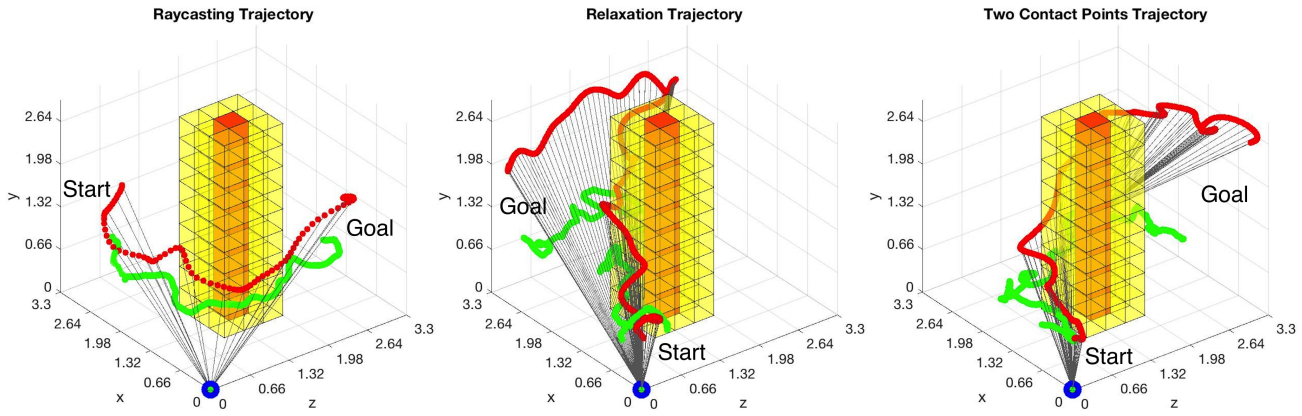
Based on the given map, we obtained two different reachable spaces from the two motion planners. The ray casting method reduced navigable space from the original free space while contact(s) point planning and relaxation kept the whole free space intact.

We totally performed 21 trials. Two trials were discarded due to the UAV platform hardware failure and one was discarded due to the UAV flying out of the range of the MoCap system. We obtained 18 planned paths with way points and contact points (CPs for ray casting were simply tether reel) and corresponding 18 executed paths captured at 120Hz, six trials for each set.

V. DISCUSSION

A. Comparison of the Two Algorithms

Ray casting works in post-reduction free spaces and the UAV cannot reach spaces blocked by ray casting. Contact point(s) planning can navigate to spaces which are not reachable with a straight tether. Tether can be properly relaxed when UAV returns to original free spaces. Multiple contact points could be formed and handled. For this particular set up, ray casting can reach 60% of the whole free space, and contact point(s) planning can reach 100%. A 40% reduction of reachable space was observed for ray casting to maintain a straight tether. Contact point planning has greater



(a) Reduced Reachable Space by Ray casting (b) Contact Point with Relaxation (c) Two Contact Points

Fig. 6. Three Different Paths Planned (Red) and Executed (Green): Red voxels represent the obstacles and yellow voxels are the occupied spaces due to map inflation. Red path is the off-line computed motion plan and green one is the actual path captured by OptiTrack motion capture system.

reachability since the UAV is de facto tetherless, but tether contact may not be acceptable in all domains. There is an open issue as to whether the tether would break or would damage the environment. Ray casting has a complexity of $\mathcal{O}(o+n^2)$. Contact point(s) planning has $\mathcal{O}(n^2+po)$ due to the extra work load to plan and relax contact points.

B. Insights on Implementation

All 18 trials were completed based on the UAV sensor feedback. However, the onboard localization error accumulates during flight, so position estimation is not precise. Fig. 6 shows three example trials. Fig. 6a shows the execution of the path generated by ray casting method. Fig. 6b and Fig. 6c use contact point(s) planning and relaxation. To be noticed is that the tether can pass through yellow voxels (inflation) and contact points can only be formed on the surfaces/edges of the red voxels (obstacles). In Fig. 6a, originally free spaces behind the obstacles are blocked by ray casting, so the robot has to forgo the short path behind the obstacles and circumvent from the front in order to maintain a straight tether through the whole flight. Fig. 6b shows the robot firstly navigates to the far end of the map, where the tether has to touch the obstacles. One contact point is planned, which is thereafter relaxed, since the robot flies back to the non-contact space and reaches the final destination with a straight tether. As we can see, the navigation accuracy decreases significantly after making the contact. In Fig 6c, two contact points are planned along the way. Although the last portion of the path is reachable directly from the tether reel, it still keeps the two contact points since obstacles are confined within the triangle formed by the waypoint, current and last contact points (Fig. 3c).

One important reason behind the navigation error is UAV internal localization error, which is determined by the choice of tethered UAV. The error is further deteriorated when the UAV is far away from the tether reel. Since the increased weight of the tether will pull the tether down due to gravity, the tether will form an arc instead of a straight line (Fig. 7). This explains why the height of the actual flight tends to be

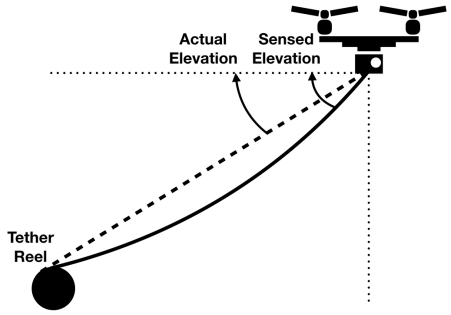


Fig. 7. Tethered UAV's Internal Localization Error

lower than the motion plan, since height is most relevant to the elevation angle. [22] investigates this effect in detail.

An examination on all 18 trials (Tab. I) of the accuracy in terms of cross track error shows that contact point(s) planning has a larger error, which is even more significant with two contact points. We presume that this is due to the error introduced by contact point position, which will accumulate with increased number of contacts made. We further investigate this presumption by looking into the segmented accuracy for different contact points (Fig. 8). When no contact is made, the accuracy (0.4198m) is comparable to the ray casting result in Tab. I. The average error increases to 1.3602m at one contact and 1.7634 at two. The increased positional error is because of two reasons: (1) Due to the lack of contact point positional feedback, the actual contact point may differ from the original motion plan at initial touch. This will shift the navigation space in the next region. (2) The assumption of fixed contact point may not hold all the time, so the contact point will move slightly during flight. This process adds random noise into the system. These two sources of error will accumulate and further deteriorate the navigation accuracy. The three stages of error profile in Fig. 8 clearly indicate the impact of increased number of contact points: the navigational precision is less satisfactory when more contact points are formed.

TABLE I

MEAN CROSS TRACK ERRORS (METER)

	Raycasting	1 contact w/ relaxation	2 contacts w/o relaxation
1	0.6963	1.0005	0.9900
2	0.5644	0.9587	0.9933
3	0.5355	0.8407	1.1895
4	0.4105	0.9940	1.1173
5	0.6026	1.0146	1.1539
6	0.5298	0.9653	1.1212
Mean	0.5565	0.9623	1.0942

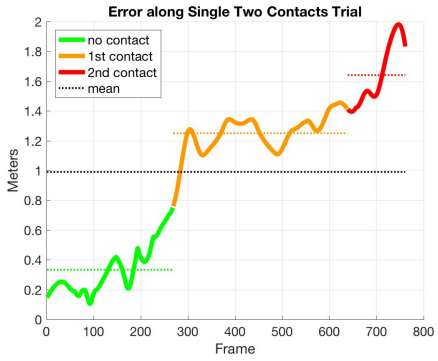


Fig. 8. Navigational Error along an Example Trial with Two Contact Points

VI. CONCLUSIONS

This paper presents and compares two motion planning methods to navigate a tethered UAV in confined spaces with obstacles. The reachable space reduction approach by ray casting provides the best navigational accuracy, but with the price of a smaller reachable space. Contact point planning allows the robot to navigate in all original free spaces as if it were tetherless. It also enables contact relaxation when necessary. However, this approach compromises motion accuracy with increased number of contact points. Two reasons were presented and errors were analyzed. The results indicate that the motion planners and executor provide an alternative way for UAV localization and navigation in indoor cluttered environments using a taut tether. They also alleviate the challenges caused by managing a tether in obstacle-occupied spaces. However, a trade-off between reachable volume and navigational accuracy exists, so full coverage of the free configuration space and high motion precision cannot be achieved at the same time.

ACKNOWLEDGMENT

This work is supported by NSF 1637955, NRI: A Collaborative Visual Assistant for Robot Operations in Unstructured or Confined Environments.

REFERENCES

[1] S. Lupashin and R. D'Andrea, "Stabilization of a flying vehicle on a taut tether using inertial sensing," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. IEEE, 2013, pp. 2432–2438.

[2] M. Schulz, F. Augugliaro, R. Ritz, and R. D'Andrea, "High-speed, steady flight with a quadrocopter in a confined environment using a tether," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*. IEEE, 2015, pp. 1279–1284.

[3] L. Zikou, C. Papachristos, and A. Tzes, "The power-over-tether system for powering small uavs: tethering-line tension control synthesis," in *Control and Automation (MED), 2015 23th Mediterranean Conference on*. IEEE, 2015, pp. 681–687.

[4] K. S. Pratt, R. R. Murphy, J. L. Burke, J. Craighead, C. Griffin, and S. Stover, "Use of tethered small unmanned aerial system at berkman plaza ii collapse," in *Safety, Security and Rescue Robotics, 2008. SSRR 2008. IEEE International Workshop on*. IEEE, 2008, pp. 134–139.

[5] R. Murphy, J. Dufek, T. Sarmiento, G. Wilde, X. Xiao, J. Braun, L. Mullen, R. Smith, S. Allred, J. Adams *et al.*, "Two case studies and gaps analysis of flood assessment for emergency management with small unmanned aerial systems," in *Safety, Security, and Rescue Robotics (SSRR), 2016 IEEE International Symposium on*. IEEE, 2016, pp. 54–61.

[6] X. Xiao, J. Dufek, and R. Murphy, "Visual servoing for teleoperation using a tethered uav," in *Safety, Security and Rescue Robotics (SSRR), 2017 IEEE International Symposium on*. IEEE, 2017.

[7] J. Dufek, X. Xiao, and R. Murphy, "Visual pose stabilization of tethered small unmanned aerial system to assist drowning victim recovery," in *Safety, Security and Rescue Robotics (SSRR), 2017 IEEE International Symposium on*. IEEE, 2017, pp. 116–122.

[8] X. Xiao, J. Dufek, T. Woodbury, and R. Murphy, "Uav assisted usv visual navigation for marine mass casualty incident response," in *Intelligent Robots and Systems (IROS)*, 2017.

[9] M. A. Minor, C. R. Hirschi, and R. O. Ambrose, "An automated tether management system for microgravity extravehicular activities," in *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, vol. 3. IEEE, 2002, pp. 2289–2295.

[10] S. Prabhakar and B. Buckham, "Dynamics modeling and control of a variable length remotely operated vehicle tether," in *OCEANS, 2005. Proceedings of MTS/IEEE*. IEEE, 2005, pp. 1255–1262.

[11] M. Krishna, J. Bares, and E. Mutschler, "Tethering system design for dante ii," in *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on*, vol. 2. IEEE, 1997, pp. 1100–1105.

[12] C. Marques, J. Cristóvao, P. Alvito, P. Lima, J. Frazao, I. Ribeiro, and R. Ventura, "A search and rescue robot with tele-operated tether docking system," *Industrial Robot: An International Journal*, vol. 34, no. 4, pp. 332–338, 2007.

[13] H. Schempf, "Self-rappelling robot system for inspection and reconnaissance in search and rescue applications," *Advanced Robotics*, vol. 23, no. 9, pp. 1025–1056, 2009.

[14] J. Yuh, "Design and control of autonomous underwater robots: A survey," *Autonomous Robots*, vol. 8, no. 1, pp. 7–24, 2000.

[15] K.-H. Oh and H.-S. Ahn, "Extended kalman filter with multi-frequency reference data for quadrotor navigation," in *Control, Automation and Systems (ICCAS), 2015 15th International Conference on*. IEEE, 2015, pp. 201–206.

[16] S. Grzonka, G. Grisetti, and W. Burgard, "A fully autonomous indoor quadrotor," *IEEE Transactions on Robotics*, vol. 28, no. 1, pp. 90–100, 2012.

[17] M. Achtelik, A. Bachrach, R. He, S. Prentice, and N. Roy, "Autonomous navigation and exploration of a quadrotor helicopter in gps-denied indoor environments," in *First Symposium on Indoor Flight*, no. 2009. Citeseer, 2009.

[18] L. R. G. Carrillo, A. E. D. López, R. Lozano, and C. Pégard, "Combining stereo vision and inertial navigation system for a quadrotor uav," *Journal of Intelligent & Robotic Systems*, vol. 65, no. 1-4, pp. 373–387, 2012.

[19] A. R. Vetrella, A. Savvaris, G. Fasano, and D. Accardo, "Rgb-d camera-based quadrotor navigation in gps-denied and low light environments using known 3d markers," in *Unmanned Aircraft Systems (ICUAS), 2015 International Conference on*. IEEE, 2015, pp. 185–192.

[20] S. Hert and V. Lumelsky, "Motion planning in r-3 for multiple tethered robots (vol 15, pg 623, 1999)," *IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION*, vol. 15, no. 6, pp. 1146–1146, 1999.

[21] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.

[22] X. Xiao, Y. Fan, J. Dufek, and R. Murphy, "Indoor uav localization using a tether," in *Safety, Security and Rescue Robotics (SSRR), 2018 IEEE International Symposium on*. IEEE, 2018.