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

Feature-constrained Active Visual SLAM for Mobile Robot Navigation

Xinke Deng, Zixu Zhang, Avishai Sintov, Jing Huang, and Timothy Bretl
Coordinated Science Lab, University of Illinois at Urbana-Champaign
{xdeng12, zzhng122, asintov, jhuang81, tbretl}@illinois.edu

Abstract— **This paper focuses on tracking failure avoidance during vision-based navigation to a desired goal in unknown environments. While using feature-based Visual Simultaneous Localization and Mapping (VSLAM), continuous identification and association of map points are required during motion. Thus, we discuss a motion planning framework that takes into account sensory constraints for a reliable navigation. We use information available in the SLAM and propose a data-driven approach to predict the number of map points associated in a given pose. Then, a distance-optimal path planner utilizes the model to constrain paths such that the number of associated map points in each pose is above a threshold. We also include an online mapping of the environment for collision avoidance. Overall, we propose an iterative motion planning framework that enables real-time replanning after the acquisition of more information. Experiments in two environments demonstrate the performance of the proposed framework.**

I. Introduction

Navigation of mobile agents in unknown environments is an important capability for autonomous robotic systems. Exploration [1], area coverage [2] and search and rescue, are some of the applications for autonomous navigation in which the agent must act efficiently to traverse the environment. The robot must map and avoid collisions, and localize itself while approaching a desired goal. Normally, sampling-based motion planning algorithms are easily utilized for known environments [3]. However, in uncertain and unknown environments the planner uses limited sensory feedback to enforce constraints. Vision offers a low-cost source of rich information of the environment.

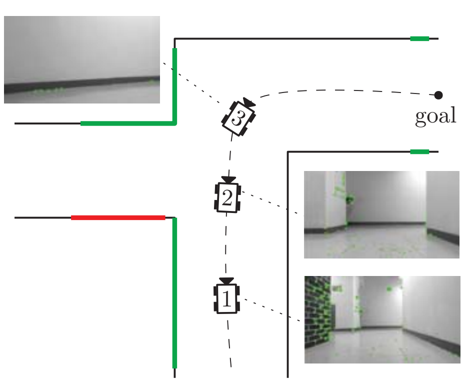


Fig. 1. A mobile robot moving along a planned path around a corner. Green lines are feature-rich areas seen by the robot while moving along the path. The red line is a hidden feature-rich area which is not seen while traversing the path. Tracking failure occurs in the third pose.

Visual Simultaneous Localization and Mapping (VSLAM) [4] deals with using sequential images to estimate the pose of one or more cameras and construct the map of the environment simultaneously in real time. Due to the low cost of the cameras and rich information from the images, VSLAM has attracted significant research attention in the last two decades. However, VSLAM still remains an open problem due to robustness issues [4]. In this paper, we identify one specific type of failure termed *tracking failure* [5]. Tracking failure refers to the incapability of associating features in the current image to map points in VSLAM. In other words, VSLAM cannot observe enough map points in order to localize itself with respect to the map currently being constructed. Tracking failure of VSLAM systems can result in inaccurate pose estimation and catastrophic results, such as the crash of aerial robots. Figure 1 illustrates a tracking failure example. Here, the robot moves on a pre-planned collision-free path while trying to make a turn at the corner. The tracking is fine during the initial motion (poses 1 and 2) due to enough visible features along the path. However, when the robot makes an attempt to turn (pose 3), tracking failure occurs because the topmost wall is poor of features.

In modern feature-based VSLAM systems, such as ORB-SLAM [5], a threshold is commonly used to determine the tracking failure. When the number of associated map points is lower than the threshold, the tracking is declared as failed. Figure 2 shows the relationship between the number of associated map points and the probability of observing tracking failure within the next five images. The data was acquired by running ORB-SLAM on two challenging sequences in EuRoC MAV [6] datasets (Vicon Room 1 03 and Room 2 03). Each sequence is evaluated for 50 times with different thresholds (3, 10, 20 and 30). It is obvious that more associated map points correlate to lower probability of tracking failure. Although lowering the threshold for a lower number of associated map points (between 30 and 50) will reduce the probability of failure, the gained benefit is marginal. In other words, merely varying the threshold offers a limited benefit. Increasing the number of associated map points, however, is significantly more beneficial. Thus, this motivates the necessity of actively adjusting the camera pose to reduce the probability of tracking failure and having more associated map points.

In this paper, our goal is to actively plan a path of a mobile robot to a goal while avoiding tracking failure. There are several challenges related to this task. The first challenge is efficiently approximating the map points that can be associated from a specific pose. We address this challenge by proposing a data-driven Map points Association

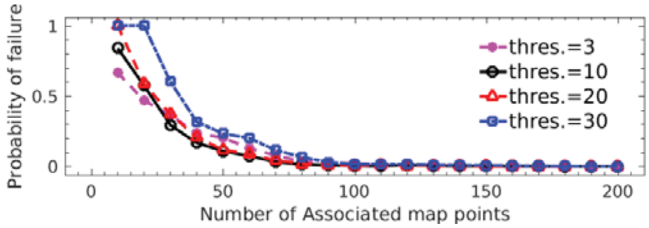


Fig. 2. The relationship between the number of observed map points and probability of tracking failure with monocular ORB SLAM on EuRoC dataset Vicon Room 1 03 and Vicon Room 2 03 sequences.

Model (MAM). The second challenge is related to real-time path planning under the VSLAM constraint. In particular, we must constrain the path based on the MAM such that it is feature-rich. We use a distance-optimal RRT* [7] planner and constrain the path to be collision-free and keep the number of tracked map points above a threshold. However, the environment map and the SLAM map are not a priori known and need to be constructed incrementally on the fly. Thus, replanning must be performed after the gathering of more information. In the example of Figure 1, the robot can explore its immediate surroundings from the last feasible pose of the contemporary path (for instance, just before pose 3). Thus, new map points would be initialized. This new information will provide the planner with more opportunities to replan a new feasible path toward the goal. If exploration is avoided, the robot will continue replanning but fail to find a feasible path due to limited information. The notion of frontier proximity exploration for uncertainties reduction will be discussed in this paper.

The main contributions of this paper are the identification of the particular tracking failure and the proposition of a navigation framework to avoid such failure. We propose to exploit internal data of the SLAM to approximate the number of associated map points and use it to constrain the path in a real-time distance-optimal planner. The implementation of the framework in ROS is open-sourced and available online.

II. Related Work

Active perception has roots in the seminal work of Bajcsy [8]. In the paper, the author points out that sensory performance can be improved by proper selection of control. Since then, efforts are made to integrate perception, path planning and control, such as in [9] and [10]. In [9], the authors propose a planning framework for active localization, whereas [10] focuses on information acquisition with multiple robots. However, neither of them focus on visual navigation. The integration of visual sensing with planning and control is commonly referred to as active vision or a Next-Best-View (NBV) problem. Prominent work in active vision by Davison and Murray [11] try to control two movable cameras on a ground robot to reduce uncertainty during localization. Map points were observed in the direction perpendicular to the uncertainty. Planning in the belief space such as in [12] and [13] have been applied to active vision as well. In [12], the authors exploit Rapidly-exploring Random Brief Trees [14] to plan a path which minimizes the localization uncertainty and considers the predicted measurements along the path with sensor models. However the approach assumes a known map, and the path planning is performed offline. In [13], the authors consider photometric information and perform online path-planning. However, the map is still assumed to be known.

Significant progress has been made in VSLAM over the last two decades. There are some successful open source VSLAM systems [5], [15], and VSLAM has been practically applied in fields such as autonomous robot navigation and, virtual and augmented reality [15], [16]. One major breakthrough in VSLAM is the PTAM by Klein and Murray [15], which separates the VSLAM problem into two threads: tracking and mapping to ensure accuracy and real-time performance. However, most of the VSLAM systems are performed passively. Active VSLAM deals with the problem of motion planning concurrently with VSLAM.

Bryson and Sukkarieh select control actions based on overall map quality in [17]. The map quality is indicated by using mutual information. In [18], the authors propose a framework called Perception-Driven Navigation (PDN), which enables the robot to revisit the previous explored visual salient area to reduce the uncertainty of SLAM. The authors use the covariance matrix of the estimated pose to indicate the uncertainty. In [19], the authors propose a path planning method to make VSLAM focus on feature-rich areas. In order to compute the density of features, a 3D mesh grid is built upon the sparse point cloud provided by PTAM [15]. An RRT* [7] based path planning algorithm is developed to minimize the cost function which considers the distance to the goal position and the feature density. In [20], the authors introduce a metric to evaluate the localization stability based on the observations, viewing angles, scale, and depth of map points. A path planning module is developed to ensure that the localization quality is high.

The failure of VSLAM systems and its avoidance by constraining the camera motion are not discussed explicitly in [18] and [19]. In addition, a mesh grid is built in [19] to inform the density of map points, which can be expensive. The work in [20] is probably the most relevant to our work due to its focus on localization stability. However, the reasons for VSLAM failure are not discussed. Lacking of intuition, the proposed metric imposes difficulties in application. In addition, the experiments for real-world navigation are limited to a two-meter cubic space. In our work, we show a navigation framework that enables a ground robot to traverse through a much larger VSLAM challenging environment like the one shown in Figure 1.

III. Problem Formulation

We consider a mobile robot operating in an indoor 2D enviroment with static obstacles. Let $\mathcal{C} \subset \mathbb{R}^2 \times \mathbb{S}$ be the configuration space of the robot in the plane. In addition, let $\mathcal{C}_{obs} \subset \mathcal{C}$ be a restricted region due to obstacles. The free configuration space is defined by $\mathcal{C}_{free} = \mathcal{C} \setminus \mathcal{C}_{obs}$. We note that any a-priori knowledge of $\mathcal{C}_{obs}$ is not available. We also assume that $\mathcal{C}_{free}$ is a path-connected subspace of $\mathcal{C}$. The robot is equipped only with a Kinect-style camera able to provide RGB and depth images in real-time and at high rates. In addition, let $\mathbf{x}_s \in \mathcal{C}$ and $\mathbf{x}_g \in \mathcal{C}$ be the current and goal positions of the robot, respectively. The robot's main objective is to reach the goal configuration while avoiding collisions with the obstacles.

The above problem is a general motion problem formulation. However, the unknown environment and the lack of direct location sensing impose some challanges. In particular, reliable localization is essential for collision avoidance and trajectory following when traversing a path to the goal. Thus, a motion planning strategy is required that uses gained information from the environement to find a feasible path to the goal. A feasible path is the one that is collision-free and enables the robot to traverse it without tracking failures. By definition, tracking failure would occure when the number of features seen by the camera is below a given threshold λ. If the number of features is below λ, the robot might lose its location and fail to reach the goal. Let $F : \mathcal{C} \to \mathbb{Z}$ be a map that yields the number of features seen at a given robot configuration. Therefore, the motion planning problem is now to find a path $\alpha : [0, 1] \to \mathcal{C}_o$ where

$$\mathcal{C}_o = \{\mathbf{x} \in \mathcal{C} : \mathbf{x} \in \mathcal{C}_{free}, F(\mathbf{x}) \geq \lambda\}, \tag{1}$$

such that $\alpha(0) = \mathbf{x}_s$ and $\alpha(1) = \mathbf{x}_g$. We note however that $\mathcal{C}_{free}$ and F are not known in advance and impose another challange addressed in this paper. Thus, feature-based VSLAM will be used to provide real-time approximation of $\mathcal{C}_{free}$ and F, i.e., $\tilde{\mathcal{C}}_{free}$ and $\tilde{F}$.

IV. Computing Associated Map Points

In order to plan a path $\mathcal{P}$ that satisfies $\mathcal{P} \subset \mathcal{C}_o$, an approximation of map F must be obtained. Such map is termed the Map points Association Model (MAM), which is used to predict the number of associated map points in a given state. Let $\mathcal{D}$ be the contemporary set of acquired map points and denote one map point as $\mathbf{p}_{\mathcal{R}}^{(i)} \in \mathcal{D}$, where $\mathcal{R}$ is the reference frame for the map point, i as the index of the map point. Let $\mathbf{T}_{\mathcal{OR}}(\mathbf{x}) \in SE(3)$ be the rigid body transformation from $\mathcal{R}$ to the camera frame $\mathcal{O}$. The map point's coordinate in the camera frame can be computed by $\mathbf{p}_{\mathcal{O}}^{(i)} = \mathbf{T}_{\mathcal{OR}}(\mathbf{x})\mathbf{p}_{\mathcal{R}}^{(i)}$. The projection $\tilde{\mathbf{p}}^i$ of the map point onto the image plane is computed as

$$\tilde{\mathbf{p}}^i = \Pi(\mathbf{p}_{\mathcal{O}}^{(i)}) \tag{2}$$

where $\Pi : \mathbb{R}^3 \to \mathbb{Z}^2$ represents the geometric projection model [21]. For a map point to be included, it must satisfy three conditions illustrated in Figure 3. First, we check if the map point is within the camera's field of view (condition 1). For each map point in VSLAM, we can retrieve its descriptor information. Thus, we define a scale invariance region (d_{min}, d_{max}) and distance d between the camera center and the map point. The second condition states that map points which satisfy $d \in (d_{min}, d_{max})$ are kept while others are discarded due to difficulty of descriptor association (condition 2).

In keyframe-based VSLAM, map points are jointly optimized with one or more keyframes. With the poses of keyframes related to the map point, we can compute its viewing direction from these keyframes, and their mean (mean keyframe viewing direction). The viewing angle difference α between the mean keyframe viewing direction and the current viewing direction $\mathbf{x}$ is then computed. Because different viewing angles might cause difficulty in descriptors associating, map points must satisfy the third condition: $\alpha < \alpha_{max}$, where α_{max} can be determined by the rotation invariance of the descriptor (condition 3).

Due to noise in the images and randomness in VSLAM's data association, the map points which satisfy the three conditions above may still not be recognized and associated to features in the images. We apply a data driven approach to address this problem. When we apply VSLAM to a sequence of images, we count the number of times each map point satisfies the conditions ($n_s^{(i)}$) and is successfully associated ($n_a^{(i)}$). A ratio $v^{(i)} \in [0, 1]$ between $n_a^{(i)}$ and $n_s^{(i)}$ is computed to indicate the association probability of this map point. By denoting the set of map points that satisfy the three conditions as $\tilde{\mathcal{D}}$, the camera model can be represented as:

$$\tilde{F}_{\mathcal{D}}(x) = [\sum_i v^{(i)}], \quad \forall \mathbf{p}_{\mathcal{R}}^{(i)} \in \tilde{\mathcal{D}}. \tag{3}$$

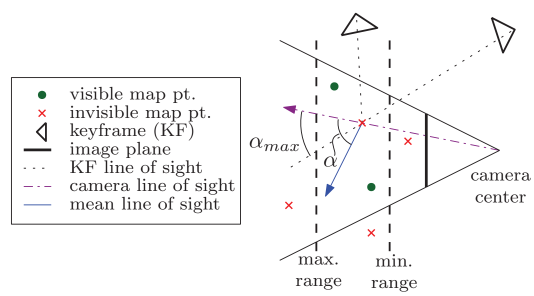


Fig. 3. Map points approximation model for computing the number of visible map points in VSLAM.

V. Navigation Framework

A. Overview

The objective of the robot is to reach the goal while avoiding obstacles. We use a distance-optimal path planner to plan a path to the goal. However, since the environment is unknown, the robot must construct a map to reduce uncertainties, identify obstacles, and localize. Localization is done using VSLAM for state estimation. Mapping the environment to reduce uncertainties is done by classifying the space in a grid structure. Then, frontier regions marking the bounds between unoccupied and unexplored regions are identified. These frontier regions are potential exploration targets to reduce uncertainties.

The planner constrains the path to cross only in explored collision-free and feature-rich regions. Thus, a direct path from the start state reaching the goal is unlikely to be acquired due to uncertainties. Therefore, we perform a series of replannings to acquire more information. In each replanning, a path will be outputted reaching the most advanced state toward the goal. After each traverse of the goal, we perform an uncertainty reduction operation where the robot rotates in place toward frontier regions in its proximity. This simple step increases the explored space and provides the planner with more opportunities to approach the goal. These steps are described in the following subsections.

B. Proximity exploration

During motion, a map is built as a 3D grid structure using OctoMap [22] and is later projected to the 2D floor plane. Similar to [23], each cell in the map is classified into four states: unexplored, occupied, free or frontier. Unexplored regions contain cells which sensory information is not yet available. Occupied cells contain obstacles and cannot be visited, while free cells are explored cells that are free of obstacles. Frontier cells are defined to be free cells that have at least one unknown neighbor cell. They are identified based on the extraction algorithm described in [23]. Let $\tilde{\mathcal{C}}_{free}$ and $\tilde{\mathcal{C}}_{obs}$ be the contemporary approximations of $\mathcal{C}_{free}$ and $\mathcal{C}_{obs}$, respectively. In each update of $\tilde{\mathcal{C}}_{free}$ and $\tilde{\mathcal{C}}_{obs}$ (in OctoMap), unexplored cells now marked as free are checked whether they have at least one neighbor. In this process, OctoMap constructs and maintains a map structure $M = \{\tilde{\mathcal{C}}_{free}, \tilde{\mathcal{C}}_{obs}, \mathcal{C}_{fro}\}$ in real-time. Subset $\mathcal{C}_{fro}$ is the contemporary information of the frontier cells. Cells in $\mathcal{C}_{fro}$ are treated as obstacles in the motion planning until they are marked as free with further gain of information.

During the traverse of a planned path, more sensory data is taken and the frontier boundaries are pushed back, i.e., the known region grows larger. With that exploration, more features are acquired for the motion planner. However, some regions are less likely to be approached for exploration due to tracking failure avoidance (discussed in the planning section). Thus, to reduce uncertainties along the path and provide the planner with more information, after each traverse of a planned path, the robot will rotate toward unexplored regions within its proximity. In particular, the robot will try to have a direct line of sight to unexplored regions. In order to do so, we identify a set of frontier cell clusters $\mathcal{W}$ as proposed in [23] and represent them by their geometric center. Then, after each traversed path, the robot will attempt to rotate in place and face frontier clusters that are within radius r and are not occluded by obstacles. Rotation is done as much as possible within the feature threshold. In other words, when the number of the associated map points is lower than a threshold (100 in our experiments), the robot will stop the proximity exploration and rotate back. This exploration enables the robot to acquire more features and gain more information of M. After doing so, the replanning has more opportunities for a feasible path toward the goal. The planning methodology is discussed next.

C. Path planning

Our path planning is based on the RRT* [7] to find a distance-optimal path $\mathcal{P} \subset \mathcal{C}_{free}$ to the goal. The RRT* planner incrementally extends a search tree in $\mathcal{C}$ with some bias toward the goal. In each iteration, a node in the tree is extended towards a randomly sampled node in the configuration space. The tree is extended to the sampled point only if the edge connecting them is feasible. That is, discretized states along the edge are checked whether they satisfy collision and map points constraints. Using the reconstructed map M, each state is checked whether it is not in collision with occupied cells or within frontier cells. In addition, the approximated number of associated map points from that state must be above the pre-defined threshold λ. This is to ensure that the robot observes enough map points along the path and avoids tracking failure. If the edge is feasible, the sampled node is added to the tree followed by the rewiring phase to maintain optimality.

The planning will run for a pre-defined amount of time in an attempt to find a solution and optimize the path. Due to the uncertainties along the motion and the constraints described above, the planner will not be able to plan a full path reaching the goal if the goal region has not been explored. Therefore, the intermediate planning output will be a feasible path reaching as close as possible to the goal. After reaching the intermediate goal, the robot will replan based on new information gained during the traversed path. The overall replanning and path traverse strategy is discussed next.

D. Algorithm

Algorithm 1 describes the main algorithm for motion toward the goal while planning, mapping the environment and proximity exploration. It starts by updating the OctoMap and VSLAM map. Then, a path is planned and executed from the current pose toward the goal. The planner will return a path close as possible to the goal based on the current knowledge of the environment. If failed to reach the goal in this iteration, frontier regions are identified and explored by simple rotation toward them. This scheme will be repeated until the goal is reached.

VI. Experiments

In our experiments we first present validation for the proposed MAM. Then, we present experimental navigation results in various scenarios to demonstrate the performance of our approach. Videos of the experiments are included in the supplementary material.

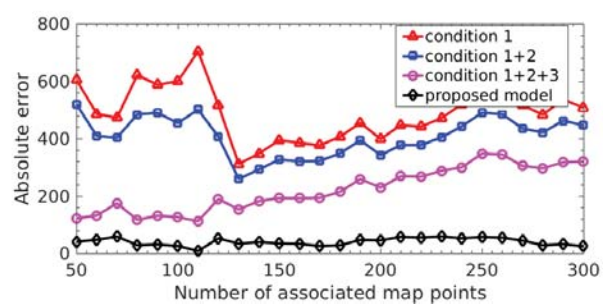


Fig. 4. Evaluation of the map points association model (MAM).

Algorithm 1 Main algorithm

Input: Start state $\mathbf{x}_s$ and goal state $\mathbf{x}_g$.

$\mathbf{x}_{cur} \leftarrow \mathbf{x}_s$
while true **do**
  $M \leftarrow \texttt{UpdateMap}()$
  $D \leftarrow \texttt{Update\_VSLAM\_Map}()$
  $\mathcal{P} \leftarrow \texttt{plan}(\mathbf{x}_{cur}, \mathbf{x}_g)$
  Execute path $\mathcal{P}$.
  $\mathbf{x}_{cur} \leftarrow \texttt{CurrentPose}(\mathcal{D})$
  if $\mathbf{x}_{cur} = \mathbf{x}_g$ **then**
    return success.
  end if
  $\mathcal{W} \leftarrow \texttt{FrontierClusters}(M)$
  for all $\mathbf{q} \in \mathcal{W}$ **do**
    $e \leftarrow \texttt{edge}(\mathbf{q}, \mathbf{x}_{cur})$
    if $\texttt{distance}(\mathbf{q}, \mathbf{x}_{cur}) < r$ and $e \in \tilde{\mathcal{C}}_{free}$ **then**
      Rotate toward $\mathbf{q}$.
    end if
  end for
  Rotate back to angle in $\mathbf{x}_{cur}$.
end while

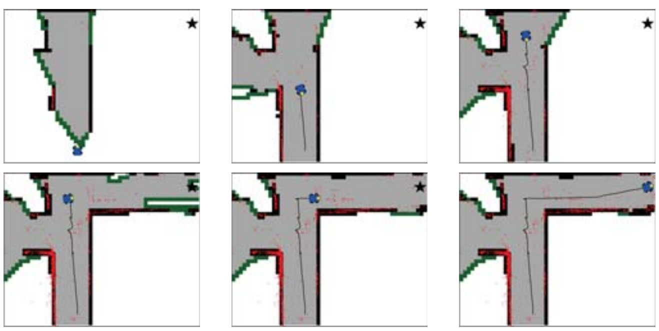


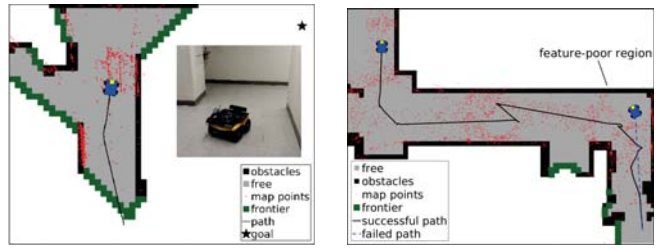
Fig. 5. Evolution of a successful trial using the proposed method in environment I. The robot manages to turn on the corner by backing up while using richer regions of features. Legend is as in Figure 6.

A. Implementation setup

Our proposed algorithm is implemented in the Robot Operating System (ROS). Thus, integration to any system supported in ROS is simple. The following experiments were conducted in various environments using the Jackal ground robot by Clearpath Robotics, on which an Xbox Kinect camera was mounted. An Intel-Core i7-6600U Ubuntu machine runs Algorithm 1 and the path following control. In particular, it runs ORB-SLAM and builds the OctoMap. In addition, it uses the Open Motion Planning Library (OMPL) [24] for the RRT* planner. The open-source implementation code is available at `https://github.com/XinkeAE/Active-ORB-SLAM2`.

B. Map points association model validation

In this section, we assess the effectiveness of our predictive model (MAM) for the number of associated map points. ORB-SLAM is performed on three ground robot navigation sequences of TUM-RGBD dataset [25] (fr2/pioneer_360, fr2/pioneer_SLAM, and fr2/pioneer_SLAM2). Each sequence is run for three trials. We record the number of actually associated map points and the number computed with MAM. We compute the absolute prediction error in different ranges of associated map points. In addition, we assess the effectiveness of our model by performing an ablation study. The results are shown in Figure 4. We compare our MAM with a predictive model while only considering the camera's field of view (map points satisfying condition 1 in section IV), considering scale invariance region and camera field of view (map points satisfying condition 1 and 2 in section IV), and considering observation direction, scale invariance region and camera field of view (map points satisfying condition 1, 2, and 3 in section IV). The results demonstrate the benefit of each condition proposed in section IV in predicting if a map point can be associated. In addition, we can see that applying the recognition probability can result in a significant improvement in prediction because of noise consideration in map points association.



(a) (b)

Fig. 6. (a) Tracking failure when running naive planning (without a feature constraint) in environment I. (b) A successful trial in environment II when using the proposed approach and a tracking failure when running naive planning.

C. Experimental results

The performance of the proposed approach is validated in two environments where passing through low feature regions is necessary. For each of the environments I and II, start and goal configurations were defined and the minimal possible path lengths (disregarding the feature tracking constraint) are $9.23\ m$ and $12.8\ m$, respectively. We first ran 10 naive trials where the distance-optimal planner considers only collisions. In all trials, tracking failure occurred while approaching a wall poor of features. Figures 6a and 6b show examples of tracking failures in both environments. It is important to note that map points seen on the free space are points on the floor. SLAM has difficulty associating these points and they are considered unreliable.

Next, we executed 10 trials of Algorithm 1 in both environments with $\alpha_{max} = 30°$ and $\lambda = 60$. An evolution of the map and path during motion in environment I is illustrated in Figure 5. Here, the robot approached the corner backwards while initializing and associating richer regions of map points. Once in the corner, it could rotate toward the goal with enough features. This solution was repeated in all successful trials. It is important to mention that without the proximity exploration phase, the robot would have just kept replanning toward the goal without acquiring new

TABLE I

EXPERIMENTAL RESULTS

	env. I	env. II
Success rate w/ constraint	90%	80%
Success rate w/o constraint	0%	0%
Avg. path length [m]	12.55	17.6

information. It will not be able, however, to approach the goal due to the feature threshold and since the planner will yield similar paths. Proximity exploration on the other hand provides the robot with more planning options. A successful traversed path in environment II is seen in Figure 6. Here, the robot approached a point just before the turn and used map points on the doorstep to acquire more map points in regions after the turn. Once performed, a feasible path around the corner could be acquired. The curve in the middle of the path is due to a feature-rich poster on the wall. The robot approached it while acquiring more information. Then, it replanned.

The success rates of the trials and the mean path length are seen in Table I. Failed trials with the proposed method are due to errors in the state estimation. During the experiments, we observed accumulating deviations due to hardware heating which affected the localization. Thus, the failed trials were the latter ones. Nonetheless, we have validated the approach and shown that tracking failure can be avoided by planning under a feature constraint.

VII. CONCLUSIONS

In this paper we have presented a framework for planning and traversing a path by a mobile robot toward a goal and through an unknown environment. The robot is equipped with a Kinect camera to map its proximity, identify obstacles and localize itself. VSLAM is used to build a map points association model, so that the number of associated map points at each pose can be predicted. With that model and the contemporary map, a real-time distance-optimal RRT* planner can plan a path that satisfies two constraints: a path free of collisions and the number of map points observed at each point is above a desired threshold. A methodology of replanning and proximity exploration enables the traverse of the environment without collisions and tracking failure. We have demonstrated the method through experiments in two environments with high success rates.

ACKNOWLEDGMENT

This work was supported by the National Science Foundation Grant No. 14-46765 and 14-27111.

REFERENCES

[1] I. Arvanitakis, K. Giannousakis, and A. Tzes, "Mobile robot navigation in unknown environment based on exploration principles," in *2016 IEEE Conference on Control Applications (CCA)*, Sept 2016, pp. 493–498.

[2] Y. Stergiopoulos, M. Thanou, and A. Tzes, "Distributed collaborative coverage-control schemes for non-convex domains," *IEEE Transactions on Automatic Control*, vol. 60, no. 9, pp. 2422–2427, Sept 2015.

[3] S. M. LaValle, *Planning Algorithms*. New York, NY, USA: Cambridge University Press, 2006.

[4] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1309–1332, 2016.

[5] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "Orb-slam: a versatile and accurate monocular slam system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[6] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart, "The euroc micro aerial vehicle datasets," *The International Journal of Robotics Research*, 2016.

[7] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The international journal of robotics research*, vol. 30, no. 7, pp. 846–894, 2011.

[8] R. Bajcsy, "Active perception," *Proceedings of the IEEE*, vol. 76, no. 8, pp. 966–1005, 1988.

[9] A.-A. Agha-Mohammadi, S. Chakravorty, and N. M. Amato, "Firm: Sampling-based feedback motion-planning under motion uncertainty and imperfect measurements," *The International Journal of Robotics Research*, vol. 33, no. 2, pp. 268–304, 2014.

[10] N. A. Atanasov, *Active information acquisition with mobile robots*. University of Pennsylvania, 2015.

[11] A. J. Davison and D. W. Murray, "Mobile robot localisation using active vision," in *European Conference on Computer Vision*. Springer, 1998, pp. 809–825.

[12] M. W. Achtelik, S. Lynen, S. Weiss, M. Chli, and R. Siegwart, "Motion-and uncertainty-aware path planning for micro aerial vehicles," *Journal of Field Robotics*, vol. 31, no. 4, pp. 676–698, 2014.

[13] G. Costante, C. Forster, J. Delmerico, P. Valigi, and D. Scaramuzza, "Perception-aware path planning," *arXiv preprint arXiv:1605.04151*, 2016.

[14] A. Bry and N. Roy, "Rapidly-exploring random belief trees for motion planning under uncertainty," in *IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 723–730.

[15] G. Klein and D. Murray, "Parallel tracking and mapping for small ar workspaces," in *IEEE and ACM International Symposium on Mixed and Augmented Reality*. IEEE, 2007, pp. 225–234.

[16] S. Weiss, M. W. Achtelik, S. Lynen, M. Chli, and R. Siegwart, "Real-time onboard visual-inertial state estimation and self-calibration of mavs in unknown environments," in *IEEE International Conference on Robotics and Automation*. IEEE, 2012, pp. 957–964.

[17] M. Bryson and S. Sukkarieh, "An information-theoretic approach to autonomous navigation and guidance of an uninhabited aerial vehicle in unknown environments," in *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*. IEEE, 2005, pp. 3770–3775.

[18] A. Kim and R. M. Eustice, "Active visual slam for robotic area coverage: Theory and experiment," *The International Journal of Robotics Research*, vol. 34, no. 4-5, pp. 457–475, 2015.

[19] S. A. Sadat, K. Chutskoff, D. Jungic, J. Wawerla, and R. Vaughan, "Feature-rich path planning for robust navigation of mavs with mono-slam," in *IEEE International Conference on Robotics and Automation*. IEEE, 2014, pp. 3870–3875.

[20] C. Mostegel, A. Wendel, and H. Bischof, "Active monocular localization: Towards autonomous monocular exploration for multirotor mavs," in *IEEE International Conference on Robotics and Automation*. IEEE, 2014, pp. 3848–3855.

[21] Y. Ma, S. Soatto, J. Kosecka, and S. S. Sastry, *An invitation to 3-D vision: from images to geometric models*. Springer Science & Business Media, 2012, vol. 26.

[22] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "Octomap: an efficient probabilistic 3d mapping framework based on octrees," *Autonomous Robots*, vol. 34, no. 3, pp. 189–206, Apr 2013.

[23] C. Zhu, R. Ding, M. Lin, and Y. Wu, "A 3d frontier-based exploration tool for mavs," in *2015 IEEE 27th International Conference on Tools with Artificial Intelligence (ICTAI)*, Nov 2015, pp. 348–352.

[24] I. A. Sucan, M. Moll, and L. E. Kavraki, "The open motion planning library," *IEEE Robotics & Automation Magazine*, vol. 19, no. 4, pp. 72–82, Dec 2012.

[25] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of rgb-d slam systems," in *Proc. of the International Conference on Intelligent Robot Systems*, Oct. 2012.