Autonomously Managing Uncertainty: Active Simultaneous Localization and Mapping

Andrew Choi, Viacheslav Inderiakin, Yuki Shirai, Yusuke Tanaka

***Abstract*— A highly multidisciplinary problem, Simultaneous Localization and Mapping (SLAM) has made astonishing progress over the past decades in handling robot perception but has traditionally lacked autonomy. Autonomous mobile robots operating in real-world environments must fulfill three core competencies: localization, mapping, and planning. Solutions to these problems are important in order to perform tasks such as exploration, search and rescue, inspection, reconnaissance, target-tracking, and others. Herein lies the purpose of active SLAM: to allow a robot to not only autonomously operate within an unknown environment, but to do so in a manner that prevents failure of SLAM. The purpose of this survey paper is two-fold. First, we intend for this paper to serve as an educational introduction to the topic of active SLAM. Second, we hope the paper serves as a resource for researchers as a comprehensive collection of state-of-the-art approaches to active SLAM.**

***Index Terms*— Robots, Autonomy, Planning, Localization, Mapping, Learning, Optimal Control, SLAM**

I. INTRODUCTION

Recently, mobile robots have gathered attention from all over the world by accessing dangerous and unknown environments unfit for human-beings such as lunar/planetary exploration [1], search and rescue in disaster areas [2], inspection for nuclear power plants [3] and others. In order to conduct such missions, autonomous robots require three key capabilities: localization, mapping, and planning. In particular, Simultaneous Localization and Mapping (SLAM) has been well-studied in the past decades [4], [5], [6], resulting in formulations that enable robots to estimate their state (odometry) and environment model (map) in unknown settings. SLAM as a framework focuses primarily on perception and has been traditionally purely passive where robot trajectory is either guided by a human or is simply a retracing of an already laid out path. On the other side of the spectrum, planning algorithms focus primarily on decision making with sensor data begin given which results in a complete map and accurate odometry. However, in real scenarios, neither a complete map or optimal path will be priori knowledge. Therefore, it is vital for robots to be able to simultaneously deal with SLAM and planning in a way that produces reliable and robust autonomous navigation in unknown environments.

To deal with such a problem, active SLAM is proposed, which consists of planning that takes into consideration SLAM performance [7], [8], [9]. So far, a number of active SLAM works have been studied using several different approaches such as optimal control theory (e.g. Model Predictive Control) and learning (e.g. Reinforcement Learning) to solve tasks falling into either navigation where the goal is to arrive at a goal destination, and/or exploration which strives to achieve an accurate/complete map.

The structure of the paper is as follows: in chapter II, we start off with a mathematical introduction to the fundamental sub-components of active SLAM. In chapter III, we outline the basic framework/formulation of active SLAM that consists of three sub-problems and explain the importance of each of these sub-problems thoroughly. In chapter IV, we explain state-of-the-art approaches to the two main problems of active SLAM: navigation and exploration. Here, we discuss the methodologies and results of these approaches. Next, in chapter V, we discuss interesting findings from our survey as well as our inferences on the matter. Chapter VI goes on to propose and discuss several future research directions that have high importance for active SLAM. Finally, in chapter VII, we conclude with an overall summary of the the paper as well as some final discussion.

II. MATHEMATICAL FOUNDATION

Before delving into the topic of active SLAM, it is essential to first introduce the necessary mathematical fundamentals required by active SLAM. Three main topics will be discussed: SLAM, planning methods, and finally, the melding of the two, active SLAM.

A. SLAM

SLAM addresses a fundamental problem in robotics where a robot must predict its odometry as well as the map through noisy observations using sensors [10]. SLAM algorithms must handle both the uncertainty stemming from the odometry as well as from the generated map itself. Following this, SLAM has often been described as a chicken-or-egg problem where an accurate map is necessary for localization and accurate localization is necessary for mapping. To solve this stochastic estimation problem without priori knowledge, several SLAM algorithms have been developed such as EKF SLAM, UKF SLAM, and FastSLAM. These SLAM iterations all make use of a large family of state estimation techniques called Bayesian filters [11] [12]. Although many more SLAM algorithms exist that all use different methodologies, here we choose EKF SLAM as an example of SLAM as EKF SLAM is widely considered the most fundamental SLAM.

The structure of EKF SLAM is as follows [13]:

1) State prediction (Odometry)

[1]University of California Los Angeles, CA 90024, USA asjchoi@ucla.edu, v.inderiakin.uk@gmail.com, yukishirai4869@g.ucla.edu, yusuketanaka@g.ucla.edu

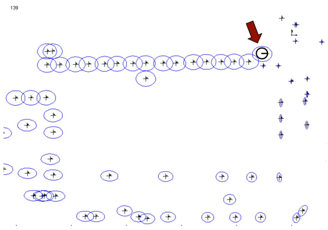

Fig. 1: Pre-loop-closure [13]

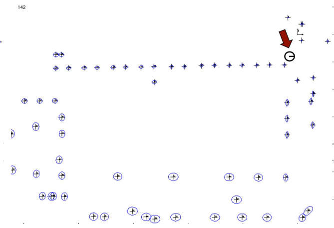

Fig. 2: Post-loop-closure [13]

2) Measurement prediction
3) Observation
4) Data Association
5) Update

The procedure of EKF SLAM can be divided into two key phases: the prediction and correction stages. In the state prediction step, the state including the mean and covariance matrix of a robot is predicted using the motion model of a robot. Next, in the measurement prediction step, future measurements are predicted through an observation model. After predicting the future state and sensor inputs, the robot then obtains measurements using sensors. The robot then associates the obtained measurements with stationary features of the environment that can be recognized reliably, known as landmarks [13]. Using these landmarks, the robot then corrects its state using the measurement information.

One key characteristic of SLAM is "loop-closure", which is when the robot re-visits already mapped locations. By conducting loop-closures, the robot is then able to reduce its map and odometry uncertainty dramatically [14] [15]. The advantage of closing a loop is that it decreases the uncertainty of SLAM by comparing the observed map with the map acquired before. The effect of loop-closure is contradicted in Fig. 1 and Fig. 2. In Fig. 1, the uncertainty of the robot and landmarks are relatively large while in Fig. 2, the uncertainty collapses because the robot closes a loop.

Loop-closure is essential for a mobile robot to limit uncertainty of SLAM.

B. Planning Methods

While SLAM is primarily a perception problem, pure planning deals with developing a sequence of actions with environment information provided. In this paper, three key methods of planning seen in active SLAM are discussed. First are sampling-based methods that attempt to find efficient trajectories through state space sampling. Second are control theoretic methods, in particular, model predictive control, also known as receding horizon control. Lastly, a relatively new and growing framework of active SLAM, reinforcement learning methods are also discussed.

1) Sampling-based Methods

Sampling-based algorithms make up one of the most popular subclasses of path-planning methods with the majority of state-of-the-art approaches for motion planning being built upon classical algorithms such as Probabilistic Roadmap (PRM) [16] and Rapidly Exploring Random Trees (RRT) [17]. These methods successfully deal with large state spaces

Algorithm 1: Rapidly-Exploring Random Trees (RRT) [17]

```
Vertices are V, Edges are E
Construct configuration space with obstacles C_obs
Set start state s_s, set k = 0, set number of vertices
  K, set drive time dt
while k is not K do
    sample x_rand ∈ C/C_obs
    find x_nearest ∈ V
    from x_nearest drive towards x_rand for dt
    arrive at x_new with trajectory of states
    if trajectory ∉ C_obs then
        add node x_new to V
        add edge (x_nearest, x_new) to E
    end
    increment k
end
```

by reducing them to a set of randomly-picked states resulting in greatly reduced computational cost when compared to continuous domain planning methods such as MPC. Sampling-based methods are *probabilistically complete* meaning that given enough time, sampling-based approaches are guaranteed to arrive at a viable path so long as one exists. In other words,

$$\lim_{t \to \infty} Probability(solution) \to 1$$

These methods are only probabilistically complete as they are unable to determine the non-existence of a path.

In algorithm 1, the psuedocode of RRT is shown where states are randomly sampled from a set of all possible states. From this, the nearest node in the current RRT tree is identified. From $x_{nearest}$, the robot then attempts to drive towards x_{rand} for a set amount of time until arriving at x_{new}. So long as such a trajectory is collision-free, x_{new} and the generated trajectory can be added to the vertices and edges of the RRT tree, respectively. A viable path is found when a node is located in the goal state/region.

One drawback of these methods is the lack of guarantee in terms of path optimality. To combat this, Karaman and Frazzoli created arguably the most widely implemented optimal motion planning algorithm RRT* [18]. This algorithm builds upon RRT by conducting rewiring operations to generate optimal paths as determined by a defined cost parameter. Variants of this algorithm are widely employed in active SLAM as can be seen in later sections.

2) Model Predictive Control

Model Predictive Control (MPC) is an established method in active SLAM as it draws attention from the robotics community [19]. MPC is one of the optimal control theories that enables the system to reach the goal state by computing the optimal action based on the predicted state using a model in a continuous domain. In particular, MPC can take into account dynamic constraints; that means in active SLAM a planner incorporates new constraints such as new obstacles as a robot moves. Thus, MPC can be used in active SLAM

Algorithm 2: Model Predictive Control (MPC)

Compute an optimal action based on predicted state
while do
  predict
  x$(\tau+1) = Ax(\tau) + Bu(\tau), \tau = t, \ldots, t+T$
  compute the optimal action
  $u_{0:N-1} = \arg\min_{u_{0:N-1}} J^*(\tau)$
  choose the 1 st input u* = u_0
  if *x(N)* $\in \mathcal{X}_N$ **then**
    break
  end
end

where the optimization in dynamic environments is essential for planning an action.

The formulation of MPC is shown in Equation 1 where $J^*(\tau)$ is the objective function, $x(\tau)$ and $u(\tau)$ is the state and the action, respectively at time τ. A and B are the co-efficient matrices describing the system and motion models, and $\mathcal{U}$ and $\mathcal{X}$ show feasible regions of $x(\tau)$ and $u(\tau)$, respectively.

$$\begin{array}{ll} \text{minimize} & J^*(\tau) = \sum_{\tau=t}^{t+T} \ell(x(\tau), u(\tau)) \\ \text{subject to} & u(\tau) \in \mathcal{U}, \quad x(\tau) \in \mathcal{X}, \quad \tau = t, \ldots, t+T \\ & x(\tau+1) = Ax(\tau) + Bu(\tau) \\ & x(t+T) = 0 \\ & \quad x(N) \in \mathcal{X}_N \end{array} \tag{1}$$

The algorithm of MPC is shown in Alg. 2. Using the motion model, the planner first predicts the future states $x(\tau)$, where $\tau = t, \ldots, t+T$. Then MPC calculates a set of T actions by solving the optimization problem. T is also called the prediction horizon. Although in this equations, we use MPC for linear dynamic systems as an example of MPC, MPC is able to deal with nonlinear dynamic system by linearizing the dynamics and stochastic dynamic systems by converting the stochastic equations into deterministic equations using mean and covariance matrix, and others.

One drawback of MPC is the computational cost. MPC predicts the future state of a system over the prediction horizon T. Hence, if MPC has a large T, it outputs the set of actions that consider the future states as well as incoming states. However, a large T will require more time to predict the future states and optimize the actions based on those future states. This trade-off makes it difficult for a mobile robot to use MPC in real scenarios.

Therefore, the key advantage of MPC is that it can deal with changes to the environment by predicting future states, and outputs the best action in the continuous domain naturally.

3) Reinforcement Learning

Traditional reinforcement learning formulates problems as Markov decision processes (MDP) and utilize a value-action function, $Q(s,a)$. This value corresponds to the total discounted sum of rewards for every state and action pair as shown below,

$$Q(s,a) = \mathrm{E}[r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + ...] \tag{2}$$
$$Q(s,a) = \mathrm{E}[\Sigma_t \gamma^t r_t] \tag{3}$$

where r_t is the reward at time step t and γ is the discount factor. By formulating the value in such a way, actions can be assigned a value that takes into consideration future rewards as well. The core purpose of the Q function is to be able to produce an optimal policy π^* that assigns an optimal action to all states. Optimality is defined by the reward function R. Tuning γ changes the weight that is given to future rewards versus immediate rewards. Bellman backups are conducted in an iterative fashion until convergence. Using a method called Temporal Difference Learning (TD Learning), sampling is conducted to find accurate approximations of the true expected value as shown below.

$$Q(s,a) = \mathrm{E}[R(s,a,s^{'}) + \gamma max_a Q(s^{'},a^{'})] \tag{4}$$
$$Q(s,a) = \sum_{samples} P(s^{'}|s,a)[R(s^{'}) + \gamma max_a Q(s^{'},a^{'})] \tag{5}$$

Note that a system model is not required; rather, the transition probabilities are accounted for implicitly through sampling. This procedure is performed until convergence of the optimal Q function, Q^*, is achieved. From here, we can produce the optimal policy.

$$Q^*(s,a) \rightarrow \pi^*(s,a)$$

One drawback of TD learning is the inability to handle large state spaces due to its tabular setting. This can be circumvented through function approximation provided by deep neural networks. Deep RL has shown impressive results in the field of robotics by its ability to develop complex optimal control policies for scenarios that are mathematically difficult to model. One of the areas of robotics that deep RL has shown promise in is that of active SLAM. Arguably the most fundamental deep RL algorithm is the Deep Q-Network (DQN). A method that enabled traditional Q-learning approaches towards computationally large or infinite state spaces through the use of deep neural networks, Mnih et al. showcased several agents that surpassed human capabilities for several Atari games [20]. In DQNs, Bellman backups are performed similarly to traditional Q-learning but updates are now applied to a neural network through gradient descent. By doing so, a neural network learns the relationship between Q values and state/action pairs.

Therefore, the key strength of DQN is that it offers generalization towards state/action pairs that have not yet been experienced so long as similar state/actions pairs have been. Alg. 3 shows the sampling and learning procedure. Detailed information can be found here [20].

C. Active SLAM

As it can be seen from sections II-A and II-B, quality of the map produced by a moving robot depends not only on SLAM performance, but also on the actions it takes. Traditional planning algorithms, however, take an action based on the best known map and consider the robot's odometry

Algorithm 3: Deep Q-Network (DQN)

Initialize weights θ
for *i in E episodes* **do**
  for *time t in ith episode* **do**
    sample (s_t, a_t, s_{t+1}, r_t)
    $Q_\theta(s_t, a_t) \leftarrow Q_\theta(s_t, a_t) +$
    $\alpha[r_t + \gamma max_{a'} Q_\theta(s_{t+1}, a^{'}) - Q_\theta(s_t, a_t)]$
  end
end

and map coordinates as precise. A naive approach such as this that does not take into consideration the uncertainty of the SLAM readings is prone to aggressive trajectories that ultimately lead to SLAM failure (i.e. causing localization accuracy to drop below the minimal acceptable value). To prevent these situations, active SLAM is used integrating SLAM and planning.

1) Coupling of Planning, Localization, and Mapping

First, a crucial distinction must be made between planning with SLAM and active SLAM as this is often a point of confusion. While both methods carry out a planning algorithm using SLAM outputs, a key difference between the methods is that active SLAM planning takes into consideration the effects of a robot's actions on its future SLAM performance. In other words, planning and SLAM are *coupled*. A graphical representation of the key difference between both methods can be seen in Fig. 3.

To perform this coupling, an uncertainty metric describing the current map and localization is incorporated into the robot's planning as a cost. Often times, multiple cost parameters are incorporated such as distance travelled and/or map coverage. Coming up with effective ways to satisfy these conflicting costs is referred to as the *exploration/exploitation dilemma*, which is talked about later in the paper. In the next section, we discuss several representations of uncertainty.

2) Representation of Uncertainty

One of the key component of the active SLAM formulation is the uncertainty representation. Both exploration and navigation algorithms require a metric that characterizes the uncertainty of the map and odometry.Several metrics exist for this purpose and are explained in the next paragraph.

One of the first and most influential papers devoted to determining the optimal uncertainty representation [21] proposed to use A-optimality criteria (i.e. a trace of the covariance matrix) as a quantifier of uncertainty. Some state-of-the-art works employ entropy [22], [23], [24], [25] as a decision criterion, but might also use D-optimality (i.e. determinant of the covariance matrix) [26], [27], [28] or even propose novel criterion like expected uncertainty [29]. Overall, there is no consensus on which decision criterion is able to provide better performance and the choice of uncertainty representation remains implementation-specific [23]

Carrillo proposed a tuningless utility function which compromise Renyi's entropy, which is a general form of Shannon

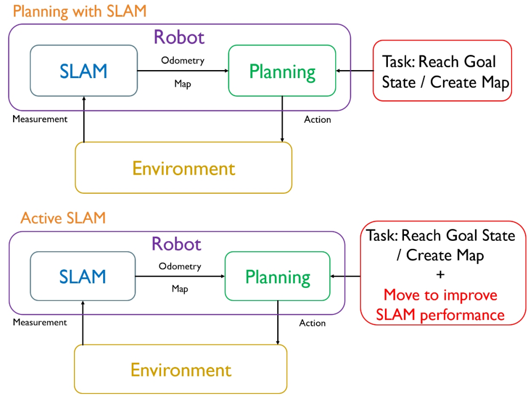

Fig. 3: SLAM with Planning vs. Active SLAM: Note the difference in the planning objectives. Although the difference is somewhat subtle as the two diagrams are nearly identical, the trajectories produced by both methods may differ significantly.

entropy. [23]. Carrillo defined information gain, α as in (6), such that it increases as pose uncertainty, σ rises. As α increases a robot tends to do exploitation and vise versa.

$$\alpha = 1 + \frac{1}{\sigma} \tag{6}$$

III. Three Fundamental Components of Active SLAM

As illustrated by Cadena et al. [6], the task of active SLAM can be broken down into three fundamental sub-problems:

1) Selection of Vantage Points
2) Finding Optimal Action through Utility Function
3) Deciding when to Terminate Active SLAM

In this section, we explain what these problems mean as well as their role in the overall problem of active SLAM.

A. Selection of Vantage Points

Vantage points are essentially a subset of states that are predicted to hold high utility. It is imperative that a subset is chosen as evaluating every possible action in the robot and map space quickly becomes computationally intractable for most real world applications.

Vantage point selection algorithms depend on the current state of the robot in terms of its odometry and map uncertainty. When the localization accuracy is high and the robot is performing an exploration strategy, points near the frontier of the known map have the highest potential for providing information of the environment and consequently, they should have higher probability of being selected [30]. However, frontier detection in itself remains to be a computationally complex problem (especially on large maps), and various approaches are proposed to reduce its computational complexity [31], [32].

In contrast, when the robot's localization accuracy drops and it follows an exploitation strategy, vantage points should be chosen in feature-rich regions of the already explored map. There is no consensus on how to place vantage points for a loop-closure. One of the approaches proposed by Lehner et al. is to define loop-closure vantage points at the center of generated submaps [24].

B. Finding Optimal Action through Utility Function

A utility function measures the value (utility) of a certain robot state given the current robot state and map. Utility functions must take into consideration the current map quality in assigning the utility of actions. A well designed utility function is necessary for a robot to be able to best determine the optimal action. It goes without saying that the design of the utility function significantly defines the active SLAM behaviour.

Map quality and robot localization uncertainty are two key performance metrics used in the active SLAM utility function. These metrics are necessary so that the robot can autonomously choose between exploration and exploitation through the utility function. Human operators make loop closures occasionally to ensure map quality. Such procedures have to be reproduced autonomously in active SLAM but only when necessary. Actions that improve SLAM performance should be rewarded in order to encourage the robot looking up known features. Another aspect of cost representation is to discount action values that could potentially cause SLAM failure, such as entering featureless area with poor localization [33] or harsh movements to SLAM sensors [34], [35]. These cost terms are subtracted from the other terms so as to reduce the corresponding utility values.

Another performance metric intrinsic for active SLAM algorithms designed for exploration can be as map coverage. This metric is included to encourage the robot to explore unknown parts of map and may be represented as a percentage of a covered map or as information gain [24]. This metric can also be expressed as the inverse of map uncertainty, but in most of cases is explicitly included in the utility function.

Other cost metrics are not mandatory in active SLAM utility functions, but they are often utilized to penalize movements that are expensive under some perspective, such as visiting a far point.

In addition to defining those representations, each of them should be weighted such that the robot can effectively accomplish the allocated tasks. Some Nobel works proposed auto tuned weighting parameters or tuneless methods on specific SLAM [23] or planning methods [45], [44] since engineer-define weights are required to tune on each environment, map representation, robot, and sensor[23].

C. Deciding when to Terminate Active SLAM

Due to the heavy computational expense of performing active SLAM, effective decision making in knowing when to end active SLAM is crucial for autonomous operations. Although this step is essential, the decision on whether or not active SLAM is still required is an open problem as it is challenging to formulate the stopping criteria on active SLAM [39].

IV. The Two Goals of Active SLAM

The overall tasks of active SLAM can be divided into two categories: exploration or navigation. Although both tasks have significant overlap, the structuring of the problem has many key differences dependent on which of these two problems active SLAM is trying to solve.

For the exploration problem, the goal is to explore the environment and generate the highest quality map possible in the shortest time as defined by the utility function. Metrics which are usually used to evaluate such algorithms include area coverage, overall map uncertainty, completion time, etc.

For the navigation problem, the goal is to achieve a destination state with the highest probability of success and in the shortest amount of time. The challenge of the navigation problem is to explore the environment just to the point that is necessary to get to the goal state. Metrics used for navigation problems may include distance travelled, completion time, etc.

A. Accomplishments in Exploration Problem

Active SLAM frameworks that attempt to solve the exploration problem have been around for several decades. One of the earliest frameworks, Dissanayake formulated the exploration problem using MPC [36], [37], [38]. As explained previously, in MPC, two steps are necessary: prediction and optimization. As a prediction step, the author used EKF to predict the future odometry and map [36], and in the optimization step, he used the trace of the covariance matrix of the robot odometry and map as a utility function which the robot minimized at every time step.

In [36], however, the author described that MPC suffers from small prediction horizons as the robot cannot get enough information for optimal planning. The obvious solution would be to simply increase the prediction horizon but this exponentially increases the computational cost. Therefore, it was proposed that by combining MPC with an attractor that provides global information to MPC, the robot is able to explore more of the map with an acceptable uncertainty. In particular, map coverage dramatically increased from 88 % to 100 % [37], [38].

Using a more information-theoretic approach, Lehner et al. suggested metrics such as Information gain (IG), match fitness, and match effect in order to balance between efficiency and map quality [24]. In this paper, two type of vantage points were defined: a frontier point and a revisit point as shown in Fig. 4. First, IG was used to represent the amount of information potentially gained at a vantage point to evaluate exploration. For loop-closures, match fitness was computed based on the number of feature, and match effect measured the map quality improvements. Under this utility function, it was shown that a robot could autonomously decide between vantage points and perform loop-closures only when necessary.

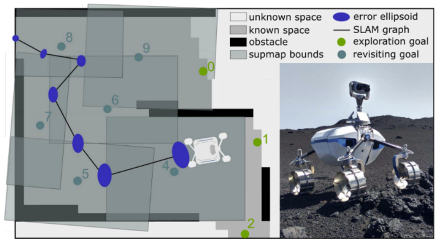

Fig. 4: Submap matching and frontier/revisiting vantage points [24].

Fig. 5: Comparison between balanced active SLAM and SLAM with naive planning. Left: the proposed method, Middle: the frontier method, Right: true map [23]. 3 minutes of exploration.

More insights into how differing utility functions affect exploration performance can be seen in [22]. Here, Vallvé and Andrade-Cetto proposed two methods: one that minimized distance travelled and the other that minimized joint path and map entropy per meter travelled, both using Pose SLAM and RRT* variants. In their work, they showcased that the efficiency and quality of map coverage was highly dependent on the formulation of the utility function with thorough analytical comparisons.

Another great work using sampling-based algorithms was conducted by Jadidi et al. which proposed a method that allowed dense map representations and incorporated the full state uncertainty into the planning process with an information-theoretic convergence criterion for incremental motion planning [39]. As explained in chapter III-C, research dealing with developing efficient stopping criteria is sparse and requires attention. Considering this, their largest contribution is the formulation of a natural automatic stopping criterion for information-driven active SLAM. Jadidi et al. shows that by computing the least upper bound of the average map entropy, regardless of the quantity of interest, an effective stopping criterion is formed. Using the proposed criterion, the authors reported that the robot was able to terminate active SLAM in appropriate scenarios.

One of the key problems in active SLAM is computational cost as active SLAM takes into account the states of both a robot and map. By dealing with more variables, this leads to expensive computation when it does prediction, optimization, and more in planning. To deal with this problem, several approaches are proposed.

Mu et al. handled computationally expensive active SLAM mapping with sampling-based path planning using topology featured graph (TFG) which represents the map as geometries instead of grid cells [40]. TFG not only requires less computational resources, but also can quantify both robot pose and features naturally to balance exploration and exploitation.

Other avenues of active SLAM research strive to reduce the computational complexity of active SLAM operation. One such example can be seen in [41], where Chaves and Eustice proposed an active SLAM framework that utilized a Bayes tree data structure for efficient planning. This tree cached the utility for past actions which let the robot avoid redundant calculations for similar actions.

Y. Chen et al. addressed a problem of computational complexity for MPC active SLAM [28]. By approximating the objective function, they managed to obtain suboptimal solutions in substantially lower time.

Another challenge that active SLAM faces is map coverage. One method that achieves better map coverage but sacrifices computational efficiency in the process is [42]. Meng et al. suggested that the planner selects all points on the edge of the observed map as exploration vantage points. Once a robot performs a action, the planner computes all possible trajectories that sequentially lead the robot through all of the vantage points and chooses the trajectory that maximizes information gain and minimizes distance traveled. Furthermore, the robot performs only the first interval of this trajectory, and then re-selects vantage points. Although this allows the robot to decrease the number of steps to achieve acceptable map coverage, operation increases in computational cost.

Chaves et al. also proposed an active SLAM framework that produced better map coverage. Often times, conducting loop-closures during exploration can lead to increased path lengths and coverage of already explored areas. To combat this redundancy, Chaves et al. developed ways of determining opportunistic loop-closures that are coverage efficient [43].

Utility functions using entropy are often incorporated with weighting parameters to balance exploration and exploitation since a robot tends to search new area regardless of map quality otherwise. However, these weighting matrices have to be re-tuned for each environment and map resolution.

Carlone et al. tackled tuneless exploration with Rao-Blackwellized particle filters [44]. They applied Kullback-Leibler divergence in order to evaluate posterior approximation, which is utilized to compute ”expected information from a policy” so that a robot can decide between exploration and exploitation autonomously without manual tuning [44]. However, this auto-tuning approach is restricted to particle filter SLAM [44], [45].

Carrillo et al. proposed a general auto-tuning utility function which compromises Renyi’s entropy along with Shannon entropy [23]. Carrillo et al. defined an information gain which increases when the pose uncertainty is high. The utility function produced actions that correct its state estimation when the gain is large. Fig. 5 illustrates the SLAM performance improvements when integrating SLAM uncertainty into planning. The map quality is poor with the frontier

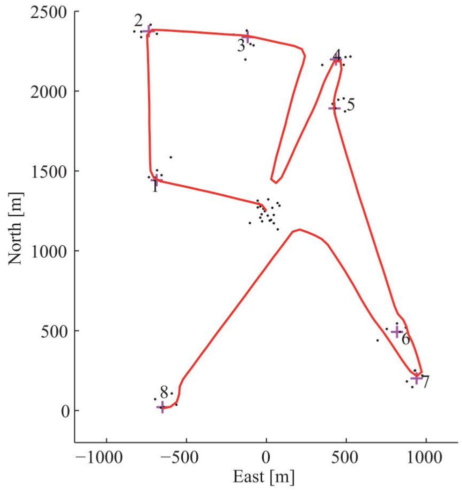

Fig. 6: The trajectory of a robot using the proposed planner in [46].

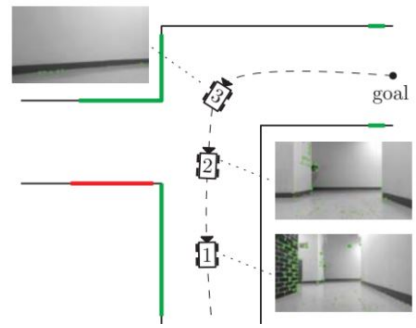

Fig. 7: Visual Slam Failure Case under Naive Planning: Robot fails at step 3. Green indicates feature rich areas. Red indicates hidden feature rich areas [33].

method which always attempts to visit unknown regions whereas the proposed active SLAM successfully captures the true map shape though it covers less area per time due to its conservative planning.

B. Accomplishments in Navigation Problem

Active SLAM frameworks for navigation have also been studied in the past decades, resulting in great works. One of the most influential works was developed by Indelman et al. [46]. In their paper, they relaxed the maximum likelihood assumption. By treating future observations and acquisitions from sensors as random variables, the proposed planner does not lose the stochastic nature of the system. This allowed for more real scenario based solutions while previous works simply used mean values. The author formulated the navigation problem using stochastic MPC. The result of the MPC planner is shown in Fig. 6. In Fig. 6, the robot starts from the center of the figure and has to visit point 1 through 8. In particular, the trajectory from points 3 to 4 show that the robot re-visits the starting area in order to decrease the uncertainty through loop-closure as the predicted uncertainty would be large.

Zhang and Scaramuzza also applied MPC to the active SLAM problem where the proposed planning method explicitly takes into account perception constraints in order to plan towards areas of high perception quality [47]. This enables a robot to take an appropriate action especially in environments with visually degraded regions. In particular, the authors used one single camera for Monocular SLAM and based on the measurements, a library of candidate trajectories are generated and evaluated in terms of perception quality, collision probability, and distance to the destination.

A number of active SLAM frameworks do not have ”guarantees” in terms of the range of the uncertainty if a robot explores at a value below a predetermined uncertainty value because the system is stochastic. However, for many real world applications, it is desired for a robot to arrive at the destination with guaranteed low uncertainty. To address this challenge, Ivanov and Campbell proposed an active SLAM framework with probabilistic guarantees [48], [49], [50]. By formulating the optimization problem with chance constraints that represent the uncertainty at the destination using entropy, the authors report that the robot arrived at the goal state with probabilistic guarantees while avoiding obstacles.

Another avenue of increasing navigation success arises from accurate prediction of future map states. In one such example, Deng et al. proposed a data-driven active visual SLAM framework that predicted the number of map points associated in a given pose [33]. The proposed framework was able to choose actions that optimized for visual SLAM input quality while running a distance optimal RRT* planner. Under the example shown in Fig. 7, rather than failing, at state 3, the robot was able to rotate and detect the hidden feature rich area through its prediction model. With this, the robot was able to successfully reach the goal state.

The approaches that utilize sampling, MPC, or RL often require significant amounts of computational effort which may not be feasible to process with an on-board microcomputer while executing SLAM simultaneously.

Rodrigues et al. introduced the concept of the potential field to overcome the active visual SLAM navigation problem with a microcomputer [52]. In physics, a force will act on the robot toward where the potential energy is low. Using this idea, two potential fields are incorporated: the goal and feature-based field. The goal and feature rich areas are represented with ’low potential energy’, and thus a robot experiences two forces which generate a force vector it follows. Following this force vector matches the robot path with the feature rich regions as shown in Fig. 8. Although this approach suffers from being stuck in non-continuous feature rich regions, the idea of utilizing potential fields is important due to their minimal computational cost.

Although sparse, reinforcement learning methods have also started to be implemented in active SLAM. One example can be seen in [53], where Li et al. proposed a deep reinforcement learning approach for active visual SLAM that trained two RL agents to process the path while SLAM

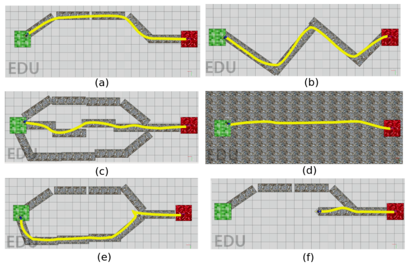

Fig. 8: Trajectories of a robot generated by potential fields [52]. Grey regions represent feature rich and green is the destination, whereas red is the start position.

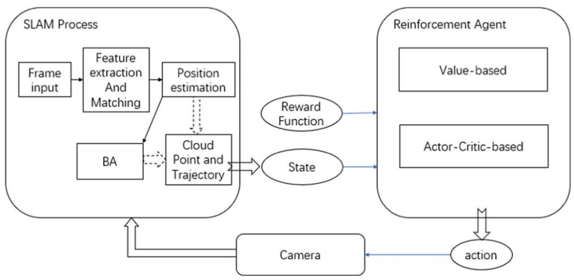

Fig. 9: Active SLAM Framework using Keyframe-based SLAM and RL [53]

is processing. In their approach, they used keyframe-based SLAM with DQN and actor-critic methods to determine the optimal action given SLAM inputs as shown in Fig. 9.

By using a reward function with a large penalty for actions that produce collisions and/or result in SLAM failure, they were able to achieve excellent results in simulation where the robot was able to successfully navigate for long periods of time without SLAM failure. Possibly the most important takeaway was that the learned agent could operate in other maps with satisfactory performance even if it did not train on those maps. Here, the strengths arising from the generability of deep RL approaches can be observed. Another method that showcases this generability can be seen in [34], [35], where Prasad et al. proposed an RL approach that utilized a DQN as an action filter for a robot carrying out naive planning with SLAM. Although not technically active SLAM, this method has many similarities in that actions are shaped in a way that enhance SLAM performance. More precisely, this DQN was able to learn the relationship between which actions were most prone to SLAM failure when carrying out monocular SLAM. Similar to [53], this deep RL framework was also generalizable to several other maps.

V. Findings and Inferences

In this section, we would like to discuss some findings we discovered across several active SLAM papers as well as provide some of our inferences on the matter. These discussions are divided into the following three sections.

1) Frequency of Planning Implementations in Active SLAM
2) Predicted Future Trends

A. Planning Method Trends

Across active SLAM papers, a trend in terms of the frequency of planning method implementations in active SLAM frameworks was observed and listed below. Some inferences as to why this trend exists is also discussed.

1) Sampling-based methods *by a large margin*
2) Model Predictive Control (MPC)
3) Reinforcement Learning (RL)

From these findings, we infer that sampling-based methods received the most research attention in the past decades due to their computational efficiency provided by their discretization of the sample space. This factor alone arguably contributed significantly to the popularity of sampling-based planning algorithms as active SLAM is frequently performed on mobile robots with limited embedded computing capabilities. Another attractive quality of sampling-based methods are their high level of tunability where users can easily trade off between path optimality and computational expense.

Next, MPC, although robust, can be very computationally expensive in active SLAM for certain horizon thresholds as it plans in the continuous domain. This has made it difficult to implement for active SLAM, especially for robot systems with limited computing capabilities [46]. Another limitation of MPC is the lack of flexible tunability that sampling algorithms possess. Whereas sampling-based methods purely sacrifice path optimality, a shorter MPC horizon can be potentially hazardous due to short-sightedness.

Finally, the least frequently observed were reinforcement learning methods. Much of this can be attributed to how RL how ”young” the field is as almost all deep reinforcement learning algorithms being implemented in active SLAM are currently less than five years old. In terms of computational comparison with the prior methods, RL has to potential to be the most efficient planning method so long as learning has been completed offline as the utility of one action can be computed with a single neural network pass. In the next section, we discuss some educated predictions as to how these trends might change in the future.

B. Predicted Future Trends

For active SLAM frameworks, we predict that MPC and RL will see significant increases in research attention. In the case of MPC, this can be simply be attributed to increases in computational limits as time goes on. As the drawbacks of MPC on embedded systems diminishes, we believe that MPC's robust performance will make it a go-to method for active SLAM so long as an accurate system model is present. Likewise, as we enter the age of data, it is becoming more and more time efficient to train robust and viable RL agents for active SLAM operations. Due to RL's capability to produce optimal control policies for systems that are too difficult to model mathematically, RL will continuously receive research attention.

Of the two methods, RL seems to have much more room for potential. This by no means indicates that sampling-based methods will stagnate, rather that such a large disparity in the types of active SLAM implementation may start to diminish.

VI. Future Work

In addition to surveying recent accomplishments in active SLAM, we propose several research directions that we believe have high potential for the future of the field.

1) Reduce Computational Complexity
2) More Comparisons between Frameworks and Baseline
3) Monotonicity of Uncertainty Representation
4) Preventing Sensor-specific SLAM Failure

A. Reduce Computational Complexity

Although we have done a rough comparison between the computational expenses between different methods in the previous section, currently regardless of the planning method used, active SLAM remains very computationally expensive across the board. Not only must the robot predict possible future poses for a set of actions but it must also predict the future map state resulting from these actions as well. As maps can have a tremendous amount of features, this operation increases significantly in terms of computational complexity. Due to this, sacrifices must be made in terms of how frequently active SLAM is run. This computational expense is also the reason for why determining when to terminate active SLAM is one of the three fundamental components to the active SLAM problem.

Finding ways to reduce the computational complexity of active SLAM in general will be very important for reliable performance in mobile robots with limited computational ability.

B. More Comparisons between Frameworks and Baseline

Another key research issue is the lack of comparisons between different frameworks/communities of active SLAM and the lack of an established baseline. Although these two problems could be seen as separate issues, we believe that they are intertwined for reasons which will become obvious in the following section.

The first issue essentially states that although there exists comparisons between two different implementations of RL active SLAM or between two different implementations of MPC active SLAM, there exists very little research in comparing between communities. Due to this, there is little consensus on how different active SLAM frameworks compare with each other in terms of performance. Much of this can be attributed to different experimental assumptions between researchers and lack of access to experimental setups. In fact, there is no consensus on how performance should even be quantified which leads us to the second issue of lack of a baseline. Even when comparisons are made, it remains difficult to establish relationships between comparisons due to the usage of different metrics by different authors. Much of this stems from issues of interchangeability as discussed in the previous section.

Although there isn't a large amount of research for active SLAM comparisons, it would be amiss to say that accomplishments haven't been made in this area. Kim and Eustice made in-depth comparisons between their framework and similar pre-existing frameworks [27]. In their work, they devote an entire section of their paper to analyze different methods of active SLAM, belief space planning, integrated navigation, and area coverage in terms of experimental assumptions, computational complexity, sensors, major operations and more. Building upon this, Indelman et al. also went on to make in-depth comparisons between Kim and Eustices' discrete planning approach with several continuous domain planning methods [46]. Still, comparisons between active SLAM frameworks are still sparse and will be an important research direction for improved understanding of active SLAM.

C. Monotonicity in Uncertainty Representation

While the number of novel/custom decision making criteria continues to increase, little research effort has been devoted to figuring out their fundamental properties. One important question that is usually overlooked is whether the metric they use is monotonic - that is, whether the change in uncertainty always causes a corresponding change in the metric's value. As Carrillo et al. states in [54], it does not always hold true. Under special circumstances, decision making criteria can remain its value or even decrease while uncertainty of a robot's pose grows. Wrong information about uncertainty may lead to selection of a non-optimal action. However, necessary conditions required for monotonicity have only been determined for A-opt, D-opt, E-opt criteria, and Shannon entropy [55]. Investigation of the monotonicity criteria for all decision making criteria is important for more reliable action selection and will require research attention in the future.

D. Preventing Sensor-specific SLAM Failure

SLAM failures that arise due to the use of specific sensors are overlooked in exploration. Carrillo's method autonomously conducts loop-closure to maintain map quality but the utility function is not SLAM fault-tolerant [23]. Visual SLAM performance highly depends on features distributions and often maps are considered to be feature rich to assume SLAM would not fail. However, as in [33], visual SLAM uncertainty could rapidly grow and lose localization by entering featureless areas. Monocular SLAM is especially vulnerable to failure arising from sudden sharp movements as it must compare each successive frame [34]. Work has been done to address such perception constraints as shown in [47]. Still, further research in considering SLAM failure resulting from sensor-specific issues must be conducted.

VII. Conclusion

In this work, active SLAM, an essential framework for autonomous operations, is explained. Active SLAM was described as planning that takes into consideration perception performance. In this report, we explained a basic general framework of active SLAM, several state-of-the-art works, interesting findings, and finally, proposed important future research directions.

The problem of active SLAM was divided into two key tasks: exploration and navigation. While active SLAM methods may differ significantly in implementation, all strive to autonomously decide between exploration and exploitation in the most effective manner.

Although active SLAM has shown positive results, several limitations remain to be solved before it starts to have real world impact. Arguably the most important issue, the high computational cost of active SLAM must be lowered.

Regardless of these limitations, current experimental results showcase the high potential of active SLAM in achieving autonomous navigation in unknown environments.

REFERENCES

[1] A. Parness et al. ”LEMUR 3: A limbed climbing robot for extreme terrain mobility in space,” IEEE International Conference on Robotics and Automation (ICRA), pp. 5467-5473, 2017.

[2] K. Zhang et al. ”Robot Navigation of Environments with Unknown Rough Terrain Using deep Reinforcement Learning,” IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR), pp. 1-7, 2018.

[3] D. Thakur et al. ”Autonomous Inspection of a Containment Vessel using a Micro Aerial Vehicle,” IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR), pp. 1-7, 2019.

[4] H. F. Durrant-Whyte and T. Bailey, “Simultaneous localisation and mapping (SLAM): Part I,” IEEE Robotics Automation Magazine, vol. 13, no. 2, pp. 99–110, 2006.

[5] T. Bailey and H. F. Durrant-Whyte, “Simultaneous localisation and mapping (SLAM): Part II,” IEEE Robotics Automation Magazine, vol. 13, no. 3, pp. 108–117, 2006.

[6] C. Cadena et al. ”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.

[7] C. Stachniss et al. ”Information gain-based exploration using Rao-Blackwellized particle filters,” Proceedings of the Robotics: Science Systems Conference, pp. 65-72, 2005.

[8] H. H. Gonzalez-Banos and J.-C. Latombe, ”Navigation strategies for exploring indoor environments,” International Journal of Robotics Research, pp. 829–848, 2002.

[9] R. Sim and N. Roy, ”Global A-optimal robot exploration in SLAM,” IEEE International Conference on Robotics and Automation (ICRA), pp. 661–666, 2005.

[10] F. Lu and E. Milios, “Globally consistent range scan alignment for environment mapping,” Autonomous Robots, vol. 4, no. 4, pp. 333–349, 1997.

[11] G. Dissanayake et al. “A review of recent developments in simultaneous localization and mapping,” International Conference on Industrial and Information Systems, pp. 477–482, 2011.

[12] Z. Wang, et al. ”Simultaneous Localization and Mapping: Exactly Sparse Information Filters,” World Scientific Publishing Company, 2011.

[13] S. Thrun et al. ”Probabilistic Robotics,” MIT Press, 2006.

[14] Y. Latif et al. “Robust loop closing over time for pose graph slam,” International Journal of Robotics Research, vol. 32, no. 14, pp. 1611–1626, 2013.

[15] M. J. Milford and G. F. Wyeth, “SeqSLAM: Visual route-based navigation for sunny summer days and stormy winter nights,” IEEE International Conference on Robotics and Automation (ICRA), pp. 1643–1649, 2012.

[16] L. E. Kavraki, P. Svestka, J. -. 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.

[17] S. M. LaValle, ”Rapidly-exploring random trees: A new tool for path planning,” Technical Report. Computer Science Department, Iowa State University, 1998.

[18] S. Karaman and E. Frazzoli, ”Sampling-based Algorithms for Optimal Motion Planning,” arXiv: 1105.1186, 2011.

[19] W. H. Kwon, K. Bruckstein, ”Stabilizing state feedback design via the moving horizon method,” International Journal of Control, vol. 37, no. 3, pp. 631–643, 1983.

[20] Mnih, Volodymyr, et al. ”Human-level control through deep reinforcement learning,” Nature, vol. 518.7540, pp. 529-542, 2015.

[21] R. Sim and N. Roy, ”Global A-Optimal Robot Exploration in SLAM,” IEEE International Conference on Robotics and Automation (ICRA), pp. 661-666, 2005.

[22] J. Vallvé and J. Andrade-Cetto, ”Active pose SLAM with RRT*,” IEEE International Conference on Robotics and Automation (ICRA), pp. 2167-2173, 2015.

[23] H. Carrillo et al. ”Autonomous robotic exploration using a utility function based on Rényi’s general theory of entropy,” Autonomous Robots, vol. 42, no. 2, pp. 235-256, 2018

[24] Lehner, Hannah, et al. ”Exploration with active loop closing: A trade-off between exploration efficiency and map quality,” IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 6191-6198, 2017.

[25] J. Heejin et al. ”Learning Q-network for Active Information Acquisition,” IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2019.

[26] Zhang, Zichao and Davide Scaramuzza. “Perception-aware Receding Horizon Navigation for MAVs,” IEEE International Conference on Robotics and Automation (ICRA), pp. 2534-2541, 2018.

[27] A. Kim and R. M. Eustice, ”Perception-driven navigation: Active visual SLAM for robotic area coverage,” IEEE International Conference on Robotics and Automation (ICRA), pp. 3196-3203, 2013.

[28] Y. Chen, S. Huang, R. Fitch and J. Yu, ”Efficient Active SLAM Based on Submap Joining, Graph Topology and Convex Optimization,” IEEE International Conference on Robotics and Automation (ICRA) pp. 5159-5166, 2018.

[29] L. Fermin-Leon et al., ”Path planning in graph SLAM using Expected uncertainty,” IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 4594-4601, 2016.

[30] B. Yamauchi, “A frontier-based approach for autonomous exploration,” IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA), pp. 146–151, 1997.

[31] P. Quin et al. ”Expanding wavefront frontier detection: An approach for efficiently detecting frontier cells,” Australasian Conference on Robotics and Automation, 2014.

[32] J. Oršulić et al. ”Efficient Dense Frontier Detection for 2-D Graph SLAM Based on Occupancy Grid Submaps,” IEEE Robotics and Automation Letters, vol. 4, no. 4, pp. 3569-3576, 2019.

[33] X. Deng et al., ”Feature-constrained Active Visual SLAM for Mobile Robot Navigation,” IEEE International Conference on Robotics and Automation (ICRA), pp. 7233-7238, 2018.

[34] V. Prasad et al. ”SLAM-Safe Planner: Preventing Monocular SLAM Failure using Reinforcement Learning,” CoRR 2016

[35] V. Prasad et al. “Data Driven Strategies for Active Monocular SLAM using Inverse Reinforcement Learning.” AAMAS, 2017.

[36] C. Leung, S. Huang and G. Dissanayake, ”Active SLAM using Model Predictive Control and Attractor based Exploration,” IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 5026-5031, 2006.

[37] C. Leung et al. ”Planning under uncertainty using model predictive control for information gathering,” Robotics and Autonomous Systems, vol. 54, pp. 898-910, 2006.

[38] C. Leung, Shoudong Huang and G. Dissanayake, ”Active SLAM in structured environments,” IEEE International Conference on Robotics and Automation, pp. 1898-1903, 2008.

[39] M. G. Jadidi, J. V. Miro, and G. Dissanayake, ”Sampling-based incremental information gathering with applications to robotic exploration and environmental monitoring,” Int. Journal of Robot. Res., vol. 38, no. 6, pp. 658-685, 2019.

[40] B. Mu et al. ”Information-based Active SLAM via topological feature graphs,” IEEE 55th Conference on Decision and Control (CDC), pp. 5583-5590, 2016.

[41] S. M. Chaves and R. M. Eustice, ”Efficient planning with the Bayes tree for active SLAM,” 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 4664-4671, 2016.

[42] Z. Meng et al. ”Intelligent robotic system for autonomous exploration and active SLAM in unknown environments,” IEEE/SICE International Symposium on System Integration (SII), pp. 651-656, 2017.

[43] S. M. Chaves, A. Kim and R. M. Eustice, ”Opportunistic sampling-based planning for active visual SLAM,” IEEE/RSJ International Conference on Intelligent Robots and Systems (ICRA), pp. 3073-3080, 2014.

[44] L. Carlone, et al. ”Active SLAM and Exploration with Particle Filters Using Kullback-Leibler Divergence,” Journal of Intelligent Robotic Systems, vol. 75, no. 2, pp. 291-311, 2014.

[45] J. Blanco, J. Fernandez-Madrigal, J. Gonzalez, "A Novel Measure of Uncertainty for Mobile Robot SLAM with Rao-Blackwellized Particle Filters," The International Journal of Robotics Research (IJRR) vol. 27, no. 1, pp. 73–89, 2008.

[46] V. Indelman, L. Carlone, and F. Dellaert, "Planning in the continuous domain: A generalized belief space approach for autonomous navigation in unknown environments," The International Journal of Robotics Research, vol. 34, no. 7, pp. 849–882, 2015.

[47] Z. Zhang and D. Scaramuzza, "Perception-aware Receding Horizon Navigation for MAVs," IEEE International Conference on Robotics and Automation (ICRA), pp. 2534-2541, 2018.

[48] A. Ivanov and M. Campbell, "An efficient robotic exploration planner with probabilistic guarantees," IEEE International Conference on Robotics and Automation (ICRA), pp. 4215-4221, 2016.

[49] A. Ivanov and M. Campbell, "Uncertainty Constrained Robotic Exploration: An Integrated Exploration Planner," IEEE Transactions on Control Systems Technology, vol. 27, no. 1, pp. 146-160, Jan. 2019.

[50] A. Ivanov, "Guaranteed Robotic Planning and Exploration in Unknown Environments," Ph.D. Thesis, Cornell University, 2019.

[51] A. Banino, et al. "Vector-based navigation using grid-like representations in artificial agents," Nature 557, pp. 429–433, 2018.

[52] R. T. Rodrigues, et al. "Low-level Active Visual Navigation: Increasing robustness of vision-based localization using potential fields," IEEE Robotics and Automation Letters 3.3, pp. 2079-2086, 2018.

[53] B. Li, et al. "Path Planning for Active V-Slam Based on Reinforcement Learning," Cognitive Systems and Signal Processing. Communications in Computer and Information Science (ICCSIP), vol. 1006, 2018.

[54] H. Carrillo et al., "On the Monotonicity of Optimality Criteria during Exploration in Active SLAM", IEEE International Conference on Robotics and Automation (ICRA), pp. 1476-1483, 2015.

[55] M. L. Rodríguez-Arévalo, J. Neira and J. A. Castellanos, "On the Importance of Uncertainty Representation in Active SLAM," in IEEE Transactions on Robotics, vol. 34, no. 3, pp. 829-834, 2018.