Article

Active SLAM: A Review On Last Decade

Muhammad Farhan Ahmed [1,‡,*], Khayyam Masood [‡] and Vincent Fremont [1]

[1] LS2N, CNRS, Ecole Centrale de Nantes, 1 Rue de la Noë, 44300 Nantes, France; Muhammad.Ahmed@ec-nantes.fr (M.F.A.); vincent.fremont@ec-nantes.fr (V.F.)
* Correspondence: Muhammad.Ahmed@ec-nantes.fr; Tel.: +33-679093615 (M.F.A.)
‡ These authors contributed equally to this work.

Abstract: This article presents a novel review of Active SLAM (A-SLAM) research conducted in the last decade. We discuss the formulation, application, and methodology applied in A-SLAM for trajectory generation and control action selection using information theory based approaches. Our extensive qualitative and quantitative analysis highlights the approaches, scenarios, configurations, types of robots, sensor types, dataset usage, and path planning approaches of A-SLAM research. We conclude by presenting the limitations and proposing future research possibilities. We believe that this survey will be helpful to researchers in understanding the various methods and techniques applied to A-SLAM formulation.

Keywords: SLAM; Active SLAM; Localization; Mapping; Control Theory

Citation: Lastname, F.; Lastname, F.; Lastname, F. Title. *Journal Not Specified* **2022**, *1*, 0. https://doi.org/

Received:
Accepted:
Published:

1. Introduction

Simultaneous localization and mapping (SLAM) is a set of approaches in which a robot autonomously localizes itself and simultaneously maps the environment while navigating through it. It can be subdivided into solving localization and mapping. Localization is a problem of estimating the pose of the robot concerning the map, while mapping makes up the reconstruction of the environment with the help of visual, visual-inertial, and laser sensors on the vehicle. The front-end handles perception tasks, which involve implementing methods in signal processing and computer vision domains to compute estimated relative local poses between the robot environment and the observed features. SLAM back-end uses optimization theory, graph theory, and probability theory to estimate a global map and trajectory. For a detailed review of SLAM methods, we can refer to [1], [2], [3], [5].

Mostly SLAM algorithms are passive where the robot is controlled manually or goes towards pre-defined way-points and the navigation or path planning algorithm does not actively take part in robot motion or trajectory. A-SLAM however tries to solve the optimal exploration problem of the unknown environment by proposing a navigation strategy that generates future goal/target positions actions which decrease in the map and pose uncertainty, thus enabling a fully autonomous navigation and mapping SLAM system. We will look for further insight into A-SLAM in its designated section 2. In Active Collaborative SLAM (AC-SLAM), multiple robots collaborate actively while performing SLAM. The application areas of A-SLAM and AC-SLAM include search and rescue [17], planetary observations [12], precision agriculture [23], autonomous navigation in crowded environments [34], underwater exploration [30] [40] [44], artificial intelligence [42], assistive robotics [74], and autonomous exploration [126].

The first implementation for an algorithm on A-SLAM can be traced back to [9], but the initial name was drafted in [10]. However, A-SLAM and its roots can be further traced back to the nineteen eighties from ideas coined by artificial intelligence and robotic exploration techniques [11]. Reviews on A-SLAM have not been presented in the last decade. Only one article from 2016 discusses A-SLAM in their review article [4]. Within this article itself, A-SLAM is not the highlight of the research, instead, the authors look at the whole topic of SLAM in its totality. The reason for mentioning it here is the fact that the research conducted in our article is focused on active and A-SLAM.

The advantage of our work includes discussing not only the internal A-SLAM components but also its application domains, limitations, and future prospects, as well as providing a comprehensive statistical review of A-SLAM from the last decade. We start with Section 2, which provides an introduction to A-SLAM. From subsection 2.0.1 to subsection 2.0.2 we discuss the A-SLAM formulation, its main ingredients and how they are connected and related to one another. From subsection 2.1 to subsection 2.3 we discuss the various techniques, application domains, and qualitative analysis results. Subsection 2.4 presents our statistical analysis on robot and sensor type usage, real robot usage, result types (simulation and analytical), SLAM method adopted, path planning approach used, dataset usage, loop closure applicability, and ROS usage. From subsection 2.5 to subsection 2.9 we present the AC-SLAM problem introduction, application domains, and qualitative and quantitative results which quantify collaboration architecture, collaboration parameters, and environment usage apart from other parameters. Section 3 to subsection 3.2 presents the limitations and future prospects for A-SLAM research. Finally we conclude by summarising this article in section 4.

2. Introduction to Active SLAM (A-SLAM)

As described earlier, SLAM is a process in which a robot maps its environment and localizes itself to it. A-SLAM deals with designing robot trajectories to minimize the uncertainty in its map representation and localization. The aim is to perform autonomous navigation and exploration of the environment without an external controller or human effort. Referring to Figure 1 we observe that in a typical SLAM system, data from the sensors typically Lidars, cameras, and IMUs is processed by the front-end module which computes feature extraction, data association, feature classification, Iterative Closest Point (ICP), and loop closure. ICP is an iterative approach which computes the transformation that optimizes/aligns the data-points/features and is used in scan-matching methods for mapping the environment. The back-end module is responsible for high computational tasks involving Bundle Adjustment (B.A), pose graph optimization, and map estimation. The back-end module outputs the global map and pose estimate of the robot. A-SLAM can be referred to as an additional module or super set of SLAM systems that incorporates waypoints and trajectory planning, and control modules using information theory, control theory, and Reinforcement Learning (RL) methods to autonomously guide the robot toward its goal. We will discuss these components in section 2.0.2. We further present the utilization of these methods in section 2.1 and 2.2.

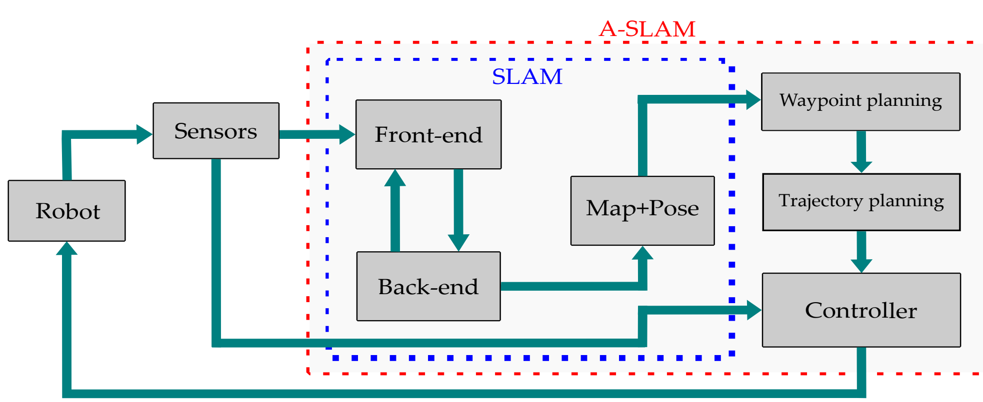


Figure 1. Architecture of SLAM and A-SLAM

In SLAM, environment exploration (to get a better knowledge of the environment) and exploitation (to revisit already traversed areas for loop closure) are maximized for better map estimation and localization. As a consequence, we have to perform a trade-off between exploration and exploitation as the prior requires maximum coverage of the environment, and the latter requires the robot to revisit previously explored areas. These two tasks may not always be applied simultaneously for a robot to perform autonomous navigation. The robot might have to solve the exploration-exploitation dilemma by switching between these two tasks.

In the following sections from 2.0.1 to section 2.0.2 we provide the basic A-SLAM formulation, its main components along with their brief definitions and functions in the A-SLAM pipeline.

2.0.1. A-SLAM formulation

A-SLAM is formulated in a scenario where the robot has to navigate in a partially observable/unknown environment by selecting a series of future actions in the presence of noisy sensor measurements that reduce its state and map uncertainties with respect to the environment. Such a scenario can be modeled as an instance of the Partially Observable Markov Decision Process (POMDP) as discussed in [6]. POMDP is defined as a 7 tuple $(X, A, O, T, \rho_o, \beta, \gamma)$, where $X \in \mathbb{R}$ represents the robots state space and is represented as the current state $x \in X$ and next state $x^{'} \in X$, $A \in \mathbb{R}$ is the action space and can be expressed as $a \in A$, O are the observations where $o \in O$, T is the state transition function between an action (a), present state (x) and next state ($x^{'}$), T accounts for robot control uncertainty in reaching the new state $x^{'}$, ρ_o accounts for sensing uncertainty, β is the reward associated with the action taken in state x, $\gamma \in (0, 1)$ takes into account the discount factor ensuring a finite reward even if the planning task has an infinite horizon. Both T and ρ_o can be expressed using conditional probabilities as Equation 1 and 2.

$$T(x, a, x^{'}) = p(x^{'} \mid x, a) \tag{1}$$

$$\rho_o(x, a, o) = p(o \mid x^{'}, a) \tag{2}$$

We can consider a scenario where the robot is in state x and takes an action a to move to $x^{'}$. This action uncertainty is modeled by T with an associated reward modeled by β, then it takes an observation o from its sensors that may not be precise in their measurements, and this sensor uncertainly is modeled by ρ_o. The robot's goal is to choose the optimal policy α^* that maximizes the associated expected reward ($\mathbb{E}$) for each state-action pair and it can be modeled as Equation 3:

$$\alpha^* = \underset{t}{\operatorname{argmax}} \sum_{t=0}^{\infty} \mathbb{E}\gamma^t \beta(x_t, a_t) \tag{3}$$

Where s_t, a_t and γ^t are the state, action, and discount factor evolution at time t. Although the POMDP formulation of A-SLAM is the most widely used approach, it is considered computationally expensive as it considers planning and decision-making under uncertainty. For computational convenience, A-SLAM formulation is divided into three main sub-modules which identify the potential goal positions/waypoints, compute the cost to reach them, and then select actions based on utility criterion which decreases map uncertainty and increases the robot's localization. We will discuss these sub-modules briefly in section 2.0.2

2.0.2. A-SLAM components

To deal with the computational complexity of A-SLAM, it is divided into three main sub-modules as depicted in Figure 2. The robot initially identifies potential goal positions to explore or exploit in its current estimate of the map. The map represents the environment perceived by the robot using its onboard sensors and may be classified as 1) topological Maps: which use a graphical representation of the environment and provide a simplified topological representation 2) metric maps: which provide environment information in the form of a sparse set of information points (landmarks) or full 3D representation of the environment (point cloud) 3) semantic maps: provide only segmented information about environment objects (like static obstacles) to the robot. Interested readers are directed to [1,4] for detailed discussions on mapping approaches. Once the robot has a map of its environment using any of the above approaches, it searches for potential target/goal locations to explore. One of the most widely used method is frontier based exploration initially used by [7], where the frontier is the border between known and unknown map locations. Using frontier based exploration has the advantage that all the environment is covered, but the disadvantage is that no exploitation task (revisiting already visited areas for loop closure) is performed which affects the robot's map estimate, we will discuss the application of this approach in section 2.1.2. An alternative approach recently used is deep reinforcement learning that will be discussed in 2.2.

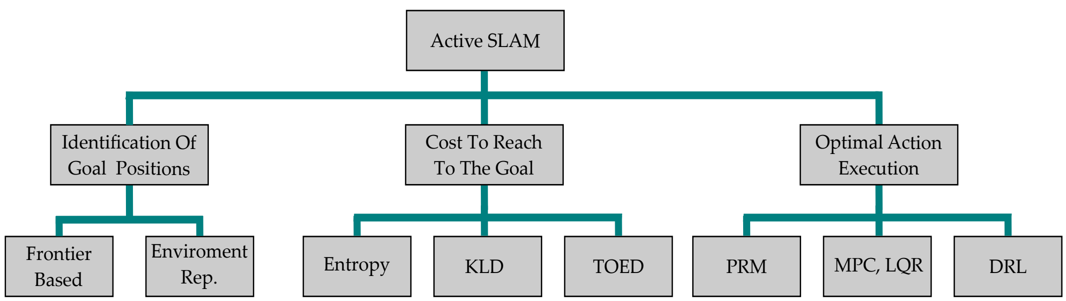


Figure 2. A-SLAM sub-modules

Once the goal position is identified, the next step is to compute the cost or utility function to that position based on some reward value of the optimal action selected from a set of all possible actions according to Equation 3. Ideally, this utility function should consider full joint probability distribution of map and robot poses, but this method is computationally expensive. Since we have a probabilistic estimation of both the robot and map, we can treat them as random variables with associated uncertainty in their estimation. The two most common approaches used in the quantification and representation of this uncertainty are Information Theory (IT), initially coined by Shannon in 1949, and Theory of Optimal Experimental Design (TOED) [64]. In IT, entropy measures the amount of uncertainty associated with a random variable or random quantity. Higher entropy leads to less information gain and vice versa. It is defined for a random variable X as $\mathcal{H}(X)$ and is shown in Equation 4. The objective is to reduce the entropy between the robot pose and map estimation as elaborated by the authors in [8].

$$\mathcal{H}(X) = -\sum_{x \in X} p(x) \log_2 p(x) \tag{4}$$

The relative entropy can also be used as a utility function that measures the form of probability distribution along with its deviation from its mean. This relative entropy is measured as Kullback-Leibler Divergence (KLD). KLD for two discrete distributions A and B on probability space X can be defined as Equation 5

$$\mathcal{D}_{KL}(A \mid B) = \sum_{x \in X} A(x) \log \frac{A(x)}{B(x)} \tag{5}$$

In A-SLAM, if we consider information driven utility functions, then entropy or KLD can be used as a metric to target binary probabilities in the grid map (occupancy grid map). Alternatively, if we consider task driven utility functions and assume a Gaussian distribution, then we can try to quantify the uncertainty in task space using TOED. In TOED, the priority of a set of actions for A-SLAM is based on the amount of covariance in the joint posterior. Less covariance contributes to a higher weight of the action set. To compare matrices for candidate action sets, different functions known as "optimality criterion" have been defined for a covariance matrix $\Sigma \in \mathbb{R}^{nxn}$ and having eigenvalues ζ_n, as 1) A-optimality, which deals with the minimization of average variance as shown in Equation 6, 2) D-optimality, deals with capturing the full covariance matrix and is defined in Equation 7, 3) E-optimality, intends to minimize the maximum eigenvalue and is expressed in Equation 8.

$$trace(\Sigma) = \sum_{i=1}^{n} \zeta_i \tag{6}$$

$$det(\Sigma) = \Pi_{i=1}^{n} \zeta_i \tag{7}$$

$$max_{1 \leq i \leq n}(\zeta_i) \tag{8}$$

TOED approaches require both the robot pose and map uncertainties to be represented as covariance matrix and may be computationally expensive, especially in landmark based SLAM where its size increases as new landmarks are discovered. Hence, IT based approaches are preferred over TOED. We will discuss the application of these approaches in section 2.1.

Once the goal positions and utility/cost to reach these positions have been identified the next step is to execute the optimal action which eventually moves/guides the robot to the goal position. Three approaches are commonly deployed:

1. Probabilistic Road Map (PRM) approaches represent a network graph representing the possible paths for the robot to select to reach the goal position. These approaches work in a heuristic manner and may not give the optimal path, additionally, the robot model is not incorporated in the planning phase which may result in unexpected movements. RRT [51], D* [50] and A* [57] are the widely used PRM methods. We identify these methods as geometry based methods and from section 2.1 to section 2.3, we discuss the application of these methods.
2. Optimal control methods like Liner Quadratic Regulator (LQR) and Model Predictive Control (MPC) are used for planning the control paths online or offline while taking into account the costs related to control effort and robot state evolution over time.
3. Deep Reinforcement Learning (DRL) methods work by maximizing the reward attained for each state-action pair during evolution of robot trajectory. We further discuss the application of these methods along with 2 in dynamic based methods in section 2.2.

The choice of selecting a suitable way-point candidate is weighted using IT and TOED as discussed in previous sections. In these methods, information gain or entropy minimization between the map and robot path guides the decision for the selection of these future way-point candidates. To generate a trajectory or a set of actions for these future way-point candidates, two main methods are adopted, namely geometric and dynamic approaches, respectively. These methods involve the usage of traditional path planners along with the Markov decision process and non-linear optimal control techniques. From section 2.1 to section 2.3, we will discuss these two methods and their utilization in research articles that are a part of this survey.

2.1. Geometry based methods

These methods describe A-SLAM as a task for the robot of choosing the optimal path and trajectory while reducing its poses and mapping uncertainly for efficient SLAM to autonomously navigate an unknown environment. The exploration space is discretized with finite random waypoints and Frontier based exploration along with traditional path planners like RRT*, D*, A* is deployed with IT and TOED based approaches including entropy, information gain, uncertainty metrics reduction. We can further classify the application of these methods as follows in the subsections below:

2.1.1. Information theory based approaches

The authors of [20] address the joint entropy minimization exploration problem and propose two modified versions of RRT*[51] called dRRT* and eRRT* respectively. dRRT* uses distance, while eRRT* used entropy change per distance traveled as the cost function. It is further debated that map entropy has a strong relationship with coverage, and path entropy has a relationship with map quality (as better localization produces a better map). Hence, actions are computed in terms of the joint entropy change per distance traveled. The simulation results proved that a combination of both of these approaches provides the best path planning strategy. An interesting comparison between information theory approaches is given in [18] where particle filters are used as the back-end of A-SLAM with frontier-based exploration (frontier is a boundary between visited and unexplored areas) [49] is deployed to select future candidate target positions. A comparison of these three methods used for solving the exploration problem and evaluating the information is discussed in the relevant subsections below:

1. **Joint entropy**: The information gained at the target is evaluated using the entropy of both the robot trajectory and map carried by each particle weighted by each trajectory's importance weight. The best exploration target is selected, which maximizes the joint entropy reduction and hence corresponds to higher information gain.
2. **Expected Map Mean**: An expected mean can be defined as the mathematical expectation of the map hypotheses of a particle set. The expected map mean can apply to detect already traversed loops on the map. Since the computation of the gain is developing, the complexity of this method increases.
3. **Expected information from a policy**: Kullback-Leiber Divergence [28] is used to drive an upper bound on the divergence between the true posterior and the approximated pose belief. Apart from the information consistency of the particle filter, this method also considers the information loss due to inconsistent mapping.

It was concluded using simulation results on various data sets referring to table 2 that most of these approaches were not able to properly address the probabilistic aspects of the problem and are most likely to fail because of high computational cost, map grid resolution dependency on performance.

The authors in [32] propose TFG SLAM, which uses geometry representation of space i,e the exploration space is represented by primitive geometric shapes, and an entropy reduction over map features is computed. It uses an entropy metric based on Laplacian approximation and computes a unified quantification of exploration and exploitation gains. An efficient sampling-based path planner is used based on a probabilistic road-map approach having a cost function that reduces control cost (distance) and collision penalty between targets. The simulation results compared to traditional grid map frontier exploration show a significant reduction in position, orientation, and exploration errors. Future improvements include expanding to an active Visual SLAM framework.

When considering Topo-matric graphs and a less computationally expensive solution, we can refer to the approach adopted by [45] which considers a scenario where we have many prior Topo-matric subgraphs and the robot does not know its initial position. A novel open-source framework is proposed which uses active localization and active mapping. A sub-map joining approach is defined, which switches between active localization and mapping. Active localization uses the maximum likelihood estimation to compute a motion policy, which reduces the computational complexity of this method.

2.1.2. Frontier based exploration

Frontiers are boundaries between explored and unexplored space. Formally, we can describe frontiers as a set of unknown points that each has at least one known-space neighbor. The work presented by [21] formulates a hybrid control switching exploration method with particle filter SLAM as the back end. It uses a frontier-based exploration method with A* [57] as a global planner and the Dynamic Window Approach (DWA) reactive algorithm as a local planner. Within the occupancy grid map, each frontier is segmented, a trajectory is planned for each segment and the trajectory with the highest map segments covariance is selected from the global cost map. The work presented in [34] deals with dynamic environments with multiple ground robots and uses frontier exploration for autonomous exploration with graph-based SLAM (iSAM) [61] optimization as the SLAM back-end. Dijkstra's algorithm-based local planner is used. Finally, a utility function based on Shannon and Renyi entropy is used for the computation of the utility of paths. Future work proposes to integrate a camera and use image feature scan matching for obstacle avoidance.

2.1.3. Path planning optimization

The method proposed by [30] exploits the relationship between the graphical model and sparse matrix factorization of graphical SLAM. It proposes the ordering of variables and a sub-tree catching scheme to facilitate the fast computation of optimized candidate paths weighted by the belief changes between them. The horizon selection criteria is based on the author's previous work utilizing an Extended information filter (EIF) and Gaussian Newton (GN) prediction. The proposed solution is implemented in Hovering Autonomous Underwater Vehicle (HAUV) with pose graph SLAM. The work presented in [44] deals with a similar volumetric exploration in an underwater environment with a multi-beam sonar. For efficient path planning, the revisit actions are selected depending on the pose uncertainty and sensor information gain.

The authors in [39] used an interesting approach that addresses the path-planning task as D* [50] with negative edge weights to compute the shortest path in case of a change of localization. This exploration method is highly effective in dynamic environments with changing obstacles and localization. When dealing with noisy sensor measurements, an interesting approach is adopted by [27] which proposes the Ambiguity-aware Robust A-SLAM (ARAS) that makes use of multi-hypothesis state and map estimates based on noisy or insufficient sensor information. This method uses the local contours for efficient multi-hypothesis path planning and incorporates loop closure.

2.1.4. Optimization in robot trajectory

The method proposed in [13] integrates A-SLAM with Ekman's exploration algorithm [53] to optimize the robot trajectory by leveraging only for global waypoints where loop closure appears and then the exploration canceling criterion is sent to SLAM back-end (based on ES-DSF information filter [52]). The exploration canceling criterion depends on the magnitude of information gain from the filter, loop closure detection, and the number of states without an update. If these criteria are met then the A-SLAM sends the exploration algorithm to stop and guides the robot to close the loop. We must note that in this approach A-SLAM is separated from the route planning and exploration process which is managed by the information filter.

In a similar approach presented by [24], the study assumes that some map prior information about the environment is available as a topological map. Then exploits this map information for active loop closure. The proposed method calculates an optimal global plan as a solution to the Chinese Postman Problem (CPP) [54] and an online algorithm that computes the Maximum Likelihood Estimate (MLE) by using non-linear optimization which computes the optimized graph with respect to the prior map and explored map. The D-optimality criterion is used to represent robot localization uncertainty. While the work presented by [12] incorporates active path planning with salient features (features with high entropy value) and ICP-based feature matching [56]. The triggering condition of A-SLAM is based on an active feature revisit and the path with the maximum utility score is chosen based on its length and map data.

2.1.5. Optimal policy selection

The definition and comparison presented in [35] formulate A-SLAM as a task of choosing a single or multiple policy type of robots trajectories which minimizes an objective function that comprises a reduction in expected costs in robot uncertainty, energy consumption, navigation time among other factors. An optimality criterion by definition quantifies the improvement of actions taken by the robot to improve localization accuracy and navigation time. A comparison between D-optimality (proportional to the determinant of the covariance matrix), A-optimality (proportional to the trace of the covariance matrix), and joint entropy is performed, and it is concluded D-optimality criterion is more appropriate for providing useful information about the robot's uncertainty contrary to A-optimality. The authors in [36] proved numerically that by using differential representations to propagate the spacial uncertainty, monotonicity is preserved for all the optimality criteria A-opt, D-opt, and E-opt (largest eigenvalue of the covariance matrix). In absolute representation using only unit quaternions, the monotonicity is preserved only in D-optimality and Shannon's entropy. In a similar comparison, the work presented in [37] concludes that A-Opt and E-opt criteria do not hold monotonicity in dead reckoning scenarios. It is proved using simulations with a differential drive robot that the D-opt criterion, under a linearized odometry method, holds monotonicity.

2.2. Dynamic based methods

Instead of using traditional path planners like A*, D*, and RRT, these methods formulate the A-SLAM a problem of selecting a series of control inputs to generate a collision-free trajectory and cover as much area as possible while minimizing the state estimation uncertainty and thus improve the localization and mapping of the environment. The planning and action spaces are now continuous (contrary to being discrete in geometry based methods) and locally optimal trajectories are computed. For the selection of optimal goal positions the similar approaches used in geometric based methods in section 2.1 are used with the exception that now the future candidate trajectories are computed using robot models, potential information fields, and control theory. Linear Quadratic Regulator (LQR), Model Predictive Control (MPC) [59], Markov decision process [60] or Reinforcement Learning (RL) [58] is used to choose the optimal future trajectories/set of trajectories via matrices that balance the need for exploring new areas and exploiting already visited areas for loop closure.

The method used by [26] uses reinforcement learning in the path planner to acquire a vehicle model by incorporating a 3D controller. The 3D controller can be simplified to one 2D controller for forward and backward motion and one 1D controller for path planning having an objective function that maximizes the map reliability and exploration zone. Therefore, the planner has an objective function that maximizes the accumulated reward for each state-action pair using the "learning from experience approach". It is shown through simulations that a nonholonomic vehicle learns the virtual wall-following behavior. A similar approach presented in [42] Uses fully convolutional residual networks to recognize the obstacles to getting a depth image. The path planning algorithm is based on a deep reinforcement learning algorithm (DRL).

An active localization solution where only the rotational movement of the robot is controlled in a position tracking problem is presented by [19]. Adaptive Monte Carlo Localization (AMCL) particle cloud is used as input and robot control commands are sent to its sensors as output. The proposed solution involves spectral clustering of the point cloud, building a compound map from each particle cluster, and selecting the most informative cell. The active localization is triggered when the robot has more than one cluster in its uncertainty estimate. The future improvements include more cells for efficient hypotheses estimation and integrating this approach into the SLAM front-end. In an interesting approach by [48] saccade movement of bionic eyes (rapid movement of the center of gaze within the visual field) is controlled. To leverage more features from the environment, an autonomous control strategy inspired by the human vision system is incorporated. The A-SLAM system involves two threads (parallel processes), a control thread, and a tracking thread. The control thread controls the bionic eyes movement to feature rich positions while the tracking thread tracks the eye motion by selecting the feature rich (ORB features) keyframes.

2.3. Geometry and dynamic based methods

These methods use geometry and dynamic-based methods mentioned in sections 2.1 and 2.2 incorporating frontier based exploration, information theory, and Model Predictive Control (MPC) for solving the A-SLAM problem.

The approach used by the authors in [15] presents an open-source multi-layer A-SLAM approach where the first layer selects the informative (utility criterion based on Shanon's Entropy [62]) goal locations (frontier points) and generates paths to these locations while the second and third layers actively re-plan the path based on the updated occupancy grid map. Non-linear MPC [63] is applied for local path execution with the objective function based on minimizing the distance to the target, controlling effort and cost of being to a nearby obstacle. One issue with this approach is that sometimes the robot stops and starts the re-planning phase of local paths. Future works include adding dynamic obstacles and the usage of aerial robots.

While an interesting approach mentioned in [23] and [29] presents a solution based on Model Predictive Control (MPC) to solve the area coverage and uncertainty reduction in A-SLAM. An MPC control switching mechanism is formulated and SLAM uncertainty reduction is treated as a graph topology problem and planned as a constrained nonlinear least-squares problem. Using convex relaxation, the SLAM uncertainty is reduced by a convex optimization method. The area coverage task is solved via the sequential quadratic programming method and Linear SLAM is used for sub-map joining.

2.4. Statistical analysis on A-SLAM

Table 1 summarizes the sensor types with descriptions used in A-SLAM. The SLAM method, path planning approaches, and publication years are also depicted. Within this table we can conclude that in most A-SLAM methods i) RGB and Lidar sensors are used as the main input data source for extracting the point cloud and image features/correspondences, ii) Pose Graph or Graph-based SLAM methods are involved, iii) Path planning algorithms based on graph search are used.

Table 2 elaborates on the robots and their respective descriptions used in A-SLAM. The datasets, ROS compatibility, and loop closure for A-SLAM are also presented. The information can be summarised as i) Ground Robots are widely used, ii) Loop closure is incorporated in 50% of implementations, and iii) ROS is used only 30% for most implementations.

Table 1. A-SLAM sensors, SLAM methods and path planning approaches.

Papers	Years	Sensors	SLAM Method	Path Planning
[12]	2017	Lidar	EKF SLAM	Active revisit Path Planning
[13]	2016	RGBD[1], Lidar[2], IMU[3],WE[16]	ES-DSF[17]	Ekman's Exploration algorithm
[14]	2018	Lidar, RGB	Hector SLAM	Artificial potential fields
[15]	2021	Lidar[4], RGBD[5]	RTAB-MAP	NMPC[18] , ACADO Framework, A*
[16]	2019	RGB	Vision Based SLAM	CAO[23]
[17]	2016	Lidar, RGB	EKF-SLAM	Maze solver Algorithm
[18]	2011	Lidar	Partical Filter SLAM	Joint Entropy, EMMI[19]
[19]	2021	Lidar	ACML	-
[20]	2015	Lidar	Pose SLAM	RRT*
[21]	2015	Lidar[6]	FastSLAM	A*, DWA[20]
[22]	2018	Lidar	Gmapping	Sequential Monte Carlo
[23]	2020	RGB	Linear SLAM	MPC
[24]	2019	Lidar	Graph SLAM	CPP[21] , OLSqO [22]
[25]	2016	Lidar, RGB	IEKF SLAM	Dijkstra, VSICP[24]
[26]	2011	IRS, Lidar[7], RGB	Matric Based Scan Matching SLAM	Reinforcement Learning
[27]	2020	Lidar, RGBD, IMU	MH-SLAM Based on iSAM2	Finding a straight line for each hypothesis
[28]	2014	Lidar	Partical Filter SLAM	Frontier based exploration
[29]	2018	ORS[30]	Linear SLAM	MPC and D-Optimality
[30]	2016	RGB	Pose graph SLAM based on GTSAM	Bayes treee , RRT*(T-RRT*)
[31]	2018	RGBD[8]	ORB-SLAM2	RRT* with MAP
[32]	2016	RGB	TFG SLAM	Probabilistic Road-map
[33]	2021	RGBD[9], Lidar[10]	Visual SLAM and Lidar SLAM	RRT* and fuzzy controller
[34]	2019	Lidar	Canonical Scan Matcher + iSAM2	Dijkstra, DOO[25]
[35]	2012	RBS[31]	EKF-SLAM	A*
[38]	2019	-	EKF Localization	A-optimally indicator
[39]	2018	Lidar 2D[11],3D[12]	Graph SLAM based ESDSF[26]	Modified D*
[40]	2019	MBS[32]	Pose-Graph SLAM	RRT*
[43]	2013	RGB	EKF SLAM	OCBN[27]
[41]	2015	Lidar, IMU	Sensor Based SLAM	-
[42]	2020	RGBD, Lidar[13]	FastSLAM	DRL[28]
[44]	2020	RGB, IMU	Pose Graph SLAM	RRT
[45]	2022	RGB	ORB-SLAM Based A-SLAM	RPP[29]
[47]	2021	Lidar, IMU	RIEKF SLAM based A-SLAM	-
[47]	2021	RGB	Object SLAM	-
[48]	2019	RGBD[14],[15], IMU	Active Visual SLAM	-

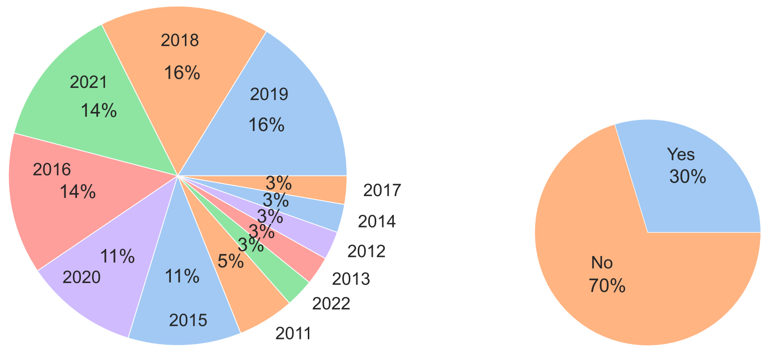
[1] Microsoft Kinect. [2] SICK LMS-100. [3] X-Sense MTI-G-700. [4] Hokuyo A2M8. [5] Intel RealSence D435i. [6] SLICK LMS 200. [7] Hokuyo URG-04LX. [8] Microsoft Kinect. [9] Intel RealSence D435 [10] RpLidar A2 [11] SICK LMS 100-10000. [12] Volodyne. [13] Rplidar A2. [14] Bionic Eyes. [15] Intel Realsence T265. [16] Wheel Encoders. [17] Exactly Sparse Delayed State Filter. [18] Non-Linear Model Predictive Control. [19] Expected Map Mean Information [20] Dynamic Window Approach [21] Chinese Postman Problem [22] Online Least squares optimization [23] Cognitive Based Adaptive Optimization [24] Visual Servoying using successive ICP [25] Dynamic Obstacle Avoidance [26] Extremely Sparse Delayed State Filter [27] Optimal Control Based navigation [28] Deep Reinforcement Learning [29] Rural Postman Problem [30] Omidirectional Range Sensor [31] Range Bearing sensor [32] Multi-beam Sonar

Table 2. A-SLAM robot types, Datasets, loop closure and ROS framework usage in A-SLAM

Papers	Robots	Data Set	Loop Closure	ROS
[13]	Clearpath Huskey A200	-	✓	-
[14]	Custom designed	-	-	✓
[15]	Festo Didactic's Robotino Omnidirectional robot	-	✓	✓
[16]	Survyer SVS	-	-	-
[17]	Khepra	-	-	-
[18]	-	ACES, Intel Research Labs, Friburg 079	-	-
[19]	Turtlebot 2	-	-	✓
[20]	-	Friburg 079	✓	
[21]	Pioneer 3-DX	-	-	✓
[23]	UAV	-	-	✓
[24]	-	MIT CSAIL, Intel Research Lab, AutoLab ROS	-	-
[25]	Non-halonomic AGV	-	-	-
[26]	Non-halonomic Ackermann mobile robot	-	-	-
[27]	-	-	✓	-
[28]	-	ACES, Intel Research Labs, Friburg 079	✓	-
[29]	-	-	✓	-
[30]	Hovering underwater vehicle (HAUV)	-	✓	-
[31]	Jackal Ground Robot by Clearpath Robotics	-	✓	✓
[32]	Turtlebot	-	-	✓
[33]	Custom designed	-	-	✓
[34]	Pioneer 3-DX, Pepper	-	✓	✓
[35]	-	DLR Dataset	-	-
[39]	Huskey	-	✓	✓
[40]	Girona 500 AUV(underwater robot)	-	-	-
[42]	TurtleBot 3	-	-	-
[44]	Bluefin HAUV Under water robot	-	✓	-
[45]	-	-	✓	-
[46]	-	-	✓	-
[47]	-	-	✓	-
[48]	Mobile robot with bionic eyes	-	✓	-

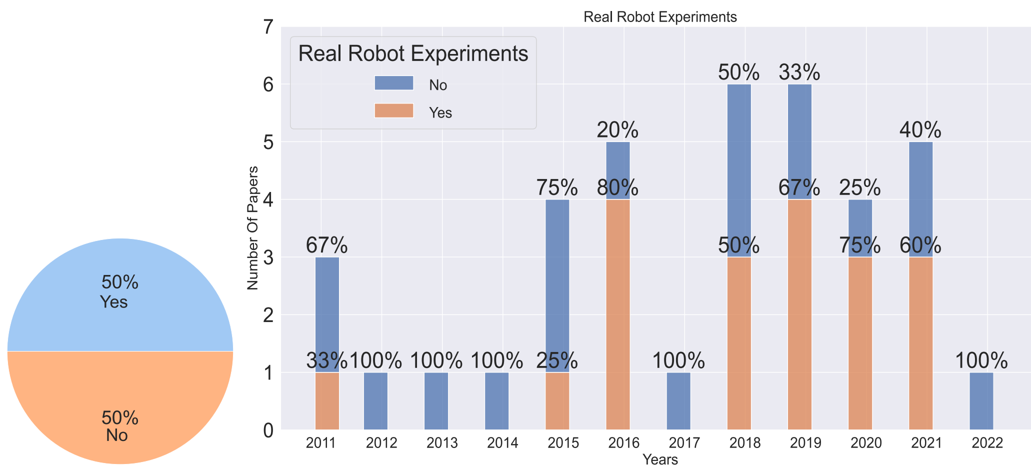
In Figure 3, the per annum selection of A-SLAM articles and the usage of ROS [128] is depicted. We can observe that almost 57% of A-SLAM articles are selected from the last four years. Although ROS is the popular environment for robots, it is deployed only in 30% of A-SLAM solutions.

From Figure 4, we can deduce that the use of real robots has increased since 2017 in experiments for A-SLAM. In Figure 5 and Figure 6 we can conclude that the use of simulation and analytical results has increased with each passing year.



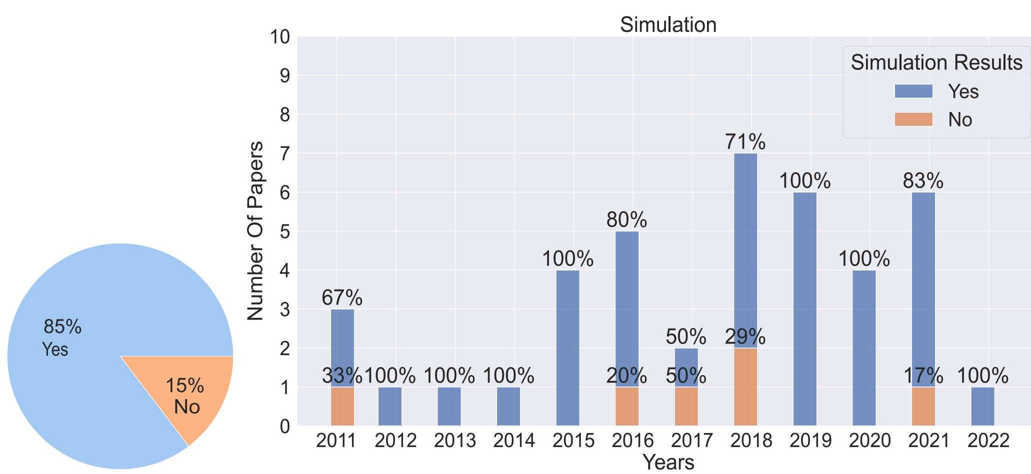
(a) Per annum article selection **(b)** ROS ussage

Figure 3. A-SLAM per annum article selection, Robot Operating System (ROS) applicability



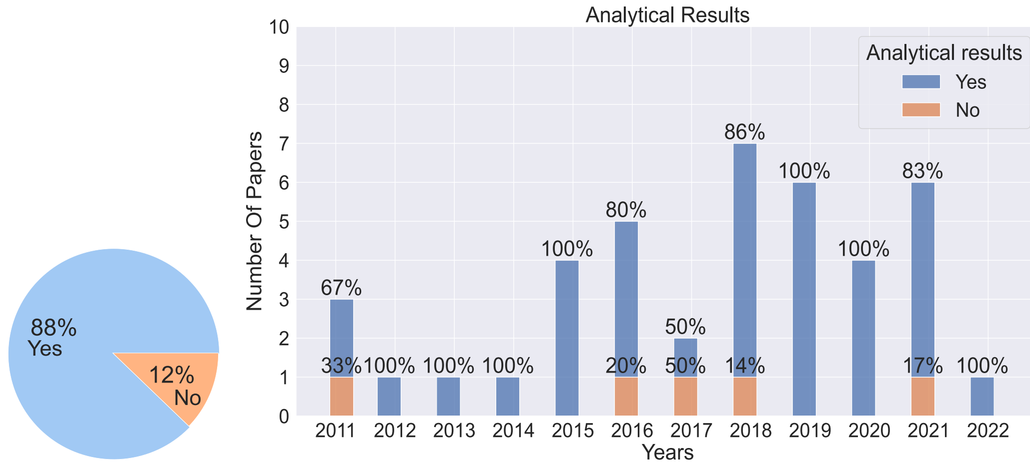
a % real robots used **b** Per annum distribution of real robot experiments

Figure 4. Real robots usage in A-SLAM and number of papers/articles



a % simulation results **b** Per annum distribution of simulation results

Figure 5. Simulation results used in A-SLAM and number of papers/articles



a % analytical results

b Per annum distribution of analytical results

Figure 6. Analytical results used in A-SLAM and number of papers/articles

2.5. *Active Collaborative SLAM (AC-SLAM)*

In AC-SLAM, multiple robots collaborate while actively performing SLAM. The data driven, information theory and control theory based approaches used A-SLAM mentioned from section 2.1 to 2.3 are also applicable in AC-SLAM with additional constraints of managing the communication and robust parameters exchange between robots. This collaboration may include exchanging parameters such as mapping info., localization info., and sensor data between a homogeneous or heterogeneous group of robots. In addition to these parameters, AC-SLAM parameters may include a) as presented by the authors in [66] and [85], incorporating the multirobot constraints induced by adding the future robot paths while minimizing the optimal control function (which takes into account the future steps and observations) and minimizing robot state and map uncertainty and adding them into the belief space (assumed to be Gaussian), b) parameters relating to exploration and relocalization (to gather at a predefined meeting position) phase of robots as described by [71], c) 3D Mapping info (OctoMap) used by authors in [80], d) path and map entropy info. as used in [81] and relative entropy, as mentioned in [82].

2.5.1. Network topology of AC-SLAM

Network topology describes how different robots communicate and exchange data with each other and with a central computer/server. This communication strategy may be centralized, decentralized, or hybrid. In a centralized communication network as presented by the authors in [65–67,82,83,85,86], a central server/agent is responsible for managing the communication and high level computational tasks while in a decentralized network as described in works of [70,71,74,75,78,81,84], each robot manages communication and computes its AC-SLAM parameters individually. Hybrid networks discussed in [72] use both centralized and decentralized approaches. We leave further discussion of network topology as it is not the main topic of this article, and focus on the scenarios and application domains in which AC-SLAM is applied. Typical application scenarios include collaborative task allocation, exploration and exploitation (revisiting already explored areas for loop closure), collaborative trajectory planning/trajectory optimization, and collaborative localization. In the following sections, we discuss the application of these application scenarios in selected literature.

2.6. Collaborative localization

In these methods, the robots switch their states (tasks) between self localization and other robot localization. The method proposed by the authors in [67] presents a novel centralized AC-SLAM method in which a deep reinforcement learning-based task allocation algorithm is used to assist agents in a relative observation task. Each agent can choose to perform it's independent ORB-SLAM [127] or localize other agents. The unique observation function is derived based on ORB-SLAM and consists of map points, keyframes, and loop closure detection components. The transition error between robots state is used to measure loss function. To learn the correspondence between the Q value and state-action pair, a novel Multi-Agent System Deep Q Network (MAS-DQN) is deployed. The large associated computational cost of this method lacks real-time application and thus a distributed learning approach is proposed in the future.

The method described in [85] defines a multi-robot belief space spanning problem as a robot collaboration problem to reduce uncertainty in state estimation. The robot belief is measured as the probability distribution of its state from the entire group and mapped environment. The proposed active localization method can guide each robot using Maximum A Posteriori (MAP) estimation of future way points to reduce its uncertainty by re-observing areas only observed by other robots in the framework of multi-robot AC-SLAM. In an interesting approach, the method presented in [83] uses multiple humanoid robot Multi-robot system (MRS) SLAM, where each robot has two working modes independent and collaborative. Each robot has two threads running simultaneously: a) the motion thread and b) the listening thread. During the motion thread, it will navigate the environment via the trajectory computed by the organizer (central server) using a D* path planner and a control strategy based on Reinforcement Learning (RL) and greedy algorithm. It also uploads its pose periodically to the organizer. During the listening thread, it will receive its updated pose from the organizer (via ORB-SLAM) and may receive the command to help other robots in the vicinity to improve their localization (chained localization).

2.7. Exploration and exploitation tasks

As mentioned earlier in A-SLAM we need to balance the need for exploration (maximizing the explored area) and exploitation (revisiting already explored areas for loop closure). In AC-SLAM, exploitation can also be achieved by moving the robot to another robot with less localization uncertainty. The authors in [79] describe a centralized AS-SLAM exploration problem (using frontier based exploration) as an efficiency optimization problem where the information gain and localization efficiency is maximized while navigation cost is penalized. For the relocalization (exploitation) phase, a function is derived in which each robot is guided towards a known landmark or another robot with less localization uncertainty. An adapted threshold criterion is defined which is adjusted by robots to escape the exploring and exploitation loop if they get stuck. To manage the limited communication bandwidth (because of a centralized architecture) a rendezvous method is proposed which relocates the robots to a predefined position if they get out of communication range. The future work proposed involves using distributed control schemes

The method described by [126] formulates the problem in topological geometrical space (the environment which is represented by primitive geometric shapes). Initially, the robots are assigned target positions and exploration is based on the frontier method and utilizing a switching cost function that takes into consideration the discovery of the target area of a robot by another member of the swarm. When the target is inside the robot's disjoint explored subspace, the cost function switches from frontier to a geodesic (distance) based navigation function.

2.8. Trajectory planning

In these methods, the path entropy is optimized to select the most informative path for AC-SLAM and to collectively plan trajectories that reduce the localization and map uncertainties. In the approach formulated in [81], the study presents a decentralized AC-SLAM method for a long planning horizon of actions for exploration and maintains estimation uncertainties at a certain threshold. The active path planner uses a modified version of RRT* in which a) the nonfusible nodes are filtered out because a non-holonomic robot is used and b) the action is chosen that best minimizes the entropy change per distance traveled. Entropy estimation is performed as a two-stage process. At first, entropy in short horizons is computed using Square-Root Information Filter (SRIF) updates and that of the short horizon is computed considering a reduction in loop closures in robot paths. The main advantage of this approach is that it maintains good pose estimation and encourages loop closure trajectories. An interesting solution is given by a similar approach to the method proposed by [82] using a relative entropy (RE) optimization method which integrates motion planning with robot localization and selects trajectories that minimize the localization error and associated uncertainty bound. A planning cost function is computed, which includes the uncertainty in the state (trace of the covariance matrix of the EKF state estimator) besides state and control cost. In an interesting and less computationally expensive approach, the authors in [73] use Support Vector Machine (SVM) Based corridor generation and Bezier curve-based continuous refinement used along with D-optimally criterion to collaboratively plan trajectories that reduce pose uncertainty in pose-graph-based SLAM. A bidding strategy is defined to reduce the pose uncertainty of the target robot, which selects the winning robot based on the least computational cost, feasible trajectory, and resource friendly criterion.

2.9. Statistical analysis on AC-SLAM

Table 3 summarizes the sensor types with descriptions used in active plus collaborative SLAM. The SLAM method and path planning approaches are also presented along with the year of publication. It can be concluded that most active and collaborative SLAM articles use i) RGB, Lidar, and IMU sensor data as input ii) Pose Graph and EKF SLAM methods are mostly used iii) Probabilistic road map-based approaches are used for path planning.

Table 4 elaborates analytical, simulation, and real robot experiments along with environment type, collaboration architecture, collaboration parameters, loop closure, and ROS framework. The information can be summarized as i) Most articles provide analytical and simulation-based results and multi-robot collaboration with up to four robots ii) Both centralized and decentralized collaboration architectures are used iii) Loop closure is encouraged while the use of ROS has been limited.

Table 3. AC-SLAM sensors, SLAM methods and path planning approaches .

Papers	Years	Sensors	SLAM Method	Path Planning
[66]	2018	Lidar, RGB	Pose Graph SLAM	Probabilistic Road Map
[67]	2020	RGBD[1], IMU	ORB-SLAM2	-
[71]	2011	Lidar, IMU	EKF-SLAM	A*, Exploration is stated as a constraint optimization problem
[72]	2019	Lidar(3D), RGBD[1],Magnetic compass, IMU	Vision Based SLAM	FSOTP[6], BIT*-H(motion planner)
[73]	2020	RGB, IMU	Active Pose Graph SLAM	RRT-Connect
[81]	2019	Lidar, IMU	Graph SLAM	-
[82]	2013	Lidar, IMU	EKF SLAM	-
[79]	2013	Lidar, RGB	EKF-SLAM	Frontier based
[80]	2017	Lidar(2D,3D),RGBD[3], RGB, IMU	Vision Based SLAM, Lidar Based SLAM	Octomap-Based navigation infrastructure
[83]	2018	RGB	ORBSLAM2	D*
[85]	2015	Lidar, IMU	Lidar SLAM	RRT*
[86]	2020	Lidar[4], RGB[5], IMU	Monte Carlo SLAM	-

[1] Microsoft Kinect. [3] Microsoft Kinect. [4] Garmin Lite V3. [5] See CAM-130 USB 3.1. [6] Fixed Start Open Travelling Salesman Problem.

Table 4. AC-SLAM results, robot types, collaboration architecture and parameters

Papers	Analytical Results	Sim. Results	Real Robots	Env.	MR[4]	Robot Types	Collab. Architecture	Collab. Parameters	Loop Closure	ROS
[66]	✓	✓	✗	🏠[1]	✓, Two	-	✤[2]	MI[4], Pose, VF[6]	✓	-
[67]	✓	✓	✗	🏠	✓, Four	-	◆[3]	MI, Pose, VF	✓	✓
[71]	✓	✓	✗	🏠	✓, Two	-	◆	MI, Pose	-	-
[72]	✓	✓	✓	🏠	✓, Two	UAV, UGV (Custom made)	◆	MI, VF	✓	✓
[73]	✓	✓	✓	🏠	✓, Two	UAV (Custom made)	✤	MI, VF	✓	-
[81]	✓	✓	✗	🏠	✓	-	◆	MI, Pose	-	-
[82]	✓	✓	✗	🏠	✓	-	✤	MI, Pose	✓	-
[79]	✓	✓	✗	🏠	✓	-	◆	MI, Pose	-	-
[80]	✓	✓	✓	🏠	✓, Two	UAV, UGV	◆	MI, Pose	✓	-
[83]	✓	✓	✗	🏠	✓	-	✤	MI, Pose	✓	✓
[85]	✓	✓	✗	🏠	✓	-	✤	MI, Pose	✓	-
[86]	✓	✓	✓	🏠	✓	Turtlebot 3	✤	MI, Pose	✓	✓

[1] Indoor Environment. [2] Centralised. [3] Decentralised. [4] Mapping Info. [5] Multi Robot. [6] Visual Features.

3. Discussion and Perspectives

We focused on A-SLAM and AC-SLAM methods, their implementation and methodology applications in selected research articles. Apart from the qualitative and quantitative analysis presented in previous sections, we would like to bring into the limelight, the limitations of A-SLAM problem and future research domains in the following sections.

3.1. *Limitations of existing methods*

We can classify the limitations of A-SLAM research into two categories described in sections 3.1.1 and 3.1.2.

3.1.1. General limitations

These limitations can be considered as open problems persisting in A-SLAM research and we can further explain them as:

- Stopping criteria: since A-SLAM is computationally expensive, we can debate on its stopping criteria as discussed by [4] i.e, the decision of when to stop the exploration task and switch to some other task such as revisiting already explored areas. The quantification of uncertainties from TOED may be used as an interesting stopping criterion, but still, this is an open research problem.
- Robust data associations: contrary to SLAM where an internal controller is responsible for robot action and the data association (association between measurements and corresponding landmarks) has less impact on robot actions, in A-SLAM a robust data association guides the controller to select feature rich positions. The qualification of these good feature/landmark positions may be difficult, especially in beyond line of sight measurements.
- Dynamic environments: contrary to SLAM, in A-SLAM the nature of the environment (static/dynamic), and the nature of the obstacles (static, dynamic) have a strong relationship with the utility function for computing future actions. Most of the A-SLAM literature deals with static environments and obstacles that may not apply to real world scenarios.
- Simulation environment: when considering DRL based approaches, the model training is constrained to a simulated environment, and contrary to deep learning approaches, an offline dataset cannot be used. The trained model may not perform optimally in real world scenarios with high uncertainty.

3.1.2. Limitations in selected literature

Some of the insights found in the surveyed articles include a) limited consideration of dynamic obstacles as only [39] and [34] consider them, b) computational complexity and real-time deployment as only [18] and [72] addresses it c) referring to subsection 2.4 and table 2 we can conclude that the ussage of loop closure task and ROS has been limited. d) lack of open-source implementation as only [45] and [31] provide open-source solutions, this requirement may be beneficial to researchers to reproduce the results, e) in AC-SLAM no article explicitly addresses the problem of managing robust inter-robot and robot-server communications while using minimum bandwidth, f) limited usage of dynamic approaches involving MPC and deep reinforcement learning-based navigation.

3.2. *Future Prospects*

1. Detection and avoidance of dynamic obstacles: for a robot to navigate autonomously in an unknown/partially known environment, it is necessary that it should be able to detect, localize and avoid dynamic obstacles. For A-SLAM, the static and dynamic obstacles avoidance mechanism is important because it represents the uncertainty propagation and hence affects the entropy of the system.

2. Lowering computational complexity for real-time applications: As discussed earlier the utility criterion in TOED and relative entropy computation are both computationally extensive tasks, thus limiting the real-time performance of A-SLAM. Formulating efficient uncertainly criteria promising real-time performances, is a challenging task.
3. Optimal control and DRL implementation: the application of optimal and robust control strategies helps to formulate the robot's action space in the continuous domain and may also provide an optimal solution. The usage of these control strategies is highly encouraged. DRL provides an alternate model free solution in which the decision making is embedded inside the network. Depp Q Networks (DQN) and double-dueling (D3QN) are applications of such DRL approaches used by [26] and [42].

4. Conclusion

This paper focused on two emerging techniques applied in simultaneous localization and mapping technology, i.e. A-SLAM and AC-SLAM. We surveyed papers published in the last decade and summed up their contributions. We started our work by recalling the A-SLAM problem, and its formal formulation, discussing sub-modules and presenting methods applied for the deployment of modern active and collaborative SLAM. We presented an extensive qualitative and quantitative analysis of surveyed research articles and presented the research domains and methodology. We finally highlighted the limitations of the present research and proposed some research axis that require attention.

Author Contributions: Conceptualization, M.F.A, K.M. and V.F.; methodology, M.F.A. and K.M.; validation, V.F.; formal analysis, K.M.; investigation, M.F.A.; resources, V.F.; data curation, M.F.A. and K.M.; writing—original draft preparation, M.F.A. and K.M.; writing—review and editing, K.M. and V.F.; visualization, M.F.A. and K.M.; supervision, V.F.; project administration, V.F.; and funding acquisition, V.F. All authors have read and agreed to the published version of the manuscript.

Funding: This research has been supported by the DIONISO project (progetto SCN_00320 - INVITALIA).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Acknowledgments: This work was carried out in the framework of the NExT Senior Talent Chair DeepCoSLAM, which were funded by the French Government, through the program Investments for the Future managed by the National Agency for Research ANR-16-IDEX-0007, and with the support of Région Pays de la Loire and Nantes Métropole.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Dhiman, N.K.; Deodhare, D.; Khemani, D. Where Am I? Creating Spatial Awareness in Unmanned Ground Robots Using SLAM: A Survey. Sadhana 2015, 40, 1385–1433, doi:10.1007/s12046-015-0402-6.
2. Saeedi, S.; Paull, L.; Trentini, M.; Li, H. Multiple Robot Simultaneous Localization and Mapping. In Proceedings of the 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems; IEEE: San Francisco, CA, September 2011; pp. 853–858.
3. Pancham, A., Tlale, N., & Bright, G. (2011). Literature review of SLAM and DATMO. 4th Robotics and Mechatronics Conference of South Africa (RobMech 2011), CSIR International Conference Centre, Pretoria.
4. Cadena, C.; Carlone, L.; Carrillo, H.; Latif, Y.; Scaramuzza, D.; Neira, J.; Reid, I.; Leonard, J.J. Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age. IEEE Trans. Robot. 2016, 32, 1309–1332, doi:10.1109/TRO.2016.2624754.
5. Zamora, E.; Yu, W. Recent Advances on Simultaneous Localization and Mapping for Mobile Robots. IETE Tech Rev 2013, 30, 490, doi:10.4103/0256-4602.125671.

6. Fox, D.; Burgard, W.; Thrun, S. Active Markov Localization for Mobile Robots. Robotics and Autonomous Systems 1998, 25, 195–207, doi:10.1016/S0921-8890(98)00049-9.
7. Yamauchi, B. A Frontier-Based Approach for Autonomous Exploration. In Proceedings of the Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA'97. "Towards New Computational Principles for Robotics and Automation"; IEEE Comput. Soc. Press: Monterey, CA, USA, 1997; pp. 146–151.
8. Stachniss, C.; Grisetti, G.; Burgard, W. Information Gain-Based Exploration Using Rao-Blackwellized Particle Filters. In Proceedings of the Robotics: Science and Systems I; Robotics: Science and Systems Foundation, June 8 2005.
9. H. J. S. Feder. (1999). Simultaneous stochastic mapping and localization. Ph.D. dissertation, Dept. Mech. Eng., MIT, Cambridge, MA, USA.
10. C. Leung, S. Huang, and G. Dissanayake. (2006). A-SLAM using model predictive control and attractor based exploration. in Proceedings of IEEE/RSJ International Conference of Intelligent Robots Systems, 2006, pp. 5026–5031.
11. A. G. Barto and R. S. Sutton. (1981). Goal seeking components for adaptive intelligence: An initial assessment. Air Force Wright Aeronautical Laboratories. Wright-Patterson Air Force Base, Dayton, OH, USA, Tech. Rep. AFWAL-TR-81-1070.
12. Li, G., Geng, Y., & Zhang, W. (2018). Autonomous Planetary Rover Navigation via A-SLAM. Aircraft Engineering and Aerospace Technology, 91(1), 60–68. https://doi.org/10.1108/aeat-12-2016-0239
13. Lenac, K.; Kitanov, A.; Maurović, I.; Dakulović, M.; Petrović, I. Fast Active SLAM for Accurate and Complete Coverage Mapping of Unknown Environments. In Intelligent Autonomous Systems 13; Menegatti, E., Michael, N., Berns, K., Yamaguchi, H., Eds.; Advances in Intelligent Systems and Computing; Springer International Publishing: Cham, 2016; Vol. 302, pp. 415–428 ISBN 978-3-319-08337-7.
14. Kim, P.; Chen, J.; Kim, J.; Cho, Y.K. SLAM-Driven Intelligent Autonomous Mobile Robot Navigation for Construction Applications. In Advanced Computing Strategies for Engineering; Smith, I.F.C., Domer, B., Eds.; Lecture Notes in Computer Science; Springer International Publishing: Cham, 2018; Vol. 10863, pp. 254–269 ISBN 978-3-319-91634-7.
15. Bonetto, E., Goldschmid, P., Pabst, M., Black, M. J., & Ahmad, A. (2022). iRotate: Active visual slam for Omnidirectional Robots. Robotics and Autonomous Systems, 104102. https://doi.org/10.1016/j.robot.2022.104102
16. Kalogeiton, V. S., Ioannidis, K., Sirakoulis, G. C., & Kosmatopoulos, E. B. (2019). Real-time A-SLAM and obstacle avoidance for an autonomous robot based on Stereo Vision. Cybernetics and Systems, 50(3), 239–260. https://doi.org/10.1080/01969722.2018.1541599
17. Nabil, M., Kassem, M. H., Bahnasy, A., Shehata, O. M., & Morgan, E.-S. I. (2016). Rescue missions bots using A-SLAM and map feature extraction. Proceedings of the 4th International Conference on Control, Mechatronics and Automation - ICCMA '16. https://doi.org/10.1145/3029610.3029624
18. Du, J., Carlone, L., Kaouk Ng, M., Bona, B., & Indri, M. (2011). A comparative study on A-SLAM and autonomous exploration with particle filters. 2011 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM). https://doi.org/10.1109/aim.2011.6027142
19. Andrade, F., LLofriu, M., Tanco, M. M., Barnech, G. T., & Tejera, G. (2021). Active localization for mobile service robots in symmetrical and Open Environments. 2021 Latin American Robotics Symposium (LARS), 2021 Brazilian Symposium on Robotics (SBR), and 2021 Workshop on Robotics in Education (WRE). https://doi.org/10.1109/lars/sbr/wre54079.2021.9605406
20. Vallve, J., & Andrade-Cetto, J. (2015). Active pose slam with RRT. 2015 IEEE International Conference on Robotics and Automation (ICRA). https://doi.org/10.1109/icra.2015.7139485
21. Trivun, D., Salaka, E., Osmankovic, D., Velagic, J., & Osmic, N. (2015). A-SLAM-based algorithm for autonomous exploration with Mobile Robot. 2015 IEEE International Conference on Industrial Technology (ICIT). https://doi.org/10.1109/icit.2015.7125079
22. Pang, L., Hu, J., Xiao, P., & Liu, S. (2018). A-SLAM based on geometry rules and forward simulation in Exploration Space. 2018 IEEE International Conference on Information and Automation (ICIA). https://doi.org/10.1109/icinfa.2018.8812347
23. Chen, Y., Huang, S., & Fitch, R. (2020). A-SLAM for mobile robots with area coverage and obstacle avoidance. IEEE/ASME Transactions on Mechatronics, 25(3), 1182–1192. https://doi.org/10.1109/tmech.2019.2963439
24. Soragna, A., Baldini, M., Joho, D., Kummerle, R., & Grisetti, G. (2019). A-SLAM using connectivity graphs as Priors. 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). https://doi.org/10.1109/iros40897.2019.8968613
25. An, S.-Y., Lee, L.-K., & Oh, S.-Y. (2015). Ceiling vision-based A-SLAM framework for dynamic and wide-open environments. Autonomous Robots, 40(2), 291–324. https://doi.org/10.1007/s10514-015-9453-0
26. Martinez-Marin, T., Lopez, E., & De Bernardis, C. (2011). An unified framework for A-SLAM and Online Optimal Motion Planning. 2011 IEEE Intelligent Vehicles Symposium (IV). https://doi.org/10.1109/ivs.2011.5940547
27. Hsiao, M., Mangelson, J. G., Suresh, S., Debrunner, C., & Kaess, M. (2020). ARAS: Ambiguity-aware robust A-SLAM based on multi-hypothesis state and map estimations. 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). https://doi.org/10.1109/iros45743.2020.9341384
28. Carlone, L., Du, J., Kaouk Ng, M., Bona, B., & Indri, M. (2013). A-SLAM and exploration with particle filters using Kullback-Leibler divergence. Journal of Intelligent & Robotic Systems, 75(2), 291–311. https://doi.org/10.1007/s10846-013-9981-9

29. Chen, Y., Huang, S., Fitch, R., & Yu, J. (2018). Efficient A-SLAM based on submap joining, graph topology and convex optimization. 2018 IEEE International Conference on Robotics and Automation (ICRA). https://doi.org/10.1109/icra.2018.8460864
30. Chaves, S. M., & Eustice, R. M. (2016). Efficient planning with the Bayes Tree for A-SLAM. 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). https://doi.org/10.1109/iros.2016.7759686
31. Deng, X., Zhang, Z., Sintov, A., Huang, J., & Bretl, T. (2018). Feature-constrained active visual slam for Mobile Robot Navigation. 2018 IEEE International Conference on Robotics and Automation (ICRA). https://doi.org/10.1109/icra.2018.8460721
32. Mu, B., Giamou, M., Paull, L., Agha-mohammadi, A.-akbar, Leonard, J., & How, J. (2016). Information-based A-SLAM via topological feature graphs. 2016 IEEE 55th Conference on Decision and Control (CDC). https://doi.org/10.1109/cdc.2016.7799127
33. Thoa Mac, T., Lin, C.-Y., Huan, N. G., Duc Nhat, L., Cong Hoang, P., & Hai, H. hong. (2021). Hybrid slam-based exploration of a mobile robot for 3D scenario reconstruction and autonomous navigation. Acta Polytechnica Hungarica, 18(6), 197–212. https://doi.org/10.12700/aph.18.6.2021.6.11
34. Mammolo, D. (2019). A-SLAM in Crowded Environments, master thesis, Autonomous Systems Lab, ETH Zurich, Zurich, 2019 https://doi.org/10.3929/ethz-b-000341706
35. Carrillo, H., Reid, I., & Castellanos, J. A. (2012). On the comparison of uncertainty criteria for A-SLAM. 2012 IEEE International Conference on Robotics and Automation. https://doi.org/10.1109/icra.2012.6224890
36. Rodriguez-Arevalo, M. L., Neira, J., & Castellanos, J. A. (2018). On the importance of uncertainty representation in A-SLAM. IEEE Transactions on Robotics, 34(3), 829–834. https://doi.org/10.1109/tro.2018.2808902
37. Carrillo, H., Latif, Y., Rodriguez-Arevalo, M. L., Neira, J., & Castellanos, J. A. (2015). On the Monotonicity of optimality criteria during exploration in A-SLAM. 2015 IEEE International Conference on Robotics and Automation (ICRA). https://doi.org/10.1109/icra.2015.7139384
38. Zhang, L., Zhang, Z., Siegwart, R., & Chung, J. J. (2019). Optimized motion strategy for active target localization of mobile robots with time-varying connectivity: Extended abstract. 2019 International Symposium on Multi-Robot and Multi-Agent Systems (MRS). https://doi.org/10.1109/mrs.2019.8901089
39. Maurovic, I., Seder, M., Lenac, K., & Petrovic, I. (2018). Path planning for A-SLAM based on the D* algorithm with negative edge weights. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 48(8), 1321–1331. https://doi.org/10.1109/tsmc.2017.2668603
40. Palomeras, N., Carreras, M., & Andrade-Cetto, J. (2019). A-SLAM for autonomous underwater exploration. Remote Sensing, 11(23), 2827. https://doi.org/10.3390/rs11232827
41. Lourenco, P., Batista, P., Oliveira, P., & Silvestre, C. (2015). Torwards uncertainty optimization in A-SLAM. 2015 54th IEEE Conference on Decision and Control (CDC). https://doi.org/10.1109/cdc.2015.7402706
42. Wen, S., Zhao, Y., Yuan, X., Wang, Z., Zhang, D., & Manfredi, L. (2020). Path planning for A-SLAM based on deep reinforcement learning under unknown environments. Intelligent Service Robotics, 13(2), 263–272. https://doi.org/10.1007/s11370-019-00310-w
43. Yang, Z. X., & Jin, S. (2013). UAV A-SLAM trajectory programming based on optimal control. Advanced Materials Research, 765-767, 1932–1935. https://doi.org/10.4028/www.scientific.net/amr.765-767.1932
44. Suresh, S., Sodhi, P., Mangelson, J. G., Wettergreen, D., & Kaess, M. (2020). A-SLAM using 3D SUBMAP saliency for underwater volumetric exploration. 2020 IEEE International Conference on Robotics and Automation (ICRA). https://doi.org/10.1109/icra40945.2020.9196939
45. Xue, W., Ying, R., Wen, F., Chen, Y., & Liu, P. (2022). A-SLAM with prior topo-metric graph starting at uncertain position. IEEE Robotics and Automation Letters, 7(2), 1134–1141. https://doi.org/10.1109/lra.2021.3133606
46. Xu, M., Song, Y., Chen, Y., Huang, S., & Hao, Q. (2021). Invariant EKF based 2D A-SLAM with Exploration Task. 2021 IEEE International Conference on Robotics and Automation (ICRA). https://doi.org/10.1109/icra48506.2021.9561951
47. Wu, Y., Zhang, Y., Zhu, D., Chen, X., Coleman, S., Sun, W., Hu, X., & Deng, Z. (2021). Object SLAM-based active mapping and robotic grasping. 2021 International Conference on 3D Vision (3DV). https://doi.org/10.1109/3dv53792.2021.00144
48. Liu, Y., Zhu, D., Peng, J., Wang, X., Wang, L., Chen, L., Li, J., & Zhang, X. (2019). Robust active visual slam system based on Bionic Eyes. 2019 IEEE International Conference on Cyborg and Bionic Systems (CBS). https://doi.org/10.1109/cbs46900.2019.9114468
49. B. Yamauchi, "A frontier-based approach for autonomous exploration," Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA'97. 'Towards New Computational Principles for Robotics and Automation', 1997, pp. 146-151, doi: 10.1109/CIRA.1997.613851.
50. A. (1994). The D* Algorithm for Real-Time Planning of Optimal Traverses
51. Naderi, Kourosh & Rajamäki, Joose & Hämäläinen, Perttu. (2015). RT-RRT*: a real-time path planning algorithm based on RRT*. 113-118. 10.1145/2822013.2822036.
52. R. M. Eustice, H. Singh and J. J. Leonard, "Exactly Sparse Delayed-State Filters for View-Based SLAM," in IEEE Transactions on Robotics, vol. 22, no. 6, pp. 1100-1114, Dec. 2006, doi: 10.1109/TRO.2006.886264.
53. A. Ekman, A. Torne and D. Stromberg, "Exploration of polygonal environments using range data," in IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), vol. 27, no. 2, pp. 250-255, April 1997, doi: 10.1109/3477.558809.
54. Sökmen, Özlem & Emeç, Şeyma & Yilmaz, Mustafa & Akkaya, Gokay. (2019). An Overview of Chinese Postman Problem. 3rd international conference on advanced engineering technologies 19-20 September 2019, Turkey

55. A. Renzaglia, L. Doitsidis, A. Martinelli and E. B. Kosmatopoulos, "Cognitive-based adaptive control for cooperative multi-robot coverage," 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2010, pp. 3314-3320, doi: 10.1109/IROS.2010.5649249.
56. He, Ying & Liang, Bin & Yang, Jun & Li, Shunzhi & He, Jin. (2017). An iterative closest points algorithm for registration of 3D laser scanner point clouds with geometric features. sensors. 17. 1862. 10.3390/s17081862.
57. Liu, Xiang & Gong, Daoxiong. (2011). A comparative study of A-star algorithms for search and rescue in perfect maze. 10.1109/ICEICE.2011.5777723.
58. W. Qiang and Z. Zhongli, "Reinforcement learning model, algorithms and its application," 2011 International conference on mechatronic science, Electric Engineering and Computer (MEC), 2011, pp. 1143-1146, doi: 10.1109/MEC.2011.6025669.
59. A. Bemporad, "Model predictive control design: new trends and tools," proceedings of the 45th IEEE conference on decision and control, 2006, pp. 6678-6683, doi: 10.1109/CDC.2006.377490.
60. Sudharman K. Jayaweera, "Markov decision processes," in signal processing for cognitive radios , Wiley, 2015, pp.207-268, doi: 10.1002/9781118824818.ch7.
61. M. Kaess, A. Ranganathan and F. Dellaert, "iSAM: Incremental smoothing and mapping," in IEEE Transactions on Robotics, vol. 24, no. 6, pp. 1365-1378, Dec. 2008, doi: 10.1109/TRO.2008.2006706.
62. Shannon, C. E. (1948). A mathematical theory of communication. The bell system technical journal, 27, 623–656.
63. Findeisen, Rolf & Allgöwer, Frank. (2002). An introduction to nonlinear model predictive control.
64. A. P$a^{'}$zman, Foundations of optimum experimantal design. Springer, 1996, vol. 14.
65. Yu, H., Li, H., & Yang, Z. (2019). Collaborative visual Slam framework for a multi-UAVs system based on mutually loop closing. Wireless and Satellite Systems, 653–664. https://doi.org/10.1007/978-3-030-19153-5_64
66. Indelman, V. (2018). Towards cooperative multi-robot belief space planning in unknown environments. Springer Proceedings in Advanced Robotics, 441–457. https://doi.org/10.1007/978-3-319-51532-8_27
67. Pei, Z., Piao, S., Quan, M., Qadir, M. Z., & Li, G. (2020). Active collaboration in relative observation for multi-agent visual simultaneous localization and mapping based on Deep Q Network. International Journal of Advanced Robotic Systems, 17(2), 172988142092021. https://doi.org/10.1177/1729881420920216
68. Patel, M., Bandopadhyay, A., & Ahmad, A. (2021). Collaborative mapping of archaeological sites using multiple UAVs. ArXiv. https://doi.org/10.48550/arXiv.2105.07644
69. Dong, S., Xu, K., Zhou, Q., Tagliasacchi, A., Xin, S., Nießner, M., & Chen, B. (2019). Multi-robot collaborative dense scene reconstruction. ACM Transactions on Graphics, 38(4), 1–16. https://doi.org/10.1145/3306346.3322942
70. Hilger, R. P. (2014). Acoustic communications considerations for collaborative simultaneous localization and mapping (thesis). California/ Naval Postgraduate School. http://hdl.handle.net/10945/44579
71. Pham, V. C., & Juang, J. C. (2011). An Improved Active SLAM Algorithm for Multi-robot Exploration . SICE Annual Conference 2011, 1660–1665.
72. Qin, H., Meng, Z., Meng, W., Chen, X., Sun, H., Lin, F., & Ang, M. H. (2019). Autonomous exploration and mapping system using heterogeneous UAVs and UGVs in GPS-denied environments. IEEE Transactions on Vehicular Technology, 68(2), 1339–1350. https://doi.org/10.1109/tvt.2018.2890416
73. Chen, Y., Zhao, L., Lee, K. M., Yoo, C., Huang, S., & Fitch, R. (2020). Broadcast your weaknesses: cooperative active pose-graph slam for multiple robots. IEEE Robotics and Automation Letters, 5(2), 2200–2207. https://doi.org/10.1109/lra.2020.2970665
74. Perdigão, J. da S. (2014). Collaborative-Control based navigation of mobile human-centered Robots (thesis). University of Coimbra. http://hdl.handle.net/10316/40415
75. Fankhauser, P., Bloesch, M., Krusi, P., Diethelm, R., Wermelinger, M., Schneider, T., Dymczyk, M., Hutter, M., & Siegwart, R. (2016). Collaborative navigation for flying and Walking Robots. 2016 IEEE/RSJ International conference on intelligent robots and systems (IROS). https://doi.org/10.1109/iros.2016.7759443
76. Yue, Y., Wen, M., Zhao, C., Wang, Y., & Wang, D. (2022). COSEM: collaborative semantic map matching framework for autonomous robots. IEEE transactions on industrial electronics, 69(4), 3843–3853. https://doi.org/10.1109/tie.2021.3070497
77. Lajoie, P.-Y., Ramtoula, B., Chang, Y., Carlone, L., & Beltrame, G. (2020). Door-SLAM: distributed, online, and outlier resilient Slam for robotic teams. IEEE Robotics and Automation Letters, 5(2), 1656–1663. https://doi.org/10.1109/lra.2020.2967681
78. Li, J., Cheng, Y., Zhou, J., Chen, J., Liu, Z., Hu, S., & Leung, V. C. (2022). Energy-efficient ground traversability mapping based on UAV-UGV collaborative system. IEEE Transactions on Green Communications and Networking, 6(1), 69–78. https://doi.org/10.1109/tgcn.2021.3107291
79. Pham, V. C., & Juang, J. C. (2013). A multi-robot, cooperative, and active SLAM algorithm for exploration. International Journal of Innovative Computing, Information and Control, 9(6), 2567–2583. http://www.ijicic.org/ijicic-12-03075.pdf
80. Meng, Z., Sun, H., Qin, H., Chen, Z., Zhou, C., & Ang, M. H. (2017). Intelligent robotic system for autonomous exploration and active slam in unknown environments. 2017 IEEE/SICE International Symposium on System Integration (SII). https://doi.org/10.1109/sii.2017.8279295

81. Ossenkopf, M., Castro, G., Pessacg, F., Geihs, K., & De Cristoforis, P. (2019). Long-Horizon Active Slam system for multi-agent coordinated exploration. 2019 European Conference on Mobile Robots (ECMR). https://doi.org/10.1109/ecmr.2019.8870952
82. Kontitsis, M., Theodorou, E. A., & Todorov, E. (2013). Multi-robot active slam with relative entropy optimization. 2013 American Control Conference. https://doi.org/10.1109/acc.2013.6580252
83. Pei, Z., Piao, S., Souidi, M., Qadir, M., & Li, G. (2018). Slam for humanoid multi-robot active cooperation based on relative observation. Sustainability, 10(8), 2946. https://doi.org/10.3390/su10082946
84. Dikoko, B. (2021). Multiple Mobile Robot Slam for collaborative mapping and exploration (thesis). University of Cape Town, Cape Town, South Africa. http://hdl.handle.net/11427/35587
85. Indelman, V. (2015). Towards multi-robot active collaborative state estimation via belief space planning. 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). https://doi.org/10.1109/iros.2015.7354035
86. Yang, S. (2020). Active sensing for collaborative localization in Swarm Robotics (thesis). Faculty of Engineering, University of Ottawa. http://hdl.handle.net/10393/40548
87. Tian, Y. (2019). Reliable and resource-aware collaborative slam for multi-robot search and rescue (thesis). Massachusetts Institute of Technology. https://hdl.handle.net/1721.1/122417
88. Trajkovic, Miroslav & Hedley, Mark. (1998). Fast Corner Detection. Image and Vision Computing. 16. 75-87. 10.1016/S0262-8856(97)00056-5.
89. Lindeberg, Tony. (2012). Scale Invariant Feature Transform. 10.4249/scholarpedia.10491.
90. Bay, Herbert & Tuytelaars, Tinne & Van Gool, Luc. (2006). SURF: Speeded up robust features. Computer Vision-ECCV 2006. 3951. 404-417. 10.1007/11744023/32.
91. E. Rublee, V. Rabaud, K. Konolige and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," 2011 International Conference on Computer Vision, 2011, pp. 2564-2571, doi: 10.1109/ICCV.2011.6126544.
92. Ahmed, Dr. Tasneem & Singh, Dharmendra & Singh, Dharmendra & Raman, Balasubramanian & Subramanian, R.. (2014). Application of KLT (Kanade-Lucas-Tomasi) Tracker for Hotspot Observation.
93. Y. Zhang and F. Wang, "Improved Optical Flow Algorithm of Moving Object Detection," 2015 Fifth International Conference on Instrumentation and Measurement, Computer, Communication and Control (IMCCC), 2015, pp. 196-199, doi: 10.1109/IMCCC.2015.48.
94. Raguram, Rahul & Frahm, Jan-Michael & Pollefeys, Marc. (2008). A Comparative Analysis of RANSAC Techniques Leading to Adaptive Real-Time Random Sample Consensus. Lect. Note. Comput. Sci.. 5303. 500-513. 10.1007/978-3-540-88688-4-37.
95. Zheng, Yinqiang & Kuang, Yubin & Sugimoto, Shigeki & Åström, Kalle & Okutomi, Masatoshi. (2013). Revisiting the PnP Problem: A Fast, General and Optimal Solution. Proceedings of the IEEE International Conference on Computer Vision. 2344-2351. 10.1109/ICCV.2013.291.
96. Liu, R., Marakkalage, S. H., Padmal, M., Shaganan, T., Yuen, C., Guan, Y. L., & Tan, U. X. (2020). Collaborative SLAM Based on WiFi Fingerprint Similarity and Motion Information. IEEE Internet of Things Journal, 1826–1840. https://doi.org/10.1109/JIOT.2019.2957293
97. Dubois, R., Eudes, A., Moras, J., & Fremont, V. (2020). Dense Decentralized Multi-robot SLAM based on locally consistent TSDF submaps.IEEE International Conference on Intelligent Robots and Systems, 4862–4869. https://doi.org/10.1109/IROS45743.2020.9341680
98. Dubois, R., Eudes, A., & Frémont, V. (2022). Sharing visual-inertial data for collaborative decentralized simultaneous localization and mapping. Robotics and Autonomous Systems. https://doi.org/10.1016/j.robot.2021.103933
99. Mur-Artal, R., & Tardós, J.D. (2017). ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras. IEEE Transactions on Robotics, 33, 1255-1262.
100. M., Thrun, S., Koller, D., & Wegbreit, B. (n.d.). FastSLAM 2.0: An Improved Particle Filtering Algorithm for Simultaneous Localization and Mapping that Provably Converges.
101. Paul J. Besl, N. D. M. (1992). A Method for registration of 3-D Shapes. IEE Tranactions on Pattern Analysis and Machine Intelligence, Vol. 2 Issue 2.
102. Triggs, B., Mclauchlan, P., Hartley, R., & Fitzgibbon, A. (n.d.). Bundle Adjustment-A Modern Synthesis
103. Grisetti, G., Kummerle, R., Stachniss, C., & Burgard, W. (2010). A tutorial on graph-based SLAM. <i>IEEE Intelligent Transportation Systems Magazine, 31–43. https://doi.org/10.1109/MITS.2010.939925
104. R. Dubois, A. Eudes and V. Frémont, "On Data Sharing Strategy for Decentralized Collaborative Visual-Inertial Simultaneous Localization And Mapping," 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2019, pp. 2123-2130, doi: 10.1109/IROS40897.2019.8967617.
105. T. Duhautbout, J. Moras and J. Marzat, "Distributed 3D TSDF Manifold Mapping for Multi-Robot Systems," 2019 European Conference on Mobile Robots (ECMR), 2019, pp. 1-8, doi: 10.1109/ECMR.2019.8870930.
106. Thibaud Duhautbout, Julien Moras, Julien Marzat. Distributed 3D TSDF Manifold Mapping for Multi-Robot Systems. 9th European Conference on Mobile Robots (ECMR 2019), Sep 2019, PRAGUE, Czech Republic. pp.1-8. ffhal-02335904
107. Stefania Bellavia, Serge Gratton, Elisa Riccietti. A Levenberg-Marquardt method for large nonlinear least-squares problems with dynamic accuracy in functions and gradients. Numerische Mathematik, Springer Verlag, 2018, 140 (3), pp.791-825. ⟨10.1007/s00211-018-0977-z⟩. ⟨hal-03126963⟩

108. M. Sanfourche, V. Vittori and G. Le Besnerais, "eVO: A realtime embedded stereo odometry for MAV applications," 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2013, pp. 2107-2114, doi: 10.1109/IROS.2013.6696651.
109. Geiger, A., Roser, M., &; Urtasun, R. (2010). Efficient Large-Scale Stereo Matching. Springer, pp. 25–38.
110. W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3d surface construction algorithm," in ACM siggraph comuter graphics, vol 21, no. 4. ACM, 1987, pp, 163-169.
111. C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel and J. D. Tardós, "ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual–Inertial, and Multimap SLAM," in IEEE Transactions on Robotics, vol. 37, no. 6, pp. 1874-1890, Dec. 2021, doi: 10.1109/TRO.2021.3075644.
112. T. Qin, P. Li and S. Shen, "VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator," in IEEE Transactions on Robotics, vol. 34, no. 4, pp. 1004-1020, Aug. 2018, doi: 10.1109/TRO.2018.2853729.
113. Kneip, Laurent & Chli, Margarita & Siegwart, Roland. (2011). Robust Real-Time Visual Odometry with a Single Camera and an IMU. Proc. of the British Machine Vision Conference (BMVC). 10.5244/C.25.16.
114. R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras," in IEEE Transactions on Robotics, vol. 33, no. 5, pp. 1255-1262, Oct. 2017, doi: 10.1109/TRO.2017.2705103.
115. Qader, Wisam & M. Ameen, Musa & Ahmed, Bilal. (2019). An overview of bag of words;Importance, implementation, applications, and challenges. 200-204. 10.1109/IEC47844.2019.8950616.
116. J. Qian, X. Mai and X. Yuwen, "Real-time power line safety distance detection system based on LOAM SLAM," 2018 Chinese Automation Congress (CAC), 2018, pp. 3204-3208, doi: 10.1109/CAC.2018.8623168.
117. Saat, Shahrizal & Rashid, WN & Tumari, MZM & Saealal, MS. (2020). HECTORSLAM 2D Mapping for Simultaneous Localization and Mapping (SLAM). Journal of Physics: Conference Series. 1529. 042032. 10.1088/1742-6596/1529/4/042032.
118. Jayanth Kumar, Thenepalle & Singamsetty, Purusotham. (2019). An open close multiple travelling salesman problem with single depot. Decision Science Letters. 8. 121-136. 10.5267/j.dsl.2018.8.002.
119. J. D. Gammell, S. S. Srinivasa and T. D. Barfoot, "Batch Informed Trees (BIT): Sampling-based optimal planning via the heuristically guided search of implicit random geometric graphs," 2015 IEEE International Conference on Robotics and Automation (ICRA), 2015, pp. 3067-3074, doi: 10.1109/ICRA.2015.7139620.
120. M. Kaess, A. Ranganathan and F. Dellaert, "iSAM: Incremental Smoothing and Mapping," in IEEE Transactions on Robotics, vol. 24, no. 6, pp. 1365-1378, Dec. 2008, doi: 10.1109/TRO.2008.2006706.
121. S. Gatesichapakorn, J. Takamatsu and M. Ruchanurucks, "ROS based Autonomous Mobile Robot Navigation using 2D LiDAR and RGB-D Camera," 2019 First International Symposium on Instrumentation, Control, Artificial Intelligence, and Robotics (ICA-SYMP), 2019, pp. 151-154, doi: 10.1109/ICA-SYMP.2019.8645984.
122. P.-Y., Ramtoula, B., Wu, F., & Beltrame, G. (2021). Towards collaborative simultaneous localization and mapping: a Survey of the current research landscape. https://doi.org/10.55417/fr.2022032
123. Zou, D., Tan, P., & Yu, W. (2019). Collaborative visual SLAM for multiple agents:A brief survey. In Virtual reality and intelligent hardware (Vol. 1, Issue 5, pp. 461–482). KeAi Communications Co. https://doi.org/10.1016/j.vrih.2019.09.002
124. J. G. Mangelson, D. Dominic, R. M. Eustice and R. Vasudevan, "Pairwise consistent measurement set maximization for robust multi-robot map merging," 2018 IEEE International Conference on Robotics and Automation (ICRA), 2018, pp. 2916-2923, doi: 10.1109/ICRA.2018.8460217.
125. Howard A. Multi-robot simultaneous localization and mapping using particle filters. The international journal of robotics research. 2006;25(12):1243-1256. doi:10.1177/0278364906072250.
126. Arvanitakis, I., & Tzes, A. (2017). Collaborative mapping and navigation for a mobile robot swarm. 2017 25th Mediterranean Conference on Control and Automation (MED).
127. Mur-Artal, R., Tardos, J. D. (2017). ORB-SLAM2: An Open-source SLAM system for monocular, stereo, and RGB-D cameras. IEEE Transactions on Robotics, 33(5), 1255–1262. https://doi.org/10.1109/TRO.2017.2705103
128. Quigley, M., Gerkey, B., Conley, K., Faust, J., Foote, T., Leibs, J., Berger, E., Wheeler, R., ; Ng, A. (2009). ROS: an open-source Robot Operating System. ICRA.t. 3,3.2,kobe, Japan http://stair.stanford.edu