Proceedings of the 2023 International Conference on Machine Learning and Automation
DOI: 10.54254/2755-2721/41/20230758

Series of advanced SLAM processes and applications

Ryan Xianqian Yuan

Canyon Crest Academy, San Diego, CA, 92130, USA

YuanX4798@my.sduhsd.net

Abstract. Simultaneous Localization and Mapping (SLAM) is an increasingly important field in robotics. SLAM enables the processes of localization and mapping to run simultaneously by using methods like feature extraction, feature matching, extended Kalman Filter, and probability distribution models, allowing robots to have more autonomy and greater efficiency. SLAM has been applied in industry robots, autonomous vehicles, augmented reality, UAVs, humanoid robots, and planetary rovers. With an increasing number of applications of SLAM, the demand for higher-performance SLAM algorithms has driven innovation and advancement in the field every year. SLAM is a highly promising algorithm for the future of robotics, but problems like uncertainty, correspondence, and time complexity prevent the full use of SLAM in robotics applications. It is essential to analyze SLAM in all aspects to apply it to future work. This article provides a detailed insight into the SLAM process, considers previous advancements and current problems, and discusses the future of SLAM.

Keywords: SLAM, Feature Extraction, Uncertainty, Correspondence, Mathematical Model.

1. Introduction

SLAM has been a constantly changing field ever since its origin in 1986. At its core, SLAM is the combined process of localization and mapping that enables the efficient self-exploration of robots. Localization and mapping run both concurrently and recursively This is because localization must be computed with respect to a map that is being built as the robot moves, while for the map to be built the localization for where the last perception of the environment was needs to be factored in accordingly. SLAM addresses the problem of exploring terrains that are unreachable to humans in an efficient and fully autonomous manner. With no human interference and without prior knowledge of the environment, robots can map their environment, localize themselves in it, and perform pre-programmed tasks.

SLAM has been applied in many fields today, including autonomous vehicles, augmented reality, UAVs, humanoid robots, and space exploration. SLAM has been used in autonomous vehicles to improve safety and efficiency by enabling vehicles to simultaneously map their surroundings and locate themselves within the map. In augmented reality, SLAM has been used to track and update the position and orientation of virtual objects in the real world. SLAM has been applied to UAVs for autonomous tasks like surveillance, inspection, and search and rescue missions. Similarly, SLAM has been used to autonomously control humanoid robots so that they can efficiently and accurately interact with objects and navigate the environment. Finally, SLAM has been proposed as an exploration algorithm for planetary rovers and deep space missions and has directly been researched and applied in NASA satellites.

The last decade has been of innovation and improvements to the SLAM algorithm. Notable algorithms and their advancements are shown in Table 1 below.

Table 1. Significant SLAM algorithms by year.

Year	SLAM Algorithm	Significance
2013	DSO (Direct Sparse Odometry)	Utilized stereo cameras for mapping
2014	LSD-SLAM	Combined monocular and stereo cameras for modelling of the environment
2015	VINS-Mono	Integrated IMU with monocular cameras to create a map of the environment
2016	ORB-SLAM2	Improved original ORB-SLAM algorithm by adding loop closure detection
2017	OKVIS	Visual-inertial SLAM system that used nonlinear optimization for camera trajectory and 3D reconstruction
2018	OpenVSLAM	Open-source SLAM system compatible with monocular, stereo and RGB-D cameras
2021	Collaborative SLAM	Combined odometry data from multiple robots to generate 3D maps
2023	Laser and Visual SLAM	Combined laser and visual sensors for improved SLAM accuracy and robustness

2. SLAM

2.1. Sensors in SLAM
SLAM algorithms utilize various sensors depending on their application. These include sonar, laser, LiDAR, RGB-D, Monocular, Stereo, IMU, and more.

Table 2. SLAM sensors and their respective applications.

Sensor	Application	SLAM Algorithm
Laser	Indoor, Outdoor, Drones	Laser-based SLAM, TinySLAM
LiDAR	Indoor, Outdoor, Drones, Deep Space	LiDAR SLAM, CT-SLAM
Monocular	Indoor and Outdoor, Satellites	Mono SLAM
Sonar	Underwater	Bat SLAM
Stereo	Indoor, Outdoor, UAV, Planetary Rover	Stereo LSD-SLAM
RGB-D	Indoor	RGB-D SLAM

2.2. SLAM Process
SLAM algorithms vary in their sensors, applications, and methods, but the overall process shown in Figure 1 remains the same. Sensor data describes the odometry data collected by the physical sensors on the robot (see Table 2 for examples). Using this sensor data, the SLAM algorithm performs mapping and localization in parallel, building a map of the environment and localizing the robot within it at the same time. The map is used to localize the robot, while the robot’s localized position is used to update relative distances in the map accordingly.

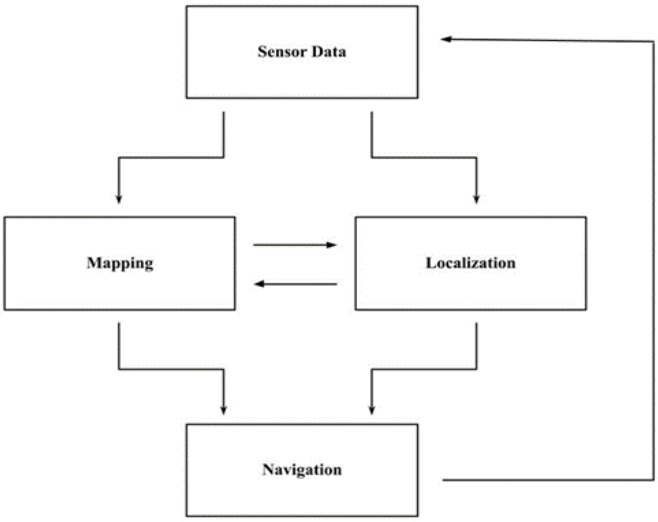

Figure 1. Conceptual SLAM Process.

Localization is not an absolute position, but rather relative distances between the robot and landmarks that are combined to provide an estimate of the robot's pose within the map. The robot's origin is the initial localization point, and from there, the more landmarks the robot localizes itself against, the more accurate the robot's estimated pose will be.

Thus, through mapping, the robot identifies as many landmarks as it can, and relative distances are used in localization to estimate the robot's pose. These relative distances between the robot and landmarks are then used to update the map to adjust for any errors, and as the robot continues to navigate through the terrain, the cycle continues. Localization and mapping are dependent on each other, but through SLAM they can occur simultaneously and create a positive feedback loop.

As the mapping and localization processes run, the robot navigates through the environment, avoiding obstacles detected in the map. As the robot moves forward, both the robot's relative location and the map are updated as new sensor data is collected. This process repeats until the robot has completed its objective.

To illustrate this concept, an example is provided in Figure 2 below. A SLAM-based rover equipped with IMU and visual odometry systems is tasked with autonomously traversing an unknown terrain and reaching the desired landmark. IMU data is used to estimate the robot's orientation and displacement, while visual odometry is processed to detect landmarks and their respective distances from the rover.

Considering the rover's relative landmarks and origin point, a map can be constructed, and its pose can be estimated within it. A path-planning algorithm can then be used to find an obstacle-avoiding trajectory from the robot's current position to its next waypoint.

As the rover navigates along this path, new landmarks are detected, which are used to update the map and re-localize the robot as it heads toward its desired landmark. This cycle of detecting new landmarks, updating the map, localizing the rover, and path planning between waypoints occurs as the robot traverses the terrain, making it an extremely efficient process as compared to a robot completing step-by-step decision-making before it acts.

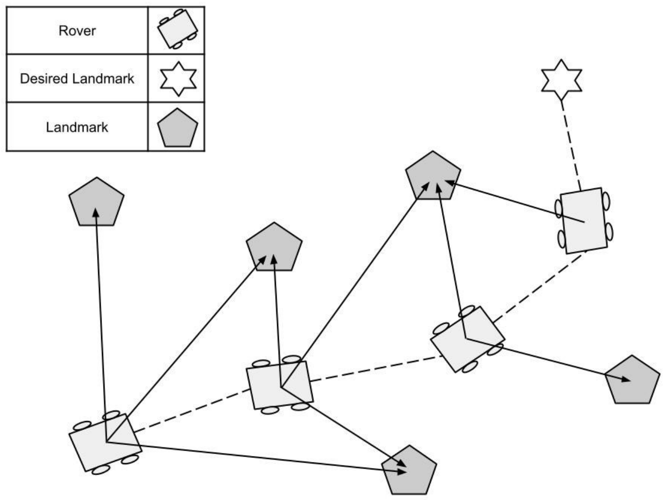

Figure 2. An illustrative depiction of a rover navigating terrain using SLAM.

2.3. *Feature Extraction*

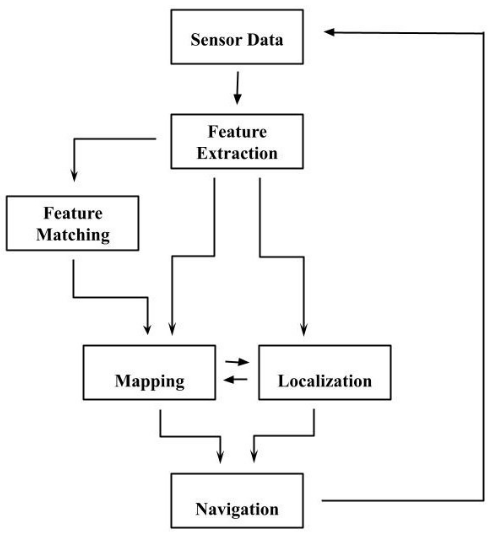

Figure 3. Feature Extraction in the SLAM Process.

Feature Extraction is a crucial component in the SLAM process. As shown in Figure 3, it acts as a preprocessing method from sensor data to mapping and localization. When building a map, SLAM extracts static features, or landmarks, that will help the localization and mapping process. Then, in Feature Matching, it matches these landmarks to existing ones, or if it does not recognize the landmark, it identifies it as a new feature on the map. If the landmark is pre-existing, then the robot uses the last stored distance and the current distance from the robot to the landmark to calculate the robot's displacement. If the landmark is a new landmark, then SLAM adds the landmark to the map and uses that landmark to localize the robot for future updates. Each time the robot collects new sensor data, the system will extract features from the visual odometry and use it to re-localize the robot and update the map.

2.4. Mathematical Model

SLAM uses a probability distribution to make probabilistic estimates of the robot's pose and model the uncertainty in the odometry data. As the robot's pose and map are constantly updated with new data, so is the probability distribution model, creating a positive feedback loop. The main variables in the model are listed below.

t : Time instant
x_t : Robot displacement from time $t – 1$
z_t : Robot position in map at time t
m_t : SLAM-generated map of terrain and landmarks at time instant t
u_t : Odometry data between time t -1 and t
d_t : Distance between robot and landmarks at time t
U_t : $\sum u_t$ Sequence of odometry data from time 0 to time t
D_t : $\sum d_t$ Robot motion or sequence of distances between the robot and landmarks from time 0 to time t

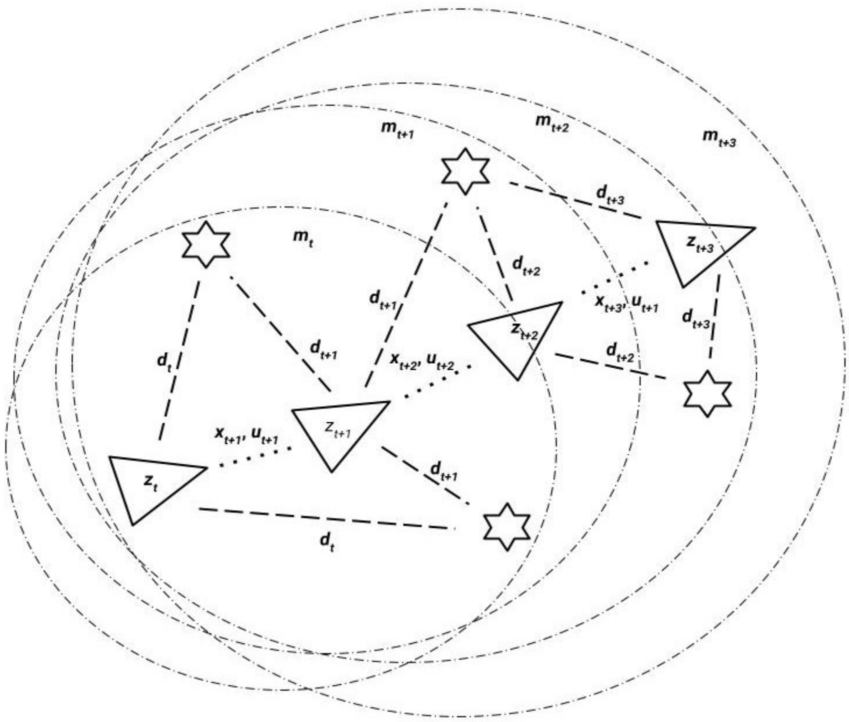

Figure 4. Visual depiction of the variables in the SLAM process.

$$P(x_t \mid x_{t-1}, u_t) \tag{1}$$

Here, the previous position at time t - 1 and the odometry in between time t - 1 and t are used to calculate the position of the robot at time t.

$$P(d_t \mid x_t, m_t) \tag{2}$$

Here, the position and map m_t at time t are used to calculate the distance between the robot and landmarks for the robot to localize itself within the map.

$$U_t = \Sigma\ u_t\ ,\ \ D_t = \Sigma\ d_t \tag{3}$$

Here, the new updated values of u_t and d_t are stored in memory in the sequences U_t and D_t, respectively.

DOI: 10.54254/2755-2721/41/20230758

$$P(zt, mt \mid Dt, Ut) \tag{4}$$

Here, visual, LiDAR, positional, and/or inertial odometry data sequence U_t is used to generate a map m_t at time t. The SLAM system then uses the distances D_t between the robot and landmarks from time 0 to t to localize itself most probabilistically within the generated map (z_t). The visual depiction of variables is shown in Figure 4.

2.5. Uncertainty in SLAM

Uncertainty is a common problem in SLAM. The most common forms of uncertainty in SLAM are location uncertainty, hardware uncertainty, and correspondence error. These problems result from both localization and mapping. In localization, inaccuracy in odometry systems can lead to hardware uncertainty, and mismatching distances to landmarks can cause location uncertainty. In mapping, misidentification of landmarks can result in correspondence issues that mess up the entire localization process. Although uncertainty is a heuristic that cannot be solved completely, there have been various methods that have decreased uncertainty in SLAM. Two such methods are sensor fusion and the Kalman Filter.

2.5.1. Correspondence Error

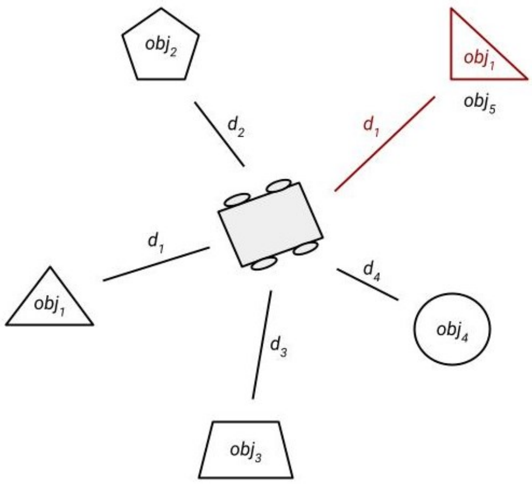

Figure 5. Misclassification from a similar landmark.

Correspondence error is when the robot misclassifies a new landmark as a previous one. This occurs when two landmarks closely resemble one another, resulting in the algorithm identifying one as the other. In Figure 5, obj_5 is similar in size and shape to obj_1, so SLAM classifies obj_5 as obj_1. This results in a series of errors in the mapping and localization process.

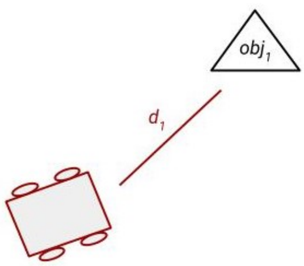

Figure 6. Resulting localization of the robot.

DOI: 10.54254/2755-2721/41/20230758

obj_5 is identified as obj_1, so the robot will evaluate the distance to obj_5 and store that as the distance d_1 to obj_1. When SLAM maps the landmarks and localizes itself, the resulting localized position of the robot will be as shown in Figure 6. This localization of the robot from obj_5 will disagree with the localized position derived from d_2, d_3, and d_4, so SLAM will attempt to account for this by using a probabilistic distribution as shown in *3.5* to balance out the contradictory value of d_1. This will result in a better-predicted position than shown in Figure 6, but the misclassification of obj_5 will drastically decrease the accuracy of the robot's localization and mapping.

2.5.2. Hardware Uncertainty

Hardware uncertainty occurs either when there is noise in the hardware that leads to inaccurate sensor data, or when the sensor data is accurate but external factors result in the subsequent calculations being off. In both cases, the inaccurate outputs will be factored into robot and landmark position estimates, likewise resulting in errors in localization and mapping.

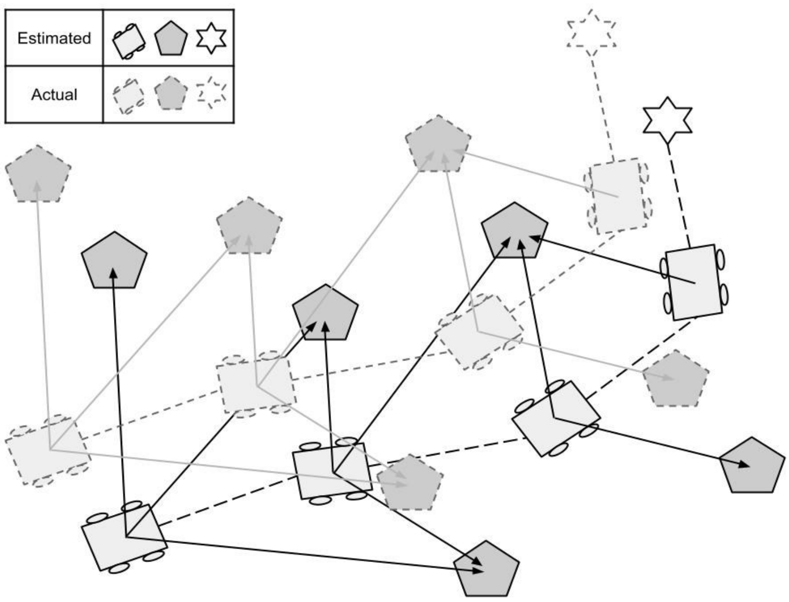

Figure 7. Estimated vs. actual position and relative distances.

Figure 7 depicts the result of hardware uncertainty on robot localization. Factors like wheel slippage and uneven terrain can be the cause of the error. Wheel odometry cannot detect slippage or uneven terrain, resulting in the estimated path diverging from the actual path as the error accumulates. Inertial and visual odometry is better at accounting for these factors, but uncertainty is still likely to arise.

Table 3. Uncertainty and sampling rates in individual odometry systems versus a combined sensor fusion.

Method	Uncertainty	Sampling Rate
Wheel Odometry	10%	10-100Hz
Inertial Odometry	5-10%	10-100Hz
Visual Odometry	1-5%	0.5Hz
Sensor Fusion	1-2%	10Hz

Sensor fusion is a method that is used to reduce hardware uncertainty. As shown in Table 3, the uncertainty for individual odometry systems is high, but a sensor fusion of multiple odometry systems drastically reduces the uncertainty of the system. Using sensor fusion, different odometry systems can be cross-compared to validate the data and balance out any sensor input errors.

2.5.3. Location Uncertainty

Location uncertainty is when the robot is unsure of its position, which most commonly results in either an insufficient number of landmarks or contradictory distance measurements. If there are not enough landmarks, then the robot cannot pinpoint its position, resulting in a larger probability distribution of the robot's location and thus more uncertainty.

Contradictory distance measurements that lead to location uncertainty can result from both correspondence error and hardware uncertainty. If hardware uncertainty is present, then the robot either has conflicting distances between accurate and inaccurate data, or the overall data is inaccurate, both resulting in location uncertainty. If a correspondence error occurs, then the distance to the incorrectly classified landmark conflicts with the distances to the correctly classified landmarks, resulting in uncertainty as the robot is unsure of which distances to factor into the localization process.

For example, in Figure 8 a robot can use the distance to three landmarks to triangulate its relative position, but if the distance to a fourth landmark disagrees with the calculated position from the three initial landmarks, then SLAM experiences location uncertainty, where the robot is unsure of its relative position because of conflicting data inputs.

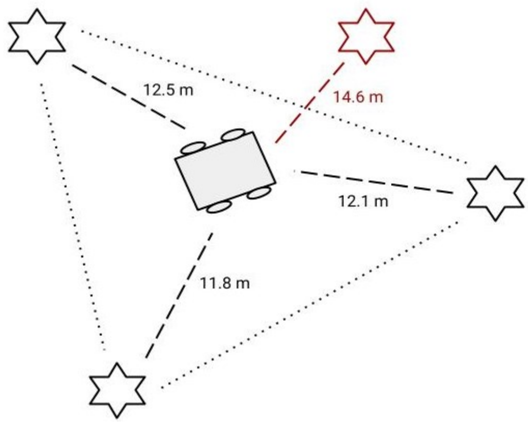

Figure 8. Location uncertainty from an unexpected measurement.

3. Conclusion

SLAM is an effective algorithm regarding autonomy and efficiency. Because localization and mapping are run simultaneously, SLAM is much more efficient than step-by-step algorithms. Moreover, the capabilities of SLAM allow for much more robot autonomy, and thus SLAM can be used in more challenging applications. SLAM is not a perfect solution to autonomous navigation, but rather an optimization that produces a heuristic result. Processes like sensor fusion, probabilistic distribution, and extended Kalman filter can minimize the effect of error on the SLAM system, but the problem of uncertainty and correspondence remains.

Straits Research predicts that SLAM technology will grow at a CAGR of 49.41%, reaching a market size of 9.42 billion USD by 2030. Between now and 2030, advancements addressing these problems are expected to be made for the growth of this field to fluctuate. The problems of location uncertainty, hardware uncertainty, correspondence, data association, and time complexity that were first brought up when SLAM was proposed are still prevalent in modern SLAM algorithms. Although these algorithms have become more efficient in dealing with these problems, they are not optimized enough to the point where SLAM is the dominating algorithm in the robotics field. From here, future work in SLAM includes (1) improving latency in sensor hardware and algorithms, (2) optimizing precision in robot and landmark position, (3) improving robustness, scalability, and handling of challenging environments, (4) integrating multiple sensors to improve overall system performance and accuracy, (5) addressing longterm mapping and exploration of large-scale environments, and (6) enabling higher-level reasoning and decision making.

References

[1] R Khairuddin, M S Talib and H Haron 2015 Review on simultaneous localization and mapping (SLAM)
[2] Hong S, Bangunharcana A, Park J M, Choi M, and Shin H S 2021 Visual SLAM-Based Robotic Mapping Method for Planetary Construction
[3] Chen Y 2017 Possibility of Applying SLAM-Aided LiDAR in Deep Space Exploration
[4] Kazerouni I A, Fitzgerald L, Dooly G and Toal D 2022 A survey of state-of-the-art visual SLAM
[5] Smith R, Self M, and Cheeseman P 1986 Estimating Uncertain Spatial Relationships in Robotics
[6] Montemerlo M, Thrun S, Koller D and Wegbriet B 2002 FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem
[7] Barros A, Michel M, Moline Y, Corre G and Carrel F 2022 A Comprehensive Survey of Visual SLAM Algorithms
[8] Vecilla M A 2021 Autonomous Navigation of Planetary Rovers
[9] Gupta A and Fernando X 2022 Simultaneous Localization and Mapping (SLAM) and Data Fusion in Unmanned Aerial Vehicles: Recent Advances and Challenges
[10] Kolhatkar C and Wagle K 2020 Review of SLAM Algorithms for Indoor Mobile Robot with LIDAR and RGB-D Camera Technology
[11] Bonetto E, Goldschmid P, Pabst M, Black M J and Ahmad A 2022 iRotate: Active Visual SLAM for Omnidirectional Robots
[12] Singandhupe A and La H M 2019 A Review of SLAM Techniques and Security in Autonomous Driving