Visual SLAM in non-stationary environments: State-of-the-Art

Robin Connor Schramm
Reutlingen University
Fraunhofer IPA
robinschramm@gmx.de

Abstract

Simultaneous localization and mapping (SLAM) is a key technology needed for autonomous mobile robot navigation. Visual SLAM (vSLAM), a type of SLAM where mainly cameras are used, plays an increasingly important role in the SLAM community. There are many solutions to the vSLAM problem in non-stationary environments, but none of them are considered truly robust and broadly applicable. Those types of environments usually have either dynamic objects in them or change in appearance over time. In this paper, the question on how the state-of-the-art for vSLAM in non-stationary environments can be defined and what current technologies look like will be answered. In the end, all approaches for vSLAM in non-stationary environments are still trying to solve specific problems, even though there exists a number of robust solutions. In conclusion, non-stationary environments differ too wildly and the approaches are too specific for there to be a general robust solution, and as such, there is no system that can be clearly considered the state-of-the-art.

Supervisor University	Prof. Dr.-Ing. Cristóbal Curio Hochschule Reutlingen cristobal.curio@reutlingen-university.de
Supervisor Organisation	Philipp Schnattinger Fraunhofer IPA philipp.schnattinger@ipa.fraunhofer.de

Informatics Inside Frühling 2021
05. Mai 2021 Hochschule Reutlingen

CCS Concepts

• Computing methodologies → Vision for robotics; Motion path planning; • Computer systems organization → Robotic autonomy;

Keywords

Computer vision, Simultaneous localization and mapping, Robot vision systems, Unmanned autonomous vehicles

1 Introduction

The simultaneous localization and mapping (SLAM) technique is one of the key enabling technologies of mobile robot navigation. Through SLAM, an autonomous mobile robot can navigate through an unknown environment while building a map to help locate itself in said environment. Stachniss et al. state in the Springer Handbook of Robotics [1] (page 1153) that if a robust, general-purpose solution to SLAM can be found, many new applications of mobile robotics will become possible. At present, there exist several robust methods for mapping environments. Those environments are, however, limited to be mainly static, structured, and of limited size. The SLAM problem in changing environments and environments including moving objects is explored through several promising approaches, can however still be considered under development. One variant of SLAM that uses cameras as the only sensor employed, called visual SLAM (vSLAM), has been a popular and widely used SLAM technique in recent years and will probably have an essential role in the future [1].

1.1 Motivation and Goals

In the Fraunhofer Institute for Manufacturing, Engineering and Automation IPA in the group industrial and commercial service robots, procedures for autonomous navigation are currently researched and in development. Especially camera systems for mobile robots play an increasingly important role due to their inexpensiveness and higher information density in comparison to conventional two-dimensional (2D) laser scanners. The SLAM problem is considered *solved* for non-changing environments, meaning there are robust methods with high accuracy and low failure rate for static, structured and limited size environments with no moving objects or changing parts of the environment [1]. However, in real-life applications like logistic environments, there are often moving objects in the vicinity of the robot, like humans or other robots. There are also permanent changes to the environment like moved furniture in office environments or moved wares in warehouses. Even more drastic changes like building or removing walls have to be considered for *lifelong SLAM* [2].

The goal of this paper is to describe the state-of-the-art for vSLAM in non-stationary environments, how advanced current techniques are and if there are robust solutions available. It also defines the term non-stationary environment. In addition, it will be answered on how to assess the robustness of a SLAM system.

1.2 Methodology

First, an overview of the terminology and fundamentals of different common approaches to SLAM as well as definitions for the terms vSLAM and non-stationary environments will be given. Next, the state-of-the-art for vSLAM in non-stationary environments will be explored by first defining how the robustness of SLAM can be assessed. Then, different problems and their approaches from recent years will be given as well as if the solutions can be called robust or solved. Finally, a discussion about successful past and possible future work will be held.

2 Basics

Stachniss et al. define in the Springer Handbook of Robotics [1] the problem of *simultaneous localization and mapping* (SLAM) as "one of the key enabling technologies of mobile robot navigation" ([1], S. 1153). SLAM addresses the problem of a mobile and autonomous robot navigating in an to the robot unknown environment. It typically consists of two parts: Building a map of the environment in any form and simultaneously locating itself in said map. Different kinds of sensors can be used to build the map, popular sensors include lasers, cameras with depth sensors or simple RGB cameras.

Formally, SLAM is a probabilistic problem. The robot's location at any time t is defined as x_t. Since the basic SLAM problem addresses only robots on a flat ground, x_t consists of its two-dimensional (2D) coordinate plus a rotational value θ, stored in a three-dimensional (3D) vector. The initial location x_0 serves as a point of reference for the remainder of the estimation algorithm. Each point $\{x_1, x_2, ..., x_T\}$ up to the terminal time T has to be calculated relative to x_0. Motion sensor data (odometry) in the robot's wheels or legs provide relative information between two points [1].

A SLAM system is typically split up between a front-end and a back-end [3]. The front-end represents usually the interface to the robot's sensors like cameras and odometry sensors and is responsible for feature extraction and feature matching in the case of visual SLAM (vSLAM). The back-end builds the map based on the data extracted in the front-end and estimates the position of the robot inside the map. The map composed in the back-end is usually represented in the form of a graph where each node usually consists of the robot's position x_t and rotation θ like described before. Each edge in the graph represents

a spatial constraint between the two robot poses [4]. This is only a general simplification since different SLAM systems can differ widely in structure and function.
Cadena et al. address in [4] the question why SLAM is needed for autonomous robots. A critical aspect of many SLAM systems and one reason why SLAM is important, is the ability to create loop closures. When the robot comes by a point that he has already passed and *remembers* that he has already visited that point, the map can be updated accordingly. Without loop closures, SLAM reduces down to odometry alone and every position the robot calculates is dependent on previous positions, which is also called dead-reckoning. Since there is no such thing as perfect sensors, over longer periods of time any odometry sensor will drift at least a little bit. So even when the robot drives only in a straight line, due to drift and accumulated errors, the resulting path can appear as a curve. Through detecting a loop closure the robot can *reset* its localization error and neutralize drift in the map. Loop closures are also useful for the robot's understanding of the real topology of the environment and helps finding shortcuts between locations.

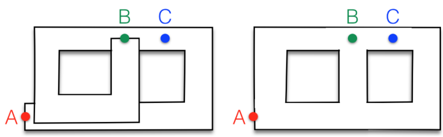

Figure 1: Left: Map built from odometry. Right: Map built from SLAM with loop closures. Taken from [4].

Figure 1 taken from Cadena et al. [4] shows the necessity of loop closures in SLAM. The starting point is A. On the left, the map is built from odometry alone. Here, point B and C can be far away from each other in the constructed map, even though they are close in reality. On the right, the map is build using SLAM with loop closures. Note that the map depicts the real topology of the scene and reveals shortcuts between the points A, B, and C.

2.1 Visual SLAM

When the only sensors employed in a SLAM system are cameras, the term visual SLAM (vSLAM) for that system is commonly used [5], [6]. There is a tendency for using vision as the only perception system for solving the SLAM problem. This is due to properties of cameras like inexpensiveness, lightness in weight, and low power consumption. Furthermore, cameras can obtain an environment's appearance, color and texture in comparison to e.g. laser scanners. Nevertheless, cameras have obstacles to overcome like resolution, changes in lightning, surfaces without texture or blurred images [6].
Cameras are also used as the primary sensor for many localization and place recognition algorithms. Place recognition is a highly relevant research topic in robotics and used primarily on long-term mobile autonomous robots and as such is not only limited to e.g. detecting loop closures in SLAM systems [7]. Examples for famous vSLAM Systems that can be considered as part of state-of-the-art for vSLAM are, as examined by Servieres et al., Vins-Mono [8], DSO [9] and ORB-SLAM2 [10].

2.2 Non-stationary environments

The term static environment is well defined since the majority of approaches to solve the SLAM problem apply to those environments. Simply put, a static environment is an environment that doesn't contain moving or changing objects, also called dynamic objects, and doesn't change over any period of time [11]. In contrast, a non-static environment contains dynamic objects in any form and/or does change over time. In most real world scenarios it is practically impossible to separate strictly into different types of environments since most environments change over time in any form and contain dynamic objects at some point in their life cycle. To tackle this problem, Hentschel and Wagner differentiate between three types of objects in [11]. They differentiate between *dynamic*, *semi-static* and *static objects*. Objects that are actively moving in a defined direction like cars, people or other robots are called dynamic objects. Objects that don't move actively in the sensory range of the robot but change in other ways or outside the range of the robot's sensors are called semi-static. This includes environmental changes due to seasons and weather like the absence or presence of rain, snow, etc., changes to objects like cars parking or objects that are moved to other places and changes to objects themselves like growing trees or painted walls and many more. Objects are called static when they are mostly invariant to changes. In addition, changes are not always permanent. Objects generally get moved at some point, buildings can change in structure and appearance over time and outdoors, the environment is constantly changing through weather and change in seasons. In conclusion, an environment that is mainly composed of semi-static objects but also includes some static objects like walls and a limited amount of dynamic elements can be called non-stationary environment. For example Morris et al. [12] developed a method for navigation in non-stationary environments. Their test-environment consisted of an office scenario in which office dividers and chairs were moved manually to block the robot's path in different runs. Also present were two people who moved around the robot. Another adjacent term is lifelong SLAM. Shi et al. [13] define the term as a SLAM system that builds and maintains a persistent map of the robot's environment. Challenges include changed viewpoints, *changed things* i. e. semi-static objects, and dynamic objects. The lifelong SLAM problem will be described further in chapter 3.3. In figure 2, some examples for non-stationary environments can be seen. These come from the OpenLORIS datasets for lifelong SLAM but are applicable for non-stationary SLAM as well. The images include dynamic objects like humans, semi-static objects like chairs and different viewing angles in common environments like a cafe and a supermarket.

Figure 2: Examples for some non-stationary scenarios, taken from the OpenLORIS datasets [13].

3 vSLAM in non-stationary environments: state-of-the-art

In this chapter it will be examined what the state-of-the-art for SLAM in non-stationary environments looks like. First, it will be explained what is needed to call a SLAM system robust. Next, the definition of the specific SLAM variant which will be explored will be given. Then, different topics with their own challenges and approaches to overcome theses challenges will be shown.

3.1 When is a SLAM system considered robust?

To begin the assessment of the state-of-the-art for vSLAM in non-stationary environments, one broad question has to be asked first: When can be a SLAM system considered robust? Robust is a term which is widely used for a multitude of different SLAM system proposals. In addition, Cadena et al. [4] link the question *SLAM is solved?*, to a systems robustness in addition to accuracy. Furthermore, SLAM itself is such a broad topic so that an answer to those questions can only exist with more specified parameters. In general, robustness is linked to error handling [4]. Failure in SLAM systems can either be algorithmic or hardware-related. It is for example crucial for SLAM systems to filter loop closures so that mostly correct loops are detected. Even when false loops are fed to the back-end the system has to be resilient enough to still function adequately. For more specific answers, Cadena et al. describe in [4] three main aspects that have to be defined to answer the question if SLAM is solved and, in conclusion, if a SLAM system is robust. Those aspects will be summarized in the following:

- **Robot:** Most importantly the type Robot i.e. the type of motion used like wheeled or legged and the robot's speed. Also the types of sensors influence the underlying SLAM system strongly. Common types in addition to odometry sensors are for example cameras and laser scanners. Since a SLAM system can be computationally demanding, the available computing power is also an important aspect to consider.
- **Environment:** The type of environment influences the desired map representation and the used SLAM system strongly. The environment can be for example a relatively simple 2D plane or consists of different heights and floors and is as such 3D. Planar environments tend to be easier for robots to understand than environments that include elevation or multiple floors. The complexity and feature richness also influence the SLAM system greatly. Another question to ask is, if artificial landmarks are present to guide the robot to an extend. Furthermore, does the environment contain mainly static, semi-static, or dynamic objects and what is the maximum size of the environment?
- **Performance:** Other important aspects include the desired accuracy. In some scenarios, the system has to perform really precise actions, in others an error of one meter can still be in the acceptable margin. The internal representation of the environments, like graph based, 2D map, 3D map, or even multiple maps, has also to be considered. Finally, the maximum time of operation is critical information in order to select the right SLAM system. Is the robot deployed only for a few hours, days or even years?

In summary, there are many configurations possible and each individual circumstance has to be evaluated separately. There are already robust solutions for a few of those configurations and can as such be considered *solved*. An example for a scenario for which there exist multiple robust solutions is SLAM in small, feature-rich, and completely stationary environments [4]. ORB-SLAM2 [10] would be an example as a highly cited paper applicable in this scenario.

Another example for robust SLAM systems are NASA's mars rovers [14], which move incredibly slow in an environment that remains mostly static and only changes in appearance over time. Some of those systems function for years and can cope with errors really well and are therefore robust systems.
Since there is such a multitude of parameters and the list given above is definitely not exhaustive, there are probably many possible SLAM systems out there that don't have any robust solutions or aren't even properly explored yet.
In the following, the underlying problem and difficulty for vSLAM in non-stationary environments as well as current solutions for each problem and their maturity will be described.

3.2 Visual place recognition

Visual place recognition is a well defined but challenging task as described by Lowry et al in [7]. First of all, the definition of a place is ambiguous. A place can be differently defined and relies on the context it is used in. It can be defined as a single point, a small area or even a complete room. In addition, it can be stored and represented in a number of ways, like as a 2D plane or a 3D point cloud for example. Usually a new place is generated when the environment as measured by the sensor is sufficiently different from the current impression of the environment. So how can a system recognize an object or place multiple times? The basic requirements for that are a map for position and a recognition system to identify already visited places. Since SLAM inherently includes localization and mapping, this requirement is fulfilled. Performing visual place recognition can be difficult though, especially in non-stationary environments where objects may move, changing the place's appearance remarkably. Other challenges are the revisiting of places from different viewpoints and objects that look very similar to each other and as such are difficult to differentiate.
In changing environments, there are two main paradigms identified by Lowry et al. [7] to remember places in non-stationary environments.

3.2.1 Remembering and forgetting

In a non-stationary environment, each place must be updated as soon as new observations are obtained by the robot. There has to be a certain balance between remembering the more *static* parts of the environment and forgetting more ephemeral changes without deleting important parts of the map or storing each object as integral part of the world. Other challenges include high memory usage, hence the forgetting part, computationally demanding and the difficult task of automatically selecting which information to retain. Hafez et al. described in their paper from 2013 [15] the use of a bag-of-words (BoW) model to apply a quality measure for determining useful features to retain [7]. Their environment consisted of crowded Indian urban outdoor settings which includes static and many dynamic objects. They use a database as long term memory in which data from previous runs gets stored. Then, the current view is fed into the BoW. Depending on the resulting reliability score, features can be augmented and eliminated over multiple runs. They call their method robust in the said context where an error greater than 20m was observed with a small percentage of 0.321%.
One recent example for the remembering and forgetting paradigm is the long-term simultaneous planning, localization and mapping (SPLAM) system proposed by Labbé et al. published 2017 [16]. They implemented a memory management mechanism which differentiates between working memory (WM) and long-term memory (LTM). Here, the WM is where the maps are processed to satisfy memory and processing constraints while in the field. After some amount of time, nodes are transferred from the WM to the LTM to

keep the WM size constant. Instead of *forgetting* visited places, they are held back for future use. When a loop closure is detected, the WM can retrieve similar nodes from the LTM to retrieve the whole relevant area, essentially *remembering* those places again. Their system fulfills the processing requirements for SPLAM in multi-session conditions. It can still be considered under development, especially for non-stationary environments. They write, that they will examine more robust failure recovery approaches for usage in non-stationary environments.

3.2.2 Multiple representations of the environments

Since deleting anything from the map is a risky process, one approach using multiple maps has been seen a few times. These systems need some kind of filter in order to retain important features and to delete obsolete features [17]. Also in the case in which places change in an cyclic manner like offices, a system remembering and forgetting runs into problems [18]. For example, over night the *forgetting* process gradually unlearns any progress made during the day and has to relearn some aspects or places again. In response, Ranganathan et al. stated in [19], that in the office scenario, three to four images per location would be needed for consistent localization. Instead of forgetting information, there should be multiple representations of the environment, either place by place or multiple maps for the entire environment.

3.3 The lifelong SLAM problem

The lifelong SLAM problem describes the challenges a robot faces when it exists in a space over periods of time. These include changes to the environment like described in chapter 2.2. Other challenges described by [2] are incremental mapping i.e. the robot should be able to wake up anywhere and connect its current location back to the map. Another problem that could occur is recovering from failure. Sometimes the sensors get blocked which disorients the robot or degrade over time. In lifelong SLAM, it is highly possible that the environment is also non-stationary, which can lead to errors. When, for example the robot stops performing its tasks to charge its battery and comes back to a changed environment to continue its tasks, thus starting a new *session*. This leads to problems because the robot's map is outdated and the robot starts to get delocalized [20]. Hence, in non-static environments there can't be an initial mapping phase. The SLAM system has to be initialized at it's starting point and the map is build up from this single reference point. In conclusion, autonomous robots face many problems in a lifelong application which makes it difficult to develop robust lifelong SLAM systems.

3.4 Use of artificial intelligence

Another recent addition is the use of artificial intelligence (AI) in different fields in the domain of long-term autonomy (LTA) in robots. Kunze et al. conducted a survey in [17] in 2018 about the use of AI in the context of LTA. They separate their paper into six different areas where AI can help autonomous robots in real-world environments. The first area is navigation and mapping with SLAM as one of the biggest problems in this category. AI can be used to filter features in a system consisting of long-term and short-term memories.

There is also the possibility for the robot to learn about the dynamics of the environment. For example Tipaldi at al. employed in [20] in 2013 used a combination of Rao-Blackwellized filter with a hidden Markov model. Their approach estimates the pose of the robot in addition to the current state of the environment. They evaluated their framework with real world data and claim, that

their approach allows for accurate and robust localization in changing environments and in addition provides up-to-date maps of them. However, the system works only with individual cells and can not reason on an object level.
Another example for the use of AI in visual localization is shown in the paper of Naseer et al. [21] from 2018. There, they developed a robust system "to perform localization under substantial seasonal changes, e.g., summer versus winter." [21]. They used a semi-dense image description with histogram of oriented gradients descriptors combined with deep covolutional neural networks (CNNs) and a directed acyclic data association graph. Their approach allows for robust matching across season which outperforms multiple existing methods.

3.5 Modeling environmental dynamics

A novel approach for solving the dynamic SLAM problem was proposed by Krajnik et al. [22] called FreMEn in 2017. FreMEn is based on the assumption that over long time periods, the environment is influenced by various processes and that some of these processes are periodical. As such, these dynamics and their underlying processes can be analysed using spectral analysis based on the Fourier transform. Their system can represent arbitrary timescales with low memory requirements while predicting future states with error rates of less than 10%. Nevertheless, FreMEn does not exceed the performance of other approaches in the same field such as [20] and is not considered robust, its simplicity may enable application in other scenarios adjacent to life-long autonomy.

3.6 Datasets

To evaluate and compare different SLAM systems a baseline is needed. There exist several well know datasets for SLAM like the TUM RGB-D benchmark [23] and KITTI dataset [24]. Those are however intended for general use and are not specific for non-stationary SLAM.
One recent addition are the OpenLORIS-Scene datasets from Shi et al [13] published in 2020. The OpenLORIS datasets are specifically designed to test lifelong SLAM algorithms for service robots. As such, the data is as close as possible to real world applications. It includes scene changes through day-night shifts, moving people, poor illumination and blur. As future addition, the data could also serve as a benchmark for incremental learning algorithms, which, as discussed above, are an important research topic for SLAM. In figure 2 there are three examples taken from the OpenLORIS dataset, which were already discussed in chapter 2.2.

4 Discussion and future work

While there are still no broadly applicable, robust systems for non-stationary environments, there was progress made in the discipline of SLAM in the last years. Non-stationary environments still cover many widely different types of environments, which require specific solutions for their challenges. There is still no broadly usable and truly robust vSLAM system which can be used in any environment with great success. Even through the great advances in the last five years, there is still a long way ahead to achieve an universal, robust and widely usable perception system for robots. Nevertheless, there were great improvements and approaches for different combinations of robot, environment, and performance were made.
Additionally, it is difficult to define an exact state-of-the-art for vSLAM in non-stationary environments. For visual SLAM alone, there are famous systems that can be considered part of the the state-of-the-art systems like Vins-Mono [8], DSO [9] and ORB-SLAM2 [10] as described in [25]. However, there is no single approach comparable to those systems in the field of non-stationary SLAM. For

one, the problem of non-stationary SLAM still poses too many challenges to overcome. Second, the topic of perception and place recognition in non-stationary environments is a broad field, incorporating various environments with too many differences. There are numerous approaches to specific environments, even ones that already can be called robust. However, there is still much to do until all of those components can be incorporated into one system that is applicable in general while needing realistic amounts of memory and computational power to be ran in an autonomous robot. In the future, it is possible that there will be a system that combines building an exact map of the environment while predicting future changes and planning ahead, while being able to not get disoriented through varying amounts of highly dynamic objects. All that while maintaining a speed so that the robot is usable in real world applications.

5 Conclusion

This paper has tried to compile the state-of-the-art of visual SLAM (vSLAM) in non-stationary environments. It has shown the basic SLAM problem and the importance of vSLAM in recent years. The term non-stationary environment was defined, which incorporates different aspects of semi-static, dynamic and lifelong SLAM. In addition, a summary of what is needed to call a SLAM system *robust* or *solved* was given. Furthermore, different problems and challenges for vSLAM in non-stationary environments as well as current solutions to those problems were shown. The state-of-the-art of this topic can not be defined clearly. Most solutions to the examined problems solve specific problems and as such, there is no broad, general solution to most problems. Nevertheless, there are numerous promising approaches and robust solutions to many problems and bit by bit the systems will evolve as we enter the robust perception age [4]. Finally, SLAM still is necessary and a key technology which will be a much discussed and researched topic in the years to come.

References

[1] B. Siciliano and O. Khatib, *Springer handbook of robotics*. 2016, pp. 1–2227, ISBN: 9783319325521.

[2] K. Konolige and J. Bowman, “Towards lifelong visual maps,” in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009, pp. 1156–1163.

[3] J. Fossel, K. Tuyls, and J. Sturm, “2D-SDF-SLAM: A signed distance function based SLAM frontend for laser scanners,” in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 1949–1955.

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

[5] S. Yang, G. Fan, L. Bai, R. Li, and D. Li, “MGC-VSLAM: A meshing-based and geometric constraint VSLAM for dynamic indoor environments,” *IEEE Access*, vol. 8, pp. 81 007–81 021, 2020, ISSN: 21693536.

[6] J. Fuentes-Pacheco, J. Ruiz-Ascencio, and J. M. Rendón-Mancha, “Visual simultaneous localization and mapping: a survey,” *Artificial Intelligence Review*, vol. 43, no. 1, pp. 55–81, 2015, ISSN: 15737462.

[7] S. Lowry, N. Sunderhauf, P. Newman, J. J. Leonard, D. Cox, P. Corke, and M. J. Milford, “Visual Place Recognition: A Survey,” *IEEE Transactions on Robotics*, vol. 32, no. 1, pp. 1–19, 2016, ISSN: 15523098.

[8] T. Qin, P. Li, and S. Shen, "Vins-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018. DOI: 10.1109/TRO.2018.2853729.

[9] R. Wang, M. Schworer, and D. Cremers, "Stereo dso: Large-scale direct sparse visual odometry with stereo cameras," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct. 2017.

[10] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[11] M. Hentschel and B. Wagner, "An Adaptive Memory Model for Long-Term Navigation of Autonomous Mobile Robots," *Journal of Robotics*, vol. 2011, J. M. Dias, Ed., p. 506 245, 2011, ISSN: 1687-9600.

[12] T. Morris, F. Dayoub, P. Corke, G. Wyeth, and B. Upcroft, "Multiple map hypotheses for planning and navigating in non-stationary environments," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 2765–2770, 2014, ISSN: 10504729.

[13] X. Shi, D. Li, P. Zhao, Q. Tian, Y. Tian, Q. Long, C. Zhu, J. Song, F. Qiao, L. Song, Y. Guo, Z. Wang, Y. Zhang, B. Qin, W. Yang, F. Wang, R. H. Chan, and Q. She, "Are we ready for service robots? The OpenLORIS-scene datasets for lifelong SLAM," *arXiv*, 2019, ISSN: 23318422.

[14] M. Maimone, Y. Cheng, and L. Matthies, "Two years of visual odometry on the Mars Exploration Rovers," *Journal of Field Robotics*, vol. 24, no. 3, pp. 169–186, 2007, ISSN: 15564959.

[15] A. H. Hafez, M. Singh, K. M. Krishna, and C. V. Jawahar, "Visual localization in highly crowded urban environments," *IEEE International Conference on Intelligent Robots and Systems*, pp. 2778–2783, 2013, ISSN: 21530858.

[16] M. Labbé and F. Michaud, "Long-term online multi-session graph-based SPLAM with memory management," *Autonomous Robots*, vol. 42, no. 6, pp. 1133–1150, 2018, ISSN: 15737527.

[17] L. Kunze, N. Hawes, T. Duckett, M. Hanheide, and T. Krajnik, "Artificial intelligence for long-term robot autonomy: A survey," *arXiv*, vol. 3, no. 4, pp. 4023–4030, 2018, ISSN: 23318422.

[18] M. Milford and G. Wyeth, "Persistent navigation and mapping using a biologically inspired slam system," *The International Journal of Robotics Research*, vol. 29, no. 9, pp. 1131–1153, 2010.

[19] A. Ranganathan, S. Matsumoto, and D. Ilstrup, "Towards illumination invariance for visual localization," in *2013 IEEE International Conference on Robotics and Automation*, 2013, pp. 3791–3798.

[20] G. D. Tipaldi, D. Meyer-Delius, and W. Burgard, "Lifelong localization in changing environments," *International Journal of Robotics Research*, vol. 32, no. 14, pp. 1662–1678, 2013, ISSN: 02783649.

[21] T. Naseer, W. Burgard, and C. Stachniss, "Robust Visual Localization Across Seasons," *IEEE Transactions on Robotics*, vol. 34, no. 2, pp. 289–302, 2018, ISSN: 15523098.

[22] T. Krajnik, J. P. Fentanes, J. M. Santos, and T. Duckett, "FreMEn: Frequency map enhancement for long-term mobile robot autonomy in changing environments," *IEEE Transactions on Robotics*, vol. 33, no. 4, pp. 964–977, 2017, ISSN: 15523098.

[23] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," *IEEE International Conference on Intelligent Robots and Systems*, pp. 573–580, 2012, ISSN: 21530858.

[24] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013, ISSN: 02783649.

[25] M. Servières, V. Renaudin, A. Dupuis, and N. Antigny, "Visual and Visual-Inertial SLAM: State of the Art, Classification, and Experimental Benchmarking," *Journal of Sensors*, vol. 2021, 2021, ISSN: 16877268. DOI: 10.1155/2021/2054828.