ArucoSLAM: Enhancing Visual SLAM with Fiducial Marker Integration

André Pardo Maurell[1 [0000−1111−2222−3333]], Kristofer Stift Kappel[1 [0000−0002−9124−8540]], Jardel dos Santos Dyonisio[1 [0009−0003−7722−7203]], Rodrigo da Silva Guerra[1 [0000−0003−4011−0901]], and Paulo Lilles Jorge Drews-Jr[1 [0000−0002−7519−0502]]

Center for Computational Science - C3,
Universidade Federal do Rio Grande - FURG, Brazil
{andremaurell3,kris,jardel.dyonisio,rodrigo.guerra,paulodrews}@furg.br
http://www.c3.furg.br

Abstract. SLAM (Simultaneous Localization and Mapping) is a technique that enables a robot to construct a map of its environment while simultaneously estimating its own position within that map. Accurate evaluation of SLAM algorithms in mobile robotics is essential to ensure the effectiveness and safety of autonomous navigation. However, traditional validation methods often face limitations in spatial coverage and environmental data specificity. This project presents a method that incorporates fiducial markers, specifically ArUco tags, strategically placed throughout the robot's operational environment, called ArucoSLAM. Using an onboard monocular camera, the system detects these markers and leverages their relative positions to both the world and the robot to correct accumulated errors during the SLAM process. Additionally, this enables loop closure through tag detection, addressing one of the major challenges in Visual SLAM. The primary goal of this approach is to enhance navigation accuracy by overcoming key limitations of traditional SLAM systems that do not employ environmental reference tags. Experimental results demonstrated that this approach improves map generation when compared to OrbSLAM3 operating without markers, as well as to scenarios where markers are present only for visual enrichment.

Keywords: Visual SLAM · Fiducial Tags · ArUco

1 Introduction

Simultaneous Localization and Mapping (SLAM) is a technology in robotics and computer vision that enables an agent to create a map of an unknown environment while simultaneously tracking its own location within that environment [1].

The main objective of SLAM is to construct a reliable and enable accurate localization in real-time, particularly in scenarios where prior environmental information is not available [16].

OrbSLAM3 [2] is one of the state-of-the-art visual SLAM solutions currently available, supporting monocular, stereo, and RGB-D scenarios. It introduces several improvements over ORB-SLAM2 [9], including more efficient keyframe management, enhanced support for depth sensors, and better relocalization in complex environments.

One of the main features of SLAM is Loop-Closure, which enables the algorithm to identify previous visited regions, allowing the correction of the map and relocalization of the robot [7].

In homogeneous environments, where textures and visual features are repetitive or lack distinctiveness, visual SLAM faces significant challenges. Since consecutive frames tend to be very similar, the extracted keypoints show little to no variation, making it difficult to recognize unique patterns and compromising the loop-closure stage [5].

In this context, incorrect loop detection may occur, leading to false positives that mistakenly indicate a previously visited location. This results in inconsistencies in the generated map and the robot's current localization [6].

Therefore, in environments with low texture variation, OrbSLAM3 struggles to identify keyframes and, consequently, has difficulties extracting keypoints. This causes the system to lose its ability to localize reliably, leading to trajectory deviation or loss within the environment [2].

To overcome these challenges, we propose to combine fiducial markers to improve SLAM robustness. Fiducial markers are visual patterns that help computer vision systems determine positions and orientations. The most common are ArUcos and AprilTags.

Therefore, ArUcos were selected for this work due to their ease of detection under different lighting conditions, their support in the OpenCV library, and their low processing requirements, making them a practical choice for embedded systems that often face varying light conditions. Moreover, integrating fiducial markers such as ArUcos into OrbSLAM3 is a viable approach to overcoming these limitations, which justifies the choice of this algorithm as the foundation for the development of ArucoSLAM, the method proposed in this work.

The experiments tested ArucoSLAM against OrbSLAM3 mapping an environment with ArUcos and relocalization, with ArUcos without relocalization and without ArUco tags.

Nonetheless, the main motivation for this work is to explore and develop a system capable of correcting accumulated SLAM errors, enhancing system robustness and map quality, assisting in robot relocalization, and generating accurate maps through the detection of fiducial markers.

The remainder of the work is divided as follows: Section 2 dives into related works and the state of the art. Section 3 explains ArucoSLAM. Section 4 details the experiments and discusses the results. Lastly, Section 5 draws conclusions and future works.

2 Related Works

There are algorithms in the literature that rely on ArUcos or other types of tags to perform SLAM [8, 13–15]. However, there is a need to combine keypoint detection and the use of fiducial markers to enhance mapping accuracy and robustness in complex environments.

UcoSLAM [11] is an algorithm that combines keyframe detection with the identification of ArUco markers to perform simultaneous mapping and localization. Its main advantage lies in its ability to operate in environments where visual features may be sparse or indistinct, using the markers to assist in robot relocalization and trajectory correction. This combination allows UcoSLAM to maintain accurate localization even under challenging conditions.

However, UcoSLAM uses an earlier version of ORB-SLAM (ORB-SLAM2 [9]), which does not benefit from the latest improvements introduced in Orb-SLAM3 [2]. Additionally, it is a system developed for ROS1, which limits its compatibility with more modern architectures, such as ROS2.

SPM-SLAM (Semantic Planar Marker SLAM) [10] is an algorithm that integrates planar marker detection and semantic information to improve SLAM performance. In addition to using keypoints for mapping, SPM-SLAM incorporates planar markers, such as ArUcos, to add a level of semantic understanding to the generated map, allowing the system not only to localize the robot but also to interpret its surrounding environment.

Despite its advantages, SPM-SLAM has the drawback of increased computational load due to the integration of semantic information, which can result in slower performance compared to approaches that use only keyframes/keypoints or simple markers. Furthermore, its reliance on additional semantic data may make it less effective in dynamic or poorly structured environments where such information is not readily available.

TagSLAM [12] is an approach that focuses exclusively on the use of fiducial tags, such as AprilTags, to perform SLAM. Unlike other methods that combine keyframes with markers, TagSLAM relies almost entirely on markers to map the environment and navigate. This approach is particularly effective in controlled environments, where visual features are scarce and marker-based mapping can be more reliable.

However, TagSLAM faces significant challenges in large or outdoor environments, as it depends entirely on the presence of fiducial tags distributed throughout the environment to function properly. In uncontrolled scenarios, where placing a large number of tags is not feasible, its application becomes limited.

3 ArucoSLAM

The main goal of ArucoSLAM is to improve current visual SLAM algorithms by combining OrbSLAM3 with fiducial tag detection. In this context, the use of fiducial tags is essential to assist with relocalization and loop closure, improving robustness and reliability.

ArucoSLAM combines classical visual SLAM methods with fiducial marker detection, such as ArUco tags. However, unlike other conventional methods, ArucoSLAM uses only an onboard monocular camera. This integration aims to overcome limitations faced by traditional systems in challenging environments, such as homogeneous areas or those with few distinct visual features.

An ArUco marker is a synthetic square marker composed of a wide black border and an internal binary matrix that determines its identifier (ID). The black border facilitates its quick detection in the image, and the binary encoding enables identification and the application of detection and error correction techniques. The size of the marker determines the size of the internal matrix. For example, a 4×4 marker is composed of 16 bits.

During navigation, the robot may face challenges such as environmental changes, lighting variations, or a lack of sufficient visual features. In such cases, fiducial tags act as fixed reference points, allowing the algorithm to accurately relocalize the robot. Upon detecting an ArUco marker, the system can calculate the robot's estimated position and relocalize it within the environment.

Moreover, fiducial tags play an important role in loop closure detection, which is the process of correcting accumulated errors along the robot's trajectory. The combination of keypoints and fiducial tags allows the system to recognize when the robot returns to a previously visited location. In contrast, when only keypoints are used, there is a greater risk of false positives in environments with similar visual features, leading the robot to mistakenly identify locations as loop closures when they are not.

In simulated environments or other real-world environments with limited visual information due to repetitive textures, the use of fiducial tags not only aids in loop closure and relocalization but also enriches the visual environment, making it easier for keyframes to be acquired during navigation and consequently allowing new keypoints to be extracted.

Therefore, ArucoSLAM uses keypoints to build the map and estimate the robot's position, while ArUco markers are used to enhance the accuracy of relocalization. Loop closure detection is performed using both keypoints and ArUco markers, allowing for the correction of trajectory errors and the generation of a more consistent map reflecting the robot's path.

The detection of ArUco markers is carried out in three stages:

1. **Capture and Preprocessing:** The images captured by the robot's cameras are processed using preprocessing techniques such as contrast adjustment and noise reduction to improve tag detection.
2. **Tag Detection with OpenCV and ArUco:** The `aruco` library from OpenCV was used to detect fiducial markers in the images. The `aruco` library is designed to quickly identify tags and provide information about their position and orientation relative to the camera.
3. **Pose Estimation:** Once detected, the fiducial tags provide information about their pose relative to the camera. The pose is represented by a homogeneous matrix that describes the transformation between the tag's coordinate

system and the camera's coordinate system. The `aruco` library returns this pose as a translation vector and a rotation matrix.

During navigation, the robot may get lost due to accumulated errors or challenges within the environment. To prevent this, ArucoSLAM uses ArUco detection, which is integrated into OrbSLAM3 through a ROS2-developed wrapper. Relocalization occurs as follows:

1. **Detection of ArUcos**: The wrapper detects ArUco markers present in the scene and calculates the robot's estimated position in the global coordinate system using homogeneous transforms.
2. **Integration via ros1_bridge**: Through the *ros1_bridge*, the position published by the wrapper on a ROS2 topic can also be read by ROS1, since the *ros_bridge* serves to translate topics between the two middleware systems (ROS1 and ROS2). The position is then used by the native functions of OrbSLAM3, acting directly on the loop closure process.
3. **Pose Correction**: When the robot's position is recalculated based on fiducial tags, its location in the environment is also adjusted.

Loop closure detection allows the correction of accumulated errors in the robot's trajectory and ensures a consistent map. The use of ArUco markers at this stage improves the system's reliability.

Systems based solely on keypoints may identify false loop closures in homogeneous environments or those with repetitive textures, which compromises the consistency of the map. Besides, since each ArUco marker is unique, real loops are more easily identified, allowing the robot to relocalize itself.

To integrate these improvements into the SLAM system, tf2 transforms were used, which are an essential part of the ROS2 library for handling transformations between different coordinate systems. The tf2 library allows the conversion between the ArUco coordinates and the global coordinates using homogeneous matrices to represent the transformations.

4 Experiments and Results

4.1 Navigation

The vehicle's navigation was carried out using the teach and repeat algorithm [4], primarily tested on the SHARK robotic base [3]. This method allows the vehicle to learn a specific path during the teaching phase and, subsequently, be able to repeat it. After teaching the desired navigation path to the vehicle, it will traverse the environment following the learned route.

During this navigation period, the vehicle will also run ArucoSLAM. This integration enables robust navigation, ensuring that the vehicle attempts to correct the accumulated errors throughout the navigation process.

4.2 Environment

In this work, a simulated environment representing a factory was used, where a vehicle operates autonomously based on paths previously taught to it. The simulated environment offers the advantage of repeatability, as test batches can be run repeatedly and in a controlled manner without any limitations, which facilitates the development and validation of SLAM algorithms.

The idea behind using this environment involves the development of a robot capable of autonomously navigating a factory, transporting loads between the docking stations within the environment.

4.3 Vehicle

The vehicle used in this simulated environment relies solely on a monocular camera, which provides all the necessary data for navigation and environment mapping. However, it is also equipped with a stereo camera, an IMU (Inertial Measurement Unit), and a lateral monocular camera, which were not used during the experiments

4.4 Fiducial Tags

The markers will be placed at strategic points in the environment, specifically at the vehicle's parking docks. The ArUco markers will be generated using an algorithm that follows the steps described below:

1. **Generation of ArUco PNGs**: Initially, the ArUco markers will be generated as PNG files. These files contain the marker images, which will later be positioned in the environment.
2. **Creation of the .json file**: After generating the PNGs, a file in `.json` format will be created to store the position and orientation information of each ArUco in the simulated environment, represented by the set $M = \{m\}$. Each marker m is described by:
   - s: the length of the marker's side.
   - $M \in SE(3)$: the pose of the marker.
   - $\mathbf{x}_i \in \mathbb{R}^3$, $i = 1, \ldots, 4$: the coordinates of the four corners of the marker, defined with respect to the center of the marker.
3. **Generation of the .sdf file**: Finally, the `.json` file will be used to generate a `.sdf` (Simulation Description Format) file, which will include all ArUco markers in the desired positions and orientations within the simulated world.

This process makes it easier to position the markers accurately and automatically, facilitating the creation of diverse and realistic scenarios, which helps in running test batches for the algorithm.

4.5 Detection of ArUcos

To detect ArUcos, the OpenCV library was used. However, although it helps with the detection and identification of the markers, it does not provide methods for locating them in the world. Therefore, since ArucoSLAM uses the ROS2 middleware, the use of TF2 was necessary to correct the location of the ArUcos in the world, as well as the robot's position in the world based on them.

The ArUco detection step within OrbSLAM3 was based on the ArUco detection module of the OpenCV library, configured to recognize a specific predefined marker dictionary. Furthermore, a wrapper was developed in ROS2 to handle the detection and send the information to OrbSLAM3, simplifying the detection process.

Figure 1 shows an ArUco marker being detected simultaneously as OrbSLAM3 extracts keypoints from the current frame, demonstrating that both processes can operate concurrently without issues. It is worth noting that the two images appear different, since OrbSLAM3 processes grayscale images and adjusts the camera resolution to better detect keyframes.

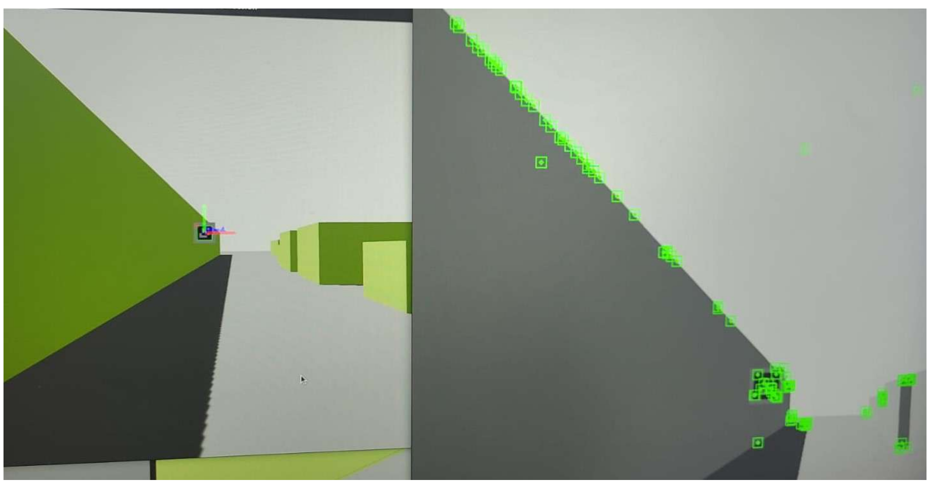


Fig. 1: ArUcos detected by OrbSLAM3

The robot's localization based on ArUco markers for loop-closure and relocalization was performed using detection and homogeneous transformations. The resulting position of the robot in the world, based on ArUco detection, is sufficient for SLAM requirements.

4.6 Ground Truth

Ground truth is the actual reference used to evaluate the accuracy of algorithms, if the robot follows a path, the ground truth will contain the exact route taken by the vehicle. Therefore, by using the teach and repeat algorithm it is possible to obtain the exact path the robot follows during its route. This allows for a comparison between ArucoSLAM and ORBSLAM3.

Furthermore, Figure 2 shows in red the absolute path followed by the robot. Moreover, Table 1 shows the results obtained from the comparison between the robot's ground truth position and the position estimated by the ArucoSLAM localization method.

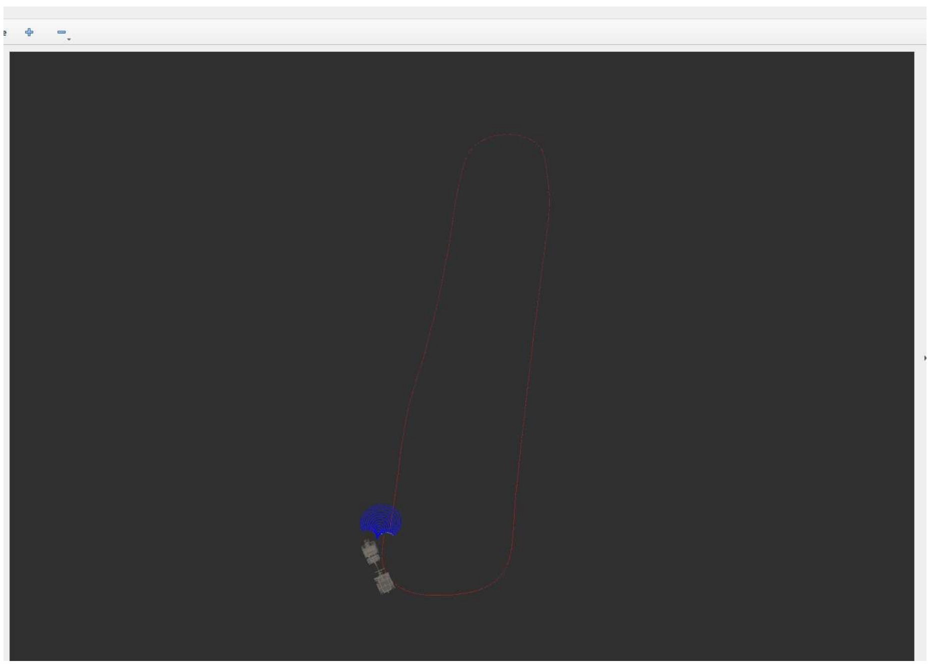


Fig. 2: Ground Truth of the path followed by the robot

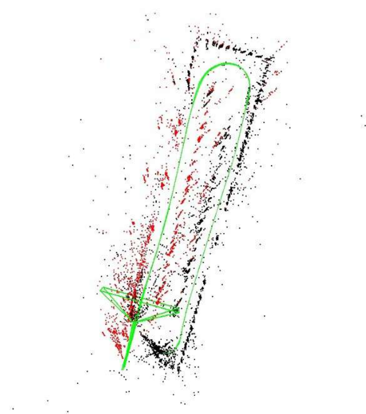
Table 1: Comparison between ground truth and ArucoSLAM position

Axis	Ground Truth	ArucoSLAM	Absolute error
x	0.048	0.040	0.008
y	0.114	0.093	0.021
z	0.301	0.301	0.000

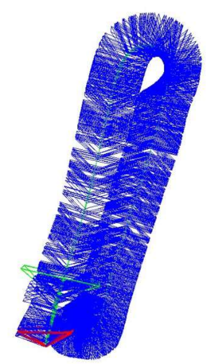
4.7 Experiment 1 - Mapping Using ArUcos and Relocalization

The experiment carried out to validate ArucoSLAM involved running the navigation along the path previously taught to the robot, but with markers added throughout the environment being navigated.

The map is generated from the keyframes detected along the path and the keypoints identified in each frame. As shown in Figure 3a, in green we have the path the robot believes it followed, in black the keypoints that were saved by the SLAM, and in red the keypoints currently being detected by the camera.



(a) Mapping with ArUcos and relocalization



(b) Keyframes acquired along the traveled path

Fig. 3: Experiments with ArUcos and Relocalization

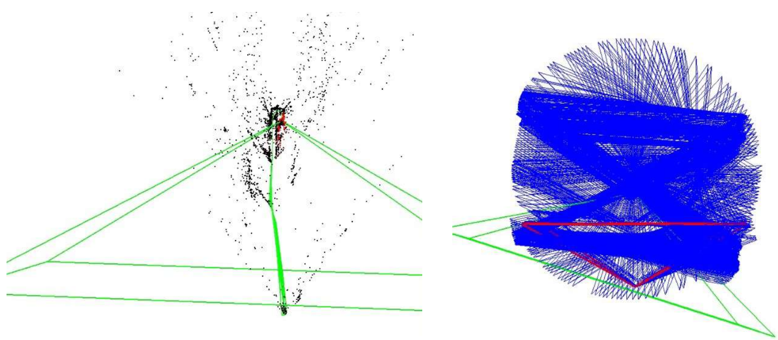
In Figure 3b, the keyframes acquired during the navigation path are shown in blue. These keyframes are represented on the map by triangles indicating the camera's field of view, where the blue triangles correspond to keyframes saved in DBoW2. It is worth noting that the image might appear somewhat blurred since the corridors navigated by the vehicle are very close together, causing the keyframes from both corridors to overlap on the map.

However, it can be observed that the saved keyframes follow an ordered direction, aligning with the format of the path traveled, which was defined by the ground truth. In Figure 3a, the result of the generated map is shown, with the keypoints represented as black dots and the traveled path as a green line. This illustrates that the green path generated by ArucoSLAM closely resembles the one defined by the ground truth, even though it may not be completely optimized.

4.8 Experiment 2 - Mapping with ArUcos in the Environment Without Relocalization

In this experiment, the markers remained in the environment where the robot would navigate. However, only OrbSLAM3 was running. The markers were left in place to provide the same visual richness in the environment as in the previous experiment, but without using the markers for relocalization accuracy.

In Figure 4b, the blue keyframes acquired during the navigation path are shown once again. It is also evident that the keyframes do not follow an ordered direction nor the path defined by the ground truth, indicating that relocalization based on OrbSLAM3's keypoints was not successful. Furthermore, in Figure 4a, we see the resulting map showing the keypoints extracted during the route. This shows that the robot was able to traverse the designated path.However, at the upper part of the trajectory, where a turn requiring relocalization occurs, it ended up getting lost and was unable to generate the map with the expected results.



(a) Mapping with ArUcos without relocalization

(b) Keyframes acquired along the traveled path

Fig. 4: Experiments with ArUcos and without Relocalization

4.9 Experiment 3 - Mapping Without ArUcos

In the mapping without the aid of any ArUco markers, the robot was able to navigate reasonably well up until the previously mentioned curve. However, due to a lack of visual information, it ended up getting lost and was no longer able to find any existing keyframes. Consequently, it lost track of the predefined route, resulting in an empty and information-less map.

4.10 Discussion

Based on the visual results, which illustrated each of the experiments conducted, it can be assumed that ArucoSLAM ends up generating a map similar to the proposed ground truth, as it benefits from the use of ArUco markers to assist with loop closure and relocalization.

It is also noticeable how important visual enrichment is in environments with limited visual information, such as simulated ones, where there is a lack of shadows and little texture on walls and objects.

However, since OrbSLAM3 is currently considered the state-of-the-art in Visual SLAM, it is worth noting that the experiments were conducted in a homogeneous environment, deliberately designed to expose the weakest aspects of OrbSLAM3, and to show that ArucoSLAM can help address those limitation.

5 Conclusion

The proposed work aimed to develop a system capable of enhancing the Visual SLAM performance of OrbSLAM3 through the integration of fiducial tags, seeking improvements in the maps generated by the developed algorithm. This approach focuses on improving relocalization and loop closure, especially in scenarios where traditional keypoints present limitations.

The experiments conducted demonstrated that the use of fiducial markers can provide estimated information about the robot's position, complementing the estimates generated by OrbSLAM3's keyframes. This integration helped correct deviations in the estimated trajectory, bringing the system closer to the ground truth. However, there are still technical aspects to be refined to solidify the practical application of the proposed method.

For future work, it is necessary to fully implement the fusion of fiducial marker data with keyframes in the DBoW2 database, so that marker prioritization occurs dynamically and is optimized in real time. Additionally, implementing the method in Stereo format and integrating the IMU could further improve the obtained results.

Therefore, the anticipated advancements in the continuation of this work will contribute to the development of a reliable, efficient, and adaptable visual SLAM system, applicable to various fields such as mobile robotics, autonomous vehicles, and navigation in complex environments.

Acknowledgments. This work was partly supported by FINEP, ANP-PRH22, FAURG, FAPERGS and CNPQ.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. 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 Transactions on robotics (2016)
2. Campos, C., Elvira, R., Gómez, J.J., Montiel, J.M.M., Tardós, J.D.: ORB-SLAM3: An accurate open-source library for visual, visual-inertial and multi-map SLAM. IEEE Transactions on Robotics (2021)
3. Dias, N.F., Dyonisio, J.D.S., Quadros, L.F.M., Dorneles, G.A., Prado, J.J.P., Rocha, M.Z., Bicho, M.C., Lemos, J.F.S.S., Bezerra, H.C., Teixeira, C.N.S., Maurell, I.P., Guerra, R.S., Drews-Jr., P.L.J.: SHARK: Stable hoverboard-driven autonomous robot kit. In: Proceedings of the V Brazilian Humanoid Robot Workshop (BRAHUR) and VI Brazilian Workshop on Service Robotics (BRASERO) (2024)
4. Dos Santos Dyonisio, J., Felipe Milczarek, L., Freitas Dias, N., Loi Brião, S., Amaral Dorneles, G., Pardo Maurell, I., Miranda Pinheiro, P., Da Silva Guerra, R., Lilles Jorge Drews-Jr, P.: Teach and repeat for path planning using bézicr curves. In: 2024 Brazilian Symposium on Robotics (SBR), and 2024 Workshop on Robotics in Education (WRE) (2024)
5. Fang, B., Zhan, Z.: A visual slam method based on point-line fusion in weak-matching scene. International Journal of Advanced Robotic Systems (2020)
6. Granström, K., Schön, T.B.: Learning to close the loop from 3d point clouds. In: International Conference on Intelligent Robots and Systems (2010)
7. Iegawa, F.N., Botelho, W.T., Santos, T.d., Pimentel, E.P., Yamamoto, F.S.: Loop closure detection in visual slam based on convolutional neural network. In: ITNG 2023 20th International Conference on Information Technology-New Generations (2023)
8. Li, X., Patel, S., Stronzek-Pfeifer, D., Büskens, C.: Indoor localization for an autonomous model car: A marker-based multi-sensor fusion framework. In: 19th International Conference on Automation Science and Engineering (CASE) (2023)
9. Mur-Artal, R., Montiel, J.M.M., Tardós, J.D.: ORB-SLAM: a versatile and accurate monocular SLAM system. IEEE Transactions on Robotics (2015)
10. Muñoz-Salinas, R., Marín-Jimenez, M.J., Medina-Carnicer, R.: Spm-slam: Simultaneous localization and mapping with squared planar markers. Pattern Recognition (2019)
11. Muñoz-Salinas, R., Medina-Carnicer, R.: Ucoslam: Simultaneous localization and mapping by fusion of keypoints and squared planar markers. Pattern Recognition (2020)
12. Pfrommer, B.: Tagslam: Flexible slam with tags. https://sifrobot.com/clinical-apps/simultaneous-localization-and-mapping-slam/ (2018)
13. Scheideman, S., Ray, N., Zhang, H.: A flexible method for performance evaluation of robot localization. In: 2020 IEEE International Conference on Robotics and Automation (ICRA) (2020)
14. Tanaka, H., Ogata, K.: Fiducial marker and camera-based passive high-accuracy 6-dof tracking for mobile objects (2023)
15. Wang, Y., Ji, Y., Liu, D., Tamura, Y., Tsuchiya, H., Yamashita, A., Asama, H.: Acmarker: Acoustic camera-based fiducial marker system in underwater environment. IEEE Robotics and Automation Letters (2020)
16. Xia, L., Cui, J., Shen, R., Xu, X., Gao, Y., Li, X.: A survey of image semantics-based visual simultaneous localization and mapping: Application-oriented solutions to autonomous navigation of mobile robots. International Journal of Advanced Robotic Systems (2020)