Visual Recognition and Localization of Industrial Robots Based on SLAM Algorithm

Wei Cui[1]*, Yuefan Zhao[2], Litao Sun[3]
Department of Electrical Engineering, Hebei Institute of Mechanical and Electrical Technology, Hebei 054000, China[1, 3]
Department of Mechanical Engineering, Hebei Institute of Mechanical and Electrical Technology, Hebei 054000, China[2]

Abstract—The front-end feature matching module of traditional SLAM systems is characterized by sparse or dense feature points, it is difficult to generate accurate camera trajectory and scene reconstruction results, in response to this problem, the author studied a fast reconstruction algorithm for any path based on V-SLAM, by using improved feature matching algorithms to accurately match feature points, the accuracy of scene sparse reconstruction and camera trajectory recovery has been improved, the backend optimization thread adopts segmented optimization matching to reduce the computational burden of reconstruction, and the performance of the V-SLAM system was improved through parallel processing, the matching results and camera trajectory error comparison results showed that the improved V-SLAM algorithm can quickly recover camera trajectory and scene reconstruction, with the development of multi-sensor collaborative coupling and multi view fusion technology, the V-SLAM method proposed by the author can add virtual 3D objects to real scenes, and the V-SLAM system can extract feature points in the screen in real-time and detect planar objects in the scene, ensure that multiple virtual objects in the scene meet geometric consistency with the actual scene, in the experiment, two objects were added to the virtual scene, users can interactively scale objects and add them without being affected by camera movements, ensuring consistency between objects and the real scene.

Keywords—SLAM algorithm; industrial robot; visual recognition; location

I. INTRODUCTION

The positioning and map creation of mobile robots are hot research topics in the field of robotics, and are also important links in navigation. There are already some practical solutions for autonomous localization of robots in known environments and map creation of known robot positions [1]. However, in many environments, robots cannot use global positioning systems for localization, and obtaining a map of the robot's working environment in advance is also difficult, or even impossible. At this point, the robot needs to create a map in a completely unknown environment with its own position uncertain, and simultaneously use the map for autonomous positioning and navigation, as well as positioning and mapping [2]. Assuming that the robot is moving in a completely unknown environment, executing control commands and observing the characteristics of the environment, both the control and observation quantities will be affected by noise interference, SLAM is the restoration of robot path and environmental feature information from a series of noisy variables. If the path of the robot is determined (such as GPS positioning), then it is a problem of building a map, where the position of the target in the environment is estimated using an independent filter; When the robot's path is unknown, the robot's path is related to the error of the map, so, the state information and map information of the robot must be estimated simultaneously. The SLAM problem includes four basic aspects: (1) How to describe the environment, the representation method of environmental maps; (2) How to obtain environmental information, robots roam the environment and record sensor perception data, which involves the problem of robot localization and environmental feature extraction; (3) How to represent the obtained environmental information and update the map based on the environmental information requires addressing the description and processing methods of uncertain information; (4) Develop stable and reliable SLAM methods. The SLAM algorithm has many important attributes that affect the uncertainty in map features and robot position estimation, including the convergence of state estimation, the data consistency of the estimation process, and the computational complexity of updating the state covariance matrix. Convergence of state estimation: As the number of observations increases, in order to reduce the uncertainty of map estimation to a limited range, the convergence of state estimation must be maintained. Firstly, the accuracy of the map is related to the position accuracy of the robot when the first environmental feature is observed, so it is necessary to ensure the convergence of the state matrix when the first environmental feature is observed [3-4]; Secondly, the update of the state covariance matrix after each measurement must be convergent relative to the matrix before the update [5]; Finally, in extreme cases, as the number of observations increases, the feature estimation becomes completely correlated. As long as these are ensured, the relationship between map features is completely determined. Consistency of data association: Data association is the key to data fusion. In the SLAM process, data association mainly completes two tasks: Detection of new environmental features and feature matching. If the data association is not accurate, it will lead to filter divergence, which has a particularly prominent impact on EKF based methods. In order to maintain the consistency of SLAM, it is necessary to update the state covariance matrix. The observation of the environment is relative to robots, so any errors in robot estimation are related to errors in map estimation. In the absence of information about the robot's position and its environmental characteristics, in order to keep the error of system state estimation within a limited range, it is necessary to maintain the consistency of state estimation. Therefore, it is necessary to update the covariance matrix between the robot states and environmental features in real-time. The scene perception effect of traditional SLAM relies on

*Corresponding Author.

radar or laser sensors and is easily affected by acquisition noise, so sometimes the scene reconstruction and perception effect are not ideal. When the scene contains objects with too single or too complex features, inter frame feature matching is too sparse or too dense, uneven allocation, and affects scene perception, bringing additional computational pressure to backend scene optimization [6]. The author uses visual sensors as the main sensing device to estimate the camera position and the V of the environment in which it is located_ SLAM (Vision based SLAM) group technology, by improving the matching algorithm and backend optimization strategy of SLAM front-end threads, V-SLAM has good scalability, flexibility, and can be applied to large-scale scene perception. The improved V-SLAM can add virtual objects to the scene in real-time, providing users with an intuitive and real-time visual experience, expressing AR effects, and obtaining real-time device parameters and posture determination play an important role in the SLAM system. The global positioning system (GPS) can greatly reduce its positioning accuracy in indoor environments or when signals are severely obstructed; Placing sensors that receive signals in specific indoor environments can obtain rich data and greatly improve positioning accuracy, but it requires pre-arranged usage scenarios and does not have good flexibility and adaptability; Lidar systems are difficult to popularize due to their high cost. V-SLAM has significant advantages in scenario adaptability, scalability, and low cost [7]. V-SLAM has natural advantages in augmented reality positioning solutions, the improved V-SLAM proposed by the author can add virtual objects while constructing the three-dimensional scene of the environment, allowing users to add selected objects to the perception scene. Currently, there are many types of SLAM algorithms, ORB- SLAM (OKVIS, viorbvins- SLAM)、DTAM (dense tracking and mapping) and LSD-SLAM (large scale direct monocular SLAM). ORB-SLAM is a SLAM algorithm based on keyframe BA, using the PTAM (pickuptruck access method) system framework. ORB SLAM uses ORB feature points for matching to improve the accuracy of bundle adjustment. DTAMLo and LSD-SLAM7 solve for motion by directly comparing pixel colors between images, these two local pixel based SLAMs have good reconstruction performance in noisy environments. They can restore 3D scenes in real-time, but LSD SLAM can only restore depth maps of semi dense scenes; Due to the need for DTAM to process the depth information of each pixel to establish a dense map, the computational complexity is high and the scene perception efficiency is low. By constructing a feature transformation network, which is composed of multiple similar affine transformation, the author matches feature points to improve the matching accuracy and efficiency, thus reducing the calculation time of front-end threads [8-9]. Fig. 1 is the flow chart of the overall visual SLAM.

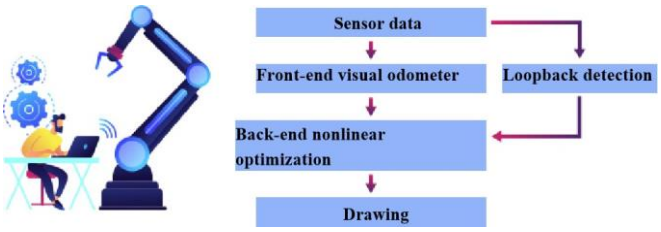

Fig. 1. Overall visual SLAM flowchart.

II. METHODS

The V-SLAM system mainly includes front-end and back-end threads, among which the front-end threads mainly pass through visual sensors such as cameras, input the acquired data into the SLAM system for initialization, put the SLAM system into tracking state, and output real-time camera pose and scene point cloud; The backend thread mainly optimizes the thread, and there may be pose deviation or scale drift during camera motion or scene switching, resulting in error accumulation, when the error accumulates to a certain extent, it will cause the algorithm module to stop working, the backend thread uses scene loop detection to correct the drift phenomenon that occurs during the camera movement process, the author uses local or global optimization to reduce the error accumulation of SLAM system [10].

A. Front End Thread Design for V-SLAM

The SLAM system initializes the system by detecting a certain number of feature points and enters camera tracking mode, the tracking stage mainly includes motion between images and scenes, image feature extraction, feature matching, and reconstruction from 2D images to 3D scenes. Visual sensors are used in unknown environments, perceiving the surrounding environment by capturing continuous environmental images, while constructing a scene using a three-dimensional sparse point cloud, with the world coordinate system feature point X; Projection to the image coordinate system, which includes the transformation from the world coordinate system to the camera coordinate system and the transformation from the camera coordinate system to the imaging coordinate system. In the world coordinate system, the feature points Xn of the environment [11-12]. The motion parameter during the shooting process in three-dimensional coordinates $[x_n, y_n, z_n]^T$ can be expressed as $C_1,...,C_i,...,C_n$, each camera motion parameter contains 3x3 camera rotation matrix Rn and position offset pn. Convert the feature point Xn in the world coordinate system to the camera's local coordinate system as shown in Eq. (1):

$$\begin{bmatrix} x_{cn} \\ y_{cn} \\ z_{cn} \end{bmatrix} = R_n \left(\begin{bmatrix} x_n \\ y_n \\ z_n \end{bmatrix} - p_n \right) \tag{1}$$

The camera projects the 3D point $[x_{cn}, y_{cn}, z_{cn}]^T$ in the camera coordinate system to the image point $[x, y]^T$ in the 2D coordinate system through perspective changes:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \left[\frac{F_x x_{cn}}{z_{cn}} + c_x, \frac{F_y y_{cn}}{z_{cn}} + c_y \right]^T \tag{2}$$

Among them, c_x, c_y represent the position of the lens optical center in the image, F_x, F_y represents the focal length

of the image along the x and y axes, and (x, y) represents the pixel position.

The environmental structure feature points in the world coordinate system are projected onto the image coordinate system, and each frame of the image input to the camera sensor is used for feature extraction and feature point matching between adjacent frames, feature point extraction and matching provide the data required for SLAM algorithm processing, which is related to the quality of camera path reconstruction and the accuracy of scene perception. High precision feature point matching algorithms are beneficial for reducing error accumulation during camera motion [13].

In order to generate accurate camera paths and high-quality scene perception, the V-SLAM system requires continuous feature point matching between hundreds or even thousands of frames of images. The feature matching algorithm proposed by the author ensures global optimization and has a certain degree of robustness through efficient region matching; At the same time, it also has good matching performance in scenes with fewer feature points or scenes where the captured image is blurry due to strong shaking of the lens [14-15].

The camera obtains consecutive frames $I_1,...,I_i,...,I_n$, using the image of the region where a feature point is located in frame Ii as a template, adjacent frame image I_{i+1} represents the image to be matched, and $W(I_i)$ represents the total change in the i-th frame image:

$$v_m = \max_{s \in N(m)} |I_i(m) - I_i(s)| \tag{3}$$

$$W(I_i) = \sum_{m \in I_i} |v_m| \tag{4}$$

Among them, the maximum difference between pixel point m and its adjacent eight pixels N (m) is taken as the variation of m. Traditional image matching algorithms first extract feature points from two consecutive frames of images, calculate the distance between feature points, and use threshold feature points as matching feature points between the detected image frames. The matching algorithm proposed by the author considers that the continuous frame images taken by the camera have little change, and a feature point image in the image frame Ii is mapped to the next frame image I_{i+1} through a certain affine transformation [16].

B. VSLAM Backend Thread Optimization

In the V-SLAM system, bundle adjustment is the core task of backend optimization. In order to reduce the cumulative error of the SLAM system, the backend threads of the SLAM system include local bundle adjustment (LBA) and global bundle adjustment (GBA) 24. LBA mainly optimizes the local scenes and camera keyframes generated during the process, while GBA optimizes all image frames and landmark pose features. The author proposes an improved LBA and segmented GBA optimization strategy. Once a new keyframe is added in the scene, LBA optimization is triggered, the feature points and camera trajectories of the current keyframe and previous frames within a certain time range are processed using improved segmented GBA [17-18].

In traditional SLAM systems, the key frame count for LBA optimization is 70-85, with a maximum of 100 frames, there is a significant consumption of redundant storage in system computing and storage. When the V-SLAM system processes captured video clips, when the visible two-dimensional feature points in a certain image frame meet the set values, they are labeled and added to the keyframe set. In local scenes, LBA optimizes the visible feature points in these keyframes. The traditional LBA optimization method involves the appearance of optimized 3D feature points in newly added keyframes, resulting in repeated optimization of visible feature points and keyframes in the scene, resulting in a large amount of redundant computation and memory consumption, greatly reducing the efficiency of SLAM backend processing threads. In the process of generating local environment, when high-precision local scene landmarks are detected, the author defines them as fixed landmarks, which can be used as reference landmarks to make other keyframes or scenes reference their positions to estimate coordinates. The same visible feature point does not need to be optimized multiple times, as the accumulation of system errors reduces the accuracy of camera trajectory and landmark coordinates. The backend system records the number of optimizations for visible feature points in each keyframe, when the point has been optimized 10 times, it is marked as a reference landmark, and LBA will no longer optimize the landmark and can estimate the camera pose of the current frame using this point. If there are visible reference landmarks in the newly added keyframes, LBA will no longer optimize them, thereby reducing the number of visible points for local optimization. For large scenes with many keyframes, when the keyframes exceed 120, the average keyframes in local LBA optimization range from 50 to 65, reducing the computational burden of the system and improving the efficiency and accuracy of local BA optimization.

In the backend optimization of V-SLAM system, GBA optimizes all global keyframes and visible feature points, resulting in high computational complexity and memory consumption, the number of optimizations far exceeds that of LBA. For the accumulation of errors in GBA optimization, the author proposes a segmented optimization to gradually eliminate error accumulation, each continuous frame calculates a set of motion variables, the relative error of continuous image frames is relatively small, and only needs to be calculated at the connection between segments, compared with traditional GBA optimization, the performance is improved to a certain extent, and the processing efficiency is improved to a certain extent. The relative pose of each consecutive keyframe remains unchanged during optimization, only visible landmarks between segments are optimized, experimental data shows that the improved backend optimization strategy has good real-time performance and high efficiency in processing camera trajectory optimization in large scenes. At the same time, in augmented reality applications, the system detects that a plane in a real-time scene is considered a homography plane that can add objects, manually adding a 3D model to the scene can provide users with a more intuitive visual experience [19].

III. Experimental Results

The author proposes a fast scene reconstruction algorithm based on improved V-SLAM, which includes parallel front-end and back-end threads, and is deployed and implemented on a personal PC, the hardware configuration is IntelCorei584002.8GHz CPU, 8GB memory, NVIDIAGTX1050Ti graphics card, OS is ubuntu16.04, and input video image resolution is 640x480. In the V-SLAM system, the front-end tracking thread projects real-world 3D points onto the 2D points of the current frame to solve the camera pose transformation, it is necessary to determine the position or offset of a feature point in the next frame. The author proposes a region matching algorithm for inter frame feature point matching to minimize errors and reduce computational pressure on backend optimization threads. Select an area with many or dense feature points, perform template affine transformation through the transformation network to match the next frame, select ROI in the region of interest in the image, and input ROI into the transformation network through certain amplification or reduction of ROI, the transformation network performs affine transformation on ROI until a transformation maximally matches a region of the next frame image, so that the feature points in the region can be matched. Good performance can also be achieved in cases of camera shake and rapid rotation, and the use of transform networks reduces the burden on the SLAM system and improves its real-time performance.

The algorithm proposed by the author was tested in offices and large indoor scenes, during the experimental shooting, the camera experienced some shaking and rapid movement, the algorithm combined RGB images and depth images for real-time processing to reconstruct a sparse 3D point cloud of the scene, the sparse reconstructed 3D point cloud was displayed in blue, the newly added 3D point cloud at the current frame in the scene is displayed in red, and the yellow trajectory represents the reconstructed camera motion trajectory. V-SLAM performs sparse reconstruction on two types of indoor environments, and the shooting data of the two scenes are shown in Table I. The SLAM method proposed by the author has good reconstruction results for large factory buildings, with the yellow trajectory being the camera shooting path trajectory.

TABLE I. Scene Information of Offices and Large Factories

Scenario Type	Track length/m	Time 1 second	Average Movement Speed/(m/s)
Office Scenarios	21.460	88.0	0.250
Large factory buildings	40.051	173.2	0.240

In terms of backend optimization, the main focus is on optimizing camera trajectories, including camera loop detection, local or global BA optimization, and scene map expansion. The author adopts a segmented optimization approach, marking feature points with optimized values reaching a certain threshold as reference landmarks without further processing, reducing the computational complexity of the system and improving real-time data processing capabilities. In the experiment, the camera trajectory was saved to a CameraTrajectory file and compared with the actual trajectory on the ground, absolute trajectory error (ATE) was used for comparison, and ATE was used to compare the SLAM system based on keyframes, as shown in Table II.

TABLE II. Path Absolute Error Analysis

Scenario Type	ATE /m			
	ORB-SLAM	PTAM	ISD-SLAM	Ours
Office Scenarios	0.100	0.745	0.874	0.096
Large factory buildings	0.121	0.727	0.895	0.104

Preprocessing was performed on each frame of the image, and a timestamp was used to compare the estimated camera pose with the actual camera pose, as shown in Fig. 2, compare the results of the error between the real trajectory and the optimized camera trajectory in the office small scene environment, by calculating the difference between camera poses in each segment, the absolute translation error is 0.09623. When estimating errors in large factory buildings, the backend optimization thread of the system performs well, as shown in Fig. 3, the actual trajectory and camera trajectory only deviate to a certain extent in local areas, while the other parts almost overlap. The absolute translation error in larger scenes is 0.10448. Root mean square error (RMSE) is also used to evaluate backend thread optimization performance, as shown in Table III. By comparing the performance parameters of various SLAM systems in the same scenario, the algorithm proposed by the author can more accurately process deep data and meet the real-time requirements of V-SLAM systems.

TABLE III. Comparison of Root Mean Square Error of Absolute Path for Key Frames of Different Algorithms

Scenario Type	RMSE /m			
	ORB-SLAM	PTAM	ISD-SLAM	Ours
Office Scenarios	0.034	-	0.385	0.030
Large factory buildings	0.381	0.332	0.356	0.327

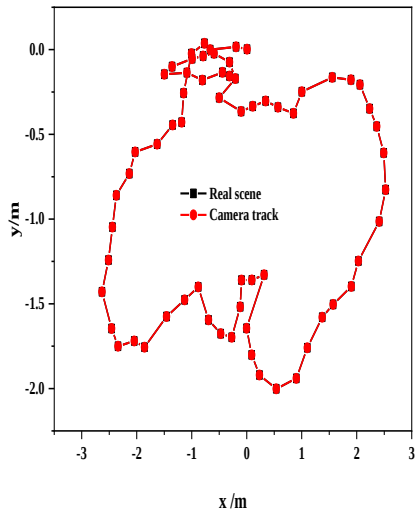

Fig. 2. Comparison of camera errors in office environments.

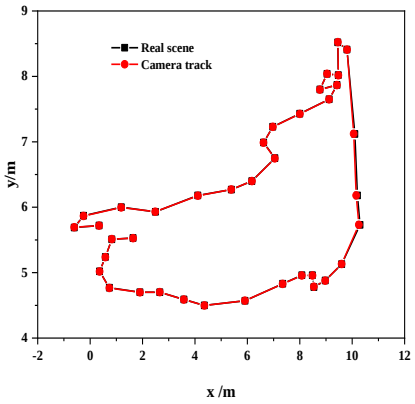

Fig. 3. Error comparison of camera trajectories in large factory environment.

The V-SLAM method proposed by the author can add virtual 3D objects to real scenes. The V-SLAM system can extract feature points from the screen in real-time and detect scene plane objects, ensuring that multiple virtual objects in the scene meet geometric consistency with the actual scene. In the experiment, two objects were added to the virtual scene, users can interactively scale objects and add them without being affected by camera movements, ensuring consistency between objects and the real scene [20].

IV. Conclusion

SLAM technology has been a research and application hotspot in recent years, applied in fields such as scene reconstruction, perception, digital city construction, VR/AR applications, drone driving, robotics, etc. SLAM includes LSDSLAM, ORBSLAM, Mono SLAM, etc. Simultaneous Location and Map Building (SLAM) plays an important role in the fields of computer vision and robotics, and also provides basic technical support for VR/AR applications. When facing scenes with relatively single or complex features, the front-end feature matching module of traditional SLAM systems is difficult to generate accurate camera trajectory and scene reconstruction results due to the sparsity or density of feature points. The author proposes an improved algorithm for arbitrary path scene reconstruction based on visual SLAM, the front-end thread uses Hessian matrix to extract and match the image features, and applies affine transformation to the region of interest to identify the feature points of adjacent frames to improve the matching efficiency, thereby reducing the original error of camera track and scene reconstruction; The backend optimization thread reduces the number of marker points to optimize the number of feature points, and uses local and global BA (bundle adjustment) methods to segment and optimize the camera motion trajectory, reducing system errors and improving the efficiency of camera trajectory optimization. The proposed method can add 3D objects in real-time to the scene. The experimental results show that the improved visual SLAM algorithm has better real-time performance than traditional SLAM algorithms.

Acknowledgment

The paper is the output of R&D Platform Special of Xingtai Technology Innovation Center for Intelligent Sensing and control of mechanical and electrical equipments.

References

[1] Chen, L. , Jin, S. , & Xia, Z. . (2021). Towards a robust visual place recognition in large-scale vslam scenarios based on a deep distance learning. Sensors, 21(1), 310.

[2] Srithar, S. , Priyadharsini, M. , Sharmila, F. M. , & Rajan, R. . (2021). Yolov3 supervised machine learning framework for real-time object detection and localization. Journal of Physics: Conference Series, 1916(1), 012032 (7pp).

[3] Salim, M. Z. , Abboud, A. J. , & Yildirim, R. . (2022). A visual cryptography-based watermarking approach for the detection and localization of image forgery. Electronics, 11(1), 136-.

[4] Schubert, S. , Neubert, P. , & Protzel, P. . (2021). Graph-based non-linear least squares optimization for visual place recognition in changing environments. IEEE Robotics and Automation Letters, 6(2), 811-818.

[5] Lu, F. , Chen, B. , Zhou, X. D. , & Song, D. . (2021). Sta-vpr: spatio-temporal alignment for visual place recognition. IEEE Robotics and Automation Letters, PP(99), 1-1.

[6] Molloy, T. L. , Fischer, T. , Milford, M. J. , & Nair, G. N. . (2021). Intelligent reference curation for visual place recognition via bayesian selective fusion. IEEE Robotics and Automation Letters, 6(2), 588-595.

[7] Hu, H. , Wang, H. , Liu, Z. , & Chen, W. . (2021). Domain-invariant similarity activation map metric learning for retrieval-based long-term visual localization. IEEE/CAA Journal of Automatica Sinica, PP(99), 1-16.

[8] He, D. , Chuang, H. M. , Chen, J. , Li, J. , & Namiki, A. . (2021). Real-time visual feedback control of multi-camera uav. Journal of Robotics and Mechatronics, 33(2), 263-273.

[9] Wan, G. , Wang, G. , Xing, K. , Fan, Y. , & Yi, T. . (2021). Robot visual measurement and grasping strategy for roughcastings:. International Journal of Advanced Robotic Systems, 18(2), 715-720.

[10] Awwad, A. . (2021). Visual emotion-aware cloud localization user experience framework based on mobile location services. International Journal of Interactive Mobile Technologies (iJIM), 15(14), 140.

[11] Torii, A. , Taira, H. , Sivic, J. , Pollefeys, M. , Okutomi, M. , & Pajdla, T. , et al. (2021). Are large-scale 3d models really necessary for accurate visual localization?. IEEE transactions on pattern analysis and machine intelligence, 43(3), 814-829.

[12] Beuth, F. , Kowerko, D. , & Hamker, F. H. . (2021). Contrasting attentional processing in visual search, object recognition, and complex tasks. Journal of Vision, 21(9), 2753-.

[13] Chen, J. , Takashima, R. , Guo, X. , Zhang, Z. , & Hancock, E. R. . (2021). Multimodal fusion for indoor sound source localization. Pattern Recognition, 115(3), 107906.

[14] Liang, J. , Zhang, J. , Pan, B. , Xu, S. , & Zhang, X. . (2021). Visual reconstruction and localization-based robust robotic 6-dof grasping in the wild. IEEE Access, PP(99), 1-1.

[15] Mahapatra, S. , & Sahu, S. . (2021). Integrating resonant recognition model and stockwell transform for localization of hotspots in tubulin. IEEE transactions on nanobioscience, 20(3), 345-353.

[16] Nguyen, T. H. , Nguyen, T. M. , & Xie, L. . (2021). Range-focused fusion of camera-imu-uwb for accurate and drift-reduced localization. IEEE Robotics and Automation Letters, PP(99), 1-1.

[17] M Servières, Renaudin, V. , Dupuis, A. , & Antigny, N. . (2021). Visual and visual-inertial slam: state of the art, classification, and experimental benchmarking. Journal of Sensors, 2021(1), 1-26.

[18] Zhao, W. , Panerati, J. , & Schoellig, A. P. . (2021). Learning-based bias correction for time difference of arrival ultra-wideband localization of resource-constrained mobile robots. IEEE Robotics and Automation Letters, PP(99), 1-1.

[19] Yang, M. , & Li, Y. . (2021). Design of intelligent safety protection robot based on global position system and machine vision. Journal of Physics: Conference Series, 1883(1), 012146 (6pp).

[20] Ali, R. , Liu, R. , He, Y. , Nayyar, A. , & Qureshi, B. . (2021). Systematic review of dynamic multi-object identification and localization: techniques and technologies. IEEE Access, PP(99), 1-1.