Comparative Analysis of Visual SLAM Algorithms across various Environments

First A. Author, Second B. Author, Jr., and Third C. Author, *Member, IEEE*

***Abstract*—These instructions give you guidelines for preparing papers for IEEE TRANSACTIONS and JOURNALS.**

***Index Terms*—About four key words or phrases in alphabetical order, separated by commas.**

I. Introduction

Historically, prior to the looming of SLAM technology[1], it has been a field on study for many decennium. localization and mapping were treated as distinct entities. Although accurate localization depends on map, mapping depends on localization. From radars and range finders, to cameras and lidars, many modalities of SLAM have been developed.

Simultaneous Localization and Mapping (SLAM)[2], [3] is a method for obtaining the 3D of an unknown environment and the sensor motion in the environment and was originally proposed to achieve autonomous control of robots in robotics. The SLAM based applications have widely used because of their pure localization, re-localization of a lost track, resource efficiency, loop closure, reliability, and support for a broad range sensor. In early SLAM algorithms, many different sensors are integrated such as Camera, Lidar, GPS, inertial sensors, rotary encoders and Radar.

In recent years, SLAM using cameras are actively booming because of the sensor configuration is simple and these Visual SLAM (vSLAM) are cost effective compare to radar and lidar. Visual SLAM is used for simultaneous building of a 3D global map of environment[4]. This 3D structure of environment can be dense or sparse, depending on implemented vSLAM. Visual SLAM methods can be classified into direct methods & feature-based methods. Feature-based methods apply a sparse image representation for reducing the scene to a set of observed feature points. Direct methods minimize photometric errors, whereas feature-based methods reduce feature reprojection errors.

Due to the significance of visual techniques in interactive robotic applications. V-SLAM can be applied to cameras to create a map of their surroundings and easily locate themselves within their work space[5]. It uses techniques such as computer vision to extract and match visual data for localization, mapping and accurate tracking of camera poses and estimating past trajectories. In vSLAM, various types of cameras are used to capture images or videos.

In this paper, we compare 4 types of algorithms i.e. ORB-SLAM2, ORB-SLAM3, DyanaSLAM, PL-SLAM in different environments like (Indoor and Outdoor). The purpose of this comparison is to identify different parameters like Speed, Accuracy, Memory utilization and power consumption. This are compared by in multiple operating domains with serval sensors like Stereo, Monocular, RGB-D to showcase each technique to generalize various environment configurations.

This paper is organized as follows. In Sections 2 and 3, the system setup and formulation. In section 4 and 5 visual SLAM pattern and methodology. In section 6, 7, 8 are Datasets, Results, Open challenges. Finally, we present the conclusion in section 9. Note that they are some vSLAM algorithms which useful to understand the algorithms and visual SLAM.

II. Framework of visual slam

vSLAM framework comprises sequential steps organized to develop the system and process its data. **Figure 01**, which illustrates the processes of vSLAM. It enables the effective implementation and operation of various scenarios, such as filtering techniques[6]. These approaches strive to provide a sophisticated visual representation of the vSLAM processes and divided into mainly three parts i.e. System input, SLAM, System output. And in SLAM they are four subtopics that are explained below.

a. Data acquisition and initialization:

In the initial phase of vSLAM, data acquisition is a critical step that involves capturing sensor data from various camera's like (monocular, Stereo, RGB-D) are them to capture visual information about the various environment[7]. The quality and frequency of data acquisition may impact the subsequent stages of vSLAM. And system initialization involves establishing the initial stage of process, which includes determining the initial position and orientation of cameras and the key components are Camera calibration, Sensor synchronization, Initial pose estimation and Environment scanning.

b. Localization:

In next phase of vSLAM, localization is used for to determine accurate position and orientation of the cameras. Precise localization is essential for generating and updating the map and this localization involves in some steps like Feature extraction, Feature matching and tracking, pose estimation, Optimization & Correction, sensor fusion and Map updating.

c. Map formation:

This phase plays a vital aspect of the vSLAM process. This map formation serves reference navigation, localization, and further analysis. And involving in initial map creation, Feature extraction and description, data association, handling various environment conditions.

d. Loop closing and process tuning:

This is the final phase in VSLAM process that is loop closing it address the problem of drift which results small errors in the predicted camera poses were leads to inaccuracies in the map. Loop closing helps to correct those errors when the system returns to previously mapped location[8]. Here also some steps are involved like loop detection, pose graph optimization. And process tuning is basically adjusting the parameters and algorithms with the vSLAM and here also some of the key aspects are include like algorithm selection, sensor fusion adjustment, real time performance, robustness to environment changes and continuous learning adaption.

III. Methodology

In this section, we provide a brief overview of the current vSLAM algorithms and techniques including their methodology, efficiency, time requirements and processing capacity. We know that SLAM system is designed to map the environment around the sensors while together determine the precise location and mapping of the sensors like monocular, RGB-D and Stereo cameras for sensor motion and scanning environment. Actually, this approach is cost-effective, easy to calibrate, and has low power consumption in monocular cameras and also depth estimation and high accuracy in RGB-D and Stereo cameras.

a. PL-SLAM

PL-SLAM (point and line SLAM) is a progressive vSLAM algorithm system designed to work in environments whereas traditional pint-based methods struggle, notably in low-textured scenes[9]. Standard techniques in vSLAM, such as ORB-SLAM usually depends on point correspondences. These methods take challenges in environment with poor textures were feature point temporarily lost dur to motion blur. So, PL-SLAM overcomes these challenges by integrating point correspondences with line-based geometric primitives. This approach is well in structured environment like indoor where line is more prevalent. And a new initialization technique is introduced in this system where it becomes well in situations where feature points are unavailable.

PL-SLAM pipeline consists of serval stages, including tracking, local mapping and loop closing, each adapted to handle both point and line features. In this SLAM line segments are detected in each frame which operates efficiently in linear time. This detected line is matched with existing lines in the map using a relational graph strategy. When a frame contains new information about the environment then it will be added as keyframe and the lines are triangulated and added to the map to boost map detail and accuracy. This line positions in the map are optimized using local Bundle adjustment (BA).

PL-SLAM is suitable for application in autonomous navigation and localization reality where precise mapping of the environment, even in low-texture or challenging visual conditions.

b. DynaSLAM

This SLAM system designed to function well in dynamic environment[10]. DynaSLAM techniques to direct, track and handle dynamic object whereas traditional SLAM systems believe static environments and consider dynamic features as oddity. DynaSLAM integrates multi-view geometry and deep learning techniques to manage dynamic scenes. It also employs convolution neural networks to segment dynamic objects, where segmentation is essential to exclude dynamics objects from the mapping and tracking process for stable and accurate mapping.

DynaSLAM pipeline consists stages where this SLAM utilizes Mask-RCNN for pixel wise sematic segmentation of dynamic objects. It includes a low-cost tracking module that localizes the camera within the created map which can get accurate camera pose estimation for effective mapping and tracking. For RGB-D cameras, DynaSLAM refines the segmentation of objects and detects new dynamic object that were static during the CNN segmentation which involves analyzing the scene from multiple view points and also reconstructs the occluded parts of the scene using information from previous frames. In monocular and stereo camera setups, images pass through the CNN for dynamic object segmentation before being used for mapping and tracking and RGB-D cameras combines CNN-based segmentation and geometric methods to improve dynamic object detection.

DynaSLAM ability is to detect and manage the dynamic environment makes it suitable for autonomous vehicles and other applications that require accurate and reliable environmental mapping.

c. ORB-SLAM2

ORBSLAM2(Oriented FAST and Rotated BRIEF SLAM2) is an advanced vSLAM system that builds on the capabilities using different types of cameras, such as monocular, stereo and RGB-D[11]. the features of this SLAM system is it has multi-sensor support and has system capabilities of real-time optimization, Map reuse, Loop closing, Relocalization. It has Bundle adjustment (BA) which performs local maps and globally after loop closes. It has Lightweight localization which uses visual odometry tracks for regions and matches to map points to allow for zero-drift localization.

ORB-SLAM2 pipeline defined in serval parallel threads and modules like tracking which responsible for real-time camera pose estimation using feature matching and motion which can do in all three camera sensors. After that local mapping that adds new keyframes and performs local BA. Loop closing detects loops and performs pose-graph optimization to correct drift and ensure global consistency. By using DBoW2 for recognizing previously visited places and for relocalization which helps to recover from tracking failures and recognizing loop closures. Feature extraction and matching are precise to rotation, scale and illumination changes which extracts both tracking and place recognition. Stereo and depth processing are extracted from rectified stereo images, where depth can be inferred. They use Bundle adjustment like motion only BA which optimizes the camera pose in 3D points, Local BA which optimizes a local key frame to keep locally consistent and full BA which is done after loop closures to refine the entire map and trajectory.

ORB-SLAM2 is versatile and can be applied in various scenarios like indoor and outdoor environment, autonomous vehicles etc.

d. ORB-SLAM3

ORB-SLAM3(Oriented FAST and Rotated BRIEF SLAM3) is a sophisticated and accurate SLAM system capable of performing vSLAM, visual-inertial and multimap SLAM using various camera setups, including monocular, stereo and RGB-D cameras, and supports both pinhole and fisheye lens models[12]. The ORB-SLAM3 can be represent multiple disconnected maps, allowing for operations such as recognition, relocalization, loop closure, and seamless map merging. This approach visual and inertial data tightly, using MAP estimation for robust and accurate SLAM which includes IMU installation phase. It supports Multimap systems allowing it to handle long periods of poor visual information. If system loses tracking, then it starts a new map, which can later be merged with existing maps when revisited.

ORB-SLAM3 pipeline is modular, consist serval interconnected stages like tracking where responsible for estimating the camera pose for each frame, detecting key points and matching them with existing map points. Mapping adds new keyframes and refines the map with bundle adjustment and relocalization corrects significant localization errors by recognizing previously visited places. Loop closing defines and corrects loop closures to improve the global consistency of the map and IMU integration is also done that IMU data is integrated to aid in pose estimation and map accuracy, especially during rapid motion or in low-texture scenes.

ORB-SLAM3 is designed to be agnostic of the camera model because various camera model providing their projection, unprojection and Jacobian functions. This system introduces an improved place recognition method that increases recall by checking geometrical consistency and also done data association which enhances the map accuracy and robustness in leveraging short-term, mid term and long-term data associations, ensuring zero drift in mapped areas and better performance in loopy environment. ORB-SLAM3 suitable for Autonomous navigation for drones and robots, AR and VR systems, vehicle navigation and 3D reconstruction.

BENCHMARKING DATASETS:

As we know SLAM traced back to nearly three decenniums. Now a days vSLAM is booming a lot and making big impact on robotics and computer vision. vSLAM has created different types of algorithms to gear specific problems in localization, mapping and considering different environment conditions. To certify and differentiate these algorithms we created datasets which plays curial role in the field of vSLAM methods.

In this section we will discuss about the datasets that are used for testing, validating and comparing various vSLAM algorithms. To estimate the performance of this methods we require a variety of different environments and sensors has shown in figure 02.

a. EuRoC MAV

EuRoC dataset is an important resource for analyzing the vSLAM algorithms. The dataset includes sequences captured in indoor environment and each sequence is rated by its complexity, speed, accuracy, change rate and variations within the camera frame. Here this dataset will support for Monocular and stereo approaches. If you want to know about EuRoC then refer[13].

b. KITTI

KITTI dataset used for autonomous driving systems by some variety of sensors, including cameras, LiDAR, GPS. This dataset captures outdoor environment and each sequence is characterized by various elements like pedestrian's and vehicle's and also in different lighting conditions. Here this dataset will support for both monocular and stereo approaches. If you want to know about KITTI then refer[14].

c. TUM RGB-D

TUM RGB-D dataset is highly used resource which provides RGB-D data capture from a camera sensor. Dataset that includes sequences captured in indoor environment obtained by using an external motion capture system. This dataset can handle occlusions, low-texture areas and depth perception problems. This dataset will support for RGB-D approach. If you want to know about TUM RGB-D then refer[15].

WHY WE USED THESE PARAMETERS?

To compare and effectively assess vSLAM algorithms, it is important to measure and define key parameters across different environments. This section gives detailed formulations of four critical parameters like accuracy, speed, memory utilization, power consumption. By these parameters you can check the efficiency and robustness of different vSLAM algorithms which enables their applicability in various environment and use cases.

a. Accuracy

This parameter refers to the reliability of vSLAM system in terms of localization and mapping. A certain algorithm can be quantified using two measures one is pose estimation error which involves in comparing the estimated pose of the camera against the ground truth were RMSE (Root mean square error).

$$\sqrt{\frac{1}{N}\sum_{i=1}^{N}(\left|p_i^{est}-P_i^{gt}\right|^2+\left|q_i^{est}-q_i^{gt}\right|^2}$$

$$\therefore\ p_i^{est}\ and\ p_i^{gt}\ is\ estimated\ position\ \&\ ground\ truth$$
$$q_i^{est}\ and\ q_i^{gt}\ is\ estimated\ position\ \&\ ground\ truth$$

And other one is mapping accuracy were the generated map matches accurately to actual environment. It can be done by overlap metrics or through comparing with high precision reference map.

b. Speed

Speed is typically measures in frame per second (FPS)

$$\therefore\ FPS=\frac{1}{T_{avg}}$$

$$T_{avg}=\frac{1}{N}\sum_{i=1}^{N}T_i$$

$$\therefore\ T_i\ is\ the\ processing\ time\ for\ frames$$
$$N\ is\ the\ total\ number\ of\ frames$$

c. Memory utilization

This refers to the number of resources consumed by vSLAM algorithm during its process. The maximum memory used by algorithm at any point is measured used by algorithm at any point is measured in Megabytes (MB) or Gigabytes (GB). The average memory utilization over a period of time can be calculated as.

$$M_{avg} = \frac{1}{T}\int_{0}^{T} M(t)dt$$

$$\therefore M(t) is\ the\ memory\ usage$$
$$T\ is\ total\ run-time$$

d. Power consumption

Power consumption is also curial parameter for compare different type of algorithms especially for mobile and embedded applications. The power consumption over a specific point measure in watt(W). It can be monitor using power measurement tools and sensors integrated into the hardware platform.

$$\therefore\ P_{avg} = \frac{1}{T}\int_{0}^{T} P(t)dt$$

IV. Results

V. Conclusion

First A. Author (M'76–SM'81–F'87) and the other authors may include biographies at the end of regular papers. Biographies are often not included in conference-related papers. This author became a Member (M) of IEEE in 1976, a Senior Member (SM) in 1981, and a Fellow (F) in 1987. The first paragraph may contain a place and/or date of birth (list place, then date). Next, the author's educational background is listed. The degrees should be listed with type of degree in what field, which institution, city, state, and country, and year degree was earned. The author's major field of study should be lower-cased.

The second paragraph uses the pronoun of the person (he or she) and not the author's last name. It lists military and work experience, including summer and fellowship jobs. Job titles are capitalized. The current job must have a location; previous positions may be listed without one. Information concerning previous publications may be included. Try not to list more than three books or published articles. The format for listing publishers of a book within the biography is: title of book (city, state: publisher name, year) similar to a reference. Current and previous research interests end the paragraph.

The third paragraph begins with the author's title and last name (e.g., Dr. Smith, Prof. Jones, Mr. Kajor, Ms. Hunter). List any memberships in professional societies other than the IEEE. Finally, list any awards and work for

References

[1] H. Taheri and Z. C. Xia, "SLAM; definition and evolution," *Eng Appl Artif Intell*, vol. 97, Jan. 2021, doi: 10.1016/j.engappai.2020.104032.

[2] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: Part I," *IEEE Robot Autom Mag*, vol. 13, no. 2, pp. 99–108, Jun. 2006, doi: 10.1109/MRA.2006.1638022.

[3] T. Bailey and H. Durrant-Whyte, "Simultaneous localization and mapping (SLAM): Part II," *IEEE Robot Autom Mag*, vol. 13, no. 3, pp. 108–117, Sep. 2006, doi: 10.1109/MRA.2006.1678144.

[4] S. Peng, F. Huang, J. Tao, B. Tie, J. Lu, and X. Zhang, "Fast 3D map reconstruction using dense visual simultaneous localization and mapping based on unmanned aerial vehicle," *International Geoscience and Remote Sensing Symposium (IGARSS)*, vol. 2018-July, pp. 5712–5715, Oct. 2018, doi: 10.1109/IGARSS.2018.8517796.

[5] X. Zhang, Q. Liu, B. Zheng, H. Wang, and Q. Wang, "A visual simultaneous localization and mapping approach based on scene segmentation and incremental optimization," *Int J Adv Robot Syst*, vol. 17, no. 6, Dec. 2020, doi: 10.1177/1729881420977669/ASSET/IMAGES/LARGE/10.1177_1729881420977669-FIG11.JPEG.

[6] G. Jia *et al.*, "Visual-SLAM Classical Framework and Key Techniques: A Review," *Sensors 2022, Vol. 22, Page 4582*, vol. 22, no. 12, p. 4582, Jun. 2022, doi: 10.3390/S22124582.

[7] Y. bao Ai *et al.*, "Visual SLAM in dynamic environments based on object detection," *Defence Technology*, vol. 17, no. 5, pp. 1712–1721, Oct. 2021, doi: 10.1016/J.DT.2020.09.012.

[8] K. A. Tsintotas, L. Bampis, and A. Gasteratos, "The Revisiting Problem in Simultaneous Localization and Mapping: A Survey on Visual Loop Closure Detection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 11, pp. 19929–19953, Nov. 2022, doi: 10.1109/TITS.2022.3175656.

[9] IEEE Robotics and Automation Society and Institute of Electrical and Electronics Engineers, *ICRA2017 : IEEE International Conference on Robotics and Automation : program : May 29-June 3, 2017, Singapore*.

[10] B. Bescos, J. M. Facil, J. Civera, and J. Neira, “DynaSLAM: Tracking, Mapping, and Inpainting in Dynamic Scenes,” *IEEE Robot Autom Lett*, vol. 3, no. 4, pp. 4076–4083, Oct. 2018, doi: 10.1109/LRA.2018.2860039.

[11] R. Mur-Artal and J. D. Tardós, “ORB-SLAM2: an Open-Source SLAM System for Monocular, Stereo and RGB-D Cameras,” 2017, doi: 10.1109/TRO.2017.2705103.

[12] C. Campos, R. Elvira, J. J. G. Rodriguez, J. M. M. Montiel, and J. D. Tardos, “ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial, and Multimap SLAM,” *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, Dec. 2021, doi: 10.1109/TRO.2021.3075644.

[13] J. Kang, Z. Sjanic, and G. Hendeby, “Technical report from Automatic Control at Linköpings universitet State-of-the-art Report of Research about Multi Sensor Image-based Navigation State-of-the-art Report of Research about Multi Sensor Image-based Navigation,” 2023.

[14] A. Geiger, P. Lenz, and R. Urtasun, “Are we ready for autonomous driving? the KITTI vision benchmark suite,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 3354–3361, 2012, doi: 10.1109/CVPR.2012.6248074.

[15] 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, doi: 10.1109/IROS.2012.6385773.

ALL BIOS ARE REQUIRED

First A. Author (M’76–SM’81–F’87) and the other authors may include biographies at the end of regular papers. Biographies are often not included in conference-related papers. This author became a Member (M) of IEEE in 1976, a Senior Member (SM) in 1981, and a Fellow (F) in 1987. The first paragraph may contain a place and/or date of birth (list place, then date). Next, the author’s educational background is listed. The degrees should be listed with type of degree in what field, which institution, city, state, and country, and year degree was earned. The author’s major field of study should be lower-cased.

The second paragraph uses the pronoun of the person (he or she) and not the author’s last name. It lists military and work experience, including summer and fellowship jobs. Job titles are capitalized. The current job must have a location; previous positions may be listed without one. Information concerning previous publications may be included. Try not to list more than three books or published articles. The format for listing publishers of a book within the biography is: title of book (city, state: publisher name, year) similar to a reference. Current and previous research interests end the paragraph.

The third paragraph begins with the author’s title and last name (e.g., Dr. Smith, Prof. Jones, Mr. Kajor, Ms. Hunter). List any memberships in professional societies other than the IEEE. Finally, list any awards and work for IEEE committees and publications. If a photograph is provided, the biography will be indented around it. The photograph is placed at the top left of the biography. Personal hobbies will be deleted from the biography.