Array 15 (2022) 100222

Contents lists available at ScienceDirect

Array

journal homepage: www.sciencedirect.com/journal/array

Visual SLAM algorithms and their application for AR, mapping, localization and wayfinding☆

Charalambos Theodorou [a,b,*], Vladan Velisavljevic [a], Vladimir Dyo [a], Fredi Nonyelu [b]

[a] School of Computer Science and Technology, University of Bedforshire, Luton, LU1 3JU, United Kingdom
[b] Briteyellow Ltd, Bedford, MK43 0BT, United Kingdom

ARTICLE INFO

Keywords:
AR
Visual SLAM
Simultaneous localization and mapping (SLAM)

ABSTRACT

Visual simultaneous localization and mapping (vSLAM) algorithms use device camera to estimate agent's position and reconstruct structures in an unknown environment. As an essential part of augmented reality (AR) experience, vSLAM enhances the real-world environment through the addition of virtual objects, based on localization (location) and environment structure (mapping). From both technical and historical perspectives, this paper categorizes and summarizes some of the most recent visual SLAM algorithms proposed in research communities, while also discussing their applications in augmented reality, mapping, navigation, and localization.

1. Introduction

Visual SLAM, according to Fuentes-Pacheco et al. [1], is a set of SLAM techniques that uses only images to map an environment and determine the position of the spectator. Compared to sensors used in traditional SLAM, such as GPS (Global Positioning Systems) or LIDAR [2], cameras are more affordable, and are able to gather more information about the environment such as colour, texture, and appearance. In addition, modern cameras are compact, have low cost, and low power consumption. Examples of recent applications that employ vSLAM are the control of humanoid robots [3], unmanned aerial and land vehicles [4], lunar rover [5], autonomous underwater vehicles [6] and endoscopy [7].

Depending on the camera type, there are three basic types of SLAM: monocular, stereo, and RGB-D. Stereo SLAM is a multi-camera SLAM that can obtain a particular degree of trajectory resolution. Additionally, stereo SLAM has the advantage of being more versatile as opposed to RGB-D SLAM which is more sensitive to sunlight and is mainly used indoors. The last two decades have seen significant success in developing algorithms such as MonoSLAM [8], PTAM [9], PTAM-Dense [10], DTAM [11] and SLAM++ [12]. However, most systems have been developed for motionless environments and their robustness is still a concern under dynamic environments. Due to the assumption that the camera is the only moving object in a stationary scene, these SLAM systems [13] are typically not applicable. Moving objects, as a consequence, would affect the system's ability to estimate camera poses. Additionally, the extra object motion introduces calculation errors and reduces the accuracy of trajectory estimation due to increased computation weight. In such environments, the SLAM algorithm is required to deal with possible errors and a certain degree of uncertainty characteristic in sensory measures.

Moreover, for virtual objects to be properly anchored in the real environment in an AR (Augmented Reality) [14] experience, it is necessary to apply tracking techniques. That means dynamically determining the viewer's pose (position and orientation) in relation to the actual elements of the scene. An alternative is the application of SLAM techniques, which aim precisely at the creation and updating of a map, as well as the location of the observer in relation to the structure of the environment. This confluence between visual SLAM and AR was the motivation for the realization of this survey. The objective of this research is to carry out a survey of the main visual SLAM algorithms, as well as their applications in AR, mapping, localization, and wayfinding. The main characteristics of the visual SLAM algorithms were identified and the main AR applications on visual SLAM were found and analysed.

As opposed to presenting a general analysis of SLAM, this survey provides an in-depth review of different visual SLAM algorithms. The survey also includes various datasets that might be considered for evaluation and different types of evaluation metrics. Existing studies in

☆ This work was supported by **Briteyellow Ltd** and by **Innovation Bridges**.
* Corresponding author. School of Computer Science and Technology, University of Bedforshire, Luton, LU1 3JU, United Kingdom.
E-mail address: Theodorou.Charalampos@study.beds.ac.uk (C. Theodorou).

https://doi.org/10.1016/j.array.2022.100222
Received 15 March 2022; Accepted 3 July 2022
Available online 8 August 2022

this area tend to describe only one SLAM algorithm, and some of them are rather old. However, to address this, a complete survey describing seminal and more recent SLAM algorithms was produced.

Even if some surveys include a description of different SLAM algorithms (e.g., Refs. [15–17]), an expanded overview of SLAM algorithms, including those recently developed, is included in this survey, a set of datasets that could be used to evaluate multiple SLAM algorithms and a set of evaluation metrics Table 1. Additionally, the limitations of the evaluation metrics have been identified, which will be explored further in the future. Through this article, we hope to help readers better understand different SLAM algorithms and how they might be applied in different fields.

Following is a description of how this survey is organized. SLAM applications were introduced in Section II. In Section III, various SLAM algorithms were discussed. In Section IV a table of different SLAM features was introduced. Section V includes a discussion of various datasets that could be used to experiment with SLAM algorithms. Section VI includes a description of the two most common used evaluation metrics. Section VII is mainly focused on discussions about SLAM and Section VIII concludes this survey.

2. Visual SLAM applications

SLAM algorithms make use of data from different sensors. Visual SLAM is a SLAM technique that uses only visual sensors which may require a monocular RGB camera [18], a stereo camera [19], an omnidirectional camera (which captures images simultaneously in all 360-degree directions) [20] or an RGB-D camera (captures depth pixel information in addition to RGB images) [21]. Following, in this section Localization, Mapping, and Wayfinding, the three main categories of vSLAM, are described in more depth, along with some relevant algorithms applicable to each category.

2.1. Localization

Localization systems assist users in identifying their location and orientation within an environment. Localization can be performed both indoors [22] and outdoor [23] using various methods. There are traditional approaches that do not involve technology, such as the use of mental maps that are built through guided exposure to the environment or auditory instructions, as well as tactile maps. Robots must be able to understand their current position in the environment in order to complete tasks securely and autonomously. This type of problem is defined as a requirement that the robot knows where it is in the environment. Laser rangefinders [24] are among the most popular sensors used in developing SLAM algorithms. For the localization of unmanned ground vehicles (UGVs) in large-scale indoor environments, a scan matching method was proposed along with an algorithm for multi-resolution likelihood mapping. As a result, searching space can be reduced during the matching process. Using a laser rangefinder, the position of the robot can be determined with considerable accuracy depending on the rangefinder's accuracy. Although the laser rangefinder enables the determination of pose, in large and highly irregular environments, where walls surrounding the environment or corridors have no variety. In an environment with simple geometry, like a park with trees and circles, a robot, for example, can easily get lost.

In recent years, interest in visual SLAM has increased significantly [25]. Visual SLAM systems typically triangulate the position of set points through successive camera frames, and simultaneously use this information to estimate the camera's pose. In essence, these systems are designed to map their surroundings in relation to their location in order to facilitate navigation.

This can be accomplished with a single 3D vision camera, unlike other forms of SLAM. As long as there are sufficient points tracked through each frame, both the orientation of the sensor and the structure of the physical environment can be quickly understood.

The goal of all visual SLAM systems is to minimize reprojection error, or the difference between the projected and actual points, by means of algorithmic adjustment known as bundle adjustment. Due to the need for visual SLAM systems to operate in real-time, location and mapping data are bundled adjusted separately but simultaneously to provide faster processing speeds then merged together.

Robots can also be equipped with a camera as another common sensor. Engle et al. propose a camera-based method for large-scale, semi-dense maps called LSD-SLAM [26] that does not require adjusting bundles of features. ORBSLAM [27] is another approach to visual SLAM proposed by Use Mur-Artal et al. which allows visual SLAM problems to be broken down into three subtasks: tracking, mapping, and optimizing. Due to the camera's sensitivity to light changes, using a monocular camera is not sufficient to solve the scaling problem. This could lead to errors during the localization process. A combination of laser rangefinder and monocular camera might be useful since those two sensors are most commonly used for robots. Although such algorithms improve the performance of feature extraction, they are limited to particular SLAM algorithms.

2.2. Mapping

Visual SLAM mapping is performed by using cameras to acquire data about an environment, followed by combining computer vision and odometry algorithms to map the environment. This allows robots to navigate themselves independently and to improve localization. The majority of robots have wheels, so measuring their distance is easy. Inertial measurement units (IMU) [28] have been added to some robots for measuring their body motion. Nonetheless, relying on odometry alone to estimate location is not helpful. However, relying on odometry alone to calculate the own position is not accurate enough because of accumulated error produced by noise. When it takes several turns for the error to accumulate, the location becomes uncertain. Loop closure plays a key role here. Considering buildings and trees to be static, there should be a loop closure [29] at the exact same place since it has already been there (loop detection) [30]. It is then possible to correct and adjust the generated map for these accumulated noises [31].

The vSLAM concept is fundamental to any kind of robotic application where the robot must traverse a new environment and generate a map. The system is not limited to robots, but can be used on smartphones and their cameras, as well. vSLAM would be one aspect of the pipeline needed by some advanced AR use cases, for example, where virtual worlds need to be accurately mapped onto real environments.

2.3. Wayfinding

Wayfinding systems must be capable of planning and communicating effective paths to users. Localizing the user and planning the path to the user's desired destinations go hand in hand. Once a user has been localized, the optimal path to destination can be determined and communicated to the user as accessible instructions. There is always a possibility that the user may veer from the recommended path for many reasons, and a smart navigation aid will be capable of dynamically replanning the path to the user's destination based on his/her new location. While remaining simple and effective, directions must include landmarks that can be sensed by the user during navigation. State filter-based SLAM algorithms such as MonoSLAM and pose-graph SLAM such

Table 1
Survey papers discussing SLAM algorithms.

Ref.	Year	Algorithm Flowcharts	Datasets	Metrics
[15]	2015	No	No	No
[16]	2017	No	No	No
[17]	2020	No	No	No
Our Work	2022	Yes	Yes	Yes

as PTAM, DTAM and LSD-SLAM can be used for wayfinding. In wearable.

RNAs [32] today there are multiple SLAM algorithms implemented, such the algorithm proposed by Saez for a 6-DOF Pose Estimation (PE) [33] by using an RNA camera. Visual Odometry (VO) algorithm [34] and entropy-based cost functions are used to determine egomotion (change in pose between two camera views) of the camera. For estimating the pose of a wearable RNA based on stereovision, a metric-topological SLAM method (SLAM++) has been proposed. As features from stereo camera images are extracted and tracked step by step, local topological maps of the area and global topological relations between the areas on the maps are updated. Stereo cameras cannot provide complete depth information about the scene, though these RNAs are not capable of detecting objects. Wearable RNAs [35] rely on RGB camera depth data due to its capability of providing more reliable depth data in feature-sparse environments. In order to estimate the camera's pose, a bundle-adjustment algorithm [36] is used and visual features are extracted from images and compared against each other. It is possible to estimate camera pose in real-time thanks to PL-SLAM [37], which separates the task of tracking and mapping into two separate threads and processes them on a dual-core computer. Recent SLAM methods align the whole image rather than matching features. However, these types of methods are typically less accurate than feature-based SLAM methods for estimating pose.

3. SLAM algorithms

In general, Visual SLAM algorithms have three basic modules: initialization [38], tracking and mapping [39]. The initialization consists of defining the global coordinate system of the environment to be mapped, as well as the reconstruction of part of its elements, which will be used as a reference for the beginning of the tracking and mapping. This step can be quite challenging for some visual SLAM applications Fig. 1. The next section of this paper is split into three categories: monocular based, stereo focused, and monocular and stereo focused vSLAM algorithms. In detail, each algorithm is described along with its advantages and disadvantages.

3.1. Monocular based

Monocular SLAM is a type of SLAM that relies exclusively on a monocular image sequence captured by a moving camera in order to perform mapping, tracking and wayfinding. A monocular image sequence is usually a set of images that are similar to each other.

PTAM. A hand-held camera can be tracked by Parallel Tracking and Mapping (PTAM) in an AR environment. In parallel threads, tracking and mapping are handled separately. The first thread tries to track the erratic motion of a hand-held device, while on the other hand the second method generates a 3D map of the point features based on previous frames. A detailed map is produced, with thousands of landmarks. Clearly visible at high frame rates Model-based systems are surpassed in terms of their accuracy and robustness by this method. In the process of mapping, there are two distinct stages [9]. A first stage involves creating an initial map with stereo techniques. After the keyframes (map points) are added to the map by tracking systems, the mapping thread refines and expands Fig. 2.

Video images captured by the hand-held camera are used to maintain real-time estimates of the camera's position in relation to the built map. After estimating the video frame, graphics can be augmented over it. To calculate the final pose, the system uses the same procedure every frame. The motion model is used to generate a pose estimate from a new frame every time the camera detects a new frame. An estimate of the frame's prior pose determines how map points should be projected into the image. A final pose estimation is computed based on the detection of coarse-scale features in the image. From these coarse matches, the camera pose is updated, and the overall pose is estimated.

PTAM is advantageous because it splits tracking and mapping into two separate tasks and processes them in parallel, allowing for batch optimization techniques that are not generally associated with real-time operations. This map only serves as a tool for tracking cameras, which is a limitation of PTAM. Virtual entities should be able to interact with the map's geometry, so it shouldn't be static. PTAM also lacks auto-occlusive capabilities, which mean that it cannot track objects without outside assistance. Another limitation of SLAM is that it is not designed to close large loops. M-estimators are a general class of extremum estimators where the objective function is the sample average. The non-linear least squares method and maximum likelihood estimation are both special cases of M-estimators. M Estimators of trackers do not take feature map uncertainties into account, but this does not affect AR applications.

MonoSLAM. The first successful SLAM algorithm in mobile robotics was monocular SLAM (MonoSLAM). By moving rapidly along the trajectory of a monocular camera in an unknown environment, natural landmarks can be reconstructed into 3D maps, and an urban environment can be mapped using sparse but persistent points. In this approach, a map of natural landmarks is created online in a probabilistic framework from a sparse but persistent set of data. A fundamental aspect of MonoSLAM is the feature-based map that is a probabilistic snapshot of the camera's current state at any given point, as well as the uncertainty in its estimations. Fig. 3 shows how the Extended Kalman.

Filter continuously updates maps when the system starts up and persists until the operation is complete. Motion of the camera and

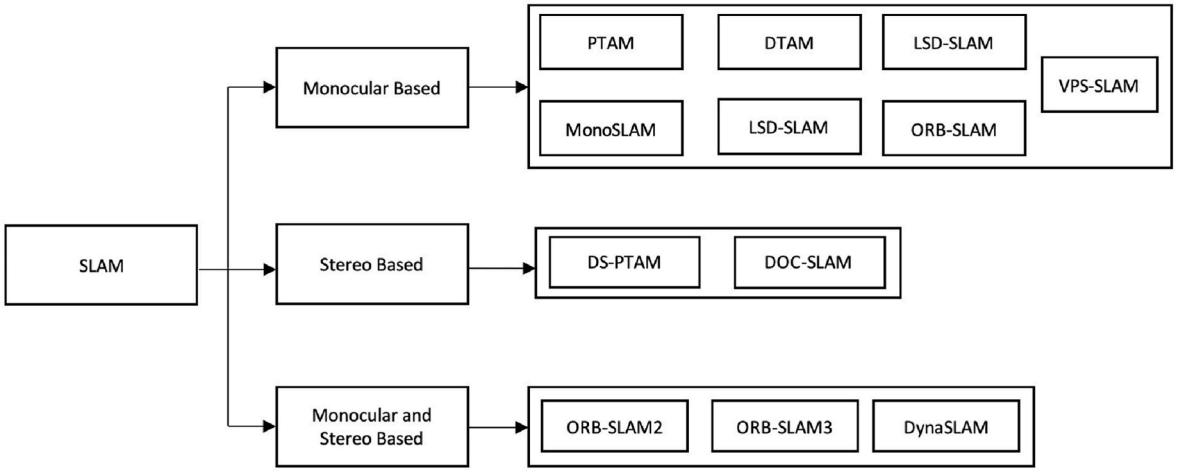

Fig. 1. Classification of Visual SLAM algorithms.

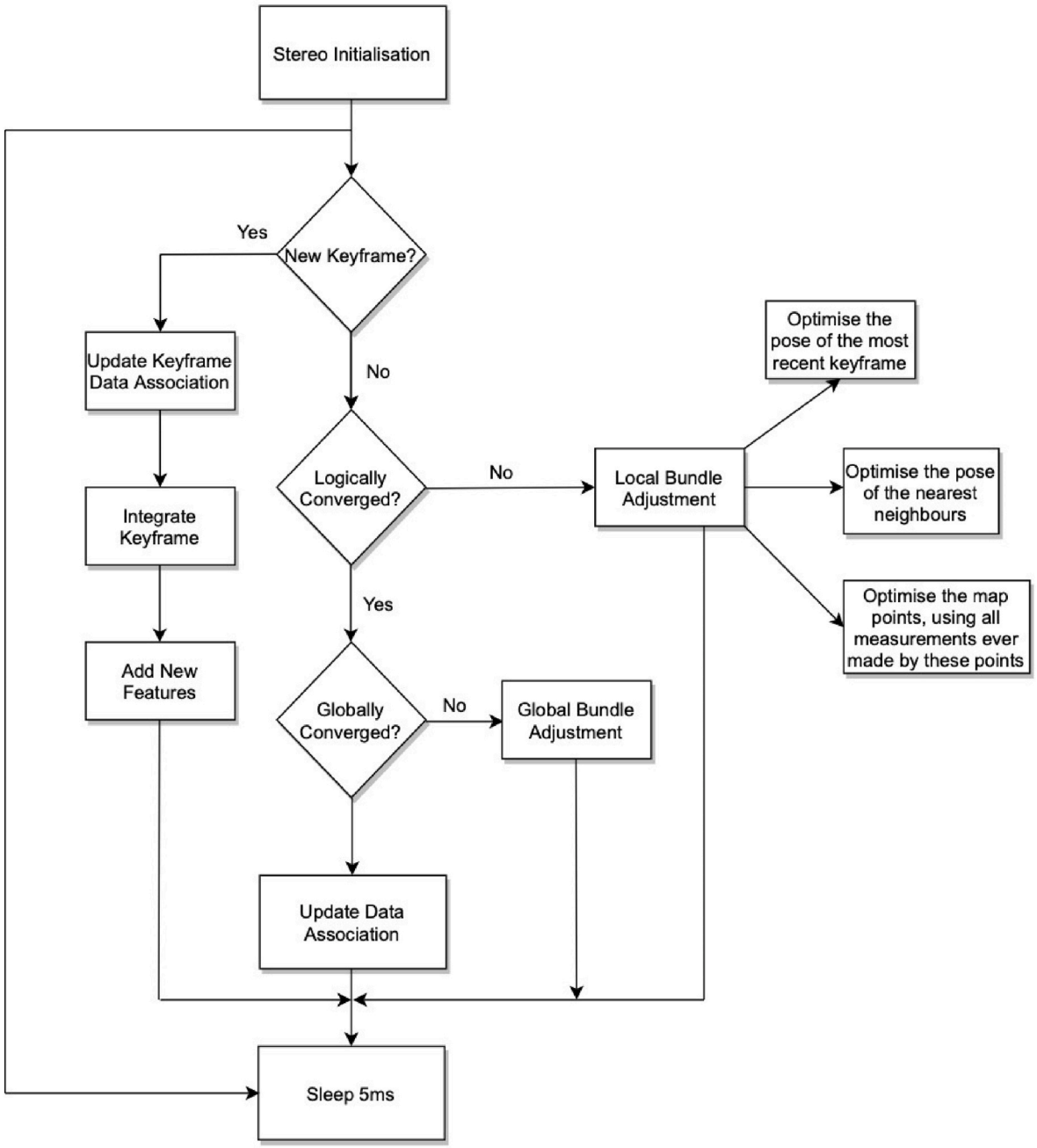

Fig. 2. PTAM system architecture.

feature observations result in updated probabilistic state estimations. A new state is added when new features are observed and, if necessary, feature deletion is also possible.

A given image measurement and camera position cannot directly be used to invert the feature measurement model to determine the location of a new feature since the feature depth is unknown. To determine the depth of a feature, several measurements from different viewpoints are required, as well as the camera's motion. Rather than trying to track the new feature for several frames, it would be better to estimate its depth by triangulating multiple views from multiple views instead of tracking it in the in-game frame. As a result, 2D tracking would be worthless since tracking a moving camera is very difficult. As well, initialization of functions in a camera with a narrow viewing angle must be completed very quickly to prevent them from being overwritten. As an alternative, after identifying and measuring a new feature, an initial 3D line should be drawn along a certain line on the map. The line grows to infinity along the direction of feature viewing, beginning at the estimated position of the camera and having Gaussian uncertainty in parameters.

Images captured with a camera are combined with computer graphics to generate composite scenes in augmented reality. Graphics must appear as if they are anchored to the 3D scene being observed in order to produce a convincing effect. An accurate understanding of the camera's motion is required in order to achieve this. The location of the object can then be fed into a standard 3D graphics engine, which then renders the image correctly.

ORB-SLAM. ORB-SLAM is a feature-based, real-time SLAM system that works outdoors and indoors in a variety of environments. As a result of the robust system, motion clutter can be tolerated, the baseline loop can be closed and the loop re-located, and the system can be fully automatic. ORB-SLAM operates on three threads at the same time: tracking, local mapping, and loop closure.

With every frame, tracking locates the camera and determines when

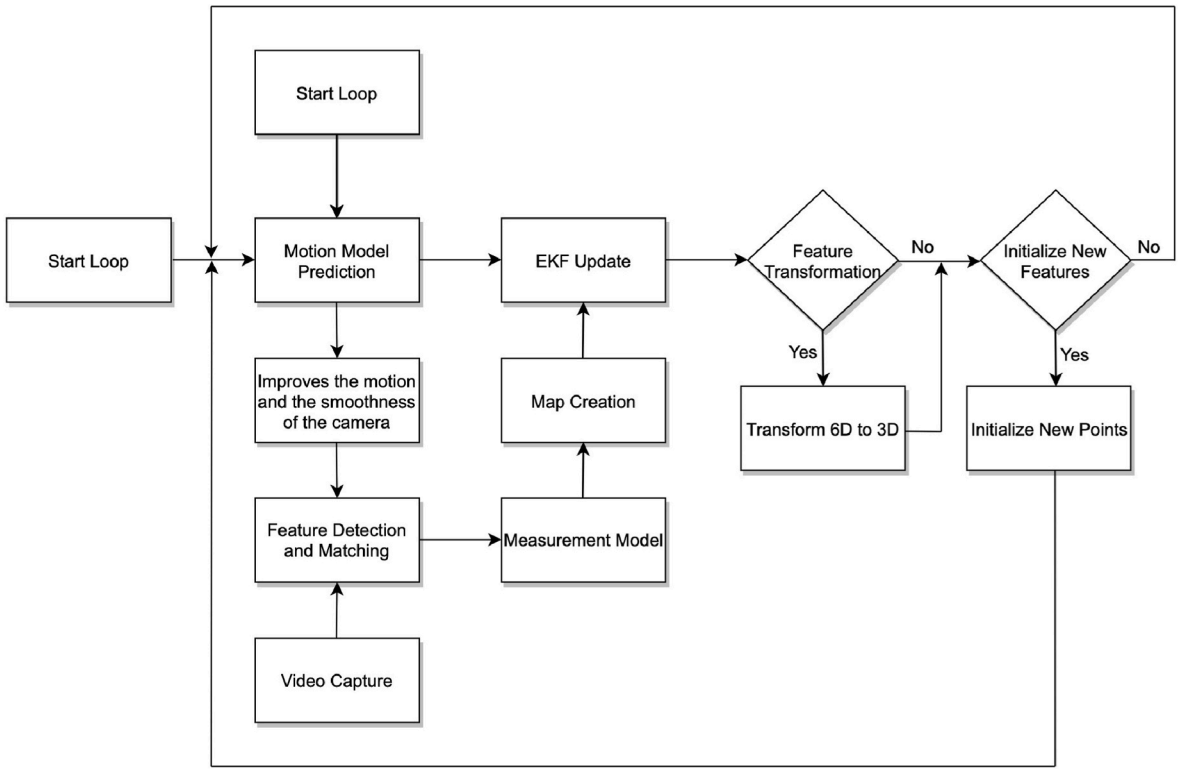

Fig. 3. MonoSLAM system architecture.

a new keyframe should be inserted. In order to optimize the pose, motion-only Bundle Adjustment (BA) was used, along with initial feature matching with the previous frame. In the event that tracking is lost (e.g., due to occlusions or abrupt movement), relocalization is performed globally using the place recognition module. The covisibility graph of keyframes that is maintained by the system is used to retrieve the local visible map based on a first estimate of camera pose and feature matching. After all local map points are found, reprojection is used to find matches, and camera pose optimization is again performed. Last but not least, the tracking thread determines if a keyframe needs to be added.

By processing new keyframes and performing local BA, local mapping is able to reconstruct the surrounding environment efficiently. By finding new correspondences for unmatched ORB, new points are triangulated in the covisibility graph. The point culling policy is applied some time after the points are created, based on the results of the tracking. This ensures that only high-quality points are retained. Also, redundant keyframes are culled by local mapping.

It is important to keep in mind that pixels that do not belong to the model can negatively affect tracking quality. Photometric errors over a certain threshold must be excluded from being included in the analysis. As the least squares method converges, this threshold is lowered with each iteration. As a result, this scheme makes observing unmodeled objects possible while tracking densely.

DTAM. Dense Tracking and Mapping (DTAM) is a real-time tracking and reconstruction approach that does not rely on feature extraction, but rather on dense, pixel-by-pixel tracking. When a large RGB hand-held camera flies over a static scene, a dense patchwork surface of millions of vertices is created. A detailed texture depth map is generated based on keyframes using the algorithm. In order to create a depth map, bundles of frames are reconstructed densely and with sub-pixel resolution.

When estimating the pose of a live camera, one can determine what motion parameters produce the best synthetic viewpoint that matches the live image.

Two stages are involved in refining live camera poses. In the first case, we estimate inter-frame rotations from the model, and then refine the model in the second case, with 6DOF full pose refinement. In both cases, the Lucas-Kanadestyle nonlinear least-squares algorithm is used to minimize an as-measured photometric cost function iteratively. To achieve the global minimum, as soon as the system is initially placed within a convex basin, the true solution must be found. As a final step, we will maximize efficiency by using a power of two image pyramid.

LSD-SLAM. In Large-Scale Direct Monocular SLAM (LSD-SLAM), there are three key components: tracking, depth map estimation, and map optimization. Whenever new camera images are captured, the tracking component automatically keeps track of them. By using the previous frame's pose as the initialization, it determines the rigid body's position based on the current keyframe. Based on tracked frames, the depth map estimation component refines or replaces current keyframes. Through relative comparison between pixels to many smaller baselines over time, and interleaved spatial regularization, the depth can be refined. Whenever the camera moves too far from the current keyframe, points from nearby keyframes are projected into the new keyframe Fig. 4.

In general, keyframes for which tracking references replace keyframes will not be further refined, since their depth maps are incorporated into the global map by the mapping optimization component. In order to detect loop closures and scale drift, a similarity transform is estimated based on scale-aware direct image alignment.

A pose graph is shown in the map as a series of keyframes, after each of which a camera image is shown, an inverse depth map is displayed, and a variance of the inverse depth is displayed. Only pixels containing all regions with sufficient gradients of intensity can be used to calculate a depth map and variance, and thus only on semi-dense scenes. An alignment metric as well as a keyframe's covariance are determined by similarity transforms.

PMDS-SLAM. Probability Mesh Enhanced Semantic SLAM (PMDS-SLAM) [40] divides pixels into meshes and integrates the motion probability information from historic frames. The probabilities are propagated to the new frame meshes. With the use of motion checks, the

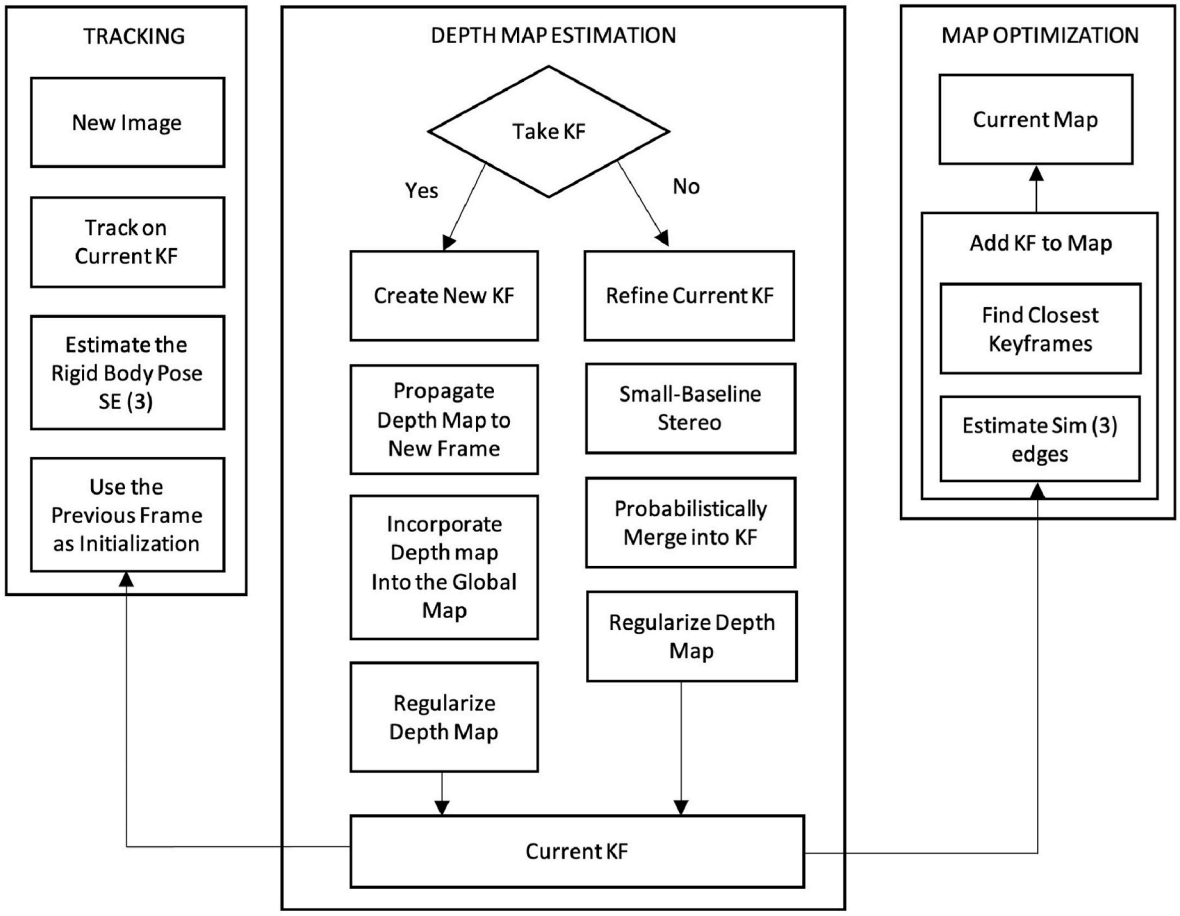

Fig. 4. LSD-SLAM system architecture.

probability of dynamic targets can be updated in the new frame, reducing its impact on tracking. This mesh probability is further used to remove dynamic feature points that are highly likely.

During PMDS-SLAM, images are tracked and captured, and the semantic information of each pixel is extracted using the Mask-RCNN segmentation technology. The superpoint segmentation technology for segmenting the current frame into a superpoint mesh is then used. The initial mesh probability is generated from that semantic prior information, and then propagated through history and into current time. As the current frame contains positions where motion is present, it calculates the motion state for superpoint mesh points at those locations. These meshes are then updated using the Bayesian probability formula. Using the dynamic area mask generated by the tracking thread based on the mesh probability, no real moving features can be observed in the result.

Camera pose is calculated using only static feature points that match each other Fig. 5.

In PMDS-SLAM, the image details are subdivided and all the targets in the scene are segmented using superpoint segmentation, as opposed to deep learning's semantic segmentation. To achieve the segmentation effect, it sprinkles superpoints randomly over the RGB input image and iteratively extends the range of superpoints as necessary. In this case, the target is not separated individually, but instead it's subdivided. Using a fast implementation method [14], this paper segments the image into superpixels by SLIC [13]. This approach can allow the semantic segmentation network to segment targets that are unrecognizable with greater accuracy, as well as pinpoint the motion feature point area more precisely, eliminating the whole contour feature point of targets that are due to partial joint motion. In comparison to ORB-SLAM2, PMDS-SLAM can significantly improve a low dynamic sequence's accuracy by more than 27.5%.

More than 90% of the improvements can be achieved for scenes that have high dynamic range. As a result of PMDS-SLAM, dynamic objects are eliminated from interference, thereby reducing pose errors.

VPS-SLAM. In the world of visual planar semantic SLAM (VPS-SLAM) [41], a lightweight and real-time framework is developed. As part of this method, visual/visual-inertial odometry (VO/VIO) is applied to the geometrical data representing planar surfaces derived from semantic objects. Using planar surfaces to estimate shapes and sizes of selected semantic objects allows for rapid, highly accurate improvements in metrics. A graph-based approach utilizing several state-of-the-art VO/VIO algorithms and the latest object detectors can estimate the robot's six degrees of freedom pose while simultaneously generating a sparse semantic model of the environment Fig. 6. No prior knowledge of the objects is needed for this approach.

Any object-based detector can be used to detect semantic objects in VPS-SLAM.

3.2. Stereo based

Stereo based vSLAM rely on feature points to estimate the camera trajectory and build a map of the environment. Feature points usually are points from all the edges in an environment. The performance of such algorithms is affected in low-textured environments, where it is sometimes difficult to find a sufficient number of reliable point features.

DS-PTAM. A stereo vision-based approach to SLAM is Distributed Stereo Parallel Tracking and Mapping (DS-PTAM). Its purpose is to build a map of an environment in which a robot can operate in real time while getting an accurate estimate of its position. By dividing tracking and mapping tasks into two independent execution threads and performing them both in parallel, S-PTAM achieves very good performance

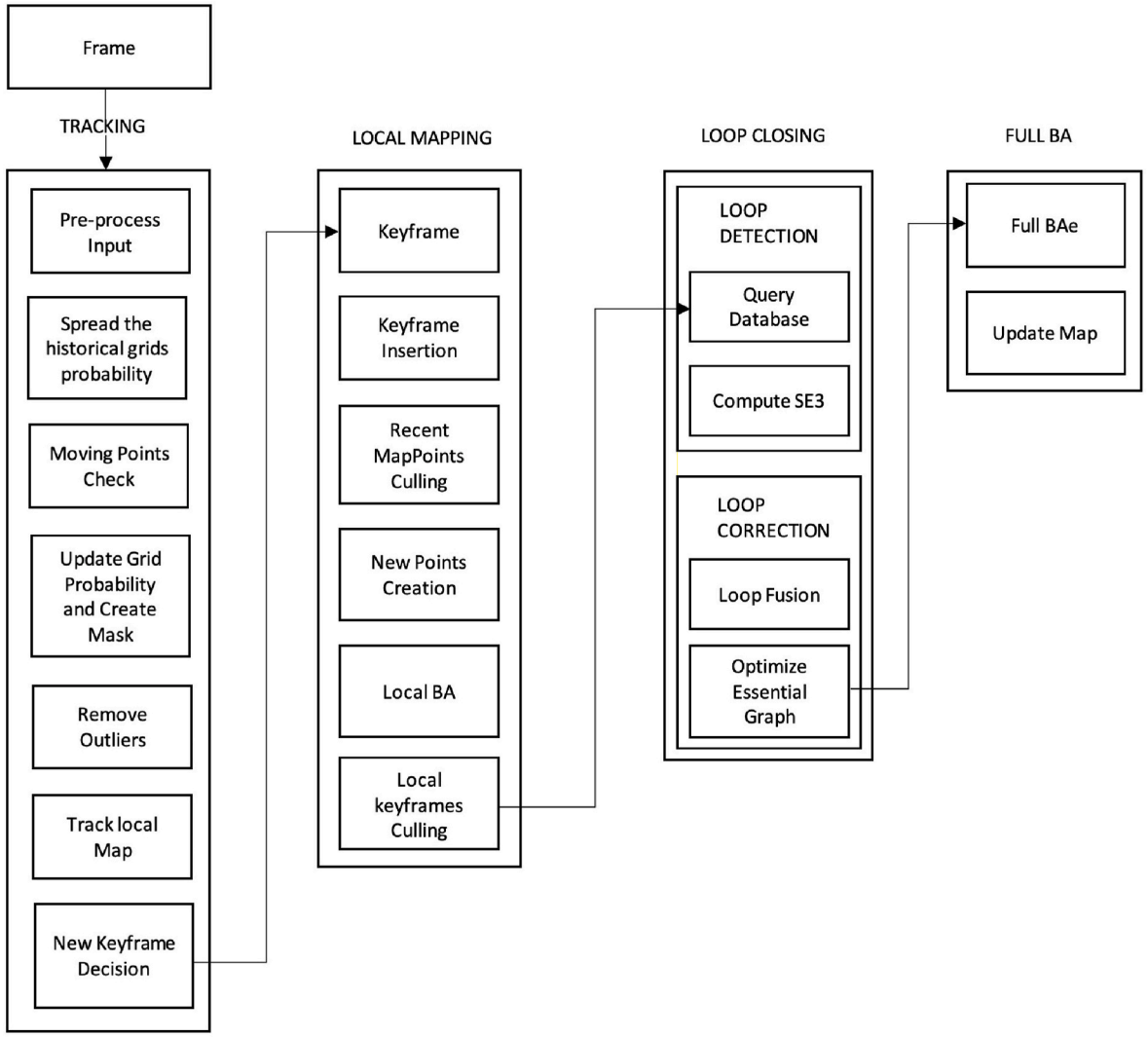

Fig. 5. PMDS-SLAM system architecture.

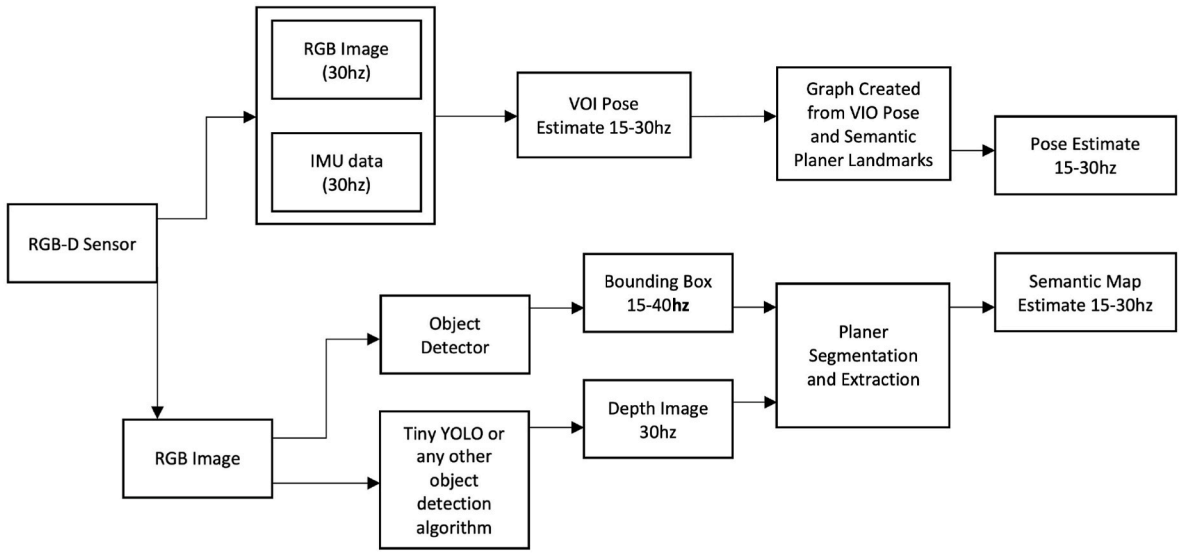

Fig. 6. VPS-SLAM system architecture.

compared to other current SLAM methods. S-PTAM assumes that the stereo camera is at the origin of world coordinates and doesn't know what its surroundings are at the beginning of the process.

The map is then initialized using the extracted features from the left and right frames using a triangulation process. Points on a map are known as triangulations. Based on the current map, the next stereo frame will be located based on the current pose of the camera. Based on previous poses, a decay-velocity model is used to determine the camera's initial pose. By processing each image in three dimensions and then plotting each point on a plane, we adjust this estimation by comparing each of its descriptors to the features that are extracted from it. In this procedure, correspondences are obtained from an observation point in space that was made by the stereo camera. These correspondences are referred to as 2D-3D matches or constraints. Additionally, stereo frames from unknown regions around the world are selected to be included in the map, which adds constraints and new points to the map Fig. 7. These frames are known as keyframes.

After moving along its path, it will obtain new stereo images that will be processed using the described method, resulting in an incremental map whose size increases with each successive iteration. The Tracker module is responsible for this functionality. A second execution thread called Mapper will run concurrently with Tracker, which will adjust camera positions and point locations, i.e., adjust the map. Bundle Adjustment is used for these refinements. Also, this thread increases the constraints between keyframes and map points by finding new matches. Further, both map points and measurements considered to be unreliable or spurious are removed.

DOC-SLAM. The Dynamic Object Culling SLAM (DOC-SLAM) [42] system is a stereo SLAM that is able to achieve good performance in highly dynamic environments by removing the actual moving objects. By combining the semantic information from panoptic segmentation together with the optical flow points, DOC-SLAM can detect potential moving objects. To accomplish dynamic object culling, a moving consistency check module determines and removes feature points in objects which are in motion.

Utilizing a direct method of estimating camera trajectory, this method reduces the time that's consumed with feature extraction and tracking. To remove the objects in motion I propose a moving consistency check module, which is an alternative to the feature-based method, which measures match points by re-projection error. In Fig. 8, static point extraction is followed by dynamic point culling to extract the key points.

VINS-Fusion (stereo) is the base on which DOC-SLAM's localization module is built. In static scenes, VINS-Fusion achieves accurate self-localization by using optimization-based state estimation.

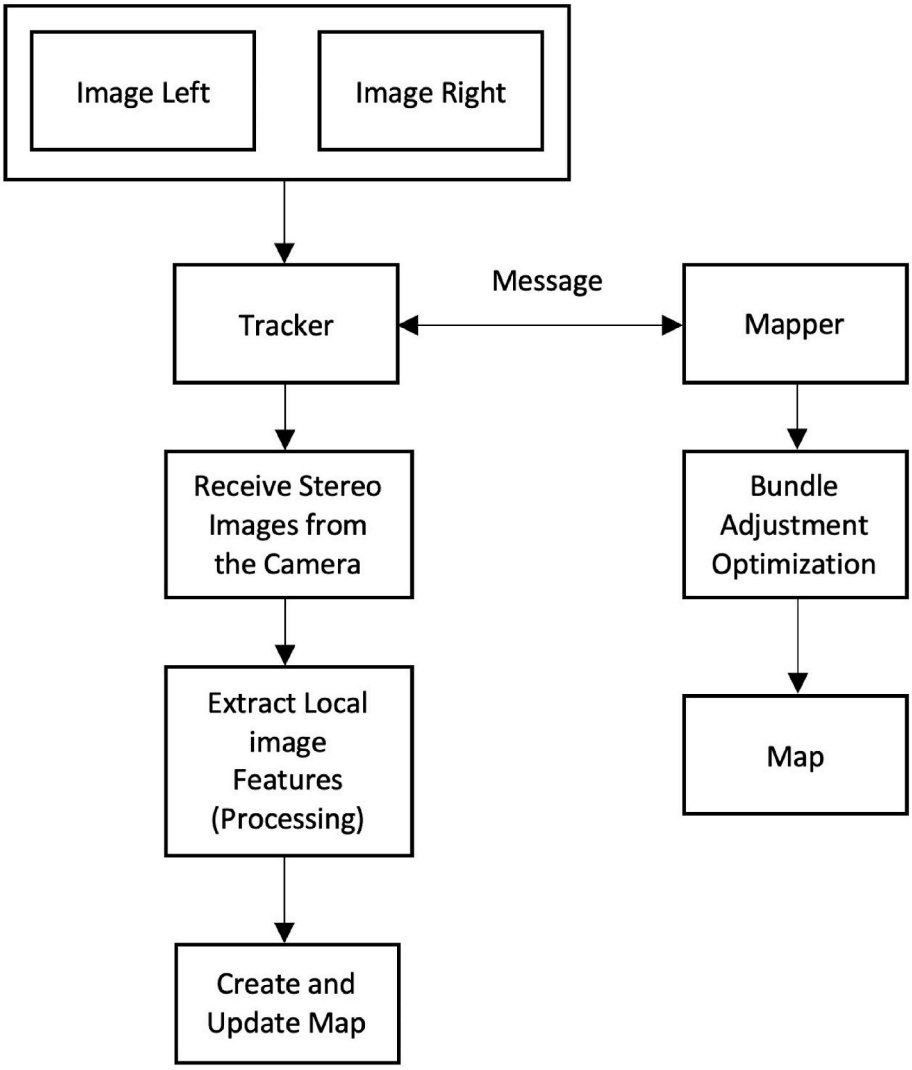

Fig. 7. DS-PTAM system architecture.

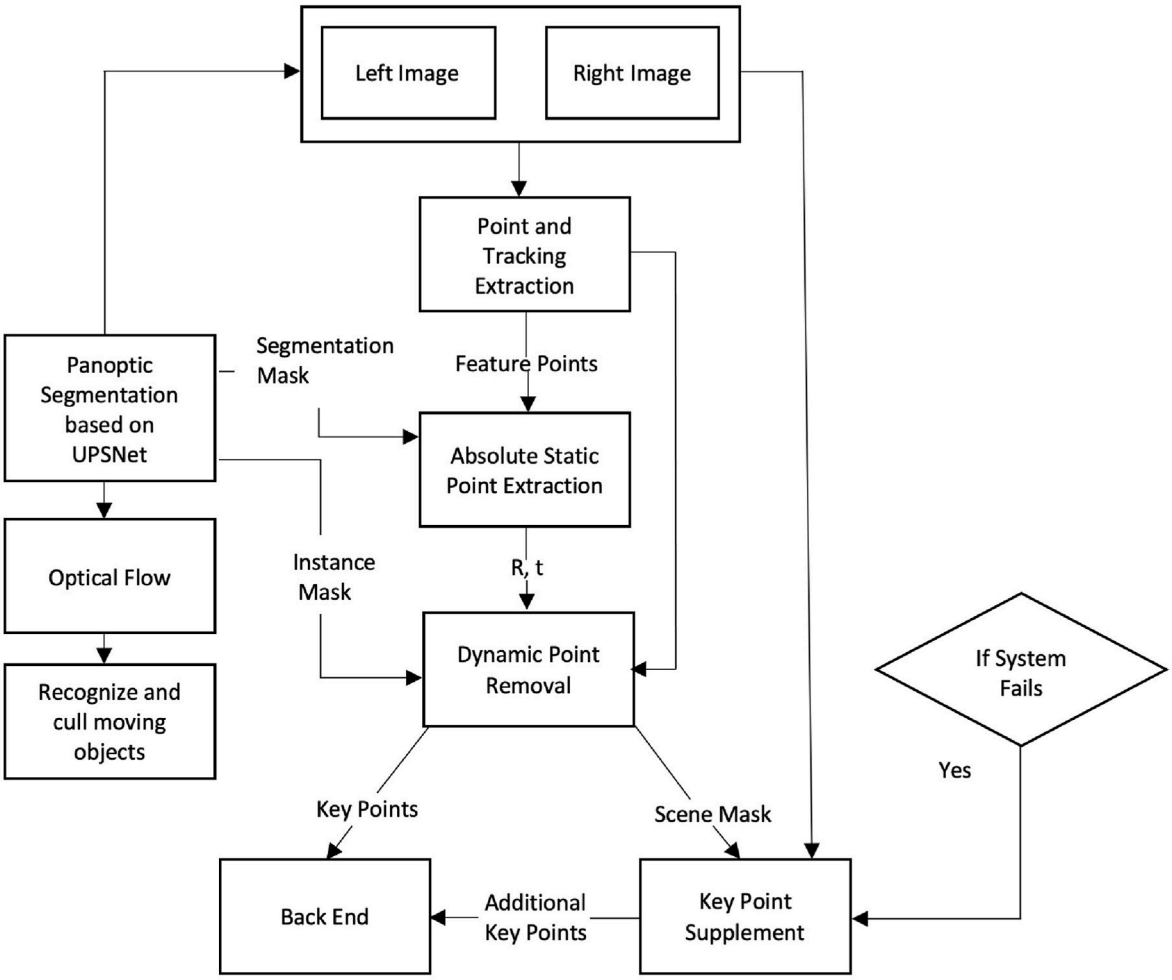

Fig. 8. DOC-SLAM system architecture.

3.3. Monocular and stereo based

Monocular and Stereo based vSLAM algorithms can perform mapping, tracking and wayfinding by using both a sequence of images or just feature points.

ORB-SLAM2. The ORB-SLAM2 system [43] is an integrated SLAM system for monocular, stereo, and RGB-D cameras that offers map reuse, loop closing, and relocalization functions. This system works by using standard CPUs in different environments ranging from small handheld devices inside the home to drones flying in factories and cars traveling through city streets. Bundled adjustments combined with metric scale observations allow for accurate trajectory estimation on the back end. For localization, the system provides a lightweight mode that utilizes visual odometry to track non-mapped regions and match those tracks to map points to ensure zero drift.

During ORB-SLAM2, a full Bundle Adjustment (BA) optimization is applied to reach the optimum solution. Since optimization would require a lot of resources, it was performed in a separate thread, thus allowing the system to create maps and identify loops while optimization was being performed. As a consequence, merging the bundle adjustment output with the existing map is challenging. During the optimization, a new loop may occur, causing the optimization to be aborted and the loop to be closed, triggering the BA Optimization process again. Upon completion of the full BA, the updated subset of keyframes should be merged, and all points that were inserted during the optimization process should be optimized by the full BA. Through the spanning tree, the corrected keyframes are propagated to the non-updated keyframes. Based on the correction applied to their reference keyframe, the non-updated points are transformed in Fig. 9.

Whenever environmental conditions do not change significantly in the long run, the localization mode can be used to enable lightweight long-term localization. If necessary, the tracking processes relocalize the camera continuously in this mode without deactivating the local mapping or loop closure threads. As part of this mode, points are mapped using visual odometry matches. In visual odometry, the 3D points that are created at each point of the current frame at the same position as the 3D points generated in the previous frame are matched with the ORB in the current frame. Localization is robust to regions that have not been mapped, but drift may accumulate. By matching map points, we ensure the existing map remains localized at all times.

DynaSLAM. The capability of dynamic object detection and background inpainting is added to ORB-SLAM2 using DynaSLAM. Whether it is monocular, stereo, or RGB-D, DynaSLAM works well in dynamic scenarios. The system is capable of detecting moving objects either using deep learning or multiview geometry. Static maps of scenes make it possible to inpaint frame backgrounds obscured by dynamic objects.

After segmenting the potentially dynamic content, the pose of the camera is tracked by analyzing the static part of the image. Because high-gradient areas tend to appear in segment contours, salient features tend to stand out. Such contour areas do not have features taken into account. This is a simpler, lighter version of ORB-SLAM2's tracking, which is implemented at this stage of the algorithm. In Fig. 10, the algorithm consists of projecting map features into an image frame, verifying correspondences within the static areas of the image, 460 and

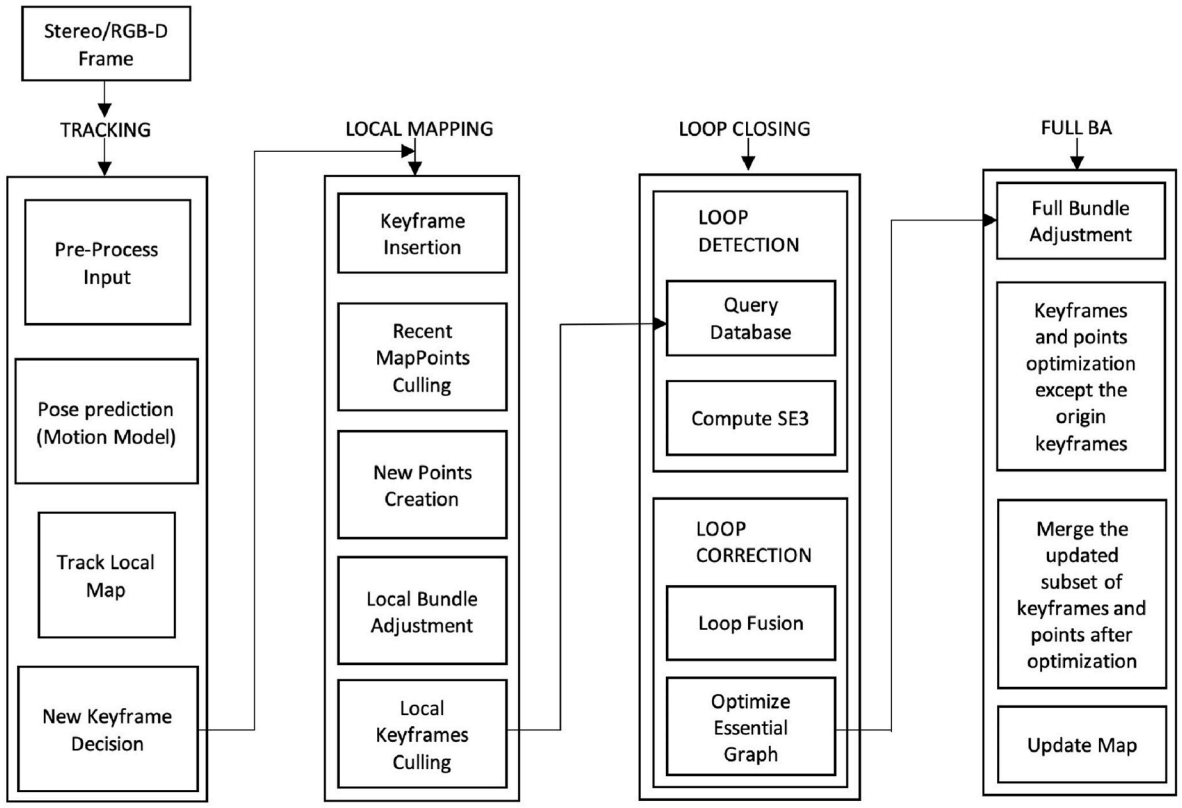

Fig. 9. ORB-SLAM2 system architecture.

optimizing the camera pose by minimizing re-projected errors.

For every removed dynamic object, background inpainting is used to reconstruct a realistic image by taking information from previous views and painting over the occluded background. After the map has been created, the synthetic frames may be used to relocate and track cameras, as well as for applications 465 such as virtual and augmented reality.

Finally, the only limitation of DynaSLAM is that it is less accurate in scenes with dynamic objects.

ORB-SLAM3. Using pin-hole and fisheye lens models, ORB-SLAM3 [44] is the first system that can perform visual, visual-inertial, and multimap SLAM.

It is the first system to rely on maximum a posteriori (MAP) estimation during the initialization of the inertial measurement unit, resulting in two to ten times higher accuracy as compared to other approaches in small and large, indoor, and outdoor environments. A new mapping method allows ORB-SLAM3 to survive periods of poor visual data, as it continues with an updated map when visual data become unavailable for some reason, while combining previous maps seamlessly as new data becomes available. In comparison with conventional odometry systems, ORB-SLAM3 maintains all the previously processed keyframe information from equally visible frames from each stage, even if they are far apart in time or from previous sessions, increasing overall accuracy.

Sensor data is processed by tracking threads, which compute the position of the current frame relative to the current map in real time. Feature projections with matched features can be made with minimal error. Further, it determines if a keyframe should be applied to the current frame. Visual-inertial modes estimate body velocity and IMU bias by including residuals from inertial sensors during optimization. Tracking threads that lose tracking attempt to relocalize the current frame throughout all Atlas' maps [44]. The tracking will resume if it has been relocalized, and the active map will be switched. If the active map is not initialized after a certain time, it becomes inactive and is stored as nonactive until it can be re-initialized from scratch.

4. Comparison

Monocular vSLAM algorithms such as MonoSLAM, PTAM, PMDS-SLAM, LSD-SLAM and ORB-SLAM lack the ability to perform well and fast in large, crowded indoor environments. This is due to various reasons. For example, due to MonoSLAM's deterministic nature, it is difficult to estimate an exact normal vector at each feature location due to the relatively simple texture patterns associated with many features, such as black on white corners, for which full warp estimation is not possible. In addition, the algorithm must be improved to handle large indoor and outdoor environments. Moreover, ORB-SLAM's performance is not robust to changes in illumination while tracking and to global illumination changes can occur in real life.

On the other hand, stereo base vSLAM algorithms such as DS-PTAM, it is tracker configuration and map configuration are prone to higher computational costs and take a longer time to execute. Because map updates are so complex, sending and receiving them was complicated. DS-PTAM has another limitation, tracking is estimated slower because the configuration works better on an unoptimized map.

Monocular and stereo based vSLAM algorithms such as ORB-SLAM2, their optimization process requires a lot of resources and merging the bundle adjustment output with the existing map is challenging. Also, if a new loop occurs, the optimization is stopped, the loop is closed, and the BA optimization starts over.

Both Monocular and Stereo based vSLAM algorithms use Bundle Adjustment and some of them use Local Bundle Adjustment. In feature-based monocular SLAM, bundle adjustment plays an important role. As part of the 6DOF camera trajectory and 3D point cloud estimation, bundle adjustment is used to estimate the 3D map (3D point cloud) based on input feature tracks. However, SLAM systems using bundle adjustments suffer from two major weaknesses. In the first place, the need for careful initialization of bundle adjustments requires the map to be estimated as accurately as possible and maintained over time, making the overall algorithm complex. A second challenge will arise during

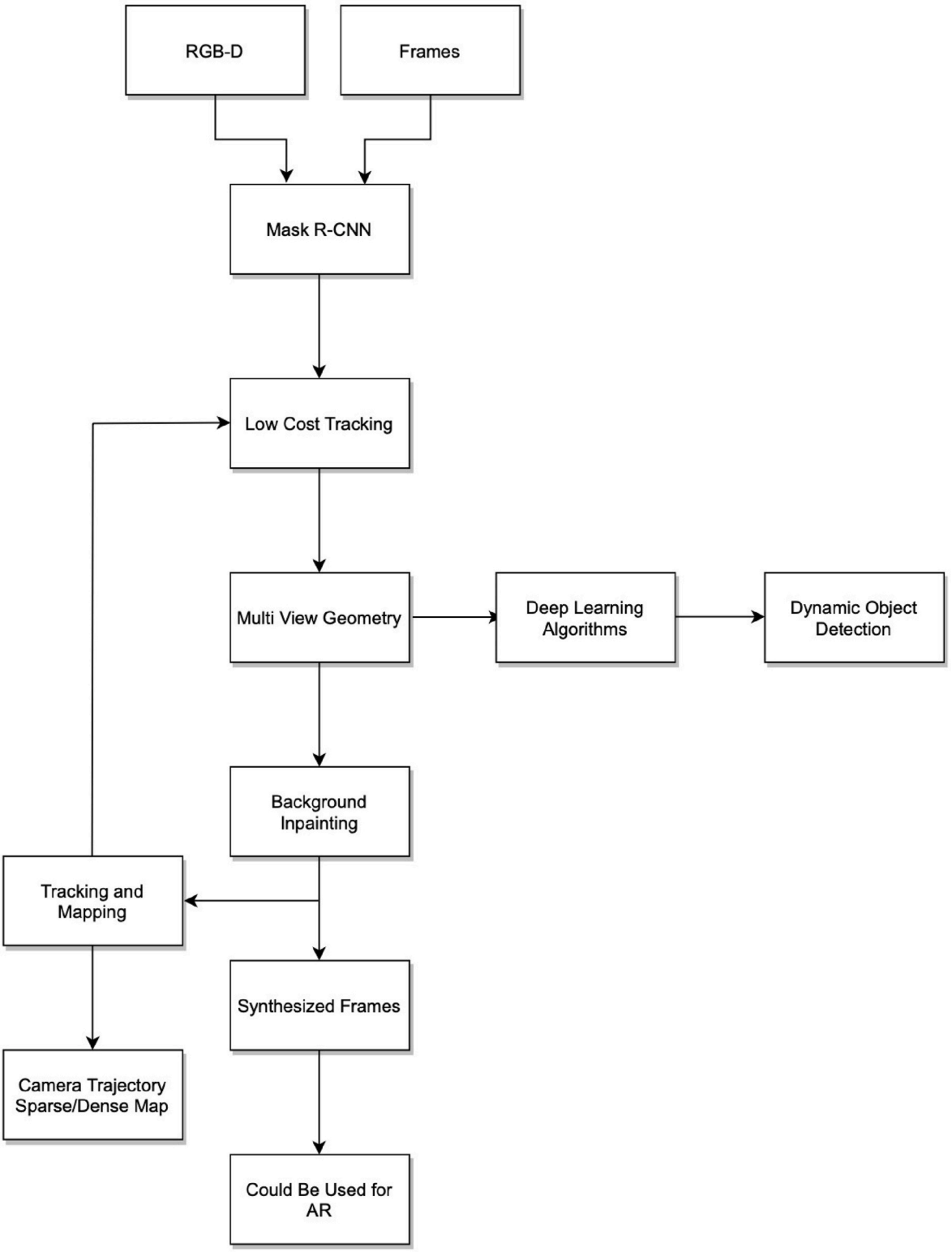

Fig. 10. DynaSLAM system architecture.

periods of slow motion or rotation when the SLAM algorithm has difficulty estimating the 3D structure (which requires an appropriate baseline).

A local bundle adjustment (LBA) is a method of estimating the geometry of image sequences taken by a calibrated camera. This approach has the advantage of reducing computational complexity, allowing real-time processing with comparable accuracy as standard (global) bundle adjustments.

A monocular and stereo based vSLAM algorithm that outperforms every-thing else is ORB-SLAM3. It is the first system that can perform visual, visual inertial, and multimap SLAM. Moreover, it is the first system to rely on maximum a posteriori (MAP) estimation during the initialization of the inertial measurement unit, achieving two to ten times higher accuracy as compared to other approaches in small and

large, indoor, and outdoor environments.

5. SLAM algorithms table

Table 2 shows a general description of the SLAM algorithms for different factors. The **purpose** of the algorithm could be either for general use, AR or Robotics. The **Camera** represents what camera can be used for each algorithm and **Environment** shows in which environment each algorithm works. Finally, Table 2 shows the **Resolution** of each algorithm and their **Estimation**.

6. Datasets

Open-source datasets will be discussed in this section which may be used to test SLAM algorithms. In this section, the most commonly used datasets were discussed, including the KITTI dataset, the EuRoc dataset, and the TUM RGB-D dataset.

KITTI Dataset. Karlsruhe Institute of Technology and Toyota Technological Institute (KITTI) [45] datasets have been used by mobile robotics and autonomous driving research. A variety of sensor technologies were used to record hours of traffic scenarios, including high-resolution RGB, grayscale stereo cameras, and a 3D laser scanner. Although incredibly popular, the dataset does not contain sufficient ground truth to allow for semantic segmentation. A total of 7481 training images are annotated using 3D bounding boxes in the KITTI dataset.

EuRoC Dataset. European RoC MAV [46] is a visual-inertial dataset collected by a Micro Aerial Vehicle (MAV). Synced IMU measurements, as well as motion and structure ground truths, are present in the dataset. Visual-inertial localization algorithms can be designed and evaluated with the dataset.

TUM RGB-D Dataset. TUM RGB-D [47] is a dataset containing images which contain colour and depth information collected by a Microsoft Kinect sensor along its ground-truth trajectory. Recording was done at full frame rate (30 Hz) and sensor resolution (640 × 480). Ground-truth trajectory information was collected from eight high-speed tracking cameras (100 Hz), using high-precision motion capture.

7. Evaluation metrics

Relative pose error (RPE). Based on a determined time interval Δ, relative pose error is a measure of the local accuracy of the trajectory. Hence, the relative pose error is indicative of the drift of the trajectory which is especially relevant for evaluating visual odometry systems. According to this definition, the relative pose error at time step i is

$$E_i := \left(Q_i e^{-1} Q_i + \Delta\right)^{-1} \left(P_i^{-} 1 P_i + \Delta\right) \tag{1}$$

The relative pose errors along a sequence of n camera poses are obtained in this way: m = n -/*Delta*. Overall time indices of the translational component, the root mean squared error (RMSE) was calculated as follows:

$$RMSE(E_{1:n}, \Delta) := \left(\frac{1}{m}\sum_{i=1}^{m} \|trans(E_i)\|^2\right)^{\frac{1}{2}} \tag{2}$$

The translational components of the relative pose error E_i are denoted by trans (E_i). In certain situations, root mean square error is preferred rather than mean error since outliers are less affected. It is also possible to compute the median rather than the mean instead, which gives outliers less influence. Also, the rotational error can be evaluated. However, most of the time the translational errors are sufficient for comparison (since rotational errors are translated by the camera when it is moved).

For systems attempting to match consecutive frames, it is necessary to incorporate the time parameter Δ = 1 which indicates the drift per frame *RMSE* ($E_{1:n}$).

When using multiple previous frames in a system, larger distributions of/*Delta* can be appropriate. For example, Δ = 20 gives the drift per second for a sequence recorded at 20 Hz. One commonly chosen (but poor) method of comparing the start point and the end point is to set Δ = n. Because this metric penalizes rotational errors more towards the end of the trajectory [48,49], it is misleading. In order to evaluate SLAM systems, it makes sense to average over all time intervals Δ for example, to compute

$$RMSE(E_{1:n}) := \frac{1}{n}\sum_{\Delta=1}^{n} RMSE(E_{1:n}, \Delta) \tag{3}$$

This expression has quadratic computational complexity in terms of trajectory length. Accordingly, it was proposed [46] that it could be approximated by composing a set of relative pose samples from a fixed number of locations.

Absolute trajectory error (ATE). For vSLAM systems, the absolute distance between the estimated and the ground truth trajectory is another important metric that can be used to assess the global consistency of the estimated trajectory. Due to the fact that both trajectories can be specified in any coordinate frame, they must be aligned first. With the Horn method [50], one can obtain a rigid-body transformation S that maps the estimated trajectory $P_{i:n}$ onto the ground truth trajectory $Q_{1:n}$ using the least-squares solution. As a result of this transformation, the absolute trajectory error can be computed as follows:

$$F_i = Q_i^{-1} S P_i \tag{4}$$

For translational components, it was proposed [47] to calculate root mean squared error over all time indices for each component, for instance

Table 2
Slam algorithms.

Algorithm	Purpose	Camera	Environment	Resolution	Estimation
MonoSLAM	General	Monocular	Indoor	Low	EKF
PTAM	AR	Monocular	Indoor	Low	BA
DS-PTAM	Robotics	Stereo	Indoor/Outdoor	Low-High	BA
PTAM-DENSE	Robotics	Monocular	Indoor	Low	BA
ORB-SLAM	General	Monocular	Indoor/Outdoor	Low-High	Local-BA
ORB-SLAM2	General	Monocular/Stereo	Indoor/Outdoor	Low-High	Local-BA
PL-SLAM	General	Monocular	Indoor	Low-High	BA
DTAM	General	Monocular	Indoor	Low	Local-BA
LSD-SLAM	General	Monocular	Indoor/Outdoor	Low-High	PG
SLAM++	General	Depth	Indoor	Low	Local-BA
DynaSLAM	General	Monocular/Stereo	Indoor/Outdoor	Low-High	BA
DOC-SLAM	Robotics	Stereo	Indoor/Outdoor	Low-High	Local-BA
PMDS-SLAM	Robotics	Monocular	Indoor/Outdoor	Low-High	Full BA
VPS-SLAM	General	Monocular	Indoor/Outdoor	Low-High	Local-BA
ORB-SLAM3	General	Monocular	Indoor/Outdoor	High	Local BA

$$RMSE(F_{1:n}) := \left(\frac{1}{n}\sum_{i=1}^{n}\|trans(F_i)\|^2\right)^{\frac{1}{2}} \quad (5)$$

By averaging the data over all possible time intervals, the RPE can also be used to evaluate the overall error of a trajectory. Translational and rotational errors are taken into account by the RPE, while only translational errors are considered by the ATE. Therefore, the RPE metric provides an elegant way to combine rotational and translational errors into a single measure. The ATE, however, generally also detects rotational errors indirectly because they show up in wrong translations.

7.1. Limitation of evaluation metrics

Many researchers today still use older techniques and algorithms to evaluate the accuracy of their SLAM algorithm. Those evaluation approaches have many limitations i.e do not work for all algorithms and especially recent ones and do not always work if the environment is large and full of obstacles. Those limitations bring the motivation to explore new ways that could potentially be used to accurately evaluate SLAM algorithms.

8. Discussion

AR systems demonstrate the great potential of visual SLAM algorithms to deal with the registration problem. The mapping of the environment enables the user to include virtual objects in their view according to their point of observation along with solving the occlusion of virtual elements by real elements by utilizing the tracking of the sensor device's pose. The results of this re620 search show a trend in the use of conventional (monocular) devices as a sensor. Among the described algorithms, ORB-SLAM can be considered the state of the art among those who use a single camera as a sensor. Although the PL-SLAM Monocular is robust, especially in environments that are poor in texture, this result is achieved at the expense of high computational capacity. The stereo version of these algorithms, ORB-SLAM2 and PL-SLAM Stereo, respectively, show similar results, although there are inherent advantages to stereo technology, such as easier scaling and map initialization. Among the algorithms that are based on depth sensors, associated or not with RGB cameras, some present promising results since the behaviour that is invariant to the ambient light is an important characteristic of these sensors. However, depth sensors are not as commonly found in devices as RGB cameras and do not usually perform well outdoors, due to the use of infrared rays.

Among the advantages that can be pointed out of the use of visual SLAM algorithms for AR applications are the high availability of low-cost cameras, both for desktop computers and for mobile devices, and the collection of non-invasive information made by optical devices. Among the disadvantages of working in low light environment there is still a challenge of initializing the map in the monocular configuration, and the difficulty of working outdoors with infrared depth cameras.

9. Conclusion

This work carried out for this survey was to explore the main techniques of visual SLAM developed in recent years with the aim of identifying their fundamental characteristics. The results show that the solutions developed for general purposes are the majority, although AR is becoming a potential application for dedicated algorithms. The most common solutions are those that perform in real time without requiring prior knowledge of the environment. In addition, the predominance of applications performed in internal environments is due to the limitation of sensors in acting externally. This scenario should only change if the sensor technology evolves to the point of overcoming this limitation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Fuentes-Pacheco J, Ruiz-Ascencio J, Rend′on-Mancha JM. Visual simultaneous localization and mapping: a survey. Artif Intell Rev 2015;43(1):55–81.

[2] Queralta JP, Yuhong F, Salomaa L, Qingqing L, Gia TN, Zou Z, Tenhunen H, Westerlund T. Fpga-based architecture for a low-cost 3d lidar design and implementation from multiple rotating 2d lidars with ros. In: 2019 IEEE sensors; 2019. p. 1–4. https://doi.org/10.1109/SENSORS43011.2019.8956928.

[3] Sheikh R, OBwald S, Bennewitz M. A combined rgb and depth descriptor for slam with humanoids. In: 2018 IEEE/RSJ international conference on intelligent robots and systems (IROS). IEEE; 2018. p. 1718–24.

[4] Shim JH, Im Cho Y. A visual localization technique for unmanned ground and aerial robots. In: 2017 first IEEE international conference on robotic computing (IRC). IEEE; 2017. p. 399–403.

[5] An P, Liu Y, Zhang W, Jin Z. Vision-based simultaneous localization and mapping on lunar rover. In: 2018 IEEE 3rd international conference on image, vision and computing (ICIVC). IEEE; 2018. p. 487–93.

[6] Zhang Q, Niu B, Zhang W, Li Y. Feature-based ukf-slam using imaging sonar in underwater structured environment. In: 2018 IEEE 8th international conference on underwater system technology: theory and applications (USYS). IEEE; 2018. p. 1–5.

[7] Xie C, Yao T, Wang J, Liu Q. Endoscope localization and gastrointestinal feature map construction based on monocular slam technology. J Infect Publ Health 2020; 13(9):1314–21.

[8] Davison AJ, Reid ID, Molton ND, Stasse O. Monoslam: real-time single camera slam. IEEE Trans Pattern Anal Mach Intell 2007;29(6):1052–67.

[9] Klein G, Murray D. Parallel tracking and mapping for small ar workspaces. In: 2007 6th IEEE and ACM international symposium on mixed and augmented reality. IEEE; 2007. p. 225–34.

[10] S. Lovegrove, Parametric dense visual slam.

[11] Newcombe RA, Lovegrove SJ, Davison AJ. Dtam: dense tracking and mapping in real-time. In: 2011 international conference on computer vision. IEEE; 2011. p. 2320–7.

[12] Salas-Moreno RF, Newcombe RA, Strasdat H, Kelly PH, Davison AJ. Slam++: simultaneous localisation and mapping at the level of objects. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2013. p. 1352–9.

[13] Ai Y, Rui T, Lu M, Fu L, Liu S, Wang S. Ddl-slam: a robust rgb-d slam in dynamic environments combined with deep learning. IEEE Access 2020;8:162335–42.

[14] Basiratzadeh S, Lemaire ED, Baddour N. Augmented reality approach for marker-based posture measurement on smartphones. In: 2020 42nd annual international conference of the IEEE engineering in medicine Bi695 ology society (EMBC); 2020. p. 4612–5. https://doi.org/10.1109/EMBC44109.2020.9175652.

[15] Huletski A, Kartashov D, Krinkin K. Evaluation of the modern visual slam methods. In: 2015 artificial intelligence and natural language and information extraction, social media and web search FRUCT conference (AINL-ISMW FRUCT). IEEE; 2015. p. 19–25.

[16] Taketomi T, Uchiyama H, Ikeda S. Visual slam algorithms: a survey from 2010 to 2016. IPSJ Trans Comput Vis Appl 2017;9(1):1–11.

[17] Covolan JPM, Sementille AC, Sanches SRR. A mapping of visual slam algorithms and their applications in augmented reality. In: 2020 22nd symposium on virtual and augmented reality (SVR). IEEE; 2020. p. 20–9.

[18] Munguia-Silva R, Mart′ınez-Carranza J. Autonomous flight using rgb-d slam with a monocular onboard camera only. In: 2018 international conference on electronics, communications and computers (CONIELECOMP). IEEE; 2018. p. 200–6.

[19] Li Y, Lang S. A stereo-based visual-inertial odometry for slam. In: 2019 Chinese automation congress (CAC). IEEE; 2019. p. 594–8.

[20] Wang S, Yue J, Dong Y, Shen R, Zhang X. Real-time omnidirectional visual slam with semi-dense mapping. In: 2018 IEEE intelligent vehicles symposium (IV). IEEE; 2018. p. 695–700.

[21] Jo H, Jo S, Cho HM, Kim E. Efficient 3d mapping with rgb-d camera based on distance dependent update. In: 2016 16th international conference on control, automation and systems (ICCAS). IEEE; 2016. 720 873–875.

[22] Zafari FG, A and leung kk. A survey of indoor localization systems and technologies. IEEE Commun Surv Tutor 2019;21(3).

[23] Ellwood SA, Newman C, Montgomery RA, Nicosia V, Buesching CD, Markham A, Mascolo C, Trigoni N, Pasztor B, Dyo V, et al. An active-radio-frequency-identification system capable of identifying colocations and social-structure: validation with a wild free-ranging animal. Methods Ecol Evol 2017;8(12): 1822–31.

[24] Misono Y, Goto Y, Tarutoko Y, Kobayashi K, Watanabe K. Development of laser rangefinder-based slam algorithm for mobile robot navigation. In: SICE annual conference 2007. IEEE; 2007. p. 392–6.

[25] Demim F, Nemra A, Boucheloukh A, Louadj K, Hamerlain M, Bazoula A. Robust svsf-slam algorithm for unmanned vehicle in dynamic environment. In: 2018 international conference on signal, image, vision and their applications (SIVA). IEEE; 2018. p. 1–5.

[26] Engel J, Schöps T, Cremers D. Lsd-slam: large-scale direct monocular slam. In: European conference on computer vision. Springer; 2014. p. 834–49.

[27] Mur-Artal R, Montiel JMM, Tardos JD. Orb-slam: a versatile and accurate monocular slam system. IEEE Trans Robot 2015;31(5):1147–63.
[28] Korkishko YN, Fedorov V, Prilutskiy V, Ponomarev V, Fedorov I, Kostritskii S, Morev I, Obuhovich D, Prilutskiy S, Zuev A, et al. Highprecision inertial measurement unit imu-5000. In: 2018 IEEE international symposium on inertial sensors and systems (INERTIAL). IEEE; 2018. p. 1–4.
[29] Quan K, Xiao B, Wei Y. Intelligent descriptor of loop closure detection for visual slam systems. In: 2019 Chinese control and decision conference (CCDC). IEEE; 2019. p. 993–7.
[30] Deng C, Luo X, Zhong Y. Improved closed-loop detection and octomap algorithm based on rgb-d slam. In: 2020 IEEE international conference on artificial intelligence and computer applications (ICAICA). IEEE; 2020. p. 73–6.
[31] Zheng J, Zhang H, Kong W, Tang K. A slam loop closure algorithm of bow incorporating the gray level of pixel. In: 2020 international conference 755 on computer vision, image and deep learning (CVIDL). IEEE; 2020. p. 360–3.
[32] Qian K, Zhao W, Li K, Ma X, Yu H. Visual slam with boplw pairs using egocentric stereo camera for wearable-assisted substation inspection. IEEE Sensor J 2019;20 (3):1630–41.
[33] Ye C, Hong S, Tamjidi A. 6-dof pose estimation of a robotic navigation aid by tracking visual and geometric features. IEEE Trans Autom Sci Eng 2015;12(4): 1169–80.
[34] Fu Z, Quo Y, Lin Z, An W, Fsvo. Semi-direct monocular visual odometry using fixed maps. In: 2017 IEEE international conference on image processing (ICIP). IEEE; 2017. p. 2553–7.
[35] Bescos B, F′acil JM, Civera J, Neira J. Dynaslam: tracking, mapping, and inpainting in dynamic scenes. IEEE Rob Autom Lett 2018;3(4):4076–83.
[36] Liu K, Sun H, Ye P. Research on bundle adjustment for visual slam under large-scale scene. In: 2017 4th international conference on systems and informatics (ICSAI). IEEE; 2017. p. 220–4.
[37] Pumarola A, Vakhitov A, Agudo A, Sanfeliu A, Moreno-Noguer F. Plslam: real-time monocular visual slam with points and lines. In: 2017 IEEE international conference on robotics and automation (ICRA). IEEE; 2017. p. 4503–8.
[38] Butt MM, Zhang H, Qiu X, Ge B. Monocular slam initialization using epipolar and homography model. In: 2020 5th international conference on control and robotics engineering (ICCRE). IEEE; 2020. p. 177–82.
[39] Spournias A, Skandamis T, Pappas E, Antonopoulos C, Voros N. En- chancing slam method for mapping and tracking using a low cost laser scanner. In: 2019 10th international conference on information, intelligence, systems and applications (IISA). IEEE; 2019. p. 1–4.
[40] Wang C, Zhang Y, Li X. Pmds-slam: probability mesh enhanced semantic slam in dynamic environments. In: 2020 5th international conference on control, robotics and cybernetics (CRC). IEEE; 2020. p. 40–4.
[41] Bavle H, De La Puente P, How JP, Campoy P. Vps-slam: visual planar semantic slam for aerial robotic systems. IEEE Access 2020;8:60704–18.
[42] Lyu L, Ding Y, Yuan Y, Zhang Y, Liu J, Li J. Doc-slam: robust stereo slam with dynamic object culling. In: 2021 7th international conference on automation, robotics and applications (ICARA). IEEE; 2021. p. 258–62.
[43] Mur-Artal R, Tard′os JD. Orb-slam2: an open-source slam system for monocular, stereo, and rgb-d cameras. IEEE Trans Robot 2017;33(5):1255–62.
[44] Campos C, Elvira R, Rodríguez JJG, Montiel JM, Tardós JD. Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. IEEE Trans Robot 2021;37(6):1874–90.
[45] Geiger A, Lenz P, Stiller C, Urtasun R. Vision meets robotics: the kitti dataset. Int J Robot Res 2013;32(11):1231–7.
[46] Burri M, Nikolic J, Gohl P, Schneider T, Rehder J, Omari S, Achtelik MW, Siegwart R. The euroc micro aerial vehicle datasets. Int J Robot Res 2016;35(10): 1157–63.
[47] Sturm J, Engelhard N, Endres F, Burgard W, Cremers D. A benchmark for the evaluation of rgb-d slam systems. In: 2012 IEEE/RSJ international conference on intelligent robots and systems. IEEE; 2012. p. 573–80.
[48] Kümmerle R, Steder B, Dornhege C, Ruhnke M, Grisetti G, Stachniss C, Kleiner A. On measuring the accuracy of slam algorithms. Aut Robots 2009;27(4):387–407.
[49] Kelly A. Linearized error propagation in odometry. Int J Robot Res 2004;23(2): 179–218.
[50] Horn BK. Closed-form solution of absolute orientation using unit quaternions. Josa a 1987;4(4):629–42.