EURASIP Journal on Wireless
Communications and Networking

REVIEW Open Access

Dynamic visual SLAM and MEC technologies for B5G: a comprehensive review

Jiansheng Peng[1,2*], Yaru Hou[1], Hengming Xu[1] and Taotao Li[1]

*Correspondence:
sheng120410@163.com

[1] College of Automation, Guangxi University of Science and Technology, Guangxi, China
[2] Department of Artificial Intelligence and Manufacturing, Hechi University, Guangxi, China

Abstract

In recent years, dynamic visual SLAM techniques have been widely used in autonomous navigation, augmented reality, and virtual reality. However, the increasing demand for computational resources by SLAM techniques limits its application on resource-constrained mobile devices. MEC technology combined with 5G ultra-dense networks enables complex computational tasks in visual SLAM systems to be offloaded to edge computing servers, thus breaking the resource constraints of terminals and meeting real-time computing requirements. This paper firstly introduces the research results in the field of visual SLAM in detail through three categories: static SLAM, dynamic SLAM, and SLAM techniques combined with deep learning. Secondly, the three major parts of the technology comparison between mobile edge computing and mobile cloud computing, 5G ultra-dense networking technology, and MEC and UDN integration technology are introduced to sort out the basic technologies related to the application of 5G ultra-dense network to offload complex computing tasks from visual SLAM systems to edge computing servers.

Keywords: Dynamic visual SLAM, MEC, B5G, UDN

1 Introduction

Artificial intelligence technology has been transforming more and more industries as it has developed rapidly in recent years, creating disruptive changes across a wide range of sectors. Among them, mobile robots, as an essential part of achieving industrial intelligence, have received widespread attention from academic and industrial sectors. SLAM (Simultaneous Localization and Mapping) technology relies on sensors to estimate the robot's position and model its surroundings to generate a map that the robot can understand and use for navigation.

SLAM with the camera as the only external sensor is called visual SLAM. Vision-based SLAM techniques have gradually developed a relatively mature algorithm system and program architecture. However, classical visual SLAM techniques can only operate normally and without interference in ideal environments, while it is challenging to maintain robustness in highly dynamic scenes. Meanwhile, deep learning-based image semantic segmentation and target detection methods have greatly improved efficiency and accuracy. As a result, numerous researchers remove or track dynamic targets by semantic tagging or target detection preprocessing and thus solve the dynamic SLAM problem.

Dynamic vision SLAM techniques are resource-intensive in terms of memory usage and processing efficiency. Various complex tasks, such as image processing and spatial recognition of the environment, can make the system particularly computationally intensive. It can be even slower for embedded systems due to limited hardware resources. Therefore, for resource-constrained mobile devices, it is challenging to apply dynamic vision SLAM technology based on deep learning efficiently. To improve mobile applications of dynamic visual SLAM techniques, offloading part of the computational tasks to the cloud is generally an effective method. However, in mobile applications, the response time is too long due to the long distance between the endpoint and the cloud, which limits the quality of service. With mobile edge computing, we can provide cloud computing capabilities and IT (Internet Technology) service environments for mobile terminals at the network's edge. The heavy computing tasks are offloaded from the mobile terminal to the high-performance mobile edge computing server, thus breaking through the hardware limitation and resource constraints of the mobile terminal.

Driven by 5G, the demand for mobile communications work is surging. Ultra-dense networks are one of the critical technologies for 5G. UDN (Ultra-Dense Network) addresses the demand for wireless access by deploying dense base stations in communities, providing huge access capacity for end devices. In 5G networks, the application of UDN solves the problem of higher data volume and more efficient transmission due to the surge in mobile work demand. At the same time, MEC (Mobile Edge Computing) technology brings richer computing resources to the mobile side. Thus, there is a complementary relationship between UDN and MEC in the age of 5G. Combining UDN and MEC can provide more mobile devices with mighty computing power to perform resource and data-intensive tasks efficiently and with low latency.

B5G (Beyond 5G) further expands and deepens the scope and field of IoT (Internet of Things) applications based on 5G and combines with artificial intelligence, big data and other technologies to realize the intelligent interconnection of everything. As one of the important innovations of B5G network architecture, MEC can effectively solve the problems of massive data transmission and limited terminal resources faced by B5G scenarios. For the new trend of future B5G/6G network development, the combination of edge computing and artificial intelligence is inevitable. AI-EC (AI over Edge Computing) enables fast visual recognition, speech synthesis, natural language processing and other services by integrating various AI (Artificial Intelligence) algorithms on the network edge devices. By reviewing the advantages and disadvantages of B5G-oriented dynamic visual SLAM and various technologies for offloading computing-intensive tasks to mobile edge servers by applying ultra-dense networks and MEC, this paper proposes current problems in this field and future development trends.

This paper is divided into four main sections. The introduction section in the first section introduces the background and significance of the research. Section 2 introduces the research results in visual SLAM through three categories: classical static visual SLAM, visual SLAM techniques in highly dynamic environments, and SLAM techniques combined with deep learning. Section 3 analyzes the basic technology research related to the application of 5G ultra-dense network to offload complex computational tasks in visual SLAM systems to edge computing servers, starting from three modules: mobile edge computing compared with mobile cloud computing technology, 5G ultra-dense networking technology, and MEC and UDN integration technology. Section 4 is the conclusion and outlook.

2 Visual SLAM technology

2.1 Brief description of the overview

Due to the rapid development of SLAM technology, there are significant differences in how SLAM has been introduced at various stages. The literature [1] addresses the operational environment of visual SLAM and compares the operational effectiveness of different SLAM schemes in indoor and outdoor environments. In particular, the paper [2] emphasizes that the visual SLAM system can crash very quickly if dynamic objects in the environment are not considered. However, do not go further to summarize the handling of SLAM in dynamic environments. Instead, the paper [3] makes a detailed summary and analysis of the tracking of dynamic objects and 3D reconstruction of moving targets. As the theory of visual SLAM tends to mature, people gradually realize that to further improve the robustness of SLAM methods, they need to resort to multi-sensor fusion. Scholars have also provided detailed reviews on SLAM systems using different sensor fusion approaches, such as vision and laser fusion [4] and vision and IMU (Inertial measurement unit) fusion [5]. Since deep learning has been successfully integrated into graphics over the last few years, more and more solutions have tried to apply it to other systems. As a result, numerous scholars have started a comparative analysis of current SLAM systems from the perspective of deep learning [6, 7].

In summary, it is easy to find that the existing review comparisons focus on individual SLAM system development and fail to consider the actual mobile platform resource constraints in combination with the key technologies that break the hardware limitations for a comprehensive comparison. To this end, this paper compares the underlying technologies related to applying 5G ultra-dense networks to offload complex computational tasks from visual SLAM systems to edge computing servers. It discusses their respective advantages and disadvantages, interconnections, and future development.

2.2 Static SLAM

The classical vision SLAM framework [8] is shown in Fig. 1 and usually consists of five modules: vision sensor, vision odometry, nonlinear optimization, loopback detection, and map construction. Among them, vision SLAM can be classified into

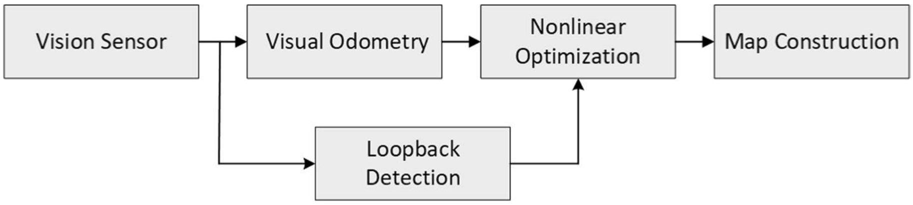

Fig. 1 Classic Visual SLAM Framework. Vision odometry, which estimates the camera motion between adjacent images and constructs a local map; nonlinear optimization, which receives the camera's positional and loopback detection information and optimizes it to obtain a globally consistent trajectory and map; loopback detection, which determines whether the robot has reached a location it has been to before; and map construction, which constructs a map based on the estimated trajectory

monocular, binocular, and RGB-3D SLAM types depending on the vision sensor from which the data are collected. Vision odometry, also known as the front-end, is tasked with acquiring the raw sensor data and preprocessing the data. Operations such as feature extraction and short- and long-term data correlation are performed to convert the geometric information into a mathematical model and send it to the backend. Nonlinear optimization, also known as the backend, is tasked with optimizing the input model for the front-end, minimizing the cumulative error in the camera pose, and optimally adjusting the map information. Loopback detection sends the detection results of camera images to the backend for processing. The accumulated error is eliminated by calculating image similarity and recognizing and comparing the scenes the robot passes through.

SLAM research first appeared in Smith's paper [9] in 1986. In that paper, the authors constructed a map consisting of a series of waypoints while recording the robot's trajectory. Much of today's work can be traced back to Davison's Mono-SLAM [10] (Monocular SLAM), the first real-time monocular vision SLAM system. In Mono-SLAM, sparse feature points are tracked using an extended Kalman filter as the backend. In the same year, Klein proposed PTAM [11] (Parallel Tracking and Mapping), which proposes and implements parallelization of the tracking and map building process and introduces a keyframe mechanism that allows for some mitigation of the computational growth problem [12]. A practical and easy-to-use SLAM system was proposed by Murartal in 2015: ORB-SLAM [13] (Oriented FAST and Rotated BRIEF SLAM). As far as mainstream SLAM feature points go, it represents the pinnacle.The system has several distinct advantages; firstly, it supports monocular, binocular, and RGB-D modes; secondly, the whole system is computed around ORB (Oriented FAST and Rotated BRIEF) features; finally ORB-SLAM innovatively uses three threads to complete SLAM, namely the tracking thread for real-time feature point tracking, the optimization thread for local BA (Bundle Adjustment), and the loopback detection and optimization thread for global bit-pose maps.

Visual SLAM can be classified into feature-based SLAM methods and direct methods depending on the image information. PTAM and ORB-SLAM are typical representatives of feature-based SLAM methods. Most of these feature-based methods rely on the local features of the image and are therefore sensitive to the texture and image quality of the environment [14]. In contrast to the feature-based method, the direct method solves the bit pose by optimizing the photometric error of all pixels between frames. It has the advantages of fast operation and low requirement for the environment texture because it does not need to extract features and compute descriptors. As a representation of the direct method, Engel proposed LSD-SLAM (Large-Scale Direct monocular SLAM) as a method in 2014 [15], which, similar to ORB-SLAM, uses a graph optimization scheme and can be applied to large-scale scenes and build semi-dense maps. However, tests indicated that the accuracy of this algorithm was slightly inferior to that of ORB-SLAM. Using a sparse direct method, Forster [16] proposed a visual odometry SVO (Semi-Direct Monocular Visual Odometry) in the same year. It uses blocks of feature point images to estimate the camera's motion instead of directly matching all pixels. The advantage of SVO over other schemes is that it is speedy and can achieve real time even on low-end computing platforms. In 2016, Engel [17] published another paper on DSO (Direct Sparse Odometry) based on the sparse direct method, which improved the operation speed of visual odometry based on the direct method to a new level.

In summary, the direct method is more robust to some scenes than the feature point-based visual SLAM system. It reduces the time for feature extraction and descriptor matching, speeds up the time for pose estimation, and provides better real-time performance of the algorithm. In addition, the direct method can recover semi-dense or dense maps that are more useful for navigation tasks. However, the direct method is based on the assumption of grayscale invariance, so illumination significantly impacts the system; moreover, the direct method suffers from tracking loss when the camera is moving fast.

2.3 SLAM technique in high dynamic environment

2.3.1 Removing dynamic targets

The classical vision SLAM systems assume a static environment. That is, the changes between adjacent frames are only due to camera movement. When these classical methods are applied to highly dynamic environments such as densely populated areas and driving areas of self-driving vehicles, the dynamic feature points extracted by the vision SLAM system on dynamic targets can directly affect the accuracy of the robot's positional estimation, making the system suffer from errors and drift, which seriously affect the visual odometry and map construction results of the vision SLAM system [18]. To address this problem, many researchers have conducted research and proposed a series of visual SLAM methods to efficiently handle dynamic scenes.

The 3D feature points in space are required to satisfy the projection relationship of the multi-view geometry in the context of a static environment. Taking the feature point method as an example, the 2 frames of images after feature matching are generally optimized simultaneously using the BA method for the 6-degree-of-freedom poses of the camera and the 3D waypoint in space. As shown in Fig. 2, I_1 ,I_2 are two adjacent frames,P_{i1} represents the observation generated by feature point P in frame I_1 , the pixel coordinates of feature point in image I_1 , P_{i2} represents the pixel coordinates of dynamic

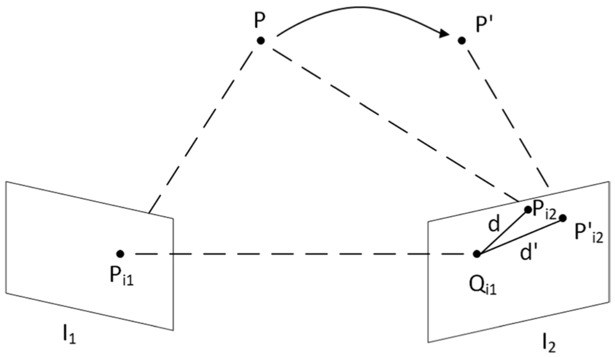

Fig. 2 Spatial point projection and matching. I_1 ,I_2 are two adjacent frames, P is a dynamic feature point, P_{i1} represents the pixel coordinates of dynamic feature point P in frame I_1 , P_{i2} represents the pixel coordinates of P in frame I_2 , Q_{i1} represents the predicted value Q_{i1} corresponding to P_{i1} calculated by BA method, P_{i2} and Q_{i1} will produce a certain deviation d between them, and the deviation needs to be reduced by continuously optimizing the transformation matrix of BA method. After the P point moves to the P' position, the corresponding matching pixel becomes P'_{i2} in the I_2 frame, and the deviation between P'_{i2} and Q_{i1} becomes d'

feature point P in frame I_2 , and Q_{i1} represents the predicted value Q_{i1} corresponding to P_{i1} predicted by BA method calculation, and there will be a certain deviation d between P_{i2} and Q_{i1} , which needs to be continuously optimized by the transformation of BA method matrix to reduce the deviation. However, if the feature point P is a moving target, when it moves to the position P', the corresponding matching pixel in the I_2 frame becomes P'_{i2} , and the deviation between P'_{i2} and Q_{i1} becomes d' , which will make the optimization of the transformation matrix of BA method deviated and cannot get the optimal transformation matrix that minimizes the reprojection error between pixels. Therefore, to improve the accuracy and robustness of visual SLAM in dynamic environments, it is necessary to eliminate the influence of dynamic targets in the environment.

Removing dynamic feature targets using reprojection errors and acquiring static scenes for mapping is effective. Tan [19] propose a novel keyframe online representation and update method to adaptively model dynamic environments so that changes in the appearance and structure of the scene can be efficiently detected and processed. Sun [20] performs pixel classification by calculating the intensity difference of continuous RGB images and quantifying the segmentation of depth images [21] to obtain the static part of the scene.

There are also methods that combine geometric information with other information for the recognition of dynamic features [22]. Ambrus [23] combined dynamic classification and multi-view geometry to dynamic segment objects and proposed methods to adjust static structures and merge new elements over time. Palazzolo [24] used RGB-D sensors to collect data and track them directly against the truncated symbol distance function TSDF [25] (Truncated Signed Distance Function) and finally used the color information encoded in the TSDF to estimate the sensor's bit pose.

In addition, there are ways to reject dynamic targets by processing feature points. Kitt [26]classified feature points by training the classifier in advance to distinguish dynamic points from fixed points, but this method cannot be used to explore unknown environments. Li [27] proposed an odometry method based on the alignment of frame and keyframe depth edge points, while assigning corresponding weight values to the keyframe depth edge points to reduce the influence of dynamic target points on the odometry, and the static weight values indicate the probability that the depth edge points are static or dynamic. The system only uses depth edge point information, so it is unsuitable for scenes lacking textures, and the maps created are sparse.

Some methods use optical flow to segment moving objects. Optical flow estimates pixel motion in 2 consecutive frames of an image in luminance mode [28]. Usually, it corresponds to the motion field in an image and can therefore be used to segment moving objects. Various moving objects are described by scene flow in 3D point clouds and optical flow in 2D images.

In 2017, Jaimez proposed Joint-VO-SF [29], which uses the k-means algorithm to cluster images based on geometric and color information to obtain discrete image blocks, uses the direct method to estimate the camera pose, and statistically determines the reprojection error of the image blocks based on the estimation results to determine whether they are moving objects or not. This algorithm has good performance in computing short-time scene streams. However, the localization accuracy is not high because the initial pose estimation is obtained by optimizing the photometric error of all pixels.

Alcantarilla [30] uses dense scene stream representation to detect dynamic targets. However, this system has distance constraints and is prone to mistakenly detect fixed points as dynamic points in weakly textured scenes. Kerl [31] treats the photometric error between two-pixel points located in different images. However, corresponding to the same spatial point as a variable finds that the distribution of the variable can be better fitted by a t-distribution after statistical analysis, and use this property to assign a weight to each feature point while constructing a weighted least squares loss function to continuously optimize the camera pose through nonlinear optimization to achieve robust localization. The shortcoming of this system is that the dynamic target points are not completely removed, which makes the positional optimization have some errors.

Geometry-based and optical flow-based methods for removing dynamic targets have similar characteristics, which can accurately detect and identify moving objects' position and motion state without knowing the scene information and can work efficiently in real time. In addition, optical flow not only carries the motion information of moving objects but also rich scene 3D structure information, which provides a reasonable basis for dense map building. However, since the optical flow method works based on the assumption of constant luminance, it is susceptible to errors caused by the influence of illumination.

2.3.2 Tracking dynamic targets

The approach of discarding dynamic targets as outliers in the positional estimation and not using them for map construction is applicable in most cases, but simply discarding the information may lead to tracking failure if the dynamic part of the image is a significant occluder or takes up too large a proportion of the image. Therefore, tracking, identifying, and adding dynamic targets to static backgrounds to build maps containing dynamic targets are very necessary to research work.

Kundu [32] addressed the emotional SLAM problem by solving the motion recovery structure and moving object tracking problems. The system outputs a 3D dynamic map containing the structure and trajectories of static and dynamic objects. During the ICRA (International Conference on Robotics and Automation) conference in 2018, Scona [33] presented a robust, dense RGB-D SLAM system that detects moving objects and reconstructs their background structure simultaneously. They simultaneously estimate the camera motion and the current static/dynamic segmentation of RGB-D images. In the next step, a dense RGB-D fusion algorithm is used to derive a 3D model of the static component of the environment based on this segmentation [34]. The camera motion estimation system is more accurate because it uses a 3D model to align frames to the model and segment static and dynamic information. The limitation of this system is that there cannot be a large number of dynamic objects within an initial number of frames so that the accuracy of the initial static scene map can be guaranteed. In the same year, Zhang [35] proposed Pose Fusion for segmenting dynamic bodies and static backgrounds in dynamic environments, which detects human nodes and segments dynamic bodies in 3D point clouds and reconstructs dense maps.

The above methods can solve the problem effectively under certain conditions. However, in the presence of many moving objects in the scene, these methods will be brutal to complete the localization and map building tasks accurately and in real time due to the lack of accurate semantic information as a priori. Deep learning has become increasingly popular for dynamic SLAM as a result, and there has been a comprehensive development of SLAM techniques that combine deep learning to identify and track dynamic targets.

2.4 Visual SLAM technique combined with deep learning

2.4.1 Removing dynamic targets

Several deep learning-based methods have gained efficiency and accuracy in recent years for semantic image segmentation and target detection [36, 37]. In many studies, semantic tagging or target detection preprocessing is used to remove potentially dynamic objects and thus solve the dynamic SLAM problem. Compared with using multi-view geometric constraints or traditional image processing to solve the camera's motion estimation in dynamic environments, it is simpler and more reliable to use semantic information with a priori knowledge to derive the camera's motion model.

Yu Chao from Tsinghua University proposed DS-SLAM [38], which combines a semantic segmentation network with a moving consistency detection method to reduce the impact of dynamic objects. At the same time, the system generates a dense semantic octree map that can be used for higher-order tasks. Experimental results show that the absolute trajectory accuracy in DS-SLAM can be improved by order of magnitude compared to ORB-SLAM2, which is one of the most advanced SLAM systems in highly dynamic environments, but the map building quality performs poorly. Xi [39] improved the DS-SLAM system by utilizing the PSP Net (Pyramid Scene Parsing Net) network with high segmentation accuracy as the segmentation network to further reduce the error of pose estimation in dynamic scenes, thus enabling the SLAM system to perform camera pose estimation in dynamic scenes more accurately. Berta [40] proposed a point-based semantic SLAM system for dynamic environments-Dyna SLAM, which uses a combination of deep learning and multi-view geometry to determine whether a feature point belongs to a dynamic object and combines a domain growth algorithm to key out all dynamic pixels in the build thread to construct a dense static map. This method removes all possible moving objects, so fewer stationary feature points remain, affecting the pose estimation. Xiao [41] proposed a new SLAM framework Dynamic-SLAM, which uses deep learning techniques to build a convolutional neural network-based object detector and combines a priori knowledge to achieve dynamic object detection at the semantic level. A velocity-invariant-based compensation model for missed detection of adjacent frames is proposed.

Deep learning can nicely complement the advantages of geometric and optical flow methods to deal with dynamic SLAM problems. Their combined use is essentially joint processing of environmental semantic, geometric and photometric information. According to Zhong [42], Detect-SLAM is a novel robotic vision system that integrates SLAM with a deep neural network-based object detector to maximize the benefit of both. Cui [43] in 2019 proposed SOF-SLAM, a semantic vision SLAM for dynamic environments. SOF-SLAM exploits the complementary properties of semantic segmentation of motion prior information and information about motion detection concerning polar geometric constraints. Using the semantic segmentation information to assist the computation of the polar geometry, dynamic features can be removed more efficiently, resulting in more accurate results. In 2020, Han and Xi [44] proposed a PSP net-SLAM based on optical flow and PSP net removal of dynamic features for improving ORB-SLAM2. Tracking is performed with the remaining features after removing the features extracted from marked dynamic objects and features with large optical flow values. Ma [45] proposed a joint semantic segmentation network and depth prediction network based on the DSO framework to solve the localization problem in dynamic scenes. The system segmented the input RGB images semantically and predicted them in depth separately, then used the original feature tracking method of DSO for feature matching and detected dynamic points by polar line constraints, and finally removed the dynamic points for camera pose estimation. Hu [46] used the Mask R-CNN (Mask Region-Convolutional Neural Network) network, for instance, segmentation of images, and first used semantic labels to estimate camera pose. Then, we use light projection to determine which objects are visible in this frame, determine whether the objects are moving by calculating the motion residuals of each object, and optimize the camera poses according to the non-moving targets.

When adding a priori semantic information to a SLAM system, it is a very critical issue to ensure real-time performance as much as possible. Redmon [47] proposed the Yolo (You Only Look Once) network architecture in 2016, which classifies and localizes objects in one step, and the real-time performance of the One-stage algorithm detection is very much improved compared to the Two-stage network. However, the detection accuracy is slightly lower compared to the Two-stage network. To filter out unstable features from moving objects, Zhang [48] used YOLOv3 [49] running in a separate thread. A novel sliding window compensation algorithm was proposed by Li [50]to reduce the error of YOLOv3 when detecting dynamic features, thus offering a new approach to dynamic object detection. Cheng [51] used YOLOv3 as an image segmentation network and assigned prior probabilities to each class of objects based on semantic information. A Bayesian model updates the dynamic probabilities of each grid to detect dynamic objects in the scene. These systems detect targets using YOLOv3. The segmented targets are labeled using bounding boxes, so the detection results cause errors by classifying static regions around dynamic targets. A

Table 1 Comparison of visual SLAM algorithms combined with deep learning to remove dynamic targets

Algorithm	Camera type	Method	Application scenarios
DS-SLAM	D	SegNet +Mobile consistency testing	I
[35]	D	PSP Net+Mobile consistency testing	I
DynaSLAM	S,M,D	Mask R-CNN+Multi-view geometry	I/O
PSP Net-SLAM	D	PSP Net+Optical flow method	I
[46]	M	YOLO v3	I
Dynamic-SLAM	M	SSD+Selective tracking algorithm	I/O
Detect-SLAM	D	SSD+Movement probability propagation	I
SOF-SLAM	D	SegNet+For polar geometric constraints	I/O
DSO-Dynamic	M	Mask R-CNN+DSO	O

M represents a monocular camera, S represents a binocular camera, D represents an RGB-D camera, I represents an indoor environment, and O represents an outdoor environment comparison of some visual SLAM algorithms that combine deep learning to remove dynamic targets is shown in Table 1.

By comparing the underlying SLAM frameworks, and technical approaches adopted by these systems, it is found that more scholars choose to develop based on the ORB-SLAM2 system, which benefits from the excellent performance of the ORB-SLAM2 system in a static environment and the easy readability of its open source code. In addition, dynamic SLAM systems based on a priori semantic information require higher hardware requirements. However, some lightweight networks can perform target detection or semantic segmentation tasks in real time with GPU acceleration, such as YOLOv3. However, if the segmentation phase is added to the visual SLAM system, the existing overall solutions integrating semantic information are not very good in real time. Therefore, integrating a semantic SLAM system in a dynamic environment into a mobile device is an effective way to consider offloading the heavy computational tasks from the mobile side to a high-performance mobile edge computing server, which will be described in detail later.

2.4.2 Tracking dynamic targets

The idea of simultaneously estimating camera motion and multiple motion targets originated from the SLAMMOT (Simultaneous localization, Mapping and Moving object tracking) work [52]. The paper develops a mathematical framework that combines SLAM and motion target tracking and demonstrates that it meets the navigation and safety requirements for autonomous driving.

For moving target tracking reconstruction, in 2017, Runz [53] proposed Co-Fusion to different segment objects in dynamic scenes based on camera motion or semantic cues, which can maintain a particular performance in dynamic scenes and track dynamic objects in the reconstructed environment during the SLAM process. The following year, Runz proposed Mask Fusion [54], a real-time, object-aware, semantic, and dynamic RGB-D SLAM system. The method provides object-level scene description by combining Mask R-CNN's target and geometric object edge detection methods to refine the instance edges and reconstruct the recognized objects in 3D while building a background map [55]. Shen [56] used a binocular camera combined with a Faster R-CNN network for target detection and drew a 3D detection frame of the object with geometric information for an autonomous driving application scenario, which incorporated semantic information homography information into a unified optimization framework. At CVPR (Conference on Computer Vision and Pattern Recognition) 2020, Huang [57] introduced Cluster VO, a stereo visual odometer that simultaneously clusters and estimates the motion of self and surrounding rigid objects. The method combines semantic, spatial, and motion information to jointly infer the clustered segmentation of each frame in an online manner.

Implementing visual SLAM in unstructured, dynamic environments requires identifying moving targets and estimating their velocities in real time. Most existing SLAM-based approaches rely on a 3D model database of objects or impose significant motion constraints. To address this problem, Jun [58] proposed VDO-SLAM (Visual Dynamic Object-aware SLAM), a robust target-aware dynamic SLAM system, which is a novel dynamic feature-based SLAM system that enables tracking of dynamic targets based on semantic information of images in the scene without additional target pose or geometric information. At the ICRA 2020 conference, Henein [59] presented a new feature-based, model-free, object-aware dynamic SLAM algorithm that uses semantic segmentation to estimate the motion of rigid bodies in a scene without estimating the object's pose or having any a priori knowledge of the object's 3D model. The algorithm generates maps of dynamic and static structures and extracts the velocity of the rigidly moving objects in the scene.

In summary, if the dynamic targets in the image are essential occluders or take up a large proportion of the image, it becomes imperative to track, identify, and add them to the static background to build a map containing dynamic targets. Combining SLAM techniques with deep learning to identify and track dynamic targets, these methods will accomplish the tracking of dynamic targets more accurately and in real time due to the acquisition of accurate semantic information as a priori, and thus locate and build maps efficiently.

3 Unloading of complex computing tasks in 5G network

3.1 Mobile edge computing and mobile cloud computing

Since mobile devices' computing and storage capacity cannot process and store huge data to meet the immediate need of deep learning tasks in SLAM systems to handle large amounts of computation, for this reason, mobile cloud computing can be used to solve this problem. Mobile cloud computing combines the advantageous features of cloud computing and mobile Internet [60]. All complex computations can be performed in the cloud, allowing mobile devices to be simplified and not requiring very complex configurations [61].

In MCC (Mobile Cloud Computing), end devices can offload low latency and high energy-consuming tasks to cloud servers for computational processing to enhance end devices' computational and storage capabilities [62]. However, although the offloading of device tasks to the cloud server gives a great performance boost to the device, it also brings new problems, as the cloud server is far away from the end device, resulting in a potentially high task execution latency, which cannot meet the demand of the SLAM system for instant map building. At the same time, the mobile terminal needs to consume more energy in the process, leading to an increase in system cost. Moreover, MCC adopts a centralized processing mode [63], and the massive data generated by a large number of terminal devices are transmitted to the cloud server for processing, which not only brings a great burden to the transmission network but also easily causes data privacy leakage and data security problems [64, 65]. Based on this, mobile edge computing was proposed in the industry [66].

Mobile edge computing can effectively solve the problems of time extension, high energy consumption, and data insecurity. In the MEC scenario, the proxy server or base station is placed closer to the mobile terminal. MEC computation offloading technique allows the terminal device to offload computationally intensive tasks to the MEC server for execution and achieve task execution latency reduction with the help of a high computational performance MEC server [67]. Meanwhile, offloading tasks from end devices to edge servers can also effectively reduce device energy consumption. MEC computation offloading techniques can solve the problems caused

Table 2 Comparison of characteristics between MCC and MEC

Method	MEC	MCC
Deployment method	Distributed	Centralized
Distance between server and mobile terminal	Close	Far
Main idea	Marginalization	Centralization
Connection method	Dedicated line connection	Wireless connection
Delay	Low	High
Computing power	Adequate	Limited
Storage capacity	Adequate	Limited
Energy consumption	Low	High
Security	High	Low

Compared with MCC, MEC has the characteristics of low latency, low energy consumption, high security, and has more sufficient storage capacity and computing power

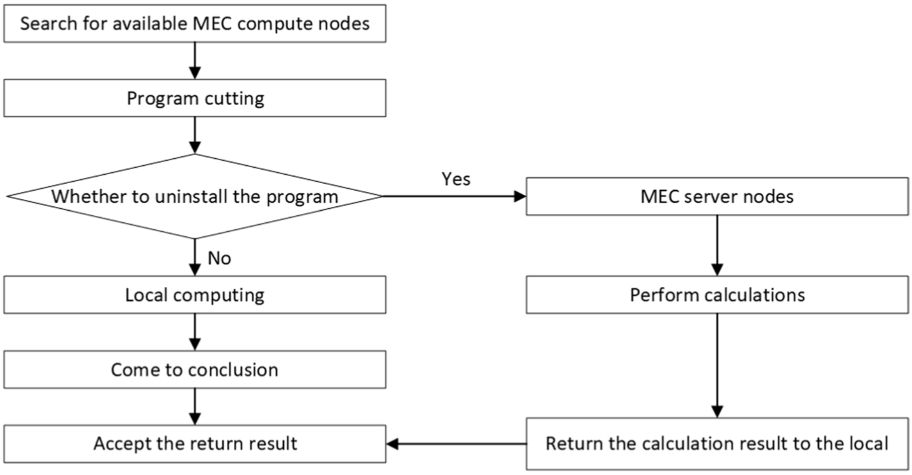

Fig. 3 Process of Calculating Offload. The computational offload process is roughly divided into six phases, including finding available MEC computational nodes, program partitioning, offload decision, program transfer, execution of computation, and return of computational results

by cloud computing while effectively solving the resource-constrained problems of mobile end devices by optimizing the network load and transmission latency of offloading computations of deep learning tasks in visual SLAM to servers. As listed in Table 2, a comparison between mobile edge computing and mobile cloud computing is presented.

Computational offloading technology [68], as one of the key technologies of MEC, refers to offloading tasks running on end devices to edge servers through reasonable offloading decisions and resource allocation strategies, using sufficient computational and storage resources of servers to complete task execution, reducing task completion delay and energy consumption of devices, and improving device performance. Compute offloading techniques have been used in cloud computing, and the only difference in compute offloading techniques from those in mobile edge computing is the different destinations for offloading. The computation offloading process [69] is shown in Fig. 3. It is roughly divided into six phases: finding available MEC computation nodes, program cutting, offloading decision, program transmission, execution of computation, and return of computation results.

3.2 5G ultra-dense networking technology

5G communication technology, the fifth generation mobile communication technology, has the following core features: ultra-high speed, ultra-high capacity, ultra-low latency, ultra-efficient, and full coverage. Compared with 4G technology, 5G communication technology has significantly improved in terms of connection rate, system capacity, number of links, and network latency. As listed in Table 3, 5G and 4G performance indicators are compared.

ITU (International Telecommunication Union) describes three scenarios for 5G applications: Enhanced mobile broadband, Massive IoT, Low Latency, and High Reliability. eMBB (Enhance Mobile Broadband): Human-centric application scenarios, focusing on ultra-high transmission data rates and guaranteed mobility with comprehensive coverage [70]. mMTC (Massive Machine Type Communication): As a result of 5G's robust connectivity, various vertical industries can be rapidly integrated, creating the conditions for the "Internet of Everything" [71]. uRLLC (UltraReliable and Low Latency Communication): The connection latency should reach the 1ms level and support a high-reliability connection in the case of high-speed movement.

The 5G vision includes three approaches to increase system capacity: increasing spectrum bandwidth, improving spectrum utilization, and cell splitting [72]. It is considered that cell splitting is the most effective means of extending the coverage of macro base stations into blind areas and increasing spatial cell reuse by deploying low-power small base stations. There was an emergence of UDN in this context [73]. The most important goal of 5G is to increase the network's capacity. 5G network adopts extensive space multiplexing technology and ultra-dense network collaboration mode, which can maximize the capacity of the network system.

Ultra-dense networking technology increases system capacity and thus frequency diversity by arranging wireless network infrastructure in high density. UDN is the deployment of more Small Cells, and Small Cells can be Femto Cells, Pico Cells, Micro Cells, etc., which usually cover a much smaller area than Macro Cells. These Small Cells in ultra-dense networking technology are small in size, flexible in backhaul, low in transmission power, easy to install, low in construction resistance, and low in cost, and more importantly, significantly reduce the transmission distance between the base station and

Table 3 Comparison of 5G and 4G technical indicators

Name of Indicator	Flow density	Connection number density	Time delay	
4G reference value	0.1Tbps/Km^2	100,000/Km^2	10ms	
5G Reference Value	10Tbps/Km^2	1 million/Km^2	1ms	
Name of Indicator	**Mobility**	**Energy Efficiency**	**User Experience Rate**	**Peak Rate**
4G reference value	350km/h	1X Boost	10Mbps	1Gbps
5G Reference Value	500km/h	100x boost	0.1–1Gbps	20Gbps

Compared with 4G, 5G has many advantages such as higher traffic density, lower latency, higher energy efficiency and faster experience rate the user, with less path loss, and improve the signal quality to a greater extent. UDN is the core technology to meet the demand for low latency, high capacity, and efficient transmission in 5G network systems.

3.3 Technologies for MEC and UDN integration

3.3.1 5G and MEC convergence architecture

The adoption of UDNs for 5G will meet future applications' needs for higher data transfer volumes and lower latency [74]. In 5G networks [75], MEC and UDN are complementary rather than competing technologies as MEC provides end users with computing and storage capabilities [76]. Due to the proximity of server resources, MEC-enabled UDNs have a significant advantage over macro-enabled MEC base stations in terms of offloading computational tasks. Due to short transmission distances, the UDN reduces the energy consumption of end devices and base stations [77]. Combined with 5G UDN is beneficial for future applications of terminal devices that require significant computational resources due to the sound 5G UDN.

Figure 4 shows the MEC deployment in the 5G network proposed by ETSI (European Telecommunications Standards Institute) [78]. As shown in Fig. 4, the left side is the 5G network, which contains a series of control plane network elements such as AMF (Authentication Management Field), SMF (Session Management Function), and PCF (Policy Control Function), as well as user plane network elements UPF (User Plane Function), access network RAN (Radio Access Network), and terminal UE (User Equipment). The right side is the MEC, which contains the MEC platform, management orchestration domain, and multiple service providing APPs (Applications). UPF is the integration point between the 5G network and the MEC. All data must pass through the UPF forwarding before it can flow to the external network. In other words, MEC devices responsible for edge computing must be connected to the UPF, a network element in the 5G core network. 5G core network design is very flexible. In order to reduce data transmission detours, the UPF is generally deployed in a lower position than the control plane network element, which is the UPF sink. For example, China Mobile's core network is divided into eight regions across the country, each managing several provinces, but only control plane elements are deployed in the server rooms of these regions. At the same

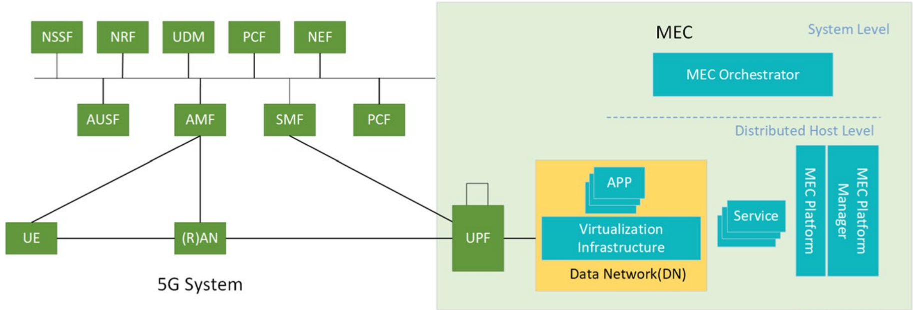

Fig. 4 5G and MEC Converged Architecture. On the left side is the 5G network, which includes a series of control plane elements such as AMF , SMF , PCF , and user plane elements UPF, access network RAN and terminal UE. On the right side is the MEC, which includes the MEC platform, management orchestration domain, and multiple service providing APPs. All data must be forwarded through the UPF before it can flow to the external network

time, UPFs are sunk to provincial centers and even local cities and counties to facilitate local data local digestion. Such an architecture provides the conditions for MEC to be deployed close to the network edge.

3.3.2 MEC unloading decision

The computational offloading process is affected by different factors [79], such as user habits, radio channel communication, backhaul connection quality, mobile device performance, and cloud server availability. Therefore, the key to computational offloading is to specify the appropriate offloading decision. There has been considerable research on offloading decisions and resource allocation in MEC networks. They are broadly classified into three categories: the goal of reducing latency, the goal of reducing energy consumption, and the goal of the trade-off between latency and energy consumption.

To enable latency-sensitive application execution latency reduction, Ning [80] proposed an IHRA (Iterative Heuristic MEC Resource Allocation) scheme for making computational offloading decisions in multi-user situations by considering the abundance of MCC computational resources and the low MEC transmission latency. Sun [81] studied the task offloading problem between vehicles and proposed an algorithm that enables vehicles to understand the offloading delay performance of neighboring vehicles during offloading. Jian [82] improves the bat swarm algorithm to solve the optimization problem of task offloading scheduling in edge computing and proposes an improved chaotic bat swarm cooperative offloading scheme, which greatly reduces the task completion delay and thus meets the demand for real-time task processing. Li [83] proposed a task offloading strategy based on intermediate nodes to balance the load between different edge nodes and shorten the task completion time.

All the above uninstallation decisions achieve the goal of reducing the time delay. However, they fail to consider the energy consumption at one end of the mobile terminal device when performing computational uninstallation. The terminal device may not function properly due to insufficient power. Thus, researchers continue to explore offloading decision schemes to minimize energy consumption. Wen [84] minimizes the total device energy by solving an optimization problem constrained by its parameters. To minimize energy consumption, Cao [85] optimizes the shared resource allocation between users and auxiliary nodes. Zhang [86] models the dependencies between tasks as sequential call graphs and investigates the execution of collaborative mobile device and cloud server tasks under random wireless channels. Mobile device energy consumption is minimized while time deadlines are met. Cuervo [87] proposed a system for offloading fine-grained energy-conscious mobile code to the infrastructure, reducing energy consumption, and modifying programs significantly. Zhou [88] minimizes the device's energy consumption through the joint optimization of resource allocation and task offloading allocation.

The above computational offloading strategies aiming at reducing energy consumption largely alleviate the shortcomings in the battery life of mobile terminal devices. However, in some systems, users prefer to minimize the sum of time and energy consumption to reduce the overall consumption of the system or to trade-off the time and energy consumption so that the total consumption of the system is relatively better and stable. Gu [89] proposed two independent heuristic matching algorithms to solve the problem of minimizing the delay under the energy consumption constraint. Li [90] proposed a system based on Q-Learning and deep Q-Learning to solve the offloading decision problem in multi-user MEC systems. Lian [91] considered the system energy consumption and time delay as the final optimization objective and designed an offloading scheme based on quantum evolution theory. Dai [92] formulated the task offloading problem in a multi-user mobile edge computing scenario as a convex optimization problem with weighted and minimized delay and energy consumption under resource-limited conditions and proposed a computational offloading mechanism based on the multiplicative method to minimize the energy consumption and task execution delay of mobile terminals. Teng [93] optimized a multi-user mobile edge computational offload system by constructing a Markov decision problem with the long-term average overhead of delay and power consumption as the optimization objective and solving it using convex optimization theory. Zhang [94] proposed a game-theoretic-based offloading scheme for the multi-user task offloading problem, which approximates the optimal solution to the theoretical optimal policy and dramatically reduces the system overhead.

By analyzing and comparing the above three types of computation offloading decisions, it can be seen that the most significant advantage of offloading computation to the edge server is the reduction of computation latency compared to computing on the mobile side. The discussion on energy consumption can be further developed under the condition of guaranteeing computation latency. Although not minimizing latency or energy consumption, the strategy of maximizing revenue can be closer to the specific application requirements. In the future, the application scenario of mobile edge computing is continuously being expanded, and the setting of computational offloading decisions largely depends on the characteristics of the task to be processed. The variable factors involved in the computation using mobile edge computing technology will become more, and only by guaranteeing the stability of the elements involved in the computation process can the application of mobile edge computing technology be better supported.

3.3.3 UND and MEC integration

Since mobile devices have proliferated and computation-intensive and latency-sensitive tasks have emerged, resources have been in high demand. This problem can be effectively solved by combining mobile edge computing with ultra-dense networks, and thus, ultra-dense edge computing emerges. UDN's base stations can be integrated with MEC by deploying MEC servers [95]. Integrating these systems allows real-time computing requirements to be met with low latency data transmission.

In the MEC, offloading tasks were a hot topic as mobile devices make offloading decisions to reduce execution delays, reduce energy consumption, and improve offloading efficiency. The problems related to resource allocation in UDNs have also received great attention. Using deep learning techniques, Zhou [96] intelligently avoided or alleviated congestion at UDN base stations. A similar method based on non-explicit ranking genetic algorithms has been proposed by Xu [97], which improves energy efficiency and spectral efficiency by allocating transmission power and resource blocks. Liu [98] proposed an optimal resource scheduling strategy with two-step joint clustering. Zhang [99] designed an augmented learning-based downlink power control algorithm for managing interference in small cell networks with dense cell networks. These studies examine the problem of resource allocation in UDNs in depth. The solution to the spectrum and power allocation problem in UDN will help reduce interference and improve performance. If a large number of devices are engaged in data-intensive or compute-intensive tasks simultaneously, it will be challenging to fulfill the demands of the devices efficiently in real time.

To address the above issues, several studies have emerged in recent years to address the problem of task offloading and resource allocation in MEC-enabled UDNs. Guo [100] proposed an optimal enumeration offloading strategy and a two-level game offloading strategy to optimize the weighted sum of delay and energy consumption. The task offloading algorithm proposed by Yang [101] is based on the game idea of UDN and is primarily designed to conserve energy under the delay constraint. An energy-collection MEC server was designed, as well as centralized and decentralized algorithms for offloading tasks, by Chen [102]. Using a long-term evolving UDN model, Bottai [103] developed an energy consumption correlation algorithm to study user terminal energy consumption. Sun [104] considered the cost of switching between tasks in a multitasking scenario with sequential constraints in their energy-aware mobility management algorithm. Guo [105]proposed a heuristic greedy offloading strategy for offloading MEC. The limitation of this method is that it does not consider the delay requirements of individual tasks but only the total delay of the whole system. Chen [106] proposed a differential evolutionary algorithm for task assignment and channel resource allocation, which reduces the task assignment problem to integer nonlinear programming. This approach is efficient, has good convergence performance, and can significantly reduce energy consumption. Huang [107] studied the IoT offloading problem in MEC-based UDNs to minimize energy consumption and delay while optimizing offloading decisions, transmission power, and radio resource allocation. In the UDEC (Ultra-Dense Edge Computing) environment, the problem becomes very complex because UDEC has a large number of edge devices and servers with heterogeneous computational resources and communication links. Therefore, exploring how to efficiently and fully utilize the system resources to improve the UDEC performance is very important.

In summary, channel resource allocation in UDN and offloading decisions in MEC directly affect user experience and network quality. The main objective of MEC's offloading decisions is to reduce task execution delay, reduce energy consumption, and improve offloading efficiency. The UDN's resource allocation problem usually involves allocating spectrum and power to reduce interference so that system performance can be improved. These traditional approaches will have difficulty handling the demands of many devices simultaneously if they perform computation-intensive or data-intensive tasks. A further challenge in UDEC is that system resources are often diverse and variable over time, complicating scheduling. To achieve computational offloading, applications must be partitioned in real time, and low overhead resources must be allocated. Therefore, it is well worth exploring the research direction of how to design dynamic and real-time scheduling schemes to improve the quality of service of edge devices.

4 Summary and outlook

There are many applications of visual SLAM, including autonomous navigation, augmented reality, virtual reality, and more. As most traditional visual SLAM techniques assume a static environment, dynamic targets in the environment will significantly affect the accuracy of the positional estimation when the SLAM system extracts dynamic feature points. For this reason, a series of visual SLAM methods for removing or tracking dynamic targets, including geometry, optical flow, and deep learning, have been proposed.

Among them, geometry-based and optical flow-based methods to remove dynamic targets can accurately detect and identify the position and motion state of moving objects without knowing the scene information. However, they are more affected by environmental factors such as illumination and weak texture of objects. The visual SLAM technique, which combines deep learning to identify and remove dynamic targets, has a significant improvement in the accuracy and robustness of localization and map building compared to the way of using multi-view geometric constraints or traditional image processing to solve the camera's motion estimation in dynamic environments by using semantic information with a priori knowledge to assist in deriving the camera's motion model. However, if the dynamic target of the image is a crucial occlusion or takes up too large a proportion of the image, discarding the dynamic target as an outlier in the positional estimation and not using it for map construction will likely lead to tracking failure. Therefore, combining deep learning to track and identify dynamic targets and adding them to static backgrounds to build maps containing dynamic targets is a future mainstream research direction for dynamic SLAM.

The technology of combining deep learning for visual SLAM removal, tracking dynamic targets, and completing dynamic scene-building maps is gradually matured. However, the extremely high real-time requirements of dynamic visual SLAM and the heavy computational tasks caused by various complex image processing and environment recognition work limit its application on resource-constrained mobile devices. The mobile edge computing offload technology allows end devices to offload computationally intensive tasks to mobile edge servers for execution, achieving reduced task execution latency and device energy consumption with the help of high computational performance mobile servers. Mobile edge computing provides additional computing and memory resources for mobile devices, breaking through the hardware limitations and resource constraints of terminals.

With the development of 5G, the combination of mobile edge computing and 5G ultra-dense network deployment has given birth to UDEC technology to accommodate greater device access and further reduce data transmission latency for mobile end devices. With high bandwidth and low latency, UDEC technology enables complex computing tasks in visual SLAM systems to be offloaded to edge computing servers with extremely low latency to meet real-time computing requirements. However, the presence of a large number of edge devices and edge servers in the UDEC environment, as well as the diverse and time-varying nature of system resources, complicates task offload decision-making. In addition, due to the highly dynamic mobility of mobile devices and the random nature of work requests, MEC servers can experience load imbalance. When a high number of tasks are offloaded to a single MEC server, it will cause server overload, and when computers run at a low load, they will waste enormous computing power. Therefore, when facing task offloading requests from multiple mobile devices, it is very important to deal with the problem of prioritizing tasks, designing an intelligent learning scheduling scheme to allocate resources, and partitioning program tasks so that the resources of edge computing servers are fully utilized. With low latency and low energy consumption as the research objectives, it is very important to study in depth how to efficiently and fully utilize system resources, determine dynamic, real-time offload decisions and channel resource allocation schemes, and improve UDEC performance in the future. In addition, it is easy to find from the schemes investigated in the paper that most proposed algorithms are not applied in real life. Due to the difference between theoretical research and practice, many other factors affect the research results. In the subsequent research work, more complex real-life scenarios should be considered to actively seek to implement the scheme technology on the ground to create economic and social values.

Abbreviations

SLAM	Simultaneous localization and mapping
IT	Internet technology
UDN	Ultra-dense network
MEC	Mobile edge computing
B5G	Beyond 5G
IoT	Internet of Things
AI	Artificial Intelligence
AI-EC	AI over Edge Computing
IMU	Inertial measurement unit
Mono-SLAM	Monocular SLAM
PTAM	Parallel tracking and mapping
ORB-SLAM	Oriented FAST and Rotated BRIEF SLAM
BA	Bundle adjustment
LSD-SLAM	Large-scale direct monocular SLAM
SVO	Semi-direct monocular visual odometry
DSO	Direct sparse odometry
TSDF	Truncated signed distance function
ICRA	International conference on robotics and automation
PSP Net	Pyramid scene parsing net
Mask R-CNN	Mask Region-convolutional neural network
YOLO	You only look once
SLAMMOT	Simultaneous localization, mapping and moving object tracking
CVPR	Conference on computer vision and pattern recognition
VDO-SLAM	Visual dynamic object-aware SLAM
MCC:	Mobile cloud computing
ITU	International Telecommunication Union
eMBB	Enhance Mobile Broadband
mMTC	Massive Machine Type Communication
uRLLC	UltraReliable and low latency communication
ETSI	European Telecommunications Standards Institute
AMF	Authentication Management Field
SMF	Session Management Function
PCF	Policy Control Function
UPF	User Plane Function
RAN	Radio access network
UE	User equipment
APP	Application
IHRA	Iterative Heuristic MEC Resource Allocation
UDEC	Ultra-dense edge computing

Acknowledgements

The authors are highly thankful to the National Natural Science Foundation of China, to the Innovation Fund of Chinese Universities Industry-University-Research, to the Research Project for Young and Middle-aged Teachers in Guangxi Universities, to the Natural Science Foundation of Guangxi Province, and to the Special research project of Hechi University. This research was financially supported by the project of outstanding thousand young teachers' training in higher education institutions of Guangxi, Guangxi Colleges and Universities Key Laboratory of AI and Information Processing (Hechi University), Education Department of Guangxi Zhuang Autonomous Region.

Author contributions

Jiansheng Peng wrote the manuscript; Yaru Hou contributed significantly to analysis and manuscript preparation; Hengming Xu contributed to the conception of the study; Taotao Li helped perform the analysis with constructive discussions. All authors read and approved the final manuscript.

Funding

The authors are highly thankful to the National Natural Science Foundation of China(NO.62063006), to the Innovation Fund of Chinese Universities Industry-University-Research (ID:2021RYC06005), to the Research Project for Young and Middle-aged Teachers in Guangxi Universities (ID: 2020KY15013), to the Natural Science Foundation of Guangxi Province (NO.2018GXNSFAA281164), and to the Special research project of Hechi University (ID:2021GCC028). This research was financially supported by the project of outstanding thousand young teachers' training in higher education institutions of Guangxi, Guangxi Colleges and Universities Key Laboratory of AI and Information Processing (Hechi University), Education Department of Guangxi Zhuang Autonomous Region.

Availability of data and materials

Not applicable.

Declarations

Ethics approval and consent to participate

Not applicable.

Consent for publication

All authors agree to publish.

Competing interests

The authors declared that they have no conflicts of interest to this work.

Received: 23 May 2022 Accepted: 21 September 2022
Published online: 01 October 2022

References

1. J.-X. He, Z.-M. Li, Survey of vision-based approach to simultaneous localization and mapping. Jisuanji Yingyong Yanjiu **27**(8), 2839–2844 (2010)
2. J. Fuentes-Pacheco, J. Ruiz-Ascencio, J.M. Rendón-Mancha, Visual simultaneous localization and mapping: a survey. Artif. Intell. Rev. **43**(1), 55–81 (2015)
3. B. Alsadik, S. Karam, The simultaneous localization and mapping (slam)-an overview. J Appl Sci Technol Trends. https://doi.org/10.38094/jastt204117 (2021)
4. C. Debeunne, D. Vivet, A review of visual-lidar fusion based simultaneous localization and mapping. Sensors **20**(7), 2068 (2020)
5. C. Chen, H. Zhu, M. Li, S. You, A review of visual-inertial simultaneous localization and mapping from filtering-based and optimization-based perspectives. Robotics **7**(3), 45 (2018)
6. L. Xia, J. Cui, R. Shen, X. Xu, Y. Gao, X. Li, A survey of image semantics-based visual simultaneous localization and mapping: application-oriented solutions to autonomous navigation of mobile robots. Int. J. Adv. Rob. Syst. **17**(3), 1729881420919185 (2020)
7. C. Wei, A. Li et al. Overview of visual slam for mobile robots. Int. J. Front. Eng. Technol. **3**(7) (2021)
8. G. Xiang, A. Engineering, *Visual SLAM XIV: From Theory to Practice* (Electronic Industry Press, 2017)
9. R.C. Smith, P. Cheeseman, On the representation and estimation of spatial uncertainty. The Int. J. Robot. Res. **5**(4), 56–68 (1986)
10. A.J. Davison, I.D. Reid, N.D. Molton, O. Stasse, Monoslam: real-time single camera slam. IEEE Trans. Pattern Anal. Mach. Intell. **29**(6), 1052–1067 (2007)
11. G. Klein, D. Murray, Parallel tracking and mapping for small ar workspaces. In *2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, pp. 225–234 (2007). IEEE
12. M. Quan, S. Piao, G. Li, An overview of visual SLAM. CAAI Trans Intell Syst. **11**(6), 768–776 (2016)
13. R. Mur-Artal, J.M.M. Montiel, J.D. Tardos, Orb-slam: a versatile and accurate monocular slam system. IEEE Trans. Rob. **31**(5), 1147–1163 (2015)
14. H. Liu, G. Zhang, H. Bao, A survey of monocular simultaneous localization and mapping. J. Comput.-Aided Des. Comput. Graph. **28**(6), 855–868 (2016)
15. J. Engel, T. Schöps, D. Cremers, Lsd-slam: large-scale direct monocular slam. In *European Conference on Computer Vision*, pp. 834–849 (2014). Springer
16. C. Forster, M. Pizzoli, D. Scaramuzza, Svo: fast semi-direct monocular visual odometry. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 15–22 (2014). IEEE
17. J. Engel, V. Koltun, D. Cremers, Direct sparse odometry. IEEE Trans. Pattern Anal. Mach. Intell. **40**(3), 611–625 (2017)
18. M.R.U. Saputra, A. Markham, N. Trigoni, Visual slam and structure from motion in dynamic environments: a survey. ACM Comput. Surv. (CSUR) **51**(2), 1–36 (2018)

19. W. Tan, H. Liu, Z. Dong, G. Zhang, H. Bao, Robust monocular slam in dynamic environments. In *2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 209–218 (2013). IEEE
20. Y. Sun, L. Ming, Q.H. Meng, Improving rgb-d slam in dynamic environments: a motion removal approach. Robot. Auton. Syst. **89**, 110–122 (2016)
21. S. Wan, Y. Xia, L. Qi, Y. H. Yang, M. Atiquzzaman, *Automated colorization of a grayscale image with seed points propagation*. IEEE Transactions on Multimedia pp. (99), 1–1 (2020)
22. J. Peng, H. Ye, Q. He, Y. Qin, Z. Wan, J. Lu, Design of smart home service robot based on ros. Mobile Inform. Syst. **2021**, (2021)
23. R. Ambrus, J. Folkesson, P. Jensfelt, Unsupervised object segmentation through change detection in a long term autonomy scenario. *2016 IEEE-RAS 16th international conference on Humanoid robots (HUMANOIDS)*, 1181–1187 (2016)
24. E. Palazzolo, J. Behley, P. Lottes, P. Gigu &egrave;re, C. Stachniss, Refusion: 3d reconstruction in dynamic environments for rgb-d cameras exploiting residuals. *2019 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, 7855–7862 (2019)
25. B. Curless, M. Levoy, *A volumetric method for building complex models from range images*. SIGGRAPH, pp. 303–312 (1996)
26. B. Kitt, F. Moosmann, C. Stiller, Moving on to dynamic environments: visual odometry using feature classification. *IEEE/RSJ 2010 international conference on intelligent robots and systems (IROS 2010)*, 5551–5556 (2010)
27. S. Li, D.A. Lee, Rgb-d slam in dynamic environments using static point weighting. IEEE Robot. Autom. Lett. **4**, 2263–2270 (2017)
28. K.P.B. Horn, G.B. Schunck, Determining optical flow. Artif. Intell. **17**, 185–203 (1981)
29. M. Jaimez, , C. Kerl, J. Gonzalez-Jimenez, D. Cremers, Fast odometry and scene flow from rgb-d cameras based on geometric clustering. ICRA, 3992–3999 (2017)
30. F.P. Alcantarilla, J.Y.J. Torres, J. Almazan, M.L. Bergasa, On combining visual slam and dense scene flow to increase the robustness of localization and mapping in dynamic environments. *2012 IEEE international conference on robotics and automation (ICRA)*, 1290–1297 (2012)
31. C. Kerl, , J. Sturm, D. Cremers, Robust odometry estimation for rgb-d cameras. ICRA, 3748–3754 (2013)
32. A. Kundu, M.K. Krishna, V.C. Jawahar, Realtime multibody visual slam with a smoothly moving monocular camera. ICCV, 2080–2087 (2011)
33. R. Scona, M. Jaimez, R.Y. Petillot, M. Fallon, D. Cremers, Staticfusion: background reconstruction for dense rgb-d slam in dynamic environments. *2018 IEEE international conference on robotics and automation (ICRA)*, 3849–3856 (2018)
34. Z. Gao, H. Xue, S. Wan, Multiple discrimination and pairwise cnn for view-based 3d object retrieval. Neural Netw. **17**, 290–302 (2020)
35. T. Zhang, Y. Nakamura, PoseFusion: dense RGB-D SLAM in dynamic human environments. *Proceedings of the 2018 international symposium on experimental robotics*, (2020)
36. S. Ding, S. Qu, Y. Xi, S. Wan, Stimulus-driven and concept-driven analysis for image caption generation. Neurocomputing **398**, 520–530 (2020)
37. Y. Zhao, H. Li, S. Wan, A. Sekuboyina, X. Hu, G. Tetteh, M. Piraud, B. Menze, Knowledge-aided convolutional neural network for small organ segmentation. IEEE J. Biomed. Health Inform. **23**, 1363–1373 (2019)
38. C. Yu, Z. Liu, X. Liu, F. Xie, Y. Yang, Q. Wei, F. Qiao, Ds-slam: a semantic visual slam towards dynamic environments. *2018 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, 1168–1174 (2018)
39. X.I. Zhihong, S. Han, H. Wang, Simultaneous localization and semantic mapping of indoor dynamic scene based on semantic segmentation. J. Comput. Appl. **39**(10), 2847 (2019)
40. B. Bescós, M.J. Fácil, J. Civera, J. Neira, Dynaslam: tracking, mapping, and inpainting in dynamic scenes. IEEE Robot. Autom. Lett. **4**, 4076–4083 (2018)
41. L. Xiao, J. Wang, X. Qiu, Z. Rong, X. Zou, Dynamic-slam: semantic monocular visual localization and mapping based on deep learning in dynamic environment. Robot. Auton. Syst. **117**, 1–16 (2019)
42. Z. Fangwei, W .Sheng, Z. Ziqi, Z. Chen, W. Yizhou, Detect-slam: making object detection and slam mutually beneficial. 2018 IEEE winter conference on applications of computer vision (WACV 2018), pp. 1001–1010 (2018)
43. L. Cui, C. Ma, Sof-slam: a semantic visual slam for dynamic environments. IEEE Access **7**, 166528–01665390 (2019)
44. S. Han, Z. Xi, Dynamic scene semantics slam based on semantic segmentation. IEEE Access **8**, 43563–0435700 (2020)
45. P. Ma, Y. Bai, J. Zhu, C. Wang, C. Peng, Dsod: Dso in dynamic environments. IEEE Access **7**, 178300–178309 (2019)
46. W. Hu, *Visual slam research on mobile robots in indoor dynamic environments*. PhD thesis, Huazhong University of Science and Technology (2019)
47. J. Redmon, K.S. Divvala, B.R.. Girshick, A. farhadi, You only look once: unified, real-time object detection. *2016 IEEE conference on computer vision and pattern recognition (CVPR)*, 779–788 (2016)
48. L. Zhang, L. Wei, P. Shen, W. Wei, G. Zhu, J. Song, Semantic slam based on object detection and improved octomap. IEEE Access **6**, 75545–075559 (2018)
49. J. Redmon, A. Farhadi, Yolov3: an incremental improvement. arXiv: Computer Vision and Pattern Recognition (2018)
50. P. Li, G. Zhang, J. Zhou, R. Yao, X. Zhang, Study on slam algorithm based on object detection in dynamic scene. *2019 international conference on advanced mechatronic systems (ICAMECHS)*, 363–367 (2019)
51. J. Cheng, H. Zhang, Q.-H.M. Meng, Improving visual localization accuracy in dynamic environments based on dynamic region removal. IEEE Trans. Autom. Sci. Eng. **17**, 1585–1596 (2020)
52. C.-C. Wang, C. Thorpe, Simultaneous localization, mapping and moving object tracking. The Int. J. Robot. Res. **26**, 889–916 (2007)
53. M. Rünz, L. Agapito, *Co-fusion: real-time segmentation, tracking and fusion of multiple objects*. ICRA, 4471–4478 (2017)

54. M. Rünz, L. Agapito, *Maskfusion: Real-time recognition, tracking and reconstruction of multiple moving objects*. ISMAR (2018)
55. Z. Gao, Y. Li, S. Wan, Exploring deep learning for view-based 3d model retrieval. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 1–21 (2020)
56. P. Li, T. Qin, S. Shen, Stereo vision-based semantic 3d object and ego-motion tracking for autonomous driving. Computer Vision - ECCV 2018(PT II), 664–679 (2018)
57. J. Huang, S. Yang, T.-J. Mu, S.-M. Hu, Clustervo: clustering moving instances and estimating visual odometry for self and surroundings. *2020 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, 2165–2174 (2020)
58. Z. Jun, H. Mina, M. Robert, I. Viorela, Vdo-slam: a visual dynamic object-aware slam system. arXiv preprint arXiv:2005.11052 (2020)
59. H. Mina, Z. Jun, M. Robert, I. Viorela, Dynamic slam: the need for speed. ICRA, 2123–2129 (2020)
60. S.D. Linthicum, Connecting fog and cloud computing. IEEE Cloud Comput. **4**, 18–20 (2017)
61. H.T. Dinh, C. Lee, D. Niyato, W. Ping, A survey of mobile cloud computing: architecture, applications, and approaches. Wirel. Commun. Mob. Comput. **13**(18), 1587–1611 (2013)
62. C. Chen, B. Liu, S. Wan, P. Qiao, Q. Pei, An edge traffic flow detection scheme based on deep learning in an intelligent transportation system. IEEE Trans. Intell. Transp. Syst. **22**, 1840–1852 (2021)
63. B. Panchali, Edge computing- background and overview. In *International Conference on Smart Systems and Inventive Technology* (2018)
64. J. Wang, L. Wu, R.K.-K. Choo, D. He, Blockchain-based anonymous authentication with key management for smart grid edge computing infrastructure. IEEE Trans. Ind. Inform. **16**, 1984–1992 (2020)
65. Z. Wang, X. Pang, Y. Chen, H. Shao, Q. Wang, L. Wu, H. Chen, H. Qi, Privacy-preserving crowd-sourced statistical data publishing with an untrusted server. IEEE Trans. Mob. Comput. **18**(6), 1356–1367 (2018)
66. Y.C. Hu, M. Patel, D. Sabella, Mobile edge computing a key technology towards 5g. ETSI White Paper **11**(11), 1–16 (2015)
67. W. Shi, J. Cao, Q. Zhang, Y. Li, L. Xu, Edge computing: vision and challenges. IEEE Internet Things J. **3**, 637–646 (2016)
68. H. Flores, P. Hui, S. Tarkoma, Y. Li, N.S. Srirama, R. Buyya, Mobile code offloading: from concept to practice and beyond. IEEE Commun. Mag. **53**, 80–88 (2015)
69. Y. Zhang, H. Liu, L. Jiao, X. Fu, To offload or not to offload: an efficient code partition algorithm for mobile cloud computing. CloudNet, 80–86 (2012)
70. L. Wu, C. Quan, C. Li, Q. Wang, B. Zheng, X. Luo, A context-aware user-item representation learning for item recommendation. ACM Trans Inform Syst **37**(2), 1–29 (2019)
71. L. Wu, J. Wang, R.K.-K. Choo, D. He, Secure key agreement and key protection for mobile device user authentication. IEEE Trans. Inform. Foren. Sec. **14**, 319–330 (2019)
72. J. Sachs, G. Wikstrom, T. Dudda, R. Baldemair, K. Kittichokechai, 5g radio network design for ultra-reliable low-latency communication. IEEE Netw. **32**(2), 24–31 (2018)
73. X. Ge, S. Tu, G. Mao, C.-X. Wang, T. Han, 5g ultra-dense cellular networks. IEEE Wirel. Commun. **23**(1), 72–79 (2016)
74. I.M. Kamel, W. Hamouda, M.A. Youssef, Ultra-dense networks: a survey. IEEE Commun. Surv. Tutor. **18**, 2522–2545 (2016)
75. H. Guo, J. Liu, J. Zhang, Computation offloading for multi-access mobile edge computing in ultra-dense networks. IEEE Commun. Mag. **56**, 14–19 (2018)
76. P. Ranaweera, A.D. Jurcut, M. Liyanage, Realizing multi-access edge computing feasibility: security perspective. In *IEEE conference on standards for communications and networking* (CSCN 2019) (2019)
77. Y. Siriwardhana, P. Porambage, M. Liyanage, M. Ylinattila, A survey on mobile augmented reality with 5g mobile edge computing: architectures, applications and technical aspects. IEEE Commun. Surv. Tutor. **23**, 1160–1192 (2021)
78. S. Kekki, W. Featherstone, Y. Fang, P. Kuure, A. Li, A. Ranjan, D. Purkayastha, F. Jiangping, D.G.V. Frydman, Mec in 5g networks. ETSI White Paper **28**, 1–28 (2018)
79. Y. Mao, C. You, J. Zhang, K. Huang, K.B. Letaief, A survey on mobile edge computing: the communication perspective. IEEE Commun. Surv. Tutor. **19**(4), 2322–2358 (2017)
80. Z. Ning, P. Dong, X. Kong, F. Xia, A cooperative partial computation offloading scheme for mobile edge computing enabled internet of things. IEEE Internet Things J. **6**, 4804–4814 (2019)
81. Y. Sun, X. Guo, J. Song, S. Zhou, Z. Jiang, X. Liu, Z. Niu, Adaptive learning-based task offloading for vehicular edge computing systems. IEEE Trans. Veh. Technol. **68**, 3061–3074 (2019)
82. C.-F. Jian, J.-W. Chen, M.-Y. Zhang, Improved chaotic bat swarm cooperative scheduling algorithm for edge computing. J. Chin. Comput. Syst. **7**, 2424–2430 (2019)
83. G. Li, Y. Yao, J. Wu, X. Liu, X. Sheng, Q. Lin, A new load balancing strategy by task allocation in edge computing based on intermediary nodes. EURASIP J. Wirel. Commun. Netw. **1**, 1–10 (2020)
84. Y. Wen, W. Zhang, H. Luo, Energy-optimal mobile application execution: taming resource-poor mobile devices with cloud clones. In *2012 Proceedings IEEE Infocom*, pp. 2716–2720 (2012). IEEE
85. X. Cao, F. Wang, J. Xu, R. Zhang, S. Cui, Joint computation and communication cooperation for energy-efficient mobile edge computing. IEEE Internet Things J. **6**(3), 4188–4200 (2018)
86. W. Zhang, Y. Wen, D.O. Wu, Collaborative task execution in mobile cloud computing under a stochastic wireless channel. IEEE Trans. Wirel. Commun. **14**(1), 81–93 (2014)
87. E. Cuervo, A. Balasubramanian, D.-k. Cho, A. Wolman, S. Saroiu, R. Chandra, P. Bahl, Maui: making smartphones last longer with code offload. In *Proceedings of the 8th international conference on mobile systems, applications, and services*, pp. 49–62 (2010)
88. J. Zhou, X. Zhang, W. Wang, Y. Zhang, Energy-efficient collaborative task offloading in d2d-assisted mobile edge computing networks. In *2019 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–6 (2019). IEEE

89. B. Gu, Z. Zhou, Task offloading in vehicular mobile edge computing: a matching-theoretic framework. IEEE Veh. Technol. Mag. **14**, 100–106 (2019)
90. J. Li, H. Gao, T. Lv, Y. Lu, Deep reinforcement learning based computation offloading and resource allocation for mec. *2018 IEEE wireless communications and networking conference (WCNC)*, 1–6 (2018)
91. X. Lian, R. Xie, T. Huang, Security-based computation offloading scheme in edge computing network. ZTE Technol. J. **182**, 41–4656 (2019)
92. D. Meiling, L. Zhoubin, G. Shaoyong, S. Sujie, Q. Xuesong, A computation offloading and resource allocation mechanism based on minimizing devices energy consumption and system delay. J. Electron. Inform. Technol. **41**, 2684–2690 (2019)
93. T. Ying-lei, L. Wei, O. Wei-ping, L. Kun, S. Mei, Queue-aware joint optimization of offloading and transmission in wireless mobile edge computing systems. J. Beijing Univ. Posts Telecommun. **42**, 14–20 (2019)
94. Z. Genshan, L. Xuning, Tasks split and offloding scheduling decision in mobile edge computing with limited resources. Comput. Appl. Softw. **36**(10), 268–273278 (2019)
95. Z. Zhao, G. Min, W. Gao, Y. Wu, H. Duan, Q. Ni, Deploying edge computing nodes for large-scale iot: a diversity aware approach. IEEE Internet Things J. **5**, 3606–3614 (2018)
96. Y. Zhou, M.Z. Fadlullah, B. Mao, N. Kato, A deep-learning-based radio resource assignment technique for 5g ultra dense networks. IEEE Netw. **32**, 28–34 (2018)
97. S. Xu, R. Li, Q. Yang, Improved genetic algorithm based intelligent resource allocation in 5g ultra dense networks. *2018 IEEE wireless communications and networking conference (WCNC)*, 1–6 (2018)
98. L. Liu, Y. Zhou, V. Garcia, L. Tian, J. Shi, Load aware joint comp clustering and inter-cell resource scheduling in heterogeneous ultra dense cellular networks. IEEE Trans. Veh. Technol. **67**, 2741–2755 (2018)
99. H. Zhang, M. Min, L. Xiao, S. Liu, P. Cheng, M. Peng, Reinforcement learning-based interference control for ultra-dense small cells. *IEEE Global Communications Conference*, 1–6 (2018)
100. H. Guo, J. Liu, J. Zhang, W. Sun, N. Kato, Mobile-edge computation offloading for ultradense iot networks. IEEE Internet Things J. **5**(6), 4977–4988 (2018)
101. L. Yang, H. Zhang, X. Li, H. Ji, V.C. Leung, A distributed computation offloading strategy in small-cell networks integrated with mobile edge computing. IEEE/ACM Trans. Netw. **26**(6), 2762–2773 (2018)
102. W. Chen, D. Wang, K. Li, Multi-user multi-task computation offloading in green mobile edge cloud computing. IEEE Trans. Serv. Comput. **12**(5), 726–738 (2018)
103. C. Bottai, C. Cicconetti, A. Morelli, M. Rosellini, C. Vitale, Energy-efficient user association in extremely dense small cell networks. In *2014 European Conference on Networks and Communications (EuCNC)*, pp. 1–5 (2014). IEEE
104. Y. Sun, S. Zhou, J. Xu, Emm: Energy-aware mobility management for mobile edge computing in ultra dense networks. IEEE J. Sel. Areas Commun. **35**(11), 2637–2646 (2017)
105. H. Guo, J. Liu, J. Zhang, Computation offloading for multi-access mobile edge computing in ultra-dense networks. IEEE Commun. Mag. **56**(8), 14–19 (2018)
106. X. Chen, Z. Liu, Y. Chen, Z. Li, Mobile edge computing based task offloading and resource allocation in 5g ultra-dense networks. IEEE Access **7**, 184172–184182 (2019)
107. Y. Chen, J. Huang, C. Lin, J. Hu, A partial selection methodology for efficient qos-aware service composition. IEEE Trans. Serv. Comput. **8**(3), 384–397 (2014)

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.