Proceedings of the 38th Chinese Control Conference
July 27-30, 2019, Guangzhou, China

Visual SLAM with RGB-D Cameras

Qiongyao Jin, Yungang Liu, Yongchao Man, Fengzhong Li
School of Control Science and Engineering, Shandong University, Jinan 250061, P. R. China
E-mail: lygfr@sdu.edu.cn

Abstract: This paper focuses on visual SLAM with RGB-D cameras (abbreviated as RGB-D SLAM), which has been an actively studied issue in the robotics community since RGB-D cameras can obtain depth information of environments simply. Firstly, two types of RGB-D cameras are introduced according to the principle of depth measurement. Secondly, the typical RGB-D SLAM algorithm framework is normally divided into four parts: visual odometry, optimization, loop closing and mapping. Thirdly, a series of landmark achievements on algorithm, open source libraries and tools, and performance evaluation of RGB-D SLAM are summarized. Finally, the advantages and the disadvantages, as well as the development trends of RGB-D SLAM are discussed.

Key Words: SLAM, RGB-D, 3D Vision, Visual Odometry

1 Introduction

When a mobile robot enters an unknown environment, it needs to sense the surroundings to determine its position in the environment and map the surroundings in real time. This is SLAM, i.e., simultaneous localization and mapping. SLAM was first proposed in 1986 [1], which is key to realize autonomy of mobile robots [2] [3]. When the camera is used to achieve SLAM, it is called Visual SLAM (vSLAM). Remarkably, cameras have the advantages of low cost, small size and light weight, and the image contains abundant information. Therefore, visual SLAM has become a hotspot of SLAM research area, and has made tremendous progress during the decades [4–6].

RGB-D SLAM is a visual SLAM method with RGB-D cameras as the sensor [7–9]. The RGB-D camera, also known as the depth camera, is a new type of low-cost depth sensor that first emerge in 2010 [7]. RGB-D cameras can provide both the RGB color image and the per-pixel depth image of the detected environment at the same time. Both monocular cameras and binocular cameras need triangulation calculation to obtain depth information of 3D spatial points to realize visual SLAM. But RGB-D cameras can obtain depth information directly by active physical measurement [7]. Compared with laser radar sensors, RGB-D cameras are relatively cheap. Compared with monocular cameras and binocular cameras, RGB-D cameras can save computing resources. Therefore, since the emergence of RGB-D cameras, RGB-D SLAM has developed rapidly in recent years [10–15].

The paper is organized as follows. Section II introduces the working principle and the structure components of RGB-D cameras. Section III demonstrates the RGB-D SLAM algorithm framework. Section IV summarizes a series of symbolic achievements of RGB-D SLAM algorithm. Section V analyzes the advantages and disadvantages of RGB-D SLAM algorithm. Section VI discusses the development trend of RGB-D SLAM algorithm, and Section VII concludes this paper.

This work was supported by the National Natural Science Foundation of China (61873146, 61603217, 61703237).

2 RGB-D Cameras

Nowadays RGB-D cameras can be divided into two classes according to their working principles. One is the RGB-D cameras based on Infrared-Structured-Light principle, e.g. the Microsoft Xbox 360 Kinect (Kinect v1) camera, which emit a beam of infrared light to the detected target, and then calculate the distance between the object and itself according to the received pattern of structured light [16]. The other is the RGB-D camera based on Time-of-Flight (ToF) principle, e.g. the Microsoft Xbox One Kinect (Kinect v2) camera, which transmits pulsed light to the target, and then determines the distance between the object and the camera according to the time of flight between the beams launched and received [17]. Next, Kinect v1 and Kinect v2 are taken as examples to introduce these two types of RGB-D cameras.

The Kinect v1 camera launched by Microsoft in 2010 is the first consumer RGB-D camera in the world. It is mainly composed of a color camera, an infrared-structured-light depth sensor, a microphone array and a motorized pivot. The middle part of the Kinect v1 camera is a RGB color camera with the resolution of 640×480 and the maximum frame rate of 30 Hz. The depth sensor of Kinect v1 is composed of the infrared transmitting and receiving devices on the two sides. The resolution of the depth sensor is 320×240 and the frame rate is 30Hz [17].

Fig. 1: The Microsoft Kinect v1 camera.

Later, Microsoft continued to launch the Kinect v2 camera in 2013, using more advanced technology and improving hardware quality. Kinect v2 has an infrared sensor with a larger size, and a depth camera with a wider field of view. The quality of the generated depth map is relatively nice.

The Kinect v2 camera has a RGB color camera on one side, which with a resolution of 1920×1080 and the maximum frame rate of 30Hz. The middle part of Kinect v2 is the TOF depth sensor, which with a resolution of 512×424 and a frame rate of 30 Hz. There is no tilting motor [17].

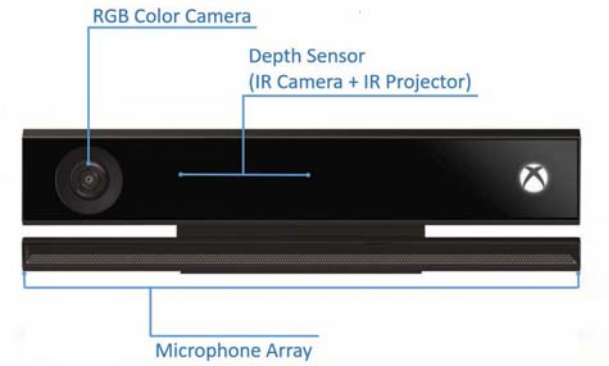

Fig. 2: The Microsoft Kinect v2 camera.

3 Algorithm Framework

When an autonomous mobile robot carrying a camera moving in a scene, it will obtain a continuous image sequence. Visual SLAM algorithm realizes real-time location and map construction simultaneously with the image sequence. After a long period of research, the framework of visual SLAM algorithm has been relatively mature and stable, especially in the static scene of rigid body, no obvious illumination change and no human interference [4–6]. RGB-D SLAM is a branch of visual SLAM, and its algorithm framework is no exception [18]. The typical RGB-D SLAM algorithm framework is shown in the Fig. 3.

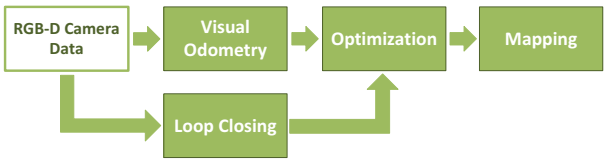

Fig. 3: The Visual RGB-D SLAM Algorithm Framework.

3.1 Visual Odometry

Visual odometry (VO) of the RGB-D SLAM algorithm aims to estimate the camera motion and the appearance of local map according to the sequential image frames of input RGB color image and per-pixel depth image pairs [19]. Visual odometry is also known as front end. According to the difference of how to use color image information, visual odometry techniques can generally be approached in two ways: through feature-based methods or direct methods [6].

Feature-based visual odometry consists of three kinds of methods, i.e. feature point-based methods, feature line-based methods and feature surface-based methods. And feature point-based methods are mainly used method in feature-based visual odometry [20].

According to feature point-based visual odometry, the key point detection and feature descriptor calculation of color images are carried out firstly, and then two adjacent frames of color images are matched according to the feature descriptor to obtain a pairs of 2D-2D feature matching points. Then, according to the depth information of depth image, the 3D spatial point corresponding to the 2D feature matching point are calculated, and the 3D-3D matching point pairs are obtained. Then Iterative Closest Point (ICP) algorithm [21] can be used to calculate the pose transformation between two adjacent frames. Finally, the motion estimation error is optimized, and the position and attitude estimation result with the least error is obtained [7, 8, 15]. Popular general purpose point features include SIFT [22], SURF [23] and ORB [24] so on.

Compared with feature point-based methods for extracting point features from color images, feature line-based methods [25, 26] and feature plane-based methods [27, 28] are for extracting line features or plane features from color images.

But direct visual odometry matches the pixel of two adjacent frames directly according to the pixel value of color images, and then obtains a pairs of 2D-2D pixel matching points [29, 30].

In conclusion, the feature-based methods has long been regarded as the mainstream method of visual odometry. It runs stably and is invariance to illumination and dynamic objects [15]. Nowadays these are mature methods. Besides, Most of feature-based visual odometry uses feature point-based methods, because general scenes can provide rich feature points [31]. Feature point-based methods can be applied to various scenes, while feature line-based methods or feature surface-based methods is only applicable to artificial scenes. In the absence of texture features and large numbers of lines and curves in the environment, methods based on edge feature and region feature is better [26, 27] .

3.2 Optimization

Because optimization is connected after the visual odometry, it is also called the back end. The main purpose of the optimization is to solve the problem of noise interference in SLAM. The real data acquired by the sensor will be noisy, so the motion estimation between two adjacent image frames estimated by the front end is noisy [32]. The motion estimation of front end is treated as the initial value of back end. From these noisy data, the state of the whole system with a maximum posteriori probability is estimated. Specifically, the back end receives the information of camera pose measured by visual odometry at different times, as well as he information of loop detection, then the back end optimizes both of them to obtain globally consistent trajectories and maps.

Optimization methods are generally divided into two classes: filter-based methods or graph optimization-based methods. In these early studies of SLAM, filter-based optimization was mostly used, especially the extended Kalman filter (EKF) [33]. But in recent years, the commonly used graph-based optimization [34] is considered to be superior to the traditional filter-based optimization, and has become the mainstream method of RGB-D SLAM [35]. Graph-based optimization can overcome the shortcomings of EKF, such as linearization error and Gaussian distribution assumption of noise. Based on the graph-based optimization method-

s, the RGB-D SLAM problem is transformed into a graph optimization problem by graph theory, and solved by least square method.

3.3 Loop Closing

Pose estimation between two frames of visual odometry will inevitably accumulate errors when it runs for a long time. The main purpose of loop closing is to solve the problem of cumulative drift over time in robot pose estimation. Loop closing detects that if the camera passes through the same place and collects similar data, then to correct the pose estimation and eliminate drift. The key difficulty of loop closing is to identify the location where the camera has arrived. Williams et al. [36] compares the loop closing methods and concludes that the image-to-image match performance is better than map-to-map match performance and image-to-map match performance. Therefore, the current mainstream method is appearance-based loop closing. The key problem is how to calculate the similarity between two images.

Appearance-based loop closing mostly uses image feature-based methods, that is, matching the feature points of two images. If the number of matches is exceed the threshold, it is considered that there is a loop. Bag-of-words (BoW) [37] is a method of describing an image by which features are on the image. The bag-of-words method uses the visual vocabulary tree to represent an image with a vector. In this way, the similarity between two images can be measured by the distance between the corresponding vectors. The commonly used bag-of-words methods include DBOW2 [38], FAB-MAP 2.0 [39] and so on.

3.4 Mapping

Mapping refers to the process of constructing a description of the scene. The expression types of maps depend on different applications such as location, navigation, obstacle avoidance, reconstruction and interaction. Common types of maps include topological maps, sparse maps, dense maps, semantic maps and so on.

Topological maps are graphs composed of nodes and edges [40]. It only considers the connectivity between nodes, ignores the details of nodes. It does not need to know the exact location of the nodes in the map. It is a compact expression. However, because topological maps are not good at expressing complex structures, it is difficult to apply them in location and navigation.

Sparse maps abstract the scene by selecting some representative landmarks in the scene to make up the map [41]. This kind of map is a sparse expression. Sparse maps can only be used for location .

Dense maps aim to describe all the details seen in the scene [42]. This kind of map is a dense expression. In the confrontation of navigation, obstacle avoidance, three-dimensional reconstruction and other needs, it is necessary to build a dense map. Both sparse maps and dense maps are metric maps.

Semantic maps have a higher level of knowledge. In the existing dense map, the robot needs to know which word the part of map represents, and which part of map the word describes [43]. When there is a need for human-computer interaction, semantic maps need to be constructed.

4 Landmark Achievements

4.1 Algorithm

Henry et al. proposed the first published RGB-D SLAM system [7]. They used SIFT feature to extract feature point from two adjacent RGB color images, and added depth data to generate 3D feature point. Random sample consensus (RANSAC) [44] was used to match 3D feature points, and ICP method was used to solve motion transformation matrix. In the back end, Tree-based network Optimizer (TORO) [45] and appearance-based loop detection are used to generate a three-dimensional point cloud map. The SLAM system proposed by Henry et al. opened up a research hotspot of RGB-D SLAM. The disadvantage of the system is that it is difficult to run in real time on CPU because SIFT has high computational complexity.

KinectFusion [10] proposed by Newcombe et al. is an RGB-D SLAM system which can reconstruct three-dimensional maps in real time on GPU based on Kinect. KinectFusion uses only depth information to generates three-dimensional point clouds from depth images, estimates camera motion through ICP, and expresses reconstruction maps with TSDF model. KinectFusion was not added in loop closing, resulting the accumulation of errors can not be eliminated. A grid model of the fixed volume is used to build maps, so only smaller scenes can be reconstructed.

RGBDSLAMv2 [13] proposed by Endres et al. uses visual odometry based on point features of SIFT, SURF and ORB to estimate the camera motion. It optimizes the global pose by graph optimization, and was added in loop closing to optimize the cumulative drift of visual odometry. It generates three-dimensional point cloud and expresses the map with OctoMap model. RGBDSLAMv2 is a comprehensive RGB-D SLAM system which integrates most of advanced technologies of SLAM, such as image features, graph optimization, loop closing, point cloud, octree-based mapping etc. It is suitable for researchers to redevelop a RGB-D SLAM system on the basis of RGBDSLAMv2. The disadvantage of RGBDSLAMv2 is that its algorithm consumes expensive computation and has poor real-time performance. The robot with the RGB-D camera need to move slowly to make RGBDSLAMv2 work well.

RTAB-MAP [14] proposed by Labbe et al. is also a comprehensive RGB-D SLAM system, which was added in all the things that RGB-D SLAM should have: visual odometry based on image features, back-end pose optimization based on graph optimization, loop closing based on bag-of-words, and mapping based on point cloud and triangular grid map. RTAB-Map is an excellent RGB-D SLAM system. Through STM/WM/LTM memory management mechanism, RTAB-Map reduces the number of nodes needed in graph optimization and loop closing. It ensures the real-time performance of the system and the accuracy of closing. It can run in large-scale scenes. The disadvantage of RTAB-Map is that it is difficult for researchers to redevelop due to the high integration level.

ORB-SLAM2 [15] proposed by Mur-Artal et al. is an open source SLAM system supporting monocular, binocular and RGB-D cameras, which is extended from the monocular ORB-SLAM algorithm. ORB-SLAM2 employs three threads: tracking, mapping and closed-loop detection. The

whole system calculates around ORB features, i.e. visual odometry based on ORB features, creates sparse maps based on ORB features, and detects loops based on the ORB visual vocabulary. ORB-SLAM2 is an excellent and easy-to-use system, which can achieve real-time operation on CPU and has a high precision. The disadvantage of ORB-SLAM2 is that only sparse maps are constructed, which is difficult to meet the needs of navigation and obstacle avoidance.

The characteristics of RGB-D SLAM algorithms above mentioned are summarized in the following table 1.

Table 1: RGB-D SLAM Algorithms

Component / Algorithm	Visual Odometry	Optimization	Loop Closing	Mapping
Henry [7]	SIFT point feature	TORO	appearance-based	3D point cloud map
KinectFusion	only depth image	frame-to-model	No	3D point cloud map and TSDF model
RGBDSLAMv2	SIFT, SURF and ORB point feature	graph optimization	BoW	3D point cloud map and OctoMap
RTAB-MAP	SURF point feature	graph optimization	BoW	3D point cloud map and triangular grid map
ORB-SLAM2	ORB point feature	graph optimization	BoW	3D point cloud map

4.2 Open Source Libraries and Tools

Open Source Computer Vision Library (OpenCV) is a cross-platform open source computer vision library. It is written in C++ language and supports interfaces of C/C++, Python, Ruby, MATLAB etc. OpenCV implements lots of general algorithms in aspect of image processing and computer vision, such as feature extraction and matching, camera calibration and pose calculation. It is convenient for researchers to achieve a SLAM system. Nowadays, two versions of OpenCV (OpenCV2 and OpenCV3) are developed and maintained at the same time [46].

A General framework for Graph Optimization (G2o) is an open source general graph optimization tool, which is the back-end optimizer commonly used in SLAM system [47]. Firstly, the non-linear optimization problem needs to be transformed into graph optimization problem. Then, the gradient descent methods such as the Gauss-Newton (G-N) method and the Levenberg-Marquart (L-M) method provided by G2o are used to solve the iterative solution.

Point Cloud Library (PCL) is an open source point cloud library. It implements a large number of common algorithms and efficient data structures related to point clouds, such as point cloud acquisition, filtering, segmentation, registration, retrieval, feature extraction, recognition, tracking, surface reconstruction, visualization and so on [48]. It is mainly used for 3D information processing, especially for 3D mapping in SLAM.

Robot Operating System (ROS) is an open source software on Linux. The main goal of ROS is to support code reuse for robot development [49]. It provides a number of services such as hardware abstraction, underlying device control, function implementation, message subscription transmission and package management. At present, there are many open source software packages to achieve different functions, such as camera acquisition, camera calibration and so on.

4.3 Performance Evaluation

In order to evaluate the performance of different RGB-D SLAM systems, researchers in Munich Polytechnic University established TUM RGB-D Benchmark [50]. It is a benchmark for visual RGB-D odometry and visual RGB-D SLAM system evaluation. It consists of TUM RGB-D SLAM Dataset and TUM RGB-D Benchmark Tools.

TUM RGB-D SLAM Dataset contains RGB-D data and ground truth data. The RGB-D data consists of color images and depth images captured by Kinect moving along real ground trajectory. The ground truth data is the real ground trajectory recorded by a high-precision motion capture system. The dataset is large and contain many scenarios such as offices, factories, etc.

Absolute Trajectory Error (ATE) and Relative Pose Error (RPE) are two prominent error measurement methods. ATE is commonly used to evaluate the performance of visual SLAM system, and RPE is commonly used to evaluate the drift of visual odometry. TUM RGB-D Benchmark Tools is an RGB-D SLAM system performance evaluation tool that can help researchers to measure the two errors.

5 Advantages and Disadvantages

Monocular vision has the problem of scale ambiguity. Monocular cameras can only get a single image at one time, and the depth information of the observed object can not be obtained by a single image. Therefore, monocular SLAM needs a moving to get two images with parallax and calculate the depth of a pixel by triangulation. Therefore, the initialization of monocular SLAM is unavoidable. Binocular cameras can obtain two parallax images at the same time, but it is difficult to obtain the depth information of each pixel of the image in real time due to the expansive computational costs. But RGB-D cameras can measure the depth of image pixel directly by physical measurement of infrared sensors. Compared with monocular SLAM and binocular SLAM, RGB-D SLAM does not have problems of initialization and expansive computation. Therefore, it is relatively easy to construct dense maps for RGB-D SLAM. In addition, RGB-D SLAM can obtain depth data accurately even there is lack of texture in the scene.

RGB-D SLAM also has many disadvantages. RGB-D cameras need to emit and receive infrared light, whether based on Infrared-Structured-Light principle or Time-of-Flight principle. Therefore, they are susceptible to the interference of sunlight and can hardly be used in outdoor environment. Besides, infrared light will interfere with each other, so it is necessary to avoid multiple RGB-D cameras being used together. In addition, RGB-D cameras are also difficult to measure the depth of transparent objects with less reflected light. Moreover, the depth measurement range of RGB-D cameras is generally less than 20 meters, and the accuracy of the RGB-D cameras is poor when measuring further afield.

6 Development Trends

One of the research hotspots of RGB-D SLAM is combining it with deep learning. Nowadays visual odometry of RGB-D SLAM system still mainly uses the low-level image features designed manually, while the deep learning method can extract the high-level semantic features of the image automatically and enhance the robustness and scalability of the

system in practical application [51]. Using deep learning to construct semantic maps can encourage high level thinking for robot to perceive environment and improve human-robot interaction. The appearance-based loop closing problem is similar to the image retrieval problem and the place recognition problem. Since the excellent performance of deep learning for image recognition and object detection, deep learning methods is potential to achieve better results in loop closing than traditional methods in the future [52].

Besides, lightweight RGB-D SLAM system is a development trend. Nowadays RGB-D SLAM algorithms need to use CPU to realize real-time operation, and some even need to use GPU [53]. If we want to apply the RGB-D SLAM system on robots and AR/VR devices, we must to make RGB-D SLAM algorithm run in real time on embedded processors [54]. And SLAM technology is only an underlying technology serving for upper applications, it can only occupy a small part of the computing resources of embedded processors, so we need to improve efficiency of RGB-D SLAM systems.

Another development trend is to fuse RGB-D cameras with other sensors to achieve SLAM systems. Nowadays, RGB-D SLAM fusion with Inertial Measurement Unit (IMU) is a research hotspot. Because of the blurred motion due to curtain shutter of camera, and the lack of overlapping areas between two adjacent frames of image, the camera motion will be hardly estimated when the camera moves too fast. But the IMU can measure the palstance and acceleration of the object. During the period when the camera data is invalid, we can realize a better motion estimation with IMU than pure visual RGB-D SLAM [55]. Fusion with IMU is a good solution for the fast motion of the camera. In turn, the camera can solve the drift problem of IMU in the slow motion [56]. They are complementary. The combination of the two sensors can improve the robustness of SLAM system, which is a promising development direction.

7 Conclusions

Focusing on the RGB-D SLAM problem, this paper firstly introduced the principles, types and products of RGB-D cameras. Then the RGB-D SLAM algorithm framework consisting of visual odometry, optimization, loop closing and mapping was introduced, as well as a series of landmark achievements. Finally, the advantages and disadvantages of RGB-D SLAM and its possible development trend were discussed.

Since the advent of RGB-D cameras, the use of RGB-D cameras as sensors to achieve SLAM has become a research topic. After many years of development, RGB-D SLAM has made a series of significant achievements, and the framework of RGB-D SLAM algorithm has been confirmed. However, there are still many challenges for RGB-D SLAM from research to practical application. In order to apply RGB-D SLAM to autonomous mobile robots and AR/VR devices, the robustness and efficiency of RGB-D SLAM needs to be improved. The combination with deep learning, the fusion with IMU, and the lightweight of RGB-D SLAM will be the future development trends.

References

[1] R. C. Smith, and P. Cheeseman, "On the Representation and Estimation of Spatial Uncertainty," The International Journal of Robotics Research, vol. 5, no. 4, pp. 56-68, Dec. 1986.

[2] M. Csorba, "Simultaneous localization and map building," Ph.D. dissertation, Univ. Oxford, Robot. Res. Group, 1997.

[3] M. W. M. G. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte and M. Csorba, "A solution to the simultaneous localization and map building (SLAM) problem," IEEE Transactions on Robotics and Automation, vol. 17, no. 3, pp. 229-241, Jun. 2001.

[4] J. Fuentes-Pacheco, J. Ruiz-Ascencio and J. M. Rendom-Mancha, "Visual simultaneous localization and mapping: a survey," Artificial intelligence review, vol. 43, no. 1, pp. 55-81, Jan. 2015.

[5] C. Cadena et al., "Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age," IEEE Transactions on Robotics, vol. 32, no. 6, pp. 1309-1332, Dec. 2016.

[6] T. Takafumi, H. Uchiyama and S. Ikeda, "Visual SLAM algorithms: a survey from 2010 to 2016," IPSJ Transactions on Computer Vision and Applications, vol. 9, no. 16, pp.1-11, 2017.

[7] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox, "RGB-D mapping: Using depth cameras for dense 3D modeling of indoor environments," Intl. Symp. on Experimental Robotics (ISER), 2010.

[8] N. Engelhard, F. Endres, J. Hess, J. Sturm, and W. Burgard, "Realtime 3D visual SLAM with a hand-held RGB-D camera," RGB-D Workshop on 3D Perception in Robotics at the European Robotics Forum, 2011.

[9] C. Audras, A. Comport, M. Meilland, and P. Rives, "Real-time dense appearance-based SLAM for RGB-D sensors," Australasian Conf. on Robotics and Automation, 2011.

[10] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. W. Fitzgibbon, "Kinectfusion: Real-time dense surface mapping and tracking," in Proc. IEEE Int. Symp. Mixed Augmented Reality, pp. 127-136, 2011.

[11] T. Whelan, H. Johannsson, M. Kaess, J. Leonard, and J. McDonald, "Robust real-time visual odometry for dense RGB-D mapping," IEEE Int. Conf. Robotics Automation, Karlsruhe, Germany, May 2013.

[12] T. Whelan, R.F. Salas-Moreno, B. Glocker, A.J. Davison and S. Leutenegger, "ElasticFusion: Real-Time Dense SLAM and Light Source Estimation," The International Journal of Robotics Research, vol. 35, no. 14, pp. 1697-1716, Dec. 2016.

[13] F. Endres, J. Hess, J. Sturm, D. Cremers and W. Burgard, "3-D Mapping With an RGB-D Camera," IEEE Transactions on Robotics, vol. 30, no. 1, pp. 177-187, Feb. 2014.

[14] M. Labbe and F. Michaud, "Online global loop closure detection for large-scale multi-session graph-based SLAM," IEEE International Conference on Intelligent Robots and Systems, Chicago, IL, pp. 2661-2666, 2014.

[15] R. Mur-Artal and J. D. Tardos, "ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras," IEEE Transactions on Robotics, vol. 33, no. 5, pp. 1255-1262, Oct. 2017.

[16] K. Litomisky, "Consumer RGB-D Cameras and their Applications," Tech. rep. University of California, page 20, 2012.

[17] Wikipedia, "Kinect," https://en.wikipedia.org/wiki/Kinect, 2018.

[18] Khalid Yousif, Alireza Bab-Hadiashar and Reza Hoseinnezhad, "An Overview to Visual Odometry and Visual SLAM: Applications to Mobile Robotics," Intelligent Industrial Systems, Vol. 1, no. 4, pp. 289-311, Dec. 2015.

[19] D. Nister, O. Naroditsky and J. Bergen, "Visual odometry," Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2004),

Washington, DC, USA, pp. I-I, 2004.
[20] T. Taketomi, H. Uchiyama and S. Ikeda, "Visual SLAM algorithms: a survey from 2010 to 2016," IPSJ Transactions on Computer Vision and Applications, vol. 9, no. 16, pp. 1-11, 2017.
[21] P. J. Besl and H. D. McKay, "A method for registration of 3-D shapes," IEEE Trans. Pattern Anal. Mach. Intell., vol. 14, no. 2, pp. 239-256, Feb. 1992.
[22] D. Lowe, "Distinctive image features from scale-invariant keypoints," Int. J. Comput. Vis., vol. 60, no. 2, pp. 91-110, 2004.
[23] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," Comput. Vis. Image Understanding, vol. 110, pp. 346-359, 2008.
[24] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," Proc. IEEE Int. Conf. Comput. Vis., 2011, vol. 13, pp. 2564-2571.
[25] C. Choi, A. J. Trevor, and H. I. Christensen, "RGB-D edge detection and edge-based registration," IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1568-1575, 2013.
[26] Y. Lu and D. Song, "Robustness to lighting variations: An RGB-D indoor visual odometry using line segments," IEEE/RSJ International Conference on Intelligent Robots and Systems, Hamburg, pp. 688-694, 2015.
[27] C. Raposo, M. Lourenco, J. P. Barreto, and M. Antunes, "Plane-based odometry using an RGB-D camera," British Machine Vision Conference, 2013.
[28] Y. Taguchi, Y.-D. Jian, S. Ramalingam, and C. Feng, "Point-plane SLAM for hand-held 3D sensors," IEEE International Conference on Robotics and Automation, pp. 5182-5189, 2013.
[29] C. Kerl, J. Sturm and D. Cremers, "Dense Visual SLAM for RGB-D Cameras," Int. Conf. on Intelligent Robot Systems, 2013.
[30] L. Ma, C. Kerl, J. Stuckler and D. Cremers, "CPA-SLAM: Consistent plane-model alignment for direct RGB-D SLAM," IEEE International Conference on Robotics and Automation, Stockholm, pp. 1285-1291, 2016.
[31] Y. Lu and D. Song, "Robust RGB-D Odometry Using Point and Line Features," IEEE International Conference on Computer Vision, Santiago, pp. 3934-3942, 2015.
[32] F. Endres, J. Hess, N. Engelhard, J. Sturm, D. Cremers and W. Burgard, "An evaluation of the RGB-D SLAM system," IEEE International Conference on Robotics and Automation, Saint Paul, MN, pp. 1691-1696, 2012.
[33] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," Transaction of the ASME-Journal of Basic Engineering, pp. 35-45, 1960.
[34] U. Frese, P. Larsson and T. Duckett, "A multilevel relaxation algorithm for simultaneous localization and mapping," IEEE Transactions on Robotics, vol. 21, no.2, pp. 196-207, 2005.
[35] H. Johannsson, M. Kaess and M. Fallon, "Temporally scalable visual SLAM using a reduced pose graph," Cambridge, USA: Computer Science and Artificial Intelligent Laboratory, MIT, 2012.
[36] B. P. Williams, M. J. Cummins, J. Neira, P. Newman, I. D. Reid and J. D. Tardos, "A comparison of loop closing techniques in monocular SLAM," Robotics and Autonomous Systems, vol. 57, no. 12, pp. 1188-1197, 2009.
[37] J. Sivic and A. Zisserman, "Video Google: A Text Retrieval Approach to Object Matching in Videos," in Proc. IEEE International Confernece on Computer Vision, pp. 1470, 2003.
[38] D. Galvez-Lopez and J. D. Tardos, "Bags of Binary Words for Fast Place Recognition in Image Sequences," IEEE Transactions on Robotics, vol. 28, no. 5, pp. 1188-1197, 2012.
[39] M. Cummins and P. Newman, "Appearance-only SLAM at large scale with FAB-MAP 2.0," The International Journal of Robotics Research, vol. 30, no. 9, pp. 1100-1123, 2011.
[40] S. P. Wilson and J. A. Bednar, "What, if anything, are topological maps for?" Developmental neurobiology, vol. 75, no. 6, pp. 667-81, 2015.
[41] P. Cavestany, A. L. Rodriguez, H. Martinez-Barbera and T. P. Breckon, "Improved 3D sparse maps for high-performance SFM with low-cost omnidirectional robots," IEEE International Conference on Image Processing, Quebec City, QC, pp. 4927-4931, 2015.
[42] T. Whelan, M. Kaess, H. Johannsson, M. Fallon, J. J. Leonard and J. McDonald, "Real-Time Large-Scale Dense RGB-D SLAM with Volumetric Fusion" The International Journal of Robotics Research, vol. 34, no. 4-5, pp. 598-626, 2015.
[43] Z. Zhao and X. Chen, "Building 3D semantic maps for mobile robots using RGB-D camera," Intelligent Service Robotics, vol. 9, pp. 297-309, 2016.
[44] M. Fischler and R. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," Commun. ACM, vol. 24, no. 6, pp. 381-395, 1981.
[45] G. Grisetti, C. Stachniss, S. Grzonka and W. Burgard, "A Tree Parameterization for Efficiently Computing Maximum Likelihood Maps using Gradient Descent," Robotics: Science and Systems, 2007.
[46] A. Kaehler and G. Bradski, "Learning OpenCV 3: computer vision in C++ with the OpenCV library," O' Reilly, 2016.
[47] R. Kummerle, G. Grisetti, H. Strasdat, K. Konolige and W. Burgard, "G2o: A general framework for graph optimization," IEEE International Conference on Robotics and Automation, Shanghai, pp. 3607-3613, 2011.
[48] R. B. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)," IEEE International Conference on Robotics and Automation, Shanghai, pp. 1-4, 2011.
[49] M. Quigley, B. P. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler and A. Ng, "ROS: an open-source Robot Operating System," 2009.
[50] J. Sturm, N. Engelhard, F. Endres, W. Burgard and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," IEEE/RSJ International Conference on Intelligent Robots and Systems, Vilamoura, pp. 573-580, 2012.
[51] L. Ma, J. Stuckler, C. Kerl and D. Cremers, "Multi-view deep learning for consistent semantic mapping with RGB-D cameras," IEEE/RSJ International Conference on Intelligent Robots and Systems, Vancouver, BC, pp. 598-605, 2017.
[52] J. Li, H. Zhan, B. M. Chen, I. Reid and G. H. Lee, "Deep learning for 2D scan matching and loop closure," IEEE/RSJ International Conference on Intelligent Robots and Systems, Vancouver, BC, pp. 763-768, 2017.
[53] N. Marniok and B. Goldluecke, "Real-Time Variational Range Image Fusion and Visualization for Large-Scale Scenes Using GPU Hash Tables," IEEE Winter Conference on Applications of Computer Vision, Lake Tahoe, NV, pp. 912-920, 2018.
[54] C. Forster, Z. Zhang, M. Gassner, M. Werlberger and D. Scaramuzza, "SVO: Semidirect Visual Odometry for Monocular and Multicamera Systems," IEEE Transactions on Robotics, vol. 33, no. 2, pp. 249-265, 2017.
[55] T. Qin, P. Li and S. Shen, "VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator," IEEE Transactions on Robotics, vol. 34, no. 4, pp. 1004-1020, 2018.
[56] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and Paul Furgale, "Keyframe-Based Visual-Inertial Odometry Using Nonlinear Optimization," The International Journal of Robotics Research, vol. 34, no. 3, pp. 314-334, 2015.