A comprehensive survey of advanced SLAM techniques

Nikolai Svishchev[1,*], *Paolo* Lino[1], *Guido* Maione[1], and *Iskandar* Azhmukhamedov[2]

[1]Dept. of Electrical and Inf. Eng., Polytechnic University of Bari, 70125 Bari, Italy
[2]Dept. of Digital Technology and Cybersecurity, Astrakhan State University, 414056 Astrakhan, Russia

Abstract. In robot navigation, precise knowledge of the robot's position and orientation is essential for accurate trajectory tracking, obstacle avoidance, and goal attainment, especially in scenarios where human supervision is limited or absent. This paper describes the different established methods in simultaneous localization and mapping (SLAM) algorithms, such as the most advanced SLAM techniques for extreme environmental conditions, including dynamic objects, illumination and brightness variability. Namely, visual information received from cameras is less susceptible to radio interference and does not depend on any additional device, such as GPS and satellite signals. The SLAM community's main approaches to solving these problems are introduced. Finally, we consider current research in the field of visual odometry (VO), as well as its practical implementation in robotics.

1 Introduction

Techniques such as odometry, visual odometry (VO), Simultaneous Localization and Mapping (SLAM), and Visual-based Simultaneous Localization and Mapping (VSLAM) are employed to achieve navigation, mapping, and localization tasks, enhancing the autonomy of robotic systems.

SLAM offers distinct advantages over traditional odometry and visual odometry methods, facilitated by the rapid advancement of computational resources. These advantages are outlined in Table 1.

Numerous papers have analysed the diverse landscape of SLAM methods [1–5]. Some, like [1, 4], delve into various SLAM concepts such as Visual-Only, Visual-Inertial, and RGB-D SLAM, along with current research trends. Others, like [3, 5], review classical optimization techniques commonly used in SLAM backend.

SLAM methods' practical implementation often involves different algorithms tailored for embedded systems [6]. Graph-based optimization techniques are prevalent [7], while deep learning-based optimization methods are also gaining traction [2, 8–11].

Review papers mainly focus on SLAM methods, with some addressing specific subproblems like path planning and navigation [12, 13]. These surveys typically emphasise

*Corresponding author: svishchev.nikolai@gmail.com

SLAM's application in control engineering rather than backend processes and methodologies.

Table 1. Comparison between SLAM and VO.

Parameter	SLAM	VO
Global Consistency	SLAM integrates both localization and mapping simultaneously, ensuring global consistency in the generated map.	VO methods often suffer from drift over time, leading to inaccuracies in the estimated pose and map.
Robustness	SLAM algorithms can handle dynamic environments and changes in scene structure, as they continuously update the map and refine the robot's pose estimation based on incoming sensor data.	VO methods may struggle in dynamic environments or when scene changes occur, leading to degraded performance.
Reduced Reliance on External Sensors	SLAM integrates data from various sensors (e.g., LiDAR, IMU, cameras, etc) to improve accuracy and robustness.	VO methods rely primarily on visual sensor data (e.g., cameras) for motion estimation.
Mapping Capability	SLAM produces a detailed map of the environment while simultaneously estimating the robot's pose within that map. This mapping capability is essential for applications such as autonomous navigation, exploration, and environment monitoring, where a precise representation of the surroundings is crucial.	VO focuses primarily on estimating the camera pose (position and orientation) relative to its initial position, thus mainly addressing the localization aspect.

[14] offer a comprehensive review of visual SLAM methods, particularly those relying solely on vision sensors. Another notable work by [15] addresses sensor fusion methods to mitigate drifting errors and inaccuracies in visual sensor data.

[16] tackle optical distortion issues in visually acquired data, providing insights into mitigating this challenge. Conversely, [17] present an introductory guide to SLAM principles, targeting newcomers to the field.

In addition to graph-based techniques, filter approaches like RANSAC and Extended Kalman Filter (EKF) are utilised in SLAM systems for state prediction, estimation, and correction [18–20]. Semantic information from neural networks is also applied to classify objects in visual data, addressing challenges in motion estimation in dynamic environments [21–25].

The paper provides a thorough overview of SLAM methods, systematically organised by hardware and software specifications. We focus on the practical implementation of these categories, analysing recent publications' results from the past five years. Table 2 compiles the main SLAM surveys to date, emphasising the discussed topics in recent years.

Table 2 illustrates the diverse spectrum of SLAM methods, tailored to various hardware, software, mobile platforms, and environmental conditions. However, the scarcity of comprehensive review papers categorising these state-of-the-art methods stems from the rapid technological advancements in hardware and software domains. This study aims to address this gap by providing a comprehensive review of recent innovations in the SLAM community over the past three years, organised by hardware, software, and environmental conditions.

Table 2. The SLAM survey literature.

Paper	Review topic	Year
[12]	SLAM methods used for quadrotor drones.	2023
[13]	Review of SLAM methods based on semantic information.	2023

Continuation of Table 2.

[16]	Review of reflective detection methods in SLAM applications.	2023
[15]	Review of camera, LiDAR, and IMU based multi-sensor fusion methods in SLAM problem.	2023
[17]	SLAM technical performance review based on designed SLAM benchmark.	2023
[14]	Survey on Visual SLAM methods only.	2022
[1]	General SLAM review with history of each SLAM category evolution.	2022
[26]	Review of SLAM methods specialised for autonomous driving vehicles.	2022
[5]	Review of SLAM methods specialised on localization strategies for autonomous mobile robots.	2021
[2]	A survey of deep learning based SLAM techniques.	2021

The paper organisation is as follows. Section 2 briefly summarise the principles and structure of the visual-based SLAM methods. Sections 3, 4 and 5 describe Visual-Only SLAM, Visual-Inertial SLAM and RGB-D SLAM methods, respectively. Finally, Section 6 concludes the paper and gives some final hints about SLAM development.

2 Visual-based SLAM methods

Several review articles on SLAM methods were published in the past, many of them described all existing techniques in SLAM, and their advantages with classical VO and visual servoing systems [1, 2]. In short, the family of VSLAM methods could be presented as a group of four main approaches (Figure 1).

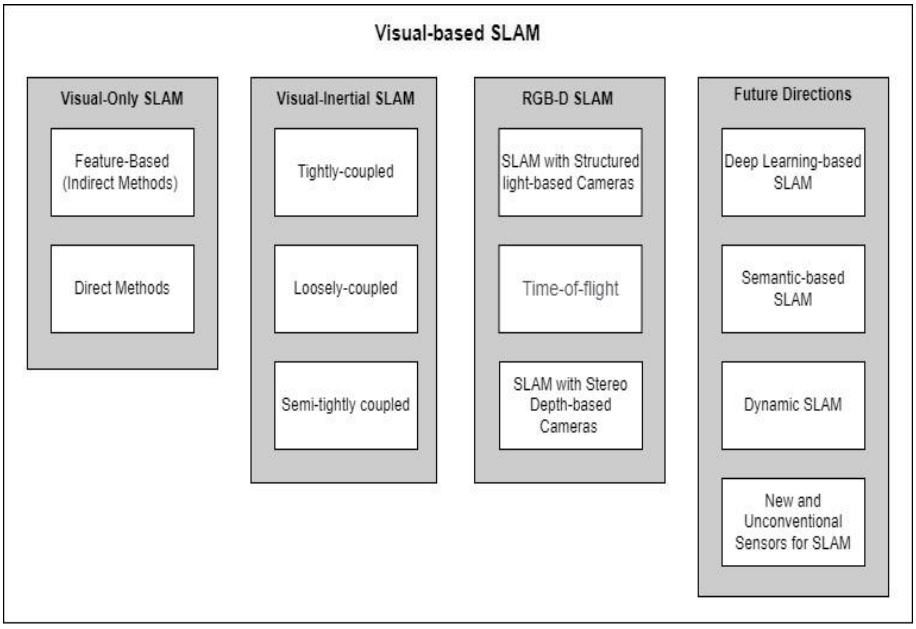

Fig. 1. Family of Visual-based SLAM methods.

Modern SLAM approaches like Visual-Only, Visual-Inertial, and RGB-D SLAM are extensively applied across various tasks. Visual-Inertial methods utilise accelerometer and gyroscope data to enhance motion estimation accuracy [7]. RGB-D SLAM employs stereo cameras or structured light sensors for depth measurement, with key points or full image information extraction techniques [27, 28].

3 Visual-only SLAM

Visual-Only SLAM (VSLAM) is a classical SLAM method that usually includes four main parts: visual sensor data integration, frontend, backend, loop-closure and map merging parts (Figure 2). The parts of Visual-Only SLAM algorithms were well described in review papers [1, 29].

Typically, SLAM systems consist of two components: a frontend system parsing sensor measurements to construct an optimization problem, and a backend solver optimising robot poses and map features. The main optimization occurs in the backend, where local and global graph optimization takes place, and the map and current position are refined. The backend employs filtering methods like the Extended Kalman Filter (EKF) [19, 20] and non-linear optimization techniques such as graph optimization for loop closure. Current major SLAM systems primarily rely on non-linear optimization due to the inefficiency of filtering methods, especially for large-scale scenes. Backend processing involves internal loop-closure optimization, ensuring globally and locally consistent maps [30, 31].

VSLAM methods are categorised into Features-Based (Indirect) and Direct methods. Features-Based methods extract feature points from images, while Direct methods use all pixel information for motion estimation, relying on photometric error minimization. Both methods further divide into dense, semi-dense, and sparse approaches. In dense methods, the map is represented as a point cloud, while sparse methods use singular points, resulting in a sparse map. Semi-dense methods utilise pixels with explicit gradients. LSD-SLAM is a classic VSLAM employing semi-dense depth maps [32-34].

Hybrid algorithms combine Direct and Indirect methods' advantages, addressing long-term navigation drift or incorporating loop-closure optimization into feature-based methods [35].

3.1 Features-based methods (indirect methods)

Indirect methods in VSLAM first of all process the measurement data to create an intermediate layer, which is implemented by extracting and matching sparse feature points, or by using a dense optical flow or by extracting straight or curved features from the flow (Figure 3), minimises a geometric error of the following reprojection error form:

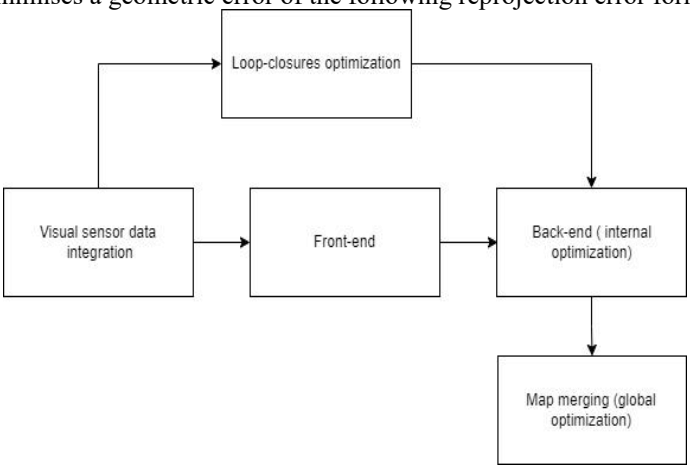

Fig. 2. Diagram of the Visual-Only SLAM.

$$T_{k-1,k} = \underset{T}{argmin} \sum_{i} \|u_i' - \pi(p_i)\|_{\Sigma}^2 \quad (1)$$

where T is the transformation matrix between two frames, which could be represented by a six-vector in the Lie algebra or by the rotation and translation matrices; is the covariance matrix associated to the scale at which a feature has been extracted; $\pi(p_i)$ is the measurement prediction function of the point p_i in space, where $\pi()$ is distribution function of prime numbers, or pi-function, which also includes the camera model. u_i is any pixel point in I_{k-1} frame, the coordinate of its projection onto a point in a space is p_i, and u_i is the coordinate of the projection of p_i onto the I_k frame. The values of the intermediate layers are then used to evaluate the 3D model of the environment and camera movement.

The ORB (Oriented FAST and Rotated BRIEF) algorithm refined BRIEF's performance during image rotation [36]. ORB-SLAM, built on ORB features, became a cornerstone in SLAM research, offering efficient feature detection and robust odometry calculation [37]. ORB-SLAM2 expanded its capabilities to monocular, binocular, and RGB-D setups, integrating closed-loop systems and map reuse [38]. However, ORB-SLAM variants still face challenges, including high computational demands, sensitivity to dynamic objects, and limited integration with inertial sensors [7].

3.2 Direct Methods

Direct methods use measurements directly in pose estimation. In passive vision, the direct method minimises the photometric error, expressed as:

$$T_{k-1,k} = \underset{T}{argmin} \sum_{i} \|I_k u_i' - I_{k-1}(u_i)\|_{\sigma}^2 \quad (2)$$

$$u_i' = \pi\left(T \cdot (\pi^{-1}(u_i) \cdot d)\right) \quad (3)$$

Where σ is the standard deviation of the Gaussian distribution, d is a depth of the image's point and other variables which were defined in (1).

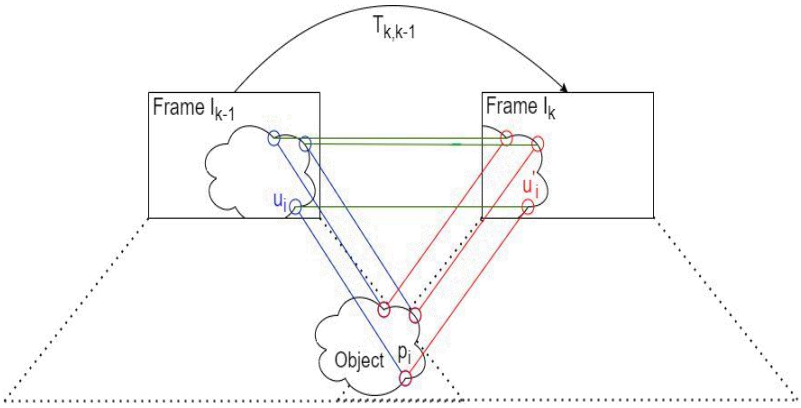

Fig. 3. The camera pose estimation method is based on the reprojection errors of feature point pairs between previous frame I_{k-1} and current frame I_k.

DTAM (Dense Tracking And Mapping) employs direct dense pose estimation for monocular VSLAM [39], utilising global energy minimization to construct dense depth maps and

computing camera positions from them. However, DTAM's highly computational processes require parallel computation on both GPU and CPU. LSD-SLAM (Large Scale Direct SLAM) addresses scalability and embeddability by proposing direct scale estimation and semi-dense scene reconstruction on CPU [32]. Yet, it suffers from sensitivity to camera parameters and lacks robustness in fast motion.

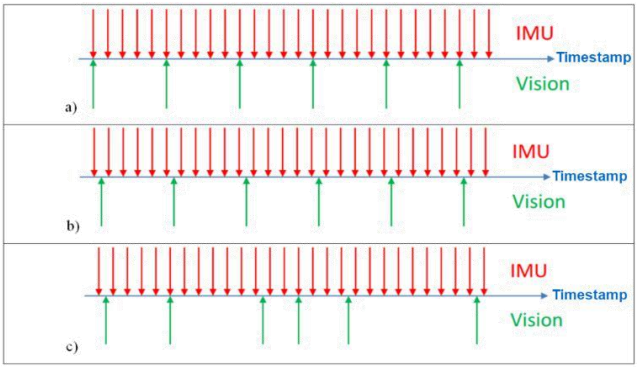

Fig. 4. a) Visual data from camera (Vision) and IMU have their own rates and timestamps aligned;b) The timestamp has a certain deviation; c) Camera and IMU rates are inconsistent and cannot be combined.

DSO (Direct Sparse Odometry) achieves high accuracy but lacks loop detection, leading to error accumulation over time [33]. LDSO extends DSO by incorporating loop-closure and global map optimization techniques [34], enhancing long-term robustness.

4 Visual-inertial SLAM

Integration of visual and inertial components in Visual-Inertial SLAM (VISLAM) enhances positioning accuracy and stability, addressing positioning ambiguity in visual data calculation with a frequency typically under 10 Hz [40]. Synchronisation of camera and IMU data timestamps is crucial for accurate data fusion. Integration levels vary (Figure 4): tightly-coupled systems [41-45], semi-tightly coupled systems [46], and loosely-coupled VISLAM [47-49].

In VISLAM, tightly-coupled methods integrate VO and IMU data equally, leveraging IMU predictions to enhance motion estimation speed and pose accuracy. Compared to loosely-coupled approaches, tightly-coupled methods utilise IMU data more effectively for pose evaluation. The tightly-coupled framework, depicted in Figure 5, optimally fuses visual and IMU cues to ensure system accuracy.

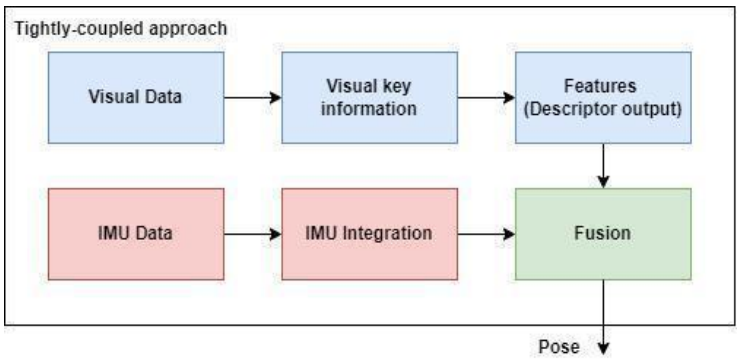

Fig. 5. Generalised block diagram of tightly-coupled approach.

In tightly-coupled VISLAM, two main approaches exist: filtering and optimization [43-45]. Filtering methods like EKF and MSCKF have shown promise, while optimization methods face challenges due to high-frequency IMU data. Combining these approaches poses pressing issues in the SLAM community [41, 42]. VISLAM holds potential for miniaturisation and cost reduction, particularly when paired with sparse direct methods, offering promising results on low-level hardware.

Loosely-coupled methods (Figure 6) treat IMU (red blocks) and visual data (blue blocks) as separate asynchronous modules, each updating independently and exchanging some redundant information. While IMU data compensates for drift and movement errors, visual data corrects cumulative IMU errors in Fusion (green block). However, visual positioning takes precedence, making this method less reliable in challenging visual environments and incapable of correcting visual measurement-induced drift.

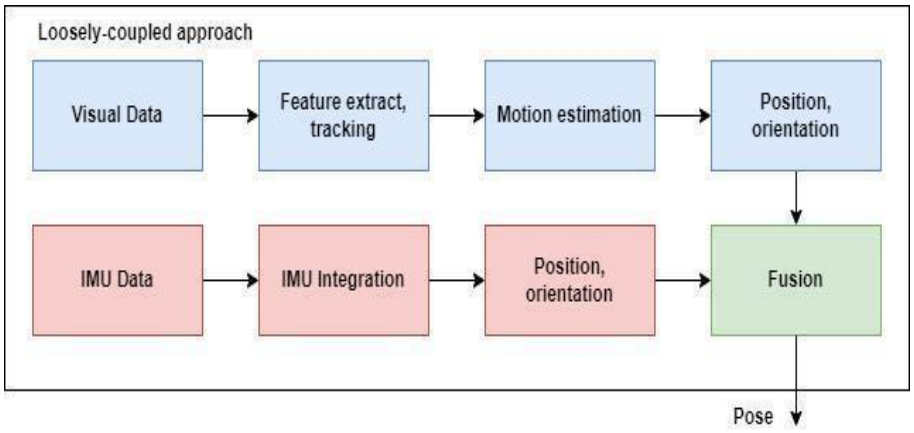

Fig. 6. Generalised block diagram of loosely-coupled approach.

Growing computing resources in embedded systems have diminished the popularity of the loosely-coupled method, as its main advantage lies in its lower computational requirements compared to the tightly-coupled approach. Tightly-coupled VISLAM methods typically don't face computational challenges on embedded systems. However, loosely-coupled methods like Empirical Mode Decomposition (EMD) or Clear Iterative EMD Interval Thresholding (EMD-CIIT) enhance stereo VO accuracy and robustness with IMU and GPS modules [48, 49]. Multi-Sensor-Fusion with Extended Kalman Filter (MSF-EKF) has gained practical popularity, as seen in [47], where MSF matches VO data with integrated

IMU and GPS data on drone controller boards. Loosely-coupled MSF offers an advantage when em-bedded directly into low-powered drone computers.

5 RGB-D SLAM

Recently, smaller and more affordable RGB-D (Red Green Blue-Depth) cameras using structured light have emerged, finding applications in SLAM algorithms [28, 50, 51]. These cameras provide real-time 3D information along with texture data, enabling direct acquisition of the environment's 3D structure in metric space, unlike monocular VSLAM algorithms.

The basic structure of RGB-D SLAM based on depth involves reconstructing the environment's 3D structure by combining multiple depth maps. The widely used Iterative Closest Point (ICP) algorithm estimates camera motion by iteratively searching for the closest points and optimising the transformation between point clouds. However, this method is sensitive to outliers and data insufficiency, often requiring manual adjustments despite attempts to mitigate issues using weight correction.

Depth-based RGB-D SLAM faces challenges in outdoor environments due to limitations of custom depth cameras designed primarily for indoor use. Issues include sunlight glare interfering with IR light projection for depth measurement and range limitations affecting depth accuracy at greater distances from the camera. Despite hardware limitations, stereo cameras without IR emitters, often equipped with integrated IMU devices, have gained popularity in the SLAM community [27, 52, 53].

6 Conclusion

Recent advancements in Visual-Only SLAM, Visual-Inertial SLAM, and other multi-sensor fusion techniques have shown promising results, driving significant interest from both researchers and industry practitioners. These developments, coupled with the growing market demand for new devices like event-based cameras, highlight the evolving landscape of SLAM technology. However, practical implementation still faces challenges, particularly in complex applications such as augmented reality (AR) and autonomous driving. While SLAM systems integrated with deep learning hold potential, their widespread adoption hinges on addressing existing performance limitations.

References

1. A. M. Barros, M. Michel, Y. Moline, G. Corre, F. Carrel, Robotics **11(24)**, 1-1 (2022)
2. X. Zhang, L. Wang, Y. Su, Pattern Recognition **113**, 107760 (2021)
3. S. Huang, J. Control Decis., **6**, 61 (2019)
4. Y. Alkendi, L. Seneviratne, Y. Zweiri, IEEE Access **9**, 76847 (2021)
5. P. K. Panigrahi, S. K. Bisoy, J. King Saud Univ. - Comput. Inf. Sci., 1 (2021)
6. S. A. S. Mohamed, M. H. Haghbayan, T. Westerlund, J. Heikkonen, H. Tenhunen, J. Plosila, IEEE Access, 1-1 (2019)
7. C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, J. D. Tardós, IEEE Trans. Robot **37**, 1874 (2021)
8. N. Dong, M. Qin, J. Chang, C. H. Wu, W. H. Ip, K. L. Yung, Comput. Commun. **186**, 153 (2022)
9. L. Ran, Y. Zhang, Q. Zhang, T. Yang, Sensors **17**, 1341 (2017)

10. Y. Fan, Q. Zhang, Y. Tang, S. Liu, H. Han, Pattern Recognition **121**, 108225 (2022)
11. A. R. Memon, H. Wang, A. Hussain, Robot. Auton. Syst. **126**, 103470 (2020)
12. G. Sonugür, Robot. Auton. Syst. **161**, 104342 (2023)
13. R. Alqobali, M. Alshmrani, R. Alnasser, A. Rashidi, T. Alhmiedat, O. M. Alia, Appl. Sci. **14**, 89 (2023)
14. I. A. Kazerouni, L. Fitzgerald, G. Dooly, D. Toal, Expert Syst. Appl. **205**, 117734 (2022)
15. J. Zhu, H. Li, T. Zhang, Tsinghua Sci. Technol. **29**, 415 (2023)
16. Q. Mo, Y. Zhou, X. Zhao, X. Quan, Y. Chen, "A survey on recent reflective detection methods in simultaneous localization and mapping for robot applications", in *2023 6th Int. Symp. Auton. Syst. (ISAS)* (IEEE, 2023)
17. J. Gaia, E. Orosco, F. Rossomando, C. Soria, IEEE Lat. Am. Trans. **21**, 1313 (2023)
18. K. Yousif, A. Bab-Hadiashar, R. Hoseinnezhad, Intell. Ind. Syst. **1**, 289 (2015)
19. M. Li, A.I. Mourikis, Int. J. Robot. Res. **32**, 690 (2013)
20. P. Geneva, K. Eckenhoff, G. Huang, "A linear-complexity EKF for visual-inertial navigation with loop closures", in *2019 Int. Conf. Robot. Autom. (ICRA)* (IEEE, 2019)
21. K. Himri, P. Ridao, N. Gracias, A. Palomer, N. Palomeras, R. Pi, IFAC-PapersOnLine **51**, 360 (2018)
22. B. Yang, X. Xu, J. Ren, L. Cheng, L. Guo, Z. Zhang, Pattern Recognit. Lett. **153**, 126 (2022)
23. Y. Tian, J. Miao, X. Wu, H. Yue, Z. Liu, W. Chen, Pattern Recognit. Lett. **153**, 75 (2022)
24. Y. B. Ai, T. Rui, X. Q. Yang, J. L. He, L. Fu, J. B. Li, M. Lu, Def. Technol. **17**, 1712 (2021)
25. L. Xiao, J. Wang, X. Qiu, Z. Rong, X. Zou, Robot. Auton. Syst. **117**, 1 (2019)
26. J. Cheng, L. Zhang, Q. Chen, X. Hu, J. Cai, Eng. Appl. Artif. Intell. **114**, 104992 (2022)
27. X. Zhao, L. Liu, R. Zheng, W. Ye, Y. Liu, Robot. Auton. Syst. **132**, 103597 (2020)
28. O. Guclu, A. B. Can, Comput. Vis. Image Underst. **184**, 31 (2019)
29. A. R. Vidal, H. Rebecq, T. Horstschaefer, D. Scaramuzza, IEEE Robot. Autom. Lett. **3**, 994 (2018)
30. M. Ferrera, A. Eudes, J. Moras, M. Sanfourche, G. Le Besnerais, IEEE Robot. Autom. Lett. **6**, 1399 (2021)
31. P. Y. Tseng, J. J. Lin, Y. C. Chan, A. Y. Chen, Autom. Constr. **140**, 104319 (2022)
32. J. Engel, T. Schöps, D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM", in *Eur. Conf. Comput. Vis.* (Springer, 2014)
33. J. Engel, V. Koltun, D. Cremers, IEEE Trans. Pattern Anal. Mach. Intell. **40**, 611 (2017)
34. X. Gao, R. Wang, N. Demmel, D. Cremers, "LDSO: Direct Sparse Odometry with Loop Closure", in *2018 IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)* (2018)
35. S. P. Li, T. Zhang, X. Gao, D. Wang, Y. Xian, Robot. Auton. Syst. **112**, 201 (2019)
36. E. Rublee, V. Rabaud, K. Konolige, G. Bradski, "ORB: An efficient alternative to SIFT or SURF", in *2011 Int. Conf. Comput. Vis.* (IEEE, 2011)
37. R. Mur-Artal, J. M. M. Montiel, J. D. Tardos, IEEE Trans. Robot. **31**, 1147 (2015)
38. R. Mur-Artal, J. D. Tardós, IEEE Trans. Robot. **33**, 1255 (2017)
39. R. A. Newcombe, S. J. Lovegrove, A. J. Davison, "DTAM: Dense tracking and mapping in real-time", in *2011 Int. Conf. Comput. Vis.* (IEEE, 2011)

40. C. Chang, S. You, Appl. Res. Comput., 642–647 (2018)
41. Y. He, J. Zhao, Y. Guo, W. He, K. Yuan, Sensors **18**, 1159 (2018)
42. G. Cioffi, D. Scaramuzza, "Tightly-coupled fusion of global positional measurements in optimization-based visual-inertial odometry", in *2020 IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)* (IEEE, 2020)
43. M. Ramezani, D. Acharya, F. Gu, K. Khoshelham, ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci. **4** (2017)
44. C. Yuan, J. Lai, P. Lyu, P. Shi, W. Zhao, K. Huang, Micromachines **9**, 626 (2018)
45. K. Sun, K. Mohta, B. Pfrommer, M. Watterson, S. Liu, Y. Mulgaonkar, C. J. Taylor, V. Kumar, IEEE Robot. Autom. Lett. **3**, 965 (2018)
46. K. W. Chiang, G. J. Tsai, H. Chang, C. Joly, N. Ei-Sheimy, Inf. Fusion **50**, 181 (2019)
47. R. Bähnemann, M. Pantic, M. Popović, D. Schindler, M. Tranzatto, M. Kamel, M. Grimm, J. Widauer, R. Siegwart, J. Nieto, J. Field Robot. **36**, 78 (2019)
48. M. Nezhadshahbodaghi, M. R. Mosavi, Measurement **183**, 109895 (2021)
49. J. M. Falquez, M. Kasper, G. Sibley, "Inertial aided dense & semi-dense methods for robust direct visual odometry", in *2016 IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)* (IEEE, 2016)
50. Y. Zhang, T. Funkhouser, "Deep depth completion of a single RGB-D image", in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit* (2018)
51. X. Gao, T. Zhang, Robot. Auton. Syst. **72**, 1 (2015)
52. S. Huang, K. Yang, H. Xiao, P. Han, J. Qiu, L. Peng, D. Liu, K. Luo, J. Visual Commun. Image Represent. **82**, 103402 (2022)
53. N. Krombach, D. Droeschel, S. Houben, S. Behnke, Robot. Auton. Syst. **109**, 38 (2018)