Graph-based Visual SLAM: Research Advances and Application Analysis Postprint

Authors: Wang Lutao, Wu Linfeng

Date: 2018-11-29T00:00:00+00:00

Abstract

Simultaneous Localization and Mapping (SLAM) refers to the technology whereby a robot estimates its own motion state while concurrently constructing an environmental map, constituting the core foundation for achieving fully autonomous robot motion in unknown environments. To attain a more comprehensive grasp of SLAM technology, this work provides a detailed analysis of the fundamental nature of the visual SLAM problem and the complexity inherent in its solution, grounded upon a retrospective examination of the developmental trajectory of visual SLAM technology over the past three decades. Particular emphasis is devoted to presenting the latest research achievements in enhancing pose estimation accuracy, constructing globally consistent maps, and improving algorithmic solution efficiency, accompanied by an analysis and comparative evaluation of current representative algorithmic implementation schemes. In anticipation of future demands arising from large-scale environments and full-lifecycle applications, the deficiencies extant within existing algorithmic frameworks and the most recent research trends are systematically summarized. Finally, the correlation between deep learning technology and visual SLAM problem solving is investigated.

Full Text

Preamble

Vol. 37 No. 1
Application Research of Computers (ChinaXiv Cooperative Journal)
Accepted Paper

Application Analyses and Research Progress of Graph-Based Visual SLAM

Wang Lutao, Wu Linfeng
(School of Computer Science, Chengdu University of Information Technology, Chengdu 610225, China)

Abstract: Simultaneous localization and mapping (SLAM) involves the concurrent construction of an environment model and estimation of a robot' s motion within it, representing a core capability for fully autonomous robot operation in unknown environments. To provide a comprehensive understanding of SLAM technology, this survey first reviews the progress made by the visual SLAM community over the past three decades, then analyzes the fundamental nature of the visual SLAM problem and its computational complexity. We focus on recent achievements in improving pose estimation accuracy, building globally consistent maps, and enhancing algorithmic efficiency, while analyzing and comparing current representative algorithmic implementations. For future large-scale, full-lifecycle applications, we summarize existing limitations and emerging research trends. Finally, we explore the potential connections between deep learning techniques and visual SLAM problem solving.

Keywords: simultaneous localization and mapping; graph optimization; data association; sparsification; deep learning
Classification: TP391.41
doi: 10.19734/j.issn.1001-3695.2018.08.0509

0 Introduction

To achieve fully autonomous operation in unknown environments, mobile robots must solve two fundamental problems: environment perception and self-localization. Simultaneous Localization and Mapping (SLAM) integrates these two tasks, enabling robots to incrementally acquire feature information about unknown environments through onboard sensors during motion while simultaneously estimating their own trajectory accurately, even without prior environmental knowledge [1]. Compared to laser-based SLAM, visual SLAM systems offer lower hardware costs and can capture environmental texture and color information, thus holding broader application prospects.

The development of visual SLAM technology over the past three decades can be summarized in two stages. The first stage (1986-2004) employed Bayesian filtering techniques to solve the SLAM problem. Smith et al. [2] conducted pioneering work by formulating SLAM as a stochastic estimation problem, treating robot poses and landmarks as random variables following certain distributions. Using motion and observation data, they applied filtering theory to achieve system state prediction, measurement updates, and online map updates, demonstrating good real-time performance. Representative algorithms include Extended

Kalman Filters (EKF) [3], Extended Information Filters [4], UKF [5], Rao-Blackwellized Particle Filters [6], and FastSLam [7].

However, filtering methods assume Markov properties for state estimation, using only adjacent frame information to estimate robot states, making it difficult to handle data association between current and historical frames. Since SLAM motion and observation equations are nonlinear functions, filtering methods employ first-order Taylor approximations to compute posterior probabilities. When strong nonlinearities exist, significant linearization errors become inevitable. Additionally, uncertainties in system parameters and observations cause error accumulation, leading to map inconsistency. Algorithmically, filtering methods must store, maintain, and update both the mean and covariance of state variables, with storage requirements growing quadratically with the number of estimated variables. Consequently, filtering methods are only suitable for resource-constrained scenarios or cases with few variables. Paz et al. addressed EKF' s scale limitations using submap segmentation, but this remained effective only within 100-meter ranges [8].

The second stage (2004-2016) focused on SLAM theoretical analysis and implementation details, with numerous studies addressing observability, convergence, and consistency. Reducing computational demands while achieving real-time performance and expanding operational scale became critical [9]. Graph optimization techniques emerged as the mainstream approach, constructing nonlinear least-squares objective functions for observation equations and treating robot poses and landmarks as optimization variables. Using Newton' s method and Levenberg-Marquardt algorithms to iteratively estimate optimal solutions significantly improved local pose and landmark estimation accuracy. Meanwhile, the sparsity inherent in SLAM problems was recognized, making globally consistent solutions attainable. Current systems can operate in 50 km environments with 1% translation error and 0.003°/m rotation error [10].

After three decades of development, the visual SLAM theoretical framework has stabilized. Key issues such as localization accuracy, trajectory drift, and globally consistent map construction have been effectively resolved under constraints of computational resources, motion models, static environments, and performance requirements. However, effective solutions remain lacking for high-performance, low-failure-rate applications in full-lifecycle, large-scale complex environments, as well as for adaptive computational resource allocation and task-driven perceptual model construction. Liu et al. [11] compared recent monocular visual SLAM techniques, while Cadena [12] summarized state estimation challenges, map representation methods, research progress, and theoretical performance guarantees, discussing related frontier topics in detail. Cuillaume [13] analyzed problems and solutions for SLAM in autonomous driving applications.

This paper focuses on analyzing the theoretical foundations of visual SLAM technology and the problems and solutions for full-lifecycle, large-scale environmental operation. We first present the SLAM system architecture, analyze issues in graph optimization methods based on theoretical foundations, and discuss recent research advances. We then summarize current mainstream algorithms and their performance metrics. Finally, for future SLAM system requirements in full-lifecycle, open large-scale scenarios, we discuss existing problems and solutions, and provide an outlook on research trends.

1 SLAM Mathematical Model

SLAM technology can be described as the process where a mobile robot incrementally builds an environmental map using feature information obtained from onboard sensors in an unknown environment while simultaneously estimating its own pose accurately, as illustrated in [Figure 1: see original paper].

In [Figure 1: see original paper], the robot's pose at time k is denoted as x_k, and landmarks are denoted as y. Based on the pose x_k at time k and observation data $z_{k,j}$ generated by observing landmark y_j at time k, the SLAM problem can be modeled using a motion model and an observation model:

$$\begin{cases} x_k = f(x_{k-1}, u_k, w_k) \\ z_{k,j} = h(x_k, y_j, \varepsilon_{k,j}) \end{cases}$$

where u_k represents motion sensor input, w_k represents state noise, and $\varepsilon_{k,j}$ represents observation noise.

According to this mathematical model, a SLAM system can be divided into front-end and back-end components, as shown in [Figure 2: see original paper]. The front-end extracts environmental features from sensor data, establishes data association between observations and landmarks as well as poses, and provides reliable initial values for back-end nonlinear optimization. Data association is critical in front-end processing, as it must address feature association across consecutive measurement frames for motion tracking while also associating current measurements with historical measurements to fulfill localization and mapping functions. The back-end performs inference and optimization on poses and landmarks to achieve globally consistent map construction and improved localization accuracy.

[Figure 3: see original paper] compares trajectory estimates before and after back-end optimization processing using the MIT Killian Court dataset. [FIGURE:3(a)] shows the optical photograph of MIT Killian Court with robot trajectory annotation. Due to sensor measurement errors and accumulated pose estimation errors, the robot' s trajectory estimate deviates significantly from ground truth, while back-end optimization can correct these estimation errors, making the trajectory estimate more consistent with the actual path.

2.1 SLAM Problem Formulation

Accurately estimating state variables from noisy observation data is the fundamental problem in SLAM. Graph optimization techniques treat the robot trajectory and landmarks as estimation variables and estimate them given observation data $Z = \{z_{k,m}\}_{k=1,m=1}^{k,j}$, maximizing the posterior probability distribution $P(X|Z)$. According to Bayes' rule, solving for maximum posterior probability equals maximizing the likelihood $P(Z|X)$ and prior probability $P(X)$:

$$X^*_{MAP} = \arg\max_X P(X|Z) = \arg\max_X P(Z|X)P(X)$$

Without prior information, $P(X)$ is constant, simplifying X^*_{MAP} to Maximum Likelihood Estimation (MLE):

$$X^*_{MAP} = \arg\max_X P(Z|X)$$

Assuming observation data $z_{k,j}$ are independent, the maximum posterior probability estimate decomposes into the product of individual observation likelihoods:

$$X^*_{MAP} = \arg\max_X \prod_{k,m} P(z_{k,m}|X) = \arg\max_X \prod_{k,m} P(z_{k,m}|x_k, y_j)$$

Assuming sensors are affected by Gaussian white noise and observations follow a Gaussian distribution with covariance matrix $\Omega_{k,j}^{-1}$, the maximum measurement likelihood can be expressed as:

$$P(z_{k,j}|x_k, y_j) \propto \exp\left(-\frac{1}{2}\|h(x_k, y_j) - z_{k,j}\|^2_{\Omega_{k,j}}\right)$$

Substituting into the previous equation and noting that maximizing posterior probability equals minimizing the negative log-likelihood function, the maximum posterior probability estimate can be explicitly formulated as solving for the joint estimation of robot poses and landmarks that minimizes the sum of squared errors between estimates and observations:

$$X^* = \arg\min_X \frac{1}{2}\sum_{k,j} \|h(x_k, y_j) - z_{k,j}\|^2_{\Omega_{k,j}}$$

where $h(\cdot)$ is an abstract nonlinear function representing mathematical models for inertial sensors, encoders, GPS, cameras, etc. When noise does not follow a standard normal distribution, the metric error $\|h(x_k, y_j) - z_{k,j}\|^2_{\Omega_{k,j}}$ in the objective function can be replaced with other norms such as ℓ_1 to increase system robustness and reduce sensitivity to outliers. Alternatively, loss functions like Huber or Tukey can replace the ℓ_2 norm.

For convenience, let the n-dimensional error vector $e = [e_1, e_2, ..., e_m]^T$ and error weight matrix $\Omega = \text{diag}(\Omega_1, \Omega_2, ..., \Omega_m)$. The objective function becomes:

$$X^* = \arg\min_{X} e^T \Omega e$$

For this nonlinear least-squares optimization problem, iterative linearization is generally employed. Given an initial value x_0, the error function $e(x)$ is first-order Taylor expanded near x_0. Taking the derivative with respect to increment δx and setting it to zero yields the incremental equation:

$$(H^T H)\delta x = -H^T g$$

where $H = J(x)$ is the Jacobian matrix and $g = e(x)$. This gives the system information vector $b = H^T g$ and Hessian matrix $H = J(x)^T J(x)$. The increment can then be solved as $\delta x = -H^{-1} b$. Iteratively solving using the Gauss-Newton method until convergence yields the optimal parameter estimate. To avoid increased approximation errors from large δx, the Levenberg-Marquardt algorithm improves stability by adding a trust region to H with a damping factor λ:

$$(H + \lambda I)\delta x = -b$$

2.2 Algorithm Essence Analysis

Pose estimation in SLAM consists of translation and rotation components. The nonlinearity of rotation calculations fundamentally makes SLAM a nonlinear optimization problem. Additionally, factors such as non-Gaussian noise, huge variable dimensions, and violations of static scene assumptions make the SLAM objective function exceptionally complex. Pose and landmark estimates easily fall into local minima, causing robot localization drift and preventing globally consistent trajectory estimation and map construction, thus failing to meet navigation and environmental reconstruction requirements. [Figure 4: see original paper] shows simulation comparisons between global optimization and convergence to local minima for spherical and toroidal surfaces.

Literature [14] combines visual odometry with 3D laser odometry to obtain initial values for optimization estimation, achieving 0.68% translation accuracy and 0.0016°/m rotation error on the KITTI dataset. Through feature point screening and tracking, applying multi-frame image feature point absolute difference summation and normalized cross-correlation techniques suppresses outlier effects on optimization results, enabling monocular visual SLAM to achieve 1% translation accuracy and rotation error below 0.003°/m on KITTI [15-17], though this remains insufficient for large-scale applications like automotive autonomous navigation.

Convergence issues in iterative solving have prompted theoretical investigations into SLAM problems, driving algorithmic advances. Huang et al. were among the first to study the non-convexity of SLAM and discuss small-scale pose graph optimization [18-20]. Knuth et al. examined state estimation error growth without loop closure [21]. Carlone et al. have continuously studied iterative convergence, first providing convergence region estimates for Gauss-Newton methods and proving the uniqueness of rotation estimation in 2D scenes, then proposing that maximum likelihood estimation is unique based on strong duality in SLAM, with semidefinite programming (SDP) yielding globally optimal pose graph solutions [22-25]. Liu et al. employed convex relaxation to avoid local minima [26]. Recognizing that SLAM nonlinearity primarily stems from rotation estimation, Carlone et al. first estimated rotation and used the results to accelerate nonlinear iteration [27]. Recently, for developing effective system failure detection and recovery strategies in safety-critical scenarios, Carlone et al. exploited Lagrangian duality in SLAM to propose effective solutions for pose estimation evaluation and verification [28].

2.3 Solving the SLAM Problem

SLAM estimation variables include all robot poses and observed landmarks. For visual SLAM, the number of landmarks far exceeds the number of poses, with dimensions reaching thousands. However, due to limited sensor fields of view and independent, finite observable landmarks, SLAM exhibits excellent sparsity that can be intuitively represented using factor graphs, as shown in [Figure 5: see original paper].

A factor graph is a two-dimensional undirected graph estimating joint probability distributions through products of multiple factor probability distributions [29]. The SLAM factor graph consists of nodes and connecting edges: circular nodes represent optimization state variables corresponding to robot and system states at different times, while rectangular nodes represent landmark observations and control inputs. Edges describe adjacent pose transformation relationships or observation errors. Since the error term generated by observing landmark l_j from pose x_k at time k is independent of other poses and landmarks, the Jacobian matrix $J(x)$ has non-zero entries only for $e_{k,j}/\partial x_k$ and $e_{k,j}/\partial l_j$:

$$J(x) = \begin{bmatrix} \frac{\partial e_{1,1}}{\partial x_1} & \frac{\partial e_{1,1}}{\partial l_1} & & & \\ \frac{\partial e_{2,1}}{\partial x_2} & & \frac{\partial e_{2,1}}{\partial l_1} & & \\ \vdots & & & \ddots & \\ \frac{\partial e_{3,4}}{\partial x_3} & & & & \frac{\partial e_{3,4}}{\partial l_4} \end{bmatrix}$$

[Figure 6: see original paper] illustrates the sparsity patterns of the Jacobian and Hessian matrices for a SLAM problem with 4 poses and 4 landmarks. Due to the sparsity of $J(x)$ and the diagonal structure of the landmark matrix H_{22}, the pose increment equation can be easily solved. Since the number of poses n is typically much smaller than the number of landmarks p, the solution scale is greatly simplified relative to the original problem.

The landmark increment equation becomes:

$$\delta l = -H_{22}^{-1}(H_{12}^{T}\delta x + b_2)$$

Substituting into the pose increment equation yields:

$$(H_{11} - H_{12}H_{22}^{-1}H_{12}^{T})\delta x = -(b_1 - H_{12}H_{22}^{-1}b_2)$$

Leveraging Hessian matrix sparsity through Schur complement elimination and linear factorization significantly improves optimization efficiency. Numerous factor graph solving frameworks have emerged, including g2o [30], TORO [31], HOG-Man [32], and COP-SLAM [33], enabling real-time solving of thousands of variables on standard PC processors and greatly advancing SLAM technology from research to application.

3 Graph Optimization SLAM Algorithm Implementation Framework

Graph optimization provides theoretical guarantees for SLAM system performance, but implementation also depends on critical technologies such as landmark perception, feature data association, and map representation, closely linked to engineering issues like program design. This section analyzes representative graph optimization-based SLAM implementations and their performance metrics.

1) PTAM

PTAM represents a revolutionary milestone in visual SLAM development, establishing the basic implementation framework with two major innovations: (1) it was the first to employ nonlinear optimization instead of traditional filtering for the SLAM back-end, and (2) it introduced the keyframe mechanism, enabling map optimization integration into real-time computation. The implementation uses two independent threads for camera pose tracking and map construction. The tracking thread responds to image data in real-time, while the mapping thread focuses on map establishment, maintenance, and updates. By only maintaining video keyframes and stably observed landmarks, the objective function can be solved efficiently. However, slow map construction or optimization may cause tracking loss. Pire et al. improved PTAM using stereo cameras, applying stereo constraints to landmark initialization, tracking, and mapping, adding real-time loop detection and correction modules, and employing local parallel bundle adjustment (BA) to optimize local maps, achieving real-time solutions with improved pose estimation accuracy [35].

2) ORB-SLAM

ORB-SLAM inherits PTAM’ s back-end nonlinear optimization scheme and keyframe processing mechanism, supporting monocular, stereo, and RGB-D video input modes, making it one of the most complete and usable modern SLAM systems [36,37]. Using ORB features for target matching and tracking, it achieves real-time computation on standard CPUs while maintaining good rotation and scale invariance. Real-time ORB descriptor extraction and offline ORB dictionary construction enable loop closure detection and relocalization during large-scale motion. ORB-SLAM employs a three-thread implementation: a real-time feature tracking thread coarsely computes landmark positions and camera poses via keyframe matching; a local BA thread maintains a covisibility graph to solve for refined landmark positions and camera poses; and a global loop detection and optimization thread eliminates accumulated estimation errors to obtain globally consistent trajectory estimates. Compared to ORB-SLAM, ORB-SLAM2 adds map reuse functionality to address localization drift when mapping fails, improving both accuracy and robustness in large-scale operations [38].

3) LSD-SLAM

LSD-SLAM employs direct methods, estimating camera motion and building semi-dense maps from pixel gradient information, avoiding keypoint extraction and descriptor computation [39]. It uses SSD metrics from five equidistant points on epipolar lines to ensure tracking stability. Depth estimation initializes with random values and performs mean normalization after estimation. Depth uncertainty measurement incorporates geometric relationships and epipolar-depth angle relationships into photometric uncertainty. Back-end optimization considers different scene scales to reduce scale drift. Direct methods are insensitive to featureless regions but highly sensitive to camera intrinsics and exposure, easily losing track during rapid camera motion. Additionally, loop detection still relies on feature point computation. S-LSD-SLAM uses stereo cameras to estimate scene depth, addressing monocular scale drift, and employs affine illumination correction to maintain photometric residual invariance under strong lighting differences between adjacent keyframes [40]. [Figure 7: see original paper] compares feature-based maps with semi-dense reconstruction, showing that semi-dense maps model gradient differences in grayscale images to display object edges and surface textures, containing more information than sparse maps.

4) SVO

SVO combines direct and feature-based methods for landmark computation and target tracking. Using 4$×$4 image patches around feature points to estimate camera pose and landmark positions, SVO neither extracts descriptors nor processes dense or semi-dense information, reducing CPU requirements and enabling real-time applications on drones, handheld AR/VR devices [41,42]. SVO’s significant contribution is introducing depth filtering for landmark position estimation. However, without back-end optimization or loop closure, SVO suffers from accumulated pose estimation errors and cannot relocalize after tracking loss.

5) SOFT

SOFT uses stereo cameras to reduce localization drift through careful selection of stably tracked features [43]. It extracts Blob and corner features within a small window in the current frame, determines correspondence through non-maximum suppression, and removes outliers using normalized cross-correlation. Motion estimation is separated into rotation and translation components: rotation uses 5-point methods with RANSAC [44], while translation uses minimum reprojection error to reduce matching errors. IMU information can further suppress outliers and optimize rotation estimation, enabling high-precision real-time pose estimation on ARM platforms.

Performance comparisons of these five algorithms in terms of pose estimation accuracy and real-time capability are shown in .

** Performance comparison of SLAM algorithms**

Algorithm	Translation Error	Rotation Error (deg/m)	Runtime per Frame (s)	Platform
S-PTAM	1.35%	0.0030	0.035	4 cores, 2.2GHz, C/C++
ORB-SLAM2	1.15%	0.0026	0.035	2 cores, 3.5GHz, C/C++
S-LSD-SLAM	1.20%	0.0027	0.045	1 core, 3.5GHz, C/C++
SVO	0.94%	0.0021	0.010	1 core, 2.5GHz, C/C++
SOFT	0.88%	0.0019	0.025	2 cores, 2.5GHz, C/C++

4 Applications and Outlook

1) Algorithm Robustness

Algorithm robustness is the primary challenge for visual SLAM systems operating over full lifecycles. Design limitations and hardware-related issues, such as sensor failures and driver errors, restrict current SLAM systems to achieving specific performance only in particular environments and on specific hardware platforms [45-48].

Among factors affecting stable SLAM operation, data association—determining correspondences between sensor measurements, between measurements and map features, or between map features—directly determines solution accuracy and real-time performance. Algorithmic failure from incorrect data association can be addressed from both front-end and back-end perspectives. In the front-end, when sensor sampling rates far exceed robot motion changes, both descriptor-based methods and optical flow [49] can effectively track observed landmarks across keyframes. To reduce accumulated errors, current observations must be associated with historical data in real-time for loop detection.

The bag-of-words method [50] is currently among the most effective loop detection techniques, significantly enhancing estimation consistency. It clusters local features extracted from images into discrete "words," then describes scenes using word histograms. Hierarchical vocabulary trees improve feature retrieval efficiency in large-scale datasets for real-time loop detection [51,52]. However, accurate visual word matching remains challenging under strong lighting variations [53]. Recent research focuses on incorporating different visual appearances into unified frameworks [54] or fusing landmark appearance with spatial relationships [55] to reduce lighting impact and enhance loop detection robustness. Literature [56] provides a detailed survey of visual place recognition methods.

Despite front-end advances, the impact of visual aliasing-induced loop detection errors on back-end estimation remains unavoidable, making it crucial to enhance back-end resistance to false data association. Common solutions include: (1) using prior knowledge to detect false loop closures and suppress outlier effects before optimization [57-61]; and (2) verifying loop detection correctness based on optimization residual errors.

The essence of graph optimization-based SLAM is solving nonlinear, non-convex optimization problems. Solvers are highly sensitive to initial values and prone to local minima, causing significant estimation bias. Ideal SLAM systems should evaluate estimation results in real-time and self-recover from failure states. Tight integration of front-end and back-end components offers an effective path to improved robustness, though related research remains limited. Additionally, using vision, IMU, GNSS, and 3D laser detectors to build customized maps and achieve relocalization in existing maps enhances SLAM applications in specific scenarios like autonomous driving [62].

2) Scalability

Scalability is another critical challenge for SLAM systems in long-term, large-scale applications such as autonomous driving, underwater exploration, and precision agriculture. As operation time and exploration range increase, the pose graph grows without bound, while iterative solving storage requirements scale linearly with variables, making the system unsustainable [63]. Reducing graph optimization complexity to maintain constant computational and storage demands primarily relies on sparsification methods and submap techniques.

Based on Markov blanket theory, factor graph sparsification can be achieved through node and edge marginalization. Ila et al. use information-theoretic methods to control node and edge addition, only incorporating non-redundant nodes and information-rich edges [64]. Johannsson et al. introduce new constraints to existing nodes to minimize new node addition, making factor graph expansion dependent on explored area rather than operation time [65]. Kretzschmar et al. studied information-theoretic marginalization criteria for nodes and edges in pose graph optimization [66]. Carlevaris and Mazuran introduced generic linear constraint factors and corresponding nonlinear graph sparsification methods [67,68]. Another sparsification approach reduces parameters by estimating continuous-time trajectories using cubic splines or B-splines in sliding window or batch modes [69-71]. Chi et al. replaced basic spline representations with Gaussian processes, where nodes in the sparsified factor graph are actual robot poses and other poses are obtained through interpolation of posterior means at given times [72].

Submap techniques decompose factor graphs into multiple submaps, using distributed computing across multiple processors for local factor graph optimization to achieve global optimization [73]. Grisetti hierarchically organizes submaps, updating only high-level and affected low-level regions with new observations [75]. Decomposing large-scale scenes into smaller regions for multi-robot mapping provides another submap processing approach [76,77]. Multi-robot mapping can be centralized or distributed: centralized processing fuses submap information at a central unit, while distributed processing maintains mapping consistency through inter-robot communication. Literature [80] provides detailed analysis of multi-robot mapping techniques.

Current scalability research focuses on simplifying factor graph optimization complexity, while many other large-scale issues require further investigation, including semantic map construction, distributed mapping robustness, and environment perception through mapping for human-like intelligence.

3) Deep Learning

Deep learning has revolutionized computer vision, and its application to SLAM problems has begun. Costante et al. use representation learning to replace geometric constraints in visual odometry, estimating robot pose from adjacent frames [78]. Eigen et al. improve data association accuracy by assigning semantic information to landmarks and jointly utilizing positional and semantic information to enhance pose and landmark estimation [79]. Liu et al. use deep networks for scene depth estimation from single images [80,81]. Semantic information also benefits loop detection and map representation [82,83].

Through complex CNN architectures, deep learning enables multi-level feature extraction for deep environmental perception, whereas traditional computer vision remains at shallow pixel or feature-level extraction. As a perception tool, deep learning can solve problems intractable for traditional methods. However, for SLAM, perception serves localization and mapping, and whether end-to-end SLAM systems using deep learning are feasible remains exploratory. Additionally, leveraging scene priors can significantly improve performance, but research on how deep network output uncertainty affects SLAM geometric processing is needed.

Future SLAM systems operating in open environments require continuous exploration and full-lifecycle learning capabilities. Deep learning success depends on training with large datasets of fixed target classes, so improving online learning and adaptability is essential for human-like intelligent SLAM systems. Developing lightweight networks for embedded SLAM applications also presents challenges for deep learning integration.

5 Conclusion

This paper reviews three decades of SLAM technology development, analyzing graph optimization-based visual SLAM theoretical models and algorithmic implementations, discussing existing problems and presenting latest theoretical results. While SLAM theory has matured and algorithmic frameworks for large-scale localization and mapping have emerged, challenges remain in robustness and scalability design for full-lifecycle, large-scale environmental applications. Building large-scale, fully autonomous SLAM systems for complex scenarios like autonomous driving, marine surveying, and precision agriculture represents the future goal. Combining traditional SLAM with deep learning for semantic-level environmental perception while fusing GNSS, inertial measurement, LiDAR, and other sensing technologies to enhance robustness in complex environments provides an effective path forward.

References

[1] Durrant W H, Bailey T. Simultaneous localization and mapping: Part I [J]. IEEE Robotics Automat Mag, 2006, 13 (3): 108-117.
[2] Smith R C, Cheeseman P. On the estimation and representations of spatial uncertainty [J]. International Journal of Robotics Research, 1986, 5 (12): 56-68.
[3] 何俊学, 李战明. 基于视觉的同时定位与地图构建方法综述 [J]. 计算机应用研究, 2010, 27 (8): 2839-2843. (He Junxue, Li Zhanming. Survey of vision-based approach to simultaneous localization and mapping [J]. Application Research of Computers, 2010, 27 (8): 2839-2843.)
[4] Thrun S, Liu Yufeng, Koller D, et al. Simultaneous localization and mapping with sparse extended information filters [J]. International Journal of Robotics Research, 2003, 23 (8): 693-716.
[5] Holmes S A, Klein G, Murray D W. An O (N^2) square root unscented kalman filter for visual simultaneous localization and mapping [J]. IEEE Trans on Pattern Analysis & Machine Intelligence, 2009, 31 (7): 1251-1263.
[6] Doucet A, Freitas N D, Murphy K, et al. Rao-blackwellised particle filtering for dynamic Bayesian networks [C]// Proc of the 16th Conference on

Uncertainty in Artificial Intelligence. 2000: 176-183.
[7] Montemerlo M, Thrun S, koller D, et al. Fast SLAM: a factored solution to the simultaneous localization and mapping problem [C]// Proc of AAAI National Conference on Artificial Intelligence. 2002: 593-598.
[8] Paz L M, Jensfelt P, Tardos J D, et al. EKF SLAM updates in O (n) with divide and conquer SLAM [C]// Proc of IEEE International Conference on Robotics and Automation. 2007: 1657-1663.
[9] Gamini D, Huang Shoudong, Wang Zhan, et al. A review of recent developments in simultaneous localization and mapping [C]// Proc of IEEE International Conference on Industrial and Information Systems. 2011: 477-482.
[10] Persson M, Piccini T, Felsberg M, et al. Robust stereo visual odometry from monocular techniques [C]// Proc of Intelligent Vehicles Symposium. 2015: 526-531.
[11] Liu Haomin, Zhang Guofeng, Bao Hujun. A survey of monocular simultaneous localization and mapping [J]. Journal of Computer-Aided Design & Computer Graphics, 2016, 28 (6): 854-871.
[12] Cadena C, Carlone L, Carrillo H, et al. Past, present, and future of simultaneous localization and mapping: Toward the Robust-Perception Age [J]. IEEE Trans on Robotics, 2016, 32 (6): 1309-1332.
[13] Cuillaume B, Zayed A, Li Yu, et al. Simultaneous localization and mapping: a survey of current trends in autonomous driving [J]. IEEE Trans on Intelligent Vehicles, 2017, 2 (3): 194-230.
[14] Zhang Ji, Singh S. Visual-lidar odometry and mapping: low-drift, robust, and fast [C]// Proc of IEEE International Conference on Robotics and Automation. 2015: 2174-2181.
[15] Buczko M, Willert V. How to distinguish inliers from outliers in visual odometry for high-speed automotive applications [C]// Proc of Intelligent Vehicles Symposium. 2016: 478-483.
[16] Persson M, Piccini T, Felsberg M, et al. Robust stereo visual odometry from monocular techniques [C]// Proc of Intelligent Vehicles Symposium. 2015: 526-531.
[17] Cvišić I, Petrović I. Stereo odometry based on careful feature selection and tracking [C]// Proc of European Conference on Mobile Robots. 2015: 1-6.
[18] Huang Shoudong, Lai Yu, Frese U, et al. How far is SLAM from a linear least squares problem? [C]// Proc of IEEE International Conference on Intelligent Robots and Systems. 2010: 3011-3016.
[19] Huang Shoudong, Wang Heng, Frese U, et al. On the number of local minima to the point feature based SLAM problem [C]// Proc of IEEE International Conference on Robotics and Automation. 2012: 2074-2079.
[20] Huang Shoudong, Dissanayake G. A critique of current developments in simultaneous localization and mapping [J]. International Journal of Robotics Research, 2016, 13 (5): 1-13.
[21] Knuth J, Barooah P. Error growth in position estimation from noisy relative pose measurements [J]. Robotics & Autonomous Systems, 2013, 61 (3): 229-244.
[22] 梁明杰, 闵华清, 罗荣华. 基于图优化的同时定位与地图创建综述 [J]. 机器人, 2013, 35 (4):

500-512. (Liang Mingjie, Min Huaqing, Luo Ronghua. Graph-based SLAM: A Survey [J]. Robot, 2013, 35 (4): 500-512.)

[23] Carlone L, Censi A. From angular manifolds to the integer lattice: guaranteed orientation estimation with application to pose graph optimization [J]. IEEE Trans on Robotics, 2014, 30 (2): 475-492.

[24] Carlone L, Rosen D M, Calafiore G, et al. Lagrangian duality in 3D SLAM: Verification techniques and optimal solutions [C]// Proc of IEEE International Conference on Intelligent Robots and Systems. 2015: 125-132.

[25] Carlone L, Calafiore G C, Tommolillo C, et al. Planar pose graph optimization: duality, optimal solutions, and verification [J]. IEEE Trans on Robotics, 2016, 32 (3): 545-565.

[26] Liu Mingjie, Huang Shoudong, Dissanayake G, et al. A convex optimization based approach for pose SLAM problems [C]// Proc of IEEE International Conference on Intelligent Robots and Systems. 2012: 1898-1903.

[27] Carlone L, Aragues R, Castellanos J A, et al. A fast and accurate approximation for planar pose graph optimization [J]. International Journal of Robotics Research, 2014, 33 (7): 965-987.

[28] Carlone L, Tron R, Daniilidis K, et al. Initialization techniques for 3D SLAM: A survey on rotation estimation and its use in pose graph optimization [C]// Proc of IEEE International Conference on Robotics and Automation. 2015: 4597-4604.

[29] Kschischang F, Frey B, Loeliger H. A. Factor graphs and the sum-product algorithm [J]. IEEE Trans on Information Theory, 2001, 47 (2): 498-519.

[30] Kümmerle R, Grisetti G, Strasdat H, et al. G2o: A general framework for graph optimization [C]// Proc of IEEE International Conference on Robotics and Automation. 2011: 3607-3613.

[31] Grisetti G, Grzonka S, Stachniss C, et al. Efficient estimation of accurate maximum likelihood maps in 3D [C]// Proc of IEEE International Conference on Intelligent Robots and Systems. 2007: 3472-3478.

[32] Grisetti G, Kummerle R, Stachniss C, et al. Hierarchical optimization on manifolds for online 2D and 3D mapping [C]// Proc of IEEE International Conference on Robotics and Automation. 2010: 273-278.

[33] Dubbelman G, Browning B. Closed-form Online Pose-chain SLAM [C]// Proc of IEEE International Conference on Robotics and Automation. 2013: 1-8.

[34] Klein G, Murray D. Parallel Tracking and Mapping for Small AR Workspaces [C]// Proc of IEEE and ACM International Symposium on Mixed and Augmented Reality. 2007: 1-10.

[35] Pire T, Fischer T, Civera J, et al. Stereo parallel tracking and mapping for robot localization [C]// Proc of IEEE International Conference on Intelligent Robots and Systems. 2015: 1373-1378.

[36] Mur-Artal R, Montiel J M, Tardós J D. ORB-SLAM: a versatile and accurate monocular SLAM system [J]. IEEE Trans on Robotics, 2015, 31 (5): 1147-1163.

[37] Mur-Artal R, Tardós J D. Fast relocalization and loop closing in keyframe-based SLAM [C]// Proc of IEEE International Conference on Robotics and

Automation. IEEE Computer Society Press, 2014: 846-853.
[38] Mur-Artal R, Tardós J D. ORB-SLAM2: an open-source slam system for monocular, stereo, and RGB-D cameras [J]. IEEE Trans on Robotics, 2016, 33 (5): 1255-1262.
[39] Engel J, Schöps T, Cremers D. LSD-SLAM: Large-Scale direct monocular SLAM [C]// Proc of the 13th European Conference on Computer Vision. 2014: 834-849.
[40] Engel J, Stückler J, Cremers D. Large-scale direct SLAM with stereo cameras [C]// Proc of IEEE International Conference on Intelligent Robots and Systems. 2015: 1935-1942.
[41] Forster C, Pizzoli M, Scaramuzza D. SVO: fast semi-direct monocular visual odometry [C]// Proc of IEEE International Conference on Robotics and Automation. 2014: 15-22.
[42] Cvišić I, Petrović I. Stereo odometry based on careful feature selection and tracking [C]// Proc of European Conference on Mobile Robots. 2015: 1-6.
[43] Fischler M A, Bolles R C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography [J]. Readings in Computer Vision, 1987, 24 (6): 726-740.
[44] 杨海燕, 罗文超, 刘国栋. 基于 SURF 算法和 SC-RANSAC 算法的图像配准 [J]. 计算机应用研究, 30 (5): 1586-1588. (Yang Haiyan, Luo Wenchao, Liu Guodong. Image registration based on SURF algorithm and SC-RANSAC algorithm [J]. Application Research of Computers, 30 (5): 1586-1588.)
[45] Frese U. Interview: is SLAM solved? [J]. Künstliche Intelligenz, 2010, 24 (3): 255-257.
[46] Ackeerman E. Dyson's robot vacuum has 360-degree camera, tank treads, cyclone suction [EB/OL]. (2014-09-04). https://spectrum.ieee.org/automaton/robotics/home-robots/dyson-the-360-eye-robot-vacuum.
[47] Jonathan H. New experimental features for Daydream. [EB/OL]. (2018-09-04). https://developers.googleblog.com/2018/09/new-experimental-features-for-daydream.html.
[48] Kuka. KUKA automates laser system for automotive supplier Proseat [EB/OL]. (2017-09). https://www.kuka.com/en-de/industries/solutions-database/2017/09/solution-systems-proseat.
[49] Scaramuzza D, Fraundorfer F. Visual Odometry: Part I: The first 30 years and fundamentals [J]. IEEE Robotics & Automation Magazine, 2011, 18 (4): 80-92.
[50] Galvez-López D, Tardos J D. Bags of binary words for fast place recognition in image sequences [J]. IEEE Trans on Robotics, 2012, 28 (5): 1188-1197.
[51] Nister D, Stewenius H. Scalable recognition with a vocabulary tree [C]// Proc of IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 2006: 2161-2168.
[52] Curless B, Levoy M. A volumetric method for building complex models from range images [C]// Proc of Conference on Computer Graphics and Interactive Techniques. New York: ACM Press, 1996: 303-312.
[53] 曾文静, 张铁栋, 姜大鹏. SLAM 数据关联方法的比较分析 [J]. 系统工程与电子技术, 2010, 32 (4): 860-864. (Zeng Wenjing, Zhang Tiedong, Jiang Daou. Analysis

of data association methods of SLAM [J]. Systems Engineering and Electronics, 2010, 32 (4): 860-864.)

[54] Churchill W, Newman P. Experience-based navigation for long-term localisation [J]. International Journal of Robotics Research, 2013, 32 (14): 1725-1749.

[55] Ho K L, Newman P. Loop closure detection in SLAM by combining visual and spatial appearance [J]. Robotics & Autonomous Systems, 2006, 54 (9): 740-749.

[56] Lowry S, Sünderhauf N, Newman P, et al. Visual place recognition: A Survey [J]. IEEE Trans on Robotics, 2016, 32 (1): 1-19.

[57] Sabatta D, Scaramuzza D, Siegwart R. Improved appearance-based matching in similar and dynamic environments using a Vocabulary tree [J]. 2010, 58 (8): 1008-1013.

[58] Sunderhauf N, Protzel P. Towards a robust back-end for pose graph SLAM [C]// Proc of IEEE International Conference on Robotics and Automation. 2012: 1254-1261.

[59] Latif Y, Cadena C, Neira J. Robust loop closing over time for pose graph SLAM [J]. International Journal of Robotics Research, 2013, 32 (14): 1611-1625.

[60] Olson E, Agarwal P. Inference on networks of mixtures for robust robot mapping [J]. International Journal of Robotics Research, 2013, 32 (7): 826-840.

[61] Carlone L, Censi A, Dellaert F. Selecting good measurements via 1 relaxation: A convex approach for robust estimation over graphs [C]// Proc of IEEE International Conference on Intelligent Robots and Systems. 2014: 2667-2674.

[62] Wolcott R W, Eustice R M. Visual localization within LIDAR maps for automated urban driving [C]// Proc of IEEE International Conference on Intelligent Robots and Systems. 2014: 176-183.

[63] Dellaert F, Carlson J, Ila V, et al. Subgraph-preconditioned conjugate gradients for large scale SLAM [C]// Proc of IEEE International Conference on Intelligent Robots and Systems. 2010: 2566-2571.

[64] Ila V, Porta J M, Andrade-Cetto J. Information-based compact pose SLAM [J]. IEEE Trans on Robotics, 2010, 26 (1): 78-93.

[65] Johannsson H, Kaess M, Fallon M, et al. Temporally scalable visual SLAM using a reduced pose graph [C]// Proc of IEEE International Conference on Robotics and Automation. 2012: 54-61.

[66] Kretzschmar H, Stachniss C, Grisetti G. Efficient information-theoretic graph pruning for graph-based SLAM with laser range finders [C]// Proc of IEEE International Conference on Intelligent Robots and Systems. 2011: 316-321.

[67] Carlevaris-Bianco N, Eustice R M. Generic factor-based node marginalization and edge sparsification for pose-graph SLAM [C]// Proc of IEEE International Conference on Robotics and Automation. 2013: 5748-5755.

[68] Mazuran M, Burgard W, Tipaldi G D. Nonlinear factor recovery for long-term SLAM [J]. International Journal of Robotics Research, 2016, 35 (1): 50-72.

[69] Bibby C, Reid I. A hybrid SLAM representation for dynamic marine environments [C]// Proc of IEEE International Conference on Robotics and Automation. 2010: 257-264.
[70] Furgale P, Barfoot T D, Sibley G. Continuous-time batch estimation using temporal basis functions [C]// Proc of IEEE International Conference on Robotics and Automation. 2012: 2088-2095.
[71] Raguram R, Chum O, Pollefeys M, et al. USAC: a universal framework for random sample consensus [J]. IEEE Trans on Pattern Analysis & Machine Intelligence, 2013, 35 (8): 2022-2038.
[72] Tong Chihay, Furgale P, Barfoot T D. Gaussian process gauss-newton for non-parametric simultaneous localization and mapping [J]. International Journal of Robotics Research, 2013, 32 (5): 507-525.
[73] Bosse M, Newman P M, Leonard J J, et al. Simultaneous localization and map building in large-scale cyclic environments using the atlas framework [J]. International Journal of Robotics Research, 2004, 23 (23): 1113-1139.
[74] Ni Kai, Dellaert F. Multi-level submap based SLAM using nested dissection [C]// Proc of IEEE International Conference on Intelligent Robots and Systems. 2010: 2558-2565.
[75] Grisetti G, Kummerle R, Stachniss C, et al. Hierarchical optimization on manifolds for online 2D and 3D mapping [C]// Proc of IEEE International Conference on Robotics and Automation. 2010: 273-278.
[76] Dong Jing, Nelson E, Indelman V, et al. Distributed real-time cooperative localization and mapping using an uncertainty-aware expectation maximization approach [C]// Proc of IEEE International Conference on Robotics and Automation. 2015: 5807-5814.
[77] Riazuelo L, Civera J, Montiel J M M. C2TAM: a cloud framework for cooperative tracking and mapping [J]. Robotics & Autonomous Systems, 2014, 62 (4): 401-413.
[78] Costante G, Mancini M, Valigi P, et al. Exploring representation learning with cnns for frame-to-frame ego-motion estimation [J]. IEEE Robotics & Automation Letters, 2015, 1 (1): 18-25.
[79] Eigen D, Fergus R. Predicting Depth, Surface normals and semantic labels with a common multi-scale convolutional architecture [C]// Proc of IEEE International Conference on Computer Vision. 2016: 2650-2658.
[80] Liu Fayao, Shen Chunha, Lin Guosheng. Deep convolutional neural fields for depth estimation from a single image [C]// Proc of IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 2015: 07-15.
[81] Cadena C, Dick A, Reid I D. Multi-modal auto-encoders as joint estimators for robotics scene understanding [C]// Robotics: Science and Systems. 2016: 1-8.
[82] Bao S Y, Bagra M, Chao Yuwei, et al. Semantic structure from motion with points, regions, and objects [C]// Proc of IEEE Conference on Computer Vision and Pattern Recognition. 2012: 2703-2710.
[83] Dame A, Prisacariu V A, Ren Carl Y, et al. Dense reconstruction using 3D object shape priors [C]// Proc of IEEE Conference on Computer Vision and Pattern Recognition, 2013: 1288-1295.