A Survey on Direct Methods for Visual SLAM: Postprint

Authors: Pan Linhao, Tian Fuqing, Ying Wenjian, Qiu Qianjun

Date: 2018-05-02T00:00:00+00:00

Abstract

Visual Simultaneous Localization and Mapping (V-SLAM) has been extensively studied in robotics, UAV navigation, autonomous driving, and other fields. Direct method V-SLAM tracks camera pose and constructs environmental maps based on the photometric consistency assumption. This paper addresses direct method V-SLAM by first briefly introducing its fundamental principles; subsequently analyzing and comparing several representative direct method V-SLAM systems; and finally discussing the advantages, disadvantages, and development trends of the direct method, along with providing a summary and outlook.

Full Text

Preamble

Title: A Survey on Direct-Method Visual Simultaneous Localization and Mapping

Authors: Pan Linhao[1], Tian Fuqing[1], Ying Wenjian[1], Qiu Qianjun[2]
[1] College of Weaponry, Naval University of Engineering, Wuhan 430033, China
[2] Naval Representative Office in the Second Academy of China Aerospace Science and Industry Corporation, Beijing 100854, China

Abstract: Visual Simultaneous Localization and Mapping (V-SLAM) has been extensively studied in robotics, unmanned aerial vehicle navigation, autonomous driving, and other fields. Direct-method V-SLAM tracks camera pose and constructs environmental maps based on the assumption of photometric consistency. This paper first outlines the fundamental principles of direct-method V-SLAM, then analyzes and compares several representative direct-method V-SLAM systems, and finally discusses the advantages, disadvantages, and development trends of direct methods, concluding with a summary and outlook.

Keywords: computer vision; simultaneous localization and mapping; direct method; structure-from-motion; multi-view geometry

0 Introduction

Simultaneous Localization and Mapping (SLAM) refers to the technology whereby a sensor-equipped agent, without prior environmental information, tracks its own position and orientation (hereinafter referred to as "pose") during motion while simultaneously constructing a consistent map of the environment structure [1-2]. Visual Simultaneous Localization and Mapping (V-SLAM) primarily employs cameras as sensors [3]. In recent years, V-SLAM technology has found applications in indoor service robots, autonomous vehicles, UAV navigation and localization, and augmented reality devices [4-6]. For instance, smartphones utilize cameras and IMUs to localize devices in real-time for AR effects, while UAVs employ cameras to track their own pose and build environmental maps for autonomous flight.

Since its proposal in 1986, SLAM research has spanned over three decades. Originating in robotics, early SLAM literature referred to it as "estimation of spatial uncertainty" [7,8]. In initial studies, SLAM was treated as a state estimation problem, with filtering algorithms dominating the field. Researchers applied state estimation theory to represent uncertainties in sensor pose and environmental structure, then used filters to update the mean and variance of state estimates [9]. To address nonlinearities in state equations, methods such as Extended Kalman Filter (EKF), Particle Filter (PF), and Unscented Kalman Filter (UKF) were applied to SLAM, yielding representative works including EKF-SLAM, UKF-SLAM, and FastSLAM [10-12].

Entering the 21st century, cameras began to see widespread use in SLAM research. In 2003, Davison proposed MonoSLAM [13], the first real-time V-SLAM system based on monocular cameras and the EKF framework. During this period, researchers recognized numerous commonalities between Structure-from-Motion (SfM) in computer vision and SLAM [14]. The nonlinear optimization technique from SfM—Bundle Adjustment (BA) [15]—was introduced to SLAM research and became the dominant method in V-SLAM. PTAM (Parallel Tracking and Mapping) [16-17], based on keyframes and bundle adjustment, was the first V-SLAM system to use nonlinear optimization as its backend. It pioneered the parallel processing of camera pose tracking and mapping as two separate threads, reducing computational load for pose tracking while optimizing mapping accuracy, making it a milestone in V-SLAM. Subsequent V-SLAM systems, whether feature-based (e.g., ORB-SLAM [18-19]) or direct-method (e.g., LSD-SLAM [20]), have drawn inspiration from PTAM' s algorithmic framework.

After more than thirty years of research, V-SLAM technology has formed a stable framework with visual odometry as the frontend, nonlinear optimization and filtering as the backend, complemented by loop closure detection and map-

ping modules. Feature-based and direct methods represent two mainstream approaches in V-SLAM. Compared to feature-based methods, direct methods have a shorter research history. However, recent years have witnessed breakthroughs in direct-method SLAM research, achieving performance comparable to or even exceeding that of feature-based methods. This survey focuses on direct-method V-SLAM, systematically analyzing and comparing several representative direct-method V-SLAM systems, discussing their advantages, disadvantages, and development trends, and providing a summary and outlook.

1 V-SLAM Basic Principles

V-SLAM technology can track camera pose in real-time from image sequences and construct three-dimensional environmental maps. To track camera motion, camera pose is represented by transformation matrices $T \in SE(3)$ [14], while scene structure is represented by spatial points $X_i \in \mathbb{R}^3$. Through the observation equation

$$\hat{x}_{ij} = h(T_i, X_j) + n_{ij}$$

we obtain the observed value $\hat{x}_{ij}$ of spatial point X_j in camera image i, where n_{ij} represents observation noise. To determine spatial points and camera pose, maximum likelihood estimation yields the state estimate

$$\hat{x} = \arg\max_x p(x|z) = \arg\max_x \prod_{i,j} p(\hat{x}_{ij}|T_i, X_j)$$

Due to the presence of observation noise, the above problem can be transformed into solving the following objective function:

$$\hat{x} = \arg\min_x \sum_{i=1}^{m} \sum_{j=1}^{n} \|h(T_i, X_j) - \hat{x}_{ij}\|^2$$

As shown in Figure 1 [Figure 1: see original paper], feature-based and direct methods parameterize the residuals in Equation (3) differently. Feature-based methods extract feature points from images and match descriptors around feature points to obtain reprojection errors e_{21}^h between corresponding feature points x_1 and x_2. Direct methods, by contrast, avoid feature point and descriptor matching altogether. Instead, they use camera pose estimates as initial values to find pixel positions u_2 corresponding to pixel points u_1 based on pixel gradients, solving for optimal camera pose by optimizing photometric errors e_{21}^I [21].

2 Direct-Method V-SLAM Systems

Current direct-method V-SLAM systems primarily include SVO (Semi-direct Visual Odometry) [22], DSO (Direct Sparse Odometry) [23], LSD-SLAM (Large-Scale Direct SLAM) [20], and DTAM (Dense Tracking and Mapping) [24] for standard cameras, and DVO [25] for RGB-D cameras. This section analyzes several representative monocular direct-method V-SLAM systems and compares their advantages and disadvantages.

2.1 SVO Algorithm Analysis

SVO is a sparse direct-method system proposed by Forster et al. in 2014, which was extended in 2017 to support wide-angle cameras, multi-camera configurations, and IMU integration [26]. SVO' s most distinctive characteristic is its frontend, which tracks camera pose based on the photometric consistency assumption, while its backend still employs traditional reprojection error minimization for global map optimization. Consequently, it is also called a semi-direct method, with reconstruction results shown in Figure 2 [Figure 2: see original paper].

The monocular SVO algorithm divides into two threads: camera pose tracking and mapping. In the pose tracking thread, SVO approximates the current frame' s pose T_k as the previous frame' s pose T_{k-1}. It projects sparse feature points stored in frame $k-1$ into the current frame k and obtains the relative motion estimate $\delta T_{k,k-1}$ between the two frames by minimizing the photometric error of feature points:

$$\delta T_{k,k-1} = \arg\min_{\delta T} \sum_{i \in \mathcal{F}_{k-1}} \|I_k(u_i') - I_{k-1}(u_i)\|^2$$

where $I(\cdot)$ represents the photometric error. To eliminate accumulated errors in feature points and pose estimation, feature points stored in keyframes are projected into the current frame k, and the optimized projection positions of feature points in the current frame are obtained by minimizing projected patch photometric errors:

$$\delta u_i' = \arg\min_{\delta u} \|I_k(u_i' + \delta u) - A_i I_{k-1}(u_i)\|^2$$

Finally, by minimizing reprojection errors between feature point positions u_i' and predicted feature point positions, optimized camera poses and feature point spatial positions are obtained respectively.

For map construction, SVO' s backend mapping thread continuously recovers spatial positions of feature points in reference frames. When the number of effective feature points falls below a specified threshold, a new keyframe is selected as the reference frame r and feature points are extracted. In subsequent

observed frames, feature point positions are matched using epipolar search, and inverse depth observations ρ_i are obtained through triangulation [27]. In SVO, a "Gaussian plus uniform distribution" model represents the distribution of inverse depth observations [28,29]:

$$p(\rho_i|\hat{\rho}_i) = \gamma_i \mathcal{N}(\rho_i|\hat{\rho}_i, \sigma_i^2) + (1-\gamma_i)\mathcal{U}(\rho_i|\rho_{\min}, \rho_{\max})$$

where γ_i is the probability that feature point i is an inlier, σ_i^2 is the variance of inlier observations, $\hat{\rho}_i$ is the inverse depth estimate, and $[\rho_{\min}, \rho_{\max}]$ is the range of the outlier distribution. SVO fuses measurement information of feature points across frames, and when the variance falls below a set threshold, feature points are saved to the environmental structure map based on estimated depth values for use by the tracking thread.

2.2 LSD-SLAM Algorithm Analysis

Unlike SVO, which extracts sparse feature points, the monocular LSD-SLAM system utilizes regions with significant pixel gradients in images for pose tracking and depth reconstruction, enabling recovery of semi-dense three-dimensional scene maps, as shown in Figure 4 [Figure 4: see original paper]. The algorithm primarily consists of three threads: camera pose tracking, map depth estimation, and global map optimization.

In the pose tracking thread, the current frame k uses the previous frame' s pose as an initial estimate. Based on the photometric consistency assumption, it compares with the nearest keyframe r to obtain the relative pose of the current frame:

$$\xi_{kr} = \arg\min_{\xi} \sum_{i \in \Omega_r} \delta(I_k, u_i, I_r, u_i')$$

where vector $\xi \in \mathfrak{se}(3)$ is the Lie algebra representation of transformation matrix T [36], and $\delta(\cdot)$ represents photometric error.

In LSD-SLAM' s map depth estimation thread, the system determines whether to create a new keyframe from the current frame based on the distance $dist(\xi_{tk})$ it has moved relative to keyframe t:

$$dist(\xi_{tk}) = \xi_{tk}^T W \xi_{tk}$$

where W is a weighting matrix. If the displacement exceeds a set threshold, the current frame becomes a new keyframe, obtaining a semi-dense pixel set projected from the previous keyframe. If the displacement does not exceed the threshold, the current frame selects an image from the frame sequence as a reference frame for epipolar search, obtaining depth observations d_i and variances

σ_i^2 for significant pixels in the current frame. It then uses EKF to update depth estimates and variances of pixels in keyframe t:

$$\begin{cases} d_i' \leftarrow \frac{\sigma_i^2 d_t + \sigma_t^2 d_i}{\sigma_i^2 + \sigma_t^2} \\ \sigma_i^{2'} \leftarrow \frac{\sigma_i^2 \sigma_t^2}{\sigma_i^2 + \sigma_t^2} \end{cases}$$

Because monocular cameras cannot compute absolute depth for pixels, scale drift occurs. Therefore, LSD-SLAM optimizes the global map created by the first two threads. For saved keyframes, LSD-SLAM controls global map scale by constraining the mean of all pixel inverse depths to be 1, enabling adjacent keyframes to represent their relative pose through similarity transformations with scale changes [36]. LSD-SLAM minimizes errors by leveraging the intrinsic relationship between scene depth and tracking accuracy to obtain similarity transformations ζ_{ji} between two keyframes:

$$\zeta_{ji} = \arg\min_{\zeta} \sum_{i \in \Omega_j \cap \Omega_i} \left(\frac{\delta_d^2(u_i, \zeta)}{\sigma_{\delta_d}^2(u_i)} + \frac{\delta_{ph}^2(u_i, \zeta)}{\sigma_{\delta_{ph}}^2(u_i)} \right)$$

where $\delta_d(u_i, \zeta)$ is the depth difference of the same pixel in the two keyframes, and $\sigma_{\delta_d}^2(u_i)$ is the variance of this depth difference. Finally, LSD-SLAM performs loop closure detection based on pose relationships and image content between keyframes [37], and uses bundle adjustment to optimize the global map [15].

2.3 DSO Algorithm Analysis

DSO is a monocular sparse direct-method visual odometry proposed by Engel in 2016, with stereo capabilities added in 2017 [38]. Like SVO, it is not a complete SLAM system and lacks a loop closure detection module. Unlike feature-based methods that require matching feature points for data association, DSO integrates data association and pose tracking within a unified nonlinear optimization framework. Unique photometric camera calibration models [39] and sliding window optimization [40,41] enable DSO to achieve excellent computational speed and tracking accuracy, pushing direct-method V-SLAM to new heights.

Since direct methods track camera pose based on image intensity values, they are susceptible to illumination changes. DSO introduces photometric calibration to compensate for vignetting (Figure 5 [Figure 5: see original paper]), exposure time, and gamma response [42]. Vignetting is the phenomenon where light intensity gradually decreases from the image center toward the periphery due to lens occlusion, calibrated by a weighting map $V(u) : \Omega \rightarrow [0, 1]$. Due to varying camera exposure modes and shutter mechanisms, different scenes have different exposure times, represented by exposure time t_{exp}. Gamma response nonlinearly maps the relationship between camera input exposure and output grayscale values, calibrated by a nonlinear response function $G : \mathbb{R} \rightarrow [0, 255]$. The grayscale mapping relationship is:

$$I_i = G(t_i V(u) B_i)$$

where B_i and I_i represent the radiance and grayscale values of frame i, respectively. The first step in DSO is grayscale calibration using Equation (13), after which the calibrated parameters are used for pose tracking.

Similar to LSD-SLAM, DSO' s frontend tracks camera pose by matching new frames with keyframes, then determines whether a new frame will become a keyframe based on certain conditions, inserting new keyframes into the backend optimization thread. DSO' s backend optimizes 5-7 keyframes (Figure 6 [Figure 6: see original paper]) within a sliding window. Pixels with converged depth are extracted from each keyframe and projected into other keyframes to obtain photometric errors:

$$E_{photo} = \sum_{i \in \mathcal{F}} \sum_{u \in \mathcal{P}_i} \sum_{j \in obs(u)} \| I_j[u'] - I_i[u] \|_\gamma$$

where u and u' are pixel coordinates in two frames, t_i and t_j are image exposure times, and a_i, b_i, a_j, b_j are photometric affine parameters that approximately achieve photometric calibration effects when actual photometric calibration is unavailable.

The sliding window ultimately optimizes the sum of photometric errors:

$$E = \sum_{i \in \mathcal{F}} \sum_{p \in \mathcal{P}_i} \sum_{j \in obs(p)} E_p^j$$

to obtain optimized keyframe poses and pixel positions, thereby maintaining a global map. Here, $\mathcal{F}$ represents the set of keyframes in the sliding window, $\mathcal{P}_i$ represents the set of pixels extracted from keyframe i, and $obs(p)$ represents the set of keyframes in the sliding window that can observe pixel p.

Unlike traditional bundle adjustment that optimizes all keyframes and feature points globally, DSO' s sliding window optimization maintains a fixed number of keyframes, requiring marginalization of excess frames. Marginalization in sliding window optimization updates the information matrix [43], preserving information from deleted frames as prior information in the information matrix, controlling computational complexity while achieving good optimization results.

2.4 Analysis and Comparison

Figure 7 [Figure 7: see original paper] shows DSO' s reconstruction results. Table 1 compares the aforementioned monocular SLAM methods.

1) Tracking Accuracy
SVO is a semi-direct visual odometry that only uses direct methods for sparse

feature point matching in the frontend, while still using reprojection errors to maintain the map in the backend. Therefore, its tracking accuracy surpasses LSD-SLAM, which uses direct pixel matching for pose tracking. LSD-SLAM' s pose tracking via pixel matching is vulnerable to environmental lighting changes and camera exposure variations, resulting in lower tracking accuracy. However, as a complete SLAM system, it can perform loop closure detection [37] and compensate for accuracy limitations through global optimization. DSO innovatively proposes photometric calibration, largely mitigating the impact of camera exposure on direct pixel matching. Sliding window optimization effectively eliminates error accumulation, yielding high localization accuracy. When using only the visual odometry module, results on the EuRoC dataset [44] show that DSO' s tracking accuracy exceeds SVO' s, while LSD-SLAM' s tracking accuracy is the least satisfactory.

2) Algorithm Efficiency

SVO cleverly avoids time-consuming descriptor computation and matching through direct image matching, and because it extracts sparse feature points, its computational efficiency is extremely high, achieving approximately 300 fps on a CPU (i7 processor/3 GHz) without requiring substantial computational resources. LSD-SLAM extracts pixel points from regions with significant gradients in images, making it a semi-dense direct method that requires solving a relatively dense information matrix, which affects algorithm efficiency to some extent. Additionally, LSD-SLAM requires loop closure detection for keyframes, resulting in lower algorithm efficiency, with processing speeds of approximately 20-30 fps on a CPU (i7 processor/3 GHz). DSO is a sparse direct method whose excellent image matching capability enables high computational speeds even at reduced image resolutions. The use of sliding window optimization controls the number of keyframes processed in the backend, keeping computational load low. There is a trade-off between DSO' s tracking accuracy and algorithm efficiency: higher image resolution generally yields higher tracking accuracy but lower efficiency. At comparable tracking accuracy, DSO' s algorithm efficiency surpasses that of LSD-SLAM.

3) Robustness to Illumination Changes

Direct methods track pose by comparing image information, making them susceptible to illumination variations. SVO computes sparse points with large pixel gradients, making it vulnerable to inconsistent environmental lighting changes and varying camera exposure times. LSD-SLAM' s robustness to illumination changes is also constrained by these factors. Photometric calibration and dynamically estimated photometric parameters make DSO more robust to image brightness variations caused by camera imaging, though its effectiveness on inconsistent environmental lighting changes is limited.

4) Robustness to Fast Motion

The ability to track camera pose during rapid camera motion reflects a SLAM algorithm' s robustness to fast motion. Unlike feature-based methods where feature descriptors can be matched despite large camera motions, direct methods

assume smooth camera motion and require good initial pose estimates for image matching. Consequently, direct methods generally exhibit poor fast-motion robustness. LSD-SLAM, which processes semi-dense pixel points during tracking, is most sensitive to camera motion speed and has poor fast-motion robustness. SVO' s high tracking efficiency and pose prediction based on image pyramid models partially compensate for this limitation. DSO can achieve fast-motion tracking by adjusting image resolution, making it more robust than LSD-SLAM.

5) Relocalization and Loop Closure Capability
Systems may lose pose tracking during operation; relocalization modules can recover lost poses. DSO lacks a relocalization module. SVO maintains a local map composed of keyframes and recovers initial pose by matching the current frame with the nearest keyframe when tracking is lost, then projects local map features into the current frame for feature matching and pose optimization to achieve relocalization. This method lacks robustness and fails when pose changes between the current frame and keyframes are large. When pose tracking is lost, LSD-SLAM uses feature point descriptors and FAB-MAP retrieval methods [37] for relocalization. Because feature descriptors are invariant to viewpoint changes, LSD-SLAM exhibits better relocalization robustness. Neither DSO nor SVO have loop closure detection modules. LSD-SLAM detects loops through feature points and FAB-MAP, and when loops occur, it optimizes global poses via pose graph optimization, demonstrating strong loop closure capability.

3 Advantages, Disadvantages, and Development Directions of Direct Methods

Direct methods establish data associations based on the special assumption of environmental photometric consistency, constructing residuals in the objective function from image intensity values. Consequently, their advantages and disadvantages are quite pronounced.

3.1 Advantages of Direct Methods

Direct methods avoid time-consuming feature point and descriptor extraction and matching processes (such as ORB [45], SIFT [46], SURF [47]), resulting in high computational efficiency. Under the same computational resources, direct methods can generally achieve hundreds of frames per second, compared to the tens of frames per second typical of feature-based methods. Moreover, unlike feature-based methods that extract only a few hundred pixel points as features, direct methods utilize information from hundreds of thousands of pixels, matching regions with pixel gradients. This allows accurate tracking even in environments with missing features or repetitive textures. The use of extensive image information also enables direct methods to recover dense or semi-dense scene maps. Additionally, without requiring feature matching, direct methods perform data association in a more holistic and robust manner, avoiding fatal impacts from feature mismatches on SLAM systems and providing higher

precision and system stability.

3.2 Disadvantages of Direct Methods

The nonlinear optimization in direct methods is based on image pixel gradients, using gradient descent to solve for optimal values in the cost function. Due to the non-convex nature of image pixels, the optimization process is prone to falling into local minima, requiring a good initial pose estimate and high image quality. Therefore, pose tracking is easily lost during fast camera motion and low frame rates. Furthermore, the photometric consistency assumption is a strong assumption with high requirements for environmental lighting. Changes in camera exposure time, shutter mechanisms, camera exposure adjustment, and environmental lighting conditions can cause tracking failure. Consequently, special cameras (such as event cameras [48-49]), global shutter lenses, and photometric calibration methods are needed. Finally, direct methods cannot use features like feature-based methods to match stored feature points and descriptors for relocalization and loop detection, necessitating innovative relocalization and loop closure modules for direct methods.

3.3 Development Directions of Direct Methods

The proposal of direct methods provides a new approach to solving V-SLAM problems, but they have not yet reached maturity, with many issues warranting further research. Direct-method V-SLAM is developing toward high precision, strong robustness, and multi-sensor fusion.

3.3.1 Multi-Sensor Fusion Direct-method V-SLAM uses cameras as sensors, making it susceptible to illumination effects and prone to pose tracking loss during lens occlusion or severe camera shake. Traditional localization methods also have limitations: IMU-based localization suffers from severe cumulative errors over time; civilian GPS offers poor accuracy and struggles to obtain position signals indoors or with severe outdoor occlusion. Single-sensor localization has inherent limitations, and multi-sensor fusion can improve system accuracy and robustness. Fusing cameras with IMU inertial devices can couple rich image information from cameras with short-term precise measurement data from IMUs [50-52], achieving excellent SLAM performance. Direct methods track pose based on pixel gradients and require good initial camera pose estimates for optimization, a problem well-addressed by IMU inertial devices. Additionally, precise pose tracking can be used to eliminate IMU bias drift. However, fusion with other sensor types increases the parameter count in direct-method V-SLAM optimization threads, increases information matrix density, and reduces system real-time performance. To address excessive optimization parameters, sliding window optimization can marginalize converged parameters as prior information for subsequent parameters. Current work on direct-method V-SLAM fused with IMU inertial devices includes [26,35].

3.3.2 Loop Closure and Relocalization Modules Direct-method V-SLAM uses pixel information directly for pose tracking and map building, lacking an image feature extraction process and making it difficult to perform loop closure detection and relocalization through image feature matching like feature-based methods. To address direct methods' difficulty in map reuse, two main research approaches exist. One, as in LSD-SLAM which extracts feature points for loop closure detection and relocalization, combines the advantages of feature-based and direct methods. Using features extracted by feature-based methods to achieve map reuse functionality, combined with direct methods' high tracking efficiency, system robustness, and rich image information usage, creates a fused system. The alternative approach uses deep learning methods to match keyframe images for map reuse. Deep learning employs neural networks to extract deeper-level features with higher recognition rates from images [53]. Using deep learning features can improve the accuracy of loop closure detection and relocalization in SLAM systems. Kendall proposed PoseNet in 2015, achieving real-time camera pose relocalization [54]. PoseNet convolutional neural networks are trained using image datasets with reconstructed 3D models and camera poses, employing an end-to-end approach for camera pose relocalization. Similar works include [55-57].

3.3.3 Fusion of Direct-Method V-SLAM and Deep Learning Deep learning is a recent research hotspot. As an end-to-end approach, deep learning can replace a module in SLAM systems or directly solve robot navigation problems using non-traditional frameworks [58,59]. Regarding the combination of deep learning and direct-method V-SLAM, Zhou' s SfM-Learner [60] estimates depth information for each image frame and tracks camera pose using CNNs based on photometric projection consistency assumptions, replacing parts of traditional SLAM system frontends. SfM-Net [61] extends SfM-Learner by computing optical flow and 3D point clouds. Similar works include [62]. CNN-SLAM [63] uses direct methods to track camera pose, combined with CNNs to estimate scene depth and perform image semantic segmentation, obtaining scene maps that combine geometric and semantic information. Additional works [64,65] have addressed feature extraction and semantic segmentation. SLAM is a geometric problem with strict closed-loop mathematical formulations, while deep learning excels in higher-level SLAM applications such as loop closure detection, image segmentation, and image semantics. The combination of both can advance SLAM technology.

4 Conclusion

In recent years, SLAM technology has been increasingly applied to virtual reality terminals, autonomous UAVs, and self-driving vehicles. The development and popularization of various hardware sensors and high-performance image processing units are driving SLAM technology toward high precision, strong robustness, and multi-sensor fusion. The proposal of direct-method V-SLAM technology provides a new approach to solving SLAM problems, compensat-

ing for some limitations of feature-based V-SLAM. However, direct methods still have many shortcomings, constrained by environmental lighting conditions, camera motion conditions, and difficulties in map reuse. This imposes higher requirements for robustness and functional expansion of direct-method V-SLAM algorithms. How to further improve direct-method V-SLAM' s robustness to environmental lighting and camera motion, and achieve map reuse functionality, will be a valuable research direction. Additionally, the combination of deep learning and SLAM technology has improved the limitations of traditional SLAM algorithms to some extent, providing ideas for more advanced SLAM applications. Applying deep learning to direct-method V-SLAM systems will be a meaningful and challenging research direction.

References

[1] Durrant-Whyte H, Bailey T. Simultaneous localization and mapping: Part I [J]. IEEE Robotics & Automation Magazine, 2006, 13 (2): 99–108.

[2] Bailey T, Durrant-Whyte H. Simultaneous localization and mapping (SLAM): Part II [J]. IEEE Robotics & Automation Magazine, 2006, 13 (3): 108-117.

[3] Fuentes-Pacheco J, Ruiz-Ascencio J, Rendón-Mancha J M. Visual simultaneous localization and mapping: a survey [J]. Artificial Intelligence Review, 2015, 43 (1): 55-81.

[4] Bonin-Font F, Ortiz A, Oliver G. Visual navigation for mobile robots: a survey [J]. Journal of Intelligent and Robotic Systems, 2008, 53 (3): 263-296.

[5] Faessler M, Fontana F, Forster C, et al. Autonomous, vision-based flight and live dense 3D mapping with a quadrotor micro aerial vehicle [J]. Journal of Field Robotics, 2016, 33 (4): 431-450.

[6] Liu H M, Zhang G F, Bao H J. A survey of monocular simultaneous localization and mapping [J]. Journal of Computer-Aided Design and Computer Graphics, 2016, 28 (6): 855–868.

[7] Smith R C, Cheeseman P. On the representation and estimation of spatial uncertainty [J]. International Journal of Robotics Research, 1986, 5 (4): 56-68.

[8] Smith R, Self M, Cheeseman P. Estimating uncertain spatial relationships in robotics [J]. Machine Intelligence & Pattern Recognition, 1990, 4 (5): 435-461.

[9] Thrun S, Burgard W, Fox D. Probabilistic Robotics [M]. Cambridge: MIT Press, 2005.

[10] Huang G Q, Mourikis A I, Roumeliotis S I. Analysis and improvement of the consistency of extended Kalman filter based slam [C]// Proc of IEEE International Conference on Robotics and Automation. 2008: 473-479.

[11] Martines-Cantin R, Castellanos J A. Unscented SLAM for large-scale outdoor environments [C]// Proc of IEEE//RSJ International Conference On In-

telligent Robots and Systems. 2005: 401-422.

[12] Montemerlo M. FastSLAM: A Factored solution to the simultaneous localization and mapping problem with unkown data association [D]. Pittsburgh, PA: Carnegie Mellon University, 2003.

[13] Davison A J, Reid I D, Molton N D, et al. Monoslam: real-time single camera slam [J]. IEEE Trans on Pattern Analysis and Machine Intelligence, 2007, 29 (6): 1052-1067.

[14] Hartley R, Zisserman A. Multiple view geometry in computer vision [M]. Cambridge: Cambridge University Press, 2004.

[15] Triggs B, McLauchlan P F, Hartley R I, et al. Bundle adjustment: a modern synthesis [C]// Proc of International Workshop on Vision Algorithms. Berlin: Springer, 1999: 298–372.

[16] Klein G, Murry D. Parallel tracking and mapping for small AR workspaces [C]// Proc of IEEE and ACM International Symposium on Mixed and Augmented Reality. Los Alamitos: IEEE Computer Society Press, 2007: 225-234.

[17] Klein G, Murry D. Parallel tracking and mapping on a camera phone [C]// Proc of IEEE and ACM International Symposium on Mixed and Augmented Reality. Los Alamitos: IEEE Computer Society Press, 2009: 83-86.

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

[19] 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.

[20] Engel J, Schöps T, Cremers D. LSD-SLAM: large-scale direct monocular SLAM [C]// Proc of Computer Vision-ECCV. Heidelberg: Springer, 2014: 834-849.

[21] Gao Xiang, Zhang Tao, et al. The fourteen lectures on visual SLAM [M]. Beijing: Publishing House of Electronics Industry, 2017.

[22] 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.

[23] Engel J, Koltun V, Cremers D. Direct sparse odometry [J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2016, pp (99): 1-14.

[24] Newcombe R A, Lovegrove S J, Davison A J. DTAM: dense tracking and mapping in real-time [C]// Proc of International Conference on Computer Vision. 2011: 2320-2327.

[25] Kerl C, Sturm J, Cremers D. Robust odometry for RGB-D cameras [C]// Proc of IEEE International Conference on Robotics and Automationa. 2013:

3748-3754.

[26] Forster C, Zhang Z C, Michael G, et al. SVO: semidirect visual odometry for monocular and multicamera systems [J]. IEEE Trans on Robotics, 2017, 33 (2): 249-265.

[27] Civera J, Davison A J, Montiel J M. Inverse depth parametrization for monocular SLAM [J]. IEEE Trans on Robotics, 2008, 24 (5): 932-945.

[28] Vogiatzis G, Hernandez C. Video-based, real-time multi-view stereo [J]. Image & Vision Computing, 2011, 29 (7): 434-441.

[29] Pizzoli M, Forster C, Scaramuzza D. REMODE: probabilistic, monocular dense reconstruction in real time [C]// Proc of IEEE International Conference on Robotics and Automation. 2014: 2609-2616.

[30] Engel J, Cremers D. Semi-dense visual odometry for a monocular camera [C]// Proc of IEEE International Conference on Computer Vision. 2013: 1449-1456.

[31] Schops T, Enge J, Cremers D. Semi-dense visual odometry for AR on a smartphone [C]// Proc of IEEE International Symposium on Mixed and Augmented Reality. Washington DC: IEEE Computer Society, 2014: 145-150.

[32] Engel J, Stückler J, Cremers D. Large-scale direct SLAM with stereo cameras [C]// Proc of IEEE//RSJ International Conference on Intelligent Robots and Systems. 2015: 1935-1942.

[33] Caruso D, Engel J, Cremers D. Large-scale direct SLAM for omnidirectional cameras [C]// Proc of IEEE//RSJ International Conference on Intelligent Robots and Systems. 2015: 141-148.

[34] Usenko V, Engel J, Stückler J, et al. Reconstructing street-scenes in real-time from a driving car [C]// Proc of International Conference on 3D Vision. Washington DC: IEEE Computer Society, 2015: 607-614.

[35] Usenko V, Engel J, Stückler J, et al. Direct Visual-Inertial Odometry with Stereo Cameras [C]// Proc of IEEE International Conference on Robotics and Automation. 2016: 1885-1892.

[36] Barfoot T D. State estimation for robotics: a matrix Lie group approach [M]. Cambridge: Cambridge University Press, 2017.

[37] Glover A, Maddern W, Warren M, et al. OpenFABMAP: an open source toolbox for appearance-based loop closure detection [C]// Proc of IEEE International Conference on Robotics and Automation. 2012: 4730-4735.

[38] Wang R, Schwörer M, Cremers D. Stereo DSO: large-scale direct sparse visual odometry with stereo cameras [C]// Proc of IEEE International Conference on Computer Vision. 2017: 3923-3931.

[39] Engel J, Usenko V, Cremers D. A photometrically calibrated benchmark for monocular visual odometry [EB/OL]. (2016-10-08) [2017-10-24].

http://arxiv.org/abs/1607.02555.

[40] Stefan L, Simon L, Michael B, et al. Keyframe-based visual-inertial odometry using nonlinear optimization [J]. International Journal of Robotics Research, 2015, 34 (3): 314-334.

[41] Sibley G, Matthies L, Sukhatme G. Sliding window filter with application to planetary landing [J]. Journal of Field Robotics, 2010, 27 (5): 587–608.

[42] Debevec P E, Malik J. Recovering high dynamic range radiance maps from photographs [C]// Proc of Conference on Computer Graphics and Interactive Techniques. 1997: 369-378.

[43] Eckenhoff K, Paull L, Huang G. Decoupled, consistent node removal and edge sparsification for graph-based SLAM [C]// Proc of IEEE//RSJ International Conference on Intelligent Robots and Systems. 2016: 3275-3282.

[44] Burri M, Nikolic J, Gohl P, et al. The EuRoC micro aerial vehicle datasets [J]. International Journal of Robotics Research, 2016, 35 (10): 1157-1163.

[45] Rublee E, Rabaud V, Konolige K, et al. ORB: an efficient alternative to SIFT or SURF [C]// Proc of IEEE International Conference on Computer Vision. 2011: 2564-2571.

[46] Lowe D G. Distinctive Image Features from Scale-Invariant Keypoints [J]. International Journal of Computer Vision, 2004, 60 (2): 91-110.

[47] Bay H, Tuytelaars T, Gool L V. SURF: speeded up robust features [J]. Computer Vision & Image Understanding, 2006, 110 (3): 404-417.

[48] Kim H, Handa A, Benosman R, et al. Simultaneous mosaicing and tracking with an event camera [C]// Proc of the British Machine Vision Conference. 2014: 1-12.

[49] Rebecq H, Horstschaefer T, Gallego G, et al. EVO: a geometric approach to event-based 6-DOF parallel tracking and mapping in real time [J]. IEEE Robotics & Automation Letters, 2017, 2 (2): 593-600.

[50] Weiss S M. Vision based navigation for micro helicopters [D]. Zurich, Switzerland: ETH Zurich, 2012.

[51] Mourikis A I, Roumeliotis S I. A Multi-State Constraint Kalman Filter for Vision-aided Inertial Navigation [C]// Proc of IEEE International Conference on Robotics and Automation. 2007: 3565-3572.

[52] Li M Y, Mourikis A I, Anastasios I. High-precision, consistent EKF-based visual–inertial odometry [J]. International Journal of Robotics Research, 2013, 32 (6): 690-711.

[53] Girshick R, Donahue J, Darrel T, et al. Rich feature hierarchies for accurate object detection and semanic segmentation [C]// Proc of IEEE Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE Press, 2014: 580-587.

[54] Kendall A, Grimes M, Cipolla R. PoseNet: A convolutional network for real-time 6-DOF camera relocalization [C]// Proc of IEEE International Conference on Computer Vision. 2015: 2938-2946.

[55] Wu J, Ma L, Hu X L. Delving deeper into convolutional neural networks for camera relocalization [C]// Proc of IEEE International Conference on Robotics and Automation. 2017: 5644-5651.

[56] Chen Z, Jacobson A, Sunderhauf N, et al. Deep learning features at scale for visual place recognition [C]// Proc of IEEE International Conference on Robotics and Automation. 2017: 3223-3230.

[57] Gao X, Zhang T. Unsupervised learning to detect loops using deep neural networks for visual SLAM system [J]. Autonomous Robots, 2017, 41 (1): 1-18.

[58] Zhu Y, Mottaghi R, Kolve E, et al. Target-driven visual navigation in indoor scenes using deep reinforcement learning [EB/OL]. (2016-09-16) [2017-10-24]. http://arxiv.org/abs/1609.05143.

[59] Gupta S, Davidson J, Levine S, et al. Cognitive mapping and planning for visual navigation [C]// Proc of IEEE Conference on Computer Vision and Pattern Recognition. Washington DC: IEEE Computer Society, 2017: 7272-7281.

[60] Zhou T, Brown M, Snavely N, et al. Unsupervised learning of depth and ego-motion from video [C]// Proc of IEEE Conference on Computer Vision and Pattern Recognition. Washington DC: IEEE Computer Society, 2017: 1851-1858.

[61] Vijayanarasimhan S, Ricco S, Schmid C, et al. SfM-Net: learning of structure and motion from video [EB/OL]. (2017-04-25) [2017-10-24]. http://arxiv.org/abs/1704.07804.

[62] Ummenhofer B, Zhou H, Uhrig J, et al. DeMoN: depth and motion network for learning monocular stereo [C]// Proc of IEEE Conference on Computer Vision and Pattern Recognition. 2016: 5622-5631.

[63] Tateno K, Tombari F, Laina I, et al. CNN-SLAM: Real-time dense monocular SLAM with learned depth prediction [C]// Proc of IEEE Conference on Computer Vision and Pattern Recognition. Washington DC: IEEE Computer Society, 2017: 6565-6574.

[64] Yi K M, Trulls E, Lepetit V, et al. LIFT: learned invariant feature transform [C]// Proc of European Conference on Computer Vision. Amsterdam: Springer, 2016: 467-483.

[65] Li X, Belaroussi R. Semi-dense 3D semantic mapping from monocular SLAM [EB/OL]. (2016-11-13) [2017-10-24]. http://arxiv.org/abs/1611.04144.

Note: Figure translations are in progress. See original paper for figures.