A Survey of Visual Odometry Algorithms: Post-print

Authors: Ci Wenyan, Huang Yingping, Hu Xing

Date: 2018-09-12T00:00:00+00:00

Abstract

Visual odometry estimates the pose of mobile robots by analyzing image stream information acquired from cameras. To provide an in-depth analysis of the current development status of visual odometry algorithms, this paper surveys the related technologies and latest research achievements of visual odometry, drawing upon several advanced visual odometry systems. First, it briefly introduces the concept and development history of visual odometry, and presents the mathematical description and classification methods for the visual odometry problem. Then, it elaborates in detail on the key technologies of visual odometry, including feature modules, inter-frame pose estimation, and drift reduction. Furthermore, it introduces the development trends of deep learning-based visual odometry. Finally, it summarizes the existing problems in visual odometry and provides an outlook on future development trends.

Full Text

Preamble

Review of Visual Odometry Algorithms

Ci Wenyan[1],[2], *Huang Yingping*[1] †, *Hu Xing*[1]

[1]School of Optical-Electrical & Computer Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China
[2]School of Electric Power Engineering, Nanjing Normal University Taizhou College, Taizhou Jiangsu 225300, China

Abstract: Visual odometry (VO) estimates the pose of a mobile robot by analyzing the image flow captured by onboard cameras. To provide a comprehensive analysis of the current state of VO algorithm development, this paper reviews relevant technologies and the latest research findings in conjunction

with several advanced VO systems. First, we introduce the concept and evolution of VO, along with the mathematical formulation and classification of VO problems. We then examine key VO technologies in detail, including feature modules, inter-frame pose estimation, and drift reduction. Additionally, we discuss recent developments in deep learning-based VO. Finally, we summarize existing challenges in VO and outline future research trends.

Keywords: machine vision; visual odometry; pose estimation; vision-based navigation; mobile robots; deep learning

0 Introduction

In mobile robot systems, accurate self-localization is essential for target detection and positioning. Traditional pose estimation methods include GPS, IMU, wheel odometry, and sonar-based localization. In recent years, camera systems have become more affordable while offering higher resolution and frame rates, and computer performance has improved significantly, enabling real-time image processing. This has given rise to a new pose estimation approach called visual odometry (VO). VO estimates the pose of an intelligent agent using image streams from single or multiple cameras. It offers low cost, operates in GPS-denied environments such as underwater and aerial scenarios, exhibits less drift than wheel odometry and low-precision IMU, and produces data that can be easily integrated with other vision-based algorithms without requiring inter-sensor calibration.

The concept of estimating camera ego-motion from consecutive image sequences was first proposed by Moravec et al. [2], who used a sliding camera to acquire visual information and completed indoor robot navigation. In 1987, Matthies et al. [3] established a theoretical framework encompassing feature extraction, feature matching and tracking, and motion estimation—a framework that remains the foundation for most VO systems today. The majority of early VO systems were applied to planetary exploration [2,4], most notably NASA' s Mars exploration program, where VO measured six-degree-of-freedom parameters when wheel odometry failed. The term "visual odometry" was coined by Nister et al. [5] in 2004, who designed a real-time VO system that achieved true outdoor robot navigation and proposed two implementation approaches: monocular and stereo vision, laying new groundwork for subsequent VO research.

A closely related field is visual simultaneous localization and mapping (V-SLAM) [6-8]. V-SLAM performs self-localization in unknown environments while reconstructing 3D structures in real-time, aiming for globally consistent trajectory estimation. This requires the robot to recognize previously visited locations through loop closure detection. In contrast, VO incrementally reconstructs local paths, focusing only on local trajectory consistency. From the perspectives of real-time performance and environmental adaptability, VO—which specializes

in local motion estimation—offers greater practical value and is more suitable for mobile robots operating over large distances.

Several review papers on VO have been published previously [1,9-11], particularly the two articles by Scaramuzza et al. [1,11] that systematically covered VO development prior to 2012. However, VO technology has advanced significantly in recent years, with numerous high-performance VO systems emerging, rendering these earlier surveys outdated. This review emphasizes integration with state-of-the-art VO systems. We begin with an overview of VO, including its mathematical formulation and classification. We then focus on key technologies: feature modules, inter-frame pose estimation, and drift reduction. For the emerging deep learning-based VO, we summarize its development and analyze its strengths and weaknesses. Recognizing the importance of algorithm evaluation for VO advancement, we also introduce three commonly used public datasets. Finally, we summarize current challenges and future trends.

---

1 Overview of Visual Odometry

1.1 Mathematical Formulation of Visual Odometry

The camera model is a function that projects the 3D world onto a 2D image plane. Numerous camera models exist, including perspective projection, omnidirectional, and spherical models, with the perspective projection model being the most fundamental and widely used. In perspective projection, distant objects appear smaller than nearby ones—a property consistent with human vision and most cameras. The geometric relationship of perspective projection is shown in [Figure 1: see original paper].

In the figure, point C is the camera optical center. The coordinate system formed by point C and the x, y, z axes is called the camera coordinate system. The x and y axes are parallel to the image u and v axes, while the z axis is the optical axis, perpendicular to the image plane. The intersection of the optical axis with the image plane is the origin of the image coordinate system. Point P represents a point in the 3D world, and point p is its projection onto the image plane.

The perspective projection equation from 3D world to 2D image plane can be expressed as:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

where $K = \begin{bmatrix} f_u & 0 & u_c \\ 0 & f_v & v_c \\ 0 & 0 & 1 \end{bmatrix}$ is the camera intrinsic parameter matrix. Here,

f_u and f_v are the focal lengths in the u and v directions, and u_c, v_c are the principal point coordinates. These parameters depend only on the camera' s internal structure and are therefore called intrinsic parameters.

Assume an intelligent agent moves in an environment with a camera rigidly mounted (no relative motion between camera and agent). The camera captures images at discrete time instants $k = 0, 1, ..., n$. The image sequence can be represented as $\mathcal{I} = \{I_0, I_1, ..., I_n\}$. The coordinate transformation from time $k-1$ to k can be expressed as:

$$T_{k,k-1} = \begin{bmatrix} R_{k,k-1} & t_{k,k-1} \\ 0 & 1 \end{bmatrix}$$

where $R_{k,k-1} \in SO(3)$ is the rotation matrix and $t_{k,k-1} \in \mathbb{R}^{3\times 1}$ is the translation vector. The homogeneous coordinates of point P in the camera coordinate system at time k are $P_k = [X_k, Y_k, Z_k, 1]^T$. Since the camera is moving, P_k in the current frame should be obtained by transforming its coordinates from the previous frame according to the camera pose:

$$P_k = T_{k,k-1} P_{k-1}$$

Both sides of this equation are homogeneous coordinates, which remain equivalent when multiplied by any non-zero constant. Therefore, we can drop the homogeneous coordinate, yielding:

$$\begin{bmatrix} X_k \\ Y_k \\ Z_k \end{bmatrix} = R_{k,k-1} \begin{bmatrix} X_{k-1} \\ Y_{k-1} \\ Z_{k-1} \end{bmatrix} + t_{k,k-1}$$

$T_{k,k-1}$ is also called the camera extrinsic parameter matrix, which is the target to be estimated in VO. Its calculation methods will be discussed later.

For brevity, we denote $T_{k,k-1}$ as T_k. Assume the set $\mathcal{T}_{0:n} = \{T_1, T_2, ..., T_n\}$ contains all relative motions between consecutive frames, and the set $\mathcal{C}_{0:n} = \{C_0, C_1, ..., C_n\}$ contains all poses relative to the initial coordinate frame at time 0. The current pose C_k can be computed from the initial pose C_0 and the relative motions as:

$$C_k = C_0 \prod_{i=1}^{k} T_i$$

The primary task of VO is to compute the relative motion T_k from images I_{k-1} and I_k, thereby recovering the complete camera trajectory $\mathcal{C}_{0:n}$.

1.2 Classification of Visual Odometry

Clarifying VO classifications helps understand the field macroscopically. VO can be categorized from different perspectives: by camera type (monocular, stereo, RGB-D), by image information usage (feature-based vs. direct methods), and by drift reduction approach (filter-based vs. nonlinear optimization). lists common VO systems and their classifications.

** Common VO Systems and Their Classifications**

System	Camera Type	Image Information	Drift Reduction
SVO[12]	Monocular	Direct method	Nonlinear optimization
PTAM[13]	Monocular	Feature-based	Nonlinear optimization
ORB-SLAM2[14,15]	Monocular, Stereo, RGB-D	Feature-based	Nonlinear optimization
VISO2[16]	Monocular, Stereo	Feature-based	Nonlinear optimization
LSD-SLAM[17]	Monocular	Direct method	Nonlinear optimization
TLBBA[18]	Stereo	Feature-based	Nonlinear optimization
MonoSLAM[19]	Monocular	Feature-based	Filter-based
DEMO[20]	RGB-D	Feature-based	Nonlinear optimization
RTAB-MAP[21]	Stereo, RGB-D	Feature-based	Nonlinear optimization

1.2.1 Monocular, Stereo, and RGB-D **Monocular VO** systems use a single camera. Their advantages include simple sensors, low cost, and strong environmental adaptability. However, the primary limitation is the inability to determine absolute depth, resulting in scale-ambiguous motion trajectories. This requires known distances between two points in space or integration with other sensors like LiDAR or IMU. Additionally, monocular depth estimation relies on triangulation across consecutive frames, and pure rotational camera motion cannot be handled. These constraints limit monocular VO applications.

Stereo VO systems use binocular or multi-camera configurations. Stereo vision obtains depth information through a fixed baseline, avoiding monocular scale ambiguity and enabling triangulation within a single frame. However, stereo cameras are more expensive and require complex calibration. Depth measurement demands stereo matching, which is computationally intensive. Moreover, the measurement range is limited by the baseline length; when measuring depths far exceeding the baseline, stereo vision degenerates to monocular vision. Currently, monocular and stereo approaches are in balanced development.

RGB-D cameras, emerging around 2010, simultaneously capture color and depth information. Compared to stereo vision, RGB-D cameras save substantial computation time for depth estimation. However, most RGB-D cameras suffer from limited measurement range and sunlight sensitivity, making them primarily suitable for indoor environments. Despite their recent introduction, RGB-D cameras have developed rapidly, with many excellent VO solutions [20,21] based on them.

1.2.2 Feature-Based and Direct Methods VO follows the theoretical framework of feature modules, inter-frame pose estimation, and drift reduction, as illustrated in [Figure 2: see original paper].

[Figure 2: see original paper] VO Implementation Flowchart

The feature module includes feature detection and matching. For each new image frame, the algorithm first detects salient, repeatable image features for pose estimation, then performs feature matching between current and previous frames. Feature matching aims to find feature point correspondences—2D points that are projections of the same 3D point in two frames. Inter-frame pose estimation includes outlier rejection and motion estimation. The resulting feature correspondences typically contain outliers that violate the mathematical model, which must be excluded as they severely impact motion estimation. Subsequently, the relative camera motion between current and previous frames is computed from the remaining feature pairs. Outlier rejection and motion estimation are usually iterative processes. Since inevitable error accumulation occurs in two-frame pose estimation, drift reduction methods are employed to obtain more accurate camera poses, primarily through filter-based or nonlinear optimization approaches.

Feature-based methods extract salient features from dense image data for computation. Feature-based VO systems operate stably with low computational cost and are robust to illumination changes and image noise. Their disadvantage is poor performance in feature-scarce scenes, such as low-texture images.

Direct methods utilize intensity information from all pixels in an image or subregion to compute camera motion. Direct VO does not require feature points, only pixel gradients. It fully exploits image information, facilitating dense map construction and other vision applications. However, direct methods are computationally intensive, unsuitable for large motions, and require the brightness constancy assumption, which can be violated by illumination changes. Based on pixel usage, direct methods can be sparse, semi-dense, or dense. Although recent direct VO systems like SVO[12] and LSD-SLAM[17] have emerged, mature solutions remain limited, and feature-based methods still dominate mainstream VO.

2 Key Technologies

2.1 Feature Module

The feature module forms the foundation for subsequent pose estimation. We discuss feature detection and matching separately.

2.1.1 Feature Detection Classic feature detection algorithms include Moravec, Forstner, Harris, Shi-Tomasi, SUSAN, FAST, SIFT, SURF, MSER, and Censure. Among these, Harris[22] and SIFT[23] are most widely used. Harris corners exhibit strong stability against noise and rotation, providing rich information and serving as a common feature detector in vision-based pose estimation systems [5,20]. However, Harris corners are sensitive to scale and affine transformations. Parra et al.[24] demonstrated that Harris corners produce false matches when scenes contain repetitive textures, suggesting SIFT features are more suitable for VO. SIFT exhibits rotation and scale invariance, along with robustness to illumination, viewpoint changes, and noise, leading many VO systems to adopt SIFT features [24,25]. SIFT' s main drawback is low computational efficiency. To meet real-time requirements, TLBBA[18] simplified SIFT for VO conditions and achieved 40 Hz feature tracking by leveraging GPU acceleration.

Recent years have seen numerous new algorithms building upon classic feature detection. In 2011, Rublee et al. proposed ORB[26] based on FAST and BRIEF, offering good rotation and scale invariance at 30-50× the speed of SIFT. ORB has been successfully applied in the renowned ORB-SLAM[14,15], demonstrating its excellence in balancing accuracy and efficiency. Also in 2011, Leutenegger et al. introduced BRISK[27], which uses an adaptive generic accelerated segment test, achieving faster feature detection than ORB. In 2012, Alcantarilla et al. proposed KAZE[28] based on nonlinear scale space theory, offering better scale and rotation invariance than SIFT. They later released A-KAZE[29] in 2013, which significantly improved computational speed.

2.1.2 Feature Matching After feature detection, each feature point and its neighborhood must be converted into a compact descriptor for matching. Classic feature descriptors include SIFT[23] and its derivative SURF[30]. SIFT has proven highly stable against illumination, rotation, scale, and viewpoint changes up to 60°. SURF approximates the difference-of-Gaussian filter with box filters for higher computational efficiency. Other SIFT-derived descriptors include PCA-SIFT[31] and DAISY[32], which primarily address SIFT's efficiency limitations.

Since 2010, binary string descriptors have emerged, including BRIEF[33], ORB[26], BRISK[27], FREAK[34], and NESTED[35], generally offering higher efficiency than floating-point descriptors. Hartmann et al.[36] compared SIFT, SURF, BRIEF, ORB, BRISK, and FREAK, finding SIFT still achieved the highest accuracy, while BRIEF was optimal when computational efficiency was

paramount. Khan et al.[37] evaluated popular feature descriptors across eight image datasets, with SIFT showing the best overall performance. Among binary descriptors, NESTED achieved the best results. Notably, their comparison results varied across datasets, indicating that feature algorithm performance is scene-dependent.

Feature matching aims to find corresponding feature pairs between two sets based on similarity metrics such as Euclidean distance (for floating-point descriptors) or Hamming distance (for binary descriptors). A critical issue is the matching search algorithm. The most common approach is Approximate Nearest Neighbor (ANN) search, which trades small accuracy losses for significant speed improvements. In 1997, Beis et al. proposed the BBF algorithm[38] based on approximate Kd-trees, which finds nearest neighbors with 95% probability while achieving 1000× speedup, making it widely adopted [23,24]. Other ANN algorithms include Spill-tree[39], hierarchical K-means trees[40], and random kd-trees[41]. Additionally, ORB-SLAM[14,15] employs a bag-of-words model[42], while LSD-SLAM[17] uses the FAB-MAP method[43] for efficient matching. Search speed can be further accelerated by adding constraints such as motion model constraints[16,44], 3D feature point position constraints[45], and epipolar constraints[24].

2.2 Inter-Frame Pose Estimation

2.2.1 Outlier Rejection Outliers primarily arise from two sources: (a) mismatches due to image noise, illumination changes, viewpoint variations, and inherent limitations of matching algorithms; and (b) moving objects in the scene. These outliers significantly impact motion estimation and must be removed for accurate results.

A classic and effective outlier rejection method is the Random Sample Consensus (RANSAC) algorithm[46], which iteratively extracts optimal subsets from data containing numerous outliers. RANSAC randomly samples a minimal subset to compute model parameters, then validates other data points against this model. After multiple iterations, the model parameters achieving highest consensus are selected as the solution, with inconsistent points classified as outliers. The subset size is typically the minimum required to solve the model (e.g., 3 for stereo vision). The key parameter is the number of iterations M, estimated by:

$$M = \frac{\log(1-p)}{\log[1-(1-\varepsilon)^s]}$$

where s is the subset size, ε is the outlier probability, and p is the desired probability of obtaining a reasonable result. [Figure 3: see original paper] shows an example of RANSAC-based outlier rejection (red "+" marks removed points, green "+" marks retained points).

RANSAC is a non-deterministic algorithm—results may vary between runs.

More iterations increase the probability of success. RANSAC has become a universal outlier rejection method in VO systems [44,47]. Recent improvements include MLESAC[48], which evaluates hypothesis similarity using a mixture model of errors rather than counting inliers, and PROSAC[49], which guides sampling when prior information about outlier likelihood is available. Rusu et al.[50] propose sampling based on most similar feature histograms. Raguram et al.[51] comprehensively analyzed RANSAC variants, proposing an adaptive real-time RANSAC algorithm (ARRSAC).

Beyond RANSAC derivatives, alternative outlier rejection methods exist. VISO2-S[16] uses triangulation voting. Some works [5,52] employ bucketing to distribute features uniformly across the image, improving VO accuracy. However, VO research on outliers from moving objects is limited—systems like PTAM[13], ORB-SLAM[14,15], and LSD-SLAM[17] assume static scenes, making them unsuitable for highly dynamic environments. To address this, Zhejiang University' s RDSLAM[53] detects scene changes online and identifies altered 3D points. Other approaches [54,55] detect only ground points. Ci et al.[56] propose a spatial position constraint method based on vehicle motion smoothness. However, these methods cannot handle scenes with short-term major changes, leaving significant room for improvement.

2.2.2 Motion Estimation Motion estimation is the core computational step in VO. Specifically, it computes the transformation matrix $T_{k,k-1}$ between current image I_k and previous frame I_{k-1}. Concatenating these single-step motions recovers the complete camera and agent trajectory. Given corresponding feature points between frames $k-1$ and k, three methods exist for computing $T_{k,k-1}$ based on point dimensionality: 3D-3D, 3D-2D, and 2D-2D.

3D-3D solves motion from 3D point pairs, typically used in stereo vision. It estimates $T_{k,k-1}$ by minimizing Euclidean distances between 3D point pairs:

$$T_{k,k-1} = \arg\min_T \sum_i \|Q_k^i - TQ_{k-1}^i\|^2$$

where Q_{k-1}^i and Q_k^i are 3D affine coordinates of feature points in frames $k-1$ and k. At least three non-collinear 3D point pairs are required. While more pairs increase computation, they improve accuracy, so typically far more than three pairs are used. Solution methods include singular value decomposition[57], quaternion-based methods[58], and Iterative Closest Point (ICP)[4].

3D-2D solves motion from 3D space points and 2D image points, applicable to both monocular and stereo vision. It minimizes 2D reprojection error:

$$T_{k,k-1} = \arg\min_T \sum_i \|q_k^i - \pi(TQ_{k-1}^i)\|^2$$

where $\pi(\cdot)$ is the projection function onto image I_k after motion transformation T. 3D-2D is also called Perspective-n-Point (PnP) and is currently the most commonly used method. Moreno-Noguer et al.[59] surveyed PnP solutions. The minimal case, P3P, requires at least 3 point pairs, with over 10 solution methods developed[60]. For outlier-contaminated cases, P3P with robust estimators like RANSAC is standard.

2D-2D solves motion parameters from 2D image point pairs, generally used in monocular VO when 3D data is unavailable (e.g., initializing the first two frames). In this case, epipolar constraints estimate the transformation, as illustrated in [Figure 4: see original paper]. Epipolar geometry constrains corresponding 3D points viewed from different perspectives. For corresponding points q_{k-1} in image I_{k-1} and q_k in I_k (both projecting from 3D point Q), the camera centers and points lie on the same plane. This coplanarity constraint yields:

$$q_{k-1}^T F q_k = 0$$

where F is the fundamental matrix containing inter-frame motion and camera intrinsics. With known intrinsic matrix K, the essential matrix E is:

$$m_{k-1}^T E m_k = 0$$

where m_{k-1}, m_k are normalized image coordinates. E encapsulates rotation and translation parameters, with translation determined up to a scale factor.

For calibrated cameras, solving 2D-2D requires at least 5 point pairs. The 5-point algorithm combined with robust estimators (particularly Nister' s efficient 5-point algorithm[61] and its improvements[62]) has become the standard for 2D-2D problems with outliers. Other methods include 6-point[63], 7-point[64], and 8-point algorithms[65]. Stewenius et al.[62] compared various 2D-2D solvers, finding the efficient 5-point algorithm offers the best overall performance. Some automotive VO systems use motion model constraints to reduce required point pairs. For example, Fraundorfer et al.[66] proposed a 3-point algorithm for known rotation angles. For planar motion (3 DOF), only 2 point pairs are needed. Scaramuzza et al.[67] exploited non-holonomic vehicle constraints to reduce motion model complexity to 2 DOF, solving vehicle motion with just 1 feature pair at 400 Hz.

Generally, 3D-2D achieves higher accuracy than 3D-3D because triangulation has high depth uncertainty, especially for distant points where depth variations cause minimal projection changes, as shown in Figure 5: see original paper. Li et al.[68] proved that depth measurement error (Δz) is proportional to depth squared (z^2), as shown in Figure 5: see original paper. This uncertainty severely impacts 3D-3D motion estimation. In 3D-2D, the cost function uses image reprojection error, largely canceling this uncertainty. To fundamentally address

depth uncertainty, Forster et al.[12] introduced depth filters in their SVO system, deriving uniform-Gaussian mixture depth filters for feature point position estimation, achieving good results.

In practice, 3D-2D is more widely used than 2D-2D in monocular VO because 3D-2D data association is faster. Accurate motion estimation requires effective outlier rejection, whose computational time is tightly coupled with the minimum feature points needed. While 2D-2D requires at least 5 pairs, 3D-2D needs only 3, making it faster. Thus, 2D-2D is typically used only for monocular VO initialization.

2.3 Drift Reduction

As described in Section 1.1, current pose C_k is computed by concatenating single-step relative motions T_i from initial pose C_0. Errors inevitably exist between each estimated T_i and actual camera motion. Smith et al.[69]' s error propagation law proves that pose error accumulates with concatenated motions —a phenomenon called drift, illustrated in [Figure 6: see original paper]. Drift reduction primarily employs filter-based or nonlinear optimization methods.

2.3.1 Filter-Based Methods Filter methods dominated early VO, with Extended Kalman Filter (EKF)[19] being most common. EKF treats current camera pose and all 3D points as state variables, updating their means and covariances. For nonlinear VO systems, EKF provides a maximum a posteriori estimate under single linear approximation. Kitt et al.[52] used Unscented Kalman Filter (UKF) for more accurate results than EKF, as UKF achieves near-third-order precision compared to EKF' s first-order Taylor expansion. Other algorithms reduce computational complexity relative to EKF, such as Sparse Extended Information Filter (SEIF)[70], Atlas framework[71], and divide-and-conquer methods[72]. However, filter methods have significant limitations: they assume Markov property (state at time k depends only on time $k-1$), still causing error accumulation. Therefore, filter methods are generally used when computational resources are limited or state estimation is simple, while nonlinear optimization has become the mainstream approach.

2.3.2 Nonlinear Optimization Methods Nonlinear optimization considers relationships between the current state and all previous states. PTAM[13] was the first system to use nonlinear optimization, introducing keyframe-based processing: rather than processing every frame meticulously, it chains several keyframes and optimizes their trajectory. Bundle Adjustment (BA) is the most widely used method, optimizing camera poses and 3D point coordinates by minimizing reprojection errors across multiple frames. When the cost function considers all frames, it' s called Global BA (GBA); when considering a fixed window of frames, it' s called Local BA (LBA). GBA offers higher accuracy than LBA but is computationally expensive. LBA is more suitable for real-time systems—for example, ORB-SLAM[14] includes an LBA optimization thread for

refined camera pose and 3D point estimation. TLBBA[18] uses a two-stage local binocular BA method, fully exploiting information and constraints from stereo sequences. Many state-of-the-art VO systems currently employ LBA[73,74].

3 Deep Learning-Based Visual Odometry

The aforementioned VO systems recover camera motion using geometric principles. In contrast, recent research has explored deep learning (DL) to reveal relationships between image optical flow and camera motion, offering new VO solutions. Konda et al.[75] first implemented DL-based VO by extracting visual motion and depth information from stereo images, using a Convolutional Neural Network (CNN) with softmax to predict camera velocity and direction changes. Kendall et al.[76] developed an end-to-end localization system using CNN, taking RGB images as input and outputting camera pose. Their 23-layer PoseNet leverages transfer learning from classification databases for complex image regression. The learned features demonstrate stronger robustness than traditional local visual features against illumination, motion blur, and camera intrinsics. They proposed using Structure-from-Motion (SfM) to automatically generate training annotations, eliminating manual pose labeling, though this is time-consuming for large-scale scenes. Since the system uses a trained CNN as an appearance map, retraining or fine-tuning is required for new environments —a major challenge for DL-based VO.

To address this, Costante et al.[77] used dense optical flow instead of RGB images as CNN input, designing three CNN architectures for VO feature learning, achieving robustness under image blur and underexposure. However, results show training data significantly impacts performance, with large errors for large inter-frame motions due to lack of high-speed training samples.

Wang et al.[78] used Deep Recurrent Convolutional Neural Networks (RCNN) to propose a novel end-to-end monocular VO framework. RCNN enables automatic feature representation learning for VO via CNN while implicitly modeling motion and data association via recurrent neural networks. Experiments on KITTI' s VO dataset showed performance comparable to state-of-the-art VO methods, though they emphasized DL-based VO complements rather than replaces traditional geometry-based VO. Recent work[79] built an autoencoder deep network to learn a nonlinear latent space representation of optical flow, jointly trained with another neural network for ego-motion estimation, enabling more robust flow descriptions and accurate motion estimation through mutual learning.

Compared to geometry-based VO, DL-based VO eliminates complex geometric motion modeling and even camera calibration parameters and scale factors. Its accuracy and robustness depend on neural network design and whether training data covers test scene variations. Currently, DL-based VO remains in its infancy, with suboptimal performance when test scenes differ significantly from

training data. Existing methods exhibit diverse network architectures, and their robustness (generalization ability) across various scenes needs improvement.

4 Algorithm Evaluation

Comparing VO system performance requires testing on identical image sequences. Several institutions provide public datasets for this purpose. We introduce three popular datasets: KITTI[80], Tsukuba[81], and TUM[82]. summarizes their basic information.

** Basic Information of Three VO Datasets**

Dataset	Camera Type	Scene	URL
KITTI	Stereo	Urban/Natural	http://www.cvlibs.net/datasets/kitti/eva
Tsukuba	Stereo	Synthetic	http://www.cvlab.cs.tsukuba.ac.jp/datas
TUM	RGB-D	Indoor	http://vision.in.tum.de/data/datasets/rg dataset

KITTI is an evaluation platform jointly created by Karlsruhe Institute of Technology and Toyota Technological Institute. Its image sequences were captured by a driving car in urban and natural environments, with diverse speeds, lighting conditions, and trajectory types. The VO module includes 22 stereo sequences (00-10 for training with ground truth, 11-21 for testing without public ground truth). KITTI provides evaluation metrics: Average Translation Error (ATE) and Average Rotation Error (ARE). [Figure 7: see original paper] shows translation and rotation errors versus path length for several popular VO systems, with quantitative results in .

Tsukuba is a photorealistic synthetic dataset—a 1-minute video with 1800 stereo pairs providing ground truth disparity and occlusion maps, plus per-frame 3D camera position and orientation. Unlike real datasets relying on GPS/IMU, synthetic datasets provide noise-free ground truth.

TUM (Technical University of Munich) provides multiple datasets for RGB-D and monocular cameras, with the RGB-D dataset being most commonly used. It contains 39 indoor sequences covering various scenes and camera motions, mostly captured by handheld Kinect cameras performing unconstrained 6-DOF motion, with some sequences from robot-mounted Kinect. Scenes are categorized by structure, texture, and dynamic objects. Ground truth from external motion capture systems is provided, along with evaluation tools. TUM metrics include Relative Pose Error (RPE) and Absolute Trajectory Error (ATE).

5 Conclusion

This paper reviewed VO technologies in conjunction with advanced VO systems. VO uses cameras instead of traditional sensors, offering lower cost. It requires no prior scene or motion information, avoids data errors from encoder inaccuracies or sensor degradation, and is unaffected by wheel slippage on uneven terrain. VO has been successfully applied in land, aerial, and underwater mobile robots, as well as in automotive and consumer electronics. Nevertheless, VO systems face limitations: lack of image texture, image blur from fast camera motion, and varying illumination and imaging conditions all degrade pose estimation accuracy. The greatest challenge is maintaining stability during long-distance outdoor operation.

Based on these challenges and recent literature, future research trends may include:

a) Diverse camera types. Beyond monocular, stereo, and RGB-D cameras, other types are emerging: omnidirectional[84], fisheye[85], and catadioptric cameras[86]. These suit different scenarios—Zhang et al.[87] experimentally showed wide field-of-view cameras suit indoor/small spaces while narrow field-of-view cameras suit outdoor/large environments. Selecting appropriate camera types based on scene relationships can enhance VO adaptability.

b) High-performance feature detection and descriptors. Recent rapid development in feature algorithms has yielded new methods like binary descriptor NESTED[35] showing excellent outlier rejection, and Desai et al.'s[88] SYBA descriptor effectively reducing drift. Utilizing higher-level image information like edges can reduce feature dependency—LSD-SLAM already exploits edge information. A promising direction combines point and edge features for better low-texture scene handling.

c) Leveraging visual moving object detection. Moving objects are a major outlier source, and their removal is crucial for VO. Visual moving object detection has achieved substantial results; integrating these with VO to exclude moving points is a viable approach. This is important for improving feature set quality and motion estimation accuracy.

d) Deep learning-based VO. DL-based VO eliminates complex geometric modeling, learning features and mapping poses directly from image sequences. Improving robustness across diverse scenes is key. Current methods show large architectural variations and limited generalization. Enhancing network generalization capabilities is essential for this emerging direction.

References

[1] Scaramuzza D, Fraundorfer F. Visual odometry: part I: the first 30 years and fundamentals [J]. IEEE Robotics & Automation Magazine, 2011, 18 (4):

80-92.

[2] Moravec H P. Obstacle avoidance and navigation in the real world by a seeing robot rover [D]. Palo Alto: Stanford University, 1980.

[3] Matthies L, Shafer S. Error modeling in stereo navigation [J]. IEEE Journal on Robotics and Automation, 1987, 3 (3): 239-248.

[4] Milella A, Siegwart R. Stereo-based ego-motion estimation using pixel tracking and iterative closest point [C]// Proc of the 4th IEEE International Conference on Computer Vision Systems. 2006: 21-21.

[5] Nister D, Naroditsky O, Bergen J, et al. Visual odometry [C]// Proc of IEEE Conference on Computer Vision and Pattern Recognition. 2004: 652-659.

[6] Patra S, Aggarwal H, Arora H, et al. Computing Egomotion with Local Loop Closures for Egocentric Videos [C]// Proc of IEEE Winter Conference on Applications of Computer Vision. 2017: 454-463.

[7] Wang S, Zhang Y, Zhu F. Monocular visual SLAM algorithm for autonomous vessel sailing in harbor area [C]// Proc of the 25th Saint Petersburg International Conference on Integrated Navigation Systems. 2018: 1-7.

[8] He Hong, Jia Yunhui, Sun Lei. Simultaneous location and map construction based on RBPF-SLAM algorithm [C]// Proc of Chinese Control And Decision Conference. 2018: 4907-4910.

[9] 江燕华, 熊光明, 姜岩, 等. 智能车辆视觉里程计算法研究进展 [J]. 兵工学报, 2012, 33 (2): 214-220. (Jiang Yanhua, Xiong Guangming, Jiang Yan, et al. A review of visual odometry for intelligent vehicle [J]. Acta Armamentarii, 2012, 33 (2): 214-220.)

[10] 李宇波, 朱效洲, 卢惠民, 等. 视觉里程计技术综述 [J]. 计算机应用研究, 2012, 29 (8): 2801-2805, 2810. (Li Yubo, Zhu Xiaozhou, Lu Huimin, et al. Review on visual odometry technology [J]. Application Research of Computers, 2012, 29 (8): 2801-2805, 2810.)

[11] Fraundorfer F, Scaramuzza D. Visual odometry, part II: matching, robustness, optimization, and applications [J]. IEEE Robotics & Automation Magazine, 2012, 19 (2): 78-90.

[12] 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.

[13] Klein G, Murray D. Parallel tracking and mapping for small AR workspaces [C]// Proc of the 6th IEEE and ACM International Symposium on Mixed and Augmented Reality. 2007: 225-234.

[14] Mur-Artal R, Montiel JMM, Tard JD. ORB-SLAM: a versatile and accurate monocular SLAM system [J]. IEEE Trans on Robotics, 2015, 31 (5): 1147-1163.

[15] Mur-Artal R, Tardós JD. ORB-SLAM2: an open-source SLAM system for monocular, stereo, and RGB-D cameras [J]. IEEE Trans on Robotics, 2017, 33

(5): 1255-1262.

[16] Geiger A, Ziegler J, Stiller C. StereoScan: dense 3D reconstruction in real-time [C]// Proc of IEEE Intelligent Vehicles Symposium. 2011: 963-968.

[17] 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.

[18] Lu Wei, Xiang Zhiyu, Liu Jilin. High-performance visual odometry with two-stage local binocular BA and GPU [C]// Proc of IEEE Intelligent Vehicles Symposium. 2013: 1107-1112.

[19] 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.

[20] Zhang J, Kaess M, Singh S. Real-time depth enhanced monocular odometry [C]// Proc of IEEE//RSJ International Conference on Intelligent Robots and Systems. 2014: 4973-4980.

[21] Labbe M, Michaud F. Online global loop closure detection for large-scale multi-session graph-based SLAM [C]// Proc of IEEE//RSJ International Conference on Intelligent Robots and Systems. 2014: 2661-2666.

[22] Harris C G, Stephens M J. A combined corner and edge detector [C]// Proc of the 4th Alvey Vision Conference. 1988: 147-151.

[23] Lowe D G. Distinctive image features from scale-invariant keypoints [J]. International Journal of Computer Vision, 2004, 60 (2): 91-110.

[24] Parra I, Sotelo M A, Llorca D F, et al. Robust visual odometry for vehicle localization in urban environments [J]. Robotica, 2010, 28 (3): 441-452.

[25] Tardif J P, Pavlidis Y, Daniilidis K. Monocular visual odometry in urban environments using an omnidirectional camera [C]// Proc of IEEE//RSJ International Conference on Intelligent Robots and Systems. 2008: 2531-2536.

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

[27] Leutenegger S, Chli M, Siegwart RY. BRISK: binary robust invariant scalable keypoints [C]// Proc of International Conference on Computer Vision. 2011: 2548-2555.

[28] Alcantarilla P F, Bartoli A, Davison A J. KAZE features [C]// Proc of the 12th European Conference on Computer Vision. 2012: 214-227.

[29] Alcantarilla P F, Nuevo J, Bartoli A. Fast explicit diffusion for accelerated features in nonlinear scale spaces [C]// Proc of International Conference on Control, Automation and Systems. 2013: 704-709.

[30] Bay H, Tuytelaars T, Van Gool L. SURF: speeded up robust features [C]// Proc of the 9th European Conference on Computer Vision. 2006: 404-417.

[31] Yan K, Sukthankar R. PCA-SIFT: a more distinctive representation for local image descriptors [C]// Proc of IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 2004: 506-513.

[32] Tola E, Lepetit V, Fua P. DAISY: an efficient dense descriptor applied to wide-baseline stereo [J]. IEEE Trans on Pattern Analysis and Machine Intelligence, 2010, 32 (5): 815-830.

[33] Calonder M, Lepetit V, Strecha C, et al. BRIEF: binary robust independent elementary features [C]// Proc of the 11th European Conference on Computer Vision. 2010: 778-792.

[34] Vandergheynst P, Ortiz R, Alahi A. FREAK: fast retina keypoint [C]// Proc of IEEE Conference on Computer Vision and Pattern Recognition. 2012: 510-517.

[35] Byrne J, Shi J. Nested shape descriptors [C]// Proc of IEEE International Conference on Computer Vision. 2013: 1201-1208.

[36] Hartmann J, Klussendorff JH, Maehle E. A comparison of feature descriptors for visual SLAM [C]// Proc of European Conference on Mobile Robots. 2013: 56-61.

[37] Khan N, Mccane B, Mills S. Better than SIFT? [J]. Machine Vision and Applications, 2015, 26 (6): 819-836.

[38] Beis J S, Lowe D G. Shape indexing using approximate nearest-neighbour search in high-dimensional spaces [C]// Proc of IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 1997: 1000-1006.

[39] Liu T, Moore A W, Gray A, et al. An investigation of practical approximate nearest neighbor algorithms [C]// Proc of International Conference on Neural Information Processing Systems. 2004: 825-832.

[40] 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.

[41] Silpa-Anan C, Hartley R. Optimised KD-trees for fast image descriptor matching [C]// Proc of IEEE Conference on Computer Vision and Pattern Recognition. 2008: 1-8.

[42] Galvez-Lopez 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.

[43] 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.

[44] Maimone M, Cheng Y, Matthies L. Two years of visual odometry on the mars exploration rovers [J]. Journal of Field Robotics, 2007, 24 (3): 169-186.

[45] Davison A J. Real-time simultaneous localisation and mapping with a single camera [C]// Proc of the 9th IEEE International Conference on Computer Vision. 2003: 1403-1410.

[46] Fischler M A, Bolles R C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography [M]// Readings in Computer Vision. San Francisco: Morgan Kaufmann, 1987: 726-740.

[47] Deigmoeller J, Eggert J. Stereo visual odometry without temporal filtering [C]// Proc of the 38th German Conference on Pattern Recognition. 2016: 19-31.

[48] Pretto A, Menegatti E, Bennewitz M, et al. A visual odometry framework robust to motion blur [C]// Proc of IEEE International Conference on Robotics and Automation. 2009: 2250-2257.

[49] Chum O, Matas J. Matching with PROSAC-progressive sample consensus [C]// Proc of IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 2005: 220-226.

[50] Rusu RB, Blodow N, Beetz M. Fast point feature histograms (FPFH) for 3D registration [C]// Proc of IEEE International Conference on Robotics and Automation. 2009: 3212-3217.

[51] Raguram R, Frahm J M, Pollefeys M. A comparative analysis of RANSAC techniques leading to adaptive real-time random sample consensus [C]// Proc of the 10th European Conference on Computer Vision. 2008: 500-513.

[52] Kitt B, Geiger A, Lategahn H. Visual odometry based on stereo image sequences with RANSAC-based outlier rejection scheme [C]// Proc of IEEE Intelligent Vehicles Symposium. 2010: 486-492.

[53] Tan Wei, Liu Haomin, Dong Zilong, et al. Robust monocular SLAM in dynamic environments [C]// Proc of IEEE International Symposium on Mixed and Augmented Reality. 2013: 209-218.

[54] Musleh B, Martin D, Escalera Adl, et al. Visual ego motion estimation in urban environments based on U-V disparity [C]// Proc of IEEE Intelligent Vehicles Symposium. 2012: 444-449.

[55] Min Qin, Huang Yingping. Motion detection using binocular image flow in dynamic scenes [J]. EURASIP Journal on Advances in Signal Processing, 2016, 2016 (1): 1-12.

[56] Ci Wenyan, Huang Yingping. A robust method for ego-motion estimation in urban environment using stereo camera [J]. Sensors (Basel), 2016, 16 (10): 1-12.

[57] Gong Piliang, Zhang Qifeng, Zhang Aiqun. Stereo vision based motion estimation for underwater vehicles [C]// Proc of the 2nd International Conference on Intelligent Computation Technology and Automation. 2009: 567-570.

[58] Horn B K P. Closed-form solution of absolute orientation using unit quaternions [J]. Journal of the Optical Society of America A, 1987, 4 (4): 629-642.

[59] Moreno-Noguer F, Lepetit V, Fua P. Accurate non-iterative o(n) solution to the PnP problem [C]// Proc of the 11th IEEE International Conference on Computer Vision. 2007: 1-8.

[60] Haralick B M, Lee C N, Ottenberg K, et al. Review and analysis of solutions of the three point perspective pose estimation problem [J]. International Journal of Computer Vision, 1994, 13 (3): 331-356.

[61] Nister D. An efficient solution to the five-point relative pose problem [J]. IEEE Trans on Pattern Analysis and Machine Intelligence, 2004, 26 (6): 756-770.

[62] Stewenius H, Engels C, Nister D. Recent developments on direct relative orientation [J]. ISPRS Journal of Photogrammetry and Remote Sensing, 2006, 60 (4): 284-294.

[63] Pizarro O, Eustice R, Singh H. Relative pose estimation for instrumented, calibrated imaging platforms [C]// Proc of Digital Image Computing Techniques & Applications. 2003: 601-612.

[64] Hartley R, Zisserman A. Multiple view geometry in computer vision [M]// Cambridge: Cambridge University Press, 2003: 1865-1872.

[65] Mirabdollah M H, Mertsching B. Single camera motion estimation: modification of the 8-point method [C]// Proc of the 6th International Conference on Intelligent Robotics and Applications. 2013: 117-128.

[66] Fraundorfer F, Tanskanen P, Pollefeys M. A minimal case solution to the calibrated relative pose problem for the case of two known orientation angles [C]// Proc of the 11th European Conference on Computer Vision. 2010: 269-282.

[67] Scaramuzza D. 1-point-RANSAC structure from motion for vehicle-mounted cameras by exploiting non-holonomic constraints [J]. International Journal of Computer Vision, 2011, 95 (1): 74-85.

[68] 李海滨, 单文军, 刘彬. 双目立体视觉测距系统误差模型的研究 [J]. 光学技术, 2006, 32 (1): 24-26. (Li Haibin, Shan Wenjun, Liu Bin. Research of error-model on two eyes stereoscopic measurement system [J]. Optical Technique, 2006, 32 (1): 24-26.)

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

[70] Thrun S, Koller D, Ghahramani Z, et al. Simultaneous Mapping and Localization with Sparse Extended Information Filters: Theory and Initial Results

[J]. Springer Tracts in Advanced Robotics, 2004, 7 (1): 363-380.

[71] Bosse M, Newman P, Leonard J, et al. An Atlas framework for scalable mapping [C]// Proc of IEEE International Conference on Robotics and Automation. 2003: 1899-1906.

[72] Paz L M, PiniÉs P, TardÓs J D, et al. Large-Scale 6-DOF SLAM With Stereo-in-Hand [J]. IEEE Trans on Robotics, 2008, 24 (5): 946-957.

[73] Engel J, Koltun V, Cremers D. Direct sparse odometry [J]. IEEE Trans on Pattern Analysis and Machine Intelligence, 2017: 1-14.

[74] Leutenegger S, Lynen S, Bosse M, et al. Keyframe-based visual-inertial odometry using nonlinear optimization [J]. International Journal of Robotics Research, 2015, 34 (3): 314-334.

[75] Konda K, Memisevic R. Learning Visual Odometry with a Convolutional Network [C]// Proc of International Conference on Computer Vision Theory and Applications. 2015: 486-490.

[76] Kendall A, Grimes M, Cipolla R. Convolutional networks for real-time 6-DOF camera relocalization [J]. Education for Information, 2015, 31: 125-132.

[77] Costante G, Mancini M, Valigi P, et al. Exploring representation learning with cnns for frame-to-frame ego-motion estimation [J]. IEEE Robotics and Automation Letters, 2016, 1 (1): 18-25.

[78] Wang S, Clark R, Wen H, et al. DeepVO: towards end-to-end visual odometry with deep recurrent convolutional neural networks [C]// Proc of IEEE International Conference on Robotics and Automation. 2017: 2043-2050.

[79] Costante G, Ciarfuglia TA. LS-VO: learning dense optical subspace for robust visual odometry estimation [J]. IEEE Robostics and Automation Letters, 2018, 3 (3): 1735-1742.

[80] Geiger A, Lenz P, Urtasun R. Are we ready for autonomous driving? The KITTI vision benchmark suite [C]// Proc of IEEE Conference on Computer Vision and Pattern Recognition. 2012: 3354-3361.

[81] Peris M, Martull S, Maki A, et al. Towards a simulation driven stereo vision system [C]// Proc of the 21st International Conference on Pattern Recognition. 2012: 1038-1041.

[82] Sturm J, Engelhard N, Endres F, et al. A benchmark for the evaluation of RGB-D SLAM systems [C]// Proc of IEEE//RSJ International Conference on Intelligent Robots and Systems. 2012: 573-580.

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

[84] Reich S, Seer M, Berscheid L, et al. Omnidirectional Visual Odometry for Flying Robots using Low-power Hardware [C]// Proc of International Conference on Computer Vision Theory and Applications. 2018: 499-507.

[85] Matsuki H, Stumberg Lv, Usenko V, et al. Omnidirectional DSO: direct sparse odometry with fisheye cameras [J]. IEEE Robotics and Automation Letters, 2018, 3 (4): 3693-3700.

[86] Ilizirov G, Filin S. Pose Estimation and Mapping Using Catadioptric Cameras with Spherical Mirrors [J]. International Archives of the Photogrammetry Remote Sensing & S, 2016, XLI-B3: 43-47.

[87] Zhang Z, Rebecq H, Forster C, et al. Benefit of large field-of-view cameras for visual odometry [C]// Proc of IEEE International Conference on Robotics and Automation. 2016: 801-808.

[88] Desai A, Lee DJ. Visual odometry drift reduction using SYBA descriptor and feature transformation [J]. IEEE Trans on Intelligent Transportation Systems, 2016, 17 (7): 1839-1851.

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