Visual Odometry Integrated Semantic Constraints towards Autonomous Driving

Abstract

Robust data association is a core problem of visual odometry, where image-to-image correspondences provide constraints for camera pose and map estimation. Current state-of-the-art visual semantic odometry uses local map points semantics, building semantic residuals associated with all classes to realize medium-term tracking of points. Considering the problem of inefficient semantic data associations and redundant semantic observation likelihood model in the visual semantic odometry, we propose a visual odometry, Local Semantic Odometry (LVSO), which is integrated with medium-term semantic constraints based on local nearest neighbor distance model. Firstly, the ERFNet model is introduced to predict semantic information, and the local nearest neighbor distance function based on breadth-first search in adaptive distance is used to establish a semantic observation likelihood model to determine the functional form of the semantic residual; then according to features of medium-term constraints, we establish a medium-term sliding window to manage the semantic map points which is used to build data associations with the latest key frame. Considering semantic segmentation reliability and under-constrained semantic residuals, a graph optimization model integrating semantic residuals and reprojection residuals is established. The algorithm is verified on the KITTI dataset. The results show that the absolute pose estimation error of LVSO is reduced by 11.37% and the relative pose estimation error is reduced by 1.67%, which improves the pose estimation accuracy of the semantic odometry and reduces drift in the pose.

Key words: autonomous driving; semantic segmentation; visual odometry

Introduction

Perception and localization is key technology for autonomous driving, and an important prerequisite for ensuring the movement of intelligent cars[1]. Vision sensors are widely used for local localization of intelligent vehicles due to their small size, low cost, abundant data and flexible installation[2]. This localization method called visual odometry is to obtain the image sequence of the environment through the on-board camera and estimate the transformation between adjacent images, so as to obtain the pose information of the vehicle. It has played an important role in the localization and navigation system[3] (especially in the GPS-free or weak GPS environment).

At this stage, the visual odometry mainly consists of feature extraction, data association, and pose estimation. The traditional visual odometry follows the framework of Nister et al.[4], that is, extracts features according to artificial features or uses direct methods[5], and estimates the pose of the camera according to the specific pixel geometric relationship between adjacent frames. However, the above low-level features are based on the grayscale of pixel, which changes a lot with distance[6]. Its data association can only be maintained in a short distance called short-term data association. Moreover, features may be too unstable to correctly match and establish reliable associations in the scene with large changes in illumination condition or lack of texture, resulting in an increase of pose estimation error between frames, which makes the localization of intelligent vehicles drift.

Semantic features are high-level features obtained by inference of low-level features[3]. Compared with low-level features, they have less dependence on illumination conditions and texture information, and can remain stable over a long distance. Therefore, it is of great significance to fuse semantic features into visual odometry to improve performance[7]. Li et al.[8] designed a visual odometry combining semantic segmentation and successfully constructed a semi-dense 3D semantic map via a multiple-view monocular camera. The research mainly focuses on improving high-level perception of environment to guide robots to complete complex tasks in semantic maps. Liang et al.[9] proposed a VO framework where feature selection algorithm selected the features of information-rich regions in the image to construct a visual odometry system by combining visual saliency maps and semantic segmentation results, and confirmed the robustness of the odometry in the case of a few feature points. Reddy et al.[10] employed a multilayer dense CRF tool to perform motion segmentation and object class labeling of images, distinguished static features from dynamic features, and tracked static features to improve the accuracy of localization in dynamic scenes. The above research enhances the robustness of visual odometry through specific feature selection mechanism. However, the algorithm in this paper fusing semantic features mainly aims to improve the accuracy of localization.

In research on fusing semantic features to improve localization accuracy, Frost et al.[11] added the prior scale information of object detected by neural network as additional measurement to bundle adjustment to guide the scale estimation of the monocular odometry and reduced the scale drift of the system. Bowman et al.[12] modelled the sensor state and the position information of semantic markers into an optimization problem, which integrated scale information, semantic information and data association, and solved the pose optimization problem under discrete data association through expectation maximization (EM) algorithm. Compared with this paper aiming to improve the accuracy of tracking in stereo system, the former research focuses on reducing scale drift in monocular system, and the latter research focuses on improving the accuracy of the relocalization and loop closure. Lianos et al.[13] proposed a new Visual Semantic Odometry (VSO) framework, which used semantic segmentation technology to obtain semantic information, and used the consistency of semantic labels to associate medium-term data into the odometry. The objective function, which used distance transformation and reprojection model to integrate the edges of segmentation results as residuals into pose and map optimization, improved the translation drift problem of the system. The semantic observation likelihood model of the algorithm is stable and reliable, but contains insignificant semantic constraints and the efficiency of observation likelihood model can be further improved. The algorithm selected map points in key frames with several intervals to establish semantic data associations, and it results in building redundant residuals similar to reprojection residuals, which weakens the effect of medium-term constraints, so the pertinence of data selection strategies needs to be further improved. Therefore, inspired by [13], this paper aims to study how to establish more effective medium-term constraints to improve localization accuracy in visual semantic odometry.

To this end, this paper designs a new visual semantic odometry based on VSO: Local Semantic Odometry (LVSO). A local neighbor projection model is used to establish semantic constraints to avoid adding inefficient residuals to the graph optimization model. The medium-term key frame sliding window is designed to manage the semantic map points, and the data associations for medium-term constraints are established with key frames, improving the localization accuracy of the odometry, and reducing pose drift.

In the rest of this paper, the structure is as follows. System Architecture section presents the framework of this whole odometry system. Semantic Observation Likelihood Model, Graph Optimization Model and Medium-term Data Associations sections derivate the semantic observation likelihood model in details and propose a graph optimization model based medium-term sliding window. Subsequently, Experiments section provides qualitative and quantitative results of

performance of LVSO on KITTI dataset[14] to demonstrate the effectiveness and accuracy of the system. Finally, a brief conclusion and discussion are given in Summary section.

System Architecture

In order to realize the localization function, the visual odometry extracts stable features from continuous images, establishes data associations through the similarity measure and establishes an optimization problem to estimate the pose according to the observation likelihood of the associated data[14]. Therefore, this paper expounds LVSO algorithm from the above three aspects. Figure 1 shows the overall system architecture of LVSO.

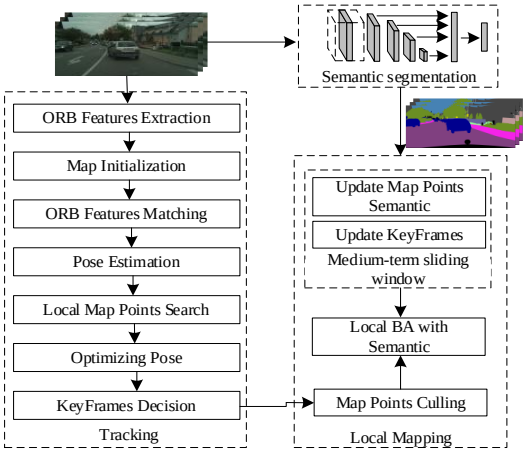


Figure 1. LVSO system architecture

LVSO includes three threads: semantic segmentation, tracking and local mapping. The semantic segmentation thread predicts the semantic by lightweight neural network ERFNet[16] to extract semantic features and passes the features to the local mapping thread through ROS (Robot Operating System).

The tracking thread uses a quad-tree structure to extract ORB[17] (Oriented FAST and Rotated BRIEF) features uniformly in the image pyramid, and matches the feature points in the stereo image in the initial key frame. It initializes the map points through triangulation[17] and establishes the initial map. Then it calculates the initial camera pose from the feature points between continuous frames. In order to further improve the pose accuracy, the system matches the local map points with the feature points in the current frame, establishes appropriate data association and uses the Bundle Adjustment(BA) method to minimize the reprojection error. It realizes short-term pose tracking and positioning between frames. Then it selects whether to establish a keyframe according to the observation quality of current frame and the state of system.

In order to trade off real-time performance between localization accuracy, the system uses a multi-threaded parallel structure to build a local map. In local map, the image frame shares the same map point with the current keyframe defined as the local keyframe. The map point in the local keyframe is close to the current frame, which is suitable for short-term constraints. Therefore, the map point is called local map point. In order to keep the number of data associations stable, the local mapping thread generates new map points according to the key frame feature points passed by the tracking thread, and adds reprojection error edges to the local BA graph optimization model. In addition, LVSO uses the local nearest neighbor distance model to model semantic feature observations and establishes medium-term constraints between medium-term map points and current keyframes through maximum likelihood estimation. After eliminated low-quality error edges and filtered by saturated linear kernel function to limit mismatches, they are added to the local BA graph optimization model to optimize the pose of the keyframe. Taking into account the pixel offset of the neural network prediction, the prediction error is integrated into the information matrix. Finally, the system projects the map points onto the semantic segmentation map using the optimized pose, and updates the class weights of the medium-term map points using the observed likelihoods of each class.

Semantic Observation Likelihood Model

From a probabilistic point of view, the pose estimation problem can be explained as

$$\begin{cases} \boldsymbol{z} = h(\boldsymbol{x}, \boldsymbol{y}) \\ \boldsymbol{x}^*_{\mathrm{MLE}} = \arg\max P(\boldsymbol{z} \mid \boldsymbol{x}, \boldsymbol{y}) \end{cases} \tag{1}$$

where $\boldsymbol{x}$ represents the pose to be estimated, $\boldsymbol{y}$ represents the feature in the map, $\boldsymbol{z}$ represents the observation data associating the map feature and the pose, and $h(\boldsymbol{x}, \boldsymbol{y})$ represents the observation equation of the current pose to the feature. The pose estimation problem can be modeled as the maximum likelihood estimation (Maximize Likelihood Estimation, MLE) problem of $\boldsymbol{x}$. Generally, the error function can be established by the least square method to form a nonlinear optimization problem[19].

$$\begin{cases} \boldsymbol{e} = \boldsymbol{z} - h(\boldsymbol{x}, \boldsymbol{y}) \\ \boldsymbol{x}^* = \arg\min\left(\sum_k \sum_j \boldsymbol{e}_{k,j}^T \boldsymbol{Q}_{k,j} \boldsymbol{e}_{k,j}\right) \end{cases} \tag{2}$$

Here, k represents the index of keyframe, j represents the index of map point feature and $\boldsymbol{Q}_{k,j}$ represents the information matrix of the observation. Semantic features are represented by classes, and the error between classes cannot be directly expressed by mathematical equation, so proper mathematical modeling of observation equation is required.

First, define the semantic observation likelihood model as: $p(S_k | Z_i = c, X_i, T_k)$, where k represents the keyframe index, i represents the map point index, vehicle pose T_k , map point location X_i and map point class Z_i , and the semantic segmentation S_k of the current frame . Based on the position of the map point and the vehicle pose, the projected pixel coordinates can be obtained, and the observation likelihood marginalizes the pixel coordinates and can be obtained by the Bayesian formula

$$\begin{aligned} & p\left(S_k \mid Z_i = c, X_i, T_k\right) \\ &= \sum_{u_{i,k}} p\left(S_k, u_{i,k} \mid Z_i = c, X_i, T_k\right) \\ &= \sum_{u_{i,k}} p\left(S_k \mid u_{i,k}, Z_i = c, X_i, T_k\right) p\left(u_{i,k} \mid Z_i = c, X_i, T_k\right) \end{aligned} \tag{3}$$

where, $u_{i,k}$ represents the projected pixel coordinates. There are two conditional independence properties in Eq.(3)

$$\begin{cases} S_k \perp X_i, T_k \mid u_{i,k}, Z_i \\ u_{i,k} \perp Z_i \mid T_k, X_i \end{cases} \tag{4}$$

The above two independences can be easily obtained. It can be seen from Eq.(4) that the projected pixel point and the map point class constitute the Markov blanket of the observation , that is, when these two variables are used as condition variables, the observation $S_{i,k}$ is independent of other variables. In addition, the projected pixel $u_{i,k}$ is determined by the map point location and the vehicle pose. Therefore, Eq.(3) can be transformed into

$$\begin{aligned} & p\left(S_k \mid Z_i = c, X_i, T_k\right) \\ &= \sum_{u_{i,k}} p\left(S_k \mid u_{i,k}, Z_i = c\right) p\left(u_{i,k} \mid X_i, T_k\right) \end{aligned} \tag{5}$$

The first term in Eq.(5) is the probability that the semantic segmentation is S_k given pixel coordinates and semantic classes. Assuming that the prior probability of each class is the same, according to the Bayesian formula, we can get

$$\begin{aligned} &p\left(S_k \mid u_{i,k}, Z_i = c\right) \propto p\left(u_{i,k}, Z_i = c \mid S_k\right) \\ &= p\left(u_{i,k} \mid S_k, Z_i = c\right) p\left(Z_i = c \mid S_k\right) \end{aligned} \tag{6}$$

The second term in Eq.(5) represents the reprojection error likelihood of ORB feature observations, which is often modeled as a two-dimensional Gaussian distribution. Within the acceptable error range, we assume that the projection probability is always 1 to simplify the calculation. At this point, Eq.(5) can be transformed into

$$\begin{aligned} &p\left(S_k \mid Z_i = c, X_i, T_k\right) \\ &= p\left(u_{i,k} \mid S_k, Z_i = c\right) p\left(Z_i = c \mid S_k\right) \end{aligned} \tag{7}$$

$p\left(u_{i,k} \mid S_k, Z_i = c\right)$ represents the probability that the pixel coordinate obtained by the projection of the map point i is $u_{i,k}$, when the given map point is of class c and the semantic segmentation of the k keyframe. Intuitively, $p\left(u_{i,k} \mid S_k, Z_i = c\right)$ should decrease the further $u_{i,k}$ is to its closest pixel in S_k with label c. Therefore, we use the local nearest neighbor distance to represent the probability

$$\begin{cases} p\left(u_{i,k} \mid S_k, Z_i = c\right) = \dfrac{e^{-\frac{1}{2\sigma^2} DT_k^{(c)}\left(u_{i,k}\right)^2}}{\alpha_{k,c}} \\ \alpha_{k,c} = \displaystyle\sum_{u_{i,k}} e^{-\frac{1}{2\sigma^2} DT_k^{(c)}\left(u_{i,k}\right)^2} \end{cases} \tag{8}$$

Here, σ represents the pixel offset error in the semantic segmentation, which is assumed to be a constant, showed in Figure 2. In Figure 2, the blue area represents area of the black car that neural network may predict, while the green area is the actual area of the car. The difference between two areas is what σ represents. $\alpha_{k,c}$ is a normalization factor to ensure that the sum of the probabilities of all pixels is equal to 1. Based on the credibility of semantic segmentation, the system searches for the nearest neighbor class points through the pruned breadth-first search algorithm within the adaptive block distance range of the projection point. Moreover, if the distance is further than $4.57\sqrt{\alpha_{k,c}\sigma^2}$, $p\left(u_{i,k} \mid S_k, Z_i = c\right)$ is regarded as zero to improve computing efficiency.

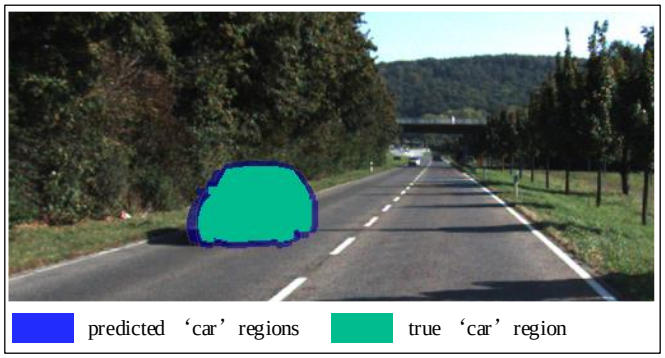


Figure 2. Pixel offset error in the semantic segmentation

The second probability in Eq.(7) is equal to the probability of occurrence of class c pixels in the semantic segmentation S_k. Ideally, this probability is equal to the ratio of the area of the corresponding class c to the area of the entire image. To facilitate computation, the system uses the Gaussian probability representation of pixels under a given semantic class

$$\begin{aligned} &p\left(Z_i = c \mid S_k\right) \\ &= \frac{\displaystyle\sum_{u_{i,k}} e^{-\frac{1}{2\sigma^2} DT_k^{(c)}\left(u_{i,k}\right)^2}}{\displaystyle\sum_{c'} \sum_{u_{i,k}} e^{-\frac{1}{2\sigma^2} DT_k^{(c')}\left(u_{i,k}\right)^2}} = \frac{\alpha_{k,c}}{\alpha_k} \end{aligned} \tag{9}$$

So far, the semantic observation likelihood model can be obtained

$$\begin{aligned} &p\left(S_k \mid Z_i = c, X_i, T_k\right) = p\left(S_k \mid u_{i,k}, Z_i = c\right) \\ &= p\left(u_{i,k} \mid S_k, Z_i = c\right) p\left(Z_i = c \mid S_k\right) \\ &= \frac{e^{-\frac{1}{2\sigma^2} DT_k^{(c)}\left(u_{i,k}\right)^2}}{\alpha_{k,c}} \cdot \frac{\alpha_{k,c}}{\alpha_k} \\ &\propto e^{-\frac{1}{2\sigma^2} DT_k^{(c)}\left(u_{i,k}\right)^2} \end{aligned} \tag{10}$$

According to Eq.(1) and (10), the pose estimation problem based on semantic observation of local neighbor distance can be established. Since exponential probability has a better mathematical form under negative logarithms, the semantic-based pose estimation problem can be expressed as

$$\begin{cases} \hat{\theta} = \arg\max\limits_{\theta}\left(P(S \mid \theta)\right) \\ \sim \arg\min\limits_{\theta}\left(-\ln P(S \mid \theta)\right) \\ = \arg\min\limits_{\theta} E_{sem}(\theta) \\ E_{sem}(\theta) = \displaystyle\sum_k \sum_i e_{sem}(k,i) \end{cases} \tag{11}$$

According to Eq.(10) and (11)

$$e_{sem}(k,i) \propto \frac{1}{\sigma^2} DT_k^{(c)}\left(\pi(T_k, X_i)\right)^2 \tag{12}$$

where the residual edge formed by the observation of the i map point for the k frame pose is mainly related to the semantic segmentation reliability and the nearest neighbor distance.

Medium-term Data Associations

Semantic features, that is, the class labels of map points, belong to high-level features that have a certain understanding of environment. This feature is hardly affected by distance, can remain stable for a long time and can be observed within a certain distance, and can establish data associations at intervals of several frames. In contrast, ORB feature points are low-level features designed based on pixel grayscale relationships in principle, and are significantly affected by distance. Even if image pyramids are used to enhance scale invariance, associations cannot be established after multi-frame intervals. Therefore, constraints established by semantic features are also called medium-term constraints. The characteristics of the two features are shown in Figure 3.

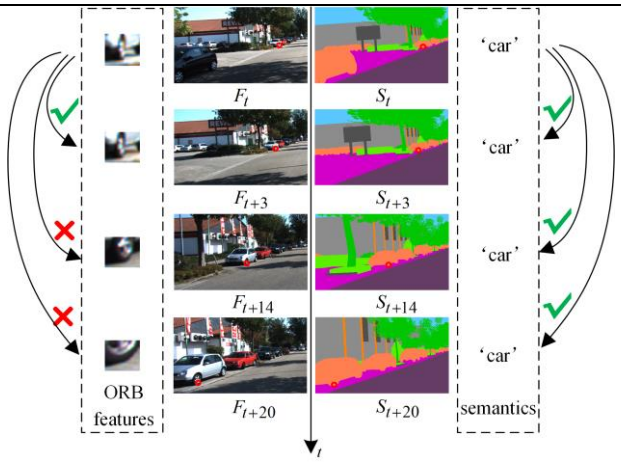


Figure 3. Characteristics of the two features

It can be seen from Figure 3 and existing research[13] that semantic features can still establish data associations after a certain time interval, thereby reducing the error accumulation of pose estimation. To this end, this paper adopts the medium-term sliding window method to establish the data association of semantic features. The system uses a certain number of keyframes before the current frame as the sliding window container for managing the semantic information of map points, but most of the map points in the keyframes with a short interval from the current frame can establish stable low-level feature constraints. Therefore, the system selects the first five frames in the sliding window container as the medium-term sliding window to establish effective medium -term constraints. The sliding window is continuously updated over time. Since the poses of these keyframes and the positions of corresponding map points have been optimized by BA at least once, the initial state of the optimization problem constructed is more accurate.

In addition, the map points in the medium-term sliding window are obtained by triangulating the ORB features, and the reliability of their positions is affected by the ORB feature matching. Moreover, the map points are limited by the observation occlusion, which may form a mismatch and reduce the accuracy of pose estimation. The system first selects map points that can be continuously observed by multiple images in the medium-term sliding window. These map points have been filtered for many times by descriptors, distances, directions and other characteristics. If the residual is too large, the map point may be occluded, and the map point will be eliminated. Finally, there is an error in the neural network prediction and the semantic class of the map point is designed as a multi-weight mode. After many optimizations, the semantic weight of the map point should tend to be a single class, so map points with scattered weight distributions are eliminated. So far, semantic observations are established between the remaining map points and current frame. Figure 4 is a schematic diagram of the data association based on the medium-term sliding window. The hollow points in the figure are the map points maintained by the keyframes in the medium-term sliding window, which are used to establish medium-term constraints with the current frame, corresponding keyframes are the keyframes in the medium-term sliding window, and the solid are local map points used to establish ORB features observation.

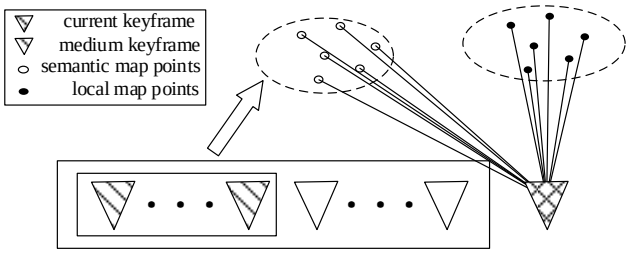


Figure 4. Medium-term data associations

Graph Optimization Model

Estimating pose only based on the Eq.(12) is an under-constrained state estimation problem and lacks structural information, that is, the residuals of map point projection in multiple positions in the semantic segmentation are the same, and the specific position cannot be determined, as shown in Figure 5.

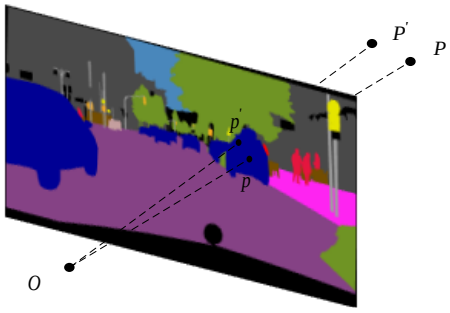


Figure 5. Under-constrained semantic observation

In Figure 5, P and P' are the two estimated positions of the same map point. Different estimation results lead to different projected point positions, but the semantic features of both are "car". Therefore, the system adopts the fusion of semantic observation and ORB feature observation to construct the following pose optimization problem

$$\{X\},\{T\} = \arg\min E_{reproj} + \lambda E_{sem} \tag{13}$$

where, E_{reproj} is the reprojection error of ORB features.

In addition, considering that there is a certain error in the prediction of the neural network, the projected pixels of map points may appear in multiple classes, and the observation likelihood needs to calculate the nearest neighbor distance for multiple classes. To this end, the system incrementally calculates the class weights of map points, which are used to weight semantic errors for each class. The class weights of map points are also represented by the Gaussian probability Eq.(10) of pixels under a given semantic class. The semantic error model considering the class weights of map points is as follows

$$\begin{cases} e_{sem}(k,i) = \sum_{c\in C} w_i^{(c)} \frac{1}{\sigma^2} DT_k^{(c)} \left(\pi(T_k, X_i)\right)^2 \\ w_i^{(c)} = \frac{1}{\alpha} \prod_{k\in\mathcal{T}_i} p\left(S_k \mid T_k, X_i, Z_i = c\right) \\ \alpha = \sum_{c\in C} \prod_{k\in\mathcal{T}_i} p\left(S_k \mid T_k, X_i, Z_i = c\right) \end{cases} \tag{14}$$

Here, α is the normalization factor to make sure that sum of weights is 1. When a keyframe is added in the sliding window, the weight of the map point is updated once. So far, the semantic error edge and the reprojection error edge have been formed. Due to the high real-time requirements of the tracking thread, the system adds two error edges to the graph optimization model optimized by the local BA. It establishes a reprojection error between local map point and the corresponding local key frame, and semantic errors between medium-term map points and latest keyframes. Due to lack of structural information and accurate location of semantic map points, the location of semantic map points in graph optimization is fixed, which acts as constraints. In order to reduce the impact of semantic feature mismatches, the system removes error edges with excessive semantic errors and optimizes the optimization in two stages. In the first stage, the Huber kernel function is set to limit the excessive error. In the second stage, we remove edges with too large error in the first stage, and no longer set the kernel function. The optimization may terminate early according to state of tracking thread. The resulting graph optimization model is shown in Figure 6.

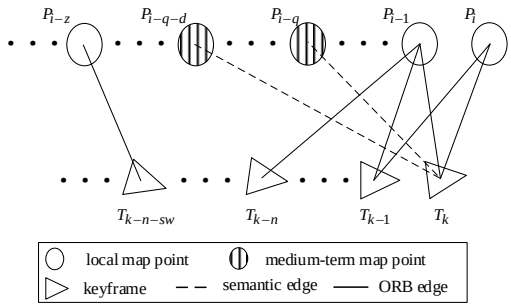


Figure 6. Graph optimization of local BA

Experiments

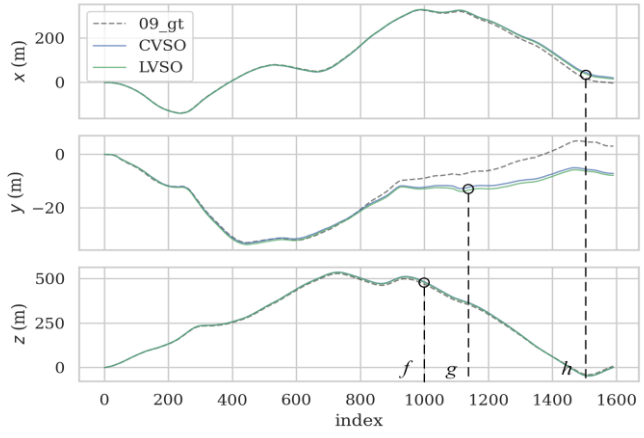
The algorithm experiment platform is configured as follows: the software environment is Ubuntu 16.04; the processor (CPU) is Intel Core I7-9750H at 2.6GHz; the RAM is 16GB; the GPU is NVIDIA GTX 1660Ti; the parallel computing framework is CUDA10.1; the deep learning framework is Pytorch; third-party libraries are Opencv 3.4.3, g2o[20], etc. Experiments were performed using the KITTI dataset[14]. The error weights in the system and the neural network prediction reliability are selected empirically in each set of experiments.

In order to verify the localization accuracy of the system, the reference system in this paper filters the local map points to establish semantic constraints, that is, the map points in the key frame before the latest frame, which is called CVSO. Referring to the standards in [21], this paper selects the root mean square error (RMSE) of the relative pose error (RPE) and the absolute trajectory error (ATE) as the evaluation standard. ATE measures the absolute deviation of the pose on the two trajectories. The experimental results are shown in Tabel 1.

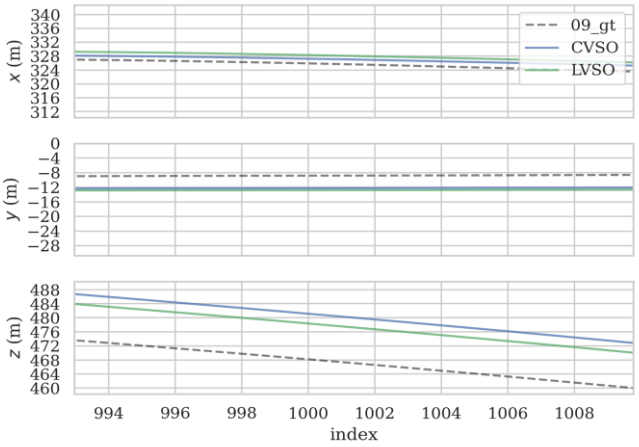
Tabel 1. ATE in the KITTI dataset

seq.	length/m	RMSE/m	
		LVSO	CVSO
00	714.26	12.816	13.341
02	5067.23	21.970	26.223
05	2205.57	6.649	5.724
06	1232.87	4.474	5.064
07	694.69	3.161	3.493
08	3222.79	15.908	19.004
09	1705.05	11.237	13.415
10	919.51	5.954	6.318

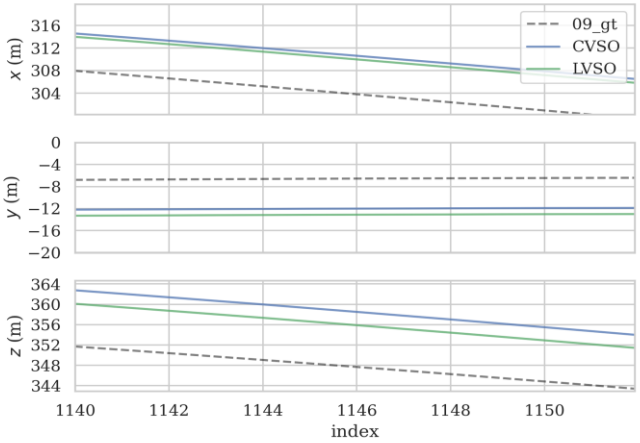
In the experiment, except for the 05 sequence, the estimation accuracy of the absolute trajectory has been improved, and the absolute trajectory error has decreased by 11.37% on average, and the maximum can reach 16.29%. Taking sequence 09 as an example, Figure 7 is the position comparison result, and the position is represented by coordinates. It can be seen that the trajectory obtained by LVSO is closer to the real trajectory.



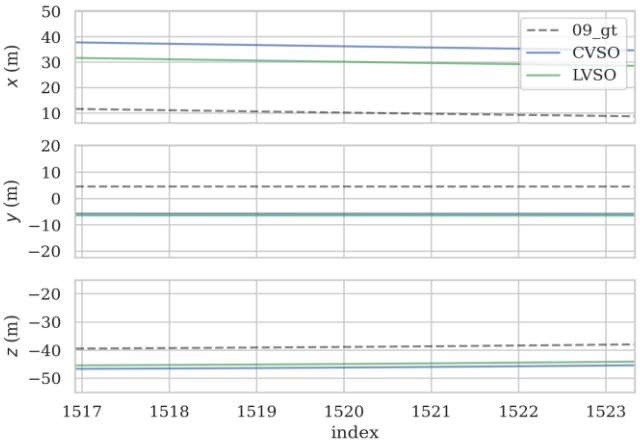
a) Trajectory estimate in xyz directions



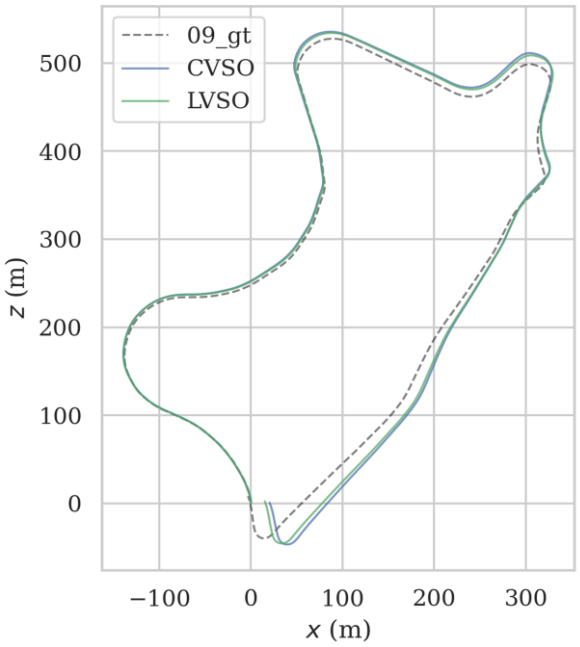
b) Trajectory estimate of *f* point in xyz directions



c) Trajectory estimate of *g* point in xyz directions



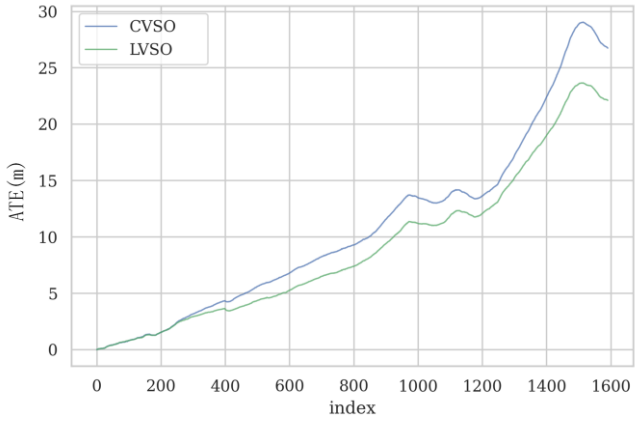
d) Trajectory estimate of *h* point in xyz directions



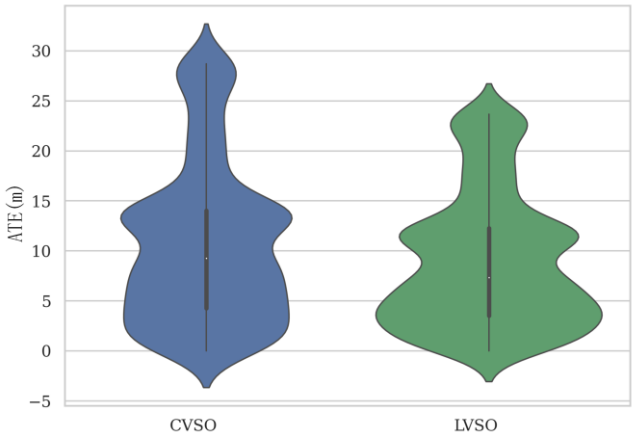
e) Overall trajectory estimation comparison

Figure 7. Comparison of trajectory estimation between LVSO and CVSO on 09 sequence

Figure 8 shows the variation of ATE with the picture frame and the overall error distribution of LVSO and CVSO on the 09 sequence. It can be seen from Figure 8 (b) that the error distribution of LVSO is more concentrated in the small error part. The upper and lower quartiles are lower than CVSO, and the positioning accuracy and stability of the system are significantly improved compared with CVSO.



a) ATE as a function of the frame index



b) Violin plot of ATE

Figure 8. ATE comparison of LVSO and CVSO on 09 sequence

The experimental results show that the data associations based on medium-term map points provide more effective semantic error edges for the graph optimization model, strengthen the medium-term constraints, and improve the accuracy of pose estimation. However, in the experiments of the 05 sequence, compared with CVSO, the localization error of LVSO increases continuously from 1500 to 2000 frames. The analysis shows that during this period, the vehicle is on an urban branch road with limited field of vision, the houses on both sides are relatively close to the vehicle, and the map points changes greatly with the observation of the driving process. Only map points at the end of the road meet requirements for medium-term map points, but the location accuracy of distant map points is limited, which leads to a decrease in the accuracy of pose estimation. On the contrary, CVSO selects the nearest generated map points, whose location and observation state are stable, the formed data association is reliable, and the pose estimation accuracy does not decrease. In Figure 9, the formal map points are stored in map observed by several key frames and the temporary map points triangulated in tracking process to enhance feature matching are not stored in map. The map points in the red box in Figure 9 are mostly selected as mid-term map points. In the follow-up, the medium-term map points can be further screened by distance to improve system stability.

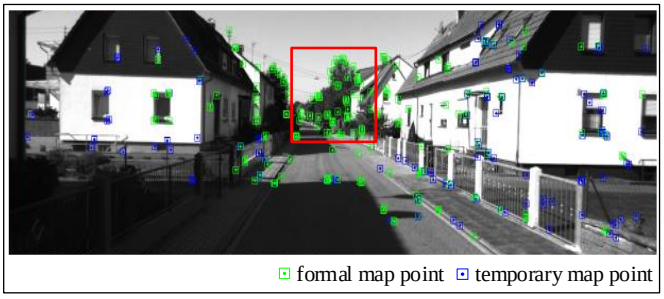


Figure 9. Medium-term map point in the 1445th image of the 05 sequence.

RPE calculates the deviation between the estimated pose change and the real change in the same distance (time) interval to obtain the relative pose error, which can be used to evaluate the pose drift of the odometry. The experiment evaluates the relative pose translation error of each 100-meter distance, and the experimental results are shown in Tabel 2.

Tabel 2 RPE in the KITTI dataset

seq.	length/m	RMSE/m	
		LVSO	CVSO
00	714.26	1.592	1.593
02	5067.23	1.702	1.699
05	2205.57	1.070	1.066
06	1232.87	1.181	1.240
07	694.69	1.064	1.090
08	3222.79	1.613	1.636
09	1705.05	1.088	1.137
10	919.51	0.988	0.998

Since semantic constraints can provide medium-term constraints to reduce odometry pose drift, the RPE of LVSO decreases by an average of 1.67% compared to CVSO, and the maximum can reach 4.76%. The use of medium-term sliding window to establish semantic constraints has a certain effect on pose drift. Taking sequence 06 as an example, the error statistics of RPE are shown in Figure 10. In the figure, the maximum and minimum errors of LVSO are much smaller than those of CVSO, and the overall error distribution is closer to zero. It can be seen that the data association strategy based on the mid-term sliding window can reduce the pose drift.

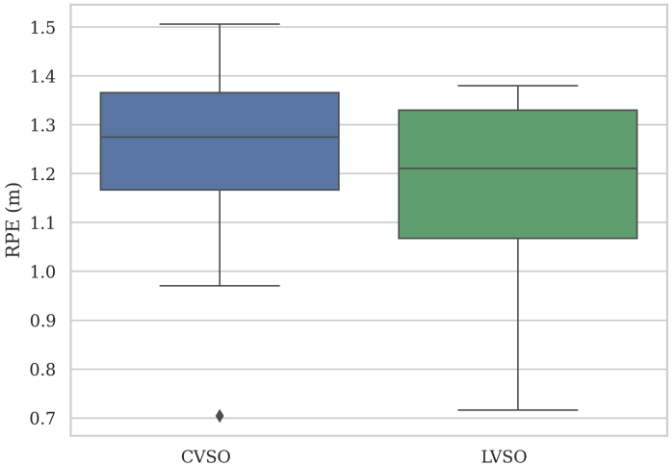


Figure 10. Boxplot of RPE

Summary

In this paper, an odometry algorithm fusing medium-term semantic is proposed. The system uses the local neighbor distance of the map point projection to represent the semantic observation and selects the map points in the medium-term sliding window to establish data associations. Then it establishes a suitable pose estimation problem through the fusion of the semantic observation error and the ORB feature observation error, thereby reducing the pose translation error. This paper verifies the effectiveness of the method on a real-world scene dataset. The drawback of our methods includes the accuracy of semantic segmentation and the adaptivity of parameters, especially the uncertainty in semantic segmentation. In the future, we expect to research on semantic segmentation network design and system parameter adaptability to various challenging scenes. We hopes to use semantic features to further improve the positioning accuracy of the odometry and reduce pose drift in odometry, so as to promote the development of intelligent automobile.

References

1. Cadena, C., Carlone, L., and Carrillo, H., et al., "Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age," *IEEE Transactions On Robotics* 32(6):1309-1332, 2016, 10.1109/TRO.2016.2624754.
2. Yousif, K., Bab-Hadiashar, A., and Hoseinnezhad, R., "An Overview to Visual Odometry and Visual SLAM: Applications to Mobile Robotics," *Intelligent Industrial Systems* 1(4):289-311, 2015, 10.1007/s40903-015-0032-7.
3. Xia, L., Cui, J., and Shen, R., et al., "A Survey of Image Semantics-Based Visual Simultaneous Localization and Mapping: Application-Oriented Solutions to Autonomous Navigation of Mobile Robots," *International Journal of Advanced Robotic Systems* 17(3):255689074, 2020, 10.1177/1729881420919185.
4. D., N., O., N., and J., B., "Visual Odometry," presented at Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004., 0027-02-20, 2004.
5. Qin, T., Li, P., and Shen, S., "VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator," *IEEE Transactions On Robotics* 34(4):1004-1020, 2018, 10.1109/TRO.2018.2853729.
6. Mur-Artal, R., and Tardos, J. D., "ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras," *IEEE Transactions On Robotics* 33(5):1255-1262, 2017, 10.1109/TRO.2017.2705103.
7. R., F. S., R., A. N., and H., S., et al., "SLAM++: Simultaneous Localisation and Mapping at the Level of Objects," presented at 2013 IEEE Conference on Computer Vision and Pattern Recognition, 2013-01-01, 2013.
8. Li, X., and Belaroussi, R., "Semi-Dense 3D Semantic Mapping From Monocular SLAM,", 2016.
9. H., L., N., J. S., and C., F., et al., "SalientDSO: Bringing Attention to Direct Sparse Odometry," *IEEE Transactions On Automation Science and Engineering* 16(4):1619-1626, 2019, 10.1109/TASE.2019.2900980.
10. N., D. R., P., S., and V., C., et al., "Dynamic Body VSLAM with Semantic Constraints," presented at 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 0028-02-20, 2015.
11. Frost, D. P., Kahler, O., and Murray, D. W., "Object-Aware Bundle Adjustment for Correcting Monocular Scale Drift," presented at 2016 IEEE International Conference on Robotics and Automation (ICRA), 2016.
12. S., L. B., N., A., and K., D., et al., "Probabilistic Data Association for Semantic SLAM," presented at 2017 IEEE International Conference on Robotics and Automation (ICRA), 0029-03-20, 2017.
13. Lianos, K., Schönberger, J. L., and Pollefeys, M., et al., "VSO: Visual Semantic Odometry" (Cham, Springer International Publishing, 2018), 246-263, 10.1007/978-3-030-01225-0_15.
14. Geiger, A., Lenz, P., and Urtasun, R., "Are we Ready for Autonomous Driving? The KITTI Vision Benchmark Suite," presented at, 2012-01-01, 2012.
15. Moravec, H. P., "Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover," *Stanford University.*, 1980.
16. Romera, E., Alvarez, J. M., and Bergasa, L. M., et al., "ERFNet: Efficient Residual Factorized ConvNet for Real-Time Semantic Segmentation," *Ieee Transactions On Intelligent Transportation Systems* 19(1):263-272, 2018, 10.1109/TITS.2017.2750080.
17. R., M., J., M. M. M., and J., D. T., "ORB-SLAM: A Versatile and Accurate Monocular SLAM System," *IEEE Transactions On Robotics* 31(5):1147-1163, 2015, 10.1109/TRO.2015.2463671.
18. Hartley, R., and Zisserman, A., "Multiple View Geometry in Computer Vision," presented at Cambridge University Press, 2000.
19. Carlone, L., "State Estimation for Robotics [Bookshelf]," *IEEE Control Systems* 39(3), 2019.
20. Kummerle, R., Grisetti, G., and Strasdat, H., et al., "G2o: A General Framework for Graph Optimization," presented at, 2011-01-01, 2011.
21. Sturm, J., Engelhard, N., and Endres, F., et al., "A Benchmark for the Evaluation of RGB-D SLAM Systems," presented at, 2012-01-01, 2012.