LMBAO: A LANDMARK MAP FOR BUNDLE ADJUSTMENT ODOMETRY IN LiDAR SLAM

Letian Zhang¹, Jinping Wang¹, Lu Jie¹, Nanjie Chen¹, Xiaojun Tan¹,*, Zhifei Duan²

¹School of Intelligent Systems Engineering, Sun Yat-sen University, Shenzhen, 518100, China,
²The Automotive Technology Center, XPeng Inc., Guangzhou, 510640, China

ABSTRACT

Existing LiDAR odometry strategies match a new scan iteratively with previous fixed-pose scans, gradually accumulating errors. Furthermore, as an effective joint optimization mechanism, bundle adjustment (BA) cannot be directly introduced into odometry due to the intensive computation of global landmarks. Therefore, this paper designs a landmark map for bundle adjustment odometry (LMBAO) in LiDAR SLAM. First, an active landmark maintenance strategy is developed to obtain a local map of limited size that enables real-time BA. Specifically, this paper keeps entire stable landmarks on the map instead of just their feature points in the sliding window and timely deletes inactive landmarks. Next, unlike visual marginalization to approximate the Gaussian distribution, and a direct and efficient marginalization strategy is performed to retain the scans outside the window to greatly simplifying the computation. Experiments show the effectiveness of LMBAO in outdoor driving.

Index Terms— Bundle adjustment odometry, SLAM, and joint optimization.

1. INTRODUCTION

Simultaneous localization and mapping (SLAM) is one of the most fundamental problems in robotic applications [1–4]. With Visual-based and LiDAR-based sensing, great efforts have been made to achieve highly accurate real-time localization. LiDAR is known as a reliable, illumination-insensitive sensor that can detect the fine details of an environment in a large area. Therefore, this paper focuses on an implementation method for LiDAR SLAM in outdoor driving.

The classic LiDAR SLAM framework [5–12] registers LiDAR scans incrementally and estimates only the pose of the current frame each time. In fact, the current frame can also improve the estimates of the historical frames, which in turn improves the estimate of the current frame. The introduction of bundle adjustment (BA), which jointly optimizes the pose of multiple frames, avoids error accumulation, and effectively lowers the drift in LiDAR SLAM [13–15].

To provide enough information for optimization, the length of the BA sliding window of map refinement is generally large (set to 20 in [13]). However, when BA is used for odometry, the computational cost is extremely limited due to the high real-time requirements, which means that the number of optimization times (equal to the length of the sliding window) is restricted (set to 8 in [15]). Thus, only information from recent frames is available in the map. The insufficient constraints for BA have implications for the applicability to odometry. LMBAO is designed to improve the utilization of prior constraints, as Fig. 1. The main contributions of our work can be summarized as follows:

1) We propose a strategy for maintaining prior constraints in local landmark maps that is independent of the sliding window. With an extremely limited capacity, the local landmark map is suitable for BA odometry to avoid cumulative errors.

2) We develop an observation count to control the lifetime of landmarks and a center drift to discard unstable landmarks, successfully avoiding occlusions and dynamic objects.

3) To speed up the computation, we perform efficient marginalization in combination with our landmark map to
fully exploit prior constraints instead of approximating the Gaussian distribution.

2. LANDMARK MAP GENERATION

2.1. Continuous Motion Model and Pose Prediction

We define the kth LiDAR scan as $S_k$ and its acquisition time within $[t_k, t_{k+1})$. The pose of $S_k$ in the coordinate system of the current scan $\{C_k\}$ is regarded as $P_k$ and the transformation matrix to the world coordinate system $\{W\}$ is $T_k \in SE(3)$.

A continuous motion model is applied to compensate for the distortion and predict the pose. Assuming that the LiDAR is constantly rotating at angular velocity $\omega \in R^3$ in $\{C_k\}$, and moving at a linear velocity $v \in R^3$ in $\{C_k\}$, as [15]. We use a rotation vector to represent $\omega$. The skew matrix of $\omega$ is defined as $[\omega]_x \in so(3)$. Then $exp : so(3) \rightarrow SO(3)$ represents the exponential map.

For any point $i$ in scan $S_k$, we can compensate the motion according to continuous motion model. Since $t_{k+1}$ is the starting time of $S_{k+1}$, the pose of next scan can be predicted as $\tilde{T}_{k+1}$ relative to $P_k$. And its state variable $\tilde{x}_{k+1} = \{\tilde{T}_{k+1}, \tilde{v}_{k+1}, \tilde{\omega}_{k+1}\}$ of $S_{k+1}$.

$$
\begin{align*}
\tilde{T}_{k+1} &= exp((t_{k+1} - t_k)\omega_k)P_k + (t_{k+1} - t_k)v_k \\
\tilde{v}_{k+1} &= exp(-(t_{k+1} - t_k)\omega_k)v_k \\
\tilde{\omega}_{k+1} &= exp(-(t_{k+1} - t_k)\omega_k)\omega_k
\end{align*}
$$

Using $\tilde{T}_{k+1}$, the feature points of $S_{k+1}$ can be accurately associated with points of previous scans in $\{W\}$, as shown in Fig. 1. The velocity and position of $S_{k+1}$ are initialised by the prediction and then optimised together several times in BA odometry. This step of pose prediction replaces the independent process of frame-by-frame position estimation in earlier systems [11–14].

2.2. Landmarks Adding to Local Map

Once a new scan arrives, feature points are first extracted from this single scan. We use the newly extracted feature points to construct global landmarks, which are spherical region. A landmark $L$ is represented by center $c_L$ and radius $r$ in types of edge or plane. The details can be found in [15].

Landmarks tracks feature points from new scans. When the new scan $S_{k+1}$ arrives, a pose prediction and motion compensation are described in Section 2.1. All landmarks in the map are also projected into $\{C_{k+1}\}$ with the predicted $\tilde{T}_{k+1}$ to accurately associate these feature points. The core idea of feature association is that a feature point in a new scan approaching a landmark is considered an observation point of the landmark and added to the landmark. Landmarks in the map continuously track points from different scans. To avoid redundancy, new landmarks are created only from feature points that are not tracked by existing landmarks.

2.3. Landmark Deletion from Local Map

To reduce the computational burden, we simultaneously delete landmarks to limit the number of points in the map. Unlike [13] and [15], which rely on the sliding window, the feature points of one scan are deleted when it slides out of the window. We adopt a more active map maintenance strategy that uses landmark active degree to distinguish landmarks and fully preserves feature points from active landmarks. The landmark active degree consists of three parts: the observation count $O_L$, the drift of the center point $c_L$ in optimization and the feature points number $N^L$.

The observation count $O_L$ is defined as $\{\omega\}$ as in (4 in this paper). Whenever a new scan arrives and $L$ successfully collects feature points in that scan, $O_L$ is increased by one; otherwise, $O_L$ is decreased by one. When $O_L$ decreases to 0, $L$ is deleted from the map. Even if $L$ still has feature points that participate in the sliding window optimization, they will be deleted directly. In this way, the constantly updating landmarks are not immediately discarded by the sliding window when they encounter unexpected obstructions, so they can always participate in the optimization.

When $L$ is created, the center point $c_L$ is computed using its feature points in $\{C_{k+1}\}$ and its global projection $c_L^0$ is initialised by the predicted $\tilde{T}_{k+1}$ and $\tilde{P}_{k+1}$. Subsequently, $S_{k+1}$ is constantly updated in the sliding window optimization. When $S_{k+1}$ gets its new pose $P_{k+1}'$ in the sliding window, the global projection of $c_L$ is recomputed as $c_L'$. The drift of $c_L$ is given by the distance between $c_L^0$ and $c_L'$. If the drift exceeds $3r$, the points collected by landmark in the next scan are far from the feature points captured at origin. Then, the landmark will be deleted as it is an unstable landmark with large drift.

Feature points number $N^L$ is designed to reduce the influence of dynamic objects, which are usually small in size. we treat some small plane landmarks as unstable landmarks whose feature points are still below the threshold $N^L < 86$ even after continuously collecting points across multiple scans, and remove them directly from the map. These three measures above effectively reduce the interference from random obstacles and the existence time of unstable landmarks.

3. BUNDLE ADJUSTMENT ODOMETRY WITH MARGINALIZATION

3.1. Residuals and Optimization Function

Each landmark contributes an observation residual to the BA optimization - a global pose optimization of the LiDAR scan in $\{W\}$. Since the pose of each scan is constantly updated during the optimization process, $p_k^j$ (the j-th point in the k-
th scan of landmark $L$ must be reprojected into $\{W\}$ as $\tilde{p}_L^i$. Then the covariance matrix $Cov$ of all points of $L$ in $\{W\}$ is

$$Cov = \frac{1}{N_L} \sum_{k=1}^{N_L} \sum_{j=1}^{N_L} (\tilde{p}_k^i - \bar{p}_L)(\tilde{p}_k^i - \bar{p}_L)^T.$$  

(2)

where $N_L$ is the points number of $L$. $\bar{p}_L$ is $\frac{1}{N_L} \sum_{k=1}^{N_L} \tilde{p}_k^i$.

For each plane and edge landmark, we construct the plane residual $r_p$ or edge residual $r_e$ as BALM [13]. And BALM proved $r_p/r_e$ can be simplified to

$$r_p = \sqrt{\lambda_1(Cov)}$$

$$r_e = \sqrt{\lambda_2(Cov) + \lambda_3(Cov)}.$$  

(3)

where $\lambda_i(Cov)$ represents the $i$th smallest eigenvalue of $Cov$.

Continuous Factor Residual is added to constrain the motion condition and avoid divergence as the pre-integration factor IMU. The residual is defined as

$$r_s(x_k, x_{k+1}) = \begin{bmatrix} R_k(t_{k+1} - t_k)v_k + p_k - p_k \\ R_k(o_k - R_{k+1}o_k + 1) \end{bmatrix}.$$  

(4)

where $R_k$ is rotation matrix of the transformation matrix $T_k$.

Global landmarks are the result of the association of feature points across multiple scans, so the position and survival time of each landmark are different. To standardize the format, each landmark is assumed to contain a total of $N_L$ points from all $K$ historical scans, and the number of points tracked by $S_k$ is $N_L$. Let $\chi = \{x_1, x_2, ..., x_K\}$ represent the set of state vectors of all historical scans, and $M_{\chi}$ and $M_{\pi}$ represent the set of all planar/edge landmarks in map, respectively. The final optimization is

$$\arg \min_{\chi} \sum_{c \in M_{\chi}} \rho(||r_c^L(\chi)||^2) + \sum_{\pi \in M_{\pi}} \rho(||r_s^L(\chi)||^2)$$

$$+ \sum_{k=1}^{K} \sum_{l=1}^{N_L} ||r_s(x_k, x_{k+1})||^2.$$  

(5)

where Huber loss $\rho(s)$ is defined as $s$ when $s \leq 1$ and as $2\sqrt{s} - 1$ when $s > 1$.

Each $r_c^L$ or $r_s^L$ contain two parts: calculation of $Cov$ as (2) and eigenvalue decomposition as (3). If the optimization formula is used directly. All feature points of the landmark are involved in (2), which correspond to the pose of all historical scans. When the optimization formula is used directly, it is the bundle adjustment of all scans which are optimized in each iteration.

Although our new map maintenance strategy can limit the number of landmarks, the cost increases and becomes unacceptable over time. Visual SLAM [16] uses the sliding window mechanism to limit the number of optimized historical scans and a marginalization strategy that approximates the Gaussian distribution. In contrast to Visual SLAM, this paper uses a direct and efficient marginalization strategy to fully store the previous landmark information.

### 3.2. Residuals Simplification with Direct Marginalization

The key problem to be solved by the marginalization is the calculation of the landmark observation residuals when a scan slides out of the window. With the sliding window(size $n$) and marginalization, we decomposing (2) of Cov calculation:

$$Cov = \frac{1}{N_L} \sum_{k=1}^{N_L} \sum_{j=1}^{N_L} (\tilde{p}_k^i - \bar{p}_L)(\tilde{p}_k^i - \bar{p}_L)^T - \frac{1}{N_L} \sum_{k=1}^{N_L} \sum_{j=1}^{N_L} (\tilde{p}_k^i)(\tilde{p}_k^i)^T - \frac{1}{N_L} \sum_{k=1}^{N_L} \sum_{j=1}^{N_L} (\tilde{p}_k^i)(\tilde{p}_k^i)^T.$$  

$$= \frac{1}{N_L} \sum_{k=1}^{K-n} \tilde{O}_k + \frac{1}{N_L} \sum_{k=n+1}^{K} \tilde{O}_k - \frac{1}{(N_L)^2} \sum_{k=1}^{K-n} \tilde{S}_k - \frac{1}{(N_L)^2} \sum_{k=n+1}^{K} \tilde{S}_k.$$  

(6)

In (6), we divide the calculation of Cov into the marginal part $O_{marg}$, $S_{marg}$ and the sliding window part $\tilde{O}_{win}$, $\tilde{S}_{win}$:

$$Cov = \frac{1}{N_L} (O_{marg} + \tilde{O}_{win}) - \frac{1}{N_L (N_L)^2} (S_{marg} + \tilde{S}_{win})(S_{marg} + \tilde{S}_{win})^T.$$  

(7)

In the sliding window mechanism, only the state variables of the scans in the sliding window $\chi_{win}$ need to be optimized, while the poses of the scans outside the sliding window $\chi_{marg}$ is fixed. That is, all its points position, is fixed in the subsequent optimization process. According to the (6), once $\chi_{marg}$ is fixed, $O_{marg}$ and $S_{marg}$ are also fixed. Thus $O_{marg}$ and $S_{marg}$ can be calculated in advance before optimization and fixed in each iterative optimization process.

The changes in $O_{marg}$ and $S_{marg}$ at each iteration correspond only to the scan sliding out of the window (the scan $S_{K-n+1}$). The $O_{marg}^{K-n+1}$ is

$$O_{marg}^{K-n+1} = O_{marg}^{K} - \tilde{O}_{K-n+1}.$$  

(8)

$\tilde{O}_{win}$ and $\tilde{S}_{win}$ contain all scans in the sliding window, and must be recalculated at each iteration. When a new scan $S_{K+1}$ is added to the sliding window, its state vector $x_{K+1}$ is initialised by the prediction with (1) and continues optimised in the window until it slides out.

Such a sliding window and a marginalization mechanism significantly reduce the number of scans in each optimization. Let $\chi_{win} = \{x_{K-n+1}, ..., x_K\}$. The final optimization is

$$\arg \min_{\chi_{win}} \sum_{c \in M_{\chi}} \rho(||r_c^L(\chi_{win})||^2) + \sum_{\pi \in M_{\pi}} \rho(||r_s^L(\chi_{win})||^2)$$

$$+ \sum_{k=1}^{K-1} ||r_s(x_k, x_{k+1})||^2.$$  

(9)

Finally, Ceras Slover library [17] and the L-M method is used to solve (9).
4. EXPERIMENTS

We conducted a series of experiments to verify the effectiveness of LMBAO with state-of-the-art LiDAR SLAM algorithms. The experiments were run on a laptop equipped with an AMD Ryzen7 5800H CPU and 16 GB RAM. The root mean square error (RMSE) of the absolute translational error (ATE) [18] is used as the evaluation index.

4.1. Evaluation on Public Datasets

In this paper, 10 sequences are selected from two public datasets UTBM [19] and ULHK [20] to evaluate LMBAO. We show odometry times of four sequences in Table 1 and Table 2. The propagation time throughout our pipeline is less than 100 ms and achieves real-time performance at 10 Hz LiDAR.

To allow a fair comparison with algorithms with a mapping step, we add a mapping step to LMBAO which is similar to Lego-loam. Compared to other methods in Table 1, our method achieves the best performance on all 10 sequences. In sequences 1-5 of the UTBM dataset, the error of LMBAO is only half that of the other two methods. The trajectories of sequences 2 and 5 are shown in Fig. 2(a)(b) with the origin and destination shown enlarged. The destination of the red LMBAO trajectory almost coincides with the origin and the whole process is closer to the ground truth. In the ULHK dataset, the sequence length is relatively short, so our accuracy has improved, but the extent is not particularly obvious.

4.2. Evaluation of BA Odometry on Campus Datasets

To validate the improvement of LMBAO in odometry, we performed ablation experiments on landmark map maintenance and marginalization using own campus dataset, in Table 2. The library loop is difficult as it passes through open streets.

LMBAO-LM, a VLOM variant, is designed only adding the marginalization mechanism as LMBAO, but without landmark map maintenance. In Fig. 2(c), after passing the small intersection A, the trajectories of Lego-loam and VLOM are obvious offset, while LMBAO matches well. Compared to Fig. 2(d)(e), LMBAO landmark map in Fig. 2(f) preserves finer structural details, such as the light pole and the ground. This demonstrates the effectiveness of our map maintenance.

5. CONCLUSIONS

This paper further develops the BA odometry for LiDAR SLAM by using an active landmark maintenance strategy to separate the map from the sliding window. The map is divided into a sliding window part involved in the joint optimization of pose and velocity, and a marginal part outside the window providing sufficient prior constraints. Experiments show the effectiveness of LMBAO in outdoor driving.
6. REFERENCES

[1] Tixiao Shan, Brendan Englot, Drew Meyers, Wei Wang, Carlo Ratti, and Daniela Rus, “Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping,” in 2020 IEEE/RSJ international conference on intelligent robots and systems (IROS). IEEE, 2020, pp. 5135–5142.

[2] Haoyang Ye, Yuying Chen, and Ming Liu, “Tightly coupled 3d lidar inertial odometry and mapping,” in 2019 International Conference on Robotics and Automation (ICRA). IEEE, 2019, pp. 3144–3150.

[3] Xingxing Zuo, Yulin Yang, Patrick Geneva, Jiajun Lv, Yong Liu, Guoquan Huang, and Marc Pollefeys, “Lic-fusion 2.0: Lidar-inertial-camera odometry with sliding-window plane-feature tracking,” in 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2020, pp. 5112–5119.

[4] Xingxing Zuo, Patrick Geneva, Woosik Lee, Yong Liu, and Guoquan Huang, “Lic-fusion: Lidar-inertial-camera odometry,” in 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2019, pp. 5848–5854.

[5] Paul J Besl and Neil D McKay, “Method for registration of 3-d shapes,” in Sensor fusion IV: control paradigms and data structures. Spie, 1992, vol. 1611, pp. 586–606.

[6] Szymon Rusinkiewicz and Marc Levoy, “Efficient variants of the icp algorithm,” in Proceedings third international conference on 3-D digital imaging and modeling. IEEE, 2001, pp. 145–152.

[7] Aleksandr Segal, Dirk Haehnel, and Sebastian Thrun, “Generalized-icp,” in Robotics: science and systems. Seattle, WA, 2009, vol. 2, p. 435.

[8] François Pomerleau, Francis Colas, Roland Siegwart, et al., “A review of point cloud registration algorithms for mobile robotics,” Foundations and Trends® in Robotics, vol. 4, no. 1, pp. 1–104, 2015.

[9] Seungpyo Hong, Heedong Ko, and Jinwook Kim, “Vicp: Velocity updating iterative closest point algorithm,” in 2010 IEEE International Conference on Robotics and Automation. IEEE, 2010, pp. 1893–1898.

[10] Patrick Geneva, Kevin Eckenhoff, Yulin Yang, and Guoquan Huang, “Lips: Lidar-inertial 3d plane slam,” in 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2018, pp. 123–130.

[11] Ji Zhang and Sanjiv Singh, “Loam: Lidar odometry and mapping in real-time.” in Robotics: Science and Systems. Berkeley, CA, 2014, vol. 2, pp. 1–9.

[12] Tixiao Shan and Brendan Englot, “Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain,” in 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2018, pp. 4758–4765.

[13] Zheng Liu and Fu Zhang, “Balm: Bundle adjustment for lidar mapping,” IEEE Robotics and Automation Letters, vol. 6, no. 2, pp. 3184–3191, 2021.

[14] Lipu Zhou, Shengze Wang, and Michael Kaess, “π-lsam: Lidar smoothing and mapping with planes,” in 2021 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2021, pp. 5751–5757.

[15] Lu Jie, Zhi Jin, Jinping Wang, Letian Zhang, and Xiaojun Tan, “A slam system with direct velocity estimation for mechanical and solid-state lidars,” Remote Sensing, vol. 14, no. 7, pp. 1741, 2022.

[16] Raul Mur-Artal, J. M. M. Montiel, and Juan D. Tardos, “Orb-slam: A versatile and accurate monocular slam system,” IEEE Transactions on Robotics, vol. 31, no. 5, pp. 1147–1163, 2015.

[17] Sameer Agarwal, Keir Mierle, and The Ceres Solver Team, “Ceres Solver,” 3 2022.

[18] Zichao Zhang and Davide Scaramuzza, “A tutorial on quantitative trajectory evaluation for visual (-inertial) odometry,” in 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2018, pp. 7244–7251.

[19] Zhi Yan, Li Sun, Tomáš Krajník, and Yassine Ruichek, “Eu long-term dataset with multiple sensors for autonomous driving,” in 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2020, pp. 10697–10704.

[20] Weisong Wen, Yiyang Zhou, Guohao Zhang, Saman Fahandezh-Saadi, Xiwei Bai, Wei Zhan, Masayoshi Tomizuka, and Li-Ta Hsu, “Urbanloco: A full sensor suite dataset for mapping and localization in urban scenes,” in 2020 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2020, pp. 2310–2316.