Supplementary Material for "BTC: A Binary and Triangle Combined Descriptor for 3D Place Recognition"

1 Applications

In this section, we demonstrate the practical applications of the BTC descriptor in two major robotic applications: SLAM and Multi-Map Alignment. Each application highlights the unique advantages and substantial enhancements offered by the BTC descriptor. When integrated into a SLAM system, the BTC descriptor provides a robust and efficient solution for loop closure detection, which consequently leads to more accurate localization and mapping. Furthermore, the BTC descriptor can be used for Multi-Map Alignment to identify co-visible areas and estimate the relative pose between different point cloud maps that are collected and constructed at different time. In the following subsections, we provide a detailed implementation of these applications and discuss the results.

1.1 SLAM with BTC Descriptor

In this section, we integrate the proposed Binary Triangle Combined (BTC) descriptor into a state-of-the-art LiDAR-Inertial system, FAST-LIO2 [1], and examine its performance in the context of a 3D LiDAR-based SLAM system.

FAST-LIO2 [1] is a real-time, tightly-coupled LiDAR-inertial odometry system. Built on an on-manifold iterated Kalman filter, FAST-LIO2 achieves high computation efficiency and is highly robust to different types of LiDARs and environments. As an odometry, it achieves a high accuracy in short-term localization and mapping, able to merge consecutive LiDAR scans into a consistent local map for local path planning and control [2, 3]. However, due to the lack of a loop closure module, FAST-LIO2 exhibites long-term drift, especially in large environments with few revisits.

Our objective is to improve the long-term accuracy of FAST-LIO2 by incorporating a BTC descriptor-based loop closure module. This integrated SLAM system proceeds by maintaining a pose graph and adhering to the following steps:

1. The SLAM system runs FAST-LIO2 as a front-end.

2. For each LiDAR scan $\mathbf{C}_j$ and its pose ${}^{W}\mathbf{T}_j$ within the world frame estimated by FAST-LIO2, the relative pose constraint ${}^{W}\mathbf{T}_{j-1}^{-1} \cdot {}^{W}\mathbf{T}_j$ is added to the pose graph. Meanwhile, the scan $\mathbf{C}_j$ is accumulated into the current submap $\mathbf{S}_i$ if the submap size does not exceed the preset number n_s scans (see Section IV.A) and its pose within the submap $\mathbf{S}_i$, denoted as ${}^{I}\mathbf{T}_k$, where I refers to the reference frame of the submap $\mathbf{S}_i$ (chosen as the frame of the first scan in the submap), is calculated. If the number of scans in the current submap $\mathbf{S}_i$ exceeds n_s, the current scan is used to initialize a new submap $\mathbf{S}_{i+1}$. In the meantime, all BTC descriptors of the current submap $\mathbf{S}_i$ are extracted (see Section IV).

3. Query the submap $\mathbf{S}_i$'s BTC descriptors in the database following the procedure in Section VI. If a loop submap $\mathbf{S}_c$ is detected with the relative 6-DoF transformation ${}^{C}_{I}\mathbf{T}$, we incorporate the loop constraints ${}^{C}_{I}\mathbf{T}$ to the pose graph. Specifically, for each scan pair, indexed as $(n_s \times c + k, n_s \times i + k)$ for $k = 0, 1, ..., n_s - 1$, in submap $\mathbf{S}_c$ and submap $\mathbf{S}_i$, respectively, we add pose constraint ${}^{C}\mathbf{T}_{k}^{-1} \cdot {}^{C}_{I}\mathbf{T} \cdot {}^{I}\mathbf{T}_{k}$ to the pose graph.

4. With the updated pose graph, which now includes both relative pose constraints between consecutive scans and loop constraints from loop closures, we use the solver GT-SAM [4] to optimize the pose graph. The optimization process corrects the poses of all scans by minimizing the error of all constraints, thereby mitigating the accumulation of drift over time. This leads to a globally consistent trajectory estimate.

To compare BTC with other methods in SLAM, we also integrate Scan Context [5] and BoW3D [6] into the above SLAM system by replacing BTC with the respective loop detection methods. As Scan Context only estimates a 1-DoF yaw for a relative pose, we use it as an initial pose for a subsequent 6-DoF pose estimation by Generalized-ICP [7]. BoW3D estimates the full 6-DoF relative pose similar to BTC, so it can be used directly.

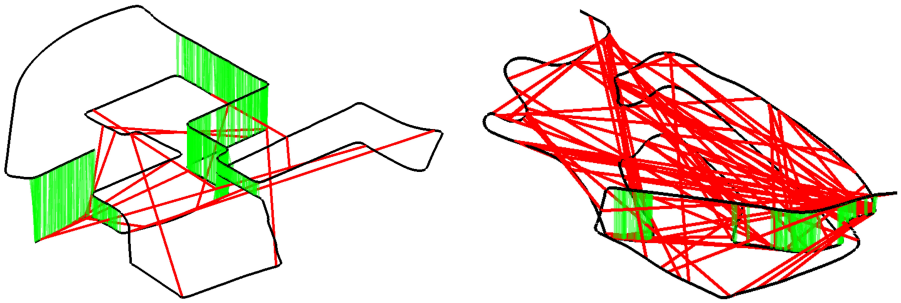
We evaluate the SLAM systems on the KITTI dataset and use the root mean square error (RMSE) as the metric to assess the global trajectory accuracy. For all methods, the loop detection parameters are set to the value that yields the maximum F1 score in their respective precision-recall curve shown in Section VIII.A.

Table 1 shows that all methods effectively reduce the accumulated error of FAST-LIO2, with BTC outperforming Scan Context and BoW3D across all sequences except sequence 07, where all methods behave similarly. To further explain the advantages of BTC over the other methods, we present the loop detection results for KITTI00 and KITTI02 in Fig. 1. As can be seen, while BTC provides more effective constraints through a higher number of true positives, it also manages to keep the number of false positives to a minimum. Additionally, as evaluated in Section VIII.C, BTC achieves more accurate relative pose estimation through plane-to-plane registration. These two factors contributed to its superior trajectory accuracy when compared to the other two methods.

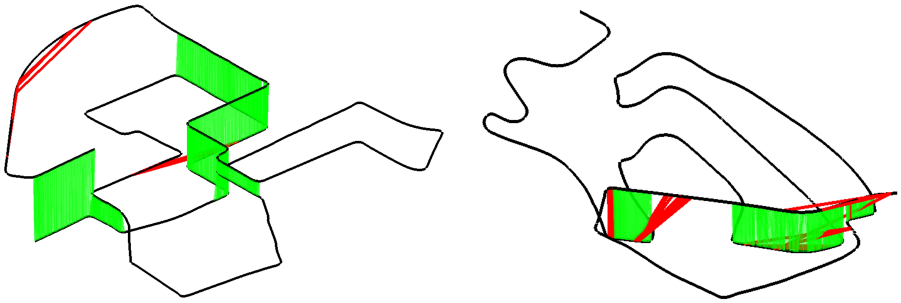
To visually display the enhancements in trajectory accuracy brought by BTC, we plot the trajectories from the initial FAST-LIO2 and that corrected by BTC loop closure on sequences KITTI 00 and 02 in Fig. 2.

TABLE 1: Root Mean Square Error (RMSE) in meters for all methods on KITTI datasets.

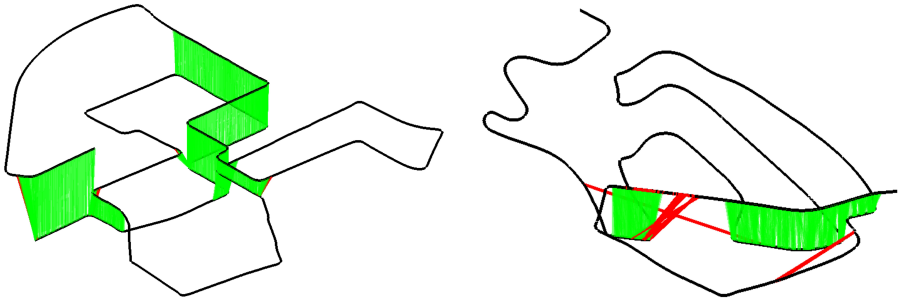
Approach	00	02	05	06	07	08
(Total length [m])	(3724)	(5067)	(2205)	(1232)	(694)	(3222)
FAST-LIO2 [1]	2.72	13.14	1.81	1.19	0.79	3.15
+BTC	**0.92**	**3.66**	**0.33**	**0.31**	**0.33**	**2.05**
+Scan Context [5]	1.53	5.34	0.62	0.33	0.33	2.11
+BoW3D [6]	1.84	10.84	0.35	0.32	0.33	2.41



(a) BoW3D. (TPs : FPs) on KITTI00 (607:16) and KITTI02 (97:90).



(b) ScanContext. (TPs : FPs) on KITTI00 (759:30) and KITTI02 (282:42).



(c) BTC. (TPs : FPs) on KITTI00 (982:11) and KITTI02 (304:12).

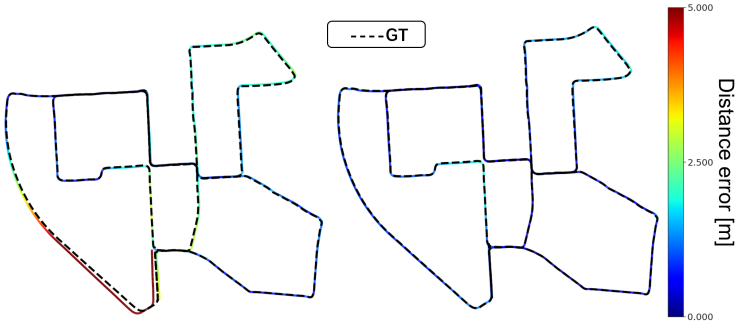
Figure 1: Comparison of loop detection results for BoW3D, Scan Context, and BTC methods on the KITTI00 (left) and KITTI02 (right) sequences when achieving the maximum F1 score. Black lines represent the LiDAR trajectories, green lines represent true positives, and red lines indicate false positives.

1.2 Multi Map Alignment with BTC Descriptor

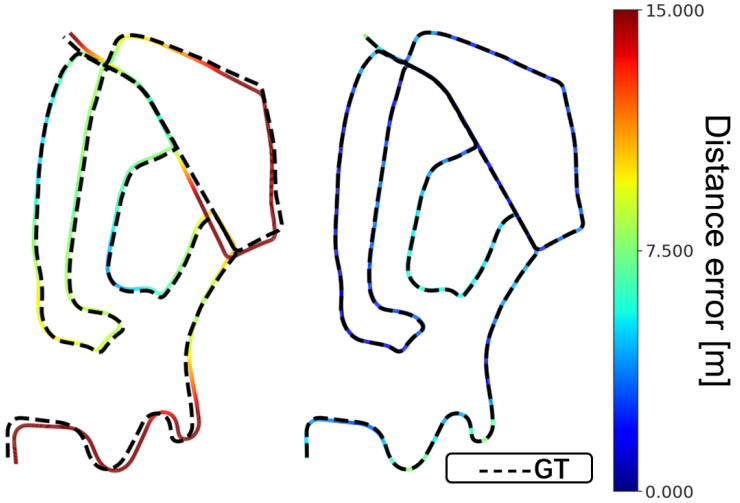
In this section, we demonstrate how the proposed BTC descriptor can be used for multi-map alignment, a crucial task for large-scale mapping. Due to the limited onboard battery, a handheld or robotic mapping device often has limited endurance. As a consequence, for a larger-scale environment, the mapping has to be conducted in multiple runs, where the map collected at each time must be merged in a post-processing stage to obtain the global map of the entire environment.

To address this task, we develop an automatic multi-map alignment system using the BTC descriptor, detailed below.

1. Multiple sequences are collected, where a new sequence has certain overlaps with previous ones.



(a) RMSE on KITTI00



(b) RMSE on KITTI02

Figure 2: Trajectory error (RMSE) of FAST-LIO2 (left) compared to the results corrected by BTC (right) on sequence 00 and sequence 02 of the KITTI dataset.

2. Each sequence is processed using the SLAM described in Section 1.1, yielding a consistent point cloud map for that sequence by incorporating the inner-sequence loop closure constraints (if any). We also save the optimized pose trajectory, based on which we update the submaps in this sequence and re-calculate the BTC descriptors of each submap.

3. Enumerate all pairs of sequences. For each pair, we use BTC descriptors of the larger sequence (in terms of path length) to construct the database and query the BTC descriptors of the smaller sequence in this database. We record all submaps in the smaller sequence that have loop submaps in the larger sequence.

4. Construct a global pose graph, where the nodes consist of all scan poses of all sequences, and the constraints consist of inter-sequence consecutive relative pose (from the SLAM on each sequence) and intra-sequence loop constraints (from the previous step).

5. Optimize the pose graph using GTSAM [4] to solve for the globally-consistent scan poses. Use the optimized scan pose to transform every LiDAR scans to obtain the global point cloud map.

We apply our multi-map alignment system to a real-world, large-scale mapping task: constructing a point cloud map of the entire HKU campus. We employ a non-repetitive scanning LiDAR Livox Avia, which has a built-in IMU and provides 10 Hz LiDAR data and 200 Hz IMU data, and collect the data in four different routes within the campus. The data collection times for each route were 1468s, 863s, 1097s, and 609s, respectively, which corresponded to trajectory lengths of 1593.9m, 1091.6m, 1097.6m, and 700.8m. In choosing each route, we ensure a certain number of loop closures within each sequence (to mitigate inter-sequence drift) and certain overlaps across sequences (to enable multi-map alignment).

After data collection, we use the multi-map alignment procedures above to process each sequence and merge their point cloud maps into the same global reference frame. In Fig. 3, we show the final aligned map with different colors denoting different sequences. As can be seen, despite the relatively small overlaps between neighboring sequences and different travel directions within the overlap areas, the automatic map fusion was still accomplished with a notable level of accuracy, even without applying any global bundle adjustment on the raw LiDAR points. Overall, this experiment showcases the efficacy of using BTC for large-scale mapping tasks, enabling users to design multiple mapping routes more effectively, as BTC is capable of achieving map fusion efficiently, even with limited overlap and casual routes. More visual illustration of the experiment can be seen in the attached video on Youtube: `youtu.be/zB4Xqi3-J9U`.

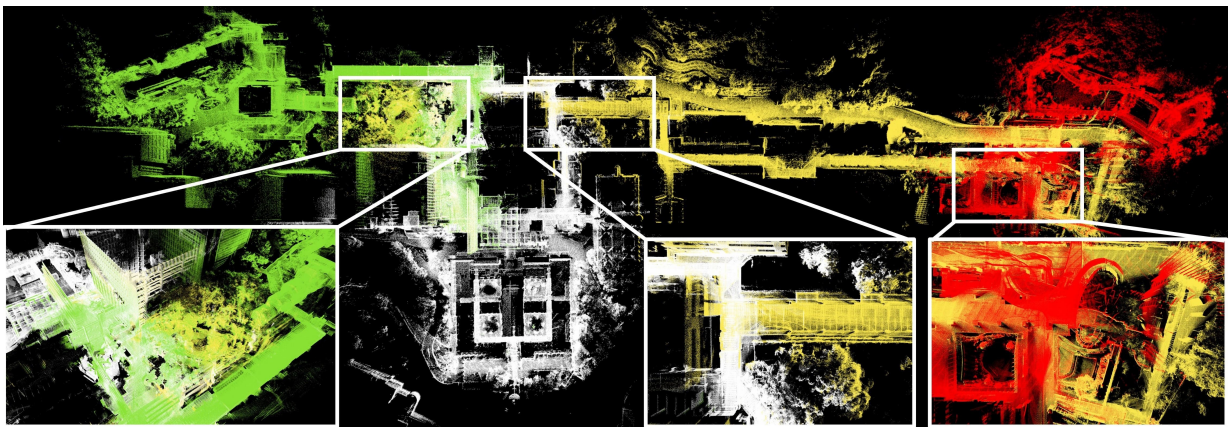


Figure 3: Aligned point cloud map of all sequences after using BTC for map fusion. Different colors represent different sequences, illustrating the small overlap between routes. Zoom-in views of overlapping areas are provided to showcase the accurate alignment of point clouds between sequences.

References

[1] W. Xu, Y. Cai, D. He, J. Lin, and F. Zhang, "Fast-lio2: Fast direct lidar-inertial odometry," *IEEE Transactions on Robotics*, 2022.

[2] G. Lu, W. Xu, and F. Zhang, "On-manifold model predictive control for trajectory tracking

on robotic systems," *IEEE Transactions on Industrial Electronics*, vol. 70, no. 9, pp. 9192–9202, 2023.

[3] Y. Ren, F. Zhu, W. Liu, Z. Wang, Y. Lin, F. Gao, and F. Zhang, "Bubble planner: Planning high-speed smooth quadrotor trajectories using receding corridors," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 6332–6339.

[4] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert, "isam2: Incremental smoothing and mapping using the bayes tree," *The International Journal of Robotics Research*, vol. 31, no. 2, pp. 216–235, 2012.

[5] G. Kim and A. Kim, "Scan context: Egocentric spatial descriptor for place recognition within 3d point cloud map," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 4802–4809.

[6] Y. Cui, X. Chen, Y. Zhang, J. Dong, Q. Wu, and F. Zhu, "Bow3d: Bag of words for real-time loop closing in 3d lidar slam," *IEEE Robotics and Automation Letters*, 2022.

[7] A. Segal, D. Haehnel, and S. Thrun, "Generalized-icp." in *Robotics: science and systems*, vol. 2, no. 4. Seattle, WA, 2009, p. 435.