LiDAR-to-LiDAR Global Localization

Himanshu Singh   Pavan Karke   Srinath Bhamidipati

December 7, 2024

Problem Statement

Perform **global localization** by matching LiDAR point clouds to a pre-built HD LiDAR map using 3D-BBS (3D Bag of Binary Words) for **feature extraction** and a Branch-and-Bound (BnB) algorithm for efficient **scan matching**.

Accomplishment

- Generation of HD LiDAR Maps, using KISS-ICP [1].
- Feature extraction and scan matching, using 3D-BBS [2].
- Depth estimation and reprojection to point clouds, using Depth Anything v2 [3].

Previous Works

- Histogram based methods to establish correspondences, coupled with outlier removal [4].
- Frame based methods that aggregate geometrical features [5].
- Bag-of-words feature representations [6].
- Branch and bound based frameworks for efficient search [7].

Dataset

- We employ Velodyne LiDAR scans and RGB images from KITTI Odometry dataset [8].
- Specifically, our demonstration will be based on hundred scans from the first sequence of the dataset.
- The dataset focuses on outdoor urban environments involving roads, vehicles, trees, buildings, etc.

Problem Formulation

- Occupancy Map

$$\mathbf{x}^* = \arg\max_{\mathbf{x} \in \mathcal{X}} \sum_{k=1}^{K} \mathcal{M}(T_{\mathbf{x}} \mathbf{s}_k)$$

  where $T_{\mathbf{x}}$ is the transformation matrix corresponding to pose $\mathbf{x}$, $\mathcal{M}(\cdot)$ denotes the occupancy function of the map $\mathcal{M}$ at a given point, and $\mathcal{S} = \{\mathbf{s}_k \in \mathbb{R}^3 \mid k = 1, \ldots, K\}$ denotes the LiDAR scan.

- Spatial Hashing Function

$$\mathcal{H}_l(\mathbf{v}) = \begin{cases} 1 & \text{if voxel } \mathbf{v} \text{ is occupied,} \\ 0 & \text{otherwise.} \end{cases}$$

Problem Formulation

- Branching

$$
\begin{aligned}
C_{c_1} = \{ & (2x + j_x, 2y + j_y, 2z + j_z, a_a c_\alpha + j_\alpha, a_\beta c_\beta + j_\beta, a_\gamma c_\gamma + j_\gamma, c_{l-1}) \\
& |(j_x, j_y, j_z) \in \{0, 1\}, j_\alpha \in \{0, ..., a_\alpha - 1\}, j_\beta \in \{0, ..., a_\beta - 1\}, \\
& j_\gamma \in \{0, ..., a_\gamma - 1\},\ a_a c_\gamma + j_\gamma, c_{l-1})\}
\end{aligned}
$$

where a_α, a_β and a_γ are the number of divisions for each rotational component of a node.

- Score Computation

$$\text{score}(\mathbf{x}) = \sum_{k=1}^{K} \mathcal{H}_l(T_{\mathbf{x}} \mathbf{s}_k)$$

$$\text{score}_{\text{upper}}(c) = \sum_{k=1}^{K} \max_{\mathbf{v} \in \mathcal{N}(T_{\mathbf{x}_c} \mathbf{s}_k)} \mathcal{H}_l(\mathbf{v})$$

where $\mathcal{N}(\cdot)$ denotes the neighborhood voxels surrounding the transformed scan point.

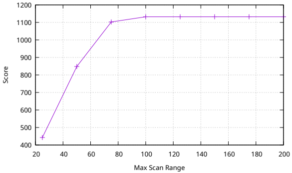
Algorithm

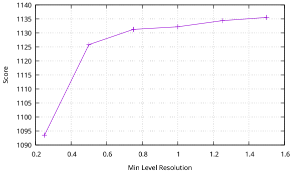
Algorithm 2 Batched 3D-BBS

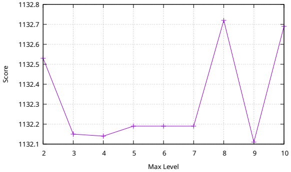
- $best\_score \leftarrow score\_threshold$
- **while** C is not empty **do**
  - Pop c from the queue C
  - **if** $\overline{score}(c) < best\_score$ **then**
    - **continue**
  - **if** c is a leaf node **then**
    - $match \leftarrow c$
    - $best\_score \leftarrow \overline{score}(c)$
  - **else**
    - Branch: Split c into nodes $\mathcal{C}_{c_l}$
    - Add $\mathcal{C}_{c_l}$ to $\mathcal{C}_{\text{CPU}}$
  - **if** $|\mathcal{C}_{\text{CPU}}| > b$ **then**
    - $\mathcal{C}_{\text{GPU}} \leftarrow \mathcal{C}_{\text{CPU}}$
    - Compute and memorize a score for each element in $\mathcal{C}_{\text{GPU}}$
    - $\mathcal{C}_{\text{CPU}} \leftarrow \mathcal{C}_{\text{GPU}}$
    - Push $\mathcal{C}_{\text{CPU}}$ onto the queue C, sorted by score
    - Clear the all elements of $\mathcal{C}_{\text{CPU}}$.
- **return** $best\_score$ and $match$ when set.

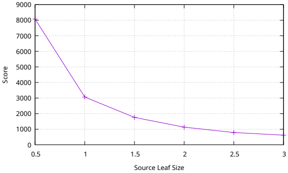
Figure: 3D-BBS Algorithm [2]

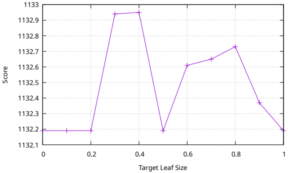
Ablation Study











LiDAR Scans

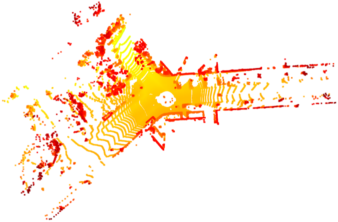


Figure: Sample LiDAR Frame

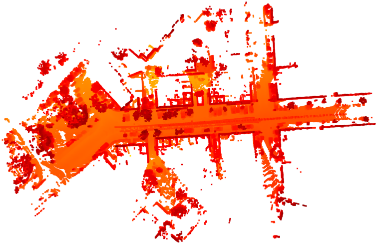


Figure: LiDAR Map generated using ICP

Image to 3D Projection

Figure: Sample RGB Image

Figure: Estimated Depth

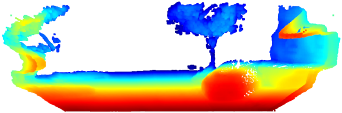


Figure: Projected Point Cloud

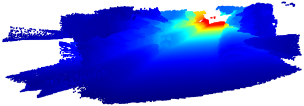


Figure: Concatenated Point Cloud

References I

[1] I. Vizzo, T. Guadagnino, B. Mersch, L. Wiesmann, J. Behley, and C. Stachniss, "Kiss-icp: In defense of point-to-point icp - simple, accurate, and robust registration if done the right way," *IEEE Robotics and Automation Letters*, vol. 8, no. 2, pp. 1029–1036, Feb. 2023, ISSN: 2377-3774. DOI: 10.1109/lra.2023.3236571. [Online]. Available: http://dx.doi.org/10.1109/LRA.2023.3236571.

[2] K. Aoki, K. Koide, S. Oishi, M. Yokozuka, A. Banno, and J. Meguro, *3d-bbs: Global localization for 3d point cloud scan matching using branch-and-bound algorithm*, 2024. arXiv: 2310.10023 [cs.RO]. [Online]. Available: https://arxiv.org/abs/2310.10023.

[3] L. Yang, B. Kang, Z. Huang, *et al.*, *Depth anything v2*, 2024. arXiv: 2406.09414 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2406.09414.

[4] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (fpfh) for 3d registration," in *2009 IEEE International Conference on Robotics and Automation*, 2009, pp. 3212–3217. DOI: 10.1109/ROBOT.2009.5152473.

[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)*, 2018, pp. 4802–4809. DOI: 10.1109/IROS.2018.8593953.

References II

[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*, vol. 8, no. 5, pp. 2828–2835, May 2023, ISSN: 2377-3774. DOI: 10.1109/lra.2022.3221336. [Online]. Available: http://dx.doi.org/10.1109/LRA.2022.3221336.

[7] W. Hess, D. Kohler, H. Rapp, and D. Andor, "Real-time loop closure in 2d lidar slam," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 1271–1278. DOI: 10.1109/ICRA.2016.7487258.

[8] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *International Journal of Robotics Research (IJRR)*, 2013.