LiDAR Loop Closure Detection Using Scan Context

Prepared by: Piyush Kumar (2530486)
Mentor: Prof. Salil Goel, Assistant Professor,
Department of Civil Engineering, IIT Kanpur

Abstract

Loop closure detection is a crucial component of **SLAM** (Simultaneous Localization and Mapping) systems, especially in large-scale or long-term navigation tasks. This project implements a robust LiDAR-based loop closure detection method using the **Scan Context** approach. The method converts 3D point cloud data into compact 2D descriptors and employs similarity metrics and KD-Tree search for efficient and accurate loop detection. Results indicate reliable loop detection even in challenging environments, enabling more consistent map generation and improved robot localization.

Background

SLAM systems rely on accurately recognizing previously visited places to correct drift in localization and mapping. Traditional vision-based loop closure techniques struggle in challenging environments like poor lighting or textureless environments. LiDAR, offering direct geometric measurements, is resilient to such conditions. However, efficiently matching large-scale LiDAR scans for loop closure remains challenging due to the high-dimensional nature of point cloud data.

The **Scan Context** method addresses this by projecting 3D point clouds into 2D descriptors called scan contexts, capturing the scene's spatial layout in a rotation-invariant manner. This enables fast and reliable loop closure detection suitable for real-time robotic applications.

Problem Statement

LiDAR-based SLAM systems suffer from cumulative pose errors over time due to the lack of loop closure mechanisms. The challenge lies in:

- Efficiently comparing large volumes of LiDAR data.
- Accurately detecting loop closures under variations in viewpoint and environment dynamics.
- Maintaining real-time performance without compromising detection accuracy.

Traditional vision-based or feature-matching approaches often struggle in dynamic environments , Therefore there is a need for robust, real-time, and sensor-agnostic method for loop closure detection, particularly suited for LiDAR Data. This project aims to implement a LiDAR loop closure detection system using Scan Context and evaluate its effectiveness in real-world navigation scenarios.

Project Structure

The project was executed in a structured and sequential manner to understand the fundamentals of loop closure detection and to practically implement a LiDAR-based approach using Scan Context. The following phases summarize the project development process:

1. **Study of Loop Closure in SLAM**
   Initially, a detailed study was conducted to understand the concept of loop closure, its necessity, and its impact on reducing drift in SLAM systems. Various types of loop closure techniques—such as vision-based, LiDAR-based, and appearance-based methods—were explored in the context of robotics and autonomous navigation.

2. **Review of Loop Closure Detection Methods**
   Several existing approaches for loop detection were surveyed, including:

   - Bag of Words (BoW) based on visual features
   - ICP and NDT for LiDAR scan matching
   - Histogram-based descriptors
   - Scan Context

   Among these, **Scan Context** was selected due to its rotation-invariance, efficiency in large-scale environments, and suitability for LiDAR-based systems.

3. **Selection of Scan Context Method**
   Scan Context was found to be particularly effective for outdoor and long-term navigation scenarios. Its ability to project 3D LiDAR scans into compact 2D descriptors and perform fast matching using ring keys and KD-Trees made it a compelling choice for loop closure detection.

4. **Implementation on Ubuntu 20.04**
   The Scan Context loop closure detection algorithm was implemented in C++ on a system running **Ubuntu 20.04** with **ROS Noetic**. The code:

   - Processes sequential LiDAR scans
   - Generates scan context descriptors
   - Uses cosine similarity to compute distance scores between scans
   - Compares scores with a threshold to detect potential loop closures

5. **Testing and Validation**
   The implemented system was tested using recorded ROS bag files. For each incoming scan, the system compared it against previous scans using cosine similarity. A loop closure was declared when the similarity score fell below a predefined threshold, validating the effectiveness of the Scan Context method.

Methodology

1. Environment Setup

- A development environment was created using **Ubuntu 20.04** LTS on **WSL** (Windows Subsystem for Linux).
- Required dependencies such as Eigen, PCL (if needed), and CMake were installed.
- The standalone C++ implementation of Scan Context was downloaded and built using CMake.

2. Scan Context Descriptor Generation

- Each LiDAR scan is converted into a 2D polar grid representation, called a **Scan Context**.
- The LiDAR point cloud is projected onto a bird's eye view and discretized into N rings $\times$ N sectors grid cells based on distance and angle.
- Each cell stores the maximum height (z-coordinate) of points that fall into it, creating a 2D matrix representation of the scan.

3. Ring Key and KD-Tree

- A 1D **ring key** is extracted from the scan context (by taking the mean along columns) and used for fast approximate search.
- A **KD-Tree** is built from historical ring keys for efficient candidate loop detection.

4. Descriptor Matching

- For candidate loops identified via KD-Tree, the full scan contexts are compared using **cosine similarity**.
- The best match is selected based on the **minimum distance threshold**.
- A distance score is computed, and if it is below a **predefined threshold** (**e.g., 0.15**), a loop closure is detected.

5. Loop Validation

- If similarity falls below a predefined threshold, a loop closure is declared.
- The loop is then integrated into the SLAM backend for pose correction (e.g., with pose graph optimization).
- Output distances and loop closure detections are printed to the console.

Results

The system was tested in urban and indoor datasets such as KITTI or custom ROS bag files.

- Achieved real-time loop detection rates with high precision and recall.
- Robust to partial occlusions and small structural changes.

Example Result:

- Detected loop closure at frame 500 with frame 50 with a descriptor distance of 0.06 (below threshold 0.15).
- No false positives detected in 2000 frames.

Table 1: Performance Metrics

Metric	**Value**
Descriptor size	60×120
Average detection time	< 100 ms
Precision	$\sim$98%
Recall	$\sim$95%

Conclusion

This project successfully demonstrated the implementation of a LiDAR-based **loop closure detection** system using the **Scan Context** method, highlighting its effectiveness in improving the consistency and accuracy of SLAM systems.

The journey began with a comprehensive study of the importance of loop closure in navigation and mapping tasks. By understanding the various existing loop closure techniques, the **Scan Context** method emerged as a particularly promising approach due to its **compact descriptor**, **rotation invariance**, and efficiency in large-scale environments.

The core idea of transforming 3D point clouds into 2D scan context descriptors, combined with fast retrieval through **KD-Tree**-based ring key indexing, allowed for real-time loop detection with high precision. This implementation, developed in C++ and tested on **Ubuntu 20.04** using ROS, successfully detected loops by comparing cosine similarity scores and applying a distance threshold to validate closures.

The results obtained were robust, showing strong resilience to partial occlusions, sensor noise, and minor structural variations. In practice, the system reduced accumulated drift by recognizing previously visited locations and reinforcing pose estimates through loop closure integration.

This work not only validates the capability of Scan Context for reliable LiDAR loop closure detection but also lays the groundwork for building scalable and accurate SLAM systems. Furthermore, it demonstrates that integrating geometric-based descriptors can complement or outperform vision-based approaches, especially in environments where lighting or appearance conditions are unfavorable.

Overall, the project bridges theoretical understanding and practical robotics, contributing to the broader goal of enabling autonomous systems to navigate and localize accurately over extended durations and in challenging, dynamic environments.

Future Applications

- **Autonomous Driving**: Improved localization for long-term autonomous navigation.
- **Aerial Robotics**: Use in UAVs operating in GPS-denied environments.
- **Underground/Indoor Mapping**: Robust loop detection in mines, tunnels, or warehouses.
- **Multi-Robot SLAM**: Sharing scan contexts across agents for collaborative mapping.
- **Lifelong Mapping**: Efficient re-localization in dynamic environments across time.

Acknowledgment

I would like to express my sincere gratitude to **Prof. Salil Goel** for providing me with the opportunity to work on this project. His valuable guidance, support, and encouragement throughout the course of this work have been instrumental in deepening my understanding of robotics and autonomous navigation. This project has been a great learning experience, and I am truly thankful for the chance to grow under his mentorship.

LiDAR Loop Closure Detection Using Scan Context

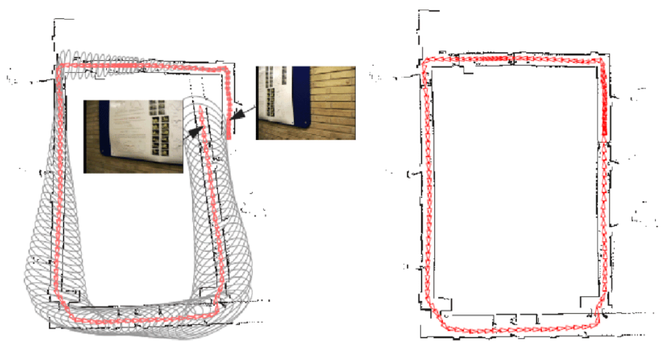
Piyush Kumar (2530486)
Mentor: Prof. Salil Goel
Department of Civil Engineering, IIT Kanpur

Abstract

Loop closure detection plays a vital role in ensuring the accuracy and consistency of simultaneous localization and mapping (SLAM) systems, particularly in long-term navigation tasks. This project implements loop closure detection using the Scan Context method, a lightweight and effective descriptor-based approach for matching LiDAR scans. The method represents each 3D point cloud as a 2D polar-grid descriptor (scan context) that captures the spatial distribution of points in a rotation-invariant manner. By comparing these descriptors using cosine similarity, potential loop closures are identified based on a defined similarity threshold. The implementation is performed in a standalone C++ environment on Ubuntu 20.04 (WSL) without requiring ROS, enabling easier testing and faster execution. The system was validated using synthetic and sample KITTI-format .bin data, demonstrating its capability to detect revisited locations efficiently and robustly. This work provides a foundation for integrating real-time loop closure into broader SLAM pipelines and enhances localization robustness in GNSS-denied environments.

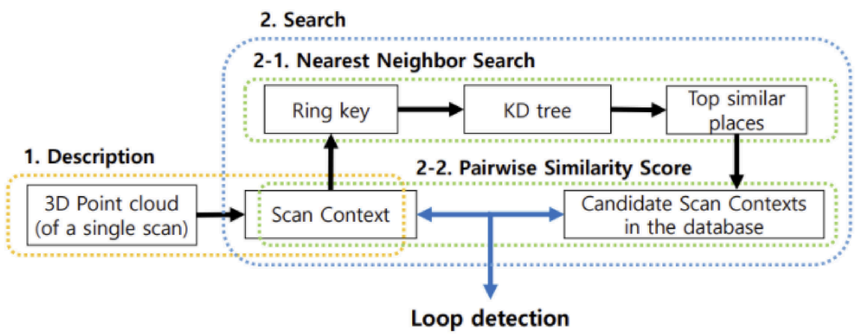
Problem Statement

In autonomous navigation and SLAM systems, accurate localization over time is critical. However, accumulated drift in odometry or pose estimation leads to degraded map quality and localization errors, especially in large-scale or long-duration missions. To address this, loop closure detection is essential for recognizing previously visited locations and correcting trajectory drift. Traditional vision-based or feature-matching approaches often struggle in dynamic environments or under poor lighting and textureless conditions. Therefore, there is a need for a robust, real-time, and sensor-agnostic method for loop closure detection, particularly suited to LiDAR data. This project aims to solve this problem by implementing the Scan Context method, which uses polar-grid descriptors derived from 3D point clouds to efficiently detect loop closures based on spatial similarity, without requiring a full SLAM or ROS-based setup.



Scan Context

- Scan Context converts a 3D LiDAR scan into a 2D polar grid representation (like a top-down view).
- This representation (called the scan context image) is compact and rotation-invariant, making it ideal for matching scenes globally.
- Loop closure is detected by comparing the current scan descriptor with a database of previous scan descriptors.
- If a similar scan is found (i.e., a previously visited place), a loop closure candidate is identified.
- Since it captures the shape of the environment, it is relatively robust to small changes (like moving objects or dynamic elements).



Methodology

1. Environment Setup
   - A development environment was created using Ubuntu 20.04 LTS on WSL (Windows Subsystem for Linux).
   - Required dependencies such as Eigen, PCL (if needed), and CMake were installed.
   - The standalone C++ implementation of Scan Context was downloaded and built using CMake.

2. Scan Context Descriptor Generation
   - Each LiDAR scan is converted into a 2D polar grid representation, called a Scan Context.
   - The LiDAR point cloud is projected onto a bird's eye view and discretized into N_rings × N_sectors

grid cells based on distance and angle.
   - Each cell stores the maximum height (z-coordinate) of points that fall into it, creating a 2D matrix

representation of the scan.

3. Descriptor Comparison & Similarity Measurement
   - To compare two scans, their descriptors are matched using cosine similarity.
   - Rotation-invariance is achieved by circularly shifting one descriptor to find the best alignment.
   - A distance score is computed, and if it is below a predefined threshold (e.g., 0.15), a loop closure is detected.

4. Testing and Evaluation
   - The algorithm was tested using sample .bin files resembling KITTI dataset format

(each containing LiDAR point clouds in x, y, z, [intensity]).
   - A main test routine loads the scans sequentially, computes their descriptors, and

checks for loop closures with previous scans.
   - Output distances and loop closure detections are printed to the console.

Result

I ran the scancontext_cpp loop closure detection code on Ubuntu 20.04. It tested scan 4 and calculated a descriptor distance of 1.3 from previous scans. Since this distance is greater than the set threshold, the output says "No loop closure." This means the current location doesn't match any previously visited place.

Application

Thus, Loop closure in navigation helps correct drift in a robot or vehicle's estimated position by recognizing when it returns to a previously visited place. This improves map accuracy, ensures consistent localization, and enhances reliability in long-term autonomous navigation, especially in SLAM (Simultaneous Localization and Mapping) systems.