LiDAR Loop Closure Detection Using Scan Context

Piyush Kumar (230752)

Mentor: Prof. Salil Goel

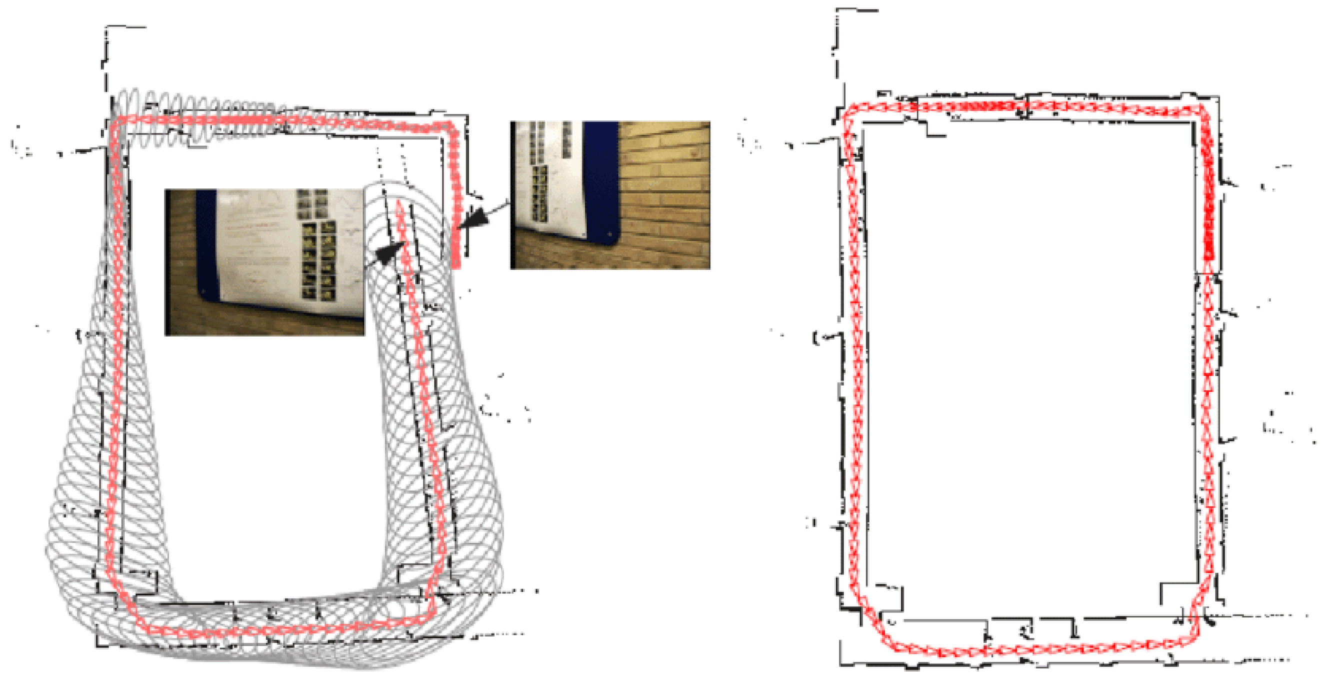
Department of Civil Engineering

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.

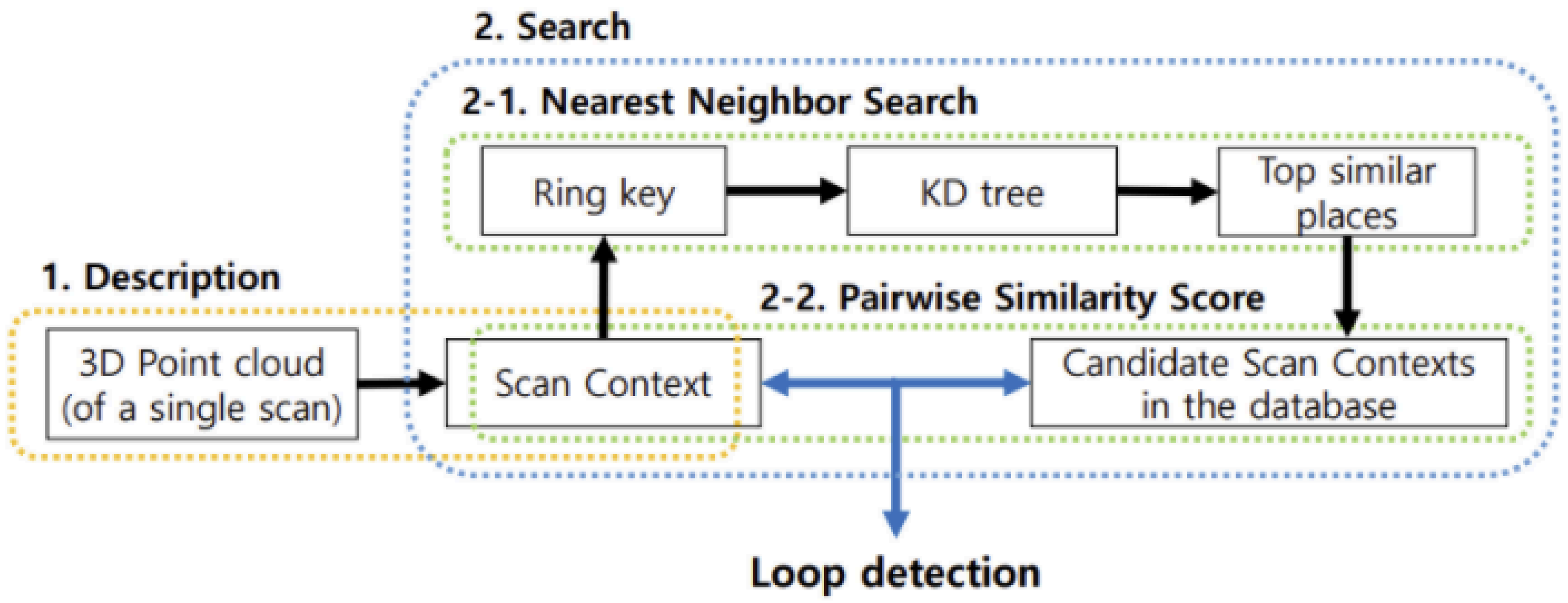
Problem Statement

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.