Super Visual LiDAR Odometry and Mapping (Super VLOAM)

Aneesh Sinha Anvesh Reddy Gummi Prakrit Tyagi Shruti Bansal Swathi Jadav
Carnegie Mellon University
5000 Forbes Avenue Pittsburgh, PA 15213
aneeshsi, agummi, prakritt, shrutib, sjadav

I. Background

Visual Lidar Odometry and mapping (VLOAM) [1] is a state-of-the-art technique used in robotics for localizing and mapping an environment and a robot in real-time. It combines the benefits of both visual and lidar sensors to accurately estimate the robot's position and map its surroundings.

VLOAM works by using the visual sensor to extract visual features at a high frequency (60 Hz) from the environment, such as corners, edges, and textures. The lidar sensor, running at a low frequency (1 Hz) on the other hand, generates a 3D point cloud of the surrounding environment and helps in correcting the visual odometry drift. By fusing the visual features and lidar data, VLOAM is able to estimate the robot's position and create a high-resolution map of the environment.

A. Motivation

By studying a state-of-the-art odometry method that fuses multiple sensors, we wished to enhance our understanding in this field. Initially, our scope of the project was to re-implement the original paper. However, following the feedback received, we decided it was more worthwhile to reuse existing code to reproduce a baseline, upon which we would attempt to improve the current method.

B. Challenges

One of the main challenges faced by VLOAM is the lack of robustness to illumination changes, occlusions, and low-texture regions, which can cause the visual data to be unreliable. Furthermore, cost of the sensor is another challenge that we would like to address by comparing performance of VLOAM with RGBD based odometry method.

C. Existing Approaches

Existing approaches to VLOAM include both feature-based and direct methods. Feature-based methods rely on detecting and tracking distinctive features in the environment, such as corners and edges, to estimate the robot's motion and build a map of the environment. VINS-Mono algorithm, which uses a monocular camera and an IMU to estimate the robot's pose and map the environment is a feature based algorithm. Direct methods, on the other hand, directly estimate the motion and map from the raw sensor data without explicitly detecting features. Direct methods include the LSD-SLAM algorithm, which uses a monocular camera to directly estimate the motion and map. Below are explanation of some of these approaches:

1) LOAM: LOAM (Lidar Odometry and Mapping) [2] is a popular method for estimating the pose of a mobile robot using a 3D lidar sensor. It was first introduced by Zhang and Singh in 2014 and has since been widely used in robotics research. One of the main advantages of LOAM is its computational efficiency, which enables it to run in real-time on a low-cost embedded system.

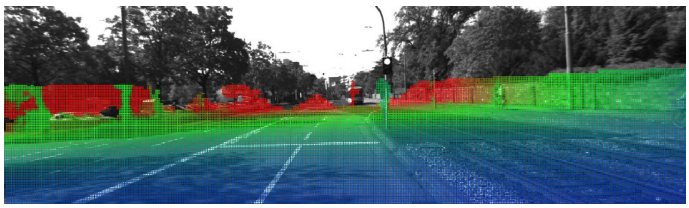


Fig. 1: Point cloud visualization

LOAM is based on the iterative closest point (ICP) algorithm, which is used to match consecutive lidar scans and estimate the relative motion between them. However, LOAM also incorporates several enhancements to improve its accuracy and robustness. For example, it uses a front-end module to extract feature points from the raw lidar data and a back-end module to optimize the estimated trajectory based on constraints from the odometry.

LOAM has been evaluated on various datasets and compared to other state-of-the-art methods. In general, it has been shown to achieve high accuracy and robustness, even in challenging environments with dynamic objects and occlusions. However, there are also some limitations to LOAM, such as its sensitivity to sensor noise and its reliance on accurate initial guess for the motion estimation. Despite its limitations, LOAM remains a popular choice for lidar-based odometry and mapping in robotics research. Many researchers have proposed extensions and variations of the original LOAM algorithm to further improve its performance and address some of its limitations.

2) VLOAM: The paper presents a method for combining visual odometry and Lidar odometry in a fundamental and first principle method to improve accuracy and speed while reducing drift [1]. The method shows improvements in performance over the state of the art, particularly in robustness to aggressive motion and temporary lack of visual features.

The proposed online method starts with visual odometry

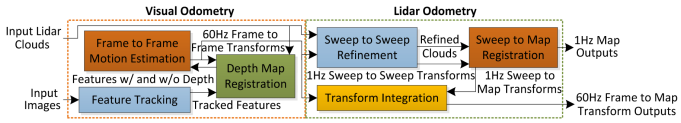


Fig. 2: Block diagram of the odometry and mapping software system.

to estimate the ego-motion (Camera kinematics) at a high frequency and to register point clouds from a scanning Lidar at low frequency. Then the motion estimates from the visual odometry are refined by matching point clouds between consecutive sweeps of lidar and finally point clouds are registered on the map.

The authors evaluate their approach using several datasets as well as benchmark dataset from KITTI and compare it to other state-of-the-art SLAM methods. The results show that their algorithm achieves low-drift, robustness to noise and occlusions, and fast processing time. The authors also demonstrate the practicality of their approach by deploying it on a real-world robotic platform and showing successful mapping and localization in challenging environments.

3) ORB-SLAM: ORB-SLAM is a reliable and comprehensive solution for monocular SLAM.It employs a policy to create and remove key frames, allowing for flexible map expansion, particularly useful in poorly conditioned exploration trajectories [3]. ORB-SLAM is capable of recognizing places from severe viewpoint changes, and its ORB features are both fast to extract and match, enabling real-time and accurate tracking and mapping without the need for multi-threading or GPU acceleration. Overall,ORB-SLAM provides interesting long-term mapping results by storing a history of different visual appearances.

4) ORB-SLAM2: ORB-SLAM2 [4] is a stereo SLAM system that was first released in 2016. It builds on the ORB-SLAM1 algorithm, but it uses a more robust feature detector and tracker, and it also uses a bundle adjustment algorithm to improve the accuracy of the estimated pose. ORB-SLAM2 is able to run in real-time on a standard laptop, and it has been shown to be effective in a variety of indoor and outdoor environments.

5) ORB-SLAM3: ORB-SLAM3 [5] is a visual-inertial SLAM system that was first released in 2018. It builds on the ORB-SLAM2 algorithm, but it also uses an inertial measurement unit (IMU) to provide additional information about the camera pose. This allows ORB-SLAM3 to track the camera pose even when there are no visual features available, such as when the camera is in a dark environment. ORB-SLAM3 is able to run in real-time on a standard laptop, and it has been shown to be effective in a variety of indoor and outdoor environments.

6) Superpoint: One notable paper on feature detection is SuperPoint: Self-Supervised Interest Point Detection and Description [6]. The paper presents a self-supervised approach to training the SuperPoint network, which allows it to learn to extract features without the need for labeled data. The authors also show that SuperPoint outperforms other feature extraction methods such as ORB and SIFT in terms of accuracy, speed and number of features detected.

7) SuperGlue: Another paper that builds on the SuperPoint algorithm is SuperGlue: Learning Feature Matching with Graph Neural Networks [7] . In this paper, the authors introduce SuperGlue, a graph neural network-based method for feature matching that uses the SuperPoint algorithm for feature extraction. The authors show that SuperGlue outperforms other feature matching methods such as SIFT and RANSAC in terms of accuracy and speed, especially in challenging scenarios such as low-texture regions and dynamic environments.

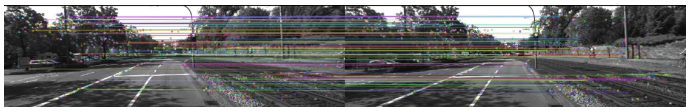


Fig. 3: Feature Correspondences

D. Dataset

We used KITTI dataset to evaluate our implementation. The KITTI dataset is a widely used dataset for autonomous driving research. It contains data from different sensors such as stereo cameras, laser scanners, and GPS/IMU localization systems. Specifically we used KITTI-Odometry dataset [8].

The KITTI-Odometry dataset contains 22 sequences, where 11 sequences (00-10) come together with ground truth trajectories for training and 11 sequences (11-21) without ground truth for evaluation. Each of the sequences have different trajectories in shape and length. Each sequence contains high resolution RGB and Grayscale stereo images, as well as Velodyne laser data. This data was recorded in the city of Karlsruhe, Germany using a car fitted with sensors.

II. Proposal

To address the challenges mentioned, we would integrate deep-learning based methods: Superpoint model for extracting keypoints from the images and Superglue for matching these features. We would then compare our method with existing method to compare these methods. We will also follow up by comparing ORB-SLAM methods with our method/current VLOAM method to understand the justification of costs.

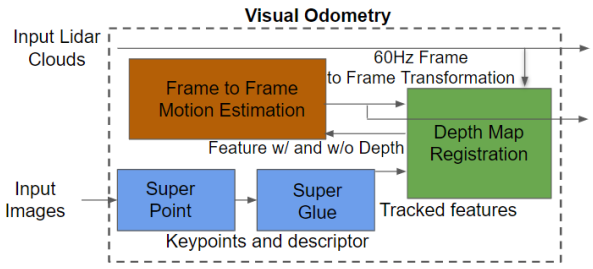


Fig. 4: Block Diagram of Visual Odometry in Super VLOAM

III. Methodology

In our proposed approach, we plan to replace the classical feature extraction and matching algorithms in VLOAM with SuperPoint and SuperGlue deep learning models respectively. We will integrate the SuperPoint and SuperGlue networks into the VLOAM pipeline, which we call Super-VLOAM.

To evaluate the performance of our proposed approach, we plan to conduct experiments using synthetic and if possible real-world datasets. We will compare the accuracy and efficiency of our Super-VLOAM model with the current VLOAM implementation. We expect that our proposed approach will improve the accuracy and robustness of VLOAM, especially in challenging scenarios such as low-texture regions and dynamic motion of the vehicle.

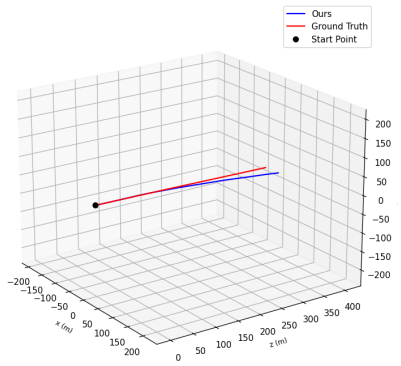
A. Work Done and Experimentation

- We have deeply understood VLOAM for getting a better perspective on the lidar and visual odometry algorithms. To save us time and effort we referred to a few github repositories. Using these repositories we visualized the rqt graph to understand the topics needed, dependencies, and frameworks needed and how the roslaunch files are written. We have used RVIZ in ROS to visualize the baseline results.
- We integrated the Superpoint and Superglue networks into VLOAM. We used pre-trained models trained using PyTorch, and converted them to the intermediate ONNX format for integration into our VLOAM (C++) code. We were able to successfully run inferences on the models on the CPU, but we are currently having trouble running them on the GPU.
- We tested Super-VLOAM on KITTI odometry sequences 04 (simple) and 05 (more complex). On the simpler sequence, the performance was comparable to existing VLOAM, which uses ShiTomasi features and ORB descriptor matching in the visual odometry block. However, on the harder sequence, we had an underwhelming performance. On each frame, the processing was about 100x slower than the classical methods. The reported RMSE is in three digits for translational errors.
  We attribute this to the model running on the CPU, which takes considerable time to process the frames. As a result, multiple frames in the sequence are skipped. The skipped frames contain more features to match, which in turn worsens the performance.
  We are currently working on optimizing the code to improve the performance of Super-VLOAM on the GPU. We believe that this will allow us to achieve comparable performance to existing VLOAM on both simple and complex sequences.
- We have reproduced results with existing ORB-SLAM2 model for the purpose of comparing Visual Odometry part of Super-VLOAM with it.

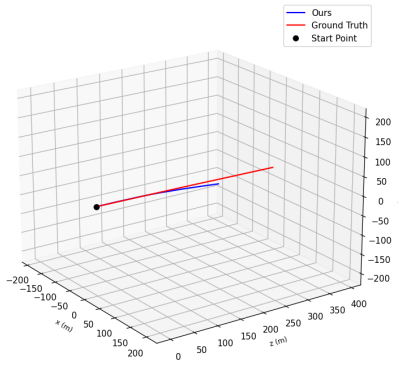
IV. Results

In this section we present results for sequence 04 and 05 of KITTI odometry dataset. In the table I we have summarized performance parameters of various implementation for comparisions. Translational error is in meters and rotational errors is in radians. In the next section we will present results in detailed for sequence 04.

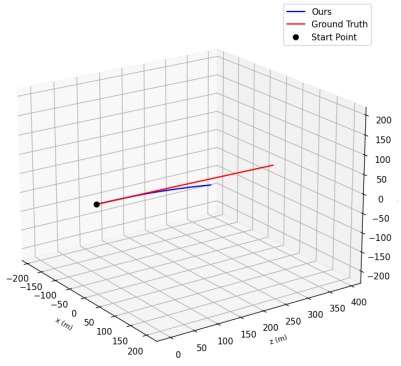
A. Sequence 04



(a) Sequence 4 path with ShiTomasi feature detector



(b) Sequence 4 path with Orb feature detector



(c) Sequence 4 path with Superpoint and SuperGlue

Fig. 5: Sequence 04 3D path comparison

Seq	Sensor	Error	VLOAM ShiTomasi	VLOAM	ORB-SLAM2	Super VLOAM
04	Visual	t_{err}	4.550	5.244	0.857%	4.555
		r_{err}	0.024	0.040	0.130%	0.044
	Lidar	t_{err}	1.863	1.585		1.485
		r_{err}	0.015	0.017		0.015
	Mixed	t_{err}	0.605	0.399		0.540
		r_{err}	0.004	0.004		0.004
05	Visual	t_{err}	26.652	68.145	9.116%	98.786
		r_{err}	0.092	0.264	1.060%	0.044
	Lidar	t_{err}	27.562	64.742		101.440
		r_{err}	0.105	0.256		0.441
	Mixed	t_{err}	25.247	66.357		102.010
		r_{err}	0.086	0.262		0.442

TABLE I: Table encapsulating performance parameters of various implementations

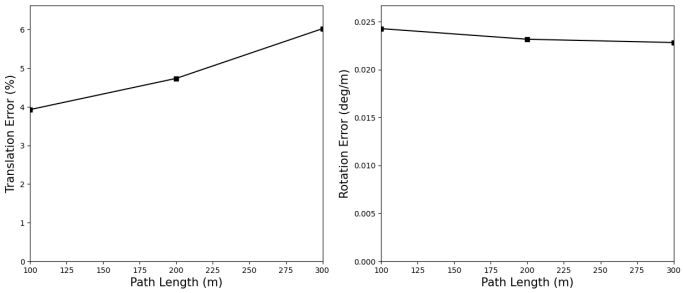


Fig. 6: Error graph for ShiTomasi feature

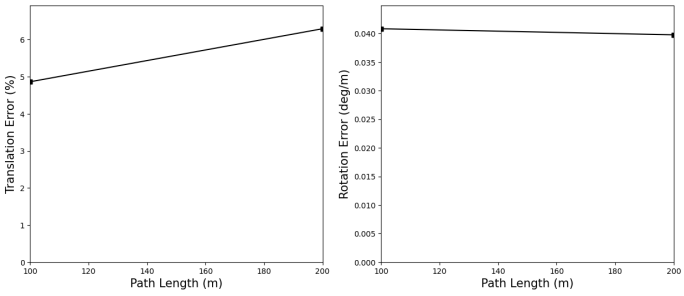


Fig. 7: Error graph for orb feature

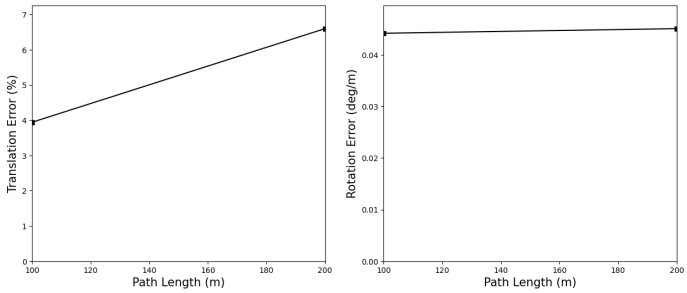


Fig. 8: Error graph for Superpoint and Superglue feature

V. Discussions

In this study, we proposed Super-VLOAM, a deep learning-based approach that improves the performance of Visual-LiDAR Odometry and Mapping by replacing classical feature extraction and matching algorithms with SuperPoint and SuperGlue models. Our experiments were aimed at demonstrating that Super-VLOAM achieves higher accuracy and robustness compared to the baseline VLOAM method, especially in challenging scenarios such as low-texture regions and dynamic motion.

We observed that Super-VLOAM has a high processing time due to the complex computations involved in deep learning models, which can be a limiting factor for real-time applications. The high processing time can prevent the extraction of features from every frame, resulting in incomplete feature maps, and a decrease in accuracy.

As future work, one potential solution to this limitation is to run Super-VLOAM on GPUs to decrease the processing time. This could significantly improve the performance of Super-VLOAM, allowing for real-time applications. Additionally, we could explore the use of other deep learning-based models that can reduce the computation time without compromising accuracy.

Furthermore, we evaluated Super-VLOAM on a limited number of sequences, including the KITTI dataset, and additional evaluation on more diverse datasets can provide a more comprehensive assessment of its performance. Future work can also explore the use of transfer learning techniques to fine-tune SuperPoint and SuperGlue models on different datasets and improve their generalization capability.

In conclusion, Super-VLOAM shows promise as a deep learning-based approach for Visual-LiDAR Odometry and Mapping. While it has a high processing time, future work can explore ways to optimize its computation time and evaluate its performance on more diverse datasets to further assess its potential for real-world applications.

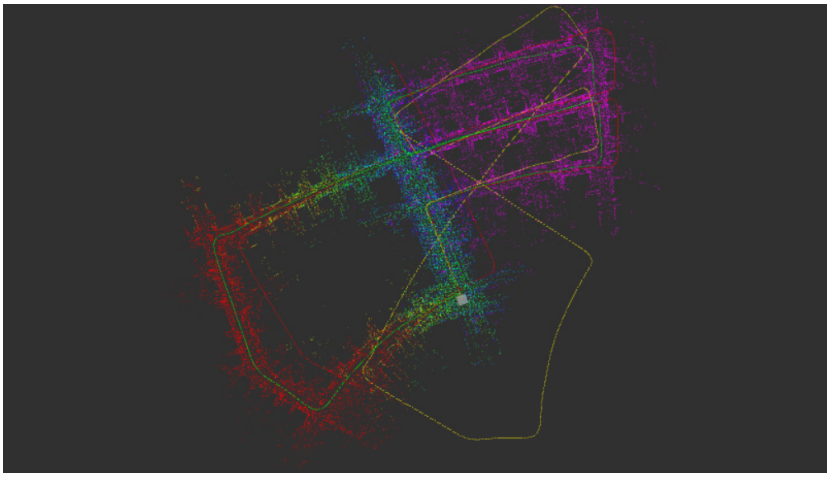


Fig. 9: Predicted Trajectory for VLOAM: Red path: Visual Odometry Yellow path: Lidar Odometry Green path: Mixed Odometry

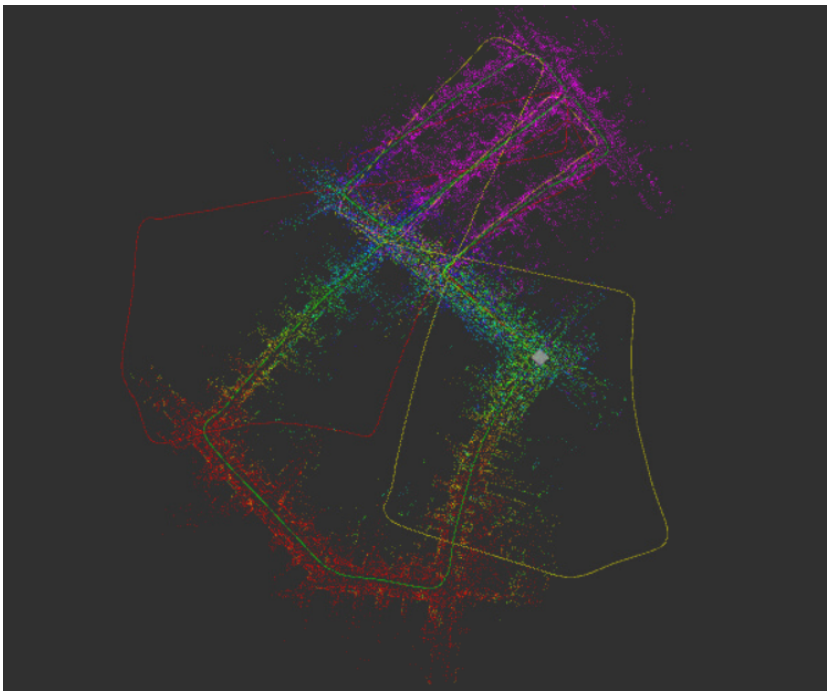


Fig. 10: Predicted Trajectory for Super VLOAM: Red path: Visual Odometry Yellow path: Lidar Odometry Green path: Mixed Odometr

REFERENCES

[1] J. Zhang and S. Singh, "Visual-lidar odometry and mapping: Low-drift, robust, and fast," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2174–2181, IEEE, 2015.

[2] J. Zhang and S. Singh, "Loam: Lidar odometry and mapping in real-time.,"

[3] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, "Orb-slam: A versatile and accurate monocular slam system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[4] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: an open-source SLAM system for monocular, stereo and RGB-D cameras," *CoRR*, vol. abs/1610.06475, 2016.

[5] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel, and J. D. Tardós, "Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, 2021.

[6] D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superpoint: Self-supervised interest point detection and description," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018.

[7] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superglue: Learning feature matching with graph neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[8] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.