Technology to Ensure Equitable Access to Automated Vehicles for Rural Areas

August 2023 | Final Report

SAFE-D
SAFETY THROUGH DISRUPTION

Texas A&M Transportation Institute

San Diego State University
Leadership Starts Here

VIRGINIA TECH TRANSPORTATION INSTITUTE
VIRGINIA TECH

Disclaimer

The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented herein. This document is disseminated in the interest of information exchange. The report is funded, partially or entirely, by a grant from the U.S. Department of Transportation's University Transportation Centers Program. However, the U.S. Government assumes no liability for the contents or use thereof.

TECHNICAL REPORT DOCUMENTATION PAGE

1. Report No. 06-004	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle Technology to Ensure Equitable Access to Automated Vehicles for Rural Areas		5. Report Date August 2023 6. Performing Organization Code:
7.Author(s) Stephen Ninan Sivakumar Rathinam		8. Performing Organization Report No.
9. Performing Organization Name and Address: Safe-D National UTC Texas A&M University Texas A&M Transportation Institute 3135 TAMU College Station, Texas 77843-3135 USA		10. Work Unit No. 11. Contract or Grant No. 69A3551747115/ Project 06-004
12. Sponsoring Agency Name and Address Office of the Secretary of Transportation (OST) U.S. Department of Transportation (US DOT)		13. Type of Report and Period Final Research Report Start: 9/2021 End: 8/2023 14. Sponsoring Agency Code

15. Supplementary Notes
This project was funded by the Safety through Disruption (Safe-D) National University Transportation Center, a grant from the U.S. Department of Transportation – Office of the Assistant Secretary for Research and Technology, University Transportation Centers Program.

16. Abstract
A significant majority of state-of-the-art autonomous sensing and navigation technologies rely on good lane markings or detailed 3D maps of the environment, making them more suited for urban communities. Conversely, many rural roads in the U.S. do not have lane markings and have irregular boundaries. These challenges are common to many small and rural communities (SRCs), which are sparsely connected and cover huge areas. The objective of this project was to develop an efficient sensing and navigation system for SRCs that uses crowd-sourced topological maps, such as OpenStreetMap, and provides high-level road network information in concert with onboard sensing systems that include lidar and cameras to localize and navigate an autonomous vehicle. The system was tested and validated on rural roads in an SRC around Bryan, TX.

17. Key Words Autonomous vehicles, rural roads, object detection, semantic segmentation, dataset, localization, particle filter	18. Distribution Statement No restrictions. This document is available to the public through the Safe-D National UTC website, as well as the following repositories: VTechWorks, The National Transportation Library, The Transportation Library, Volpe National Transportation Systems Center, Federal Highway Administration Research Library, and the National Technical Reports Library.

19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 29	22. Price $0

Form DOT F 1700.7 (8-72) Reproduction of completed page authorized

Abstract

A significant majority of the state of the art autonomous sensing and navigation technologies rely on good lane markings or detailed 3D maps of the environment and are more suited for urban communities. On the other hand, a large number of rural roads in the U.S. do not have lane markings and have irregular boundaries. These challenges are common to many small and rural communities (SRCs), defined as an incorporated city, town or village with a population of less than 50,000. These communities are sparsely connected and cover huge areas. The objective of this work is to develop an efficient sensing and navigation system for SRCs. To this end we develop a novel Rural Road Detection dataset for training and evaluation of sensing algorithms. We also propose road descriptors along with an initialization technique for localization that allows for fast global pose estimation on crowd sourced topological maps such as the Open Street Map (OSM). We test our algorithms on (real world) maps and benchmark them against other map based localization as well as SLAM algorithms. Our results show that the proposed method can narrow down the pose to within 50 centimeters of the ground truth significantly faster than the state of the art methods.

Acknowledgements

This project was funded by the Safety through Disruption (Safe-D) National University Transportation Center, a grant from the U.S. Department of Transportation – Office of the Assistant Secretary for Research and Technology, University Transportation Centers Program.

Table of Contents

List of Figures

List of Tables

Introduction

The current role of road transport cannot be overstated. It serves as a crucial lifeline, particularly in the United States, where it dominates as the primary mode of transportation. Despite approximately 81% of the American population residing in urban and suburban areas, rural roads account for about 68% of the country's road network [1]. This striking disparity in road distribution arises from the vast expanses of sparsely populated rural regions, resulting in a situation where there are more roads than people. To put it into perspective, there are 9,494 lane miles per 100,000 residents in rural areas, compared to a mere 1,056 in urban areas. At the same time, residents of rural communities must cover greater distances to access basic amenities.

The impact of significantly lower population density in rural areas extends to various aspects, including road safety. Traffic incidents on rural roads tend to have higher risk of death or injury due to limited access to emergency care. This stark reality is exemplified by the fact that approximately 43% of all highway fatalities occur on rural roads, despite only 19% of the population residing in these regions [1]. In addition, rural areas encompass 85% of counties where at least 20% of the population is over 65 years old [2]. Consequently, the development of a robust transportation system is crucial to cater to the needs of the aging population residing in these areas.

Moreover, considering that a significant portion of agricultural lands and industries are situated in rural areas, rural roads also play a vital role in facilitating freight transportation. These roads serve as conduits for substantial volumes of freight. In fact, it is estimated that rural roads account for approximately 46% of a truck's vehicular miles traveled [1].

The surge of research in the field of autonomous vehicles (AVs) holds great promise in terms of improved road safety, enhanced connectivity, better access to facilities, and efficient logistics and freight transportation. While these advantages are undoubtedly valuable in urban environments, they are even more crucial for small and rural communities (SRCs)—communities characterized by sparse connectivity and vast geographic coverage and defined as any incorporated city, town, or village with a population of less than 50,000. As of 2019, there were 18,723 SRCs in the U.S. [1, 2].

Following the Defense Advanced Research Projects Agency (DARPA) Grand Challenge in 2004, AVs have been highly researched, which is why significant progress has been made in their development. A prevailing trend in this domain has been the division of the self-driving task into distinct subtasks, including sensing, planning, and acting. Among these, the sensing or perception problem stands as the foundation of the entire system, as subsequent decisions and actions rely on the vehicle's ability to "see" its surroundings. In recent years, perception systems have increasingly incorporated machine learning and deep learning techniques [3], requiring a substantial amount of data for training purposes.

The scarcity of annotated rural datasets has resulted in limited progress in the development of perception systems capable of addressing the unique challenges and characteristics of rural roads. The significance of this issue is illustrated in Figure 1, showing a road detection model trained on the KITTI road dataset. While the model attained a respectable mean Intersection over Union (mIoU) score of 0.8 on the validation set, its performance significantly deteriorated when tested on rural roads. Specifically, the model failed to accurately detect the road surface, highlighting the fact that rural roads fall within an unfamiliar data distribution for the model.

(a) Rural road detections from model a trained on KITTI road dataset.

(b) Rural road detections after the model was retrained on rural road data.

Figure 1. Annotated photos. Domain adaptation problem in machine learning.

To address these challenges and facilitate advancements in AVs for rural areas, this research offers the following contributions:

1. Rural Road Detection Dataset (R2D2): By making the R2D2 accessible, we aim to stimulate the development and refinement of algorithms specifically designed to tackle the intricacies of rural road environments. R2D2 offers real-world data from rural roads, serving as a valuable resource to evaluate the performance and effectiveness of various sensing algorithms. This dataset will enable researchers and developers to assess their systems and compare them against standardized metrics.

2. Global Localization using Road Descriptors: We proposed a novel road descriptor as a concise feature representation for rural roads. We used these road descriptors to generate an initial pose belief, which was then passed to a particle filter for localization. The choice of initial belief significantly impacts the rate of convergence of particle filter-based localization algorithms, especially for global localization, where the search space is very large. The road descriptor search (RDS) technique we introduced helps in the selection of this initial belief. We demonstrated this through simulations and real-world tests by comparing the performance of the localization algorithm with and without the generation of an initial belief.

Background

As previously stated, in the realm of AVs, the prevailing trend has been to break down the self-driving task into distinct subtasks: sensing, planning, and acting, with the sensing or perception subtask being the most important. This is because subsequent decisions and actions heavily rely on the information gathered from the vehicle's surroundings. In recent years, perception systems have increasingly embraced the utilization of machine learning and deep learning methodologies [3]. However, an inherent challenge faced by deep learning-based methods is domain adaptation, which hinders their effectiveness and necessitates a substantial availability of extensive datasets.

The domain adaptation problem applies in the case of rural road scenes as well. The applicability of AVs in rural driving scenarios is less pronounced due to three key factors: the structure, scale, and sparsity of features, all of which are intricately linked to the nature of rural roads.

First, the structure of rural roads differs significantly from their urban counterparts. Urban roads generally exhibit a well-defined and consistent structure, whereas rural roads display considerable variations. These variations encompass inconsistent road markings and diverse road surfaces, such as gravel or dirt roads, in addition to the typical asphalt or concrete found in urban areas.

Another challenge arises from the scale and sparsity of features when employing dense maps for rural roads. Urban scenes are typically characterized by a plethora of distinctive features, including traffic signs, buildings, curbs, and lane markings, among others. Conversely, rural communities encompass vast areas with low population densities. Consequently, rural roads are predominantly surrounded by vegetation, with a scarcity of features crucial for localization. Figure 2 illustrates this disparity, where the urban scene exhibits numerous informative landmarks like buildings, sidewalks, and lane markings, whereas the rural scene lacks such prominent features.

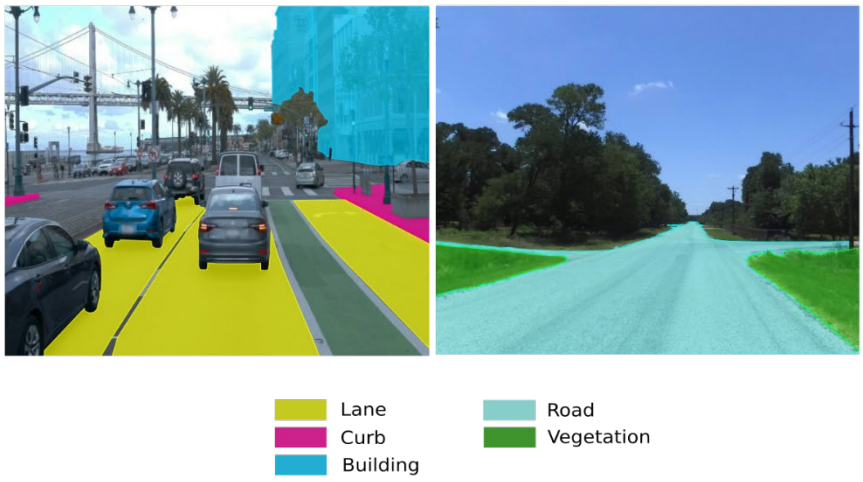


Figure 2. Color-coded images. Difference between urban (left) and rural (right) scenes.

Table 1 presents a comprehensive comparison of the prominent datasets used in the field of AVs. Over the past 15 years, with the advent of deep learning for computer vision, a substantial number of datasets have been released for evaluating perception algorithms. Initially, most datasets focused on object localization and classification in 2D images. However, in 2008, the CamVid dataset [4] emerged as one of the first datasets specifically designed for semantic segmentation. It comprised 700 images captured in the vicinity of Cambridge, UK, accompanied by pixel-wise annotations.

Table 1. Comparison of AV Datasets

Dataset	Year	Landscape	Camera	LiDAR (Channels)	RADAR	GPS,IMU	Annotations	Image Labels	Point cloud Labels
CamVid	2008	Urban	Mono	-	-	-	700	Semantic	-
Mapillary Vistas	2016	Urban, Rural	Mono	-	-	-	25K	Semantic	-
BDD100K	2017	Urban	Mono	-	-	-	100K	Semantic+ Bounding Box	-
CaRINA	2016	Urban, Suburban	Stereo	32C	Yes	GPS	900	Semantic	Semantic
KITTI	2012	Urban	Stereo	64C	No	Yes	15K	Bounding Box	Bounding Box
KAIST	2018	Urban	Stereo	16Cx2	No	Yes	9K	Bounding Box	Bounding Box
A2D2	2019	Urban	Mono	16Cx5	No	Yes	42K	Semantic	Semantic
nuScenes	2019	Urban	Mono	32C	Yes	Yes	40K	Semantic+ Bounding Box	Semantic+ Bounding Box
Waymo	2019	Urban	Mono	130C	No	Yes	200K	Semantic+ Bounding	Semantic+ Bounding
R2D2	2023	**Rural**	Stereo	128C	No	Yes	10.5K	Bounding Box	Semantic

Subsequently, numerous large-scale image-only datasets were introduced, with notable examples including BDD100K [5], Mapillary Vistas [6], and D2-City [7]. CityScapes [8] expanded on this concept by providing instance-level semantic labels for images collected from 50 different cities.

Further advancements in AV perception underscored the importance of utilizing multiple sensor modalities, particularly for challenging scenarios like driving in adverse weather conditions or at night when passive sensors such as cameras face significant limitations. Active sensors such as radar and lidar demonstrated improved performance in such situations while also offering redundancy within the system. This led to a demand for datasets that provide time-synchronized data from multiple sensors.

The KITTI dataset [9] was a pioneering effort in this direction, offering camera and lidar data from approximately 1.5 hours of urban driving. It included 2D bounding boxes for images and 3D bounding boxes for lidar point clouds. Building upon KITTI, the SemanticKITTI dataset [10] provided semantic labels for point clouds.

While KITTI primarily provided images from a front-facing camera, the NuScenes dataset [11] introduced the concept of scene understanding by offering 360-degree surround view data from

six cameras, lidar, and radar. The dataset also included 2D and 3D maps with detailed semantic annotations, collected in Singapore and Boston. The Waymo Open Dataset [12] utilized a 130-channel lidar and provided data from urban and suburban environments primarily in California.

It is evident that each of these datasets focuses on capturing data from urban landscapes. For rural roads, the CaRINA dataset [13] provides road detection data and benchmarks from selected urban and semi-urban scenarios in Brazil. However, its main emphasis lies in the detection of road surfaces in images and point clouds. Another dataset, HSI Road [14], offers hyperspectral images depicting various types of road surfaces, such as asphalt, gravel, and sand.

Based on our examination of the most popular driving datasets, we have identified several notable commonly shared characteristics:

- Multi-Modal Data: As previously emphasized, the integration of multiple sensing modalities leverages the strengths of different sensors, leading to the development of robust perception systems.
- Synchronization: It is crucial for the data collected from various sensors to be time synchronized. This synchronization enables a fair comparison between algorithms that rely on specific subsets of sensors.
- Calibration and Formatting: Accurate calibration of all sensors is essential, and the corresponding calibration parameters must be provided. Without proper calibration, the data cannot be effectively utilized. Furthermore, the formatting of both the data and annotations should facilitate easy integration into existing workflows.

To the best of our knowledge, no existing dataset for rural roads fulfills all the aforementioned requirements. In response to this gap, we have created the R2D2 to address this need.

R2D2: Rural Road Detection Dataset

Platform

For data collection purposes, we utilized a 2018 Lincoln MKZ vehicle that was retrofitted with a drive-by-wire system. In addition, we employed the following sensors:

- Ouster OS1-128: This high-resolution lidar sensor features 128 beams and a horizontal angular resolution of 0.3 degrees.
- Zed Stereo Camera: We employed a high-definition stereo camera capable of estimating depth within a range of up to 40 meters.
- Xsens MTi 680G: This inertial measurement unit (IMU) incorporates an accelerometer, gyrometer, and a built-in Global Navigation Satellite System (GNSS) receiver.

To illustrate our data collection platform and the sensor positioning, refer to Figure 3. To ensure a 360-degree field of view for the lidar, we elevated that sensor above the vehicle's roof using a

mounting fixture. The stereo camera is situated at the front of the car, positioned just below the lidar sensor to maximize the overlap in their respective fields of view. As depicted in Figure 3, the GPS+IMU unit is affixed to the vehicle's body, positioned above the rear axle.

Figure 3. Diagram. Data collection platform.

Route Selection

In contrast to urban roads, rural roads exhibit structural variations, including inconsistent, faded, or even absent road markings, as well as the absence of curbs and sidewalks. Moreover, the texture of the road surface differs, with gravel or dirt being prevalent alongside the asphalt or concrete commonly found on urban roads. When selecting routes for data collection, we prioritized those that effectively capture the diverse structural and surface characteristics observed on rural roads. All data collection occurred within the rural areas surrounding Bryan, Texas. Figure 4 provides a graphical representation of the routes employed for data collection. Broadly speaking, the dataset encompasses samples from three distinct types of road conditions:

- Highways: These roads exemplify well-structured thoroughfares, complete with lane markings and various traffic signs.
- Rural single-lane roads: This category encompasses roads bordered by vegetation on either side.
- Farmlands: The dataset also includes roads flanked by farmland on both sides, providing a unique context for analysis.

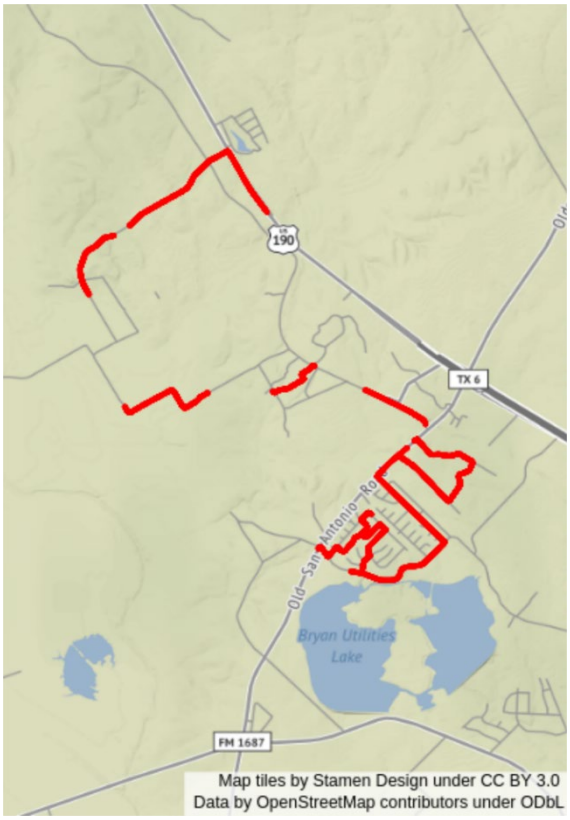


Figure 4. Map. Routes used for data collection.

Calibration

To calibrate the stereo cameras, we employ a checkerboard pattern, as depicted in Figure 5. The intrinsic parameters obtained from this calibration process are readily available with the dataset. As for the extrinsic calibration between the lidar and stereo camera, we employed a target-based calibration method proposed by Verma et al. [15]. With this method, we utilized a checkerboard as the calibration target. By isolating the points from the lidar that correspond to the checkerboard, along with their corresponding images, we established a set of 40 image-point cloud pairs for estimating the extrinsic parameters. Following the calibration procedure, we achieved a mean re-projection error of 7 pixels. Figure 5 illustrates an example of lidar points being projected onto a camera image after calibration.

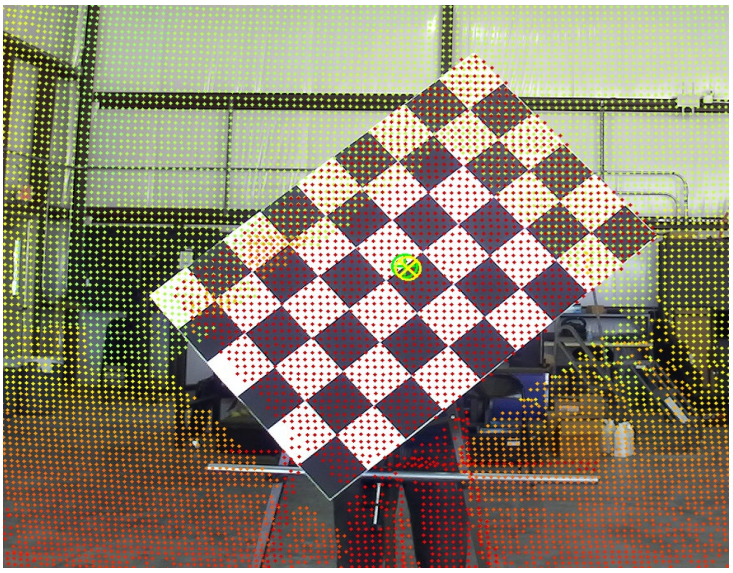


Figure 5. Photo and lidar points. Lidar camera calibration setup.

Annotations

For data annotation, we focused on three key classes that we considered most relevant to the self-driving task: the road, road users, and traffic signs. The road class encompasses all points in the environment that would typically be identified as a drivable surface by an average human driver. Road users include various sub-classes such as vehicles, pedestrians, cyclists, trucks, animals, and other types of vehicles. Lastly, the traffic signs class includes elements such as stop signs, traffic lights, warning signs, and signs related to road construction or maintenance. These classes were manually annotated in both the lidar point clouds and camera images. In the following section, we detail the annotation process and labeling procedures for the lidar point clouds.

Semantic Point Cloud Annotations

Given the high resolution of the lidar sensor employed in our dataset, a single point cloud contains a substantial amount of data, with 131,072 points. Annotating each point individually for every point cloud would be an arduous and time-consuming task. Instead, we adopted the approach proposed for the SemanticKITTI dataset [10], wherein point cloud registration is performed between successive frames. This registration allowed us to estimate the transformations between consecutive frames in the dataset, thus constructing a comprehensive map of the entire sequence. Consequently, all objects of interest within the map can be annotated simultaneously, significantly reducing the time required for labeling the point clouds.

Experiments and Results

To establish baseline results, we conducted training experiments using state-of-the-art models for semantic segmentation and object detection on our dataset. The following sections provide detailed explanations of these baseline experiments.

Semantic Segmentation

For the semantic segmentation baselines, we employed RangeNet++ [16], a fully convolutional model specifically designed for semantic segmentation. We trained the model using three different backbones: Darknet21 [17], Darknet53 [17], and Squeezeseg [18]. Two different sizes of input range images were used for training: 128x1024, which corresponds to the full-size images obtained from our point clouds, and 64x512. We use two different image sizes to provide baselines for model performance improvement by using a large image size. From the results in Table 2, we see that the performance improvement is marginal for Darknet21 with a 128x1024 image resolution. This suggests that most features are intact after reducing the image to 64x512 size.

Table 2. Results from Training Segmentation Model

Backbone	Image Size	Accuracy	mIoU
Darknet21	64x512	97.9	0.14
Darknet21	128x1024	98.02	0.15
Darknet53	64x512	98.5	0.235
SqueezesegV2	64x512	98.1	0.173

The loss function employed was weighted cross entropy, and Stochastic Gradient Descent (SGD) was used as the optimizer. The initial learning rate was set to $1e^{-2}$ and decayed at a rate of 0.05% per epoch. All models underwent training for 150 epochs, and we recorded performance metrics such as accuracy and mIoU. The results obtained from the training process are presented in Table 2. Additionally, Figure 6 provides an illustrative example showcasing the model predictions after training, alongside their corresponding ground truth values.

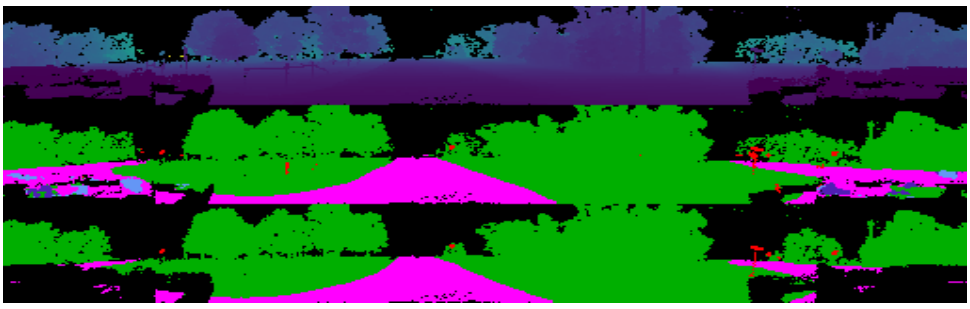


Figure 6. Color-coded images. Qualitative results from training Segmentation model on R2D2.

Object Detection

Since cameras are cheap and provide color-rich images, object detection in camera images serves to complement detections in lidar point clouds. This would be particularly useful for rural roads where obstacles may be occluded by vegetation. To establish baseline results, we conducted training experiments using state-of-the-art object detectors on the image data from R2D2. Specifically, we trained two variants of YOLOv5 [19], namely YOLOv5s and YOLOv5m. Each

model was trained using two different input image sizes: 640 pixels and 860 pixels. The model weights were initialized using pre-trained weights from training on the COCO Image dataset. The training data and validation data were split with a ratio of 95:5, respectively. All models underwent training for 300 epochs, with early stopping if there was no performance improvement in the last 100 epochs. Optimization was performed using SGD with a learning rate of 0.01. Other settings, such as anchor box scale and aspect ratios, were set to their default values.

The YOLOv5s model achieved a mean Average Precision (mAP50) score of 0.39 when using 860-sized input images. However, we observed that this model suffered from poor recall, particularly in detecting smaller objects that may be partially occluded. On the other hand, the YOLOv5m model achieved a mAP50 score of 0.41. Complete training results from the baseline experiments are provided in Table 3, and an example showcasing the results after training appears in Figure 7.

Table 3. Object Detection Results

Model	Image Size	mAP50	mAP0.5:0.95
$YOLOv5s$	640	0.33	0.16
$YOLOv5s$	864	0.39	0.21
$YOLOv5s$	1280	0.43	0.25
$YOLOv5m$	640	0.38	0.2
$YOLOv5m$	864	0.41	0.23

Figure 7. Annotated photos. Qualitative results from training Object Detection models on R2D2.

Global Localization using OpenStreetMap

Accurate pose estimation, which refers to determining the position and orientation of a mobile robot in its environment, is crucial for effective navigation. Similar to how humans rely on their understanding of the surroundings, robots rely on this ability, too. AVs often use detailed 3D maps created in advance to enhance their perception and estimate their pose based on sensor data. However, this approach is not well-suited for rural areas, which are sparsely connected and cover large regions.

In such cases, topological maps like OpenStreetMap [20] provide a valuable alternative. However, localizing a vehicle accurately using these maps, especially for global localization over large areas, is a challenging task. A GPS sensor could be used to aid localization, but GPS sensors provide low precision and may lose signal intermittently. Due to this, the team could not rely solely on a GPS sensor for localization. To address this challenge, we proposed the use of road descriptors along with an initialization technique for localization, enabling fast and reliable estimation of the vehicle's global pose. We evaluated our algorithms on real-world maps spanning areas as large as 36 square kilometers. The results demonstrate that our proposed method significantly outperforms state-of-the-art techniques, narrowing the estimated pose to within 1.5 meters of the ground truth in a much shorter time.

Methodology

The main task in the localization problem is to estimate a robot's pose in a predefined map. In other words, it means calculating the belief of a robot's pose, $bel(x_t)$ at time t, by combining information from previous sensor measurements $z_{1:t}$, control inputs $u_{1:t}$, and map information m. In other words,

$$p(x_t \mid z_{1:t}, u_{1:t}, m) = \frac{p(x_t, m) \sum_i^n p(x_t \mid u_t, x_{t-1}^i) bel(x_{t-1}^i)}{p(z_t \mid m)}$$

Road Descriptors

As shown in the above equation, the motion model relies on the previous state x_{t-1} and the input u_t. Since this is a recursive algorithm, the initial state needs to be set for the first iteration. The choice of initial belief has a significant impact on the convergence rate of Monte Carlo localization algorithms [21], especially for global localization tasks where the search space is extensive (this will be demonstrated later in the results section). To narrow down the search space and improve the initial belief, we proposed road descriptors that incorporate road geometry information at specific positions on the map. In rural scenes, roads serve as the primary visual features since they often lack consistent features such as buildings found in urban environments.

The road descriptor for a given point p on the map is represented by a 2D binary array D, where the rows correspond to the distances between p and the road features, and the columns correspond to the angles formed by the position vectors of the road features with respect to p. The values in D are generated through a ray casting operation, as depicted in Figure 8. To construct one row of D, a ray of length r is projected outward from p. This process is repeated for all angles θ in 1-degree intervals. The value in D is determined based on whether the ray intersects with a road point or not, following the specified rule:

$$D(r, \theta) = \{1 \, , if \; ray \; lands \; on \; a \; road \; point \; 0, \;\; otherwise$$

The outcomes of this computation for a given radius at two different road positions are depicted in Figure 8. By repeating this process for multiple radii, multiple rows can be generated, and the

road descriptor for point p can be calculated, as illustrated in Figure 9. In our framework, these road descriptors are precalculated for each node on the OpenStreetMap and stored in a lookup table for the initialization step.

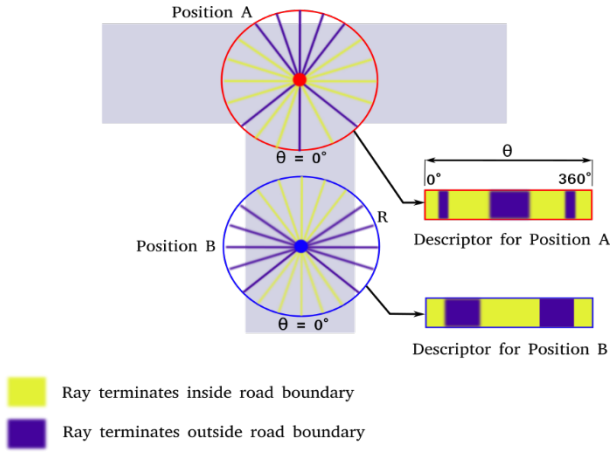


Figure 8. Diagram. Road descriptor generation.

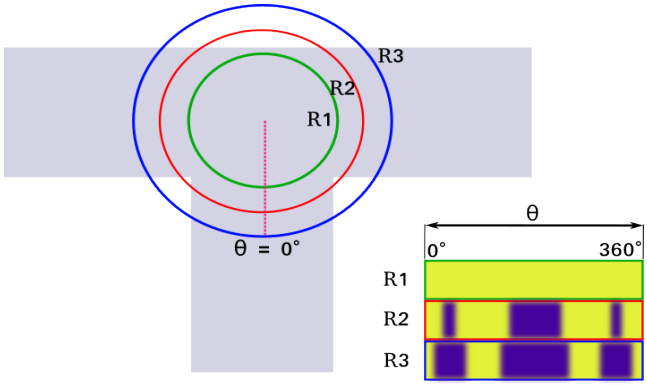


Figure 9. Diagram. 2D road descriptor.

The measurements z_t for the filter consists of lidar point clouds with labels indicating road points. To establish a correlation between point cloud data and road descriptors, the point clouds are transformed into lidar descriptors to enable a descriptor search during the initialization phase. This is achieved by projecting the point clouds onto a bird's-eye view (BEV) image, as shown in Figure 8. The lidar descriptor L is then generated using the same ray casting process described earlier for road descriptors. In this case, the ray casting starts from the center of the BEV image.

Now, to generate the initial belief $bel(x_{t0})$, a two-step approach is employed. Firstly, the lidar point cloud is converted into the descriptor form, which serves as the query descriptor L for the subsequent search steps. Initially, a search is conducted to identify the top 1,500 positions on the map where the road descriptors exhibit similarity to the query descriptor. Since descriptor values depend on orientation, they are flattened by calculating the row sum for each row and converting them into a one-dimensional vector, as shown in Figure 10. This step renders their rotation invariant. The similarity score used is defined as:

$$Similarity\ Score = \frac{1}{||L - D||_2}$$

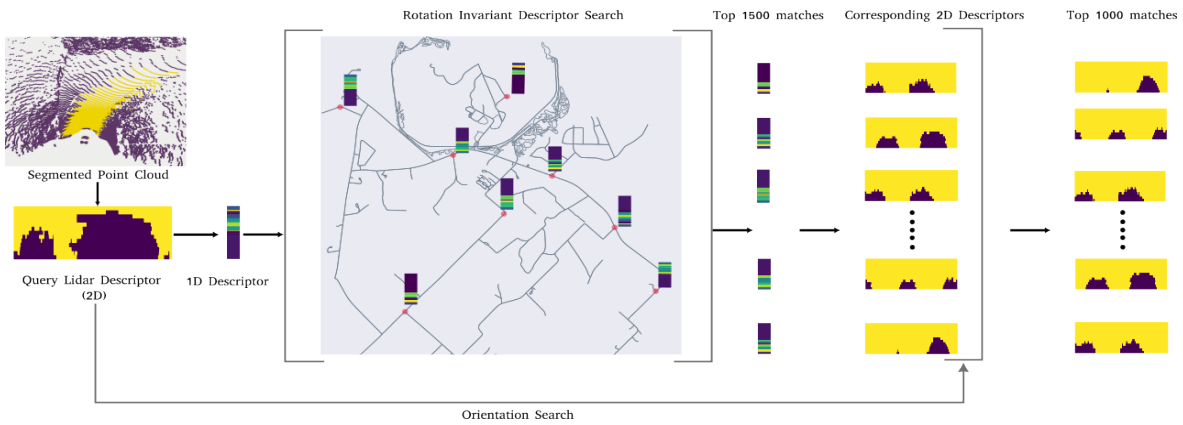


Figure 10. Diagram. Steps for initialization of particles.

In the second step, to further limit the number of matches for a given query descriptor, the orientation is taken into consideration. In this case, the complete 2D descriptors for L and R are utilized. To achieve this, for each position in S, all orientations ranging from 0 to 2π and their corresponding descriptors are examined. They are then sorted based on their similarity score, and the top 1,000 are chosen as potential poses for initializing the particle filter.

Regarding the motion model, we have the control input $u_t = [de, dn, d\theta]^T$, which represents the estimated change in pose between time steps $t-1$ and t.

The control input u_t and the initial estimates x_{t-1} are utilized to generate a pose hypothesis by sampling from the probability distribution as follows:

$$p(x_{t-1}) = N(x_t + u_t, q)$$

Measurement Model

In the measurement model, a distance function is employed to assign probability mass to each pose hypothesis $\underline{x_t}$ within the set $\underline{X_t}$.

To compute this distance function for a given pose hypothesis $\underline{x_t}$, we transform the segmented point cloud into the map frame using the pose represented by $\underline{x_t}$.

Based on the distance $d(p(road), i)$ between the nearest road edge and a road point $p(road)$ in the segmented cloud, we estimate $p(z_t|x_t, m)$ as follows:

$$p(z_t\,|x_t, m) = \prod_{i=1}^{i=n} \delta(d_{p_{road}}, i)$$

where n is the number of road points in the segmented point cloud and δ represents the distance function, which is a Gaussian with zero mean and covariance r:

$$\delta\left(d_{p_{road}}, i\right) = N(0, r)$$

The distance function returns the maximum probability for projected points $p_{road,i}$ that are close to road edges on the map. In effect, this means a greater probability mass will be assigned to poses where the road points $p_{road,i}$ are well aligned to the road geometry in the map.

The probabilities of all pose hypotheses $\underline{x_t} \in \underline{X_t}$ are updated in this way.

The poses are then sampled based on the assigned probability mass. Finally, the pose estimate x_t is then estimated as the weighted average over all poses.

$$x_t = \frac{\sum_i \quad p(x_i)x_i}{\sum_i \quad p(x_i)}$$

Experiments and Results

To showcase the benefits of utilizing road descriptors to initialize prior beliefs, we conducted two experiments. Initially, we employed OpenStreetMap to simulate lidar road detection. This was performed to assess the effectiveness of the road descriptor technique independently of the road segmentation algorithm's performance. Subsequently, we evaluated the global localization performance of the MapLite [22] algorithm using real-world data, both with and without the inclusion of road descriptors, to validate the effectiveness of the proposed approach.

Simulation Tests

To carry out the simulation, instead of utilizing real-world point clouds with road labels as described in the Methodology section, we generated a virtual BEV image from OpenStreetMap. This image provides a top-down snapshot of the map region surrounding the actual pose x_t. The virtual BEV image served as the basis for generating the lidar descriptor L as detailed in the Methodology section. Subsequently, the lidar descriptor was employed as the query descriptor for the RDS process.

To simulate a route, we utilized position and orientation information derived from OpenStreetMap nodes to generate ground truth measurements. For each ground truth measurement, the RDS was performed over a map segment spanning 36 sq. km. The lidar descriptor obtained from the virtual BEV image was then utilized as the query descriptor. For each query descriptor, we identified the top 1,500 poses on the map that exhibited similar descriptors. If the true pose was within 5 m of the top 1,500 matches, we considered the descriptor search to have converged.

The results of this simulation are illustrated in Figure 11. The segments of the route where the RDS converged are highlighted in red, orange, or yellow, based on the distance to the ground truth.

If RDS failed to converge within 15 meters of the ground truth position, the corresponding nodes along the route are left uncolored.

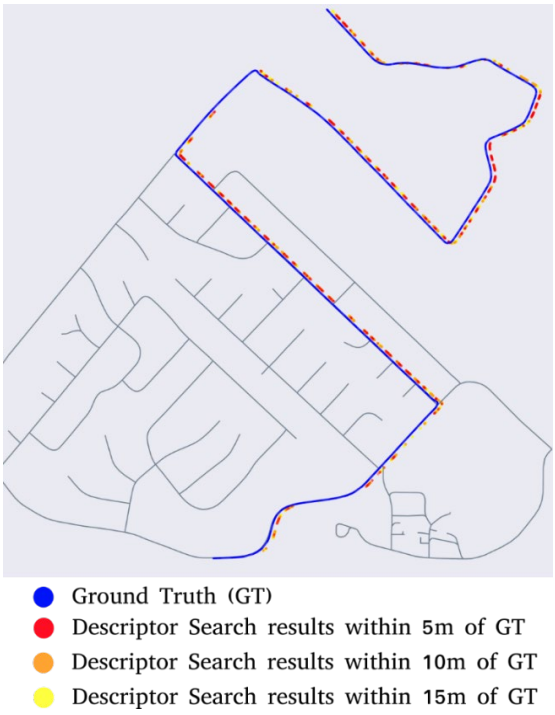


Figure 11. Map. Simulation results.

Analysis of the simulation results revealed that the descriptor search technique demonstrates the highest effectiveness in route segments featuring road attributes such as turns and intersections. However, it encounters difficulties in segments consisting solely of straight roads. This is particularly evident in the outcomes of the real-world test for Route 2 (Figure 13), where localization is lost in the straight portions of the route but subsequently regained in segments with intersections and turns.

Real-world Tests

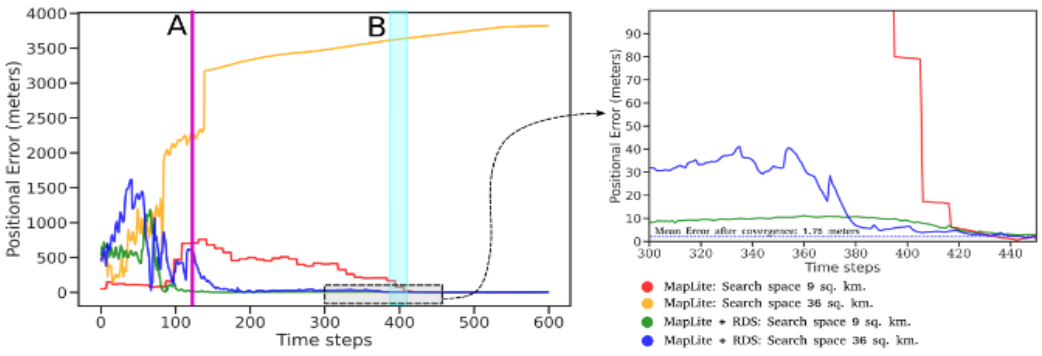
The performance evaluation of the particle filter, with and without the proposed initialization using RDS, was assessed using real-world data collected from two routes near Bryan, Texas. To gather the data, a test vehicle equipped with a 128-channel lidar sensor, a GNSS receiver with a horizontal sensing accuracy of 2.0 meters, and sensors for wheel speed and steering angle were employed.

To detect the road surface, a range image-based segmentation method inspired by RangeNet++ [13] was utilized. The model was trained on the Texas A&M AV Rural Road Dataset [23], which comprises 2,800 annotated range images specifically designed for road point detection. The dataset also includes vehicle bus data, such as GPS, IMU, steering, brake, throttle inputs, and wheel speed measurements. Altogether, the dataset encompasses approximately 15 minutes of recorded driving data.

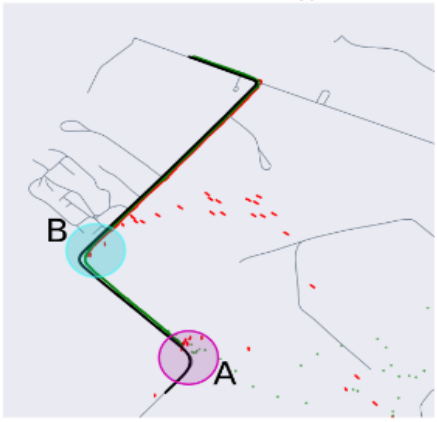
GPS measurements were intentionally excluded from all tests and only utilized as ground truth for performance evaluation purposes. The control input u_t was generated using a bicycle model based on measurements of wheel speed and steering angle. To optimize runtimes, the point clouds were downsampled using a voxel grid with voxel sizes of 2x2 meters, and the distance to the nearest edge was precomputed for all map points. Similarly, road descriptors were precomputed for all nodes on the map.

To emphasize the computational advantages of the proposed approach, the size of the search space on the map was varied. Two tests were conducted for each of the considered routes. The first test involved a search space of 9 sq. km, followed by a second test with a search space of 36 sq. km. For both tests, the MapLite algorithm was first initialized with 90,000 particles uniformly distributed across the entire search space. Subsequently, MapLite was tested again with RDS initialization, where particles were initialized around the top 1,500 matches returned by the descriptor search algorithm.

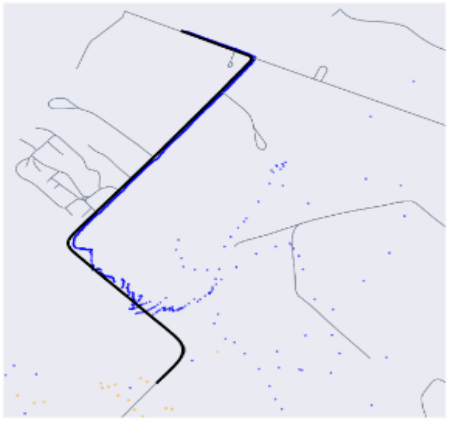
The first route (Route 1) consisted of data that the road segmentation model had previously encountered, as it was used during training. The localization results for Route 1 are depicted in Figure 12. It is evident that the time required for the MapLite algorithm to converge to the true position was considerably longer compared to when RDS was utilized for particle initialization.



(a) Positional error of Localization estimates - Route 1



(b) Position Estimates with initial search space of 9 sq.km



(c) Position Estimates with initial search space of 36 sq.km

Figure 12. Graphs and maps. Results from real-world tests - Route 1.

The utilization of RDS initialization demonstrates a notable reduction in time to convergence for both the 9 sq. km and 36 sq. km search spaces. In Figure 12, during the 9 sq. km search test, it is evident that the MapLite algorithm converged only after encountering road features up to turn B, as indicated by the decrease in error after approximately 400 time steps, where each time step is 100 milliseconds. Conversely, when RDS initialization was employed, the algorithm rapidly converged after around 100 time steps, corresponding to the vehicle reaching turn A, showcasing a significantly faster convergence compared to the previous scenario.

Due to the larger search space, the algorithm requires more time to converge for the 36 sq. km search space. As depicted in Figure 12c, when the algorithm is initialized without the descriptor search, it converges to an erroneous position on the map, leading to substantial position errors. This signifies that the number of particles used was insufficient relative to the size of the search space. However, when road descriptors are incorporated, the position estimate aligns closely with the ground truth after the vehicle reaches turn A, albeit with a slight degree of error remaining. This is attributed to a few particle clusters concentrated in other regions of the map. However, as the vehicle progresses to turn B, these alternative clusters are subsequently eliminated, resulting in the algorithm converging to the ground truth position.

Route 2 poses a more demanding test case due to the increased complexity and density of the road network in the region. Furthermore, this route was not previously encountered by the road segmentation model. Similar to Route 1, we conducted two tests using the MapLite and MapLite + RDS algorithms, with the results displayed in Figure 13.

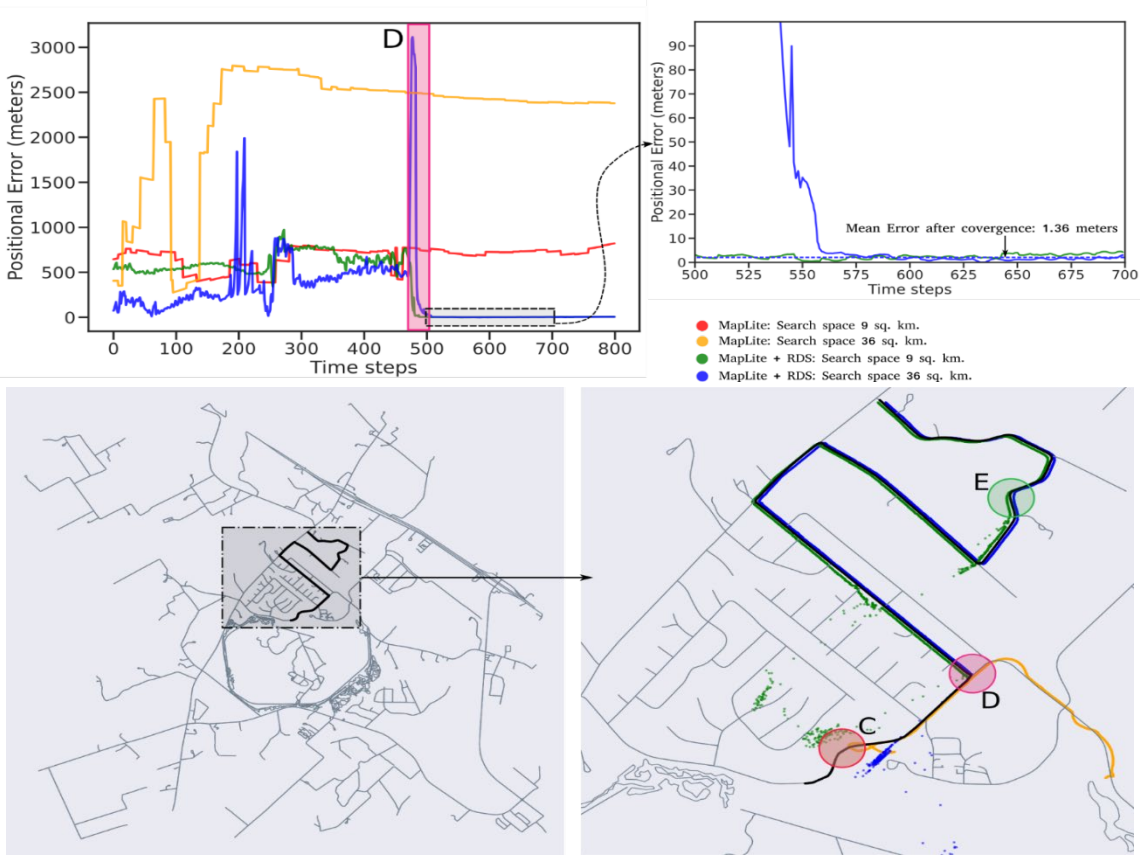


Figure 13. Graphs and maps. Results from real-world tests - Route 2.

It is evident that the MapLite algorithm alone fails to converge in both the 9 sq. km and 36 sq. km test scenarios, reaffirming that the number of particles employed was insufficient in relation to the size and intricacy of the considered map. However, when road descriptors are incorporated in both tests, the algorithm successfully converges to the actual vehicle position. The notable fluctuations in positional errors observed in Figure 13 during the initial phase can be attributed to the presence of particles in other parts of the map that exhibit similar road geometry, such as turns C and E illustrated in Figure 13. Nevertheless, as the vehicle reaches the intersection at D, those particle clusters are eliminated, leading to the convergence of the pose estimate to the ground truth position after approximately 500 time steps. Furthermore, the proposed algorithm can run in real time, as the computational time for each iteration is less 100 milliseconds, which is less than the time between two lidar measurements (~100-120 milliseconds).

Conclusions and Recommendations

Conclusion

In this project, we introduced the R2D2, a pioneering resource for evaluating AV perception algorithms specifically designed for rural road scenarios. What sets this dataset apart is its focus on rural road environments and its utilization of a synchronized multi-modal sensor suite

(including lidars, cameras, GPS, and IMU sensors). This research went beyond sensor data by providing semantic labels for point clouds and object labels for camera images, thereby enabling comprehensive perception model training. By making this dataset available to the research community, our aim is twofold: to drive advancements in machine vision and AV perception and to ensure the inclusivity and accessibility of AV technologies in rural communities.

Furthermore, we present a localization algorithm designed to enable global localization on rural roads. Our proposed algorithm leverages road descriptors to generate an initial belief, thereby augmenting the state-of-the-art approaches. This algorithm serves as the foundation for implementing a lidar-based, GPS-denied localization technique for global localization. Experimental results demonstrate that our algorithm can accurately recover a vehicle's pose, achieving a mean error of 1.5 meters even in expansive maps spanning up to 36 sq. km. The performance improvements facilitated by the RDS initialization make this method particularly suitable for addressing the kidnapped robot problem - a situation where we do not have a good initial guess of the robots pose.

Future Work

While this work makes significant contributions in the field of rural road perception and localization, there are several avenues for further exploration and improvement. Some potential directions for future research include:

1. Integration of additional sensor modalities: Although our dataset incorporates lidar, camera, GPS, and IMU sensors, future work could explore the integration of other sensor modalities, such as radar or thermal imaging, to further enhance perception capabilities in rural environments. By combining multiple sensor inputs, the robustness and accuracy of perception algorithms can be improved.
2. Advanced road descriptor techniques: While the utilization of road descriptors has proven effective in our localization algorithm, there is room for exploring more sophisticated road descriptor techniques. Investigating advanced feature extraction methods and incorporating contextual information can potentially enhance the accuracy and robustness of the descriptor matching process.
3. Dynamic mapping and localization: Our current approach focuses on global localization in static rural road environments. Future research could extend this work to address dynamic mapping and localization challenges, such as handling moving obstacles or road construction zones. Developing algorithms that can adapt to dynamic changes in the environment will be crucial for real-world deployment of AVs in rural areas.
4. Optimization of runtime efficiency: While our algorithm demonstrates promising performance, there is potential for optimizing the runtime efficiency. Exploring techniques such as parallelization, distributed computing, or efficient data structures can help reduce computational overhead and enable real-time operation in resource-constrained settings.

5. Real-world deployment and validation: To validate the effectiveness of the proposed algorithm in practical scenarios, it is essential to conduct real-world deployment and rigorous field testing. Future work should focus on conducting extensive experiments and evaluations on different rural road datasets and under various environmental conditions to assess the algorithm's robustness and generalizability.

By pursuing these avenues for future research, we can further advance the state-of-the-art in rural road perception and localization, ultimately bringing us closer to the realization of safe and reliable AVs in rural areas.

Additional Products

Education and Workforce Development Products

- Graduate Student (Stephen Ninan) had his M.S. thesis supported by this Safe-D project.
- Graduate Student (Stephen Ninan) volunteered as a Finals Judge for Texas Junior Science and Humanities Symposium in April 2022.

Technology Transfer Products

1. The R2D2 has been submitted to the IEEE Intelligent Transportation Systems Conference (ITSC) and is under consideration.
2. The work on global localization is also under review with the IEEE ITSC. It is available at the following link: https://arxiv.org/abs/2202.07049
3. Graduate student (Stephen Ninan) presented the work from this project to multiple OEMs, including Rivian Automotive and Lucid Motors.
4. Stephen Ninan presented an invited talk on this project at the Transportation Research Board's Automated Road Transportation Symposium (ARTS) in July 2022. Title of the presentation: AVs in Rural America.

Data Products

1. Robot Operating System (ROS) Package for this project with all code and drivers: https://github.com/nsteve2407/osm-localization
2. Dataset collected over the course of this project:
    a. AV Rural Road Dataset was uploaded to the VTTI Dataverse and is accessible at: https://doi.org/10.15787/VTT1/AOHI5N
    b. R2D2 is available on the following website: https://autonomy.engr.tamu.edu/r2d2/

References

[1] https://www.bts.gov/rural, "Rural Transportation Statistics," 16 August 2022. [Online]. Available: https://www.bts.gov/rural. [Accessed 30 May 2023].

[2] J. Cromartie, "Rural Aging Occurs in Different Places for Very Different Reasons," 20 December 2018. [Online]. Available: https://www.usda.gov/media/blog/2018/12/20/rural-aging-occurs-different-places-very-different-reasons. [Accessed 30 May 2023].

[3] "[1910.07738] A Survey of Deep Learning Techniques for Autonomous Driving," 17 October 2019. [Online]. Available: https://arxiv.org/abs/1910.07738. [Accessed 30 May 2023].

[4] G. J. Brostow, J. Shotton, J. Fauqueur, R. Cipolla and G. Brostow. [Online]. Available: http://www0.cs.ucl.ac.uk/staff/G.Brostow/bibs/RecognitionFromMotion_bib.html. [Accessed 30 May 2023].

[5] "BDD100K: A Diverse Driving Dataset for Heterogeneous Multitask Learning," [Online]. Available: https://openaccess.thecvf.com/content_CVPR_2020/papers/Yu_BDD100K_A_Diverse_Driving_Dataset_for_Heterogeneous_Multitask_Learning_CVPR_2020_paper.pdf. [Accessed 30 May 2023].

[6] "Mapillary," [Online]. Available: https://www.mapillary.com/dataset/citation. [Accessed 30 May 2023].

[7] "[1904.01975] D^2-City: A Large-Scale Dashcam Video Dataset of Diverse Traffic Scenarios," 3 April 2019. [Online]. Available: https://arxiv.org/abs/1904.01975. [Accessed 30 May 2023].

[8] "[1604.01685] The Cityscapes Dataset for Semantic Urban Scene Understanding," 6 April 2016. [Online]. Available: https://arxiv.org/abs/1604.01685. [Accessed 30 May 2023].

[9] A. Geiger, P. Lenz and R. Urtasun, "The KITTI Vision Benchmark Suite," [Online]. Available: https://www.cvlibs.net/datasets/kitti/. [Accessed 30 May 2023].

[10] "[1904.01416] SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences," 2 April 2019. [Online]. Available: https://arxiv.org/abs/1904.01416. [Accessed 30 May 2023].

[11] "Please use the following citation when referencing nuPlan," [Online]. Available: https://www.nuscenes.org/publications#. [Accessed 30 May 2023].

[12] K. H. D. X. C. A. P. V. T. P. G. J. Z. Y. C. Y. C. B. V. V. Sun P, "Scalability in perception for autonomous driving: Waymo open dataset.," *IEEE/CVF conference on computer vision and pattern recognition 2020,* 2020.

[13] d. S. T. R. L. R. D. M. C. A. F. B. M. H. A. O. F. W. D. Shinzato PY, "CaRINA dataset: An emerging-country urban scenario benchmark for road detection systems.," *IEEE 19th international conference on intelligent transportation systems,* 2016.

[14] L. H. Y. Y. T. S. T. Z. L. J. Lu J, "Hsi road: A hyper spectral image dataset for road segmentation," *IEEE International Conference on Multimedia and Expo (ICME),* 2020.

[15] "[1904.12433] Automatic extrinsic calibration between a camera and a 3D Lidar using 3D point and plane correspondences," 29 April 2019. [Online]. Available: https://arxiv.org/abs/1904.12433. [Accessed 30 May 2023].

[16] A. Milioto, "RangeNet ++: Fast and Accurate LiDAR Semantic Segmentation," *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).*

[17] A. F. Joseph Redmon, "YOLO9000: Better, Faster, Stronger," *arXiv,* 2016.

[18] A. W. X. Y. K. K. Bichen Wu, "SqueezeSeg: Convolutional Neural Nets with Recurrent CRF for Real-Time Road-Object Segmentation from 3D LiDAR Point Cloud," *arXiv,* 2017.

[19] "ultralytics/yolov5: v7.0 - YOLOv5 SOTA Realtime Instance Segmentation," 22 November 2022. [Online]. Available: https://doi.org/10.5281/zenodo.3908559. [Accessed 30 May 2023].

[20] O. contributors, "OpenStreetMap," 2017. [Online]. Available: https://www.openstreetmap.org/.

[21] W. B. F. D. S. T. Dieter Fox, "Monte Carlo Localization: Efficient Position Estimation for Mobile Robots," 1998.

[22] T. K. M. R. B. S. K. G. D. B. I. G. L. P. a. D. R. Ort, "Maplite: Autonomous intersection navigation without a detailed prior map," *IEEE Robotics and Automation Letters,* 2020.

[23] S. Ninan and S. Rathinam, "Autonomous Vehicle Rural Road Dataset," [Online]. Available: https://dataverse.vtti.vt.edu/dataset.xhtml?persistentId=doi:10.15787/VTT1/AOHI5N.