A-RoI: Adaptive Region of Interest to Enhance Obstacle Awareness for Autonomous Driving

C.A. Rakshith Ram[1], Abhishek Thakur[2], and P. Rajalakshmi[2]
[1]*Center for Interdisciplinary Programs*, [2]*Department of Electrical Engineering*
Indian Institute of Technology Hyderabad, India
email: sm22mtech12003@iith.ac.in, ee20resch11014@iith.ac.in, raji@ee.iith.ac.in

***Abstract*—Autonomous vehicles navigating dynamic environments demand adaptive perception and planning for safe operation. Crucially, navigational decisions rely on understanding the position and behaviour of surrounding objects. Traditional, static Regions of Interest (RoIs) for obstacle detection prove inadequate in meeting these dynamic and context-dependent requirements. This paper presents Adaptive Region of Interest (A-RoI), a novel technique that dynamically updates the RoI on a LiDAR HD map based on the vehicle's real-time position and planned trajectory. Integrating obstacle positions with the A-RoI within a Bird's Eye View (BEV) framework enables effective obstacle perception, significantly enhancing navigational decision-making. Real-time validation on a sensor-equipped autonomous vehicle demonstrates the effectiveness of the proposed approach. Future research will explore enhancing A-RoI's adaptability through multi-modal sensor fusion and incorporating learning-based methodologies for further improvements in perception and decision-making capabilities.**

***Index Terms*—Adaptive Region of Interest (A-RoI), LiDAR HD map, Bird's Eye View (BEV), Obstacle perception.**

I. Introduction

A motion planning system determines a sequence of high-level driving actions or manoeuvres to safely accomplish a vehicle's driving mission under various road conditions. The planned manoeuvres must account for traffic rules, road structure, and interactions with both static and dynamic objects in the environment. The planner's high-level decisions should ensure not only vehicle safety but also efficient motion through the environment.

To achieve this, the behaviour/motion planner requires an accurate understanding of the vehicle's position, the location of obstacles relative to the vehicle, and an assessment of whether the planned path is safe to follow. This enables informed decision-making for navigation, obstacle avoidance, and adaptive manoeuvring in real-world scenarios.

One key advantage of Adaptive RoI (A-RoI) is its ability to filter out obstacles that do not lie within the vehicle's intended path. This is particularly useful in scenarios involving sharp turns, where a static RoI may fail to account for the vehicle's trajectory, potentially misidentifying obstacles as relevant when they do not pose an immediate threat. By dynamically adjusting to the vehicle's movement, A-RoI ensures more accurate obstacle perception, reducing false positives and improving decision-making for safe navigation.

The code and a demonstration video will be made available here following the publication of this paper.

Fig. 1 illustrates various obstacle perception scenarios where the proposed Adaptive Region of Interest (A-RoI) may profoundly help motion planning in autonomous vehicles. In a scenario depicted by (a), a static RoI may correctly detect an obstacle directly in the vehicle's path. However, in (b), when the planned trajectory involves a curve, the static RoI remains fixed relative to the vehicle, leading to false positive detections of obstacles outside the actual path. In (c), when an obstacle is close to the planned trajectory, slowing down may be necessary, and the A-RoI effectively provides feedback in such cases. Additionally, in (d), when the vehicle needs to deviate from its original path to avoid obstacles, the A-RoI monitors approaching vehicles from the rear, which aids in safe manoeuvring. Incorporating such dynamic feedback from the A-RoI significantly aids the motion planning process, leading to more adaptive and context-aware decision-making.

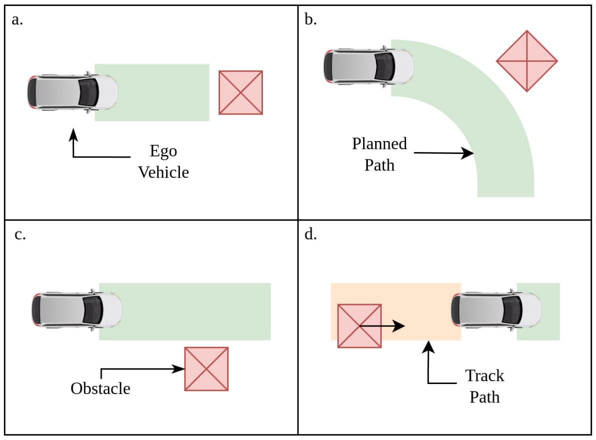


Fig. 1: Various scenarios depicting the potential use cases of using A-RoI.

In this paper, we provide a detailed discussion of the components of A-RoI and the methodology used to generate it. We also explain how the A-RoI and obstacles perceived around the vehicle are transformed into a Bird's Eye View (BEV) representation. All of the steps mentioned above are essential in enhancing motion and behaviour planning, which is the main goal of the proposed method.

II. Literature Survey

The Bird's Eye View (BEV) is a standard representation in autonomous driving, simplifying object detection, tracking, and path planning. Deep learning-based BEV object detection has achieved state-of-the-art results [1]. Combining BEV with adaptive RoIs can further enhance perception.

LiDAR sensors provide rich 3D information about the environment, making them essential for autonomous vehicle perception. Simultaneous Localization and Mapping (SLAM) techniques, such as LOAM and its variants [2], [3], [4], enable the creation of high-definition (HD) maps and accurate vehicle localization. These HD maps serve as the foundation for many perception and planning algorithms. [5] demonstrates the use of LiDAR data for robust object detection and tracking in challenging urban environments.

Fusing data from multiple sensors (e.g., LiDAR, cameras, radar) can improve the robustness and accuracy of perception systems. [6] proposed a multi-sensor fusion approach for object detection.

Recent works, such as [7], have explored deep learning methods for road boundary extraction, which can be used to define RoIs; however, these approaches often incur significant computational costs. Unlike prior methods, our proposed Adaptive Region of Interest (A-RoI) is computationally light and dynamically updates RoI boundaries in real-time, integrating obstacle positions within a Bird's Eye View (BEV).

III. Methodology

This section details the methodology employed for the ego-vehicle localization and pose extraction, the construction of the A-RoI's various components, obstacle perception with Principal Component Analysis (PCA), coordinate transformations, and finally, the process of visualizing the A-RoI and obstacles in a Bird's Eye View (BEV). The process is as outlined in Fig. 2.

A. Ego Vehicle Localization and Pose extraction

The autonomous vehicle used in this implementation is equipped with a Velodyne VLP-16 sensor for mapping and localization purposes and a Livox HAP sensor for obstacle/object detection. The system utilizes a high-definition (HD) LiDAR map generated using the A-LOAM (Advanced LiDAR Odometry and Mapping) algorithm [2], which processes LiDAR point clouds to create an accurate environmental representation. Localization is then performed using the NDT (Normal Distributions Transform) algorithm [8], which aligns real-time LiDAR scans with the HD map to estimate the vehicle's precise position and orientation, ensuring accurate navigation. Additionally, predefined waypoints are provided for the vehicle to follow. Once localization is established, the vehicle's pose parameters $[x, y, z, \phi, \theta, \psi]$ are extracted. The position coordinates (x, y, z) are obtained from the "`/ndt_pose`" ROS (Robot Operating System) topic, while the roll, pitch, and yaw angles (ϕ, θ, ψ) are derived from the "`/euler_angle`" topic published by the NDT Localizer.

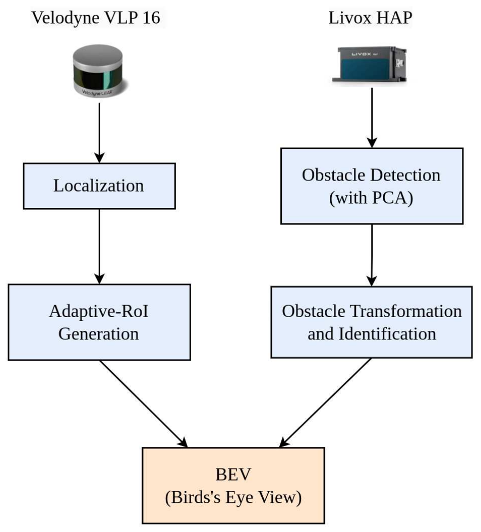


Fig. 2: Flowchart of the methodology followed.

B. Generating various components A-RoI

The Adaptive Region of Interest (A-RoI) consists of five key components, primarily designed to monitor critical areas around the vehicle for safe navigation. These components include:

a). Forward Path RoI – This region covers the planned trajectory that the vehicle will follow, ensuring a clear and obstacle-free path. If any obstacles are detected within this region, the ego vehicle's velocity can be adjusted accordingly. The RoI is constructed using upcoming waypoints as a reference, forming a shape that aligns with the trajectory. The width and length of this RoI can be adjusted as parameters.

b). Lateral Right RoI – Positioned to the right of the planned path, this region helps detect obstacles that may interfere with the vehicle's movement. If an obstacle is detected near the path, the ego vehicle can reduce its speed as necessary to ensure safe navigation.

c). Lateral Left RoI – Similar to the Lateral Right RoI, this region monitors obstacles on the left side of the planned trajectory. If obstacles are detected in either the left or right RoI, the system can take corrective actions, such as slowing down the vehicle to avoid potential collisions.

d). Rear Monitoring RoI – This region is used to monitor traffic approaching from behind, which is particularly important when the autonomous vehicle needs to change lanes or overtake an obstacle. The RoI is built using waypoints that the ego vehicle has already traversed as a reference.

e). Ego-Vehicle Safety RoI – Defined as a rectangular area surrounding the ego vehicle, this RoI acts as an emergency buffer zone. If an obstacle enters this region, the system can trigger emergency braking to prevent collisions. This RoI dynamically adjusts according to the ego vehicle's pose, and its dimensions can be modified as needed.

The mathematical framework for constructing the components follows this structure: For each waypoint, starting from

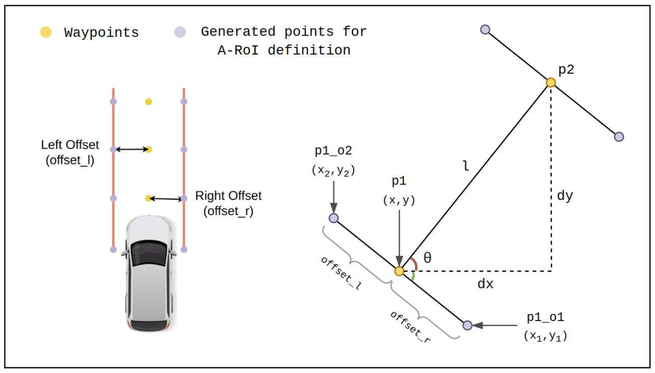


Fig. 3: Geometry of the keypoint construction for A-RoI.

the nearest waypoint to the current position up to a defined window length, the right and left offset points are computed as follows (depicted in Fig. 3:

Let the Cartesian coordinates of any two consecutive waypoints be:

$$p_n \equiv (x_n, y_n) \quad \text{and} \quad p_{n+1} \equiv (x_{n+1}, y_{n+1}) \tag{1}$$

The coordinates of $p_n\_r \equiv (x_n\_r, y_n\_r)$, which is the right offset point, can be determined by:

$$x_n\_r = x_n + (\text{offset}\_r * dy/l) \tag{2}$$

$$y_n\_r = y_n + (\text{offset}\_r * dx/l) \tag{3}$$

Similarly, the coordinates of point $p_n\_l \equiv (x_n\_l, y_n\_l)$, which is the right offset point, can be determined by:

$$x_n\_l = x_n - (\text{offset}\_l * dy/l) \tag{4}$$

$$y_n\_l = y_n + (\text{offset}\_l * dx/l) \tag{5}$$

where,

$$dx = (x_{n+1} - x_n) \quad \text{and} \quad dy = (y_{n+1} - y_n) \tag{6}$$

$$l = \sqrt{dx^2 + dy^2} \tag{7}$$

Once the offset points are determined for each waypoint within the specified window, the Adaptive Region of Interest (A-RoI) is constructed by forming a set of rectangles between consecutive offset points. Specifically, each rectangle is defined using the four points $(p_n\_l, p_n\_r, p_{n+1}\_r, p_{n+1}\_l)$. These rectangles collectively form a continuous adaptive region that dynamically adjusts based on the vehicle's planned trajectory.

The Forward Path RoI, Lateral Right RoI, Lateral Left RoI, and Rear Monitoring RoI are all constructed using the same geometric principles. Each of these RoIs is influenced by three key tunable parameters: *offset_right*, which defines the lateral extension to the right; *offset_left*, which defines the lateral extension to the left; and *window_size*, which determines the number of waypoints considered in forming the A-RoI. These parameters allow for flexibility in adjusting the size and coverage of the RoIs based on the vehicle's navigation and environmental conditions.

In addition to these regions, a separate Ego Vehicle Safety RoI is defined. It is a fixed rectangular region surrounding the vehicle. This safety RoI is customizable, with its four defining coordinates referenced from the LiDAR mounting point, ensuring that it accounts for the actual vehicle dimensions and required safety margins. It serves as a static buffer zone around the vehicle to ensure obstacle-free navigation and safe manoeuvring.

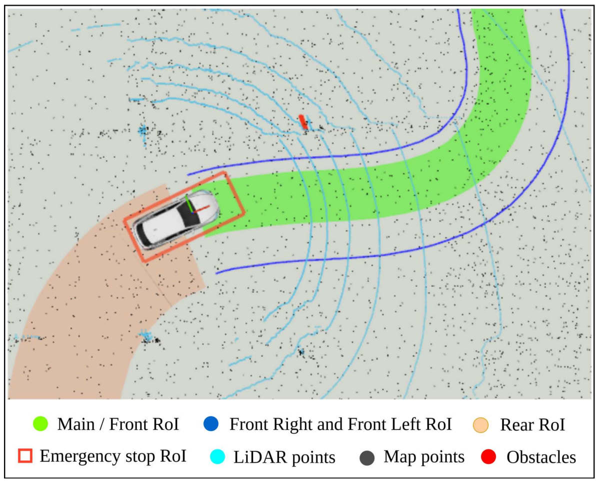


Fig. 4: Bird's Eye View (BEV) of the ego vehicle, A-RoI and obstacles visualized on a LiDAR HD map using RViz.

C. Obstacle perception and Principle Component Analysis (PCA) application

The Livox HAP LiDAR sensor is used for obstacle detection in this implementation. A DBSCAN-based clustering method is used to detect clusters in the point cloud after ground points have been filtered out. This implementation also uses Principal Component Analysis (PCA), a useful tool for identifying the primary orientation and dimensions of an object or cluster in point cloud data. The goal is to compute the Oriented Bounding Box (OBB) for the input 3D LiDAR point cloud cluster. The methodology consists of the following steps:

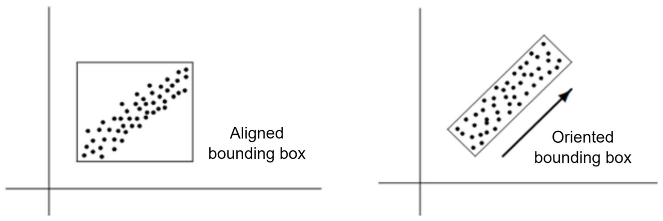


Fig. 5: Comparison of axis-aligned and oriented bounding boxes for a set of data points.

First, the centroid of the point cloud is computed as the arithmetic mean of all the points. Let $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ represent the set of n points, where each point $\mathbf{x}_i =$

(x_i, y_i, z_i) is a 3D coordinate. The centroid $\mathbf{c}$ is calculated as:

$$\mathbf{c} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i \tag{8}$$

Next, the point cloud is centred by subtracting the centroid from each point, resulting in the centred point set $\mathbf{X}' = \{\mathbf{x}'_1, \mathbf{x}'_2, \ldots, \mathbf{x}'_n\}$, where:

$$\mathbf{x}'_i = \mathbf{x}_i - \mathbf{c} \tag{9}$$

The covariance matrix $\mathbf{C}$ is then computed for the centred points, capturing the spread and correlation of the data in 3D space. The covariance matrix $\mathbf{C}$ is given by:

$$\mathbf{C} = \frac{1}{n-1}\sum_{i=1}^{n} \mathbf{x}'_i (\mathbf{x}'_i)^T \tag{10}$$

Eigenvalue decomposition is performed on the covariance matrix to find the eigenvectors $\mathbf{v}_i$ and eigenvalues λ_i. The eigenvectors represent the directions of the largest variances in the point cloud and form the principal axes for the OBB:

$$\mathbf{C}\mathbf{v}_i = \lambda_i \mathbf{v}_i \tag{11}$$

These eigenvectors $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$ define the orientation of the OBB. To align the point cloud with these principal directions, the centred points are transformed into the new coordinate system defined by the eigenvectors. This transformation is achieved by multiplying the centred points by the transpose of the matrix $\mathbf{V}$ (which is formed by the eigenvectors):

$$\mathbf{x}''_i = \mathbf{V}^T \mathbf{x}'_i \tag{12}$$

In the transformed space, the bounding box is calculated by determining the minimum and maximum extents of the points along each of the principal axes:

$$\mathbf{min}_j = \min(\mathbf{x}''_i), \quad \mathbf{max}_j = \max(\mathbf{x}''_i) \tag{13}$$

where $j = 1, 2, 3$ represent the three principal directions. The corners of the bounding box are computed in this transformed space based on these minimum and maximum values.

Finally, the OBB is transformed back into the original coordinate system by multiplying the bounding box corners by the eigenvector matrix $\mathbf{V}$ and adding back the centroid:

$$\mathbf{x}_i^{\text{OBB}} = \mathbf{V}\mathbf{x}''_i + \mathbf{c} \tag{14}$$

This process produces the vertices of the Oriented Bounding Box (OBB), which is aligned with the principal axes of the point cloud and accurately reflects its orientation in 3D space as shown in Fig. 5. The axis-aligned bounding box is aligned with the LiDAR coordinate system axes and does not consider the inherent orientation of the data. In contrast, the OBB is rotated to align with the principal components of the point cloud, providing a tighter fit that better captures the data's natural alignment.

D. Point Transformation Using Sequential Rotation and Translation

This section describes the coordinate transformations used to align obstacle bounding boxes (detected by Livox) with the map frame (referenced by Velodyne). It uses two sequential transformations:

1. Transformation from Livox to Velodyne frame (T_2): Because the Livox and Velodyne LiDARs are at different locations on the vehicle, the obstacle detections from the Livox need to be transformed into the Velodyne's coordinate frame. This transformation accounts for the relative rotation (θ) and translation (t_{x2}, t_{y2}) between the two sensors.

2. Transformation from Velodyne to Map frame (T_1): The Velodyne's current pose and orientation relative to the map are given by the vehicle's yaw angle (γ) and position (t_{x1}, t_{y1}). This transformation aligns the obstacle bounding box (now in the Velodyne frame) with the global map coordinates.

The corner points of the obstacle bounding box are represented as given in equation []. The Z coordinates are ignored as we finally need our representation in a BEV:

$$\mathbf{p} \equiv [x_i, y_i, 1]^T \tag{15}$$

The individual transformation matrices are defined as:

$$\mathbf{T_2} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & t_{x2} \\ \sin(\theta) & \cos(\theta) & t_{y2} \\ 0 & 0 & 1 \end{bmatrix} \tag{16}$$

$$\mathbf{T_1} = \begin{bmatrix} \cos(\gamma) & -\sin(\gamma) & t_{x1} \\ \sin(\gamma) & \cos(\gamma) & t_{y1} \\ 0 & 0 & 1 \end{bmatrix} \tag{17}$$

They are transformed sequentially, first by T_2 and then by T_1. The final transformed point p' is calculated as:

$$\mathbf{p'} = T_1 \cdot (T_2 \cdot p) \tag{18}$$

$$\mathbf{p'} \equiv [x'_i, y'_i, 1]^T \tag{19}$$

The final transformed points are then visualized as 2D bounding boxes in the BEV using RViz software.

E. Obstacle Presence Detection and Feedback from A-RoI

Once all the A-RoI components and obstacle bounding boxes are transformed into world coordinates, the next step is to determine whether any obstacles lie within the defined regions and provide appropriate feedback. This feedback is communicated via a ROS topic to facilitate real-time decision-making. If an obstacle is detected within any of the RoIs, the distance of its nearest point from the vehicle is computed and included in the ROS message.

To check for intersections between obstacles and the A-RoI, we utilize the `.intersects()` function from the `Shapely` library. This function determines whether two geometric shapes, such as polygons, lines, or points, share any common space. Internally, it relies on the Dimensionally Extended Nine-Intersection Model (DE-9IM), a spatial relationship framework used in computational geometry.

If any of the $\mathbf{p}'$ lies in the A-RoI, the distances $\mathbf{d_i'}s$ between all $\mathbf{p_i'}s$ in A-RoI and the ego vehicle are calculated and the minimum euclidean distance $min(\mathbf{d_i'})$ is returned.

F. Projecting and Visualizing A-RoI and obstacles in a Bird's Eye View (BEV)

The Bird's Eye View (BEV) is a key tool in automotive Advanced Driver Assistance Systems (ADAS) that offers a comprehensive 360-degree top-down visualization of the vehicle's surrounding environment. This method allows us to efficiently assess the spatial relationship between the vehicle, obstacles, and regions of interest in real time. Utilizing tools like RViz, which is well-integrated with ROS, we can visualize the BEV to gain a clear, intuitive representation of the surroundings.

In this methodology, we leverage the precise coordinates of the bounding boxes representing obstacles, the positions of the A-RoI, and the ego vehicle's location, all of which are referenced to a common world frame. By projecting these elements onto a 2D plane, we create a unified view of the environment as shown in the Fig. 4. This visualization in RViz offers an intuitive way to display the surroundings, highlighting the A-RoI zones, obstacles, and the ego vehicle's position relative to them.

The resulting BEV representation facilitates subsequent steps in the decision-making process, such as motion, behaviour and trajectory planning

Algorithm 1: PCA and Sequential Transformation for Obstacle Alignment and Region of Interest Check

```
Input: Set of 3D corner points
       $C = \{(x_1, y_1, z_1), \ldots, (x_4, y_4, z_4)\}$ from
       obstacle cloud.
Output: Feedback from A-RoI of the obstacles in the
        environment and their distances.
1  Compute centroid $\mathbf{c} = (\bar{x}, \bar{y}, \bar{z})$ and center the points
     $C_i' = C_i - \mathbf{c}$;
2  Construct covariance matrix $\Sigma$ from $C_i'$ ;
3  Perform eigen decomposition to compute eigenvalues
     $\lambda_1, \lambda_2, \lambda_3$ and eigenvectors $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$;
4  Align the points to the principal axes using $\mathbf{v}_1$ and
     $\mathbf{v}_2$;
5  Apply transformation matrix $\mathbf{T_2}$ for Livox to
     Velodyne LiDAR transformation;
6  Apply transformation matrix $\mathbf{T_1}$ for the current pose
     and orientation of Velodyne LiDAR;
7  foreach 2D point $p = (x_i, y_i)$ do
8      Apply sequential transformation $\mathbf{p'} = \mathbf{T_1} \cdot \mathbf{T_2} \cdot \mathbf{p}$;
9  end
10 Check if any of the transformed points $\mathbf{p}'$ lie within
     the A-RoI and return feedback;
11 Return $\mathbf{min}(\mathbf{d_i'})$ for all $\mathbf{p_i'}$ present in A-RoI if any.
```

IV. Implementation and Results

The proposed algorithm was practically implemented and tested on an Autonomous Campus Shuttle (shown in Fig. 6) at the TiHAN (Technology Innovation Hub on Autonomous Navigation) testbed facility. The vehicle is equipped with drive-by-wire capabilities, enabling the deployment of autonomous driving algorithms and other related systems.

A LiDAR HD map was recorded at the testbed facility, serving as the basis for testing the proposed A-RoI algorithm. Fig. 7 illustrates various test scenarios where obstacles are positioned at different points relative to the ego vehicle's position and predicted path. The A-RoI algorithm successfully provided feedback in each of these scenarios, as observed in the terminal outputs presented in the same figure.

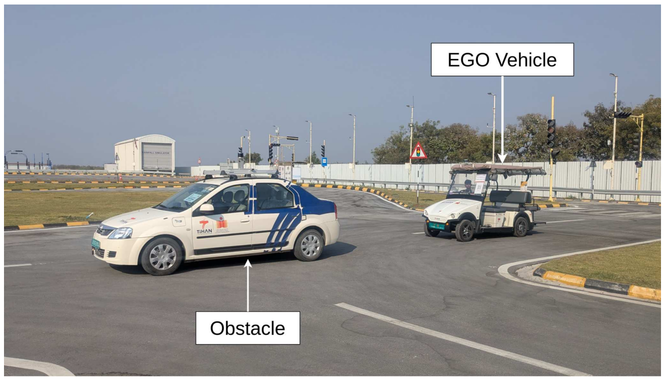


Fig. 6: Picture of the Ego vehicle and obstacle vehicle taken during the experimentation in TiHAN facility.

Scenario Descriptions:

a). A car is positioned on the path of the ego vehicle, intersecting with the Forward Path RoI. In the terminal, the TRUE indicator appears for the main RoI, and the distance to the obstacle is calculated as 8.75 m.

b). A car is parked near the path of the ego vehicle, but it does not lie within the Forward Path RoI. Instead, it intersects with the Lateral Right RoI. This is confirmed by the TRUE message, and the distance is computed as 4.16 m.

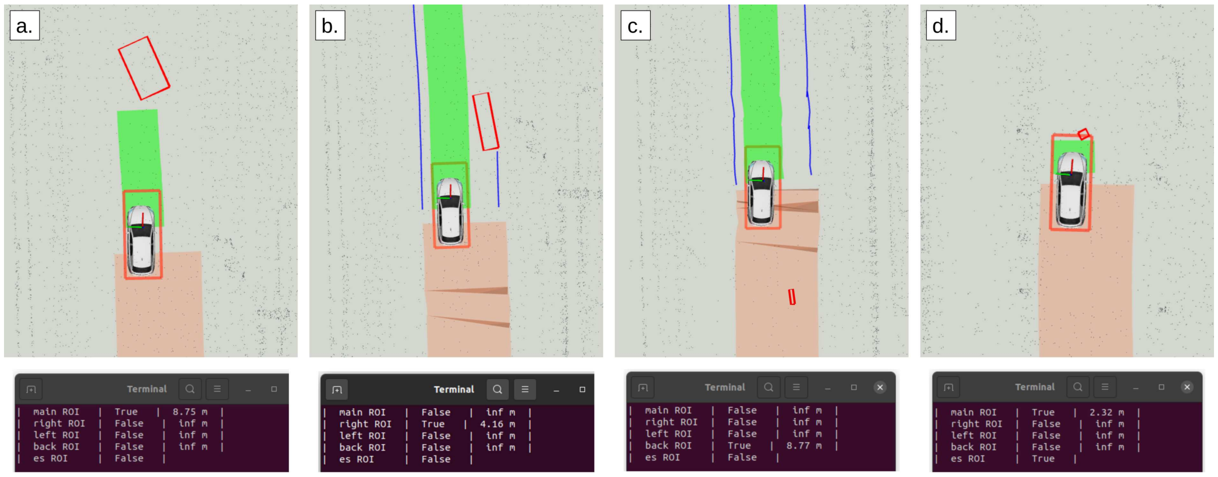
c). A person is approaching the ego vehicle from the rear. The Rear Monitoring RoI detects the object within its bounds, as indicated by the terminal message, with the distance calculated at 8.77 m.

d). A person directly intersects the path of the ego vehicle, entering the bounds of the Safety RoI. In this case, the ROS message is published immediately, and the vehicle is commanded to apply full brakes to prevent a collision.

These results demonstrate the effectiveness of the A-RoI algorithm in detecting obstacles at various positions relative to the ego vehicle and ensuring appropriate responses, such as obstacle distance calculation and emergency braking when necessary.

V. Conclusion

This paper introduced A-RoI, a novel Adaptive Region of Interest technique for enhancing motion and behaviour planning in autonomous vehicles. A-RoI dynamically updates the region of interest on a LiDAR HD map based on the vehicle's position and trajectory, overcoming the limitations



```
|  main ROI   |  True   |  8.75 m  |
|  right ROI  |  False  |   inf m  |
|  left ROI   |  False  |   inf m  |
|  back ROI   |  False  |   inf m  |
|  es ROI     |  False  |
```

```
|  main ROI   |  False  |   inf m  |
|  right ROI  |  True   |  4.16 m  |
|  left ROI   |  False  |   inf m  |
|  back ROI   |  False  |   inf m  |
|  es ROI     |  False  |
```

```
|  main ROI   |  False  |   inf m  |
|  right ROI  |  False  |   inf m  |
|  left ROI   |  False  |   inf m  |
|  back ROI   |  True   |  8.77 m  |
|  es ROI     |  False  |
```

```
|  main ROI   |  True   |  2.32 m  |
|  right ROI  |  False  |   inf m  |
|  left ROI   |  False  |   inf m  |
|  back ROI   |  False  |   inf m  |
|  es ROI     |  True   |
```

Fig. 7: Snapshots of RViz visualization of the vehicle's environment during various test scenarios.

of static RoIs. Five key A-RoI components were presented, each targeting critical areas around the vehicle. Integrating obstacle positions within a Bird's Eye View enables effective perception and decision-making. Real-world tests on an autonomous shuttle demonstrated A-RoI's ability to accurately detect obstacles and trigger appropriate responses. Future work will explore multi-modal sensor fusion and learning-based methods to further improve A-RoI's adaptability and performance.

Acknowledgment

This work is supported by the DST National Mission Interdisciplinary Cyber-Physical Systems (NM-ICPS), Technology Innovation Hub on Autonomous Navigation and Data Acquisition Systems: TiHAN Foundation at Indian Institute of Technology (IIT) Hyderabad.

References

[1] Lang, Alex H., Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. "Pointpillars: Fast encoders for object detection from point clouds." In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 12697-12705. 2019.

[2] J. Zhang, S. Singh, "LOAM: Lidar odometry and mapping in real-time," InRobotics: Science and Systems Vol. 2, No. 9, pp. 1-9, 2014.

[3] H. Wang, C. Wang, C.L. Chen and L. Xie, "F-LOAM: Fast LiDAR Odometry and Mapping," 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Prague, Czech Republic, 2021, pp. 4390-4396, doi: 10.1109/IROS51168.2021.9636655.

[4] A. Thakur, B. Anand, H. Verma and P. Rajalakshmi, "Real Time Lidar Odometry and Mapping and Creation of Vector Map," 2022 8th International Conference on Automation, Robotics and Applications (ICARA), Prague, Czech Republic, 2022, pp. 181-185, doi: 10.1109/ICARA55094.2022.9738576.

[5] A. Thakur and P. Rajalakshmi, "L3D-OTVE: LiDAR-Based 3-D Object Tracking and Velocity Estimation Using LiDAR Odometry," IEEE Sensors Letters, vol. 8, no. 7, pp. 1-4, July 2024, Art no. 6008004, doi: 10.1109/LSENS.2024.3416411.

[6] Liang, Ming, Bin Yang, Shenlong Wang, and Raquel Urtasun. "Deep continuous fusion for multi-sensor 3d object detection." In Proceedings of the European conference on computer vision (ECCV), pp. 641-656. 2018.

[7] Liang, Justin, Namdar Homayounfar, Wei-Chiu Ma, Shenlong Wang, and Raquel Urtasun. "Convolutional recurrent network for road boundary extraction." In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9512-9521. 2019.

[8] A. Thakur and P. Rajalakshmi, "LiDAR-Based Optimized Normal Distribution Transform Localization on 3-D Map for Autonomous Navigation," IEEE Open Journal of Instrumentation and Measurement, vol. 3, pp. 1-11, 2024, Art no. 8500211, doi: 10.1109/OJIM.2024.3412219.

[9] E. Takeuchi and T. Tsubouchi, "A 3-D Scan Matching using Improved 3-D Normal Distributions Transform for Mobile Robotic Mapping," 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing, China, 2006, pp. 3068-3073, doi: 10.1109/IROS.2006.282246.

[10] B. Anand, M. Senapati, V. Barsaiyan and P. Rajalakshmi, "LiDAR-INS/GNSS-Based Real-Time Ground Removal, Segmentation, and Georeferencing Framework for Smart Transportation," in IEEE Transactions on Instrumentation and Measurement, vol. 70, pp. 1-11, 2021, Art no. 8504611, doi: 10.1109/TIM.2021.3117661.

[11] Wei, Junqing, Jarrod M. Snider, Tianyu Gu, John M. Dolan, and Bakhtiar Litkouhi. "A behavioral planning framework for autonomous driving." In 2014 IEEE Intelligent Vehicles Symposium Proceedings, pp. 458-464. IEEE, 2014.

[12] L. Sun, W. Zhan, C. -Y. Chan and M. Tomizuka, "Behavior Planning of Autonomous Cars with Social Perception," 2019 IEEE Intelligent Vehicles Symposium (IV), Paris, France, 2019, pp. 207-213, doi: 10.1109/IVS.2019.8814223.

[13] Du, Jiayuan, Shuai Su, Rui Fan, and Qijun Chen. "Bird's Eye View Perception for Autonomous Driving." Autonomous Driving Perception: Fundamentals and Applications (2023): 323-356.

[14] A. Thakur, C. A. Rakshith Ram and R. Pachamuthu, "LiDAR Sensing-Based Exponential Adaptive Cruise Control and Steering Assist for ADAS," in IEEE Sensors Journal, vol. 25, no. 2, pp. 3597-3607, 15 Jan.15, 2025, doi: 10.1109/JSEN.2024.3512418.

[15] Kumar, Nitish, S. Abhilash, Abhishek Thakur, Omkarthikeya Gopi, Ayush Dasgupta, Arpitha Algole, Bhaskar Anand et al. "TIAND: A Multimodal Dataset for Autonomy on Indian Roads." In 2024 IEEE Intelligent Vehicles Symposium (IV), pp. 688-694. IEEE, 2024.

[16] TiHAN: TiHAN IITH Autonomous Navigation Testbed [Online]. Available: https://tihan.iith.ac.in/