2024 European Control Conference (ECC)
June 25-28, 2024. Stockholm, Sweden

V2X based Vehicle Environment Perception and Occupancy Analysis for Dynamic Pedestrian Behaviour

Swaraj Tendulkar[1], Mayank Yogesh Khandelwal[1], Frank Schrödel[1]

***Abstract*—This research paper addresses the topics of the environment perception domain to realise the solution for connected autonomous mobility by using simulation softwares and real life sensors in tandem. The camera and lidar sensor simulation is performed for a specific scenario arising in the environment and the simulation results, sensor readings and the binary occupancy grids received from sensor simulations are analysed. The RealSense Depth Camera is used to perform visual mapping of the static environment for static obstacle perception and map generation whereas the SICK 2D lidar is used to create a dynamic probabilistic occupancy grid for perceiving the dynamic obstacles in the environment. The results from the sensors in simulation and real life are compared, analysed and validated. The dynamic probabilistic occupancy is used as a foundation to further develop an occupancy prediction model which predicts the future occupancy of any dynamic obstacle based on its velocity and direction of motion. Furthermore, a framework is conceptualised for the Vehicle to Everything (V2X) communication system which includes the identification and determination of the essential communication infrastructure, type of data, recipients, rate of data transfer and ROS communication nodes.**

I. Introduction

Autonomous Driving has always been a subject of evolution since the landmark autonomous vehicle was introduced in the 1980s, by Carnegie Mellon University's Navlab [1] and ALV [2] projects funded by the Defense Advanced Research Projects Agency (DARPA) of the United States of America originating in 1984 and Bundeswehr University Munich's EUREKA Prometheus Project and Mercedes-Benz in 1987 [3].

Due to the risk of collisions in terms of driving, integration of autonomous driving solutions with communication between different entities in the environment forms an important parameter for sustainable connected mobility through V2X. V2X can be described as the mode of technology that facilitates the communication between a vehicle and different entities like vehicles, pedestrians and infrastructure in its environment [9]. One of the major standards for V2X information exchange is Dedicated Short Range Communication (DSRC) developed in the USA [6]. Vehicle to Vehicle (V2V) allows vehicles in proximity to form a mesh network and exchange data which helps to make better decisions using data exchange among the existing nodes [7]. This industrial standard of communication further helps to develop the data transfer framework for the vehicle Onboard Units (OBU) in this paper. The DSRC roadside stationary units and its integration with Wi-Fi, Worldwide Interoperability for Microwave Access (WiMAX), and Long Term Evolution (LTE) network creates a heterogeneous wireless network for Connected Vehicle Technology applications [10]. These existing protocols lay the foundation to the development of the Roadside Stationary Unit (RSU) data transfer framework and data processing in this paper. As the communication protocol is effectively defined, there are few state of the art resources published in [14] regarding the type of data which is supposed to be transferred in a connected environment. This research paper presents a novel approach for the type of data to be transferred between the vehicles and their connected environment infrastructure. Initially, the individual autonomous vehicle's environment data processing is taken into consideration. The next section gives an idea of the dynamic occupancy grid mapping technique which is the base towards identifying the pedestrian occupancies in real time and creating an occupancy prediction model for detecting the pedestrian's future occupancy.

II. Fundamentals

In the research as per [4], a Bayesian filtering technique is used for environment representation and a machine learning approach is adopted for long term prediction of dynamic obstacles. The dynamic occupancy grid mapping solution is based on Sequential Monte Carlo Filtering technique. This grid is given as a generic data input to the particle filter which estimates the spatial occupancy and velocity distribution. For each particular cell, a quasi time continuous probability $P_O(k)$ is extracted. According to [4], the desired prediction result

$$P_O(k) = (P_O(1), P_O(2), P_O(3), ...)$$

where k defines the time step for each sequence element. Furthermore, this sequence is segmented in static and dynamic parts.

*We gratefully acknowledge the financial support for this research by the Thüringer Ministerium für Wirtschaft,Wissenschaft und Digitale Gesellschaft for the project Hochschulübergreifende Forschergruppe Vernetztes und Kognitives Fahren.

Swaraj Tendulkar[1], Mayank Yogesh Khandelwal[1] and Frank Schrödel[1] are with Faculty of Mechanical Engineering, Schmalkalden University of Applied Sciences, 98574 Schmalkalden, Germany.

The probability equation is updated and can be described by

$$P_O(k) = P_{O,s} + (P_{O,d}(1), P_{O,d}(2), P_{O,d}(3), ...)$$

$$= P_{O,s} + P_{O,d}(k)$$

- $P_O(k)$ = Total Probability of Occupancy;
- $P_{O,s}$ = Probability of Static Occupancy (Constant);
- $P_{O,d}(k)$ = Probability of Dynamic Occupancy (Variable);

The equation above has two parts. $P_{O,s}$ which is the constant occupancy probability of static environment whereas $P_{O,d}(k)$ is a sequence which describes the occupancy probability of dynamic objects over several time steps. In the absence of dynamic objects, the probability of occupancy $P_O(k)$ is always equal to the static environment occupancy $P_{O,s}$. When a dynamic obstacle moves in the environment, the corresponding dynamic cell occupancy probability value at the particular time instant $P_{O,d}(k)$ increases. This in turn increases the total occupancy probability value $P_O(k)$. When object leaves the grid cell, the occupancy probability value decreases and becomes equal to the probability of static occupancy [4]. The following section describes the development and implementation of the Dynamic Occupancy Grid using lidar data.

III. Implementation

A. Dynamic Occupancy Grid Implementation

The entire implementation process is carried out in the ROS Noetic environment on Linux 20.04. The 2D lidar is used for dynamic occupancy grid generation to make sure that the dynamic object occupancy gets detected adequately to form a probabilistic occupancy distribution. This acts as a local map which updates over time to create real-time occupancy grids for dynamic path planning. For implementing this part, the Gmapping algorithm is used due to its familiarity, ease of operation and understanding [11]. The default Gmapping node allows the user to represent the occupancies in a binary format. If the occupancy probability threshold is over 25%, then the pixel is considered occupied. The occupied grid cells publish a value of 100 in the map data whereas the unoccupied grid cells return a value of -1. To increase the scope of the occupancy distribution, it is important to convert this binary occupancy grid into a probabilistic occupancy grid [11]. In a probabilistic occupancy grid, the occupancy values from 0 to 100 % are divided into range of probability values.

Table 1: Probabilistic Occupancy Data

Probability of Occupancy	Colour in B/W	Colour in Costmap	Exported Value to map data
0.0	White	Black	-1
0.1 - 0.2	Light Gray	Blue	20
0.2 - 0.4	Gray	Purple	40
0.4 - 0.75	Dark Gray	Red	70
0.75 - 0.1	Black	Pink	100

The table 1 shows the percentage occupancy and the corresponding probabilities, colours in Black & White and cost map channels along with the exported map data values. This improves the performance of the occupancy grid and improves the understanding by providing a much comprehensive set of data as well as visualisation.

B. Occupancy Prediction Model Implementation

Consider a scenario in which a car approaches a cross walk and a pedestrian is about to cross or two cars are approaching a junction. If the vehicle relies on the sensors alone to sense the pedestrian or the car, it wont be able to detect the obstacle until the car or the pedestrian is in the sensor range. In such a case, the ego vehicle would require to make an abrupt maneuver or braking action. To avoid such a scenario, the velocity and direction of the pedestrian or the car can be recorded and its movement can be predicted to give an estimate of the probability of future occupancy in the next time instants. Through this prediction model, the approaching ego vehicle can estimate the future probability of occupancy of the approaching pedestrian or vehicle and apply the necessary control action smoothly. The map grid in RViz is split according to the set resolution of 0.05 ppi. According to the velocity of the object, the number of grid cells it will cover in the next instance can be given by

$$P = \frac{2V}{0.1}$$

where P is the number of cells the object will cover in 1 second. The velocity V is assumed constant and the direction of travel is in one direction. This occupancy prediction model's considerations are basic and further improvements can be made by following a more advance approach as per [15]. Depending on the distance of the object from the lidar sensor, the initial occupancy information can be received from the data projected in the lidar visualization and subsequently the initial occupancies can be shown on the graph. For reference, the pedestrian crossing the side walk is considered. The pedestrian is assumed to have a constant velocity of 0.1 m/s in the right direction as viewed from the ego vehicle waiting for the pedestrian to cross the zebra crossing. According to the defined occupancies ranges in the previous section, the similar colour scheme is used in the graphical representation to represent the particular occupancy probabilities. Initially at $t = 0$, the pedestrian occupies the the given spaces and a range of occupancy probability regions are provided around it indicating its potential movement in the next instance of time. The next section provides an overview on the type of data to be transferred and the V2X framework to realize the same.

C. V2X Framework Implementation

The vehicle Onboard Unit's ROS communication framework is divided into three sections to process the inputs and derive the necessary outputs for the Roadside Stationary Unit as illustrated in Figure 1. The first section receives GPS coordinates in the `nav_odom` topic from the GPS sensor of the vehicle.

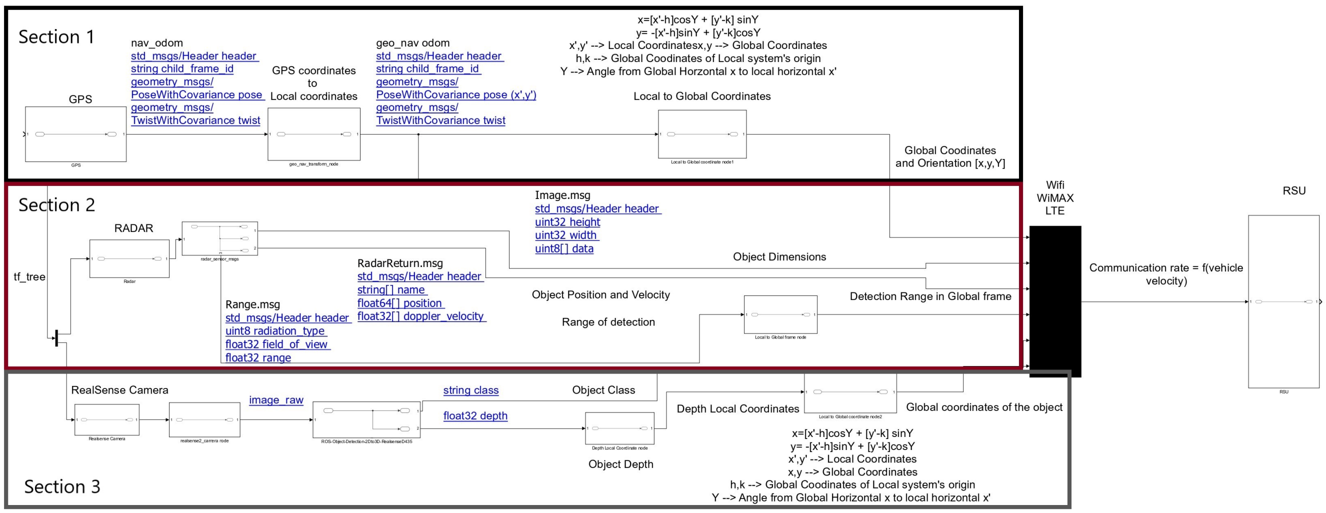


Fig. 1. Onboard Unit Data Transfer Framework

This topic is subscribed to by the `geo_nav_transform_node` to convert it into local coordinates. This node publishes the coordinates in the `geo_nav_odom` message. This message is further subscribed by the custom node created to convert the local coordinates to the global coordinates. These coordinates are then given to the RSU. The second section involves data from the radar sensor. The radar sensor data is accessed by the `radar_sensor_msgs` node and publishes several message topics in which `Image.msg`, `RadarReturn.msg` and `Range.msg` are the three messages in consideration. The `Image.msg` provides the object dimensions in terms of height and width as a float32 variable. The `RadarReturn.msg` provides the velocity of the object in the `doppler_velocity` topic and position of the object as a float64 variable type. The `Range.msg` provides the range of the sensor as a float32 variable. This range and position data is received as a reference to the local frame of the ego vehicle as the sensor is added to the tf tree of the vehicle. The third section provides RealSense camera parameter transfer. Here, the camera provides the camera feed to the `realsense2_camera` node which publishes the `image_raw` topic. This `image_raw` topic is used as an input to the YOLO V5 algorithm which detects the object and its depth data. It outputs the class of the object and its corresponding depth in meters. This depth is aligned with respect to the ego vehicle local coordinates as the camera is also added in the tf tree. These local coordinates of the objects are converted into global coordinates and are transmitted with their class in the form of an object list to the RSU through Wifi, WiMAX or LTE communication [10]. The communication rate is a function of the vehicle velocity as formulated in [12].

Based on the data received, the traffic optimizer operates the traffic signals to ensure sustainable traffic flow in the environment. The traffic optimizer is also responsible for providing the necessary input to the Traffic Navigation Recommender. This unit sends further recommendations to the vehicle OBUs for efficiently navigating through the traffic [12]. Subsequently, the RSU sends a filtered object list to specific vehicle OBU depending on the objects in close proximity to that vehicle. This logic is implemented by first converting the radar detection range which is in the global frame from cartesian global coordinates to parametric coordinates in the form of radius r and angle θ. Then the vehicle global coordinates and the object global coordinates are also converted to the parametric form. The object list and the corresponding object parametric coordinates are added together in the same array whereas the parametric range coordinates and vehicle coordinates are combined in a single array. The parametric coordinates of the object and the parametric range coordinates are compared to check which objects are within range of the sensor. These object classes are filtered and the condensed object list and the corresponding object parametric coordinates are published. These parametric coordinates are converted to global cartesian coordinates and then converted to the local coordinates and transmitted to the specific OBU. The RSU receives this data from all OBUs in the environment and publishes a unique filtered object list for the vehicle in the specified range via DSRC in tandem with Wifi, WiMAX and LTE technology [10]. This communication framework developed in this section as well as the dynamic occupancy grid and prediction model developed in the previous sections need sensor perception data from individual vehicles. This perception data development is explained in the next section.

IV. Development and Results

A. Development Approach

In the case of this research paper, an environment perception solution is developed for a pedestrian crossing in front of the vehicle. The sensor simulation data collection and analysis is a part of the Software in Loop phase. The sensor readings and occupancy data received from the simulations in

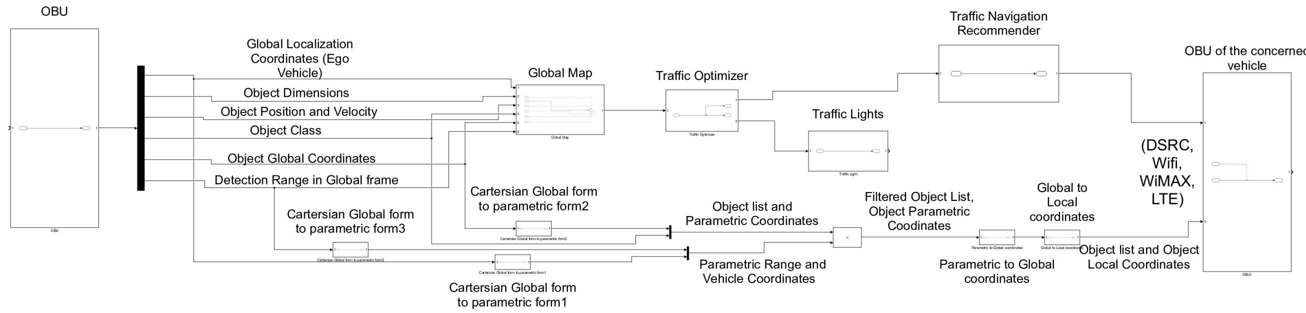


Fig. 2. Roadside Stationary Unit Data Transfer Framework

Carla create a base to further evaluate the data received from physical sensors. The Hardware in Loop phase introduces the RealSense Depth Camera and SICK 2D lidar. After the development of the algorithm for occupancy data, it is tested by mounting the sensor setup on the Pioneer 3DX mobile robot. The sensor data and algorithm functionality for the particular scenario perception is recorded. The software and hardware solutions are then validated and verified by Software in Loop and Hardware in Loop methods.

B. Software Execution

A road network contains the necessary geometric mesh data [.fbx file] as well as the OpenDRIVE data [.xodr file] which gives information on the direction of travel allowed in particular lanes and junctions. The .xodr file contains all the motion and maneuverability information as well as road section details which is exported to Carla [13]. Carla executes a powerful Python API which facilitates the users to control various aspects in relation with simulation including traffic generation, weather, sensors and pedestrian behaviors. The sensors used in simulation include a DVS camera and 3D lidar. Dynamic Vision Cameras calculate the changes of intensity in the form of a stream of events, which encode per-pixel brightness changes instead of capturing images at fixed instant. They can provide better quality of visual information in high-speed dynamic scenarios acting in high-dynamic range environments due to higher dynamic range, no motion blur and high temporal resolution. The 3D lidar sensor generates a 3D point cloud of the objects in the surrounding which is essential for environment perception.

C. Simulation Results

The simulation results for the scenario consisting of the pedestrian crossing in front of the ego vehicle are elaborated in this section. As defined in the implementation, this scenario is created in Carla and simulated to receive the mentioned sensor simulation data. The constructed simulation and the sensor results are illustrated in Figure 3. For the scenario, we spawn the vehicle at a spawn point near the zebra crossing. The top left window represents the 3D lidar feed whereas the top right window represents the DVS camera feed. The pedestrian behaviour is of two types namely the group of pedestrians which are static and are standing on the zebra crossing and the group of pedestrians walking across on the crosswalk both depicted together in Figure 3. This approach verifies whether the sensors are capable of perceiving the static and dynamic behaviour of the pedestrians and representing it in the form of a dynamic binary occupancy grid at a particular time instant. One key point of address is that standing pedestrians on the zebra crossing are not visible as effectively in the DVS camera feed. On the contrary, the 3D lidar point cloud data captures standing as well as walking pedestrians as seen in the lidar feed and correspondingly in the binary occupancy grid represented in the top centre window.

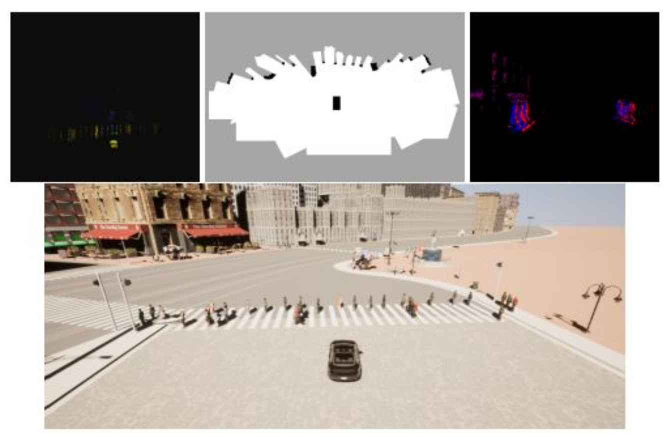


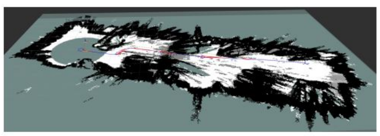
Fig. 3. Sensor Simulation Result

D. Hardware Execution

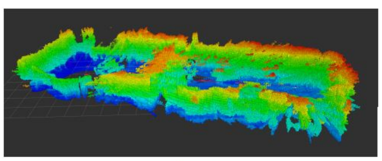
The sensor setup includes the RealSense depth camera and SICK 2D lidar mounted on the robot frame. All the sensors are connected to the vehicle OBU where the data is analysed and algorithms are implemented on ROS Noetic. The robot is two wheeled with a differential drive steering system. Further details are provided in [5]. The RealSense depth camera is used for static environment mapping due to the detailed RGB and depth data of the environment. For the visual mapping of the test environment, the Real Time Appearance Based Mapping (RTAB) algorithm is selected due to the utilisation of the visual characteristics of the camera. To implement visual mapping using the RTAB algorithm, the steps are followed as per [8].

E. Experimental Results

After the installation of all the necessary nodes and dependencies, the camera is setup for performing visual mapping. The environment in consideration is a closed environment used for testing this mapping algorithm. This environment is chosen as the task here is to statically map the environment and do not allow any dynamic obstacles to enter the camera frame. The RTAB algorithm allows the environment representation in different ways [8]. Out of all these different interpretations, three of them represent the environment comprehensively if considered together and give some key features if considered individually. The 2D binary occupancy grid provides a 2D black and white occupancy representation of the target environment as shown in Figure 4. Even though the 2D occupancy grid creation takes the 3D information of the object and maps it on a 2D surface, it fails to show the 3D orientation and structure of the object in the map. The 3D probabilistic colour occupancy grid in Figure 4 allows the representation of the environment and its 3D information in the form of a colour occupancy zone with each zone representing a range of occupancy probability.



2D Binary Occupancy Grid



3D Probabilistic Occupancy Grid

3D Point Cloud

Fig. 4. Static Environment Representation

The only drawback of this representation is that even though the occupancies are effectively represented, the appearance of the object still remains unknown in the map representation. To tackle this shortcoming, a 3D point cloud map is considered. From the point cloud map, the appearance of the object can be figured out. The drawback of this type of map representation is that it does not provide any occupancy information. The environment representations are illustrated in Figure 4. After launching the lidar node, the Gmapping node is launched. When the Gmapping node is launched, it provides a link between the map and the odom link. A static transform link is implemented between the laser mount link and the map link to pass on the laser scan input. The lidar is now set to detect the dynamic movement of the pedestrian crossing. After detecting the pedestrian, the algorithm creates a range of occupancy probabilities in the grid cells around this object. As seen in the Figure 5, the pedestrian is represented accurately in terms of a probabilistic occupancy grid at its starting position and is featured by a rectangular box in both the figures.

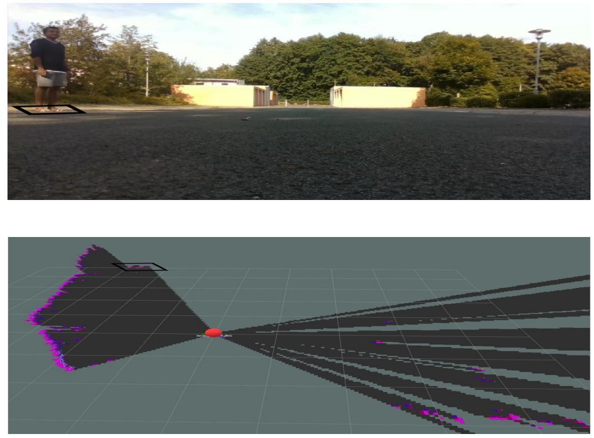


Fig. 5. Initial Scene with 2D Dynamic Occupancy

The distribution shows a pink grid cell area at the initial position of the pedestrian which represents the algorithm's confidence in its occupancy. As seen from the Figure 5, the occupancy data shows limitations in determining the occupancies in front of the robot as the laser scan extends indefinitely in the environment and cannot register object data points. As the pedestrian walks across in front of the robot on its designated path to cross the road, sensor registers a scan at the intermediate positions thereby increasing the occupancy at these locations for that time instant and decreases as the pedestrians moves further. The laser scanner further registers the scan at the next instances until the final position as shown in Figure 6.

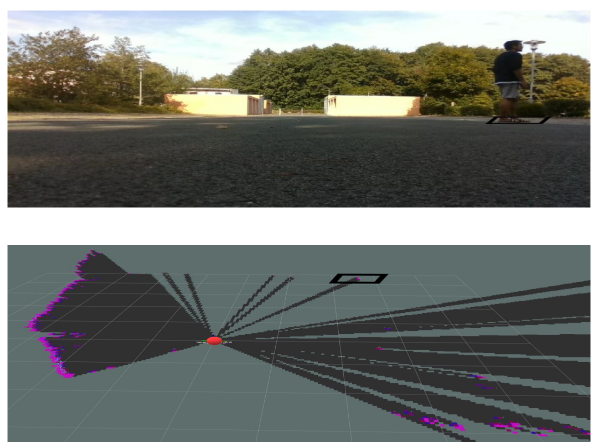


Fig. 6. Final Scene with 2D Dynamic Occupancy

By the time it reaches its final position denoted by the rectangular box, the occupancy at the final location increases and is represented in pink grid cells and subsequently the occupancy decreases at the initial position as well as the intermediate positions.

Since, the laser cannot detect any obstacles past the pedestrian, the laser takes the pedestrian's intermediate positions into account. It creates free occupancies upto the intermediate scan positions even after the pedestrian is no longer present there. Hence, in an environment with less obstacles in the lidar range, it can provide such limited data of the environment which makes it necessary to use it with other sensors like camera.

F. Occupancy Prediction Results

As per the implementation section of the occupancy prediction model, the predicted dynamic probabilistic two dimensional occupancy of the pedestrian received from the SICK 2D lidar and its prediction over the next 30 seconds is represented on a 3D graph in which the X and Y axis describe the 2D occupancy grid over time t as shown in Figure 7-

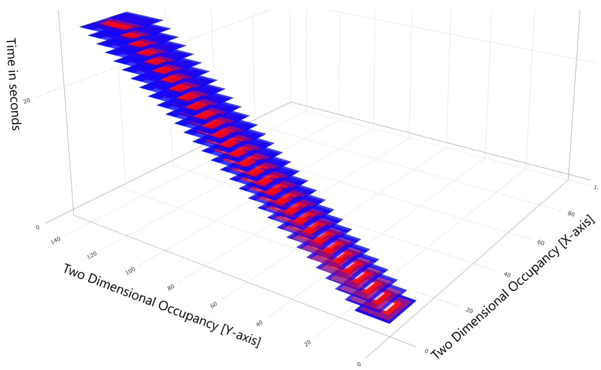


Fig. 7. Occupancy Prediction Model

As the time increases, the probability distribution moves in the given direction with the specified shift in cells at every time instant depending on the equation given in the implementation section. Here, the assumption is made that the pedestrian moves in the positive Y direction with a velocity of 0.1 m/s. Initially, the lidar can predict the probability of occupancy with greater confidence indicating large high probability of occupancy zones. As the confidence of occupancies is more at the first instant. The distribution at $t = 0$ shows smaller blue and purple region indicating low probabilities of occupancy. Since the movement of the pedestrian cannot be fully confirmed to have a constant direction and velocity, the uncertainty in the prediction of occupancies increases over time. This means that the confidence with which the model can predict the occupancy of the pedestrian decreases with time. This results in the graph to show a gradual decrease in the pink region of the highest probability over time and increase in blue region representing a low probability of occupancy. The degree of uncertainty in the pedestrian's movement leads to a greater area of occupancy probability distribution with time as shown in Figure 7. Subsequently, the distribution at $t = 30$ shows a greater blue region over the boundaries with smaller purple and red regions indicating a greater degree of uncertainty of occupancy. After a careful consideration of these factors, the necessary decision making must be performed to reduce the risk of collision thereby creating a robust automated driving solution.

V. Conclusion

The involvement of visual mapping for static environment mapping provided a detailed representation of the environment which can be used to map the target environment. Furthermore, the lidar laser scan data helped to create a dynamic probabilistic occupancy grid which improved the way the ego vehicle would perceive the dynamic objects. As a diversion from the traditional binary occupancy grid, the probabilistic occupancy grid and its representation provided a much improved occupancy data which formed the base to create a prediction model to predict their future occupancy data. The onboard sensor data collected from different vehicles served as a foundation for further conducting tests on the framework established for V2V and Vehicle to Infrastructure (V2I) communication. The concepts described in this paper lay the groundwork for establishing a sustainable connected urban mobility solution that relies not only on onboard sensor data but also on data received from the interconnected environment.

References

[1] "Carnegie Mellon". Navlab: The Carnegie Mellon University Navigation Laboratory. The Robotics Institute. Retrieved 2014-12-20.

[2] T. Kanade (February 1986). "Autonomous land vehicle project at CMU". Proceedings of the 1986 ACM fourteenth annual conference on Computer science - CSC '86. pp. 71–80. ISBN 0897911776. S2CID 2308303.

[3] J. Schmidhuber (2009). "Prof. Schmidhuber's highlights of robot car history". 4th Conference on Artificial General Intelligence, 7 Aug 2011.

[4] S. Hoermann, M. Bach, K. Dietmayer (2017). "Dynamic Occupancy Grid Prediction for Urban Autonomous Driving: A Deep Learning Approach with Fully Automatic Labeling". 2018 IEEE International Conference on Robotics and Automation (ICRA).

[5] S. Tendulkar, V.P. Uppalapati, H. Busireddy, Y. Strigina, T. Parmar and F. Schrödel. "Robots in Town – Low-Cost Automated Logistic Concept". On Journ of Robotics & Autom. 1(5): 2023. OJRAT. MS.ID.000524.

[6] R. F. Atallah, M. J. Khabbaz, and C. M. Assi, "Vehicular Networking: A Survey on Spectrum Access Technologies and Persisting Challenges", Elsevier Vehic. Commun., July 2015, vol. 2, issue. 3, pp. 125–149.

[7] M. Kawser, S. Sajjad, S. Fahad, S. Ahmed, H. Rafi (2019). "The Perspective of Vehicle-to-Everything (V2X) Communication towards 5G". 19. 146-155. IJCSNS International Journal of Computer Science and Network Security, April 2019

[8] SLAM with D435i, Intel RealSense, https://github.com/IntelRealSense /realsense-ros/wiki/SLAM-with-D435i, accessed July 2023.

[9] R. Daddanala, V. Mannava, L. Tawalbeh, M. A. Ramahi. "Vehicle to Vehicle (V2V) Communication Protocol: Components, Benefits, Challenges, Safety and Machine Learning Applications". CoRR, abs/2102.07306, 2021.

[10] K. Dey, A. Rayamajhi, M. Chowdhury, P. Bhavsar, J. Martin, (2016). "Vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication in a heterogeneous wireless network – Performance evaluation". Transportation Research Part C: Emerging Technologies.

[11] Slam_Gmapping, Ros-Perception, https://github.com/ros-perception/slam _gmapping, accessed August 2023

[12] CAR 2 CAR Communication Consortium, "Survey on ITS-G5 CAM statistics". December 2018

[13] A. Kanakagiri. "Development of a Virtual Simulation Environment for Autonomous Driving using Digital Twins". Master Thesis, Technische Hochschule Ingolstadt, September 2021.

[14] CAR 2 CAR Communication Consortium, "White Paper Connected and Cooperative Automated Driving (CCAD)". May 2023

[15] P. Zechel, R. Streiter, K. Bogenberger and U. Göhner. "Pedestrian Occupancy Prediction for Autonomous Vehicles". 2019 Third IEEE International Conference on Robotic Computing (IRC), Naples, Italy, 2019.