Literature Review on Motion Prediction of Vulnerable Road Users using Occupancy Grid Maps

Rutger Dirks

April 27, 2021

3ME DEPARTMENT OF COGNITIVE ROBOTICS

DELFT UNIVERSITY OF TECHNOLOGY

SUPERVISORS: EWOUD POOL AND DARIU GAVRILA

Acronyms

AV Autonomous Vehicle. 4, 5, 8, 11, 17, 23

BBA Basic Belief Assignment. 12, 13

BDD100K Berkely DeepDrive 100K. 19

BEV bird's-eye view. 9, 18

DST Dempster-Shafer Theory. 2, 9, 12, 13, 17

ECP2.5D Eurocity Persons 2.5D. 19

FOD Frame of Discernment. 12

KITTI Karlsruhe Institute of Technology and Toyota Technological Institute. 19

NN Artificial Neural Network. 2, 14, 17

OGM Occupancy Grid Map. 2, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 22, 23, 25, 26

STIP Stanford-TRI Intent Prediction. 18

VRU Vulnerable Road User. 4, 5, 8, 18

Contents

1 Introduction

In 2017, the European Union counted around 25300 fatalities on the EU roads, of which 46% were vulnerable road users. Although the EU roads account for the safest roads in the world, the number of road fatalities have stagnated in the past few years. At this rate, the EU's goal of reaching fewer than 16000 fatalities in 2020 could not be achieved. Especially vulnerable road user fatalities have not decreased at the same pace as the overall population. Any progress to increase the safety of vulnerable road users will have a significant impact to the road fatality rate. [1]

Autonomous Vehicles (AVs) are expected to increase road safety because the autonomy would erase the effects of human error [2], since more than 90% of the accidents is caused by human error [3].
However, AVs are not yet safe enough to deploy on the roads and still a lot of research should be done before fully autonomous vehicles can be introduced to the market [4] [2].

One of the current challenges to autonomous driving is to capture road user intent and to make accurate real-time trajectory predictions of other road users [5]. Especially predicting the behavior of Vulnerable Road Users (VRUs) is important [5] [6] because good anticipation of a VRU's behaviour results in faster reactions and thus can prevent more accidents [7]. Predicting VRU behavior is additionally challenging compared to predicting other (motorized) road users, because "VRU motion follows complex patterns constrained by static and dynamic obstacles along the path." [8]

This literature study gives an overview of what the current research areas and challenges for the future are regarding motion prediction of vulnerable road users. Based on this literature study I will write a research proposal for my Master thesis.

This chapter describes the problem of motion prediction and elaborates on current solutions and their limitations. Then, one main research question and five sub-questions are asked to investigate the stated problem. This chapter ends with a short overview of the content of each chapter.

1.1 Problem Description

The problem of motion prediction is an inferencing task. Based on evidence in the form of current and/or past observations and obtainable experience, future states (trajectories) of one or more actors must be predicted for a pre-determined time horizon, within a certain accuracy and certainty, which needs to be executed within a specific time limit. This problem statement will be explained more elaborately in these following paragraphs.

Actors, in this problem statement, are traffic participants including, but not limited to, cars, riders, cyclists and pedestrians (VRUs).

The evidence that supports the actor's predictions can have the form of current and/or past observations, and obtainable experience. The current and past observations are sensor data acquired from the environment (observations) by either static observers (the sensors are static and record the environment) or dynamic observers (the sensors move within the environment while recording it) or both. Then, obtainable experience are generalizations of traffic, including traffic behavior and traffic environments (experience). These generalizations can be based on statistics, they can be based on assumptions, and they can be learned using historic observations.

Based on the investigated literature, the future states can be represented as one of the three following forms in the predictions.

1. The predictions are estimated future state values of an actor's centroid, including its coordinates and sometimes orientations, which are projected into the environment representation.
2. The predictions are estimations of the future actor's representation within the current environment representation.
3. The predictions are estimations of the entire future environment representation.

Finally, the problem statement mentions a time horizon, that the predictions must be within a certain accuracy and certainty, and that it needs to be executed within a specific time limit. The time horizon is the amount of time into the future that the prediction must span. Ideally, the time horizon, the accuracy, the certainty, and the execution time of the predictions should be infinitely, exact, complete, and instantaneously respectively. However, this is not possible in practice. Therefore, I think a suitable prediction method should perform further into the future, more accurate, with more certainty, and faster than a human can perform the task. Within the context of AVs, this is a reasonable requirement in order for the AV to be able to perform safer than humans can.

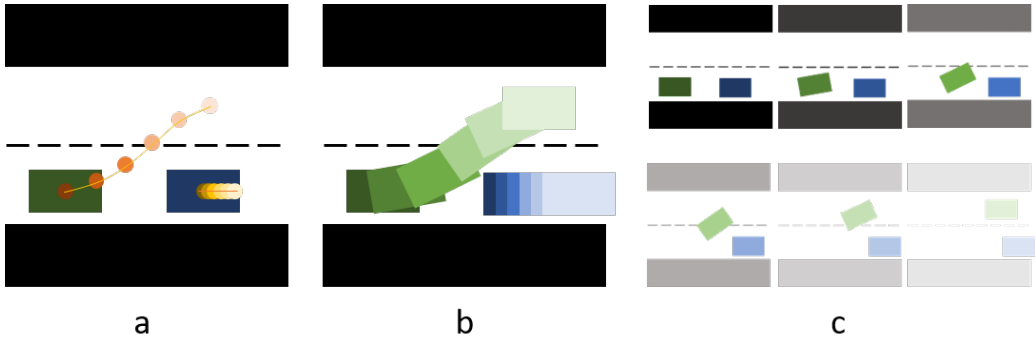


Figure 1: This image shows the three prediction forms encountered in the literature. The black and white parts represent a road. The green and blue squares are representations of vehicles. The predictions are shown using a more fading color the further the prediction goes into the future. Figure a shows the prediction of future centroids projected into the environment representation. Figure b shows the prediction of future actor representations within the environment (the complete vehicles are predicted instead of the centroids only). Figure c shows the prediction of the entire future environment. In this case, also a prediction of the environment (road) is provided.

1.2 Current Solutions

Table 1 shows an overview of recent motion prediction papers and their respective methods. It is notable that most methods use a form of deep learning to predict trajectories in any of the forms as described in the problem description. What is also striking, is that relatively few papers have performed research which includes VRUs in which the observer is dynamic (from a vehicle's point of view). Still, it is important that VRU behavior prediction is investigated. VRU collisions are likely to be fatal which makes accurate and early motion prediction necessary in order to respond safely to their actions [35], [8], [55]. Therefore, VRU behaviour prediction is a relevant research topic regarding AVs.
A major challenge in this research area is to predict VRU motion for the long term. However, research on this topic faces a multitude of challenges. Accurate long-term predictions require the incorporation of environmental and social context. Moreover, implementing uncertainty and multimodality of the predictions and making the predictions fast enough for real-time applications in AVs are also challenges that need to be solved. These major challenges and recent research to find solutions are described in the following subsections.

1.2.1 Long-term motion prediction

For motion prediction, a prediction for which the prediction time horizon is more than 2 seconds is considered long-term [48]. Compared to other road users, VRUs are highly maneuverable which makes it challenging to predict their behavior, especially for the long-term [43], [35], [35]. Most prediction models assume that VRU behavior is intention-driven and that those intentions can be inferred from certain cues, other than dynamics [35]. Therefore, detecting cues related to VRU behavioral changes is important for achieving accurate long-term predictions [20]. These cues can be related to the VRU's environment (e.g. traffic lights, zebra's, obstacles) or to the VRU's social interactions (e.g. evasion of other VRU's, distancing due to social norms, nearing another VRU). The following two paragraphs elaborate on the incorporation of environmental and, respectively, social context in prediction models. Besides using cues to predict VRU's behavior, implementing the uncertainty of the predictions and incorporating multimodality is also researched, as well as making the models fast enough so that they can be implemented real-time on AVs. These topics are also discussed below.

1.2.2 Environmental context

Environmental context consists of the static and dynamic obstacles and semantic meaning within an actor's surroundings. Research suggested to incorporate environmental context because an actor's trajectory is highly influenced by its environment. Scene context can constrain the actor in its planned motions [8], [30], [41], [31]. Especially to anticipate critical situations, taking into account the environmental context is expected to increase the prediction accuracy [55]. The challenge is how to model that environmental context so that it is semantically representative, highly discriminative and generalizable to a variety of complex scenes so that it can be used as evidence to enable inferences about an actor's future trajectory [21].
[20] Shows that leveraging prior knowledge about an actor's environment can improve the trajectory predictions. Therefore, they suggest to extend the incorporation of environmental context even more with the expectation that it will further increase the prediction accuracy.

Source	Year	Goal Type	Observer	Actors	Method	Cue type	Environment representation	Future Work
[9]	2009	TP	SO	Ps	MDP	APS	DOGMa	ISC, MISC
[10]	2013	TP, MM	DO	Ps	GPDM, PHTM, KF, IMM KF	SI	GPCS	ED, MISC
[11]	2013	TP	DO	Ps	EKF, IMM	APS	GPCS	
[12]	2014	TP, ISC	DO	Ps	DBN, SLDS	APS, Other	3DWC	IEC, MISC
[13]	2014	TP, MM	SO	Ps	GPDM, PHTM, KF, IMM KF	Other	3DWC	ED
[14]	2015	TP	DO	Vs	RRT+GMM	APS, CMI	GPCS+Image	
[15]	2015	TP, ISC	SO	Ps	GP	APS	GPCS	
[16]	2016	TP, ISC	SO	Ps	Social-LSTM	APS	GPCS	ED, IEC
[17]	2016	TP, DL	SO	Ps	CNN	SI, ACS, Other	Image	
[18]	2016	TP	SO	RUs	ASNSC	APS	GPCS, Raster-GPCS	MISC
[19]	2017	TP, DL	DO	Vs	LSTM	APS	OGM	
[20]	2017	TP, IEC, MM	DO	Cs	LDS	APS, RT	GPCS	IEC
[21]	2017	TP, IEC, DL	SO	Ps	SSCN, CNN	CL, SI	GPCS+Image	ED
[22]	2017	TP, ISC, IEC	DO	Rus	IOC RNN	APS, CL, Other	GPCS	ED
[23]	2018	TP, ISC, IEC, E2E, DL, UA	DO, SO	RUs	RNN	OGM	DOGMa	ED
[24]	2018	TP, DL	DO	Vs	CNN, FC, LSTM	SI	Raster-Image	
[25]	2018	TP	SO	Ps	MLP, RNN, TCN	APS	GPCS	ISC
[26]	2018	TP, DL	DO	Vs	LSTM	APS, SI, Other	Raster-Image	
[27]	2018	TP, ISC, DL	SO	Vs	LSTM	APS	GPCS	IEC
[28]	2018	TP, IEC	SO	Ps	ASNSC	APS, Other	GPCS, Raster-GPCS	ISC, IEC
[29]	2018	TP, ISC, DL	SO	Ps	RNN, GAN	APS	GPCS	
[30]	2018	TP, ISC, IEC	SO	Ps	CNN, LSTM	APS, OGM	GPCS	MISC
[31]	2018	TP, ISC, IEC, DL	SO	Ps	LSTM	SI	GPCS+Image	ED, ISC
[32]	2018	TP, DL	SO	Ps	CNN	APS	GPCS	ISC, MISC
[33]	2018	TP, MM, ISC	SO	Ps	CNN	APS, TL	GPCS+Image	IEC, MISC
[34]	2018	TP, ISC, IEC, E2E, DL	DO	Vs	CNN-Transformer	Other	GPCS, 3D-OGM	ED, IEC
[35]	2018	TP, MM, E2E, DL	DO	Ps	ANN, LSTM	SI, ACS	GPCS	ED, IEC, MISC
[36]	2018	TP, ISC, IEC, E2E, DL, MM, UA	DO	RUs	CNN	OGM	OGM	MISC
[37]	2019	TP, MM, DL, UA	DO	RUs	CNN	ACS, SI	DOGMa	
[38]	2019	TP, UA, MM	DO	Vs	VGG-FC, GMM	APS, SI, Other	GPCS	
[39]	2019	TP, MM	DO	Vs	RNN, CNN	APS, SI	GPCS	ED
[40]	2019	TP, ISC, DL	DO	RUs	LSTM	APS, SI	GPCS	ED, IEC
[41]	2019	TP, ISC, IEC, DL	SO	Ps	LSTM, GAN	APS, SI	GPCS+Image	
[42]	2019	TP	DO, SO	RUs	LSTM-CNN	APS, Other	3DWC	MISC
[43]	2019	TP, DL	DO	VRUs	RNN, LSTM, GRU	APS, CL, ACS, Other	Images	
[44]	2019	TP, DL	SO	VRUs	B-LSTM, IRL	APS, SI	GPCS+Image	IEC, ISC
[45]	2019	TP, ISC, IEC, E2E, DL, UA	DO	RUs	CNN, LSTM	OGM	OGM	ED, MISC
[46]	2019	TP, ISC, IEC, E2E, DL, MM, UA	DO	RUs	CNN, LSTM	OGM	OGM	MISC
[47]	2019	TP, ISC, IEC, E2E, DL, UA	DO	RUs	CNN, LSTM	OGM	OGM	ED, MISC
[48]	2020	TP, ISC, IEC, E2E, DL, MM, UA	DO	RUs	CNN	OGM	DOGMa	ISC
[7]	2020	TP, UA	DO	Vs	CNN, FC, LSTM	SI	Raster-Image	
[49]	2020	TP, DL	SO	Ps	GRU, LSTM	APS	GPCS	
[50]	2020	TP, ISC, IEC, E2E, DL,UA	DO	Vs	RNN-Transformer	RT, OGM, SI, ACS	DOGMa	ED, ISC
[8]	2020	TP, DL	DO	VRUs	CNN	SI, APS, RT, TL	Raster-Image	
[51]	2020	TP, MM, ISC	SO	Ps	Social-VRNN, LSTM, CNN	APS, OGM	GPCS+Image, DOGMa	MISC
[52]	2020	TP, ISC, IEC, E2E, DL, UA	DO	RUs	CNN, LSTM	OGM	OGM	MISC
[53]	2020	TP, ISC, IEC, E2E, DL, UA	DO	RUs	CNN, LSTM	OGM	OGM	MISC
[54]	2020	TP, ISC, IEC, E2E, DL	DO	RUs	CNN	OGM	DOGMa	

Table 1: This table shows an overview of recent literature that does research on a form of motion prediction regarding trajectories of various road users. Abbreviations are used to make the table clear to understand. The abbreviations are listed below per column category.

Goal Type: Multi-Modal (MM), Trajectory Prediction (TP), Real-time (RT), End-to-end (E2E), Using Deep Learning Methods (DL), Incorporate Social Cues (ISC), Incorporate Environmental Cues (IEC), Uncertainty-Aware (UA).
Observers: Dynamic Observer (DO), Static Observer (SO).
Actors: Road Users (RUs), Vulnerable Road Users (VRUs), Pedestrians (Ps), Cyclists (Cs), Vehicles (Vs).
Methods: Markov Decision Process (MDP), Gaussian Process Dynamical Models (GPDM), Probabilistic Hierarchical Trajectory Matching (PHTM), Kalman Filter (KF), Interacting Multiple Model Kalman Filter (IMM KF), Extended Kalman Filter (EKF), Rapid Random Tree Search (RRT+), Gaussian Mixture Model (GMM), Gaussian Process (GP), Gaussian Process Dynamical Model (GPDM), Long Short-Term Memory (LSTM), Convolutional Neural Network (CNN), Augmented Semi Nonnegative Sparse Coding (ASNSC), Linear Dynamical System (LDS), Spatially Static Context Network (SSCN), Inverse Optimal Control (IOC), Recurrent Neural Network (RNN), Fully Connected Layer(s) (FC), Multi-Layer Perceptron (MLP), Temporal Convolutional Network (TCN), Generative Adversarial Network (GAN), Articifial Neural Network (ANN), Visual Geometry Group Neural Network (VGG), Inverse Reinforcement Learning (IRL).
Cue Types: Actor Current State (ACS), Occupancy Grid Map (OGM), Scene Image (SI), Road topology (RT), Actor Past States (APS), Cost Map Image (CMI), Class labels (CL), Traffic Lights (TL), Other.
Environment Representation: Occupancy Grid Map (OGM), Dynamic OGM (DOGMa), 3D OGM (3D-OGM), Camera (RGB)-Image (Image), Rasterized scene (Raster) Ground Plane Coordinate System (GPCS), 3D World Coordinates (3DWC).
Future work: Incorporate Environmental Cues (IEC), Incorporate Social Cues (ISC), Expand Data (ED), Miscellaneous (MISC).

1.2.3 Social Context

Social context consists of the static and dynamic actors, their semantic meaning, and their interactions within an actor's surroundings. When an actor navigates, it constantly tries to anticipate movements of surrounding actors to envision more than one feasible path it could take in order to adjust its own path and maneuver past or aside those surrounding actors [41], [31]. Therefore, one of the main factors that influence an actor's behavior is the its social context. It is expected that by incorporating social context into prediction models, the accuracy will increase significantly [30].
Especially for pedestrian behavior, the social and environmental context is expected to be the leading evidence for predicting their behavior because pedestrians "are not expected to use any active forms of communication when interacting with vehicles and other road users" [55].

1.2.4 Uncertainty and Multi-modality

For the safety of AVs, uncertainty estimations for predictions are critical [7]. If an AV is aware of the confidence of its predictions, it can adjust its planned course to minimize the risk of collisions [38]. For example, if an AV predicts that a pedestrian will not cross the road, but the confidence of that prediction is very low, the AV can adjust its course by leaving more room for the pedestrian in case the prediction is false. Without knowing the confidence of a prediction, such risk-avoiding course adjustments cannot be made. Besides including uncertainty estimations, it is important that predictions are multimodal. VRU behaviour is inherently multimodal because they easily change their course which is highly dependent on their environment [37], [39]. Incorporating multimodality is a challenge, because it requires good probability estimations and knowledge of the different modes. It either requires explicit labeling of the modes prior to training [39], or the modes could be learned. Because probability estimations are needed to make multimodal predictions, these methods are often researched together.
[10] compares four different multimodal prediction models related to Bayesian principles. These models return the probability of stopping (whether a nearing pedestrian will stop, instead of cross the road). However, only two motion modes are considered in this research. [20] Uses a Mixture of Linear Dynamical Systems to predict the probabilities of a cyclists going in one of the five pre-determined direction modes. This model also takes into account information about the road topology to enhance the predictions. The downside of these Bayesian models is that the computation time increases when more modes are added and when more context information is incorporated in the predictions. Moreover, adding more modes is a challenges in complex environments in which it is unclear what directions an actor can go to. Therefore, deep learning models are devised that can include multimodality in the predictions by learning the modes in a latent space and incorporating learned features of the VRU's context.
[35], [37], [51] investigated deep learning models that, respectively, predict multimodal trajectories by training a network to learn the parameters of a Mixture Density Network that predicts multiple trajectories, train a network to choose the best of M hypothetical trajectory modes, and train a variational RNN that learns a conditional distribution of available modes from which the output Gaussian Mixture Model can be sampled. Although these methods all provide probabilities for the different available trajectory modes, still they do not provide a confidence estimation the model has about the multimodal predictions. [38]'s research adds a confidence estimator to their multimodal prediction method that estimates the confidence of multiple predictors. The predictor with the highest confidence value is considered for the trajectory prediction.

1.2.5 Real-time motion prediction

Fast real-time computation is important in order to implement the prediction model on AVs. [7] Addresses the importance of real-time execution of prediction models. They state that certain prediction methods (e.g. Bayesian Networks and Inverse Reinforcement Learning) are inefficient and thus not feasible for real-time applications such as AVs. Therefore, it is important that the prediction methods are efficient enough so they can be applied online in AVs to increase VRU safety. Their method of using a CNN to predict future trajectories was succesfully tested real-time in an AV. However, this method only predicts future trajectories of vehicles, not VRUs.
[51] stresses the importance for real-time predictions for applications in AVs as well. They developed the Social-VRNN that takes samples from a variational RNN's conditional distribution to predict pedestrian trajectories. Previous research used other methods, such as Generative Adversarial Networks (GANs), which required more samples for accurate predictions compared to the Social-VRNN, making the Social-VRNN much faster. However, this method does not yet perform in real-time.
[8] aims for the most efficient motion prediction models without it affecting the accuracy because it "is necessary to achieve real-time inference onboard an SDV in crowded urban environments comprising large number of VRUs" [8]. Without real-time online motion prediction, AVs cannot make use of these methods and the safety of VRUs will not improve. They developed a real-time motion prediction method, using a CNN, for all road users. Their solution shows promising results, but the dataset requires more VRU data to match the state-of-the-art prediction accuracy.

1.3 Research Questions

State-of-the-art research shows that Deep Learning methods provide means to include environmental and social context, as well as uncertainty information, to make better, long-term, path predictions including those of VRUs. Several Deep Learning methods use OGMs, a grid-based representation of the environment obtained from sensors on AVs, as input data to predict future OGMs for the long-term. OGMs contain occupancy data of the AV's environment which is often extended with uncertainty information of the occupied areas. Furthermore, the OGM can be extended with additional information layers such as semantics and dynamics of the occupied regions. Correlations of the environment and its actors can be captured when the OGM is processed by a Deep Learning network. Predicting OGMs is expected to be more accurate, also for long-term predictions, due to the amount of information that can be stored in the OGM. Attempts to make real-time OGM predictions are succeeding (Mohajerin [47] shows that it can take 60ms to predict OGMs 1s into the future), making this environment representation method a promising one for state prediction research.

Because of the potential of OGMs, this literature review focuses on the prediction of OGMs using Deep Learning. The following main research question is asked: *"What are currently the best methods to perform OGM prediction?"* This question will answered using the answers of the following sub-questions:

1. What OGM form is most suitable to use in OGM prediction methods?
2. What dataset is most suitable to generate OGM sequences for OGM prediction?
3. What is the best metric to determine the quantitative accuracy of a predicted OGM?
4. What Deep Learning network is best to use for OGM prediction?
5. What method provides the best OGM predictions?

1.4 Chapter overview

The posed research questions will be answered in the following chapters of this literature review. Chapter 2 provides information about OGMs and answers the first research question. In chapter 3, criteria are set which an ideal dataset should meet to use it for OGM prediction. This chapter will answer the second research question. The third research question will be answered in chapter 4, in which several metrics to determine the quality of OGMs are compared and tested against some criteria that are required for accurate error estimations of OGM predictions. Chapter 5 provides an overview of deep learning networks that process sequential data for prediction purposes. In this chapter, the answer to the fourth research question is provided. The last research question will be answered in chapter 6, in which several methods are highlighted that use OGMs for their state predictions. Chapter 7 contains a conclusion in which the main research question of this literature review is answered. The final chapter is a research proposal for my Master thesis to which the conclusion of this literature review provides the basis.

2 Occupancy Grid Mapping

In 1985 Moravec and Elfes [56] investigates the use of sonar measurements to map the surroundings of a robot. By gathering sonar measurements from multiple points of view (having several sensors on the robot) and from multiple instances in time, information about the robot's surroundings is accumulated. This information is used to compute the state of each cell within a rasterized 2D bird's-eye view (BEV) map of the robot's surroundings is 'occupied', 'unoccupied', or 'unknown' (when there is no sensor data available of that cell), as can be seen in Figure 2. This is called an OGM.

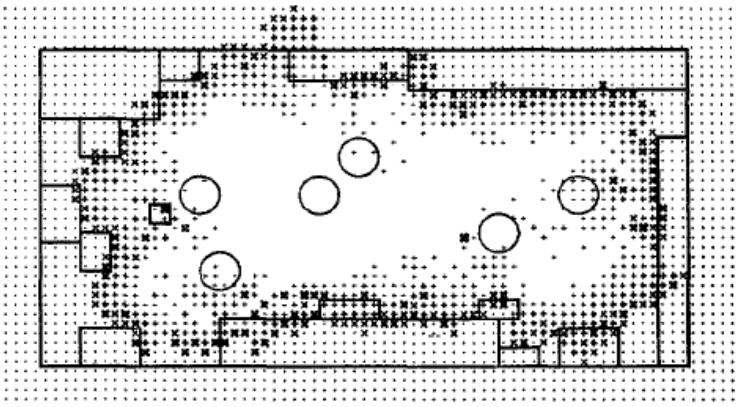


Figure 2: This image shows an OGM of a robot's surrounding environment. The solid lines represent the outline of the room and of the major objects that formed the robot's environment. The large circles in the image are the positions at which the robot performed measurements of its surroundings. The measurements are used to generate the OGM, which is a BEV raster in which each cell represents the state of an area within the environment. Each cell contains a symbol or white space. Empty areas with a high certainty factor are represented by white space. Empty areas with lower certainty factors are represented by "+" symbols of increasing thickness the lower the certainty. Occupied areas are represented by "x" symbols, and Unknown areas by "." symbols. Image from [56].

Formally, the OGM is often represented in a tensor or matrix, m, of which each element (cell), $m_{i,j} : i \in [0, M], j \in [0, N]$, represents the occupancy state of an area in the robot's surrounding world. The size of that area depends on the sensor's and grid cell's resolution. The occupancy probability of a cell $m_{i,j}$ is a discrete random variable with two exclusive and exhaustive values, occupied (O) and empty (E), resulting in equation 1 [57]. Therefore, knowing only the probability of a cell being occupied $P[m_{i,j} = O]$, which is a number between 0 and 1, is enough to determine a cell's state. Given a cell's probability of being occupied, one out of three different states is assigned to that cell based on its probability value. If the probability value is close to 0, the state is 'Empty', while with a probability value close to 1 the state is 'Occupied'. If a cell's probability of being occupied is 0.5, which means that the probability of the cell being empty is also 0.5, it will be assigned the 'Unknown' state. This is the case for cells corresponding to areas that are unobserved or areas of which conflicting measurements are obtained.

$$P[m_{i,j} = O] + P[m_{i,j} = E] = 1 \tag{1}$$

For simplicity, the notation $m_{i,j}$ will be used for the case where the cell is occupied $m_{i,j} = O$ and $\neg m_{i,j}$ for the empty state $m_{i,j} = E$. These state values can be determined in several ways. In section 2.1, two probabilistic methods to obtain occupancy state values and generate OGMs will be discussed. The first method uses an inverse sensor model and the second uses a forward sensor model. In section 2.2, an evidential method for OGM generation that uses the DST will be discussed. Then, in section 2.3, some alternative, less common OGM generation methods are briefly discussed. Besides ways to generate the traditional OGM maps that provides information about the cell occupancy, there are extended OGM forms that contain more information. Section 2.4 will elaborate on some of those extended forms. Finally, section 2.5 concludes this chapter by answering the question: "What OGM form is most suitable to use in OGM prediction methods?"

2.1 Probabilistic Occupancy Grid Mapping

In probabilistic occupancy grid mapping, Bayesian reasoning is used to estimate the state of each cell in the grid. Two processing stages are required to compute the probability of the cell states. First, a sensor measurement is interpreted using a sensor model. Second, the sensor reading is used to update the OGM cell's estimates using Bayes' theorem. [58] [59]. The two main sensor models that are used to interpret the sensor data are the inverse sensor model [58] and the forward sensor model [60]. The selected sensor model affects the assumptions that can be made about the OGM and thus affects the performance and computing time of the updated OGM cell states. Both sensor models and their implementation of the two processing stages, using Bayes' theorem, will be explained in the next two subsections. In the third subsection, both models will be compared.

2.1.1 Bayesian update with inverse sensor model

The inverse sensor model is formulated according to formula 2. It signifies the conditional probability of the complete occupancy grid map of the environment m, given the complete sets of sensor measurements $z_{1:T}$ and corresponding robot poses $x_{1:T}$. This model obtains the map state probability inversely to how the measurements are generated, since measurements are generated *given* the map (environment). [61].

$$p(m|z_{1:T}, x_{1:T}) \tag{2}$$

Given an OGM's size of $M \times N$, there are $2^{M \times N}$ possible grid configurations. As a result, the required computing power needed to estimate a complete OGM scales exponentially with the grid size. To tackle this, each cell in the OGM is assumed to be conditionally independent given the measurements and the poses, which results in equation 3, where $m_{i,j}$ stands for a single grid cell's occupied state. This computation requires much less computing power. [62] [61].

$$p(m|z_{1:T}, x_{1:T}) = \prod_{i,j} p(m_{i,j}|z_{1:T}, x_{1:T}) \tag{3}$$

Also, a static world assumption is made in equation 4 which means that a measurement z at time t is conditionally independent from the previous measurements and poses, given the OGM cell's state knowledge. Using Bayes' rule, equation 5 shows the static assumption in the form that requires the conditional probability of the cell's state given the measurement and pose. This is easier to compute because the cell's state is binary as opposed to the measurement, which can have a much larger range depending on the sensor [61].

$$p(z_t|m_{i,j}, z_{1:t-1}, x_{1:t}) = p(z_t|m_{i,j}, x_t) \tag{4}$$

$$p(z_t|m_{i,j}, x_t) = \frac{p(m_{i,j}|z_t, x_t)p(z_t|x_t)}{p(m_{i,j}|x_t)} \tag{5}$$

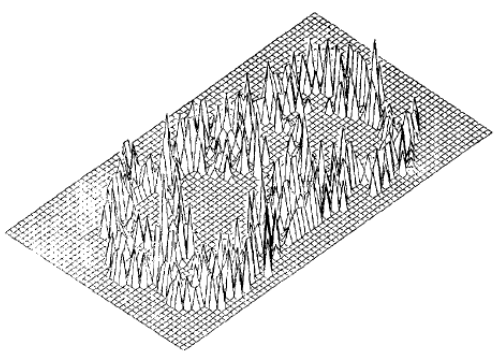
By assigning the estimated OGM as the prior estimation for the next time step and given equations 3 and 5, Bayes's theorem can be used for every grid cell $m_{i,j}$ to update the OGM m recursively every time new sensor data becomes available. This is shown in equation 6 [63].

$$p(m_{i,j}|z_{1:t}, x_{1:t}) = \frac{p(z_t|m_{i,j}, z_{1:t-1}, x_{1:t})p(m_{i,j}|z_{1:t-1}, x_{1:t})}{p(z_t|z_{1:t-1}, x_{1:t})} = \frac{p(m_{i,j}|z_t, x_t)p(z_t|x_t)p(m_{i,j}|z_{1:t-1}, x_{1:t})}{p(m_{i,j})p(z_t|z_{1:t-1}, x_{1:t})} \tag{6}$$

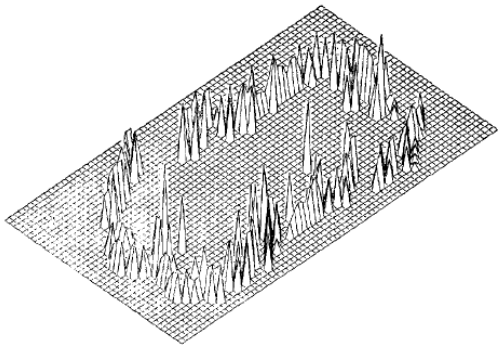
Because sensors are never ideal, the sensor model transforms a single sensor reading $p(z_t|x_t)$ to a Gaussian range around the reading. This way, a measured cell's neighbors are also given a measurement probability value to consider the inaccuracy of the sensor. Then, equation 6 computes for each (observed) OGM cell a probabilistic estimate of its occupancy state. Unobserved cells are set to have a $p(m_{i,j})$ of 0.5 and are thus considered 'unknown'. Afterwards, a threshold is set which determines at what probability a grid cell is considered occupied (Figure 3b), a two-dimensional grid map can be created that can be used for purposes such as motion planning and environment interpretation (Figure 2).

2.1.2 Bayesian update with forward sensor model

The forward sensor model is formulated in formula 7. When comparing to the inverse sensor model, this model also makes the static world assumption (i.e. a measurement is conditionally independent from the previous measurements and poses), but does not assume that the grid cell's are conditionally independent. It is a generative model that is modeled as a likelihood. Given the world that is represented by the OGM m, and the complete set of poses $x_{1:T}$, this formula would give the most likely set of sensor measurements $z_{1:T}$ [61].



(a) This figure shows the occupied areas in the map where the height of the peaks shows the occupancy certainty factors.[56]



(b) This figure shows the occupied areas in the map after thresholding the occupancy certainty factors.[56]

Figure 3: Thresholding of the occupancy grid map data.

$$p(z_{1:T}|m, x_{1:T}) \tag{7}$$

To find the OGM, the likelihood formula 7 needs to be maximized by iteratively adjusting m. To do so, a reliable sensor model needs to be made. Not all measurements are caused by obstacles, some are erroneously made. Therefore, it is assumed that all sensor measurements, z, can be subdivided into the following three cases to model the sensor. [60] [61].

1. **Non-random:** A non-random measurement is caused by an obstacle that lies within the sensor's range.
2. **Random:** A random measurement covers the remaining causes of a sensor measurement such as specular reflections or false positives.
3. **Maximum reading:** When the sensor does not detect an obstacle it will return a value that is equal to the maximum range z_{max} of the sensor.

These cases can be modeled using binary correspondence variables, $c_{t,k}, c_{t,*}$ and $c_{t,0}$ respectively, which are linked to their specific probability values. The respective correspondence variable is equal to 1 if the measurement corresponds to that particular case. This in turn determines the conditional probability of the measurement given the c_t value, which results in equation 8 [60] [61]

$$p(z_t, c_t|m, x_t) = p(z_t|m, x_t, c_t)p(c_t|m, x_t) \tag{8}$$

where

$$p\left(c_t \mid m, x_t\right) = \begin{cases} p_{\text{rand}} & \text{if } c_{t,*} = 1 \\ p_{\max} & \text{if } c_{t,0} = 1, K_t \geq 1 \\ \left(1 - p_{\text{rand}} - p_{\max}\right) \prod_{i=1}^{k-1} \left[\left(1 - p_{hit}^{(i)}\right)\right] p_{hit}^{(k)} & \text{if } c_{t,k} = 1, k \geq 1 \end{cases} \tag{9}$$

and where p_{rand} is the random probability, p_{max} is the maximum reading probability, and p_{hit} is the probability function of the obstacle's coverage within the sensor. K_t and k are the total number of obstacles and an obstacle instance respectively. By taking the logarithm of equation 8 and regarding the complete set of sensor data, the expected log-likelihood can be computed. The OGM m can be found when the log-likelihood is maximized using the Expectation Maximization(EM) algorithm. However, this algorithm requires a high computational effort, which makes it impossible to perform real-time on AVs compared to the inverses sensor model [60]. Recent research uses maximum a posteriori (MAP) inference instead of the EM algorithm to increase the convergence speed of the OGMs, but real-time performance has not yet been acquired [64].

2.1.3 Comparison of the sensor models

In this subsection, the inverse and forward sensor models are compared. Figure 4 shows the difference between the inverse sensor model and the forward sensor model, which can be compared to the ground truth.

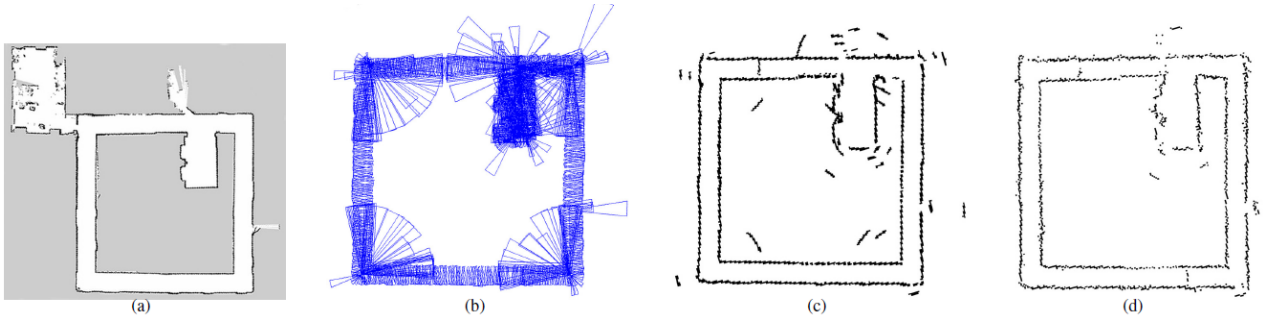


Figure 4: The experimental results from Carvalho's research [61] which compares the inverse sensor model with the forward model. Figure (a) is the ground truth map. Figure (b) are the taken measurements. Figure (c) is the computed OGM using the inverse sensor model. Figure (d) is the computed OGM using the forward sensor model.

The forward sensor model is more accurate but also requires much more computational effort [61]. Both probabilistic methods have in common that they cannot tell the difference between incompleteness (i.e. ignorance) and uncertainty. When the occurrence of an event does not hold absolute confidence, this lack of confidence can have two causes. The first cause could be that due to a lack of information another possible event is not excluded. This lack of information is defined as ignorance. A second cause could be that the occurrence of the event is not definitely true, which is defined as uncertainty [65]. The Dempster-Shafer Theory (DST) [66] [67], however, can distinguish between ignorance and uncertainty. This theory can be used to generate better OGMs. In the following subsection, evidential occupancy grid mapping, using the DST, will be explained.

2.2 Evidential Occupancy Grid Mapping

In evidential Occupancy Grid Mapping, the Dempster-Shafer Theory (DST) is used to estimate the state of the occupancy grid cells. This method, as opposed to the probabilistic method, can distinguish ignorance (the cell is either empty or occupied) from uncertainty (there is evidence that the cell is both empty and occupied with a certain belief). However, more computational power is required to for the evidential method compared to the probabilistic method [59]. The following section will first explain the mathematics behind DST. Then, its application on OGMs is discussed.

2.2.1 The Dempster-Shafer Theory

The DST formalizes the transferable belief model. The model defines a discrete Frame of Discernment (FOD) which contains the set of possible states of a system. In the case of an OGM, the FOD is $\Omega = \{E, O\}$, with E for Empty and O for Occupied. A mass function M is defined that maps the powerset 2^Ω of Ω to the domain $[0\ 1]$. The powerset is the set of all subsets of Ω including the empty set $\emptyset$ and itself ($2^\Omega = \{\emptyset, E, O, \Omega\}$). If A is an element in 2^Ω, then $M(A)$ represents the amount of evidence (mass) that supports hypothesis A within the domain $[0\ 1]$. Then, two properties are set to the mass function M. First, $M(\emptyset) = 0$, and second, formula 10 verifies that the sum of the masses of each hypothesis in the powerset is equal to 1. This means that it is assumed that the powerset 2^Ω is complete and that no evidence will be obtained that supports none of the hypotheses in the FOD. A mass function with these two properties is called a Basic Belief Assignment (BBA) mass function. [59]. A BBA mass function allows for the assignment of lower and upper bounds of a probability interval, belief $Bel()$ and plausibility $Pl()$ respectively, that represent the support to the hypotheses $A \in 2^\Omega$.
Belief is the sum of the masses of all the hypothesis's subsets including the hypothesis, as is shown in equation 11, where A and B represent hypotheses (e.g. for the Ω-hypothesis, this would be the mass of Ω and the subsets of Ω: E and O]). In equation 12 it can be seen that the plausibility is computed by taking 1 minus the sum of the masses that exclude the hypothesis (e.g. for the hypothesis O, the plausibility $Pl(O)$ would be 1 minus $M(\emptyset)$ and $M(E)$ which is 0.3). [59].

$$\sum_{A \in 2^\Omega} M(A) = 1 \tag{10}$$

$$Bel(A) = \sum_{B|B \subset A} M(B) \tag{11}$$

$$Pl(A) = \sum_{B|B \cap A \neq \emptyset} M(B) = 1 - \sum_{B|B \cap A = \emptyset} M(B) \tag{12}$$

An example of a BBA mass function is shown in table 2, given the powerset $2^{\Omega} = \{\emptyset, E, O, \Omega\}$. In this example, a sensor reading obtains information that a certain grid cell is empty. The sensor reading's probability of being reliable is 0.7 and 0.3 for being unreliable. This information can be used to assign a subjective probability (mass) to each hypothesis, which sum up to 1. A reliable sensor will give a true reading, so the hypothesis of the grid cell being empty ($m(E)$) is assigned a mass of 0.7. However, given that there *is* a grid cell ($m(\emptyset) = 0$), the sensor reading has a probability of 0.3 that it is unreliable. This does not mean that the grid cell is occupied with a probability of 0.3, but it means that it's state is uncertain with that probability. Therefore, a mass of 0.3 is assigned to the hypothesis $m(\Omega)$ that states that the grid cell is either empty or occupied. The hypothesis of the cell being occupied ($m(O)$) will have a mass of 0, since there is no evidence that supports it. [68]. Subsequently, the belief $Bel()$ and the plausibility $Pl()$ of the hypotheses are computed, giving the lower and upper probability bounds for the hypotheses.

Table 2: An example of a BBA mass function.

Hypothesis	Mass	Belief	Plausibility
$M(\emptyset)$	0	0	0
$M(E)$	0.7	0.7	1.0
$M(O)$	0	0	0.3
$M(\Omega)$	0.3	1.0	1.0

2.2.2 Applying DST to generate OGMs

To generate an evidential OGM, given independent sensor data, each grid cell is assigned a mass function $M_{i,j}$ with beliefs and plausibilities. If new sensor data about a grid cell is obtained, the DST method can fuse the mass functions of the current cell state and the new sensor data according to the joint mass equations 13 and 14. [59].

$$M_1 \oplus M_2(A) = \left\{ \begin{array}{ll} \frac{M_{1\cap2}(A)}{1-M_{1\cap2}(\emptyset)} & A \neq \emptyset \\ 0 & A = \emptyset \end{array} \right\} \tag{13}$$

Where $\cap$ is the conjunctive rule with B and C hypotheses and A the joint hypothesis:

$$M_{1\cap2}(A) = \sum_{B,C\in2^{\Omega}|A=B\cap C} M_1(B) \cdot M_2(C) \tag{14}$$

Then, a probability measure can be taken from the mass function according to equation 15. Here, A and B represent hypothesis subsets of powerset 2^{Ω}. The cardinality (number of elements in a subset) is denoted as two vertical bars $||$.

$$P(A) = \sum_{B\in2^{\Omega}} M(B) \cdot \frac{|A \cap B|}{|B|} \tag{15}$$

This equation is not bijective, meaning an infinite number of mass functions can be found given the same probability. This is because information that distinguishes ignorance from uncertainty is lost when the mass function is transformed to a probability. This lost information in probabilities is exactly what makes the DST method a more accurate but also a more computational method for OGM generation. [59]. The processing time of the DST method is about 1.5 times higher than that of the inverse sensor model probabilistic method [69]. Figure 5 shows the qualitative differences between the probabilistic approach (center) and the DST approach (right).

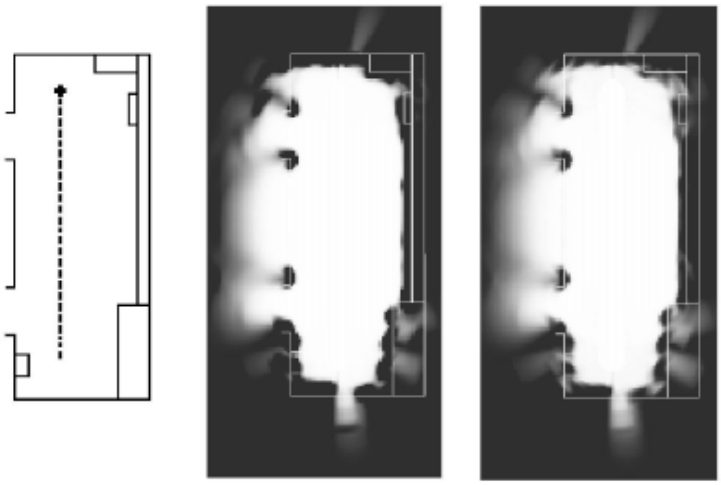


Figure 5: Ribo and Pinz [69] compare the probabilistic method (center) with the DST method (right) to generate OGMs. The ground truth schematic is shown left.

2.3 Alternative methods for Occupancy Grid Mapping

Besides the probabilistic and evidential approach to generate OGMs, there are more methods to determine which cells in the grid are occupied. Two methods, the possibilistic method and the NN method are discussed in this section. These methods are less common than the probabilistic and evidential methods, therefore they are not explained in depth in this literature review. For more in depth information about the methods it is advised to read the papers from which the methods are cited. At the end of this section comparison methods of OGMs are discussed.

2.3.1 Possibilistic Occupancy Grid Mapping

Oriolo [70] defines the OGM as two fuzzy sets where one set is empty (E) and the other occupied (O). Based on measurements, each cell is assigned a partial membership to states E and O. This partial membership allows to process and distinguish insufficient (ignorance) from uncertain information. Ribo and Pinz [69] compare this fuzzy method (also called possibilistic) with the probabilistic and evidential methods and conclude that the possibilistic method is more conservative and thus more robust towards outliers compared to the other two methods. However, its robustness also causes loss of information and the emergence of artifacts. Like the DST method, the possibilistic method requires about 1.5 times more computation time than the inverse sensor model probabilistic method.

2.3.2 NN-based Occupancy Grid Mapping

Thrun [71] uses an Artificial Neural Network (NN) to generate an OGM from measurements. In a simulated environment, the NN is trained to interpret sensor data and estimate a confidence that a cell is occupied. This trained NN is then tested in a real environment and shows it can model unknown environments efficiently, however the network cannot be trained until convergence because then it would overfit on the simulated environment and perform worse in real situations. Collins [72] compares the NN method with the probabilistic method. The NN method performs worse that the probabilistic one because it has a tendency to overestimate the free space beyond the actual environment borders. It also tends to model the sensor's extremities (maximum value due to no detection) as occupied areas, while the probabilistic approach's algorithm will ignore these extremities. Van Kempen [73] takes the use of NNs to generate OGMs even further. Van Kempen proposes a deep inverse sensor model together with a PointPillars architecture (often used to process LiDAR data) extended with an evidential prediction head to estimate an evidential OGM, including the uncertainties. The network is trained using a synthetic dataset. The network is then tested on synthetic data and on real-world data. It shows promising results on both synthetic data and real-world data. It performs better than classical approaches on the synthetic data, but the generalization capabilities are not sufficient yet to have accurate results on the real-world data. In future research, van Kempen suggests to use a more diverse dataset to train the network on for better generalization.

2.3.3 Comparison methods for OGMs

Metrics to measure and decide what OGM generation method is the best are not easy to define. The computation time and memory requirements to generate the OGM can be determined and used as a metric to compare efficiency of the

OGM methods. Most other methods to determine and compare the quality of OGMs are qualitative (visual insepction) [69]. Collins [72] summarizes some quantitative metrics based on image similarity measures and one that bases the OGM quality on the usefulness of the map for a robot to find paths, compared to the ground truth. More about metrics to use on OGMs will be discussed in section 4.

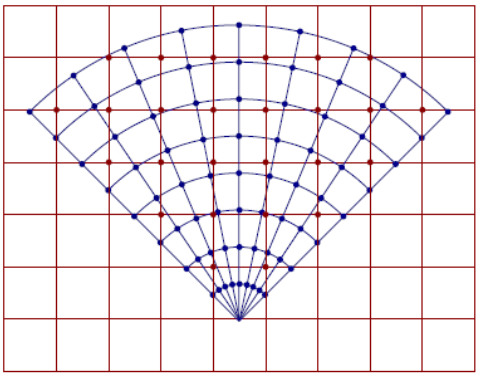
Besides ways to generate the traditional OGM maps that provide information about the cell occupancy, there are extended OGM forms that contain more information. The next subsections will elaborate on some of those extended forms.

2.4 Occupancy Grid Map extended forms

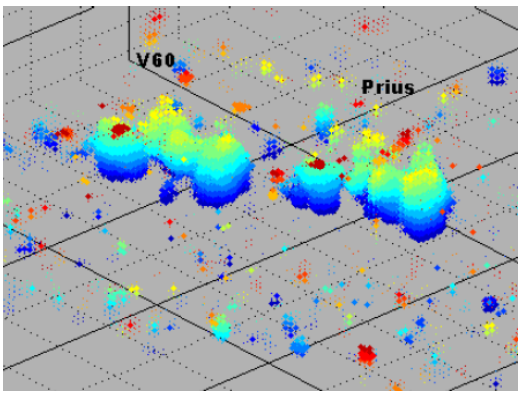
Many variations of the OGM have been invented and researched that represent environments in more complex methods, including for example 3D information [74], semantic segmentation of the cells [75], or dynamic information of the occupied regions [76].

2.4.1 3D Occupancy Grid Mapping

Degerman [74] proposes a way to create 3D Occupancy Grid Maps using 3D RADAR data. Degerman uses a binary Bayes filter to estimate the occupancy probabilities of the assumed independent voxels (grid cells in 3D). The RADAR data is obtained in spherical coordinates (radius r, scanning angle θ, and azimuth angle ϕ) after which those measurements are mapped to the Cartesian coordinates of the 3D OGM. As shown in Figure 6a, the spherical measurement grid cells are dense in short range while the distances between them increase for longer ranges. Therefore trilinear interpolation is used to combine the spherical data points when they are mapped to the Cartesian coordinates. An example of the resulting 3D occupancy grid is shown in Figure 6b.



(a) The 2D image of the measured spherical grid points (blue) and the Cartesian grid points that are computed by performing interpolation on the surrounding spherical points. [74]



(b) An example of a 3D occupancy grid where each voxel with a likelihood value above a certain threshold is visualized. The blue color represents lower levels while the red color represents higher levels. The clusters of occupied voxels represent two cars (V60 and Prius). [74]

2.4.2 Semantic Occupancy Grid Mapping

While classical OGMs only contain information about occupancy, Lu [75] proposes a method that utilizes the semantics of the environment in an OGM. The method they propose is an end-to-end convolutional neural network with a variational autoencoder-decoder part that takes monocular RGB camera data as input and outputs an OGM with an additional semantics channel. They distinguish the environment with four labels: road, sidewalk, terrain, non-free space. The network is trained and evaluated on the Cityscapes [77] and KITTI [78] datasets and the results show that the network is robust to sparse input data and a weak ground-truth. Figure 7 shows an illustration of the semantic OGM method.

2.4.3 Dynamic Occupancy Grid Mapping

Besides having information about semantics in an OGM, for purposes such as motion planning and tracking, information about the environment's dynamics is also desired. Danescu [79] proposes a particle based method for tracking the dynamic driving environment in occupancy grids. This method represents the world as a 2D BEV grid in which each

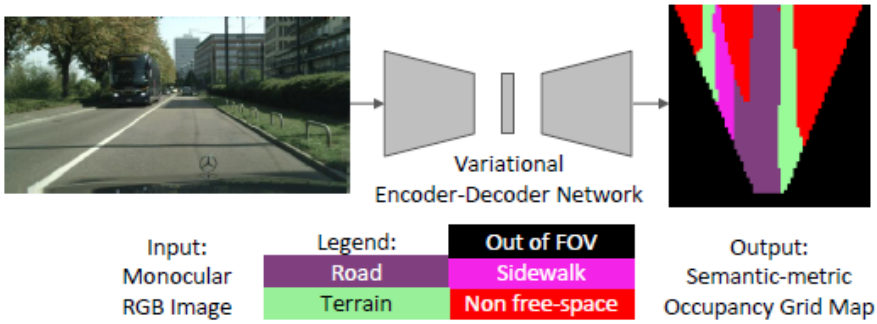


Figure 7: An illustration of the semantic OGM method that takes a monocular RGB image as input and outputs a semantic OGM [75].

obstacle is represented by a set of particles that are located in the grid's cells. Each particle has their own position and speed and can move between cells based on their dynamics. Also, particles can be created or destroyed over time. The occupancy probability of a cell C is estimated by the ratio of the number of particles in that cell and the total number of particles allowed for a single cell N_C. The number of particles allowed for a single cell is predetermined, while the total number of particles in the grid, N_S is not fixed and is dependent on the number of obstacles in the environment. By taking the average speed of all particles within a grid cell, the speed estimation of that cell is estimated. If obstacles in the environment overlap, the particles in a single cell can have different speeds. Clustering of the speeds can be used to model the speeds of multiple obstacles that overlap in a cell. This way, a tracking algorithm is used to create, update, and destroy particles to estimate the occupancy and dynamic state of the real world. These states generally are saved in separate channels of the grid, one for the occupacy state, and two for the speed and orientation of the grid cells.

Nuss [76] continues this research and proposes an improved method, using the Dempster-Shafer theory, to assign particles to grid cells. They name this method the Dynamic Occupancy Grid Map (DOGMa). An example is shown in figure 8.

Figure 8: A Dynamic Occupancy Grid Map example in which the static parts (occupancy information) are shown in black and white, with black occupied and white emtpy space. The dynamic information is shown in color. The color code represents the direction of the grid cell corresponding to the color wheel at the bottom right. The saturation value of the color determines the velocity, where the velocity is higher if the saturation is higher. [76].

Besides the 3D OGM, semantic OGM and the DOGMa, there are more variations to occupancy grid maps. In some papers, the term OGM is not used however, making it sometimes hard to define what is still an OGM and what isn't. For example, Wu [54] proposes a method to predict future environments using Bird's Eye View (BEV) maps. The paper states that they define an alternative to OGMs, which they call a BEV map. The BEV map is a grid map with discrete occupancy information in each cell. The BEV map has multiple channels. Each channel represents another height layer of the environment, which basically means that the whole BEV map is a 2D pseudo-image, because all layers together form a 3D representation of the environment. In principle, this approach is similar to a 3D OGM. In Wu's method, a Deep

Learning Network is used to apply semantics to the BEV map and estimate temporal features (the dynamics). In essence, Wu's method creates OGMs that are 3D and incorporate semantics and dynamics. In this literature review, environment representations such as Wu's are also considered OGM variations.

2.5 What OGM form is most suitable to use in OGM prediction methods?

What OGM form is best depends on the requirements of the prediction method. Table 3 shows an overview of the different OGM generation methods. In the case of AVs, there will be a trade-off between accuracy and computation time. If a fast method is desired, the inverse probabilistic method would be the best choice. The downside is that the accuracy will be lower compared to most other methods and there is no distinction between ignorance and uncertainty. If a higher accuracy is desired and there is the means to carry a higher computational load, the DST or possibilistic generation methods are more suitable because of their high accuracy and distinction of ignorance and uncertainty. The possibilistic method is more conservative than the DST method, which results in fewer outliers but more artifacts and information loss in the OGMs compared to the DST method. In the context of AVs, having artifacts in the OGMs and a higher information loss both might result in dangerous situations in which the AV either perceives something that is not present, or does not perceive something that is present. Therefore, the better option would be the DST method. Due to the high computational effort of the forward sensor model, and because the NN-based method is not developed enough to acquire a good accuracy in real situations, these options are considered the least suitable for OGM prediction purposes.

Table 3: This table gives an overview of the properties of each OGM generation method based on the information in this chapter.

	Accuracy	**Computation Effort**	**Ignorance-Uncertainty Distinction**
Inverse Probabilistic	Average	Low	No
Forward Probabilistic	High	High	No
DST	High	Average	Yes
Possibilistic	High	Average	Yes
Neural Network Based	Low	Unknown	Yes

Further, regarding the extended forms, each extension will provide the OGM with more information. Generally, when performing predictions, having more information will result in better predictions. The trade-off in this case is again one of accuracy versus computation effort. Whether the amount of additional information is worth the additional computation effort depends on the goal of the predictions.
The next section will discuss the available datasets that are used to obtain Occupancy Grid Maps from for research and what metrics are used to determine the accuracy of those OGMs.

3 Datasets

When researching OGM prediction using deep learning networks, it is important to choose a dataset (or multiple datasets) that optimally fulfills the requirements for that purpose. The dataset will be used to train and evaluate the deep learning network on. Not only the performance of deep learning network, but also the time that is required to pre-process, train, and evaluate the network is greatly dependent on what dataset is used. Also, the performance of a network should be compared reliably. Therefore, it is important that a dataset is chosen on which other research has based their network training and evaluation on, so there are enough results for reliable comparisons. To ensure that the optimal dataset is chosen for OGM prediction research, the following list of criteria is devised which the dataset has to meet. Based on how the dataset scores on this list, an informed decision can be made.

1. The dataset contains data of ego-vehicle centered traffic scene sequences that provide at least 2D BEV information of the environment.
2. The dataset provides enough diverse data to train and evaluate a network on.
3. The dataset contains traffic scene sequences, or provides means to easily generate them.
4. The sequences have enough frames per sequence and a frame frequency that is suitable for capturing road user behavior and trajectories.
5. The sequences contain various traffic actors including VRUs.
6. The dataset data, and any OGM that can be generated from it, provides a resolution that is high enough to distinguish and to track VRUs.
7. The sequences show a diversity in environmental properties, containing VRUs, that may influence the generated OGMs (e.g. urban vs rural, dense vs sparse traffic).
8. If any extended OGM form is required for research, the dataset should also provide means to attain data that is required for such an extended form including ground truth data.
9. Results of other research using the dataset for generating and predicting OGMs is available for comparison.

There is a number of existing datasets that are closest to meeting all criteria. Three datasets were obtained for a purpose that includes motion prediction. Five datasets were obtained for object detection purposes. The last two suitable datasets were mainly obtained to research semantic segmentation on. All the datasets are recorded using a vehicle equipped with one or multiple sensors that drives through real traffic. The datasets are discussed below in three subsections based on their original purposes: Motion prediction, Object Detection, and Semantic Segmentation. At the end of this chapter is a conclusion about the datasets.

3.1 Tracking and Motion prediction Datasets

The following three datasets are obtained for tracking or motion prediction purposes. The Stanford-TRI Intent Prediction (STIP) dataset [80], the Argoverse dataset [81] and the RoboCar dataset [82].

The STIP dataset [80] is obtained using three cameras (left, front and right) with a 1216x1936 resolution at 20 Hz positioned on the recording vehicle. The data is recorded in dense urban areas in 8 US cities (in California and Michigan). The dataset contains 923.48 minutes of driving scenes, which comes down to 1,108,176 frames. The data contains 350K pedestrian instances annotated with 2D bounding boxes at 2 fps which are interpolated to cover all frames with pedestrians. There are over 25K pedestrian tracks with a median length of 4 seconds, at 20 Hz. 556 sequences are selected in which busy intersection are recorded. These sequences are subdivided into a training set containing 2525 pedestrians in 102.37 minutes of video, and a test set containing 823 pedestrians in 23.43 minutes of video.

For the Argoverse dataset [81] two 107K points LiDARs, with a range of 200m, at 10 Hz together with seven 1920x1200 resolution cameras in a 360 degrees view setting at 30 Hz, with two additional 2056x2464 resolution front cameras at 5 Hz were used to obtain the data. Localization data is obtained from GPS and motion sensors. The data is recorded in 2 US cities (Pittsburgh and Miami). The dataset contains over 320K five-second sequences containing the centroid of each tracked object in 2D BEV sampled at 10 Hz. Together these are over 19K minutes containing over 11K tracked road users and obstacles labeled out of 17 categories including vehicles, pedestrians and cyclists.

The RoboCar dataset [82] data is collected using three LiDARs with each a range of 50m at 50 Hz, and four cameras (one 1280x960 resolution camera in front at 16 Hz, and three 1024x1024 resolution cameras at 11.1 Hz at the rear and on the sides). A GPS INS combination is used for localization. The data is recorded in Oxford, UK. A 1000 km of traffic scenes are recorded and subdivided into 360s second sequences. Vehicles, pedestrians and other road users are recorded in the dataset, however none are labelled. It is therefore unkown how many pedestrians or vehicles are recorded.

3.2 Object Detection Datasets

This subsection will discuss five datasets that were originally obtained for object detection purposes. The Eurocity Persons 2.5D (ECP2.5D) dataset [83], the Berkely DeepDrive 100K (BDD100K) dataset [84], the Karlsruhe Institute of Technology and Toyota Technological Institute (KITTI) dataset [85], the nuScenes dataset [86], and the Waymo Open dataset [87].

The ECP2.5D dataset [83] is obtained using a Velodyne HDL-64E LiDAR, which has a range of 120m and a rotation frequency of 5-20 Hz, and a 1920x1014 resolution front camera at 20 Hz. The localization data is obtained using a GPS INS combination. The recordings were made in 30 different cities in 12 diverse European countries. The dataset contains 136K persons in 46K frames. Together there are 218K pedestrians and 19K riders. More than 140K 2D person annotations are made, of which around 136K have 3D information.

The BDD100K dataset [84] is collected using crowd-sourcing. Drivers could upload their data obtained by using a 1280x720 resolution front camera at 30 Hz together with GPS/IMU localization data. Recordings were mostly made in San Francisco and the Bay Area, New York, and Berkeley. The dataset contains data for object detection, but also for tracking and semantic segmentation, since object detection is not the sole purpose of the BDD100K dataset. The dataset contains 100K driving videos, each of a duration of 40 seconds. For all 100K videos, 10 object categories are annotated with 2D bounding boxes and 8 lane marking categories and the vehicle's drivable area are annotated on a pixel-level. Other annotations for object detection and semantic segmentation are done for every 10th second in each video. For those 10K sample frames 40 object classes are annotated on a pixel-level. For the tracking task, 2K 40-second videos containing about 400K frames are annotated at 4 Hz. The dataset contains 130.6K track identities and 3.3M bounding boxes for the training and validation set. There are 129K instances of pedestrians and about 1M vehicles in the dataset.

The KITTI dataset [85] is obtained using a Velodyne HDL-64E LiDAR with a 120m range at 10 Hz and four 1392x512 resolution cameras (two color cameras and two grayscale cameras) at 10 Hz. A GPS/IMU combination is used for localization data. The data is recorded in one city (Karlsruhe, Germany). Besides object detection, this dataset also provides data for stereo vision, optical flow, visual odometry, SLAM, and 3D object tracking tasks. Dynamic objects of 8 different classes, including vehicles and pedestrians, are annotated in the form of 3D bounding box tracklets in the LiDAR point-clouds, which are projected into the camera images. 9400 pedestrians are annotated and 3300 riders. The dataset contains 22 sequences in traffic that together span a distance of 39.2 km and consist of 41K frames at 10 Hz.

For the nuScenes dataset [86], a LiDAR that can generate a 1.4M point 360 degree view at 20 Hz, five radars around the vehicle with a 250m range at 13 Hz, and six 1600x900 resolution cameras around the vehicle at 12 Hz are used to collect data. Also a GPS/IMU and CAN bus data is collected for localization. The dataset is recorded in 2 cities (Boston and Singapore). There is 15 hours of data that together span a total distance of 242 km of diverse driving conditions. The nuScenes dataset is collected for object detection as well as object tracking. Therefore, it contains 1000 sequences of 20 seconds, annotated at 2 Hz. Objects of 23 classes are annotated, including vehicles, pedestrians, and riders. The nuScenes paper [86] compares its dataset against the KITTI dataset [85]. It concludes that training on the KITTI dataset, with its smaller size compared to nuScenes, affects a network's performance.

The Waymo Open dataset [87] (in short Waymo), is obtained using five LiDAR sensors and five cameras (three 1920x1280 resolution front cameras and two 1920x1040 side cameras) sampled at 10 Hz. The dataset was originally recorded in 3 cities (Phoenix, San Francisco, and Mountain View) and later extended with another 3 cities (Los Angeles, Detroit, and Seattle). Besides object detection, the Waymo dataset's extension focuses on data for motion prediction. It now contains 103K 20 second sequences at 10 Hz (together over 20M frames and 574 hours of data). It contains 10.8M objects with tracking IDs, labels for three object classes (vehicles, pedestrians, and cyclist), and 3D bounding boxes of each of those objects. Each sequences contains 3D map data and is further broken down into 9 second windows (1 second of historic frames and 8 seconds of future frames) with 5 second overlap for motion prediction purposes.

3.3 Semantic Segmentation Datasets

This section discusses two datasets that are originally generated for semantic segementation tasks. The Apolloscape dataset [88] and the Cityscapes dataset [77].

Apolloscape [88] is a dataset which is collected for the purpose of scene parsing (semantic segmentation on pixel-level). The data is obtained using two VUX-1HA LiDARs and a VMX0CS6 camera system with two 3384x2710 resolution cameras. Localization data is collected with a IMU/GNSS combination. The dataset contains almost 144K frames with pixel-level annotations for semantic segmentation. Furthermore, 89K instance-level annotations are provided for movable objects. 25 different labels are annotated covering five groups. Also, 28 lane markings are annotated. Together there are 543K pedestrian instances and 1.99M vehicle instances in the dataset.

The Cityscapes dataset [77] is obtained using a 1920x1080 resolution stereo camera (OnSemi AR0331) at 17 Hz. 16 bit HDR and 8 bit LDR RGB images are recorded. Localization data is collected with a GPS/odometry sensor combination and the outside temperature is measured as well. The dataset is recorded in 50 different cities, primarily in Germany but also in neighboring countries. From 27 cities, 5000 images are selected for dense pixel-level annotation. Annotations are done on every 20th frame of a 30-frame video sequence. For the remaining 23 cities, coarse annotation is performed on a single image every 20 seconds or 20 meters driving distance (whichever comes first). This yields another 20K annotated images. 30 classes are annotated which are grouped into 8 categories. The dataset contains 24.4K annotated pedestrians and 41K vehicles. Humans and vehicles are also annotated on an instance level.

3.4 What dataset is most suitable to generate OGM sequences for OGM prediction?

Table 5 shows an overview of the datasets that are discussed in the previous subsections. Based on the properties of each dataset it is evaluated how well they meet the criteria that were set at the beginning of this chapter. Table 4 shows an overview of how well each datasets scores per criterion.

	STIP [80]	Argoverse [81]	ECP2.5D [83]	BDDK100 [84]	KITTI [85]	NuScenes [86]	Waymo [87]	Apolloscape [88]	Cityscapes [77]	RoboCar [82]
1	N	Y	Y	N	Y	Y	Y	Y	Y	Y
2	A	A	A	G	B	A	A	B	G	B
3	A	G	U	G	B	G	G	U	U	U
4	G	G	U	U	G	B	G	U	G	G
5	Y	Y	Y	Y	Y	Y	Y	Y	Y	U
6	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
7	A	A	A	A	A	A	G	A	A	A
8	-	-	-	Pixel Semantics	-	Velocity data	-	Pixel Semantics	Pixel Semantics	-
9	-	Only OGM gen	-	-	OGM gen and pred	Only OGM gen	OGM gen and pred	-	Only OGM gen	OGM gen and pred
Scores	4.5	**6.5**	4.0	5.5	5.5	**6.5**	**7.5**	4.5	**7.0**	4.5

Table 4: The table shows for each dataset how well it scores per criterion. The scores are based on the information in table 5. The following letters are used to evaluate each criterion: **G:** Good, **A:** Average, **B:** Bad, **U:** Unknown: **-:** None/Not available/Not found, **Y:** Yes, **N:** No.
At the bottom row, each dataset is given a score based on which letter is given at the evaluation, and whether there is useful information found for criteria 8 and 9. A 'G', a 'Y', 'OGM gen and pred', and any available additional data for an extended OGM form provide 1 point, an 'A' and 'only OGM gen' provides 0.5 points, and the 'N', 'B', 'U', and '-' provide 0 points. Below is the list of criteria.
1. The dataset contains data of ego-vehicle centered traffic scene sequences that provide at least 2D BEV information of the environment. **2.** The dataset provides enough diverse data to train and evaluate a network on. **3.** The dataset contains traffic scene sequences, or provides means to easily generate them. **4.** The sequences have enough frames per sequence and a frame frequency that is suitable for capturing road user behavior and trajectories. **5.** The sequences contain various traffic actors including VRU. **6.** The dataset data, and any OGM that can be generated from it, provides a resolution that is high enough to distinguish and to track VRU. **7.** The sequences show a diversity in environmental properties, containing VRU, that may influence the generated OGM (e.g. urban vs rural, dense vs sparse traffic). **8.** If any extended OGM form is required for research, the dataset should also provide means to attain data that is required for such an extended form including ground truth data. **9.** Results of other research using the dataset for generating and predicting OGM is available for comparison.

Dataset	Year	Sensors			Sampling Frequency	Diversity of Locations	Dataset size	Diversity of labels	# Pedestrians	# Vehicles	Extended OGMs	OGM generation method	OGM prediction comparison
STIP [80]	2020	-	3 RGB cameras of res 1216x1936 (20 Hz)	-	20 Hz	1 country, 8 cities, dense urban areas	556 sequences, ~2h of recording, ~150K frames	2 classes (pedestrian, vehicle)	~3K	-	-	-	-
Argoverse [81]	2019	2 VLP-32 LiDARs with 200m range (10 Hz)	7 RGB cameras in 360 deg setup of res 1920x1200 (30 Hz) and 2 in stereo view of res 2056x2464 (5 Hz)	GPS/ Odometry sensor	10 Hz	1 country, 2 cities, dense urban areas	~333K 5s sequences, 1006h of recording, spanning ~290km	17 classes	~1.5K	~8K	-	[89]	-
ECP2.5D [83]	2020	Velodyne HDL-64E LiDAR with 120m range (5-20 Hz)	1 RGB camera of res 1920x1024 (20 Hz)	GPS/INS	-	12 countries, 31 cities, dense urban areas	~46K frames	7 classes	~218K	-	-	-	-
BDDK100 [84]	2020	-	1 camera of res 1280x720 (30 Hz)	GPS/IMU	-	(mostly) 1 country, 4 cities, dense urban areas	~100K 40s sequences	3 categories, 20 classes	~129K	~1M	Pixel-level semantic segmentation	-	-
KITTI [85]	2012	Velodyne HDL-64 LiDAR with 100m range (10 Hz)	4 cameras (2 color 2 grayscale) of res 1392x512 (10 Hz)	GPS/IMU	10 Hz	1 country, 1 city, dense urban areas	22 sequences	2 categories, 8 classes	~9.4K	~100K	-	[45] [47] [52]	[45] [47] [52]

NuScenes [86]	2019	32 beam LiDAR 1.4M points/s (20 Hz) + 5 Radars with 250m range (13 Hz)	6 cameras of res 1600x900 cameras (12 Hz)	GPS/IMU, CAN bus data	2 Hz	2 countries, 2 cities, dense urban areas	1000 20s sequences	23 classes, 8 attributes	~200K	~500K	velocity information from Radar with 0.1km/h accuracy	[89] [90]	-
Waymo [87]	2020	5 LiDAR sensors (1 mid range, 4 short range)	5 cameras (3 front of res 1920x1280 and 2 side of res 1920x1014)	-	10 Hz	1 country, 3 cities, urban and suburban areas	103K 20s sequences	3 classes	~2.8M	~6.1M	-	[52] [53]	[52] [53]
Apolloscape [88]	2018	2 VUX-1HA LiDARs with 420m range	1 VMX-CS6 camera system with 2 cameras of res 3384x2710 (no depth information)	IMU/GNSS	-	1 country, dense urban areas	-	5 categories, 35 classes, additional 28 kinds of lane markings	~543K	~1.99M	Pixel-level semantic segmentation	-	-
Cityscapes [77]	2020	-	1 stereo camera of res 1920x1080 (OnSemi AR0331) (17 Hz)	GPS, outside temperature, in-vehicle odometry sensors	17 Hz	1 country, 50 cities, dense urban areas	A 'large set' of sequences	8 categories, 30 classes	~24.4K	~41K	Pixel-level semantic segmentation	[91]	-
RoboCar [82]	2016	2 SICK LMS-151 2D LiDARs with 50m range (50 Hz) and 1 SICK LD-MRS 3D LiDAR with 50m range (12.5 Hz)	4 cameras (1 front of res 1280x960 at 16 Hz, 3 for sides and rear of res 1024x1024 at 11.1 Hz)	GPS/INS	-	1 country, 1 city, urban area	360s sequences of unknown amount	No labels	-	-	-	[23] [92]	[23]

Table 5: This table shows an overview of the datasets that are most likely to be suitable for OGM prediction purposes.

4 Metrics

The quantitative evaluation of OGM prediction methods depends highly on what metric is used to assess a method's performance. The metric determines which errors are considered and how much they weigh in the evaluation. For OGM prediction, it is important that the predictions that ensure safe motion planning for the AV are evaluated as 'better', than predictions that might cause accidents. Ensuring safe motion planning means that no (additional) dangerous situations are caused when the AV executes a path that is planned in the OGM predictions, compared to a path that is planned in the ground truth OGM. This means that a prediction must deviate minimally from the ground truth. Also, AVs have less time to correct their paths when there are large deviations or deviations close to the AV. Large displacements and errors close to the AV should therefore be considered worse than small displacements and errors further from the AV. Moreover, if there are deletions or additions of actors in the predictions, the AV might perform dangerous confronting or evasive maneuvers respectively. Therefore, deletions and additions should be penalized more than displacements. These demands result in the following list of criteria that a metric must meet in order to evaluate an OGM method that ensures that safer methods are considered better.

1. The metric can evaluate the OGM as a whole.
2. The metric can evaluate the OGM on local and on global scale.
3. The metric negatively weighs small displacements less than big displacements.
4. The metric negatively weighs errors close to the AV (often the center of the OGM) more than errors further from the AV.
5. The metric negatively weighs additions and deletions more than displacements.

4.1 Occupancy Grid Map Metrics

Mean Squared Error: [52], [45], [53]
Image Similarity: [52], [53], [93]
TP, TN, SSIM/S100: [47]
F1 measure: [23], [46]
FPR, TPR (ROC-curve): [36], [46]

Motion planning metric: Plan a (or multiple) paths in GT, plan a (or multiple) paths in the prediction. THe more similar the paths are, the better the OGM prediction is?
Experiment: Test each metric on a couple of images in which each of the criteria is tested.
TABLE: Metric INFO

4.2 What is the best metric to determine the quantitative accuracy of a predicted OGM?

5 Sequential input Deep Learning Networks

Prior knowledge about Deep Learning can be found here:

5.1 RNN

RNN Encoder-Decoder: [94]
LSTM: [95]
GRU: [96]

5.2 Sequential Convolutional Neural Networks

Temporal Convolutional Network: [97]
U-Net: [98]
Temporal Shift: [99]

5.3 Transformer Networks

Transformer: [100]

5.4 What Deep Learning network is best to use for OGM prediction?

6 Prediction Methods using Occupancy Grid Maps

6.1 Occupancy Grid Map Prediction Methods

6.1.1 PredNet based OGM predictors

Itkina [45] proposes to utilize Lotter's [101] PredNet, which is an architecture that contains a Convolutional LSTM originally used for video prediction, to predict future occupancy grid map data based on past grid maps. This network is expected to be suitable for Occupancy Grid Map (OGM) prediction because an OGM is similar to a frame in a video.

A grid map is an environmental representation that subdivides an area into single cells and estimates the states of each cell based on sensor data. Occupancy grid maps are a form of grid maps were each cell state is represented by its occupancy probability. [102]. These probabilities can be estimated with methods such as Bayes' inference or the Dempster-Shafer Theory [103]. For many robotics applications, including AVs, not only the occupancy state of its surroundings is of interest, but also the type of occupancy, such as its class. Nuss [102] proposed a method to fuse the class labeled information from camera images with the occupancy data from laser data to create occupancy grid maps with labeled objects. OGMs are a suitable environment representation for motion prediction because they can generate probabilistic maps without having data of the entire environment or making assumptions of agent behavior [45].

The convolutional layers are used to correlate the occupied cells in the grid map based on contextual information. Itkina [45] compares the PredNet's performance with a baseline that assumes a static environment (for the short period of time the predictions last), with a FCN network, and with a particle filter predictor. Itkina [45] also compares the performance of the network with multiple variations of the input OGMs. Using only static OGMs is compared with using an additional Dynamic OGM (DOGMa), which includes an additional dimensional layer to the grid map that contains dynamic state information (e.g. velocity). Furthermore, the benefits of the Dempster-Shafer environment representation in the OGMs is compared with using a probabilistic alternative. Comparisons are done qualitatively and with the MSE metric that compares each cell of the OGMs.
The research of Itkina [45] found that the PredNet outperforms the other investigated methods. Also, the DST-based method performs better than the probability based method, and the difference in performance between the OGM and the DOGMa almost none. For longer predictions, the objects in the OGM become blurry or even disappear from the environment.
To counter the blurriness and object disappearance for longer term predictions, Lange [52] proposes the Attention Augmented ConvLSTM (AAConvLSTM) for Environment Prediction. Lange [52] stresses the importance of OGM predictions, because this "approach facilitates the use of occupancy state estimation under uncertainty to update the belief of surroundings" [52]. However, the predictions must be reliable, so that is why Lange [52] tries to reduce blurriness by implementing an attention mechanism that originated from creating long-term dependencies in language processing into the PredNet architecture. Comparing the results with using the original PredNet architecture, the AAConvLSTM performs better both qualitatively and based on the MSE and Image Similarity (IS) metrics. However, blurriness and disappearance of objects remains.
Toyungyernsub [53] also attempts to counter the blurriness and disappearing objects that resulted from Itkina's [45] method. They propose a double-prong ConvLSTM network based on the PredNet architecture which separately predicts the static and dynamic input OGMs and fuses the results in a joined prediction of the OGM. The method outperforms the original PredNet both in terms of the IS and MSE metrics as qualitatively. The blurriness and disappearance of objects reduces significantly. For future work, incorporating multi-modality into the predictions is advised because the predicted object orientations were not always correct due to the variety of directions the objects could head for.
The networks of Itkina [45], Lange [52], and Toyungyernsub [53] are trained on the KITTI [78], Waymo [87] and KITTI [78], and Waymo [87] datasets respectively.

6.1.2 Convolutional neural network based OGM predictors

Hoermann [36], proposes to predict Dynamic Occupancy Gridmaps (DOGMas) using a convolutional neural network. The network requires one DOGMa of the current timestep and predicts the future 0 to 3 seconds. It learns to label the objects, static or dynamic, to provide one static channel output and several dynamic channel outputs that together form an output DOGMa. Only one input DOGMA is used because it is hypothesized that most information necessary for prediction can be found in the dynamic representation and the relation between the cells and not necessarily from the past DOGMas. The paper argues that providing a DOGMa as output is beneficial because it is independent of the sensor setup since the network does not process the raw sensor data directly. Therefore, the network can be trained with varying sensors that provide the input DOGMas. Results show that the proposed method is better than a particle filter approach. The proposed method performs multimodal predictions, can distinguish static from dynamic objects, and

provides more accurate predictions than the particle filter. This is because the convolutional layers capture the DOGMa's cell dependencies where particle filters assume independent cells. However, due to uncertainty longer term predictions become vaguer.
Schreiber [46] builds upon Hoermann's [36] research by expanding the convolutional neural network with an LSTM encoder and decoder, and skip connections with ConvLSTMs. Schreiber [46] states that the prediction of future DOGMas is comparable to a video prediction problem die to the image-like structure of the DOGMas. The LSTMs are expected to exploit spatial and capture temporal correlations, while the skip connection ConvLSTMs are expected to handle missing data due to occlusion. Schreiber's [46] method shows better F1-score results and better qualitative results compared to Hoermann's [36] previous research. The static predictions remain sharp for long term predictions, even for parts of the environment that became partially occluded in the input sequence. Partially occluded dynamic obstacles are predicted accurately as well and there is multimodality in the predictions. Communication signals and road signs are not taken into account. These cues could improve the (multimodal) predictions. Also, the data is only recorded in a stationary scenario, so the performance of the network with egomotion is not researched.

6.1.3 Deep tracking based OGM predictors

Dequaire's [23] research is based on end-to-end tracking of objects using OGMs as environment representations. The research proposes a framework that learns a model of world dynamics in an unsupervised manner. The model can predict the unoccluded state, inluding occupancy, semantics, and dynamic behavior, of the world given a sequence of partially observable input OGMs. This partially observable sequence of the environment provides varying information about the objects that are present. The sequence is used as input for a GRU with dilated convolutional layers to track and classify the partially occluded objects. Dequaire [23] argues that predicting the future is the same as tracking and classifying the environment given partially occluded OGMs of the past and 'future' OGMs that are completely occluded. The network will fill in the future occupancy in the completely occluded OGM based on the information from the past OGMs. Different network configurations' results are compared using the F1 measure and IoU to assess tracking performance and scene semantics quantitatively.
Mohajerin [47] reasons that having OGMs as output provides drivable space information without the need of several stages (detection, classification, tracking, predicting, occupancy grid update) which is required in the classic approach. Therefore, Mohajerin [47] builds upon Dequaire's [23] research and proposes a multi-step prediction of OGMs with a ConvLSTM network architecture. The suggested method learns the difference between consecutive OGMs as a compensation matrix which it then uses to predict the future OGM. This difference learning architecture outperforms Dequaire's [23] architecture in predicting future OGMs based on the Structural Similarity Index Metric (SSIM). Mohajerin [47] did discuss that the provided ground truth data (from the KITTI [78] dataset) is an inadequate target to which the predictions should be compared, since the ground truth only contains the visible borders (to the AV) within its FOV, while the model predicts the whole environment. As a result, the network can accurately predict occupied cells outside the FOV of the ground truth data which is then erroneously considered a false positive because the GT data is not complete.

6.1.4 MotionNet multi-channel BEV map predictor

Wu's [54] performs perception and motion prediction with 3D LiDAR data as input. A sequence of LiDAR sweep 3D point clouds synchronized to the current time frame is converted to BEV maps by voxellizing the point cloud and representing it as a 2D pseudo-image where the height dimension corresponds to the image channels. So it is similar to having a multi-channel OGM where each channel is another height. This representation makes convolution possible. The MotionNet makes use of spatio-temporal convolution (STC) blocks that consist of standard 2D convolutions for capturing spatial features, followed by a 3D convolution that captures the temporal features. The output of the network is a BEV grid cell map with occupancy information, object labels, and motion information in 3 channels per timestep. This is similar to a DOGMa with labelling information. Based on the L2 distance metric between the predicted objects and the ground truth, MotionNet outperforms other networks especially because of its ability to distinguish static and dynamic obstacles well. This is because networks learns to classify static objects and links a zero velocity value to them.

6.2 State Prediction Methods

6.2.1 Transformer based predictors

Li [50] proposes an end-to-end network that performs object detection and trajectory prediction of an AV's environment using it's raw LiDAR and camera data as input in the form of BEV 3D occupancy grid map data. The first part of the network performs multi sensor object detection where LiDAR and camera data are fused to provide spatial and representational features of the surrounding actors. These features are provided as a spatial input sequence of actor data to the Transformer part of the network. The transformer outputs actor interaction features that are processed by an autoregressive

recurrent model to predict the future states and features of the actors. These features can be projected onto a grid map for the AV's motion planning purposes. This network is trained and evaluated on the ATG4D [34] and NuScenes [86] datasets. The Transformer method outperforms all other methods that were trained on the ATG4D and NuScenes datasets, based on the Detection Average Precision (AP), Average and Final Displacement Error (ADE, FDE), and Trajectory Collision Rate (TCR) metrics. Qualitative results show that collision avoidance is done well compared to the baseline Transformer method. This method did not yet include pedestrians and bicyclists. Because of their more unpredictable behavior, more research should be done to verify that this network also works well on VRU motion prediction.

6.2.2 LiDAR based OGMs for state predictor input

Luo [34] proposes a real-time end-to-end 3D detection, tracking and motion prediction network which only uses a CNN network and performs a task within 30ms. An AV's LiDAR data is encoded by a 4D tensor of 3D voxel OGMs in space over several time frames. Having 3D input voxel OGMs makes the learning progress easier since the network can use priors about typical object sizes. A CNN performs 3D convolutions on the voxel OGM to detect, track, and predict objects by outputting 3D bounding boxes of the predicted future object locations. For motion forecasting the network performs well and can predict 10 frames with an average L2 distance of 0.33 meter. It has a recall of 92.5%. It performs qualitatively well for both static and dynamic object forecasting. However, due to the sparsity of the LiDAR 3D points, some objects are not detected properly and therefore not forecasted well. The network is also not tested for pedestrians and longer term predictions.

6.3 What method provides the best OGM predictions?

7 Discussion and Conclusion

8 Research Proposal

Comparing the accuracy of future Occupancy Grid Map predictions with future state predictions focussed on VRUs provided with the ECP dataset.

8.1 Why VRU behavior?

VRU behavior prediction is more challenging that predicting vehicle behavior due to the sudden and swift trajectory changes that VRUs can make. Additionally, VRUs are vulnerable, so VRU safety is more important than vehicle safety.

What output representation (OGM or states) will provide more accurate results for VRU path prediction?

Experiment: Predict future behavior of VRUs with both output representations and compare the results

8.2 Why embed environmental cues in a grid map representation?

Why could using grid maps improve the accuracy of the predictions? (static environment is incorporated, semantic segmentation of static env possible, ...)

Experiment: Compare using grid map representation with a state representation (e.g. just coordinates). This requires a metric that can be used to compare those representations or a way to translate on representation to the other and then compare it with the same metric.

OR

Experiment: Investigate the benefits of extended OGM forms. Compare the prediction results of a plain OGM input with a semantic OGM input. Compare the results of both the OGM output representation and the state output representation.

8.3 What method performs better with missing data?

Experiment: Compare the OGM output representation accuracy with the state output accuracy when there is missing input data (empty OGMs). Also, compare the difference in the OGM ouput accuracy with and without missing data. Do the same for the state output representation. See if one of the two methods is robuster (lower decrease in accuracy with missing data).

8.4 What method performs better with sensor fusion input/input from multiple sensors?

8.5 Why the ECP dataset?

It is large and diverse.

It has LiDAR and camera data that can be aligned to form a 3D pointcloud with semantic segmented labelling to create a BEV map of the environment.

A subset of the ECP dataset is made which contains tracks for motion prediction purposes.

Experiment: Use the ECP dataset to perform the experiments on.

References

[1] E. Commission, "Road safety in the european union – trends, statistics and main challenges," European Commission, April 2018.

[2] J. Cui, L. S. Liew, G. Sabaliauskaite, and F. Zhou, "A review on safety failures, security attacks, and available countermeasures for autonomous vehicles," *Ad Hoc Networks*, vol. 90, p. 101823, 2019.

[3] E. Commission, "Intelligent transport systems - road."

[4] R. Okuda, Y. Kajiwara, and K. Terashima, "A survey of technical trend of adas and autonomous driving," in *Technical Papers of 2014 International Symposium on VLSI Design, Automation and Test*, pp. 1–4, IEEE, 2014.

[5] E. Ohn-Bar and M. M. Trivedi, "Looking at humans in the age of self-driving and highly automated vehicles," *IEEE Transactions on Intelligent Vehicles*, vol. 1, no. 1, pp. 90–104, 2016.

[6] I. Cara and E. de Gelder, "Classification for safety-critical car-cyclist scenarios using machine learning," in *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, pp. 1995–2000, IEEE, 2015.

[7] N. Djuric, V. Radosavljevic, H. Cui, T. Nguyen, F.-C. Chou, T.-H. Lin, N. Singh, and J. Schneider, "Uncertainty-aware short-term motion prediction of traffic actors for autonomous driving," in *The IEEE Winter Conference on Applications of Computer Vision*, pp. 2095–2104, 2020.

[8] F.-C. Chou, T.-H. Lin, H. Cui, V. Radosavljevic, T. Nguyen, T.-K. Huang, M. Niedoba, J. Schneider, and N. Djuric, "Predicting motion of vulnerable road users using high-definition maps and efficient convnets," *arXiv preprint arXiv:1906.08469v2*, 2020.

[9] B. D. Ziebart, N. Ratliff, G. Gallagher, C. Mertz, K. Peterson, J. A. Bagnell, M. Hebert, A. K. Dey, and S. Srinivasa, "Planning-based prediction for pedestrians," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3931–3936, IEEE, 2009.

[10] C. G. Keller and D. M. Gavrila, "Will the pedestrian cross? a study on pedestrian path prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 2, pp. 494–506, 2013.

[11] N. Schneider and D. M. Gavrila, "Pedestrian path prediction with recursive bayesian filters: A comparative study," in *German Conference on Pattern Recognition*, pp. 174–183, Springer, 2013.

[12] J. F. P. Kooij, N. Schneider, F. Flohr, and D. M. Gavrila, "Context-based pedestrian path prediction," in *European Conference on Computer Vision*, pp. 618–633, Springer, 2014.

[13] R. Quintero, I. Parra, D. F. Llorca, and M. Sotelo, "Pedestrian path prediction based on body language and action classification," in *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pp. 679–684, IEEE, 2014.

[14] J.-H. Park and Y.-W. Tai, "A simulation based method for vehicle motion prediction," *Computer Vision and Image Understanding*, vol. 136, pp. 79–91, 2015.

[15] P. Trautman, J. Ma, R. M. Murray, and A. Krause, "Robot navigation in dense human crowds: Statistical models and experimental studies of human–robot cooperation," *The International Journal of Robotics Research*, vol. 34, no. 3, pp. 335–356, 2015.

[16] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social lstm: Human trajectory prediction in crowded spaces," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 961–971, 2016.

[17] S. Huang, X. Li, Z. Zhang, Z. He, F. Wu, W. Liu, J. Tang, and Y. Zhuang, "Deep learning driven visual path prediction from a single image," *IEEE Transactions on Image Processing*, vol. 25, no. 12, pp. 5892–5904, 2016.

[18] Y. F. Chen, M. Liu, and J. P. How, "Augmented dictionary learning for motion prediction," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2527–2534, IEEE, 2016.

[19] B. Kim, C. M. Kang, J. Kim, S. H. Lee, C. C. Chung, and J. W. Choi, "Probabilistic vehicle trajectory prediction over occupancy grid map via recurrent neural network," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 399–404, IEEE, 2017.

[20] E. A. Pool, J. F. Kooij, and D. M. Gavrila, "Using road topology to improve cyclist path prediction," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, pp. 289–296, IEEE, 2017.

[21] D. Varshneya and G. Srinivasaraghavan, "Human trajectory prediction using spatially aware deep attention models," *arXiv preprint arXiv:1705.09436*, 2017.

[22] N. Lee, W. Choi, P. Vernaza, C. B. Choy, P. H. Torr, and M. Chandraker, "Desire: Distant future prediction in dynamic scenes with interacting agents," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 336–345, 2017.

[23] J. Dequaire, P. Ondrúška, D. Rao, D. Wang, and I. Posner, "Deep tracking in the wild: End-to-end tracking using recurrent neural networks," *The International Journal of Robotics Research*, vol. 37, no. 4-5, pp. 492–512, 2018.

[24] N. Djuric, V. Radosavljevic, H. Cui, T. Nguyen, F.-C. Chou, T.-H. Lin, N. Singh, and J. Schneider, "Motion prediction of traffic actors for autonomous driving using deep convolutional networks," in *arXiv preprint arXiv:1808.05819, 2*.

[25] S. Becker, R. Hug, W. Hübner, and M. Arens, "An evaluation of trajectory prediction approaches and notes on the trajnet benchmark," *arXiv preprint arXiv:1805.07663*, 2018.

[26] S. H. Park, B. Kim, C. M. Kang, C. C. Chung, and J. W. Choi, "Sequence-to-sequence prediction of vehicle trajectory via lstm encoder-decoder architecture," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1672–1678, IEEE, 2018.

[27] N. Deo and M. M. Trivedi, "Convolutional social pooling for vehicle trajectory prediction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1468–1476, 2018.

[28] G. Habibi, N. Jaipuria, and J. P. How, "Context-aware pedestrian motion prediction in urban intersections," *arXiv preprint arXiv:1806.09453*, 2018.

[29] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, "Social gan: Socially acceptable trajectories with generative adversarial networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2255–2264, 2018.

[30] M. Pfeiffer, G. Paolo, H. Sommer, J. Nieto, R. Siegwart, and C. Cadena, "A data-driven model for interaction-aware pedestrian motion prediction in object cluttered environments," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1–8, IEEE, 2018.

[31] H. Manh and G. Alaghband, "Scene-lstm: A model for human trajectory prediction," *arXiv preprint arXiv:1808.04018*, 2018.

[32] N. Nikhil and B. Tran Morris, "Convolutional neural network for trajectory prediction," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 0–0, 2018.

[33] N. Radwan, W. Burgard, and A. Valada, "Multimodal interaction-aware motion prediction for autonomous street crossing," *The International Journal of Robotics Research*, p. 0278364920961809, 2018.

[34] W. Luo, B. Yang, and R. Urtasun, "Fast and furious: Real time end-to-end 3d detection, tracking and motion forecasting with a single convolutional net," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 3569–3577, 2018.

[35] E. Rehder, F. Wirth, M. Lauer, and C. Stiller, "Pedestrian prediction by planning using deep neural networks," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1–5, IEEE, 2018.

[36] S. Hoermann, M. Bach, and K. Dietmayer, "Dynamic occupancy grid prediction for urban autonomous driving: A deep learning approach with fully automatic labeling," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2056–2063, IEEE, 2018.

[37] H. Cui, V. Radosavljevic, F.-C. Chou, T.-H. Lin, T. Nguyen, T.-K. Huang, J. Schneider, and N. Djuric, "Multimodal trajectory predictions for autonomous driving using deep convolutional networks," in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 2090–2096, IEEE, 2019.

[38] X. Huang, S. G. McGill, B. C. Williams, L. Fletcher, and G. Rosman, "Uncertainty-aware driver trajectory prediction at urban intersections," in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 9718–9724, IEEE, 2019.

[39] C. Tang and R. R. Salakhutdinov, "Multiple futures prediction," in *Advances in Neural Information Processing Systems*, pp. 15424–15434, 2019.

[40] Y. Ma, X. Zhu, S. Zhang, R. Yang, W. Wang, and D. Manocha, "Trafficpredict: Trajectory prediction for heterogeneous traffic-agents," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 6120–6127, 2019.

[41] A. Sadeghian, V. Kosaraju, A. Sadeghian, N. Hirose, H. Rezatofighi, and S. Savarese, "Sophie: An attentive gan for predicting paths compliant to social and physical constraints," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1349–1358, 2019.

[42] R. Chandra, U. Bhattacharya, A. Bera, and D. Manocha, "Traphic: Trajectory prediction in dense and heterogeneous traffic using weighted interactions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8483–8492, 2019.

[43] H. Xiong, F. B. Flohr, S. Wang, B. Wang, J. Wang, and K. Li, "Recurrent neural network architectures for vulnerable road user trajectory prediction," in *2019 IEEE Intelligent Vehicles Symposium (IV)*, pp. 171–178, IEEE, 2019.

[44] K. Saleh, M. Hossny, and S. Nahavandi, "Contextual recurrent predictive model for long-term intent prediction of vulnerable road users," *IEEE Transactions on Intelligent Transportation Systems*, 2019.

[45] M. Itkina, K. Driggs-Campbell, and M. J. Kochenderfer, "Dynamic environment prediction in urban scenes using recurrent representation learning," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pp. 2052–2059, IEEE, 2019.

[46] M. Schreiber, S. Hoermann, and K. Dietmayer, "Long-term occupancy grid prediction using recurrent neural networks," in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 9299–9305, IEEE, 2019.

[47] N. Mohajerin and M. Rohani, "Multi-step prediction of occupancy grid maps with recurrent neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10600–10608, 2019.

[48] S. Hörmann, *Long-Term Prediction using Grid Based Environment Models for Urban Autonomous Driving*. PhD thesis, Universität Ulm, 2020.

[49] A. Das, E. S. Kolvig-Raun, and M. B. Kjærgaard, "Accurate trajectory prediction in a smart building using recurrent neural networks," in *Adjunct Proceedings of the 2020 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2020 ACM International Symposium on Wearable Computers*, pp. 619–628, 2020.

[50] L. L. Li, B. Yang, M. Liang, W. Zeng, M. Ren, S. Segal, and R. Urtasun, "End-to-end contextual perception and prediction with interaction transformer," *arXiv preprint arXiv:2008.05927*, 2020.

[51] B. Brito, H. Zhu, W. Pan, and J. Alonso-Mora, "Social-vrnn: One-shot multi-modal trajectory prediction for interacting pedestrians," *arXiv preprint arXiv:2010.09056*, 2020.

[52] B. Lange, M. Itkina, and M. J. Kochenderfer, "Attention augmented convlstm forenvironment prediction," *arXiv preprint arXiv:2010.09662*, 2020.

[53] M. Toyungyernsub, M. Itkina, R. Senanayake, and M. J. Kochenderfer, "Double-prong convlstm for spatiotemporal occupancy prediction in dynamic environments," *arXiv preprint arXiv:2011.09045*, 2020.

[54] P. Wu, S. Chen, and D. N. Metaxas, "Motionnet: Joint perception and motion prediction for autonomous driving based on bird's eye view maps," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11385–11395, 2020.

[55] M. Á. S. UAH, C. Salinas, and J. A. UAH, "D4.4 model for the prediction of vrus intentions," 2020.

[56] H. Moravec and A. Elfes, "High resolution maps from wide angle sonar," in *Proceedings. 1985 IEEE international conference on robotics and automation*, vol. 2, pp. 116–121, IEEE, 1985.

[57] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," *Computer*, vol. 22, no. 6, pp. 46–57, 1989.

[58] A. Elfes *et al.*, "Occupancy grids: A stochastic spatial representation for active robot perception," in *Proceedings of the Sixth Conference on Uncertainty in AI*, vol. 2929, p. 6, Morgan Kaufmann, 1990.

[59] J. Moras, V. Cherfaoui, and P. Bonnifait, "Evidential grids information management in dynamic environments," in *17th International Conference on Information Fusion (FUSION)*, pp. 1–7, IEEE, 2014.

[60] S. Thrun, "Learning occupancy grid maps with forward sensor models," *Autonomous robots*, vol. 15, no. 2, pp. 111–127, 2003.

[61] J. Carvalho and R. Ventura, "Comparative evaluation of occupancy grid mapping methods using sonar sensors," in *Iberian Conference on Pattern Recognition and Image Analysis*, pp. 889–896, Springer, 2013.

[62] University of Pennsylvania, "Robotics: Estimation and learning. 3.2.1. occupancy grid map.." https://www.coursera.org/lecture/robotics-learning/3-2-1-occupancy-grid-map-0QuFW?redirectTo=%2Flearn%2Frobotics-learning%3Faction%3Denroll, 2021.

[63] N. Chebrolu and C. Stachniss, "Msr course - 03 occupancy grid mapping with known poses (chebrolu)." https://youtu.be/x_Ah685BFEQ?t=3204, 2020.

[64] V. Dhiman, A. Kundu, F. Dellaert, and J. J. Corso, "Modern map inference methods for accurate and fast occupancy grid mapping on higher order factor graphs," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2037–2044, IEEE, 2014.

[65] W. Liu, *Propositional, Probabilistic and Evidential Reasoning: Integrating numerical and symbolic approaches*, vol. 77. Springer Science & Business Media, 2001.

[66] A. Dempster, "Upper and lower probabilities induced by a multivalued mapping. annals of mathematical statistics. vol. 38. p. 325-339," 1967.

[67] G. Shafer, *A mathematical theory of evidence*, vol. 42. Princeton university press, 1976.

[68] G. Shafer, "Dempster-shafer theory," *Encyclopedia of artificial intelligence*, vol. 1, pp. 330–331, 1992.

[69] M. Ribo and A. Pinz, "A comparison of three uncertainty calculi for building sonar-based occupancy grids," *Robotics and autonomous systems*, vol. 35, no. 3-4, pp. 201–209, 2001.

[70] G. Oriolo, G. Ulivi, and M. Vendittelli, "Fuzzy maps: a new tool for mobile robot perception and planning," *Journal of Robotic Systems*, vol. 14, no. 3, pp. 179–197, 1997.

[71] S. B. Thrun, "Exploration and model building in mobile robot domains," in *IEEE international conference on neural networks*, pp. 175–180, IEEE, 1993.

[72] T. Collins and J. Collins, "Occupancy grid mapping: An empirical evaluation," in *2007 mediterranean conference on control & automation*, pp. 1–6, IEEE, 2007.

[73] R. van Kempen, B. Lampe, T. Woopen, and L. Eckstein, "A simulation-based end-to-end learning framework for evidential occupancy grid mapping," *arXiv preprint arXiv:2102.12718*, 2021.

[74] J. Degerman, T. Pernstål, and K. Alenljung, "3d occupancy grid mapping using statistical radar models," in *2016 IEEE Intelligent Vehicles Symposium (IV)*, pp. 902–908, IEEE, 2016.

[75] C. Lu, M. J. G. van de Molengraft, and G. Dubbelman, "Monocular semantic occupancy grid mapping with convolutional variational encoder–decoder networks," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 445–452, 2019.

[76] D. Nuss, S. Reuter, M. Thom, T. Yuan, G. Krehl, M. Maile, A. Gern, and K. Dietmayer, "A random finite set approach for dynamic occupancy grid maps with real-time application," *The International Journal of Robotics Research*, vol. 37, no. 8, pp. 841–866, 2018.

[77] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3213–3223, 2016.

[78] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[79] R. Danescu, F. Oniga, and S. Nedevschi, "Modeling and tracking the driving environment with a particle-based occupancy grid," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 4, pp. 1331–1342, 2011.

[80] B. Liu, E. Adeli, Z. Cao, K.-H. Lee, A. Shenoi, A. Gaidon, and J. C. Niebles, "Spatiotemporal relationship reasoning for pedestrian intent prediction," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3485–3492, 2020.

[81] M.-F. Chang, J. Lambert, P. Sangkloy, J. Singh, S. Bak, A. Hartnett, D. Wang, P. Carr, S. Lucey, D. Ramanan, *et al.*, "Argoverse: 3d tracking and forecasting with rich maps," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8748–8757, 2019.

[82] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 Year, 1000km: The Oxford RobotCar Dataset," *The International Journal of Robotics Research (IJRR)*, vol. 36, no. 1, pp. 3–15, 2017.

[83] M. Braun, S. Krebs, and D. M. Gavrila, "Ecp2. 5d-person localization in traffic scenes," in *2020 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1694–1701, IEEE, 2020.

[84] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, and T. Darrell, "Bdd100k: A diverse driving dataset for heterogeneous multitask learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2636–2645, 2020.

[85] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3354–3361, IEEE, 2012.

[86] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11621–11631, 2020.

[87] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine, *et al.*, "Scalability in perception for autonomous driving: Waymo open dataset," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2446–2454, 2020.

[88] X. Huang, P. Wang, X. Cheng, D. Zhou, Q. Geng, and R. Yang, "The apolloscape open dataset for autonomous driving and its application," *IEEE transactions on pattern analysis and machine intelligence*, vol. 42, no. 10, pp. 2702–2719, 2019.

[89] T. Roddick and R. Cipolla, "Predicting semantic map representations from images using pyramid occupancy networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11138–11147, 2020.

[90] A. Loukkal, Y. Grandvalet, T. Drummond, and Y. Li, "Driving among flatmobiles: Bird-eye-view occupancy grids from a monocular camera for holistic trajectory planning," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 51–60, 2021.

[91] T. Hehn, J. F. Kooij, and D. M. Gavrila, "Fast and compact image segmentation using instance stixels," *IEEE Transactions on Intelligent Vehicles*, 2021.

[92] L. Wang, B. Goldluecke, and C. Anklam, "L2r gan: Lidar-to-radar translation," in *Proceedings of the Asian Conference on Computer Vision*, 2020.

[93] A. Birk and S. Carpin, "Merging occupancy grid maps from multiple robots," *Proceedings of the IEEE*, vol. 94, no. 7, pp. 1384–1397, 2006.

[94] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[95] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[96] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," *arXiv preprint arXiv:1409.1259*, 2014.

[97] C. Lea, M. D. Flynn, R. Vidal, A. Reiter, and G. D. Hager, "Temporal convolutional networks for action segmentation and detection," in *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 156–165, 2017.

[98] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241, Springer, 2015.

[99] J. Lin, C. Gan, and S. Han, "Tsm: Temporal shift module for efficient video understanding," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 7083–7093, 2019.

[100] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *arXiv preprint arXiv:1706.03762*, 2017.

[101] W. Lotter, G. Kreiman, and D. Cox, "Deep predictive coding networks for video prediction and unsupervised learning," *arXiv preprint arXiv:1605.08104*, 2016.

[102] D. Nuss, M. Thom, A. Danzer, and K. Dietmayer, "Fusion of laser and monocular camera data in object grid maps for vehicle environment perception," in *17th International Conference on Information Fusion (FUSION)*, pp. 1–8, IEEE, 2014.

[103] A. P. Dempster, "A generalization of bayesian inference," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 30, no. 2, pp. 205–232, 1968.