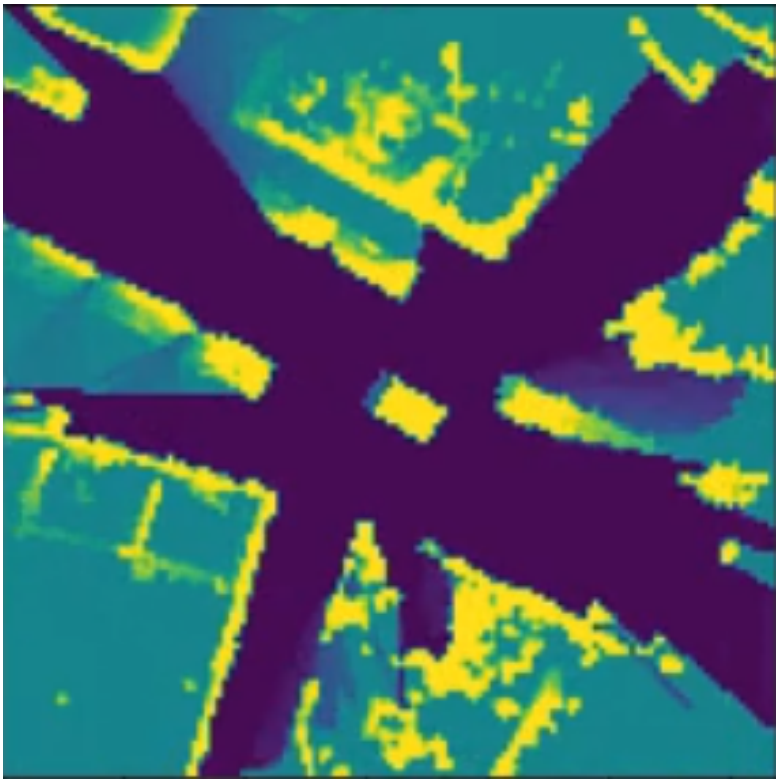
Literature Review on Motion Prediction of Vulnerable Road Users using Occupancy Grid Maps

Rutger Dirks

June 8, 2021



3ME Department of Cognitive Robotics - Intelligent Vehicles

Delft University of Technology

Supervisors: Ewoud Pool, Hidde Boekema, and Dariu Gavrila

Acronyms

AAConvLSTM Attention Augmented Convolutional Long Short-Term Memory (ConvLSTM). 32, 43

AV Autonomous Vehicle. 4, 5, 6, 8, 9, 12, 18, 19, 27, 29, 30, 36, 38, 40, 41

BBA Basic Belief Assignment. 13, 14

BDD100K Berkely DeepDrive 100K. 21

BEV bird's-eye view. 4, 10, 19, 20, 24, 37

CNN Convolutional Neural Network. 33, 34, 41

ConvLSTM Convolutional Long Short-Term Memory. 1, 31, 32, 33, 34, 36, 37

DOGMa Dynamic Occupancy Grid Map. 2, 31, 32, 33, 34, 37, 38, 39, 40

DST Dempster-Shafer Theory. 2, 10, 13, 14, 18, 31, 32, 40, 42, 43

ECP2.5D Eurocity Persons 2.5D. 20, 21

FOD Frame of Discernment. 13

FPR False Positive Rate. 29, 34

GRU Gated Recurrent Unit. 35, 36

IS Image Similarity. 2, 27, 29, 30, 32, 33, 38, 40, 43

KITTI Karlsruhe Institute of Technology and Toyota Technological Institute. 21, 22, 31, 32, 36

LSTM Long Short-Term Memory. 31, 32, 35

MeSE Median Squared Error. 37

MFE Motion-Flow Extraction. 36

MSE Mean Squared Error. 2, 27, 28, 29, 30, 32, 33, 34, 35, 37, 38, 39, 43

NN Artificial Neural Network. 2, 15, 18

OGM Occupancy Grid Map. 2, 3, 4, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43

PPV Positive Predictive Value. 29

RNN Recurrent Neural Network. 31, 35, 36, 41

ROC Receiver Operating Characteristic. 2, 27, 29, 30, 34, 35, 38, 39

SSIM Structural Similarity Index Measure. 2, 27, 28, 29, 30, 36, 38, 40, 41, 43

STC Spatio-Temporal Convolution. 37

STIP Stanford-TRI Intent Prediction. 20, 21, 22

STM Spatial Transformer. 35

TNR True Negative Rate. 36

TPR True Positive Rate. 29, 34, 36, 38

VRU Vulnerable Road User. 4, 5, 6, 8, 19, 20, 24

Contents

1 Introduction

In 2017, the European Union counted around 25300 fatalities on the EU roads, of which 46% were vulnerable road users (e.g. pedestrians and (motor-)cyclists). 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 the motion patterns of VRUs are complex and mutable depending on the static and dynamic obstacles they encounter [8]. Therefore, a motion prediction method is needed that can also accurately predict the motion of VRUs.

Currently, there are some promising methods to predict the motion of the entire AV's environment (including VRUs). These methods make use of an Occupancy Grid Map (OGM) to capture the AV's environmental context. An OGM is a rasterized bird's-eye view (BEV) map corresponding to the AV's environment that contains occupancy information in each grid cell of the raster. Given the AV's environmental sensor data (e.g. from a LiDAR, camera, or radar), each grid cell's state in the OGM is updated to a new occupancy value (Empty, Occupied, or Uncertain). Deep learning can be used to predict future OGMs sequences of an AV's environment, given past OGM sequences. Since an OGM's form is similar to an image - a matrix containing values between 0 (Empty) and 1 (Occupied) - convolutional neural networks can be used to efficiently extract useful features from it [9]. Then, a sequential neural network can be used to process a sequence of the extracted features to capture the past states and behavior of the AV's environment. The information of the past is then used by the neural network to generate OGM predictions that describe the future environment of the AV.

This literature study gives an overview of what the current research areas and challenges for the future are regarding motion prediction using OGMs. The conclusion of this study forms the basis for the Master thesis research proposal at the end of this literature review.

In the following sections of this introduction, first, the problem of motion prediction is described. Then, current solutions to motion prediction challenges and their limitations are briefly discussed. After discussing the challenges and limitations of current solutions, it is explained why OGM prediction methods are a promising all-encompassing solution to those current challenges. Then, one main research question and four sub-questions are posed to investigate the stated motion prediction problem regarding OGM prediction methods. The following chapters provide information from literature to answer the main and sub-questions posed in this introduction. 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 experience, future states (trajectories) of one or more actors (i.e. traffic participants such as cars, cyclists, and VRUs) 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.

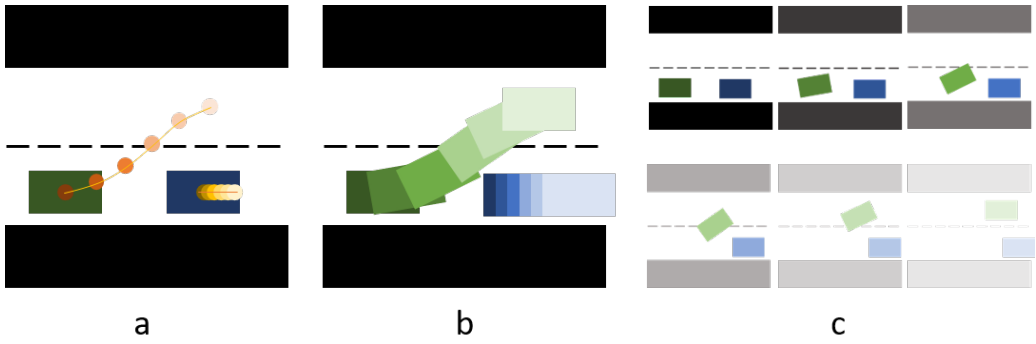


Figure 1: This image shows the three prediction forms encountered in the literature. The black and white parts represent a road. The green and blue squares are representations of vehicles. The predictions are shown using a more fading color the further the prediction goes into the future. Figure a shows 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.

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 instantaneous, respectively. However, this is not possible in practice. Therefore, I think a suitable prediction method should predict further into the future, be more accurate, have more certainty, and be 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.

1.2 Current Solutions

Over the past decade, much research has been done to find methods that solve the previously mentioned motion prediction problem. Table 9 in Appendix A 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 [10], [8], [11]. Therefore, VRU behavior 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 [12]. Compared to other road users, VRUs are highly maneuverable which makes it challenging to predict their behavior,

especially for the long-term [13], [10], [10]. Most prediction models assume that VRU behavior is intention-driven and that those intentions can be inferred from certain cues, other than dynamics [10]. Therefore, detecting cues related to VRU behavioral changes is important for achieving accurate long-term predictions [14]. 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], [15], [16], [17]. Especially to anticipate critical situations, taking into account the environmental context is expected to increase the prediction accuracy [11]. An example of how the use of traffic light information influences the predictions is shown in figure 2. 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. This way, it can be used as evidence to enable inferences about an actor's future trajectory [18]. [14] 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.

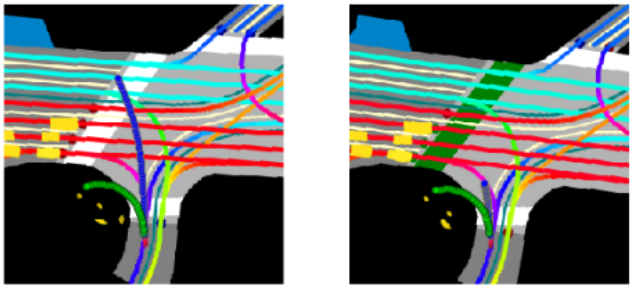


Figure 2: This image shows an example of how the environmental context can be used to influence the predictions. On the left, the prediction (the blue trajectory) of the bicyclist (red) is shown when the traffic light in the scene is green. On the right, the bicyclist's prediction is shown when the traffic light turned red. The green trajectory is the ground truth. [8]

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 [16], [17]. Figure 3 shows how [16]'s prediction method takes into account social context. 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 [15]. 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" [11].

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 [19]. 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 [20], [21]. 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

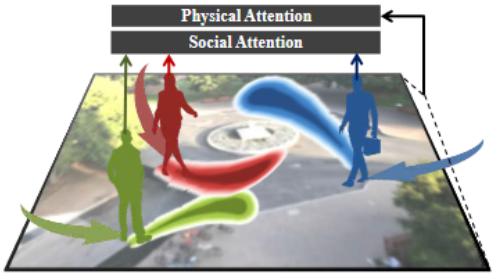


Figure 3: This image shows a schematic of how [16] uses the social context, and the scene context, to predict the behavior of the pedestrians (the green, red, and blue surfaces). The pedestrians are predicted to avoid each other while continuing in the general direction they walked in the past (the arrows). [16]'s method predicts physically and socially plausible trajectories.

training [21], or the modes could be learned. Because probability estimations are needed to make multimodal predictions, these methods are often researched together.

[22] 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. [14] 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.
[10], [20], [23] 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. [19]'s research adds a confidence estimator to their multimodal prediction method that estimates the confidence of multiple predictors (See figure 4). The predictor with the highest confidence value is considered for the trajectory prediction.

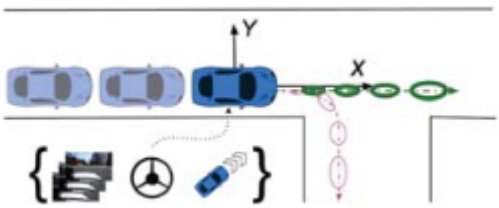


Figure 4: This image shows a schematic of how multimodal predictions can be made, given a past trajectory. Since the road allows for two directions the vehicle can move in, two predictions are made. One prediction in green, going straight. The other in red, going to the right. The prediction that goes straight forward is given the highest confidence value (thick green), which means that that trajectory is considered for the prediction. [19]

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.
[23] 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.

1.2.6 The promising solution

The state-of-the-art research in the previous paragraphs show 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 convolutional neural 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 [24] shows that it can take 60ms to predict OGMs 1s into the future), making these methods, using OGMs as environment representation, a promising one for motion prediction research. All in all, the motion prediction methods that use OGMs can provide solution to all challenges stated in the previous paragraphs. That is why OGM prediction is a promising solution that is worth investigating, which leads to the following section about this literature reviews research questions.

1.3 Research Questions

Because of the potential of OGMs, this literature review focuses on the prediction of OGMs using Deep Learning. Currently, state-of-the-art research regarding motion prediction in traffic scenes makes use of OGMs. To give an overview of these OGM prediction methods and to find out which method provides the best predicitons, the following main research question is asked: *"What is the best method to perform traffic scene OGM prediction using Deep Learning?"*

In this review, several sub-questions are asked in order to answer the main question. The reasoning behind the sub-questions, followed by the sub-questions is described below.

First, it is important to know what an OGM is and how they can be generated from an AV's sensor data. Research shows that OGMs can be generated using several methods [25] [26] [27]. Besides several generation methods that compute occupancy information for each OGM's grid cell, the OGMs can also be extended with other information such as semantics [28], dynamics [29], and 3D information [30]. Combinations of the different generation methods and data extensions will result in different OGM forms. This leads to the first sub-question: *What OGM form is most suitable to use in OGM prediction methods?*

Second, to generate the OGMs, a dataset should be used that contains sequences of traffic scene sensor data from an AV's point of view. It is important that occupancy information can be extracted from the data. Additionally, it is a perk if the datasets also contains 3D, semantic, or dynamic information about the traffic scenes for the extended OGM forms. There exist several datasets (view table 4) that contain this kind of data. To find out which of these datasets is the best to use for OGM generation and prediction, the second sub-question is asked: *What dataset is most suitable to generate OGM sequences for OGM prediction?*

Third, to determine which OGM prediction method performs best, the quality of the OGM predictions need to be evaluated per method so the different methods can be compared. When evaluating, the OGM predictions will be compared to the ground truth OGMs. The more the predictions are similar to the ground truth, the better that prediction method performs. Evaluation can be done by visually comparing the predictions to ground truth OGMs [26]. However, this evaluation method is subjective and thus can result in unreliable performance scores. Therefore, [25] summarizes some quantitative evaluation methods that use image similarity principles to evaluate OGMs. Quantitative metrics can provide

an objective means for performance evaluation. However, each metric measures a different aspect of the OGM. It is therefore important that a metric is chosen that best evaluates the similarity accuracy between OGMs. This leads to the third sub-question: *What is the best quantitative metric to determine the accuracy of a predicted OGM?*

Fourth, there are several state-of-the-art deep learning methods that perform OGM prediction. To find out which of those methods are the best ones, they must be evaluated and compared with each other. To answer the main questions, this fourth sub-question is asked: *What Deep Learning method generates 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 sub-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 sub-question. The third sub-question will be answered in chapter 4, in which several metrics to determine the quality of OGMs are compared according to some criteria that are required for accurate and safe (related to AV's) error estimations of OGM predictions. The fourth sub-question will be answered in chapter 5, in which an overview of OGM prediction methods using deep learning is provided. Chapter 6 contains a conclusion in which the main research question of this literature review is answered. Then, chapter 7 discusses some topics for future work that result from this literature study. 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 [31] investigated 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 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 5. This is an example of environment representation called an Occupancy Grid Map (OGM).

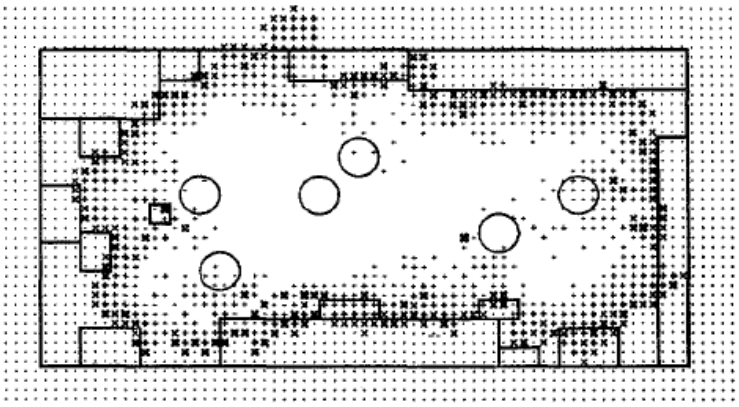


Figure 5: This image shows an OGM of a robot's surrounding environment. The solid lines represent the outline of the room and of the major objects that formed the robot's environment. The large circles in the image are the positions at which the robot performed measurements of its surroundings. The measurements 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 [31].

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 [32]. 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. [33] [34]. The two main sensor models that are used to interpret the sensor data are the inverse sensor model [33] and the forward sensor model [27]. 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). [35].

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

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. [36] [35].

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

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 [35].

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

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

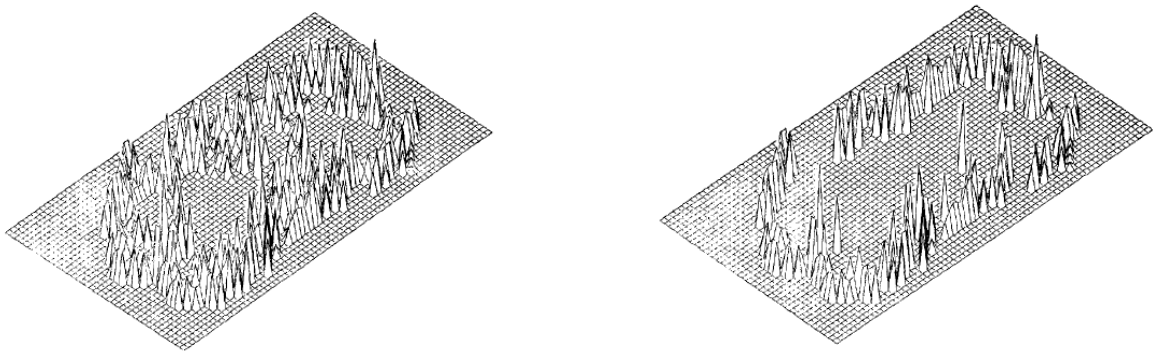
By assigning the estimated OGM as the prior estimation for the next time step and given equations 3 and 5, Bayes's theorem can be used for every grid cell $m_{i,j}$ to update the OGM m recursively every time new sensor data becomes available. This is shown in equation 6 [37].

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

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 6b), a two-dimensional grid map can be created that can be used for purposes such as motion planning and environment interpretation (Figure 5).

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 $z_{1:T}$ [35].



(a) This figure shows the occupied areas in the map where the height of the peaks shows the occupancy certainty factors.[31]

(b) This figure shows the occupied areas in the map after thresholding the occupancy certainty factors.[31]

Figure 6: 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. [27] [35].

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 [27] [35]

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

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 [27]. 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 [38].

2.1.3 Comparison of the sensor models

In this subsection, the inverse and forward sensor models are compared. Figure 7 shows the difference between the inverse sensor model and the forward sensor model, which can be compared to the ground truth.

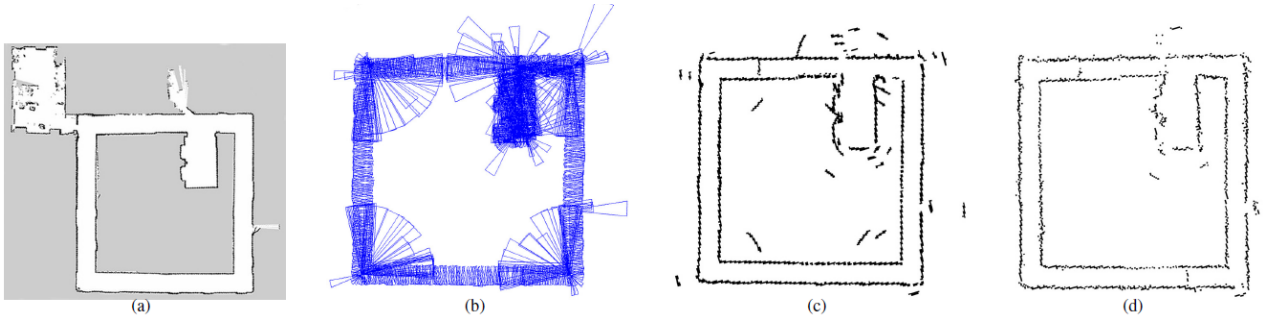


Figure 7: The experimental results from Carvalho's research [35] which compares the inverse sensor model with the forward model. Figure (a) is the ground truth map. Figure (b) are the taken measurements. Figure (c) is the computed OGM using the inverse sensor model. Figure (d) is the computed OGM using the forward sensor model.

The forward sensor model is more accurate but also requires much more computational effort [35]. 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 [39]. The Dempster-Shafer Theory (DST) [40] [41], 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 [34]. 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. [34]. 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). [34].

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

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

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

An example of a BBA mass function is shown in table 1, 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. [42]. Subsequently, the belief $Bel()$ and the plausibility $Pl()$ of the hypotheses are computed, giving the lower and upper probability bounds for the hypotheses.

Table 1: 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. [34].

$$M_1 \oplus M_2(A) = \left\{ \begin{array}{ll} \frac{M_{1\cap 2}(A)}{1-M_{1\cap 2}(\emptyset)} & A \neq \emptyset \\ 0 & A = \emptyset \end{array} \right\} \tag{13}$$

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

$$M_{1\cap 2}(A) = \sum_{B,C \in 2^{\Omega} | A = B \cap C} M_1(B) \cdot M_2(C) \tag{14}$$

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 $||$.

$$P(A) = \sum_{B \in 2^{\Omega}} M(B) \cdot \frac{|A \cap B|}{|B|} \tag{15}$$

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. [34]. The processing time of the DST method is about 1.5 times higher than that of the inverse sensor model probabilistic method [26]. Figure 8 shows the qualitative differences between the probabilistic approach (center) and the DST approach (right).

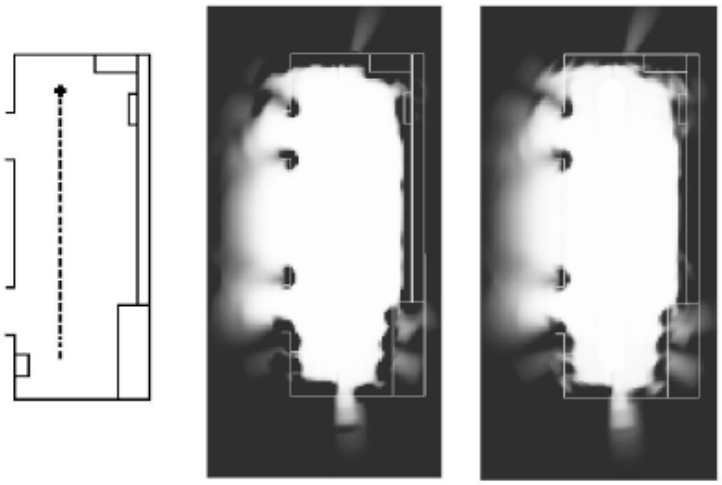


Figure 8: Ribo and Pinz [26] compare the probabilistic method (center) with the DST method (right) to generate OGMs. The ground truth schematic is shown left.

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 [43] 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 [26] 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 [44] 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 [25] 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 [45] 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 inspection) [26]. Collins [25] 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 evaluate OGMs will be discussed in section 4.

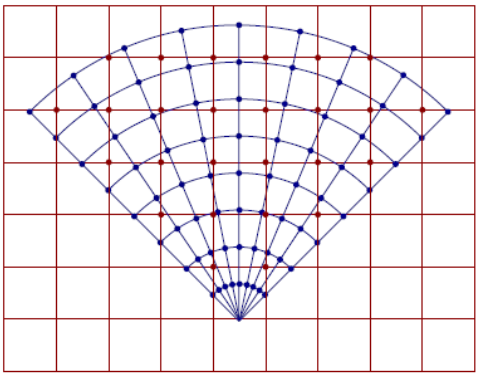
Besides ways to generate the traditional OGM maps that provide information about the cell occupancy, there are extended OGM forms that contain more information. The next subsections will elaborate on some of those extended forms.

2.4 Occupancy Grid Map extended forms

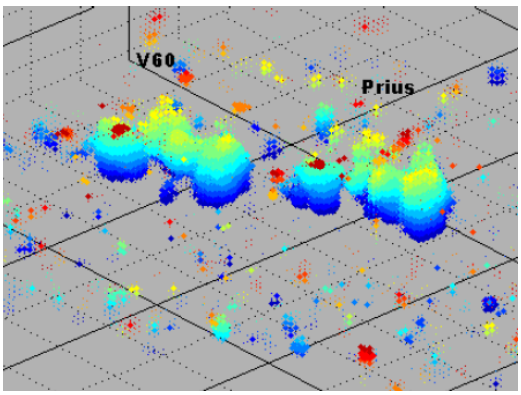
Many variations of the OGM have been invented and researched that represent environments in more complex methods, including for example 3D information [30], semantic segmentation of the cells [28], or dynamic information of the occupied regions [29].

2.4.1 3D Occupancy Grid Mapping

Degerman [30] 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 9a, 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 9b.



(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. [30]



(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). [30]

2.4.2 Semantic Occupancy Grid Mapping

While classical OGMs only contain information about occupancy, Lu [28] 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 [46] and KITTI [47] datasets and the results show that the network is robust to sparse input data and a weak ground-truth. Figure 10 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 [48] 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

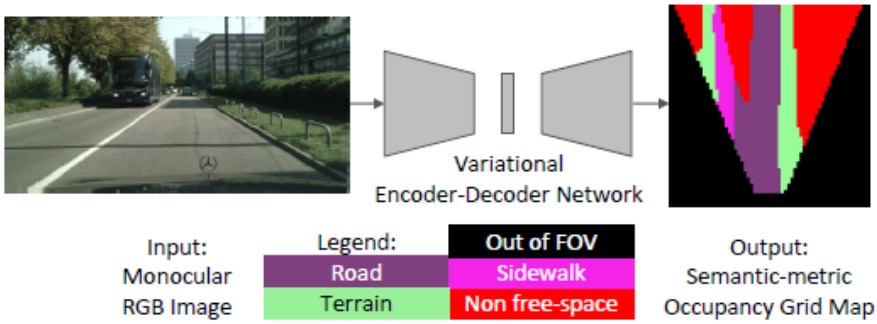


Figure 10: An illustration of the semantic OGM method that takes a monocular RGB image as input and outputs a semantic OGM [28].

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 [29] 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 11.

Figure 11: 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. [29].

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 [49] 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 2 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 best 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 2: 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 using deep learning networks, it is important to choose a dataset (or multiple datasets) that optimally fulfills the requirements for its purpose. The dataset will be used to train and evaluate the deep learning network on. So, if the datasets is faulty, the resulting model will be faulty as well. A faulty model will predict faulty OGMs, which could cause dangerous situations if it is used by an AV and affects its decisions regarding motion planning. This chapter first discusses what the requirements are for a dataset that is used for OGM prediction. Then, based on those requirements, nine criteria are devised that are used to evaluate the adequacy of a dataset. Subsequently, ten datasets are evaluated against the criteria and compared with one another. A dataset top 3 is presented in this chapter's conclusion in which the question "*What dataset is most suitable to generate OGM sequences for OGM prediction?*" is answered.

In the case of OGM prediction in a traffic scene context, a dataset is desired that contains BEV environmental data of ego-vehicle centered traffic scenes that can be used to generate OGM sequences. Moreover, if the dataset contains data that can be used to generate an extended form of the OGM, as is discussed in chapter 2, it would be beneficial.

It is important that the dataset's traffic scenes reflect the diversity of real world situations. The way a dataset's data is sampled from a population can influence how much the dataset reflects reality. A model that is trained on biased data could generate inaccurate predictions [50]. If a traffic scene dataset is obtained in only one geographic, environmental, or cultural location, the dataset will contain some kind of bias towards the behavior, representation, and environmental context of that location [50]. For example, [51]'s research shows that geographical location (rural vs urban), and factors such as age, gender, and education influence the behavior of drivers. Also, [52] shows that traffic risk perception differs per culture and country. This can result in different traffic behavior between countries given the same traffic scene context. It is therefore important that a dataset contains data from multiple locations in varying geographical, cultural, and environmental areas. Also, the dataset should be diverse in the kind of traffic participants it contains. Besides vehicles and pedestrians, it is desired that the dataset contains other traffic participant classes (e.g. cyclists, riders). The more diverse the dataset is, the smaller the dataset bias is, and the better the prediction accuracy is of the model that is trained on that dataset. Besides having diverse data, the performance of a deep learning network also increases with a logarithmically increasing dataset size [53]. Therefore, a dataset should be large besides being diverse. For large-scale datasets, even rare traffic cases are expected to be captured enough times so the deep learning network can generalize those situations accurately.

Regarding the traffic scene sequences, literature does not mention the ideal frame rate that OGM sequences should have to generate optimal predictions, [54] reviewed literature on what the influence of frame rate of visual perceptions is on human perceptual performance. In [54]'s reviewed papers, humans are situated in virtual environments in which the visual information of the environment is updated at a variable frame rate. The research shows that higher frame rates increase the performance (accuracy, reaction time, recognition) at which humans completed certain perceptual tasks. Frame rates below 10 Hz showed sharp performance degradations, while frame rates higher than 15 Hz did not increase performance significantly. Therefore, a frame rate of 10 Hz is considered the minimum threshold for accurate human perceptual understanding. Using humans as reference for the performance of deep learning networks, a frame rate of 10 Hz is also considered as the minimal threshold for accurate OGM prediction. So, it is desired that a dataset has at least a 10 Hz sampling rate of its environment. Furthermore, the traffic scene sequences should be long enough. The sequences have to be split into a part that represents the 'past' to use as input sequences for the deep learning network, and a ground truth of the 'future' to compare the predictions with. The research by [55] uses 0.5s of past OGMs to predict up to 1.5s of future OGMs. So, a sequence of 2s is sufficient to perform OGM prediction using this method. If a margin of 0.5s is considered for the length of the past information and for the length of the prediction horizon, a dataset should provide OGM sequences of at least 3s long.

Aside from the contents of the dataset, the quality of the sensors with which the dataset is recorded is essential. The resolutions of the sensors should be high enough to capture the smallest traffic participant's (VRUs) behavioral patterns. 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 a dataset has to meet in order to be suitable for OGM prediction. The criteria are based on the 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 rate that is suitable for capturing road user behavior and trajectories.
5. The sequences contain various traffic actors including vehicles and 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. The dataset provides data, including ground truth, that is required for generating extended OGM forms.
9. Results of other research using the dataset for generating and predicting OGMs is available for comparison.

A total of ten datasets coming from three categories are found that contain ego-vehicle centered traffic scene sequences. Three datasets were created for a purpose that includes motion prediction. Five datasets were created for object detection purposes. The last two datasets were mainly created to research semantic segmentation on. All the datasets are recorded using a vehicle equipped with one or multiple sensors that drives through real traffic.

To evaluate the datasets against the criteria, information about each dataset is gathered and put into a table (see table 4). The first criterion is met if the dataset contains 2D BEV information about the environment. From the recording vehicle's point of view, having 2D BEV information means that the vehicle should gather 3D sensor data which can be projected onto the 2D top view plane. The information about the sensors provides the data to evaluate this first criterion. The second criterion is tested using information about the number of categories and classes that are labelled in the datasets. If the dataset contains sequences of its sensor data, the third criterion is met. The fourth criterion is tested by evaluating the sampling frequency and length of the sequences. To test the fifth criterion, the number (and presence) of vehicles and pedestrians is evaluated. The information about the sensors is used to test the sixth criterion. The seventh criterion is evaluated by examining the diversity of sampling locations. Then, the eighth criterion is met if the dataset contains data to generate extended OGMs. Finally, the ninth criterion is tested by evaluating whether there exists literature that generates OGMs from the dataset and whether there is literature that uses those generated OGMs for OGM prediction. The evaluation and comparison of each dataset against the criteria is shown in table 3.

The most interesting characteristics of each dataset are highlighted below in three subsections based on the dataset's original purposes: Motion prediction (section 3.1), Object Detection (section 3.2), and Semantic Segmentation (section 3.3). At the end of this chapter (section 3.4) a conclusion about the datasets is provided.

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 [56], the Argoverse dataset [57] and the RoboCar dataset [58].

The STIP dataset [56] is obtained using three cameras (left, front and right) with a 1216x1936 resolution at 20 Hz positioned on the recording vehicle. An example of the STIP dataset's camera images is shown in figure 12. However, these are not stereo cameras, so no 3D information is obtained. This dataset does therefore not meet the first criterion. 3D information can be computed from the monocular images using monocular unsupervised depth estimation techniques [59]. However, this probably will not yield as reliable results compared to stereo vision data and the time spent on generating 3D data could also be spent on training a network using a dataset that does contain 2D BEV information. A remarkable property of this dataset is that the sampling frequency of the camera images is 20 Hz. When comparing the STIP dataset's sampling rate to the other dataset's rates (see table 4, it can be seen that this dataset, together with the Eurocity Persons 2.5D (ECP2.5D) [60] dataset has the highest sampling rate.

The Argoverse dataset [57] is recorded in only two cities in one country. It therefore contains little geographic diversity compared to the ECP2.5D [60] and Cityscapes [46] datasets. On the other side, the Argoverse dataset [57] contains 333K 5-second sequences and labels 17 different classes, including vehicles, pedestrians and cyclists. Furthermore, [61] has

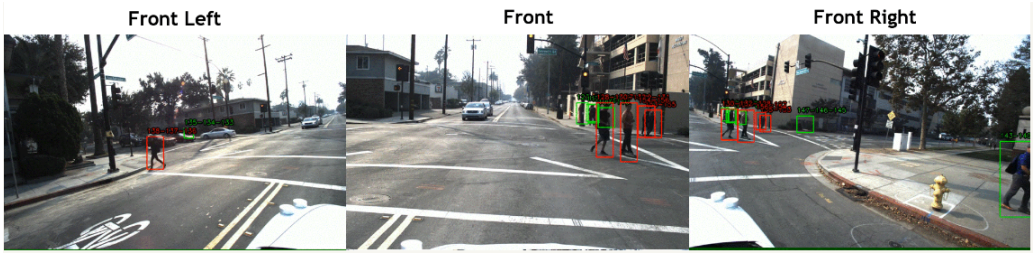


Figure 12: This image shows an overview of the STIP [56] dataset's camera images. Pedestrians are annotated with cross (red) not-cross (green) labels. [56]

already used this dataset to generate OGMs. Examining [61]'s research could therefore speed up the OGM generation time.

The RoboCar dataset [58] is collected in only one city and no data is labeled. It is therefore not a diverse dataset compared to the other datasets. The dataset contains 360-second sequences, but of an unkown number. What is beneficial about this dataset is that [62] and [63] used it for OGM generation before, where [62] also performed OGM prediction.

3.2 Object Detection Datasets

This subsection will discuss five datasets that were originally obtained for object detection purposes. The ECP2.5D dataset [60], the Berkely DeepDrive 100K (BDD100K) dataset [64], the Karlsruhe Institute of Technology and Toyota Technological Institute (KITTI) dataset [65], the nuScenes dataset [66], and the Waymo Open dataset [67].

The ECP2.5D dataset [60] contains 15-second sequences in which seven different classes are labeled. This is average compared to the other datasets. This dataset is remarkable because it is recorded in 31 cities in 12 countries which makes this the most diverse datatset regarding geographic locations. Also, the sampling frequency of the sequences is 20 Hz, which is the highest together with the STIP dataset [56]. An example of the camera and LiDAR data of the ECP2.5D dataset [60] is shown in figure 13.

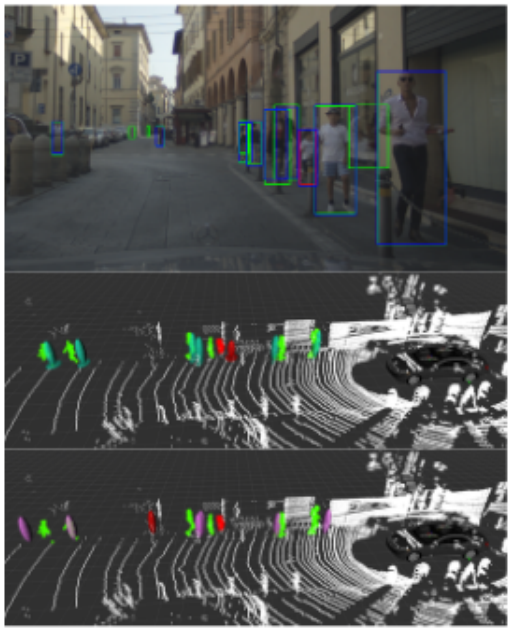


Figure 13: This image shows an example of how the ECP2.5D [60] dataset contains camera and LiDAR data in which pedestrians are annotated.

The BDD100K dataset [64] 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 which makes it average on geographic diversity. Because of the crowd-sourcing, large amounts of data could be obtained. The dataset contains 100K 40-second sequences which makes it one of the largest datasets compared to the other datasets. Moreover, the dataset contains pixel-level semantic segmentation of the vehicle's drivable area. A downside of this dataset is that it is obtained using solely a monocular camera. Therefore, some unsupervised depth estimation technique must be performed to gather 3D data, like the STIP dataset [56].

The KITTI dataset [65] is the smallest dataset (of which the number of sequences is known) compared to the other datasets. It only contains 22 sequences and is obtained in only one country. The benefit of using this dataset is that it has been used by [68], [24], and [55] to generate and predict OGMs. So, there is enough research to compare prediction methods with that use the KITTI dataset [65].

For the nuScenes dataset [66], 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. However, the sampling frequency of the data is only 2 Hz. This is the lowest sampling frequency compared to other datasets. Figure 14 shows an example of the six camera images overlayed with the LiDAR point cloud of the nuScenes dataset [66]. The advantage of this dataset is that it contains velocity information from the radar data. Besides that, the nuScenes dataset is collected for object detection as well as object tracking. Therefore, it contains 1000 sequences of 20 seconds. Objects of 23 classes are annotated, including vehicles, pedestrians, and riders. So, the nuScenes dataset [66] is larger and more diverse than most other datasets. The nuScenes paper [66] compares its dataset against the KITTI dataset [65]. It concludes that training on the KITTI dataset, with its smaller size compared to nuScenes, affects a network's performance.

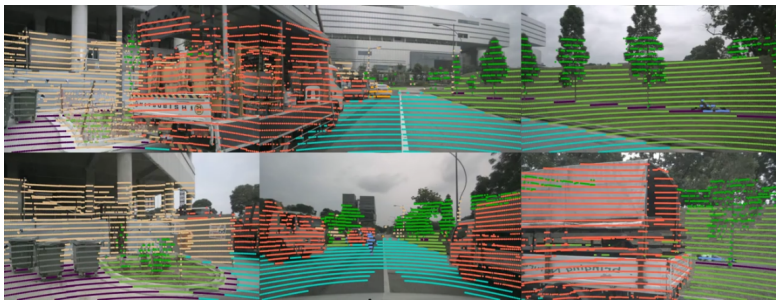


Figure 14: This image shows an example of the images of the six cameras from the nuScenes dataset [66]. The images are overlayed with the semantic segmented LiDAR data.

The Waymo Open dataset [67] (in short Waymo), was originally recorded in 3 cities (Phoenix, San Francisco, and Mountain View) and later extended with another 3 cities (Los Angeles, Detroit, and Seattle). This makes the dataset average on geographic diversity. Besides object detection, the Waymo dataset [67] also 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 sequence 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. This makes the dataset one of the largest datasets. Moreover, [55] and [69] used the Waymo dataset to generate and predict OGMs, so there is research to compare prediction methods with using the Waymo dataset [67].

3.3 Semantic Segmentation Datasets

This section discusses two datasets that are originally generated for semantic segementation tasks. The Apolloscape dataset [70] and the Cityscapes dataset [46].

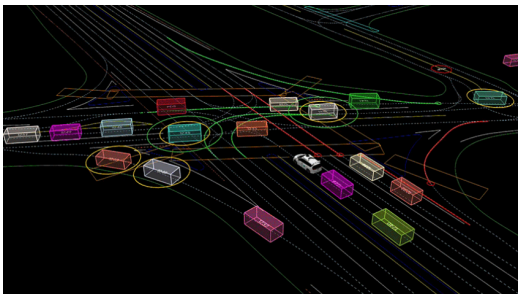
Apolloscape [70] is a dataset which is collected for the purpose of scene parsing (semantic segmentation on pixel-level). Therefore, the dataset contains almost 144K frames with pixel-level annotations for semantic segmentation. Furthermore, 89K instance-level annotations are provided for movable objects. 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. This makes the dataset diverse regarding traffic participants. The downside is that the dataset is recorded in only one city, and there is no information available about presence of sequences (only individual frames).

Figure 15: This image shows an example of the Cityscapes [46] dataset. The camera image is semantically segmented. Each object class in the image is given a different label (color).

The Cityscapes dataset [46] 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. An example an annotated image from the dataset is shown in figure 15. 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. Therefore, this dataset is diverse regarding both geographic locations and traffic participants. The dataset has also been used to generate OGMs by [71]. The downside of this dataset is that the number of sequences is not known, besides that it is a 'large set'.

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

Table 4 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. If a criterion is met, the score of the dataset goes up by 1 point. If the criterion is met partially, or in average quality compared to the other datasets, the score goes up by 0.5 point. If a criterion is not met, or if it is met in a bad quality compared to other datasets, the dataset will not get any points for that criterion. Also, if information about a criterion is not found or not available, no points are given to the criterion. Table 3 shows an overview of how well each datasets scores per criterion. More information about the symbols and what score is linked to them is written below the table.



(a) This image shows an overview of the labeled vehicles and lane markings of the Waymo [67] dataset.

(b) This image shows a front camera image of the Waymo [67] dataset.

Figure 16: Two examples of the Waymo [67] dataset.

With a score of 7.5, the Waymo [67] dataset (shown in figure 16) is the most suitable dataset to generate OGM sequences for OGM prediction. While the diversity of the Waymo dataset is average, compared to the other datasets, and there is no (available) data to extend the OGM form, the Waymo dataset meets all other criteria in good quality which sums

up to the score of 7.5. Other suitable datasets on the second and shared third places are the Cityscapes [46], Argoverse [57] and NuScenes [66] datasets with a score of 7.0, 6.5 and 6.5 respectively. Unlike the Waymo dataset, the Cityscapes dataset provides pixel-level semantic segmentation data and the Nuscenes dataset provides velocity data and semantic LiDAR data. In the case this additional data is desired for research on a specific OGM prediction method, the Cityscapes or Nuscenes datasets can be more suitable to use than the Waymo dataset. Also, conveniently, for the four best scoring datasets there is literature available that provides code to generate OGMs from the data. These sources can be found in table 4.

	STIP [56]	Argoverse [57]	ECP2.5D [60]	BDDK100 [64]	KITTI [65]	NuScenes [66]	Waymo [67]	Apolloscape [70]	Cityscapes [46]	RoboCar [58]
1	✗	✓	✓	✗	✓	✓	✓	✓	✓	✓
2	±	±	±	+	−	±	±	−	+	−
3	±	+	±	+	−	+	+	U	U	U
4	+	+	+	U	+	−	+	U	+	+
5	✓	✓	✓	✓	✓	✓	✓	✓	✓	U
6	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
7	±	±	±	±	±	±	+	±	±	±
8	N	N	N	Pixel Semantics	N	Velocity data, LiDAR Semantics	N	Pixel Semantics	Pixel Semantics	N
9	N	Only OGM gen	N	N	OGM gen and pred	Only OGM gen	OGM gen and pred	N	Only OGM gen	OGM gen and pred
Scores	4.5	**6.5**	5.5	5.5	5.5	**6.5**	**7.5**	4.5	**7.0**	4.5

Table 3: The table shows for each dataset how well it scores per criterion. The scores are based on the information in table 4. The following symbols are used to evaluate each criterion: **+:** Good, **±:** Average, **−:** Bad, **U:** Unknown: **N:** None/Not available/Not found, **✓:** Yes, **✗:** No.
At the bottom row, each dataset is given a score based on which symbol is given at the evaluation, and whether there is useful information found for criteria 8 and 9. A '+', a '✓', 'OGM gen and pred', and any available additional data for an extended OGM form provide 1 point, an '±' and 'only OGM gen' provides 0.5 points, and the '✗', '−', 'U', and 'N' 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 vehicles and VRUs. **6.** The dataset data, and any OGMs 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.** The dataset provides data, including ground truth, that is required for generating extended OGM forms. **9.** Results of other research using the dataset for generating and predicting OGMs 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 [56]	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 [57]	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	-	[61]	-
ECP2.5D [60]	2020	Velodyne HDL-64E LiDAR with 120m range (5-20 Hz)	1 RGB camera of res 1920x1024 (20 Hz)	GPS/INS	20 Hz	12 countries, 31 cities, dense urban areas	15s sequences, 46K frames	7 classes	~218K	-	-	-	-
BDDK100 [64]	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 [65]	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	-	[68] [24] [55]	[68] [24] [55]

NuScenes [66]	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 and LiDAR semantics	[61] [72]	-
Waymo [67]	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, 6 cities, urban and suburban areas	103K 20s sequences	3 classes	~2.8M	~6.1M	-	[55] [69]	[55] [69]
Apolloscape [70]	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 [46]	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	[71]	-
RoboCar [58]	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	-	-	-	[62] [63]	[62]

Table 4: This table shows an overview of the datasets that are most likely to be suitable for OGM prediction purposes.

4 Metrics

AVs can have collision avoidance systems that make use of traffic scene predictions to estimate whether the ego-vehicle will collide with any of the surrounding traffic participants. Based on these systems the AV can make decisions to drive as safe as possible by preventing or mitigating collisions. For example, if a collision is predicted, the ego-vehicle can decide to adjust its velocity to prevent or mitigate the collision, or it will decide to perform an evasive maneuver [73] [74]. [73]'s system has even shown to be robust to noise in the environment representation. However, due to prediction errors, such as artifacts or displacement errors of traffic participants, the AV might make the wrong decisions (e.g. perform an evasive maneuver when in reality there is no imminent collision, or fail to perform an evasive maneuver when there is actually a collision risk). This can be dangerous for the AV and its surroundings. Besides for the task of collision avoidance, prediction of an AV's surroundings can also improve the efficiency and safety of motion planning by AVs in general. 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. So, for the safety of AVs and its surroundings, good predictions are important. Since this literature study focuses on OGM predictions, a metric is desired that compares the predicted OGMs with the ground truth OGMs.

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. What the best metric is for OGM prediction is therefore investigated in this chapter. To ensure safety, a metric is desired to meet the following demands. The more similar a prediction is to the ground truth, the better the metric should score the prediction. Also, global errors such as noise do not affect robust AV motion planning systems. Therefore, global errors should be weighed less in the metric score than local errors, such as artifacts or displacement errors. Then, prediction errors that are more likely to cause the AV to perform dangerous maneuvers, should have a worse score compared to prediction errors that are likely harmless. Furthermore, AVs have less time to correct their paths when there are large displacement errors or errors close to the AV. So, these errors should be considered worse than small displacement errors or errors further from the AV. Moreover, if there are artifacts, deletions or additions of occupied clusters, 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 predictions 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.

A selection of five metrics that are used to measure the quality of OGMs compared to the ground truth is made based on papers that research OGM prediction methods. These metrics are evaluated in this section based on the criteria described in this chapter. First, the Mean Squared Error (MSE) is evaluated in section 4.1. Second, the Structural Similarity Index Measure (SSIM) is evaluated in section 4.2. Third, section 4.3 evaluates the Image Similarity (IS) metric. Fourth, secion 4.4 evaluates the F1 score and Receiver Operating Characteristic (ROC) metrics together, because those metrics are based on similar mathematical principles. Finally, section 4.5 answers the sub-question: "*What is the best quantitative metric to determine the accuracy of a predicted OGM?*"

4.1 Mean Squared Error (MSE)

The Mean Squared Error (MSE) metric is used by [68], [55], and [69] to compare the OGM predictions with the ground truth. [55] and [69] also use the Image Similarity (IS) metric which is discussed later in this chapter. The Mean Squared Error (MSE) metric formula is shown in equation 16, where I is an $m \times n$ OGM, K its prediction, and i and j the OGM coordinates.

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i,j) - K(i,j)]^2 \tag{16}$$

The equation evaluates the predicted OGM as a whole and sums up the square of each cell's difference in value compared to the ground truth. So, the higher MSE, the more its cell's values differ from the ground truth on a global scale. [69]'s

research shows that the MSE increases with the prediction time horizon of the OGM, which is expected because of the accumulation of prediction errors with every time step. Because of its simplicity, the MSE is used to assess not only OGMs but images in general. However, there are implicit assumptions when using the MSE [75]. The main assumptions are that the MSE treats spatial positions of the grid cells, the sign of the error, and the relationship between the prediction and the ground truth independently, while it treats each cell with equal importance. Therefore, the MSE cannot recognize relations within an OGM, consider context, or differentiate local errors from global errors. The MSE would thus not be able to assess displacements, deletions or additions. It is, however, possible to give errors close to the OGM's center a higher weighting by multiplying those errors by a penalty factor.

4.2 Structural Similarity Index Measure (SSIM)

The Structural Similarity Index Measure (SSIM) is used by [24] to evaluate the predicted OGMs in their research. The SSIM is based on the underlying principle "that the human visual system is highly adapted to extract structural information from visual scenes." [75]. This principle makes humans sensitive in identifying structural distortions in images, which is the reason that the SSIM is designed to simulate that principle by focusing on image structure. Besides being sensitive for structural distortions, the human visual system also naturally compensates for non-structural distortions such as luminance (brightness) and contrast, since natural lighting causes a large variability in these distortions. The influence of luminance and contrast is separated from the structural distortions for better visual comprehension and recognition. The SSIM task is therefore subdivided into three comparisons: luminance, contrast, and structure [76]. These parts are explained as follows.

First the luminance of the OGM is computed. The luminance is assumed to be the mean intensity of an OGM. Equation 17 shows the formula for computing the luminance μ_X, where $m \times n$ is the size of the OGM and X represents an OGM. The luminance value of the ground truth and the prediction are compared using equation 18, where X and Y represent the predicted OGM and ground truth respectively and $C_1 = (K_1 L)^2$, where L is the dynamic range of the pixel intensity values (255 for 8-bit grids), and $K_1 \ll 1$ is a small constant.

$$\mu_X = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} X_i(i,j) \tag{17}$$

$$l(X,Y) = \frac{2\mu_X \mu_Y + C_1}{\mu_X^2 + \mu_Y^2 + C_1} \tag{18}$$

Second, the OGM's contrast is computed. The contrast ratio is assumed to be the standard deviation σ_X of the OGM, shown in equation 19. The contrasts of the prediction and the ground truth are then compared using equation 20, where $C_2 = (K_2 L)^2$, L is the dynamic range of the pixel intensity values, and $K_2 \ll 1$ is a small constant.

$$\sigma_X = \left(\frac{1}{mn-1} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} (X(i,j) - \mu_X)^2 \right)^{\frac{1}{2}} \tag{19}$$

$$c(X,Y) = \frac{2\sigma_X \sigma_Y + C_2}{\sigma_X^2 + \sigma_Y^2 + C_2} \tag{20}$$

Third, equation 21 is used to compare the structures between the prediction and the ground truth, where C_3 is a small constant and σ_{XY} is computed using equation 22.

$$s(X,Y) = \frac{\sigma_{XY} + C_3}{\sigma_X \sigma_Y + C_3} \tag{21}$$

$$\sigma_{XY} = \frac{1}{mn-1} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} (X(i,j) - \mu_X)(Y(i,j) - \mu_Y) \tag{22}$$

After the comparisons are performed, the SSIM combines them according to equation 23, "where $\alpha > 0$, $\beta > 0$ and $\gamma > 0$ are parameters used to adjust the relative importance of the three components." [76]. For simplicity, these values can be set as $\alpha = \beta = \gamma = 1$ and $C_3 = C_2/2$ [76]. The SSIM ranges from 0 to 1, where 0 means that the evaluated image is completely different from the ground truth, while a value of 1 means that the evaluated image is identical to the ground truth.

$$SSIM(x,y) = [l(x,y)]^\alpha \cdot [c(x,y)]^\beta \cdot [s(x,y)]^\gamma \tag{23}$$

The SSIM has the beneficial property that it can be used globally as well as locally (by separately evaluating spatial sections of an OGM). Also, because the SSIM is focused on structural similarities, it is expected that small displacements

are weighed less than large displacement, and that displacements are weighed less than additions or deletions, since displacements retain the original structure, while the additions and deletions disrupt the original structure. Finally, similarly to the MSE, the SSIM can give a higher weighting to errors in the center of an OGM if it is subdivided into sections and the center sections are given a higher error penalty.

4.3 Image Similarity (IS)

The Image Similarity (IS) metric is used in [55] and [69] to measure the predictions' quality of the scene structure compared to the ground truth. [77] developed the Image Similarity (IS) metric which they later used in [78]'s research on finding similarities between OGMs in order to merge them to generate bigger maps. [77] defines the IS metrics according to equations 24 and 25.

$$IS(m_1, m_2) = \sum_{c \in C} d(m_1, m_2, c) + d(m_2, m_1, c) \tag{24}$$

$$d(m_1, m_2, c) = \frac{\sum_{m_1[p_1]=c} min\{md(p_1, p_2) | m_2[p_2] = c\}}{\#_c(m_1)} \tag{25}$$

Where C denotes the set of 'colors', in the case of an OGM empty, unknown and occupied. $m_i[p]$ is the color c of grid m_i at position $p = (x, y)$. $md(p_1, p_2) = |x_1 - x_2| + |y_1 - y_2|$ is the Manhattan-distance between p_1 and p_2. $\#_c(m_i) = \#p_1 | m_i[p_1] = c$ is the number of pixels in m_i with color c [77].
The IS metric is the sum over all the grids colors (grid cell states) of the average Manhattan-distance of the cells with color c (i.e. empty, unknown or occupied) in grid map m_i to the nearest cell with color c in grid map m_j ($d(m_1, m_2, c)$) and the average Manhattan-distance vice versa [77]. The higher the IS value, the larger the difference between the evaluated image and the ground truth.

The IS metric, similarly to the SSIM, compares image structures. Therefore, this metric can be used both on a global and local (spatial subsections) scale to compare OGMs. Also, because IS metric evaluates based on structure, it is expected that small displacements are weighed less than large displacements, and displacements in general are weighed less than deletions and additions. Like the other metrics, higher weights can be given to spatial subsections that match the AV's location (center of the grid map) to weigh errors close to the AV more than errors farther away from the AV.

4.4 F1 score and Receiver Operating Characteristic (ROC)-curve

To evaluate OGM prediction quality, [62] uses the F1 score (equation 29), [79] the Receiver Operating Characteristic (ROC)-curve, and [80] uses both. These two metrics together make use of the True Positive Rate (TPR) (Recall) in equation 26, Positive Predictive Value (PPV) (Precision) in equation 27, and False Positive Rate (FPR) (Fall-out) in equation 28 to determine how well the predictions match the ground truth. Here, a positive condition signifies that a grid cell is occupied, where a negative condition signifies that it is empty.

$$recall = TPR = \frac{\sum TruePositive}{\sum ConditionPositive} \tag{26}$$

$$precision = PPV = \frac{\sum TruePositive}{\sum PredictedConditionPositive} \tag{27}$$

$$fall\text{-}out = FPR = \frac{\sum FalsePositive}{\sum ConditionNegative} \tag{28}$$

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{29}$$

The F1 score combines the precision (PPV) and recall (TPR) of a predicted OGM to generate a score that is 1 if the prediction is identical to the ground truth and 0 if the prediction does not match the ground truth at all (meaning no true positives are predicted). The recall is the probability that a positive grid cell in the ground truth is also predicted as positive. Precision is the probability that a predicted positive grid cell is correctly predicted.

The ROC-curve is a graphic metric in which the TPR is plotted against the FPR at various threshold values γ for which a grid cell is considered occupied. When predicting OGMs, it is desired that the TPR is maximized and the FPR is minimized. However, this performance depends partly on the threshold γ. If $\gamma = 0$, all grid cells with a value above 0 are considered occupied (positive). This means that the TPR is equal to 1, but the FPR is equal to 0. The opposite happens

	MSE	SSIM	IS	F1/ROC
1.	✓	✓	✓	✓
2.	✗	✓	✓	✗
3.	✗	✓	✓	✗
4.	✓	✓	✓	✓
5.	✗	✓	✓	✗

Table 5: This table gives an overview of the metrics that can be used to evaluate OGM predictions. For each criterion (numbers 1 to 5), the table shows whether the metric meets it (✓) or not (✗). The criteria are as follows.
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.

when $\gamma = 1$. Those outcomes are not desired, so the optimal γ is often found between 0 and 1.

The F1 score and ROC-curve evaluate an OGM only on a global scale where each grid cell is regarded independently from other grid cells. Therefore, the metric does not distinguish large displacements from small ones or additions and deletions from displacements. It is possible to locally weigh some grid cells more than others if a penalty factor is implemented.

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

The evaluation of the criteria which a metric must meet to be suitable for OGM prediction is shown in table 5. Based on this chapter's information about the different metrics, the most suitable metrics are the SSIM and IS metrics. These metrics meet all the criteria. However, to determine whether the SSIM is more suitable than the IS metric, or whether there is another metric that is more suitable than the metrics from this chapter, more research should be done to investigate OGM evaluation metrics.

5 Occupancy Grid Map prediction methods using Deep Learning

OGM prediction is the prediction of a sequence of future OGMs given a past sequence of OGMs. This literature review focuses on OGM prediction using deep learning. The goal is to train a deep learning network until it accurately generalizes traffic behavior and extrapolates that behavior to make predictions. To achieve this, first, OGM data is required to train the network. The data, in the form of OGM sequences, can be generated from a dataset, as is discussed in chapters 2 and 3. Second, a suitable network architecture is required which can process OGM sequences. The network should extract spatial and temporal features of traffic behavior to generate predictions. Third, a suitable loss function is required to evaluate the network's performance during training to update its weights. Like a metric, as discussed in chapter 4, the loss function should ideally evaluate the predictions based on safety assurance. The chosen dataset, the network architecture, and the loss function that is used to update the network's weights all influence the prediction results. Finally, a metric is used to quantitatively evaluate different deep learning OGM prediction methods. By comparing the metric outcomes for each method, the best performing method can be picked.

This chapter gives an overview of current ego-vehicle traffic scene OGM prediction methods. All described methods are deep learning based. First, section 5.1 discussed three methods that are based on the PredNet [81] architecture. Second, section 5.2 discusses two DOGMa prediction methods. Third, section 5.3 discusses two methods that are based on a deep tracking network architecture. Fourth, 5.4 discusses a method that is based on the MotionNet [49] architecture. After these methods are reviewed, the loss functions that are used to update the network's weights in each of the methods are evaluated in section 5.5. This chapter ends with section 5.6, which provides the answer to the final sub-question of this literature study: "*What Deep Learning method generates the best OGM predictions?*"

5.1 PredNet-based OGM predictors

This section covers three methods that are all based on Lotter's [81] Predictive Coding Network (PredNet) architecture. First, Itkina [68] adjusts PredNet to be suitable for OGM prediction. Second, Lange [55] attempts to improve Itkina's [68] architecture by adding an attention mechanism that better learns dependencies between sequential grid cells. Third, Toyungyernsub [69] extends Itkina's [68] architecture to separate the static from the dynamic environment so the network can better distinguish dynamic behavior.

Initially, Itkina [68] proposes to utilize PredNet, an architecture based on the human brain predictive coding principle. This principle hypothesizes that the human brain continually predicts its incoming stimuli (e.g. visual stimuli from the eyes), compares these predictions against the actual stimuli, and generates an error signal to update the predictions. To mimic this principle, the PredNet architecture consists of multiple concatenated modules that all contain the same four parts: an input layer, a representation layer, a prediction layer, and an error representation layer (See figure 17). The input layer obtains features from the OGM (i.e. the actual stimulus) using convolutions and pooling. Then, a prediction of those features is made by the prediction layer (analogous to the brain's predictions), using the representation layer's features. The error representation layer generates an error by subtracting the input features from the predicted features (like the brain's generated error signal). After non-linearizing, the error is linked to both the representation layer and the output of the module. The representation layer learns temporal and spatial patterns from the error and the representation layers of neighboring modules using a ConvLSTM. Recurrently, the prediction layer then uses the representation layer's features to make predictions of the input features. Originally, the ConvLSTM was used for video prediction [81].

Itkina [68] deems the network to be suitable for OGM prediction as well because an OGM's format (a grid with elements ranging from 0 to 1) is similar to an image. Moreover, an OGM sequence is then considered similar to a video. Therefore, Itkina [68] repurposes the ConvLSTM [82] for OGM prediction. The convolutional part of the ConvLSTM is efficient in obtaining spatial features from image-like data structures. Subsequently, the Long Short-Term Memory (LSTM) [83] is a type of Recurrent Neural Network (RNN) architecture that can process sequential data and keeps track of long-term dependencies in the features obtained from the OGM sequences.

To train the PredNet, Itkina [68] generates OGMs from the KITTI [47] dataset. The L1-loss is used as the loss function as shown in equation 30. The L1-loss is the sum of absolute errors between the grid cells in the ground truth OGM I of size $m \times n$, and its prediction K, where i, j are the coordinates of the OGM. This loss function assumes an independence between grid cells. Itkina [68] compares the PredNet's performance with a baseline that assumes a static environment (for the short period of time the predictions last), with an FCN network, and with a particle filter predictor. Itkina [68] also compares the performance of the network with multiple variations of the input OGMs. Using only static OGMs is compared with using an additional DOGMa, which includes an additional dimensional layer to the OGM that contains dynamic state information (e.g. velocity), as discussed in chapter 2.4. Furthermore, using DST to generate the OGMs is

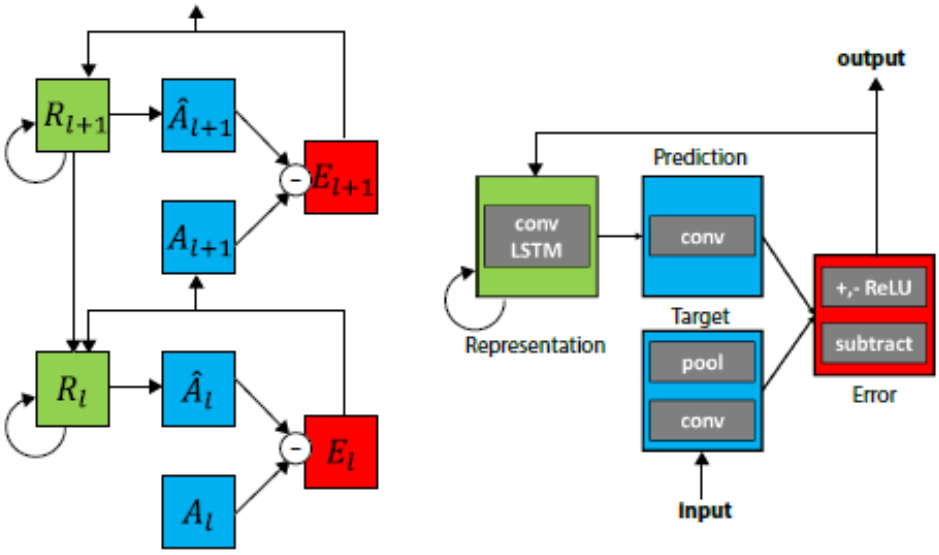


Figure 17: "Predictive Coding Network (PredNet). Left: Illustration of information flow within two layers. Each layer consists of representation neurons (R_l), which output a layer-specific prediction at each time step ($\hat{A}_l$), which is compared against a target (A_l) to produce an error term (E_l), which is then propagated laterally and vertically in the network. Right: Module operations for case of OGM sequences." [81].

compared with using the probabilistic alternative. Comparisons are done qualitatively, by visually comparing, and with the Mean Squared Error (MSE) metric that compares each cell of the OGMs.

$$\text{L1-loss} = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} |I(i,j) - K(i,j)| \tag{30}$$

The research of Itkina [68] 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 [55] proposes the Attention Augmented ConvLSTM (AAConvLSTM) for environment prediction. Lange [55] stresses the importance of OGM predictions, because this "approach facilitates the use of occupancy state estimation under uncertainty to update the belief of surroundings" [55]. However, the predictions must be reliable. Therefore, Lange [55] tries to reduce blurriness by implementing an attention mechanism into the PredNet architecture, which originated from creating long-term dependencies in language processing. Attention augmented convolutions replace the regular convolutions of the original ConvLSTM to form the AAConvLSTM. These attention augmented convolutions highlight inter-dependencies between spatial and temporal dimensions of the OGM sequences. Figure 18 shows an example of the attention mechanism applied to an image. Using attention augmented convolutions, compared to regular convolutions, allows the LSTM to store more relevant information in its long-term memory. This, in turn, Lange [55] expects to improve the network's prediction accuracy.

Lange [55] trains the AAConvLSTM PredNet using the Waymo [67] and KITTI [47] datasets. The L1-loss is used as the loss function (See equation 30), which evaluates each grid cell independently. When comparing the results with using the original PredNet architecture and another baseline, the AAConvLSTM performs better both based on qualitative visual comparison and based on the MSE and Image Similarity (IS) metrics. However, blurriness and disappearance of objects remains.

Toyungyernsub [69] also attempts to counter the blurriness and disappearing objects that resulted from Itkina's [68] method for long-term predictions. Itkina [68] concluded that there was almost no difference between using an OGM or a DOGMa as input. However, Toyungyernsub [69] expects that incorporating dynamic information directly in the network's architecture can solve the problems of Itkina's [68] network. Therefore, they propose a double-prong ConvLSTM network

Figure 18: An example of the attention mechanism applied to an image [84]. The mechanism places its attention on the STOP sign as it learned that this sign is the most relevant information it should retain in its memory.

based on the PredNet architecture (figure 19 shows the network's pipeline). This network splits its architecture in static and a dynamic prong. The static prong takes as input the static OGMs and makes static predictions, while the dynamic prong does the same for the DOGMas. Instead of generating DOGMas by using a particle filter to obtain velocity data (see chapter 2.4), Toyungyernsub [69] uses object detection and tracking to determine each grid cell's velocity between two separate OGM frames. Subsequently, the static and dynamic OGM predictions are fused to form a joint prediction as the output of the network.

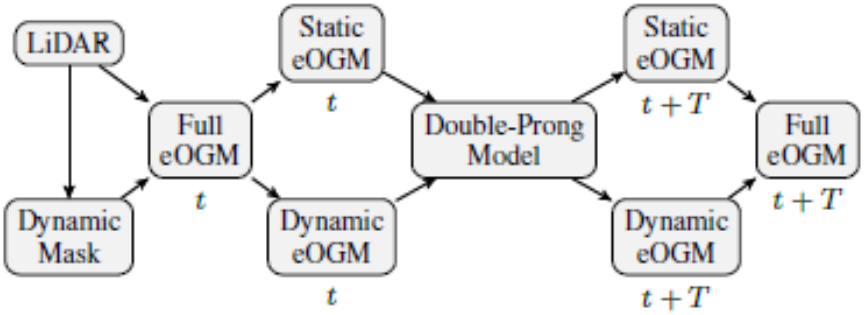


Figure 19: The Double-Prong network pipeline proposed by Toyungyernsub [69]. This network separately predicts the future static and dynamic OGMs and then fuses them to create the complete OGM prediction.

The double-prong network is trained on the Waymo [67] dataset and uses the L1-loss as loss function (See equation 30), which evaluates each grid cell independently. The method outperforms the original PredNet and two other baselines both in terms of the MSE and IS metrics and by visual comparison. 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.

5.2 DOGMa predictors

This section elaborates on two methods for DOGMa prediction. First, Hoermann's [79] method is discussed which uses a Convolutional Neural Network (CNN). Then, Schreiber's [80] method is discussed, which builds upon Hoermann's [79] research and uses encoder-decoder ConvLSTMs instead of a CNN.

Hoermann [79] proposes to predict Dynamic Occupancy Grid Map (DOGMa) using a Convolutional Neural Network (CNN). The network requires one DOGMa of the current time step and predicts up to 3 seconds into the future with time steps of 0.5 seconds. 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. Then, about 0.25% of a DOGMa contains dynamic grid cells, so if the network falsely predicts dynamic grid cells, the resulting loss would marginally affected. Therefore, the network learns to segment static and dynamic parts of the DOGMa. This is done, so that the loss can be split up into a static loss and a dynamic loss. By weighing dynamic grid cells relatively more than static grid cells in the loss, the network is prevented from ignoring the dynamic grid cells in its predictions. The output of the network is a multichannel OGM with one channel containing the occupancy of static grid cells and several channels, for each predicted time step one, containing the future occupancy of dynamic grid cells. The network's architecture is a downscaling cascade of CNNs followed by upscaling deconvolutions. It is based on Noh's [85] semantic segmentation network (See figure 20) but it replaces the unpooling layers with deconvolutions. This is because deconvolution kernels can be learned which allows for more parameters to generate predictions.

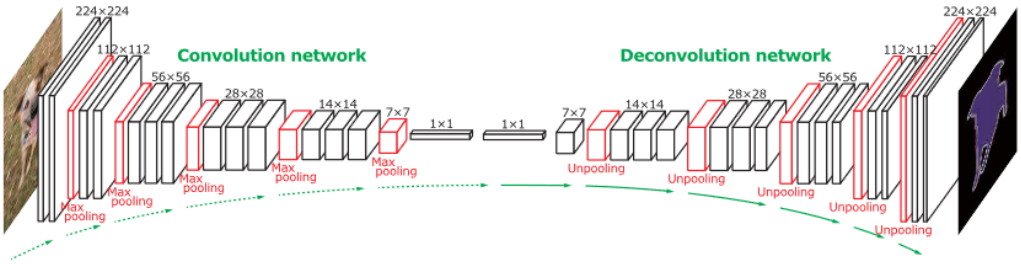


Figure 20: Noh's [85] CNN architecture that forms the basis for Hoermann's [79] DOGMa prediction network. In Hoermann's [79] architecture, the unpooling layers are replaced by deconvolution layers.

Hoermann's [79] network is trained on their own recorded dataset. The MSE is used as loss function (See equation 16 in chapter 4), in which the loss is subdivided into weighed static and dynamic parts. This loss assumes independent grid cells. The results are evaluated against a particle filter baseline using the True Positive Rate (TPR),False Positive Rate (FPR), and Receiver Operating Characteristic (ROC) as metrics and by visual comparison. The evaluations shows that the proposed method is better than a particle filter baseline. 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 [80] builds upon Hoermann's [79] research by expanding the CNN with a ConvLSTM encoder and decoder containing ConvLSTMs and with ConvLSTM skip-connections between the down-scaling and up-scaling parts of the network (See figure 21). The ConvLSTMs in the encoder and decoder capture spatial and temporal correlations. The skip-connections convey high resolution features, including occluded objects, to the up-scaling layers. Like Hoermann's [79] network, Schreiber's [80] network also splits the static and dynamic predictions to compute the loss separately. The network outputs a multi-channel OGM. One channel contains the occupancy of static grid cells, and for every predicted time step there are channels with the occupancy of the dynamic grid cells.

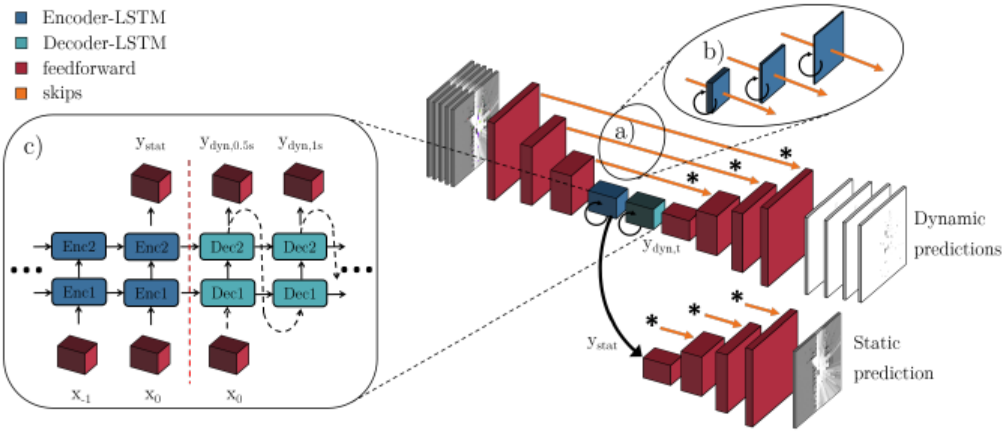


Figure 21: Schreiber's [80] ConvLSTM encoder-decoder network with ConvLSTM skip connections.

To train the network, Schreiber [80] recorded their own dataset. The loss function for the static environment is the L1-score (equation 30) and the MSE (equation 16 in chapter 4) for the dynamic environment. Both loss functions evaluate each grid cell independently. The F1-score and the ROC are used as metric to compare Schreiber's [80] network with Hoermann's [79] network, Dequaire's [62] network (which is discussed in the following section), and a particle filter. The methods are also compared based on visual quality. Schreiber's [80] method shows better F1-score results and better qualitative results compared to Hoermann's [79] 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. However, the data is only recorded in a stationary scenario. Therefore, the performance of the network with egomotion should researched in the future to investigate the accuracy of the predictions in driving scenarios.

5.3 Deep tracking based OGM predictors

In this section, two OGM prediction methods are described that use deep tracking based network architectures. First, Dequaire's [62] method is discussed which bases its network's architecture on the Deep Tracking framework introduced by Ondruska [86]. Then, Mohajerin's [24] method is discussed which builds upon Dequaire's [62] research by implementing a difference learning architecture.

Dequaire's [62] main research problem is to uncover the true, unoccluded state of the world in terms of OGMs, given a sequence of partially observable OGMs. Dequaire [62] generates two-channel OGMs. One channel contains the occupancy data (0 for Empty, 1 for Occupied), the other channel contains visibility information (0 for occluded, 1 for unoccluded). By uncovering the unoccluded state of a partially observable OGM, tracking (partially) occluded objects through the OGM sequence can be done more accurately. Furthermore, Dequaire [62] argues that OGM prediction is the same as uncovering the unoccluded state of a completely occluded OGM, given the past sequences of OGMs. Besides tracking and prediction, Dequaire [62] also assumes that if a network learns to predict traffic behavior, it must learn to distinguish shapes and motion patterns of the observed traffic participants. With this knowledge, the network can also be trained to perform semantic segmentation if class labels are provided in the OGMs during network training. This is because different object classes (i.e. pedestrians, cyclists, vehicles) have distinct shapes and motion patterns. To perform the tasks of tracking, prediction, and semantic segmentation, Dequaire [62] proposes a network based on Ondruska's [86] recurrent deep tracking network which contains RNNs. However, to improve this network's memory Dequaire [62] uses the Gated Recurrent Unit (GRU) in their network architecture (See figure 22). A Gated Recurrent Unit (GRU) is similar to an LSTM in that it retains memory for a longer term compared to an RNN. However, the GRU has fewer parameters than the LSTM, which makes it more efficient and more suitable for training on small datasets compared to the LSTM. Furthermore, to compensate for the ego-motion in between an OGM sequence's time steps, Dequaire [62] implements a Spatial Transformer (STM). The STM is a learnable module that uses odometry data to translate the GRU's hidden state of the previous time step to the current time step.

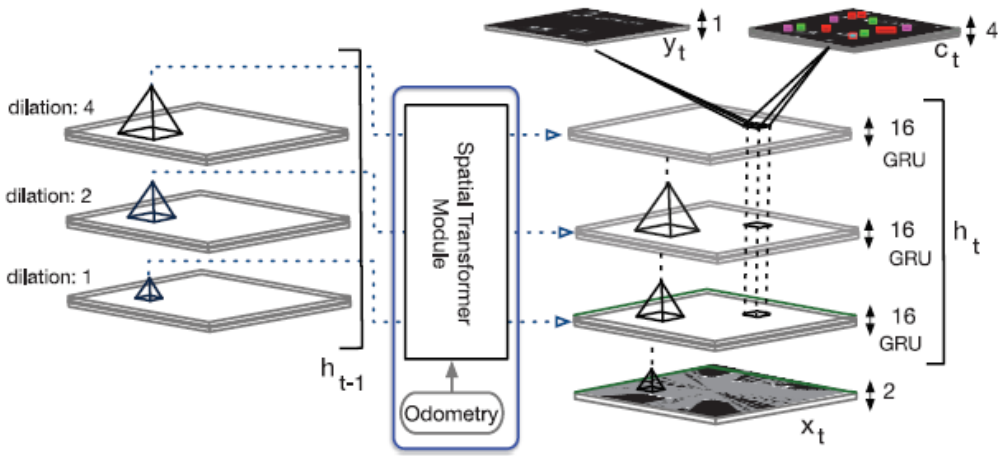


Figure 22: Dequaire's [62] GRU-based OGM prediction network architecture. The latent features from the previous time step go through the Spatial Transformer (STM) to compensate for the ego-motion in between the OGM frames.

$$\text{Cross-Entropy loss} = -\frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}(I(i,j)\cdot log(p(K(i,j))) - (1-I(i,j))\cdot log(1-p(K(i,j))) \tag{31}$$

The network is trained on Dequaire's [62] self-recorded dataset. The traffic scenes in the dataset are in no point in time fully observable. Therefore, the network is only trained to correctly predict the occupancy of OGM's occluded states that are visible in future ground truth OGMs. This is because, predictions cannot be evaluated if the ground truth does not contain visible data of the predicted content. The loss function that is used to train the network's predictions is the (binary) cross-entropy loss (equation 31). The cross-entropy loss compares the rounded values (0 or 1) of grid cells in the ground truth OGM I of size $m \times n$ with the predicted occupancy probabilities of K, where i, j are the coordinates of the OGM. The loss increases logarithmically with the error. This loss function assumes an independence between grid cells. The prediction results are compared against Ondruska's [86] Deep Tracking network using the F1-score and by visual comparison. The results show that Dequaire's [62] GRU-based network performs better than Ondruska's [86] Deep Tracking network.

Mohajerin [24] builds upon Dequaire's [62] research and proposes a difference learning method for the prediction of OGMs using ConvLSTMs in the network architecture. Figure 23 shows the difference learning architecture. The suggested method learns the difference between consecutive OGMs using a Motion-Flow Extraction (MFE) algorithm. The output of the MFE algorithm is a tensor of the same height and width as the OGMs which contains movement information in X and Y directions for each grid cell. This information is encoded and added to the encoded representation of the input OGM in the network's architecture. Subsequently, the encoded OGM and movement information flows through a ConvLSTM core after which it is decoded and results in the predicted change of motion for each grid cell of the OGM. This predicted change of motion is added to the input OGM and processed by a feed-forward classifier layer that provides the prediction of the full OGM for one time step. This predicted OGM is inserted back into the network to predict the next time step. This is iterated for the number of predicted time steps that are desired.

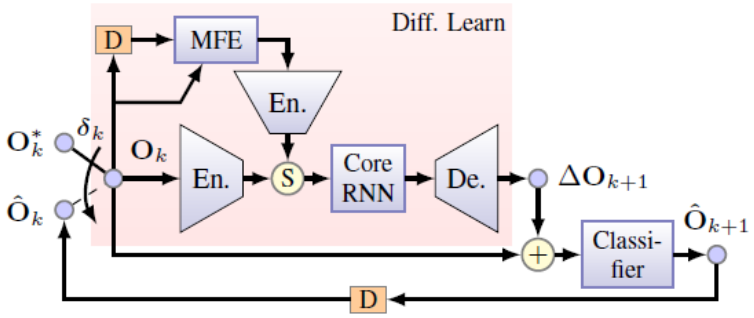


Figure 23: Mohajerin's [24] ConvLSTM-based difference learning OGM prediction network architecture. Two consecutive OGMs go through the MFE module to obtain a movement information for each grid cell. The encoded movement information is added to the current encoded OGM and processed by the Core RNN (the ConvLSTM). After decoding the result, it is processed by the Classifier to predict an OGM for one time step. To predict the next time step, the predicted OGM is inserted as the new input of the network.

$$\text{L2-loss} = \sum_{i=0}^{m-1}\sum_{j=0}^{n-1}(I(i,j) - K(i,j))^2 \tag{32}$$

The network is trained on the KITTI [47] dataset. A summation of three loss functions is used to train the network: the cross-entropy loss (equation 31), the L2-loss (equation 32), and the SSIM (equation 23 in chapter 4). The L2-loss is the sum of squared errors between the grid cells in the ground truth OGM I of size $m \times n$, and its prediction K, where i, j are the coordinates of the OGM. The cross-entropy loss and the L2-loss assume an independence between grid cells. However, the SSIM loss evaluates the structures between grid cells and thus considers the dependencies between grid cells. Mohajerin's [24] network is compared to a close approximation of Dequaire's [62] network using the TPR, True Negative Rate (TNR), and the SSIM metrics. The difference learning architecture outperforms Dequaire's [62] architecture in predicting future OGMs based on the SSIMs. Mohajerin [24] mentions that the provided ground truth data (from the KITTI [47] dataset) is an inadequate target to which the predictions should be compared, since the ground truth only contains the visible borders within the AV's 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.

5.4 MotionNet multi-channel OGM predictor

This last section discusses Wu's [49] MotionNet OGM prediction method. Wu [49] 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 represents a different height in the environment. This representation makes convolution possible. The MotionNet is a sptio-temporal pyramid network (See figure 24). It contains an encoder with multiple Spatio-Temporal Convolution (STC) blocks that consist of standard 2D convolutions for capturing spatial features, followed by a 1D convolution. The encoder is followed by a decoder using deconvolutions. Between the encoder and decoder there are temporal pooling skip-connections that convey temporal features at different scales to retain both global and local spatio-temporal contexts. The network contains three output heads each providing an output with specific information. The first head provides semantic segmentation of the OGM. The second head provides an OGM prediction for N time steps. The third head provides a classification of whether a grid cell is dynamic or static. Together, the three heads from a multi-channel OGM output with object labels, predictions, and dynamic information. This is similar to a sequences of DOGMas with semantic information.

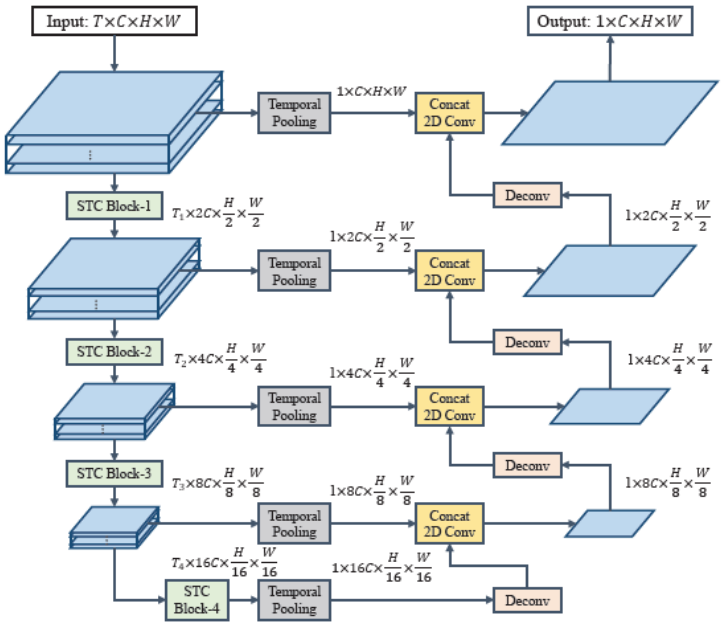


Figure 24: MotionNet, Wu's [49] spatio-temporal pyramid network containing STC building blocks and temporal pooling skip connections to predict OGMs.

The network is trained on the Nuscenes [66] dataset. The loss functions that are used are the L1-loss (equation 30) for the predictions, and the cross-entropy loss for the dynamic and semantic information (equation 31). Both loss functions assume independence between grid cells. The network is compared with Schreiber's [80] ConvLSTM encoder-decoder network, and four other baselines. The MSE metric and Median Squared Error (MeSE) metric (the median value of the squared errors of the OGM's cells) are used for evaluation. The evaluation is done by comparing the results at different velocities of the grid cell dynamics: at $0m/s$, $\leq 5m/s$, and $> 5m/s$. Wu's [49] MotionNet outperforms the other networks especially because of its ability to distinguish static and dynamic obstacles well.

5.5 Loss Functions

In the previous sections, several loss functions were used to train the deep learning networks. The loss function that is used influences what errors are penalized and thus how a network's weights are updated. The loss functions, similarly to metrics, evaluate the resemblance between a ground truth OGM and its prediction. So, the better a loss function can determine the quality of a prediction, the more accurate the weights of a network can be updated to achieve accurate

	L1	L2	MSE	Cross-Entropy	SSIM
1.	✓	✓	✓	✓	✓
2.	✗	✗	✗	✗	✓
3.	✗	✗	✗	✗	✓
4.	✗	✗	✗	✗	✓
5.	✗	✗	✗	✗	✓

Table 6: This table gives an overview of the loss functions that are used in the OGM prediction methods. For each criterion (numbers 1 to 5), the table shows whether the loss function meets it (✓) or not (✗). The criteria are as follows (the same criteria as the metrics criteria from chapter 4).
1. The loss function can evaluate the OGM as a whole. **2.** The loss function can evaluate the OGM on local and on global scale. **3.** The loss function negatively weighs small displacements less than big displacements. **4.** The loss function negatively weighs errors close to the AV (often the center of the OGM) more than errors further from the AV. **5.** The loss function negatively weighs additions and deletions more than displacements.

predictions. Therefore, this section evaluates the loss functions that are used in the previous sections. Since metrics and loss functions fulfill the same purpose, the criteria to evaluate the metrics are also used to evaluate the loss functions. An additional criterion for loss functions is that they must be (pseudo-)differentiable to backpropagate the error through the deep learning network. However, in this case, all loss functions are (pseudo-)differentiable since they have been used in the networks described earlier in this chapter. Table 6 shows how well the loss functions meet the criteria. The criteria are listed below the table. Because all loss functions except the SSIM assume an independence between the grid cells of the OGM, they only meet the first criterion. The SSIM does consider dependencies between grid cells, since it evaluates structures. Therefore, as discussed in chapter 4, the SSIM meets all criteria.

5.6 What Deep Learning method generates the best OGM predictions?

The best results of each method that is described in this chapter are shown in table 8. Also, table 7 shows an overview of the OGM prediction methods. Since not all methods are trained on the same datasets and evaluated using the same metrics, it is difficult to compare them. Even within the same method's subcategories (PredNet, DOGMa prediction, Deep Tracking, and MotionNet) there is no consistent use of datasets and metrics. If the use of different datasets is not considered, [55] provides the lowest MSE and the lowest IS (on the Waymo [67] dataset), given the available information. Furthermore, Mohajerin's [24] method has the highest TPR. Then, when looking at the ROC curves that compare Schreiber's method [80] with Hoermann's method [79] in figure 25, Schreiber's method [80] performs better. Also, Schreiber's method [80] has the highest F1-score. Unfortunately, Mohajerin's [24] SSIM score cannot be compared because no other methods are evaluated using that metric. After this analysis, still no conclusive answer can be given about what method generates the best OGM predictions. Especially since chapter 4 concludes that the SSIM and IS metrics are more suitable to evaluate OGMs compared to the other metrics. This makes it hard to determine how much weight a result from a specific metric should be given compared to the other metrics. Therefore, comparing the methods using different metrics is not feasible. In conclusion, to answer the question: "*What Deep Learning method generates the best OGM predictions?*", the different methods should be evaluated using the same metric(s).

Paper/Method	Dataset(s)	Network(s)	Input	Output	Loss Function(s)	Metric(s)
Itkina [68]	KITTI [65]	PredNet-based ConvLSTM	DOGMa, DST OGM	DST OGM	L1-Loss	MSE
Lange [55]	KITTI [65], Waymo [67]	PredNet-based AAConvLSTM	DST OGM	DST OGM	L1-Loss	MSE, IS
Toyungyernsub [69]	Waymo [67]	PredNet-based ConvLSTM Double-Prong	DOGMa, DST OGM	DST OGM	L1-Loss	MSE, IS
Hoermann [79]	Own	CNN-based Grid Predictor Model	DOGMa	OGM	MSE	ROC, TPR, FPR
Schreiber [80]	Own	ConvLSTM-Encoder-Decoder	DOGMa	OGM	L1-Loss, MSE	ROC, F1-score
Dequaire [62]	Own	Convolutional GRU	OGM with semantics	OGM with semantics	Cross-entropy	F1-score
Mohajerin [24]	KITTI [65]	ConvLSTM-Encoder-Decoder-based variations	OGM	OGM	Cross-entropy, L2-Loss, SSIM	TPR, TNR, SSIM
Wu [49]	NuScenes [66]	MotionNet CNN-based	OGM at multiple heights	DOGMa with semantics	L1-Loss, Cross-entropy	MSE, MeSE

Table 7: This table contains an overview of properties of the different OGM prediction methods.

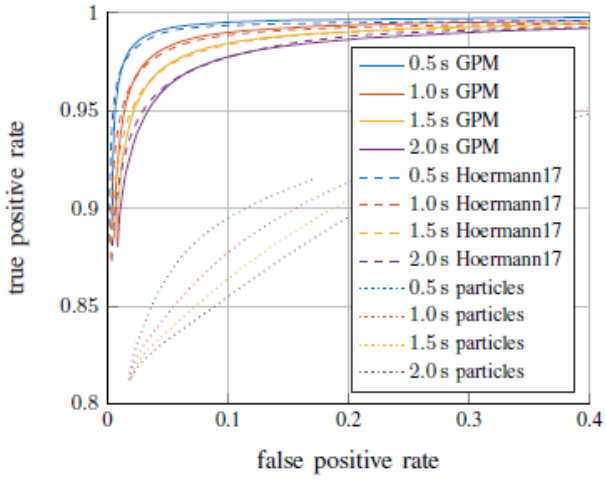


Figure 25: The ROC curves of Schreiber's method [80] (GPM), compared to Hoermann's method [79] and a particle filter.

			Results					
	Paper/Method	**Datasets**	**MSE** ($\cdot 10^{-2}$)	**IS**	**TPR**	**ROC**	**F1-Score**	**SSIM**
PreiNet	**Itkina [68]**	KITTI [65]	3.295 ± 0.1625	✗	✗	✗	✗	✗
	Lange [55]	KITTI [65],Waymo [67]	KITTI: 3.62 ± 0.0012, **Waymo: 2.33 ± 0.0063**	KITTI: 6.91 ± 0.06, **Waymo: 3.51 ± 0.04**	✗	✗	✗	✗
DOGMa	**Toyungyernsub [69]**	Waymo [67]	3.17 ± 0.0007	5.86 ± 0.058	✗	✗	✗	✗
	Hoermann [79]	Own	✗	✗	Min: 0.75 (0.5s), Max: 0.81 (2.0s)	✓	✗	✗
	Schreiber [80]	Own, Nuscenes [66]	Nuscenes: 5.92 [1] [49]			✓	**Best: 0.9 (0.5s), Worst: 0.8 (2.0s)**	✗
Deep Tracking	**Dequaire [62]**	Own	✗	✗	✗	✗	[80]'s dataset: Best: 0.8 (0.5s), Worst: 0.6 (1.5s) [80]	✗
	Mohajerin [24]	KITTI [65]	✗	✗	**0.8836** for TNR of 0.9928	✗	✗	**0.9841**
Motion Net	**Wu [49]**	NuScenes [66]	3.42 [2]	✗	✗	✗	✗	✗

Table 8: This table shows the results for each OGM prediction methods categorized by the metric(s) that is/are used to evaluate the method. Some methods are trained multiple times using different datasets. In that case, the datasets are mentioned before each result.

[1] This is the weighted average MSE based on [79]'s finding that 0.25% of an OGM consists of dynamic grid cells. The assumption is made that the number of grid cells with the velocities of $\leq 5m/s$ and $> 5m/s$ are of equal amount.

[2] This is the weighted average MSE based on [79]'s finding that 0.25% of an OGM consists of dynamic grid cells. The assumption is made that the number of grid cells with the velocities of $\leq 5m/s$ and $> 5m/s$ are of equal amount.

6 Conclusion

This conclusion will give an answer to the main question of this literature study: "*What is the best method to perform traffic scene OGM prediction using Deep Learning?*" To answer this question, an investigation has been done to find the most suitable OGM form, dataset, metric, and deep learning method, by answering the following four 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 quantitative metric to determine the accuracy of a predicted OGM?*
4. *What Deep Learning method generates the best OGM predictions?*

The answers to the sub-questions, that together answer the main question are as follows. First, the most suitable OGM to use in prediction methods is the DST generated OGM. Whether a specific OGM extension should be used depends on the goal for the prediction. Generally, the more extensions, the more accurate the predictions, but the more computation power is required to train and execute the prediction network. Second, the most suitable dataset to generate OGM sequences for OGM prediction is the Waymo [67] dataset for generating OGMs without extensions, the Cityscapes [46] dataset for generating OGMs with semantic segmentation on pixel level, and the Nuscenes [66] dataset to generate DOGMas using radar velocity data and to add semantic segmentation on pixel level. Thirds, the best quantitative metric to determine the accuracy of a predicted OGM is either the SSIM or the IS metric. Fourth, the deep learning method that generates the best OGM predictions cannot be determined yet. The investigated methods cannot be compared well, because they are either trained on different datasets or evaluated using different metrics.

7 Future Work

Based on the conclusion of this literature study, some ideas for future research are proposed in this section. The proposals below are categorized into four categories: Datasets, Metrics, Loss functions, and Methods. Subsequently, chapter 8 proposes a Master thesis research which elaborates on the Loss Functions future work.

Datasets This literature review gave an extensive overview of the plethora of datasets that contain traffic scene information from an ego-vehicle point of view. However, most datasets were not recorded with the purpose to use them for OGM generation and (motion) prediction. Therefore, not all datasets have been used before for these purposes, given the found literature. Based on the set criteria for the datasets, if there was literature that showed that a dataset has previously been used to generate or predict OGM, its score increased. Intuitively, it is probable that the motion prediction datasets are more likely to be used for OGM generation and prediction compared to the other datasets. Therefore, those datasets are more likely to score higher based on the set criteria. To even out the dataset's playing field, future research could be done by investigating whether OGMs can be generated and predicted for each dataset. Then, the focus could lie more on the other criteria. Furthermore, the datasets are evaluated according to criteria that were set based on literature about dataset quality and based on practical reasons. Perhaps, given some other criteria, the datasets would have scored differently. Therefore, this review proposes a research that investigates the quality of traffic scene datasets used for OGM generation and prediction. To evaluate the datasets, more extensive research could be done to set up adequate criteria that a dataset should meet.

Metrics The evaluated metrics in this literature review were metrics that were previously used in papers about OGM prediction. This review concluded that, based on the set criteria, the best metrics to evaluate OGMs are the SSIM and IS metrics. To determine which of the two metrics is better, research could be done that compares the metrics against other criteria. For example, the criteria in this literature study were based on ensuring traffic safety. To test traffic safety, an investigation in a traffic simulation could be done to evaluate the behavior of an AV and its surroundings, given different OGM predictions. The metric that best correlates its score of the OGM predictions with the resulting traffic safety would be the best metric. Furthermore, OGMs have a similar structure to images. Therefore, there might be other metrics, originally used for image comparison, that could be more suitable to assess OGMs compared to the ones in this literature study. Research should be done to find more metrics and compared those to the metrics assessed in this review.

Loss Functions Chapter 5.5 assesses the various loss functions that were used to train the different OGM prediction methods. When evaluating the loss functions against the criteria that were set for the metrics, it is found that only one loss function meets all criteria. This loss function is the SSIM. The reason why it meets all set criteria is because it considers grid cell dependencies compared to the other loss functions which only assess each grid cell independently. According to the metrics criteria, the former would provide a better assessment of the network's output compared to the latter. This raises the question whether the use of a loss function that considers grid cell dependencies, such as the SSIM, improves how a network is trained compared to using a loss function which evaluates each grid cell independently. Therefore, future work is proposed that compares the effect of loss functions that consider independent grid cells with loss functions that consider a dependency between grid cells.

Methods In this literature study, only OGM prediction methods are evaluated. However, there are other methods found in literature that use OGMs as input but only predict the future states of the traffic participants within the OGM instead of predicting the complete future OGMs. Two examples of state prediction methods using OGM sequences as input are given as follows. A recent state-of-the-art method by Li [87] proposes an end-to-end Transformer network that performs object detection and trajectory prediction of an AV's environment using its raw LiDAR and camera data as input in the form of 3D OGMs. A Transformer network is a sequential data processing network that does not use a form of RNN in its architecture. It makes use of the attention principle, as discussed in chapter 5.1. Also, Luo [88] proposes a real-time end-to-end 3D detection, tracking, and motion prediction network which only uses a CNN and performs the prediction task within 30ms. They use a sequence of 3D OGMs as input and perform 3D convolutions on it in the network to predict the motion of traffic participants. Both these methods achieve state-of-the-art results in state prediction. Future research could be done to investigate whether an adjusted version of these methods can be used for OGM prediction. Furthermore, the methods that were evaluated in this review could hardly be compared due to the fact that most of the methods were trained on different datasets and evaluated using different metrics. In the future, to compare the methods better, they should trained on the same dataset and evaluated using the same metrics.

8 Research Proposal

Based on the conclusions of the literature study, the following research plan is proposed. First, a title is presented in section 8.1. Then, some background knowledge describing the research problem is described in section 8.2. Subsequently, the research questions are proposed in section 8.3. After that, the methodology is explained in section 8.4. This is followed by a planning in section8.5 to execute the proposed research.

8.1 Title

Investigating the effect of loss functions on Occupancy Grid Map (OGM) prediction methods.

8.2 Background

Like metrics, loss functions assess the quality of a network's performance. A loss function is used to train network, while a metric is used to evaluate a trained network. Loss functions compute the error between a network's output and the ground truth and back-propagate it to update the network's weights. This is how a network is trained. A metric and a loss function share the same purpose, except that a loss function should be differentiable so its derivative can be determined for back-propagation. Therefore, it makes sense that a loss function is tested against the same criteria as a metric to determine its ability to evaluate a specific task. In this case the task is to compare OGMs. In the literature study, the loss functions that were used in the different OGM prediction methods were evaluated against the criteria that were set for the metrics. The literature study discovered that most OGM prediction methods using deep learning networks are trained using a loss function that does not consider grid cell dependencies. These loss functions only meet one of the five criteria (see chapter 5.5). One loss function, which meets all five criteria, considers grid cell dependencies in its loss function.

From this background, a problem arises. Several deep learning networks that are trained to predict OGMs, use different loss function that score differently when evaluating them against the metric criteria. This raises the question whether using loss functions that consider grid cell dependencies when evaluating OGMs result in better performing networks compared to using loss functions that evaluate each grid cell independently. Additionally, if said case occurs, would the magnitude of improvement of the network's performance be consistent in multiple networks? This results in the following main research questions and two sub-questions in the next section.

8.3 Research Questions

The main question for this research proposal results from the background and problem statement and is as follows:

What is the effect of different loss functions in Deep Learning methods for Occupancy Grid Map (OGM) prediction of traffic scenes?

This main question can be answer by investigating the answer to the following two sub-questions:

1. *Does using a loss function that considers grid cell dependencies when evaluating OGMs when training a network improve its results compared to using a loss function that evaluates each grid cell independently?*
2. *Is the effect of different loss functions consistent when they are used to train different deep learning networks?*

To answer the questions a methodology is described in the following section.

8.4 Methodology

To answer the research questions, multiple deep learning networks have to be trained, using various loss functions, to perform OGM predictions. As described in the literature study, to perform OGM predictions several components are required. A OGMs generation method is required, a dataset that contains suitable data for OGM generation and prediction should be chosen, the various loss functions must be chosen to train the networks, the networks that predict the OGMs should be chosen, and metrics to evaluate and compare the network's results are required. Finally, experiments can be done to answer the research questions. The choices for the components are discussed in the following subsections.

8.4.1 OGM generation method

The OGMs will be generated using the DST method. The literature study shows that this method is the most suitable one to use for OGM prediction. Moreover, this method has been used before by the PredNet-based OGM prediction methods, which gives a practical reason to use it together with one of those methods.

8.4.2 Dataset

The Waymo [67] dataset will be used because the literature study shows it is the best dataset for OGM prediction. Furthermore, no dynamic or semantic information is required for this research, therefore the second and third best datasets are not considered.

8.4.3 Loss Functions

The MSE and L1-loss are used as loss functions that evaluate each grid cell independently. These loss functions are already used in most OGM prediction methods (see table 7), therefore those loss function have been proven to provide acceptable training results. The SSIM and IS are chosen as loss functions that consider dependencies between grid cells. Both functions are differentiable. Table 5 shows that these functions meet al criteria set for metrics. It is therefore expected they also function well as loss functions. Furthermore, the SSIM is used as loss function by [24], which proves that this loss function provides acceptable training results.

8.4.4 Deep Learning Network Architecture

First, Lange's [55] AAConvLSTM network architecture will be used since it scored highest compared to the other PredNet methods on the Waymo [67] dataset (see table 8). Moreover, the code for Lange's [55] network is available online. Also, the DST method is chosen to generate the OGMs, which Lange also uses in their research. This will make it easier to fit the OGM's format to Lange's architecture. Lange's network originally uses a L1-loss for training. Second, Mohajerin's [24] network architecture will be used to compare whether the effects of different loss functions are consistent between different network architectures. Mohajerin [24] originally uses the SSIM as a loss function in its network.

8.4.5 Metrics

The MSE, IS, and SSIM are used as metrics to evaluate the results of the predictions. The MSE is chosen because it has been used most often by the other methods (see table 7). The IS and SSIM are chosen because they both met all metrics criteria (see table 5).

8.4.6 Experiments

First, the Waymo [67] dataset is used to generate OGM sequences using the DST method. Second, the deep learning networks are adjusted, if necessary, to fit the OGM sequence format as input. Third, each network is trained multiple times on the Waymo [67] dataset. Each time a different loss function is chosen to train the network. Fourth, the results of each trained network's predictions are compared using the chosen metrics. Based on these comparisons, the research questions can be answered.

8.5 Planning

This planning gives a monthly overview about when this research proposal will be executed.

June Analysis of the Waymo [67] dataset.

July Generation of OGM sequences from the dataset.

August Preparation the deep learning networks for training.

September Training the deep learning networks.

October Analysis of the results and writing thesis.

November Writing thesis.

December Graduation.

A Appendix A

Source	Year	Goal Type	Observer	Actors	Method	Cue type	Environment representation	Future Work
[89]	2009	TP	SO	Ps	MDP	APS	DOGMa	ISC, MISC
[22]	2013	TP, MM	DO	Ps	GPDM, PHTM, KF, IMM KF	SI	GPCS	ED, MISC
[90]	2013	TP	DO	Ps	EKF, IMM	APS	GPCS	
[91]	2014	TP, ISC	DO	Ps	DBN, SLDS	APS, Other	3DWC	IEC, MISC
[92]	2014	TP, MM	SO	Ps	GPDM, PHTM, KF, IMM KF	Other	3DWC	ED
[93]	2015	TP	DO	Vs	RRT+GMM	APS, CMI	GPCS+Image	
[94]	2015	TP, ISC	SO	Ps	GP	APS	GPCS	
[95]	2016	TP, ISC	SO	Ps	Social-LSTM	APS	GPCS	ED, IEC
[96]	2016	TP, DL	SO	Ps	CNN	SI, ACS, Other	Image	
[97]	2016	TP	SO	RUs	ASNSC	APS	GPCS, Raster-GPCS	MISC
[98]	2017	TP, DL	DO	Vs	LSTM	APS	OGM	
[14]	2017	TP, IEC, MM	DO	Cs	LDS	APS, RT	GPCS	IEC
[18]	2017	TP, IEC, DL	SO	Ps	SSCN, CNN	CL, SI	GPCS+Image	ED
[99]	2017	TP, ISC, IEC	DO	Rus	IOC RNN	APS, CL, Other	GPCS	ED
[62]	2018	TP, ISC, IEC, E2E, DL, UA	DO, SO	RUs	RNN	OGM	DOGMa	ED
[100]	2018	TP, DL	DO	Vs	CNN, FC, LSTM	SI	Raster-Image	
[101]	2018	TP	SO	Ps	MLP, RNN, TCN	APS	GPCS	ISC
[102]	2018	TP, DL	DO	Vs	LSTM	APS, SI, Other	Raster-Image	
[103]	2018	TP, ISC, DL	SO	Vs	LSTM	APS	GPCS	IEC
[104]	2018	TP, IEC	SO	Ps	ASNSC	APS, Other	GPCS, Raster-GPCS	ISC, IEC
[105]	2018	TP, ISC, DL	SO	Ps	RNN, GAN	APS	GPCS	
[15]	2018	TP, ISC, IEC	SO	Ps	CNN, LSTM	APS, OGM	GPCS	MISC
[17]	2018	TP, ISC, IEC, DL	SO	Ps	LSTM	SI	GPCS+Image	ED, ISC
[106]	2018	TP, DL	SO	Ps	CNN	APS	GPCS	ISC, MISC
[107]	2018	TP, MM, ISC	SO	Ps	CNN	APS, TL	GPCS+Image	IEC, MISC
[88]	2018	TP, ISC, IEC, E2E, DL	DO	Vs	CNN-Transformer	Other	GPCS, 3D-OGM	ED, IEC
[10]	2018	TP, MM, E2E, DL	DO	Ps	ANN, LSTM	SI, ACS	GPCS	ED, IEC, MISC
[79]	2018	TP, ISC, IEC, E2E, DL, MM, UA	DO	RUs	CNN	OGM	OGM	MISC
[20]	2019	TP, MM, DL, UA	DO	RUs	CNN	ACS, SI	DOGMa	
[19]	2019	TP, UA, MM	DO	Vs	VGG-FC, GMM	APS, SI, Other	GPCS	
[21]	2019	TP, MM	DO	Vs	RNN, CNN	APS, SI	GPCS	ED
[108]	2019	TP, ISC, DL	DO	RUs	LSTM	APS, SI	GPCS	ED, IEC
[16]	2019	TP, ISC, IEC, DL	SO	Ps	LSTM, GAN	APS, SI	GPCS+Image	
[109]	2019	TP	DO, SO	RUs	LSTM-CNN	APS, Other	3DWC	MISC
[13]	2019	TP, DL	DO	VRUs	RNN, LSTM, GRU	APS, CL, ACS, Other	Images	
[110]	2019	TP, DL	SO	VRUs	B-LSTM, IRL	APS, SI	GPCS+Image	IEC, ISC
[68]	2019	TP, ISC, IEC, E2E, DL, UA	DO	RUs	CNN, LSTM	OGM	OGM	ED, MISC
[80]	2019	TP, ISC, IEC, E2E, DL, MM, UA	DO	RUs	CNN, LSTM	OGM	OGM	MISC
[24]	2019	TP, ISC, IEC, E2E, DL, UA	DO	RUs	CNN, LSTM	OGM	OGM	ED, MISC
[12]	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	
[111]	2020	TP, DL	SO	Ps	GRU, LSTM	APS	GPCS	
[87]	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	
[23]	2020	TP, MM, ISC	SO	Ps	Social-VRNN, LSTM, CNN	APS, OGM	GPCS+Image, DOGMa	MISC
[55]	2020	TP, ISC, IEC, E2E, DL, UA	DO	RUs	CNN, LSTM	OGM	OGM	MISC
[69]	2020	TP, ISC, IEC, E2E, DL, UA	DO	RUs	CNN, LSTM	OGM	OGM	MISC
[49]	2020	TP, ISC, IEC, E2E, DL	DO	RUs	CNN	OGM	DOGMa	

Table 9: 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).

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] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *2017 International Conference on Engineering and Technology (ICET)*, pp. 1–6, Ieee, 2017.

[10] 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.

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

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

[13] 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.

[14] 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.

[15] 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.

[16] 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.

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

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

[19] 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.

[20] 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.

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

[22] 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.

[23] 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.

[24] 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.

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

[26] 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.

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

[28] 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.

[29] 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.

[30] 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.

[31] 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.

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

[33] 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.

[34] 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.

[35] 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.

[36] 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.

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

[38] 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.

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

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

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

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

[43] 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.

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

[45] 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.

[46] 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.

[47] 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.

[48] 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.

[49] 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.

[50] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan, "A survey on bias and fairness in machine learning," *arXiv preprint arXiv:1908.09635*, 2019.

[51] T. Nordfjærn, S. H. Jørgensen, and T. Rundmo, "An investigation of driver attitudes and behaviour in rural and urban areas in norway," *Safety science*, vol. 48, no. 3, pp. 348–356, 2010.

[52] T. Nordfjærn, Ö. Şimşekoğlu, and T. Rundmo, "Culture related to road traffic safety: a comparison of eight countries using two conceptualizations of culture," *Accident Analysis & Prevention*, vol. 62, pp. 319–328, 2014.

[53] A. Torralba and A. A. Efros, "Unbiased look at dataset bias," in *CVPR 2011*, pp. 1521–1528, IEEE, 2011.

[54] J. Y. Chen and J. E. Thropp, "Review of low frame rate effects on human performance," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 37, no. 6, pp. 1063–1076, 2007.

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

[56] 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.

[57] 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.

[58] 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.

[59] C. Godard, O. Mac Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 270–279, 2017.

[60] 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.

[61] 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.

[62] 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.

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

[64] 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.

[65] 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.

[66] 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.

[67] 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.

[68] 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.

[69] 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.

[70] 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.

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

[72] 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.

[73] S. Annell, A. Gratner, and L. Svensson, "Probabilistic collision estimation system for autonomous vehicles," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 473–478, IEEE, 2016.

[74] H. Wang, Y. Huang, A. Khajepour, Y. Zhang, Y. Rasekhipour, and D. Cao, "Crash mitigation in motion planning for autonomous vehicles," *IEEE transactions on intelligent transportation systems*, vol. 20, no. 9, pp. 3313–3323, 2019.

[75] Z. Wang and A. C. Bovik, "Mean squared error: Love it or leave it? a new look at signal fidelity measures," *IEEE signal processing magazine*, vol. 26, no. 1, pp. 98–117, 2009.

[76] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

[77] A. Birk, "Learning geometric concepts with an evolutionary algorithm," *environment*, vol. 4, p. 3, 1996.

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

[79] 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.

[80] 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.

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

[82] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo, "Convolutional lstm network: A machine learning approach for precipitation nowcasting," *arXiv preprint arXiv:1506.04214*, 2015.

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

[84] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *International conference on machine learning*, pp. 2048–2057, PMLR, 2015.

[85] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proceedings of the IEEE international conference on computer vision*, pp. 1520–1528, 2015.

[86] P. Ondruska and I. Posner, "Deep tracking: Seeing beyond seeing using recurrent neural networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, 2016.

[87] 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.

[88] 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.

[89] 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.

[90] 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.

[91] 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.

[92] 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.

[93] 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.

[94] 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.

[95] 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.

[96] 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.

[97] 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.

[98] 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.

[99] 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.

[100] 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*.

[101] 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.

[102] 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.

[103] 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.

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

[105] 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.

[106] 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.

[107] 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.

[108] 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.

[109] 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.

[110] 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.

[111] 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.