Mid-term status report on KISSaF: AI-based situation interpretation for automated driving

Prof. Dr. Anne Stockem Novo[a,b], Dr. Marco Stolpe[c], M. Sc. Christopher Diehl[d], M. Sc. Timo Osterburg[d], Univ.-Prof. Dr.-Ing. Prof. h.c. Dr. h.c. Torsten Bertram[d], Dr. Vijay Parsi[e], M. Sc. Nils Murzyn[e], Dr. Firas Mualla[e], Dr. Georg Schneider[e], and B. Sc. Philipp Töws[f]

[a]ZF Automotive Germany GmbH, Freiligrathstraße, 45881 Gelsenkirchen, Germany

[b]Institute of Computer Science, Ruhr West University of Applied Sciences, Duisburger Str. 100, 45479 Mülheim an der Ruhr, Germany

[c]ZF Automotive Germany GmbH, Hansaallee 190, 40547 Düsseldorf, Germany

[d]Institute of Control Theory and Systems Engineering, TU Dortmund University, Otto-Hahn-Straße 8, 44227 Dortmund, Germany

[e]ZF Friedrichshafen AG, Uni-Campus Nord D5 2, 66123 Saarbrücken, Germany

[f]INGgreen GmbH, Carl-Spaeter-Straße 76, 56070 Koblenz, Germany

Abstract

KISSaF is a publicly funded project with four project partners from industry and academia. The aim of project KISSaF is the development of a robust scene prediction model for automated driving. State-of-the-art Deep Learning methods are used for a complete and reliable forecasting of the traffic scene with large time horizons. The underlying environment modeling uses a graph-based representation of the scene. A prototype vehicle has been built-up for data recording. This data is the central part for model development, improvement and testing. A framework is currently setup for a scenario-based test approach and performance can be judged under realistic conditions with integrated maneuver planning.

1 Introduction

The acronym KISSaF stands for "KI-basierte Situationsinterpretation für das autonome Fahren", meaning *AI-based scene prediction for automated driving*. It is a publicly funded project by the Bundesministerium für Wirtschaft und Klimaschutz (BMWi) running from January 2021 until June 2023. The project consortium consists of four partners from industry and academia based in Germany. The overall product goal is the development of a robust situation prediction of traffic scenes.

Automated driving is a key technology for the automotive market. Despite of recent advances, there still exist some hindering factors for a series market production of self-driving cars with level 3 or higher. One of the central difficulties is to anticipate the development of a scene for all kind of scenarios and to guarantee a fail-safe handling of the vehicle. Especially in dense traffic or scenarios which involve different stakeholders, e.g. bicycles, pedestrians, busses and passenger cars, this is a challenging task. For instance, the number of traffic participants increases complexity. Humans tend to make sudden decisions, therefore the participation of bicycles or pedestrians introduces different time scales. Motorcyclists or heavy vehicles, like busses or trucks, are underrepresented, often leading to corner case situations. These problems are currently being addressed within this project by massively collecting realistic driving situations (> 200.000 km) with a wide variation of different scenarios. The model improvement brings together state-of-the-art models with very low prediction errors for several seconds ahead.

Each project partner takes the lead for one of four aspects of the final product: The central part of the project is the build-up of a measurement vehicle which is used for data collection. The sensor data is fused and converted to a graph-based environment model. State-of-the-art deep learning techniques are applied in order to model the evolution of the environment. This model is integrated with trajectory planning where it is then evaluated towards different key performance indicators.

The subproject details and status are given in Section 2. In Section 3 the next steps for reaching the project goal are described.

2 Subproject description

2.1 Build-up of measurement vehicle

The measurement vehicle is an Opel Insignia (Fig. 1) equipped with multiple sensors: 4 short-range HELLA corner radars, a forward-facing mid-range radar (ZF-inhouse product MRGen21), a forward-facing MobilEye camera and an IBEO Lidar system consisting of six components for a 360° perception. An additional Lidar-based lane tracker is mounted on the roof, serving mainly for exact positioning relative to the driving lanes. The sensor system is complemented with global positioning data (GPS) and for a confined area also with high-definition map (HD-map) data generated by the company *3D mapping solutions*.

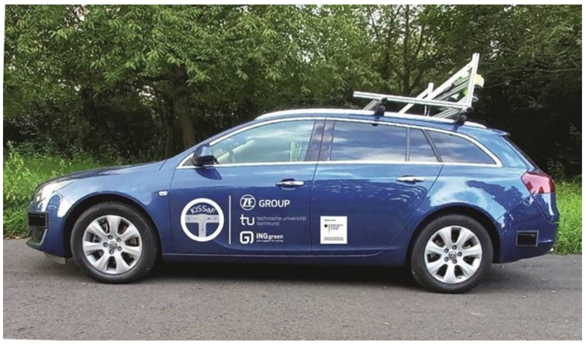

Figure 1 The KISSaF measurement vehicle equipped with radars, camera and lidars.

The vehicle setup and the formal admission process by the German association for technical inspection (TÜV) has been completed. First data taking campaigns have happened already and the data is currently being prepared to be used for model development.

The data processing step covers mainly a labelling of the meta data with information about the specific event. The entire dataset is managed within a Microsoft Azure project.

2.2 Environment modelling

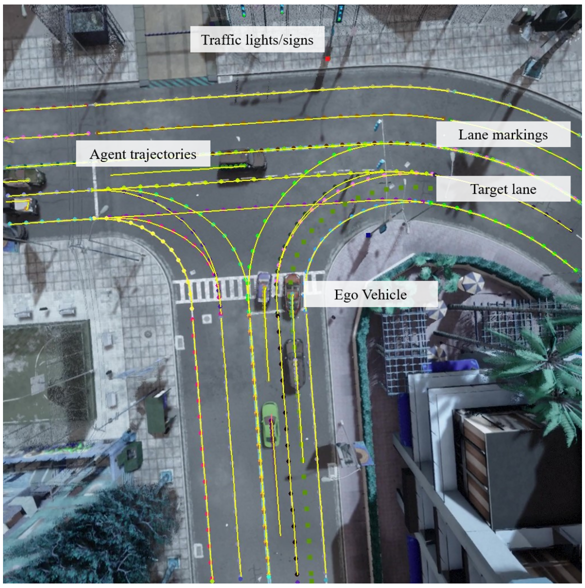

Figure 2 Example of the KISSaF graph-based environment model in Carla [5].

2.2.1 Environment representation

This section describes the representation of the environment that is used as an input to the neural network. In recent literature, vectorized representations are increasingly used over rasterized ones [1], [2], [3], [4]. Besides the better data efficiency, they pose lower requirements for the neural networks processing the data, while increasing the accuracy of following tasks such as prediction [1]. Following current developments, the environment of the ego vehicle is represented as a graph, as shown in figure 2. First, the object lists created by the different sensors are combined into a global list by using high-level fusion and tracking [6]. The global object list and HD-map information is then used to create the environment graph. The nodes of the graph represent points in the 3D world holding further semantic information. Nodes are organized in polylines, representing gathered information about objects. Connections between nodes encode further relationships such as predecessor, successor and lane assignments. The overall environment model thus is a list of polylines.

Encoding semantic information in nodes enables the representation of different data interpretations in the same format. A polyline represents a trajectory for vehicles, while it also represents the course of lane markings as detected in the most recent timestamp. Besides agent trajectories and lane markings, the environment model contains information about traffic lights and signs. Beyond the perceived environment, high-level decisions on lane changes are represented by adding a target lane to the environment model, enabling the prediction module to condition on these high-level decisions.

Nodes are represented by their state $s_n = [x_n, y_n, \psi_n, v_{\mathrm{x},n}, v_{\mathrm{y},n}, a_{\mathrm{x},n}, a_{\mathrm{y},n}, l_n, w_n, I_{\mathrm{class},n}, p_{\mathrm{exist},n}, T_n, \mathbf{s}_{\mathrm{c},n}]$, with x_n, y_n as x and y coordinate, ψ the heading, v_x, v_y, a_x, a_y the velocity and acceleration in x and y direction, length l, width w, a class identifier I_class, existence probability p_exist, the perception timestamp T and class specific semantic information $\mathbf{s}_\mathrm{c}$ of node n. Since only the x and y coordinates are used in the node state, the 3D points are projected onto the 2D ground plane. The lane marking class provides additional class specific semantic information about the marking type, legal lane changes, the marking colour and marking width.

The environment model has interfaces to the Carla simulator [5] and the data format of the KISSaF data. Using the graph environment model as an interface, the prediction network can operate using both simulated and real-world data. This also allows a closed-loop evaluation of all modules up to the planning module in simulation and in the test vehicle.

2.2.2 Weather recognition

Apart from traffic participants and traffic-related objects, the environmental model also includes environmental conditions, which consist of daytime and the current weather. Most weather effects such as rain, fog and snow can drastically impair visibility, whereas rain and snow additionally reduce grip and increase the difficulty of driving. Therefore, knowledge about the current weather conditions can improve both the estimation of the quality of the ego vehicle's perception and the prediction of other traffic participants' behavior. For these reasons, recognition of the current weather conditions is one of the goals of KISSaF.

Several approaches for weather recognition based on data from Lidar [7], [8] or camera [9], [10] already exist, however nearly all of them focus on one sensor only. Furthermore, usually only single frames without their temporally related neighbor frames are considered. To combine all of this data for weather recognition, a hybrid fusion framework is developed, which fuses the data both on a low- and mid-level. The low-level fusion enables the network to directly correlate raw data, whereas the mid-level fusion is necessary to combine the information represented in different modalities, such as images and point clouds. On a separate level, temporal relationships can be exploited either by prior fusion or through the use of a recurrent neural network such as long short-term memory [11]. The resulting estimation of weather conditions is integrated into the environment model, which then provides it to the scene prediction.

2.3 Scene prediction modeling

In order to predict the movement of traffic agents, one of the state-of the-art approaches is to represent the whole scene as 1-dimensional data. Therefore, vectors are utilized to represent map data and agent movement. The architecture utilized to process this kind of vector information follows the architecture as introduced by Gao et al. [1]. Every vector is assigned a lane ID if it contains map information or an agent ID if it contains dynamic information. All vectors which are assigned to the same ID are subject to an PointNet-like encoder network [12]. This encoder network computes a so-called polyline subgraph which is a node of the global interaction graph. This global graph is processed by a transformer architecture which updates every subgraph with all other subgraphs. The aim of this operation is to model interaction between all elements of a scene. These updated subgraphs are subject to a decoder (predictor), which generates displacement in x and y direction. An overview of this model is presented in Fig. 3.

2.3.1 Coupled prediction and planning

Traditional system architectures neglect the interaction between the prediction and planning module of an automated vehicle. This could result in a conservative, non-human-like driving maneuver. In this project, the prediction and planning should be coupled. Therefore, the developed conditional prediction model answers *What If* questions.

A first approach is proposed in the work of Diehl et al. [13]. It solves the prediction, planning, and control problem jointly in an interpretable learning-based fashion. A stochastic dynamics model conditioned on the action of the automated vehicle is used for a one-step prediction, whereas the sampling of actions is guided by a learned behavior cloned policy. A conditional variational autoencoder [14] models different future scene evolutions by sampling of a latent variable.

Future work will condition the graph-based prediction on the high-level route of the automated vehicle. The conditioned interaction-aware forecast of the future joint distribution is then used during planning.

2.4 Evaluation

It is planned to evaluate the developed scene prediction in two ways. First, the differences between predicted and real trajectories of all vehicles surrounding the ego vehicle are calculated using common metrics. This will allow for comparison with other approaches found in the existing literature on trajectory prediction. Second, the scene prediction will be evaluated in the context of particular driving functions, e.g. the automated lane change. Such driving functions may require the definition of specific key performance indicators (KPIs).

In both aforementioned cases, metrics can be evaluated in the context of different scenarios. Since the number of scenarios one could think of is too large for an explicit listing, the approach taken in the project is to describe scenes by a limited set of properties. Scenarios can then be described by a combination of conditions on the properties and a standard set of logical operators. Furthermore, the description of scenes by properties allows to correlate them with good or bad KPIs, giving insights into which kind of scenarios are particularly challenging, or to identify scenarios which are underrepresented, like corner cases.

The following subsections present the used or newly defined KPIs in more detail and give examples of the extracted scene properties and their application in the evaluation of scenarios.

2.4.1 Generic Metrics

Average Displacement Error (ADE). The average displacement error (ADE) can be seen as the L_2-norm between all predicted points and the true points over the trajectories of all traffic participants in a scene:

$$\text{ADE} = \frac{\sum_{i=1}^{n} \sum_{t=T_{obs}+1}^{T_{pred}} \sqrt{(\hat{x}_i^t - x_i^t)^2 + (\hat{y}_i^t - y_i^t)^2}}{n\left(T_{pred} - (T_{obs}+1)\right)} \tag{1}$$

Here, n is the number of trajectories (i.e. traffic participants), T_{obs} the first point of the trajectory which we can still observe (i.e. the timepoint at which the prediction started), T_{pred} the final predicted trajectory point, $\hat{x}_i^t$ and x_i^t the predicted and real longitudinal coordinates of the t's trajectory point for trajectory i, and $\hat{y}_i^t$ and y_i^t the predicted and real lateral coordinates of the t's trajectory point for trajectory i.

Final Displacement Error (FDE). The final displacement error (FDE) measures the L_2-norm only between the predicted final destination and the true final destination at timepoint T_{pred} over the trajectories of all traffic participants in the scene:

$$\text{FDE} = \frac{\sum_{i=1}^{n} \sqrt{(\hat{x}_i^t - x_i^t)^2 + (\hat{y}_i^t - y_i^t)^2}}{n} \tag{2}$$

Probabilistic Error Measures. In the case of predicting multiple possible futures, also known as predicting multiple modes, each possible future has assigned some probability. The error measure minADE calculates the minimum ADE over the k most probable futures, while the error measure minFDE calculates the minimum FDE over

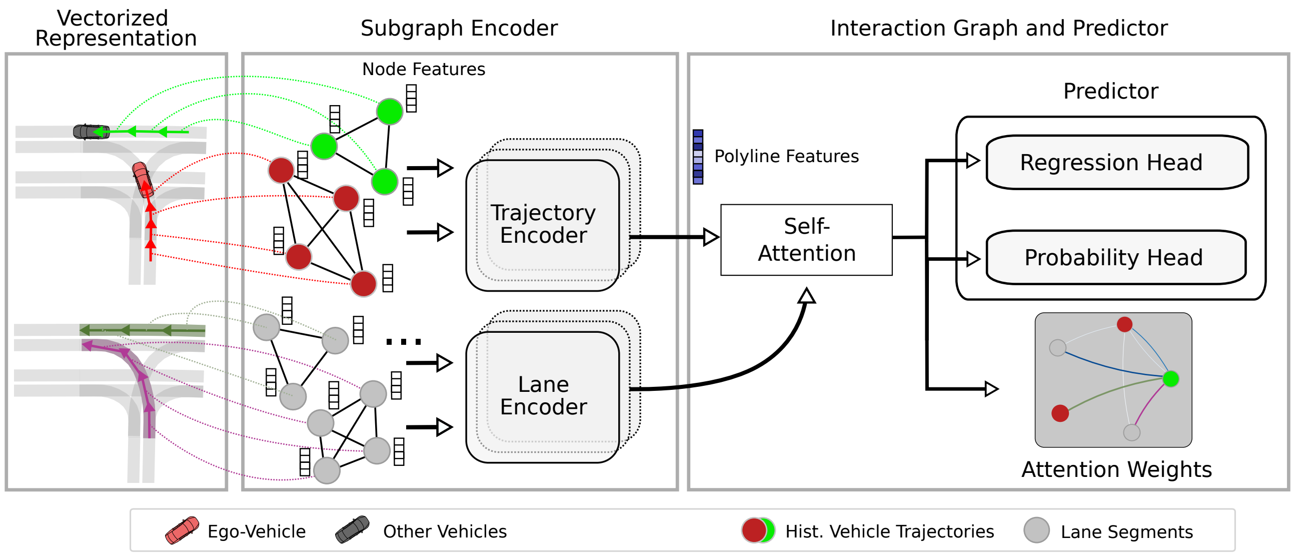

Figure 3 The neural network architecture. The past trajectories of all agents and the lane information is presented to their respective subgraph encoders in a vectorized format. The predictor extracts the trajectories and probabilities for the trajectories from the polyline features via a Multi Layer Perceptron (MLP). Additionally, the networks estimates the attention weights for the individual nodes.

the k most probable futures.

Miss Rate. A miss occurs if the maximum L_2-norm between the predicted trajectory points and the ground truth is larger than some threshold value for the k most likely predictions. The miss rate is the proportion of misses over all agents.

2.4.2 KPIs for Driving Functions

Results of the scene prediction can be used in several different automated driving functions in ADAS and AD systems. In the following, examples for the most basic kinds of automated longitudinal and lateral movements are given.

Automated Braking or Acceleration. Automated lane keeping involves reacting to other traffic participants, mainly in front of the ego vehicle. As long as the set speed is not reached and all safety distances can be kept, the ego vehicle may accelerate. Braking becomes necessary whenever the vehicle in front of the ego vehicle (also called lead vehicle) decelerates or vehicles are changing onto the ego lane (known as cut-in).

Especially cut-ins are potentially dangerous, since other vehicles might not keep the safety distance when changing lanes. The earlier cut-ins can be anticipated, the earlier the ego vehicle can reduce its speed to avoid strong braking or collisions. Similarly, strong braking might be avoided if braking of the lead vehicle can be predicted well.

There are different types of metrics which will be used to evaluate the scene prediction's performance in the aforementioned contexts. First, the deviation between the predicted and real acceleration or deceleration of vehicles can be measured, e.g. with the L_2-norm. Then, it can be assessed how many braking, acceleration or cut-in events are detected correctly (or not). Here, it is also possible to calculate true and false positives and negatives. Furthermore, it will be measured how early an event could be predicted, especially in comparison to a simple version based on constant velocity assumptions or a purely reactive version which does not use any kind of prediction.

Automated Lane Change. For the automated lane change, it is important to predict which gaps on the chosen target lane are feasible for a merge-in or not. Feasible means that once the ego vehicle has reached the gap, certain safety conditions will hold long enough such that the merge-in can be performed. If several gaps seems feasible, one might be chosen at trigger time (e.g. once the driver has activated the turn-indicator), for a later merge-in.

There are several ways the automated lane change can be realized. Without going into further details, metrics used in the project will measure how often the feasibility of gaps was correctly predicted or not, at trigger time and for different time horizons.

2.4.3 Evaluation in Scenarios

The scene prediction not only takes into account the historic movement patterns of other vehicles, but also environmental conditions like traffic signs, lane topology and markings, daytime or weather. The combination of all such factors results in a huge amount of possible scenarios which might potentially result in different prediction performance and would need to be tested. Unfortunately, it cannot be known in advance which factors have the biggest influence. It is thus also difficult to explicitly design or list scenarios which should be tested. Therefore, in the project we take a different approach, and describe scenes by a limited set of properties. These can then be queried for and correlated with performance metrics.

Properties of scenes would include time related information, environmental conditions (like weather or the presence of certain traffic signs, lane markers, ramps, exits, etc.), position and speed of other traffic participants and their classification (like pedestrian, bicycle, vehicle or truck), traffic density, and information about lane changes and gaps.

Storing scene properties together with performance metrics in some database allows for querying scenes fulfilling certain conditions and aggregating metrics related to the queried scenes. For instance, we might be interested in the average ADE over all scenes which involve dense traffic and vehicles with an average speed of 100 km/h. Furthermore, the correlation of certain properties with performance metrics can be determined using statistical methods. Finally, it becomes possible to identify scenarios which are underrepresented, e.g. using density-based or subspace clustering.

Correlation analysis and the identification of corner cases will be performed once first scene prediction results on the collected data are available.

3 Future steps

During the second half of the project, massive data taking campaigns will be conducted for recording real data. This will be done on German highways, rural roads and cities in order to guarantee a wide variety of different traffic scenes. Meta data will be extracted for labeling purposes and efficient data selection. Furthermore, the recorded data will be transformed into a format which is required by the model and will be stored in a database.

The vehicle data will then be used for modelling and evaluation: The environment representation model will be applied to the different sensor data as described in Sec. 2.2. For a limited area, HD map data will be added to the fusion model. This environment representation will then be used for scene prediction (Sec. 2.3). Finally, the best model generated on our own dataset will be compared against reference models with the approach detailed in Sec. 2.4.

4 Literature

[1] Gao, Jiyang; Sun, Chen; Zhao, Hang; Shen, Yi; Anguelov, Dragomir; Li, Congcong; Schmid, Cordelia: VectorNet: Encoding HD Maps and Agent Dynamics from Vectorized Representation, *arXiv 2005.04259*, 2020.

[2] Liang, M.; Yang, B.; Hu, R.; Chen, Y.; Liao, R.; Feng, S.; Urtasun, R.: Learning lane graph representations for motion forecasting. *Conference on Computer Vision*, 2020.

[3] Luo, C.; Sun, L.; Dabiri, D.; Yuille, A.: Probabilistic multi-modal trajectory prediction with lane attention for autonomous vehicles. *International Conference on Intelligent Robots and Systems*, 2020.

[4] Zeng, W.; Liang, M.; Liao, R.; Urtasun, R.: LaneR-CNN: Distributed Representations for Graph-Centric Motion Forecasting. *International Conference on Intelligent Robots and Systems*, 2021.

[5] Dosovitskiy, A.; Ros, G.; Codevilla, F.; Lopez, A.; Koltun, V.: CARLA: An Open Urban Driving Simulator. *Conference on Robot Learning*, 2017.

[6] Aeberhard, M.: Object-level fusion for surround environment perception in automated driving applications. *VDI Verlag*, 2017.

[7] Heinzler, R.; Schindler, P.; Seekircher, J.; Ritter, W.; Stork, W.: Weather influence and classification with automotive lidar sensors, *IEEE Intelligent Vehicles Symposium, Proceedings*, 2019.

[8] Sebastian, G.; Vattem, T.; Lukic, L.; Christian, B.; Schumann, T.: RangeWeatherNet for LiDAR-only weather and road condition classification, *IEEE Intelligent Vehicles Symposium* 2021.

[9] Kondapalli, C. P. T.; Vaibhav, V.; Konda, K. R.; Praveen, K.; Kondoju, B.: Real-time rain severity detection for autonomous driving applications, *IEEE Intelligent Vehicles Symposium*, 2021.

[10] Ibrahim, M. R.; Haworth, J.; Cheng, T: Weathernet: Recognising weather and visual conditions from street-level images using deep residual learning, *International Journal of Geo-Information*, 2019.

[11] Hochreiter, S.; Schmidhuber, J.: Long Short-term Memory, *Neural Computation*, 1997.

[12] Qi , C. R.; Su, H.; Mo, K.; Guibas, L. J.: PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation, *Conference on Computer Vision and Pattern Recognition*, 2017 .

[13] Diehl, C.; Sievernich, T.; Krüger, M.; Hoffmann, F.; Bertram, T.: UMBRELLA: Uncertainty-Aware Model-Based Offline Reinforcement Learning Leveraging Planning, *Conference on Neural Information Processing, Machine Learning for Autonomous Driving Workshop*, 2021.

[14] Kingma, D. P.; Welling, Max: Auto-Encoding Variational Bayes, *International Conference on Learning Representations*, 2014.

This research was supported by the Federal Ministry for Economic Affairs and Energy on the basis of a decision by the German Bundestag.