Generating Compact Models for Traffic Scenarios to Estimate Driver Behavior Using Semantic Reasoning

Ilya Dianov, Karinne Ramirez-Amaro and Gordon Cheng

Abstract— **Driving through a constantly changing environment is one of the main challenges of autonomous driving. To navigate successfully, the vehicle should be able to handle a variety of possible situations on the road by constantly analyzing the traffic environment and determine which objects might influence its current behavior. This paper presents an artificial intelligence method to improve the perception and situation awareness of autonomous vehicles by detecting and extracting meaningful information from different traffic scenarios, and inferring the correct driving behavior for each of them. Our method uses a state of the art technique based on semantic reasoning previously used for recognizing human activities in cooking scenarios. This algorithm has been adapted and extended to the automotive domain by introducing new object properties such as *ObjectInFront*, *ObjectActedOn*, *MoveForward*, *Turn*. The main advantage of our proposed method is its adaptability to different mobile domains without any additional training. First, our system is trained on traffic situations. The obtained semantic models are later used to autonomously navigate a mobile robot in an indoor environment by utilizing the acquired knowledge and inference from the automotive domain. The results show that the overall positive classification rate for traffic scenarios recognition is 90.14% of the cases. In addition, the average processing and behavior generation time for the implemented system is 0.177 seconds, which allows the mobile robot to react online to the newly encountered situations.**

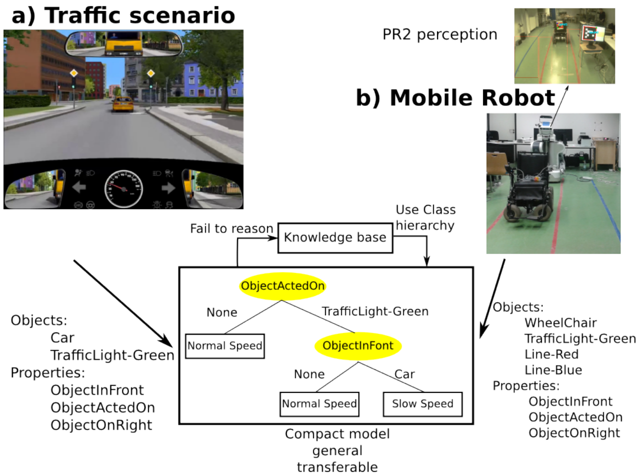

Fig. 1: General overview of our system which shows two examples where we reuse the learned semantic models: a) shows an example of the new traffic scenario and b) depicts an example of the domain transference of learned model[1].

I. INTRODUCTION

Recent developments in autonomous driving show that autonomous vehicles can bring a lot of benefits to our society e.g. reduction of traffic incidents, increased mobility for the elderly and disabled people, more efficient traffic flow, reduction of fuel consumption and many more [1]. In order to achieve that, these vehicles have to overcome many problems such as moving in a dynamic environment, processing vast amount of data from different sensors, handling driving rules, path planning and collision avoidance [2]. One of the main challenges is understanding the encountered traffic situations to estimate the proper driving behavior by considering the meaningful traffic participants and relations between them. However, finding this meaningful information represents another set of challenges such as complexity of the perceived environment, which leads to problems of having partial observable information. Typically, to solve this problem it is necessary to implement a sophisticated method which can recognize, analyze and extract contextual information about each scenario, learn and use it to determine vehicle behaviors when this scenario is encountered again [3]. The extracted contextual data is represented as spatio-temporal relationships between the ego vehicle and traffic environment [4]. This context information can be obtained from the generation of dynamic maps [5] which will enhance the tracking of the objects around the vehicle. However, there are a vast amount of objects which can influence the vehicle motion depending on their behavior, and learning all possible traffic scenarios involving all these different objects would be computationally expensive in terms of execution and memory. Thus, semantic representation and knowledge methods can greatly improve such systems.

The goal of this paper is to create a simple and general ontology model which can be used to describe different scenarios in mobile domains. For this reason, we propose a method that can automatically recognize and extract spatial relationships from such scenarios using a simple perception system and create a reasoning mechanism which can infer appropriate vehicle behavior for each situation. Fig. 1 shows an example of our proposed system, exemplifying how the model of the same behavior can be applied to two different scenarios. First, the compact general model was created for the input scenario involving the yellow car which is not moving in front of the vehicle and the traffic light is showing a green light. To avoid collision the vehicle has to slow

Faculty of Electrical Engineering, Institute for Cognitive Systems, Technical University of Munich, Germany ilya.dianov@tum.de, karinne.ramirez@tum.de and gordon@tum.de

[1] The traffic scenario image is a courtesy of Verlag Heinrich Vogel. www.fahren-lernen.de

down even if the traffic light is green. A similar situation was simulated using the PR2 robot: the monitor represents the traffic light and there is a wheelchair standing in front of the robot path. Using such knowledge our system can detect that both car and wheelchair are mobile entities therefore both will infer the same behavior: slow speed.

This paper is organized as follows: Section II describes related work. Section III introduces the framework consisting of the object properties, semantic rules and the ontology. Section IV shows the application of the generated framework to mobile robot domain, and Section V presents conclusion and an outlook on possible future work.

II. RELATED WORK

There are multiple approaches for analyzing and estimating traffic scenarios. The method presented in [6] models scenarios as a state space, containing information about the vehicle, properties of its surroundings, possible behaviors and trajectories of the vehicles using Dynamic Bayesian Network (DBN) to predict the driver's behavior in its current scenario. A similar approach by Agamennoni et al. [7] introduces feature functions to characterize dynamic relationships between traffic participants to form context models which are used by DBN to predict states of all objects influencing vehicle behaviors. In contrast, [8] introduce evidential grids which utilizes geographic information from digital maps to detect navigable space for the vehicles. However, this approach is used only for obstacle avoidance and do not incorporate any data about the road infrastructure (for example traffic signs). Another method is described in [9] where a tree-like hierarchy of classifiers is introduced. Each node in the hierarchy is predefined and predicts only one certain property of an input object and can activate a child node if a specific output is predicted. In [10] a Bayesian network is used to find impact of traffic situations to each participant and predict their behavior, where participants are represented as nodes connected with conditional distribution functions. However all these approaches require a complicated perception system, large amount of samples for each driving situation and do not extrapolate and exploit semantic relations for prediction use.

Another group of methods for analyzing traffic situations are based on description logic. An ontology can be used to describe road intersections by mapping atomic concepts to specific geometric primitives [11]. Another way of using an ontology is to represent lanes and vehicles moving on them as a graph like network and detect conflicts between traffic participants at the same intersection [12]. To analyze more complex situations an ontology can be represented as a knowledge base with hierarchical structure consisting of atomic concepts and relations between them [13]. This method can be extended to find dependencies in interaction of the traffic participants to infer their likely behavior in current situations [4]. However, the described methods are using manually generated rules for reasoning, which are very specific to each situation. Moreover, their ontology representations are created for driving scenarios and cannot be applied to any other mobile domains without significant changes.

Conversely, our method provides a framework for autonomous learning semantic rules from sensory data which together with the ontology, allows to transfer knowledge from the traffic domain to other mobile domains.

III. SEMANTIC REASONING AND KNOWLEDGE REPRESENTATION

The overview of the created framework is shown in Fig. 2. The contextual information perceived by the vehicle is processed to detect objects and their spatial properties and then stored in the ontology. The reasoning module utilizes knowledge from the ontology to recognise traffic scenarios and uses semantic rules to infer driving behavior with respect to the road context (traffic rules and objects in driving environment). The semantic rules are generated using the decision tree classifier trained on the driving tests obtained from an online driving school[2].

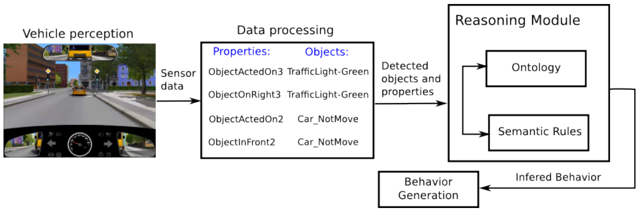

Fig. 2: Framework overview[3].

A. Identification of Object Properties

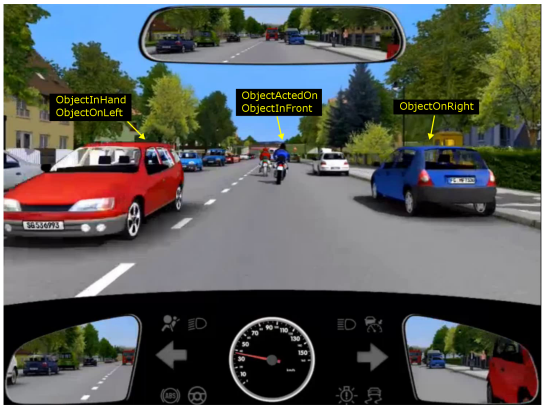

Fig. 3: Example of detecting properties from traffic situation, the input video was obtained from the online driving school[3].

To analyze different driving scenarios the video tests described above were used. Each video was manually annotated, and we found that for each traffic participant influencing the behavior of the vehicle at the current time point there is a set of common properties which are always applicable regardless of the participant type. These properties

[2] www.fahren-lernen.de

[3] The traffic scenario image is a courtesy of Verlag Heinrich Vogel. www.fahren-lernen.de

evaluates spatial relationships between the participants and the ego vehicle. Each traffic participant is represented as an abstract object and the following properties are defined:

1) *ObjectInHand*: the object is very close to the ego vehicle and can cause a collision. In Fig. 3 the red car is considered to be *ObjectInHand*, because if the vehicle would turn left it will crash with that car.
2) *ObjectActedOn*: the object is in the range of interest of the ego vehicle and might require a certain action, but neglecting the execution of this action will not lead to a collision. In Fig. 3 both motorcycles are considered to be *ObjectActedOn* because normally the vehicle should reduce speed to keep proper distance, but even if it maintain current speed there will be no collision until the vehicle would reach the motorcycles (but then they become *ObjectInHand*).

The above properties were inspired by the similar ones defined in [19] and readapted to driving scenarios. At any particular time point the object can have only one of these two properties (either *ObjectInHand* or *ObjectActedOn*) but not both of them. Additionally, we define the following properties:

3) *ObjectInFront*: the object is in front of the current driving path of the ego vehicle. In Fig. 3 both motorcycles are *ObjectInFront* because both of them are moving on the same lane as the ego vehicle.
4) *ObjectOnLeft*: the object is on the left possible path of the ego vehicle. In Fig. 3 the red car is *ObjectOnLeft* because it is moving on the lane which is to the left of the current driving path of the ego vehicle.
5) *ObjectOnRight*: the object is on the right possible path of the ego vehicle. In Fig. 3 the blue car is *ObjectOnRight* because even if there is no separation line, the car is parked on a different lane according to driving rules, and this lane is to the right of the current driving path of the ego vehicle.

For automatic detection of the above *Object Properties* we define the formulas shown in Table I. Where $l = \sqrt{x^2 + z^2}$ is the distance to the object, r_1 and r_2 are distance thresholds, x_{obj}, x_{left} and x_{right} are x coordinates of the Cartesian position of the object , left boundary and right boundary of the driving tube respectively. The framework do not require global positions of the objects or the ego vehicle for property detection and utilize only their local positions with respect to the ego vehicle. Fig. 4 shows an example of properties detection, where the blue dot represents the ego vehicle:

- The $obstacle_1$ (green dot) is detected as *ObjectInHand*.
- The $obstacle_3$ (yellow dot) is *ObjectActedOn*.
- The $obstacle_1$ (green dot) is *ObjectInFront*.
- The $obstacle_2$ (purple dot) is *ObjectOnLeft*.
- The $obstacle_3$ (yellow dot) is *ObjectOnRight*.

TABLE I: Definition of *Object Properties*.

Name	Formula
ObjectInHand	$l < r_1$
ObjectActedOn	$r_1 < l < r_2$
ObjectInFront	$x_{left} < x_{obj} < x_{right}$
ObjectOnLeft	$x_{obj} < x_{left}$
ObjectOnRight	$x_{obj} > x_{right}$

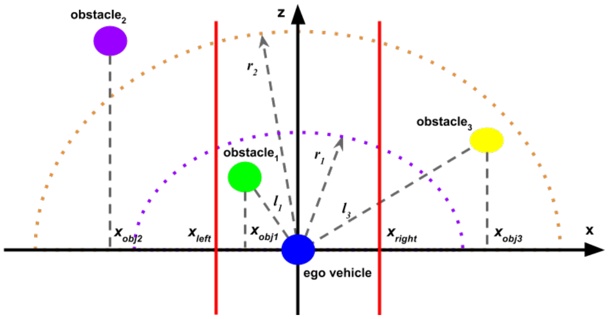

Fig. 4: Graphical Example of the *Object Properties* detection.

In addition, we identified that traffic participants of the same type can have different properties which we called *Instance Properties*. For the data obtained from the video tests we define the following *Instance Properties*:

- MoveToward: the object is moving towards the vehicle.
- MoveForward: the object is moving forward the vehicle.
- NotMove: the object is not moving.
- ChangingLane: the object is changing lane.
- Crossing: the object is crossing the path of the vehicle.
- Turn: the object is turning.

It is always possible to add additional *Instance Properties* to generate a more accurate model of traffic scenarios.

B. Semantic Rules

In order to map the *Object properties* to the ego vehicle driving activities, a decision tree classifier was build similar to [20]. The training data sample contains the current *Object Properties*:

1) *ObjectInHand (None, Something)*
2) *ObjectActedOn (None, Something)*
3) *ObjectInFront (None, Something)*
4) *ObjectOnLeft (None, Something)*
5) *ObjectOnRight (None, Something)*

where *Something* represents an object with certain instance property (for example *Vehicle_MoveToward*) and *None* is used if there is no object with that particular *Object Property*. And a target concept value which describes a current vehicle behavior:

$$Class : VehicleActivity\{NormalSpeed, SlowSpeed, NormalStop, LaneChange\}.$$

Here is an example of a training sample:

$$\{None, Vehicle\_MoveToward, None, Vehicle\_MoveToward, None, NormalSpeed\}.$$

It is possible to have two separate sets of *Object Properties* for different *Classes* in order to describe the situation in more detail. For example, the class *Vehicle* can have the properties:

$$\{ObjectInHand1, ObjectActedOn1, ObjectInFront1,\\ ObjectOnLeft1, ObjectOnRight1\}$$

where 1 indicates that this is a first set of properties. While, the class *Pedestrian* can have a different set of properties:

$$\{ObjectInHand2, ObjectActedOn2, ObjectInFront2,\\ ObjectOnLeft2, ObjectOnRight2\}.$$

where 2 indicates that this is a second set of properties. In this case, the training sample should contain combination of both properties, which means that each ego vehicle activity can be represented by several combinations of *Object properties* (in contrast to [20]). To learn a target concept value from the data samples we trained a decision tree classifier based on the C4.5 algorithm [14]. The information gain is defined as follows:

$$Gain(S,P) = Entropy(S) - \sum_{v \in Values(P)} \frac{|S_v|}{S} Entropy(S_v)$$

where $Values(P)$ is the set of all possible values of the *Object properties*, and $S_v = s \in S|P(s) = v$.

C. Ontology Model

The formal definition of an ontology is "an explicit specification of a conceptualization" [15], in other words it is an unambiguous representation of the knowledge about a certain domain. An ontology usually consists of terms organized in hierarchical structure and relationships between those terms. Our proposed ontology was built as a knowledge base using Knowrob [16] and represented in the Web Ontology Language (OWL) [17]. It consists of the terminological box (*TBox*) and the assertional box (*ABox*) [18] .

1) *Tbox* describes concepts in the ontology, which are usually called *Classes*. Each concept has a set of properties called attributes. Relations between concepts are represented by rules and axioms.
2) *Abox* describes instances of concepts.

The terminology box consists of:

1) *Classes* which represent different types of objects detected from the traffic environment.
2) *Object Properties*: properties described previously, which are common for all classes or for a specific class.
3) *Instance Properties*: these properties are defined for each object, and two objects belonging to same class can have different instance properties.

The assertional box was defined as following:

- *Abox* consists of instances which belong to classes defined in *Tbox*. All perceived objects are contained in *Abox* and if any of the *Object Properties* are held, their instances are created and placed in *Abox*.

To implement the obtained semantic rules in the Reasoning module and connect them with the ontology the Knowrob *Computable Classes* were used. The *Computable* defines a semantic relation between instances of classes representing possible behavior of the ego vehicle and *Object Properties*. For example:

```
if $(o_i = Vehicle)$ & $ObjectInHand2(o_i)$ & $ObjectInFront2(o_i)$ & $(NotMove(o_i))$ then
    $NormalStop$
end if
```

where o_i is an object detected in the traffic environment. The *Computables* were implemented using Prolog, which provides a useful functionality for ontology description, knowledge inference, searching and pattern matching.

IV. EXPERIMENT

To prove that our proposed methodology is working in different mobile domains, first the framework described in Section III was applied to create the semantic rules for traffic scenarios. Next, to test the robustness of our system in the real environment we used it to navigate PR2 mobile robot. The robot environment contains unknown objects and we will demonstrate how without any additional training the PR2 can utilize the system build for traffic scenarios and generate correct behavior by taking advantage of knowledge and inference.

A. Semantic Tree Training

TABLE II: Confusion matrix of the recognized driving behavior.

		Predicted Class			
		NormalSpeed	NormalStop	SlowSpeed	LaneChange
Actual Class	NormalSpeed	94.4%	0	5.6%	0
	NormalStop	0	98.5%	1.5%	0
	SlowSpeed	3.6%	0	96.4%	0
	LaneChange	0	0	0	100%

For training we used 14 video samples of the driving tests (see Section III). Each sample had a length of 15 seconds and contained 150 video frames. The video tests consist of different complex traffic scenarios involving multiple road participants, traffic signs and pedestrians. In the first experiment, each frame of the input video was manually annotated to obtain training samples, containing a sequence of detected objects and its properties as well as the class of the recognized driving behavior. A decision tree was generated in the Weka data mining system [21] and was tested on a 3 new video samples of the driving tests containing previously untrained traffic scenarios. The resulting classification rate was 94.6%, and the confusion matrix is shown in Table II. The partial decision tree is shown on Fig. 5, where each driving situation is identified by a set of specific properties and instances, which allows a compact and general representation.

By analyzing the obtained results, we concluded that segmenting each frame of the video was redundant, and instead it is better to extract only 5 samples for each traffic situation recognized in the input video. To prove this theory, a second experiment was conducted using the same video samples for training and testing as before, and the resulting classification

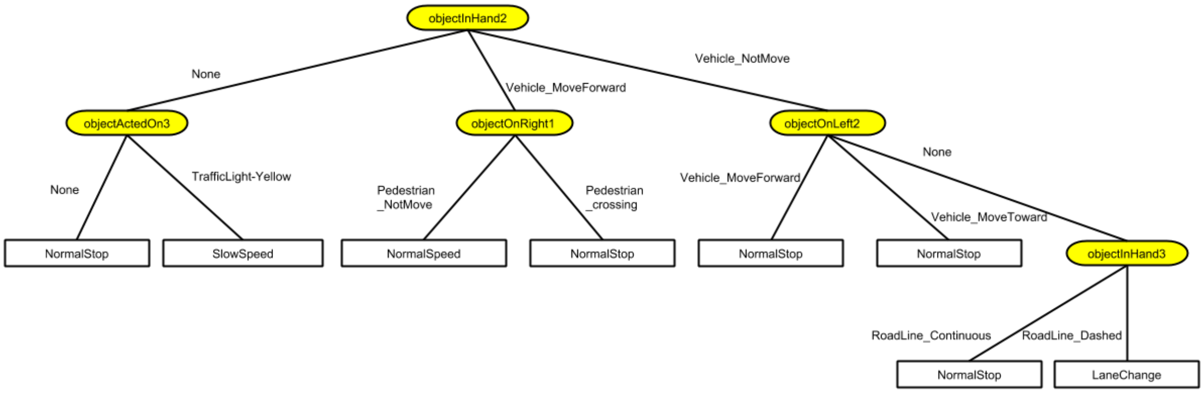

Fig. 5: Partial decision tree obtained from the first experiment instance.

rate was 90.14%. The confusion matrix generated for the new setup is shown in Table III. The resulting decision tree remained the same as the one obtained in the first experiment, which clearly indicates that our proposed method do not require large amount of data for training.

TABLE III: Confusion matrix for the second experiment with less training samples.

		Predicted Class			
		NormalSpeed	NormalStop	SlowSpeed	LaneChange
Actual Class	NormalSpeed	92.5%	0	7.5%	0
	NormalStop	0	97.8%	2.2%	0
	SlowSpeed	6.8%	0%	93.2%	0
	LaneChange	0	0	0	100%

B. Integration with the Robotics Domain

The framework was integrated with a PR2 mobile robot and the overview of the created system is shown on Fig. 6. The robot behavior is generated reactively based on the perception data obtained from the robot camera.

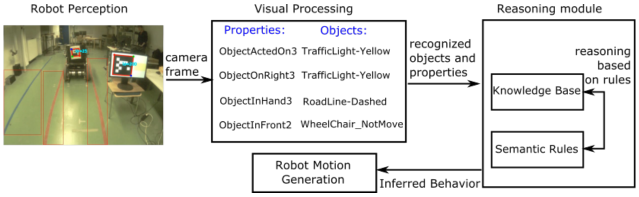

Fig. 6: Overview of the system for automatic recognition of driving scenarios implemented on the PR2.

First, each image frame obtained from the robot right camera is processed to detect existing objects and their properties. The realistic detection of objects and pedestrians is out of the scope of this paper, that is why the visual processing module was implemented using the OpenCV and the aruco library [22], which allows to detect AR markers and obtain their 3D position and orientation by using only one camera. Additionally, the color based detection was used to recognize lines of the driving tube in the testing area. For evaluation only positions in 2D space were used, because the robot and most objects it encounters cannot move in vertical direction. When objects and properties are obtained, the system uses the Reasoning module described in Section III to infer the correct behavior in a current situation (Fig. 6). To prove that the robot can reuse the semantic rules obtained from the decision tree trained on traffic situations the testing scenario shown on Fig. 7 was created. The robot can move inside the lane formed by the red lines, or by the middle red and blue lines. The middle red line represents a dashed road line, and the blue one represents a continuous road line. On his path PR2 encounters the wheelchair which is not moving and has the property *ObjectInHand2*. Moreover, the robot always perceives the dashed road line which can be crossed for overtaking and has property the *ObjectInHand3*. Using the obtained data from the tree shown in Fig. 5, the following *Computable* will be called[4]:

```
if $ObjectInHand2(o_i)$ & $ObjectInFront2(o_i)$ & $(NotMove(o_i)$ & $(o_i = Vehicle)$ & $ObjectInHand3(o_j)$ & $(o_j = RoadLine\_Dashed)$
then
    $LaneChange$
end if
```

where o_i and o_j are objects detected in traffic environment. However, the object o_i with the *ObjectInHand3* property is not the *Vehicle* but the *WheelChair* which means that the direct execution of the *Computable* would fail and additional information from the ontology will be requested. The ontology will infer the class hierarchy of the object o_i and detect that the class *WheelChair* is a subclass of *Vehicle* :

$$o_i \sqsubseteq WheelChair \sqsubset LightVehicle \sqsubset Vehicle$$

Consequently, the computable defined above will be executed and the *LaneChange* behavior will be generated.

The performance of the system is shown in Table IV which clearly indicates that the average reaction time for each perceived situation is 0.177 seconds[5]. This means that

[4] Please note that this is a simplified example of the Prolog implementation.

[5] Our system was implemented on a computer with the Intel(R) Core(TM) i5 CPU 750 2.67 GHz and 4GB memory.

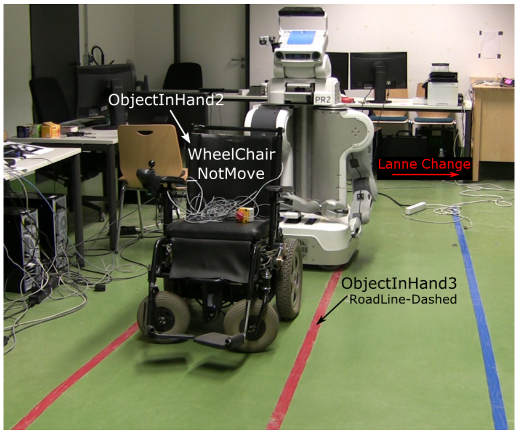

Fig. 7: Overview of the experimental setup.

the mobile robot, i.e. PR2, takes around 0.177 seconds to make a decision. It is possible that the inference time will improve when a faster perception system is used.

TABLE IV: Performance for each input frame obtained from the PR2 camera.

Type of the operation	Average execution time in sec
Image processing and object detection	0.0685529
Properties detection	0.0166563
Behavior inference	0.0917922
Total	**0.1770014**

V. CONCLUSIONS

In this paper a framework for recognition and extraction of driving situations using an artificial intelligence method was presented. This framework improves perception and situation awareness of autonomous vehicles in dynamic environments and creates general compact models of different traffic scenarios which are used to reason on road contexts. First, we trained our system in the traffic domain, and with the obtained model we tested new unknown scenarios obtaining a classification rate of 90%. Next, to demonstrate the robustness of our method we tested it on the new environment with the PR2 mobile robot. The robot reused the framework obtained from the first experiment to recognise new and previously untrained situations by taking advantage of the proposed ontology and inference methods. This framework allowed the robot to navigate successfully in an indoor environment with an average reaction time of 0.177 seconds. The results show that knowledge and semantic reasoning allows to apply our framework to different mobile domains by using the same semantic rules to enable mobile agents to correctly infer driving behaviors without additional training.

The methodology was implemented using a simple perception system, and though it used only a frontal camera, the system can be easily extended for full range detection using different types of sensors or data from dynamic maps, which will require only one additional property for objects at the back of the vehicle.

REFERENCES

[1] Beiker, S. A., ”Legal aspects of autonomous driving.” Santa Clara L. Rev., vol.52, no.4, pp.1145-1156, 2012.

[2] Aeberhard, M.; Rauch, S.; Bahram, M.; Tanzmeister, G.; Thomas, J.; Pilat, Y.; Homm, F.; Huber, W.; Kaempchen, N., ”Experience, Results and Lessons Learned from Automated Driving on Germany’s Highways.” Intelligent Transportation Systems Magazine, IEEE , vol.7, no.1, pp.42-57, Spring 2015.

[3] Armand, A.; Filliat, D.; Ibanez-Guzman, J., ”Detection of unusual behaviours for estimation of context awareness at road intersections”. In in Proceedings of the 5th Planning, Perception and Navigation for Intelligent Vehicles Workshop (IROS 2013), 2013.

[4] Armand, A.; Filliat, D.; Ibanez-Guzman, J., ”Ontology-based context awareness for driving assistance systems.” Intelligent Vehicles Symposium Proceedings, 2014 IEEE , vol.227, no.233, pp.8-11, June 2014.

[5] Florez, S. A. R.; Frmont, V.; Bonnifait, P.; Cherfaoui, V., ”Multi-modal object detection and localization for high integrity driving assistance.” Machine Vision and Applications, vol.25, no.3, pp.583-598, 2014.

[6] Gindele, T.; Brechtel, S.; Dillmann, R., ”A probabilistic model for estimating driver behaviors and vehicle trajectories in traffic environments.” Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on , vol.1625, no.1631, pp.19-22, Sept. 2010.

[7] Agamennoni, G.; Nieto, J.I.; Nebot, E.M., ”Estimation of Multivehicle Dynamics by Considering Contextual Information.” Robotics, IEEE Transactions on , vol.28, no.4, pp.855,870, Aug. 2012.

[8] Kurdej, M.; Moras, J.; Bonnifait, Ph.; Cherfaoui, V., ”Map-Aided Evidential Grids for Driving Scene Understanding.” IEEE Intelligent Transportation Systems Magazine, vol.7, no.1, pp.30-41 Jan. 2015.

[9] Bonnin, S.; Kummert, F.; Schmudderich, J., ”A Generic Concept of a System for Predicting Driving Behaviors.” Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on , vol.1803, no.1808, pp.16-19, Sept. 2012.

[10] Gindele, T.; Brechtel, S.; Dillmann, R., ”Learning Driver Behavior Models from Traffic Observations for Decision Making and Planning.” Intelligent Transportation Systems Magazine, IEEE , vol.7, no.1, pp.69,79, Spring 2015.

[11] Hummel, B.; Thiemann, W.; Lulcheva, I., ”Scene understanding of urban road intersections with description logic.” Logic and Probability for Scene Interpretation, 8091, 2008.

[12] Regele, R., ”Using Ontology-Based Traffic Models for More Efficient Decision Making of Autonomous Vehicles.” Autonomic and Autonomous Systems, 2008. ICAS 2008. Fourth International Conference on , vol.94, no.99, pp.16-21, March 2008.

[13] Hulsen, M.; Zollner, J.M.; Weiss, C., ”Traffic intersection situation description ontology for advanced driver assistance.” Intelligent Vehicles Symposium (IV), 2011 IEEE , vol.993, no.999, pp.5-9, June 2011.

[14] R. Quinlan, C4.5: Programs for Machine Learning. San Mateo, CA:Morgan Kaufmann Publishers, 1993.

[15] Gruber, T. R., ”A translation approach to portable ontology specifications.” Knowledge acquisition, vol.5, no.2, pp.199-220, 1993.

[16] Tenorth, M.; Beetz, M., ”KnowRob: A knowledge processing infrastructure for cognition-enabled robots.” The International Journal of Robotics Research, vol.32, no.5, pp.566-590, 2013.

[17] Dean, M.; Schreiber, G.; Bechhofer, S.; van Harmelen, F.; Hendler, J.; Horrocks, I.; Stein, L. A.,” OWL web ontology language reference.” W3C Recommendation February, 10, 2004.

[18] De Giacomo, G.; Lenzerini, M., ”TBox and ABox reasoning in expressive description logics.” KR, vol.96, pp.316-327, 1996.

[19] Ramirez-Amaro, K.; Beetz, M.; Cheng, G., ”Understanding the Intention of Human Activities through Semantic Perception: Observation, Understanding and Execution on a Humanoid Robot.” Advanced Robotics Journal, vol. 29, issue 5, 2015.

[20] Ramirez-Amaro, K.; Beetz, M.; Cheng, G., ”Automatic segmentation and recognition of human activities from observation based on semantic reasoning.” IROS 2014, 2014 IEEE/RSJ International Conference on , vol.5043, no.5048, pp.14-18, Sept. 2014.

[21] Hall, M.; Frank, E.; Holmes, G.; Pfahringer, B.; Reutemann, P.; Witten, I. H., ”The WEKA data mining software: an update.” ACM SIGKDD explorations newsletter, vol.11, no.1, pp.10-18, 2009.

[22] Garrido-Jurado, S.; Muoz-Salinas, R.; Madrid-Cuevas, F. J.; Marn-Jimnez, M. J., ”Automatic generation and detection of highly reliable fiducial markers under occlusion.” Pattern Recognition, vol.47, no.6, pp.2280-2292, 2014.