Combining Onthologies and Behavior-based Control for Aware Navigation in Challenging Off-road Environments

Patrick Wolf, Thorsten Ropertz, Philipp Feldmann and Karsten Berns
Robotics Research Lab, Dep. of Computer Science, TU Kaiserslautern, Kaiserslautern, Germany

Keywords: Off-road Robotics, Ontology, Behavior-based Control, Navigation, Simulation.

Abstract: Autonomous navigation in off-road environments is a challenging task for mobile robots. Recent success in artificial intelligence research demonstrates the suitability and relevance of neural networks and learning approaches for image classification and off-road robotics. Nonetheless, meaningful decision making processes require semantic knowledge to enable complex scene understanding on a higher abstraction level than pure image data. A promising approach to incooperate semantic knowledge are ontologies. Especially in the off-road domain, scene object correlations heavily influence the navigation outcome and misinterpretations may lead to the loss of the robot, environmental, or even personal damage. In the past, behavior-based control systems have proven to robustly handle such uncertain environments. This paper combines both approaches to achieve a situation-aware navigation in off-road environments. Hereby, the robot's navigation is improved using high-level off-road background knowledge in form of ontologies along with a reactive, and modular behavior network. The feasibility of the approach is demonstrated within different simulation scenarios.

1 INTRODUCTION

What was long assumed to be futuristic will soon characterize our everyday life: The vision of self-driving cars—especially in the on-road domain—has taken manifest form in recent years (Thrun et al., 2006), (Ziegler et al., 2014). The development of autonomous road vehicles profits from the structuredness of the environment and the availability of certain regulations and standards. This greatly simplifies the situation assessment for autonomous vehicles. In contrast, the off-road sector with frequently changing environmental conditions and as well as highly unstructured, rough, and dangerous surroundings still remains an unsolved area of research. Autonomous vehicles operating off-road are constantly exposed to unpredictable situations as for instance poor visibility caused by rain, dust, or mud. Additionally, properties of scene objects, as rocks, tree trunks, or versatile surface conditions, as well as the corresponding object correlations have a huge impact on the traversability estimation and navigation. Behavior-based systems (BBS) have shown to be suited for handling such difficult environments by relying on a modular design with sophisticated arbitration mechanisms (Berns et al., 2011). The research area of artificial neural networks offers promising results in image recognition (Valada et al., 2017) which is of indispensable importance in the field of mobile robotics. Unfortunately, pure reactive sensor data-based processing limits the set of possible actions since not all navigation relevant factors are perceivable. In contrast, a human operator relies heavily on world knowledge, which is used to achieve highly advanced navigation maneuvers and to handle uncertain scenarios with incomplete information.

Through applying semantic knowledge models, autonomous robots can exploit human experience. This offers an enormous advantages for task planning and navigation since strategies can be independently selected on a corresponding scenario. Furthermore, sensor data analysis may utilize such experience to identify faulty signals. An appropriate technology incorporate the background knowledge is the Semantic Web Technology (SWT). It structures information in a semantic model and was originally developed to handle the rapidly growing amount of data on the world wide web to make it accessible to search engines and likewise (Hitzler et al., 2009). The SWT is based on so-called ontologies. Ontologies are knowledge models used to link information with semantic relationships. They are suited for capturing, exchanging, and deriving information in a machine-processable as well as human-understandable form.

Wolf, P., Ropertz, T., Feldmann, P. and Berns, K.
Combining Onthologies and Behavior-based Control for Aware Navigation in Challenging Off-road Environments.
DOI: 10.5220/0007934301350146
In *Proceedings of the 16th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2019)*, pages 135-146
ISBN: 978-989-758-380-3

Therefore, ontologies are predestined be used as a semantic knowledge base to acquire information for situation-aware navigation in off-road environments.

This paper provides a novel methodology for off-road navigation by combining behavior-based control and ontologies. It is structured as follows: Section 2 presents an overview of state of the art applications of ontologies in robotics. Next, an overview to the integrated Behavior-Based Control architecture iB2C (Section 3) is given, which was used for robot control and ontology integration. An off-road ontology, the Scene INterpretation Engine (SINE), is suggested in Section 4. Section 5 demonstrates the integration and interaction of a behavior-based robot controller and SINE. Experimental results are given in section 6, where the approach was tested in simulation. Section 7 summarizes the presented approach and concludes.

2 RELATED WORK

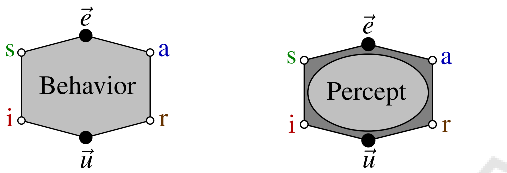
The deliberation of autonomous systems is a highly active research area of robotics. Hereby, deliberation aims at enabling a robot to fulfill its task in a variety of environments. It has an impact on acting, learning, reasoning, planning, observing, as well as the monitoring of the surroundings (Ingrand and Ghallab, 2017). Hereby, knowledge modeling with ontologies is a well known technique. The Open Mind Common Sense Project (OMICS) (Gupta et al., 2004) focuses the area of indoor robotics and provides a knowledge base which was created by more than 3000 volunteers and includes more than 1.1 million statements. Similarly, the KnowRob-Map (Tenorth et al., 2010) enables autonomous household robots to perform complex tasks in indoor environments. Spatial and encyclopedic information about objects and their environment enables a robot to determine the type and function of a detected object. Another mapping approach, the multiversal semantic map (MvSmap) extends metric-topological maps by semantic knowledge (Ruiz-Sarmiento et al., 2017). Through the identification of object and room types mobile robots can distinguish working environments as kitchen, living rooms, and bedrooms based on the detected objects located in the room.

There exist also various approaches using ontologies for outdoor scene descriptions. Recordings of outdoor scenarios can be described with the help of an ontology in sentences of natural language. Here, primitive units are extracted from an image by stochastic processes. Attributes as well as relationships of the units are evaluated by an ontology based on predefined proposition. A pool of sentence templates can be selected and enhanced according to the theme of the visual content (Nwogu et al., 2011). Such approaches have shown to provide promising results and have been further investigated in several works (Farhadi et al., 2009), (Yao et al., 2010), (Kulkarni et al., 2011). It is especially important that a complete scenario catalogue exists for every task a vehicle should fulfill to achieve a formal approval of a vehicle. Nonetheless, the number of critical scenarios is hardly manageable for a vehicle with a high degree of automation. Therefore, an ontology as knowledge-based system for the generation of traffic scenarios and testing of automated vehicles in road traffic is suggested by the authors of (Bagschik et al., 2017). It supports the identification of possible scenario permutations in road traffic scenarios and is intended to automate the creation and testing of road traffic scenarios. The method uses an ontology along rules of the Semantic Web Rule Language (SWRL) (Horrocks et al., 2004) to determine motion maneuvers. Thus, a Semantic Web Rule (SWR) was implemented in the ontology for each maneuver, while entities and rules were generated by permutation and logical thinking.

The assessment of the current risk-level is an important development aspect for mobile systems. Unfortunately, pure object recognition does not provide sufficient information to safely operate a robot since type and behavior of objects are of great importance for the assessment of the degree of hazard. In the past, ontologies have been also used to target the risk assessment of road scenarios of autonomous vehicles (Mohammad et al., 2015). The approach focuses the problem of assigning a semantic meaning to a perceived environment similar to humans who utilize their gathered experience. Therefore, inference rules of the SWRL are formulated in the ontology to assign risk factor classes to the objects. The methodology uses multiple sets of rules for the four risk classes (high risk, medium risk, low risk, and no risk) of a pedestrian crossing a road. The current risk is derived based on risk assessment knowledge of a driver and ontology information. Thereby, a hazardous scenario where a pedestrian crosses a road demonstrated the feasibility of the approach. Thus, the behavior of the pedestrian is essential for the assessment of the risk. On the one hand, the pedestrian could stay on the sidewalk and move away from the road leading to a low risk. On the other hand, the pedestrian could also move carelessly towards the road yielding a high risk. In addition, environmental factors as visibility and weather influence the risk assessment and have to be considered for the risk determination.

3 BEHAVIOR-BASED CONTROL

The integrated Behavior-Based Control (iB2C) architecture (Proetzsch, 2010), (Ropertz et al., 2017) has been developed at the Robotics Research Lab of TU Kaiserslautern. The underlying idea is that the overall system behavior emerges from the interaction of rather simple behavior components which realize only little functionality. In iB2C, there exist different basic component types for control and perception. *Behavior* modules are used for command execution, while *Percept* modules are suited for sensing and data processing by considering respective data quality information (see Fig. 1).



(a) Behavior $B = (f_a, F)$. (b) Percept $P = (f_a, f_{p_u}, F)$.

Figure 1: Basic iB2C units (Ropertz et al., 2017).

BBS are robust against environmental changes due to the partially overlapping functionality and the ability to adapt to the surroundings by using dynamic arbitration. Contradicting control and perception information is resolved through fusion modules which coordinate the interaction of network components and combine parallel data flows. All iB2C components provide a standardized common interfaces consisting of stimulation s and inhibition i, which allows to adjust the maximum relevance of a module in the current system state. The target rating output r indicates the contentment of the behavior and is defined by the activity function $f_a(\vec{e})$. The behavior's activity $a = \min(s \cdot (1 - i), r)$ reflects the actual relevance of the behavior in the current system state and is used by fusion behaviors to perform the arbitration process or to activate or inhibit other network elements. In addition, each behavior component provides an application specific interface consisting of the input vector $\vec{e}$ and output vector $\vec{u}$ containing arbitrary control and sensor data. Thereby, the output vector is defined by the transfer function $F(\vec{e})$. For coordination purposes, there are different fusion approaches predefined, namely the *Maximum Fusion* and the *Weighted Average Fusion*. The former implements a winner-takes-all methodology, where the behavior with highest activity, or respectively best data quality, gains the control. The latter admits influence with respect to the total activity ratio of every connected module.

4 OUTDOOR ONTOLOGY DESIGN

In the following, the ontology design for an outdoor scene description tailored to autonomous mobile off-road robots in rough environments is presented. Thereby, the risk assessment of the current scene is emphasized.

4.1 Entity Identification

A first step in ontology design is the identification of entities which influence the autonomous off-road navigation. This requires the modeling of various decision-relevant factors that enable risk assessment. Hereby, the entity determination needs to be complete and all relevant aspects have been considered. Unfortunately, this is hard to achieve due to the scene complexity and uncertainty in off-road environments. To illustrate those issues, a simple forest path and its specific characteristics are examined in more detail (Fig. 2). Hereby, an initial set of assumptions is collected, which is used for modeling rules of the ontology.

Figure 2: Forest path with obstacle.

The scene can be semantically segmented into two major classes: pathway and off-track. This simplistic segmentation is especially meaningful for the risk assessment of different navigation scenarios. In general, it is assumed that a path has a better traversability than driving off-road. A pathway should not contain large obstacles and its surface is often flat. Thus, a robot should navigate along a path as long as possible. Likewise, there is a higher navigation risk beside the paths due to a higher probability of large obstacles and rougher surfaces. Additionally, surface material properties are relevant for the risk assessment of a situation and have to be regarded. The robot's traction differs strongly while driving on gravel, sand, or forest road. Furthermore, the type of path geometries is relevant. Exemplary classes for the ontology are *curve*, *straight*, *uphill*, *downhill*, and *flat*. A human driver assess driving situations based on accumulated experience and learned knowledge. Therefore, occluded regions are approached at a lower velocity, as for instance an area behind a hill top. Other environmental factors as bad weather and illumination changes also trigger more cautious driving behavior. Corresponding types of precipitation that can affect the risk assessment are for instance *fog*, *snow*, and *rain*. However, since thy can influence not only visibility but also the road surface, e.g. by forming icy roads, the current *temperature* is also included in the class descriptions of the ontology. A very important and potentially most dangerous source of danger in a scene are obstacles. Therefore, different types of objects have been identified for risk assessing ontology.

The example presented in Fig. 2 shows a forest road section, where a part is blocked by a fallen tree stump. Such obstacles may present a risk depending on their size, the construction of the vehicle's chassis, and the current velocity. Identified stationary objects include *stones* and *fallen tree stumps*, *trees*, *bushes*, *steps*, and *tall grass*. The class description for tall grass was included since it often causes undesirable effects during navigation. Usually, mobile off-road robots rely on local obstacle maps for near field navigation as metric or grid maps to determine occupied areas or vehicle collisions with the underground (Wolf et al., 2018b). These obstacle maps are usually generated using geometric information provided by distance data. Therefore, obstacles are regarded as blocking without considering semantic knowledge. E.g.tall grass is recognized as a barrier, while the incorporation of semantic information would allow the robot to pass through the area. Similar cases are avoided and availability is increased without decreasing safety using modeled knowledge.

Usually, autonomous systems expose nearby humans to a very high risk. Therefore, they are separately considered by the ontology and classified as dynamic objects in order to keep the risk as low as possible. Classes corresponding to *moving objects* strongly increase the risk potential for a scene. Additionally, the motion direction of objects is explicitly modeled within the ontology as class descriptions. This enables the evaluation of the definite high risk potential of an mobile object as suggested by (Mohammad et al., 2015).

In addition, class descriptions for the identified risk and a steering direction recommendation are defined to reasonably react to potential risks in the scene. The risk is categorized into four classes, *no risk*, *low risk*, *medium risk* and *high risk*. The steering direction recommendation consists of two class descriptions: *left steering* and *right steering*.

4.2 Class Hierarchy

The entities and properties identified in Section 4.1 were divided into groups of membership and modeled within a class hierarchy (Fig. 3). Thereby, the editing tool Protégé[1] (Musen, 2015) was used to create the ontology. It provides an overview of the class hierarchy.

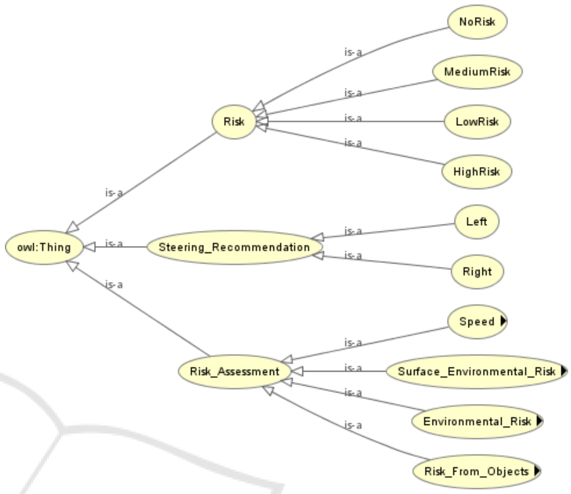


Figure 3: The ontology class hierarchy for off-road scene risk assessment and a steering direction proposal.

Risk and *Steering_Recommendation* summarize the entities provided as derived results to the robot control system. *Risk* is specified in four subclasses: *NoRisk*, *LowRisk*, *MediumRisk*, and *HighRisk*. The *Steering_Recommendation* class contains two subclasses *Left* and *Right*. *Risk_Assessment* is the main class and summarizes the other class descriptions. Subclasses of *Risk_Assessment* include the *Speed* related risk sources previously identified. Likewise, the *Surface_Environmental_Risk* class summarizes underground surfaces of the environment. Similarly, the class *Environmental_Risk* describes the risk based on environmental influences such as visibility and weather conditions. *Risk_From_Objects* describes the most important subgroup of risk factors: all types of objects and their properties.

The class *Risk_From_Objects* (Fig. 3) is further refined in Fig. 4. It describes obstacles together with their attribute class descriptions including the *Type_of_Objects*, which divides the type of objects into two categories. The first category are ground objects which are described by the *Ground* class. Its specified class descriptions are *Path* and *PathSide* classes. The individuals of the class descriptions can be populated with the attributes of *Sur-*

[1]https://protege.stanford.edu/

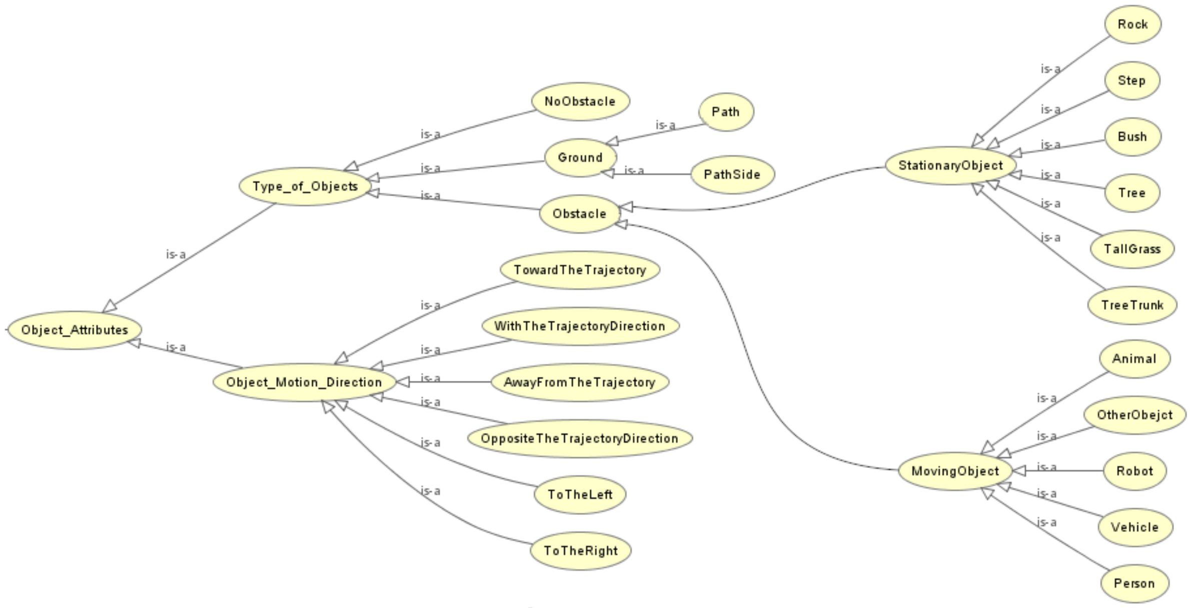


Figure 4: The ontology class hierarchy for the subclass group *Risk_From_Objects* with the subclass groups *Object_Attributes*. It contains the class *Type_of_Objects* which combines the object types for *Ground*, *Obstacles*, and *NoObstacles*. The class *Object_Motion_Direction* contains a collection of attributes for the directions of movement: *TowardTheTrajectory*, *AwayFromTheTrajectory*, *WithTheTrajectory*, *ToTheLeft*, *ToTheRight*, *WithTheTrajectoryDirection*, and *OppositeTheTrajectoryDirection*. The subclass *Ground* encapsulates the entities *Path* and *PathSide*. *Obstacle* specifies the classes *MovingObject* and *StationaryObject*. Motion objects are *Person*, *Robot*, *Animal*, *Vehicle*, and *OtherObject*. Stationary objects are *TreeTrunk*, *Tree*, *Bush*, *Step*, *TallGrass*, and *Rock*.

face_Environmental_Risk using object property relations. The second category are *Obstacle* classes. Here, *MovingObjects* and *StationaryObjects* are distinguished. *StationeryObjects* includes the class descriptions *TreeTrunk*, *Tree*, *Bush*, *Step*, *TallGrass*, and *Rock*. The entities *Person*, *Robot*, *Animal*, *OtherObject*, and *Vehicle* have been assigned to the class *MovingObjects*. *Robot* is listed separately because it has all properties of the class *MovingObjects* without being an obstacle. Objects which have been not classified through an object recognition system are assigned to the *OtherObject* class in this knowledge model. This class is interpreted as dangerous with respect to risk assessment. Thus, a worst case assumption for unknown objects is applied. The class *NoObstacle*, which is on an equal hierarchical level of as *Ground* and *Obstacle* serves to describe a scene were no obstacle exists. This is required due to the open world assumption of the ontology. Thus, a class as *NoObstacle* which is disjunctive to the *Obstacle* class can be used to paraphrase the absence of a statement. In contrast, a closed world assumption can simply be implicitly inferred from the absence of a statement.

4.3 Object and Data Properties

Additional predicates are required to describe the relationships between the recognized individuals and their specific characteristics in order to implement a situation-conscious scene description called assertional knowledge. Object properties define entity-data correlations and provide attributes of individuals. In the suggested approach, object and data properties are separated into functional and non-functional characteristics. In the first case, properties of the assertional box can be derived from existing knowledge about the classes and their detected attributes with the help of rules defined in ontology. In the second case, properties of non-functional characteristics serve to note perceived attributes of the scene. These functional relationships build on each other to derive query-able results for the risk assessment and a steering direction suggestion to avoid risks.

4.4 Rules of the Ontology

SWRL is especially suitable for expressing complex relationships in ontologies by recursively connecting the described rules. Due to the importance of this property, SWRL was selected as the expressive rule language. Thereby, recognized factors are semantically linked in order to make a meaningful statement. Ontology rules are defined to derive statements which in turn are needed for the evaluation of rule definitions on a higher level.

An example of the functional object property *hasHighRisk* is provided. It assigns an instance of the class *HighRisk* to the robot individual if the following rule applies:

$$\begin{aligned} &Robot(?r) \wedge Obstacle(?o) \wedge HighRisk(?hr) \\ &\wedge hasSafetyDistance(?o, false) \\ &\wedge isBypassableOnRoad(?o, false) \\ &\wedge isBypassableOffRoad(?o, false) \\ &\wedge isOverdriveable(?o, false) \\ &\rightarrow hasHighRisk(?r, ?hr) \end{aligned} \tag{1}$$

Rule (1) is fulfilled if all atoms in the rule body are fulfilled. In this case, all data properties required to fulfill the rule body are of functional character. Next, the rule is used to propagate a possible trace until the lowest level is reached. Therefore, a rule that derives the data property $hasSafetyDistance = false$ of an obstacle is defined as

$$\begin{aligned} &Robot(?r) \wedge Ground(?g) \wedge Obstacle(?o) \\ &\wedge hasGoodVisibility(?r, true) \\ &\wedge isSlippery(?g, false) \\ &\wedge HighSpeed(?v) \\ &\wedge hasSpeedClassification(?r, ?v) \\ &\wedge hasDistance(?o, ?d) \\ &\wedge hasGrownDepth(?o, ?gd) \\ &\wedge swrlb : divide(?halfgd, ?gd, 2) \\ &\wedge swrlb : subtract(?sd, ?d, ?halfgd) \\ &\wedge swrlb : lessThan(?sd, 25) \\ &\rightarrow hasSafetyDistance(?o, false) \end{aligned} \tag{2}$$

The body of rule (2) requires that there is an instance of *Robot*, *Ground*, and *Obstacle* present. In addition, the functional data property *hasGoodVisibility* has to be true for the robot. The functional data property *isSlippery* is expected to be false for the ground instance. The safety distance to be maintained is doubled in the case that one of both data properties fulfills the complementary statement. Additionally, speed classification is required to determine the required safety distances. Therefore, the rule body asks for an instance of *HighSpeed* with the object property *hasSpeedClassification*. This shows that there exist several rules for *hasSafetyDistance* in order to consider all possible cases. Furthermore, *hasGrownDepth* queries the size of the obstacle in the direction of travel and subtracts the half value from the query distance of *hasDistance* between the obstacle and the robot. It further checks whether the result is smaller than the required distance of 25 m to maintain the safety distance.

The above rule requires functional relationships of other rule-based object and data properties for evaluation. In this case, the rules (3) – (5) apply.

$$\begin{aligned} &Robot(?r) \wedge OptimalWeather(?wc) \\ &\wedge hasWeatherCondition(?r, ?wc) \\ &\rightarrow hasGoodVisibility(?r, true) \end{aligned} \tag{3}$$

The data property *hasGoodVisibility* requires further rule definitions for the classes *Weather_Condition* and *Visibility_Condition*.

$$\begin{aligned} &Robot(?r) \wedge OptimalWeather(?wc) \\ &\wedge Ground(?g) \wedge Asphalt(?m) \\ &\wedge hasWeatherCondition(?r, ?wc) \\ &\wedge hasGroundMaterial(?g, ?m) \\ &\rightarrow isSlippery(?g, false) \end{aligned} \tag{4}$$

The same applies to the *isSlippery* data property. Here, rules for *Weather_Condition*, *Ground_Material*, and *Ground_Quality* as well as their different combinations have to be defined. Additionally, the rule temperature further increases the overall rule count.

$$\begin{aligned} &MovingObject(?m) \wedge HighSpeed(?hv) \\ &\wedge hasSpeed(?m, ?v) \\ &\wedge swrlb : greaterThan(?v, 8.33333) \\ &\rightarrow hasSpeedClassification(?m, ?hv) \end{aligned} \tag{5}$$

Last but not least, the property *hasSpeedClassification* requires the assignment of a speed class to a *MovingObject*. Here, the example for high speed is depicted.

The given example clearly shows the complexity of relationships and the large number rules which is required for scene analysis. SWRL rules require to be represent as conjunctions of atoms, which forces the formulation of another set of rules for every disjunction of a logical statement. The open world assumption implies that a rule has to be modeled for fulfillment and in addition explicitly for the non-fulfillment. The latter cannot be implicitly closed from the inverse fulfillment of a rule, which leads to a huge amount of rules. This simple scenario already contains more than 200 SWRL rules.

4.5 Scene Interpretation Engine

The presented off-road ontology is part of the Scene Interpretation Engine (SINE). SINE serves as knowledge database along with other world knowledge as for instance OpenStreetMap data (Fleischmann et al., 2016). It enables an independent perception and control design since the ontology retrieves sensor data through generic and standardized interfaces (Schäfer et al., 2008). Therefore, arbitrary sensor data are transformed into a common representation for data exchange. This enables the consideration of various information sources as sensor-based object segmentation, or simulation data. Furthermore, this fosters extensibility as there are further sophisticated learning approaches expected to be available in near future.

5 BEHAVIOR-BASED ONTOLOGY ASSESSMENT

This section discusses the incorporation of the presented off-road ontology SINE into the behavior-based robot control architecture REACTiON for robust off-road navigation (Wolf et al., 2018a). It is implemented using iB2C and features data quality driven perception processes. Data is evaluated through a single- and multi-sensor quality assessment (Ropertz et al., 2017), which adapts dynamically to disturbances. Furthermore, reactive low-level (Ropertz et al., 2018a) and fail-safe systems (Wolf et al., 2018a) ensure a safe and robust navigation in cluttered environments.

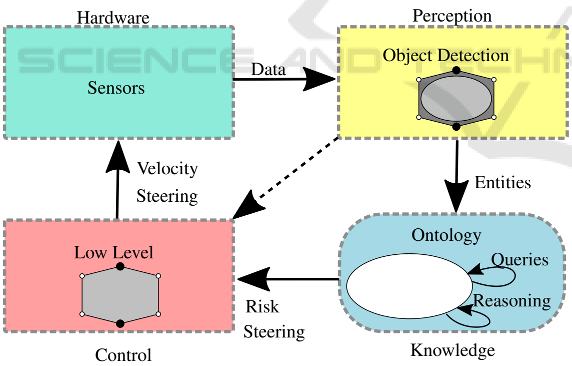


Figure 5: Conceptual representation for ontology-control interaction.

A simplified scheme for ontology-control interaction is depicted in Fig. 5. The hardware abstraction provides sensor data processed by the robot's perception. Object detection systems like a deep learning algorithm share information through standardized and common scene interfaces. It transmits detected entities including their positions, dimensions, velocities and other non-functional object and data properties to the knowledge database. The ontology processes scene data as classified entities, obstacles, path sections, and individuals of the corresponding class description. Further data are triple axioms in the assertional box of the ontology as well as object and data properties which are read directly from the scene. Newly created axioms of ontology are processed including their terminological knowledge and the defined rules for the properties with functional character. Finally, conclusions are derived with the help of a reasoner. The ontology uses queries for risk level determination and steering recommendation requests. The results are transmitted to the control system of the robot which performs different safety checks as collision prediction, roll-over avoidance, and centrifugal acceleration limitation. Here, the ontology information about the safety state of the system is transformed into a behavior signal and is considered for behavior network arbitration. In contrast to the given representation, a standard robot control system would directly connect perception and control for motion generation. An example for ontology interaction with the robot's low-level control depicted in Fig. 6.

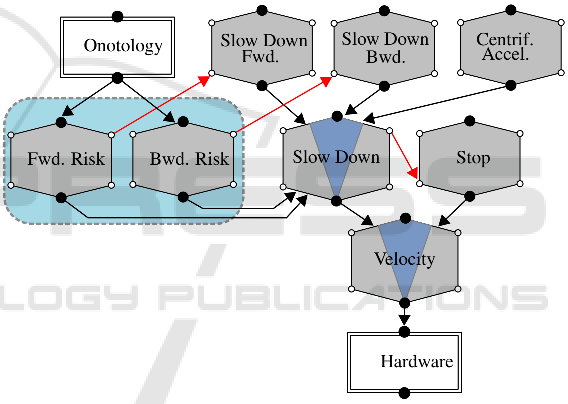


Figure 6: Low level collision avoidance incorporates ontology risk knowledge.

The low-level safety system (Ropertz et al., 2018b) uses different behaviors for safety state evaluation. A velocity provided by a higher level controller is modified through the *Slow Down Forward*, *Slow Down Backward*, and *Centrifugal Acceleration* units. The robot's resulting *Velocity* is determined by the *Slow Down* fusion, while its default behavior is *Stop*. Finally, the speed information is processed by the *Hardware* interface.

In the extended version, the *Ontology* assesses the control system through additional behaviors. The *Forward Risk* and *Backward Risk* behaviors may adjust the velocity based on the determined risk level to minimize a potential risk caused by the environment. Additionally, they are capable of disabling the respective slow down behavior if there is no risk present. This is meaningful in the case of a spurious obstacle detection, which may occur in the presence of tall grass, light vegetation, or likewise.

Similarly, SINE may adapt the robot's evasion behaviors according to the risk factor and steering recommendation. Therefore, potential risks can be avoided at an early stage of navigation. Thus, the ontology may control the robot completely or can partially suggest control values to the low level controller.

6 EXPERIMENTS

The presented ontology was tested in various simulation scenarios. Hereby, the robot's driving behavior using a behavior controller, the pure ontology, as well as a hybrid control approach for navigation were compared. Each, the navigation test was repeated and the results compared to each other.

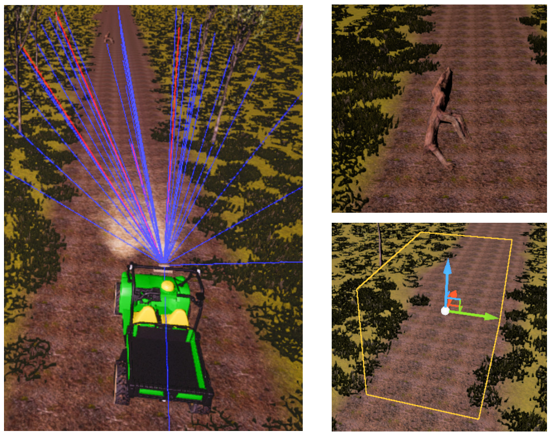
The control software and the onotology were implemented using Finroc, a C++/Java-based robot control framework with real-time capabilities, zero-copy implementations, and lock free data exchange (Reichardt et al., 2012).

The scenarios were tested in the Unreal Engine (UE 4) which was used for simulation. Finroc and the UE 4 share data via an engine plugin. Different tests were done using the simulated robot GatorX855D of the RRLab, TU Kaiserslautern. In addition to the sensor setup as described by (Ropertz et al., 2017), a ray-trace actor of the Unreal Engine was used as object detector. Therefore, standard sensor data for robot control were available. Additionally, SINE was able to operate with perfect classification data. This enabled the comparison of the robot's standard navigation behavior, ontology-based navigation results, and the combination of both approaches without dependencies on the actual classifier.

6.1 Simulated Object Detector

The simulated object detector is a perfect classificator. On the one hand side, it is used to test the onotology with undisturbed sensor data which prevents undesired effects on the final control result. On the other hand side, it enables the testing with arbitrary properties of detected objects. Even if current deep learning approaches are very promising, they are not yet capable of detecting every individual aspect of a scene. Nonetheless, it can be assumed that future learning approaches will be powerful enough to test the presented onotology framework with real sensor data.

The object detector is implemented as a specification of the UE 4 actor class. It is placed on the robot next to the top stereo camera as shown in Fig. 7a. It



(a) Line trace object detector. The blue lines indicate a visible, red lines a hidden object.

(b) Stationary obstacle (top). Volume-based entity (bottom).

Figure 7: Entities in the simulation environment.

has the same field of view as the corresponding stereo camera system. Therefore, only objects in the visible range of the robot are regarded by the ontology. The ray-trace actor provides detailed information about visible objects which can be arbitrary tagged. For this purpose an invisible volume body in the form of a frustum is used, which covers the camera's field of vision. The frustum is measured by the render depth and the aperture angle of the viewing cone of a camera actor. During the simulation only the objects that overlap with the frustum have to be found out in order to read out the visible scene.

All objects to be detected are actor classes themselves like the obstacles contained in the scene (Fig. 7b, top). A line trace is used to check whether the imaginary line between the object detector and the object is free of collisions to check the visibility. Additional rectangular volume bodies were placed on the respective areas in order to recognize path and path side sections in the simulation (Fig. 7b, bottom). Every type of object was additionally labeled with properties to be recognized and processed by the object detector. Further information as height, width, depth, location, and speed of the detected objects can be read directly from their attributes. Specialized functions provide more data as distance of objects to the robot, their motion direction, location of object edges, distances to their lateral obstacles, and their location with respect to the path or path sides.

6.2 Environment and Scenarios

Multiple scenarios were tested in simulation to demonstrate the different aspects of the onotology and its impact on the robot control. Therefore, the robot navigated in a forest environment and completed six different scenarios. Each scenario was repeated three times with a different control setup (behavior-based, ontology, combined). An overview to the experimental results is available in Fig. 8. Additionally, recorded trajectories, velocities, risk level, and steering recommendations are presented.

Scenario 1: Passing an Obstacle. The first scenario was passing a stationary obstacle on a straight pathway (Fig. 8a). Here, the robot's trajectory is nearly similar for each test run. The obstacle is always passed on the right hand side. A minor difference is that the onotology-based approach starts to evade earlier than the combined and behavior-based approaches. Nonetheless, a clear difference concerning the vehicle's velocities can be observed. The REACTiON controller advances the obstacle with a high velocity of about $2.5\,\mathrm{m\,s^{-1}}$. It starts to decelerate close to the obstacle (2.5 m). In contrast, the other controllers carefully approach the obstacle due the corresponding risk level of a medium risk. The robot's steering is re-adapted after the obstacle was passed. Then, the risk level decreases to no risk and the robot is allowed to accelerate again. Thereby, all approaches share a rather slow velocity ($1\,\mathrm{m\,s^{-1}}$) during passing the obstacle location.

Scenario 2: Passing an Obstacle During Heavy Rain. The next scenario is the repetition of the first one with the vision occluded by heavy rain (Fig. 8b). Interestingly, the standard behavior controller selects a different route to evade the obstacle and therefore navigates off-road. This is caused by camera disturbances causes due to the weather. The ontology and combined approach follow the trajectory from the previous trial. Nonetheless, the overall detected risk is in general higher (medium to high) and the ontology controller navigates very cautiously ($< 1\,\mathrm{m\,s^{-1}}$). In contrast, the raw behavior controller navigates with a constant high velocity of $2.5\,\mathrm{m\,s^{-1}}$ and only slows down during reentering the track. In general, the combined approach navigates faster than the ontology controller but slows down more strongly in front the obstacle. The steering recommendation is similar to the first scenario.

Scenario 3: Navigating over a Hilltop. The third scenario is the navigation over a hilltop (Fig. 8c). Here, all approaches share a similar trajectory and velocity. The velocities decrease in the slope uphill and slightly increase downhill. The ontology detects a low risk in front of the hill and no risk after the hilltop.

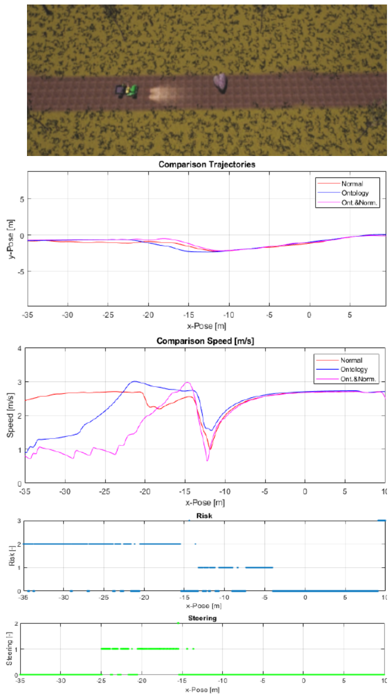
Scenario 4: Pedestrian Crossing. The fourth scenario involves two pedestrians walking close to the robot (Fig. 8d). Again the trajectories are similar, while the velocities differ strongly. The behavior controller slows down ($2\,\mathrm{m\,s^{-1}}$) when the pedestrian is close (3 m). It accelerates ($2.5\,\mathrm{m\,s^{-1}}$) after the person passed the robot and slows down again in front of the second person. Hereby, the robot finishes decelerating when the pedestrian has already passed since the motion vector of the person towards the robot is not considered by the behavior controller. It can be assumed that such a behavior would endanger a real person. In contrast, the onotology detects a medium risk when the person is detected and maximum risk when it is nearby. During the maximum risk phase, the robot is forced to slow down to $< 0.5\,\mathrm{m\,s^{-1}}$. The robot is allowed to accelerate again after the critical situation is over.

Scenario 5: Sharp Curve Navigation. In the fifth scenario, the robot has to follow a sharp curve (Fig. 8e). The onotology controller stays exactly on the path, while the combined and behavioral approaches overshoot the curve. This results from the stereo camera's opening angle in combination with the robot's occupancy map. The robot is not able to access uncertain regions of the map which have not been analyzed before. The onotology overcomes this problem through a tailored rule. The vehicle's velocities are rather similar and a nearly constant risk level of low risk is present.

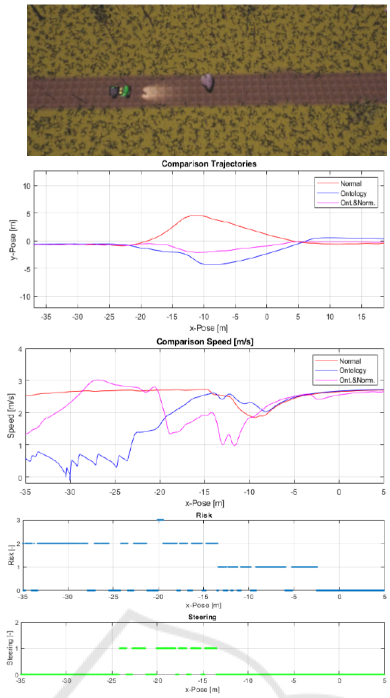
Scenario 6: Driving through Tall Grass. The final scenario was driving through tall grass (Fig. 8f). Hereby, the onotology controller was able to carefully navigate through the grass ($< 2\,\mathrm{m\,s^{-1}}$), while the other approaches stopped in front of the spurious obstacle. The onotology detected a medium risk in front of the grass and no or respectively a low risk behind. The iB2C controller approached the grass rapidly and then stopped completely. The combined controller slowly approaches the grass but was overridden from the low level safety. This was possible since a risk level was present and the ontology was not able to completely disable the low level safety features.

7 CONCLUSIONS AND FUTURE WORK

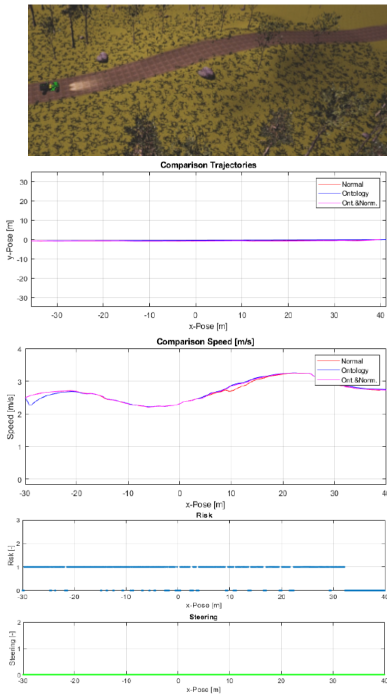
The paper presented a novel approach for situation-aware scene assessment by combining a behavior-based control approach with an off-road ontology.



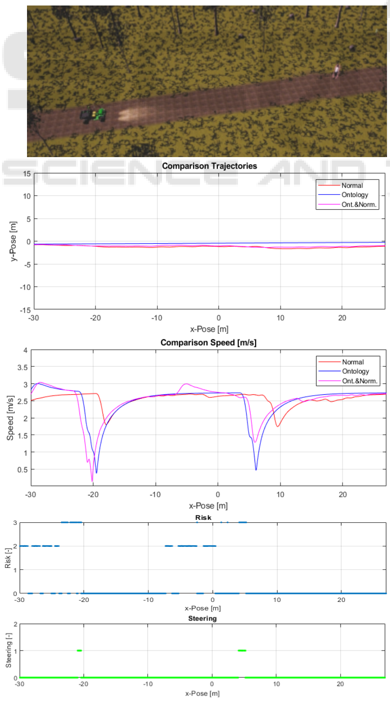
(a) Obstacle



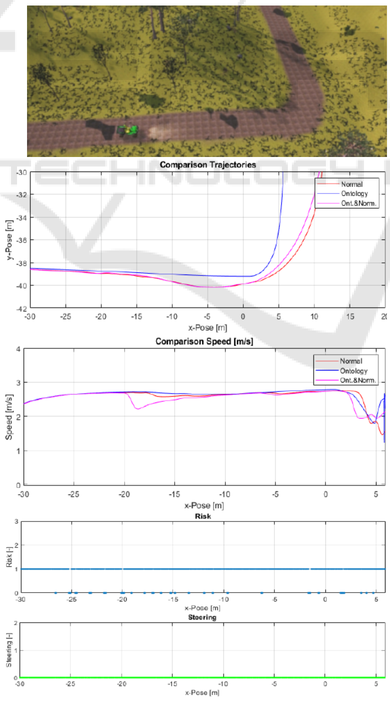
(b) Rain and Obstacle



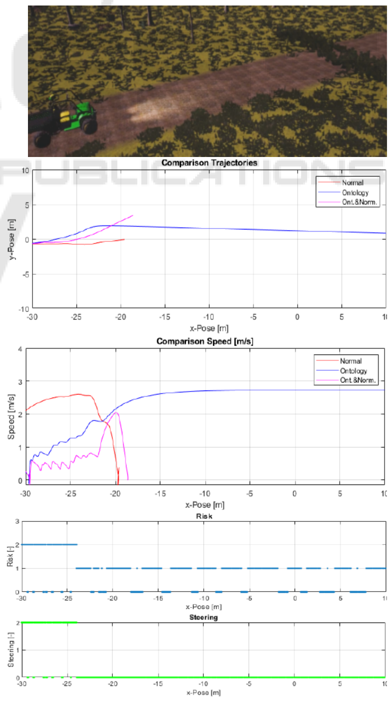
(c) Hilltop



(d) Pedestrian



(e) Curve



(f) Tall grass

Figure 8: Results of different simulation trials. Each test run was executed three times using the standard behavior controller (red), the pure ontology based controller (blue), and the combined approach (pink). The subfigures depict (from the top): scenario, trajectory, vehicle speed, ontology risk level (from no risk = 0 to high risk = 3), and the steering recommendation (none = 0, right = 1, left = 2).

Initially, state of the art onotologies have been summarized and evaluated with respect to the given scenario. Next, the integrated behavior-based control architecture iB2C was outlined, which acts as basis for the behavioral controller and interaction concepts. Further, an off-road onotology design has been presented. Hereby, the entity identification was described in detail followed by an overview over the onotology class hierarchy. After the summary of object and data properties, the rules of onotology were derived and explained along with an example for high risk determination. The onotology is part of the scene interpretation engine SINE, which embeds it into a knowledge framework featuring standardized and common interfaces for arbitrary data exchange and independent control and perception design. In a following step, the assessment of onotology knowledge by a behavior-based control approach was given. Hereby, the transformation of risk levels into iB2C meta data and the impact on network arbitration was explained using an exemplary low level safety system. Finally, the approach was tested in six different simulation scenarios, where the pure ontology-based controller, behavior-based controller, and the combined approach had to fulfill similar tasks. The corresponding results were compared to each other and discussed in detail.

Future work should target the extension of the knowledge base by more entities and respective rules so that a wider variety of scenarios can be handled. Furthermore, the onotology should be used to detect erroneous sensor readings and determine the corresponding data quality. An additional extension could subject the storage of a scene. This would enable the analysis of a scene history and allow an improved planning process as well as a better understanding of an environment. Hereby, knowledge transfer to other robots could be possible. This would result in a kind of driving school for mobile robots. In the intermediate future, the approach can be tested with powerful learning approaches in real world scenarios. Nonetheless, they should sense sufficient environmental information as for instance surface conditions, weather, and object classes to satisfy the properties which are required for the rule analysis of this onotology.

REFERENCES

Bagschik, G., Menzel, T., and Maurer, M. (2017). Ontology based scene creation for the development of automated vehicles. *arXiv preprint arXiv:1704.01006*.

Berns, K., Kuhnert, K.-D., and Armbrust, C. (2011). Off-road robotics - an overview. *KI - Künstliche Intelligenz*, 25(2):109–116.

Farhadi, A., Endres, I., Hoiem, D., and Forsyth, D. (2009). Describing objects by their attributes. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1778–1785. IEEE.

Fleischmann, P., Pfister, T., Oswald, M., and Berns, K. (2016). Using openstreetmap for autonomous mobile robot navigation. In *Proceedings of the 14th International Conference on Intelligent Autonomous Systems (IAS-14)*, Shanghai, China. Best Conference Paper Award - Final List.

Gupta, R., Kochenderfer, M. J., Mcguinness, D., and Ferguson, G. (2004). Common sense data acquisition for indoor mobile robots. In *AAAI*, pages 605–610.

Hitzler, P., Krotzsch, M., and Rudolph, S. (2009). *Foundations of semantic web technologies*. Chapman and Hall/CRC.

Horrocks, I., Patel-Schneider, P. F., Boley, H., Tabet, S., Grosof, B., Dean, M., et al. (2004). Swrl: A semantic web rule language combining owl and ruleml. *W3C Member submission*, 21:79.

Ingrand, F. and Ghallab, M. (2017). Deliberation for autonomous robots: A survey. *Artificial Intelligence*, 247:10 – 44. Special Issue on AI and Robotics.

Kulkarni, G., Premraj, V., Dhar, S., Li, S., Choi, Y., Berg, A. C., and Berg, T. L. (2011). Baby talk: Understanding and generating image descriptions. In *Proceedings of the 24th CVPR*. Citeseer.

Mohammad, M. A., Kaloskampis, I., Hicks, Y. A., and Setchi, R. (2015). Ontology-based framework for risk assessment in road scenes using videos. *Procedia Computer Science*, 60:1532–1541.

Musen, M. A. (2015). The protégé project: a look back and a look forward. *AI matters*, 1(4):4–12.

Nwogu, I., Zhou, Y., and Brown, C. (2011). An ontology for generating descriptions about natural outdoor scenes. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 656–663. IEEE.

Proetzsch, M. (2010). *Development Process for Complex Behavior-Based Robot Control Systems*. PhD thesis, Robotics Research Lab, University of Kaiserslautern, München, Germany.

Reichardt, M., Föhst, T., and Berns, K. (2012). Introducing FINROC: A Convenient Real-time Framework for Robotics based on a Systematic Design Approach. Technical report, Robotics Research Lab, Department of Computer Science, University of Kaiserslautern, Kaiserslautern, Germany.

Ropertz, T., Wolf, P., and Berns, K. (2017). Quality-based behavior-based control for autonomous robots in rough environments. In Gusikhin, O. and Madani, K., editors, *Proceedings of the 14th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2017)*, volume 1, pages 513–524, Madrid, Spain. SCITEPRESS – Science and Technology Publications, Lda. ISBN: 978-989-758-263-9.

Ropertz, T., Wolf, P., and Berns, K. (2018a). Behavior-based low-level control for (semi-) autonomous vehicles in rough terrain. In *Proceedings of ISR 2018: 50th International Symposium on Robotics*, pages

386–393, Munich, Germany. VDE VERLAG GMBH. ISBN: 978-3-8007-4699-6.

Ropertz, T., Wolf, P., Berns, K., Hirth, J., and Decker, P. (2018b). Cooperation and communication of autonomous tandem rollers in street construction scenarios. In Berns, K., Dressler, K., Fleischmann, P., Görges, D., Kalmar, R., Sauer, B., Stephan, N., Teutsch, R., and Thul, M., editors, *Commercial Vehicle Technology 2018. Proceedings of the 5th Commercial Vehicle Technology Symposium – CVT 2018*, pages 25–36, Kaiserslautern, Germany. Commercial Vehicle Alliance Kaiserslautern (CVA), Springer Vieweg. ISBN 978-3-658-21299-5, DOI 10.1007/978-3-658-21300-8.

Ruiz-Sarmiento, J.-R., Galindo, C., and Gonzalez-Jimenez, J. (2017). Building multiversal semantic maps for mobile robot operation. *Knowledge-Based Systems*, 119:257–272.

Schäfer, B.-H., Proetzsch, M., and Berns, K. (2008). Action/perception-oriented robot software design: An application in off-road terrain. In *IEEE 10th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, Hanoi, Vietnam.

Tenorth, M., Kunze, L., Jain, D., and Beetz, M. (2010). Knowrob-map-knowledge-linked semantic object maps. In *Humanoid Robots (Humanoids), 2010 10th IEEE-RAS International Conference on*, pages 430–435. IEEE.

Thrun, S., Montemerlo, M., Dahlkamp, H., Stavens, D., Aron, A., Diebel, J., Fong, P., Gale, J., Halpenny, M., Hoffmann, G., Lau, K., Oakley, C., Palatucci, M., Pratt, V., Stang, P., Strohband, S., Dupont, C., Jendrossek, L.-E., Koelen, C., Markey, C., Rummel, C., van Niekerk, J., Jensen, E., Alessandrini, P., Bradski, G., Davies, B., Ettinger, S., Kaehler, A., Nefian, A., and Mahoney, P. (2006). Winning the darpa grand challenge. *Journal of Field Robotics*.

Valada, A., Vertens, J., Dhall, A., and Burgard, W. (2017). Adapnet: Adaptive semantic segmentation in adverse environmental conditions. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4644–4651.

Wolf, P., Ropertz, T., Berns, K., Thul, M., Wetzel, P., and Vogt, A. (2018a). Behavior-based control for safe and robust navigation of an unimog in off-road environments. In Berns, K., Dressler, K., Fleischmann, P., Görges, D., Kalmar, R., Sauer, B., Stephan, N., Teutsch, R., and Thul, M., editors, *Commercial Vehicle Technology 2018. Proceedings of the 5th Commercial Vehicle Technology Symposium – CVT 2018*, pages 63–76, Kaiserslautern, Germany. Commercial Vehicle Alliance Kaiserslautern (CVA), Springer Vieweg. ISBN 978-3-658-21299-5, DOI 10.1007/978-3-658-21300-8.

Wolf, P., Ropertz, T., Oswald, M., and Berns, K. (2018b). Local behavior-based navigation in rough off-road scenarios based on vehicle kinematics. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 719–724, Brisbane, Australia.

Yao, B. Z., Yang, X., Lin, L., Lee, M. W., and Zhu, S.-C. (2010). I2t: Image parsing to text description. *Proceedings of the IEEE*, 98(8):1485–1508.

Ziegler, J., Bender, P., Schreiber, M., Lategahn, H., Strauss, T., Stiller, C., Dang, T., Franke, U., Appenrodt, N., Keller, C. G., Kaus, E., Herrtwich, R. G., Rabe, C., Pfeiffer, D., Lindner, F., Stein, F., Erbs, F., Enzweiler, M., Knoppel, C., Hipp, J., Haueis, M., Trepte, M., Brenk, C., Tamke, A., Ghanaat, M., Braun, M., Joos, A., Fritz, H., Mock, H., Hein, M., and Zeeb, E. (2014). Making bertha drive – an autonomous journey on a historic route. *IEEE Intelligent Transportation Systems Magazine*, 6(2):8–20.