Advanced scene aware navigation for the heavy duty off-road vehicle Unimog

Patrick Wolf, Axel Vierling, Thorsten Ropertz, Karsten Berns
Robotics Research Lab, Department of Computer Science, TU Kaiserslautern, P.O. Box 3049, 67663 Kaiserslautern, Germany
E-mail: {patrick.wolf, vierling, ropertz, berns}@cs.uni-kl.de

Abstract. This paper describes a novel approach for scene aware navigation in rough off-road environments using behavior-based control, graph-like world descriptions, and ontologies to determine correlations of surrounding objects and environmental properties. A geometry-based obstacle segmentation and convolutional neural networks for classification determine available scene items integrated into low and higher-level scene representations that are processed by a semantic control. Additionally, the surface’s shape, vehicle kinematics, and dynamics and object properties are considered for the planning of low-level trajectories that are hierarchically integrated into the framework. The approach was implemented using behavior-based control and was evaluated using a Unimog U5023 off-road truck in simulation and real-world scenarios.

1. Introduction
Self-driving on-road vehicles are getting more and more reliable and safe [1, 2]. In contrast, autonomous off-road navigation in cluttered environments remains a still unsolved problem. Various research approaches present specific solutions [3, 4], but no general answer to the off-road problem is available by now. Even though related application areas such as search and rescue, construction, or agriculture, can enormously benefit from high automation degrees. Additionally, legal challenges are less restrictive for off-road autonomy, so that commercial solutions are likely to be earlier deployed on the market.

The research project Autonomous Driving of Commercial Vehicles in Off-Road Environments on the Example of the Unimog focuses on the development of control and perception concepts for complex commercial vehicles which operate in such environments. As a result, a focus is safe and robust navigation within cluttered, unstructured areas. The demonstrator, an Unimog U5023 special truck, exhibits extreme off-road mobility, enabling maneuvers that are restricted to other vehicles. Therefore, resulting perception and control concepts can be scaled-down towards less challenging surroundings and can be easily transferred to other mobile systems.

Critical aspects of robust and safe control are navigation and environmental perception. Missing structures, three-dimensionality, and versatility of rough environments make the perception particularly challenging. Also, frequently changing conditions disturb sensors and exacerbate correct scene interpretation. Therefore, commonly used two-dimensional on-road representations cannot be easily adapted to off-road robotics. So far, behavior-based control

1 The project work has been financed by the European Union from the European Regional Development Fund (ERDF/EFRE).
systems (BBS) are suitable for managing operations within uncertain environments. BBS consists of a massive amount of relatively simple behaviors that realize the complex overall system behavior by continuously interacting. Dynamic arbitration mechanisms and overlapping functionality enable high robustness against unforeseen situations and changes [5].

Nonetheless, semantic scene interpretation is another crucial challenge for robust, safe, and intelligent robot behavior. Even though reactive control provides robustness, specific scene correlations have to be interpreted correctly to enable more convenient control decisions. For instance, dynamic obstacles should be considered differently than static impediments to improved safety. Also, slip factors, or the robot’s stability within a scene differs tremendously for different ground material properties. Deep Convolutional Neural Networks provide sufficient object detection and segmentation capabilities to classify a scene. Nonetheless, a typical data structure for high-level semantic processing is required, which seamlessly integrates into the existing behavior-based control. It has to support enough abstraction to provide real-time performance, store many scene objects, and enable accessing and processing such information. Semantic information has to be directly processed by the robot control. Additionally, control and perception processes should be decoupled by using standardized interfaces to enable independent detection, tracking, segmentation, and semantic planning on abstract object properties.

The paper is organized as follows: Section 2 presents related work on scene interpretation using path segmentation and deep learning. Additionally, an overview of ontologies in mobile robotics and scene representations is provided. The integrated Behavior-Based Control (iB2C) architecture is introduced in Section 3. Next, different segmentation strategies are presented in Section 4. Low-level, geometry-based obstacle detection and semantic interpretation based on deep learning are considered. Corresponding data structures for low-level surface and obstacle representations are stated in Section 5, where kinematic properties of the Unimog U5023 are considered to evaluate the current pathway. Section 6 presents a structure for abstract scene representation and interpretation which stores high-level scene objects and properties. It uses an ontology to interpret the current scene semantically. Section 7 presents a behavior-based implementation of the framework concerning off-road mobility. An evaluation of the proposed control and perception concepts is stated in Section 8, where simulation and real-world experiments are presented. Finally, Section 9 summarizes and discusses the results and provides an outlook on future work.

2. Related Work

Image segmentation, scene interpretation, and knowledge representations have been widely studied in the literature. In the following, related work to deep learning and path segmentation, ontologies for autonomous driving, and data structures for scene abstraction are presented.

2.1. Path Segmentation and Deep Learning

One can separate the road detection task into two main classes: vanishing and appearance-based approaches. The vanishing point approach tries to find a vanishing point that can also be used to control the vehicle directly. For example, in [6], Gabor wavelet filters are used to compute individual pixels’ texture orientation. This information is then used to compute an intersection of each pixel’s orientation. This idea can also be extended to use a voting scheme to determine the vanishing point, see [7]. If this point is used to control the vehicle, it is a comprehensible approach.

If not, then the second approach seems to be the more promising one. In 2005 an algorithm that focused on the texture in front of the vehicle was used to win the DARPA challenge [3]. In 2008 [8] used Deep Belief Networks to make use of the whole image and therefore enable the system to detect pathways in a greater distance. In [9] additional information about the ground plane is used to improve navigation results.
More recently, appearance-based approaches had an enormous increase in popularity due to the rise of Convolutional Neural Networks (CNNs). Here, the idea is to find out which parts of the image belong to the road and assign each pixel a label. In this way, more advanced knowledge about the scene can be included. The first promising result was [10], where a Deep CNN was used to segment road scenes in an urban environment. In 2017 a DCNN was also used to directly navigate a robot in a test environment using Deep Reinforcement Learning, see [11]. Since the off-road environment is highly unstructured, it took some time to transfer these results into the outdoor domain. In the past, also, this problem has seen some sophisticated solution approaches. In [12], a set of different experts is used to segment on- and off-road paths. Each expert is a Deep CNN and is trained on a different modality. A fusion of each of these experts takes place, which makes the approach highly robust to different lighting and environmental conditions. The segmentation distinguishes between six different semantic classes: Trail, Grass, Vegetation, Obstacle, Sky and Void.

2.2. Ontologies for Autonomous Driving

Ontologies are commonly known in the context of the semantic web. Thereby, terms are extended by additional semantic information. The same concept is applicable for describing scenes, as presented in [13]. In contrast to other purely object-centric image description techniques, the authors propose a holistic scene ontology. Thus, not only properties of individual entities are described, but also binary and tertiary relationships between relevant primitives.

Ontologies are also commonly used for high-level planning in the area of autonomous driving in on-road scenarios. The authors of [14], for example, propose a high-level ontology-based abstract world model representing intersections and their properties to enable an efficient decision-making process based on simple rules. Instead of building up a single model comprising low-level geometric data as well as its interpretation, a hierarchical model is introduced that provides low-level information like geometrical data for the low-level spatial trajectory planning and separates the high-level interpretation into a topological model incorporating additional semantic data like speed limits and turning restrictions. The higher abstraction level perfectly matches the high-level planning requirements and thus increases the planning efficiency while making the whole process more robust at the same time. Due to the strongly structured environment in on-road scenarios, the relatively strong abstraction of geometric information suits the planner’s needs, but in unstructured off-road environments, the information loss is too high by introducing just two abstraction levels. Instead, the approach proposes a more flexible representation that allows for adjusting the abstraction level according to the current needs.

2.3. Scene Representations

Scenes can be represented in various forms depending on their level of abstraction. Simple scene representations are, for example, basic occupancy grid maps [15]. They provide geometrical information describing the static scene and are especially suitable for low-level path planning. But missing information about individual objects and dynamics makes them more or less useless in more challenging environments like strongly unstructured off-road environments.

Addressing individual objects and their properties on different abstraction levels are especially important in 3D games. Scenes in today’s computer games are so complex that advanced forms of scene representation and management have been developed. The OpenSceneGraph (OSG) [16], for example, allows for managing a massive number of complex objects consisting of sub-objects. It provides a high-level graphics application interface and consists of a set of nodes organized in a hierarchical tree data structure to speed up the rendering process. OSG is used in various applications like virtual reality, simulation, and geolocalization [17]. By defining rather generic entities in this modular way, arbitrary information about individual objects can be stored and reused. But OSG is particularly tailored to rendering and comes with a vast
feature set incorporating visualization functions using OpenGL. In contrast, the work at hand focuses on the planning aspect and exploits the basic concept and data structures while tailoring the actual implementation of robot navigation.

3. Integrated Behavior-Based Control (iB2C)

The integrated Behavior-Based Control (iB2C) architecture [18, 19] 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 relatively 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 behaviors are suited for sensing and data processing by considering respective data quality information (see Fig. 1).

![Basic iB2C units](image)

**Figure 1:** Basic iB2C units [19].

BBS are robust against environmental changes due to the partially overlapping functionality and the ability to adapt to the surroundings using dynamic arbitration. Contradicting control and perception information is resolved through fusion modules that coordinate network components and combine parallel data flows. All iB2C components provide common standardized interfaces consisting of stimulation $s$ and inhibition $i$, which allows the adjustment of the maximum relevance of a module in the current system state. The target rating output $r$ indicates the behavior’s contentment 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 to activate or inhibit other network elements. 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 the highest activity, or respectively best data quality, is forwarded. The latter admits influence concerning the total activity ratio of every connected module.

4. Obstacle and Semantic Segmentation

The robot navigation and perception are arranged hierarchically into levels. On the lowest perception level exists a simple obstacle segmentation to enable a fast and reactive control. More abstract properties are determined on upper levels and processed by high-level controllers to enable a smart and scene aware navigation.

The demonstrator vehicle’s perception system uses a mix of several stereo cameras, 2D- and 3D-laser scanners (see Fig. 2). 2 Stereolab ZED cameras are mounted to the front and rear of the U5023. Additionally, 2 RealSense D435 monitor to the front right and left corners. The front sensors are supported by a Robosense RS-LiDAR-16 and SICK LMS 511. Similarly, another SICK LMS 511 faces to the rear. The vehicle sides have 2 SICK TiM 561 scanners and 2 Robosense RS-BPearl 3D lasers attached. All sensors create a 360 degree distance data
Figure 2: The autonomous robot Unimog U5023 for rough off-road navigation.

Figure 3: Laser sensor view in down-hill scenario.

around the vehicle (Fig. 3). In general, point cloud generation is not restricted to those sensors and might be extended, for instance, by radar or other data of depth-sensing devices.

Obstacle Segmentation

The lower level obstacle segmentation determines objects on a geometry base to detect missing high-level semantic segments. The segmentation pipeline includes a data quality estimation to remove low quality, distorted, or error-affected data [20]. The very dense stereo point cloud is downsampled using a hash-based filter approach to reduce computation times. As a result of this, point duplicates are identified by discretizing input data and building a 3D-hash of each point. Therefore, each duplicate point can be found in constant time. Since laser data is more sparse, no additional downsampling is applied. Next to sensor data time-stamp checks, depth error models for the available sensors are applied, and each measurement is annotated by a quality value [19]. In the specific example of the Unimog U5023, RealSense D435 [21], and ZED [22] error models have been applied. The laser evaluation uses either a ray-based scatter model (SICK) or constant accuracy (Robosense) specified by the sensor manufacturer. A Sigma Filter processes the individual point clouds, which removed low-quality points before the point clouds are combined. Now, low-quality points exceed the target deviation value required for the perception task (here obstacle segmentation). Typically, laser-data have a higher sensing precision than stereo distance data. Higher quality data replace lower quality information in the same spatial region during the combination step of the point clouds. The obstacle segmentation follows the point merge step. The segmentation is realized by combining the approaches of [23] and [24]. A local grid is used to determine the point density and distribution to estimate the surface’s shape. With this, clusters of non-surface points are recognized as obstacles. Obstacle data is passed to a fast and reactive low-level control [25].

Semantic Segmentation

For higher-level scene understanding, surrounding objects are detected and classified. Therefore, a first semantic segmentation of an input image is created. In such a segmented image, where every pixel is assigned to one class, one can not necessarily distinguish between multiple objects of the same category if they are next to each other in the image plane. Therefore, distance data distinguishes between such objects of the same class. Of course, this approach will not match the exact number of objects, but this does not limit the given process. If two objects of the same class, e.g., two stones, are not distinguishable by the image and the point data, they are interpreted as one single obstacle without losing any information regarding the navigation. A similar approach to [12] is used to create the segmentation, where two expert neural networks construct the segmented image. Both networks use the AdapNet architecture and are trained on a different modality: RGB data and distance data. Instead of using the given
six classes for their semantic segmentation, namely Void, Trail, Vegetation, Grass, Obstacle and Sky, a total number of eight classes are available by adding a 7th and 8th semantic class Light Vegetation and Light Obstacle. The demonstrator, the U5023, is a capable off-road vehicle and, therefore, able to drive over small plants or smaller stones, but of course, a real pathway is preferable. The basic architecture of AdapNet remains, and the same data set is utilized, but with some adaptations. Mainly, two experts are used. One for the RGB data and one for the depth image, since the used system's sensor setup, is similar.

The data set is split up into four different parts, as stated in Table 1.

| Part        | Containing |
|-------------|------------|
| 1 Pretraining | 60         |
| 2 Refined training | 20       |
| 3 Validation  | 10         |
| 4 Testing     | 10         |

![Training cycle of adapted data set.](image)

Where part 1 is directly taken from the dataset used in [12], including the same semantic classes. Part 2 contains the newly added classes. Part 3 is used for validating the training parameters. Part 4 is used for testing the performance of unseen data. The complete training cycle can be seen in Fig. 4.

The newly added semantic classes are quite similar to the previously used ones. Therefore, visual features do not differ much, and transfer learning is used to introduce the additional classes, see, e.g., [26, 27]. The benefit is that no adaptation of the whole data set is required, saving an enormous amount of time since labeling the data is time-consuming work. The newly created labeling then looks like Fig. 5.

![Original image](image) ![Annotation as in [28](image) ![Adapted Annotation](image)

**Figure 5:** Original image compared to the ground truth of the semantic segmentation according to [28]. The image denotation is as follows: Sky is blue, Road light grey, Vegetation dark green, Grass light green, Light Vegetation ochre, Obstacle black, Light Obstacle dark grey, Void means no annotation.
5. Low-Level Obstacle and Surface Representation

Grid maps are data structures suited for reactive low-level operations. Neighboring cells can be easily traversed and checked for local navigation. Additionally, they act as short-term memory to overcome blind spots and suppress measurement noise by considering previous sensor readings. Furthermore, they decouple perception and control development processes due to a common grid cell representation for arbitrary data exchange [29]. The proposed framework uses multiple low-level representations for reactive control.

Occupancy Map  The low-level obstacle segmentation data is gathered within an occupancy grid map to determine the environment’s traversability. The map content shifts in discrete steps according to the vehicle’s motion [20]. Hereby, there exist the classes Obstacle, Ground, Unvisited. Initially, all grid cells are marked as unvisited (grey), considered as blocked for motion planning. The segmented ground points (green) are added to corresponding map cells. Afterward obstacle regions (red) are inserted for occupancy determination (see Fig. 6a).

Similarly, the distance data quality information is stored within the grid cells (low deviation: ultraviolet, high deviation: infrared). Typically, laser data has higher precision than stereo information but is more, therefore, more sparse. Therefore, stereo data can interpolate in the robot’s near field (see Fig. 6b). The collision prevention systems consider the grid cell quality to adapt braking distances or respective safe steering angles to treat lower quality obstacle detections.

A virtual tentacle sensor evaluates the set of possible trajectories. Each tentacle models a circular arc which corresponds to a possible steering command [30]. The length of each curve states the motion length until an obstacle is reached. The evaluated tentacle set is provided to the control network, which selects the optimum curve for navigation.

Elevation Map  Since the U5023 can navigate within extremely cluttered environments, reactive control purely based on occupancy grid maps is not sufficient to navigate safely within the environment. Therefore, a shiftable elevation grid map is proposed as an additional data structure for reactive planning. Distance data is directly inserted into the map, which averages

Figure 6: Corresponding low level grid representations for reactive control. The current set of possible trajectories is evaluated for obstacles and body frame collisions.

The proposed new dataset is suited to train the given architecture. From this trained segmentation network, all the necessary information is extracted to detect each class’s actual objects.
the height per cell (encoded by shades of grey, cyan for negative slopes) to calculate the 3D surface of the environment (see Fig. 6c). Sensor noise and negative obstacles result in gaps but can be filtered by considering the vehicle’s height and wheel dimensions. A drawback of averaging input points is that overhanging points create steep surfaces which do not exist in reality. Therefore, a point is removed from the input set if it does not intersect with the robot’s working space. Usually, it is only possible to detect negative obstacles correctly if the vehicle moves close towards them. Often occultation expenses to the sensors’ perspective [31]. A simple filter considers the tire diameter and maximum immersion, allowing us to close smaller gaps within the surface map.

**Off-Road Vehicle Unimog** The Unimog U5023 has unique kinematic properties that support off-road navigation. The surface’s shape and vehicle kinematics have a direct impact on the traversability evaluation. Therefore, tire contact points on the surface grid are predicted to estimate the traversability.

Relevant properties are the vehicle’s frame bend, which can change up to 60 cm, and the axle articulation up to 30°. Portal axes enable a high ground clearance of up to 50 cm, and a low center of gravity. The vehicle can climb slopes up to 45°. Maximum embankment angles are 44° (front) and 51° (rear). Ramp and tilt angles are 30° and 38°, respectively. Furthermore, the vehicle’s dimension are 6 m length, 2.48 m width, and 2.818 m height [32].

**Surface Contact considering Vehicle Kinematics** The elevation grid map is used for a body collision analysis on the current trajectory considering the vehicle kinematics [33]. Therefore, the robot’s pathway is evaluated for frame contact points, tiling stability, and the ability to climb smaller obstacles. Due to the off-road capabilities of the U5023, tire contact points are calculated in 3D to matching correctly on corresponding elevation grid cells. They define the vehicle’s 3D orientation on the surface grid. Therefore, vehicle kinematics have to be explicitly modeled (Fig. 7).

![Figure 7: Kinematic properties of U5023 as axle articulation and their impact on surface contact points of tires (red).](image)

The wheel positions are determined by $\vec{w}_{i,j}$ with $i \in \{ \text{front, rear} \}$ and $j \in \{ \text{right, left} \}$:

\[
\vec{w}_{\text{rear},j} = \vec{k} \pm \frac{s_{\text{wheelbase}} \vec{n} \times \vec{d}}{2}
\]

\[
\vec{w}_{\text{front},j} = \vec{w}_{\text{rear},j} + s_{\text{axlebase}} \vec{d}
\]

where $\vec{n}$ is the vehicle’s frame normal, $\vec{d}$ the current direction, $s_{\text{wheelbase}}$ the wheelbase length, $s_{\text{axlebase}}$ the axle base length, and $k$ the kinematic center. The elevation map is used to derive
the current height of each wheel to compute axle orientations $\vec{a}_i$:

$$\vec{a}_i = \vec{w}_{i,\text{right}} - \vec{w}_{i,\text{left}}$$  \hspace{1cm} (3)

There exist two possibilities for contact point determination depending on the currently computed axle shrinking for the maximum possible articulation ($\phi = 30^\circ$ for U5023). Either all wheels touch the ground, or a single tire is lifted to the air. With this, the vehicle’s frame always stabilizes three or more wheels. In the first case, the four wheel contact normal $\vec{n}$ and support $\vec{s}$ are given by

$$\vec{n} = \vec{a}_{\text{rear}} \times \vec{c}_{\text{rear2front}}$$  \hspace{1cm} (4)

$$\vec{s} = \vec{w}_{\text{rear, left}} + \frac{\vec{a}_{\text{rear}}}{2}$$  \hspace{1cm} (5)

where $\vec{c}_{\text{rear2front}}$ is the vector from the center of the rear axle to the center of the front axle, with

$$\vec{w}_{i,\text{rl2fl}} = \vec{w}_{\text{front, left}} - \vec{w}_{\text{rear, left}}$$  \hspace{1cm} (6)

$$\vec{c}_{\text{rear2front}} = -\frac{\vec{a}_{\text{rear}}}{2} + \vec{w}_{i,\text{rl2fl}} + \frac{\vec{a}_{\text{front}}}{2}$$  \hspace{1cm} (7)

where $\vec{w}_{i,\text{rl2fl}}$ points from the left rear wheel $\vec{w}_{\text{rear, left}}$ to the left front wheel $\vec{w}_{\text{front, left}}$. For the second scenario, the three-wheel contact, all possible frame combinations are calculated similarly to the first approach. Thereby, the center of gravity (CoG) is determined for every tire contact combination using axle weights and mass distribution. The CoG solution is selected based on previous frame normals $\vec{n}_{i-1}$ and the smallest angle difference between $\vec{n}_{i-1}$ and the set of normal candidates.

The current trajectory is analyzed by estimating the tire contacts and normals for each tentacle bin. Both are used for body collision checks by comparing grid cell heights against the maximum ground clearance. Additionally, the normal determines whether the vehicle can fall over or violates kinematic constraints. It also provides an estimate for the roughness of the environment, considering the normal angle differences. Similar to the occupancy map, a virtual tentacle sensor provides elevation evaluated data to control.

6. High Level Scene Representation

While the different grid map types allow for safe navigation in unstructured outdoor environments, they have shortcomings in incorporating higher-level information important for sophisticated planning and control. An object-based scene graph is proposed for representing the environment and storing and managing high-level semantic information to extend the robot’s capabilities. Scene graphs are commonly used in 3D computer graphics applications to describe scenes by defining spatial and logical relationships hierarchically.

6.1. Scene Graph

The scene graph structure is depicted in Fig. 8. A scene $t\text{Scene}$ consists of various items derived from $t\text{Item}$. They represent the objects in the scene. Here, objects can be real objects with a physical counterpart or virtual items that can trigger specific behavior, exemplary. Items are addressed by an identifier and have a pose in the scene.

Each item provides an oriented bounding box representing the physical extents of the corresponding real object or the effective range of the virtual object, respectively, to perform spatial organization. While in most scene graph implementations, axis-aligned bounding boxes are used for efficiency reasons, this approach uses oriented bounding boxes since the actual extents of objects in all directions are usually not measurable at once due to the limited
Figure 8: Class diagram of the scene graph.

Figure 9: Grid map and corresponding bounding box (BB) representations of objects.

perspective. Additionally, the traversable space in rough outdoor environments is strongly limited such that the coarse approximation limits the area too strong as shown in Fig. 9. Thus, the better approximation using oriented bounding boxes is exploited with the downside of higher computational costs. But in contrast to scenes in simulation scenarios and visualizations, a coarse grain object segmentation is sufficient for planning. E.g., a perceived bush can be represented as a single object and need not be further subdivided up to individual leaves. Hence, the number of objects in the scene representation is much lower, absorbing efficiency losses.

In addition to spatial information, information concerning the dynamics and time is required. Therefore, the timestamp of the last detection and the corresponding sensor of the previous detection is stored, which allows for updating the scene and removing ghost objects that are only temporary anomalies in sensor data. Storing timestamps also enables the estimation of dynamic properties like velocities and trajectories over the navigation approach’s time.

The scene provides an interface to the objects and propagates changes in the scene to scene observers `tSceneListener`. Examples of scene observers are different views into the scene, which only needs to change if the scene also changed.

Scene managers `tSceneManager` are introduced to access the scene efficiently and tailored to the planning modules’ needs. They also keep track of changes in the scene and organize items for efficient access. One example is the Octree scene manager, which uses Octree space partitioning [34] to access objects based on their position and extents as defines by the bounding boxes.

New objects are entered into the scene by performing segmentation of 3D distance data. Additionally, semantic information provided by classification approaches is considered and associated with data segments. Afterward, the physical extents of the segments are calculated as well as their possible center. Then, the scene is checked for objects close to the detected one.
Those objects’ shapes and classes are compared to decide whether the item is a re-detection. A new object is entered into the scene graph, and a unique id is assigned. Otherwise, the already present object is updated to incorporate new information.

6.2. Ontology
Navigation in rough off-road environments requires knowledge about scene and relations between objects in the scene in addition to the relatively superficial and straightforward sensor data. This additional knowledge can be incorporated into the navigation system by exploiting ontologies. Classification algorithms as described in Section 4 can be applied to derive additional information about objects from sensor data. Identified object classes are attached to the scene object representation (items). After updating the scene representation and applying the classification result, the scene interpretation engine (SINE) [35] processes the scene and correlates object properties as well as temporal and spatial relations of objects in the scene to derive additional knowledge based on predefined rules. Thereby, assumptions about the environment that humans learned over time can be incorporated. E.g., we know that if we detect a path that seems to be driven rather frequently, then any vegetation on this path is usually flexible and thin such that they should not be treated as obstacles. Thus, if a bush is perceived and entered into the scene representation while the vegetation is on top of a detected road or path, then SINE marks the item as traversable.

7. Navigation
The navigation approach was implemented using behavior-based control and is embedded into the REACTION architecture for safe and robust navigation within rough off-road environments [36].

![Figure 10: Implementation of the perception system using iB2C.](image)
Figure 11: Control incorporates scene knowledge to suppress unindented slow downs.

The iB2C perception framework (Fig. 10) evaluates data for quality assessment through percept behaviors. First, Stereo Images and Laser Data are provided by the Hardware interface. The camera data is used to generate a stereo Point Cloud, which is analyzed for Diparity error and quality filtered using a Sigma Filter. Similarly, laser data is analyzed for Scatter and also filtered. The Combined Point Cloud is fed into the Elevation Map. The map considers the robot’s position and orientation data for map shifts and computes the Elevation Tentacles set that considers the kinematics of the U5023. Analogous, depth information is used for Obstacle Segmentation. Object and ground points are inserted into the Occupancy Map, based on which the Obstacle Tentacles are calculated. Both low-level tentacle sets are provided to a Reactive Control, which evaluates the best trajectory. In addition to lower-level perception, image and point data are used by the Semantic Segmentation to extract (light) obstacles, pathways, grass, (light) vegetation, and sky. The corresponding semantic classes and obstacle segments are inserted into the Scene Graph, which processes the scene using SINE. The Scene Awareness evaluates the high-level scene representation, which influences the low-level reactive control and processes High-Level Commands from higher planning units. Scene awareness can override specific low-level commands by exploiting ontology evaluated world knowledge stored in the scene graph.

The iB2C control system evaluates a set of object properties of the Scene Graph. An extract is depicted in Fig. 11, where Obstacles, Pathways, and Light Vegetation are considered for control and evaluated by percepts. Every percept’s activity is coupled to an object’s spatial position and its intersection with the currently measured trajectory and may suppress specific low-level slowdown behaviors. An obstacle is ignored if the high-level system labels a similar object with a different class. Nonetheless, the presence of close Obstacles suppresses this mechanism. Additionally, obstacles which are detected by the Obstacle Tentacles or Elevation Tentacles but not by the Scene Graph still trigger a Slow Down or respectively an evasion. Missing scene
objects can result from scene graph temporal and semantic filtering as object merging on higher abstraction levels. Therefore, it can be compensated by the reactive low-level control system. The underlying idea is that both low-level mechanisms provide conservative but safe navigation. Only higher-level of world knowledge may identify spurious or false positive obstacles and suppress them. In general, there exist two slow down mechanisms, an obstacle- and elevation-based. The Obstacle Slow Down is stimulated by Obstacle Tentacles and processes the current velocity, steering commands, and the tentacle set. A nearby Pathway or Light Vegetation inhibits the behavior via the Traversable Obstacle fusion. Similarly, Elevation Tentacles stimulate Elevation Slow Down which can be suppressed only by Light Vegetation. The High Level Velocity provides a Velocity to the Safe Velocity fusion. It is inhibited by the Slow Down fusion, which processes both lower-level representations. The robot’s default behavior is not to move, which is represented by the Stop behavior. It has to be explicitly overwritten by the forwarded Velocity command. Therefore, Slow Down supports the Stop command by suppressing Velocity. The Safe Velocity fusion resolves to contradict motion command by considering their activity via a weighted average fusion. The resulting control values are provided to the fail-safe level and hardware controller.

8. Experiments

The benefits of the proposed approach are presented in various tests where the improved navigation capabilities of the U5023 are shown. The scene awareness enables detecting traversable passages even if low-level map representations contain false-positive obstacles that block the desired trajectory.

Implementation

The presented concepts are implemented using Finroc, a C++, Java-based framework with zero-copy, lock-free implementations, and real-time capabilities [37]. The U5023 is equipped with a set of sensors for environmental and vehicle state perception and actuation of degrees of freedom [38]. The internal vehicle data as speed, steering values, gear, etc. are accessed via the body implement CAN bus. In addition to the distance sensors described in Section 4, a Microstrain 3DM-GX5-25 IMU, an u-blox NEO-7P GNSS are used for localization.

Simulation

The system was tested in simulation and real-world rough off-road scenarios. For simulation, the Unreal Engine 4 (UE 4) was used. The Finroc framework provides an UE 4 engine plugin for data exchange. The kinematic properties as axle-shrinking, gearbox, or adjustable tire pressure of the U5023 have been modeled in detail. So, all described degrees of freedom are also represented in the simulation model. Also, sensors have been modeled to show similar characteristics and sensing errors as the real hardware [39].

Scenarios

The U5023 was tested in various simulated and real off-road scenarios. Real-world test locations are the Daimler test environment gravel pit Otígheim, Germany, as well as the ZAK dumping ground, Kaiserslautern, Germany. The gravel pit is specially designed to test and demonstrate the Unimog’s degrees of freedom and was therefore 3D-reconstructed for a representative simulation testing area [40]. Simulation tests with nearly real-world quality are especially important since the real hardware’s test efforts are very high. Additionally, safety-critical tests as extreme inclination driving or roll-over prevention tests cannot be tested to the possible theoretical limits.

Task and Challenges

The U5023’s task in scenarios was the approach of predefined control points. The points were placed at specific landmarks. Alternatively, the robot navigated
Figure 12: Different simulation and real-world off-road test scenarios for perception and control tests.

Figure 13: Semantic segmentation in gravel pit using a custom depth stencil buffer.

Figure 14: Pathway segmentation of simulated camera images in meadow environment.

 manually, and the control points were recorded on the trajectory. On the gravel pit, a round track of approximately 1.5 km was traveled. At ZAK, multiple round tracks of 500 m each were

Figure 15: Gravel pit perception tests to overcome spurious obstacles and data gaps on pathways.
followed. In both test locations, the vehicle followed existing pathways and rough areas as an unsurfaced floor, obstacles, slopes, rubble mounds, or bump waves.

The autonomous control and perception systems’ frequent occurring issues are false-positive obstacle classifications that cause the low-level safety systems to stop unintended. Often such conditions are easily travelable for a human driver. Examples are suddenly appearing dust clouds, water puddles, or light vegetation. The latter class of obstacles is non-rigid, and a human driver can easily pass even if it sometimes collides with the robot’s working space. Therefore, the tests aimed to avoid unintended stops, evasion maneuvers, or slowdowns caused by spurious obstacles.

Initial tests have been done in simulation (Fig. 14). Here, the semantic segmentation was directly generated from the simulation using a custom depth stencil buffer in UE4. Additionally, a path detection framework was applied. The elevation map functionality, occupancy map, and scene graph were tested before real-world experiments were performed using UE4.

The real-world tests used a limited set of semantic classes as light vegetation, obstacle, and pathway for testing, as depicted in Section 7. Fig. 15 and Fig. 16 show occurrences of safety stop events in the gravel pit and ZAK environments. Due to weather conditions at the gravel pit, there were several water puddles on the pathways, which were considered as an obstacle by the point cloud segmentation. At ZAK, pathways are frequently watered to avoid dust. However,
sudden dust clouds are still a common phenomenon. Further, in both scenarios, light vegetation as small bushes narrowed some passages. It could be observed that the incorporation of scene knowledge suppressed an obstacle on a classified pathway. However, during a safety system suppression, the vehicle’s safety relies on a valid classification/ semantic segmentation result and a correct scene risk estimation to avoid semantically valid objects like other vehicles, rocks, etc.

9. Conclusion and Outlook

This paper presented an approach for behavior-based scene aware navigation on the example of the off-road vehicle Unimog U5023. Low-level perception provides an obstacle segmentation and traversability estimation based on ground elevation data considering kinematic properties as pendulum axes and axle articulation of the U5023. Additionally, higher-level semantic segmentation uses deep learning to classify object classes and provide corresponding information to a high-level scene representation, the scene graph. Ontologies implemented by SINE are used to evaluate scene information and modify scene graph contents. Light obstacles detected by lower-level perception can be ignored if high-level semantic tags them as traversable. Therefore, a reactive behavior implementation on a reduced set of semantic classes was presented. Finally, this approach was tested in simulation and real-world tests as the gravel pit or ZAK environments. The tests showed promising effects on the navigation behavior of the U5023. The robot was able to overcome blocked pathways by considering information about detected ways.

Future work investigates the handling of dynamic obstacles within the scene, as pedestrians or other vehicles. Also, passively dynamic objects as loose rocks or soft sand will be addressed in the future.

References

[1] 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, Brench C, Tamke A, Ghanaat M, Braun M, Joos A, Fritz H, Mock H, Hein M and Zeeb E 2014 IEEE Intelligent Transportation Systems Magazine 6 8–20 ISSN 1939-1390
[2] Aeberhard M, Rauch S, Bahram M, Tanzmeister G, Thomas J, Pilat Y, Homm F, Huber W and Kaempchen N 2015 IEEE Intelligent Transportation Systems Magazine 7 42–57 ISSN 1939-1390
[3] 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, Strothband 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 Journal of Field Robotics Accepted for publication
[4] Maturana D, Chou P W, Uenoyma M and Scherer S 2018 Real-time semantic mapping for autonomous off-road navigation Field and Service Robotics ed Hutter M and Siegwart R (Cham: Springer International Publishing) pp 335–350 ISBN 978-3-319-67361-5
[5] Berns K, Kuhnert K D and Armbrust C 2011 KI - Künstliche Intelligenz 25 109–116
[6] Rasmussen C 2004 Grouping dominant orientations for ill-structured road following Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004. vol 1 pp I–470–I–477 Vol.1 ISSN 1063-6919
[7] Kong H, Audibert J Y and Ponce J 2010 IEEE Transactions on Image Processing 19 2211–2220 ISSN 1057-7149
[8] Hadsell R, Erkan A, Sermanet P, Scoffier M, Muller U and LeCun Y 2008 Deep belief net
learning in a long-range vision system for autonomous off-road driving 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems pp 628–633 ISSN 2153-0858

[9] Fleischmann P, Kneip J and Berns K 2016 An adaptive detection approach for autonomous forest path following using stereo vision 2016 14th International Conference on Control, Automation, Robotics and Vision (ICARCV) pp 1–6

[10] Badrinarayanan V, Kendall A and Cipolla R 2017 IEEE Transactions on Pattern Analysis and Machine Intelligence

[11] Bohez S, Verbelen T, Coninck E D, Vankeirsbilck B, Simoens P and Dhoeidt B 2017 CoRR abs/1703.04550 (Preprint 1703.04550)

[12] Valada A, Vertens J, Dhall A and Burgard W 2017 Adapnet: Adaptive semantic segmentation in adverse environmental conditions 2017 IEEE International Conference on Robotics and Automation (ICRA) pp 4644–4651

[13] Nwogu I, Zhou Y and Brown C 2011 An ontology for generating descriptions about natural outdoor scenes 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops) pp 656–663

[14] Regele R 2008 Using ontology-based traffic models for more efficient decision making of autonomous vehicles Fourth International Conference on Autonomous and Intelligent Systems (ICAS’08) pp 94–99 ISSN 2168-1864

[15] Thrun S and B¨ u A 1996 Integrating grid-based and topological maps for mobile robot navigation Proceedings of the Thirteenth National Conference on Artificial Intelligence - Volume 2 AAAI’96 (AAAI Press) pp 944–950 ISBN 0-262-51091-X

[16] Wang R and Qian X 2010 OpenSceneGraph 3.0: Beginner’s Guide (Packt Publishing) ISBN 1849512825, 9781849512824

[17] Sherman W R, Penick M A, Su S, Brown T J and Harris F C 2007 Vfire: an immersive visualization experience for wildfire spread analysis 2007 IEEE Virtual Reality Conference pp 243–246 ISSN 1087-8270

[18] Proetzsch M 2010 Development Process for Complex Behavior-Based Robot Control Systems Ph.D. thesis Robotics Research Lab, University of Kaiserslautern München, Germany

[19] Ropertz T, Wolf P and Berns K 2017 Quality-based behavior-based control for autonomous robots in rough environments Proceedings of the 14th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2017) vol 1 ed Gusikhin O and Madani K (Madrid, Spain: SCITEPRESS – Science and Technology Publications, Lda) pp 513–524 iSBN: 978-989-758-263-9

[20] Wolf P, Ropertz T and Berns K 2020 Behavior-based obstacle detection in off-road environments considering data quality Informatics in Control, Automation and Robotics vol 495 ed Gusikhin Olegand Madani K (Cham: Springer International Publishing) pp 786–809 iSBN: 978-3-030-11292-9, DOI: 10.1007/978-3-030-11292-9_39

[21] Ahn M S, Chae H, Noh D, Nam H and Hong D 2019 Analysis and noise modeling of the intel realsense d435 for mobile robots 2019 16th International Conference on Ubiquitous Robots (UR) pp 707–711

[22] Ortiz L, Cabrera E and Gonçalves L 2018 Electronic Letters on Computer Vision and Image Analysis 17

[23] Fleischmann P and Berns K 2015 A stereo vision based obstacle detection system for agricultural applications Proceedings of Field and Service Robotics (FSR) (Toronto, Canada)

[24] Zolyński G 2018 Cognitive Maps for Autonomous Machines in Construction and Civil Engineering Ph.D. thesis Technische Universität Kaiserslautern https://www.dr.hut-verlag.de/978-3-8439-3628-6.html, Verlag Dr.Hut, ISBN: 978-3-8439-3628-6

[25] Ropertz T, Wolf P and Berns K 2018 Behavior-based low-level control for (semi-) autonomous vehicles in rough terrain Proceedings of ISR 2018: 50th International
Symposium on Robotics (Munich, Germany: VDE VERLAG GMBH) pp 386–393 iSBN: 978-3-8007-4699-6
[26] Shin H C, Roth H R, Gao M, Lu L, Xu Z, Nogues I, Yao J, Mollura D and Summers R M 2016 IEEE Transactions on Medical Imaging 35 1285–1298 ISSN 0278-0062
[27] Pan S J and Yang Q 2010 IEEE Transactions on Knowledge and Data Engineering 22 1345–1359 ISSN 1041-4347
[28] Valada A, Oliveira G, Brox T and Burgard W 2016 Deep multispectral semantic scene understanding of forested environments using multimodal fusion International Symposium on Experimental Robotics (ISER 2016)
[29] Schäfer B H, Proetzsch M and Berns K 2008 Action/perception-oriented robot software design: An application in off-road terrain IEEE 10th International Conference on Control, Automation, Robotics and Vision (ICARCV) (Hanoi, Vietnam)
[30] von Hundelshausen F, Himmelsbach M, Hecker F, Mueller A and Wünsche H J 2009 Driving with Tentacles - Integral Structures for Sensing and Motion (Berlin, Heidelberg: Springer Berlin Heidelberg) pp 393–440 ISBN 978-3-642-03991-1
[31] Braid D, Broggi A and Schmiedel G 2006 The terramax autonomous vehicle concludes the 2005 darpa grand challenge Intelligent Vehicles Symposium, 2006 IEEE (IEEE) pp 534–539
[32] Daimler AG, Mercedes Benz Special Trucks Daimlerstraße 1, 76744 Wörth, Deutschland 2017 The Unimog. Setting standards for off-road mobility.
[33] Wolf P, Ropertz T, Oswald M and Berns K 2018 Local behavior-based navigation in rough off-road scenarios based on vehicle kinematics 2018 IEEE International Conference on Robotics and Automation (ICRA) (Brisbane, Australia) pp 719–724 dOI: 10.1109/ICRA.2018.8460631
[34] Meagher D 1982 Computer graphics and image processing 19 129–147
[35] Wolf P, Ropertz T, Feldmann P and Berns K 2019 Combining ontologies and behavior-based control for aware navigation in challenging off-road environments Proceedings of the 16th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2019) INSTICC (Prague, Czech Republic: SCITEPRESS – Science and Technology Publications, Lda) pp 135–146 dOI: 10.5220/0007934301350146, ISBN: 9789897583803
[36] Wolf P, Ropertz T, Berns K, Thul M, Wetzel P and Vogt A 2018 Behavior-based control for safe and robust navigation of an unimog in off-road environments Commercial Vehicle Technology 2018. Proceedings of the 5th Commercial Vehicle Technology Symposium – CVT 2018 ed Berns K, Dressler K, Fleischmann P, Görges D, Kalmar R, Sauer B, Stephan N, Teutsch R and Thul M Commercial Vehicle Alliance Kaiserslautern (CVA) (Kaiserslautern, Germany: Springer Vieweg) pp 63–76 iSBN 978-3-658-21299-5, DOI 10.1007/978-3-658-21300-8
[37] 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
[38] Berns K, Wolf P and Hemer S 2020 CVC News 1 4–10
[39] Wolf P, Groll T, Hemer S and Berns K 2020 Evolution of robotic simulators: Using ue 4 to enable real-world quality testing of complex autonomous robots in unstructured environments Proceedings of the 10th International Conference on Simulation and Modeling Methodologies, Technologies and Applications INSTICC (SCITEPRESS – Science and Technology Publications, Lda)
[40] Berns K and Wolf P 2017 CVC News 2 8–11