The 2009 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 11-15, 2009 St. Louis, USA

Topological Large-Scale Off-road Navigation and Exploration
RAVON at the European Land Robot Trial 2008

T. Braun[1], H. Schaefer[1], K. Berns[1]

Abstract— **A large-scale navigation system for autonomous off-road robots is presented which uses a topological map to navigate to a previously unseen target location. During path traversal, the system relies on a local navigation layer to avoid obstacles not modeled in the map. Data from this local layer is abstracted to learn both realistic topological edge cost measures and local traversability maps which allow more efficient route selection during topological exploration. On the topological level, a technique to handle impassable route segments in the map is presented and an exploration strategy that allows to discover new routes to the goal is introduced. The performance of the proposed concept is experimentally validated on the robot RAVON at the 2nd Military European Land Robot Trial 2008.**

I. Introduction

The capability to navigate autonomously to an unseen target position is a key requirement for off-road robots in disaster relief, military, or commercial applications. Practical approaches addressing such scenarios often use aerial images depicting the operation area prior to robot deployment. With this information, a human operator can supply a coarse topological map containing a set of waypoints and routes across apparently traversable terrain. During navigation, the autonomous system must then deal with two problems: It must circumvent local obstacles not modeled in the initial map and it must detect erroneous and perhaps entirely impassable routes (e. g. caused by outdated or inaccurate remote imagery). In this case, the robot must adapt the planned path on a more global scale. Many established long range robot navigation systems approach this task by constructing a detailed, dense global traversability map and by planning paths to the goal using D*-type algorithms. This strategy has been shown to yield good results. However, the handling of large global maps is computationally expensive and depends on accurate robot localization at all times.

In this paper, a large-scale navigation system for autonomous off-road robots is presented which retains the highly abstract, coarse topological world representation throughout the robot's entire operation time in order to avoid these drawbacks. The system plans paths using a topological map and relies on a local navigation layer to avoid obstacles along the way using a metrical short-term memory. The efficiency of global path planning is optimized over time by incorporating the experiences of the local layer into the map as a topological edge cost measure. Local detail is only considered temporarily to steer the robot around obstacles and to guide it to the next waypoint. In case this local strategy fails, more detail is added to the global map. A method is proposed to detect and mark impassable routes in the topological map and a novel exploration strategy is presented which adds new routes if all available paths to the goal have been blocked. The selection of the most suitable route candidate is based on compact local traversability maps, which are attached to each waypoint reached. The proposed navigation strategy avoids the computational costs of detailed global world modeling. Additionally, temporarily inaccurate robot localization is not problematic since global data registration is not required and navigation waypoints are only loosely coupled via topological links.

II. Related Work

Large-scale robot navigation supported by remote imagery has received considerable interest in recent years. [1] presents a method to extract traversability information for robot navigation from a variety of data sources such as overhead visual images, digital elevation maps or LIDAR scans. [2] uses visual images to extract a global traversability grid map by considering intensity variability and slope direction and performs global path planning using the A*-algorithm. [3] proposes a more robust approach based on prior cost grid maps which finds a path that does not require lengthy detours if grid cells along the route turn out to be intraversable. In [4] , an extension for long range path planning in large grid maps is described which interpolates metrical positions to avoid the 'stepped paths' common to other grid-based path planners and reduces the computational cost of D*-based planning by using variable grid resolutions. In contrast to approaches based on large traversability grids, robust long-range navigation or exploration using topological graphs has not gained much attention. [5] presents an approach for accurate waypoint-based navigation using GPS, but does not consider obstacle avoidance or impassable routes. Another topological navigation approach is presented in [6], but here too, the focus is placed on topological place recognition, not on robust navigation. Finally, the robotic systems developed for the Darpa Grand Challenge events in 2004 and 2005 navigated using a series of waypoints [7]. However, since departure from the assigned drive corridor was not required (even prohibited), no long-range topological exploration strategies were published for these systems.

III. Overview

Figure 1 presents a schematic overview of the proposed navigation system. The user can provide an initial topological

[1]Robotics Research Lab (Department of Computer Sciences)
University of Kaiserslautern
Gottlieb-Daimler-Str. Building 48, D-67663 Kaiserslautern / Germany
braun, bhschaef, berns@cs.uni-kl.de

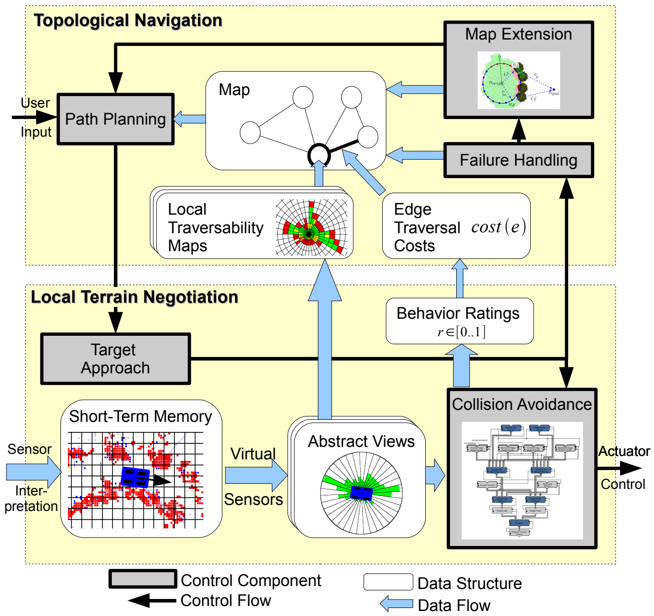
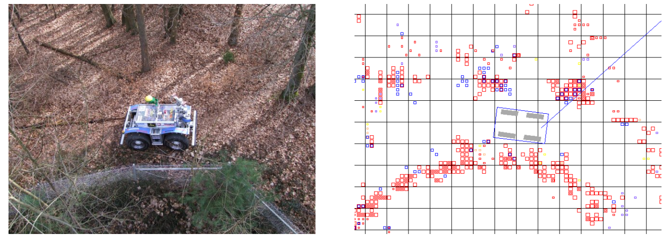
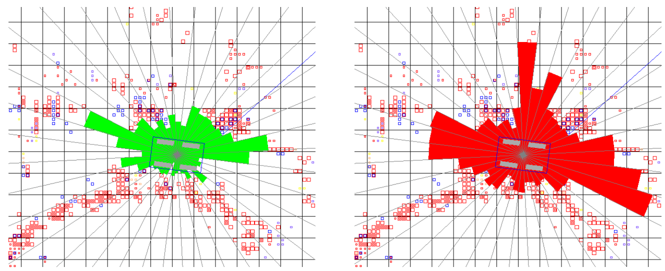


Fig. 1. Overview of the Topological Navigation System



(a) Cluttered off-road terrain (b) Corresponding STM

Fig. 2. Terrain is locally modeled using a grid-map based short-term memory holding terrain traversability properties



(a) View reflecting terrain which is easily traversable at low costs (b) View reflecting terrain which is potentially traversable at high costs

Fig. 3. Virtual sensors generate different views of the STM

map using a GIS tool and then specifies a target node to initiate navigation. On the basis of this map, path planning generates a list of waypoints (all map nodes included in the current path are referred to as *waypoints* from now on), which are subsequently transferred to the local terrain negotiation. The target approach module in this layer then tries to steer the robot to the goal. Along the way, obstacles are circumnavigated by the collision avoidance component which relies on abstract sensor views extracted from a local, metrical short-term memory (Sect. IV). Once a waypoint is reached, abstract views created from the current short-term memory are abstracted into local traversability maps which model the traversability around the waypoint (Sect. V-C). Furthermore, the behavioral activities of the obstacle avoidance modules is used to learn a realistic topological edge cost measure (Sect. V-B). If local navigation does not suffice to reach the target nodes the failure is represented in the topological map and, if this blocks all valid paths to the target node, the map is extended with a promising new route candidate using an explorative map extension strategy (Sect. V-F). After that, the path planner is restarted.

IV. Local Terrain Negotiation

Local terrain negotiation must ensure that the robot remains manoeuvrable at all times. In off-road terrain, this makes local ranking maneuvres indispensable since dead ends and narrow driving situations might not be detectable until actually stuck. Furthermore, limited sensor range and angular coverage requires a frequently updated internal representation of the local terrain around the robot. In order to build this representation, sensor data from scenes such as shown in Fig. 2a is analyzed to detect ground and obstacle structures. Abstract terrain data is then aggregated into a spatial *short-term memory (STM)* (Fig. 2b) in terms of terrain roughness, obstacle rigidity, and height [8]. The STM is realized as an orientation-fixed grid map scrolling in the robot's working coordinate system to yield constant storage requirements and access times. *Virtual sensors* cast specific *aspects*[1] of the terrain into independent abstract views (e. g. Fig. 3a and 3b) which represent the standardized modular interface to the robot's control system [9].

The collision avoidance system is implemented according to the behavior-based architecture iB2C[2] [10]. Particular behavior modules monitor limited portions of the local terrain which are provided in terms of abstract views. Each behavior makes very simple decisions like rotating away from obstacles, backing up when a dead end was found, and preferring easy terrain. Multiple behaviors are combined using standardized fusion modules to form a hierarchical network which shows an emerging overall behavior. For that purpose, behaviors compute meta signals which are used to communicate the behavior's contentment with the current situation (Target Rating r) and its will to contribute to the overall system behavior (Activity a).

V. Topological Navigation

A. Map Definition and Path Planning

Topological path planning is performed using the map stored as a directed graph $G = (N, E)$ holding a set of nodes N and edges E. Each node $n \in N$ specifies the 3D coordinates of one map node, while each edge $e \in E$ is an ordered pair of the edge's start and end nodes with $n, n' \in N \wedge n \neq n'$. For each edge e, a function $\textsc{cost}(e)$ quantifies the traversal cost of that edge. Given a start node n_s and a mission target

[1]Aspects in that context have multiple properties comprising virtual sensor range, resolution, relevant terrain properties, etc.

[2]iB2C → integrated behavior-based control

node n_t, the path planner calculates the cheapest path from n_s to n_t using the dijkstra algorithm and the COST function. The local terrain negotiator is then commanded to approach the waypoints stored in P in sequence.

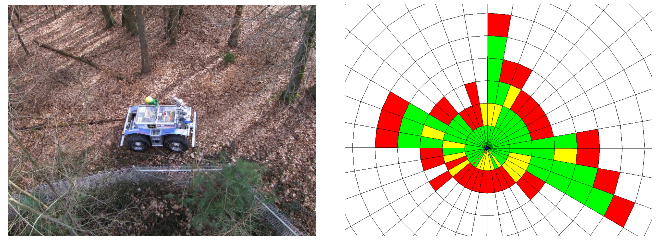
B. Edge Traversal Cost Learning

Upon arriving at a waypoint, the cost of the traversed edge e is estimated in retrospection to improve future path planning. In [11], a method was presented to determine edge traversal costs with continually improving accuracy by observing the target ratings r exported by the low-level control layer behaviors (Sect. IV). The generated cost measure is multi-dimensional and quantifies both the amount of obstacle avoidance reactions along the route as a *risk* cost and the consumed energy as an *effort* cost. The observed costs are subjected to a spatio-temporal integration and a statistical treatment to model the inherent cost uncertainty. Scalar aggregation then yields a single cost value which is the return value of $\text{COST}(e)$.

C. Local Traversability Maps

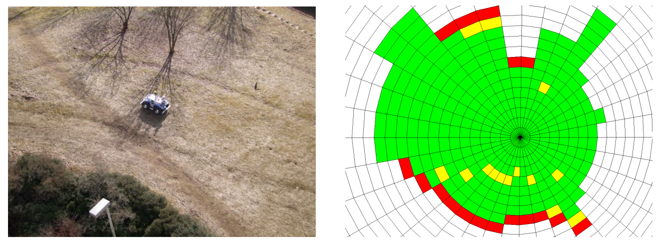
Edge cost learning transfers traversability information acquired by the local navigation layer *between two navigational waypoints* to the topological path planner in the form of an abstract cost ‘summary’. In oder to capture the traversability data *around a waypoint* similarly compact, a set of metrical *local traversability maps (LTMs)* centered at each topological node are generated each time the robot reaches a waypoint. These maps are vital for the exploration strategy (V-F) because they allow the cost estimation of new potential route alternatives. We propose a map layout using polar sectors with constant angular extent $\phi = 10°$. Each sector is subdivided into *seclets* of constant metrical length $l = 1\,\text{m}$ (see Fig. 4b). Compared to a grid map, this partially polar segmentation reduces the impact of growing distance on sensor coverage, as each seclet is covered by an image region of constant width (for cameras) or a fixed number of range measurements (for laser scanners).

To remain compact, each seclet only stores risk and effort *cost modifiers* (r, w) for the covered terrain patch. The modifiers vary from 0 for free, flat terrain to 1 for most likely impassable or extremely steep areas. Additionally, certainty measures (θ_r, θ_w) are added which express the confidence about the modifiers, ranging from 0 for a total lack of information up to 1 for complete certainty. For each node, three different methods are used to fill three separate LTMs. Two techniques use data from a large baseline stereo camera system to conduct a shape respectively appearance based terrain analysis within a range of about 8 – 35 m around the waypoint [12]. As the focus of this paper is not on terrain traversability estimation, these methods are not discussed further. The third method reuses the short-term memory of the local terrain negotiation (Fig. 2b) to extract short range (0 – 10 m) traversability data. For this, two virtual sensors (IV) are instantiated at the waypoint location with 360° field of view and polar sectors with opening angle ϕ. Each sector in the two resulting views contains the distance



(a) Terrain (b) Traversability Map

Fig. 4. Local traversability map of cluttered terrain



(a) Terrain (b) Traversability Map

Fig. 5. Local traversability map of an open field

to the first ‘relevant’ terrain feature. One view considers all terrain features as relevant which cannot be traversed at full speed (Fig. 3a). The second view represents features which cannot be traversed *at all* (Fig. 3b).

Both views are cast into a LTM by setting cost modifiers of 1 for all seclets corresponding to locations where the virtual sensor detects totally impassable terrain, a modifier of 0.5 for seclets covering slowly traversable terrain and 0 for seclets over free space. The confidence measures θ_r and θ_w for each seclet are set proportionally to the seclet area successfully classified by the virtual sensors. Figure 4b shows the LTM created from the two views in Fig. 3. Red, yellow and green seclets represent cost modifiers of 1, 0.5 and 0. Figure 5 shows a second LTM of less cluttered terrain.

D. Topological Edge Cost Prediction

Using the local traversability maps of a map node, traversal costs of up-to-now untraversed topological edges emerging from that node are predicted. This prediction relies on the assumption that the robot will try to travel directly towards the end node of the edge. In this case, all seclets on the direct connection between the edges’s starting node at the map center and its end node will influence the predicted traversal cost. The cost of traversing a single seclet is estimated from its cost modifier values r, w and confidences (θ_r, θ_w) to interpolate linearly between three parameters which quantify risk and effort costs (c_r, c_w) when negotiating:

1) a definitely *free* seclet: $\left(c_r^{\text{free}}, c_w^{\text{free}}\right)$
2) a definitely *blocked* seclet: $\left(c_r^{\text{block}}, c_w^{\text{block}}\right)$
3) an *unknown* seclet: $\left(c_r^{0}, c_w^{0}\right)$

The interpolation ensures that a confidence value of 0 results in the costs $c^{\,0}$ regardless of the (in this case totally unreliable) cost modifier values. As the confidence increases, the resulting costs approach a linear mixture of $c^{\,\text{free}}$ and $c^{\,\text{block}}$, weighted according to the values of the cost modifiers.

Mathematically, this is expressed by

$$c_r = \theta_r(r \cdot c_r^{\text{block}} + (1-r)\cdot c_r^{\text{free}}) + (1-\theta_r)\cdot c_r^0$$
$$c_w = \theta_w(w \cdot c_w^{\text{block}} + (1-w)\cdot c_w^{\text{free}}) + (1-\theta_w)\cdot c_w^0$$

Suitable values for c^0, c^{free} and c^{block} are extrapolated from the cost information learned from previous edge traversals a) averaging the costs of all traversed seclets b) using only costs generated from locations in seclets that were estimated to be free and c) using only costs generated from locations corresponding to blocked seclets.

E. Navigation Failure Handling

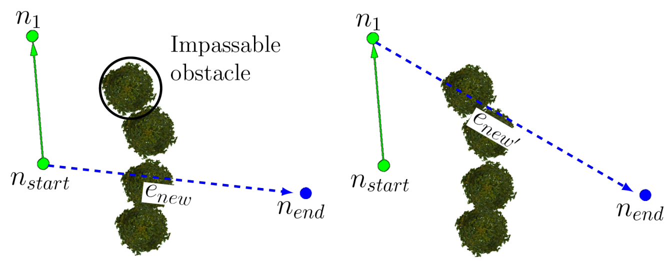
If the local navigation strategy fails to reach a given waypoint, this event needs to be detected and entered into the topological map so that subsequent path planning steps avoid the edge that lead to the failure. Then, a different path must be calculated, or, if no valid paths are found, a new connection needs to be explored. We propose to detect traversal failures by limiting the time during which the robot is *not approaching the waypoint* with a minimum velocity v_{min}. The time limit t_{max} and v_{min} are set according to the amount of clutter expected on the way. In order to represent a detected traversal failure, all map edges $e \in E$ are extended with a type attribute $t \in \{\texttt{speculative}, \texttt{traversable}, \texttt{impassable}\}$ with the following interpretation: A `speculative` edge has not been traversed yet and it is thus not known whether the edge is indeed traversable. `Traversable` edges have been successfully traversed at least once and accurate cost information has been generated using the cost learning scheme. Finally, `impassable` edges cannot be negotiated and thus are excluded from further path planning. In order to represent all information that has been gained during the failed navigation attempt of the edge $e = (n, n')$, the following map update is performed:

- Alter edge e to type `impassable`
- Insert node n_{fail} at the robot's current position
- Insert edge (n, n_{fail}) of type `traversable`
- Insert edge (n_{fail}, n') of type `impassable`
- Insert edge (n_{fail}, n) of type `speculative`

The new node n_{fail} and the connecting edges save the partial progress that the robot could make along edge e. Now, the path planner is invoked to generate a new path from n_s to n_t using the updated map. If a path is found, the robot proceeds. Otherwise, the navigator generates a new route using the exploration strategy presented in the next section.

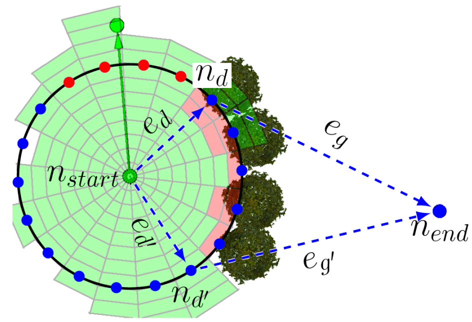
F. Explorative Map Extension

In order to reach a waypoint n_{end} to which no valid path from a node n_{start} can be found, a set of hypothetical new routes is considered which link these previously unconnected nodes. For each route, the total traversal cost is determined using the COST function for `traversable` edges and LTM-based cost prediction for `speculative` edges. Finally, the cheapest new route is selected and inserted permanently into the topological map. The other alternatives are discarded. This ensures that only minimal and necessary additions are introduced into the map. The map extension algorithm considers



(a) Starting from n_{start} (b) Starting from n_1

Fig. 6. Route candidate generation I - Direct connections



(a) Two detour routes starting from n_{start}

Fig. 7. Route candidate generation II - Detour connections

two different classes of possible map additions. First, the insertion of single new edges between already existing nodes is considered. For this purpose, new edges e_{new} between the start node n_{start} or *any node that is reachable from it* and n_{end} are hypothetically inserted into the map. Figure 6 shows two examples of such route candidates. The second set of route candidates is generated by splitting the edge e_{new} into two edges e_d, e_g and placing a *detour node* n_d in between. Thus, if e_{new} originally led from n_{start} to n_{end}, e_d and e_g now connect n_{start} with n_d and n_d with n_{end}, respectively. Different candidates are generated by rotating n_d circularly around the start node at a fixed distance d and using the LTM sector angle ϕ as angular step for the rotation. This takes maximal advantage of the information contained in n_{start}'s LTMs. Figure 7 shows two candidate routes with detour nodes for n_{start} as start node.

Algorithm 1 formalizes the proposed topological map extension approach. PATH(n_1, n_2) denotes a function that computes a path P containing only `speculative` or `traversable` edges leading from node n_1 to node n_2, if one exists. Otherwise, an invalid path with infinite cost is returned. COST(P) is defined as the accumulated cost of traversing a path P computed using the aforementioned combination of learned and predicted edge traversal costs.

G. Preventing Edge Duplication

Algorithm 1 does not insert a new hypothetical edge e_{new} if an edge already exists between the nodes to be linked. This ensures that map extension does not generate duplicate edges and thus, that routes containing impassable edges are not 're-opened' again. Similar to the detection of potential *edge* duplicates, the second part of Alg. 1 checks for *node*

Algorithm 1: Route candidate generation

Data: Graph G = (N, E) *// the global topological map*
Data: Set S *// the set of valid start nodes*

$S \leftarrow \{n_{\text{start}}\} \cup$ all nodes $n \in N$ for which $\texttt{Path}\ (n_{\text{start}}, n)$ exists
Edge $best\_edge \leftarrow$ nil; Float $best\_cost \leftarrow \infty$
foreach $n \in S$ **do** *// main loop*
  // evaluate direct connections ...
  if $(n,\ n_{end}) \notin E$ **then** *// don't create duplicate edges*
    Edge $e_{\text{new}} \leftarrow (n,\ n_{\text{end}})$
    $E \leftarrow E \cup e_{\text{new}}$
    Path $P \leftarrow \texttt{Path}\ (n_{\text{start}}, n_{\text{end}})$
    if $\texttt{Cost}\ (P) < best\_cost$ **then**
      $best\_edge \leftarrow e_{\text{new}}$
      $best\_cost \leftarrow \texttt{Cost}\ (P)$
    end
    $E \leftarrow E\ /\ e_{\text{new}}$
  end
  // evaluate detour node connections ...
  for $i \leftarrow 0$ **to** $2\pi/\phi$ **do**
    $n_d.\vec{p} \leftarrow$ compute detour node position with angle $i\phi$
    // don't overlap existing nodes
    if $\neg\exists a \in N with\, \text{dist}(a.\vec{p}, n_d.\vec{p}) < min\_dist$ **then**
      $N \leftarrow N \cup (n_d.\vec{p})$
      $E \leftarrow E \cup (n,\ n_d) \cup (n_d,\ n_{\text{end}})$
      Path $P \leftarrow \texttt{Path}\ (n_{\text{start}}, n_{\text{end}})$
      if $\texttt{Cost}\ (P) < best\_cost$ **then**
        $best\_edge \leftarrow e_{\text{d}}$
        $best\_cost \leftarrow \texttt{Cost}\ (P)$
      end
      reset E and N
    else
      handle node overlap special case
    end
  end
end
return *(best_edge, best_cost)*

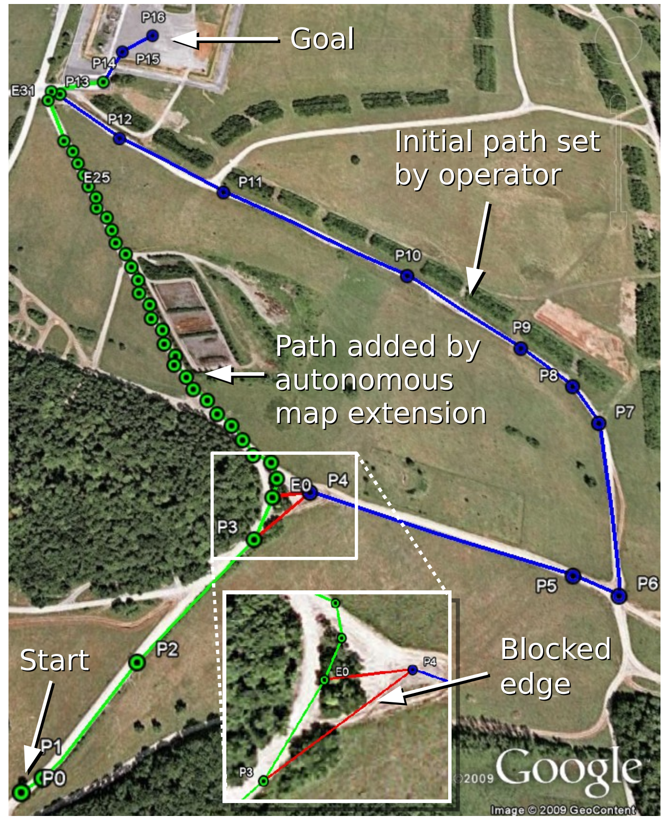


Fig. 8. Autonomous map extension during ELROB 2008

duplications by testing whether the detour node n_d is too close (below min_dist) to an already existing node. If such a situation occurs (marked red in Fig. 7), the current route hypothesis may still be salvaged if e_d and e_g can be connected to the overlapped node instead of the current detour node n_d without creating duplicate edges. If this is impossible, the current hypothesis is skipped. Otherwise, the map extension evaluates the modified route candidate.

H. Triggering Conditions for Map Extension

The presented map extension technique is executed by the topological navigation system every time no valid path can be found for a mission assigned by the user. In this case, the start node for map extension is set to the map entry node n_s and the end node to the target node n_t. Furthermore, map extension is triggered by local navigation failures if the failure handling breaks the last connection to the mission target node. In this case, the map extension algorithm is invoked with the extension start set to the node n_{fail} inserted at the current robot position. The extension end node remains identical to the target node. The map extension algorithm can also be used in a less obvious way to provide improved guidance for the behavior-based local terrain negotiation during the traversal of `speculative` map edges. In this case, the path planner checks the length of a speculative edge e to be traversed against a predefined maximum length $l_{\text{max}} > d$. If the edge is too long, it is removed from the map and the map extension algorithm is called with n_{start} set to the starting node of e and n_{end} set to e's end node. In order to avoid reinsertion of the removed edge e, the direct connection route candidates are disabled in this case. Now, based on the local traversability map of n_{start}, a new detour connection is constructed, which preferably heads towards easily traversable space. This eases the job of the piloting layer and makes the emergence of a well predictable line-of-sight trajectory more likely. It is important to note that this *Speculative Edge Split* method is usually applied several times for a given initial edge, as each split only reduces the length of the second inserted edge e_g by the detour node radius d. In effect, this replaces a long edge with a sequence of shorter segments that follow a path of locally optimal traversability. In essence, this technique implements a local cost optimization strategy on the basis of the traversability information acquired in the LTMs.

VI. Experiments

The presented topological navigation system was tested in a real world scenario on the off-road robot RAVON [13] during the European Land Robot Trial (ELROB) 2008 in Hammelburg, Germany (`http://www.m-elrob.eu/`). Both the map extension and the speculative edge split techniques have been key factors for the success of RAVON during the ELROB 2008. The goal was to travel $\approx 1\,\text{km}$ through a priori unknown terrain, with both start and end point of the track made public just before the start. The GoogleEarth® software was used to set up an initial topological path from the start to the end of the qualification track. RAVON was then commanded to traverse this path fully autonomously. Figure 8 shows a GoogleEarth® screenshot of the terrain covered during the competition together with an overlay of the final map constructed by the robot's navigation system. The small circles indicate topological nodes and the connections between them represent edges. The color scheme is as follows: green lines indicate `traversable` edges and nodes with attached LTMs, while blue and red signifies `speculative` and `impassable` elements, respectively.

Initially, the robot was commanded to travel from the node `P0` on the bottom of the image to the target node `P16` visible on the top, along the route specified by the intermediate `P` nodes. The local navigation layer successfully followed this route until the connection of `P3` and `P4` had to be traversed. Here, a mixture of debris and high vegetation, both not visible in the aerial image, blocked the path. Consequently, the edge traversal failed. In response to that, the topological failure handling algorithm marked the edge as impassable and inserted a new node `E0` at the current robot position. As this broke the only connection to the goal node `P16`, the map extension algorithm was called for the first time. Based on the LTM of `E0`, it inserted a new direct connection from `E0` to `P16`, starting off in the direction of a well traversable dirt road.[3] Upon initiating the traversal of the new connection, the speculative edge split strategy was invoked repetitively and split the extremely long direct connection iteratively into shorter segments with a length of 20 m each (the used detour node radius d). Using the information obtained from the local short-term memory, the new detour nodes were all placed accurately along the well traversable dirt road. The edge splitting terminated once one of the inserted detour nodes coincided with the `P13` node, which subsequently replaced that node to avoid edge duplication. This way, the path planner 'jumped back on track' and followed the originally set waypoints to enter the military compound. At this time, the qualification run was ended by the jury because the time limit was reached. Nevertheless, this successful demonstration of global navigation skills led to the qualification of the team for the final scenario of the competition. This was accomplished by just 3 other teams out of 11 contestants. In the final run, it could re-use both the cost information and the adapted topological structure learned during the qualification and was able to arrive at the target location well within the assigned time frame. RAVON also achieved the highest possible scores for navigation autonomy in both the qualification and the final runs, as it was the only *fully* autonomous system that passed the qualification.

VII. Conclusion and Outlook

This paper presented a novel strategy for long-range robot navigation in off-road terrain. Path planning is performed using an initially user-provided topological map. Path execution is carried out by a local navigation layer which models obstacles only temporarily in order to reduce the computational burden and the dependency on accurate robot localization. To preserve information relevant for global navigation, the short-term memory of the local layer is used to construct compact local traversability maps around map nodes. Furthermore, edge costs are learned from the reactions of the behavior-based local navigation layer. In order to cope with navigation failures, a novel exploration algorithm was proposed which generates new connection hypotheses from two sets of possibilities using the stored traversability information. After evaluation of all valid hypotheses, only the candidate with the lowest predicted costs is actually retained to keep the map as compact as possible.

The proposed navigation system was tested in a real-world scenario during ELROB 2008 and impressively demonstrated the robust interplay between local terrain negotiation and topological navigation. With this strategy, the robot successfully recovered from a traversal failure and autonomously constructed a new path along a well traversable road without relying on *any* previous information about that road or even an explicit road detection strategy. Future work consists of improving the information transfer between local terrain negotiation and the topological map, especially by providing a sounder mapping from traversability to cost modifiers and the accompanying confidence measures.

[3] Although this road is the best track to the goal, the other, more difficult road was initially planned since it would have yielded a higher score.

References

[1] D. Silver *et al.*, "Experimental analysis of overhead data processing to support long range navigation," in *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2006, pp. 2443 – 2450.

[2] A. Howard *et al.*, "Global and regional path planners for integrated planning and navigation: Research articles," *Journal of Robotic Systems*, vol. 22, no. 12, pp. 767–778, 2005.

[3] D. Ferguson *et al.*, "Planning with imperfect information," in *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, vol. 2, September 2004, pp. 1926–1931.

[4] ——, "Multi-resolution field d*," in *Proceedings of the International Conference on Intelligent Autonomous Systems (IAS)*, March 2006.

[5] S. Shair *et al.*, "The use of aerial images and gps for mobile robot waypoint navigation," *IEEE/ASME Transactions on Mechatronics*, vol. 13, no. 6, pp. 692–699, Dec. 2008.

[6] A. Vale, "Mobile robot navigation in outdoor environments: A topological approach," Ph.D. dissertation, Institute for Systems and Robotics, Lisboa, June 2005.

[7] S. Singh, Ed., *Journal of Field Robotics – Special Issue on the Darpa Grand Challenge (Part I)*. Wiley Blackwell, 2006, vol. 23, no. 8.

[8] H. Schäfer *et al.*, "3D obstacle detection and avoidance in vegetated off-road terrain," in *IEEE International Conference on Robotics and Automation (ICRA)*, Pasadena, USA, May, 19–13 2008, pp. 923–928.

[9] ——, "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, 2008.

[10] M. Proetzsch *et al.*, "The behaviour-based control architecture iB2C for complex robotic systems," in *30th German Conference on Artificial Intelligence (KI)*, Osnabrück, Germany, September 2007, pp. 494–497.

[11] T. Braun *et al.*, "Topological edge cost estimation through spatio-temporal integration of low-level behaviour assessments," in *Intelligent Autonomous Systems (IAS-10)*, W. Burgard *et al.*, Eds. IOS Press, 2008, pp. 84–91.

[12] T. Braun, *Cost-Efficient Global Robot Navigation in Rugged Off-Road Terrain*, ser. RRLab Dissertations. Verlag Dr. Hut, 2009, ISBN 978-3-86853-135-0.

[13] C. Armbrust *et al.*, *Emerging Robotics Technology for Humanitarian De-mining and Risky Interventions*. Woodhead Publishing Limited, 2009, ch. RAVON — The Robust Autonomous Vehicle for Off-road Navigation, to be published.