Lecture Notes in Mobility

Tim Schulze
Beate Müller
Gereon Meyer *Editors*

Advanced Microsystems for Automotive Applications 2015

Smart Systems for Green and Automated Driving

EPoSS
European Technology Platform on Smart Systems Integration

VDI|VDE|IT

Springer

Lecture Notes in Mobility

Series editor

Gereon Meyer, Berlin, Germany

More information about this series at http://www.springer.com/series/11573

Tim Schulze · Beate Müller
Gereon Meyer
Editors

Advanced Microsystems for Automotive Applications 2015

Smart Systems for Green and Automated Driving

Springer

Editors
Tim Schulze
VDI/VDE Innovation + Technik GmbH
Berlin
Germany

Gereon Meyer
VDI/VDE Innovation + Technik GmbH
Berlin
Germany

Beate Müller
VDI/VDE Innovation + Technik GmbH
Berlin
Germany

ISSN 2196-5544 ISSN 2196-5552 (electronic)
Lecture Notes in Mobility
ISBN 978-3-319-20854-1 ISBN 978-3-319-20855-8 (eBook)
DOI 10.1007/978-3-319-20855-8

Library of Congress Control Number: 2013936959

Springer Cham Heidelberg New York Dordrecht London

Printed on acid-free paper

Springer International Publishing AG Switzerland is part of Springer Science+Business Media (www.springer.com)

Preface

Green vehicles that provide a high degree of energy efficiency and are powered by electricity or other alternative fuels have been on the agenda of politicians and industry alike since many years now. Public funding of research and innovation in electric vehicles in the European Green Vehicles PPP for example has triggered the development of products, which have been launched successfully to the market recently. At the same time, the topic of automated driving is increasingly raising public attention: highly automated driving, providing yet unsurpassed levels of road safety, efficiency, productivity and social inclusion, seems to be feasible on motorways by the year 2020.

Smart systems combining sensing, cognitive processing and actuation are essential not just for the electrification but also for the automation of the automobile and will remain a subject of research and innovation for quite a while. Particularly for applications in complex environments like city traffic and at higher levels of automation on the way to self-driving capabilities, perception of the driving environment is a challenging task to be addressed, e.g. by multi-sensor systems and sensor data fusion. Furthermore, connectivity and eventually a certain level of artificial intelligence will be needed to ensure the safety of the system. Thus, new links between the smart systems, the Internet of Things and the robotics communities should be created in order to fully embrace the research and innovation needs for the years to come. A first attempt for this has been made earlier in 2015 when the European Technology Platform on Smart Systems Integration (EPoSS) presented a "European Roadmap on Smart Systems for Automated Driving" and contributed to an even broader roadmap activity as a member of the core team of the connectivity and automation task force of the European Road Transport Research Advisory Council (ERTRAC).

The key enabling technologies for the automobile of the future have always been the topic at the International Forum on Advanced Microsystems for Automotive Applications (AMAA) at an early stage. Thus, the topic of the 19th AMAA 2015, held in Berlin on 7–8 July 2015, is "Smart Systems for Green and Automated Driving". The AMAA organisers, VDI/VDE Innovation + Technik GmbH together with EPoSS, greatly acknowledge the support given for this conference, particularly from the European Union through the Coordination Action "Global Opportunities for Small and Medium Sized Enterprises in Electric Mobility" (GO4SEM).

The papers in this book, a volume of the Lecture Notes in Mobility book series by Springer, were written by leading engineers and researchers who have attended the AMAA 2015 conference to report their recent progress in research and innovation. The papers were peer-reviewed by the members of the AMAA Steering Committee and are made accessible worldwide. As the organisers and the chairman of the AMAA 2015, we would like to express our great appreciation to all the authors for their high-quality contributions to the conference and also to this book. We would also like to gratefully acknowledge the tremendous support we have received from our colleagues at VDI/VDE-IT.

Berlin
June 2015

Tim Schulze
Beate Müller
Gereon Meyer

Advanced Microsystems for Automotive Applications 2015

Funding Authority

European Commission

Supporting Organisations

European Council for Automotive R&D (EUCAR)
European Association of Automotive Suppliers (CLEPA)
Strategy Board on Electric Mobility (eNOVA)
Advanced Driver Assistance Systems in Europe (ADASE)
Zentralverband Elektrotechnik- und Elektronikindustrie e.V. (ZVEI)
Mikrosystemtechnik Baden-Württemberg e.V.

Organisers

European Technology Platform on Smart Systems Integration (EPoSS)
VDI/VDE Innovation + Technik GmbH

Steering Committee

Mike Babala, TRW Automotive, Livonia, MI, USA
Serge Boverie, Continental AG, Toulouse, France
Geoff Callow, Technical & Engineering Consulting, London, UK
Kay Fürstenberg, Sick AG, Hamburg, Germany
Wolfgang Gessner, VDI/VDE-IT, Berlin, Germany

Conference Chair

Contents

Part I
Driver Assistance and Vehicle Automation

Autonomous Parking Using Previous Paths

Christoph Siedentop, Viktor Laukart, Boris Krastev, Dietmar Kasper, Andreas Wedel, Gabi Breuel and Cyrill Stachniss

Abstract This paper is about mapping the drivable area of a parking lot for autonomous parking. Manual map creation for automated parking is often impossible, especially when parking on private grounds. One aspect is that the number of private properties is very large and private parking should not be included in public maps. The other aspect is that an owner and operator of a car often has very specific ideas of where the car may be driven. Our approach creates maps using just the previously driven paths. We describe the drivable area through triangles using established methods from Computer Graphics. These triangles are generated by overlaying circles of a certain radius over the driven paths. These circles create a so-called alpha-shape and approximate the drivable area. The description through

V. Laukart developed the methods in relation to α-shapes and the Delaunay triangulation. B. Krastev enabled the search. C. Siedentop wrote the paper and provided the evaluation.

C. Siedentop (✉) · V. Laukart · B. Krastev · D. Kasper · A. Wedel · G. Breuel
Daimler AG, Autonomous Driving, HPC G023, 71059 Sindelfingen, Germany
e-mail: Christoph.Siedentop@daimler.com

V. Laukart
e-mail: Viktor.Laukart@daimler.com

B. Krastev
e-mail: Boris.Krastev@daimler.com

D. Kasper
e-mail: Dietmar.Kasper@daimler.com

A. Wedel
e-mail: Andreas.Wedel@daimler.com

G. Breuel
e-mail: Gabi.Breuel@daimler.com

C. Stachniss
Institute for Geodesy and Geoinformation, University of Bonn, Nussallee 15, 53115 Bonn, Germany
e-mail: cyrill.stachniss@igg.uni-bonn.de

T. Schulze et al. (eds.), *Advanced Microsystems for Automotive Applications 2015*,
Lecture Notes in Mobility, DOI 10.1007/978-3-319-20855-8_1

triangles ("Delaunay triangulation") allows for fast retrieval and easy expansion with new paths. Finally, a simple conversion of the triangulation into a Voronoi diagram enables fast path searching. In this paper we thus present an efficient framework for determining drivable areas and allows searching for a drivable path. Finally, we show that this method enables real-time implementation in an autonomous car and can cope with new obstacles at planning time.

Keywords Autonomous parking · Mapping · Drivable area · Delaunay triangulation · Alpha shapes

1 Introduction

Each owner and user of private property will have individual and varying notions on where she would like her car to drive. Especially for parking it may be perfectly acceptable for one user to drive across grass, and not desirable at all to another driver. This example illustrates how there cannot be one set of rules determining drivable areas. We solve this problem by learning from the driver where the car is allowed to drive. On top of that our approach creates a safeguard on all other sensors that the vehicle might have.

As we hope the benefits are obvious, the challenges should not be taken lightly. We wish to traverse the area as few times as possible but need to integrate new traversals into the drivable area. Finally, our solution must be suitable for a real-time system.

The task can be summarised in the following four steps:

1. Taking a set of reference driving through the terrain and noting where the car was parked.
2. Create a map with the drivable terrain.
3. Plan a new path from a start position to one of the parking spots.
4. Avoid obstacles and re-plan the path as obstacles appear.

In the next section we describe existing research for the task of determining drivable areas. Section 3 describes our approach to this task and makes up the primary contribution of this paper. In Sect. 4 we describe our experimental platform and the experiments we undertook. This is followed by a results and an outlook section.

2 Preliminaries

2.1 Explanation of Terms

2.1.1 Duality in Graphs

Duality in graphs mean that one graph can be converted into the other. The general concept is more refined but for this paper the following explanation may be used: The graph of the Delaunay triangulation becomes the Voronoi graph by placing a vertex inside each triangle and connecting vertices of neighboring triangles with an edge.

2.1.2 A* Search

The A* search algorithm is the graph search algorithm when a heuristic exists. It improves the well-known Dijkstra's algorithm [3, Chap. 24.3] by the use of a heuristic. A heuristic is an oracle that can guess the cost to the goal. A valid and often used heuristic is the straight-line distance to the goal. The A* algorithm is complete in that it will always find a solution if one exists and optimal in that it will find the shortest path in the fastest way possible (measured by nodes expanded and without preprocessing). Necessary is only an admissible heuristic. For an exemplary overview the reader is referred to Wikipedia (*A* search algorithm*) or [15, Sect. 3.5.2] of "Artifical Intelligence - a modern approach". The original paper is [10].

2.2 Related Research

Classifying drivable and non-drivable terrain has often been the focus of robotics research: in the context of all-terrain robots [13, 17], and even autonomous driving [4]. What all these approaches have in common is their effort to predict drivability. This is either done through camera based systems, LIDAR, IR-distance sensors or a combination thereof. We, however, are interested in sensor-less drivability classification. First, because none of these approaches can guarantee perfect precision. For example, Poppinga et al. [13] achieve at best 100 % accuracy but at worst an unacceptable 83 %.

Secondly, we want our approach to work independently of LIDAR. As such the only other ground-sweeping sensors are cameras. Radar and ultrasonic sensors ignore the ground unless mounted in different configurations.

Our usage of α-shapes to compute a drivable map is motivated by several papers in the field, which show their geometric and computational desirability. McCarthy et al. use α-shapes as a map to determine when a path does not exist [11]. Alzantot and Youssrf [2] describe indoor floor plans using—among other things—α-shapes.

A treatment on necessary sampling conditions (specifically density) of points to construct α-shapes is given by Sakkalis and Charitos in [16].

Worral et al. [18, Fig. 6] use the heading information to generate road maps using α-shapes. They adapt the circumcircle test such that points with a heading difference above 90 degrees are ignored. With their method different lanes are constructed which can be used to determine intersections.

2.3 System Setup

The vehicle is equipped with localising sensors. From a combination of DGPS, Radars and cameras a vehicle position is found to an accuracy of 2 cm. Our approach, however, does not depend on any specific accuracy. Instead, any accuracy in localisation propagates through our algorithm.

In addition to the presented algorithm, there is also working obstacle detection. This is a security layer that informs the car of non-static objects that would be encountered from day to day. We later show that new points can be added at runtime easily. Obstacles detected by the sensors are added at runtime and planned around in real time (Fig. 5c).

The vehicle is operating in a publicly accessible parking area, such as a corporate car park, or a single parking spot on domestic property. We tested our approach in both situations.

3 Method

With localisation we collect the vehicle's travelled path which we describe through a left and right polyline. This can be sampled to generate a set of points.

A good representation for these points, which is often employed in Computer Graphics, is to describe the shape as a triangulation. The Delaunay triangulation [6] has good and stable properties, so it is often used for this purpose. These good properties mean that it avoids "skinny" triangles and is the dual to the Voronoi Diagram.

In our approach we first determine a Delaunay triangulation (Sect. 3.1). The boundaries of the drivable area is computed using an α-shape algorithm (Sect. 3.2). Once the Delaunay triangulation (DT) has been cleaned up, the dual to the DT is a Voronoi graph that gives us a searchable graph for path planning (Sec. 3.3). Instead of searching through the state space for the vehicle, we search in its 2D-projection into an Euclidean map—the above Voronoi diagram. We retrieve several paths and check for the optimal trajectory among them. We do so to facilitate various optimality criteria, like maximum distance to objects, minimized jerk, and others. The details of this can be seen in Sect. 3.4.

3.1 From Trajectory to Delaunay Triangulation

Having driven the vehicle around the parking lot or in the vicinity of a single parking space, the localisation module provides us with a number of trajectories. Each trajectory is a set of tuples describing the locations of the vehicle x_i, y_i, its headings θ_i and the corresponding time t_i. It is assumed that the area the car covered while recording the trajectory is safe for driving.

To determine a drivable region these trajectories have to be converted into a representation of the drivable area. Grid approaches are conceivable but we decided to describe the area through polygons. This makes it scale independent with low memory consumption, and in our case allows fast querying.

The polygonal description was picked to be α-shapes [2].

In the following we describe the algorithm employed to generate a so called Delaunay triangulation [6]. The DT is a suitable input to the α-shape algorithm.

Given a set of driven trajectories $\mathcal{T}$ with each trajectory consisting of configuration tuples (x, y, θ, t), we generate a set of points $\mathcal{P}$ relating to the inscribed rectangle of the car. A given configuration is thus transformed to four points.

Next the Delaunay triangulation $DT(\mathcal{P})$ is found on these points $\mathcal{P}$. For this there exists a choice between four algorithms. *Randomised incremental approach*, *Divide & Conquer*, *Sweep Line*, *Sweep hull*, and the *Flip* algorithms [12]. The *flip algorithm* is quadratic in complexity, and is here ignored. Furthermore, the divide and conquer and the sweepline algorithm suffer from numeric instability. We chose the randomised incremental approach [5] because it offers the same $O(|\mathcal{P}| \log |\mathcal{P}|)$ complexity as most other algorithms but additionally allows cheap insertion of further points. This is a property that will be beneficial later on.

Before the algorithm is started, duplicate points must be removed. We do this by lexicographically sorting points and then removing duplicates.

The defining feature of the Delaunay triangulation is that for any two adjacent triangles a flip of the shared edge will not result in a larger minimal angle. Compare Fig. 1 for an example.

3.2 From Delaunay-Triangulation to α-Shapes

The Delaunay triangulation is only an intermediary step to determine the drivable area. α-Shapes are best understood through a special case, namely the convex hull. This case occurs as $\alpha \to \infty$. The more general α-shapes allow concave segments to the hull and even topological holes. The parameter α controls how indented the concave indents may become.

Formally correct, if and only if a disc with radius α can be placed in the plane such that no points lie within it, that dics is not part of the α-hull [8]. Points on the α-hull define the vertices of the α-shape.

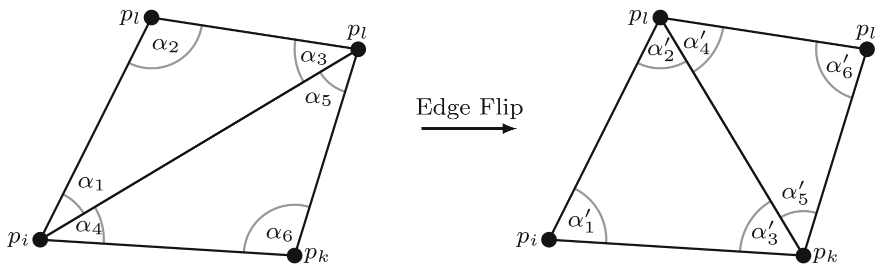

Fig. 1 The Delaunay criterion requires that two adjacent triangles *maximise* the *minimal interior angle*. A simple flip of the shared edge restores the criterion. Specifically, in the *left figure* α_1 is the smallest angle. In the *right-hand figure* α'_4 has become the smallest angle. Because $\alpha'_4 > \alpha_1$ the flip of the diagonal from the *left to the right figure* maximised the minimal angle. Therefore the *right-hand side* is the correct triangulation

The requirements for α-shapes are (1) sufficiently dense sampling of points, and (2) reasonably uniform distribution of points. Both of these requirements are easily met in our application by sampling in uniform distances along the trajectories.

Given a Delaunay triangulation the α-shape can be constructed in a straightforward manner. For each triangle in DT the circumcircle is checked for its radius. If the radius is larger than α, the triangle is not part of the α-shape. Finally, all outer points and its edges are found by marking all triangles as unvisited and then visiting all triangles and recursively marking all neighbours as visited. Only outer vertices need to be kept. Because every triangle is visited exactly once the complete α-shape algorithm possess $O(|\mathcal{P}|)$ complexity.

3.3 Delaunay as the Dual to Voronoi: Easy Search

Some readers will know that the Delaunay triangulation (DT) is the dual to the Voronoi graph. After having purged undrivable areas from the DT, all the triangles make up the drivable area. As such the Voronoi encompasses all, but not more, of the drivable area.

The path along the vertices of the Voronoi diagram can be converted to a corridor as constraints for the optimisation: Selecting the dual edges of the Delaunay triangulation provides us with the left and right boundaries of the corridor.

3.4 K-Shortest Paths*

The A* algorithm (see Sect. 2.1) provides the optimal solution given a suitable modeling of the query. For computational reasons we found that searching in the

2D-space through the Voronoi graph gives a solution faster and within the embedded environment of an autonomous street vehicle. It is self-evident that this will not guarantee an optimal solution. By evaluating several local optima of the 2D-search graph we believe that one of them will provide the global optimimum after optimisation. In practise this yields subjectively optimal paths.

This approach is also suitable when not all costs can be integrated into the search graph. In our case these are, among others, dynamic costs such as lateral and longitudinal acceleration and jerk.

Finally, providing multiple possible paths potentially allows an occupant to make a subjective and individual selection of the path to be taken.

We implemented this through a combination of the A* algorithm [10] with the K-shortest loopless path algorithm from Yen [19]. K-shortest loopless paths was extended by Eppstein [9] and both algorithms were combined by Aljazzar and Leue [1] to $K*$ (Fig. 2).

In Fig. 3 the multiple different solutions are shown.

The multiple solution can then be checked for the final orientation of the car. Those found suitable can be optimised with a local, convex cost function [20]. Of those, the lowest final cost is used and passed to the control system.

4 Evaluation and Results

The presented approach was tested on a Mercedes-Benz E-Class fitted with radar sensors and a stereo camera system. A similar system is described in Dickmann et al. [7]. For this paper it suffices to know that there is complete coverage from radar sensors. We can also build on functioning localisation and obstacle detection.

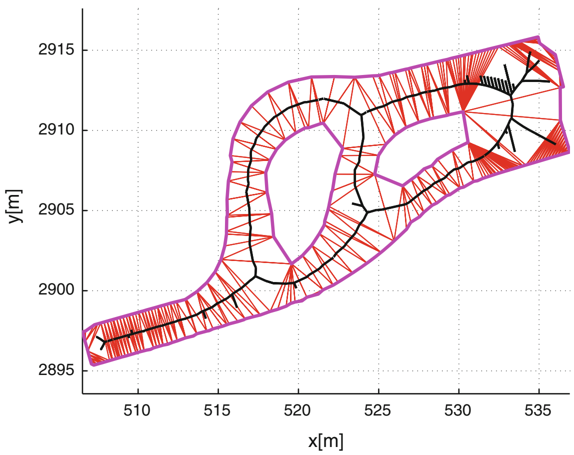

Fig. 2 A parking scenario: The computed drivable area is shown with a *magenta* outline. The Delaunay Triangulation after α-shape cleanup is shown in *red*. Finally, the *black* lines visualise the Voronoi Diagram

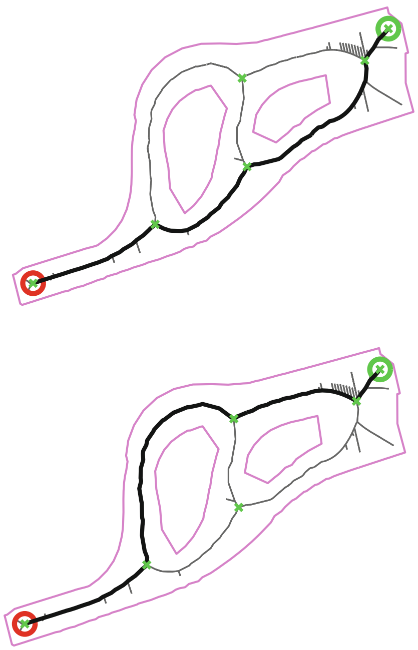

Fig. 3 An example of *k shortest paths* returned for a parking scenario (compare Fig. 2). The searched path is from the *large red circle* to the *large green circle*. *Small green crosses* represent a road graph of the parking lot. The *black lines* indicate the four shortest path along the Voronoi Diagram (shortest *top left*, longest *bottom right*). The *thin magenta line* represents the drivable area boundaries and the dimension is approximately 35 m by 25 m

Our measurements produce good results on two different parking lots of around 50 m × 100 m in size. Please refer to Figs. 4 and 5.

Because of its real-time capabilities the algorithm can cope with obstacles as they appear. When that happens, the α-shape is updated and a new corridor is calculated through the Voronoi diagram. All of this happens at run-time.

As our approach assumes that traversed terrain is safe for driving on, there are cases where the assumption and thus the algorithm could fail. Imagine a structure as typically found in car repair shops. Two narrow planks that the car drives over to be inspected from below. Similar structures can occasionally be found in parking lifts. Here a lateral offset of a few centimeters would turn the same trajectory undrivable. It is important to know that our system cannot cope with such a situation. In most circumstances, however, a parking vehicle will drive where the ground between the left and right tires is drivable as well.

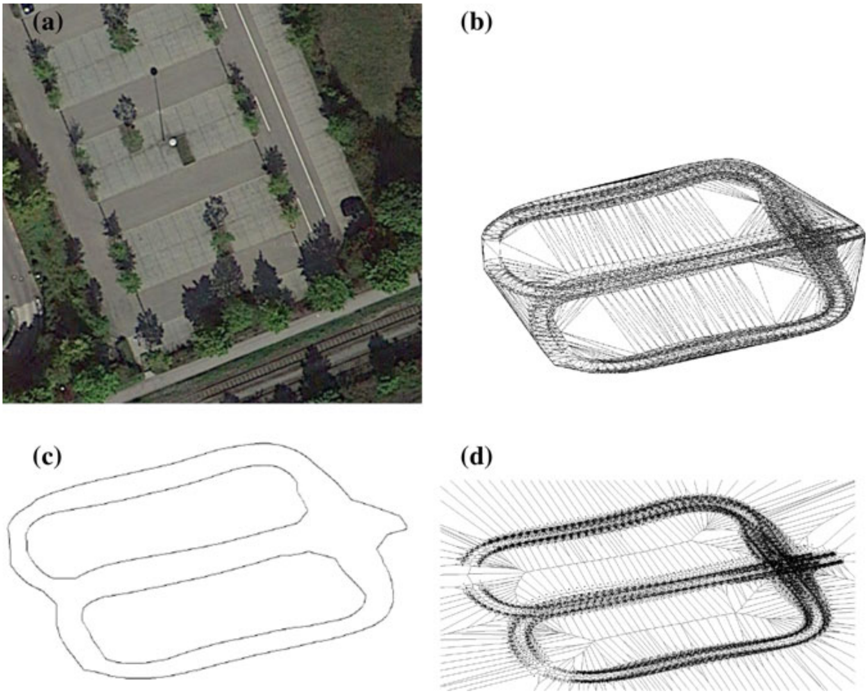

Fig. 4 The steps from driving the trajectory to the final Voronoi diagram. **a** An aerial view of the parking lot (© Google Maps 2015). **b** Delaunay triangulation. **c** α-shape of the same parking lot with an α value of 2.08 m exactly the vehicle width. **d** Voronoi diagram of the parking lot

5 Outlook

There are two omissions in the approach. The first is that the approach cannot detect very thin static obstacles. We call this the thin-fence-problem. If two trajectories were driven parallel to a fence on both sides of the fence the algorithm would falsely classify the railing as drivable.[1]

The second omission is when used as a tool for search. Here our solution is necessary for search but not sufficient. For example, for forward-backward maneuvering we rely solely on the local planner and no costs are taken into consideration during search through the Voronoi. In more general terms, the search through the Voronoi graph generates path candidates. It does not, however, search through the state space of the vehicle which would need to include at least orientation and driving directions.

[1]Points will be generated at $\alpha/2$ to each side of the two center lines. If the paths are less than 2α apart the inner points will be less than α apart and hence merged in the α-shape.

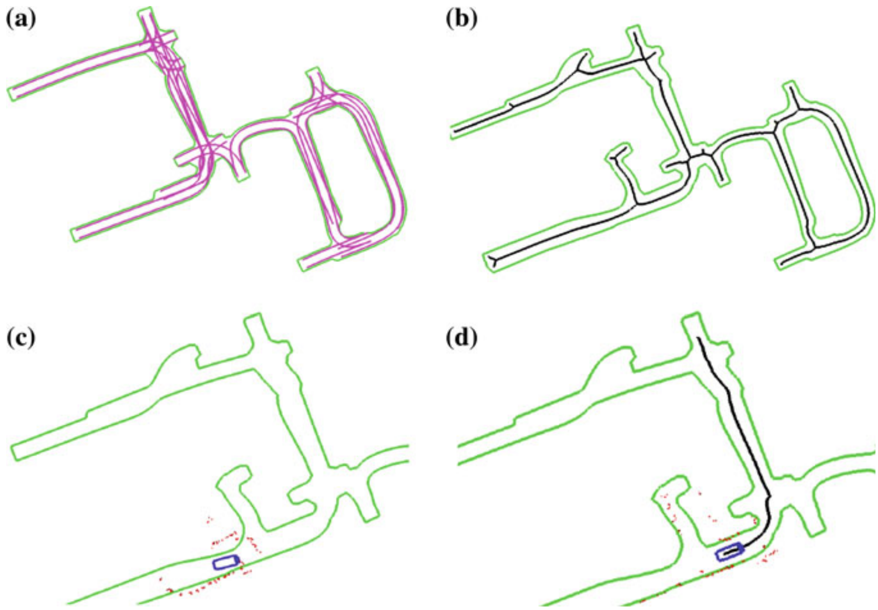

Fig. 5 A different example. **a** Drivable area creation through driven trajectories. **b** *Black center line* shows the Voronoi Diagram. **c** Drivable area has some new obstacles shown here as *red dots*. **d** The vehicle has selected a traversable path

On the other hand, there are many further developments that would benefit from our solution. For example, Ziegler et al. [21] use a Voronoi diagram as a heuristic for the obstacle-aware distance through the map and amend it with a vehicle-kinematic-aware obstacle free distance function as heuristics. It may be beneficial to use the structure of the Voronoi to add an adaptive grid. Rufli and Siegwart [14] have suggested using the major axis of an unstructured environment. We would suggest using the Voronoi diagram to provide a more informed structure than the major axis.

6 Conclusion

In this paper we developed an algorithm which enables a car to autonomously park itself without the prior existence of a map. Our focus was on determining the drivable area and planning a suitable path. We presented an application of the Delaunay triangulation in combination with α-shapes that generates a suitable map representation. We believe that our novel mapping approach is not only helpful in this parking situation but also at other low speed scenarios, especially those where we can rely on sensors to detect drivability in previously visited areas. The benefit of the presented solution is to guarantee never to drive where a user has not been before.

By relying only on driven trajectories and combining this "sensor" with the α-shape algorithm we have shown a feasible system that enables a car to park itself autonomously. Furthermore, our approach naturally lends itself as a structure for path planning and future research: both global planning over the resulting Voronoi diagram becomes simple, and local trajectory optimisation is provided with the appropriate constraints.

We have implemented our approach in an autonomous vehicle. Experimental results show that non-expert drivers are able to teach the vehicle where to drive and where to park.

References

1. Aljazzar H, Leue S (2011) K*: a heuristic search algorithm for finding the k shortest paths. Artif Intell 175(18):2129–2154
2. Alzantot M, Youssef M (2012) Crowdinside: automatic construction of indoor floorplans. In: Proceedings of the 20th international conference on advances in geographic information systems. ACM, pp 99–108
3. Cormen TH, Leiserson CE, Rivest RL, Stein C (2009) Introduction to algorithms, 3rd edn. The MIT Press, Cambridge
4. Dahlkamp H, Kaehler A, Stavens D, Thrun S, Bradski GR (2006) Self-supervised monocular road detection in desert terrain. In: Robotics: science and systems. Philadelphia
5. de Berg M, Cheong O, van Kreveld M, Overmars M (2008) Computational geometry: algorithms and applications. Springer Science & Business Media, Berlin
6. Delaunay B (1934) Sur la sphere vide. Izv. Akad. Nauk SSSR, Otdelenie Matematicheskii i Estestvennyka Nauk 7(793–800):1–2
7. Dickmann J, Appenrodt N, Bloecher H-L, Brenk C, Hackbarth T, Hahn M, Klappstein J, Muntzinger M, Sailer A (2014) Radar contribution to highly automated driving. In: 11th European radar conference (EuRAD). IEEE, pp 412–415
8. Edelsbrunner H, Kirkpatrick D, Seidel R (1983) On the shape of a set of points in the plane. IEEE Trans Inf Theory 29(4):551–559
9. Eppstein D (1998) Finding the k shortest paths. SIAM J Comput 28(2):652–673
10. Hart PE, Nilsson NJ, Raphael B (1968) A formal basis for the heuristic determination of minimum cost paths. IEEE Trans Syst Sci Cybern 4(2):100–107
11. McCarthy Z, Bretl T, Hutchinson S (2012) Proving path non-existence using sampling and alpha shapes. In: IEEE international conference on robotics and automation (ICRA). IEEE, pp 2563–2569
12. Mehlhorn K, Näher S (1999) LEDA: a platform for combinatorial and geometric computing. Cambridge University Press, Cambridge
13. Poppinga J, Birk A, Pathak K (2008) Hough based terrain classification for realtime detection of drivable ground. J Field Robot 25(1–2):67–88
14. Rufli M, Siegwart R (2010) On the design of deformable input-/state-lattice graphs. In: IEEE international conference on robotics and automation (ICRA). IEEE, pp 3071–3077
15. Russell S, Norvig P (2009) Artificial intelligence: a modern approach, 3rd edn. Prentice Hall Press, Upper Saddle River
16. Sakkalis T, Charitos Ch (1999) Approximating curves via alpha shapes. Graphical models and image processing 61(3):165–176

17. Schadler M, Stückler J, Behnke S (2014) Rough terrain 3d mapping and navigation using a continuously rotating 2d laser scanner. KI-Künstliche Intelligenz 28(2):93–99
18. Worrall S, Orchansky D, Masson F, Nieto J, Nebot E (2010) Determining high safety risk scenarios by applying context information
19. Yen JY (1970) An algorithm for finding shortest routes from all source nodes to a given destination in general networks
20. Ziegler J, Bender P, Dang T, Stiller C (2014) Trajectory planning for berthaa local, continuous method. In: Intelligent vehicles symposium proceedings. IEEE, pp 450–457
21. Ziegler J, Werling M, Schroder J (2008) Navigating car-like robots in unstructured environments using an obstacle sensitive cost function. In: Intelligent vehicles symposium. IEEE, pp 787–791

Dynamic eHorizon with Traffic Light Information for Efficient Urban Traffic

Hongjun Pu

Abstract The electronic horizon (eHorizon) is an emerging technology supporting advanced driver assistance systems (ADAS) with respect to fuel efficiency and road safety. Using the static road attributes provided by the eHorizon, new kinds of ADAS applications become possible. Considering the factor that the traffic dynamics, especially in intersection areas, has also a significant impact to the fuel consumption and traffic efficiency, this paper is devoted to the extension of the current products of eHorizon with traffic light information. It introduces a concept for the presentation of the traffic light information in accordance to the ADASIS v2 specification, so that today's subscribers of eHorizon, i.e. the ECUs with ADAS applications, can use the dynamic data with only minimal modification.

Keywords ADAS · eHorizon · Traffic light · Energy efficiency · Traffic efficiency · Road safety

1 Introduction

For diverse reasons, road traffic will remain one of the greatest challenges in the coming years, especially in and around cities. This is because of the globalization of the world economy, the urbanization of developing countries and the re-urbanization of industrial countries due to aging population. Also the individualization of the consume markets, e.g. e-commerce, causes additional traffic related to ware-delivery. All these factors lead to increased road traffic and requirements on intelligent solutions for traffic and energy efficiency.

H. Pu (✉)
Continental Automotive GmbH, Philipsstrasse 1, 35576 Wetzlar, Germany
e-mail: hongjun.pu@continental-corporation.com

T. Schulze et al. (eds.), *Advanced Microsystems for Automotive Applications 2015*,
Lecture Notes in Mobility, DOI 10.1007/978-3-319-20855-8_2

The electronic horizon (eHorizon) is an emerging technology that supports advanced driver assistance systems (ADAS) with respect to fuel efficiency and road safety. Current products of eHorizon extract information from a geo-database and provide certain road attributes like intersection, slope, curvature, speed limit and lane information over a well specified CAN-interface.

In this way, nearly all ECUs in the vehicle can get aware of the roads ahead, as if they have an electronic "eye" observing the horizon. Using the data of eHorizon, new kinds of driver assistance functions for energy-efficiency and road safety become possible. For example, a predictive vehicle cruise control of the truck manufacturer Scania using road geometry information can save up-to 3 % fuel consumption compared to a convenient cruise control [1]. In similar ways, the hybrid and electrical vehicles are expected to achieve more energy-efficiency by deploying a driving and charging strategy adaptive to road properties.

Although predictive road information contributes a lot to the optimization of the operations of gear box, fuel supply and braking system, no longitudinal control for urban traffic would be green enough, as long as the traffic dynamics in intersections are not involved. The vehicle motion at intersections, governed by the traffic light, has a significant impact to the total fuel consumption (or energy balance for electrical and hybrid vehicles) of a city route.

Considering the above intersection problem, this paper presents a solution to provide traffic light information in intersection areas by extension of the eHorizon. After a short over view of eHorizon we will discuss the technologies enabling the transmission of traffic light information into the vehicles. Then, as main contribution of the paper, a concept for presenting the traffic light status and phases on the eHorizon will be introduced and discussed. We will illustrate how the dynamic traffic light information can be embedded in the standard structure of eHorizon, so that the subscribers of eHorizon data, i.e. the electronic control units (ECU) with ADAS applications, need only minimal modification in order to use this dynamic information.

2 The eHorizon

The development of eHorizon was initialized by the ADASIS Forum established in 2001 by a group of car manufacturers, in-vehicle system developers and map data companies. Goal of the ADASIS Forum was to develop a standardized interface between digital map and ADAS applications. The first system and interface specification of ADASIS Forum, e.g. ADASIS v1, was worked out within the EU-funded project PReVENT/MAPS&ADAS. The current version, e.g. ADASIS v2 [2], are widely accepted by the automotive industry as de facto standard. Ress et al. published in 2008 a well structured description of ADASIS v2 [3].

2.1 Stubs and Paths

Paths and stubs are the basic elements of eHorizon describing the road segments and the branch-points. As shown in Figs. 1 and 2, a path with a path-ID represents a road link and stubs are defined at the intersections and branching-points on the path. Sub-paths with its own path-ID may start from each stub.

As the name implicates, only road links on the horizon, i.e. ahead of the ego-vehicle and are reachable in reasonable time, are presented as paths of the eHorizon. Further, the paths are one-dimensional, i.e. all road attributes are positioned by their relative distances to the origin of the path.

2.2 ADAS Attributes

While the paths and stubs present the basic geometry of the road, the useful information for the ADAS applications are provided by the so called ADAS attributes. These are the road classes, scopes, curvatures, speed limits, etc. defined on certain positions and segments of a path.

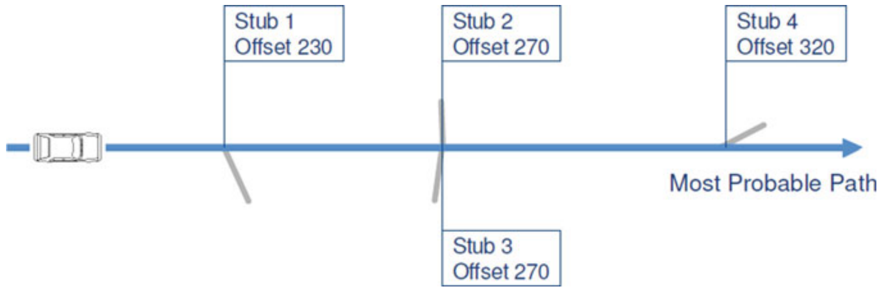

Fig. 1 Stubs of the eHorizon [3]

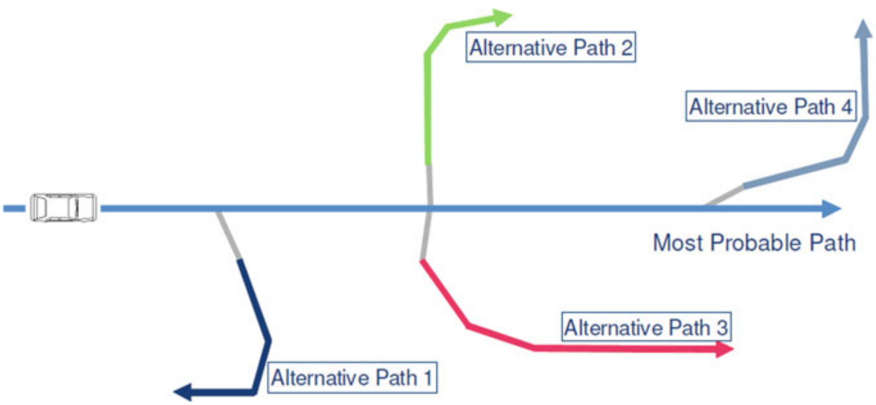

Fig. 2 Sub-paths starting from stubs of the eHorizon [3]