2014 IEEE Intelligent Vehicles Symposium (IV)
June 8-11, 2014. Dearborn, Michigan, USA

A Traffic Knowledge Aided Vehicle Motion Planning Engine Based on Space Exploration Guided Heuristic Search

Chao Chen[1] and Markus Rickert[1] and Alois Knoll[2]

***Abstract*— A real-time vehicle motion planning engine is presented in this paper, with the focus on exploiting the prior and online traffic knowledge, e.g., predefined roadmap, prior environment information, behaviour-based motion primitives, within the space exploration guided heuristic search (SEHS) framework. The SEHS algorithm plans a kinodynamic vehicle motion in two steps: a geometric investigation of the free space, followed by a grid-free heuristic search employing primitive motions. These two procedures are generic and possible to take advantage of traffic knowledge. In this paper, the space exploration is supported by a roadmap and the heuristic search benefits from the behaviour-based primitives.**

Based on this idea, a light weighted motion planning engine is built, with the purpose to handle the traffic knowledge and the planning time in real-time motion planning. The experiments demonstrate that this SEHS motion planning engine is flexible and scalable for practical traffic scenarios with better results than the baseline SEHS motion planner regarding the provided traffic knowledge.

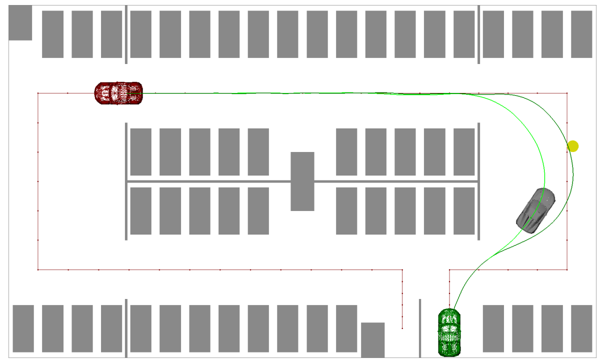


Fig. 1. Motion planning for navigating through a car park. Start (*green frame*), goal (*red frame*), prior knowledge of static obstacles (*grey*) and a roadmap (*red*) are provided. The initial result from SEHS engine (*dark green line*) is adapted (*light green line*) after detecting an obstacle (*yellow*).

I. INTRODUCTION

In context of autonomous driving, vehicle motion planning is rather different from common robot motion planning problems. The generic robotic motion planning concentrates on exploiting the high dimensional configuration space of a robot with many degrees of freedom [1], while vehicle motion planning deals with nonholonomic constraints in a relatively low dimensional configuration space [2], which is a combination of a two-dimensional position (x, y) and orientation θ. Furthermore, most industrial robot tasks are repetitive in a relatively static environment with few human interactions. Therefore, the planning can be offline, and have plenty time to search for a cost-efficient solution. In contrast, an autonomous vehicle carries out its motion in daily traffic involving human drivers and passengers. The environment is dynamic and the interactions happen in runtime, which require the planner not only to find a solution to reach the goal, but also to plan a suitable motion according to the traffic rules and human convenience. And the whole procedure must be online.

In order to build such an online traffic-knowledge-aware autonomous motion agent, the motion planner needs to handle information such as roadmaps, traffic signals, human preferred driving manners, etc., which are generally called *Traffic Knowledge* in this paper. Other than the "hard" constraints from the physical constraints such as collision and kinodynamic limitations, some of those requirements are rather "soft" as optimization criteria for post-processing. However, such information may help the planning in the first place as additional constraints or heuristics for a faster convergence or a better result. Further discussion and introduction of related works are conducted in Section II.

Space exploration guided heuristic search (SEHS), as introduced in [3], has two phases. The space exploration part takes circles to flood through the free space and constructs a path corridor, called *Circle-Path*, from the start position to the goal position. Then, the heuristic search procedure employs this circle-path as heuristics to propagate states using primitive motions and searches for a sequence of motions to reach the goal. With the help of circle-path as direction and step-size guidances, the heuristic search achieves excellent performance in a grid-free manner. In this paper, the SEHS algorithm is taken as the core of the online motion planning engine, which in addition considers the traffic knowledge and updates the vehicle motion in real-time. Details are presented in Section IV and III.

The goal of the SEHS motion planning engine is to solve the motion planning problem in more realistic traffic scenarios as demonstrated in Figure 1. The vehicle should navigate through a car park to a certain location, where it can automatically park into a free parking slot. As the environment is semi-structured, a roadmap is predefined. And the vehicle is required to follow the roadmap while driving inside the park. The sensing range of the vehicle is limited. Therefore, the planning engine needs to verify the planned motion against live detected objects and update the motion in time when necessary. More examples are demonstrated in Section V.

[1]Chao Chen and Markus Rickert are with fortiss GmbH, affiliated institute of Technische Universität München, Munich, Germany

[2]Alois Knoll is with Robotics and Embedded Systems, Technische Universität München, Munich, Germany

978-1-4799-3637-3/14/$31.00 ©2014 IEEE

II. Related Work

In general context of robotic motion planning, the motion planning of nonholonomic mobile robot is well studied. The geodesic of the configuration space, i.e., the shortest path without considering the collisions, is provided in [4] for forwards driving, [5] for bidirectional driving and [6] if the driving curvature is continuous. These results can be used as the configuration space metric in random sampling methods such as [7] [8] and their advanced variations. They can also be regarded as heuristics for the grid-based search methods like [9]. Besides, the kinodynamic or dynamic model can be directly exploited by RRT-based methods [10] [11] or control-based approaches [12] [13]. However, these algorithms concern mainly about the kinematic or kinodynamic constraints of the robot and the collisions with obstacles. In practical autonomous driving applications, other aspects have to be evaluated, such as traffic rules, energy consumption, human convenience, etc. And the planner must produce the result online, that a motion either reaches the goal or brings the vehicle to a safe state if the goal is unreachable.

A solution is decomposing a traffic scenario into a number of well specified driving tasks, e.g., following, overtaking, turning, parking. And each driving task can be further broken down into sub-tasks until some primitive motions or behaviours are reached. These atomic manoeuvres can be performed with behaviour-based methods and combined to achieve a complex mission [14]. The development of advanced driver assistant systems started just from these small building blocks, such as adaptive cruise control, lane change assistant, collision avoidance system, parking assistant, etc. However, the integration of these sub-functions in a unified framework usually leads to a hierarchical architecture, which increases the complexity of the overall system and requires sophisticated configurations of the components according to different traffic situations. In [15], different configurations of the sub-modules are designed for intersection handling and lane driving. Another approach is to do the planning first in the common way and then optimize the result according to the additional requirements. In [16], a roadmap is built regarding the environment topology and safety distance. The path is smoothed with a cost function including the deviation cost from the roadmap. However, the result could be a local optimum in contrast to one from a method which considers the roadmap in the planning phase.

In real-time applications, the planner operates with perception and control modules. In the *"Sensing, Planning, Acting"* paradigm, the three components run in a sequential manner with the assumption that each one is fast enough to produce inputs for the subsequent module. To be "fast enough" is a big challenge in the implementation for online motion planning. However, the timing condition for planning can be relaxed if the system guarantees that the vehicle is always in a safe state during the planning and even when the planning result is negative. These aspects are not always explicitly discussed, but rather regarded as engineering details [17] [18] [19].

III. Traffic Knowledge

The traffic knowledge is a general concept proposed in this paper, which stands for all kinds of information related to the vehicle motion, e.g., road networks, traffic signs, common driving behaviours. Two kinds of traffic knowledge are taken as examples to show the flexibility and extensibility of the SEHS framework, which is able to take benefit directly in the planning phase. The space exploration evaluates the geometric properties of the free space. The roadmap already contains certain topological knowledge, which can accelerate the initial exploration. The heuristic search takes primitive motions for the states propagation. A good choice of the atomic motions will dramatically improve the search performance and produce more human desirable results.

A. Roadmap

The space exploration can be initialized with a roadmap. In practice, not all the physically free space is allocated for the vehicle motion. A vehicle is supposed to stay in its lane. And a lane shift is performed only when necessary. However, this behaviour is not easy to accomplish by the motion planning alone. Usually, an extra intelligent entity needs to evaluate the traffic situation and decide whether it is suitable to swing out or go back into the lane. In SEHS motion planning, if the initial exploration is constructed according to a roadmap, it is more likely to obtain a roadmap oriented motion. Meanwhile, the planner still has the possibility to explore and search in the free space to avoid collisions.

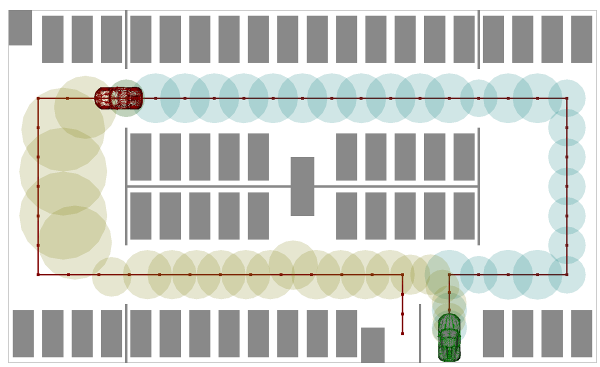


Fig. 2. Roadmap based space exploration in a car park scenario: A roadmap consists of way-points (*red dots*) and connections (*red lines*). The circle-path based on the roadmap (*blue circles*) compares with the circle-path without considering the roadmap (*yellow circles*).

In this paper, the roadmap is defined with a graph of way-points. The graph vertices are way-points in 2D coordinates. The graph edges are line segments connecting them. Without loss of generality, the start and goal positions are assumed to be inside the space exploration circles of the way-points. The according way-points are treated as the start and goal in a shortest path problem to get a roadmap path. Then, circles are created at these way-points to construct a circle-path, as the blue circles in Figure 2. If an edge is too long for the neighbouring circles to overlap, it is interpolated for intermediate circles. Figure 2 also shows another advantage of roadmap aided space exploration. The yellow circles are the result from space exploration without considering the roadmap, which is suboptimal for the motion planning task. This indicates the roadmap knowledge can also contributes to the correctness of the exploration result.

After the roadmap based space exploration, the SEHS planner carries on to verify the initial circle-path against the live sensing inputs and updates the circle-path incrementally if necessary. Examples of the roadmap aided SEHS motion planning are demonstrated in Section V.

B. Driving Behaviour

A primitive motion is a combination of control inputs that defines an atomic transaction between vehicle states. In the SEHS framework, the length of a primitive is calculated regarding the circle size. Other parameters such as acceleration and steering just take a simple metric, which is invariant throughout the planning. It is possible to adapt this trivial lattice to different driving behaviours to improve the search efficiency.

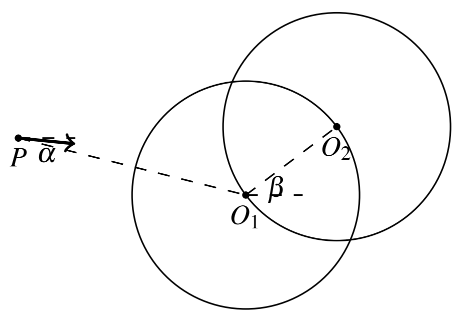


Fig. 3. Steering behaviour based on SEHS circle positions. α is the angle for the next circle O_1 at the point P. β is the angle for the after next circle O_2 at the point O_1. The arrow shows the combined driving direction at P.

Figure 3 shows an example of determining steering behaviours based on the relative circle positions. In this case the next two circles are considered. If the current state is at the point P, the optimal driving direction to reach the next circle O_1 has an angle α, which varies abruptly when the next circle changes to O_2. Therefore, the angle β between O_1 and O_2 is taken as a revision. Assuming the length of PO_1 is d and the radius of O_1 is r. The desired driving direction θ at P can be calculated with the Equation 1. The value of θ is linear between α and β according to the distance d. k is a constant factor. Thus, the vehicle travels towards the next circle when it is far away and gradually steers to the after next circle when it is getting closer.

$$\theta = \beta + (\alpha - \beta)\frac{\min(d, kr)}{kr}, k \geq 1 . \tag{1}$$

Thus, every state in SEHS heuristic search obtains a desired driving direction from the circle-path. By comparing this direction with its own steering and driving direction, a driving behaviour is chosen from left, right steering and straight driving, which have none uniform primitive motion lattices as Figure 4. In this paper, the behaviour-based steering lattices are created with the help of normal distributions. The SEHS planner obtains more states in the desired direction with a finer resolution. Of course, the motion lattice can also be modified in the dimension of acceleration by comparing the current vehicle velocity with the desired speed.

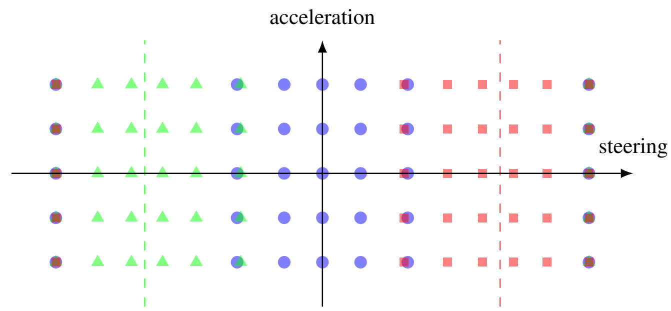


Fig. 4. Behaviour-based steering motion primitives: X-axis is for steering. Y-axis is for acceleration. The blue dots are straight driving; the green triangles are left steering; the red rectangles are right steering. The dashed lines are the according mean values.

More natural primitive motions can be created with a close study of the human driving behaviours. The SEHS planner provides information in the space exploration phase to choose the right primitive in the heuristic search, which improves the heuristic search efficiency and produces more human like motion result.

IV. SEHS Motion Planning Engine

The SEHS motion planning engine encapsulates the SEHS framework and provides interfaces to the traffic knowledge. The space exploration exploits the roadmap knowledge with the prior environment model. The heuristic search adapts the motion primitives to the vehicle states and the circle-path with steering behaviours. These two procedures are managed by the engine to operate in different modes, such as planning, replanning and verification.

A. Engine Components

The components of the SEHS motion planning engine is illustrated in Figure 5. They are divided into three types: the planning components, which perform the SEHS algorithm; the knowledge components, which provide information about the traffic situation, the vehicle model and driving behaviours; the management component, which controls the operation of the engine.

The planning components are the core of the planning engine. The two steps of the SEHS algorithm can be performed separately for specific tasks. For example, the *Space Exploration* can be used to detect environmental changes which may trigger a replanning. The *Heuristic Search* can be employed to verify a planned motion.

The knowledge components are the interfaces to the prior knowledge and the real-time sensing. The *Road Map* component contains the information of the road network, which is a database obtained from navigation system or Car-to-Infrastructure communications. The *Behaviour* component is a list of driving behaviours with the according conditions. The *Vehicle Model* contains the kinematics or dynamics

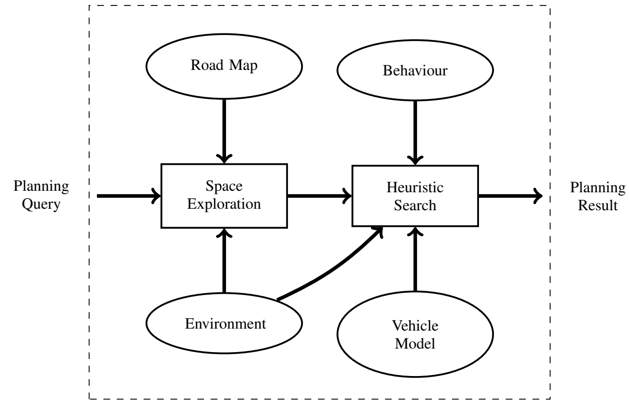


Fig. 5. SEHS motion planning engine components: The knowledge components (ellipses) and the planning components (rectangles) are encapsulated in the engine (dashed box), which has management functionalities.

model of the vehicle with the limits of the control parameters. The *Environment* component consists of prior and post models of the environment and offers the collision check and distance query services. The prior environment knowledge is used to initialize the space exploration based on the roadmap, while the post detections are referred during the incremental updating of the circle-path. As the prior obstacles are static, the planned motion verification only need to consider the post detected objects, which saves a lot effort in collision check.

The management component takes care of two things: the timing and the operation mode in order to achieve an anytime safe motion guarantee with minimum working load. The idea is that an update is only necessary when the motion is invalid, e.g., a collision is detected, or the motion is no longer optimal. The first situation is easy to check by verifying the planned motion against the real-time environment. The second one is almost as difficult as replanning. However, with the help of space exploration, it is easy to answer this question in certain circumstances. If an optimal motion is found within a circle-path, this motion is still optimal if the new circle-path is enclosed in the older one. In addition, according to the roadmap, if an previous invalid way-point is valid in the new circle-path, there could be a better solution according to the roadmap criterion. Thus, the engine is able to avoid redundant replanning by doing motion and circle-path verification instead. When a replanning is required, the SEHS planner can proceed incrementally with the knowledge gathered in the previous iterations, which is discussed in the next subsection.

B. Engine Operation

As mentioned above, the engine may have different operation modes: planning, verification and replanning. The planning procedure is the basic routine of SEHS motion planning as presented in [3]. When a planning result is available, the vehicle begins to execute the planned motion. The motion planning engine starts a verification thread to make sure the motion is always safe until the goal is reached. If not, a replanning thread is triggered. The verification thread runs synchronously with the environment perception, while the replanning is asynchronous. The states flow diagram is illustrated in Figure 6.

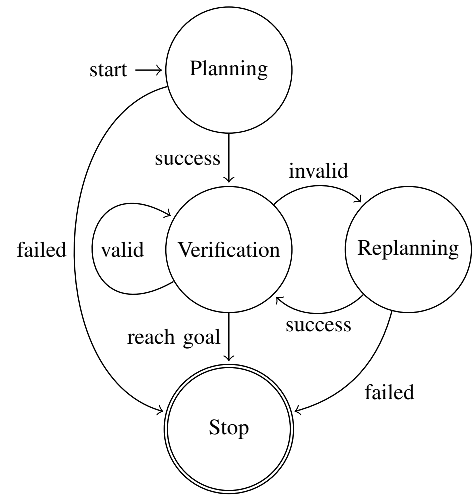


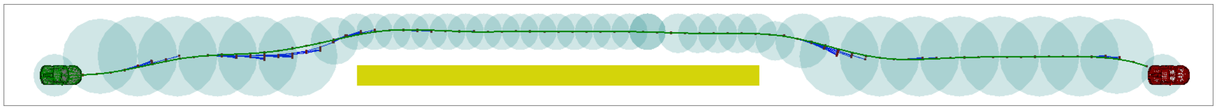
Fig. 6. SEHS motion planning engine states diagram.

In the verification procedure, the circle-path is evaluated by updating the radius of the circles, which could result in three situations as follows:

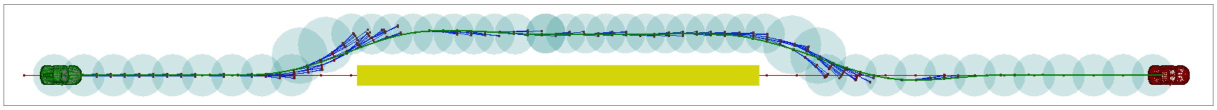
1) A gap is found in the circle-path, the replanning thread starts with incremental space exploration to fill the gap and proceeds with the incremental heuristic search.
2) The new radii are no larger than the previous ones, i.e., the new circle-path is enclosed by the old one. An incremental heuristic search starts only when a collision exists in the planned motion.
3) If a change of the circle-path involves a way-point, a replanning is started all over from the roadmap based space exploration.

The incremental heuristic search is realized by reusing the states propagation result from the last planning procedure. The state tree is updated first by removing the branches from the states in the past and the states outside the current circle-path. The costs of the rest states are updated according to the current space heuristics. The collisions are verified in a lazy manner, which only happens when the node is investigated for expansion. Thus, a large amount of time could be saved during the replanning.

An important issue of the motion planning engine is the time limit for the replanning. Instead of a fixed time duration for each planning operation, the SEHS engine intents to give the planner as much time as possible. This is achieved with the idea of the safe motion. Through verification, the engine knows when a collision could occur. The planned motion is truncated before that point and appended with a safe motion, which is usually a pull over and brake manoeuvre. Thus, any internal state from current to the beginning of the safe manoeuvre can be chosen as the start state for the replanning. And the whole time duration to the start state is available for the replanning. A trade-off should be made here between the time and space for replanning. Because a longer planning



(a) SEHS motion planning without traffic knowledge



(b) Roadmap and behaviour-based primitives aided SEHS motion planning

Fig. 7. Overtaking Scenario: The green and red frames are the start and goal respectively. The moving obstacles are modelled as a long yellow rectangle when considering its velocity and a safety margin. The roadmap is in red. The circle-path and state propagation are in blue. The result motion is in green.

time means a closer start state to the obstacle, which results in less space for replanning, and vice versa.

V. Motion Planning Scenarios

Two scenarios are evaluated with the SEHS motion planning engine. The first example is a revisit of the overtaking scenario in [3], which demonstrates the traffic knowledge aided SEHS motion planning. The second case is a navigation scenario through a car park as introduced in Section I. In this scenario, the sensing range of the vehicle is limited to show the real-time collision avoidance and replanning capability of the motion planning engine. The implementation is in C/C++ and executed on a PC with an Intel® Core™ i7-870 2.93 GHz processor and 8 GB RAM.

A. Overtaking

As illustrated in Figure 7, the road has two lanes with the same driving direction. A slow vehicle is driving in front of a fast vehicle on the right lane. The fast vehicle should overtake the slow vehicle to reach the goal position. The slow vehicle is modelled as a long yellow rectangle obstacle regarding the speed and a certain safety margin. The speed of the fast vehicle is $20\,\mathrm{m\,s^{-1}}$ and the steering speed is $0.2\,\mathrm{m^{-1}\,s^{-1}}$, which is the change rate of the trajectory curvature.

A general human driving behaviour in this case is first performing a lane shift to the left, bypassing the slow vehicle, and then switching back to the right lane. The baseline SEHS planner explores the whole free space and uses uniform distributed motion primitives in the search. Figure 7(a) shows that it drives the vehicle in the middle of the road, where is optimal regarding the distance to obstacles. In contrast the roadmap aided SEHS engine planner with behaviour-based primitives stays mostly inside a lane, and make lanes switches only at the begin and end of the overtaking manoeuvre in Figure 7(b), which compliances better to human behaviour and traffic rules.

Table I compares the performance of four variations of SEHS planners: the baseline version, the behaviour aided version, the roadmap aided version and the both traffic knowledge version. The time costs for space exploration and heuristic search are listed. The memory requirement is evaluated by the number of nodes in the heuristic search. The values are the averages of 20 simulations with random sampled start and goal positions in the defined regions.

TABLE I

Evaluation of traffic knowledge aided SEHS motion planning

Method	SE Time (ms)	HS Time (ms)	Nodes
Baseline SEHS	3	22	620
Behaviour SEHS	3	16	495
Roadmap SEHS	1	76	2204
Behaviour + Roadmap	1	31	1151

Comparing the baseline SEHS and the driving behaviour aided SEHS, the behaviour-based motion primitives reduced the number of search nodes, which results in a shorter search time. The roadmap aided SEHS takes less time in space exploration, because the initial circle-path is obtained from the roadmap. However, it takes more effort in heuristic search to generate a roadmap oriented motion. Combing the roadmap and driving behaviour knowledge, the number of heuristic search nodes is reduced to a half for this scenario.

B. Car Park

In the car park scenario, a semi-structured park environment is provided as the prior knowledge, which contains the static obstacles (grey) and a roadmap (red). An additional object (yellow), which is not prior known, blocks the roadmap. The vehicle drives with a maximum speed of $3\,\mathrm{m\,s^{-1}}$ and has a sensing range of 12 m, i.e., it can only detect the object within a maximum distance of 12 m. Therefore, the motion planning engine should be able to replan the motion in time to avoid the collision and still reach the goal.

The result is illustrated in Figure 8. The initial planning takes 1 ms in space exploration, which is mainly based on the roadmap. Then, a motion is planned close to the roadmap, which is the dark green line on the right. The whole initial planning procedure takes under 200 ms.

When the vehicle carries out the planned motion, the verification thread is running. Each verification takes just 1 ms, as it only verifies the circle-path and checks collision along the trajectory regarding the post detected objects. As soon as the obstacle is detected, the trajectory becomes invalid and a collision is predicted in 5.44 s according to the vehicle motion. The trajectory is truncated and an intermediate point is chosen as the start state for replanning. In this case, the replanning has a time margin of 200 ms. The rest trajectory

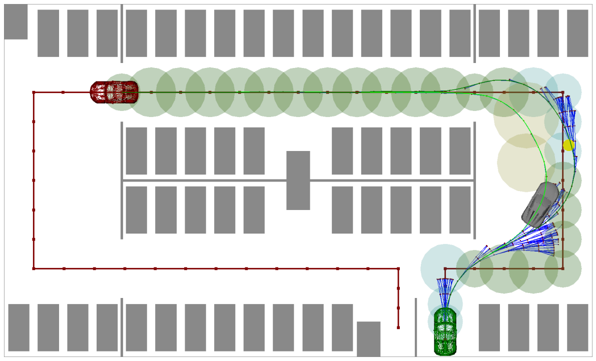


Fig. 8. Motion planning for a vehicle navigation through a car park with prior known static obstacles (*grey*) and a post detectable objects (*yellow*). A roadmap (*red*) is provided to aid the space exploration (*light blue circles*). The sensing range of the vehicle is limited. The SEHS engine plans a path (*dark green line*) from start (*green frame*) to goal (*red frame*), and updates the motion (*light yellow circles* and *light green line*) when the obstacle is detected.

is modified for a decelerate to stop manoeuvre before the collision. While the vehicle keeps driving along the truncated trajectory, the planner carries out the incremental SEHS motion planning. The incremental space exploration takes 2 ms. The whole replanning duration stays in 100 ms range. A new path is found as the light green line left around the obstacle.

In this case, we see that the SEHS motion planning engine can online adapt the vehicle motion to environment changes. No periodic replanning takes place here, but with the help of a fast verification thread, the replanning happens only when necessary. Furthermore, a backup plan is always ready to bring the vehicle to a safe state even when the replanning fails.

VI. Conclusion and Future Work

The basic SEHS motion planning framework is extended to exploit traffic knowledge in the planning phase. The roadmap orients the vehicle motion to the road network. The driving behaviour based motion primitives improve the efficiency of the vehicle states propagation. Thus, the traffic knowledge aided SEHS motion planner can produce better vehicle motions regarding the traffic scenarios. And a motion planning engine is developed to integrate all the components and take care of the real-time execution. It provides interfaces between traffic information sources and the motion planner: the navigation system updates the roadmap; the sensing module models the present environment; flexible motion primitives are chosen based on the vehicle state and human driving behaviours.

This paper focuses on the general structure and procedure aspects of traffic knowledge aided motion planning. There are still many detail work to do. For example, the motion of the obstacles could be considered to model tracked moving objects. The roadmap can contain more information such as speed limit, traffic priority, lane driving direction, etc. More advanced motion primitives can be designed according to the study of the behaviour of human drivers. Furthermore, there are still other traffic knowledge available such as traffic light signals, information exchange through Car-to-Car communication and so on. The two generic stages of SEHS motion planning provide the possibility to integrate these knowledge into different phases of the intelligence of autonomous driving.

Acknowledgements

This work is partially funded by the German Federal Ministry of Economics and Technology under grant no. 01ME12009 through the project RACE[1].

References

[1] S. LaValle, *Planning Algorithms*. Cambridge University Press, 2006.

[2] J.-P. Laumond, *Robot Motion Planning and Control*. Springer, 1998.

[3] C. Chen, M. Rickert, and A. Knoll, "Combining space exploration and heuristic search in online motion planning for nonholonomic vehicles," in *Proc. IEEE Intelligent Vehicles Symposium*, 6 2013, pp. 1307–1312.

[4] L. Dubins, "On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents," *American Journal of Mathematics*, vol. 79, no. 3, pp. 497–516, 1957.

[5] J. Reeds and L. Shepp, "Optimal paths for a car that goes both forwards and backwards," *Pacific Journal of Mathematics*, vol. 145, no. 2, pp. 367–393, 1990.

[6] T. Fraichard and A. Scheuer, "From reeds and shepp's to continuous-curvature paths," *IEEE Transactions on Robotics*, vol. 20, no. 6, pp. 1025–1035, 12 2004.

[7] L. Kavraki, P. Švestka, J.-C. Latombe, and M. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.

[8] S. LaValle, "Rapidly-exploring random trees: A new tool for path planning," Department of Computer Science, Iowa State University, Tech. Rep., 1998.

[9] S. Koenig, M. Likhachev, and D. Furcy, "Lifelong planning a*," *Artificial Intelligence Journal*, vol. 155, no. 1-2, pp. 93–146, 2004.

[10] S. LaValle and J. Kuffner Jr., "Randomized kinodynamic planning," *The International Journal of Robotics Research*, vol. 20, no. 5, pp. 378–400, 2001.

[11] A. Shkolnik, M. Walter, and R. Tedrake, "Reachability-guided sampling for planning under differential constraints," in *Proc. IEEE International Conference on Robotics and Automation*, 5 2009, pp. 2859–2865.

[12] J. Barraquand and J.-C. Latombe, "Robot motion planning: A distributed representation approach," *The International Journal of Robotics Research*, vol. 10, no. 6, pp. 628–648, 1991.

[13] S. Quinlan and O. Khatib, "Elastic bands: Connecting path planning and control," in *Proc. IEEE International Conference on Robotics and Automation*, 5 1993, pp. 802–807.

[14] R. Brooks, "A robust layered control system for a mobile robot," Massachusetts Institute of Technology, Tech. Rep. A.I Memo 864, 9 1985.

[15] C. Baker and J. Dolan, "Traffic interaction in the urban challenge: Putting boss on its best behavior," in *Proc. IEEE/RSJ International Conference on Intelligent Robotos and Systems*, 9 2008, pp. 1752–1758.

[16] D. Dolgov and S. Thrun, "Autonomous driving in semi-structured environments: Mapping and planning," in *Proc. IEEE International Conference on Robotics and Automation*, 5 2009, pp. 3407–3414.

[17] C. Urmson, J. Anhalt, D. Bagnell, C. Baker, R. Bittner, *et al.*, "Autonomous driving in urban environments: Boss and the urban challenge," *Journal of Field Robotics*, vol. 25, no. 8, pp. 425–466, 2008.

[18] M. Montemerlo, J. Becker, S. Bhat, H. Dahlkamp, D. Dolgov, *et al.*, "Junior: The stanford entry in the urban challenge," *Journal of Field Robotics*, vol. 25, no. 9, pp. 569–597, 2008.

[19] J. Funke, P. Theodosis, R. Hindiyeh, G. Stanek, K. Kritatakirana, *et al.*, "Up to the limits: Autonomous audi tts," in *Proc. Intelligent Vehicles Symposium*, 6 2012, pp. 541–547.

[1] http://www.projekt-race.de/