2015 IEEE Intelligent Vehicles Symposium (IV)
June 28 - July 1, 2015. COEX, Seoul, Korea

Path Planning with Orientation-Aware Space Exploration Guided Heuristic Search for Autonomous Parking and Maneuvering

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

***Abstract*— Due to the nonholonomic constraints of the vehicle kinematics, parking and maneuvering a car in a narrow clustered space are among the most challenging driving tasks. This paper introduces an extended version of Space Exploration Guided Heuristic Search (SEHS) method, called Orientation-Aware Space Exploration Guided Heuristic Search (OSEHS), to solve the path planning problems for parking and maneuvering. This method considers the orientation of a vehicle in the space exploration phase to achieve knowledge about driving directions. Such information is exploited later in the heuristic search phase to improve the planning efficiency in maneuvering scenarios. This approach is not bound to the specific domain knowledge about a parking or maneuvering task, but obtains the space dimension and orientation information through a generic exploration procedure. Therefore, it is convenient to integrate the maneuvering ability into a general SEHS motion planning framework. Experiments show that the OSEHS approach produces better results than common random-sampling methods and general heuristic search methods.**

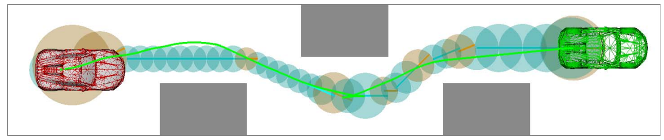


Fig. 1. Path planning for maneuvering with Orientation-Aware Space Exploration Guided Heuristic Search (OSEHS): A path planning problem is defined with a start pose (*red vehicle frame*), a goal pose (*green vehicle frame*), and prior knowledge of the static environment (*grey*). The space exploration returns a circle-path with directed circles (*circles with line segments indicating the orientations*), including the possible maneuvering region (*orange circles*) and the possible driving directions as forward (*cyan circles*) and reverse (*violet circles*). A maneuvering motion (*green line*) is planned by the heuristic search algorithm with the space and direction information.

I. INTRODUCTION

Parking a car into a small parking lot and maneuvering it through a narrow corridor are rather common driving scenarios in a clustered urban environment. Because of the nonholonomic constraints as a limited steering radius, the lateral motion of a vehicle is restricted and the orientation cannot be changed without a longitudinal movement. An advanced driver assistance system (ADAS) is extremely helpful to human drivers in these situations with respect to safety, efficiency, and convenience. Motion planning for parking and maneuvering tasks plays an important role in such ADAS and is an essential milestone to achieve fully automated driving.

The automobile industry has already brought some passive assistant functions such as parking distance control or 360° top-view to the market to help the driver with the limited field of view. Recently, more active features have been developed, which take over the steering or even the acceleration and braking during a parking maneuver. These functions are carefully designed for well-defined tasks and therefore lack the flexibility to solve a general maneuvering problem. On the one hand, it is hard to identify a specific maneuvering situation to apply a particular algorithm. On the other hand, a general motion planning method is inefficient for such specific problems. This paper introduces a heuristic search approach which extends the generic Space Exploration Guided Heuristic Search (SEHS) framework [1] to deal with general parking and maneuvering problems.

The SEHS method consists of two steps: The first one is called *Space Exploration*, which explores the workspace with circles for a path corridor that consists of overlapping circles, called *Circle-Path*. The circle-path provides knowledge about the topology and dimension of the free-space. In the second stage, a *Heuristic Search* algorithm performs a forward search along the circle-path while adapting the search steps to the circle size. Thus, the SEHS method convergences efficiently to a solution with a trade-off between path length and safety distance. However, as the vehicle orientation and kinematics are ignored in the exploration phase, the distance estimation from a circle-path may be sub-optimal when extra maneuvering effort is required. In this paper, the space exploration is further developed to consider the vehicle orientation in addition to the knowledge about driving directions. As a result, the Orientation-Aware Space Exploration Guided Heuristic Search (OSEHS) planner can identify the area where extra maneuvering is required with suggested driving directions as shown in Fig. 1. By modifying the cost function or selecting the primitive motions, the heuristic search algorithm improves the performance of path planning for parking and general maneuvering tasks.

II. RELATED WORK

During parking or maneuvering, the vehicle dynamics are less relevant, but the nonholonomic constraint of the minimum steering radius of a vehicle. In such a scenario, the driving speed is relatively slow and even steering at standstill is possible. A system like this is locally controllable [2], therefore, a path corridor in the configuration space guarantees the existence of a valid motion inside this passage. In case of parking or maneuvering, such a configuration

[1]Chao Chen and Markus Rickert are with fortiss GmbH, An-Institut Technische Universität München, Munich, Germany
[2]Alois Knoll is with Robotics and Embedded Systems, Technische Universität München, Munich, Germany

space passage is usually narrow and hard to obtain in a clustered environment. A potential-based method may be trapped in local minima and sampling-based methods are inefficient to explore through the narrow passage [3]. The grid-based heuristic search method [4] faces the trade-off between completeness and efficiency, while a narrow passage requires a fine resolution to resolve the short maneuvering motions. Furthermore, a heuristic estimation based on the grid distance provides little information about the driving direction during the maneuvering, especially when a vehicle should move back and forth at the same spot to adjust its orientation. Several methods are proposed to deal with the narrow passage problem in workspace [5], [6], [7], [1] with space decomposition or exploration approaches. However, the robot orientation and moving direction are not examined in these methods.

The parking problem can be solved with a behavior-based method if the system has sufficient prior knowledge about a well-defined parking scenario. In [8], an iterative method is proposed to steer a vehicle with sinusoidal functions into a parking lot. Collision avoidance is implemented based on the measurement from ultrasonic distance sensors. [9] and [10] apply a two-step approach, which first plans a collision-free curve without considering the vehicle kinematic and then converts this curve to a feasible trajectory. In [11] and [12], a path for parking is planned with a pre-defined schema for a specific scenario. In order to provide a unified method for different parking situations, [13] suggests an RRT-based algorithm. However, the drawback of random sampling methods is the uncertainty in path quality and planning time. [14] transforms the path planning problem to a local static optimization problem and iteratively constructs the solution with a heuristic to change driving directions. Most of these methods are specially designed for parking scenarios and are difficult to integrate into a general motion planning framework.

In [15], it is demonstrated that the SEHS method can be customized for specific traffic scenarios by modifying the exploration circles and primitive motions. This flexibility is further exploited in this paper to obtain the orientation knowledge and the according heuristics for maneuvering. In Section III, an orientation-aware space exploration procedure is introduced. Section IV presents several modifications of the heuristic search to make use of the orientation knowledge. The whole OSEHS framework is verified in Section V with three examples and compared with the results from three reference methods.

III. Orientation-Aware Space Exploration

In SEHS, the space exploration algorithm treats a vehicle as a holonomic robot without orientation. It can directly move from the center point of one circle to another. In case of clustered environments, the circle-based space exploration will create a circle-path that arbitrarily changes the moving direction to achieve the shortest path corridor. Furthermore, the space exploration terminates when a circle reaches the goal position, regardless of a mismatch between

---

Algorithm 1: SpaceExploration($c_{\text{start}}, c_{\text{goal}}$)

$S_{\text{closed}} \leftarrow \emptyset$;
$S_{\text{open}} \leftarrow \{c_{\text{start}}\}$;
while $S_{\text{open}} \neq \emptyset$ **do**
  $c_i \leftarrow \text{PopTop}(S_{\text{open}})$;
  if $f[c_{\text{goal}}] < f[c_i]$ **then**
    return success;
  else if $!\text{Exist}(c_i, S_{\text{closed}})$ **then**
    $S_{\text{open}} \leftarrow \text{Expand}(c_i) \cup S_{\text{open}}$;
    if $\text{Overlap}(c_i, c_{\text{goal}})$ **then**
      if $f[c_i] < g[c_{\text{goal}}]$ **then**
        $g[c_{\text{goal}}] = f[c_i]$;
        $\text{parent}[c_{\text{goal}}] = c_i$;
    $S_{\text{closed}} \leftarrow \{c_i\} \cup S_{\text{closed}}$;
return failure;

---

the final traveling direction and the goal orientation. These facts increase the discrepancy between the estimated circle-path distance and the actual motion distance, which requires additional search effort to close the gap. The idea of orientation-aware space exploration is to consider the robot orientation as the orientation of the circles with respect to the exploration direction. In this case, the traveling direction can be distinguished between forward and reverse.

Algorithm 1 is the algorithm for space exploration in SEHS [1]. It follows the standard heuristic search procedure with a closed set S_{closed} and an open set S_{open}. The heuristic cost h is the Euclidean distance to the goal c_{goal}, while the actual cost g is the distance from the start c_{start} via the circle centers. In each iteration, the circle c_i with the least total cost $f = g + h$ from the open set is chosen. If it does not exist in the closed set, child circles are expanded on the selected parent. The child circles are added to the open set and the parent circle is moved to the closed set. If the function $\text{Overlap}(c_i, c_{\text{goal}})$ found a circle overlapping with the goal circle, the cost to reach the goal is updated. The algorithm repeats until the goal cost is smaller than the total cost of any open set circles. Then, a circle-path is generated through backtracking. Several subroutines should be modified to perform the space exploration in an orientation-aware manner.

The function $\text{Expand}(c_i)$ calculates the center point and radius of a child circle, as well as the orientation. As demonstrated in Fig. 2, the orientation is defined by a vector connecting the parent and child center points. Regarding the parent orientation, the angle with a smaller difference in $[-\pi/2, \pi/2]$ is selected for the orientation of the child circle. The driving direction is determined by the relative orientation and position between the parent and child circles. Furthermore, different margins can be applied in the longitudinal and lateral directions of a directed circle to decide the circle radius in narrow space. As a result, the circle-path provides more precise space information for the specific

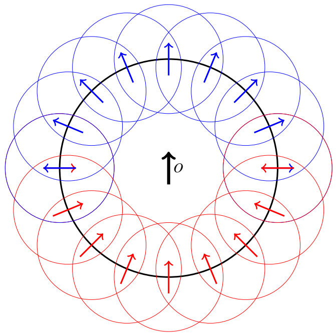


Fig. 2. Orientation-aware circle expansion: The arrows indicate the directions of the circles. The parent circle is in black and expands children at uniformly interpolated center points. The direction of a child is decided by the relative position of the center point. Based on the orientation, the driving direction of a circle is decided as forward (*blue*) and reverse (*red*).

driving direction.

The heuristic distance metric should also consider the orientations (1). $\|\vec{p}_i - \vec{p}_j\|$ is the Euclidean distance between the center points $\vec{p}_i$ and $\vec{p}_j$. $|\theta_i - \theta_j|$ is the absolute angle difference bounded in $[0, \pi/2]$. Assuming a vehicle has a maximum turning curvature of k, the traveling distance required to change its orientation about $|\theta_i - \theta_j|$ is greater or equal to $|\theta_i - \theta_j|/k$. Therefore, the maximum of the two types of distance returns an admissible heuristic estimation.

$$d_{i,j} = \max(\|\vec{p}_i - \vec{p}_j\|, |\theta_i - \theta_j|/k) \tag{1}$$

The function $\text{Exist}(c_i, S_{\text{closed}})$ checks the geometric relationship between the circles to skip redundant ones during the exploration. In SEHS, a circle is skipped when its center locates inside an existing circle, as it can be covered by the circle and its descendants. The OSEHS determine this relationship also regarding the orientation. According to the distance function (1), the equal distance points of a directed circle form the surface of a cylinder in a three dimensional space of (x, y, θ), as illustrated in Fig. 3. Thus, if another circle is centered inside a cylinder regarding a closed set circle and its radius, it is redundant in the exploration.

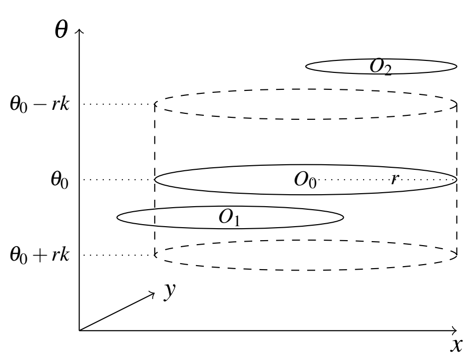
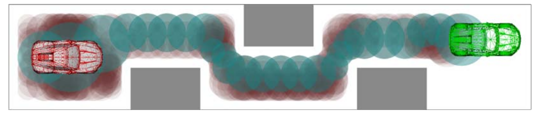
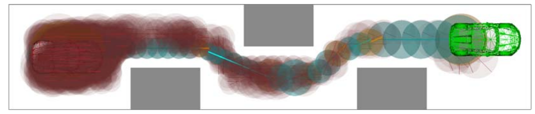


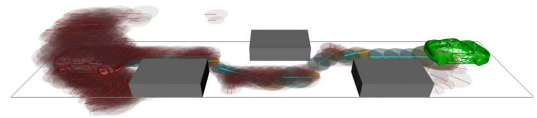
Fig. 3. Geometric relationship between the directed circles. The circle O_0 has a radius of r and an orientation of θ_0. All the states closer than a distance r to it lie in the cylinder centered at O_0 with a radius r and a height $2rk$. k is the maximum curvature of the vehicle. O_1 is redundant to O_0, as its state is inside the cylinder of O_0. O_2 is not redundant, despite its center point is inside the circle O_0 when projected to the x-y plane.



(a) Space exploration of SEHS method (bird view)



(b) Space exploration of OSEHS method (bird view)



(c) Space exploration of OSEHS method (side view)

Fig. 4. Space exploration of SEHS and OSEHS methods: The *brown* circles are the circles evaluated during the space exploration. The directed circles are illustrated in a three dimensional space with the orientation angle as the z-axis. The orientation is also indicated with a line from the circle center.

With the modifications above, the space exploration of OSEHS creates more circles than the general SEHS, as shown in Fig. 4. Comparing the top views of the exploration results in Fig. 4(a) and Fig. 4(b), the SEHS explores a larger portion of the workspace with less circles, while the orientation-aware version concentrates on a smaller neighborhood of the circle-path. This is more obvious in Fig. 4(c) with the side view of the directed circles in the three dimensional space. As the distance function penalizes direction changes of the circles, the result is a compromise between the Euclidean distance through the circle centers and the cost of orientation changes along the directed circles.

IV. Orientation-Aware Heuristic Search

The output from the orientation-aware space exploration is a sequence of directed circles. A preprocess is performed with the circle-path to decide the possible driving directions and the maneuvering regions. If the direction vector points to the next circle, forward driving is preferred, otherwise reverse driving is suggested. Due to the limited turning radius, a vehicle cannot always follow the direction of the circles. When a vehicle travels a distance d with a curvature k, the orientation change is $d \times k$. Therefore, if the orientation difference between two adjacent circles is larger than $k_{\max}$ proportional to the Euclidean distance between the centers, extra maneuvering in these circles is expected to follow the circle-path, which means both forward and reverse driving should be performed. After the preprocess, the circles are specified with three possible driving directions: *Forward*, *Reverse* and *Bidirectional*. Different primitive motions and cost functions are applied for the three situations.

Algorithm 2 is the general search procedure of SEHS, which is similar to the exploration algorithm and replaces the circles with vehicle states. The heuristic estimation is calculated according to the distance along the circle-path.

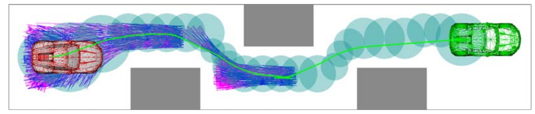
Algorithm 2: HeuristicSearch($\{c_i\}, \vec{q}_{\text{start}}, \vec{q}_{\text{goal}}$)

$S_{\text{closed}} \leftarrow \emptyset$;
$S_{\text{open}} \leftarrow \{\vec{q}_{start}\}$;
while $S_{\text{open}} \neq \emptyset$ **do**
  $\vec{q}_i \leftarrow \text{PopTop}(S_{\text{open}})$;
  if $f[\vec{q}_{\text{goal}}] < f[\vec{q}_i]$ **then**
    return success;
  else
    $c_i \leftarrow \text{MapNearest}(\vec{q}_i)$;
    if $!\text{Exist}(\vec{q}_i, c_i, S_{\text{closed}})$ **then**
      $S_{\text{open}} \leftarrow \text{Expand}(\vec{q}_i, c_i) \cup S_{\text{open}}$;
      if $h[\vec{q}_i] < R_{\text{goal}}$ **then**
        $\text{GoalExpand}(\vec{q}_i, \vec{q}_{\text{goal}})$;
      $S_{\text{closed}} \leftarrow \{\vec{q}_i\} \cup S_{\text{closed}}$;
return failure;

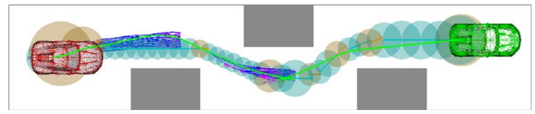
State expansion $\text{Expand}(\vec{q}_i, c_i)$ is done with primitive motions. Furthermore in each iteration, the chosen state $\vec{q}_i$ is mapped to the nearest circle c_i from the circle-path to enable motion step adaptation and efficiently resolve the similar states by $\text{Exist}(\vec{q}_i, c_i, S_{\text{closed}})$ with a circle-based clustering. A direct goal expansion $\text{GoalExpand}(\vec{q}_i, \vec{q}_{\text{goal}})$ is attempted when a state is inside a range R_{goal} from the goal state to accelerate the convergence. The orientation-aware heuristic search benefits from the direction knowledge mainly in the heuristic distance estimation and state expansion.

In the heuristic estimation, (1) is employed to define the distance metric. A state is mapped to the closest circle according to the distance metric. The next circle closer to the goal along the circle-path has a higher priority in case of a tie. As in the SEHS algorithm, the distance estimation consists of two parts. The first part is the distance to the next circle center. The second part is the remaining distance along the circle-path. As discussed in Section III, the distance function always returns a value that is less than or equal to the actual path distance of a vehicle. This estimation is admissible with respect to following the circle-path.

Another improvement of the orientation-aware heuristic search is that different groups of primitive motions are selected for the different types of circles. In [15], it is demonstrated that by choosing a suitable set of primitive motions according to the flow direction of the circle-path, a boost of the heuristic search is achieved for lane-following and overtaking maneuvers. In case of parking and maneuvering, the direction of forward or reverse driving is a hint for such decisions. According to the direction type of the closest circle, different primitive motions or cost factors are determined. For the circles with forward and reverse driving, the motions with matching direction are chosen to create the child states, or the motions in the wrong direction are punished with a larger cost. In case of circles for bidirectional driving, the vehicle can perform a cusp motion with less penalty and the forward and reverse motions are treated equally in the cost calculation. Thus, the algorithm favors the states in the suggested direction, so that maneuvering behaviors are more likely to be exploited in the pre-identified maneuvering regions.



(a) Heuristic search of SEHS method



(b) Heuristic search of OSEHS method

Fig. 5. Heuristic search of SEHS and OSEHS methods: The *blue* and *magenta* lines are the node expanded by the heuristic search for forward motion and reverse motion respectively.

Fig. 5 compares the heuristic search procedure in SEHS and OSEHS with selected primitive motions according to the driving direction. In general, the orientation-aware heuristic search takes fewer iterations to reach the goal. The nodes are denser around certain regions where maneuvering is required. In contrast, the SEHS method covers much larger areas with a lot more nodes in total as in Fig. 5(a). The strategy with different sets of primitive motions is applied in Fig. 5(b).

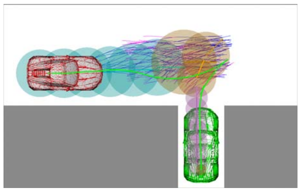
V. EXPERIMENTS

Three scenarios are verified in the experiments: two parking scenarios and a maneuvering problem. The results of OSEHS are compared to the outcomes from a sampling-based RRT method [16], the hybrid A* algorithm [17], and the general SEHS approach [1]. All the algorithms are implemented in C++ and executed on a machine with a 2.9 GHz CPU and 8 GB RAM.

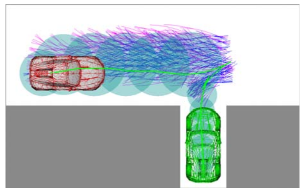
The vehicle is modelled with a $4\,\text{m} \times 2\,\text{m}$ rectangle. The maximum curvature of a turning motion is $0.2\,\text{m}^{-1}$. Three curvature values are selected for the left, right, and straight primitive motions with forward and reverse directions. In SEHS and OSEHS, the step size is decided by multiplying the circle radius with a factor and halved when no solution is found. The resolution of the states is also proportional to the circle radius. The hybrid A* algorithm takes the same set of primitive motions with a constant step size and grid resolution. The RRT method extends the tree with the same set of atomic control inputs and explores the bias of the goal states periodically. The performance of RRT is measured with the mean result of 20 trials.

A. Cross Parking Scenario

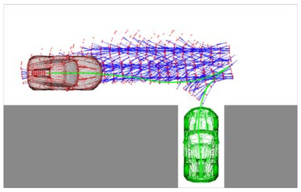
In the cross parking scenario, the vehicle's goal is to park into a perpendicular parking lot with 20 cm space margin to both left and right sides. The vehicle needs to perform a maneuvering to back into the parking space. The solutions are illustrated in Fig. 6. In Fig. 6(a), the parking maneuver is divided into three parts according to the types of the circles: forward driving, maneuvering, and reverse driving, as



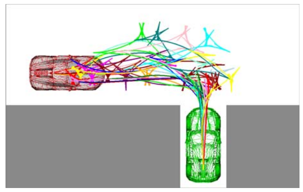
(a) Result of OSEHS method



(b) Result of SEHS method



(c) Result of Hybrid A* method



(d) Result of RRT method

Fig. 6. Cross parking scenario. The progress of the heuristic search methods are illustrated with *blue* and *magenta* lines for the nodes with forward driving and reverse driving. The results from the multiple trials of RRT method are in different colors.

TABLE I

RESULTS OF THE CROSS PARKING SCENARIO

Planner	Nodes	Collision Queries	Time in ms
OSEHS	2 902	32 132	48
SEHS	7 777	89 060	105
Hybrid A*	12 205	80 884	220
RRT	2 414	101 020	244

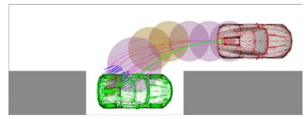
indicated with the different colors. The result is a trajectory that follows this *three segment strategy*.

The results from different planning algorithms are listed in Table I. The hybrid A* employs a grid of 0.5 m resolution for the distance heuristic and to resolve the states. The orientation resolution is 0.1 rad. The step size for the primitive motions is 0.5 m. The OSEHS algorithm shows the best time performance. The grid heuristic used by hybrid A* and the circle-path heuristic applied by SEHS do not contain the vehicle orientation information, therefore they spend a longer time with more nodes to find a solution as illustrated in Fig. 6(b) and Fig. 6(c). The RRT method is close to hybrid A* in planning time, however the motions planned by RRT vary in the length and the number of cusps due to the random samples as in Fig. 6(d).

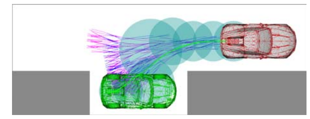
B. Parallel Parking Scenario

In the parallel parking scenario, the vehicle's goal is to park into a parallel parking lot with a 40 cm margin to the front and back. The paths from the four methods are presented in Fig. 7. In Fig. 7(a), two maneuvering regions are suggested by the orientation-aware space exploration. One is in the middle of driving in reverse into the parking space, and the other is around the end position. However, the heuristic search found a path without direction change in the first region for the large free-space. Cusp segments are only planned in the second area.

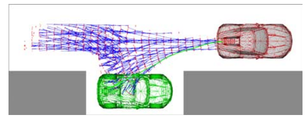
Table II compares the results from the four algorithms.



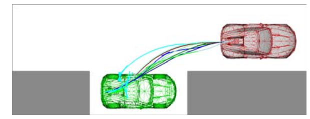
(a) Result of OSEHS method



(b) Result of SEHS method



(c) Result of Hybrid A* method



(d) Result of RRT method

Fig. 7. Parallel parking scenario. The progress of the heuristic search methods are illustrated with *blue* and *magenta* lines for the nodes with forward driving and reverse driving. The results from the multiple trials of RRT method are in different colors.

TABLE II

RESULTS OF THE PARALLEL PARKING SCENARIO

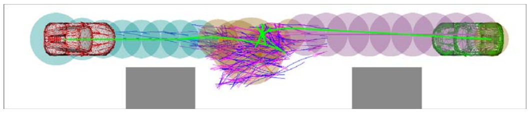
Planner	Nodes	Collision Queries	Time in ms
OSEHS	2 083	18 241	33
SEHS	5 714	58 229	90
Hybrid A*	6 300	61 206	126
RRT	24 231	1 207 970	14 507

The parameters are identical to the cross parking scenario. The OSEHS demonstrates the best time performance in this scenario. The advantage of the SEHS algorithms against the hybrid A* is reduced. Unlike the first scenario which requires a driving direction change in the middle of the maneuver, this scenario only requires maneuvering at the end of the path. Therefore, hybrid A* can follow the grid heuristic with the reverse motions until it reaches the neighborhood of the final position, where the Reeds-Shepp metric provides a good heuristic estimation and is even able to directly reach the goal. However, the grid size and the resolution of the hybrid A* method is reduced to 0.4 m to achieve the result. The RRT algorithm only has a 50% success rate, with the number of nodes limited to 30000. This scenario requires maneuvering close to the obstacles, which creates a narrow passage in the configuration space for the RRT. As in Fig. 7(d), the paths show a less diversity than in Fig. 6(d). The cross parking scenario is easier for the RRT planner, as the vehicle can adjust its orientation outside the parking lot.

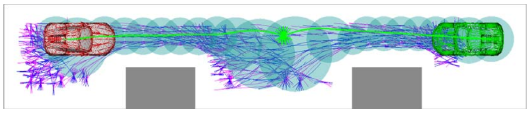
C. Maneuvering Through Narrow Space and Turn-Around

The maneuvering with turn-around scenario is more difficult than the example in Fig. 1, as the goal pose has an orientation opposite to the start configuration. The vehicle must perform a 180° turn during the maneuver. The solutions are shown in Fig. 8. In Fig. 8(a), the space exploration with directed circles suggests to drive forward at the beginning, move in reverse at the end, and to make the turn in the middle where the free-space is relatively large.

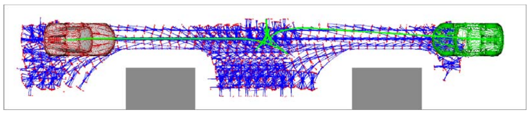
According to the quantitative results in Table III, the OSEHS has the best time performance. The RRT method is at the third place, as it is convenient for the vehicle to make the 180° turn in the middle between the two obstacles, where



(a) Result of OSEHS method



(b) Result of SEHS method



(c) Result of Hybrid A* method

(d) Result of RRT method

Fig. 8. Maneuvering with turn-around scenario. The progress of the heuristic search methods are illustrated with *blue* and *magenta* lines for the nodes with forward driving and reverse driving. The results from the multiple trials of RRT method are in different colors.

TABLE III

RESULTS OF THE MANEUVERING WITH TURN-AROUND SCENARIO

Planner	Nodes	Collision Queries	Time in ms
OSEHS	2845	43475	63
SEHS	12451	126735	148
Hybrid A*	35235	276401	746
RRT	1987	74821	181

the free-space is largest and receives the hightest number of random samples. The vehicle can easily turn with random motions and continues driving to the goal. However, the paths from RRT in Fig. 8(d) still contain more direction changes and unnecessary motions than the search methods. The hybrid A* has the longest planning time in this scenario, because the heuristic of grid-distance without orientation points directly to the goal position in spite of the orientation difference between the start and goal states. Therefore, large number of nodes need to be evaluated to compensate for the discrepancy between the heuristic estimation and the actual cost as in Fig. 8(c). The SEHS method performs better because it adapts the step size and resolution to the circle radius, which results in fewer nodes to close the gap between heuristic estimation and actual motion distance as in Fig. 8(b).

VI. CONCLUSION AND FUTURE WORK

The Orientation-Aware Space Exploration Guided Heuristic Search investigates not only the free space geometry, but also evaluates the driving directions to provide better heuristics for maneuvering through narrow space. The heuristic search takes advantage of such orientation knowledge to improve the search efficiency by adapting the cost metric and the primitive motions. The result of the examples shows that it converges faster than the basic SEHS method and other state of the art algorithms, especially when the planer does not know where maneuvering is required.

Another advantage of the OSEHS method is that it shows constant performance in different maneuvering scenarios. As a result, the OSEHS algorithm can easily be integrated in an autonomous driving system without extra effort to decide when and how to perform the dedicated path planning for the specific parking or maneuvering problems. As future work, an evaluation with a highly automated driving platform is planned.

REFERENCES

[1] 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*, 2013, pp. 1307–1312.

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

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

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

[5] O. Brock and L. Kavraki, “Decomposition-based motion planning: A framework for real-time motion planning in high-dimensional configuration spaces,” in *Proc. IEEE International Conference on Robotics and Automation*, 2001, pp. 1469–1474.

[6] E. Plaku, L. Kavraki, and M. Vardi, “Impact of workspace decompositions on Discrete Search Leading Continuous Exploration (DSLX) motion planning,” in *Proc. IEEE International Conference on Robotics and Automation*, 2008, pp. 3751–3756.

[7] M. Rickert, A. Sieverling, and O. Brock, “Balancing exploration and exploitation in sampling-based motion planning,” *IEEE Transactions on Robotics*, vol. 30, no. 6, pp. 1305–1317, 2014.

[8] I. Paromtchik and C. Laugier, “Motion generation and control for parking an autonomous vehicle,” in *Proc. IEEE International Conference on Robotics and Automation*, 1996, pp. 3117–3122.

[9] B. Müller, J. Deutscher, and S. Grodde, “Continuous curvature trajectory design and feedforward control for parking a car,” *IEEE Transactions on Robotics and Automation*, vol. 15, no. 3, pp. 541–553, 2007.

[10] H. Vorobieva, S. Glaser, N. Minoiu-Enache, and S. Mammar, “Automatic parallel parking with geometric continuous-curvature path planning,” in *Proc. IEEE Intelligent Vehicles Symposium*, 2014, pp. 465–471.

[11] P. Jeevan, F. Harchut, B. Mueller-Bessler, and B. Huhnke, “Realizing autonomous valet parking with automotive grade sensors,” in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010, pp. 3824–3829.

[12] C. Löper, C. Brunken, G. Thomaidis, S. Lapoehn, P. Fouopi, *et al.*, “Automated valet parking as part of an integrated travel assistance,” in *Proc. IEEE Conference on Intelligent Transportation Systems*, 2013, pp. 2341–2348.

[13] L. Han, Q. H. Do, and S. Mita, “Unified path planner for parking an autonomous vehicle based on RRT,” in *Proc. IEEE International Conference on Robotics and Automation*, 2011, pp. 5622–5627.

[14] P. Zips, M. Böck, and A. Kugi, “Fast optimization based motion planning and path-tracking control for car parking,” in *Proc. IFAC Symposium on Nonlinear Control Systems*, 2013, pp. 86–91.

[15] C. Chen, M. Rickert, and A. Knoll, “A traffic knowledge aided vehicle motion planning engine based on space exploration guided heuristic search,” in *Proc. IEEE Intelligent Vehicles Symposium*, 2014, pp. 535–540.

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

[17] 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.