2019 IEEE Intelligent Transportation Systems Conference (ITSC)
Auckland, NZ, October 27-30, 2019

Implementing Voronoi-based Guided Hybrid A* in Global Path Planning for Autonomous Vehicles*

S. Sedighi, D.-V Nguyen, P. Kapsalas, K.-D Kuhnert

Abstract— **Planning a global path to navigate autonomous vehicles from a generic perspective defines the overall maneuvers and performance of autonomous vehicles. Inefficient and time-consuming approaches limit the performance of autonomous vehicles in planning a path to reach the desired target position. This paper presents a low-cost and computationally efficient approach of fusing the well-known Hybrid A* search algorithm with Voronoi diagram path planning to find the shortest possible non-holonomic route in a hybrid (continuous-discrete) environment for autonomous vehicles in valet parking applications. The primary novelty of our method stems from two points: at first, Voronoi diagram is exerted to introduce an improved and application-aware waypoint creator to produce the correct waypoints for the Hybrid A* algorithm and then the derived shortest optimum path regarding the non-holonomic constraints of the urban vehicles is planned using Hybrid A* search algorithm. The method has been extensively tested and validated and proven to up to 45% (30% on average) faster than the basic Hybrid A* algorithm.**

I. Introduction

Planning an optimal path for autonomous vehicles is the main part of the decision maker of each autonomous system, which directly affects the performance of autonomous vehicles and their impact on other surrounding moving objects [1]. Path planning for autonomous vehicles is divided into three main sections; global, local and collision avoidance. Global planning provides to the ego-vehicle a general and overall geometry set of waypoints from the initial to the target position regarding the kinematics of the ego vehicle. Following the global planned path, the local planner plans an optimum path locally regarding the constraints of the ego vehicle and collision avoidance which defines the local maneuvers to avoid any possible unpredicted collisions to the surrounding stationary and moving objects. The assembly of above-mentioned sections provides the platform of the decision maker of each autonomous driving system [1] (Fig. 1).

Planning a path and finding an optimum set of waypoints from the initial position in a configuration space (C_s) to the desired target configure, has been always the major concern in autonomous driving. The way of planning a path defines the optimality of the maneuvers of the autonomous agent [2]. Defining the correct definition of an optimum path for a path planner is relatively subjective. Some researchers define this term as the shortest possible geometrical route from the initial to the goal position [2].

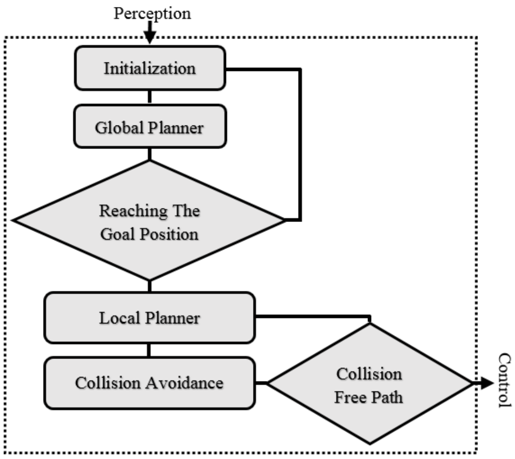

Figure 1. An example of an open-loop decision maker work flow.

Finding the shortest path leads the moving vehicle to save computing time, energy and the required implementing maneuvers. Regarding the non-holonomic agents, the optimal path is the shortest possible route which satisfies the non-holonomic constraints of the vehicle which may affect the length of the planned path (Fig. 2). In this case, the optimal path is selected regarding the kinematics, speed, gear, steering angle, wheelbase, width, steering ratio and other parameters of the vehicle [2]. Additionally, planning a path for autonomous urban vehicles requires more parameters such as safety, margin to the moving and stationary objects, the computational time of the planner and the deterministic characteristic of the decision maker [3].

The main path planner algorithms in mobile robotics established the way of path planning for autonomous cars. Dubins curves and later Reeds-Shepp were the first path planners which were used for non-holonomic autonomous agents [4]. These methods fusing with Clothoid curves could provide an optimal local path in a fully collision-free

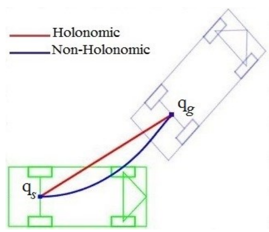

Figure 2. The comparison between holonomic and non-holonomic vehicles regarding the shortest path from the sart (q_s) to the goal (q_g) position.

*Research supported by Panasonic Automotive System Europe GmbH.

S. Sedighi is with Panasonic Automotive Systems Europe GmbH, Langen, Germany (e-mail: saeid.sedighi@uni-siegen.de).
D.-V Nguyen is with Panasonic Automotive Systems Europe GmbH, Langen, Germany (e-mail: duongvan.nguyen@eu.panasonic.com).
P. Kapsalas is with Panasonic Automotive Systems Europe GmbH, Langen, Germany (e-mail: petros.kapsalas@eu.panasonic.com).
K.-D Kuhnert is with university of Siegen, Siegen, Germany (e-mail: kuhnert@fb12.uni-siegen.de).

configuration space [5], [6]. To plan a global path, local planners were merged to graph-based planners such as Visibility and Voronoi diagram planners [7].

Stochastic planners such as road-map and Rapidly Random Trees (RRT) were used for global planning of autonomous vehicles in a given configuration space [8], [9]. Some works implemented discrete search engines such as A* to plan a continuous path for non-holonomic vehicles. Hybrid A* is one of these approaches which provides a smooth drivable path in a global scale for an autonomous vehicle [10].

However, Hybrid A* provides non-holonomic path from the start (q_s) to the goal position (q_g), the route is not guaranteed to be the optimal one. In this method, the optimality of the performance of the algorithm is constrained due to the limitations of the planner such as the limited number of the selected steering angles (heading resolution), runtime and inefficient heuristic cost function of the search engine of the planner [11]. These constraints are more visible when the planner is implemented to plan a global path for an autonomous vehicle in a valet parking scale map in which the given initial and target locations are locally close but far from each other due to the driving aspect (Fig. 3).

This paper introduces a new method of global path planning using an implementation of Voronoi-based diagram to guide the A* search engine of Hybrid A* algorithm to find its optimal non-holonomic path in a hybrid (continuous-discrete) environment faster. This method employs Voronoi diagram algorithm to find the shortest possible holonomic path from the initial to the given goal position to establish enough waypoints to support Hybrid A* search engine in finding its route faster. Using this method, the average computing time of Hybrid A* algorithm was reduced up to 45% (on average by 30%) for big discrete maps (e.g. valet parking use cases). Additionally, the driving style of the planned path regarding the safety margins to the side obstacles (roadsides) in the case of valet parking were improved as well.

A. Related Works

Between all the methods of path planning in Artificial Intelligence (AI), Voronoi-based planner is one the methods which was used in other fields of science for centuries (since 1644) where Voronoi like diagrams were used to present the disposition of the matter in the solar system [12]. Voronoi diagram is applied in computer science in several disciplines from associative file searching and cluster analysis to motion planning for holonomic mobile robots. Due to the low construction time ($O(nlogn)$) of Voronoi diagram this method has been commonly implemented in path planning for mobile robots by embedding the skeleton in the Voronoi diagram of polygonal shapes [13].

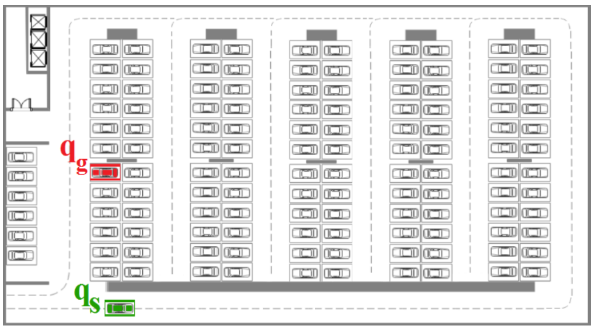

Figure 3. A schematic view of a valet parking area including the start (q_s) and the goal (q_g) positions.

Using the fast marching to support Voronoi diagram, this method was used in path planning of the automotive industry as well [14]. Some works used Voronoi-based graph for improving the cost function of heuristic search engines (such as A*). There, Voronoi approach was mainly used in parallel to other methods such as potential field functions to provide a more effective cost function for each state of the given configurations space [11].

To plan a path for an urban autonomous vehicle with non-holonomic constraints the major matter which needs to be considered is planning a continuous non-holonomic path (as the output of the planner) by using discrete inputs (such as occupancy grid map). To cover these two issues, a lot of holonomic path planners have been upgraded to implement on autonomous urban vehicles using discrete occupancy grid cell maps. Hybrid A* is one of these methods which basically merges the discrete and continuous behavior of different path planners to find a drivable path for autonomous urban vehicles [10]. This method (was introduced in DARPA Urban Challenge in 2007) benefits from A* as a well-known discrete search algorithm to plan a path for non-holonomic vehicles by changing the way which A* search engine expands regarding the kinematic constraints of a vehicle. This method implements Voronoi regions as a cost function to improve its potential field cost function [9]. However, this method is a practical algorithm and frequently implemented in motion planning for autonomous urban vehicles, still this method requires improvements from several aspects. Two main concerns of the planner which need to be improved are first following several waypoints if it's required and second having a better performance regarding the computational time for the real-time applications.

Some works improved Hybrid A* algorithm to follow several tracking waypoints [15]. However, this method improved the waypoint tracking of the algorithm, still, the runtime of the algorithm is a big concern in autonomous driving in urban applications. Here we implement a Voronoi-based diagram path planner to create sufficient waypoints to steer the search engine of Hybrid A* globally with special attention to the local safety margins of roadsides. Using this method, the runtime of the algorithm was improved by 30% and additionally the safety concern regarding the safety margin to the obstacles was improved comprehensibly.

II. VORONOI-BASED GRAPH

Voronoi region and Voronoi diagram are some of the fundamental components of applied geometry. Since the middle of fifteen centuries [12], Voronoi-based graphs have been used in several disciplines, from astronomy to robot motion planning [13]. In all applications, regardless of the Voronoi region or Voronoi diagram, the main basic geometrical concept of all is identical which is keeping the

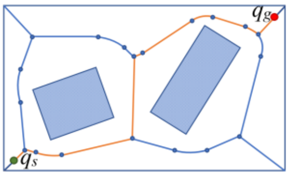

Figure 4. The shortest holonomic path from the start to the goal position using Voronoi-based graph.

lateral distance as much as possible to each obstacle of the map or the provided configuration space (C_s) (Fig. 4) [7].

To implement the Voronoi-based graph on path planning for urban autonomous vehicles, the basic method of the algorithm must be improved from several aspects. If $P = \{p_0, \cdots, p_n\}$ is a set of limited (n) number of separate points (or polygon vertexes) in a 2D area ($p_i \in R^2$), let $D_E(p_i, p_j)$ be the Euclidean distance between p_i and p_j then for all $j \neq i$ ($i, j \in \{0, \cdots, n\}$), the Euclidean distance-based Voronoi regions $V_{AE}(P)$ and Voronoi edges $V_{LE}(P)$ of C_s are defined in $O(n)$ time as follow:

$$V_{AE}(p_i) = \{p \mid p \in R^2, D_E(p, p_i) < D_E(p, p_j)\}, \quad (1)$$

$$V_{AE}(P) = \{V_{AE}(p_0), \cdots, V_{AE}(p_n)\}, \quad (2)$$

$$V_{LE}(p_i) = \{p \mid p \in R^2, p \notin V_{AE}(P), D_E(p, p_i) = D_E(p, p_j)\}, \quad (3)$$

$$V_{LE}(P) = \{V_{LE}(p_0), \cdots, V_{LE}(p_n)\}. \quad (4)$$

In the case of path planning for mobile autonomous agents, p (a valid state) is selected from the obstacle-free (C_f) section of the given configurations space (C_s):

$$P = \{p \mid p \in (R^2 \cap C_f)\}. \quad (5)$$

Considering the polygonal obstacles ($K=\{k_0,\cdots,k_n\}$) in which the borders are weighted in order to include a safety margin of r to the obstacle, the Voronoi-based collision avoiding $V_{OE}(P)$ is defined as:

$$k_{oi} = k_i \cup \{p \mid p \in R^2, D_E(p, k_i) \leq r\}, \quad (6)$$

$$K = \{k_{o0}, \cdots, k_{on}\}, \quad (7)$$

$$V_{OE}(p_i) = \{p \mid p \in (R^2 \setminus K), D_E(p, p_i) < D_E(p, p_j)\}, \quad (8)$$

$$V_{OE}(P) = \{V_{OE}(p_0), \cdots, V_{OE}(p_n)\}, \quad (9)$$

where k_{oi} is the polygonal obstacles number i which is weighted by r amount of safety margin. To prepare a smooth Voronoi curve between and in the case of moving in parallel to obstacles, the edges of weighted polygonal obstacles (e.g. k_{oi}) which are longer than L_0, are discretized to N vertices:

$$N(k_{oi}) = \left(v_i - v_{(i+1)}\right) / L_0, \quad (10)$$

where N is the number of the discretized vertices, k_{oi} is the weighted polygonal obstacle number i and v_i and $v_{(i+1)}$ are its two connected vertices. L_0 is the size of the discretization length (minimum distance between two points) which is selected depending on the resolution of the provided occupancy map. In our vision-based auto parking system the perception level of our autonomous driving system provides an occupancy grid cell map with a resolution of 4×4 (cm^2) per cell. Here we assume L_0 carries a double size of the provided grid cell.

Another modification of the basic Voronoi-based diagram is avoiding driving between the wide areas. As mentioned, in the vision-based auto parking systems, the vehicle fuses the data from the side cameras with all the collected data from other sensors to detect the proposed free parking spaces to park [16], [17]. To detect the parking slots on both sides of the area simultaneously, the vehicle drives in the middle of the parking area and scan both sides of the road at the same time (Fig. 5).

In a wide area, the vehicle must stay at the right side of the road with a minimum safety distance of D_r to the side obstacles:

$$V_{AE}(p_i) = \{p \mid p \in (R^2 \cap C_f), D_{wa}(p)\}, \quad (11)$$

$$D_{wa}(p) = \begin{cases} D_E(p,p_i) < D_E(p,p_j), & D_E(p_i,p_j) < W_O \\ D_E(p,p_i) < D_r, & D_E(p_i,p_j) \geq W_O \end{cases}, \quad (12)$$

and for the Voronoi edges $V_{LE}(P)$:

$$V_{LE}(p_i) = \{p \mid p \in (R^2 \cap C_f), p \notin V_{AE}(P), D_{we}(p)\}, \quad (13)$$

$$D_{we}(p) = \begin{cases} D_E(p,p_i) = D_E(p,p_j), & D_E(p_i,p_j) < W_O \\ D_E(p,p_i) = D_r, & D_E(p_i,p_j) \geq W_O \end{cases}, \quad (14)$$

where D_{we} and D_{wa} respectively are Voronoi edges and Voronoi areas in a wide area. The width parameter W_O is defined depending on the kinodynamic of the vehicle. In some auto parking assistance systems, parking with just one required maneuver is one of the key parameters of the path planner system. Due to this criterion, W_O is assumed to be 4 times of the size of the car width ($4\times W_v$ (car width)) and D_r to be $1.5\times W_v$. In this case, while the vehicle explores the parking area to detect a free space to park, the initial position of the vehicle is well-prepared to terminate the parking maneuver for

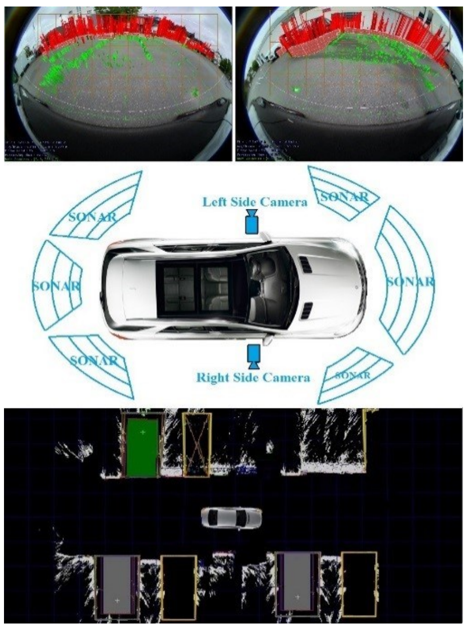

Figure 5. Panasonic vision-based auto parking system.

almost all the detected parking spaces with just one required maneuver [16].

III. ARTIFICIAL POTENTIAL FIELD

In motion planning, over the last decades, the artificial potential field has been frequently allied in motion planning. This method is commonly implemented to define the cost functions in order to avoid any collision to obstacles (which are supposed to carry the maximum potential value) and navigate the ego vehicle to find the target area in which the lowest potential field is located [18]. Extracting the correct potential function which provides the enough resources to convert the given environment including obstacles to the energy waves is the main part of using this method [19]. All the approaches in calculating the potential field for path planning have the following fundamental concept:

$$U(p) = U_a(p) + U_r(p), \tag{15}$$

$$F(p) = -\nabla U(p), \tag{16}$$

where U_a sets as the attractive potential field depending on the distance of the vehicle to the target position. U_r defines the repulsive potential depending on the distance to the obstacles or other restricted areas to visit. $F(p)$ defines the amount of implemented force on the position p of the vehicle.

In global path planning and in particular in the case of Hybrid A* algorithm a simple version of the artificial potential field-based cost function is technically sufficient to sustain the search engine of A* algorithm regarding the collision avoidance of local obstacles [10]. Using this method, the C_f of the provided occupancy grid map is gradient due to the relative locations of its states to the obstacles and the goal position:

$$U_a(p) = \begin{cases} C_a(D_E(p,q_g))^2, & D_E(p,q_g) < D_{max} \\ C_a D_{max}(2D_E(p,q_g) - D_{max}), & D_E(p,q_g) \geq D_{max} \end{cases}, \tag{17}$$

and in an environment with n polygonal obstacles $(k_0,\cdots,k_n)$:

$$U_{ri}(p,k_i) = \begin{cases} C_r\,(R_{max}^{-1} - D_E(p,k_i)^{-1})^2, & D_E(p,k_i) < R_{max}, \\ 0, & D_E(p,k_i) \geq R_{max} \end{cases} \tag{18}$$

$$U_r(P) = \sum_{i=0}^{n} (U_{ri}), \tag{19}$$

where C_a, C_r, D_{max} and R_{max} are the control constants, D_E defines the Euclidean distance between two points and U_{ri} is the repulsive field of obstacle k_i. Using (17) and (19), each grid cell of C_f is biased by which the decision maker of autonomous driving systems avoids any collision to the obstacles and plan more realistic motions which are well oriented to the target position. The artificial potential field of each state of the map can be corrected using a Voronoi-based cost function (distance from the closest Voronoi-based edge),

$$\rho(x,y) = \left({}^{A}\!/_{(A+d_o(x,y))}\right)\left({}^{d_v(x,y)}\!/_{(d_v(x,y)+d_o(x,y))}\right) \times \left({}^{(d_o(x,y)-D)^2}\!/_{D^2}\right), \tag{20}$$

where $\rho(x,y)$ defines the corresponding refined artificial potential field for a position defined by an X-Y Cartesian location. Constants A and D are selected due to the planning strategy, $d_o(x,y)$ and $d_v(x,y)$ are the cost functions related to the artificial potential field and Voronoi-based cost respectively.

Apart from the benefits of exerting the artificial potential field cost functions in path planning, its drawbacks such as expensive cost approaches and getting trapped in its local minima must be taken in the consideration while implementing this method [19]. Fusing Voronoi-based waypoints with the potential cost function can avoid these problems significantly.

IV. HYBRID A* PATH PLANNER

Hybrid engineering approaches in science merge several branches of engineering to benefit the potential of each to dampen the weak points of the others [20]. In the case of Hybrid A* algorithm, the "hybrid" term is used because the algorithm engages an occupancy grid cell map in which the environment is illustrated in a discrete domain and implements a discrete-based search method (e.g. A*) to plan a drivable path in a continuous domain [9].

A* algorithm is a well-known method in path planning [1], [21]. The normal A* search algorithm uses an occupancy grid cell map starting from q_s node and assigns a cost for each surrounding successor node (state) using a cost function as:

$$\rho(p_i) = G(p_i) + H(p_i) + NH(p_i),\ i \in \{0,\cdots,n\}, \tag{21}$$

where p_i is the corresponding state, $\rho(p_i)$ is the related cost of the node p_i, $G(p_i)$ defines the cost to reach the actual node from the initial position $q_s(x_s, y_s, \theta_s)$ and $H(p_i)$ is the heuristic cost to reach the target node $q_g(x_g, y_g, \theta_g)$ from the current node and $NH(p_i)$ is the non-holonomic costs of the vehicle to reach q_g.

In the case of holonomic vehicles, the non-holonomic cost function is set to be zero and the next successor node (or state) with the least cost is selected to move ahead by executing a linear (left-right, up-down) maneuver [22]. For non-holonomic vehicles, applying the kinematics of the ego vehicle, the future motion is predicted depending on the speed, gear, steering angle and other parameters of the vehicle (this method is known as Hybrid A* [10]):

$$\beta = (L/W_b) \times \tan\alpha,\ R = L/\beta, \tag{22}$$

$$X_i = X_{i\text{-}1} + (R \times (\sin(\theta+\beta) + \sin\theta)), \tag{23}$$

$$Y_i = Y_{i\text{-}1} + (R \times (\cos\theta - \cos(\theta+\beta))), \tag{24}$$

$$\theta_i = (\theta_{(i\text{-}1)} + \beta) \bmod 2\pi, \tag{25}$$

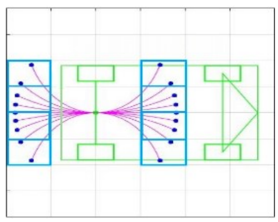

Figure 6. Hybrid A* on a grid map with its successor nodes regarding the heading resolution.

where X_i, Y_i and θ_i present the position of the vehicle in i moment. L defines the desired travel length, W_b is wheelbase and α steering angle (heading) of the ego vehicle and R and β are the dynamic parameters.

This way of selecting successor nodes propel the search engine of the algorithm to select the right nodes which a non-holonomic vehicle can execute [11]. Fig. 6 illustrates how Hybrid A* algorithm appoints the next successor nodes depending on the different positions of the steering angle (heading resolution) of the ego vehicle. Due to the way of selecting successor nodes which are limited discrete states, the planned path ($Q(q)$) Using Hybrid A* path planner, must be post-processed for the desired optimizations regarding the jerk, distance to obstacles, minimum driving distance, minimum energy and so on to avoid any unnecessary maneuver [10].

However, Hybrid A* is a capable path planner for autonomous vehicles, still, its waypoint tracking, safety margin to the obstacles in particular at turning corners and the computing time of the planner must be improved in the case of being applied in the real-time applications. Here we present an advancement of the Hybrid A* algorithm using Voronoi diagram in which the computing time of the algorithm improved beside that the vehicle keeps the safety margin to the side obstacles as much as required.

V. VORONOI-BASED HYBRID A*

To improve the basic version of Hybrid A* algorithm to plan trajectories in real-time applications for the valet parking map scales, Voronoi-based graph was exerted to create the planning waypoints which lead the search engine of A* approach to find its successor nodes much quicker with a simple cost function. This method is explained in detail below:

Input: the occupancy grid map of the environment (C_s), the initial ($q_s(x_s, y_s, \theta_s)$) and target ($q_g(x_g, y_g, \theta_g)$) positions of the vehicle, the vehicle specifications (Veh) and the polygonal obstacles (K_o).

Output: the optimum path ($Q(q)$) for an autonomous non-holonomic urban vehicle from q_s to q_g by executing the following steps:

Step one: the provided polygonal obstacles are refined to add more nodes on edge vectors of polygons in order to create a smoother path (section II).

Step two: the shortest holonomic path from q_s to q_g is planned by using the Voronoi-based graph search [14]. In this step, in addition to the mentioned improvements of the Voronoi-based graph search, we add another feature to Voronoi planner to boost the computing time of the algorithm. Here, all remaining obstacle-end edges must be eliminated from the network:

$$V'_{LE}(p_i) = \{p \mid p \in V_{LE}(p_i), \forall\, k_j \in K \ \exists\ D_E(p,k_j) \le L_c\}, \quad (26)$$

$$V'_{LE}(P) = \{V'_{LE}(p_0),\cdots, V'_{LE}(p_n)\}, \quad (27)$$

$$V_{LE}(P) = V_{LE}(P) \setminus V'_{LE}(P), \quad (28)$$

where $V'_{LE}(P)$ is the set of the configurations which are on the obstacle-end edges and L_c defines the size of the provided grid cells.

Step three: the vertices of the optimum holonomic path is extracted from the planned route as waypoints. Additionally, the mid-points of the edges with the turn vertices at their ends are appended to the planned path. Adding these waypoints to the list of waypoints avoids any unpredicted close maneuvers to the obstacles. Assume the vertices v_L and v_R are the vertices of the turn number i in the direction of the travel:

$$v_{mi}=\begin{cases}[(v_L(x)+v_R(x))/2\,,(v_L(y)+v_R(y))/2], & D_E(v_L,v_R)<W_{oc}\\ [v_{mid}(x),\, v_{mid}(y)], & D_E(v_L,v_R)\ge W_{oc}\end{cases}, \quad (29)$$

$$\theta_{RL}=\arctan\left((v_L(y) - v_R(y))\,/(v_L(x) - v_R(x)\right), \quad (30)$$

$$v_{mid}(x) = D_{Cmax}\times \cos(\theta_{RL}) + v_R(x), \quad (31)$$

$$v_{mid}(y) = D_{Cmax}\times \sin(\theta_{RL}) + v_R(y), \quad (32)$$

where $v_{mi}(x)$ and $v_{mi}(y)$ are the X-Y Cartesian position of the vertex v_{mi} (the same for v_L and v_R), v_L and v_R indicate the left and right side-vertices of each turn in the direction of the travel respectively. W_{oc} is the arbitrary acceptable distance between v_L and v_R and D_{Cmax} is the safety margin to each side of the turn. $D_E(v_L,v_R)$ is the Euclidean distance between v_L and v_R.

Step four: The collected tracking waypoints are employed to direct the Hybrid A* as a multi-goal path planning case [15]. Here, a very simple version of a potential field can be applied to the Hybrid A* cost function to eliminate unnecessary nodes in order to satisfying the dynamic obstacle avoidance criteria (section III).

Step five: the shortest non-holonomic path ($Q(q)$) which is a set of control points or states (q_i) ($i \in \{0,\cdots,n\}$) and planned using Hybrid A* is optimized to avoid any unnecessary maneuver of the vehicle:

$$Q(q) = \{q_0, q_1,\cdots, q_n\}, \quad (33)$$

$$\Delta q_i = q_{i+1} - q_i\,,\ \Delta\gamma = \angle q_{(i+1)} - \angle q_i, \quad (34)$$

$$d_{oi} = \min(D_E(q_i, K)), \quad (35)$$

$$a\sum_{i=0}^{n-1}\left(|q_i - d_{oi}| - D_r\right) + b\sum_{i=0}^{n-1}\left(\left(\Delta\gamma/\Delta q_i\right)^2 - R_{min}^{-1}\right) + c\sum_{i=0}^{n-1}\left(\Delta q_{i+1} - \Delta q_i\right)^2, \quad (36)$$

where $Q(q)$ is the planned path and q_i ($i \in \{0,\cdots,n\}$) is a member of that. a, b and c are the optimization parameters, R_{min} is the minimum turning radius of the vehicle, d_{oi} is the minimum Euclidean distance between the control point (q_i) and the provided polygonal obstacles (K) and $\Delta\gamma$ is the variation of the orientation between the control points (calculated using (30)). The aforesaid steps create a new approach for planning the optimum global path for an autonomous vehicle in a dynamic urban environment.

VI. TESTS AND VALIDATIONS

Due to the importance of safety, efficiency and the reliability of the global path planners of autonomous vehicles, in Panasonic Automotive Systems Europe GmbH this method was implemented on our test vehicles for more than 1500 different (real and simulated) scenarios of occupancy grip map, moving and stationary obstacles, the initial and target positions of the vehicle. The occupancy gild-based map of each scenario was created off and online using our vision-based object detection (perception part). The tests and validations of the program were accomplished as it is illustrated in Fig. 7.

To confirm the results of the proposed method, the same inputs (e.g. polygonal map, start and goal position) were provided for Voronoi-based guided Hybrid A* (this paper), the basic Hybrid A* algorithm [11] and RRTs [9] (as one of the benchmark path planners) and the performance of the algorithms were compared.

The results of this evaluation considerably proved that the Voronoi-based guided Hybrid A* has the best performance of global planning for a large provided occupancy grid map (with the size of 200×200 m^2) among all the mentioned methods.

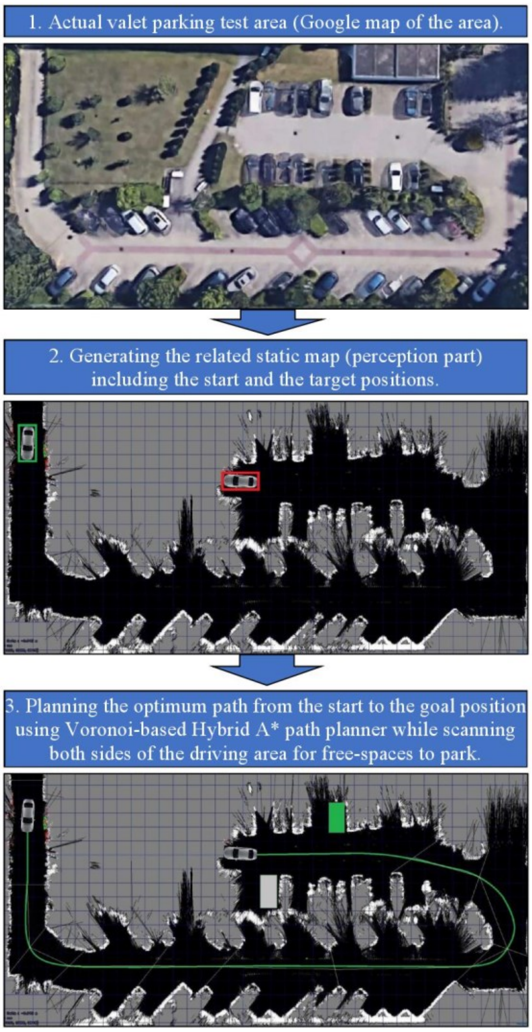

Figure 7. Test and validation of Voronoi-based guided Hybrid A* path planner for actual vision-based valet parking scenarios.

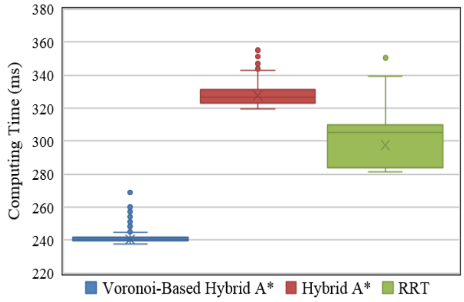

Figure 8. The comparison between the computing times of Voronoi-based and the basic Hybrid A* and RRTs planners for the identical inputs.

The results of this study evidence that Voronoi-based guided Hybrid A* is a very practical method in path planning for autonomous vehicles in valet parking applications. Voronoi-based guided Hybrid A* provides a comprehensive improvement on the original version of Hybrid A* such as the number of visited node, path length and the computing time of the algorithm.

The required computational time of the algorithms to find the shortest non-holonomic path to the target position illustrate that Voronoi-based guided Hybrid A* plans its non-holonomic optimum path with around 30 % and 23% respectively less required time in comparison to the basic Hybrid A* and RRTs path planners (Fig. 8).

The most important improvement of Voronoi-based guided Hybrid A* is the number of visited nodes (occupancy grid cells) while planning a path. Following the provided waypoints, assists the planner in avoiding visiting unnecessary nodes in particular in critical use cases such as Fig. 3. Visiting fewer grid cells, in addition to the saving of computing time, allows the planner to manipulate smaller sizes of grill cells which increase the accuracy and optimality of the planned path.

Due to the limitation of the runtime in real-time applications for autonomous systems, Hybrid A* implements its planning process for occupancy grid maps with low to normal resolution (around 15×15 cm^2) of the provided grid cells and allying a less precise heading resolution of (5°). These limitations decrease the accuracy of the planned path using Hybrid A* [11].

Using Voronoi-based Hybrid A* the optimum non-holonomic path for an autonomous vehicle is planned while visiting up to 40% fewer grid cells (nodes) in comparison to the basic algorithm of Hybrid A* (while using the same size of grid cell) (Fig. 9). Fewer visited nodes allows the autonomous

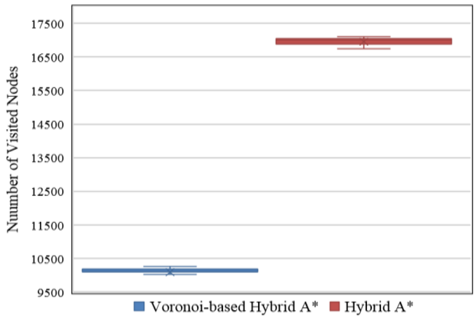

Figure 9. The comparison between the number of visited nodes in Voronoi-based and the basic Hybrid A* algorithms for the identical

system to upgrade the resolution of the occupancy grid-based map and the heading resolution up to respectively 4×4 cm^2 and 3° in a big grid map (up to 200×200 m^2). A higher resolution of the occupancy grid map assists the planner to plan a safer and smoother trajectory due to the more available states to proceed.

In comparison to the basic algorithm of Hybrid A*, the Voronoi-based guided planner has a preferable performance regarding the safety margin to the side obstacles, length and the drivability of the planned path. By comparing the length of the planned path in Voronoi-based and the basic Hybrid A* algorithms, however the safety margin criterion of the planning is carefully considered by Voronoi-based planner, there is still a visible difference between the lengths of the planned routes. The tests and validations of the algorithms (e.g. Fig. 10) reveal that planned paths using Voronoi-based Hybrid A* are on average 20% shorter than the planned path using the basic Hybrid A* algorithm (Fig. 11). In comparison to RRTs, the planned path using Voronoi-based Hybrid A* is on average 30% shorter, due to tracking of the shortest holonomic waypoints to the target position.

Using waypoints to guide the search engine of Hybrid A* improves the predictability and reliability of the planner. As it is shown in Fig. 8, 9 and 11, the tolerances and variations (the deviation between the mean value, maximum and minimum of the measuring parameters) in the performance of the three mentioned planners retain their lowest values for Voronoi-based guided Hybrid A* in all the tests (i.e. computing time, number of visited nodes and length of planned path).

Another feature of the planner is the stability of the performance of the Voronoi-based guided Hybrid A* algorithm to prove the deterministic characteristic of the algorithm. The results of the test and validation of the mentioned method demonstrate that the Voronoi-based guided Hybrid A* planner could successfully plan the optimum global path to the given target area for 100% of the proposed scenarios (if a path was available) under around 250 (*ms*). This feature of the global path planners has an essential effect on autonomous urban vehicles.

VII. Results

In this paper, a new approach of global and local motion planning for autonomous urban vehicles we introduced by reconstructing the well-known Hybrid A* algorithm using a set of leading waypoints which are created by a Voronoi-based graph.

By implementing our improvement on the basic algorithm of Hybrid A* path planner, the performance of the basic

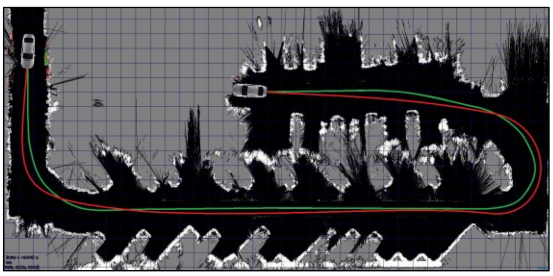

Figure 10. An example of a comparison between Voronoi-based guided Hybrid A* and Hybrid A* in planning a non-holonomic path.

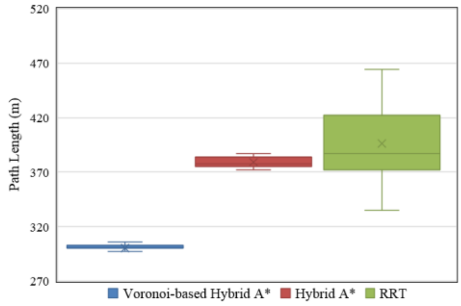

Figure 11. Comparison between the length of the planned paths by Voronoi-based and the basic Hybrid A* and RRTs planners for the identical inputs.

method was enhanced significantly. The results which were achieved by implementing the algorithm on almost 1500 different scenarios (real and simulated) of map and vehicle locations prove that using the Voronoi-based diagram algorithm to locate the correct waypoints to steer the main search engine of Hybrid A* improves the performance of the decision-making level of an autonomous urban vehicle for valet parking applications. Having a reasonable running time of the algorithm proves its application for real-time autonomous systems in order to deal with fully dynamic environments.

To verify the improvement and performance of the proposed method, the algorithm was tested precisely over 1500 different use cases. The same inputs were provided for the basic algorithm of Hybrid A* and RRTs (as one of the current benchmark path planners) and the performances of all the algorithms regarding the computational time, number of visited nodes and length of the planned routes were recorded and analyzed. The results of the mentioned benchmarking proved that in comparison to the basic Hybrid A* algorithm, the average reduction of the running time of the proposed method was about 30% (up to 45%) in finding the shortest non-holonomic path to the target position while visiting up to 40% fewer nodes (grid cells). Visiting fewer grid cells allows the planner to increase the resolution of using occupancy grill map (in our case up to 4×4 cm^2) and planning a more accurate and smoother path.

Evaluating the performance of Voronoi-based guided Hybrid A* against RRTs, shows a better performance regarding the runtime of the algorithm. Voronoi-based guided Hybrid A* plans the required non-holonomic path while requires on average 23% less computing time.

In comparison to the basic Hybrid A* and RRTs algorithms, however, using the proposed method of this paper the safety margin to the obstacles is carefully considered which keeps the ego vehicle far enough from the side obstacles, the length of the planned path is still around 20% and 30% respectively shorter than the basic version of Hybrid A* and RRTs.

Using a Voronoi-based graph lead the decision maker of an autonomous urban vehicle to plan trajectories which are located in the middle of the parking area. In Advanced Driver-Assistance Systems (ADAS) applications; driving in the middle of the parking area provides an opportunity for the vision-based parking slot finders (free space finder) to detect the free spaces to park at both sides of the parking area simultaneously.

The proposed algorithm was tested and validated for both static and dynamic scenarios which demonstrates the capabilities of the applications of Voronoi-based guided Hybrid A* algorithm for real valet parking functions where the vehicle deals very dynamic environments. The mentioned test and validation use cases were implemented by MATLAB R2015a on Win7 64-bit with Intel(R) Xeon(R) CPU 3.50 (*GHz*) processor.

VIII. Conclusions

In this paper, an approach of optimum global path planning for autonomous urban vehicles in valet parking applications was presented. This method benefits the advantages of Voronoi-based diagram algorithm to improve the performance of Hybrid A* path planner by creating the guiding waypoints for the search engine of the planner.

As it was proved, merging two different path planners reduced the runtime of the decision-making section of autonomous vehicles by 45%. Having a faster decision maker helps the whole AI of autonomous vehicles to make decisions faster at the right time to handle the dynamic scenarios which may save lives as well.

The results of this work demonstrate the great application of hybrid methods in path planning for the automotive industry in particular for autonomous driving applications. As it was proved in this paper, using a holonomic path planner (Voronoi-based graph) helped a non-holonomic planner (Hybrid A*) to find its optimal route much faster and more accurate by providing the planning waypoints. In this case, the non-holonomic constraints of graph-based planners and the heuristic cost function of Hybrid A* as a non-holonomic planner were improved at the same time by merging these two methods in one path planning algorithm.

Hybrid approaches in engineering provide an opportunity to benefit the advantages of different methods individually and put down their weak points at the same time. Back to this paper, fusing of path planning methods might be applied to other path planners to improve their performance as well.

References

[1] S. M. LaValle, *Planning Algorithms*, Cambridge: Cambridge University Press, New York, NY, USA, pp.63-295, August 2006.

[2] H. Choset, K. M Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. Kavraki and S. Thrun, *Principles of Robot Motion: Theory, Algorithms, and Implementations*. MIT press, ISBN: 9780262033275, pp.373-465, 2005.

[3] B. Paden, M. Cap, S. Z. Yong, D. Yershov and E. Frazzoli, "A Survey of Motion Planning and Control Techniques for Self-Driving Urban Vehicles," *IEEE Transl. Intelligent Vehicles*, vol. 1, no. 1, pp. 33-55, March 2016.

[4] J. A. Reeds, L. A. Shepp, "Optimal paths for a car that goes both forwards and backwards," Pacific J. Math. Vol. 2, pp. 367-393, 1990.

[5] A. Scheuer and T. Fraichard, "Continuous-curvature path planning for car-like vehicles," *IEEE/RSJ Int. Conf. Intelligent Robot and Systems. Innovative Robotics for Real-World Applications*. *IROS* 1997, Grenoble, France, vol. 2, pp. 997-1003, 1997.

[6] T. Fraichard, A. Scheuer, "From Reeds and Shepp's to continuous-curvature paths," *IEEE Transl. Robotics*, pp.1025-1035, June 2004.

[7] M. de Berg, O. Cheong, M. V. Kreveld and M. Overmars. *Computational Geometry: Algorithms and Applications*, Springer-Verlag TELOS Santa Clara, CA, USA, vol.3, pp. 283-332, 2008.

[8] K. Marti and S. Qu, "Optimal trajectory planning for robot considering stochastic parameters and disturbances – computation of an efficient open-loop strategy," *J. Intell. Robot. Syst.*, pp. 19–23, 1996.

[9] Y. Kuwata, G. A. Fiore, J. Teo, E. Frazzoli and J. P. How, "Motion planning for urban driving using RRT," 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, Nice, 2008, pp. 1681-1686.

[10] D. Dolgov, S. Thrun, M. Montemerlo and J. Diebel, "Practical search techniques in path planning for autonomous driving" Proc. of the First *International Symposium on Search Techniques in Artificial Intelligence and Robotics* (*STAIR*-08), June 2008.

[11] D. Dolgov, S. Thrun, M. Montemerlo and J. Diebel, "Path Planning for Autonomous Vehicles in Unknown Semi-structured Environments," *The International Journal of Robotics Research*, vol. 29, no. 5, pp. 485–501. Apr. 2010.

[12] A. Okabe, B. Boots and K. Sugihara, "Spatial Tessellations: Concepts and Applications of Voronoi Diagrams," John Wiley & Sons, Inc., New York, NY, USA, 1992

[13] P. Bhattacharya and M. L. Gavrilova, "Voronoi diagram in optimal path planning," *4th International Symposium on Voronoi Diagrams in Science and Engineering* (*ISVD* 2007), Glamorgan, pp. 38-47, 2007.

[14] S. Garrido, L. Moreno, M. Abderrahim and F. Martin, "Path Planning for Mobile Robot Navigation using Voronoi Diagram and Fast Marching," 2006 *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, pp. 2376-2381, 2006.

[15] J. Petereit, T. Emter, C. W. Frey, T. Kopfstedt and A. Beutel, "Application of Hybrid A* to an Autonomous Mobile Robot for Path Planning in Unstructured Outdoor Environments," ROBOTIK 2012; *7th German Conference on Robotics*, Munich, Germany, pp.1-6, 2012,

[16] S. Sedighi, D. Nguyen, K. D. Kuhnert, "Introducing a Parking State Machine and Path Planner Algorithm for Vision-Based Perpendicular Parking Scenarios," *International Conf. on Control, Decision and Information Technologies* (*CoDit* 2019), Paris, France, 2019.

[17] D.-V. Nguyen, *Perceptual Inference for Autonomous Navigation*. Germany: Südwestdeutscher Verlag für Hochschulschriften, ISBN: 9783838139883, 2014, ch. 5.

[18] C. W. Warren, "Global path planning using artificial potential fields," *Int. Conf. on Robotics and Automation, Scottsdale*, vol. 1, pp. 316-321, 1989.

[19] D. Fu-guang, J. Peng, B. Xin-qian, W. Hong-jian, "AUV local path planning based on virtual potential field, Mechatronics and Automation," *IEEE Int. Conf. Mechatronics and Automation,* 2005, Niagara Falls, Ont., Vol. 4, pp. 1711-1716, 2005.

[20] S. Seda, M. R. Tolun and R. Hassanpour, "Hybrid expert systems: A survey of current approaches and applications." *International J. Expert Systems with Applications*. vol. 39, no. 4, pp. 4609-4617, March 2012.

[21] J. Pearl, Heuristics: *Intelligent Search Strategies for Computer Problem Solving*. Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA, 1984, 501-535.

[22] P. E. Hart, N. J. Nilsson and B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths" in *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100-107, July 1968.