GRAPH BASED PATH PLANNING IN UNKNOWN ENVIRONMENTS USING VORONOI DIAGRAMS

A. Das\textsuperscript{1}, P. Mukherjee\textsuperscript{1}, C. Wang\textsuperscript{1}, G. Salas\textsuperscript{1}, S. Bedi\textsuperscript{2}, S. L. Waslander\textsuperscript{3}

\textsuperscript{1}Graduate Student, \{adas, pmukherj, c25wang, gsalas\}@uwaterloo.ca
\textsuperscript{2}Professor, sanjeev.bedi@uwaterloo.ca
\textsuperscript{3}Assistant Professor, stevenw@uwaterloo.ca

ABSTRACT

This work presents a method for online path planning based on graph traversal for autonomous mobile systems. A method using the Voronoi diagrams of obstacle dense areas for efficient planning is proposed. This method creates collision free paths for the vehicle in a receding horizon fashion on a per time-step basis, for environments where no prior map data is available. Simulation results of the proposed method running on a four wheel Ackermann drive vehicle are presented. Furthermore, the method is experimentally shown to produce good results on a custom-made platform in an autonomous circuit-race setting.

INTRODUCTION

Autonomous online path planning is a core problem to be solved if robust and dependable autonomous behaviour in mobile platforms is to be achieved. In this regard, Voronoi diagrams offer an efficient method to partition the free space of the environment. The Voronoi diagram partitions the space such that the region edges are the furthest possible from obstacles. Various path planning techniques which utilize Voronoi diagrams have been developed, and these aim to maximize the vehicle’s distance from the obstacles, thus producing a relatively “safe” path for traversal. For example, a graph search over a Voronoi diagram is used by [1] to create a collision free path for a UAV that minimizes a weighted cost that penalizes total path length and the distance from radar sites. In [2], a generalized Voronoi diagram (GVD) is used to plan a path for a vehicle in a planar workspace with polygonal obstacles. The GVD and its properties are described in detail by [3], who also describes PRM sampling strategies based on generalized Voronoi diagrams. In order to smooth the path generated by the GVD graph search, [4] and [5] utilize Bezier curve smoothing and corner cutting techniques to construct smooth roadmaps for traversal. Finally [6] use GVD’s to construct a Voronoi potential field. This utilizes the GVD to rescale the potential field according to distances of a point to the GVD edges. This results in the formation of low potential areas near the GVD edges, which greatly assists in navigation through tight or confined corridors - a common issue with standard potential field path planning algorithms. However, almost all of these works assume complete environmental awareness prior to the vehicle operation. The works also do not offer the computational efficiency required for online real-time implementation of such algorithms. This work presents a method that can be used online for real-time path planning in unknown environments assuming adequate obstacle features are present. It is also assumed that the obstacles can be represented as points. An explanation of the algorithm, simulation and implementation results are presented. As a result of this design, this work was awarded the Best Design and placed first in the circuit race contest at the 2010 International Autonomous Robot Racing Competition (IARRC) held in Windsor, Ontario, Canada in July 2010.

PLANNING STRATEGY

A. Voronoi Diagrams

In this section a brief introduction of the definition of the Voronoi diagram in 2D is given. Consider a set of points \( P = \{p_0, p_1, \ldots, p_n\} \) on the two dimensional Euclidean space \( \mathbb{R}^2 \). For the proposed method, these points are considered to be obstacles. The Voronoi diagram partitions \( \mathbb{R}^2 \) into regions based on the obstacle points. Each obstacle point in the set \( P \) has a Voronoi region, call it \( R(p_i) \). For any point \( x \in R(p_i) \), \( p_i \) is the closest sample to \( x \) using a Euclidean distance metric [7]:
\[
R(p_i) = \{ x : \|p_i - x\| \leq \|p_j - x\|, \forall j \neq i \}
\] (1)

Figure 1: Example of a Voronoi diagram where the obstacle points are lane boundaries

Figure 1 illustrates a Voronoi diagram where the obstacle points are vehicle lane boundaries. It is worth noting that the Voronoi edges are centred between surrounding pairs of obstacles. It is this phenomenon that makes Voronoi diagrams a useful tool for obstacle avoidance.

B. Proposed Method

The previous and related work using Voronoi diagrams for path planning all assume that the environment and obstacles are static and well known prior to operation. The proposed method utilizes a receding horizon planning paradigm by which the vehicle plans locally at every time-step of execution. This is done with no prior map information as the planning is solely based on observed obstacles and features at every time step. Thus, the proposed method will perform in any unstructured and unknown environment assuming that there are adequate features and obstacles to construct the Voronoi diagrams.

To demonstrate the algorithm, consider a situation where the vehicle must drive and stay within the bounds of a lane. The boundaries of the lane can be considered to be constructed from discrete points, and obstacles present within the boundary of this lane will be considered obstacle points as well. In practice, these features can be constructed from a variety of sensors such as LIDAR, ultrasonic measurements, vision systems, etc. The obstacle and feature data is conveyed in the vehicle co-ordinate frame, thus the vehicle is always at point (0, 0).

At every time-step, a set of obstacle features is obtained from the sensor suite of the vehicle, define this as the set \( P \in \mathbb{R}^2 \). From this, the Voronoi diagram of the obstacle points is constructed, as seen in Figure 1. The points at which three or more Voronoi edges meet is defined as a Voronoi vertex, and the set of these vertices is \( V \in \mathbb{R}^2 \). In order to find a traversable path, the Voronoi vertices are connected in a graph, as seen in Figure 2. Denote the set of all edges in the graph as the set \( E \subseteq V \times V \), and the full graph as \( G = (V, E) \). While it is possible to use the edges in the Voronoi diagram itself as the vehicle path, the construction of a connected graph between the Voronoi vertices allows the vehicle to traverse shorter paths through the environment. However, extra computation is required to ensure the connected graph is collision free which is an acceptable trade-off in the given scenario.

In order to ensure that the path is traversable and collision free, collision checks between every edge of the connected graph and obstacle points is performed. Between every obstacle point \( p_i \) in \( P \) and every graph edge \( e_j \) in \( E \), the distance between the two is ensured to be less than \( L_{\mu} \), a user controlled parameter which determines the minimum clearance desired between an obstacle point and the vehicle. This parameter controls the size of gaps in the environment which the vehicle can travel through. Erroneous obstacle measurements, as well as the geometry of the obstacle configuration may lead to the creation of outlier Voronoi vertices which should not be considered part of the feasible vehicle path. In order to ensure that the outlier Voronoi vertices are not connected, the maximum length of an edge is restricted to be less than \( L_e \). The subset of collision free edges between Voronoi vertices is defined as,
\[ E_C = \{ e : d(e, p_i) \geq L_p, \|e\| \leq L_e \} \]  
(2)  
where \( d(e, p_i) \) is the function which computes the perpendicular Euclidean distance between edge \( e \) and point \( p_i \). The collision free graph is \( G_C = (V_C, E_C) \), where \( V_C \) is the set of vertices connected to an edge in \( E_C \). The collision free graph can be seen in Figure 3.

In order to perform a graph traversal over \( G_C \), a goal point for the vehicle must be selected. Since the algorithm is performed at every time step, the current goal point \( v_g \) can be selected as the Voronoi vertex in \( V_C \) which is the farthest away from the vehicle. The goal point is:

\[ v_g = \max_{v_j \in V_C} \|v_j\| \]  
(3)  

Figure 4: Lowest Cost path for vehicle. The vehicle is at position (0,0)

Once the goal point has been determined, the lowest cost path through the graph can be determined using a graph traversal algorithm such as Dijkstra’s algorithm or A*. In this case, the cost associated with each edge is simply the length of the edge itself. The final vehicle path for the current time-step can be seen in Figure 4.

SIMULATION RESULTS

The proposed method is tested in a scenario outlined by the 2010 IARRC competition. Simulation studies are performed on a four wheel Ackermann ground vehicle navigating a course where the lane boundary is constructed by pylons. The vehicle’s LIDAR is also simulated, from which obstacle features are extracted. This is done using a simple peak detection algorithm which constructs obstacles from the local maxima and minima of the LIDAR scan, as seen in Figure 5.

The vehicle motion is simulated using the bicycle steering model,

\[
\begin{bmatrix}
x_t \\
y_t \\
\theta_t \\
\end{bmatrix} =
\begin{bmatrix}
x_{t-1} + v \cos(\theta_{t-1}) dt \\
y_{t-1} + v \sin(\theta_{t-1}) dt \\
\theta_{t-1} + \frac{v}{L} \tan(\delta) dt \\
\end{bmatrix}
\]  
(4)

where \( x \) and \( y \) are the vehicle’s position, \( \theta \) and \( \delta \) are the vehicle’s heading and steering angle respectively, \( v \) is the vehicle velocity, \( L \) is the vehicle wheelbase and \( dt \) is the simulation time-step. For the simulation, a constant velocity of \( \frac{3m}{s} \) is commanded. The steering controller for this simulation is a simple proportional controller which provides control effort quadratically proportional to the heading error,

\[ \delta = \text{sgn}(\theta_d - \theta_t)K(\theta_d - \theta_t)^2, \quad \delta \in [\delta_{\text{min}}, \delta_{\text{max}}] \]  
(5)

where \( \theta_d \) is the desired heading given by the proposed path planner, \( \theta_t \) is the current heading, and \( K \) is the controller gain, and \( \delta_{\text{min}} \) and \( \delta_{\text{max}} \) is the minimum and maximum steering angle respectively. This style of quadratic proportional controller is utilized to dampen the effect of small heading errors caused by sensor noise and other disturbances, while still aggressively correcting for large heading errors.

Figure 5: Feature extraction from typical LIDAR scan. The minima correspond to obstacle points

Figure 6: Simulation of full track

Using the bicycle model and quadratic steering controller, the motion of the vehicle is simulated through the course. This can be seen in Figure 6. It should be noted that the vehicle begins the simulation with no prior data of the environment.

PLATFORM DESIGN

The platform designed to participate in the 2010 IARRC and experimentally test the proposed path planning method is a custom made 1/12 scale monster truck chassis. The platform is built on a commercially available chassis modified for computer control. It is designed with a diverse suite of sensors that include a Novatel OEMV3 GPS (#1 in Figure 7) for position estimation and Hokuyo UTM-30LX LIDAR (#5 in Figure 7)
measurement system for obstacle detection and feature extraction. It is also custom fitted with optical encoders to allow for closed loop feedback velocity control (#3 in Figure 7). Other sensors include a vision system and an IMU board (#2 and #4 in Figure 7). The system has a 2.1Ghz dual-core processor and ARM7 low level microcontroller, allowing for complete on-board processing. A schematic of the vehicle sensor layout is shown in Figure 7.

Figure 7: Sensor layout for the vehicle.

EXPERIMENTAL RESULTS

The proposed path planning algorithm was implemented on the University of Waterloo IARRC racing platform. The racing platform and the output of the implemented path planner can be seen in Figure 8. During testing, the vehicle was able to navigate complex course features such as centre islands, switchback turns, and hairpins without collision. The computation time of the algorithm was negligible relative to the 40 Hz scanning rate of the LIDAR. Footage of a full circuit run including the path planner output can be found in [8].

Using the proposed receding horizon Voronoi based path planner, a secondary platform provided by Clearpath Robotics achieved a first place finish in the circuit race competition at the IARRC 2010 competition, and also won first place in the design contest.

Figure 8: Experimental results for vehicle navigating circuit course. (a) The physical vehicle driving through the course. (b) Output of the Voronoi based path planning algorithm implemented on the vehicle

CONCLUSION

This paper proposed a graph based approach using Voronoi diagrams for motion planning in 2D. Most previous work in this area assumed a known map of the environment. The proposed path planner is unique in this sense since it requires no prior map or environment information. The locally planned path is computed at every time-step in a receding horizon fashion. This allows for the vehicle to traverse unknown environments assuming that there are adequate obstacle features present.

The proposed path planner was implemented on a scale 1/12 model, and entered in the 2010 IARRC competition. Using the proposed path planning algorithm, a secondary vehicle achieved a first place finish in the circuit race competition and a first place finish in the design competition.

REFERENCES


