Path Planning in Urban Area Using Local Features of the Road System

Csaba ANTONYA
Department of Automotive and Transport Engineering, Transilvania University of Brasov, 29 Eroilor Blvd., 500036 Brasov, Romania

Keywords: Path planning, Search algorithm, Navigation, Travel time.

Abstract. In this paper is proposed a tool for path planning in urban area. The proposed method can anticipate travel time for vehicles even for future dates and hours or can decide whether to revise the current path based on the local features of the road system ahead. Travelling time and waiting time are analyzed on several features of the road system, like pedestrian crossing, traffic lights, and roundabouts. These features are typical for cities in Europe, built around medieval settlements. The method can also be expanded to include other features as traffic accidents, meteorological conditions, cultural or sport activities which will perturb the normal traffic conditions. Finally, a simulation is performed to evaluate the proposed method using the A-star graph searching algorithm.

Introduction
Most of the small and medium towns and cities in Western and Center Europe were inhabited continuously since the Neolithic period. The hearts of these are mainly well-preserved medieval towns. The roads were used mainly by carts, wagons, carriages, pedestrians and horses. The current states of the road network in the core of the cities are based on the development in the first half of the last millennium. After the industrial revolution, around the core of the cities new development were made during the 19th Century and in the first half of the 20th Century, when the number of vehicles on the road was far smaller than in the current period. These road networks in and around the medieval cities were not conceived for the present heavy traffic and the exponential growth in the number of vehicles on the road. Many improvements followed, like the construction of new roads and roundabouts, introducing traffic lights and signalized cross intersections, adding new lanes on existing roads, transforming narrow roads into one ways.

Turn-by-turn real-time navigation services are provided by the automobile navigation systems or handheld devices. These are based on satellite positioning system, accurate maps and software like Garmin, TomTom, iGo or Google Maps. The path planning can be refined with data obtained on the traffic message channel, which is broadcasting warning messages about the traffic ahead, or analyzing the live traffic as on Google Maps [1].

In this paper a new method for path planning in urban area is proposed. By analyzing the traveling time and waiting time on different features of the road system ahead, like pedestrian crossing, traffic lights, and roundabouts, the travel time for vehicles can be anticipated even for future dates or the current path can be revised.

Path Planning in Urban Area
The task of the path planning or route finding algorithms is to reach the specified goal in a search-tree with minimal cost (the cost could be time, fuel price, traveled distance etc.) [2,3]. The uninformed or blind search methods like Breadth first search, Uniform cost search, Depth first search, Depth limited search, Iterative Deepening, Bi-directional Search are all too slow for most real world problems [4]. The informed or heuristic approaches are more efficient. The Heuristic function evaluates the relative desirability of expanding a node and helps to reduce the number of alternatives [5]. These algorithms are A-Star (A*), Pure Heuristic Search, Iterative-Deepening A*, Hybrid-state A*, Recursive Best-First Search, D*, Field D* [6,7,8] etc.
The A* algorithm is an optimized version for a single destination of the Dijkstra Algorithm [9], because it is only focusing to reach the destination node from the current node, not to reach every other nodes in the search tree. The heuristic function, when applied to a state, returns a number that is an estimate of the merit of the state, with respect to the goal. A* is fast, for any given heuristic, no algorithm can expand fewer nodes than A*. The speed depends on the quality of the heuristic, poor one degenerate to uniform cost, perfect one is just going down the tree to the goal. Generally, we are somewhere in between the two situations above and the speed depends on the quality of the heuristic. The algorithm has two lists, an open list from which the next node for expansion is selected and a closed list of nodes that have been expanded.

For the path-planning in urban area the 2D grid-based search tree is employed [9]. The road network is transformed into a uniform 8-connected 2D grid representation. Moving from one grid cell to another can have different travel cost. This travel cost is measured in time intervals (seconds) and it was established with the average of 25 measurement taken at different time on a working day (at 10am, 1pm, and 6pm). The measurements are presented in Table 1. The travelling time depends on the feature of the road system came across when moving in a neighbor cell. These features are the pedestrian crossing, roundabouts and traffic lights. In case there is no special feature, then the travel time depends on the normal cruising speed in straight direction (travelling North, East, South or West) or diagonal direction. These movement costs are taken into account when deciding how to evaluate various locations on the grid.

<table>
<thead>
<tr>
<th>Feature:</th>
<th>Time on workday [hours]</th>
<th>Average [s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pedestrian crossing</td>
<td>10.1 s</td>
<td>11.3 s</td>
</tr>
<tr>
<td>Roundabout</td>
<td>23.3 s</td>
<td>23.3 s</td>
</tr>
<tr>
<td>Traffic light – green</td>
<td>46 s</td>
<td>46 s</td>
</tr>
<tr>
<td>Traffic light – red</td>
<td>76 s</td>
<td></td>
</tr>
<tr>
<td>Travelling to a N-E-S-W neighbor</td>
<td>5.7 s</td>
<td>6.03 s</td>
</tr>
<tr>
<td>Travelling to a diagonal neighbor</td>
<td>8.06 s</td>
<td>8.52 s</td>
</tr>
</tbody>
</table>

The road network of the city of Brasov, Romania, is presented in Figure 1. The city was founded by the Teutonic Knights in 1211, and around the medieval city center the new part was developed in the middle of the 20th Century. For crossing the city from South to Nord (starting point on the lower right
corner to a destination on the upper left corner) there are 4 different routes available, beside the one which is considered the main route (marked with E60).

On the road network from Figure 1 a grid was overlaid and a matrix of discrete grid points (or cells) was obtained (presented in Figure 2). This is transformed into a tree structure, a hierarchy of connected nodes, where each node represents a particular position on the road. The A* search algorithms will find a solution by traversing the tree structure of the 8-connected nodes on the grid. The solution is the path from the initial position of a vehicle to a “goal” node, representing the desired destination. The constructed heuristic function for the search algorithm is the Manhattan distance on the square grid from the goal to each location on the grid.

The A* algorithm is checking the adjacent squares starting from the initial position and expands one by one the nodes on the open list if it seems promising. It only focuses to reach the goal node and key to determining which node to use next is to minimize the following equation:

\[ F = G + H \]  \hspace{1cm} (1)

where G is the time to move from the starting point to a given square on the grid by following the road to get there, and H is the estimated travel cost (time) to move from that given position on the grid to the destination, obtained from the heuristic function.

The solution of the algorithm is the connected cells in the tree structure, representing the quickest path to reach the goal position from the current position.

![Figure 3. The obtained path on the road system with street names.](image)

**Results and Conclusions**

In Figure 3 is presented the solution for the quickest route for reaching the destination from the starting position. The simulation corresponds to crossing the city at 5pm. The travelling time through different features for the considered time were obtained with linear interpolation. The proposed algorithm is calculating 608 seconds for this journey. This result is almost the same as the routes offered by three well-known navigation software (Figure 4: a - Google Maps, b - iGo and c -}
TomTom), which were all offering the same route and the travel time of approximately 10 min with live traffic data included (note that the grid in Figure 3 is the mirror image of the maps in Figure 4).

In the presented case the features considered were pedestrian crossing, roundabouts and traffic lights. The method can also be expanded and other features or events can be included, like road
accidents, meteorological conditions (heavy rain, fog, snow and ice on the road), cultural or sport activities which will perturb the normal traffic conditions.

A disadvantage of the A* search algorithm is that requires large amounts of memory, because it needs to maintain a list of unsearched nodes. In the proposed method the A* can be replaced with other, more efficient search method like Iterative Deepening A*, Simplified Memory Bounded A*.

Acknowledgement

This paper is supported by the Romanian Government, specifically MEN – UEFISCDI authority under the program PNII “Partnerships in priority areas”, under the project number 240/2014 - NAVIYEYES, supporting the collaboration between the company Route 66 and University Transilvania of Brasov.

References


