AUV PATH PLANNING: AN A* APPROACH TO PATH PLANNING WITH CONSIDERATION OF VARIABLE VEHICLE SPEEDS AND MULTIPLE, OVERLAPPING, TIME–DEPENDENT EXCLUSION ZONES

Kevin P. Carroll, Stephen R. McClaran, Eric L. Nelson, David M. Barnett,
Donald K. Friesen, and Glen N. Williams
Computer Science Department
Texas A&M University
College Station, TX 77843-3112

Abstract

This paper describes an implementation of a path planner suitable for an Autonomous Underwater Vehicle (AUV). The path planner maintains bathymetry, exclusion zone, obstacle, and ocean current databases to facilitate planning. An A* algorithm is used to generate path corridors along great circle routes.

Introduction

A path planner is a necessary component of an AUV. While an AUV is at sea, unforeseen events may violate constraints of a previously planned path. As a result, it may be necessary for the AUV to plan a new path subject to additional constraints. This paper describes the implementation of an A* based path planner for the Autonomous Underwater Vehicle Controller Project at Texas A&M University.

The path planning unit is capable of maintaining a quadtree database of depth information, obstacles, and exclusion zones; verifying a previously planned path; generating a new path between successive goal points; and generating a path to the nearest point of a "safe" region. If a path does not exist within the given constraints, the path planner returns a reason for the failure.

The path planner attempts to find a three–dimensional corridor that does not intersect any nonentry zones. Nonentry zones include areas more shallow than the corridor's maximum depth, land masses, obstacles, and active exclusion zones. Exclusion zones are polygonally defined regions that have time–dependent restricted access. The path planner may insert navigation points to guide the vehicle around nonentry zones or to wait until exclusion zones become inactive before entering. Navigation points may also be inserted by the path planner to vary the speed of the vehicle so that time constraints may be satisfied with minimal resource consumption.

The path planner's real–time performance in open, constricted, and real–world situations will be discussed. Additional factors affecting performance such as the number of different vehicle speeds, the quadtree resolution, and the number of nonentry zones "blocking" the goal will be evaluated and discussed.

Databases

The path planner maintains several databases to facilitate the planning process. These include bathymetry, exclusion zone, obstacle, and ocean current databases.

Bathymetry Database

Five minute resolution bathymetry data for the entire world, and fifteen second data for specific regions, was obtained from the National Geophysical Data Center. The path planner's "map" utility allows users to select an area of interest from a graphical representation of the bathymetry database. Once an area is selected, the utility generates a corresponding quadtree representation subject to depth and granularity constraints specified by the user.

Since the planner need only be concerned with depths as deep as the maximum operating depth of the vehicle, water deeper than this limit may be set to the maximum operating depth. This can decrease the size of the quadtree considerably. Furthermore, the granularity of the quadtree may be controlled by specifying the minimum depth difference necessary to force a quadrant subdivision. Figure 1 shows a

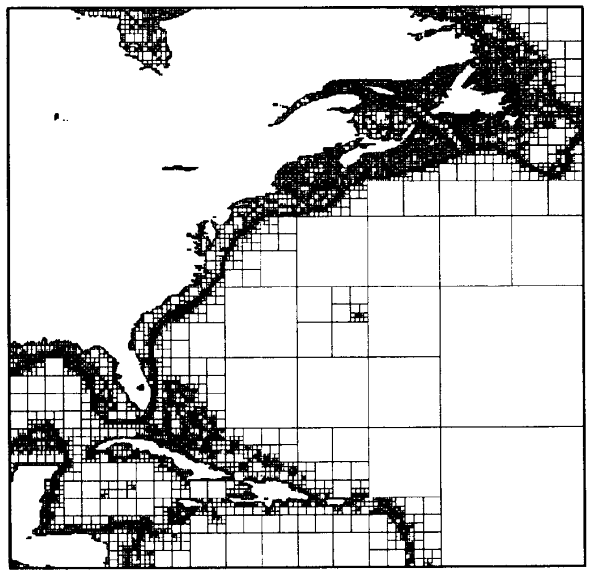

Figure 1. Bathymetry Data Quadtree

graphical representation of quadtree bathymetry data for a selected region.

Ocean Current Database

An ocean current database was created with data obtained from multiple sources of oceanographic literature. The path planner uses this database to estimate current direction and magnitude during the planning process.

Specialty Regions Database

The path planner maintains a list of specialty regions that are "overlayed" on the bathymetry quadtree [2]. Currently, the planner utilizes exclusion zones and obstacles. Exclusion zones are polygonally defined regions with time–dependent restricted access. Obstacles are radially defined regions with restricted access. These regions extend from the ocean surface to the ocean floor. The path planner is capable of updating the specialty regions database at any time.

When the path planner receives an exclusion zone update, the planner creates a quadtree representation of the zone which is then merged

0-7803-0704-6/92$03.00 ©IEEE

with the bathymetry quadtree. Each quadrant of the bathymetry quadtree contains references to each zone that the quadrant intersects. Thus during planning, a path segment that crosses a zone boundary can be easily recognized. A zone is considered "active" during the period between the zone on and zone off times and "inactive" otherwise.

When the planner receives an obstacle update, the point/radius defined obstacle is converted into a polygonally defined region. The obstacle may then be treated the same as an exclusion zone that is always "active."

Path Planning

The path planner implements an A* search with consideration of variable vehicle speeds, ocean currents, and multiple, overlapping, time–dependent exclusion zones.

Features

The path planner has a number of features that make it especially suitable for AUV path planning. These features include monitoring resources, planning with multiple speeds, planning with corridors, planning along great circles, supersampling large quadrants, and refining paths.

Monitoring Resources: The path planner keeps track of mission parameters such as time and fuel resources for each proposed path segment. The expected remaining resource levels are returned after completing the planning process.

Planning with Multiple Speeds: The user may specify a set of speeds for the path planner to use during planning. For each node expansion in the A* search, the cost of travel is computed for each of the allowable speeds. This enables the path planner to choose a speed for each path segment that will satisfy time constraints while minimizing resource consumption.

Planning with Corridors: The path generated by the planner consists of a set of corridors which have both width and depth. Figure 2 shows the corridors that are examined for obstructions during a typical A* search.

Planning along Great Circles: All path segments generated by the path planner lie along great circles [1]. The user may specify the vehicle's maximum allowable distance from the great circle. The planner approximates the great circle route with a set of corridors such that the distance between the great circle and the center–line of each corridor never exceeds the specified limit.

Supersampling Large Quadrants: As the A* search proceeds, each visited node is "expanded" to produce a list of its successors. The successors of node *n*, will be the centroids of all quadrants adjacent to the quadrant containing node *n*. However, if an adjacent quadrant is quite large, travelling to its center may produce an unnecessarily long path. Thus the user may supply a maximum distance allowed between adjacent nodes. Quadrants producing successor distances exceeding this limit are "supersampled." Consequently, a single, large adjacent quadrant may be searched using a more detailed representation. This procedure produces more direct paths through large quadrants.

Refining Paths: The path generated by the A* search will be minimal cost. However, the path is constrained to go through the centers of quadrants (except in the case of supersampling) which may result in a suboptimal path. A post–processing phase examines the path to determine if any nodes can be removed. A node is considered unnecessary if its removal results in a less costly path that remains collision free.

Path Generation

The path planner is given a start taskpoint and a goal taskpoint along with constraints about the path that is to be generated between them (see Table 1). The planner will attempt to generate a minimum–cost,

Table 1
Taskpoint and Path Constraint Information

Taskpoint	Path Constraint
Latitude Longitude Minimum Depth Desired Depth Maximum Depth Minimum Time Desired Time Maximum Time	Minimum Depth Desired Depth Maximum Depth Width Fuel Resources

collision–free, great circle path that meets the time and resource constraints. A path is considered collision–free if it does not cross obstacles, active exclusion zones, or land masses (collectively referred to as nonentry zones).

In the event that a direct path does not exist between the start and goal taskpoints, the planner will insert navigation points to guide the AUV around nonentry zones. If an exclusion zone blocks the desired path, the planner will determine if the AUV can safely travel through the zone (if it is currently inactive), wait for the zone to time out (if it is currently active), or travel around the zone. If the planner chooses to go through a zone, a loiter point will be inserted near the entry point of the zone to ensure that the vehicle remains outside of the zone while it is active. A navigation point will be placed near the exit point of the zone to ensure that the vehicle leaves an inactive zone before it becomes active.

The path planner may not be able to produce a path within the imposed constraints. If this occurs, the planner provides the user with information describing the nature of the failure. Possible reasons for failure include a taskpoint position being outside the bathymetry database boundary, the depth of a taskpoint being too shallow, and the goal being unreachable within either the allocated time or fuel resource constraints.

Sample Path Planning Scenarios

The path planner was implemented in the 'C' programming language on a Sun 4/60 SPARCstation (12 MIPS, 1.5 MFLOPS).

Scenario 1: A simple path exists between the start and goal as shown in Figure 2. The planner generates a simple, great circle path. Quadrants that have been expanded are marked with an "x"; the starting position is labeled with "ckpoint_1"; and the goal position is labeled with "ckpoint_2." This example demonstrates several features.

- Corridor examination: Corridors (as opposed to lines) are examined for obstruction during the A* search. (All examined corridors are drawn.)
- Great circles paths: Long paths are broken into sets of corridors that follow great circle routes. (Paths are drawn as sets of corridors. Note the visible curvature of long paths.)
- Supersampling of large quadrants: Supersampling large quadrants may provide better paths. (The lower left quadrant has been supersampled.)
- Initial path: The initial path is a minimal cost path constrained to go through the centers of quadrants. If quadrants have been supersampled, the points obtained through the supersampling process may also be included in the initial path. (The initial path is drawn as a set of dark, unshaded corridors.)
- Refined path: Since the initial path is constrained to go through the centers of quadrants, it may contain many unnec-

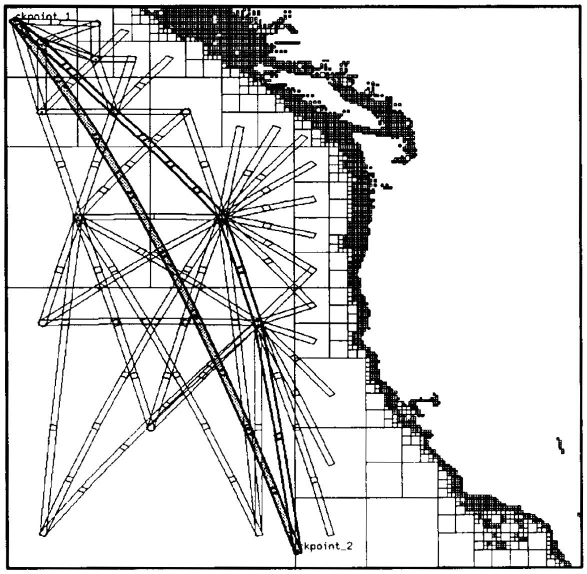

Figure 2. Corridors Displayed During Planning

essary points. These points are removed to produced a refined path. (The refined path is drawn as a set of dark, shaded corridors.)

Scenario 2: The start and goal are positioned such that the terrain obstructs a direct path between them. In this case, the planner must insert several navigation points to lead the vehicle safely around land masses. Figure 3 demonstrates a successful path planned

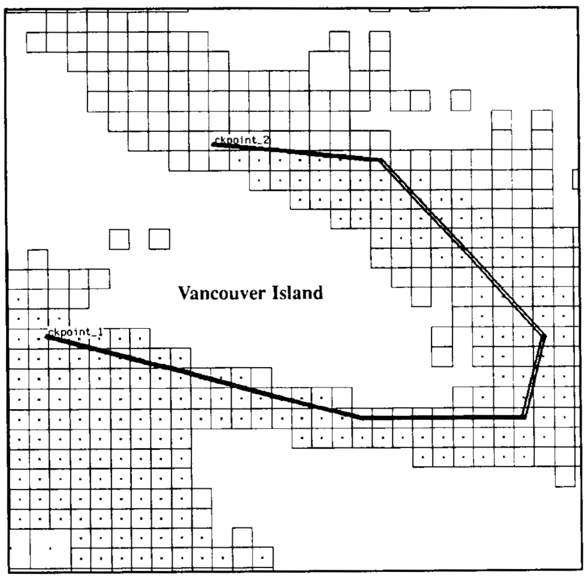

Figure 3. Planning Around Land Masses

through the Strait of Juan de Fuca. Note that a larger path width may produce a different path. (Compare Figures 3 and 4.)

Figure 5 shows a path planned at a depth of 300 meters in the Gulf of St. Lawrence. This path follows the Laurentian Valley due to the depth constraint.

Scenario 3: A field of obstacles obstructs a direct path between the start and goal. (Obstacles may be used to restrict a vehicle's ac-

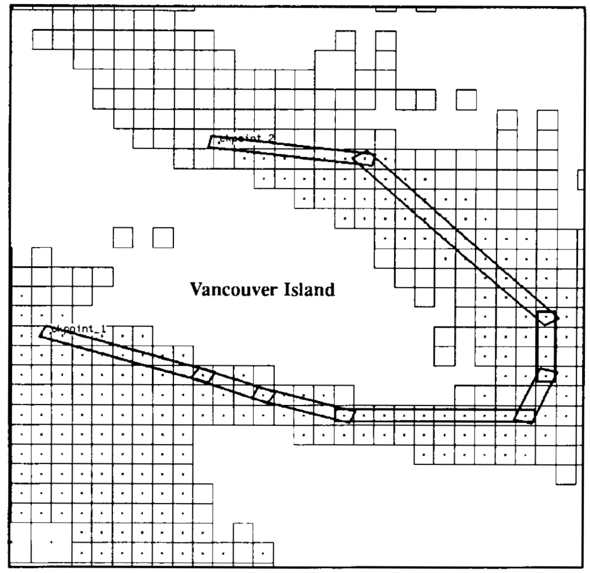

Figure 4. Larger Path Width

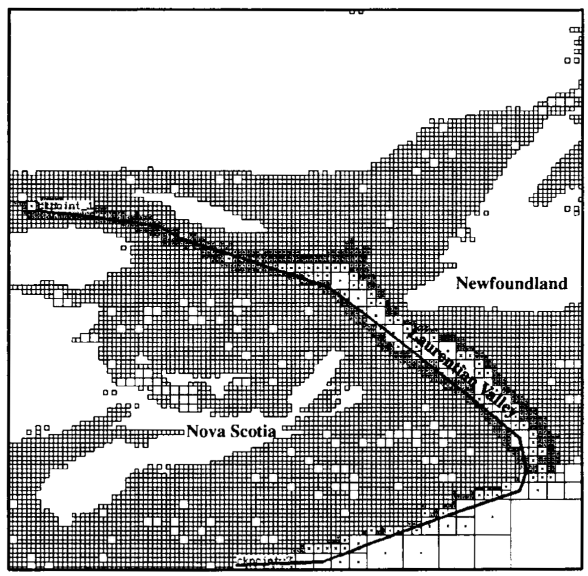

Figure 5. Depth Constraint: 300 meters

cess to an area.) First, obstacles are positioned such that a direct path between the start and goal does not exist. (See Figure 6.) Second, an obstacle is positioned such that the present entrance to the obstacle field is blocked (see Figure 7), and finally, an obstacle is positioned such that the present exit from the obstacle field is obstructed (see Figure 8).

Scenario 4: A direct path between the start and goal is obstructed by a set of "active" exclusion zones. (Exclusion zones may be used to restrict a vehicle's access to an area on a time–dependent basis.) In this example, the zones will be active during the entire journey. Thus, the vehicle must remain free from the zones at all times. Figure 9 demonstrates the planner's ability to plan a path through (around) multiple, complex, active exclusion zones.

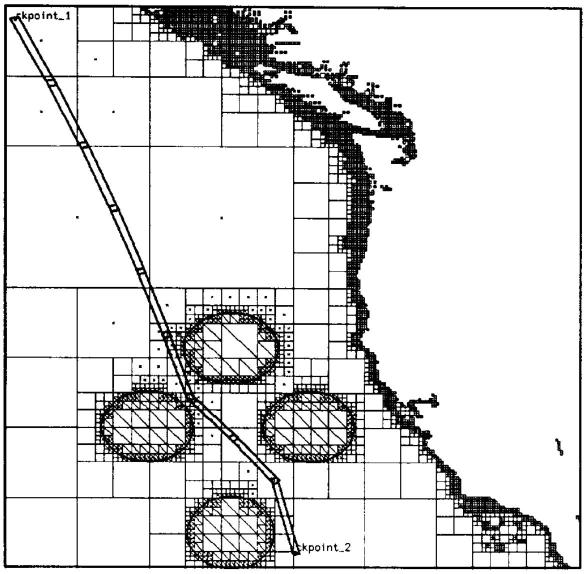

Figure 6. Obstacle Field

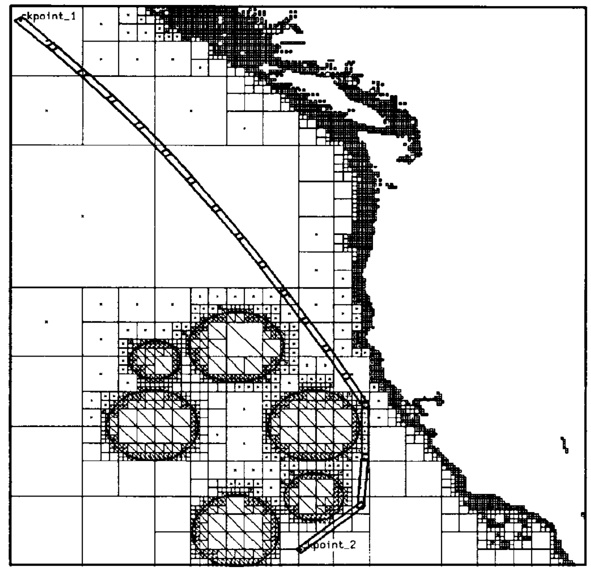

Figure 8. Previous Exit Blocked

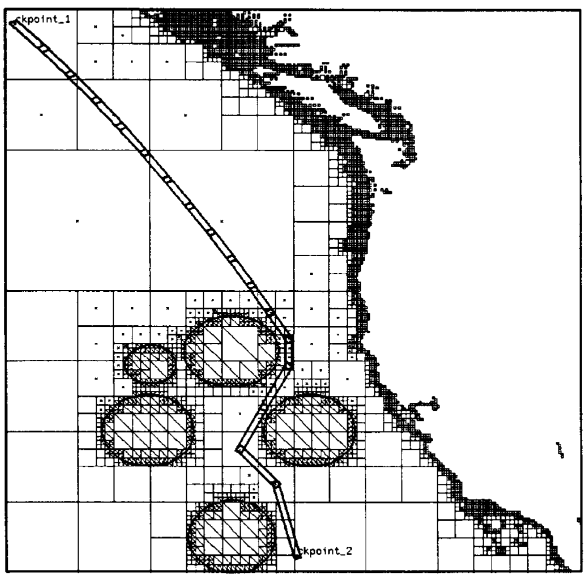

Figure 7. Previous Entrance Blocked

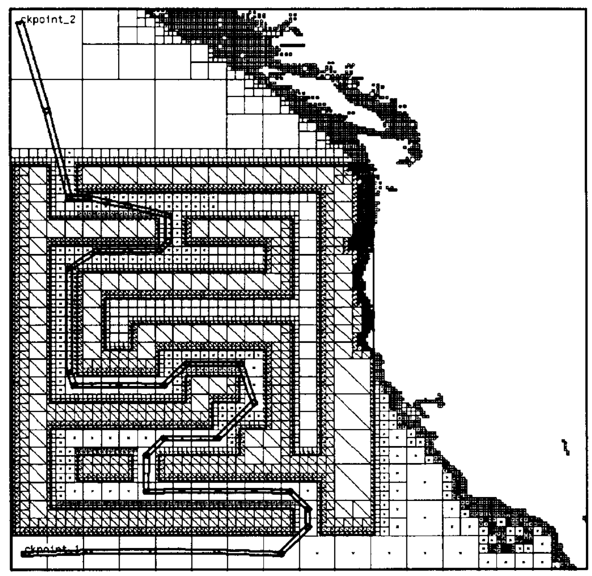

Figure 9. Multiple Complex Exclusion Zones

Scenario 5: **A direct path between the start and goal is obstructed by an exclusion zone that will be both active and inactive during the vehicle's journey (Figure 10).** When the vehicle arrives at the zone, it is not yet active. However, the zone will become active before the vehicle can travel completely through it in the direction of the goal. Consequently, the least-cost path involves traveling through the zone before it becomes active, traveling just outside the zone while it is active, and finally, traveling through it again after it returns to its inactive state. Thus, the planner inserts a navigation point to ensure the vehicle will exit the zone before it becomes active, and inserts a loiter point to ensure the vehicle will not re-enter the zone until it becomes inactive again.

Scenario 6: **The vehicle must wait outside a zone before entering (Figure 11).** The planner determines that the zone cannot be crossed while it is inactive and that traveling around the zone is too costly. The planner therefore inserts a loiter point just outside the zone. Since the planner is considering multiple speeds, the arrival time of the loiter point will be set such that it coincides with the time that the zone becomes inactive. If the vehicle travels at its minimum speed and still arrives before the zone becomes inactive, the loiter point's departure time indicates to the vehicle that it must "loiter" at that point until the zone becomes inactive.

Results

This paper describes the implementation of an A* based path planner for an AUV. The path planner maintains a number of databases to facilitate the planning process. These include bathymetry, exclusion zone, obstacle, and ocean current databases. Features of the path

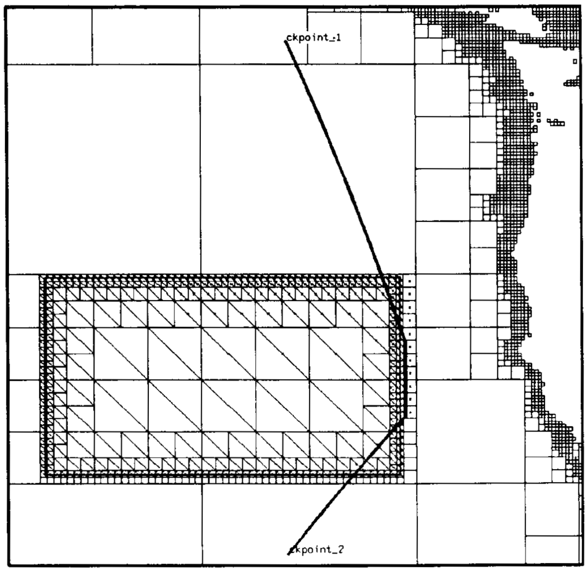

Figure 10. Exclusion Zone Becomes Active During Planning

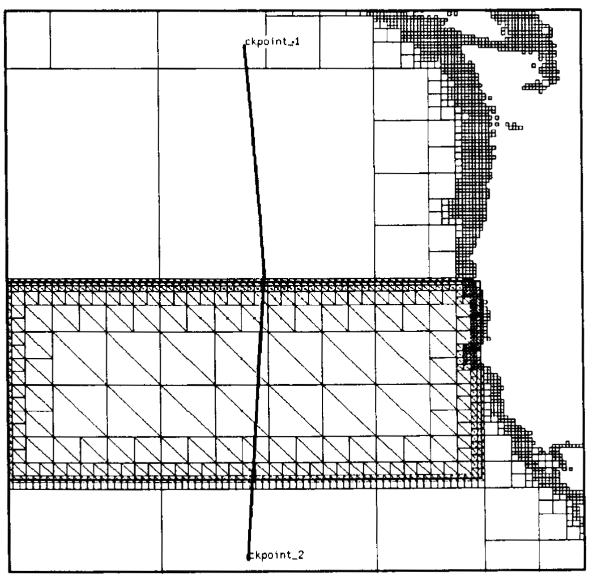

Figure 11. Vehicle Must Loiter Outside of Zone

planner discussed are monitoring resources, planning with multiple speeds, planning with corridors, planning along great circles, supersampling of large quadrants, and refining paths. Numerous examples are provided that show the planner's response to various situations.

The examples demonstrate that the path planner is able to generate simple paths, paths around land masses, paths at specific depths, paths through obstacle fields, and paths through/around multiple complex, time–dependent exclusion zones.

This implementation has been found to be quite efficient in most cases. Tables 2 through 4 show test results from scenario 1, while Table 5 shows the test results from Figure 5. Similar results were obtained in most situations. These tables do, however, show that the distance between ocean current samples and the number of planning speeds significantly impact the planning time. In most cases the increase in planning time was acceptable for the benefits gained.

However, in cases such as Figure 5, where the path must cross hundreds of small, adjacent nodes, the number of combinations of

Table 2

Number of Planning Speeds			1
Great Circle Distance between Start and Goal			1288 nm
Length of Planned Path			1288 nm
Number of Node Expansions			33
Original Number of Navigation Points			5
Number of Navigation Points After Path Refinement			2
Current Sampling Distance (nm)	1	10	100
CPU Time(sec)	9.5	1.9	1.1

Table 3

Number of Planning Speeds			3
Great Circle Distance between Start and Goal			1288 nm
Length of Planned Path			1288 nm
Number of Node Expansions			31
Original Number of Navigation Points			5
Number of Navigation Points After Path Refinement			2
Current Sampling Distance (nm)	1	10	100
CPU Time(sec)	62.1	8.7	3.4

Table 4

Number of Planning Speeds			5
Great Circle Distance between Start and Goal			1288 nm
Length of Planned Path			1288 nm
Number of Node Expansions			23
Original Number of Navigation Points			5
Number of Navigation Points After Path Refinement			2
Current Sampling Distance (nm)	1	10	100
CPU Time(sec)	103.4	14.0	5.0

Table 5

Number of Planning Speeds			1
Great Circle Distance between Start and Goal			431.4 nm
Length of Planned Path			770.5 nm
Number of Node Expansions			1274
Original Number of Navigation Points			110
Number of Navigation Points After Path Refinement			9
Current Sampling Distance (nm)	1	10	100
CPU Time(sec)	114.9	93.3	92.8

path segments with different speeds becomes too great and the planning time outweighs the benefits gained. This problem may be resolved by controlling the number of speed changes allowed during the nodal expansion process. Currently, the planner allows speed changes to occur at each node during expansion. When a node is expanded, path segments are generated for each adjacent neighbor at each planning speed. A possible solution is to allow speed changes to occur only at specific distance intervals. This will make the number of speed changes independent of the quadtree resolution.

Acknowledgements

This work was performed as part of the Autonomous Underwater Vehicle Controller Project at the Texas A&M Research Foundation and was supported by the Office of Naval Technology/Naval Surface Warfare Center (White Oak Detachment) under contract number N60921-88-C-0208.

References

[1] Rogowski, Steve. *Computers for Sea and Sky*. Morris Plains, NJ: Creative Computing Press, 1982.

[2] Shneier, Michael. "Calculations of Geometric Properties Using Quadtrees," *Computer Graphics and Image Processing* 16, 1981, 296–302.