Proceedings of the 1992 IEEE
International Conference on Robotics and Automation
Nice, France - May 1992

Coordination and Control of Multiple Autonomous Vehicles

David L. Brock
David J. Montana
Andrew Z. Ceranowicz
BBN Systems and Technologies Corporation
33 Moulton Street
Cambridge, MA 02138

Abstract

The DARPA SIMNET project allows hundreds of soldiers to train together in a virtual air, land and sea environment through a network of interactive simulators. In addition to the manned simulators, the virtual environment also contains a large number of autonomous vehicles coordinated by an operator at a single workstation. The autonomous vehicles are responsible for lower level path planning, collision avoidance and formation following. In this paper we describe routines for maneuvering among large obstacles, smaller objects and moving vehicles.

1 Introduction

Simulation has become vital for training military forces — both in increasing effectiveness and reducing cost. The DARPA SIMNET project allows hundreds of soldiers to train together in a virtual air, land and sea environment through a network of interactive simulators, figures 1-1. In addition to the manned simulators, the virtual environment is also populated by a large number of autonomous vehicles called Semi-Automated Forces (SAF) which are controlled by an operator at a single workstation, figure 1-2. The SAF operator provides higher-level supervision to the autonomous units, while lower level control such as obstacle avoidance, formation keeping, bridge crossing, road following, etc., is the responsibility of the automated system. The SAF vehicles are essentially simulated mobile robots operating in a complex environment with other autonomous vehicles and manned simulators. Although the autonomous vehicles perform a variety of mission specific tasks, in this paper we address the issues of collision avoidance and formation keeping.

2 Obstacle Avoidance

The SAF vehicles and manned simulators operate together on a simulated terrain. Each simulated terrain is a model of an actual physical location. The terrain database consists of polygonal surfaces that represent the different soil types as well as the various terrain elements such as lakes, rivers, roads, railroads, trees, power lines and buildings. Surface elevations acquired through the Defense Mapping Agency (DMA) and terrain elements from the US Geological Service are used to define areas ranging from 2500 to over 100,000 square kilometers. One basic function of the SAF units is to move from one location to another while navigating among the various obstacles and vehicles.

To simulate the limited visibility available to the vehicles, only the terrain features falling within a disk about the center of a SAF vehicle are accessible to the obstacle avoidance algorithm. Although non-visible terrain features would be available to real vehicles through on-board maps, the current implementation relies on higher-level navigation from the SAF operator.

For the purposes of collision avoidance the terrain elements are grouped as follows: large polygonal obstacles including rivers, lakes and tree canopies, smaller polygons such as tree lines and buildings, and small obstacles such as moving vehicles, trees and towers. Separate collision avoidance algorithms are employed for each of these obstacle classes.

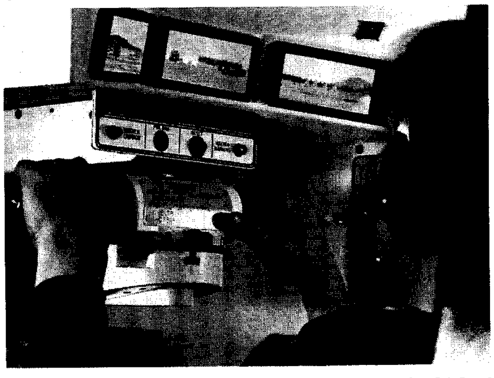

Figure 1-1. A distributed network of hundreds of vehicle simulators allows soldiers to train together in a virtual air, land and sea environment. Computer generated images provide real-time three-dimensional displays of the environment.

0-8186-2720-4/92 $3.00

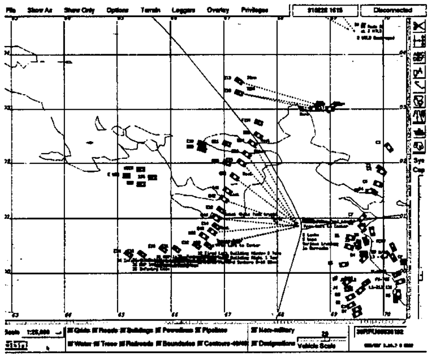

Figure 1-2. The virtual environment is populated by hundreds of manned vehicles and a large number of autonomous vehicles called Semi-Automated Forces (SAF), controlled by an operator at a single workstation.

2.1 Large polygons

Large polygons representing rivers, lakes and tree canopies must be circumnavigated by the autonomous vehicles. We assume the vehicle is represented by a point and that the extent of the obstacle is only known within a radius r about the vehicle. The objective is to move from a start s to a goal point g. We assume the current position and goal are known to the vehicle at all times (this is reasonable given Global Positioning System (GPS) data available to real vehicles).

The problem is thus to navigate from a start to goal among a finite collection of closed curves given a limited visibility radius. Solutions previously proposed include tracing the entire boundary of each obstacle encountered and leaving from the closest boundary point to the goal [Lumelsky 1986]. Other strategies include starting and leaving obstacles only on the points defined by the intersection of the line segment connecting the start to the goal [Lumelsky 1987, 1988]. These algorithms provide provable convergence to the goal (if a path exists), as well as upper and lower bounds on the possible path lengths. The algorithms are initially formulated under the assumption of a zero visibility radius $r = 0$ then relaxed to include some finite value. While the algorithms are shown to converge for any arrangement of obstacles (with certain obvious restrictions: simply closed boundaries, locally finite obstacles and finite obstacle/line intersections), they generate longer paths than necessary for the more commonly encountered obstacles present in our simulation, such as the case when the start and the goal lie outside the convex hull of the intervening obstacles. The objective is thus to employ an algorithm which is both complete and efficient for a majority of cases.

Consider the intuitively simple algorithm A of moving straight toward the goal if possible and, if not, tracing the contour of the obstacle until the vehicle is realigned with the goal, figure 2-1a. This algorithm has been shown to fail even when convergence is possible, see figure 2-1b; however in most cases it will succeed, as well as generate efficient trajectories. For example, if we can close a path around the start, goal and obstacles, and construct a line segment from the goal to the path, then the algorithm will converge, which is proven as follows.

Proposition *If there exists a simple closed path P enclosing the start, goal and obstacles, such that P is star convex about the goal, and a straight line segment L from the goal to P which does not intersect the obstacles, then algorithm A will converge.*

Proof Consider the space S defined by the area within P less the line segment L and the goal g, and consider the mapping from circular to rectangular coordinates, figure 2-2. Thus motion in the radial direction is equivalent to the Pledge Algorithm [Abelson 1980], which is shown to converge in all cases where convergence is possible. The Pledge Algorithm is simply

1. Move toward goal until an obstacle is encountered
2. Turn left and move around the obstacle until then heading is toward the goal and the number of turns is zero.
3. Go to 1. □

Clearly, convergence is assured if the start and goal are outside the convex hull of the obstacle. However, the algorithm does not ensure convergence in the general case. In our simulation the large obstacles represent lakes, tree canopies and rivers, thus cycles about these obstacles would easily be recognized by the operator. We are however developing an algorithm, which through alternating clockwise and counterclockwise decisions span the graph produced by the contact and leave points on the obstacles, and thus guarantee convergence while maintaining efficiency for the simple cases.

The preceding discussion assumes the visibility radius r is zero; however given the finite radius used in our simulation we anticipate the trajectory generated in the case $r = 0$ and move directly toward that point. In addition clockwise or counterclockwise circulation about an obstacle is generally an *a priori* decision; however we choose the direction of travel based on the closest crossing point (bridge or ford site) in the case of a river and in the case of a tree canopy or lake, we choose the direction of minimum angle at the initial contact point, since this generally minimizes travel length for convex objects.

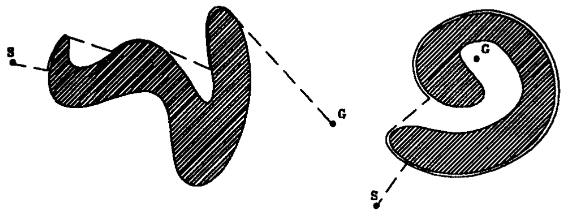

Figure 2-1. Moving directly toward the goal whenever possible generally yields efficient paths, but fails to converge in some cases.

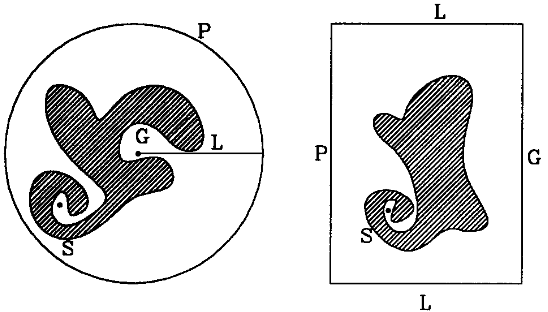

Figure 2-2. Moving directly toward the goal whenever possible is equivalent to the Pledge Algorithm under the appropriate mapping.

2.2 Small polygons

Intermediate goal points generated by the previous algorithm are target points to be achieved while avoiding smaller obstacles. Tree lines and buildings are completely within the visibility radius of the vehicles, hence it is possible to simply move along the visibility graph defined by the configuration space vertices. This is essentially the same algorithm as described previously when the visibility radius tends to infinity. The configuration space obstacles [Lozano-Pérez 1983] are constructed in real-time based on a circular automaton whose radius encloses the vehicle.

2.3 Small obstacles

Ground vehicles must avoid trees, towers and other ground vehicles. On the other hand, air vehicles, such as helicopers and planes, must primarily avoid each other. In any case it is necessary to develop an efficient algorithm to avoid both stationary and moving objects.

Initially a potential field approach was attempted [Khatib 1985]; however this produced undesirable path modifications while moving away from an obstacle. By considering only velocity components into an obstacle, as in the generalized potential field method [Krogh 1984], this problem was corrected, however, the issues of potential minima and stability remain.

Problems with potential field methods including local minima, limit cycles and undesirable oscillations have been identified both theoretically and experimentally by Koren [Koren 1991] and Mans [Mans 1991]. In addition, these methods do not anticipate the trajectories of the obstacles, as estimated from their instantaneous velocities. We have therefore devised a computationally simple yet robust algorithm based on the assumption of constant obstacle velocity.

2.3.1 Constant Velocity Assumption Given obstacle position and velocity, we can estimate future positions, assuming constant velocity, and provide velocity deviations to avoid collision. Consider the following algorithm.

Algorithm Let $\mathbf{p}_1$ and $\mathbf{v}_1$ be the position and velocity of the vehicle and $\mathbf{p}_2$ and $\mathbf{v}_2$ position and velocity of the obstacle.

```
1. Compute relative position
       $\mathbf{p}_r = \mathbf{p}_2 - \mathbf{p}_1$
2. Ignore distant obstacles
       if $\|\mathbf{p}_r\| > p_{max}$ then exit
3. Compute relative velocity
       $\mathbf{v}_r = \mathbf{v}_1 - \mathbf{v}_2$
4. Ignore past obstacles
       if $\mathbf{p}_r \cdot \mathbf{v}_r \leq 0$ then exit
5. Compute projected distance
       $d = \mathbf{v}_r \times \mathbf{p}_r / \|\mathbf{v}_r\|$
6. Sufficient passing distance
       if $|d| > d_{min}$, then exit
7. Compute velocity perpendicular
       $\mathbf{n} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \mathbf{v}_r / \|\mathbf{v}_r\|$
8. Compute projected time of closest point
       $t = \mathbf{p}_r \cdot \mathbf{v}_r / \|\mathbf{v}_r\|^2$
9. Change velocity
       $\Delta \mathbf{v}_1 = -\frac{\mathrm{sgn}(d) G(|d| - d_{min})}{\max(t, t_{min})} \mathbf{n}$
```

Note that if d in step 6 is greater than zero, passing is to the right and if d is less than zero passing is to the left. This algorithm presents a number of advantages including (i) robustness, it avoids all single obstacle collisions conforming to the constant-velocity assumption; (ii) minimal computation, practical implementations usually satisfy one of the exit conditions (steps 2, 4 or 6); (iii) minimum velocity deviation, a component perpendicular to the relative velocity achieves maximum passing distance for minimum change; and (iv) smooth paths, proper parameter selection ensures gradual trajectory changes.

The avoidance algorithm has several parameters which effect its performance including the maximum relative position p_{max}, the minimum passing distance d_{min}, minimum reaction time t_{min} and gain parameter G. A smaller value of p_{max} corresponds to less computation, increased reliability of projected positions and a more narrow scope of obstacle consideration. A larger value of p_{max} allows more reaction time hence smoother trajectories, but runs the risk of multiple vehicle consideration and aberrant behavior. The minimum passing distance d_{min} defines the minimum distance for safe passage. Thus smaller values correspond to smoother trajectories and less deviation for the intended routes, though increases the chance for collision due to errors in the estimated states. The minimum passing distance must of course compensate for the dimensions of the vehicles. The minimum reaction time t_{max} limits the maximum acceleration and is selected based on performance of the vehicles. The gain parameter G also affects the reaction time of the vehicles and the smoothness of the trajectories.

2.3.2 Deviations from the constant-velocity assumption Deviations from the constant-velocity assumption occur in our system. First, the moving obstacles are other vehicles which are also performing obstacle avoidance and thus violating the constant-velocity assumption. Assuming exact sensing, the velocity estimates of both vehicles are identical. Hence, velocity corrections will be $\mathbf{\Delta v_1} = -\mathbf{\Delta v_2}$, and the the avoidance of the second vehicle reinforces the first. However, given errors in the velocity estimate, in the case when the projected passing distance is very small, it is possible that the vehicles will disagree on the sign of d. Hence, $\mathbf{\Delta v_1} \approx -\mathbf{\Delta v_2}$, and their avoidance efforts cancel each other out. Two non-exclusive solutions to this problem are (1) to incorporate a condition to slow the vehicles until an avoidance direction is mutually decided upon, and (2) to vary slightly the avoidance parameters among individual vehicles so that the two terms do not sum to zero.

Second, vehicles moving in formation must selectively perform collision avoidance. While collision avoidance is necessary during formation or direction changes, it is inappropriate for vehicles rejoining their unit to avoid the ones they are suppose to follow. Therefore collision avoidance is temporarily disabled for lagging vehicles overtaking their units.

Third, vehicles reaching goal points will either stop or change direction. In either case their velocities will change, thus violating the constant-velocity assumption. Hence, the algorithm was modified between steps 6 and 7 to only avoid collisions prior to the current goal point. For potential collisions further in the future, we wait until we have the information required for accurate projection before performing avoidance.

Fourth, since the algorithm assumes single vehicle interaction, simultaneous intersection between multiple vehicles can cause aberrant behavior. Proper selection of p_{max} and d_{min} can reduce this occurrence, though simultaneous interactions are not addressed in this implementation.

Fifth, vehicles can gradually change direction. Ground vehicles and helicopters tend to make rapid turns at specific goal points, the consequences of which are discussed above. However, planes usually make gradual, wide turns. Such turns, which occupy a large fraction of the planes' flight time, cannot be compensated for by minor modifications to the above algorithm but rather require a completely different algorithm, which we now describe.

2.3.3 Arc-line avoidance The movement of a plane through a turn can be modeled approximately as a constant-speed motion along a circular arc. We assume that each plane can accurately estimate not only the velocity of other planes but also their roll and hence their acceleration. (Assuming constant altitude, acceleration equals gravity times the tangent of the roll.) Hence, each plane can project the motion of itself and other planes through their current turns. Planes can also anticipate the direction they will be heading when they end their current turn and thus can also anticipate this direction for other planes in their units, which will generally be heading the same direction. Therefore, they can project the time at which a turn ends and constant velocity motion resumes. Since we cannot accurately project future turns, our best model of the future trajectory of a plane is as motion along an arc (which has length zero when the plane is not currently turning) followed by linear motion. We call this "arc-line" motion.

We now give an algorithm for performing avoidance based on the avoider and obstacle executing arc-line motion. One constraint is that the paths of the planes are carefully planned and should not be varied. Therefore, we perform avoidance by altering the speeds of the planes without changing the paths. This works in all but the highly unusual case when two planes of the same force fly straight at each other.

The algorithm is as follows:

1. Find the intersection of the paths. (There will be zero, one or two intersection points.) For the i^{th} intersection, there is a pair of times (t_{ai}, t_{bi}), which are the times of the intersection point along the paths of the vehicles a and b.

2. If there are no intersection points such that $\frac{t_{ai}+t_{bi}}{2} < t_{max}$ and $|t_{ai} - t_{bi}| < \Delta t_{min}$, then exit. (Here, t_{max} is a parameter determining how far ahead in time to look, and Δt_{min} is the desired

minimum difference between times at the intersection.)

3. If there are two intersection points that satisfy the above conditions, choose one based on the following criteria:
   - if $t_{a1} \leq t_{a2}$ and $t_{b1} \leq t_{b2}$, choose the first point
   - if $t_{a2} \leq t_{a1}$ and $t_{b2} \leq t_{b1}$, choose the second point
   - if $|t_{a1} - t_{b1}| \leq |t_{a2} - t_{b2}|$, choose the first point
   - otherwise, choose the second point
4. Change the speed s_a by the quantity

$$\Delta s_a = \frac{A_1 \mathrm{sgn}(t_b - t_a)(\Delta t_{min} - |t_a - t_b|)s_a}{\max(\frac{t_a+t_b}{2}, A_2)}$$

Note that if both planes use the same velocity and roll estimates, the change in speed for plane b avoiding plane a satisfies $\Delta s_b / s_b = \Delta s_a / s_a$, and hence the avoidance efforts reinforce each other.

3 Formation following

In addition to the control of individual vehicles, it is desirable to coordinate their motion within formations and groups of formations. These formations are composed of autonomous units or aggregates of manned simulators and autonomous vehicles. Each formation has a leader which is usually, though not necessarily, located near the head of the group. Each member of the group has a station point representing the desired position of the vehicle relative to the leader. The objective is to stay approximately in formation while maneuvering among the obstacles and other vehicles. If the trailing vehicles were to simply move toward their assigned positions, they would fail to anticipate impending collision, as well as undergo large, undesirable trajectories in response to small changes in their leader's direction, figure 3-1a. Therefore we have developed the following algorithm which maintains more uniform motion.

Algorithm Let $\mathbf{p_l}$, $\mathbf{d_l}$ and $\mathbf{v_l}$ be the position, direction and velocity of the leader and $\mathbf{p_f}$ and $\mathbf{v_f}$ the position and velocity of the follower.

1. Compute station point

$$\mathbf{s} = \mathrm{T}(\mathbf{d_l}, \mathbf{p_l}),$$

where T is the homogeneous transform based on the direction $\mathbf{d_l}$ and the point $\mathbf{p_l}$.

2. Compute projection magnitude

$$p_{mag} = p_s \|\mathbf{v_l}\| + p_o,$$

where p_s is a scale factor and p_o is an offset.

3. Compute projection point

$$\mathbf{p} = p_{mag}\mathbf{d_l} + \mathbf{s}$$

4. Compute error

$$\mathbf{e} = \mathbf{p} - \mathbf{p_f}$$

5. Move only in the direction of leader or when far from the group

if $\|\mathbf{e}\| < e_{max}$ and $\mathbf{e} \cdot \mathbf{d_l} < 0$, then $\mathbf{v_f} = 0$ exit

6. Compute velocity

$$\mathbf{v_f} = \frac{\|\mathbf{v_l}\|\|\mathbf{e}\|}{p_{mag}}\mathbf{d_l}$$

The idea is to project the station point forward in the direction of the leader's intended motion. Motion is then allowed only if the follower is either a significant distance from the group or its velocity is in the same direction as the leader's. By scaling the velocity proportional to the distance from the projected point, the vehicle will either speed up or slow down to fall in line with the formation. Figure 3-1b shows the resulting trajectories for the following vehicles.

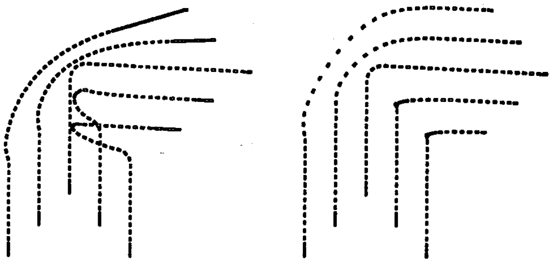

Figure 3-1. Moving directly to the proper station points causes unnecessary motion within the formation. Projecting station points in the direction of the intended unit direction smoothes trajectories and reduces unnecessary motion.

4 Implementation

The vehicle control algorithms and dynamics are simulated on a network of MIPS 2000 computers. Up to one thousand vehicles including SAF vehicles and manned simulators have simultaneously operated in the same virtual environment. For the ground vehicles the dynamic model includes acceleration and braking limits based on vehicle type, surface gradient and soil type. For air vehicles a full dynamic model is employed which incorporates inertia, lift, drag and thrust effects based on vehicle attitude and air speed.

Control of the SAF vehicles is supervised on a network of MIPS 3000 Magnum Workstations and Symbolics 3600s. SAF vehicle obstacle avoidance and formation keeping are illustrated in figures 4-1, 4-2 and 4-3.

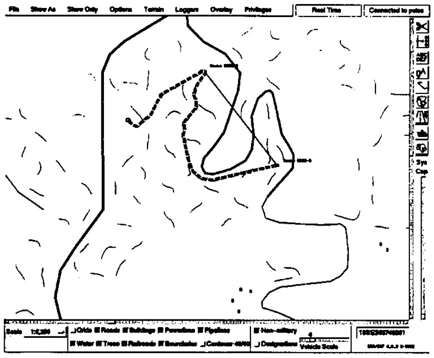

Figure 4-1. A single ground vehicle is commanded to follow a route defined by a line segment. The vehicle moves from its current position to the start and then to the end of the route while avoiding the river and tree lines. The dashed line represents the movement of the vehicle, which is moving generally from left to right. The line spanning the screen to the left is a road and the more convoluted line near the center is a river. The shorter lines represent tree lines and the small rectangular boxes near the bottom center are buildings.

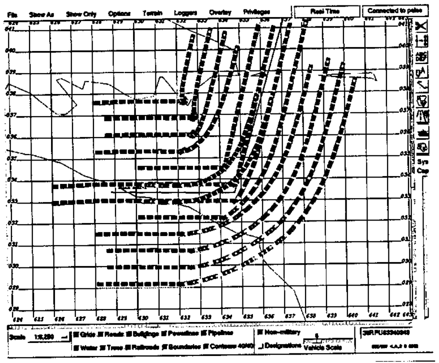

Figure 4-2. A formation composed of 15 vehicles is required to follow a a route with a 80° bend. By anticipating the future position of the unit, the followers can adjust speed and direction to maintain the uniformity of the formation.

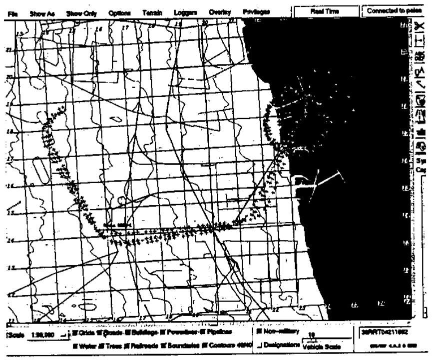

Figure 4-3. A formation of four air vehicles are given a simple route. The positions of the vehicles within the formation are reassigned to produce trajectories of equal length through the turn. The arc-line avoidance algorithm ensure the vehicles avoid collision while in the turns and circling the end point of the route.

5 Discussion and Conclusion

The algorithms presented here have generally performed well; however, multiple vehicle intersection and simultaneous interaction of vehicles and obstacles have still produced collisions. Algorithms are now being developed to address simultaneous vehicle and obstacle interaction to yield collision free motion. In every case, better models and estimates of projected position have produced superior results.

Higher-level planning and mission specific tasks have been incorporated into the autonomous vehicles; however, lower level reactive behavior is still necessary to allow immediate response to sudden changes from either the manned simulators or the SAF operators.

Acknowlegements

This work was supported in part by DARPA SIMNET programs under contract numbers MDA972-90-C-0061 and MDA972-89-C-0060 and in part by USA ARMY ETL contracts for the ODIN program DACA76091-C-0005 and DACA76091-C-0006.

References

Abelson, H. and diSessa, A., "Turtle Geometry," MIT Press, Cambridge, MA, 1980.

Khatib, O., "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots," *Proceedings of the IEEE International Conference on Robotics and Automation,* St. Louis, MS, March 1985 , p. 500-505.

Koren, Yoram and Borenstein, Johann, "Potential Field Methods and Their Inherent Limitations for Mobile Robot Navigation," *Proceedings of the IEEE International Conference on Robotics and Automation,* Sacramento, CA, April 1991 , p. 1398-1404.

Krogh, B.H., "A Generalized Potential Field Approach to Obstacle Avoidance Control," *International Robotics Research Conference,* Bethlehem, PA, August 1984.

Lozano-Pérez, T, "Spatial planning: A configuration space approach," *IEEE Transactions on Computing,"* 32 1983, p. 108-120.

Lumelsky, V. and Stepanov, A., "Dynamic Path Planning for Mobile Automaton with Limited Information on the Environment," *IEEE Transactions on Automatic Control,* Vol. AC-31, No. 11, November 1986.

———, "Path-Planning Strategies for a Point Mobile Automaton Moving Amidst Unknown Obstacles of Arbitrary Shape," *Algorithmica,* 3(4) 1987.

Manz, Allan, Liscano, Ramiro and Green, David, "A Comparison of Real-time Obstacle Avoidance Methods for Mobile Robots," *Proceedings of the International Symposium on Experimental Robotics,* Toulouse, France 1991.