Preprints of the Third International
Symposium on Experimental Robotics
Kyoto, Japan Oct. 28-30, 1993.

A Unified Approach to Planning, Sensing and Navigation for Mobile Robots

Alexander Zelinsky[1] and Shin'ichi Yuta[2]

Intelligent Machine Behaviour Section[1]
Electrotechnical Laboratory
Tsukuba 305, Japan

Intelligent Robotics Laboratory[2]
Tsukuba University
Tsukuba 305, Japan

Abstract

Much of the focus of the research effort in path planning for mobile robots has centred on the problem of finding a path from a start location to a goal location, while minimising one or more parameters such as length of path, energy consumption or journey time. Most of the experimental results reported in the literature have centred on simulation results. Only a small subset of the reported results have been implemented on real autonomous mobile robots. It is the goal of our research program to develop path planning algorithms whose correctness and robustness can be tested and verified by implementation on our experimental, self-contained and autonomous mobile robot - the Yamabico [Yuta *et.al.* 91].

1. Introduction

Much of the focus of the research effort in path planning for mobile robots has centred on the problem of finding a path from a start location to a goal location, while minimising one or more parameters such as length of path, energy consumption or journey time. Most of the experimental results reported in the literature have centred on simulation results. Only a small subset of the reported results have been implemented on real autonomous mobile robots. It is the goal of our research program to develop path planning algorithms whose correctness and robustness can be tested and verified by implementation on our experimental, self-contained and autonomous mobile robot - the Yamabico which was reported by Yuta *et.al.* in [1].

The primary drawbacks of most reported path planning methods is that they operate in known environments and only find the shortest path between the start and goal positions. However completely known environments rarely exist in practise, for example a map of a building or factory can be readily obtained and used to perform path planning for a mobile robot. However it is impossible to guarantee that the map is accurate, since it does not record the presence of people or obstacles such as boxes left in passage ways. Environments are in reality not static, they are dynamic i.e. constantly changing. This introduces uncertainty into the problem of path planning. To be useful an autonomous mobile must be able to handle the unknown. Path planning in unknown environments as studied by Chatila [2], Crowley [3], Moravec [4] and Thorpe [5], has taken the approach that the environment is completely unknown and the mobile robot must incrementally map the environment using the robot's onboard sensors. It can be argued that this is the correct approach to the problem. However, it has not produced significant results. This is due to the limitations of the current generation of sensing technology. Current sensors either have serious limitations in their accuracy e.g. ultrasonic sensors or they require enormous computation and memory needs e.g. vision sensors. Building accurate maps with current sensing technology is a tall order. At present there are no methods available which allow a mobile robot to construct an accurate representation of the environment quickly using onboard processing.

Planning paths that only minimise the distance to a goal are usually unsuitable for implementation on an actual mobile robot. The safety of the robot is important particularly since there are uncertainties in the sensor data, such as the exact shape and position of obstacles. This problem is compounded by the uncertainty in the dynamic control of a robot i.e. the precise position of the robot is not always known by the robot's control system. Thus both minimum distance to a goal and safety of the robot need to be considered simultaneously during path planning. This problem could be solved by adding an external localisation system which would maintain an accurate estimate of robot's position. This involves adding beacons or artificial landmarks to the environment. This adds unnecessary artificial structure to the environment. One goal of our research is to construct mobile robot navigation schemes that operate in unstructured indoor environments.

In this paper we present an approach to the implementation of a navigation system on a mobile robot using a unified approach to planning. A unified approach uses a combination of known and unknown environment path planning techniques. The environment in which a mobile robot must operate in is usually well known e.g. a factory layout. The plan of the environment e.g the design drawings of a building, can be supplied to the robot's path planner. The robot will plan a global path to the goal through the known environment using a vertex graph model and the A* algorithm. However, this plan is insufficient to allow a robot to execute the path relying solely on dead reckoning. Therefore, the global path planner must also consider which landmarks in the environment should be tracked for position estimation and the sensing positions to optimally observe the selected landmarks. Hence, the model of the environment must not only include geometric data, but it must also include the types of surfaces and the characteristic features of the landmarks which can be measured by sensors. Once a plan for the robot's mission has been constructed, the robot executes the planned path while observing specific landmarks.

If the robot unexpectedly encounters an obstacle which interferes with the execution of the planned mission, the robot must avoid the obstacle. In the literature there have been various local obstacle avoidance procedures proposed. A popular approach has been to use potential fields that were pioneered by Khatib [6]. The problems of using potential fields is well documented in the literature by Koren *et. al.* [7]. The worst problem affecting potential fields is the local minima problem. The other approach to local obstacle avoidance is heuristic navigation. Heuristic navigation as studied by Borenstein *et. al.* [8] and Chattergy [9] guide the robot to the goal by using rules to decide, based on local sensor information, which path of those available is "best". Heuristic path planners can solve a wide variety of path planning problems, but a problem can always be found where a particular heuristic strategy fails. The appealing feature of heuristic obstacle avoidance procedures is that they require little processing and can produce rapid "reactive" behaviours. However, in these approaches since no planning is done the resulting execution paths may not be optimal and may lead to a failure of the mission.

In this paper we present a local obstacle avoidance scheme called "reactive planning". In this scheme we use the "path transform" which was developed by the first author [10]. The path transform is a grid based planning method. This method considers both minimum distance to a goal and the safety of the robot. The path transform can be regarded as a "numeric potential field" which has the desirable properties of potential field path planners without suffering the local minima problem. Once the robot detects an obstacle with its sensors, it builds a local map of the environment which includes an accurate representation of the freshly sensed obstacle. Making a grid based map using sensor data is a straight forward procedure. Using a grid based planner over a small area is computationally inexpensive. The robot then plans a path which avoids the obstacle and allows the robot to rejoin the originally planned global path. The reactive planning approach yields a fast, efficient and robust local obstacle avoidance method.

This paper will present the research results of implementing the unified planning approach to mobile robot navigation on the Yamabico. Video results showing experimental results of planning global paths and sensing points, execution of the global path, and reactive planning will be shown.

The problem of planning a path of complete coverage of an environment by a mobile robot has received little research attention. A path of complete coverage is a planned path in which a robot sweeps all areas of free space in an environment in a systematic and efficient manner. Possible applications for paths of complete coverage include autonomous vacuum cleaners, lawn mowers, security robots, land mine detectors etc. This paper will also present a solution to this problem based upon an extension to the "path transform". This solution is in keeping with unified approach to path planning. This paper will present the experimental results showing planning and execution of complete coverage paths.

The Yamabico robot is equipped with two driving wheels mounted on a central axis and has optical and ultrasonic range sensors (see Figure 1). The ultrasonic sensors are used to track landmarks and detect the presence of unforeseen obstacles, thus preventing collisions. The optical range sensor is also used to detect unexpected obstacles. However, it's primary purpose is to accurately sense the obstacles, and feed this data to the reactive planner. Each hardware function of the robot is modularised to run on a single board computer. A master module implements decision making and coordinates the actions of each hardware module via a shared communications bus.The master module is programmed to control the operations of the robot with the sensor based ROBOL/0 language developed in your laboratory by Suzuki *et. al.* [11]. Software on the master module runs under our specially developed multi processing MOSRA operating system.

Figure 1. The Yamabico Robot

2. Path and Sensing Planning

The Yamabico mobile robot navigates by using a a map of the operational environment. The robot's map is a 2 dimensional linked list of vectors. The vectors do not only represent the geometry of the environment, they also record which obstacle surfaces can be sensed by the robot's sensors. For example, the 2 dimensional projection onto the floor plane of a table will result in a rectangular shape. The robot should plan a path that avoids the table, however it should not use the table to perform landmark tracking with its ultrasonic sensors. For a real autonomous robot to execute a planned path it must able to correct it's position during path execution time. However, the robot must plan where it should correct its estimated position. Path planning is done by building a visibility graph and then performing a graph search. Refer to Figure 2 for an example of a vector map of an environment and the corresponding visibility graph which is used for path planning (the heavy line shows the solution path). Details of path planning with the consideration of the cost of is presented by Nagatani *et. al.* in [12].

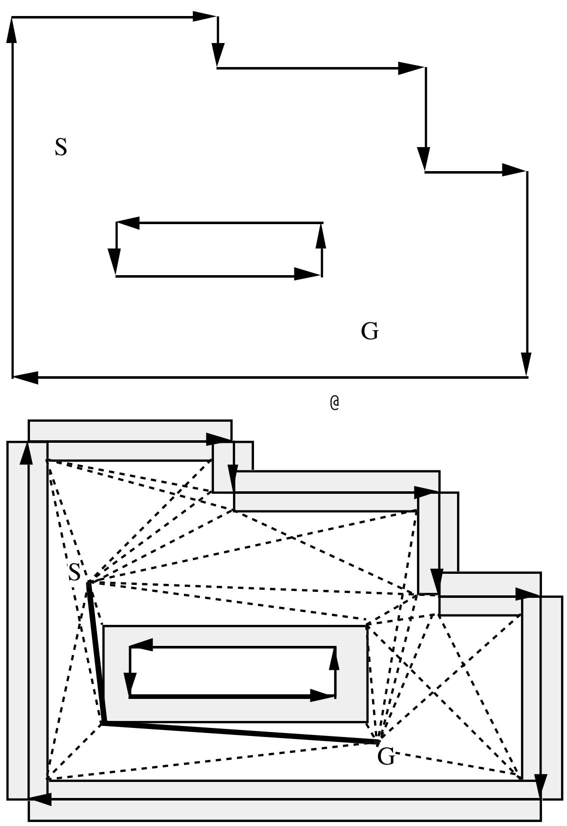

Figure 2.Vector Map and Path Planning Visibility Graph

Once a path has been planned the Yamabico mobile robot executes the path. The robot monitors the execution of the path by using dead reckoning and observing landmarks with its ultrasonic sensors. The robot uses the ultrasonic sensors to measure distances to landmarks, such as walls. The robot's vector map is used to look up the expected range measurement to the landmark. This measurement is compared against the actual range measurement to the landmark. After the sighting of the landmark the estimate of the robot's position in the environment is adjusted for errors. Refer to Figure 3 for an example of the robot monitoring the execution of a path from S to G. In this example the robot attempts to execute the path shown with the broken lines. The actual path executed by the robot is shown with the unbroken line. The robot senses for landmarks and corrects the robot's position at the tracking points T.

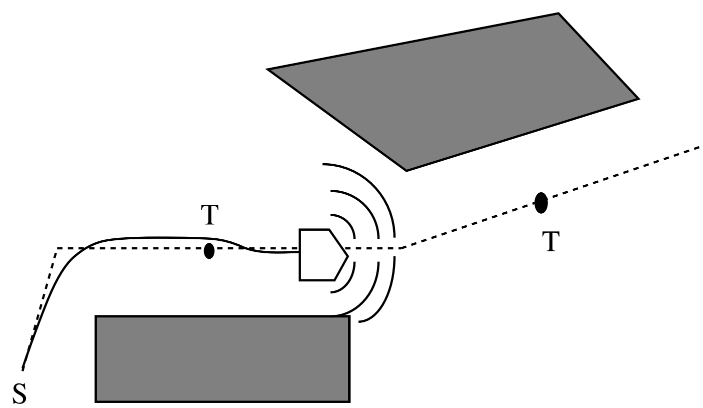

Figure 3. Landmark Tracking during Path Execution

3. Reactive Planning

A novel approach to the problem of findpath using distance transforms (Refer Figure 4) was presented by Jarvis *et. al.* [13]. This approach considers path planning to be finding all paths from the goal location back to the start location. This planning approach propagates a distance wave front through all free space from the goal. From any starting point in the environment the shortest path to the goal is traced by following path of steepest descent. If no downhill path exists then we can conclude the goal is unreachable. Despite the disadvantage of computing all paths from the goal location the distance transforms have a number of advantageous properties. These include; supporting multiple robots and multiple goals. For example paths which favour or avoid certain areas in the environment. The

Jarvis *et. al.* approach suffers the problem of moving too close to obstacles while it minimises the length of the path to the goal.

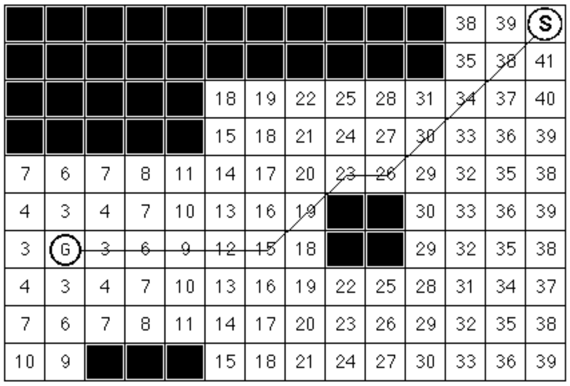

Figure 4. Distance Transform

Using the path transform approach presented. by Zelinsky in [10] instead of propagating a distance wave front from the goal, another wave front is propagated which is a combination of the distance from the goal together with a measure of the discomfort of moving near obstacles. This has the effect of producing a distance transform which has the properties of potential fields *without* local minima.

The distance transform is extended to include safety from obstacles information in the following way. Firstly, the distance transform is inverted into an "obstacle transform" where the obstacle cells become the goals. The resulting transformation yields the minimum distance for each free cell from the boundary of the closest obstacle cell. Lastly, a second distance transform is generated through free space from the goal location using a new cost function. This cost function is refered to as the "path transform" (*PT*). The path transform for a cell *c* is defined as:

$$PT(c) = \min_{p \in P} \left(\text{length}(p) + \sum_{c_i \in p} \alpha \text{obstacle}(c_i) \right)$$

where *P* is the set of all possible paths from the cell *c* to the goal, and $p \in P$ i.e. a single path to the goal. The function *length(p)* is the length of path *p* to the goal. The function *obstacle(c)* is a cost function generated using the values of the obstacle transform. It represents the degree of discomfort the nearest obstacle exerts on a cell *c*. The weight α is a constant ≥ 0 which determines by how strongly the path transform will avoid obstacles.

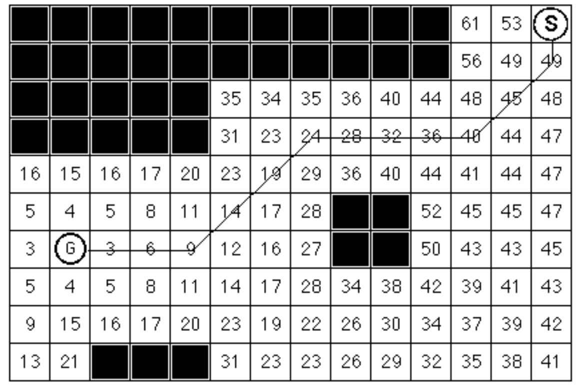

Figure 5. Path Transform with $\alpha = 0$

The value stored in the path transform of each cell represents the minimum propogated cost to the goal. The propogated cost is a weighted sum of the distance to the goal and the cost of approaching obstacles. Since the path transform determines the costs of all paths to the goal from each cell it therefore does not contain local minima. The path to the goal is found by tracing the path of steepest descent. Examples of the path transform with different values of α are shown in Figures 5 and 6. Figure 5 shows the path transform with $\alpha = 0.5$, resulting in a solution path that takes a safer path to the goal. Figure 6 shows the path transform with $\alpha = 1.0$, resulting in a solution path which takes the safest path to the goal.

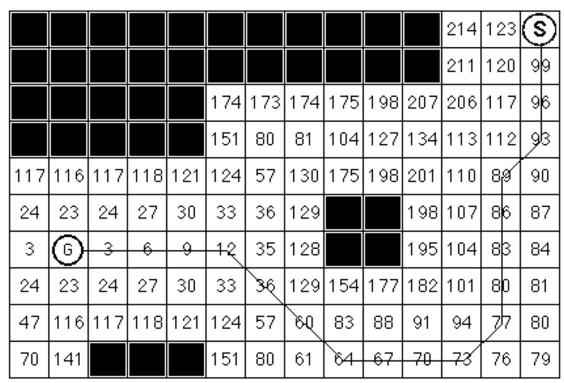

Figure 6. Path Transform with $\alpha = 1.0$

The cost function *obstacle(c)* that was found to be most effective was a cubic function that ranged from zero at a fixed distance, set by the user, to a maximum cost at zero distance. Such a cost function has the advantages of: creating a saddle between the repulsion peaks of neighbouring obstacles, ease of computation, and has its effects bounded in a local area.

There has been other work which is similar to the path transform reported by Barraquand and Latombe [14]. Barraquand and Latombe construct a numeric potential field using a grid representation. This is done in three (3) steps. Firstly, they compute an "obstacle transform" of the free space cells, from which they extract a "distance skeleton" which represents a digitised Voronoi diagram of the free space in the environment. Joining the highest values in the obstacle transform with line segments will yield a distance skeleton. They then connect the goal cell to the distance skeleton. Secondly, they compute a distance transform from the goal cell to all cells that are members of the distance skeleton. Thirdly, they compute another distance transform from the member cells of the distance skeleton to all the remaining free space cells in the environment. The end result is a "numeric" Voronoi diagram of the environment. This method is more complicated to compute than the path transform. Also, since it maximises clearance from obstacles it can divert the solution too far from the shortest path. Another drawback is because Barraquand and Latombe do not specifically consider clearance information to generate their numeric potential, their method can guide the robot through narrow free space channels that are close to the goal there by endangering the robot. They counter this problem by removing channels or branches of the Voronoi diagram that are too narrow. In effect they grow selected obstacles, and thereby lose the completeness of their solution. The path transform does not suffer this drawback.

The Yamabico mobile robot navigates by following the paths planned by the global planner, and uses the path transform to deal with unexpected obstacles. Once a path has been planned the Yamabico mobile robot executes the path. The robot monitors the execution of the path by using dead reckoning and observing landmarks, such as walls with its ultrasonic sensors. If an unexpected obstacle is encountered along the planned path trajectory by the ultrasonic or laser sensing systems, the Yamabico stops. The robot then uses its laser range finding system to determine the shape and location of the obstacle. The robot scans the environment to the left and right of the obstacle. A grid map covering an area of 4.0m x 3.0m with a grid cell size of 0.20m is projected onto the environment, the map is centred on the mobile robot. The relevant portions of the vector map together with the laser sensor data are projected onto the grip map. A goal position is placed 1.5m behind the sensed obstacle along the original planned trajectory. The Yamabico robot is almost cylindrical in shape and has a diameter of approximately 0.36m. Collision testing can therefore be efficiently implemented by growing the obstacles in the environment by 0.20m. Refer to Figure 7 for an example of reactive planning using the path transform. The first diagram shows the environment, the planned path and the unexpected obstacle. The second diagram shows the local grid map, with the sensor data and environment map data, and the obstacle avoidance path.

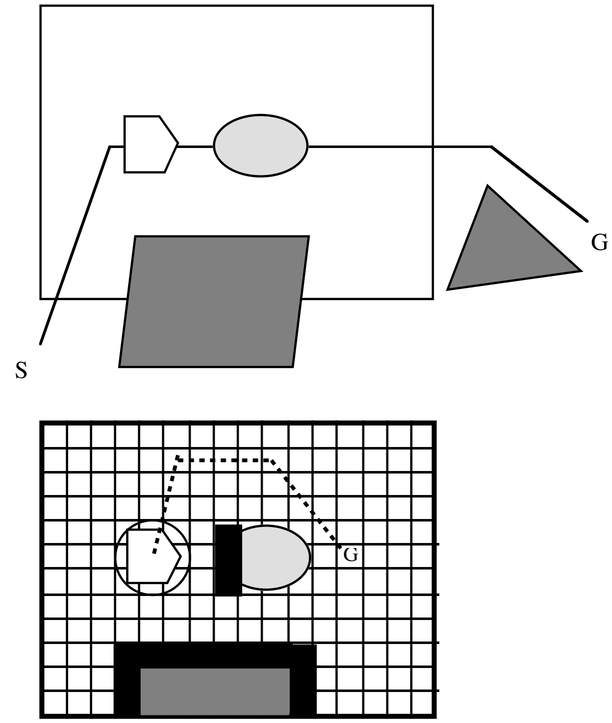

Figure 7. Reactive Planning

The proposed collision avoidance scheme described above was initially implemented on AMROS simulator developed in our laboratory by Kimoto *et. al.* [14]. This simulator models the kimematics of the the locomotion system, and has realistic models of the robot's ultrasonic and laser sensors. The simulator is programmed in the exactly same manner as the Yamabico robot. The simulator proved to be a valuable debugging tool. Collision avoidance software could be tested on the simulator before it was used on the robot. Figure 5 shows an example of output from the AMROS Simulator of the Yamabico navigating down an obstacle strewn corridor. The corridor is 13m long and 2.5m wide. The robot's navigation task is to move down the corridor, as each unexpected obstacle is encountered the robot makes a local map, and places a goal behind the obstacle further down the corridor. Using this strategy the robot is able to avoid three obstacles, back its way out a dead-end, and deduce that the corridor is blocked off, and abandons the mission. Figure 8 shows output from the AMROS simulator.

Start

Figure 8. Simulation Results

Finally the reactive planner was implemented and tested on the Yamabico. The Yamabico was successfully able to navigate down a corridor 20m long containing numerous unknown obstacles. Experimental results showed that once the robot detected an unexpected obstacle with its perception sensors, a 100 degree laser scan of the environment in front of the robot (50 degrees left and right of the robot),making a local map of the environment and computing the path transform can be done in less than 3 seconds.

4. Planning Paths Of Complete Coverage

Much of the focus of the research effort to date has centred on the problem of finding a path from a start location to a goal location, while minimising one or more parameters such as length of path, energy consumption or journey time. A mobile robot should be capable of finding paths which ensure the complete coverage of an environment. The path transform can be adapted to find such a path. To find the path of complete coverage, instead of descending along the path of steepest descent to the goal, the robot follows the path of steepest ascent. In other words the robot moves away from the goal keeping track of the cells it has visited. The robot only moves into a grid cell which is closer to the goal if it has visited all the neighbouring cells which lie further away from the goal. A full discussion of paths of complete coverage is provided in Zelinsky *et. al.* [16]. Refer to Figure 9 for an example of the path of complete coverage.

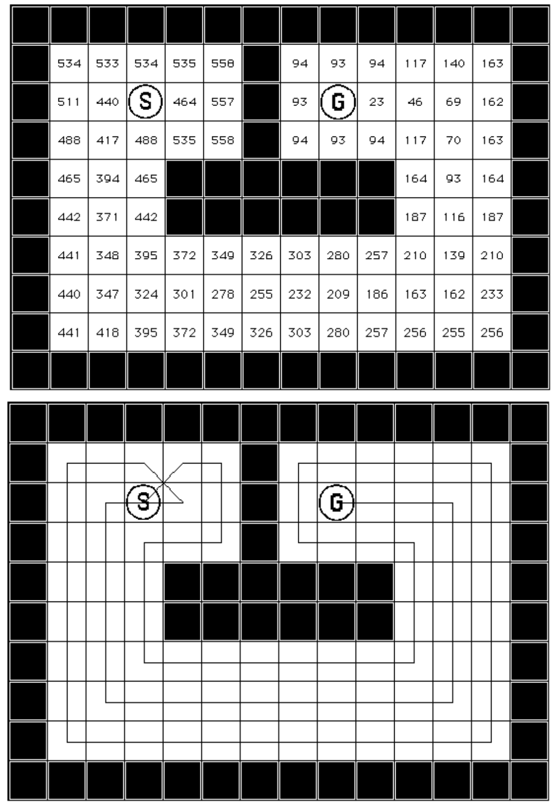

Figure 9. Path of Complete Coverage

To correctly execute the planned path of complete coverage the robot must navigate with great accuracy. To rely on dead reckoning alone to execute a long and complicated path is unsatisfactory. The Yamabico periodically re-adjusts its position estimate. This is done using the techniques described earlier in this paper. The precomputation of the landmark sensing points can be elegantly done by using data needed to compute the path transform. The path transform is computed by using a combination of the obstacle and distance transforms. The obstacle transform knows the exact distance that each free space grid cell is from the closest obstacle filled grid cell. We modify the obstacle transform to include directional information i.e. in which direction the closest obstacle lies. The Yamabico is then able to correct its position estimate dynamically. Refer to Figure 10. for output from the AMROS simulator of a path of complete coverage.

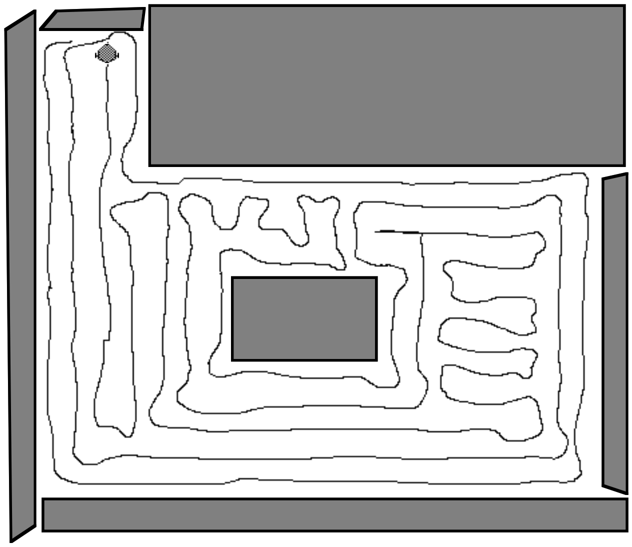

Figure. 10 Simulation Result

The path of complete coverage was successfully implemented in an environment which was similar to the one shown in the simulation.

5. Conclusions

This paper presented a technique for unifying navigation schemes for a mobile robot in known and unknown environments. A method was presented to join global and local path planners that takes into account maintaining the estimated position of the robot. An efficient method for local path planning based on numeric potential fields called the path transform was also presented. It was also shown how the path transform could be easily adapted to create a new type of path which ensures complete coverage of the environment by a mobile robot. The validity of the reported approaches was demonstrated by implementation of the navigation schemes on our self-contained autonomous robot - the Yamabico.

Acknowledgement

The authors are extremely grateful to Mr. Katsumi Kimoto for providing his simulator, which was an extremely valuable tool in debugging the robot programs. Special thanks are extended to Mr. Keiji Nagatani for his helpful assistance during the course of this research.

References

[1] S. Yuta, S. Suzuki and S. IIda, "Implementation of a Small Size Experimental Self-Contained Autonomous Robot - Sensors, Vehicle Control, and Description of Sensor based Behaviour", Proceedings of 2nd International Symposium on Experimental Robotics, June 25-27, 1991, Toulouse, France.

[2] R. Chatila, "Path Planning and Environment Learning", European Conference on Artificial Intelligence, pp211-215, July 1982.

[3] J.L. Crowley, "Navigation for an Intelligent Mobile Robot", IEEE Journal of Robotics and Automation, RA-1 No. 1, pp31-41, March 1985.

[4] H.P. Moravec, "Obstacle Avoidance and Navigation in the Real World by a Seeing Rover", Phd dissertation, Stanford University, September 1980.

[5] C.E. Thorpe, "FIDO:Vision and Navigation for a Robot Rover", Phd dissertation, Carnegie Mellon University, Department of Computer Science, December 1984.

[6] O. Khatib, "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots", International Journal of Robotics Research, Vol. 5 No.1, pp90-98, 1986.

[7] Y. Koren and J. Borenstein, "Potential Field Methods and their Inherent Limitations for Mobile Robot Navigation", Proceedings of IEEE International Conference on Robotics and Automation, pp1398-1404, April 1991.

[8] J. Boreinstein and Y. Koren, "Real-Time Obstacle Avoidance for Fast Mobile Robots in Cluttered Environments", Proceedings of IEEE International Conference on Robotics and Automation, May 1990, Cincinatti, pp 572-577.

[9] R. Chattergy, "Some Heuristics for the Navigation of a Robot", International Journal of Robotics Research, Vol. 4 No.1, pp59-66, 1985.

[10] A. Zelinsky, "Environment Exploration and Path Planning Algorithms for a Mobile Robot using Sonar", Phd dissertation, University of Wollongong, Department of Computer Science, October 1991.

[11] S. Suzuki, M.K. Habib, J. Iijima and S. Yuta, "How to Describe the Mobile Robot's - Sensor based Behaviour?", Journal of Robotics and Autonomous Systems, No.7, 1991, North-Holland, pp 227-237.

[12] K, Nagatani and S. Yuta, "Path and Sensing Point Planning for Mobile Robot Navigation to Minimize the Risk of Collision", Proceedings of International Conference on Intelligent Robots and Systems (IROS), July 26-30, 1993, Yokohama, pp2198-2206.

[13] R.A. Jarvis and J.C. Byrne, "Robot Navigation: Touching, Seeing and Knowing", Proceedings of 1st Australian Conference on Artificial Intelligence, November 1986.

[14] J. Barraquand and J.-C. Latombe, "Robot Motion Planning: A Distributed Representation Approach", International Journal of Robotics Research Vol. 10 No.6, pp628-649, 1991.

[15] K. Kimoto and S. Yuta, "A Simulator for Programming the Behavior of an Autonomous Sensor-Based Mobile Robot", Proceedings of International Conference on Intelligent Robots and Systems (IROS), July 7-10, 1992, Raleigh, pp 1431-1438.

[16] A. Zelinsky, R.A. Jarvis, J.C. Byrne and S. Yuta, "Planning Paths of Complete Coverage of an Unstructured Environment by a Mobile Robot", to be

presented , International Conference of Advanced Robotics - ICAR '93, November 2-5, 1993, Tokyo.