Proceedings of the 2004 IEEE
International Conference on Robotics & Automation
New Orleans, LA • April 2004

A Genetic Algorithm Based Area Coverage Approach for Controlled Drug Delivery using Micro-Robots

WeiMin Tao
Brooks Automation Inc
Mountain View, California, USA
t_weimin@hotmail.com

Mingjun Zhang
Agilent Technologies
Palo Alto, CA 94304
mingjunzhang@ieee.org

Tzyh-Jong Tarn
Washington University
St. Louis, MO 63130
tarn@wuauto.wustl.edu

***Abstract*—This paper describes a Genetic Algorithm (GA) based approach for area coverage using micro-robots. The method can be used for controlled drug delivery or tumor treatment using micro-robots. The algorithm aims to find a near-optimal path that covers a given area entirely, except obstacles defined as biological barriers or drug side effect restricted zones. The proposed GA approach is a dynamic on-line path planning approach, which is able to achieve planning during motion and to response to detected obstacles. Different from point-to-point path planning, the proposed operators for the GA are specially designed for area coverage. Comparisons of the approach with high-level static optimal search algorithms and low-level fixed path planning approaches are also presented. Simulation results are given to show the effectiveness of the approach.**

Keywords-Genetic algorithm; area coverage; drug delivery; tumor treatment; micro-robots.

I. INTRODUCTION

Controlled drug delivery is the definite way for future drug delivery [1]. Controlled drug delivery is the phasing drug administration to the need of a condition at hand so that an optimal amount of drug is applied at an exact location to cure and control the condition completely in minimum time. Controlled drug delivery using micro-robots shows promising futures. One challenge for controlled drug delivery using micro-robots is the area coverage, i.e., how to delivery drugs to exact and all tumor sites without or less side effects? This is a problem of micro-robot area coverage. An efficient algorithm is needed to completely survey tumor zones and deliver the drug to proper locations for treatment. In the meantime, avoid the area of "obstacles", which are defined as areas that drug side effects are strongly restricted, or areas no therapeutic effects are expected, or barriers that micro-robots cannot traverse through, such as Brain-Blood Barrier, etc.

Many approaches may be used for robot area coverage. The focus of this work will be the genetic algorithm (GA) based area coverage. One reason for choosing GA-based approach is that the GA approach has been proven to be effective and efficient in solving many practical problems. A number of previous works were done on applying GA approach to point-to-point path planning [2][3][4]. On the other hand, only a few works have been reported in the open literatures on area coverage using GA approach. Ref. [5] introduced a CGA approach for learning gait cycles for area coverage in a rectangle area with known path pattern. Genetic algorithm based approach may be an interesting topic for area coverage in general.

The area coverage path planning in this paper is defined as: given an arbitrary boundary of planar area with populated obstacles, plan robot path to guide the robot traverse through the area completely. In drug delivery context, it means given a bounded zone of tumor tissues with biological restrictions, plan a path for robot traversing the infected zone completely and release drug at infected center locations.

Clearly, the path planning problem is different from previous work on point-to-point path planning in the open literatures. It's also different from CGA approach in the sense that the optimized object is a robot's coverage path instead of motion control.

In this study, the drug delivery area is represented as a digitized planar map, which is further divided into various equal sized square map grids. Assume the micro-robot's drug delivery device has an effective delivery radius of R_s and the drug delivery round with radius R_s is just good enough to cover a square map grid. To cover an area (spread drug) is to traverse through all the centers of the grid cells (the center of a grid cell). It is also better to target and release drug at the center of a grid cell. Areas with obstacles are those grid cells, which will block robot's movement. To illustrate the idea, Fig.1 shows a simple square shape of drug delivery area, where the path is shown in dashed lines with arrows to represent moving directions. S is the starting point, which can be specified anywhere in the map or outside the map.

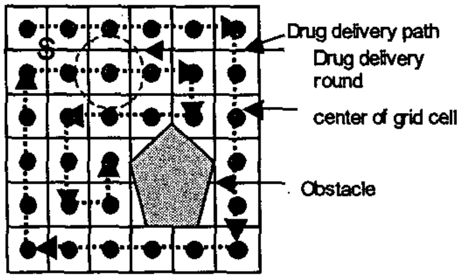

Fig. 1 Area coverage for drug delivery

The objective of path planning for area coverage in this paper is to find a near optimal collision-free path, which connects all the centers of grid cells free of obstacles. Since

0-7803-8232-3/04/$17.00 ©2004 IEEE

some obstacles are unknown prior to drug delivery and have to be discovered during the course of robot's movement, the path-planning algorithm has to be flexible and dynamic to cope with on-line environmental changes.

Traditional off-line planners with incremental map building approaches are costly in both processing time and memory utilization. They are usually inflexible due to their optimal search nature [6]. On the other hand, area coverage approaches with fixed path plan (e.g. back-and-forth boustrophedon motion) may not be able to achieve high efficiency in some circumstances due to lack of global path optimization [7][8][9]. For example, a robot may need to make too many turns to cover an area, which is constraint by obstacles and with relatively small open areas with respect to robot size. On-line Goal Selection (OGS) with behavior based approach has certain path flexibility [10]. But lacking global path optimization will still lead to low efficiency.

The proposed GA approach has the following advantages:

- The GA evolutionary feature leads to near global path optimization, which will result in highly efficient paths.
- Dynamic on-line path planning allows planning during motion to increase planning efficiency and to cope with uncertainties of the environment, especially newly sensed biological changes (obstacles).
- By selecting the population number and generation cycle number N, we can control the planning cost (e.g. processing time and memory utilization) to meet application requirements.

In next section, the new GA-based area coverage approach will be introduced. Simulation results and conclusions are given in sections 3 and 4, respectively.

II. GA APPROACH for AREA COVERAGE

1. Overview

A. *Genetic Algorithm Procedure*

In this study, a chromosome refers to a path connecting the centers of a number of map grid cells, which the robot will follow through for area coverage. Since the robot will move cell by cell, the starting point of chromosomes has to be changed dynamically whenever a new path is adopted. The ending point of a chromosome is not fixed and need to be determined by applying GA operators.

At the beginning, all the chromosomes in the population only have starting points. Once started, a chromosome in the population is randomly selected and an operator is selected based on probability distribution, which is adjusted dynamically according to an adaptive probability scheme. The operator is applied to the chromosome to create an offspring. By applying operators, the chromosomes will start to expand to link up with other grid cells and get improved. Offspring chromosomes and all other chromosomes in the population are evaluated using evaluation function that measures the fitness of the chromosome for area coverage. The bigger the value of the evaluation function, the better is the chromosome. If the worst chromosome in the population is worse than the new offspring, the worst chromosome will be replaced with the new offspring. This completes one evolutionary cycle and results in a new population. The evolutionary cycle continues until all grid cells (possible drug targeting sites) are visited by the micro-robot.

B. *Dynamic on-line Path Planning*

Dynamic on-line path planning is implemented for two main reasons:

- Planning during motion to increase efficiency.
- In response to uncertainties in the environment, such as growing tumors sensed recently.

The micro-robot's movement is a consecutive process moving from current grid cell to next one along current best chromosome. During its motion, the micro-robot also keeps updating its chromosome population by applying operators. Such update will make the chromosomes in the population evolve to become better ones. Dynamic on-line path planning is implemented as follows:

- GA process is applied while the robot is moving. The GA process is cycles of selecting a chromosome, applying an operator, updating the population. Each cycle will produce one generation of chromosome.
- The micro-robot will update its current chromosome every N generations by re-selecting the best chromosome from the population and setting its next grid cell as the starting point of the new chromosome. Currently N is set to be 100, which ensures the GA process will complete before the robot reaches the current goal cell so that the next goal cell is always the result of the N generations GA process.
- All chromosomes' starting points will be updated once the robot reaches its current goal cell, and the new GA process will be started on the population with new starting points.

Whenever the micro-robot senses a new obstacle, the map grid cells covered by the obstacle become infeasible and so are some of chromosomes. The GA process starts running for at least N generations and till a feasible path is found.

After the robot moves to its current goal cell, in addition to update starting point, all chromosomes must also be modified to response to the change of starting point. This is done by the following two mechanisms:

- *Deleting part of the path.* Starting from the beginning of the chromosome, the algorithm deletes the nodes of the chromosome until a node coincides with the new starting position. The resulting chromosome is the rest of the original chromosome. This works only if there is a node in the original chromosome that coincides with the new starting position.
- *Going to the point.* Starting from the new starting position, a chromosome is generated to the starting position for the original chromosome. From then on, it follows the original chromosome.

While implementing the algorithm, *deleting part of the path* is tested first. If it is unsuccessful, the second method will be applied. The reason for doing that is that the *deleting part of the path* is computational cheaper and would not cause overlaps in the area covered. However, this also means that all the information stored in the deleted part is lost.

C. *Expandable Chromosomes*

Different from GA approach for point-to-point planning that always starts from some initialized existing chromosomes with fixed starting and ending points, the ending point for area coverage is open. Hence the chromosomes have to be expandable to cover all unvisited points from their starting points. Operators need to be designed with such capability.

2. Operators

An operator is an evolution function to be applied to chromosomes to evolve the population. Six operators are proposed and implemented in our system. Later simulation shows that this is sufficient for a terrain of irregular shape. The application of these six operators is probabilistic. These operators are described below:

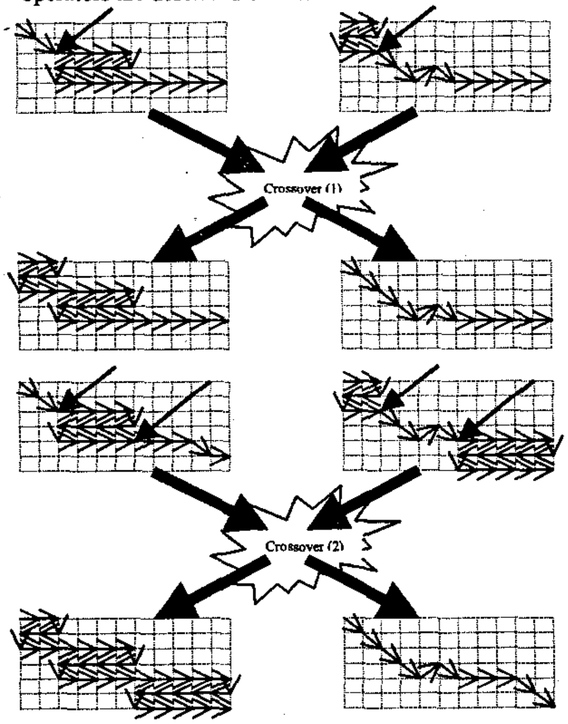

Fig. 2 Crossover operator

Crossover recombines two parent chromosomes into two new ones. There are two types of operators:

1). The parent chromosomes are divided randomly into two parts and then recombined. The way of recombination could be the first part of the first chromosome with the second part of the second chromosome; or, the first part of the second chromosome with the second part of the first chromosome.

2). The parent chromosomes are divided randomly into three parts respectively and then recombined. The way of recombination could be the first part of the first chromosome with the second part of the second chromosome, followed by the third part of the first chromosome; or the first part of the second chromosome with the second part of the first chromosome, followed by the third part of the second chromosome.

It should be noted that crossover$_2$ is essentially crossover$_1$ applied twice to different points. Due to the relatively higher expensive nature of this operator (O(n^2) time instead of O(n) time by the other operator, where n is the number of nodes), it may be an advantage to have crossover$_2$. Crossover operator is illustrated in Fig. 2. In this diagram, the vertices of the squares are the center of grid cells, so robot will move from on vertex to the next one. The arrow shows the chromosome's direction along which the robot will move. The upper two chromosomes are parents and the lower two are offspring.

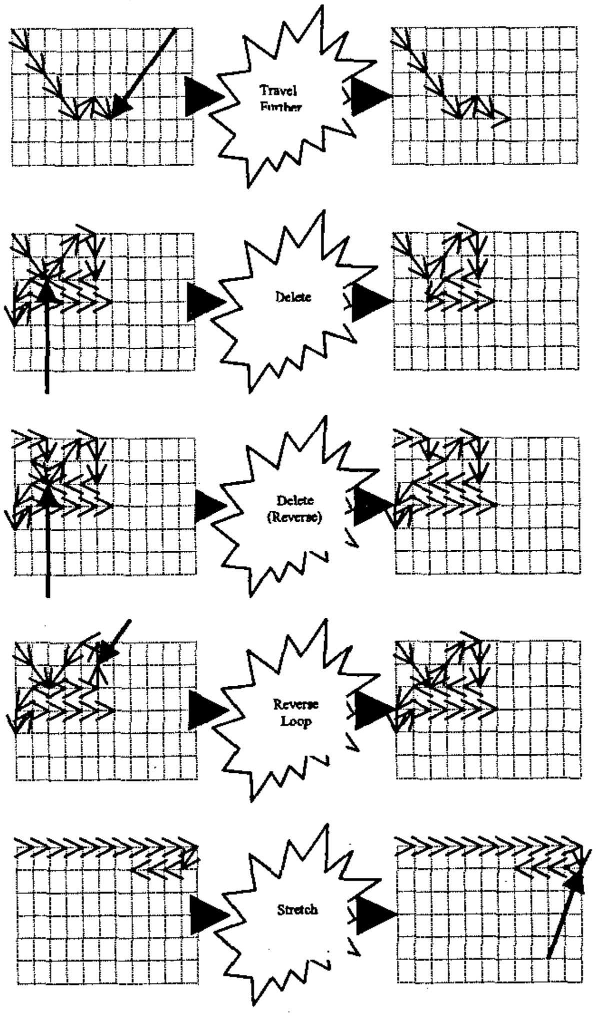

Fig. 3 Five other operators

Travel Further extends the chromosome by one node in any of eight directions (left, right, up, down, diagonal left up, diagonal left down, diagonal right up, diagonal right down). Highest priority is given to travel in the same direction, subsequent traveling directions are ordered by turning angle with less turning angle first. In addition, higher probability is also given to going to a point that has not been visited. This operator realizes the expandable chromosomes for area coverage.

Delete: deletes a point in the chromosome that has been traversed earlier in the same chromosome. In order to reconnect up the chromosomes, more than one point may be deleted.

Reverse Delete: deletes a point in the chromosome that will be traversed later in the same chromosome. Similar to **Delete**, more than one point may be deleted in order to make the chromosome feasible.

Reverse loop: randomly finds a loop in the chromosome and then reverse the direction in which the loop is traversed.

Stretch: inserts a node into the middle of the chromosome. Nodes that have the best effect on the change in turning angles of the chromosome are given precedence. In addition, higher probability is given to a point that has not been visited.

Fig. 3 shows how these five operators work, where all the left chromosomes are parents and the right ones are offspring.

Applying each operator to chromosomes will generate children chromosomes. The value of the evaluation function for these offspring will be compared with the current chromosome population and replace the one, which is the worst chromosome in the population.

3. Evaluation

Evaluation is the process to measure the value (fitness) of the chromosomes for area coverage goal. Chromosomes are classified as two classes: infeasible chromosomes and feasible chromosomes. Infeasible chromosomes are zero length chromosomes and chromosomes that pass through obstacles or boundaries. All other chromosomes are feasible chromosomes.

In the algorithm, the value of each chromosome is evaluated by the following factors:

Eval(p)= [distinct(p)+forgetfulness(p)]/[length(p)+turns(p)]

The higher the value of Eval(p), the better the quality of the chromosome. Also, if the chromosome is identical to another chromosome in the population:

$$Eval(p) \leftarrow Eval(p)/1.5$$

This prevents identical chromosomes to flood over the entire population.

The distinct, forgetfulness, length, turns are defined as follows:

- *distinct(p)* = number of distinct points that are marked *NotVisited* that are traversed by the chromosome. A chromosome with more distinct points have higher Eval(p) value. This will encourage evolution to select a chromosome with more NotVisited points.
- *forgetfulness(p) = forgetrows(p) + toforget(p).*
  where:
  - *fogetrows(p)* = the number of rows in the map that the robot can forget if it traversed this chromosome. The robot can forget row *i* if and only if it can forget all the rows before *i* and none of the points in row *i* is marked *NotVisited.*
  - if *i-1* is the maximum row number of the rows that forgetrows can forget,
    toforget(p) = the fraction of row *i* that is covered if chromosome p is traversed.

  The *forgetfulness* factor helps to guide the micro-robot along a general path to find tumor site and deliver drug. This is necessary, since after the robot finished a row, it can forget it. *Forgetfulness* gives more merit to a chromosome which could remove some sub-area been visited entirely. It would also help to ease the memory constrains for storing the digitized map. However, care must be taken to ensure that after forgetting the rows, the resultant map is not disjointed because of the irregular boundaries or because of obstacles

- *length(p)* = the length of chromosome p. The bigger the length, the smaller the evaluation value. This evaluation factor leads to finding a chromosome which has the shortest length but still cover the whole area. In other words, this evaluation factor will minimize the overlapping travel distance for area coverage.
- *turns(p)* = the sum of all the turns in the chromosome, where a turn value is (angle of turn)/(2*π). We consider too many turns and sharp turning angle as energy expensive and harmful to a physical robot. This function penalizes too many turnings of the chromosome for evolving a smoother and more efficient path.

4. Adaptive Operator Probability

An operator probability is initially set randomly, and adjusted based on its effectiveness in improving chromosomes. Ref. [3] reported significant improvement of adaptive operator probability vs. constant operator probability for point-to-point path planning. This paper adopted a similar approach for area coverage. The adaptation of an operator's probability is governed by the Performance Index (PI) of that operator. Adaptation takes place every N generation. Let *k* denotes the number of adaptations that have occurred since generation zero.

PI of an operator *i* is defined as:

$$PI_i(k) = e_i(k-1) \times p_i(k-1)$$

where

- $e_i(k-1)$ = the ratio between the number of times operator *i* improves a chromosome and the total number of times it is applied since the last adaptation. It measures the effectives of *i*.

- $p_i\ (k-1)$ = the probability of applying operator *i* since the last adaptation. $p_i\ (0)$ is fixed before the algorithm is run.

The new probability of applying operator *i* after the *kth* adaptation will then be:

$$p_i\ (k) = \frac{PI_i}{\Sigma_{j=1}^{6} PI_j}$$

However, $p_i(k)$ is further modified so that no $p_i(k)$ is less than a threshold $p_{threshold}$, since no $p_i(k)$ is zero. It makes sense to give each operator at least some chances.

The $p_i(k-1)$ factor represent the calculation of $PI_i(k)$ so that the history of the operator is somewhat reflected in the new probabilities. It can also serve to stabilize the probabilities so that there will not be sudden changes to the probabilities. However, this also means that the operators are not able to adapt as fast as without this factor.

Ideally, the operation time and operation side effects for the operators can also be included as factors in the calculation. However they are not included in this implementation because the operation time is not available on the platform.

III. SIMULATION RESULTS

To test the proposed GA approach, we have developed a micro-robot simulation software using Visual C. Tests have been conducted to evaluate the performance of this approach. Test results have also been compared with OGS approach that has no global path optimization function. Fig. 4 to Fig. 7 show two examples of test results. In these pictures, the micro robot is represented as a small blue square with a tail and two eyes. The obstacles are represented as polygons. The chromosome population of GA approach is set to 30.

Fig. 4 and Fig. 5 show two sample results from GA approach and OGS approach respectively. Both tests use the same terrain map (irregular boundary and two obstacles). We use overlapped travel distance and total turning angle to measure the efficiency of a planned path. Obviously, the lesser the total path length is, the lesser the overlapped travel distance is. By placing the path length and total turning angle into the evaluation function, the GA algorithm is to minimize these two parameters.

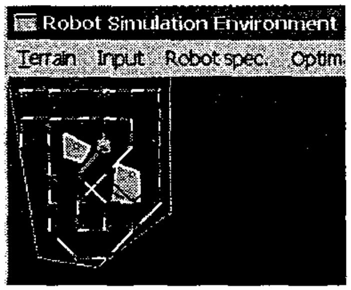

Fig.4 Area Coverage track with GA based approach

Fig. 4 shows the drug delivery path with GA approach. In this simulation, the robot kinematics model and physical size is not incorporated. So the track is a series of straight lines connecting the centers of all feasible grid cells. Initially we set probability of 0.5 to **travel further** operator and 0.1 to the rest of operators. The robot starts to move after 500 generations and reselect the current best path every 100 generations. Obviously, the area coverage path has very small amount of overlapping travel distance and total turning angle. This shows the effectiveness of near global path optimization.

Fig. 5 shows the drug delivery path with OGS approach without global path optimization. OGS approach selects its next goal cell in a fixed pattern from its nearby layer of cells. The pattern is associated with user input to realize back-and -forth, zigzag and spiral path plan. In this simulation, we choose spiral path plan for area coverage and the robot kinematics model and physical size is incorporated. Some grid cells were not connected by robot's track due to slightly bigger drug delivery radius. However, those capability differences from GA approach won't affect performance comparison of the two approaches. Obviously we can see more overlapping travel distance and bigger total turning angle in the resultant coverage path here.

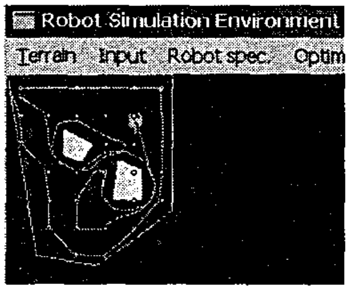

Fig. 5 Area Coverage track with OGS approach

Fig. 6 and Fig.7 show another example comparing the GA based approach to OGS approach with back-and-forth path pattern. In this map, there is only one obstacle.

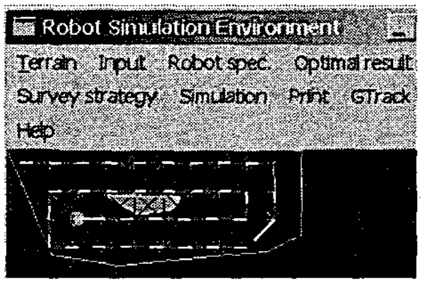
Robot Simulation Environment
Terrain Input Robot spec. Optimal result
Survey strategy Simulation Print GTrack
Help

Fig. 6 Area coverage track with GA based approach

Fig. 6 shows the drug delivery path with GA based approach, apparently, this path features no overlapping travel distance and smallest total turning angle (though this is not a general conclusion).

Fig.7 shows the same map using OGS approach. Here, the back-and-forth path pattern was selected. Apparently, this path pattern caused the robot to make many turns to cover the area, which leads to very big total turning angle. The overlapping travel distance is fairly small but not zero.

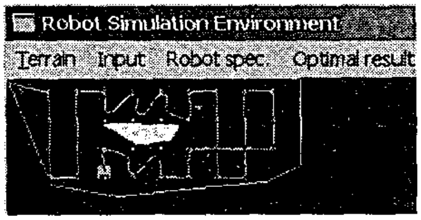

Fig. 7 Area coverage track with OGS approach

Various tests have been conducted and observed significant amount of improvements in overlapping distance and turning angle with GA approach, though individual results may be quite different depending on their given map.

IV. CONCLUSIONS

A GA based approach is proposed in this paper for area coverage. It can be used for drug delivery or tumor treatment using micro-robots. The method features dynamic on-line path planning and near global path optimization. The "expandable chromosome" concept is proposed and implemented. Simulation results have qualitatively showed the efficiency of this approach.

With the GA evolutionary feature, this approach has advantage over traditional optimal path planning approaches, which usually have high computational cost, such as planning time and memory utilization. This approach also has advantage over fix path plan approaches due to its near global optimization feature. Our simulation results showed that more efficient paths could be generated, especially in some types of terrain (e.g. obstacle free area is relatively small to robot physical dimension), which may result in low efficiency with fixed path plan.

The discussion of this paper is restricted to algorithm design only. It is assumed that the micro-robots can be made and sense as well as actuate well. There are still many research topics for physical implementation of the micro-robots for drug delivery. Certainly, area coverage is one of the immediate needs for control of drug delivery micro-robots. On the other hand, the proposed GA algorithm may be used for robot area coverage or survey in general as well.

REFRENCES

[1] B. Berner and S. M. Dinh, Electronically Controlled Drug Delivery, CRC Press, May 1998.

[2] T.Shibata and T. Fukuda, "Intelligent Motion Planning by Genetic Algorithm with Fuzzy Critic", Proceedings of the 8th IEEE International Symposium on Intelligent Control, Chicago, pp.565-570, August 25-27, 1993.

[3] J. Xiao and L. Zhang, "Adaptive Evolutionary Planner/Navigator for Mobile Robots", IEEE trans. Evolutionary Computation, Vol. 1, No. 1, April 1997.

[4] Kazuo Sugihara, "GA-based On-line Path Planning for SAUVIM " IEA/AIE, Vol. 2, pp329-338, 1998.

[5] Gary B. Parker, "Learning Area Coverage Using the Co-Evolution of Model Parameters", Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2002), pp. 1286-1293, July 2002.

[6] J.C.Latombe, Robot Motion Planning, Kluwer Academic Publishers, Norwell, MA, 1991.

[7]. S.Hert, S.Tiwari and V.Lumelsky, A Terrain-covering algorithm for an AUV, Autonom. Robots 3, 1996, pp91-119.

[8] J.VanderHeide and N.S.V.Rao, Terrain coverage of an unknown room by an autonomous mobile robot, Technical report ORNL/TM-13117, Oak Ridge National Laboratory, Oak Ridge, TN (1995).

[9] Ercan U. Acar, Howie Choset, Sensor Based Coverage of Unknown Environments: Incremental Construction of Morse Decompositions, The International Jounal of Robotics Research, Vol. 21, No.4, April 2002, pp. 345-366.

[10] Tao Wei Min, How Khee Yin, "A Decentralized Approach for Cooperative Sweeping by Multiple Mobile Robots", IEEE/RSJ Intelligents Robot and Systems Conference (IROS98), Victoria, Canada. 1998