# A Two Phase Recursive Tree Propagation based Multi-Robotic Exploration Framework with Fixed Base Station Constraint

Piyoosh Mukhija, K Madhava Krishna and Vamshi Krishna

*Abstract*— **A multi-robotic exploration with the requirement of communication link to a fixed base station is presented in this paper. The robots organize themselves into roles of maintainers of communication (hinged robots or robot nodes) or explorers of the environment ensuring that every robot is in contact with the base station directly or through the hinged robots. A two phased strategy for the same is presented. The first phase is characterized by a recursive growth of trees that starts from the root node or the base station and then repeated from other nodes of the hitherto grown tree in a depth first fashion. The second phase constitutes the recursive tree growth invoked repeatedly from the frontier nodes. While the first phase rapidly explores areas around the base station in a concentric fashion, the second phase extends the depth of the explored area to increase the limits of coverage. The strategy is consistent in that none of the robots loose contact with the base station. Extensive simulations confirm the efficacy of the method and comparisons portray performance gain in terms of exploration time and absence of deadlocks vis-a-vis the few methods previously reported in the literature.**

## I. INTRODUCTION

A multi robotic exploration algorithm where the robots are required to always maintain a communication with a fixed base station is presented in this paper. The novelty of this work lies in its method that provides for higher per unit time information gain and very low computation and exploration times compared to various methods that tackle the problem of exploration with base-station connectivity constraint. The continuous connectivity to base-station is considered important as the robots other than the base-station itself are assumed to be of minimalistic configuration (capable of only basic autonomy such as obstacle avoidance), being guided by base station over the ad-hoc network. Also, the exploration scenario may contain other sensor data (images, ground-analysis, etc.) beyond map information, too large or time critical for prolonged containment by such simple robots. The problem addressed here is also novel having been addressed only once before by the current authors in [1] in the context of exploration rather than point by point coverage as in [9], [10]. In other words, the robots need not visit every point in the map, but only such that their sensors are able to capture the whole map.

This work was not supported by any organization.

Piyoosh Mukhija is a research student in the Department of Computer Science, IIIT Hyderabad, Gachibowli, Hyderabad, Andhra Pradesh, India piyoosh.mukhija@research.iiit.ac.in

K. Madhava Krishna is with the Faculty of Computer Science, IIIT Hyderabad, Gachibowli, Hyderabad, Andhra Pradesh, India mkrishna@iiit.ac.in

Vamshi Krishna is a research student in the Department of Computer Science, IIIT Hyderabad, Gachibowli, Hyderabad, Andhra Pradesh, India vamshi.krishna@research.iiit.ac.in

An earlier method, presented in [1], introduced the concept of an *Exploration Tree* which is grown by allocating and extending communication nodes for the exploration process. The method proposed in this paper uses the same concept of Exploration Tree and enhances this exploration process greatly by providing new and better techniques for the tree growth in terms of exploration and computation efficiency. The method from [1] would henceforth be referred to as Single Hinged Method (SHM) since only one hinge point (node) is allotted and expanded at a time for the tree growth in it. While the SHM guaranteed a direct or indirect contact of all the robots with the base station throughout the exploration, it was considerably slow due to lack of parallelism. The computation time involved in pruning the node layout configurations (simulated look-ahead) is of exponential nature with respect to the number of robots. However, the new proposed method, termed as Multiple Hinged Method (MHM), is aptly designed to overcome these limitations. Allotting and expanding multiple hinge points (nodes) ensures that the robots are well utilized, their net information gain per unit time is significantly higher and minimizes the time loss in rollback processes in the SHM. Instead of hinging along certain number of directions, MHM tries to cover the frontiers by fitting the boundary with overlapping cells (explained in the section on *Wavefront Expansion*), thus reducing the computation time drastically by making its characteristics almost linear with respect to number of robots. This difference is clearly indicated from the observations in table I.

Furthermore, this method may also be compared to various other possible methods such as a chain or line sweep, all of which have their own set of disadvantages/problems. For example, a radial-chain sweep that looks like a simple approach would require precise coordinated motion of each robot, and would have problems in handling large obstacles (especially with base-station connectivity constraint). Rendezvous methods, which cater to the robots with higher configuration requirements, aren't applicable here at all owing to the problem definition (requirement of continuous/real-time connectivity with base-station).

## II. RELATED LITERATURE

From the viewpoint of exploration some of the earliest approaches have been due to [4] that traded of information gain with distance to be travelled while allocating robots to frontier locations through a bidding process. These methods built upon the frontier exploration strategy presented for a single robot in [11]. A similar method was presented in [2]

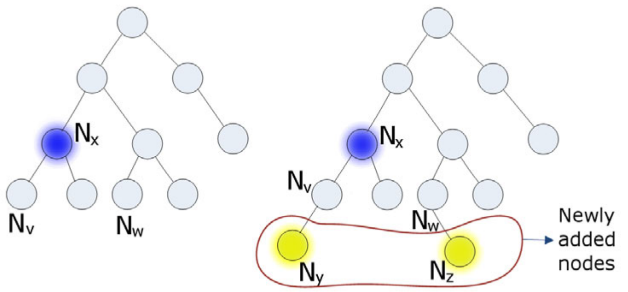



Fig. 1. Wavefront Expansion being applied on $N_x$ during the Recursive Tree Growth Procedure. New leaf nodes $N_y$ and $N_z$ were created and were assigned parents $N_v$ and $N_w$ in the exploration tree. The processing of $N_y$ and $N_z$ later takes place immediately after $N_v$ and $N_w$ respectively.

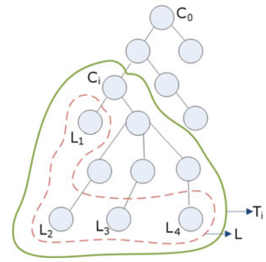



Fig. 2. A subtree $T_i$ with its root node $C_i$ within the exploration tree. The set $L$ is the set of leaf nodes belonging to $T_i$.

while [3] fused multi robot localization and map fusion along with exploration. Very recently [5] showed how the choice of the metric affects the exploration time in a significant manner.

In the broader context of multi robot navigation with constraints, a variety of problems have been tackled. [8] shows how robots maintain formation constraints while navigating from one location to another. In [7] the problem of maintaining a network of mobile robots while exploring an environment is presented. Typical of behaviour based approaches modelled over a push pull potential field paradigm, the authors are silent about the maintenance of the network constraints especially in the wake of challenging obstacle configurations. A method for sensor based coverage with a k-connected constraint was presented in [12] while [6] propose a strategy of routing a network of robots through various locations where network constraints are required to be maintained. [13] propose a generalized framework for tightly coupled multi robot planning problems, wherein they adapt their framework for navigating a set of robots to various goals with the requirement that each robot is in line of sight of communication with one other robot.

However, none of the above approaches tackle the problem of terrain exploration with multiple robots with the base station constraint. The problem is addressed once before [9], [10] in the context of point by point coverage rather than exploration. In other words, the method of [9], [10] for coverage problem requires every free cell in the map be visited instead of the exploration problem addressed in this effort, which requires every free cell be "seen" by the sensors.

## III. Methodology

Here we use a role-based approach as the one described in [1]. Each robot serves the role of either an *explorer* or a *communication link station (node)*. When acting as a node, a robot may not move. It acts as a repeater along the link chain (communication backbone made of nodes, also referred to as hop path) and provides a communication link to explorers within its range. At all times, the distance between two nodes or between an explorer and its closest node is never more than their communication range $r_{comm}$. The area within $r_{comm}$ of a node is termed as a *node cell* or simply *cell*, which is explored using frontier exploration method with all the available explorer robots. The areas/points lying on the periphery of a cell (just within $r_{comm}$) constitute the *cell boundary*. Here, we use the terms cell and node interchangeably and the transition of a robot's role from explorer to node is termed *hinging* and vice-versa as *unhinging*. The map is divided into *grid cells* which define the maximum resolution for the obstacles and robot positions.

In the MHM algorithm, the explored cells are considered as tree nodes, which form an *exploration tree* (denoted by $T_0$ throughout the text). In the first phase of the algorithm, this tree starts with a single node i.e. the base station cell ($C_0$) and it grows as the algorithm proceeds. Section III-D describes the MHM algorithm for exploration in detail using a procedure known as Recursive Tree Growth (RTG) (described in section III-B). The RTG involves processing the tree-nodes belonging to the exploration tree in a mechanism akin to Depth-First Traversal. This processing, done on each individual tree-node, is termed as *Wavefront Expansion* (described in section III-A), and it creates more cells, which are added as leaf nodes to the tree after being explored. The first phase ends when no more cells can be added to the tree and all the nodes in the exploration tree have been traversed by RTG. It should be noted that the depth-first processing of the tree automatically includes the newly added nodes as well. e.g. if $B$ is a leaf node resulting from the processing of $A$, and is assigned as a descendent of $A$ or another node due to be processed, $B$ will also be processed after its immediate parent is processed (see Figure 1).

The second phase of the algorithm involves finding the frontier closest to the base station, creating the shortest hop path to it and executing RTG with the node at the frontier as the base. This process from finding closest frontier to executing RTG is repeated until no frontier exists (for closed areas) or all the frontiers are out of reach (for open areas).

### A. Wavefront Expansion

The wavefront expansion is the fundamental building block of the MHM algorithm. It chooses multiple hinge points and propagates the tree in parallel fashion. The amount of hinge points chosen depends on the number of robots still available to hinge and the hierarchy of the node in the tree. At the top of the exploration tree, $T_0$ the number

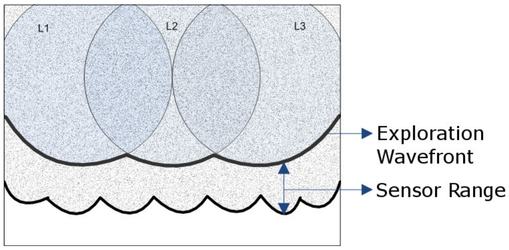



Fig. 3. Wavefront determined from 3 leaf cells $L_1$, $L_2$ and $L_3$. The shaded area has been explored and unshaded area is unexplored.

of hinge points due to wavefront expansion is higher and it progressively reduces as wavefront expansion proceeds downwards in depth first fashion.

In the exploration tree ($T_0$), if we consider a cell $C_i$ ($C_i \in T_0$) under processing, there exists a sub-tree $T_i$ ($T_i \subset T_0$) such that $T_i$ consists of $C_i$ and all its descendants, and $C_i$ is its root node (see Figure 2). As shown in the same figure, we define a set $L$ containing all the leaf nodes of $T_i$. Now, the *Exploration Wavefront (EW)* associated with cell $C_i$ may be defined as the collection of points belonging to the boundary of cells in $L$, close to the exploration frontiers. In other words, for each cell in $L$, every boundary point having an exploration frontier within a distance $R_s$ (Sensor Range) would belong to the EW associated with $C_i$ (Figure 3).

We arrange the points in EW associated with $C_i$ in an ordered set $B$ based on their bearing from the base station and choose the first point in the set as a node position $P_i$. Next, we filter the set $B$ by removing all the points (including $P_i$) from $B$ within distance $r_{comm}$ (communication range of each robot) from $P_i$. The process of taking the first point from $B$ and filtering $B$ is repeated until set $B$ is empty, thus generating a set of $H$ potential node positions (points taken from $B$). Now, we hinge the robots at these positions if:

$$H \leq X_{available} - X_{min}, \quad (1)$$

where $X_{available}$ is the number of explorers available before hinging and $X_{min}$ is a limiting parameter defined as the minimum number of explorers required for exploration during expansion. When the robots are hinged at these $H$ positions, we explore the new area under their $r_{comm}$ using the remaining explorers and the newly created cells are added to the exploration tree as described in section III-E.1. This procedure, starting from finding leaf nodes and determining EW to the addition of cells in $T_0$ (exploration tree), may be repeated multiple times until the $n^{th}$ iteration, when $H_n$ (hinges required in $n^{th}$ iteration) violates (1). Application of this repeated procedure on the considered cell $C_i$ is called as the *Wavefront Expansion of Cell $C_i$*. Once (1) is violated, the RTG algorithm chooses a child node of the node of cell $C_i$ and begins wavefront expansion from there. Since every wavefront expansion process invokes several iterations of computing points on EW and expanding those, this entails unhinging robots from nodes from a previous expansion. This is described in section III-B below.

### B. Recursive Tree Growth

The process of Wavefront Expansion (WE) is executed on each node of a subtree $T_b$ of $T_0$, starting from its root node $C_b$ in a depth-first fashion. This procedure is termed here as *Recursive Tree Growth (RTG)* for $T_b$. We execute the RTG on the exploration tree $T_0$, starting from base node $C_0$ in the first phase of the MHM algorithm. Similarly, in the second phase, this procedure is executed on some subtrees with their root nodes at the exploration frontier. As mentioned earlier, the WE of each node may generate more nodes, which are added to the tree as leaf nodes and hence, the depth-first traversal includes hitherto added nodes as well.

When applying Wavefront Expansion on node $N_j$ we reuse the robots hinged for a previous WE (WE of a higher node). Hence, the nodes which are not the part of the communication backbone to $N_j$ or its children, are unhinged and reused for further expansions. These nodes exist as subtrees of siblings of the nodes in the backbone to $N_j$. This is done as follows:

1) Consider $A$ as the set of nodes forming the hop path (backbone) from base station to $N_j$ (incl. $N_j$).
2) Find another set $B$ which consists of the siblings (in the exploration tree) of each member of $A$.
3) Now, $B$ is the set of the root nodes of all the subtrees whose robots need to be recalled. Hence, employ the tree-breaking algorithm described in III-E.2 on set $B$. This returns all the robots from those subtrees to the cells in the backbone.

### C. Embellishments to RTG

In our implementation, the RTG procedure has been further embellished with a cascade of two procedures for leaf nodes viz. dynamic relaxation of minimum explorers constraint in (1) and use of a SHM when dynamic relaxation doesn't work. We also prune the depth processing of the tree using the fact that if no Exploration Wavefront is available for a tree $T_k$, none of its children would have an exploration wavefront. The details of these embellishments are not discussed due to space constraints.

### D. Baseline Algorithm

This section describes the MHM algorithm using the above mentioned procedures. As the algorithm starts, all the robots are in the role of explorers and are placed within the base-station communication range.The algorithm proceeds as follows:

1) Construct a cell with base-station as its central node and explore the traversable area within the direct communication range of the base station using frontier exploration.
2) Initialize the exploration tree $T_0$ with base cell $C_0$ as its root node.
3) Set the current node to be processed ($C_i$) to $C_0$.
4) Execute Wavefront Exploration on cell $C_i$ and add the resulting cells to the $T_0$ as leaf nodes (as described in section III-E.1).
5) Mark $C_i$ as processed.

6) If an unprocessed child $C_j$ exists for $C_i$, set the current node to be processed ($C_i$) to $C_j$ and repeat from step 4. Otherwise, proceed to next step.
7) If $C_i$ is not same as $C_0$, set $C_i$ to the parent of $C_i$ and repeat step 6. Otherwise, proceed to the next step in order to start the second phase.
8) Find the exploration frontier closest to the base station.
9) Determine the number ($n_h$) and positions (ordered list $L_h$) of nodes required to create a path to the frontier including a node to be hinged just before the frontier (to create cell $C_f$). The list $L_h$ should contain the node positions in the order of hop sequence starting from the base station to the frontier.
10) If $n_h < R_t$, where $R_t$ is the total number of robots, proceed to the next step. Otherwise, jump to step 18.
11) Break all existing sub-trees of $T_0$ (described in III-E.2) and retract any existing node-chains (like tree-breaking since a chain is same as a tree with only one child per node).
12) Extend a node chain to the frontier by sequentially hinging the nodes at the positions specified by the $L_h$.
13) Set the current node to be processed ($C_i$) to $C_f$.
14) Execute Wavefront Exploration on cell $C_i$ and add the resulting cells to the $T_0$ as leaf nodes (as described in section III-E.1).
15) Mark $C_i$ as processed.
16) If an unprocessed child $C_j$ exists for $C_i$, set the current node to be processed ($C_i$) to $C_j$ and repeat from step 14. Otherwise, proceed to next step.
17) If $C_i$ is not same as $C_f$, set $C_i$ to the parent of $C_i$ and repeat step 16. Otherwise, repeat from step 8.
18) Stop execution as the exploration is finished.

### E. Other Procedures

Various other routines mentioned above are described here:

*1) Addition of Cell Nodes to Exploration Tree:* Wavefront Expansion or SHM lead to creation of new cells, that are to be added as leaf nodes in the Exploration Tree ($T_0$). For each of these cells $C_n$, we find the corresponding parent cell in $T_0$ in the manner as follows:

- Find a set $J$ consisting of tree-nodes from $T_0$ such that the hinged-robot corresponding to each tree-node $C_i$ is within communication range of $C_n$.

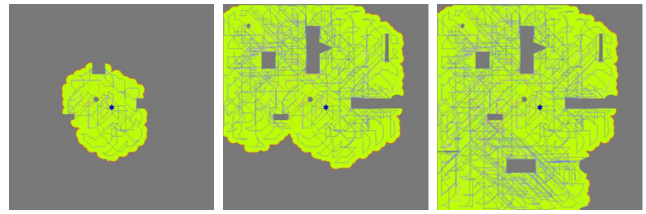
$$J \equiv \{C_i : (dist(C_i, C_n) < r_{comm}) \,|\, (C_i \in T_0)\}, \qquad (2)$$

where $dist(C_x, C_y)$ gives the distance between the hinged robots corresponding to $C_x$ and $C_y$.

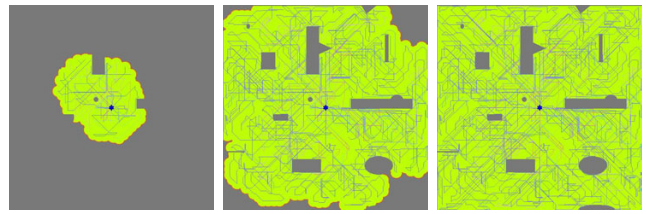
- From the set $J$, find a subset $K$ such that $K$ contains the nodes with the shortest hop path to base station.

$$K \equiv \left\{ \underset{C_j \in J}{\arg\min} (hopcount(C_j, C_0)) \right\}, \qquad (3)$$

where $hopcount(C_x, C_y)$ gives the distance between cells $C_x$ and $C_y$ in terms of hops. The set $K$ will contain one or more nodes, and is defined as the *Parent Candidate Set for $C_n$*.



(a) SHM exploration progress at 2479 simulation ticks (b) SHM exploration progress at 41407 simulation ticks (c) SHM exploration progress at 83942 simulation ticks



(d) MHM exploration progress at 2420 simulation ticks (e) MHM exploration progress at 41362 simulation ticks (f) MHM exploration progress at 82780 simulation ticks

Fig. 4. Progress of SHM and MHM at different exploration time instants

- Pick one of the members of $K$ randomly and assign $C_n$ as its child in $T_0$.

Thus, we assign parents to each of the new cells by iterating the above procedure for each of them.

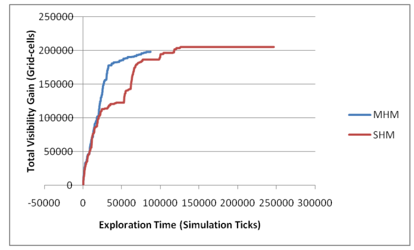
*2) Breaking a tree:* During the expansion of a subtree, robots from other existing subtrees need to be reused for maximal tree growth. The following algorithm provides an efficient unhinging process to break those trees. Consider the set $B$ mentioned in Section III-B, consisting of the root nodes of the trees to be broken. We push this set as root level in a stack. For each hinged cell/node $B_i$ in $B$, we find the children of $B_i$ and create a set $C$ containing the children of all the nodes in $B$. This set $C$ is pushed into the stack too. Now, considering $C$ as the parent set (analogous to $B$), we find a set containing all the children of elements in $C$ and push the resulting set in the stack. This process is repeated until no children can be found for the elements in a set. Now, we can repeatedly pop a set $C$ from the stack, move all the explorers to the parent cell for each element of $C$, and unhinge the corresponding node robot in each element of $C$. This process of popping a set and unhinging its cells is repeated until the stack is empty, i.e. we have unhinged the root nodes of the subtrees to be broken. Thus, by the end of this algorithm, all the robots from the broken subtrees will arrive in the parent cells of the provided root nodes. It should be noted that this procedure only recalls robots (nodes and explorers) from the cells belonging to a given tree. It doesn't modify the exploration tree structure in any way.

## IV. Simulation Results

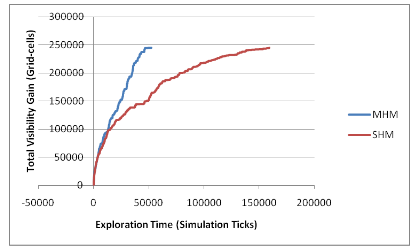
We show simulations on an AMD Turion 64 bit processor running Ubuntu with kernel 2.6.28 with 1 GB RAM. The graphic interface is through QT. Various tests were conducted, involving permutations of different sensor ranges, maps and number of robots. Also, the maps used were of various types (in terms of obstacle configuration such as large/small obstacles, rooms/outdoor scenario, bounded/unbounded, etc.) and sizes. However, for simplicity in explaining the results and due to space constraints, only a subset of bounded maps of same size have been given. The figures 4(a)-4(c) show snapshots of the area in one of the bounded maps explored by the SHM at various instances of exploration time while the figures 4(d)-4(f) show the area explored by the MHM around similar instants. Since the actual time taken by the robots for exploration would depend on the speed of the robot and the size of each grid-cell (1 grid-cell = 1 pixel in map), we consider a grid-cell as a normalized unit of distance and hence, the time taken by a robot to travel the size of a grid-cell as a simulation tick. The exploration ticks are calculated i.e. time for exploration is measured when any of the robots is moving. The improvement is vividly seen through larger areas explored and larger obstacle portions mapped at those same instances due to the current algorithm. The paths traversed by the robots are shown in blue/orange lines, the explored areas in light green and the obstacles and unexplored area in gray. The map used is of size 512x512 pixels. The $r_{comm}$ used for generating the displayed results is 40 pixels and sensor range is 15 pixels. The sensor model is similar to a laser rangefinder. The results in given tables use different number of robots mentioned in each row of the table along with corresponding map used. None of the tables show results for robots more than 20 since the large computation time (order of months) in SHM makes the tests nearly impossible. However, the exploration trend for MHM alone is available for number of robots upto 50 in figure 7, simulated on a bounded map of size 512x512 and same communication/sensor parameters.

Graphs shown in figures 5(a)-5(c) compare the performance of the SHM versus the MHM in terms of the visibility gained per unit time averaged over various runs for different maps with varying obstacle configuration, but same size and different number of robots. For each graph, the map and the number of robots used in the simulation was same for both the approaches and 5 runs were executed to generate the average characteristic values. These graphs plot the total visibility (grid-cells explored) on the ordinate and time in simulation ticks on the abscissa. The blue line has been plotted for MHM while red line represents SHM characteristics. We observe from these graphs that while the area explored by both these methods are almost the same, the MHM explores the same area about 4 times faster than SHM on an average. Some of these maps used for generating the results are shown in figure 6.

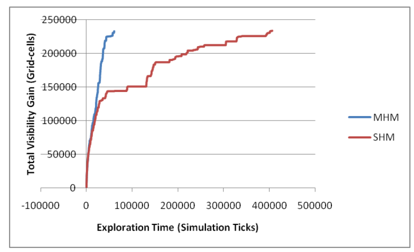
Graph in the Figure 7 shows how the exploration characteristics of MHM vary with number of robots. In the graph, total visibility gained by any instant of time is shown on the ordinate and the exploration time in simulation ticks is shown on the abscissa. In the graph, various plots correspond to different number of robots used, but all of them were obtained for the same map and hence are related for comparative analysis. For the plots with larger number of robots, the average slope is higher, which implies that the




(a) For map o1 with 10 robots

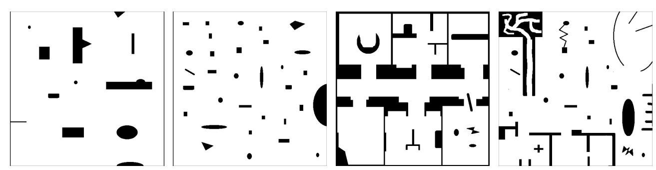



(b) For map u1 with 14 robots




(c) For map v1 with 12 robots

Fig. 5. Visibility-Exploration Time comparison between SHM and MHM



(a) Map u1 (b) Map u3 (c) Map o1 (d) Map v1

Fig. 6. Some sample maps used for testing SHM and MHM implementations

visibility gain per unit exploration time is high and so is the speed of exploration. Thus, a higher number of robots doesn't just increase the spatial coverage, but gives a good boost to the speed of exploration as well.

Table I compares the performance of the SHM versus MHM in terms of computation time used to explore maps such as those in figure 6 on the computers used for simulation. Both the algorithms were simulated on the same computer for each combination of map and robot count. For robot values larger than 16, the SHM tests took too long (order of days) and hence only an estimate is provided for such numbers for comparison with MHM. The remaining times are averaged over various runs and maps and all the resultant times are tabulated in columns 2 and 3 due to SHM and MHM respectively. It was made sure that the map explored in both the methods were same so that all

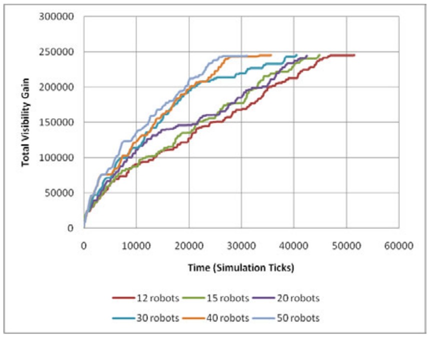



Fig. 7. Trend of variation in MHM's exploration characteristics for different number of robots.

TABLE I
COMPARISON OF SHM AND MHM IN TERMS OF COMPUTATION TIME

| Robots | Map (code) | SHM Computation Time (d:hh:mm:ss) | MHM Computation Time (d:hh:mm:ss) |
|---|---|---|---|
| 10 | o1 | 0:00:43:55 | 0:00:01:59 |
| 10 | u1 | 0:03:46:18 | 0:00:07:11 |
| 10 | v1 | 0:01:19:54 | 0:00:02:16 |
| 11 | u1 | 0:06:37:35 | 0:00:05:27 |
| 12 | o1 | 0:02:46:54 | 0:00:02:07 |
| 13 | o1 | 0:09:47:08 | 0:00:01:16 |
| 17 | u3 | 0:34:09:00 | 0:00:01:03 |

the environment characteristics remain the same for both computations. Once again, the performance gain due to MHM vis-a-vis SHM is evident as the computation times are drastically reduced for MHM and the computation time increment per robot is significantly less than SHM.

Finally, we compare the amount of coverage/exploration reach of both the methods in large maps that are large enough to ensure that they cannot be explored in totality when the number of robots are lesser than an upper bound. The inability to cover the entire area is evidently due to the constraint of retaining communication with the base. Table II compares the performance of both the algorithms in terms of total final visibility, for same parameters (map, robot count, etc.) in each case. The table shows both the methods having almost similar reach in terms of ability to explore or cover an unknown area. They differ essentially in the time to explore an area as well as in the amount of visibility or information gained per unit time.

TABLE II
COMPARISON OF SHM AND MHM IN TERMS OF TOTAL COVERAGE (DETERMINED IN AN UNBOUNDED MAP)

| Robots | Area explored by SHM (Grid-cells) | Area explored by MHM (Grid-cells) |
|---|---|---|
| 10 | 430719 | 423421 |
| 11 | 509272 | 489689 |
| 12 | 599174 | 590780 |

*A video of the simulation demonstrating our implementation of the current method accompanies the paper and may be referred for a better understanding of the algorithm.*

## V. CONCLUSIONS

A novel method of interleaving multi robotic exploration with construction of a tree network that satisfies the constraint that all robots must maintain a communication to the fixed base station is presented. The method involves finding multiple hinge locations at the boundary of the explored area that is within the communication range of at-least one hinged robot and expanding them in a parallel fashion. The number of points that get hinged and the manner in which they get hinged is decided by the recursive tree growing process that invokes within it the Wavefront Expansion module one or more times. Also, the addition of new cells has a new approach of fitting overlapping cells at the boundary. The algorithm shows significant performance gain in terms of time taken for exploration, the visibility or information gain per unit time as well as computational time over a very recent approach that addresses the same problem. Apart from the two approaches there does not seem to be any other approach in the literature that has addressed the problem of multi robot exploration with the base station constraint.

## REFERENCES


[1] Piyoosh Mukhija, Rahul Sawhney and K Madhava Krishna, "Multi Robotic Exploration with Communication Requirement to a Fixed Base Station", AAMAS 2010

[2] W. Burgard, M. Moors, C. Stachniss, and F. Schneider,"Coordinated Multi Robot Exploration", IEEE Transactions on Robotics, 21(3), 2005.

[3] J. Ko, B. Stewart, D. Fox, K. Konolige, and B. Limketkai, " A Practical, Decision-theoretic Approach to Multi-robot Mapping and Exploration", IROS 2003

[4] R. Simmons, D. Apfelbaum, W. Burgard, D. Fox, M. Moors, S. Thrun and Hakan Younes. "Coordination for Multi-Robot Exploration and Mapping", In Proc. of AAAI, 2000.

[5] Rahul Sawhney, K Madhava Krishna and K Srinathan, "On Fast Exploration in 2D and 3D Terrains with Multiple robots", AAMAS 2009 .

[6] Alejandro R. Mosteo*, Luis Montano, and Michail G. Lagoudakis, "Multi-Robot Routing under Limited Communication Range"

[7] Jose Vazquez, Chris Malcolm, "Distributed Multirobot Exploration Maintaining a Mobile Network", 2nd International IEEE Conference on Intelligent Systems, 2004.

[8] Guilherme A. S. Pereira, Aveek K. Dasy, Vijay Kumar and F. M. Campos, "Formation Control with Configuration Space Constraints", in IROS 2003

[9] Martijn N. Rooker and Andreas Birk, "Multi Robot Exploration under the Constraints of Wireless Networking", in Control Engineering Practise Journal, 2007

[10] Martijn N. Rooker and Andreas Birk, "Communicative Exploration with Robot Packs", in Lecture Notes in AI, 2006

[11] Yamauchi, "A Frontier-Based Approach for Autonomous Exploration", IEEE International Symposium on Computational Intelligence in Robotics and Automation, 146-151.

[12] Sameera Poduri and Gaurav S. Sukhatme "Constrained Coverage for MobileSensor Network". ICRA (2004), 165-171

[13] Nidhi Kalra, Dave Ferguson, and Anthony Stentz, "A Generalized Framework for Solving Tightly-coupled Multirobot Planning Problems", in ICRA 2007