On Fast Exploration in 2D and 3D Terrains with Multiple Robots

Rahul Sawhney
rahulsawhney@research.iiit.ac.in

K. Madhava Krishna
mkrishna@iiit.ac.in

Kannan Srinathan
srinathan@iiit.ac.in

International Institute of Information Technology, Hyderabad, India

ABSTRACT

We present a fast multi-robotic exploration methodology for 2D and 3D terrains. An asynchronous exploration strategy is introduced which shows significant improvements over the existing synchronous ones. A per-time visibility metric is being utilized by the algorithm. The metric allots the same weight for points for next view whose visibility over time ratios are equal. The outcome of this is that while the number of points visited to explore a terrain is nearly the same as other popular metrics found in literature, the time length of the paths are smaller in this case resulting in reduced time exploration. The results have been verified through extensive simulations in 2D and 3D. In 2D multiple robots explore unknown terrains that are office like, cluttered, corridor like and various combinations of these. In 3D we consider the case of multiple UAVs exploring a terrain represented as height fields. We introduce a way for calculating expected visibilities and a way of incorporating explored features in the per-time metric. The maximum height of the UAV at each location is governed by the so called exposure surface, beneath which the UAVs are constrained to fly. We also show performance gain of the present metric over others in experiments on a Pioneer 3DX robot.

Categories and Subject Descriptors

I.2.9 Robotics : Multi-Robotics

General Terms

Algorithms

Keywords

Exploration, Multi robots, UAV, 3D, Terrain, Metric

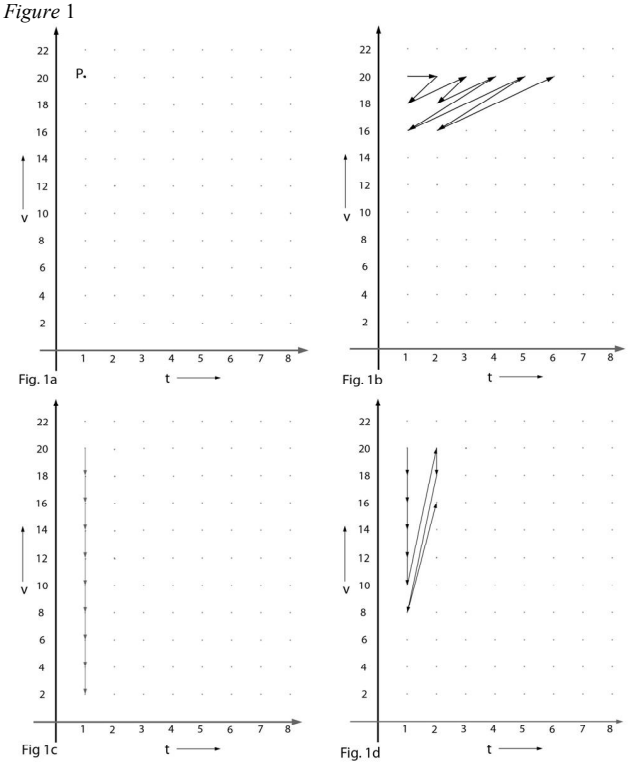
1. INTRODUCTION

The problems of coverage and exploration are of fundamental importance in mobile robotics and, in an eclectic sense, have significance in agent systems in general. While the former generally refers to gathering information from known environments, the later refers to gathering information from unknown ones. Creating map of unknown environments is intrinsic to several robotic applications like reconnaissance, search and rescue, planetary exploration, cleaning, sweeping or mowing. In general, to do so efficiently requires good exploration strategies. In particular, the robots need to ascertain, based on current information, what areas are worthwhile to explore. They also need to apportion exploration tasks effectively amongst themselves, achieving coordination and reducing idle time, in order to map efficiently.

This work aims to provide a fast multi-robot exploration methodology for both 2D and 3D environments. One of the main contributions of this effort is to determine the role of per-time visibility metric for a multi-robotic/agent/UAV exploration problem in 2D and 3D terrains. In [12] and [13], the efficacy of the metric was shown for fast coverage of 3D terrains whose representations are known. A natural extension to that effort would be in sensor based coverage of unknown terrains, which is essentially a terrain exploration problem.

The contributions of the paper apart from the choice of the metric include introducing a superior asynchronous exploration strategy, a new method of computing expected visibility in 3D terrains factoring height information, and an extension of the frontier exploration framework to 3D terrains represented as height fields where the explored terrain features and UAV elevation plays a crucial role in the choice of next point of visit.

Figure 1



Cite as: On Fast Exploration in 2D and 3D Terrains with Multiple Robots, Rahul Sawhney, K. Madhava Krishna, Kannan Srinathan, *Proc. of 8th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2009),* Decker, Sichman, Sierra and Castelfranchi (eds.), May, 10–15, 2009, Budapest, Hungary, pp. 73–80

Our motivation for the choice of per-time visibility metric is characterized through *figs* .1(a)-1(d). The abscissa of the figures represents the time, *t,* to reach the point for next view in an exploration process. The y-axis or the ordinate represents the expected *new* visibility, *V,* upon reaching the point. We define '*V'* for a given point on the terrain as the total number of unexplored terrain points visible from that point. Each dot or point in *fig.* 1(a) is a possible location for a next view with its <*x,y*> location representing the tuple <*t,V*>. For example point '*p'* in *fig.* 1(a) has <1,20> as its <*t,V*> tuple. The dots basically represent the options available for deciding the next point of visit. For a given UAV/robot, the terrain point corresponding to '*p'* would take one time unit to reach and twenty unexplored terrain points are expected to be visible from there. It should be kept in mind that the *t,V* values in *fig.* 1 and their discretized representation are shown as such, only for providing an intuitive understanding of our motivations behind the choice of metric.

The popular exploration metrics such as [1,2] take the form of *V-βt*, where *V* is typically the expected utility of a point as in [2] or expected new visibility as in [1] and *t* is the time taken to reach that point or some variant such as path cost.

For sake of illustrating the idea we choose the top 10 points or dots in *figs.* 1(a)-1(d) in descending order of evaluation of the metric. For *V-βt* type metric with β =2 , *fig*. 1(b) , the top ten points that are chosen are (1,20), (2,20), (1,18), (3,20), (2,18), (1,16), (4,20), (5,20), (2,16), (6,20). The arrows in *fig.* 1(b) show how the search for the next best point proceeds for the first ten points. The average visibility of the first ten points chosen by the *V-2t* metric is 18.8 units, average time is 2.7 units and the average value of the ratio *V/t* is 10.

The first ten points for *V-βt* metric with β =20 results in an average *V* value of 11, *t* value of 1 and the ratio *V/t* is 11. The search for the next best point for the first 10 points is as shown in *fig.* 1(c).

The top ten points as chosen by the per-time visibility or *V* over *t* or the *V/t* metric however results in an average *V* value of 15.2, average *t* value of 1.3 and the ratio *V/t* is 12.5. *Fig.* 1(d) shows how the search proceeds.

As said previously, the dots represent possible points of visit. At any step during exploration, only some of the options would be available out of which the best one, as evaluated by the metric, would be chosen. For most part of the exploration one of the top options would be available. In an extended run, we can expect most of the top options to be chosen roughly equal number of times. Hence, a reason for computing the aforementioned values can be given as follows. If the next point choosing algorithm has exactly one of the top ten points available to choose, each of the 10 ten points having equal chances of being available, the average values of the quantities *V, t, V/t* computed is same as the expected value of these quantities. Thus, with *V-2t* metric we can expect to see possibly 10 new points on average in unit time, while with *V/t* metric we can expect possibly 12.5 new points on average in unit time.

This empirical analysis results in the following observations:

i. For low β values, β = 1, 2, etc, the points chosen have higher values of *V,t.* The points chosen are from top rows in the figures, indicative of high *V* values. The algorithm is dominantly greedy on *V* than on *t.* The *V/t* value turns out to be poor

ii. For high β values, β = 30, 40, etc, points chosen have lower values of *V ,t* They are almost always chosen from the left most columns. The ratio *V/t* is marginally better than low β values. The algorithm is greedy on *t* than *V* and degrades to a closest point first method.

iii. Points chosen based on the *V/t* metric have the highest *V/t* ratio as expected and the algorithm is greedy on *V/t* values. Points chosen generally progress from left most columns initially and then oscillate between top rows and left columns

iv. That no single value of β will always be able to choose the same point as chosen by the *V/t* metric unless β value is changed every time the point for next best view is to be chosen. For example consider two points with <*V,t*> values as <20,1> and <110,11>. The *V/t* metric prefers the point <20,1>. For *V-βt* metric to prefer <20,1> , we require that $\beta > 9$. For the next view, we are given points <100,60> and <10,50>. The *V/t* metric prefers <100,60>. For *V-βt* to also make the same choice one requires that $\beta < 9$ which contradicts the previous entailment of $\beta > 9$.

Our method has been simulated in 2D in various kinds of environments with MobileSim a simulator from Mobile Robotics Inc that takes into account both sensor and process noises. It has also been extensively tested in 3D terrains and on the Pioneer-3DX moving in various kinds of 2D environs. While the above discussion shows points chosen by *V/t* metric is almost always different from other metrics the superior performance of this metric is vindicated in extensive simulations in 2D and 3D worlds. An intuitive explanation for the better performance of this metric is that it captures the notion of maximizing expected new visibility and minimizing travel time better than others through the ratio *V/t.* Its complete independence from any other parameters or constants that needs to be tuned or learned also makes it an appealing choice.

2. RELATED WORK

Frontier exploration became popular in the last decade due to [3] and was further extended to a multi-robotic framework due to the seminal work of [1,2]. While [2] focuses on the gains achieved due to coordination between robots by sharing of map information and reducing the utility of frontier points in the vicinity of an allotted point, [1] came up with an elegant bidding process. Earlier Rekeleitis et.al, [4] covered terrains with multiple robots in a Boustrophedon fashion where at-least one robot was stationary and posed as an observer. In the same paper the authors had opined the need for taking into account the scanning time apart from travel costs. In [5] a decision theoretic approach to multi-robotic exploration was presented where the essential theme was to decide whether a robot should explore the terrain or to verify the hypothesis of other robots whose states are not mapped into a common reference frame. Unlike previous approaches robots started from unknown reference frames and

original contributions were made as to how to localize a robot in the map of another robot. In [6,7] authors present an algorithm for coverage and exploration of a terrain using a sensor network. In a very recent seminal contribution [8], bounds are derived for a minimum views to explore an unknown polygonal terrain.

In contrast the approaches for exploration in 3D terrains are relatively lesser. The notion of coverage of known terrains rather than exploration of the unknown has gained more prominence in 3D worlds. In [9] Eidenbenz presents an approach for minimum number of guards to cover a known 3D terrain. Recently Joho and others present a paper for single robot exploration in 3D worlds [10] through a metric that is analogous to V-βt. This and earlier approaches such as [11] are more focused on generating 3D maps and deal with single robots operating on multiple planes. The degree of freedom along the height axis is almost zero as they are ground robots. Height variations occur in those approaches only due to undulations of the ground plane. Our survey showed no work based on 3D heightmap based multi-robot exploration

The methods of [1,2] are closest to our method in terms of the framework used. Apart from the choice of the metric, our method differs from them in the way frontier allocation is done and the visibility computations are made. Both use a synchronous allocation, while we propose the use of an asynchronous one. While [2] counts all the cells that are unknown within the sensing range of the sensor centered at a frontier cell, we utilize ray tracing to compute the expected visibility. The performance gain due to ray tracing is that it avoids going to dead frontier locations whose views are mostly blocked by obstacles around, one that a simple counting procedure fails to detect. The method of [1] while presenting a method of how to reduce the utility of a frontier cell close to an already existing frontier cell, is silent on how to compute the expected visibility at a frontier location. Apart from these differences, this work extends the exploration framework to 3D terrains.

3. METHODOLOGY

3.1 Formulation

We initially pose the more generic 3D version of the problem and later mention the modifications required for the 2D version.

Given an unknown but bounded terrain T, described as a set of points $T = \{p_1, p_2, \ldots, p_N\}$, where each $p_i = \{x_i, y_i, z_i\}$ with z_i denoting the terrain height at $\{x_i, y_i\}$, and a set $R = \{r_1, r_2, \ldots, r_m\}$ of m UAV, find m paths that completely explore T, such that exploration time is as reduced as possible and the height of the path over any point p_i is constrained to lie beneath the exposure surface at that point. In other words, the height h_j of an UAV, r_j, is such that $h_j \in (z_i, ez_i)$, where $ez_i > z_i$ is the height of the exposure surface, E, at the point p_i. The exposure surface, E , which is known to us a priori, is basically defined as a height field which constrains the UAVs to stay within a certain safe height over the exploration area. The UAVs are constrained to lie beneath an exposure surface to avoid being noticed by hostile observers.

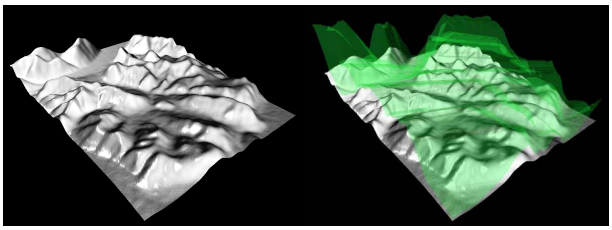


Figure 2 : a) A sample terrain b) With Exposure surface rendered on top

3.2. The Allotment Process

Let M_v be the mapped part of the terrain at an instant v. In 3D, M_v, is prone to be discontinuous and can be represented as a union of such discontinuous regions, $M_v = M_{v1} \cup M_{v2} \cup \ldots \cup M_{vq}; M_v \subset T$. *Fig.* 2 shows a UAV exploring the underlying terrain. It can be seen that the mapping process results in discontinuous explored regions. The mapping strategy is along the lines of occupancy grid [14] and is not explained here due to brevity of space. Let all the UAVs be present in the initially explored sub-map M_{v1}. Then M_{t1} would be the dominant sub-map within which all the UAVs are constrained to stay at any time during exploration. The UAVs cannot move into another sub-map because of the unexplored regions in between. Let $F = \{f_1, f_2, \ldots, f_s\}$ be a set of s candidate frontier points in the dominant sub-map, M_{v1}, at v; the subscript v has been removed henceforth for simplicity. Each frontier is represented as $f_i = \{x_{fi}, y_{fi}, z_{fi}\}$. Here $\{x_{fi}, y_{fi}\} = \{x_i, y_i\}$ are same as the x,y coordinates of the terrain point beneath that was mapped, i.e., $\{x_i, y_i\} \in \{x_i, y_i, z_i\} \subseteq M_1$ while z_{fi} is the height of the frontier point that lies between the exposure surface above and the mapped terrain beneath, i.e., $z_{fi} \in (z_i, ez_i)$.The algorithm first uniformly samples the candidate frontier points in M_{v1} . Then for each such sampled candidate point, the space between the underlying terrain and the exposure surface is sampled again and all the resultant sampled points constitute F .

3.2.1 Synchronous and Asynchronous Allotment

Let t be the time taken by a UAV, r_j, to reach a frontier point f_i. and let V be the expected new visibility, to be gained at f_i .The allocation process then allots a frontier point f_i to a UAV r_j provided the per-time visibility metric, $M_{PTV} = \frac{V}{t}(z_f - z_{av})^{\gamma} z_f^{1-\gamma}$ evaluates to a maximum. In case of Synchronous allocation, the metric should evaluate to maximum for that frontier – UAV pair among all possible pairs. In case of Asynchronous allocation, the metric should evaluate to a maximum, among all possible pairings of only the stopped UAVs with the frontier points. Pseudocode for the two allocation strategies have been provided. Note that the asynchronous algorithm will work even without a central overseer function. Since generally, multiple UAVs will not stop at the same time, the joint frontier allocation function can be replaced by an individual one, without any significant performance degradation.

```
procedure find_Time_Paths_Sync(Terrain T, Outposts O, UAV R)
ListOfFrontiers = FrontiersFromRegionInitiallyExplored

do
UnallocatedUAVs = set of all UAVs R
  while UnallocatedUAVs is not an empty set
        find (e_q, r) = arg max_i [M_PTV(e_i, r)];
            ∀e_i ∈ ListOfFrontiers ; ∀r ∈ UnallocatedUAVs
        add_Point_To_Path (r, e_q)
        reduce_VisibilityValue_Of_All_Points_Expected_
                                              To_Be_Seen (e_q)
        mark_All_Points_Expected_To_Be_Seen (e_q)
        remove UAV r from UnallocatedUAVs
  end while

  wait_For_All_UAVs_To_Reach_Their_Frontiers_And_Take_Scan()
  reset_VisibiltyValue_Of_All_Unseen_Marked_Points()
  unmark_All_Unseen_Marked_Points()
  update_Frontier_List (ListOfFrontiers)

 while (ListOfFrontiers! = null)
 end

procedure find_Time_Paths_ASync(Terrain T, Outposts O, UAV R)
ListOfFrontiers=FrontiersFromRegionInitiallyExplored

do
StopppedUAVs = set of UAVs which have scanned and stopped
  while StopppedUAVs is not an empty set
        find (e_q, r) = arg max_i [M_PTV(e_i, r)];
            ∀e_i ∈ ListOfFrontiers ; ∀r ∈ StopppedUAVs
        add_Point_To_Path (r, e_q)
        reduce_VisibilityValue_Of_All_Points_Expected
                                        _To_Be_Seen(e_q, r)
        mark_All_Points_Expected_To_Be_Seen (e_q, r)
        remove UAV r from StopppedUAVs
 end while

 reachedUAV = wait_Till_Some_UAV_Has_Reached_It's_Frontier
                                        _And_Taken_Scan()
 reincrease_VisibiltyValue_Of_All_Unseen_
                                  Marked_Points( reachedUAV )
 unmark_All_Unseen_Marked_Points( reachedUAV )
 update_Frontier_List (ListOfFrontiers)

while (ListOfFrontiers! = null)
end
```

Here z_f , is the height of the frontier point as detailed before, z_{av} .is the average height of the terrain patch centered at the terrain point $p = \{x, y, z\} \subseteq M_1$. The frontier being considered lies directly above this terrain point at height z_f. Once allotted the frontier UAV pair is removed from the list of possible pairs; the potential visibility of all the points expected to be seen from that frontier is significantly diminished. This ensures that multiple robots do not tend to explore the same unexplored regions.

3.3 Factoring Height Information

The height information is incorporated in the computation of the metric through the $(z_f - z_{av})^{\gamma} z_f^{1-\gamma}$ terms. The first term captures the relative height at which the frontier point is above the underlying terrain. An estimate of the underlying terrain region is calculated by taking average of constant sized patch of explored terrain beneath. Given two competing points with same expected visibility gain, *V,* and frontier height z_f the algorithm chooses that point whose relative height over the terrain beneath is more. Essentially, we expect if a point is at a better height with respect to terrain beneath it is better poised to see a wider surrounding area all other factors remaining the same. Similarly if the absolute height of a frontier point is higher it is expected see more than a point for which all other variables remain the same. These two ideas are captured through the coupling of relative and absolute heights (*fig*. 3). The parameter, γ denotes the relative importance of the two terms serving as a weighing factor. In the results section we show comparative analysis with and without factoring the height information.

Note that if a frontier point has its relative height larger than a certain limit (which is a little less than the scanning range of the UAV's sensor), it is immediately removed from the competition since very little of the terrain can be seen from such a point.

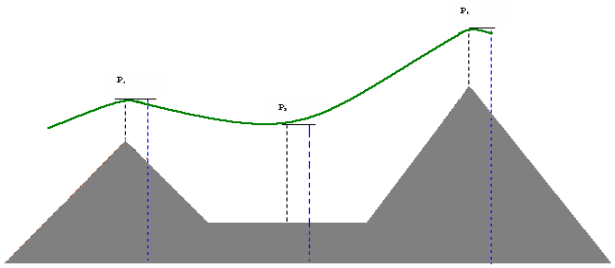


Figure 3 : The blue and black dotted lines indicate the absolute heights and relative heights respectively.

3.4 Computing Expected Visibility Gain

Note that because of a limited scanning range, as one goes closer towards the underlying terrain, a larger spread could be expected. But at the same time, the closer we get to the terrain, more would be the chances of the spread being blocked by surrounding terrain features. For the methodology to work in the desired manner, a good estimate of expected visibility gain from

a given frontier is important. Our method of ascertaining expected visibility involves simulated ray tracing with terrain extrapolation over unexplored regions. In order to calculate expected visibility from a frontier point the unexplored terrain around it is linearly extrapolated as $z_{unexplored} = (1 - (dist / scan\,range))z_{av}$, where *dist* is the distance (on the XY plane) of the frontier point from the unexplored terrain point whose height is being extrapolated. We chose such an extrapolation scheme because of the typical nature of the explored terrain near frontier regions. Usually, the terrain at the frontier regions of main sub-map would be a peak region with the unexplored part typically sloping down. This is because the UAVs are scanning from above the terrain. (*fig. 4*)

Ray tracing is simulated from the prospective frontier point over this extrapolated explored terrain to get an estimate of the set of possibly visible unexplored points, *Spv,* from that frontier. Use of ray tracing is especially useful in 3D, where the explored terrain features will cause significant occlusions. The expected visibility is calculated from the set *Spv* as follows :

$V = \sum_i w_{pi}, \forall p_i \in S_{PV}$. The weight w_{pi} of each unexplored point p_i is diminished significantly , when it's included in the *Spv* set of a UAV. This reduces the likelihood of a frontier allocation which possibly sees a lot of such points. This ensures that the multiple robots don't tend to explore the same unexplored regions. The weight w_{pi} of each unexplored point, which is going to be seen by at least one UAV, is given as $w_{pi} = 1/n_r$, where n_r is the number of robots expected to see that unexplored point based on allotment made to frontiers so far.

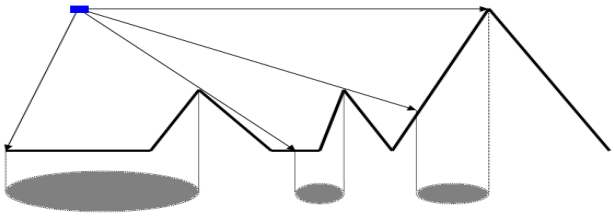


Figure 4 : The blue square indicates the UAV. Due to the underlying terrain features, the region seen by the UAV consists of discontinuous patches of the terrain. The first patch which contains the UAV would be part of the main sub-map.

3.5 Modifying for 2D Terrains

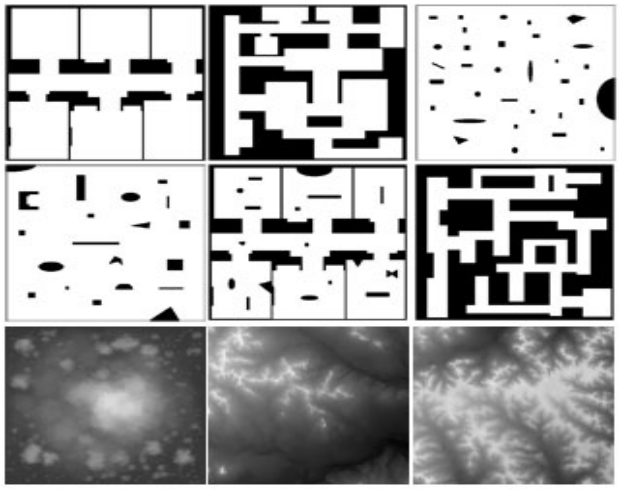
The main modification as for 2D terrains go is in the computation of expected new visibility. Unlike 3D the height information of the frontier point is not considered and neither is there a linear interpolation of the terrain while computing the metric. In other words, M_{PTV} is simply, $M_{PTV} = \frac{V}{t}$, where symbols have the same connotations as before. The expected visibility *V* used in the M_{PTV} metric is computed by ray casting procedures. Omni-directional rays are cast from the selected frontier point and proceed till they meet the closest obstacle or till the sensing limit is reached. All unexplored points that fall in the cast rays are counted to provide the expected visibility at that frontier point. Once an unexplored point is accounted for by a frontier its weight values are diminished in the same manner as explained in section 3.4. Apart from this modification in computing the expected visibility the rest of the algorithm remains identical.

4. SIMULATIONS

The simulations were performed over various classes of terrains, some of which are shown in *fig*.5. 2D simulations were done on several office, corridor and unstructured environments and their hybrids, such as the first six in *fig.* 5. 3D simulations were performed on terrains, represented as heightmaps, with varying degrees of undulations and feature distribution, such as the last three in the figure. The simulations in 2D were performed on MobileSim, a mobile and multi robotic simulator that accounts for both sensor and process noise providing a realistic simulation counterpart for the real world scenario.

For all comparisons that are shown in the subsequent sections the abscissa denotes the number of robots and the ordinate the exploration time averaged over several runs across several terrains. Scanning times have been taken into account in all results and a limited scanning range was assumed.

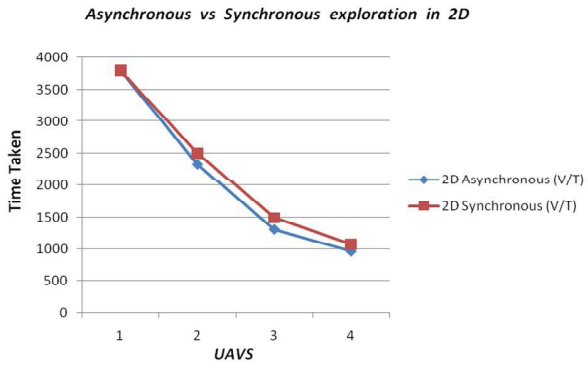
Figure 5 : Simulations were carried out in varied environments

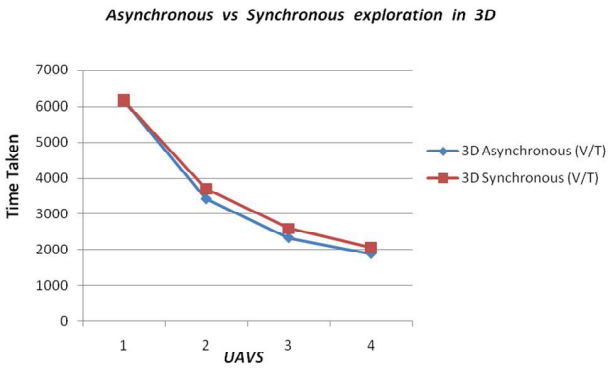


4.1 Asynchronous Vs Synchronous

The following graphs compare the results of asynchronous allotment of frontiers vis-à-vis synchronous allocation with the underlying metric being M_{PTV}. The asynchronous plot is shown in blue while the synchronous in red The performance gain due to asynchronous update was of the order of 10% in 2D and 9% in 3D, on average This is along expected lines since in a synchronous process much time is lost while waiting for all robots to reach their respective frontiers before a new allotment is computed. Similar performance gain was obtained due to asynchronous allotment for the *V-βt* metrics not shown here. Asynchronous approach, apart from being superior in performance, could be more efficient in terms implementation on multi-robot systems. It can be deployed in a decentralized

manner without requiring a central overseer function. Both [1] and [2] use the synchronous allocation mechanism which requires a "central mapper" for allocation and co-ordination. For rest of the comparisons in this paper the asynchronous frontier allotment is the de facto method.

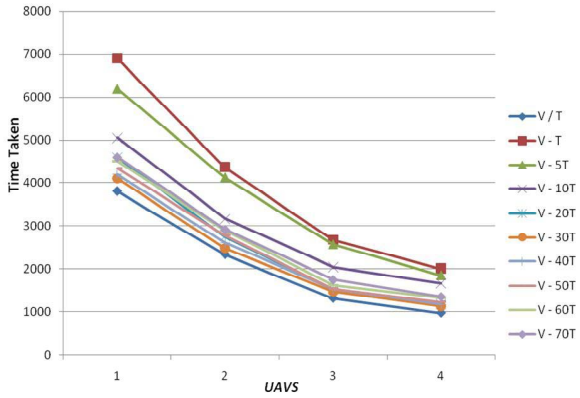




4.2 Simulations in 2D

The graph of *fig. 6* compares the V over t metric with the V-βt for β values ranging from 1 to 70. The color legend is also shown in the same figure. The results were averaged over 160 test runs per metric per agent group.

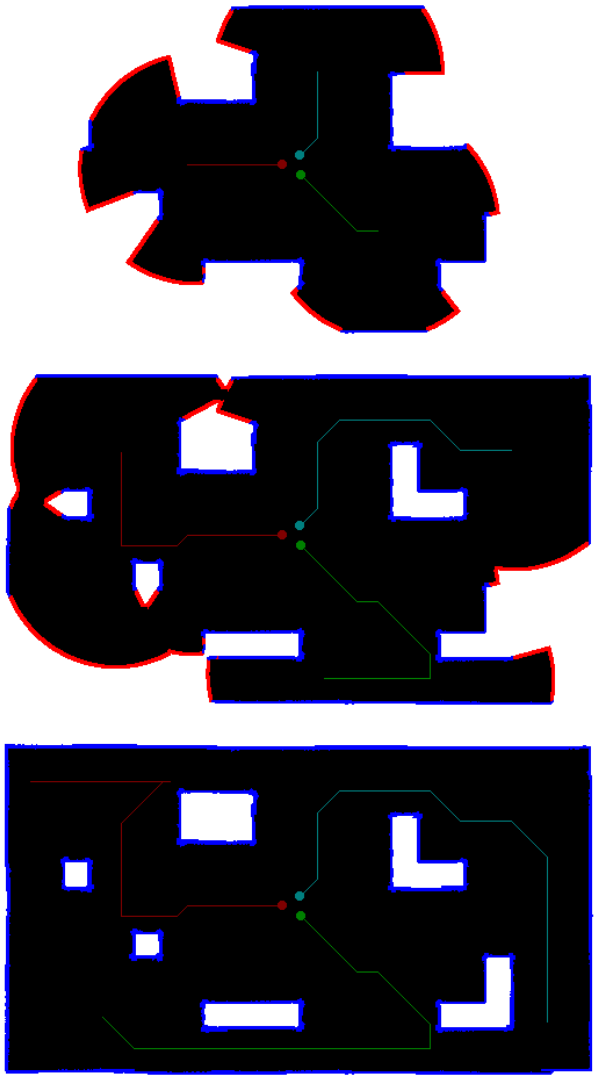
Figure 6 : Comparison of V/t and V- βt metrics in 2D



The V over t metric explored terrains at around 10% faster, on average, than the best performing β metric. β=*30* had the best average performance amongst the tried V-βt metrics. Note that the best βs were arrived upon by an exhaustive search over V-βt metrics and is very dependent on the kind of terrain being explored.

Snapshots of an exploration with 3 robots , using M_{PTV} metric with asynchronous allocation have been shown in *fig.* 7

Figure 7 : Exploration with 3 robots in MobileSim. Frontier points are shown in red, with obstacle and wall boundaries in blue.



4.3 Simulations in 3D

The graph of *fig.* 8 compares the V over t metric with the V-βt for β values ranging from 1 to 70. The color legend is also shown in the same figure. The results were averaged over 120 test runs per metric per agent group. The V over t metric explored terrains at

around 9% faster, on average, than the best performing β metric. β=40 had the best average performance amongst the tried V-βt metrics. Again, note that the best βs were arrived upon by an exhaustive search over V-βt metrics and is very dependent on the kind of terrain being explored.

Snapshots of an exploration with 3 robots, using V over t metric with asynchronous allocation have been shown in *fig.* 9

Figure 8 : Comparison of V/t and V- βt metrics in 3D

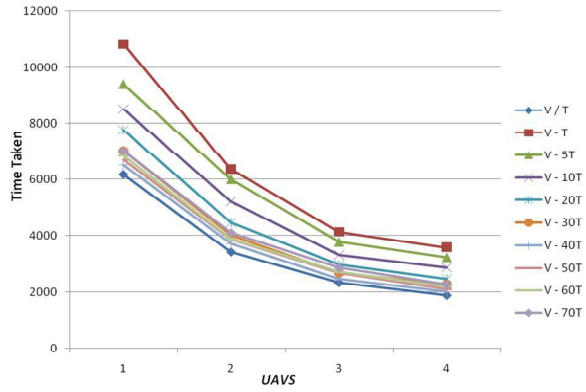
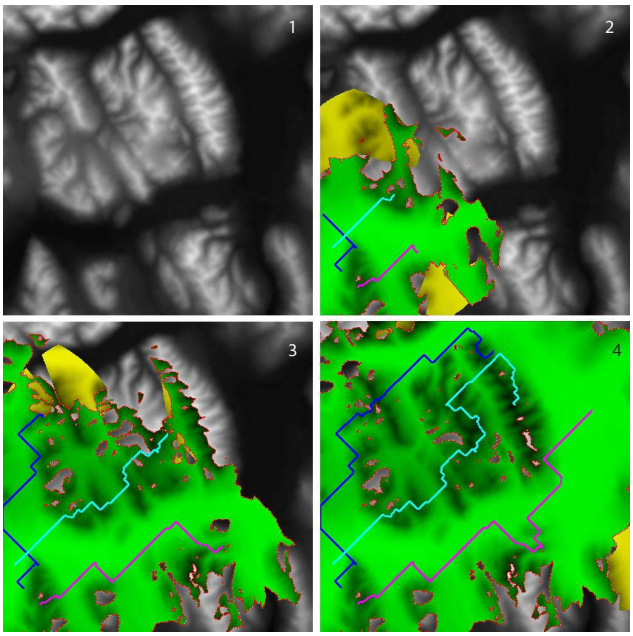
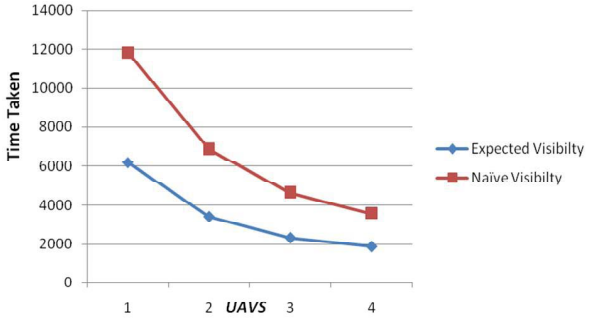


Figure 9 : Progression of the algorithm: the unexplored regions of the terrain are depicted in shades of gray, the explored regions in shades of green. The yellow regions mark the regions of expected visibility. The boundaries between the explored and unexplored regions form the frontiers (marked in red). The lines indicate the paths taken by the individual robots. Note the very discontiguous nature of the explored areas



The performance due to expected new visibility computations mentioned in 3.4 versus a naive method that merely counts all unexplored points in a spherical ball centered at the frontier point as the new expected visibility , is compared in *fig.* 10. The plots due to the current method are in blue while the naive visibility method is in red. The naïve method is actually a 3D extension of [2].

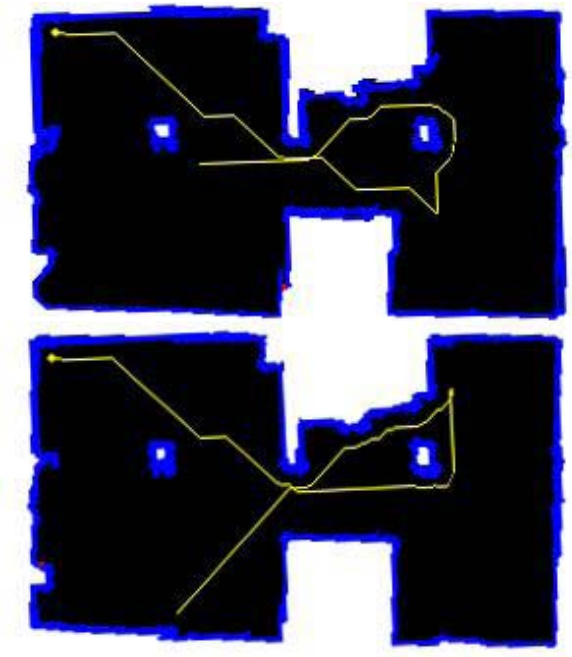
Figure 10 : Comparison of Expected Visibility with naïve method in 3D



5. EXPERIMENTAL RESULTS

Real world exploration experiments in varied environments, using a single Pioneer 3DX robot were also carried out, to verify efficacy of V over t metric. The paths generated by V over t metric and V-βt metric , for β=25 , the best performing β, in a typical room like environment have been shown. V over t performs slightly better in terms of exploration times.

Figure 11: Path generated by V/T (first) and V-25T (second). Path generated by V/T is more compact.



Per metric, 36 experimental runs were carried out. It was observed that the paths generated by V over t were generally more compact than V-βt. V over t exploration times in the environments tested were generally better than V-βt metrics and comparable with the best performing β *metric*; with around 7% improvement on average. The trends were similar to ones shown

in *fig.*6 - single robot. The explorations were carried out on a P3DX with limited laser range. SLAM using scan matching was utilized to do away with localization errors.

Figure 12: The exploration environment for the shown exploration paths

6. CONCLUSIONS

This paper has confirmed the efficacy of the per-time visibility metric for fast terrain exploration in 2D and 3D worlds over and above metrics that subtract time or path cost from visibility or information gain, i.e. metrics of the form V-βt. Comparisons were made in 2D over several classes of terrains such as office like, corridors, unstructured and various combinations of those and in 3D over terrains of varying degrees of undulations. The gain of the V over t metric was around 10%in 2D and 9% in 3D over the most competitive of V-βt metrics when β values varied from 1 to 70. The paper also confirms that the performance of V-βt was best when β values were neither too small nor too large. Small β values result in the algorithm becoming excessively greedy on V tending to chose points far away to avail a high visibility value at a cost of high time costs, while large β values degrade to a closest point first algorithm greedy on t. The V over t metric is independent of such search over parameter values for optimum performance and hence is attractive due to this reason also. Real time experimental comparisons on a Pioneer 3DX further confirm the aptness of this metric.

Moreover the paper has also presented a new method of computing expected visibility in 3D terrains based on terrain interpolation. The performance of the algorithm is much superior when using this method of computing the expected visibility vis-à-vis an algorithm that merely counts all unexplored points in a spherical ball centered at the frontier point. The paper also shows the performance gain in asynchronous allotment of frontiers than with synchronous allotment, since in a synchronous process much time is lost while waiting for all robots to reach their respective frontiers before a new allotment is computed. Asynchronous approach, apart from being superior in performance, is more efficient in terms implementation. It is deployable in a decentralized manner on multi-robot systems.

7. REFERENCES

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

[2] 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.

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

[4] I. M. Rekeleitis, G Dudek and E Millios, " Multi-robot Collaboration for Robust Exploration", *IEEE ICRA*, 2000

[5] D. Fox, J. Ko, K. Konolige, B. Limketkai, D. Schulz, and B. Stewart, "Distributed Multi-robot Exploration and Mapping", *Proc of IEEE*, 2006

[6] M. Batalin and G. S. Sukhatme, (2007), "The Design and Analysis of an Efficient Local Algorithm for Coverage and Exploration Based on Sensor Network Deployment", In *IEEE Transactions on Robotics*, 23(4):661-675, Aug 2007

[7] S. Poduri and G. S. Sukhatme "Constrained Coverage for MobileSensor Network". *IEEE ICRA* (2004), 165—171

[8] S. K. Ghosh, J. W. Burdick, A. Bhattacharya and S. Sarkar, Online Algorithms with Discrete Visibility: Exploring Unknown Polygonal Environments, Special issue on Computational Geometry approaches in Path Planning, *IEEE Robotics & Automation Magazine*, vol. 15, no. 2, pp. 67-76, 2008.

[9] S Eidenbenz, "Approximation Algorithms for Terrain Guarding", Information Processing Letters,82(2002): 99-105

[10] Dominik Joho, Cyrill Stachniss, Patrick Pfaff, and Wolfram Burgard, "Autonomous Exploration for 3D Map Learning", *Autonome Mobile Systeme (AMS)*. Kaiserslautern, Germany, 2007.

[11] Surmann H, N¨uchter A, Hertzberg J, "An autonomous mobile robot with a 3D laser range finder for 3D exploration and digitalization of indoor environments", *Robotics and Autonomous Systems,* 45(3-4):181–198, 2003.

[12] R. Sawhney, K.M. Krishna, K. Srinathan, M. Mohan "On reduced time fault tolerant paths for multiple UAVs covering a hostile terrain". AAMAS (3) : 1171-1174, 2008

[13] M. Mohan, R. Sawhney, K.M. Krishna, K. Srinathan, M.B. Srikanth, "Covering hostile terrains with partial and complete visibilities: On minimum distance paths". IROS : 2572-2577, 2008

[14] A. Elfes. Occupancy Grids: A Probabilistic Framework for Robot Perception and Navi-gation. PhD thesis, ECE Department, Carnegie Mellon University, 1989