2008 IEEE/RSJ International Conference on Intelligent Robots and Systems
Acropolis Convention Center
Nice, France, Sept, 22-26, 2008

Covering Hostile Terrains with Partial and Complete Visibilities: On Minimum Distance Paths

Mahesh Mohan, Rahul Sawhney , K Madhava Krishna, K Srinathan and M B Srikanth

Abstract— **We present a method for finding paths for multiple Unmanned Air Vehicles (UAVs) such that the sum over their lengths is minimum as they cover a 3D terrain (represented as height fields). The paths are constrained to lie beneath an exposure surface to ensure stealth from enemy outposts. The exposure surface is also computed as a height field. The algorithm greedily clusters the terrain such that gain in visibility per distance would be higher for intra-cluster points than points across clusters. Paths generated on clusters formed by such a per distance visibility metric are reduced by more than 25% over other related decoupled methods. The method is extended to cover terrains with partial visibilities. The advantage of the coupled metric extends under constrained visibility also. We again show performance gain by comparing with an existing decoupled algorithm that solves a similar problem of minimum distance terrain coverage with constrained visibility. The paper reveals that decomposing the terrain based on visibility first and then distance is always better than the other way round to cover the terrain in shorter distances.**

I. INTRODUCTION

THE paper attacks the following problem: Given a terrain *T*, represented as a height field and a set *O* of hostile outposts or towers, find paths for multiple UAVs such that, the terrain is covered, the overall sum of path lengths is minimized and none of the UAVs is visible from any outpost.

We then, extend the algorithm to the case where the points in *T* are considered seen only if it is viewed from within a cone; the bisector of the cone is normal to the plane containing the point. If the point is seen from outside the cone it needs to be seen once more, from another viewing angle to be considered as completely seen. The partial visibility constraint allows for more than one sighting of terrain points to compensate for viewing errors and blurred images that occur due to oblique viewing of the terrain.

There have been no known approaches for the problem amongst the literature we surveyed. The original problem of seeking to place a minimum number of guards to cover the entire 3D terrain is NP hard and [10] presents the best approximate version with an upper bound that would be no worse than a logarithmic increase in the number of points over the optimal solution. The problem if attacked based on methods present in literature could take the following two ways. In the first method, we find the number of points required to guard the terrain based on the dominating set (output points of [10]). The traveling salesman problem [8] extended to its multi-agent version would then partition the points output by [10] into clusters through a distance metric. The points in each cluster are visited through an approximation algorithm such that the sum of the path lengths is minimal. In the second method we adapt the traveling tourist [9] to the current scenario. The original set of terrain points is first clustered into the number of salesmen or UAVs based on a distance metric. For each of the clusters the points are visited by an implementation of the traveling tourist.

Both these methods of hierarchically decomposing the terrain through visibility followed by distance or distance followed by visibility produced inferior results compared to the current method of partitioning, that couples both distance and visibility in a single metric. It is noted that, when we seek to maximize visibility along with minimizing distance simultaneously it is possible to visit more guard positions than that output by [10] yet reduce the actual distance traversed. Fig. 1 depicts this motivating example.

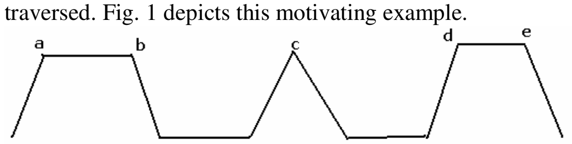

Fig 1. A single dimension terrain with the minimum number of points to cover the terrain given by ***a***, ***c*** and ***e***. If we have two UAV to cover the terrain they can cover through a shorter distance by one of them visiting points ***a*** and ***b*** and other ***d*** and ***e***. If we go purely based on minimum number of points to visit one UAV stays at ***a*** while other needs to visit ***c*** and *e* that takes longer distance.

Further, we show that the coupled decomposition extends its advantages to terrains even when visibility is constrained. We compare the current method with an approach [16] that attacks the problem of minimum distance coverage with constrained visibility and show performance gain. [16] first decomposes based on visibility to compute a set of guards that cover the terrain and further an optimal order to visit those guards. The method presented in [16] is for a single robot and uses a randomized version. To have fair comparisons, we compare paths due to the current method with that of [16] with its randomizations removed.

II. RELATED WORK

Coverage robotics has attracted a lot of attention in the

The first four authors are with the IIIT-Hyderabad. The last author is currently with the Massachusetts Institute of Technology, USA.

last half a decade and the literature is rich with several versions and themes of it. Some have focused on visiting every point in known or unknown maps [1, 2], others have attacked the problem that every point within a terrain is seen or explored [3, 4], while others have deployed sensor networks to maximize coverage over an area [5, 6].

In this paper we attack the problem of finding paths for multiple UAVs such that they cover the terrain with minimum sum over distance paths. Other pertinent approaches [11, 12] attack the problem of stealth navigation in presence of known observers albeit for a 2D terrain for a single robot. In [13] a multi robot approach to the problem was presented. [17] presents a robust (fault tolerant) method for multi-robot coverage in 2D. It however uses a decoupled metric (cellular decomposition) and pertains to exhaustive geographical search (sweeping). There have been approaches that have tackled coverage and formation under constraints for UAV [14, 15] but the visibility information has been neglected or simplified as circles beneath the UAV. In other words they fail to take into account the rich terrain information that comes from a height map or a terrain map.

III. METHODOLOGY

A. Problem definition

Given a terrain T, described as a set of points $T = \{p_1, p_2, \ldots, p_N\}$, where each $p_i = \{x_i, y_i, h_i\}$ with h_i denoting the terrain height at $\{x_i, y_i\}$, a set of outposts $O = \{o_1, \ldots, o_p\}$ and a set $R = \{r_1, r_2, \ldots, r_m\}$ of m UAV, find m paths that cover T, such that sum of the path lengths is as reduced as possible and none of the paths is visible from any $o_i \in O$.

Due to presence of outposts it may not be possible to see every point of T. We first delineate the points that can actually be covered and then see all those with as reduced sum of path lengths as possible. We find the exposure surface E of T as the set of points $E = \{e_1, e_2, \ldots, e_n\}$, where each $e_i = \{x_i, y_i, z_i\}$. E characterizes the maximum height z_i above the given point p_i that an UAV can reach without being seen by any outpost [7]. E, at times, may be composed of isolated regions $E_1, E_2, \ldots, E_q$ such that no point in E_i is connected to any point in E_j unless they pass through a point seen by at-least one outpost. For example, Fig. 2 shows an exposure surface partitioned into multiple isolated regions.

Fig.2 : Partitions in the exposure surface. Due to the presence of the outposts, the exposure surface need not always be contiguous. The green square indicates the outpost. The blue regions are the areas not visible to the outpost. UAV from one blue region cannot move into another blue region without becoming visible to the outpost

If the number of such partitions is more than the number of UAVs, some partitions would not be visited and points seen from them left uncovered. This inability to cover all points seen by E arises only because of the inability of an UAV to move into an isolated region without being watched by an outpost. We assume the UAV takes snapshots at discrete locations to cover the terrain. Piecewise linear paths in 3D are used to simulate UAV motion that can be executed by aerial vehicles like choppers. This may not be kinematically feasible for UAVs that move along continuous curvature paths. This however does not affect the main motivation of this work that by visiting more points the UAVs can cover the terrain in shorter distances and the advantages of a coupled metric over decoupled one.

B. The algorithm

The algorithm itself consists of two phases. In the first step, starting points for the UAVs are determined on a contiguous exposure surface, E, such that they are sufficiently far apart and afford high visibility. In the second step, the paths are built using a resource allocation mechanism. A point becomes a UAV's next point of visit when the gain in visibility for distance traveled is the maximum for that point-UAV pair over all other point-UAV pairs. It is to be noted that the UAVs move on E to cover S_c, the set of terrain points visible from E, $S_c \subset T$. The algorithm is described in Listing 1.

1) The spread out phase:

Let at some instant in this phase, the points of the set $EA = \{e_1, e_2, \ldots, e_p\}$, $EA \subseteq E$ be assigned to one of the UAVs in the set, $R_s = \{r_1, r_2, \ldots, r_s\}$, $R_s \subseteq R$. In other words every UAV in R_s has exactly one point from EA allotted to it. A point $e_e \in E - E_a$ then becomes the starting point for any $r_r \in R - R_s$ provided the factor $S_o = \Delta d^{1-\beta} V^{\beta}$ evaluates to a maximum for e_e. The factor, S_o, is called the spread out factor, V is the amount of terrain visibility at e_e and Δd is the change in length of the path by adding e_e to EA. Let the original length of the path due to points in EA be d_p, then the path due to addition of e_e evaluates to d_e through the edge breaking rule. This rule adds e_e to EA by connecting at one of the endpoints of the path or by breaking one of the p-1 edges, whichever of these possibilities gives the least path length increment. Then, $\Delta d = d_e - d_p$. Distance between any two points on E is computed by moving along E or under it or both, whichever is shortest. Essentially the motivation of this phase is to obtain good guesses from where the UAVs can takeoff. Intuitively we want the UAVs to be far apart and at locations that have large visibility. This is done by first measuring the proximity of a chosen point to the already allotted points by the edge breaking rule, by computing Δd. The points that maximize S_o (proximity along with visibility) are selected and assigned to UAVs as their starting points.

While the performance of the algorithm is dependent on the initial guess of the starting points, the motivation is not. In other words even if the UAVs are constrained to start off from starting points not computed by the algorithm, paths obtained by the coupled metric are invariably shorter than those obtained in a decoupled manner.

2) Resource allocation:

In order to decide where each UAV moves next, we define a metric 'M' to quantitatively describe the feasibility of a point $e_q \in E - EA$ as the next point from which a terrain snapshot is taken. The metric M_{PDV}, (PDV signifying per distance visibility) for any such point, e_q, is defined as:

$$M_{PDV} = \frac{V(e_q)^{\alpha}}{(\Delta d_r(e_q))^{1-\alpha}} \qquad (1)$$

Here $V(e_q)$ is the visibility of point e_q and $\Delta d_r(e_q)$ is the difference in distance due to incorporation of e_q in the path of r. That is, $\Delta d_r(e_q)$ computes the difference in path distances after and before insertion of e_q in the path of r. The insertion itself takes place through the edge breaking rule mentioned in the previous subsection. $V(e_q)$, the visibility of point e_q, is the total number of unseen points in S_c seen by e_q. We allocate the next point to that UAV for which (1) is the maximum among all point-UAV pairs. This is determined as: $(e_q, r_u) = \text{argmax}_{i,j} M_{PDV}(e_i, r_j); \forall e_i \in E - EA, \forall r_j \in R_s$

C. *Choice of the Metric and Parameters:*

The central idea of the metric M_{PDV} is that it should choose terrain points that give as high a gain in visibility for as less a distance traveled. In other words the algorithm seeks points on the terrain that maximizes the visibility gained per distance or per-distance visibility, *PDV*. The term V/d captures the notion of per distance visibility well. As with all objective functions or metrics there are weighing constants. If the objective function or metric is additive the constants are multiplicative while metric that are ratios have constants in their powers. Hence $V^{\alpha}/d^{1-\alpha}$ is a good choice for the metric, with α being the weighing constant, termed the coupling parameter in the current context.

The coupling parameter α determines the amount of relative preference given to visibility over distance traveled. When α=1, the exposure surface is decomposed or clustered based on terrain visibility alone. When α=0 the terrain is clustered based on distance. The effects of α on the results obtained and a suitable value for it are further discussed in the Section IV based on empirical data obtained through extensive testing.

Similarly for the spread out phase the requirement is to have a high visibility and distance. Hence the multiplicative metric ΔdV. The coupling parameter β determines the importance of these factors through the metric $\Delta d^{1-\beta} V^{\beta}$. A good choice of β was found to be $\beta \in [0.2, 0.25]$ based on empirical data. Parameter β facilitates to choose points that are far off from each other as well as possessing high visibility

D. *Incorporating partial visibility*

In this section, we outline the changes to the algorithm that we need to incorporate a restricted viewing angle for the sensors. For this section, we define the following:

1) A point p is said to be normally visible from q if and only if q is in the central cone of p. (i.e. the segment pq should subtend an angle less than $\theta < \theta_0$ with the terrain's normal at p)
2) A point p is said to be partially visible from q if and only if q is in the intermediate cone of p. (i.e. the segment pq should make an angle $\theta_0 < \theta < \theta_{\max}$ with the terrain's normal at p)
3) A point p is considered seen if it has been viewed normally by some point q or if it has been viewed partially by two opposite points a,b (the projection of line segment ab on surface S subtends an angle of more than 120 degrees at p).

The notion of partial visibility is introduced for two reasons, firstly, the sensors should not pick up the surface at too grazing an angle and secondly, the data sensed should provide sufficient detail for inspection. Hence it is justifiable that points that are normally viewed need not be viewed again but points viewed partially may be considered seen if they are viewed partially from two opposite points. The original problem remains unchanged, except for the kind of visibility relations that are considered for terrain coverage.

In the previous section, the visibility of a point was considered to be the number of points to which line of sight visibility could be established. This is modified to include only those points which fall in the viewing cone of the point being viewed. The visibility of a point is considered to be *noOfPointsSeenNormally* + 0.5 **noOfPoints SeenPartially*

procedure findMinLengthPaths($Terrain\ T, Outposts\ O, UAV\ R, \alpha$)

$E =$ generateExposureSurface(T, O)

$EA = \phi$ // *set of alloted points*

$unseenPoints =$ getViewableTerrainPoints(T, E)

$noOfPoinsInPath = 0$

//spread out phase

while ($noOfPointsInPath\ != noOfRobots$)

do

$e_q = \arg\max_i \left[\Delta d_{EA}(e_i)^{1-\beta}\ \Delta V(e_i)^{\beta}\right]; \quad \forall e_i \in E - EA, \forall r_j \in R_s$

$EA = EA + e_q$

$noOfPointsInPath++;$

$unseenPoints = unseenPoints -$ pointsSeenBy(e_q)

end while

//resource allocation phase

while ($unseenPoints\ != null$)

do

$(e_q, r_u) = \arg\max_{i,j}\left[M_{PDV}(e_i, r_j)\right]; \forall e_i \in E - EA, \forall r_j \in R_s$

$\Delta V_{e_q} =$ numberOfUnseenPoints (e_q)

$\Delta d_{e_q} = \arg\min_i \left[\Delta d_r(e_q)\right]; \ \forall r_i \in R_s$

addPointToPath (r_u, e_q)

$unseenPoints = unseenPoints -$ pointsSeenBy(e_q)

end while

end

Listing 1. Pseudocode for the algorithm of section IIIB

The original algorithm remains unchanged except for the computation of $\Delta V(e_q)$. The weight 0.5 assigned to partially visible points is to bias the metric in favor of normally visible points. It also indicates that point needs to be seen from one more viewing direction and hence the weighing factor.

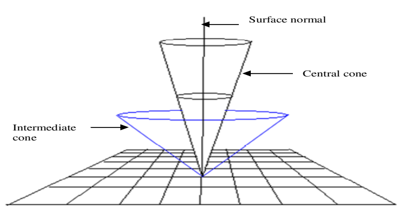

Fig. 3. Visibility cones for a point: observers in the central cone can view the point completely whereas observers in the intermediate cone can only view the point partially. The point is assumed to be invisible to observers outside the intermediate cone.

IV. SIMULATION AND RESULTS

A. The standard case

We have carried out our simulations in 3D environments, although the method is equally advantageous in planar coverage too.

Table 1 compares the current method 'C '(with $\alpha = 0.7$) with two decoupled methods, A and B. As mentioned in section I, method A first finds the minimum number of points required to guard the terrain and visits these guard points by multiple UAVs such that the overall path length is reduced.

Method	No Of UAV	Mean Distance	Std. Dev
A	2	350.13	111.25
B		380.43	109.74
C		271.32	117.64
A	3	290.56	118.02
B		429.34	131.87
C		235.19	115.15
A	4	220.19	133.38
B		450.72	118.51
C		175.92	120.28
A	5	183.67	116.63
B		472.56	138.35
C		147.75	98.1

Table 1: Comparison of the average total path lengths and their standard deviation for different methods and different number of UAV

For method B, the given terrain is partitioned into k clusters (where k is the number of UAVs) based on a distance metric. For each of the clusters the dominating set of points to see terrain points viewable from that cluster, is found from [10]. Paths are computed that visit the points in each cluster, such that the sum of their path lengths is approximately minimal. An approximate version of the Christophedes algorithm [11] is used to compute these paths. For C, the clustering process itself gives the path lengths at the end through the edge breaking procedure. Since the paths are better than those obtained by methods A and B consistently, no path refinement algorithm was used after the clusters were formed.

The first column of the table denotes the method used, the 2nd the number of UAVs. The third reports the average distance taken by the methods to cover a terrain based on simulation runs over 45 terrains. The last column reports the standard deviation of the runs. The terrains were selected such that they had all kinds of features like hills, valleys, ridges, flat surfaces and undulations.

The tabulations clearly show that the current method performs better by giving path lengths that are at-least 25% shorter on an average than method A and by a much larger values with method B. The large standard deviations indicate the high non uniformity in the set of terrains chosen in the runs.

Experimental results show that the performance of the methods A and C generally improves when the number of UAVs is increased whereas it reduces drastically when method B is used. Figure 3 depicts this trend where distances decrease with robots for methods A and C, while it increases for method B. This indicates that method B may not be a suitable approach to solving this problem.

The counterintuitive results of B are attributed to the fact that when terrain is decomposed or clustered first based on distance, the visibility relations between inter terrain points are lost. When clusters are formed based on distance first and visibilities considered only within the cluster, points chosen in another cluster, say c_2, may see several points (sometimes all) points within cluster c_1. Yet in this process one still ends up finding a tourist path for points in c_1 by decomposing it based on visibility. This results in lot more redundancy in visibility among the points visited and this redundancy increases as more clusters are formed or when the number of UAVs increases, resulting in increasing distance.

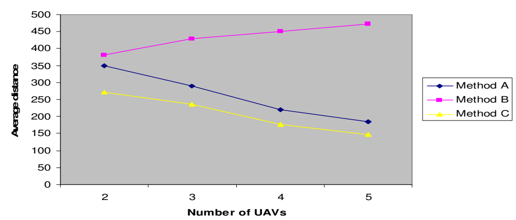

Fig. 4: Behavior of methods as number of UAV is increased. The blue line (circles) is for method A, pink line (squares) for method B and yellow line (triangles) is for method C.

The only increase in computation time over other methods is the use of edge breaking that requires breaking of the edges of the path computed so far to come up with the best path length. This is done every time a point needs to be allocated to a cluster. Due to the competitiveness of edge breaking the path refinement at the end of clustering process is only optional, this is not the case in methods A and B.

B. The effect of α

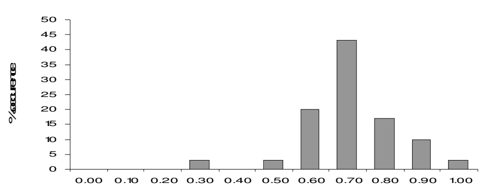

Fig 5. Variation of results with α

Fig. 5 shows how often a given value of alpha performs better than other values of alpha. For example, when $\alpha = 0.7$, it is better than all other values of α, 43% of the time. Clearly the trend in the results suggest that the most suitable value of α is 0.7. It is to be noted that this does not mean $\alpha = 0.7$ is the optimum value. But rather that $\alpha = 0.7$ performs better more often than other values for this set of terrains.

The amount of error introduced by fixing the value of α is described in Fig. 6. Since we have fixed the value of α at 0.7, there might be some cases in which some other value of α would have yielded better results. Fig.6 shows the percentage increase in the distance, caused when some other value of α performs better, and the percentage of the cases it

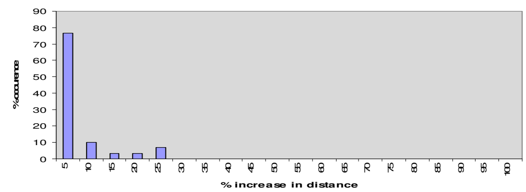

Fig 6. % Increase in distance due to choosing $\alpha = 0.7$ over the α that yielded the least distance

happens. For example, the first bar of the graph shows that the increase in distance is merely between 0% and 5 %, 73% of the time. This justifies our choice of $\alpha = 0.7$.

The optimum value of α is dependent on terrain parameters such as mean and standard deviation of terrain visibility. This underlines the need for a coupled metric, since this degree of coupling is lost in a decoupled search. Finding the optimum value of α for a given terrain is the subject of ongoing research.

C. The effect of starting position

The method is susceptible to start positions of UAVs. Typically the spread out phase gives a good initial guess of the starting points. However when the starting points are provided by the user, the main motivation of this paper still holds ; that is, with arbitrary starting points , paths generated by a coupled decomposition are always better than paths generated by decoupled decomposition. For a separate set of simulations, the coupled decomposition with $\alpha = 0.7$ showed paths lengths reduced by 18 % over method A and more than 40% for method B .

D. The case of partial visibility

When the visibility of the point was constrained (as described in section III), the overall path lengths increased, though the coupled metric still performed better than the decoupled methods. The results are tabulated in Table 2.

Table 2 shows the results for different number of UAVs and their corresponding improvements. Here the comparisons are shown only between methods A and C, since the method B was observed to be an unsuitable approach. The first column of the table denotes the method used, the 2nd the number of UAV. The third reports the average distance taken by the methods to cover a terrain based on simulation runs over 45 terrains.

The case when $\alpha = 1$ and a single UAV corresponds to the method described in [16] with the modification to the manner in which the guards are chosen. In the method A as implemented here, the points (equivalent of guards in [16]) are chosen from a sample set that constitutes all the unchosen points in the exposure surface, whereas in [16] the

Method	No Of UAV	Mean Distance
A	1	1190
C		1047
A	2	900.4
C		796.9
A	3	757
C		697.3
A	4	670.6
C		610.7

Table 2: Comparison of the average total path lengths for different methods and different number of UAV when the visibility is restricted

guards are chosen from a randomly generated subset of all the remaining points. Both in [16] and the adapted version of [16], the method A, the guard is chosen from the sample set greedily.

As the results show, the use of the coupled metric clearly gives an improvement of 10% over the use of decoupled metric. Fig.7 shows a sample set of paths obtained. Imposing the restriction on visibility causes the visibility information to be less rich at any point on the exposure surface leading to reduction in the improvement of coupled method over the decoupled method to 10% from the earlier 25%, nonetheless the advantages of the coupled metric is evident.

V. Conclusion

This paper has attacked the problem of covering a 3D terrain specified as height fields with multiple UAVs with minimum sum over path lengths and with stealth from hostile outposts. The problem has been extended to the case where terrain points are partially visible depending on view angle. The principal motivation is that by visiting more points than the number output by the best terrain guarding algorithms one can cover the terrain with shorter path lengths. A metric that couples distance and visibility captures this motivation. The path lengths obtained with this metric were shorter by at-least 25% than the methods that use a decoupled metric. Further this work reveals if hierarchical clustering or decomposition of the terrain is resorted, it is better to decompose first based on visibility and then on distance rather than the other way round. Unlike these methods that need to hierarchically decompose the terrain based on visibility followed by distance or vice versa the current method using the coupled metric does a one time decomposition of the terrain.

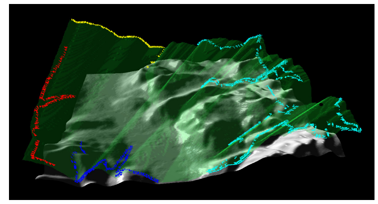

Fig. 7. Paths obtained when partial visibility is considered. Outward projecting surface normals at the terrain edges cause the generated paths to be close to the edges. The restriction on visibility causes paths to zigzag over the terrain's features to cover it.

Further the advantage of the coupled metric over the decoupled methods was seen for cases where the terrain data set was restricted by visibility constraints. This confirms that the advantages of coupled decomposition can be generalized

References

[1] N Hazon, F Mieli and G Kaminka, "Towards Robust Online Multi-Robot Coverage", *Proc ICRA,* 2006, pp: 1710-1715

[2] H Choset, "Coverage for Robots- a Survey of Recent Results", *Annals of Math and AI*, 31:113-126, 2001

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

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

[5] M.A. Batalin and G.S. Sukhatme. Spreading out: A local approach to multi-robot coverage. In *Proc. of the 6th Internat. Symposium on Distributed Autonomous Robotic Systems*, page 373382, 2002

[6] Sameera Poduri and Gaurav S. Sukhatme "Constrained Coverage for Mobile Sensor Network". *IEEE ICRA (2004)*, 165—171

[7] M Mohan, K M Krishna and M Srikkanth, "Increasing Coverage and Preserving Covertness for UAV Moving in Undulated Terrain", *Proc of IICAI-2007, to appear*

[8] N. Christofides, "Worst-case analysis of a new heuristic for the travelling salesman problem", *Report 388*, Graduate School of Industrial Administration, Carnegie Mellon University, 1976

[9] S. Guha and S. Khuller, "Approximation Algorithms for Connected Dominating Sets", *Algorithmica,* 20(4), 1998.

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

[11] M. S. Marzouqi and R. A. Jarvis. "New visibility-based path-planning approach for covert robotic navigation*". Robotica,* volume 24 of Issue 6, pages 759.773. Springer, 2006.

[12] M. S. Marzouqi and R. A. Jarvis, "Covert Robotics: Covert Path Planning in Unknown Environments"

[13] A D Tews, G Sukhatme and M Mataric, "A Multi Robot Approach to Stealth Navigation in Presence of Obsevers", *Proc of ICRA,* 2004

[14] Patrick Vincent, Izhak Rubin: A Framework and Analysis Of Cooperative Search Using UAV Swarms, *2004 ACM Symposium on Applied Computing*

[15] Randal Beard, Timothy McLain: Multiple UAV Cooperative Search Under Collision Avoidance And Communication Constraints, *2003 IEEE Conference on Decision and Control.*

[16] T Danner and L Kavaraki, "Randomized Planning for Short Inspection Paths", *ICRA 2000*

[17] N Hazon and G Kaminka, "Redundancy, Efficiency and Robustness in Multi-robot Coverage", *Proc. ICRA*, 2005,pp:735-741