

Multi-Robot Mobility Enhanced Hop-Count Based Localization in Ad-Hoc Networks

Terence Chung Hsin SIT¹, Zheng LIU^{2,3}, Marcelo H. ANG Jr.¹, and Winston Khoon Guan SEAH³

¹Department of Mechanical Engineering, ²Department of Electrical & Computer Engineering,
National University of Singapore

³Institute for Infocomm Research,
Agency for Science, Technology and Research, Singapore.

Abstract

The localization problem is important in mobile robots and wireless sensor network and has been studied for many years. Among many localization methods, the hop-count based approach is simple and scalable; however, the localization accuracy is not satisfactory if the node density is low. To solve this problem, in this paper a multi-robot approach is proposed to utilize the cooperation and mobility of the robots to improve the node distribution (density), thus enhancing the hop-count based localization. By an auction based task allocation scheme, the robots can negotiate with the static sensor nodes and then select the most suitable robots to move to the area of sparse node density, thus increasing the localization accuracy for the static sensor nodes. On the other hand, the robots also can localize themselves with the help of the static sensor nodes. The efficacy of this approach is shown by simulation.

Keywords

Cooperative robotics, auction based task allocation, wireless sensor networks, ad-hoc networks, hop-count based localization

1. Introduction

In robotics research, it is important and necessary to know the location of the mobile robots with respect to the environment or the global coordinate for many kinds of applications, such as exploration and map building, distributed sensing, pattern formation and marching, and robot soccer. For instance, in Parker's approach of Cooperative Multi-robot Observation of Multiple Moving Targets (CMOMMT, [1]), one critical assumption is that the robots are able to localize themselves with respect to a global coordinate system; without this location information, robots cannot effectively cooperate to work. From the perspective of wireless sensor networks research, localization is also meaningful in applications of data aggregation, velocity estimation, and geographic-aware routing. For example, to achieve better routing with less overhead, Ko and Vaidya [2] propose the Location Aided Routing (LAR) algorithm that utilizes the location information to limit the search space for route discovery; therefore the power consumption and overhead are greatly reduced.

Due to the sensor limitation, environment confinement, and cost constraints, it is difficult to achieve satisfactory localization for both mobile robots and wireless sensor networks. In robotics, usually two classes of localization schemes are used. One class is to get the position via positioning systems/equipments, such as the Global Positioning System (GPS). However, positioning systems are usually expensive and the implementation is constrained by the environment, e.g., GPS cannot work in the regions without the satellite coverage. Another class is to get the position relying on the processing of the data obtained through sensing or intercommunications, e.g., SLAM (Simultaneous Localization and Map building). However, such methods usually require complex computation (in sensor data processing and matching) and the location estimation may take a long time to "converge" to the real location gradually. Furthermore, because this kind of approach usually uses Kalman filter [3] or particle filter [4] to predict and refine the location estimation, if the robot team is large or the environment is complex, the state space (for Kalman filter) or the number of particles (for particle filter) will be excessively large and the whole system may not work appropriately. Therefore, the application of this approach is normally limited to small

robot groups or small environment. In addition, the traditional localization methodologies are highly dependent on the quality of the sensors. To improve the localization accuracy, the system designers have to select accurate but expensive sensors. Obviously, this is also not applicable for large robot groups or mass sensor networks.

For wireless sensor networks, especially the ad-hoc networks, the localization is also a hard problem due to the constraints in form factor (small size), bandwidth, power, and cost. In current research, the most well known localization method is the multilateration, which can estimate the location of the nodes based on its distance to the neighbors and the neighbors' location [5]. Such algorithms need to know the distance among the sensor nodes; however, for mass sensor networks, it is difficult to get acceptable distance information by the simple ranging sensors or the received signal strength index (RSSI). To solve this problem, Approx Point-in-Triangulation (APIT) is proposed to reduce the reliance on the information of the distance among neighbors [6]. However, this approach is still not range free in that it requires knowing the changes in range.

In recent years, the hop-count based localization scheme is proposed and studied for infrastructure-less wireless ad-hoc networks [7]. This localization scheme can be simply implemented in ad-hoc networks without adding extra sensors or equipments. However, if the nodes are sparsely distributed in the networks, the location estimation may be very coarse because the hop count is inaccurate. To address this problem, we propose a robotics approach that deploys multiple robots to increase the node density of the nodes in sparse areas in the network to improve the localization precision. On the other hand, from the view of the robots, the robots in the networks can also locate themselves using the hop-count based localization. The incorporation of robotics to ad-hoc networks is beneficial to both.

In this paper, the hop-count based localization method and its constraints are introduced in Section 2. Then, our multi-robot "intelligent" mobility enhanced localization approach is presented in Section 3. Following which, the simulation results are shown in Section 4. Finally, Section 5 concludes this paper.

2. Hop-count based Localization

In wireless ad-hoc networks, one node can only directly "talk" with its neighbors within one hop range. If the destination node is beyond this range, the communication packets have to be ferried by intermediate nodes through multi-hops. Based on this communication methodology, the hop count (number of hops) between two nodes is a metric to represent the distance between them. This is the basis for hop-count (connectivity) based localization.

To apply hop-count based localization in ad-hoc networks [8], two assumptions should be satisfied. One is that the position of some nodes in the network is known, and they serve as the reference/anchor nodes who broadcast their position throughout the entire network; the other is that the "dumb" nodes, which need to learn their position, can find the shortest links connecting to at least three reference nodes through one hop or multi-hops. Based on these two assumptions, the dumb nodes, by finding the shortest path (minimal number of hops) to the reference nodes, can roughly estimate the distance to the reference nodes based on the hop count; and then, since the position of the reference nodes is known, the dumb nodes can estimate their position by triangulation [9]. In theory, the localization error of the hop-count based localization is in the same order of the magnitude of the hop distance. For many applications that do not have stringent accuracy requirements, such accuracy is acceptable.

Comparing to other localization methods, the hop-count based localization only needs the hop-count information, which is a by-product of the routing protocols for wireless ad-hoc sensor networks, e.g., Ad-hoc On Demand Distance Vector Routing [10]. Therefore, it normally does not incur much communication overheads, and is scalable for mass sensor networks. The most well known hop-count based localization algorithms are DV-Hop [11] and Hop-TERRAIN [12] algorithms.

For the hop-count based localization, one main drawback is that the location estimation accuracy depends largely on the node density of the network. If the node density is low, e.g., one node has very few neighbors, the link between dumb nodes (to be localized) and reference nodes may take more hops compared with

dense networks [13]. This problem is illustrated in Fig. 1, where the arrows indicate the shortest link (the connection with lowest number of hops) from node A to B. If the node density is high (1-a), node A can easily find a “straight” path to node B; however, if the node density is low (1-b), the shortest link from node A to node B has to be “indirect”. Therefore, although the real distance between node A and B are the same in both scenarios, the hop count between them is different. Obviously, the overestimated hop count number in the sparse networks may lead to large error in distance estimation.

To address this problem, two classes of approaches are proposed. One is to adjust the distance estimation according to the node density. For example, DV-Hop algorithm [11] refines the location estimation by adjusting the hop distance according to the hop-counts among reference nodes; density-aware hop-count based localization (DHL) [14] adjusts the hop distance of the node according to its number of neighbors. By these means, if the network is sparse and the link between nodes takes more hops, the hop distance can be reduced and thus the errors in distance estimation are reduced. This class of methods usually generates large communication overhead and cannot perform well in non-uniform networks.

Another class of approaches to address the problem with regard to sparse networks is to increase the node density of the sparse area. Lim and Rao introduce mobile nodes to the traditional static ad-hoc networks [15]. The idea is to let multiple robots move randomly, thus may occupy the node sparse area and then serve as the intermediate nodes to “bridge” the static nodes with shorter path. This approach is shown through simulation to be an improvement over traditional, static, hop-count based localization methods. However, random mobility usually takes an excessively long time for the mobile robots to move to the demanding locations where nodes are sparse; in other words, the performance gains from random mobility are not reliable or predictable. In our research, we plan to address this problem by enabling the mobile robots to move to the most demanding area intelligently. The details are introduced in the next section.

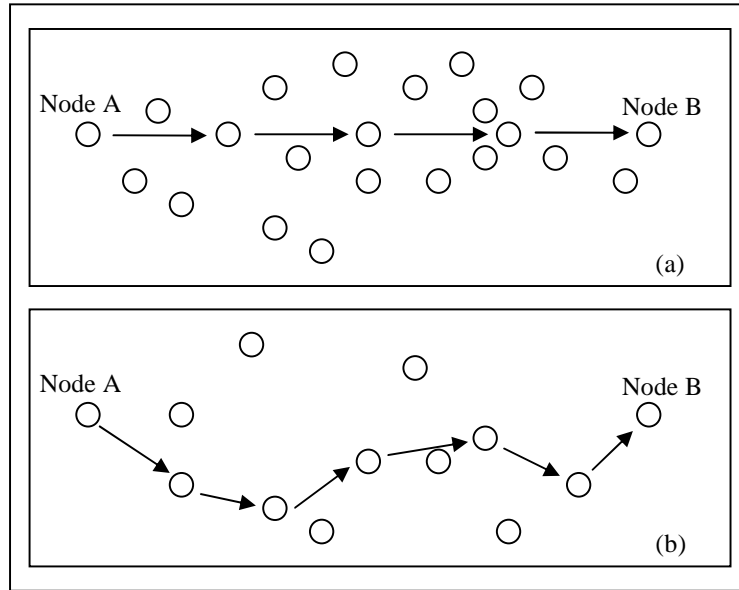


Fig. 1. Same Distance, Different Hop Count

3. Our “Intelligent” Approach

To enhance the hop-count based localization, our research aims to enable the mobile robots to move to the node sparse areas; thereby increasing the node density to improve the localization precision. In this section, we present our approach by introducing the environment, main research issues, and our solutions.

3.1 Environment and Assumptions

A simple diagram of the task environment is shown in Fig. 2. The locations of the static and mobile nodes are unknown.

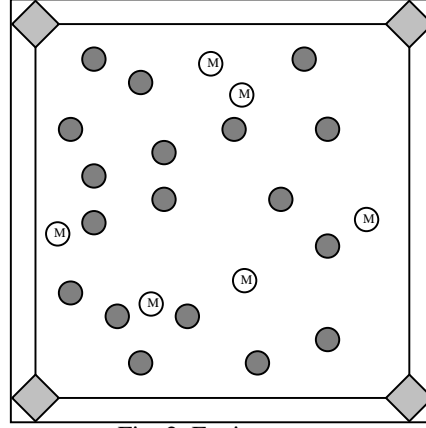


Fig. 2. Environment

(\diamond : reference node; \bullet : static node; \odot : mobile/robot node)

The following are the characteristics of the environment and the mobile robots:

- The environment is a large bounded area.
- There are static nodes scattered throughout the environment. Several (at least 3) of them are the reference/anchor nodes that have accurate knowledge of their positions in the environment. This location information is broadcast to the entire network.
- A team of mobile nodes (robots) can move in the environment. The robot team can be heterogeneous in functionality.
- Static nodes and mobile nodes are distinct. They are able to communicate by wireless ad-hoc communications.
- For non-reference/anchor nodes and mobile nodes, no absolute positioning means (such as GPS) are available.
- The mobile nodes consist of simple robot with sensors, e.g., the range sensor to detect the obstacles and static nodes. When a static node is within the sensor range of the robot, the robot can detect and identify the ID of the static node.
- The static nodes and mobile nodes can fail at any time.
- In this network, the multi-hop network communication protocol and the Media Access Control (MAC) ([16, 17]) can enable the nodes to find the shortest path (link with lowest hop number) to communicate.

In addition, we make the following assumption:

- The nodes (both static and mobile) are fully connected without any partitioning: one node can communicate with any other node through one hop or multi-hop wireless link.

3.2 Main Research Issues

The main purpose of our research is to enable the mobile robots to move to the most demanding area, thus increasing the node density to improve the localization precision for the hop-count based localization scheme. This is an integration of the capabilities of ad-hoc networks with cooperative robotics, so as to achieve efficiency and accuracy not previously achievable using localization methods from ad-hoc networks or robotics alone.

Essentially, this is a task allocation problem, in which the movements to specific locations are the tasks to be performed, and the resources in question are the mobile robots. In general, there are mainly four research issues that need to be examined as follows:

- Where are the most desirable locations to move to? – where to move
- Once these locations are identified, how should the robots cooperate in order to move to these locations? In other words, which robot(s) is most suitable to move to the desired area? – who to move
- Considering a robot that has been allocated a location to move to, how should the robot move to that location? – how to move
- What if the robot fails? How should the algorithm recover from failure? – failure recovery

3.2.1 Where to Move

Considering the main research issues, the first important point is to identify the locations requiring robots to move to. As outlined in previous research work, the accuracy of hop-count based localization is affected by the node density in the network: areas with low node density are more likely to generate localization errors. At this juncture, it is meaningful to introduce an additional assumption: the number of neighbors of a node is an indicator of the local node density of that area. This is a reasonable assumption as a node with few neighbors within direct communications range would suggest that area is sparse in node density. In our approach, the robots are going to move to the static nodes with few neighbors. To achieve this, we propose an auction based task allocation scheme to assign robots to move to the sparse area. Regarding the auction, the static nodes with insufficient numbers of neighbors (and hence likely in low node density regions) will become the auctioneer who issues a move request, retrieves the bid from the robot(s), and then assigns the robot(s) to move. Comparing to the MURDOCH algorithm, which is an auction based task allocation framework “built upon a principled, resource centric, publish/subscribe communication model” [18], our approach is unique in the following aspects:

- In our approach, the auctioneer to initiate tasks is a static sensor node, and not a robot. In most real applications, the number of static sensors is much larger than the number of robots; obviously, it is easier to let the static nodes initiate tasks because they have higher chances to find the tasks. For example, we can deploy many cheap static thermal sensors to monitor the temperature of the room and if there is a fire, the sensor can call robots for help.
- In our approach, the negotiation among robots and static nodes is through the ad-hoc communications, which let the information transmitted hop-by-hop. Based on this communication methodology, it is easy to limit the communications within a sub-region of the environment by setting the maximal communication hops. This can reduce the communication overheads, and not affect the system performance because in many cases only the robots near to the task area should be involved.

In the environment, all the static nodes in the node sparse area have very few neighbors. Obviously, they will all require the robots to approach them. In our approach, there is only one active auctioneer at a time. Hence there is a need to select which static node to be the auctioneer. Since the node with fewest neighbors is likely to have the largest localization estimation error, it is given the priority to become the auctioneer.

Each static node makes a 1-hop broadcast to its neighbors. All nodes will therefore know the number of 1-hop neighbors it has. If a node has very few neighbors (i.e., less than “least neighbor threshold”), it will broadcast this information to the entire network. All nodes will therefore know nodes that need help and can determine which node has the least number of neighbors. That node then becomes the auctioneer. The details of this distributed algorithm are shown in Algorithm 1. Each static node executes this algorithm.

Algorithm 1. Selection of Auctioneer. This algorithm is executed for each static node i .

1. Initialize all values: Let N_i = number of neighbors of node i , L = ID of the static node with lowest number of neighbors = 0, N_L = number of neighbors of node L = 0
2. Broadcast $\langle i, \text{“hello”} \rangle$ to one hop neighbors
3. Keep listening for a while (several times more than communications time)
 - If it receives one hop neighbor’s broadcast during this time, e.g., $\langle j, \text{“hello”} \rangle$, $N_i = N_i + 1$
4. Check the value of N_i
 - If $N_i \geq \text{least-neighbor-threshold}$ (e.g., 6), wait until a signal is received before going to Step 1.

- (Waiting means i is in node dense area, no need participate in the auction)
5. Let $L = i$; $N_L = N_i$
 6. Broadcast $\langle i, N_i \rangle$ throughout the entire network
 7. Keep listening for a while (several times more than communications time)
 - Every time it receives a message, e.g., $\langle m, N_m \rangle$,
 - If $N_m < N_L$, $L = m$; $N_L = N_m$ (choose the node with the lower number of neighbors)
 - If $N_m = N_L$ and $m < L$, $L = m$; $N_L = N_m$ (if two nodes have same number of neighbors, choose the one with lower ID)
 8. If $L = i$, then node i becomes the auctioneer, and the auction starts
 Else, wait until a signal is received before going to Step 1.

In this distributed auctioneer selection algorithm, the static nodes need to be coordinated to broadcast and wait. A node is randomly selected as a coordinator at the start. (For example, it can be node 0.) This coordinator starts the selection process according to Algorithm 1, and sends the triggers required (Steps 4 and 8). When a node is selected as the auctioneer, it will request robots to approach. This node (auctioneer) will also serve as the coordinator to select the next auctioneer. Since the static node does not have absolute positioning sensors or equipments, the location information of this node is estimated by hop-count based localization.

3.2.2 Who to Move

Knowing where to move, the next issue is to choose the most suited robot(s) to move to the area. In the proposed auction scheme, the auctioneer will broadcast the task (its own position) throughout the entire network. The robot who receives this message will give a bid (robot's position and speed) if it is free (i.e., the robot is not executing the task assigned by the previous auctioneers). Then, the auctioneer, in order to minimize the time that the robot(s) may spend in moving to the locations, chooses the most suitable robot(s) to do the task by awarding a contract. For this purpose, a metric using both position and speed is designed and used to select the most suitable robot(s): the nearest robot(s) with the highest speed is (are) awarded the contract (i.e., assigned the task to move towards the auctioneer). Details of the auction are presented in Algorithms 2 (from point of view of auctioneer) and 3 (from point of view of robot). It should be noted that in the auction, since the static nodes and mobile robots do not have absolute positioning sensors, their location information is estimated by hop-count based localization.

Algorithm 2. Auction on Auctioneer's Side (for the selected static node which is the auctioneer)

1. Estimate its own location $\langle x_i, y_i \rangle$ by hop-count based localization
2. Set $RobotPool$ = number of robots already recruited = 0;
 R = number of robots needed = $least_neighbor_threshold - N_i$
3. Broadcast $\langle i, x_i, y_i, \text{"auction request"} \rangle$ to all robots
4. Keep listening for a while
 - If it receives a reply from robot r during this time, e.g., $\langle r, B_r \rangle$
 (r is the ID of robot, B_r is the bid offered by robot r)
 - Record this information in a queue Q , in which the robots are ordered by the value of bid
 (The highest is at the beginning).
 - $RobotPool = RobotPool + 1$.
5. Compare R and $RobotPool$
 - If $R > RobotPool$, available robots are not enough,
 - Unicast a contract $\langle i, x_i, y_i, \text{"contract"} \rangle$ to each of the robots in Q .
 - Let $R = R - RobotPool$
 - Goto Step 3.
 - If $R \leq RobotPool$,
 - Unicast contracts $\langle i, x_i, y_i, \text{"contract"} \rangle$ to each of the first R robots in Q
6. Finish auction

(Note that B_r is calculated by the robot giving the bid using Algorithm 3.)

Algorithm 3. Auction on Robots' Side (e.g., robot r)

1. Estimate the location $\langle x_r, y_r \rangle$ by hop-count based localization
2. Keep listening for a while
 - If it receives an auction request from the auctioneer, e.g., $\langle i, x_i, y_i, \text{"auction request"} \rangle$
 - Reply a bid as $\langle r, B_r \rangle$. $B_r = 1/\text{dist}(\langle x_i, y_i \rangle, \langle x_r, y_r \rangle) + \text{speed}(r)$
3. Keep listening for a while
 - If it receives contract from the auctioneer, e.g., $\langle i, x_i, y_i, \text{"contract"} \rangle$
 - Set status as busy
 - Else goto step 1.

When the auctioneer has assigned the tasks to the selected robots, it will keep monitoring its neighborhood. If the required robots have arrived and thus the density of the neighborhood (number of neighbors) reaches the least-neighbor-threshold, the auctioneer will release the recruited robots by cancelling the contracts, and this static node will not “call for help” (hold new auctions) in future. The freed robots can participate in the next auction. (A robot that is working on a task will not participate in new auctions until it is freed.)

Now we compare our approach with a well-known task allocation scheme for robots in sensor networks proposed by Batalin and Sukhatme [19, 20]. Our approach is different in following aspects:

- In our task allocation scheme, at any time, only one single static node holds the auction. This auctioneer will send the task information to the robots and then allocate the task based on the bid received. In [19, 20], all the static nodes in the sensor networks will set the navigation field as a landmark (motion direction suggestion) for the robots in the vicinity.
- For our auction based task allocation scheme, two auctions cannot be held together, the static nodes will hold the auction sequentially, according to the priority (inverse of the number of neighbors). In [19, 20], the tasks can be allocated in parallel. The robots will choose the task based on the suggestions of the nearby static nodes.

We note that the main difference in our auction based task allocation is that in our work, the static node is the master that controls the robots, while in Batalin and Sukhatme's work, the robots are the masters that only ask for suggestions from static nodes and they can even violate the suggestions based on their own consideration. Since our research objective is to enhance the hop-count based localization by deploying the robots to node sparse area, it is more suitable to let the static node (indicator of node sparse area) control the robots.

In addition, our approach may generate less communication and computation overhead because only the static nodes involved with the task will perform actions (auction); other static nodes are free and their job is merely delivering messages. The drawback is that the system cannot assign multiple tasks simultaneously; the tasks have to be queued and auctioned one by one. On the other hand, Batalin and Sukhatme's approach requires the participation of all the static nodes. Even if a static node is far away from the task and has no robot neighbor, it still needs to calculate the task requirement. However, this approach can simultaneously assign multiple tasks to the robots. In our task scenario of mobility enhanced localization, there are more tasks (static nodes needing robots) than resources (robots); in this case, the ability of multi-task concurrent assignment may not be meaningful because even though a robot can receive many task enquiries, it can only perform one task at a time.

3.2.3 How to Move

When a robot is assigned a target position, it needs to move towards the target with obstacle avoidance based on its local sensing and communications. One simple solution is to use the Artificial Potential Field (APF) method [21]. Essentially, the APF approach models the goal and obstacles (neighboring robot) as sources for attractive and repulsive forces, respectively. The vector sum of these forces is then used to determine the motion of the robot.

In our approach, when a robot is assigned a task (target area), the location of the target area becomes the attractive force for the robot. It should be noted that both target location and robot location are estimates based on the localization scheme.

3.2.4 Failure Recovery

The final main research issue concerns fault tolerance. Owing to the assumption that the robots may fail at any time, as well as the possibility that the robots may fail to find a path towards the goal due to obstacles along the way or the local minimum problem inherent in APF methods [22], there is thus a need to consider task failure on the part of the robot. To address this, contract renewal is used, whereby the assigned task expires in a preset time interval. If a robot cannot reach the target in the set time interval, or loses communications with the auctioneer of the task, it is deemed to have failed in its task. Then, this robot will give up current task and participate in the next auction. On the auctioneer's side, the static node which issued the move request will restart another auction to assign the work to other robot(s) to replace this failed one(s).

A further consideration is that the static nodes may also fail at any point in time. This may, for example, be in the form of communications failure. A possible problem with this might be that robots allocated to a task by the auctioneer (a static node with communications failure) may not be freed from the task if the static node fails to free it from the contract. This can be dealt with through a similar implementation of a "time out" for the mobile nodes. That is, if the mobile node has not been freed from the task after a sufficiently long time period, the robot will assume static node failure, and then free itself to participate in the next auction.

3.2.5 Localization

Both static and mobile nodes need to estimate their positions to enable the "intelligent" mobility. In auction based task allocation, the auctioneer needs to broadcast its position to the robots and on the other hand, the robots need to give a bid containing their location information. In addition, to move to the target area, the robots also need to know the position of the target.

In our work, the reference nodes know their respective positions and broadcast their positions to the nodes within the communications range. This information is then propagated throughout the entire network since we assume that the network is fully connected. By the method introduced in [11], the average hop distance can be calculated and together with the hop count to the reference nodes, the node can roughly estimate its distance to the reference nodes. Since the locations of the reference nodes are known, a node's position can then be calculated by triangulation.

For the triangulation in each "dumb" node or robot, the estimated distance to the reference nodes and the location of the reference nodes are the input. Obviously, the error in distance estimation will lead to the error in location estimation. However, the trend of these two kinds of errors may not be consistent. If the errors of distance estimation are of the same nature, i.e., overestimate or underestimate, they may cancel themselves out in the triangulation in a "least-squares-error sense", and finally the localization errors may be small. For example, consider one node at the center of the environment and the four reference nodes are at the corners which are away from this node with the same distance. If there are errors in the estimation of the distance from this node to the four reference nodes, and the errors are almost the same, no matter how large the errors are, the location estimation of this node (by triangulation) is always nearly at the center of the environment. Therefore, in this paper, we use both distance and location estimation error to evaluate the performance of the localization algorithms.

3.2.6 Communications

In ad-hoc networks, some kind of multi-hop network communication protocol and the Media Access Control (MAC) can enable the nodes to find the shortest link (lowest hop count) to communicate [16, 17]. Since the focus of this paper is on task allocation, we do not concern much on the problems associated with transmissions. In our approach, we assume the entire network can update the information periodically with reasonable transmission delay.

4. Simulation Results and Discussion

4.1 Environment and Settings

To justify the efficacy of our “intelligent” motion approach, we benchmark it against the “random” motion approach [15]. The simulation environment and settings are as follows:

- The simulator: Webots [23].
- Simulation environment: 4.5x4.5m square area including 4 reference nodes, 30 static nodes, and 10 mobile robots. The robots, reference nodes, and static nodes are represented by circles with radius less than 0.1m. This is a reduced scenario simulating a museum that has many sensors and some security guards.
- In the environment, the reference nodes are placed at the 4 corners; the static nodes are placed as shown in Fig. 3. The initial position of the robots is random. The density of the static node in the environment is $1.48/m^2$. This is actually sparse sensor network because the communication range and sensor range of the robots and static nodes are quite small. If the distribution of the static nodes is uniform, in average one static node has only 1.35 static node neighbors.
- The communications radius of each node is 0.8m.
- The update interval of the entire network by wireless communication is 1.6 second.
- The result is the average of 50 episodes. Each simulation episode is 10000 steps long. 1 simulation step is about 0.16 second in real world.
- For our “intelligent” motion approach. Different *least-neighbor-thresholds* of “dense” are tested. For example, when the threshold is 6, if a static node has 2 neighbors, it will require 4 robots to approach; however, if the static node has 7 neighbors, it will not require any robot to come. The thresholds we tested are 4, 6, and 8 for the thirty static nodes.

4.2 Results and Discussions

As mentioned previously in 3.2.5, the error in distance estimation will affect the resultant location estimation during the triangulation process, but the trend of these two kinds of errors may not be consistent. Therefore, we use both distance and location estimation error to evaluate the performance of the localization algorithms. Besides, since the robots are moving and their real location is always changing, we only use the estimation error of the static nodes to evaluate the performance of the localization algorithms.

For each dumb node, the distance error is the average of the differences (absolute values) between estimated distance and real distance to the four reference nodes; the location error is the distance between estimated location and its real location. In our simulation, we use the average of the 30 dumb nodes’ distance error and location error to evaluate the system performance.

Fig. 4 shows the average distance and location estimation errors in the end of the simulation. We can find that intelligent mobility (threshold = 4, 6) achieves better localization performance than random mobility as we expected. By the auctions, the robots can proactively and intelligently move to the most demanding area (node sparse area) to serve as “bridges” to shorten the link between static nodes and reference nodes. Figs. 5 and 6 show the decreasing of distance/location estimation errors during the simulation (when threshold = 6). The intelligent mobility can reduce the error faster than the random mobility. Observing the behavior of the robots in the simulation, we also find that in intelligent mobility mode, when several robots are assigned the same target node to approach to, they usually move in a group. This motion pattern can increase the chance to construct the “bridge” between static nodes and reference nodes.

Besides, we also find that different threshold of “dense” results in different performance. As shown in Fig. 4, the threshold as 4 (threshold and static node ratio is 0.13) generates the best result. With the increasing of threshold number, the performance becomes worse, and when the threshold is 8, our intelligent mobility is even worse than the random mobility mode. The possible explanation is that if the threshold is too large, the static nodes will require more robots to approach. This may slow down the progress because more time is needed to wait for the last robot to come before the target static node can release the robots to work for the next task. Furthermore, when the neighbor number is above a threshold, e.g., “magic number = 6” [24], increasing neighbor number may not greatly improve the localization precision.

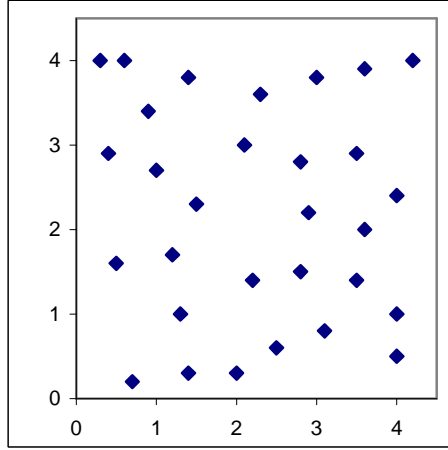


Fig. 3. Distribution of the Static Nodes

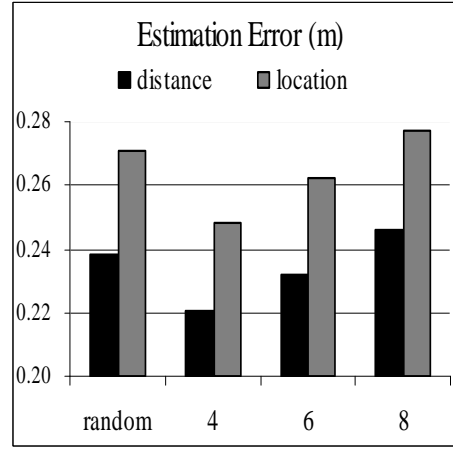


Fig. 4. Distance and Location Estimation Error

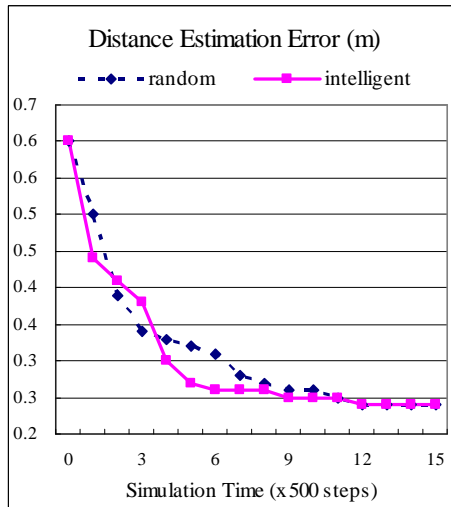


Fig. 5. Distance Estimation Error vs. Time

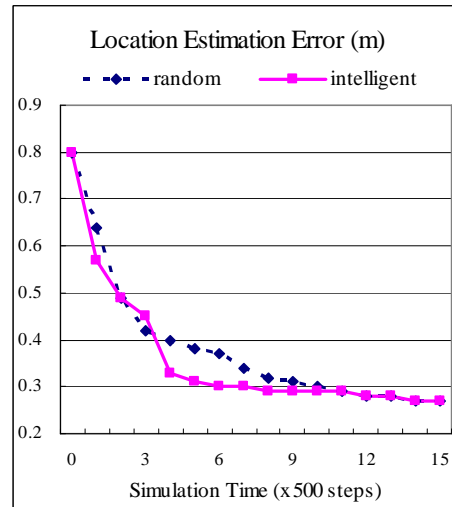


Fig. 6. Location Estimation Error vs. Time

Another significant difference between random and intelligent mobility is in the performance fluctuations. This can be seen from Figs. 7 and 8, which show the distance and localization error for each approach after 10000 simulation steps for each simulation episode. Random mobility produces much more variation in performance than intelligent mobility across the 50 runs of the simulation. These significant fluctuations in performance between each run for random mobility are evident from both performance perspectives of distance error and localization error. The standard deviation of the distance/location estimation errors of the 50 runs in the last simulation step is shown in Fig. 9. This shows that the intelligent mobility mode can achieve more repeatable and consistent estimations. The lack of repeatability and consistency for random mobility can be attributed to the fact that the mobile robots move in random, unpredictable directions, and

hence the improvement in localization information depends largely on chance in that the robots happen to move to locations with low node density. In contrast, the performance gains through the use of intelligent mobility are much more repeatable and consistent, due to the fact that the mobile robots know the approximate locations of the areas of low node density, and thus moving purposely towards those locations to improve localization accuracy.

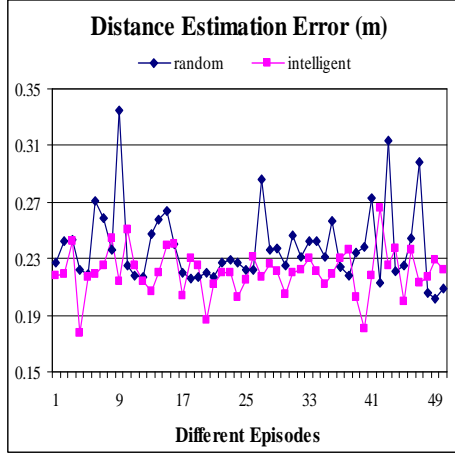


Fig. 7. Distance Estimation Error

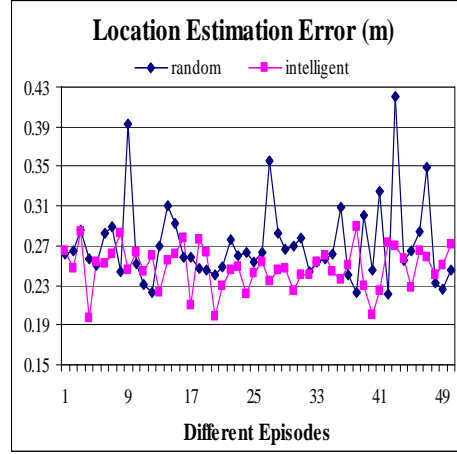


Fig. 8. Location Estimation Error

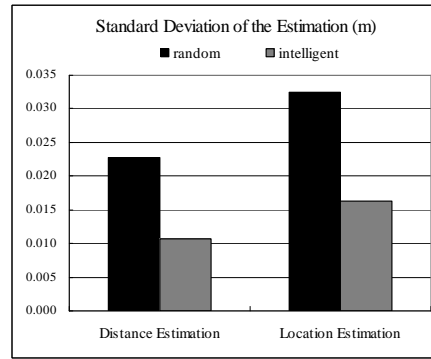


Fig. 9. Standard Deviation of the Estimation

In this paper, we only compare our intelligent mobility approach to the random mobility approach proposed by Lim and Rao [15]. To the best of our knowledge, there are no other algorithms that address improved hop-count based localization using mobility.

5. Conclusion and Future Work

In this paper, an intelligent multi-robot approach is proposed with the aim to improve the accuracy of hop-count based localization in a wireless ad-hoc network. This algorithm is proposed with due consideration to efficiency and fault tolerance. The core of this algorithm lies in the auction based task allocation algorithm with no fixed auctioneer such that any node in the network can “call for help” through self discovery of its surrounding node density.

Through simulations, it is shown that this new intelligent mobility model out-performs the current model of random mobility, both in terms of distance error and location error. The intelligent mobility model is also found to produce more consistent and repeatable results as compared to random mobility. To further test

the efficacy of this approach, we are planning to implement it in real robots and wireless sensor networks, as a part of TARANTLAS project [25].

In this paper, the threshold of the minimum number of neighbors for the surrounding area of a node to be considered sufficiently dense is also investigated. However, it is yet unknown how to determine the optimal threshold. Intuitively, this is a function of the environment size, number of nodes in the area, and the communications radius of each node. In our future research, we plan to test more scenarios to find a method to determine the optimal threshold of “dense”.

In our current approach, we have not put much emphasis on the communication issues such as transmission throughput, contention, or route failure recovery. We only simulate the update interval (delay) in our approach. To further validate our algorithms, we need to make the assumptions on communications more realistic in our future research.

Acknowledgment

We are grateful to Joo Ghee Lim and Kian Hsiang Low for the numerous discussion and feedback on the ideas presented in this paper.

References

- [1] L. E. Parker, Distributed Algorithms for Multi-Robot Observation of Multiple Moving Targets, *Autonomous Robots* (2002)12-3.
- [2] Y. B. Ko and N. H. Vaidya, Location-Aided Routing (LAR) in Mobile Ad Hoc Networks, *Wireless Networks* 6 (2000) pp. 307-321.
- [3] S. Thrun, Y. Liu, D. Koller, A.Y. Ng, Z. Ghahramani, and H. Durrant Whyte, Simultaneous Localization and Mapping with Sparse Extended Information Filters, *Robotics Research* (2004) 23: pp693-716.
- [4] A. Howard, Multi-robot simultaneous localization and mapping using particle filters, in: *Proceedings of Robotics Science and Systems*, Jun 2005.
- [5] A. Savvides, H. Park, and M. B. Srivastava, The n-Hop Multilateration Primitive for Node Localization Problems, *Mobile Networks and Applications* (2003), Vol. 8, pp443-451.
- [6] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. Abdelzaher, Range-free localization schemes for large scale sensor networks, in: *Proceedings of MOBICOM*, 2003.
- [7] M. Frodigh, P. Johansson, and P. Larsson, Wireless ad hoc networking: the art of networking without a network, in: *Ericsson Review*, No. 4, 2000.
- [8] R. Bischoff and R. Wattenhofer, Analyzing connectivity-based multi-hop ad-hoc positioning, in: *Proceedings of the 2nd IEEE PerCom Conference*, Florida, USA, 2004.
- [9] J. J. Caffery, A new approach to the geometry of TOA location, in: *Proceedings of IEEE Vehicular Technology Conference*, Massachusetts, USA, 2000.
- [10] C. E. Perkins and E. M. Royer, Ad-hoc on demand distance vector routing, in: *Proceedings of 2nd IEEE Workshop on Mobile Computing Systems and Applications (WMCSA)*, 1999.
- [11] D. Niculescu and B. Nath, DV based Positioning in Ad Hoc Networks, *Telecommunication Systems* (2003) 22: 1-4, pp. 267-280.
- [12] C. Savarese, J. Rabaey, and K. Langendoen, Robust positioning algorithms for distributed ad-hoc wireless sensor networks, in: *Proceedings of General Track: 2002 USENIX Annual Technical Conference*, 2002, pp.317-327.
- [13] D. Niculescu and B. Nath, Ad hoc positioning system (APS), in: *Proceedings of GLOBECOM*, San Antonio, USA, 2001.
- [14] S. Y. Wong, J. G. Lim, S.V. Rao, and Winston K.G. Seah, Hop-count localization with density and path length awareness in non-uniform wireless sensor networks, in: *Proceedings of IEEE Wireless Communications and Networking Conference 2005*, New Orleans, LA, USA, Mar 13-17, 2005.
- [15] J. G. Lim and S. V. Rao, Mobility-enhanced positioning in ad hoc networks, in: *Proceedings of IEEE Wireless Communications and Networking Conference 2003*, New Orleans, USA, 2003.
- [16] J. Kulik, W. R. Heinzelman, and H. Balakrishnan, Negotiation-based protocols for disseminating information in wireless sensor networks, in: *Proceedings of 5th ACM/IEEE Mobicom Conference*, Seattle, USA, 1999.
- [17] S. Singh and C. S. Raghavendra, PAMAS – Power Aware Multi-Access Protocol with Signalling for Ad Hoc Networks, *ACM Computer Communications Review* (July 1998).
- [18] B. P. Gerkey and M. J. Mataric, Sold! Auction Methods for Multirobot Coordination, *IEEE Transactions on Robotics and Automation* (October 2002) Vol. 18, No.5, pp.758-768.
- [19] M. A. Batalin and G. S. Sukhatme, Sensor network-based multi-robot task allocation, in: *Proceedings of IEEE/RSJ International Conference on Intelligent Robotics and Systems (IROS)*, 2003.
- [20] M. A. Batalin and G. S. Sukhatme, Using a sensor network for distributed multi-robot task allocation, in: *Proceedings of IEEE International Conference on Robotics and Automation*, New Orleans, USA, 2004.
- [21] B. H. Krogh and C. E. Thorpe, Integrated path planning and dynamic steering control for autonomous vehicles, in: *Proceedings of IEEE International Conference on Robotics and Automation*, San Francisco, USA, 1986, pp. 1664-1669.
- [22] Y. Koren and J. Borenstein, Potential field methods and their inherent limitations for mobile robot navigation, in: *Proceedings of IEEE International Conference on Robotics and Automation*, California, USA, April 7-12, 1991.
- [23] Cyberbotics, <http://www.cyberbotics.com/>
- [24] L. Kleinrock and J.A. Silvester, Optimum transmission radii for packet radio networks or why six is a magic number, in: *Proceedings of IEEE Nat. Telecomm*, 1978.

[25] TARANTULAS project, <http://tarantulas.i2r.a-star.edu.sg/>

Biography



Terence Chung Hsin SIT received the B.Eng. degree in the Department of Mechanical and Production Engineering from the National University of Singapore in 2004.



Zheng LIU received the B.Eng. degree in Department of Automation from Tsinghua University, Beijing, China, in 2001. Currently, he is a Ph.D. student in Department of Electrical & Computer Engineering, National University of Singapore. His research focuses on multi-robot systems including multi-robot exploration, searching, target tracking, mobile sensor networks, and multi-robot concurrent learning.



Marcelo H. Ang, Jr. received the B.S. degrees (Cum Laude) in Mechanical Engineering and Industrial Management Engineering from the De La Salle University, Manila, Philippines, in 1981; the M.S. degree in Mechanical Engineering from the University of Hawaii at Manoa, Honolulu, Hawaii, in 1985; and the M.S. and Ph.D. degrees in Electrical Engineering from the University of Rochester, Rochester, New York, in 1986 and 1988, respectively. His work experience include heading the Technical Training Division of Intel's Assembly and Test Facility in the Philippines, research positions at the East West Center in Hawaii and at the Massachusetts Institute of Technology, and a faculty position as an Assistant Professor of Electrical Engineering at the University of Rochester, New York. In 1989, Dr. Ang joined the Department of Mechanical Engineering of the National University of Singapore, where he is currently an Associate Professor. In addition to academic and research activities, he is actively involved in the Singapore Robotic Games as its founding chairman. His research interests span the areas of robotics, mechatronics, automation, computer control, and applications of intelligent systems methodologies.



Winston Khoon Guan SEAH received the Dr.Eng. degree from Kyoto University, Kyoto, Japan, in 1997. He is a Senior Scientist in the Network Technology Department of the Institute for Infocomm Research (I2R). Prior to I2R, he had been a Principal Member of Technical Staff, and director of the Internet Technologies programme in the Institute for Communications Research. He is also an Adjunct Associate Professor in the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. Concurrently, he is an adjunct faculty in the Graduate School for Integrative Science and Engineering, and the Department of Computer Science in the National University of Singapore. He is actively involved in research and development in the areas of mobile ad hoc and sensor networks, co-developed one of the first quality of service (QoS) models for mobile ad hoc networks (MANET), and has given keynote talks on the area of QoS for MANETs. His latest research focuses on mobility-enhanced protocols and algorithms for C3 and

sensing applications in terrestrial and oceanographic ad hoc sensor networks. He is also on the TPC of numerous conferences and reviewer of papers for many key journals and conferences in the area of MANET and sensor networks. He serves on the Steering Committee of the Asia-Pacific IPv6 Task Force, and is also an associate of the Singapore Advanced Research and Education Network (SingAREN.) He is a Senior Member of the IEEE. [Homepage: <http://www1.i2r.a-star.edu.sg/~winston>]