

Interfering Mobile Target Motion Planning in Wireless Sensor Networks

Jia-Liang Lu^{†‡}, Zhengzheng Xu[†], Lionel Croix[‡], Anis Darwich[‡], Min-You Wu[†]

[†]Dept. of Computer Science & Engineering, Shanghai JiaoTong University, Shanghai, China

[‡]Universite de Lyon, INSA-Lyon, CITI, F-69621, Villeurbanne, France

[†]{jialiang.lu, zhengzheng, mwu}@sjtu.edu.cn, [‡]{lionel.croix, anis.darwich}@insa-lyon.fr,

Abstract—In surveillance related Wireless Sensor Networks (WSNs) applications, mobile targets can utilize intelligent motion planning in robotics to avoid detection, which could impose great threats on the sensor field. In this paper, we propose countermeasures against mobile target motion planning. The proposed solution enables the sensors to interfere the localization and map generation on mobile targets so as to expose them in the sensor field. We compare the behaviors of a mobile target in naive mode, smart mode with Simultaneous Location And Mapping (SLAM) and interfered mode where sensor nodes use different puzzle algorithms to disturb it. The simulation results show that the puzzle algorithms can interfere SLAM on mobile targets and enhance the target detection performance in a WSN. We also build an experimental testbed with Iris motes and a mobile robot to validate our solution.

I. INTRODUCTION

Target detection applications built on Wireless Sensor Networks (WSNs) are envisioned to ensure real-time detection of adversary targets. The role of the WSNs is to detect and report potential targets and suspect events that can impose a threat on the deployed field. Most existing researches emphasize the intelligence of sensors and have developed different mechanisms to promote detection and monitoring abilities of WSNs. Huang et al. discuss the coverage problem in wireless sensor networks with 2D and 3D models [1], [2]. Goldenberg et al. [3] study the coverage and connectivity problem on collaborative mobile sensors to ensure target detection in sensor field.

However, the intelligence of target should also be taken into consideration. Equipped with techniques such as probing, exploring [4] and navigating [5], adversary targets can impose great threat on the deployed sensor field. In [6], targets are considered smart and able to collect sensor information for planning a minimal exposure path. To deal with rational targets, some pioneer works have proposed countermeasures. In [7], it is shown that a few additional sensors on the critical positions can greatly limit the freedom of mobile targets and increase the detection probability. Optimal placement of sensors for intrusion detection is discussed in [8]. The latter two works require sensors to be deployed in specific pattern ahead of the arrival of rational targets, and thus lack flexibility.

Different from previous work, we propose a solution that can react upon the arrival of rational targets. We first investigate the intelligence of rational targets. To plan a minimal exposure path, rational targets need to build a real-time map

of the sensor field. SLAM [9] is a typical work studying how robots can incrementally build a consistent map of an unknown environment and simultaneously determine its location within the map. The key idea of our work is to interfere the map generation process of rational targets so that targets are not able to correctly plan a smart path based on the map. We introduce three collaborative puzzle algorithms for sensor nodes to interfere the SLAM process on mobile rational targets. We build a demonstration system with Iris motes and a mobile robot. The proposed puzzle algorithms are shown efficient in interfering the mobile target motion planning.

The rest of the paper is organized as follows. In Section II, we detail the entire motion planning process on mobile targets. Three puzzle algorithms for sensor nodes to interfere mobile rational targets are introduced in Section III. Section IV presents the demonstration system we build to emulate a surveillance application. The performance of the puzzle algorithms are evaluated in Section V, which shows that they can efficiently disturb the motion planning process of mobile target. Section VI concludes this work and provides some perspectives on its extension.

II. TARGET INTELLIGENCE

In this section, we investigate the possible intelligence and capability of a mobile target within the scope of this work. To make smart movement in a sensor field, a mobile target should be able to (i) get information from the sensor field (ii) generate a map and be aware of its location on the map and (iii) plan its path. These tasks are to be achieved through probing techniques, SLAM and movement planning.

A. Probing Techniques

A mobile target can be equipped with vision or laser sensors to seek the presence and location of sensor nodes. These techniques have been known for decades, and there are various methods to hide the sensor nodes in the field such as steganography in [10]. In this paper, we deal with another probing technique, that is, the target records the statistic properties on radio signals to locate sensor nodes or make realistic estimations through SLAM process.

B. SLAM principle

SLAM is a process by which a mobile robot (target) can build a map of an environment, and at the same time use

this map to deduce its location [11]. Let us consider a mobile robot moving through an environment taking observations over a number of unknown landmarks by its probing modules (on-board sensors or communications). At a time instant k , the following quantities and associated sets should be defined on the target [9]:

- \mathbf{x}_k : The state vector of location and orientation.
- $\mathbf{X}_{0:k} = \{\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_k\} = \{\mathbf{X}_{0:k-1}, \mathbf{x}_k\}$: The history of location.
- \mathbf{u}_k : The control vector, applied at time k .
- $\mathbf{U}_{0:k} = \{\mathbf{u}_0, \mathbf{u}_1, \dots, \mathbf{u}_k\} = \{\mathbf{U}_{0:k-1}, \mathbf{u}_k\}$: The history of control inputs.
- \mathbf{m}_i : A vector describing the location of the i th landmark.
- $\mathbf{m} = \{\mathbf{m}_1, \mathbf{m}_2, \dots, \mathbf{m}_n\}$: The set of all landmarks.
- \mathbf{z}_{ik} : An observation taken from the target over the i th landmark at time k .
- $\mathbf{Z}_{0:k} = \{\mathbf{z}_0, \mathbf{z}_1, \dots, \mathbf{z}_k\} = \{\mathbf{Z}_{0:k-1}, \mathbf{z}_k\}$: The set of all landmark observations.

In probabilistic form, the SLAM requires that the probability distribution $P(\mathbf{x}_k, \mathbf{m} | \mathbf{Z}_{0:k}, \mathbf{U}_{0:k}, \mathbf{x}_0)$ be computed for all time k . The Extended Kalman Filter (EKF) can linearize the mean and the covariance of the process, and hence it is used in many works on SLAM [12], [13]. A typical EKF model can be given as follow:

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, k-1) + \mathbf{q}_{k-1} \quad (1)$$

$$\mathbf{z}_k = h(\mathbf{x}_k, k) + \mathbf{r}_k \quad (2)$$

\mathbf{r}_k is the measurement noise and \mathbf{q}_{k-1} is the process noise. The dynamic model function f and the measurement model function h are both nonlinear. The EKF is applied in two steps: prediction and update. During the prediction step, the next state of the system is predicted given the previous measurements. During the update step, the current state of the system is estimated given the current measurements.

We use a SLAM algorithm in which the localization method is based on range measurement. A target obtains the range by measuring the quality of reception signal when sniffing into the radio communications between sensor nodes. We use the euclidean distance d to denote the range between the target T and each node $n_i = (x_i, y_i)$ broadcasting its coordinates. Each sensor node is then associated with an ID by the target as a landmark on its map.

$$h_i = d(T, N_i) = \sqrt{(x - x_i)^2 + (y - y_i)^2} \quad (3)$$

From the measurement function, the corresponding jacobian is used in the update process of the EKF.

$$\frac{\delta h_i}{\delta x} = \frac{x - x_i}{\sqrt{(x - x_i)^2 + (y - y_i)^2}} \quad (4)$$

$$\frac{\delta h_i}{\delta y} = \frac{y - y_i}{\sqrt{(x - x_i)^2 + (y - y_i)^2}} \quad (5)$$

This allows the target to build the resulting n by n jacobian matrix H_x , where n is the number of measurements made.

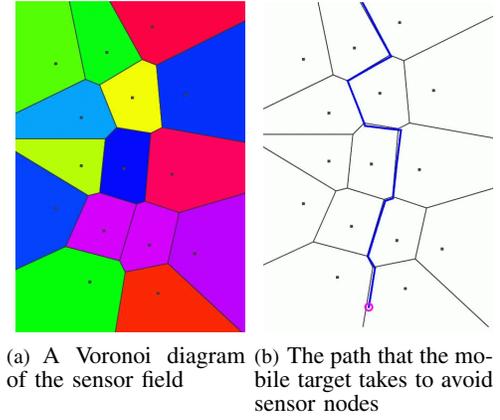


Fig. 1. Motion plan with Voronoi diagram

$$H_x = \begin{pmatrix} \frac{\delta h_1}{\delta x} & \frac{\delta h_1}{\delta y} & 0 & \dots & 0 \\ \frac{\delta h_2}{\delta x} & \frac{\delta h_2}{\delta y} & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{\delta h_n}{\delta x} & \frac{\delta h_n}{\delta y} & 0 & \dots & 0 \end{pmatrix} \quad (6)$$

C. Movement Planning

Before each move, a mobile target runs a movement planning procedure to ensure that the movement is safe. All verifications and computations are based on the map and location information obtained by SLAM. For instance, Fig. 1(a) shows the Voronoi diagram of a sensor field. The positions of sensor nodes are estimated and corrected with the previous detections by mobile target through SLAM. The movement planning algorithm then builds a path as shown in Fig. 1(b), which has the maximal distance to each nodes on this map. The next move follows the segment in the path if the target is already located on this path, otherwise it is toward the path.

The efficiency of the movement planning depends on the accuracy of the map and location obtained by SLAM. Therefore interfering SLAM will cause significant degradation on the correctness of movement planning. To this end, we propose a set of puzzle algorithms for sensor nodes.

III. INTERFERING TECHNIQUES WITH PUZZLE ALGORITHMS

We denote R_c as the communication range of a sensor node. In a target detection application, the whole sensor field should be fully covered by sensor nodes. Therefore, the sensing range R_s should be at least $R_c/2$. If we assume that the detection accuracy of sensor nodes is uniform in its sensing coverage area, a mobile target would be detected no matter how it plans its movement. Unfortunately, the detection accuracy attenuates with the increase of distance from the sensor nodes. Therefore, the accuracy of detection is low when the mobile target is far from the sensor nodes. This is the prerequisite to planning a minimal exposure path.

To this end, we define an accurate sensing range R_a , which is smaller than R_s . Within this range the mobile target is fully

exposed to a sensor node and a sensor can call for actions against the target.

The goal of motion planning on a mobile target is to ensure that its distance from any sensor node is larger than R_a and to keep as far as possible from sensor nodes. On the other side, the goal of the puzzle algorithms is to disturb the motion planning and to mislead the mobile target into the accurate sensing range of any sensor node.

The coordinate of a sensor node can be decomposed into the distance and the orientation from the target to the sensor. It also provides a guideline for designing the puzzle algorithms: sensor nodes may generate collaborative errors on the distance or the orientation. Hereby, we propose three puzzle algorithms corresponding to faking distance, faking orientation and the combination of both.

In *Translation* algorithm, the sensor nodes generate their fake coordinates using geometric translation. They report fake locations that are with a certain distance away from their real locations in a fixed direction. This distance is carefully handled by sensor nodes so that from the perspective of the robot, the locations of the landmarks are static. As a result, the target will take the same move as its previous one and the motion planning is interfered.

The second is *Rotation* algorithm which changes the direction from the target to any sensor nodes. Similar to the translation strategy, we change nodes' coordinates by adding a constant deviation on the direction. It interferes the angle computation in SLAM, and causes it to fail in the correction of trajectory. Therefore the map generation is disturbed and the motion planning is interfered.

Hybrid algorithm is a combination of the above two algorithms. Sensor nodes first apply the *Rotation* algorithm, then *Translation* algorithm. The WSN elects a trap node in the first place, and broadcasts its identity among the network. After that, it calculates the angle difference between the current target direction and the trap node angle. The WSN turns around the target without being detected by SLAM. And they applied the translation algorithm to make sure that the target go straight to the trap node.

IV. SYSTEM IMPLEMENTATION

Our experimental testbed is composed of a WSN, a mobile robot and a workstation. Figure 2 illustrates the system architecture. The WSN is formed by several nodes exchanging control messages and a gateway. The robot can overhear these messages and obtain the nodes' coordinates. The base station communicates with the gateway and the robot to trace the traffic on the network and the robot's trajectory. It is in charge of results collection.

The WSN is established with Iris motes based on TinyOS. The nodes communicate over IEEE 802.15.4. An MIB520 gateway configures and manages the WSN as well as transfers the traffic information of the network to the base station. The robot is equipped with LM3S1968 ARM microcontroller. XBee series 1 chip (802.15.4 compatible) is used as communication module.

The software architecture is presented in Figure 3. The robot and rational target moves in the WSN in a discrete manner. At each step, the robot first overhears the communication among sensor nodes to obtain their coordinates. Then the measurement function is computed to get the distance between the robot and the sensors. The EKF in (4) is applied to obtain the corrected coordinates of the robot. Given the coordinates, the next motion of the robot is planned based on a Voronoi diagram strategy. The robot aims at moving across the WSN on a minimal exposure path.

When the puzzle algorithm is enabled, the sensors collaborate to generate false location information to mislead the robot. The puzzle algorithm takes advantage of the mechanisms of SLAM. This makes the misleading purpose undetectable from the perspective of the robot.

V. PERFORMANCE EVALUATION

This part presents both the experimental results and the simulation results. The following parameters remain the same in both experimentation and simulation: $R_c = 30 \text{ units}$, $R_s = 15 \text{ units}$ and $R_a = 5 \text{ units}$.

A. Experimentation Results

We use the network topology given in Figure 5. The robot starts at point S and attempts to cross the deployed field and reach the final destination F . We generate four mobile target trajectories under four different experimental conditions: *Ideal trajectory*, with neither transmission error nor SLAM; *Under error trajectory*, transmission errors are added to the data gained from robot's probing; *With SLAM*, SLAM is added to correct transmission errors; and *Interfered trajectory*, the hybrid puzzle algorithm is activated on the WSN, while transmission errors and SLAM are still present.

The following facts can be observed on Figure 7:

- 1) The motion planning works well under ideal environment. It is the optimal path for the mobile target.
- 2) The transmission errors disturb the motion planning, as a result, the robot deviates from the optimal path.
- 3) SLAM corrects the movement of the target and keeps the target close to the optimal path. The reason is that SLAM is efficient in correcting Gaussian noise known as independent errors.
- 4) The consequence of puzzle algorithm on the target is obvious. The target is misled thus entering the accurate sensing range of a sensor.

In order to quantify the impact of transmission errors, SLAM and puzzle algorithm on the target's motion planning, we run 100 tests under this experimentation scenario and collect: minimal distance from any node; proximity which indicates the average distance from the closest node during the whole path; and path length before the robot is detected or reaches the destination.

Figure 8 shows that the transmission errors bring the mobile target closer to sensor nodes than in the ideal trajectory. The SLAM process implemented on the target does not really improve this minimal distance but does not decrease it either.

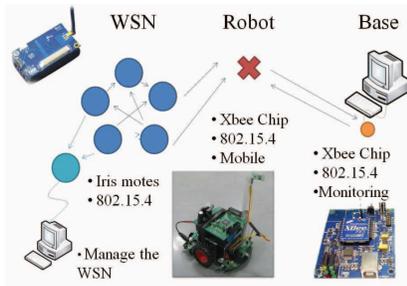


Fig. 2. System hardware architecture

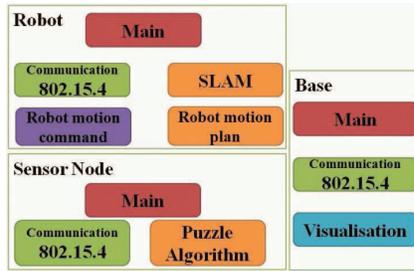


Fig. 3. System software architecture

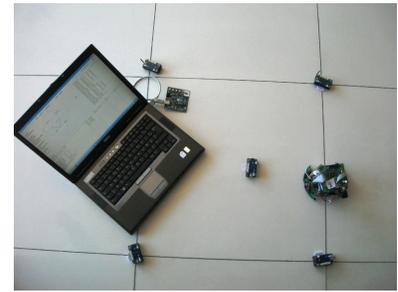


Fig. 4. Experimental testbed

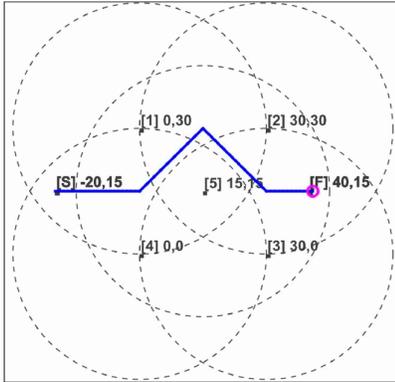


Fig. 5. The topology of the WSN



Fig. 6. Experimentation setting up

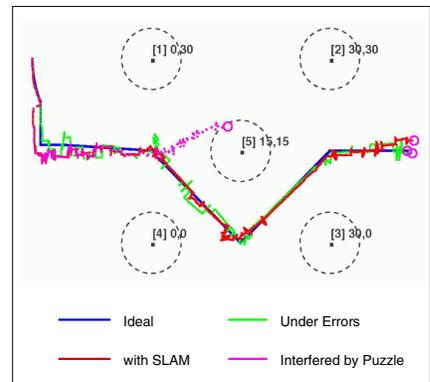


Fig. 7. Four trajectories

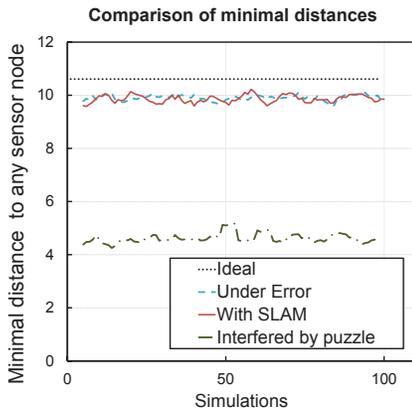


Fig. 8. The minimal distance to any sensor node in the network. The bigger it is, the worse the target detection is.

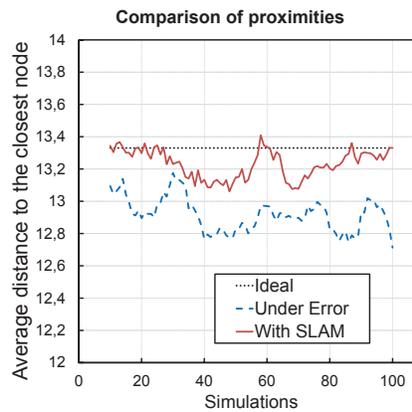


Fig. 9. The average distance to the closest nodes during a target's movement.

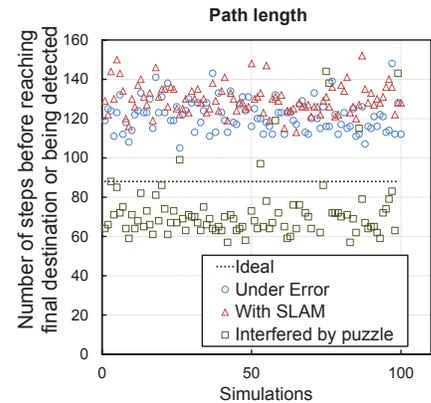


Fig. 10. The SLAM on mobile target is always interfered by puzzle algorithm, and is detected before going through the sensor field.

When puzzle algorithm is applied on sensor nodes to interfere the target motion planning, the minimal distance chute into the accurate sensing range, because the target is misled to a sensor node.

Figure 9 shows that the mobile target's proximity in the error environment is lower than in the ideal case. When SLAM is activated, the average distance to sensor nodes increases in comparison to the trajectory with error.

As shown on Figure 10, when transmission errors are added, the *Path Length* is longer in comparison to the ideal

TABLE I
AVERAGE PATH LENGTH (IN UNITS)

Ideal	Under Error	With SLAM	Interfered by puzzle
88	121	130	77

environment. The SLAM process increases the path length to a larger extent by correction with historical measurements. The average path length of the 100 tests is given in Table I. The puzzle algorithm can lead the target to be detected with high accuracy before the target step out of the sensor field.

B. Simulation Results

We runs 100 simulations on random network topologies composed of specified number of nodes. The average distance between sensor nodes are controlled to be approximate to the sensing range. Three puzzle algorithms are evaluated. We compare the detection ratio and puzzle path length.

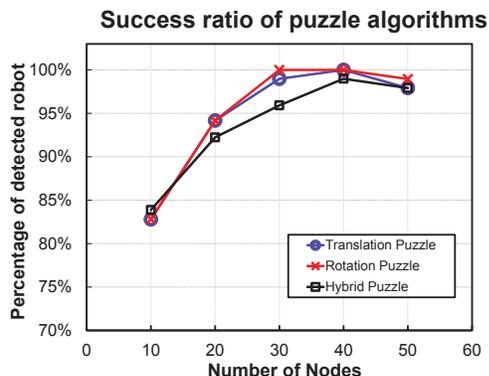


Fig. 11. Detection ratio for different puzzle

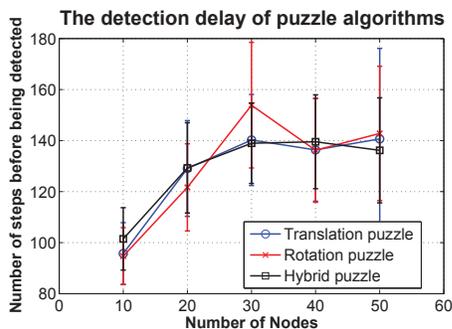


Fig. 12. Path Length for different puzzles

Figure 11 shows that the more the network size increases, the higher the detection ratio is. The translation and rotation puzzle algorithms achieve almost the same detection ratio. It is unexpected that the hybrid algorithm is not as good as the other two. It might be caused by the uncertainty brought about by the transmission errors. Therefore probabilistic solutions (translation and rotation puzzle) work better than deterministic solutions such as hybrid puzzle.

Figure 12 shows that the *Path Length* under different puzzle algorithms are very close to each other. More importantly, when density of nodes becomes important, the value of path length is stable. It is a good property which means that the detection speed can be estimated despite the motion of targets.

VI. CONCLUSION

In this work, we provide a complete paradigm for target detection in WSNs. We assume that intelligent mobile targets can use motion planning to travel through the sensor field and avoid being accurately detected. The motion planning is composed of probing techniques, SLAM for location and

map generation and movement planning. We propose a new approach for sensor nodes to interfere the motion planning on mobile target so as to expose them. We focus on three puzzle algorithms generating collaborative errors that cannot be eliminated by SLAM. The experiment and the simulation results confirm that these puzzle algorithms can effectively mislead the mobile target and enhance the target detection probability in surveillance applications. We are currently investigating the scenario in which the robot uses RSSI values to calculate the positions of the nodes and find the minimal intrusion path.

ACKNOWLEDGEMENT

This research was supported by NSF of China under grant No.61073158, No.61100210 and Shanghai Post-Doc grant No.09R21413700. The authors would like to thank the Embedded System Lab of Shanghai Jiao Tong University for their technical support on the robot.

REFERENCES

- [1] C.F. Huang and Y.C. Tseng. The coverage problem in a wireless sensor network. *ACM Workshop on Wireless Sensor Networks and Applications (WSNA)*, pages 115–121, 2003.
- [2] D. Huang, M. Mehta, D. Medhi, and L. Harn. Location-aware key management scheme for wireless sensor networks. *Proc. of the 2nd ACM workshop on Security of Ad Hoc and Sensor Networks*, Washington, D. C. , USA., pages 29–42, October 2004.
- [3] D.K. Goldenberg, L. Lin, and A.S. Morse. Towards mobility as a network control primitive. *The Fifth ACM Int'l Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, pages 163–174, 2004.
- [4] C. Stachniss and W. Burgard. Exploring unknown environments with mobile robots using coverage maps. *Proc. of the 18th int'l Conference on Artificial intelligence, Acapulco, Mexico*, pages 1127–1132, 2003.
- [5] G. DeSouza and A. Kak. Vision for mobile robot navigation: a survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(2):237–267, February 2002.
- [6] W. Zhang, M. Li, W. Shu, and M. Wu. Smart path-finding with local information in a sensory field. *Proc. of Int'l Conference on Mobile Ad-hoc and Sensor Networks, HongKong, China*, pages 119–130, 2006.
- [7] S. Zhou, M.Y. Wu, and W. Shu. Improving mobile target detection on randomly deployed sensor networks. *Int'l Journal of Sensor Networks*, 6(2):115–128, 2009.
- [8] F. Anjum, D. Subhadrabandhu, S. Sarkar, and R. Shetty. On optimal placement of intrusion detection modules in sensor networks. *Proc. of the 1st International Conference on Broadband Networks, San Jose, USA*, pages 690–699, October 2004.
- [9] H. Durrant-Whyte and T. Bailey. Simultaneous Localisation and Mapping (SLAM): Part I the essential algorithms. *Robotics and Automation Magazine*, 13(2):99–110, June 2006.
- [10] C. Turner. A steganographic computational paradigm for wireless sensor networks. *Proc. of the 6th Int'l Conference on Innovations in Information Technology, Al-Ain, United Arab Emirates*, pages 270–274, 2009.
- [11] J. Aulinas, Y. Petillot, J. Salvi, and X. Llado. The SLAM problem: a survey. *Frontiers in Artificial Intelligence and Applications*, 184:363–371, 2008.
- [12] A.J. Davison and D. Murray. Simultaneous localization and map-building using active vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):865–880, 2002.
- [13] P.M. Newman and J.J. Leonard. Consistent, convergent, and constant-time SLAM. *International Joint Conference on Artificial Intelligence (IJCAI), Acapulco, Mexico*, pages 1143–1150, 2003.