A Distributed Tracking Algorithm for Target Interception in Face-Structured Sensor Networks

Efren L. Souza*†, Richard W. Pazzi† and Eduardo F. Nakamura*‡
efren@icomp.ufam.edu.br, richard.pazzi@uoit.ca and eduardo.nakamura@fucapi.br
* Federal University of Amazonas – UFAM
† University of Ontario Institute of Technology – UOIT
‡ Analysis, Research and Tech. Innov. Center – FUCAPI

***Abstract*—An addition to the target tracking application is the presence of a tracker that aims to intercept the target. The WSN provides the position of the target to the tracker, so that it can move towards the target. The objective is to reach the target in the shortest possible time, using the fewest number of nodes to save energy. In this paper, we propose and evaluate a Tracking Algorithm for Target Interception in face-structured sensor networks (TATI). TATI reduces the number of active nodes, activating only the faces that are feasible to be achieved by the target until the next sampling. Furthermore, the algorithm uses the nominal communication range of the nodes to shorten the route of the tracker. Simulation results show that TATI can save up to 15% in energy consumption, and it can keep the tracker around 10 meters closer to the target compared to the baseline.**

I. Introduction

A Wireless Sensor Network (WSN) is a special type of ad-hoc network composed of resource-constrained devices, called sensor nodes [1]. A sensor network may be designed to monitor an environment for the occurrence of a set of possible events, such as target presence in target tracking applications [2].

Traditional tracking applications aim to collect sensor data about one or more targets (mobile objects of interest) to estimate their positions in the sensor field, and then report the result to a static sink node [3]. On the other hand, tracking applications can be formulated to include a tracker [4][5][6]. The tracker must follow the target, but it can not detect the target, then the sensor network should help the tracker to approach the target. The sensor node carrying the track information acts as a beacon that guides the tracker. The tracker follows the route of beacons to approach the target. Usually, a set of nodes is selected to track the target, while the remaining nodes stay in sleep mode to save energy.

In this paper, we propose and evaluate the Tracking Algorithm for Target Interception in face-structured sensor Networks (TATI). This algorithm is applied to assist a tracker in chasing a target. To this end, sensor nodes cooperate to adjust the route between the target and the tracker dynamically, while reducing the amount of active nodes to save energy even further. A sensor network is organized into face-structure to facilitate node cooperation, distinguish different areas in the sensor network, and solve the hole problem [4][7]. The position and velocity of the target are used as parameters to define the faces that should be activated, reducing the number of active nodes. In addition, the TATI calculates query points based on nominal communication range to shorten the route followed by the tracker.

The remainder of the paper is organized as follows. Section II presents the related works about target tracking with tracker. In Section III we show the problem formulation and the definitions used in this paper. In Section IV we describe our proposal for target tracking in sensor networks that assists the tracker in chasing the target. In Section V we present our experimental methodology and quantitative evaluation. Finally, in Section VI we present our conclusions and future work.

II. Related Works

The naive target tracking method with tracker is based in successive floodings. The tracker starts a flooding on the network in order to request the target position, then it receives the response and moves toward that position. This process is repeated as long as the tracker intends to approach the target. High energy consumption occurs in this process since the entire network will take part in each sample [4].

More elaborated methods use just a fraction of nodes to track the target. Those methods usually configure the network in face structures [7] in order to assist the selection of the set of nodes that must be activated. Tsai et al. [4] propose the Dynamic Object Tracking (DOT) algorithm. It selects the node closer to the target and activates all nodes of its adjacent faces, ensuring that there is active nodes around the target. The remaining nodes stay in the sleep state to save energy. In addition, the sensor nodes cooperate to shorten the route between target and tracker to chase the target quickly and maintain an accurate tracking route.

Other methods use predictions to reduce the amount of active faces and save more energy. Bhuiyan et al. [5] use a simple regression-based prediction method to compute velocity and direction of the target. The prediction is used to choose a minimal number of appropriate sensor nodes that are the nearest to the target, once an informed selection of nodes with the best data for cooperation can save power. Hsu et al. [6] use that same regression-based prediction to estimate the face that the target will visit next. During tracking, the sensor node closer to the target activates the other nodes of the same face in order to track the target. If the target is going to leave the territory of the current face, the predicted face must be activated. Regression-based prediction also is used in Ji et al. [8], they combine a hexagon algorithm with face-structure model that limits the next target position to $1/6$ of the face area to decrease the number of active faces.

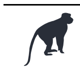
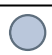
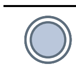
978-1-4799-3780-6/14/$31.00 ©2014 IEEE

III. OVERVIEW

In the target tracking formulation of this work, a tracker is assisted by the network to approach a target. The algorithms applied in this scenario must provide a route that makes target interception easier. In addition, they must reduce the number of nodes that effectively track the target in order to save energy. Target is the object of interest, it moves randomly on the sensor network and it can be detected by sensor nodes. Although the tracker can not detect target, it has communication and mobility resources, thus it can query sensor nodes about the target position.

Sensors nodes are all clock synchronized. Each node knows its nominal communication range and location, which can be acquired from Global Positioning System (GPS) or some distributed localization system [9]. All nodes have the same communication and sensing ranges. Sensor nodes that are not participating in the tracking turn off their radios to save energy. Nodes can operate in five states: sleep, awake, active, near, and beacon. These states define the level of power consumption of the node and its function in the tracking. Table I shows the elements that act on tracking.

TABLE I. TARGET TRACKING ELEMENTS.

Picture	Element
	Target – Mobile object tracked by sensor nodes, e.g., person, animal, or vehicle
	Tracker – Mobile object that must follow the target, e.g., person or robot. A tracker can not detect the target; the sensor network should help the tracker to approach the target
	Sleep Node – Node in a state of energy saving. It does not participate in tracking, but periodically wakes at a predefined period and listens to the communication channel to check wakeup packets
	Active Node – Node that effectively tracks the target. It can sense the target, receive or transmit data at any time
	Beacon Node – Beacon nodes compose a sequence that represents the target trajectory. The tracker queries the beacons to discover the direction to follow
	Near Node – Node closest to the target. It computes the target position and selects nodes that must be activated

Reducing the number of nodes in active state and the number of messages sent are the most important factors that contribute to energy savings. However, the set of active nodes should be selected carefully to avoid losing the target.

Figure 1 is an example of target tracking in a face-structure sensor network. Only nodes closer to the target stay in active state, while the remaining nodes stay in sleep state to save energy. Beacon nodes act as a mark to guide the tracker to chase the target. They represent the target trajectory.

Face routing is used to construct a face-structure in a distributed way in order to facilitate node cooperation, to distinguish different areas in the sensor field, and to solve the hole problem [7][6].

IV. TRACKING ALGORITHM FOR TARGET INTERCEPTION

The TATI algorithm is applied to assist the tracker in approaching the target in a face-structured sensor network. This algorithm saves energy reducing the active faces and shortening the tracker route by using the nominal communication range information. In the remainder of this section we describe the TATI scheme in detail.

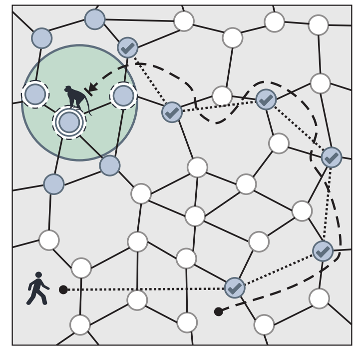


Fig. 1. Overview of target tracking with tracker.

A. Network Setup

TATI reserves an initial time to setup the sensor network. During setup time, the sensor network is organized in faces in a distributed way with Gabriel Graph (GG) [4] or Relative Neighborhood Graph (RNG) [6].

B. Target Discovery

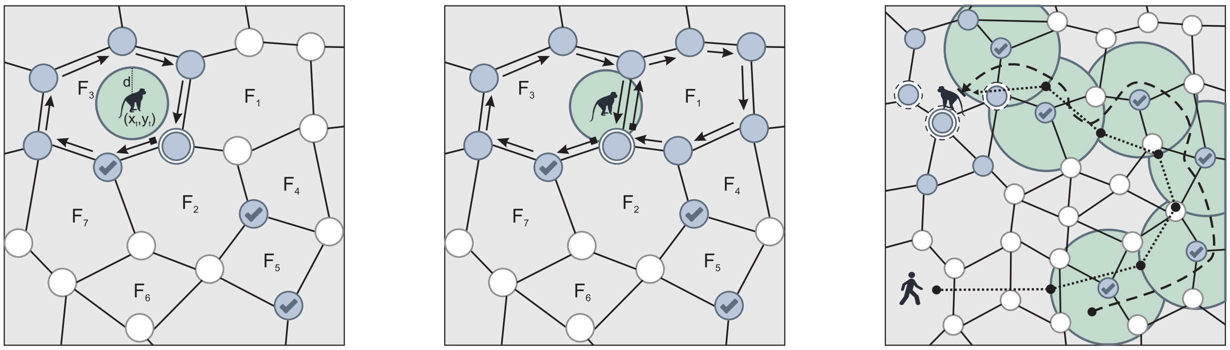
The target position is unknown before the tracking, and then sensor nodes cooperate to find the target. When nodes are in awake state, the tracker starts a flooding on the network with *TargetRequest* message to request the position of the target. The closest node to the target changes its state to near, activates all nodes of its adjacent faces, and becomes the beacon of the face in which the target was found. This node also sends the target information to the tracker. Once the target is found, tracking is maintained by active nodes.

C. Target Tracking

TATI selects the set of active nodes based on face structure and target displacement. The node in near state sends a *Wakeup* message periodically to nodes of the faces that must be activated. This message is forwarded following the face sequence until it reaches all nodes of the faces.

The nodes that receive a *Wakeup* message change their states to active and start cooperating to detect the target. The nodes that detect the target broadcast a *TargetDetected* message among themselves. They use this message to compute the position of the target and to determine which one of them will be the new near node. Thus, the role of near node switches among nodes while the target moves on the face, so the set of active nodes is dynamically adjusted with the target movement.

A near node is responsible for originating *Wakeup* messages. The near node selects which of its adjacent faces must be activated based on the maximum distance d traveled by the target between two samplings. Faces that must be activated intersect the circle with center (x_t, y_t) and radius d, where (x_t, y_t) is the current position of the target, as shown in the Figures 2(a) and 2(b).



(a) Only face F_3 is activated.

(b) Faces F_1 and F_3 are activated.

(c) Tracker approaches target using query point.

Fig. 2. Face activation and query points.

A near node becomes beacon when the target crosses from one face to another. Each face traversed by the target has a beacon node that records the trajectory of the target.

D. Tracker Route

Beacon nodes record the trajectory of the target. These nodes can be seen as a doubly-linked list, since each node knows its immediately previous and next beacons.

The tracker approaches the first beacon in order to query the path it should follow, then the tracker moves toward the reported position. After arriving at its destination, the tracker performs another query. After the beacon node informs the query point, it returns to either awake or sleep state.

Previous works use beacon location as a query point. TATI uses nominal communication range r_n, tracker position (x_o, y_o), and next beacon location (x_b, y_b) to compute a query point (x_q, y_q) that reduces the tracker route. Thus, (x_q, y_q) is the intersection between the straight line that connects the points (x_o, y_o) and (x_b, y_b) to the circle of center (x_b, y_b) and radius r_n, as shown in Figure 2(c).

V. EVALUATION

A. Methodology

Performance evaluation is conducted using ns-2 simulations. The default network consists of 81 sensor nodes distributed on a sensor field of 200×200m^2. The communication range of the nodes is 40m to enable communication among all nodes by multiple hops. Sensing range is 30m to ensure that there are at least three nodes covering any point on sensor field. The target mobility model is the Correlated Random Walk (CRW) [10] with a correlation degree of 0.70 and velocity is 10m/s.

We model the energy consumption according to different operation modes of the CC2420 radio [11]. We consider that each message has a size of 32 bytes. Table II shows the parameters used in the simulations.

Each plotted point averages 35 random topologies to ensure more reliable results and each run simulates 2000 seconds. The error bars represent the confidence interval for 99% of confidence.

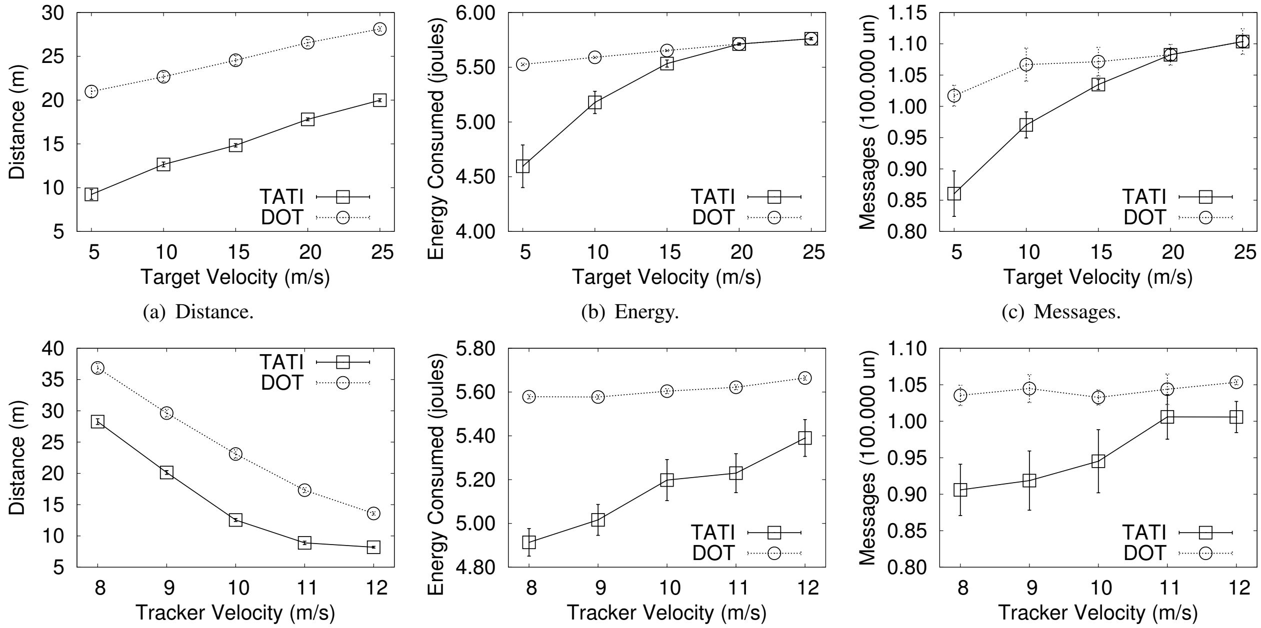
B. Simulation Results

1) Target Velocity: In this set of evaluations, we vary target velocity from 5m/s to 25m/s. Tracker velocity is always equal to target velocity. The Figure 3(a) shows the distance between target to tracker when target velocity is increased. That distance increases with velocity because there is a delay to detect the target, calculate its position, and report the results to tracker. Meanwhile, the target continues to move. The faster it moves, the more it moves away. TATI can reduce this distance because the query points minimize the tracker route, while DOT uses just beacon location. Figure 3(b) shows that TATI consumes less power than DOT when the target velocity is less than 20m/s. This is because target displacement between two samples is small enough so that it enables TATI to activate only a few faces. In cases where velocity exceeds 15m/s, the energy consumption of both methods are equivalent, because target displacement forces TATI to activate all adjacent faces of the near node, similar to DOT. A face is activated using messages that traverse all of its nodes. TATI activates fewer faces, thus fewer messages are sent during tracking, as shown in Figure 3(c). When target velocity exceeds 15m/s, the amount of messages sent increases in both methods. This is because the target is lost more often, and thus a flooding is performed on the network to recovery the target.

2) Tracker Velocity: This section evaluates the velocity difference between the target and tracker. Target velocity is fixed at 10m/s, while tracker velocity varies from 8m/s to 12m/s, to consider cases where tracker is slower or faster than the target. In Figure 3(d), TATI keeps the tracker closer to the target because query points minimize the tracker route. When the tracker is slower than the target, it has difficulty approaching the target, even if the tracker route is minimized, because the target is always moving at faster speeds. In

TABLE II. ENERGY PARAMETERS.

Parameter	Value
Initial Energy	18720 joules
Sensing	2×10^{-5} watts
Transmit	3132×10^{-5} watts
Receive	3546×10^{-5} watts
Idle	76.68×10^{-5} watts
Sleep	3.6×10^{-5} watts
State transition	5×10^{-5} watts



(a) Distance. (b) Energy. (c) Messages.

(d) Distance. (e) Energy. (f) Messages.

Fig. 3. Evaluations results.

Figure 3(e), TATI consumes less power than DOT because it uses less active faces during tracking. Both methods consume more energy when tracker velocity increases, specially TATI. The reason is that when the tracker is close to the target, it often queries the last beacon to continue approaching the target. For this same reason, the amount of messages sent increases, as depicted in Figure 3(f).

VI. Conclusions and Future Work

In this paper, we propose and evaluate the TATI algorithm using DOT as baseline. The TATI algorithm reduces energy consumption since fewer faces are active during the tracking task. It also reduces the route traveled by the tracker by using query points. These points are the shortest distance between tracker position and the communication area of beacon nodes.

TATI records the maximum displacement of the target between two samplings to select the faces that should be activated. Therefore, TATI activates a smaller number of faces when compared to DOT. However, the energy consumption results of these methods are equivalent when the target velocity is high (20m/s or more).

In general, TATI keeps the tracker closer to the target, because it uses the nominal communication range information to reduce the tracker path.

Future directions include: (i) development of a recovery mechanism that avoids flooding to recover a lost target; (ii) evaluation of the impact of node localization errors; (iii) comparison of different planarization algorithms.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: A survey," *Computer Networks*, vol. 38, no. 4, pp. 393–422, 2002.

[2] E. F. Nakamura, A. A. F. Loureiro, and A. C. F. Orgambide, "Information fusion for wireless sensor networks: Methods, models, and classifications," *ACM Computing Surveys*, vol. 39, no. 3, p. 55, 2007, article 9.

[3] F. Hajiaghajani, M. Naderan, H. Pedram, and M. Dehghan, "HCMTT: Hybrid clustering for multi-target tracking in wireless sensor networks," in *Proceedings of the 4th International Conference on Pervasive Computing and Communications Workshops (PERCOM'12)*, Lugano, Switzerland, March 2012, pp. 889–894.

[4] H.-W. Tsai, C.-P. Chu, and T.-S. Chen, "Mobile object tracking in wireless sensor networks," *Computer Communications*, vol. 30, no. 8, pp. 1811–1825, 2007.

[5] M. Bhuiyan, G. jun Wang, L. Zhang, and Y. Peng, "Prediction-based energy-efficient target tracking protocol in wireless sensor networks," *Journal of Central South University of Technology*, vol. 17, no. 2, pp. 340–348, 2010.

[6] J.-M. Hsu, C.-C. Chen, and C.-C. Li, "POOT: An efficient object tracking strategy based on short-term optimistic predictions for face-structured sensor networks," *Computers & Mathematics with Applications*, vol. 63, no. 2, pp. 391–406, 2012.

[7] Q. Huang, S. Bhattacharya, C. Lu, and G.-C. Roman, "FAR: Face-Aware Routing for mobicast in large-scale sensor networks," *ACM Transactions on Sensor Networks*, vol. 1, no. 2, pp. 240–271, 2005.

[8] X. Ji, Y.-Y. Zhang, S. Hussain, D.-X. Jin, E.-M. Lee, and M.-S. Park, "FOTP: Face-based Object Tracking Protocol in wireless sensor network," in *Proceedings of the 4th International Conference on Computer Sciences and Convergence Information Technology (ICCIT'09)*, Los Alamitos, USA, November 2009, pp. 128–133.

[9] H. A. B. F. Oliveira, A. Boukerche, E. F. Nakamura, and A. A. F. Loureiro, "An efficient directed localization recursion protocol for wireless sensor networks," *IEEE Transactions Computing*, vol. 58, no. 5, pp. 677–691, 2009.

[10] H. Wu, B.-L. Li, T. A. Springer, and W. H. Neill, "Modelling animal movement as a persistent random walk in two dimensions: expected magnitude of net displacement," *Ecological Modelling*, vol. 132, no. 2, pp. 115–124, 2000.

[11] H. Wang, X. Zhang, F. Naït-Abdesselam, and A. Khokhar, "DPS-MAC: an asynchronous MAC protocol for wireless sensor networks," in *Proceedings of the 14th international conference on High performance computing (HiPC'07)*, Goa, India, December 2007, pp. 393–404.