

DETECTING MOVEMENTS OF A TARGET USING FACE TRACKING IN WIRELESS SENSOR NETWORKS

M.Chandhraleka¹, Dr.M.Pushparani²

*¹M.C.A, ²Professor and Head, Department of Computer Science, Mother Teresa Women's
University, (India)*

ABSTRACT

Target tracking is one of the key applications of wireless sensor networks (WSNs). Existing work mostly requires organizing groups of sensor nodes with measurements of a target's movements or accurate distance measurements from the nodes to the target, and predicting those movements. These are, however, often difficult to accurately achieve in practice, especially in the case of unpredictable environments, sensor faults, etc. In this paper, we propose a new tracking framework, called Face Track, which employs the nodes of a spatial region surrounding a target, called a face. Instead of predicting the target location separately in a face, we estimate the target's moving toward another face. We introduce an edge detection algorithm to generate each face further in such a way that the nodes can prepare ahead of the target's moving, which greatly helps tracking the target in a timely fashion and recovering from special cases, e.g., sensor fault, loss of tracking. Also, we develop an optimal selection algorithm to select which sensors of faces to query and to forward the tracking data. Simulation results, compared with existing work, show that Face Track achieves better tracking accuracy and energy efficiency.

Keywords: Mobile Network, Topology, Routing Protocol, Wireless Sensor Network, Ad Hoc Network.

I. INTRODUCTION

A wireless sensor network (WSN) consists of a large number of sensors which are densely deployed over a large area. Each sensor monitors a physical environment and communicates via wireless signals. With the advancements in hardware miniaturization and wireless communication technologies, WSNs have been used in various applications such as education, warfare, and traffic monitoring. Regardless of the applications, extending the network lifetime is a critical issue in WSNs. This is because the sensors are battery-powered and generally difficult to be recharged. One of the main objectives. Unlike detection, a target tracking system is often required to ensure continuous monitoring, i.e., there always exist nodes that can detect the target along its trajectory. In target tracking applications, idle listening is a major source of energy waste. In this project, we propose Face Track, a framework to detect movements of a target using face tracking in a WSN, which does not fall into existing categories and is, to the best of our knowledge, the first of its kind. The concept of Face Track is depicted in Fig. 1, and is inspired by computational geometry, geographic routing, and face routing, in particular.

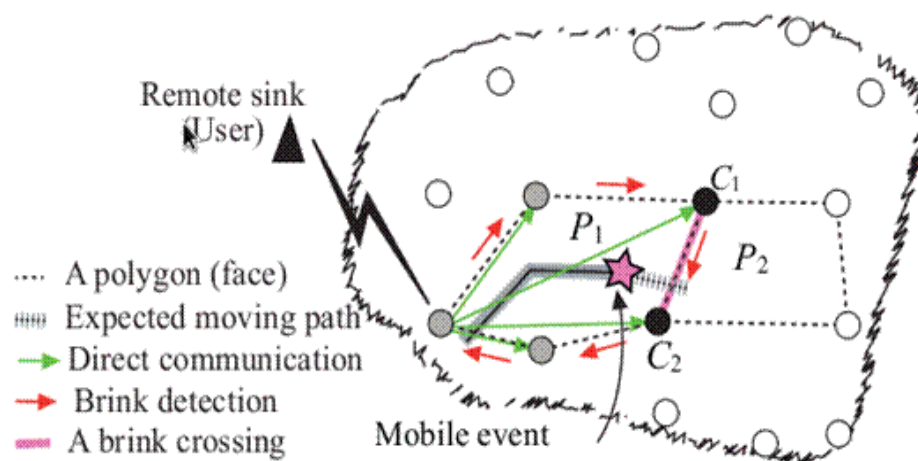


Fig. 1. An example application with a sink showing a vehicle being tracked through a polygonal-shaped area.

Fig. 1 Illustrates a Typical Scenario of an Enemy Vehicle Tracking Application. Sensor Nodes are Informed When the Vehicle Under Surveillance is Discovered, While Some Nodes (Such as Black Nodes) Detect the Vehicle and Send a Vigilance Message to the Nodes on the Vehicle's Expected Moving Path, So as to Wake Them Up. Thus, the Nodes (Such As Grey Nodes) in the Vehicle's Moving Path Can Prepare in Advance and Remain Vigilant in Front of it as it Moves.

II. REVIEW OF LITERATURE

Target tracking is one of the key applications of wireless sensor networks (WSNs). Existing work mostly requires organizing groups of sensor nodes with measurements of a target's movements or accurate distance measurements from the nodes to the target, and predicting those movements. These are, however, often difficult to accurately achieve in practice, especially in the case of unpredictable environments, sensor faults, etc. In this paper, we propose a new tracking framework, called FaceTrack, which employs the nodes of a spatial region surrounding a target, called a face. Instead of predicting the target location separately in a face, we estimate the target's moving toward another face. We introduce an edge detection algorithm to generate each face further in such a way that the nodes can prepare ahead of the target's moving, which greatly helps tracking the target in a timely fashion and recovering from special cases, e.g., sensor fault, loss of tracking. Also, we develop an optimal selection algorithm to select which sensors of faces to query and to forward the tracking data. Simulation results, compared with existing work, show that FaceTrack achieves better tracking accuracy and energy efficiency. We also validate its effectiveness via a proof-of-concept system of the Imote2 sensor platform.[1]

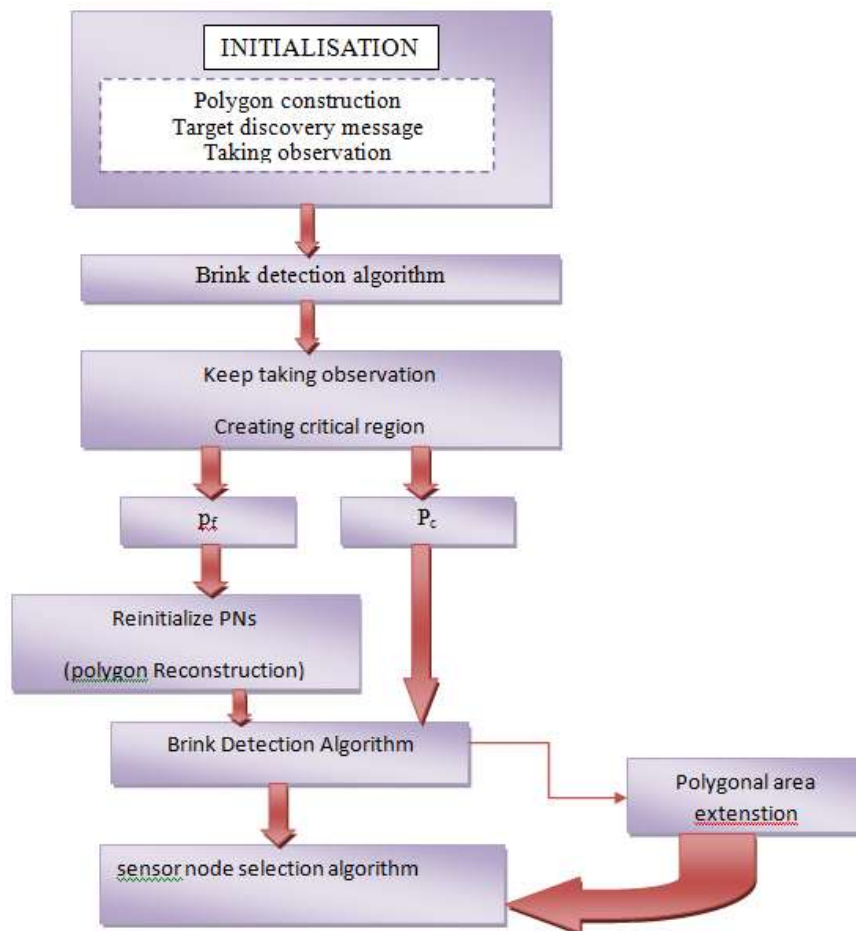
The distinctive features of mobile ad hoc networks (MANETs), including dynamic topology and open wireless medium, may lead MANETs suffering from many security vulnerabilities. In this paper, using recent advances in uncertain reasoning originated from artificial intelligence community, we propose a unified trust management scheme that enhances the security in MANETs. In the proposed trust management scheme, the trust model has two components: trust from direct observation and trust from indirect observation. With direct observation from

an observer node, the trust value is derived using Bayesian inference, which is a type of uncertain reasoning when the full probability model can be defined. On the other hand, with indirect observation, also called secondhand information that is obtained from neighbor nodes of the observer node, the trust value is derived using the Dempster-Shafer theory, which is another type of uncertain reasoning when the proposition of interest can be derived by an indirect method. Combining these two components in the trust model, we can obtain more accurate trust values of the observed nodes in MANETs. We then evaluate our scheme under the scenario of MANET routing. Extensive simulation results show the effectiveness of the proposed scheme. Specifically, throughput and packet delivery ratio can be improved significantly with slightly increased average end-to-end delay and overhead of messages.[2]

Mobility-assisted data collection in sensor networks creates a new dimension to reduce and balance the energy consumption for sensor nodes. However, it also introduces extra latency in the data collection process due to the limited mobility of mobile elements. Therefore, how to schedule the movement of mobile elements throughout the field is of ultimate importance. In this paper, the on-demand scenario where data collection requests arrive at the mobile element progressively is investigated, and the data collection process is modelled as an $M=G=1=c-NJN$ queuing system with an intuitive service discipline of nearest-job-next (NJN). Based on this model, the performance of data collection is evaluated through both theoretical analysis and extensive simulation. NJN is further extended by considering the possible requests combination (NJNC). The simulation results validate our models and offer more insights when compared with the first-come-first-serve (FCFS) discipline. In contrary to the conventional wisdom of the starvation problem, we reveal that NJN and NJNC have better performance than FCFS, in both the average and more importantly the worst cases, which offers the much needed assurance to adopt NJN and NJNC in the design of more sophisticated data collection schemes, as well as other similar scheduling scenarios.[3]

Recently, the research focus on geographic routing, a promising routing scheme in wireless sensor networks (WSNs), is shifting toward duty-cycled WSNs in which sensors are sleep scheduled to reduce energy consumption. However, except the connected- k neighborhood (CKN) sleep scheduling algorithm and the geographic routing oriented sleep scheduling (GSS) algorithm, nearly all research work about geographic routing in duty-cycled WSNs has focused on the geographic forwarding mechanism; further, most of the existing work has ignored the fact that sensors can be mobile. In this paper, we focus on sleep scheduling for geographic routing in duty-cycled WSNs with mobile sensors and propose two geographic-distance-based connected- k neighborhood (GCKN) sleep scheduling algorithms. The first one is the *geographic-distance-based connected-neighborhood for first path* (GCKNF) sleep scheduling algorithm. The second one is the *geographic-distance-based connected-neighborhood for all paths* (GCKNA) sleep scheduling algorithm. By theoretical analysis and simulations, we show that when there are mobile sensors, geographic routing can achieve much shorter average lengths for the first transmission path explored in WSNs employing GCKNF sleep scheduling and all transmission paths searched in WSNs employing GCKNA sleep scheduling compared with those in WSNs employing CKN and GSS sleep scheduling.[4]

3. Methodology



3.1 Initialization

The system initialization, including initial polygon construction in the plane. A node has all of the corresponding polygons' information after the WSN planarization. Initially, all nodes in the WSN are in a low-power mode and wake up at a predefined period to carry out the sensing for a short time. We presume that a sensor node has three different states of operation, namely, active (when a node is in a vigilant manner and participates in tracking the target), awakening (when a node awakes for a short period of time), and inactive (when a node is in a sleeping state). We consider that a sensor should be kept awake so long as its participation is needed for a given task. In the beginning, when a target is detected by some nodes, that communicate to all of its adjacent neighbors with their detection information, and reconstruct the polygon.[1]

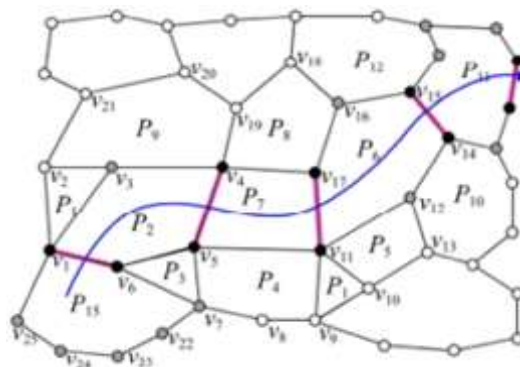


Fig 2. Detecting Target's Movements through Polygons

Fig. 2 illustrates the target movement detection through the polygons. The target is initially detected by sensors v1 and v6 (shaded black to indicate the CNs) in the polygon P15, and the rest of the corresponding nodes (shaded grey) in P15 are in the vigilant manner, and the rest of the nodes in the sensor network are in the inactive state when the target is in P15. As shown in Fig. 2, the target travels through the polygons.[1]

3.2 Brink Detection Algorithm

We introduce an edge detection algorithm, which is used to reconstruct another conceptual polygon, called a critical region, by generating an edge, called a brink, to the active polygon, Pc. As the brink is generated on the boundary of In this algorithm, the edges of Pc are mapped by the brinks. As the target moves to a brink, the target is focused on a spot, called a follow spot. In the follow spot, a brink between CNs can be similar to an 'automatic door.' Often found at supermarket entrances and exits, an automatic door will swing open when it senses that a person is approaching the door. The door has a sensor pad in front to detect the presence of a person about to walk through the doorway. Therefore, the door can be called an entrance door or entrance brink.[1][2]

3.3 Methods for Face Tracking

1. Topology Formation.
2. Predicting the target and creating Local Active environment.
3. Face Tracking
4. Modified Face Tracking protocol.

3.3.1 Topology Formation

All sensors are deployed initially. Each sensor updates their information to its neighbor sensor. This is called Initial Neighbor Discovery.[1][4]

3.3.2 Predicting the Target and Creating Local Active Environment

All sensors communicate with each other and updates the routing information once object is detected creates a Local Active environment predicts the Target movement and sends the information to base station. .[1][4]

3.3.3 Face Tracking

Once Target is detected creates an Awake region and based on the prediction results assigns Sleep scheduling to individual sensors at synchronized time and the graph is plotted for Energy efficiency in comparison with the Existing concept along with Throughput, Packet Delivery ratio. .[1][4]

3.3.4 Modified Face Tracking Protocol

In this phase we are synchronizing the proposed PPSS protocol, i.e., Local Active environment with Boundary selection nodes in which the sensors along the boundary of the field region are activated, thus the Mobile target that comes from different directions are detected, once it detects the Moving object along the boundaries, it will start sending the information about the mobile target to the base station, so we are enhancing the proposed concept to detect multiple target along with improved power efficiency.

IV. NETWORK COMPONENTS

The root of the hierarchy is the TclObject class that is the super class of all OTcl library objects (scheduler, network components, timers and the other objects including NAM related ones). As an ancestor class of

TclObject, NsObject class is the super class of all basic network component objects that handle packets, which may compose compound network objects such as nodes and links.

The basic network components are further divided into two subclasses, Connector and Classifier, based on the number of the possible output DATA paths. The basic network and objects that have only one output DATA path are under the Connector class, and switching objects that have possible multiple output DATA paths are under the Classifier class.[3]

V. INVOKING OTCL PROCEDURES

There are four different methods to invoke an OTcl command through the instance, tcl. They differ essentially in their calling arguments. Each function passes a string to the interpreter that then evaluates the string in a global context. These methods will return to the caller if the interpreter returns TCL_OK. On the other hand, if the interpreter returns TCL_ERROR, the methods will call tkerror{ }. The user can overload this procedure to selectively disregard certain types of errors.[4]

- 1. Passing Results to/from the Interpreter:** When the interpreter invokes a C++ method, it expects the result back in the private member variable, tcl-> result.
- 2. Error Reporting and Exit:** This method provides a uniform way to report errors in the compiled code.

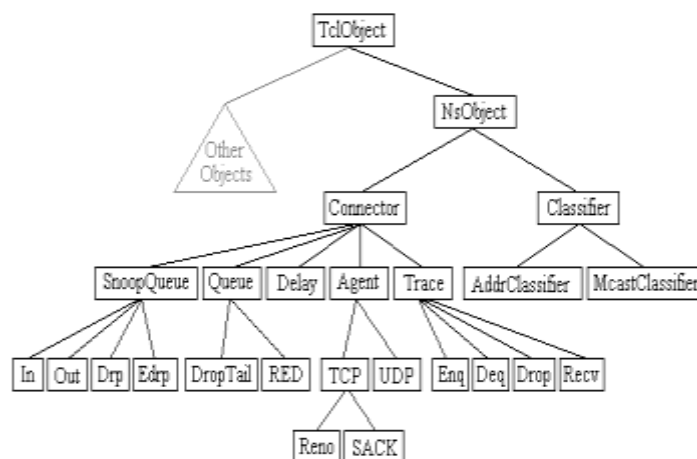


Fig 3. OTcl Class Hierarchy

VI. CONCLUSION

The main functionality of a surveillance wireless sensor network is to track an unauthorized target in a field. The challenge is to determine how to perceive the target in a WSN efficiently. We proposed a unique idea to achieve a WSN system for detecting movements of a target using polygon (face) tracking that does not adopt any prediction method. Evaluation results demonstrated that the proposed tracking framework can estimate a target's positioning area, achieve tracking ability with high accuracy, and reduce the energy cost of WSNs.

REFERENCES

- [1].** Detecting Movements of a Target Using Face Tracking in Wireless Sensor Networks , Guojun Wang, Member, IEEE, Md Zakirul Alam Bhuiyan, Member, IEEE, Jiannong Cao, Senior Member, IEEE , and Jie Wu, Fellow, Member IEEE.vol.25,no.4.

- [2]. Security Enhancements for Mobile Ad Hoc Networks with Trust Management Using Uncertain Reasoning
Zhexiong Wei, Helen Tang, F. Richard Yu, Maoyu Wang, and Peter Mason. DOI
10.1109/TVT.2014.2313865
- [3]. Evaluating Service Disciplines for On-Demand Mobile Data Collection in Sensor Networks Liang He,
Member, IEEE , Zhe Yang, Student Member, IEEE , Jianping Pan, Senior Member, IEEE , Lin Cai, Senior
Member, IEEE , Jingdong Xu, and Yu (Jason) Gu, Member, IEEE VOL. 13, NO. 4, APRIL 2014.
- [4]. Sleep Scheduling for Geographic Routing in Duty-Cycled Mobile Sensor Networks. Chunsheng Zhu,
Student Member, IEEE, Laurence T. Yang, Member, IEEE, Lei Shu, Member, IEEE, Victor C. M. Leung,
Fellow, IEEE, Joel J. P. C. Rodrigues, Senior Member, IEEE, and Lei Wang, Member, IEEE. VOL. 61,
NO. 11, NOVEMBER 2014
- [5] Z. Wang, W. Lou, Z. Wang, J. Ma, and H. Chen, "A Novel Mobility Management Scheme for Target
Tracking in Cluster-Based Sensor Networks," Proc. Sixth IEEE Int'l Conf. Distributed Computing in
Sensor Systems (DCOSS), pp. 172-186, 2010.
- [6] L.M. Kaplan, "Global Node Selection for Localization in a Distributed Sensor Network," IEEE Trans.
Aerospace and Electronic Systems, vol. 42, no. 1, pp. 113-135, Jan. 2006.