

On Mining Moving Patterns for Object Tracking Sensor Networks

Wen-Chih Peng, Yu-Zen Ko and Wang-Chien Lee[†]

Dept. of Computer Science
National Chiao Tung University
Hsinchu, Taiwan, ROC
email: yzkou,wcpeng@csie.nctu.edu.tw

Dept. of Computer Science and Engineering[†]
The Pennsylvania State University
University Park, PA 16802 USA
email: wlee@cse.psu.edu

Abstract

In this paper, we propose a heterogeneous tracking model, referred to as HTM, to efficiently mine object moving patterns and track objects. Specifically, we use a variable memory Markov model to exploit the dependencies among object movements. Furthermore, due to the hierarchical nature of HTM, multi-resolution object moving patterns are provided. The proposed HTM is able to accurately predict the movements of objects and thus reduces the energy consumption for object tracking. Simulation results show that HTM not only is able to effectively mine object moving patterns but also save energy in tracking objects.

1 Introduction

Various energy conservation schemes for object tracking sensor networks have been extensively studied in the literature [4][7]. In this paper, we focus on prediction-based object tracking sensor networks. The existing prediction-based object tracking sensor networks mostly used speeds and directions of objects to predict the next movement of objects. We argue that tracked objects such as human or animals tend to have their own moving patterns since the behaviors of human or animals are likely to be regular [3]. Thus, efficient techniques for obtaining moving patterns of objects are very important for energy conservation in object tracking sensor networks.

Our goal is to propose an efficient data mining mechanism for deriving object moving patterns and utilize the object moving patterns for energy saving prediction-based object tracking sensor networks. To facilitate collaborative data collection processing in object tracking sensor networks, cluster architectures are usually used to organize sensor nodes into clusters (with each cluster consisting of a cluster head and sensors). Similar to [6], we consider the sensor network which consists of heterogeneous sensor nodes of various functions and roles and thus propose a

heterogeneous tracking model (referred to as HTM) in the cluster architecture, in which a large number of inexpensive sensor nodes perform sensing operations and a limited number of powerful sensor nodes (standing for cluster heads) offer data collection, queries and mining capabilities. For scalability, the cluster heads (referred to as CH) recursively form a hierarchical architecture for efficiently mining and queries. As such, a multi-resolution data collection mechanism is naturally supported by our HTM model, where the higher-level cluster heads will maintain coarse object moving patterns and the low-level cluster heads will have more precise object moving patterns. Based on the obtained object moving patterns, the cluster heads predict the object movements. If the prediction fails, a recovery procedure will be executed by waking up sensor nodes within the coverage region of the cluster head. Based on HTM, only a bound number of sensor nodes need to participate in the recovery procedure.

The rest of the paper is organized as follows. Preliminary is described in Section 2. A detailed description of our proposal mechanism is presented in Section 3. Performance study is conducted in Section 4. This paper concludes with Section 5.

2 Preliminaries

Assume that all nodes (i.e., low-end sensor nodes and cluster heads) have unique sensor identifications and these sensor nodes are well time-synchronized. Suppose that each low-end sensor node is a logical representation of a set of sensor nodes which collaboratively detect an object. When a low-end sensor detects an object, this sensor node will inform the corresponding level-0 cluster head of the detected object identification, object arrival time and its sensor identification. In other words, the location of an object is represented as a sensor identification and the moving log for each object is viewed as a moving stream, which is composed of a series of symbols (i.e., sensor identifications). Even though the cluster heads are intrinsically powerful sensor

nodes, cluster heads still have the storage constraint. Furthermore, a monitored region may exist a large amount of objects. Thus, it is impossible for cluster heads to store the whole moving records for all objects and let alone scanning moving records multiple times for mining object moving patterns. Hence, in this paper, we shall address the problem of mining moving patterns in one scan. Notice that the movement of an object is in general dependent on the previous location of the object. Consequently, the variable memory Markov (referred to as VMM) model, which has been shown to be very effective in capturing dependences and in obtaining sequential models in one scan, is very appropriate for discovering object moving behavior.

Due to the storage constraint in cluster heads and the dependence of movements (i.e., the consecutive movement is highly dependent on the current location), mining object moving patterns can be regarded as a VMM model training. For each object, we use a variation of a suffix tree called *emission tree* [5] to maintain its VMM model and one corresponding buffer in a cluster head is used to hold the most recently segment of moving records. In order not to distract users from the main theme of this paper, interested readers are referred to [5] for the detailed procedure of constructing an emission tree.

VMM model is trained on the fly and not all the tree nodes are stable for predicting. Consequently, there are two nodes in an emission tree: one is the *mature node* and the other is *immature node*. Mature nodes are those tree nodes in which a certain amount of records are collected and the conditional probabilities are stable. When predicting the next movement, only mature node is participated in prediction.

To justify whether a node is mature or not, we explore L_∞ distance, which is defined as follows:

Definition 1: For a node y with the corresponding probability table denoted as x and the number of tuples in table x is n , $L_\infty(x, x') = \max(|d_1^x - d_1^{x'}|, |d_2^x - d_2^{x'}|, \dots, |d_n^x - d_n^{x'}|)$, where d_i^x represents the probability value of tuple i in table x and x' is the probability table after updating.

In light of Definition 1, for node y , if $L_\infty(x, x') \leq \alpha$ for β times of successive updates, node y becomes a mature node, where α and β , specified by system planners, are application dependent parameters.

3 In-Network Mining Moving Patterns

For each cluster head, the proposed prediction-based object tracking scheme consists two phases: 1) data collection and mining; and 2) prediction. In data collection and mining phase, all lower-end sensors will turn on and monitor the whole sensing region to collect moving records of objects. Once collecting enough moving records of objects,

CHs will be in prediction phase. If the prediction fails, sensor nodes cannot capture the location of the object. In this case, a recovery procedure will be executed in the cluster head. In what follows, we elaborate on each phase and the recovery procedure.

3.1 Data Collection and Mining Phase

At the beginning, since there are no moving records available, sensor nodes and cluster heads turn on their power and radio to monitor objects. When a low-end sensor detects a moving object, this sensor node will send a message containing the time and the identification of object sensed to the cluster head in charge of this sensor node. After receiving the detection message, the cluster head will have a moving record represented as (i, Obj_j, t) meaning that sensor i detects object Obj_j at time t . Given moving records of objects, each cluster head will exploit the VMM model for mining object moving patterns. Note that without collecting moving records in the sink, our proposal heterogeneous tracking model is able to achieve in-network mining in that mining object moving patterns is performed while moving records are forwarded to the sink. Furthermore, cluster heads at higher levels have coarse object moving patterns, whereas cluster heads at lower levels contain finer object moving patterns. Note that if an object Obj_i is in the coverage of cluster head CH_j , CH_j will predict the movement of Obj_i and justify the correctness of the prediction. To decide whether a cluster head can transit to the prediction phase or not, we define one threshold value to verify. If the average prediction hit rate is larger than δ , the cluster head will be in the prediction phase.

Note that since the object movements usually exhibit locality (i.e., the objects seldom move between high level clusters), the moving record incoming rate for upper level cluster heads is slower than that for lower level cluster heads. In order to speed up the VMM model training in upper level emission trees, we have to set smaller values of α and β for node maturity verification.

3.2 Prediction Phase

Consider an example in Figure 1, where Figure 1(a) shows the emission tree of Obj_i in CH_j and Figure 1(b) is the buffer of CH_j for Obj_i . Given the most recent moving records of the buffer, CH_j traverses the emission tree along the path $root \rightarrow node C \rightarrow node EC \rightarrow node FEC$. Assume that node FEC is not mature yet and the probability threshold for the determination of candidate sensor nodes is 0.5. Thus, the traversing terminates at node EC . The probability table of node EC shows that the sensor node B will be activated since the corresponding probability of node B satisfies the probability threshold (i.e., 0.5). If sensor node

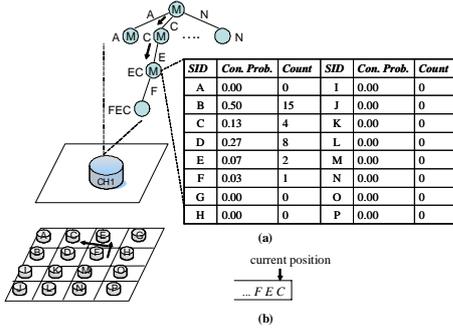


Figure 1. An example of utilizing an emission tree to predict the next movement of an object.

B is able to detect Obj_i , sensor node B will send an ACK message to CH_j and one moving record (i.e., (sid, Obj_i, t)), will be appended to the buffer for next location prediction. On the other hand, if sensor node B cannot detect Obj_i , the recovery process has to be started. Moreover, the cluster head appends an asterisk symbol to the buffer for Obj_i . From the above description, from the feedback of sensor nodes (i.e., Acknowledge message), though a cluster head is in the prediction phase, these information are beneficial to train VMM models as well.

3.3 Recovery Procedure

When cluster head CH_j cannot precisely predict the sensor nodes to detect object Obj_i , the recovery procedure will be performed to track object Obj_i . The possible location of object Obj_i can be conducted into two cases. One is that Obj_i is still in the coverage region of CH_j . The other is that object Obj_i moves into the coverage regions of other cluster heads. According to the observations above, cluster head CH_j will first check whether Obj_i is still in the coverage region of CH_j by waking up all sensor nodes under the coverage region of CH_j . If one of these sensor nodes captures Obj_i , the sensor node still sends an ACK message to CH_j . Otherwise, CH_j informs its higher level cluster head (i.e., the parent node of CH_j) since higher level clusters have coarse moving patterns of Obj_i used to predict possible cluster heads of object Obj_i . By traversing the emission tree of Obj_i and selecting the node with the highest probability, one cluster head, in which object Obj_i is likely to appear, will be determined. Then, the selected cluster head will downward to the lowest level cluster heads to estimate the fine moving patterns of object Obj_i . If the object is still not found in the selected cluster head, the parent cluster head will select other cluster heads to perform the recovery procedure. If none of these cluster heads finds the missing

object, the parent cluster head will forward the upper level cluster head for the recovery purpose. As mentioned before, each level cluster head has different resolution object moving patterns. In light of these multi-resolution object moving patterns, only a bounded number of sensor nodes activate for the recovery procedure, showing the strength of our proposal heterogeneous tracking model.

4 Performance Study

4.1 Simulation Model

In the experiment, we consider a three-level heterogeneous tracking model, where 9 low-end sensors are deployed in each level-0 cluster. Hence, there are 16 level-0 CHs, 4 level-1 CHs, one level-2 CH, and the number of low-end sensors is 144. To simulate the object movements, we generate VMM model trees for each object in each cluster head. In addition, the city mobility model [1][2] is used to simulate object movements with locality. With the model, each object has a probability p_1 to determine whether it should leave its current level-1 cluster, and a probability $1 - p_1$ to stay. In the former case, the object will choose a level-1 cluster as the next position according to its VMM model tree in the level-2 CH (It may stay in the current level-1 cluster). In the latter case, the object has a probability p_0 to determine whether it should leave its current level-0 cluster, and a probability $1 - p_0$ to stay. Similarly, in the former case, the object will choose a level-0 cluster as the next position according to its VMM model tree in the parent. In the latter case, the object will stay in its current level-0 cluster. In all cases above, the VMM model looking up procedure is repeated until the object has decided to move to which low-end sensor monitored region. The probability p_i is determined by an exponential probability $p_i = e^{-C \cdot 2^{i+1}}$, where C is a positive constant. A higher value of C means higher locality. The value of δ used to justify whether the cluster head shall be in the prediction phase is set to 0.5 and the probability threshold for the number of sensors in the prediction phase (i.e., γ) is set to 0.2.

4.2 Experiments of PES and HTM

To conduct the experiments of PES scheme in [4], we set the pause time to 5 seconds (i.e., each object will change its speed and direction every 5 seconds) and employ the INSTANT heuristic for prediction. The sampling and reporting frequency are once per second. As mentioned earlier, once the prediction is not correct, the recovery procedure will be performed by waking up sensor nodes. The number of sensor nodes waken up under PES and HTM is shown in Figure 2. In Figure 2, the number of sensor nodes waken by HTM is significantly smaller than that of PES. By utilizing

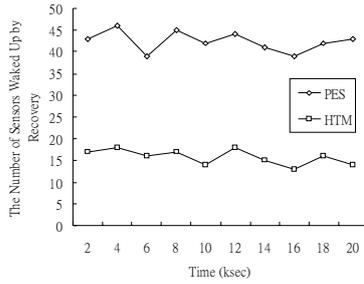


Figure 2. The average number of nodes participated in the recovery procedure.

object moving patterns, HTM is able to accurately predict the movements of objects and then the number of executing the recovery procedure is small. Furthermore, due to the very hierarchical nature of HTM, once the recovery procedure is performed, only a bound number of sensors activate for tracking objects.

4.3 Sensitivity Analysis of HTM

To examine the impact of locality parameter C on the prediction hit rate of HTM, we set β to 10. Each CH can only store 500 emission tree nodes. Figure 3 shows the average prediction hit rate of level-0 cluster heads. It can be seen in Figure 3 that the prediction hit rate tends to increase as the value of C increases. Note that with higher value of C , the movements of objects tend to move nearby such that CHs can obtain more moving records for mining. In other words, objects with higher value of C will reflect significant regular moving behaviors and thus in light of object moving patterns, CHs are able to precisely predict the movements of objects. Furthermore, with smaller value of α , the number of mature nodes in an emission tree tends to increase, leading to a higher prediction hit ratio.

5 Conclusions

In this paper, based on the fact that the movements of the tracked objects generally exhibit periodic behaviors, we proposed a heterogeneous tracking model, called HTM, to efficiently mine object moving patterns and track objects. Specifically, we use a variable memory Markov model to mine object moving patterns. Furthermore, due to the hierarchical nature of HTM, multi-resolution object moving patterns are provided. In light of object moving patterns obtained, our proposed HTM is able to accurately predict the movements of objects and thus reduces the energy consumption for object tracking. Performance of the proposed

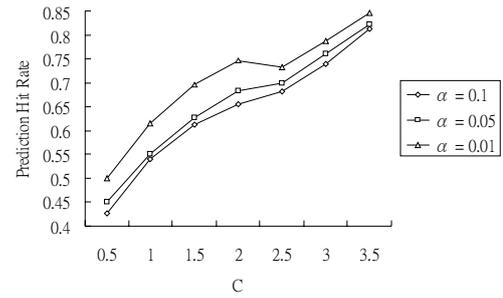


Figure 3. The impact of locality parameter on the prediction hit rate.

HTM was analyzed and sensitivity analysis on several design parameters was conducted. Simulation results showed that HTM is able to not only effectively mine object moving patterns but also efficiently track objects.

References

- [1] H. T. Kung and D. Vlah. Efficient Location Tracking Using Sensor Networks. In *Proceeding of 2003 IEEE Wireless Communications and Networking Conference(WCNC)*, pages 16–20, March 2003.
- [2] C.-Y. Lin and Y.-C. Tseng. Structures for in-network moving object tracking in wireless sensor networks. In *BROADNETS*, pages 718–727. IEEE Computer Society, 2004.
- [3] W.-C. Peng and M.-S. Chen. Developing Data Allocation Schemes by Incremental Mining of User Moving Patterns in a Mobile Computing System. In *Proceeding of IEEE Transactions on Knowledge and Data Engineering, Volume 15*, pages 70–85, 2003.
- [4] Y. Xu, J. Winter, and W. C. Lee. Prediction-based Strategies for Energy Saving in Object Tracking Sensor Networks. In *Proceedings of the 2004 IEEE International Conference on Mobile Data Management(MDM'04)*, 2004.
- [5] J. Yang and W. Wang. Agile: A general approach to detect transitions in evolving data streams. In *ICDM*, pages 559–562. IEEE Computer Society, 2004.
- [6] M. Yarvis, N. Kushalnagar, H. Singh, A. Rangarajan, Y. Liu, and S. Singh. Exploiting heterogeneity in sensor networks. In *Proceedings of 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2005)*, 2005.
- [7] W. Zhang and G. Cao. DCTC: Dynamic Convoy Tree-Based Collaboration for Target Tracking in Sensor Networks. *IEEE Transactions on Wireless Communication*, September 2004.