International Journal of Signal Processing, Image Processing and Pattern Recognition
Vol.7, No.2 (2014), pp.345-354
http://dx.doi.org/10.14257/ijsip.2014.7.2.32

Classification of RFID-enabled Trajectory using Pattern Recognition Approach

Ying Shen[1] and Weihua Zhu[2]

[1]Department of Engineering, Xinyu College, Xinyu, Jiangxi, China
[2]Department of Computer Science, Xinyu College, Xinyu, Jiangxi, China
Erinshen2010@163.com

Abstract

RFID technology has been widely used in many fields. Lots of information could be obtained from the enormous RFID devices. Therefore, classification of the trajectory of these moving objects is significant to predict the trends of the moving objects. This paper introduces an innovative algorithm, using pattern recognition approach. This algorithm could be divided into several steps. First of all, the coarse-fine layer classification approach is used for clustering the trajectories. That aims to cut down the trajectories according to the difference phases. Secondly, in the sub-set of the trajectory, searching set is established through detecting the neighbor domains which have been classified at different phases. Finally, the hierarchical classification approach is used for classifying the sub-trajectory. By using the approach, the experimental results imply the feasibility and practicality of the proposed approach on figuring out the familiar trajectory of the RFID-enabled moving objects. It is observed that With the increasing of neighbor value , the linear trends is obvious from the figure, thus, the neighbor query decreases and the magnification ratio outperforms to TRACLUS method. Additionally, the proposed algorithm uses coarse-fine strategy at phases-to-phases, saving the time spend on large number of distance calculation at various phase.

Keywords: *We would like to encourage you to list your keywords in this section*

1. Introduction

RFID technology has been widely used in many fields like real-time manufacturing, logistics control and supply chain management, retailing, and transportation supervision [1-3]. Thus, lots of information could be obtained from the enormous RFID devices [4]. RFID technology is able to track and trace the moving objects in the logistics network via the trajectory. Therefore, classification of the trajectory of these moving objects is significant to predict the trends of the moving objects.

Classification approaches are widely studied due to its great significance in pattern recognition [5]. However, traditional methods usually focus on the trajectories of whole moving objects. That may miss some familiar trajectories. In practical aspects, certain specific areas might be focused by the users for making various decisions [6]. Thus, it is important to figure out the familiar trajectories in specific area.

In order to address this practical challenge, this paper introduces an innovative algorithm, using pattern recognition approach. This algorithm could be divided into several steps. First of all, the coarse-fine layer classification approach is used for clustering the trajectories. That aims to cut down the trajectories according to the difference phases. Secondly, in the sub-set of the trajectory, searching set is established through detecting the neighbor domains which

ISSN: 2005-4254 IJSIP

have been classified at different phases. Finally, the hierarchical classification approach is used for classifying the sub-trajectory. By using the approach, the experimental results imply the feasibility and practicality of the proposed approach on figuring out the familiar trajectory of the RFID-enabled moving objects.

The rest of this paper is organized as follows. In Section 2, the problem description is reported by giving the definition and the distance of two trajectories. In Section 3, the proposed algorithm is demonstrated based on the pattern recognition methods in terms of coarseness classification, fineness classification, and detailed algorithms. The classification of trajectories is illustrated by establishing the graphs using the proposed pattern recognition method in Section 4. Experiments and results are discussed in Section 5 to verify the neighbor query times and run-time from comparing the proposed method with a traditional classification method. Finally, in Section 6, conclusions are presented and future work is highlighted to finish this paper.

2. Problem Description

The classification of RFID-enabled trajectory could be descripted by a set of definitions and the distance measurement. This section gives the details as follows.

2.1. Definitions

Given the set of moving objects trajectory, $T = \{T_1^r, T_2^r, \ldots, T_n^r\}$, n is the total number of trajectories. C is a set which is generated by the proposed algorithm, $C = \{C_1, C_2, \ldots, C_m\}$.

According to the description, there are several definitions:

Definition 1: (Sub-trajectory). The trajectory of a moving object is a sequence which contains a rich set of point with multi-dimensional perspectives. It could be expressed as $T_i^r = p_1 p_2 p_3 \ldots p_i \ldots p_L$, L is the length of T_i^r. In order to classify the trajectory, a phase of a trajectory $p_{c1} p_{c2} p_{c3} \ldots p_{ck}$ ($1 \le c1 < c2 < \ldots < ck \le L$) is defined as a sub-trajectory of T_i^r. All the phases of T_i^r is expressed as $P(T_i^r)$.

Definition 2: (Trajectory Cluster). Trajectory cluster is a set of trajectory phases. A trajectory could be divided into several sub-phases $p_i p_j$ ($i < j$), p_i and p_j belong to a same trajectory. A trajectory cluster is a set of sub-phases, which are belonged to a same trajectory [7]. They are clustered given the distance measurement like neighbors. Since a trajectory could be divided into several phases, difference sub-phases could be included in different trajectory cluster. Thus, a trajectory may be included in many clusters.

2.2. Distance Measurement

The distance of trajectory is measured using the classic method in pattern recognition. The distance measurement function $Dist$ contains three key elements: $d_{\perp}$, d_{Θ} and d_{Λ}. $d_{\perp}$ is a measurement for the vertical distance; d_{Θ} is for the parallel distance; and d_{Λ} is for measuring the angular distance.

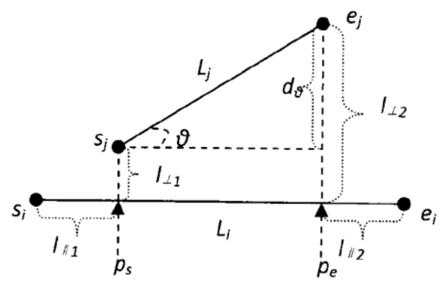

Figure 1. Distance Measurement of Trajectory Phases

Figure 1 illustrates the distance measurement of trajectory phases. The distances are defined as follows:

Definition 3: (Vertical Distance $d_\perp$). p_s and p_e are the projection points of s_j and e_j in L_i. $l_{\perp 1}$ is the Euclidean distance of s_j and p_s. $l_{\perp 1}$ is the Euclidean distance of e_j and p_e. Then the vertical distance $d_\perp(L_i, L_j)$ could be obtained from:

$$d_\perp(L_i, L_j) = \frac{l_{\perp 1}^2 + l_{\perp 2}^2}{l_{\perp 1} + l_{\perp 2}} \tag{1}$$

Definition 4: (Parallel Distance d_Θ). $l_{\Theta 1}$ is the minimal Euclidean distance from p_s to the points of s_i and e_i. Similarly, $l_{\Theta 2}$ is the maximum Euclidean distance from p_e to the points of s_i and e_i. Thus, the d_Θ could be defined as:

$$d_\Theta(L_i, L_j) = \min(l_{\Theta 1}, l_{\Theta 2}) \tag{2}$$

Definition 5: (Angular Distance d_Λ). $|L_j|$ is the length of L_j, θ is the minimal angle of L_i and L_j $0^\circ \le \theta \le 180^\circ$. Thus, the angular distance d_Λ could be calculated as:

$$d_\Lambda(L_i, L_j) = \begin{cases} |L_j| \times \sin\theta, if & 0^\circ \le \theta \le 90^\circ \\ |L_j|, if & 90^\circ \le \theta \le 180^\circ \end{cases} \tag{3}$$

From the definition, the formula (1) (2) (3) contain the distance measurement from three dimensions. Vertical distance $d_\perp$ is calculated from the location distance of different trajectory where the sub-phases are picked out. Parallel distance d_Θ is got from the location distance of same trajectory where the sub-phases are chosen. The parallel distance of two neighbor sub-phases is zero permanently. Angular distance is the different of directions of different sub-phases. When the directions of sub-phases are obviously different, angular distance d_Λ is the length of L_j, $d_\Lambda = |L_j|$. Therefore, the distance of two phases is the sum of these three dimensions:

$$Dist(L_i, L_j) = d_\perp + d_\Theta + d_\Lambda \tag{4}$$

3. Classification of RFID-enabled Trajectory

The aim of classification of RFID-enabled trajectory is to find out the points which indicate the changes of directions and trends of trajectories. These points are named as characteristic points [8]. In order to speed up the classification, this paper defines two types of classifications using the pattern recognition approach. They are coarseness and fineness classification. The purposes are to figure out the sub-phases which are required and meet the constraints as well as to reduce the data space caused by fine division.

3.1. Coarseness Classification

In order to ensure the accuracy and practicality, MDL (Minimum Description Length) is used for coarseness classification. MDL contains two parts. First is the L(H) part, which is the sum of sub-phases' length, and the other is L(D|H) which is the sum of differences of the sub-phases. Both parts could be got from the following formula.

$$L(H) = \sum_{j=1}^{L-1} \log_2(len(p_{cj} p_{cj+1})) \tag{5}$$

$$L(D \mid H) = \sum_{j=1}^{L-1} \sum_{k=c_j}^{c_{j+1}-1} \{\log_2(d_{\perp}(p_{cj} p_{cj+1}, p_k p_{k+1})) + \log_2(d_{\Lambda}(p_{cj} p_{cj+1}, p_k p_{k+1}))\} \tag{6}$$

According to the (5) (6), the algorithm of coarseness classification is descripted as follows:

Input: $T = \{T_1^r, T_2^r, \ldots, T_n^r\}$, $T_i^r = p_1 p_2 p_3 \cdots p_i \cdots p_L$

Output: $C = \{C_1, C_2, \ldots, C_m\}$

Algorithm of Coarseness Classification:

(1) CP=$NULL$;
(2) for each $TR_i \in T$ do
(3) { StarIndex=1;
(4) length=1;
(5) CP_i={ p_1};
(6) while StartIndex+length≤leni do
(7) { curr=StarIndex+length;
(8) compute L(H) and L(D|H) of line segments $p_{startIndex}\ p_{currIndex}$ for no partitioning;
(9) s= L(H) + L(D|H);
(10) compute L(H) and L(D|H) of line segments $p_{startIndex}\ p_{currIndex}$ for partitioning;
(11) s'= L(H) + L(D|H);
(12) if s< s' then
(13) { CP_i= CP_i+{ $p_{currIndex-1}$ };
(14) StartIndex=curr-1;
(15) length=1;
(16) }
(17) else length=length+1;
(18) }
(19) CP_i= CP_i+{ p_{leni} } ;
(20) CP = CP +CP_i;
(21) }
(22) each line segment L in C denoted as p_jp_{j+k*b} in CP (b is the base unit);

3.2. Fineness Classification

The fineness classification aims to figure out the upper and down boundary of the sub-phases' distances. After that, a cutting strategy could be worked out so as to reduce the time costs. To this end, it is assume that if $lb(L_i, L_j, Dist) > \varepsilon$, ε is the threshold of distance, it is no need to carry out the fineness classification comparing with the coarseness of L_i, L_j. If $ub(L_i, L_j, Dist) \le \varepsilon$, after fineness classification of L_i, L_j , it is no need to calculate the distance of fineness sub-phases.

The fineness classification algorithm based on the assumption is given as follows:

Input: $C = \{C_1, C_2, ..., C_m\}$ from coarseness classification

Output: $F = \{l_1, l_2.,,, l_n\}$ the set of fineness classification with sub-phases

Algorithms:

(1) F=null;

(2) for each pair of $L_i, L_j \in C(L_i \neq L_j)$ do

(3) { if $lb(L_i, L_j, dist) > \varepsilon$ then break;

(4) else fine partition L_i ,L_j , and insert fine line segments into F;

(5) if $ub(L_i, L_j, dist) \leqslant \varepsilon$ then

(6) $\forall\ l_i \in L_i$ and $\forall\ l_j \in L_j$

(7) $N(l_i) = N(l_i) + \{l_j\}$; $N(l_j) = N(l_j) + \{l_i\}$;

(8) else for each pair of $l_i \in L_i, l_j \in L_j$ do

(9) if $dist(l_i, l_j) \leqslant \varepsilon$ then

(10) do as line 6,7

4. RFID-enabled Trajectory Classification using Pattern Recognition Approach

Using the pattern recognition approach in classifying the RFID-enabled trajectory, this section reports on the entire method based on the coarseness and fineness classification. This method is based on graph theory. Several steps are detailed as follows.

4.1. Establishment of Graph G

Graph G could be established and each node presents a sub-phase of trajectory in the data set. If the sub-phase distance from l_i to l_j is the minimal value within the k distances of various nodes to l_j, l_i is the nearest k neighbor object of l_j. These two nodes have an edge with a weight ω, which indicates the familiarity of the sub-phases. If ω is smaller, the distances are larger. According to this rules, a graph G with lots of nodes and edges could be established. The establishment is to find out the nearest k neighbor objects from F and then add the weighted edges to them.

4.2. Division of Graph G

The division of G is to divide the graph into several sub-graphs which do not have the connections. Each sub-graph is used for the layer clustering. Firstly, the graph G is divided into two sub-graphs with familiar sizes. These two sub-graphs have to meet the requirements of minimal sum of weight of all the edges in both graphs. This step is named minimal cutting-down. Secondly, each sub-graph is regarded as an initial graph, which the first step is carried on the graph. The processes are carried on until all the nodes in sub-graphs meeting a certain criteria.

4.3. Mergence

The mergence operation is based on the layered clustering approach which is implemented on the initial cluster from the previous set. The connection and similarity of C_i and C_j are used for deciding the degree of the friendship of both sub-clusters. The friendship function is defined as:

$$RI(c_i, c_j) \times RC(c_i, c_j) \tag{7}$$

Given the mergence operation, there are several definitions to ease the classification algorithm using pattern recognition.

Definition 6: (Interconnection $RI(c_i, c_j)$). Sub-cluster c_i, c_j have the interconnection relation which is defined as:

$$RI(c_i, c_j) = \frac{|EC(c_i, c)|}{\frac{1}{2}(|EC(c_i)| + EC(c_j))} \tag{8}$$

$EC(c_j)$ is the sum of ω from the deleted edges when carrying out minimal cutting-down. $EC(c_i, c_j)$ is the sum of sub-cluster which connects c_i and c_j.

Definition 7: (Relative Similarity $RC(c_i, c_j)$). The relative similarity of two sub-clusters c_i and c_j is defined as:

$$RC(c_i, c_j) = \frac{TC(c_i, c_j)}{\frac{|c_i|}{|c_i| + |c_j|} \times TC(c_i) + \frac{|c_j|}{|c_i| + |c_j|} \times TC(c_j)} \tag{9}$$

$TC(c_i)$ is the average weight of the edges which are deleted when carrying out minimal cutting-down process.

4.4. Classification Algorithm

Based on the interconnection and relative similarity of the sub-clusters, the classification algorithm based on the pattern recognition is as follows:

Input: $T = \{T_1^r, T_2^r, \ldots, T_n^r\}$, $T_i^r = p_1 p_2 p_3 \ldots p_i \ldots p_L$

Output: $O = \{c_1, c_2, \ldots c_{numc}\}$

Algorithm:

```
for each TRi   ∈T do
    execute   coarse_partition; get a set of coarse line segments C;
    execute   prune_segments; get a set of fine line segments F;
CreatKnnGraph();
Partition(CLUSTERING C)
{ Bisection(C, C1, C2, minctw, minsum)   }
  (1) Merge(C[ ],C1[ ])
  (2) {   compute similarity(C1,C2)
  (3)       {   abs_conclose(C1,m11,m12);
  (4)        abs_conclose(C2,C21,m22);
  (5)        inter_conclose(C1,C2,n1,n2)
  (6)        k1=len(C1); k2=len(C2);
  (7)        r1=2*n1/(m11+m21);
  (8)        r2=(k1/(k1+k2))*m12+(k2/(k1+k2))*m22;
  (9)          sim=r1*r2;
  (10)         return   sim;      }   }
```

5. Experiments and Discussions

The experiments are carried out in Pentium IV 2 GHz CPU, 2 GB Memory, the operation system is Windows XP Professional SP2. We examine the TRACLUS algorithm with the proposed algorithm in this paper to compare the experimental results.

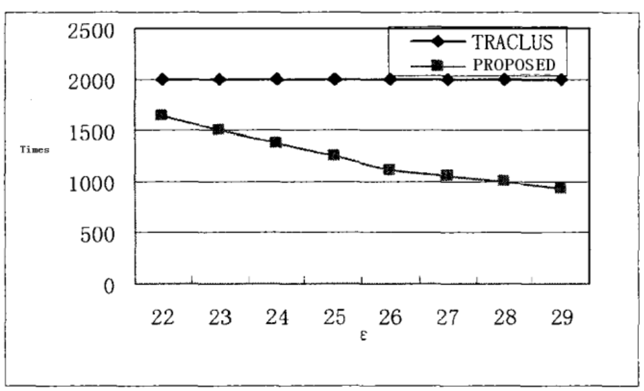

Figure 2. Query Times Comparison in Neighbor Sub-phases

From Figure 2, keeping the MinLns, the proposed algorithm in this paper greatly reduces the times spent on querying in neighbor sub-phases. TRACLUS algorithm has to query the neighbor domain for each sub-phase [9]. It is observed that no matter the value of ε , the query times will not change. However, for the proposed algorithm, due to the use of coarseness and fineness classification based on pattern recognition, the query times are significantly reduced. The cutting-down operation attributes the reduction of calculation time when the ε increases. With the increasing of ε, the linear trends is obvious from the figure, thus, the neighbor query decreases and the magnification ratio outperforms to TRACLUS method.

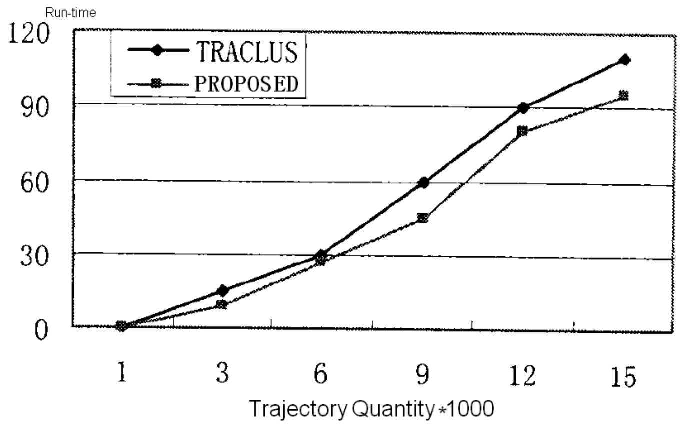

Figure 3. Comparisons of Run-time

From the Figure 3, the proposed algorithm in this paper outperforms to TRACLUS comparing the time spent on dealing with the trajectory classification. The proposed algorithm uses coarse-fine strategy at phases-to-phases, saving the time spend on large number of distance calculation at various phase. Additionally, the pattern recognition based method reduces the sub-phase set of the trajectory. It is easy for execute the classification and speeds up the execution time. However, TRACLUS algorithm uses the DBSCAN approach to work out the similar trajectory and sub-trajectory [10]. It is sensitive to parameters, thus, the results are greatly impacted by the parameters. While, the proposed algorithm uses the layered classification manner, the results are relative independent to the parameter. Thus, the time costs are relative low.

6. Summary

This paper is based on a pattern recognition approach to figure out the RFID-enabled trajectory which is classified so as to find the characteristics and predict the moving tends of the objects. The aim is to fulfill the gap of inefficiency of classification approach on RFID-enabled trajectory. Currently, the algorithms used for classifying the datasets are relative weak due to the low granularity of enormous RFID data.

The proposed algorithm is compared with TRACLUS in terms of query time and execution times. From the experiment results, it is observed that With the increasing of ε, the linear trends is obvious from the figure, thus, the neighbor query decreases and the magnification ratio outperforms to TRACLUS method. Additionally, the proposed algorithm uses coarse-fine strategy at phases-to-phases, saving the time spend on large number of distance calculation at various phase.

Future work could be carried out from the following perspectives. First of all, the pattern recognition-based algorithm could be extended in other fields like routing and scheduling. The problems are similar with the objective functions and constraints. Secondly, the coarseness and fineness classification could be improved by adding more distances measurements since only the Euclidean distance is considered in this paper. And other distances could be extended like Hausdorff distance *etc*.

References

[1] R. Y. Zhong, Z. Li, A. L. Y. Pang, Y. Pan, T. Qu and G. Q. Huang, "RFID-Enabled Real-Time Advanced Planning and Scheduling Shell for Production Decision-Making", International Journal of Computer Integrated Manufacturing, vol. 26, no. 7, **(2013)**, pp. 649-62.

[2] G. Ferrer, S. Heath and N. Dew, "An RFID Application in Large Job Shop Remanufacturing Operations", International Journal of Production Economics, vol. 133, no. 2, **(2011)**, pp. 612-21.

[3] R. Y. Zhong, G. Q. Huang, Q. Y. Dai and T. Zhang, "Mining Sots and Dispatching Rules from RFID-Enabled Real-Time Shopfloor Production Data", Journal of Intelligent Manufacturing, DOI: 10.1007/s10845-012-0721-y, **(2013)**.

[4] G. Q. Huang, T. Qu, Y. F. Zhang and H. D. Yang, "RFID-Enabled Product-Service System for Automotive Part and Accessory Manufacturing Alliances", International Journal of Production Research, vol. 50, no. 14, **(2012)**, pp. 3821-40.

[5] R. Y. Zhong, L. Y. Pang, Y. Pan, T. Qu and G. Q. Huang, "Rapshell for RFID-Enabled Real-Time Shopfloor Production Planning, Scheduling and Execution", Paper presented at the Proceeding of the Computers and Industrial Engineering, Cape Town, South Africa, vol. 42, **(2012)** July 16-18, pp. 28-1-28-10.

[6] M. L. Wang, T. Qu, R. Y. Zhong, Q.Y. Dai, X. W. Zhang and J. B. He, "A Radio Frequency Identification-Enabled Real-Time Manufacturing Execution System for One-of-a-Kind Production Manufacturing: A Case Study in Mould Industry", International Journal of Computer Integrated Manufacturing, vol. 25, no. 1, **(2012)**, pp. 20-34.

[7] R. Y. Zhong, Q. Y. Dai, T. Qu, G. J. Hu and G. Q. Huang, "Rfid-Enabled Real-Time Manufacturing Execution System for Mass-Customization Production", Robotics and Computer-Integrated Manufacturing vol. 29, no. 2, **(2013)**, pp. 283-92.

[8] M. L. Wang, Q. Y. Dai, R. Y. Zhong and G. Q. Huang, "RFID-Enabled Real-Time Mechanical Workshop Training Center", International Journal of Engineering Education, vol. 28, no. 5, **(2012)**, pp. 1199-212.

[9] R. Y. Zhong, G. Q. Huang, Q. Y. Dai, K. Zhou, T. Qu and G. J. Hu, "RFID-Enabled Real-Time Manufacturing Execution System for Discrete Manufacturing: Software Design and Implementation", 2011 IEEE International Conference on Networking, Sensing and Control (ICNSC), **(2011)**.

[10] R. Y. Zhong, Q.Y. Dai, K. Zhou and X. B. Dai, "Design and Implementation of DMES Based on RFID", 2nd International Conference on Anti-counterfeiting, Security and Identification, Guiyang, **(2008)** August 20-23, pp. 475-477.

Authors

Ying Shen, an associate professor from department of engineering, Xinyu College, got the master degree of communication and information system in 2003 from Nan Chang University. She has been published a rich set of related papers in international journals and conferences. Her main research area is digital image processing.

Weihua Zhu, a senior lecture from department of Computer Science, got the master degree of communication and information system from Nan Chang University. He has been published a large number of articles in international journals and conferences. His main research areas are algorithm design and embedded system.

