

Efficient RFID Data Imputation by Analyzing the Correlations of Monitored Objects

Yu Gu¹, Ge Yu¹, Yueguo Chen², Beng Chin Ooi²

¹ Northeastern University, China
{guyu, yuge}@ise.neu.edu.cn

² National University of Singapore, Singapore
{chenyueg, ooibc}@comp.nus.edu.sg

Abstract. As a promising technology for tracing the product and human flows, Radio Frequency Identification (RFID) has received much attention within database community. However, the problem of missing readings restricts the application of RFID. Some RFID data cleaning algorithms have therefore been proposed to address this problem. Nevertheless, most of them fill up missing readings simply based on the historical readings of independent monitored objects. While, the correlations (spatio-temporal closeness) among the monitored objects are ignored. We observe that the spatio-temporal correlations of monitored objects are very useful for imputing the missing RFID readings. In this paper, we propose a data imputation model for RFID by efficiently maintaining and analyzing the correlations of the monitored objects. Optimized data structures and imputation strategies are developed. Extensive simulated experiments have demonstrated the effectiveness of the proposed algorithms.

1 Introduction

Radio Frequency Identification (RFID) [1, 2] is experiencing fast developments in recent years. Today, an increasing number of products in our everyday life are attached with RFIDs, which provide fast accessibility of specifications as well as spatio-temporal information of the products. A RFID application is mainly composed of readers and tags. Readers are transponders capable of detecting tags within a distance from them. Tags are attached to the monitored objects. They can either actively transmit or passively respond to a unique identification code when they are close enough to a reader. By means of RFID technology, objects in the physical world can be easily identified, catalogued and tracked. With the tracking function, RFID can be widely applied in applications such as supply chain management [3], human activity monitoring and control [4], etc.

In RFID tracking applications, a huge amount of RFID readings generate a large number of rapid data streams containing spatio-temporal information of the monitored objects. However, there are many dirty data within the spatio-temporal RFID streams. Especially, missing readings are very common because of various factors such as RF collision, environmental interruption, and metal

disturbance. Therefore, the RFID data streams usually contain a large percentage of defective spatial-temporal information. Data imputation, as a kind of data cleaning techniques, is quite important to improve the quality of RFID readings.

Existing studies [5,6] on physical RFID data imputation focus mainly on the analysis of historical readings of independent monitored objects, and ignore the correlations of trajectories of monitored objects. We observe that in many RFID applications, objects are being perceived as moving in swarms. Groups of the monitored objects such as humans, vehicles, herds and commodities are often moving together within many local tracks. For example, in a smart RFID museum scenario, many visitors are expected to move around the premises with their friends or family. Obviously, the positions of partners within a group are very useful for estimating the positions of missing RFID tags within the group. However, there are three important challenges in applying such group moving information to RFID data imputation:

- **Mutation.** The partnership of the monitored objects may change over time. For example, visitors may re-choose their partners because of the mutative interests about exhibition in the museum scenario.
- **Chaos.** People or objects in different groups may accidentally merge at the same location simultaneously. For example, a special show will attract unacquainted visitors at the same time.
- **Ambiguity.** When readers cannot obtain a reading from a tagged object in the monitoring space, we say an **empty reading** occurs. An empty reading can be generated from two cases: **missing reading** (readers fail to read a tag although the tag is covered by their sensing regions) or **vacant reading** (happens when the tagged object is not covered by any sensing region of a reader). It is difficult to identify whether an empty reading is a missing reading or a vacant one.

Figure 1 illustrates the above challenges. The circle and rectangle represent people from different organizations. *Chaos* can be found in region1 where o_1 and o_3 gather. When o_1 moves to region2, a *mutation* occurs because o_1 's partner will change to o_5 . *Ambiguity* exists between o_6 and o_7 when they both are not read by readers since we cannot distinguish whether it is because the tags have been moved out the sensing regions of readers (vacant readings) or they are disturbed (missing readings).

To our knowledge, this is the first paper that seeks to utilize the grouped trajectory information of monitored objects for RFID data imputation. Algorithms on handling the above challenges are proposed. The paper is organized as follows. Section 2 outlines the related work. Section 3 introduces the correlation model of RFID monitored objects. Section 4 proposes the techniques to efficiently maintain the correlations. Section 5 illustrates the optimized data imputation methods for improving accuracy. Section 6 provides the experimental analysis and Section 7 concludes the paper.

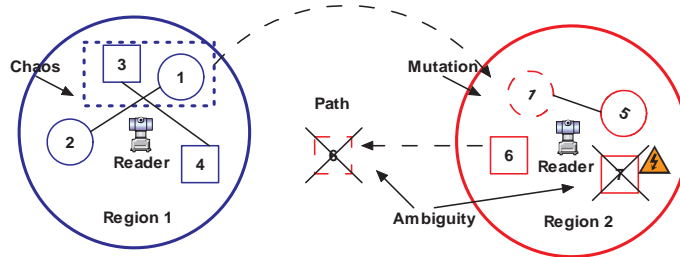


Fig. 1. Challenges for using group moving information.

2 Related Work

The proliferation of RFID has generated new problems for database community. Many data cleaning techniques [5, 6, 8, 9] have been proposed to improve the accuracy of RFID data collected from noisy environments. One important goal of RFID data cleaning is to fill up the missing readings in noisy communication conditions. For such data imputation based on physical readings, one commonly used technique is smoothing filter. All missing readings will be filled up if one reading is successfully read within one smoothing window. For example, a pipeline framework, which allows users to set the size of the smoothing window, is used to clean various dirty data in [5]. However, a fixed smoothing window size cannot well capture the dynamics of tag motion. A statistics-based approach is proposed in [6] to dynamically adjust the size of smoothing windows. Whereas, the method is only applicable to the situation of a single reader due to the limitation of the statistics model. For a general RFID application with multiple regions across the space, the statistics model will be ineffective. The cost optimization problem for historical readings based methods is discussed in [7], when additional context features and transition models are supposed to be predicted.

Different from the physical data imputation, some work has focused on cleaning data related to specific application semantics. For example, Khoussainova et al. [8] propose a cleaning strategy based on a probabilistic model. Rao et al. [9] propose to delay the data imputation to the time when queries are issued. The definition of dirty data is associated with specific applications. Moreover, it can only be applied to static data in a database, and not to the RFID streaming applications. Our spatio-temporal closeness model is also substantially different from the one in general sensor networks [10] as our model is used in the context of RFID, where moving objects are tracked.

3 RFID Monitored Objects Correlation Model

For simplicity, we assume that an RFID reader r periodically senses readings from a tagged object o if o is in the sensing region of r . Suppose R, O represent

the sets of reader IDs and object IDs respectively. A RFID reading is modelled as a ternary tuple $p = \langle i \in R, j \in O, t \rangle$, representing an object o_i is detected by a reader r_j at time stamp t . In our RFID model, we assume that RFID readers are independent, i.e., no sensing regions of two readers cover with each other. The sensing region of a reader r_i is called a logic region \mathcal{Y}_i . The space that is not covered by any logic region is called the dead region, denoted as $\tilde{\mathcal{Y}}$. Those tagged objects in any logic region are supposed to be detected by a reader if no missing readings occur. While, those tagged objects will not be sensed by any readers if they are in $\tilde{\mathcal{Y}}$.

All the detected objects within \mathcal{Y}_i at time stamp t is denoted as $\mathcal{Y}_i(t) = \{o_j | \exists p = \langle i, j, t \rangle\}$, While $\tilde{\mathcal{Y}}(t)$ represents all the tagged objects in the dead region at time stamp t . We define all the objects with empty readings at time t as $\mathcal{O}(t)$ and all the objects with missing readings at time t as $\phi(t)$. Therefore, we have $\mathcal{O}(t) = \phi(t) \cup \tilde{\mathcal{Y}}(t)$. If we use $\Delta(t)$ to represent all the tagged objects in the RFID working space at time t , we have $\Delta(t) = \mathcal{O}(t) \cup (\bigcup_i \mathcal{Y}_i(t))$. Note that $\Delta(t)$ is dynamic because tagged objects join and leave the RFID working space dynamically. Readers deployed at the entrances and exits can identify the changes of $\Delta(t)$. We define a function $\mathcal{R}^t(o_i)$ specifying the logic region that an object belongs to:

$$\mathcal{R}^t(o_i) = \begin{cases} k & \text{iff } o_i \in \mathcal{Y}_k(t) \\ * & \text{iff } o_i \in \mathcal{O}(t) \end{cases} \quad (1)$$

We can use a discrete stream $\mathcal{S}(o_i)$ specifying the logic regions of an object locating at a series of time stamps. For example, $\mathcal{S}(o_1) = 1111***222***33$. Note that a $*$ in a discrete stream is an empty reading. It can either be a missing reading or a vacant reading. During the estimation process, $*$ is replaced with 0 if it is estimated as a vacant reading. Otherwise, $k \in R$ replaces with $*$ when it is estimated as a missing reading and o_i is estimated at the logic region \mathcal{Y}_k . A discrete stream $\mathcal{S}(o_i)$ is transformed to a fully estimated stream, noted as $\tilde{\mathcal{S}}(o_i)$, if all $*$ s in $\mathcal{S}(o_i)$ have been estimated. An example of fully estimated stream is $\tilde{\mathcal{S}}(o_1) = 1111002222000033$. We also define the actual stream $\bar{\mathcal{S}}(o_i)$ of an object as a sequence of the actual positions (k for \mathcal{Y}_k and 0 for $\tilde{\mathcal{Y}}$) of o_i in different time stamps. For example, $\bar{\mathcal{S}}(o_1) = 1111000222000333$. If we ignore those consecutive repetitive readings and those empty readings within a discrete data stream $\mathcal{S}(o_i)$, we get a logic stream $\hat{\mathcal{S}}(o_i)$. For example, $\hat{\mathcal{S}}(o_1) = 123$.

We can measure the error number $e(\tilde{\mathcal{S}}(o_i))$ of an estimated stream by counting the number of incorrect estimated entries in $\tilde{\mathcal{S}}(o_i)$, comparing with the actual stream $\bar{\mathcal{S}}(o_i)$. For example, $e(\tilde{\mathcal{S}}(o_1)) = 2$. Given a number of observed streams $\mathcal{S}(o_i)$ ($i \in O$), our problem is to achieve the following two level goals:

- Level1: Recover $\hat{\mathcal{S}}(o_i)$ from $\mathcal{S}(o_i)$, which is enough for some simple location sequence query in RFID applications.
- Level2: Estimate $\tilde{\mathcal{S}}(o_i)$ from $\mathcal{S}(o_i)$, so that $e(\tilde{\mathcal{S}}(o_i))$ can be as small as possible. $\tilde{\mathcal{S}}(o_i)$ is quite useful for the exact spatio-temporal queries of tagged objects.

Note that the pure temporal smoothing method or its variance cannot even be applicable for the first level goal. For example, for $\mathcal{S}(o_1) = 1111*****33$, we may get $\bar{\mathcal{S}}(o_1) = 13$ if those empty readings are simply ignored. For the level2 goal, because the staying periods of different objects in some regions may be quite different, the size of smoothing window cannot be efficiently defined. Furthermore, the dynamic smoothing window technique will not work due to the reader number limitation [6].

To utilize the group moving information for estimating the empty readings, we need define the correlation of moving tagged objects effectively. One intuition is that if two objects stay in the same region for a longer time, they will have higher probability to be the partners. We first assume that there are no missing readings, the correlation of two tagged objects o_i and o_j , denoted as $\lambda_1^t(o_i, o_j)$ can be defined as follows:

$$\lambda_1^t(o_i, o_j) = \begin{cases} \lambda_1^{t-1}(o_i, o_j) + 1 & \text{iff } \exists k, o_i, o_j \in \Upsilon_k(t) \\ \lambda_1^{t-1}(o_i, o_j) & \text{iff } o_i, o_j \in \tilde{\Upsilon}(t) \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

The above definition handle both the *chaos* and the *mutation* problems. The correlation of two objects increases if they stay in the same logic region longer. While, the correlation drops to zero if two objects are probably in different logic regions, which happens in *mutation* scenarios. However, the *ambiguity* is still a problem for such a definition because an empty reading in $\mathcal{S}(o_i)$ can also be a missing reading. If we can distinguish between $\tilde{\Upsilon}(t)$ and $\phi(t)$, the correlation can be modified as:

$$\lambda_2^t(o_i, o_j) = \begin{cases} \lambda_2^{t-1}(o_i, o_j) + 1 & \text{iff } \exists k, o_i, o_j \in \Upsilon_k(t) \\ \lambda_2^{t-1}(o_i, o_j) & \text{iff } o_i, o_j \in \tilde{\Upsilon}(t) \\ 0 & \text{iff } \bigvee (o_i \in \phi(t) \wedge o_j \notin \tilde{\Upsilon}(t)) \vee (o_j \in \phi(t) \wedge o_i \notin \tilde{\Upsilon}(t)) \\ & \text{otherwise.} \end{cases} \quad (3)$$

However, in real cases, what we can see is $\emptyset(t)$ instead of $\tilde{\Upsilon}(t)$. When $o_i \in \emptyset(t)$, it may be in the same or different places with o_j , keeping the correlation stable is the tradeoff choice according to the three challenges.

Thus ,we present our basic correlation definition as:

$$\lambda^t(o_i, o_j) = \begin{cases} \lambda^{t-1}(o_i, o_j) + 1 & \text{iff } \exists k, o_i, o_j \in \Upsilon_k(t) \\ \lambda^{t-1}(o_i, o_j) & \text{iff } o_i \in \emptyset(t) \vee o_j \in \emptyset(t) \\ 0 & \text{otherwise.} \end{cases} \quad (4)$$

Based on the correlations among objects, given a tagged object o_i , the reference partner of o_i can be defined as $\mathcal{P}^t(o_i)$, such that $\nexists o_j, \lambda^t(o_i, o_j) > \lambda^t(o_i, \mathcal{P}^t(o_i))$. Note that there may be multiple objects maximizing the correlation to an object o_i . We randomly choose one of them as $\mathcal{P}^t(o_i)$.

In order to incrementally maintain $\lambda^t(o_i, o_j)$ from $\lambda^{t-1}(o_i, o_j)$, a complete graph is required, namely $G(t) = (V, E)$, in which the vertex set is $V = \Delta(t)$

and the edge set is $E = \{(o_i, o_j, \kappa) | \forall o_i, o_j \in \Delta(t) \wedge i < j, \kappa = \lambda^t(o_i, o_j)\}$. The vertex is fully linked with weights and it cannot be divided into sub-graphs. A matrix \mathcal{M}^t , $\mathcal{M}^t[i][j] = \lambda^t(o_i, o_j)$ can be employed for dynamic maintenance of such a graph. The algorithm is easy to be implemented, but the time and space complexity are both $O(N^2)$, where $N = |\Delta(t)|$. The cost is quite high in RFID applications when the number of monitored objects is huge. Using the adjacency list to maintain the graph will not reduce the time complexity either. Therefore, we need to re-consider the $\lambda^t(o_i, o_j)$ definition and therefore compress the graph structure based on the the concept of reference partner.

Obviously, reference partner can be utilized in the correlation definition when an empty reading occurs. The optimized correlation definition is given in equation(5). Note that λ will be replaced with $\bar{\lambda}$ in computing $\mathcal{P}^t(o_i)$ in this case. Furthermore, the optimized correlation definition can lead to improved maintenance model which will be illustrated in the next section.

$$\bar{\lambda}^t(o_i, o_j) = \begin{cases} \bar{\lambda}^{t-1}(o_i, o_j) + 1 \text{ iff } (\exists k, o_i, o_j \in \mathcal{Y}_k(t)) \\ \quad \vee (o_i \in \mathcal{O}(t) \wedge o_j \notin \mathcal{O}(t) \wedge \mathcal{R}^t(\mathcal{P}^{t-1}(o_i)) = \mathcal{R}^t(o_j)) \\ \quad \vee (o_j \in \mathcal{O}(t) \wedge o_i \notin \mathcal{O}(t) \wedge \mathcal{R}^t(\mathcal{P}^{t-1}(o_j)) = \mathcal{R}^t(o_i)) \\ \quad \vee (o_i \in \mathcal{O}(t) \wedge o_j \in \mathcal{O}(t) \wedge \mathcal{R}^t(\mathcal{P}^{t-1}(o_i)) = \mathcal{R}^t(\mathcal{P}^{t-1}(o_j))) \\ 0 \quad \quad \quad \text{otherwise.} \end{cases} \quad (5)$$

4 Optimized Maintenance of Correlation

4.1 Properties of Optimized Correlation

The optimized correlation has some desired properties which can be utilized for reducing the maintenance cost. They are: (1) $\bar{\lambda}^t(o_i, o_i) \geq \bar{\lambda}^t(o_i, o_j)$; (2) $\bar{\lambda}^t(o_i, o_j) = \bar{\lambda}^t(o_j, o_i)$; (3) $o_i \in \mathcal{Y}_k(t) \wedge o_j \in \mathcal{Y}_l(t) \wedge k \neq l \Rightarrow \bar{\lambda}^t(o_i, o_j) = 0$. Furthermore, an advanced property for $\bar{\lambda}^t$ can be inferred as follows:

Theorem 1. $\forall i, j, k$, Suppose $\bar{\lambda}^t(o_i, o_k) = \max\{\bar{\lambda}^t(o_i, o_k), \bar{\lambda}^t(o_i, o_j), \bar{\lambda}^t(o_j, o_k)\} \Rightarrow \bar{\lambda}^t(o_i, o_j) = \bar{\lambda}^t(o_j, o_k)$

Proof. Suppose $\bar{\lambda}^t(o_i, o_k) = \kappa_1 > \bar{\lambda}^t(o_j, o_k) = \kappa_2$, according to equation (5), $\bar{\lambda}^{t-\kappa_2}(o_i, o_k) = \kappa_2 - \kappa_1$ and $\bar{\lambda}^{t-\kappa_2}(o_j, o_k) = 0$. At the time point $t - \kappa_2$, if $o_k, o_i, o_j \notin \mathcal{O}(t)$, $\mathcal{A}^{t-\kappa_2}(o_i) = \mathcal{A}^{t-\kappa_2}(o_k) \neq \mathcal{A}^{t-\kappa_2}(o_j)$. Hence, $\bar{\lambda}^{t-\kappa_2}(o_i, o_j) = 0$. If $o_k \in \mathcal{O}(t)$, $o_i, o_j \notin \mathcal{O}(t)$, $\mathcal{A}^{t-\kappa_2}(o_i) = \mathcal{A}^{t-\kappa_2}(\mathcal{P}^{t-\kappa_2-1}(o_k)) \neq \mathcal{A}^{t-\kappa_2}(o_j)$. By analogy of other situations, we can prove $\bar{\lambda}^{t-\kappa_2}(o_i, o_j) = 0$. Also, $\forall t - \kappa_2 < \tau \leq t$, we can infer $\mathcal{A}^\tau(o_i)$ (or $\mathcal{A}^\tau(\mathcal{P}^{\tau-1}(o_i))$) = $\mathcal{A}^\tau(o_k)$ (or $\mathcal{A}^\tau(\mathcal{P}^{\tau-1}(o_k))$) = $\mathcal{A}^\tau(o_j)$ (or $\mathcal{A}^\tau(\mathcal{P}^{\tau-1}(o_j))$). So $\bar{\lambda}^\tau(o_i, o_j) = \bar{\lambda}^{\tau-1}(o_i, o_j) + 1$ and $\bar{\lambda}^t(o_i, o_j) = \kappa_2 = \bar{\lambda}^t(o_j, o_k)$. Also, the situations for $\bar{\lambda}^t(o_i, o_k) = \bar{\lambda}^t(o_k, o_j)$ and $\bar{\lambda}^t(o_i, o_k) > \bar{\lambda}^t(o_k, o_j)$ can be inferred in the similar way.

Based on the properties of $\bar{\lambda}$, we can get a sub-graph $G_k(t) = (V, E)$ for each $\mathcal{Y}_k(t)$, where the vertex set is $V = \mathcal{Y}_k(t)$ and the edge set is $E = \{(o_i, o_j, \kappa) | \forall o_i, o_j \in \mathcal{Y}_k(t) \wedge i < j, \kappa = \bar{\lambda}^t(o_i, o_j)\}$. Also, $\mathcal{O}(t)$ will correspond to a special sub-graph

to link them directly and assign the $\bar{\lambda}$ value at the linking position as 1 according to our $\bar{\lambda}$ definition.

- **Splitting principle.** Suppose e_k, e_m and e_n are three consecutive elements in a $\bar{\lambda}$ -List and $\bar{\lambda}^t(o_k, o_m) = \bar{\lambda}^t(o_m, o_n) = \kappa$. According to Theorem 1, $\bar{\lambda}^t(o_k, o_n) \geq \kappa$. However, if $\bar{\lambda}^t(o_k, o_n) > \kappa$, e_m can be inferred to be inserted into a $\bar{\lambda}$ -List having contained e_k and e_n , at $t - \kappa + 1$. Whereas according to the regrouping principle, e_m will not lie between e_k and e_n . Therefore, we can testify $\bar{\lambda}^t(o_k, o_n) = \kappa$. Furthermore, combining it with Theorem 1, for any two objects o_i, o_j in $\bar{\lambda}$ -List, $\bar{\lambda}^t(o_i, o_j)$ can be obtained by calculating the minimal $\bar{\lambda}$ between o_i and o_j . Thus, a list can be split into some sub-lists when some $\bar{\lambda}$ value drops to 0. Note that $\mathcal{P}^t(o_i)$ will be the o_i 's neighbor in a $\bar{\lambda}$ -list according to the analysis above.

4.2 Optimized Maintenance Strategies

In this section, we will propose three incremental $\bar{\lambda}$ maintenance strategies based on different splitting and regrouping principles. The proposed maintenance strategies show different performances in various conditions. Suppose $N = |\Delta(t)|$, N_r is the number of logic regions plus a dead region and $N_o = N/N_r$ represents the average number of objects in each region. Besides, there is a hidden metric P_c reflecting the average probability that a object will change its partners at the next time stamp.

Multiple Scans (MS) Strategy In order to compute L^t from L^{t-1} , we can scan each L_i^{t-1} for multiple times. At each time, we abstract the objects in the same L_j^t . We illustrate the procedure in Figure 3. Suppose the input is $\mathcal{Y}_1(t) = \{o_8, o_9, o_{10}, \dots\}$, $\mathcal{Y}_2(t) = \{o_{14}, o_{22}, \dots\}$ and $\mathcal{O}(t) = \{o_5, o_{20}, o_{26}, \dots\}$. 1-1 represents the first step in the first loop. $\min\lambda_i$ representing the minimum λ_i in L_i^{t-1} for each loop will be maintained in an incremental manner. For the objects in $\mathcal{O}(t)$, reference partner is required to determine which list to insert. For example, $\mathcal{P}^{t-1}(o_5) = o_{20} \wedge \mathcal{R}^t(o_{20}) = *$, therefore, o_5 will be inserted into the special list L_*^t for the dead region. While $\mathcal{P}^{t-1}(o_{26}) = o_{22} \wedge \mathcal{R}^t(o_{22}) = 2$, so o_{26} will be inserted into the region list L_2^t .

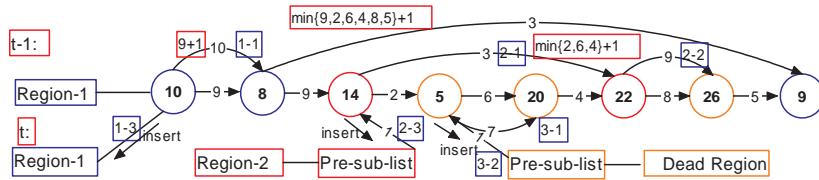


Fig. 3. An illustration of the MS procedure

The key steps of the MS algorithm is illustrated in Algorithm 1. The processing cost is related to P_c . If $P_c = 0$, the average complexity to maintain a

L_i^{t-1} is just $O(N_o)$. If $P_c = 1$, the average complexity to maintain a L_i^{t-1} will be $O(N_o^2)$. Therefore, the total time complexity to transform L^{t-1} to L^t will be between $O(N_o N_r)$ and $O(N_o N)$.

Algorithm 1 The Multiple Scans Algorithm

Input: L^{t-1}

Output: L^t

1. **for** $L_i^{t-1} \in L^{t-1}$ **do**
 2. **while** $L_i^{t-1} \neq \emptyset$ **do**
 3. reset $min\lambda_i$
 4. $\alpha = L_i^{t-1}.head$
 5. **if** $\alpha \in \mathcal{O}(t)$ **then**
 6. $\alpha = \mathcal{P}^{t-1}(\alpha)$
 7. **for** $\beta = o_i \in L_i^{t-1}$ **do**
 8. maintain $min\lambda_i$
 9. **if** $\beta \in \mathcal{O}(t)$ **then**
 10. $\beta = \mathcal{P}^{t-1}(\beta)$
 11. **if** $\mathcal{R}^t(\alpha) = \mathcal{R}^t(\beta)$ **then**
 12. $\beta.\lambda = min\lambda_i + 1$
 13. $\hat{L}_{\mathcal{R}^t(\alpha)}^{t-1}.add(\beta)$
 14. $L_i^{t-1}.remove(\beta)$
 15. $L_{\mathcal{R}^t(\alpha)}^t.link(\hat{L}_{\mathcal{R}^t(\alpha)}^{t-1})$
-

Single Scan (SS) Strategy Compared to MS, SS scans each sub-list L_i^{t-1} for only once. It inserts each o_i in the sub-list into the corresponding L_j^t . However, in order to maintain the $\bar{\lambda}$, for each L_j^t , the current $min\lambda_j$ must be maintained. All the L_j^t must be scanned to maintain the $min\lambda_j$ with each insertion. We illustrate the basic procedure in Figure 4 with the same input of Figure 3.

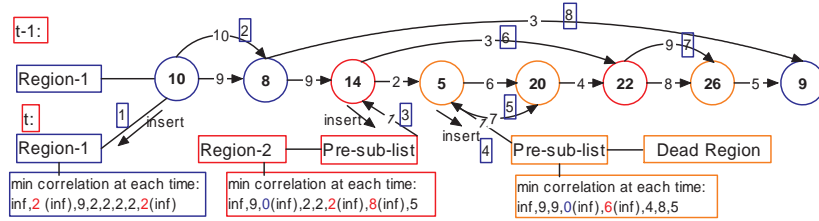


Fig. 4. An illustration of the SS procedure

The key steps of the SS algorithm is illustrated in Algorithm 2. The average complexity to maintain a list is $O(N_o N_r)$. The total complexity to transform from L^{t-1} to L^t will be $O(N_r N)$.

Algorithm 2 The Single Scan Algorithm

Input: L^{t-1} **Output:** L^t

1. **for** $L_i^{t-1} \in L^{t-1}$ **do**
 2. **for** $\beta = o_i \in L_i^{t-1}$ **do**
 3. **if** $\beta \in \mathcal{O}(t)$ **then**
 4. $\beta = \mathcal{P}^{t-1}(\beta)$
 5. **for each** L_i^t **do**
 6. **if** $\min\lambda_i > \beta.\lambda$ **then**
 7. $\min\lambda_i = \beta.\lambda$
 8. **if** β is the first in L_i^{t-1} to insert $L_{\mathcal{R}^t(\beta)}^t$ **then**
 9. $\min\lambda_{\mathcal{R}^t(\beta)} = 0$
 10. $\beta.\lambda = \min\lambda_{\mathcal{R}^t(\beta)} + 1$
 11. $\min\lambda_{\mathcal{R}^t(\beta)} = \infty$
 12. $L_{\mathcal{R}^t(\beta)}^t.add(\beta)$
 13. $L_i^{t-1}.remove(\beta)$
-

Interval Tree-based Scan (ITS) Strategy When computing the $\bar{\lambda}$ of any two objects, we can construct an interval tree to calculate the $\bar{\lambda}$ directly, instead of incrementally maintaining $\min\lambda_i$. First, all $\bar{\lambda}$ values will be indexed as an ordered array. The leaf nodes of the tree are all the objects, and the internal nodes are the index of the children nodes with smaller $\bar{\lambda}$. The bounds of the index interval for each parent node is recorded. The complexity to construct such a tree for a sub-list will be $O(N_o \log N_o)$. However, the extra space cost for the inter nodes is required. To calculate $\bar{\lambda}^t(o_i, o_j)$, from the root node, dichotomy is used until finding the corresponding left bound node and the right bound node. Then, the two correlation scores will be compared. The smaller one is the result. Note that in the list structure, $\bar{\lambda}^{t-1}(o_i, o_j)$ is stored in the node representing o_j . Therefore, the corresponding left bound index should be incremented by 1. For example, in Figure 5 which illustrates the procedure of ITS, to compute $\bar{\lambda}^{t-1}(o_8, o_9)$, namely $\bar{\lambda}^{t-1}(o_{index=2}, o_{index=8})$, the left index bound should be 3 and the right index bound should be 8. The scan of the tree needs $O(\log N_o)$ time complexity. In this way, total time complexity to transform L^{t-1} to L^t will be $O(N \log N_o)$.

MS, SS and ITS are suitable for different scenarios. MS is suitable for the situation when P_c is small and SS is suitable for the situation when N_r is small. ITS can be efficient when P_c and N_r are both large. Because no algorithm can be dominant over others in all cases, we can make the choice adaptively according to the applications.

5 Remedy-based RFID Data Imputation Strategy

Defining and maintaining the correlations are only the basis of the data imputation strategy. One way of data imputation is to construct the missing readings

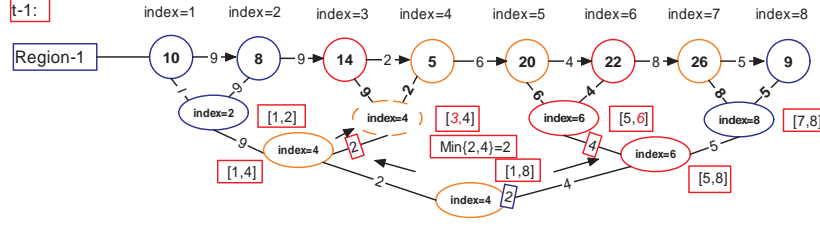


Fig. 5. An illustration of correlation computation based on interval tree

directly based on \mathcal{P}^{t-1} . We call it the direct imputation. However, the direct imputation may cause regular errors due to the change of partners. For example, for $\mathcal{S}(o_1)=111***222$ and $\mathcal{S}(o_2)=11*****33$, if $\mathcal{P}^3(o_2) = o_1$ and $\mathcal{P}^7(o_2) = o_1$, the direct imputation may result in $\tilde{\mathcal{S}}(o_2) = 111000233$. But obviously, the real case is much more likely $\tilde{\mathcal{S}}(o_2) = 110000033$, interpreted as o_1 leaves o_2 and leads to a different region. Therefore, we introduce a remedy-based RFID data imputation strategy. Instead of implementing imputation right away, the remedy-based RFID data imputation will check the states of two objects in the next logic destination region. Although the imputation is delayed, it is very useful to improve the accuracy of the results. Especially, the remedy-based RFID data imputation is quite useful for those objects having no substantial partners with them. We illustrate the strategy in Algorithm 3.

Algorithm 3 Remedy-based RFID Data Imputation

1. **if** $o_i \in \mathcal{O}(t) \wedge \mathcal{P}^{t-1}(o_i) \notin \mathcal{O}(t)$ **then**
 2. $o_i \rightarrow \mathcal{P}List.add(\mathcal{P}^{t-1}(o_i))$
 3. **if** $o_i \notin \mathcal{O}(t) \wedge \mathcal{P}^t(o_i) \notin \mathcal{O}(t)$ **then**
 4. **if** $o_i \rightarrow \mathcal{P}List \neq \emptyset$ **then**
 5. $\mathcal{S}_\alpha = o_i \rightarrow \mathcal{P}List.searchbyID(\mathcal{P}^t(o_i))$
 6. **if** $\mathcal{S}_\alpha \neq \emptyset$ **then**
 7. **for** $\alpha \in \mathcal{S}_\alpha$ **do**
 8. $o_i.interpolate(\alpha.t, \alpha.r)$
 9. $o_i \rightarrow \mathcal{P}List.removeALL()$
-

The algorithm can be combined with the correlation maintenance strategy directly by introducing an adjunctive data structure $\mathcal{P}List$, which is used to reserve the possible reference partners. We denote the size of $\mathcal{P}List$ as $N_{\mathcal{P}}$ and the additional search cost $O(N_{\mathcal{P}})$ will be incurred when the imputation is triggered. However, it is trivial compared to the maintenance of correlations. The response delay of imputation is potentially determined by the time spent during the dead region for the corresponding objects.

As a pre-processing method, data imputation will serve the user's query. However, remedy-based imputation may lead to inaccurate results for the upper

query because of the delay. RFID-oriented query can be based on database [11] or data stream [12]. For database query [11], the RFID readings will be inserted into a table, e.g., R-Table, and SQL query can be executed over the table. In this way, as long as the query is not incurred before the imputation judgement is finished, there will be no problems. Correspondingly, $o_i.interpolate(\alpha.t, \alpha.r)$ in the database context will be interpreted the following SQL:

UPDATE R-Table Set Region= $\alpha.r$ WHERE Object= $o_i.id$ AND time= $\alpha.t$.

But for the RFID data stream processing, automata model is usually employed to detect complex event pattern [12]. The data will not be stored into the database and query is executed in the continuous manner. For the common queries involving sequence, simultaneous, conjunction and disjunction, although the response will be delayed but the correct results can be guaranteed. However, for the query when negation operation is involved, there may be false positive result produced. For example, for the query $SEQ(A, NEG(B))_w$, which represents some object doesn't come to region B after leaving region A within w period, if the imputation cannot be finished within w , a false composite event will be notified. Fortunately, we can employ a similar version of algorithm 3 as the alternative method to solve the problem. The main idea is to first utilize direct imputation method. Then, delete the imputation data later once it is judged to be unreasonable. It can be obtained by simply modifying algorithm 3. Note that this alternative method is not suitable for the occasions when algorithm 3 is applicable. Therefore, we need to choose the proper remedy-based RFID data imputation strategy according to the query categories.

6 Experiments

6.1 Experiment Settings

In this section, we report the evaluation of our proposed model by simulated experiments, in terms of the efficiency and accuracy. All experiments were conducted on a PC of 2.6G Hz with 1G memory. The algorithms were implemented with C++. We designed three kinds of simulated data sets in a museum scenario.

- **DataSet1.** The locations of the monitored objects (visitors) are totally random at any time. In this case, objects hardly have stable partners, and P_c will be very high.
- **DataSet2.** Most monitored objects will have partners at each time stamp. The partners may change casually.
- **DataSet3.** All the monitored objects will have their partners all the time. The partners seldom change during the whole simulation period.

6.2 Maintenance Cost

We evaluate the processing time of three correlation maintenance strategies in this section. In order to better illustrate the flexibility in complex situation, the results of tests on dataset1 are given in Figure 6. In this case, P_c can be inferred

by computing $\min(N_o, N_r)$, which helps us to observe its impacts on computational cost. The granularity of simulation time represented by T is second and the total cost during the simulation is used to illustrate the efficiency.

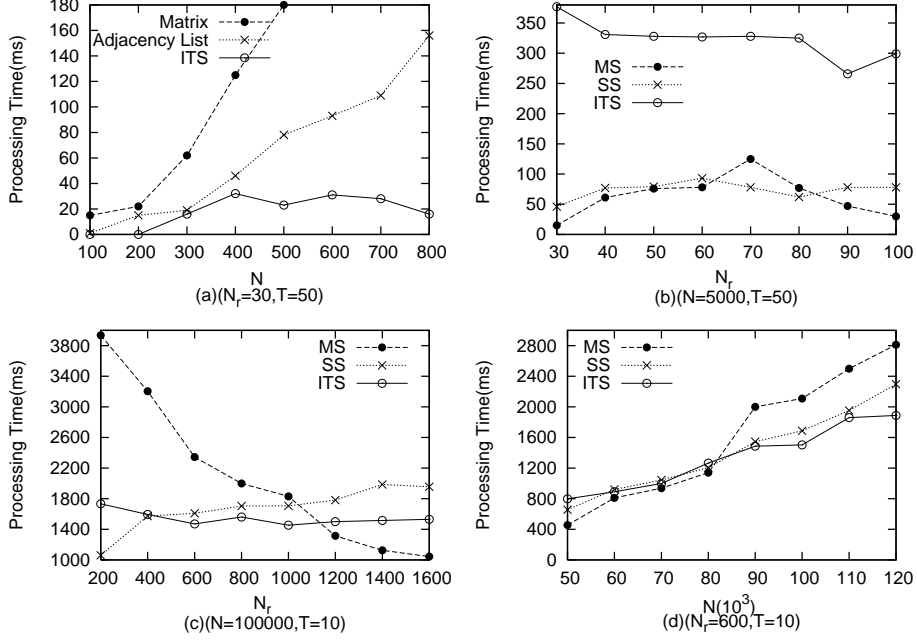


Fig. 6. Processing time comparison of different correlation algorithms

In Figure 6(a), we tested the efficiency of different correlation maintenance models. We can find that the list-based model based on the correlation λ will be much more efficient than the graph-based model based on the correlation λ . The matrix-based graph maintenance is very time-consuming. The advantage of ITS over adjacency-list-based method will become quite prominent with the increase of N . In Figure 6(b), we compared the different optimization maintenance methods when $N = 5000$. ITS will consume some computational costs in constructing the tree and index, which is comparable to the maintenance of correlations in this scale of N . Therefore, compared to MS and SS, the cost of ITS will be higher. The contrast of MS and SS is not very obvious in this situation.

In Figure 6(c) and Figure 6(d), we studied the impacts of N and N_r on the computational cost of various correlation maintenance methods when a huge number of objects are involved. As we can see from these two figures, neither MS, SS nor ITS can dominate the others under different values of N and N_r , which is substantially accordant with our theory analysis. For dataset2 and dataset3, similar results can be found for the computational time of different correlation maintenance methods.

6.3 Accuracy

Because the direct imputation and remedy-based imputation methods will incur very low computational cost compared to the maintenance of correlations, we simply studied the accuracy of different correlation definitions and data imputation strategies. The compared methods include direct imputation and remedy-based imputation based on λ and $\bar{\lambda}$ (opt- λ in Figure 7). The error rates for our level1 goal and level2 goal of different imputation methods are illustrated in Figure 7. For the level1 goal, we compared with the temporal smoothing method, which is the ideal method in this case if imputation is simply based on the temporal information. We take Dataset2 and Dataset3 as the testing datasets to illustrate the accuracy evaluation because partnerships in Dataset1 can hardly be preserved.

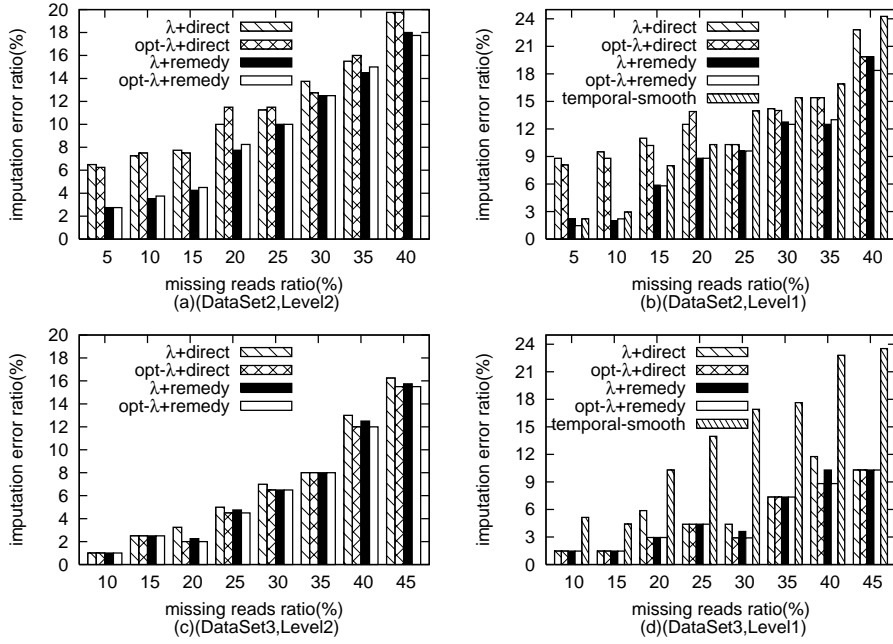


Fig. 7. Error ratio comparison of different data imputation strategies

From Figure 7(a)-(d), we can see that the accuracy under the $\bar{\lambda}$ definition and the λ definition is quite close. While, the remedy-based method will improve the accuracy in Dataset2 where more mutations exist. Even for the level1 goal, compared to the temporal-smoothing method, the correlation-based imputation method will be more accurate, especially when the missing ratio is large. Compared to dataset2, the accuracy of our proposed imputation model is better in dataset3 where partners seldom change. Moreover, our methods can also gain

relatively higher accuracy in the very noisy environment(e.g., with 40% missing ratio).

7 Conclusion

In this paper, we have proposed a novel RFID data imputation mechanism based on analyzing the correlations of monitored objects. Basic correlation model is first inferred to solve the three key challenges: mutation, chaos and ambiguity. The $\bar{\lambda}$ correlation is proposed to gain important correlation relationship between objects. Using this relationship, related optimization strategies about correlation maintenance are discussed. A remedy-based data imputation strategy is introduced to improve the accuracy. Finally, we have identified the effectiveness of the proposed models and methods through extensive experimental studies.

Acknowledgement

This research was partially supported by the National Natural Science Foundation of China under Grant No.60773220 and 60873009.

References

1. Want, R.: An introduction to RFID technology. *IEEE Pervasive Computing* (2006) Vol.5(1) 25-33
2. Want, R.: The Magic of RFID. *ACM Queue* (2004) Vol.2(7) 40-48
3. Asif, Z., Mandviwalla, M.: Integrating the supply chain with RFID: A Technical and Business Analysis. *Communications of the Association for Information Systems* (2005) Vol.15 393-427
4. Philipose, M., Fishkin, K.P., Perkowitz, M., Patterson, D.J., Fox, D., Kautz, H., Hahnel, D.: Inferring activities from interactions with objects. *IEEE Pervasive Computing*(2004) Vol.3(4) 10-17
5. Jeffery, S.R., Alonso, G., Franklin, M.J.: A pipelined framework for online cleaning of sensor data streams. In *Proceedings of ICDE (2006)* 140-142
6. Jeffery, S.R., Garofalakis, M., Franklin, M.J.: Adaptive cleaning for RFID data streams. In *Proceedings of VLDB (2006)* 163-174
7. Gonzalez, H., Han, J., Shen, X.: Cost-conscious cleaning of massive RFID data sets. In *Proceedings of ICDE(2007)* 1628-1272
8. Khoussainova, N., Balazinska, M., Suci, D.: Towards correcting input data errors probabilistically using integrity constraints. In *Proceedings of MobiDE (2006)* 43-50
9. Rao, j., Doraiswamy, S., Thakkar, H., Colby, L.S.: A deferred cleansing method for RFID data analytics. In *Proceedings of VLDB(2006)* 175-186
10. Vuran, M.C., Akyildiz, I.F.: Spatial correlation-based collaborative medium access control in wireless sensor networks. *IEEE/ACM Transactions on Networking (TON)*(2006) Vol.14(2) 316-329
11. Wang, F.S., Liu, P.Y.: Temporal management of RFID data. In *Proceedings of VLDB(2005)* 1128-1139
12. Wang, F.S., Liu, S., Liu, P.Y.: Bridge physical and virtual worlds: complex event processing for RFID data streams. In *Proceedings of EDBT(2006)* 588-607