Triggered Moving Range Queries over RFID Monitored Objects

Yu Gu, Ge Yu, Na Guo

Northeastern University, Shenyang, 110004, China
E-mail: {guyu,yuge,guona}@ise.neu.edu.cn

Received submit date

Abstract As a promising technology for monitoring and tracing the product flows and human activities, Radio Frequency Identification (RFID) has received much attention within database community. Moving range query over RFID data streams is one of the most important spatio-temporal queries to support valuable information analysis. However, the uncertainty of the monitored object location challenges the query strategy. Novel models and methods are desired. In this paper, we propose a probability evaluation model in the RFID-enabled monitoring environments and design various query processing and optimization techniques under the scenarios of triggered moving range queries. The extensive experimental evaluation verifies the efficiency and effectiveness of our proposed model and methods.

Keywords RFID, probability, location uncertainty, moving range query

1 Introduction

The Radio Frequency Identification (RFID) [1] is experiencing a fast development in recent years. Today, more and more moving objects or products in our everyday lives have been attached with RFID tags, which provide fast accessibility of the objects’ spatio-temporal information. Compared to the traditional identification method such as bar code, RFID technique is fast, automatic and robust. On the other hand, RFID systems are relatively cheap compared to GPS equipments. An RFID application is mainly composed of readers and tags. Readers are transponders capable of detecting tags within a distance. Tags are attached to the monitored objects, which can either actively transmit or passively respond a unique identification code when close to a reader. Based on the 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 [2], human activity tracking [3], 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. Efficient range, k-NN or trajectory query over the data streams are quite useful for supporting advanced analysis. Figure 1 illustrates an RFID-enabled smart museum scenario where art exhibits are displayed in the fixed exhibition area located in different exhibition rooms. Exhibition rooms are linked by passageways. The exhibits are monitored by the RFID readers, and tagged visitors (monitored objects) walk from one exhibit to another. The data stream reflecting the id, location and the sensing time of monitored objects will be continuously produced only when the objects are in the reader sensing regions. However, compared to the exact positioning techniques, there are two important challenges for efficient spatio-temporal queries due to location uncertainty.

- **Imprecision.** When an object is in the sensing region, we can only acquire which reader senses the object instead of the exact position in this region.

- **Discontinuity.** Due to consideration of cost and application demands, readers are not
possible or necessary to be deployed to monitor each inch of the space. During the time when walking from one exhibit to another, the location information of monitored object is vacant.

In this paper, we focus on an important query for the moving object management: spatial range query. Traditional range query over moving objects can be classified into different categories: (1) Based on historical data, current data or predicative data. (2) Based on time point or time period. (3) Snap-shot pull query or continuous push query. (4) Static query with specified query range or moving query. Specifically, a moving range query [4] has an associated moving object namely the focal object of the query. The spatial region of the query moves continuously as the query’s focal object moves. For example, querying the visitors with distance smaller than 100 meters to a stolen exhibit in the last 30 seconds, is a snap-shot static query based on time period and historical data. Also, the query for a VIP visitor to continuously monitor visitors around him within 100 meters in the real-time fashion is a continuous focus-moving query based on time point and current data. Besides, continuous query is usually periodically time-driven. However, in RFID applications, in order to avoid the impact of discontinuity and improve the efficiency and correctness, the query is desired to be triggered once the object enters a new reader sensing region. Different from the time driven manner, this continuous query can be considered to be event driven. We call this kind of RFID-enabled moving query as triggered moving range queries. As another example illustrated in Figure 2, for an international horticultural exposition, RFID can be widely deployed at major places of interest and service facilities. Paths exist between different visiting areas which contain adjacent scenic spots. The tourists can be offered tags for intelligent services. By real-time range query, a tourist can find nearby scenic spots, toilets and restaurants. Also, the group tourists can keep friends and families moving not too far from each other. Furthermore, the tour guide can monitor the tourists to avoid falling behind. We can deploy such RFID equipments in the hospital, railway station, office building and campus to implement interesting monitoring, tracking and inferring. In addition, range queries can be the basis for high-level analysis and decision-making. Obviously, compared to GPS positioning method, this kind of RFID-based deployment is more convenient and economical. However, the new query framework is desired to be designed considering traditional GPS-based query methods are not suitable. Especially, due to the location uncertainty of the tagged objects, we can only conduct probabilistic query to figure out those possible target objects for RFID-based range query. Therefore, relative probability inference and query optimization techniques are needed. Furthermore, the model and strategy designed for triggered moving range queries are helpful to for designing other types of RFID range queries variants, k-NN queries, trajectory analysis and aggregation statistics.

Fig. 2. An international horticultural exposition scenario

To our knowledge, this is the first work aiming to model the probabilistic moving range query
over RFID spatio-temporal data streams. Algorithms on executing and optimizing triggered moving range query are proposed. The paper is organized as follows. Section 2 outlines the related work. Section 3 introduces RFID spatio-temporal data modeling. Section 4 analyzes the probability modeling for RFID range query. Section 5 illustrates the basic and two optimized RFID moving range query strategies. Section 6 gives the experimental analysis and Section 7 concludes the paper.

2 Related Work

The proliferation of RFID raises new problems for database community. The state-of-art works mainly focus on data cleaning techniques [5, 6] considering original reading inaccuracy and complex event processing model over RFID event streams [7]. For uncertain RFID data management, Cascadia [8] and Laha [9] discuss how to infer the advanced semantics-rich complex events from the probabilistic low-level RFID physical reading information. The spatio-temporal query based on location uncertainty is neglected. Tran et al. [10] discuss the probabilistic positioning and data cleaning problems in a special warehouse monitoring scenario where readers are mobile and tagged commodities are relatively static. The model and methods are not suitable for our range query background where readers are static and tags are dynamic.

For the general moving object management, range query is widely studied. But most works focus on historical data based on database [11] instead of data streams. For the static range query over moving objects, R-tree-based variance structures [12] are widely used. Specifically, some specially designed index structures are proposed to speed up probabilistic range query. Cheng et al. [13] propose PTI (Probability Threshold Indexing) based on R-tree with additional data distribution information, which can improve the efficiency of the probability threshold query. Based on PTI, efficient evaluation of imprecise location-dependent queries are proposed by Chen et al. [14]. U-tree index [15] is constructed for multi-dimensional data following some probabilistic distribution. Based on the concept of CFB (Conservative Functional Box), U-tree can quickly filter objects instead of processing each object in the database. Also, UTR-tree [16] index is proposed to track the uncertain trajectory of network-constrained moving objects using a mixed R-tree structure. Furthermore, an extended TPR-tree namely TR(s,d)-tree is proposed by Huang et al. [17] to index moving objects with uncertain speed and direction. Different from those potentially R-tree and MBR based structure, U-grid index is proposed by Kalashnikov et al. [18] to efficiently cope with probabilistic range query based on uncertain histogram and quadtree.

The works above only focus on the static range query. Gedik et al. [4] present the processing techniques of moving range query over moving objects. Specifically, based on the current position and velocity, motion sensitive bounding boxes for moving objects as well as the query point are designed. Efficient index is built based on the motion sensitive bound box instead of the exact position of the moving object. However, different from the inherent location uncertainty applications, the position can be exactly obtained at any time in this scenario. For RFID applications, due to the specificity of location uncertainty, the RFID monitored objects are inferred to be dispersed in some pre-specific regions or discretely located at some possible points in different routes, so available index structure is not quite efficient especially for our triggered moving range query.

3 RFID Spatio-temporal Data Modeling

An RFID reader $r$ periodically senses readings from a tagged object $o$ if $o$ is in the sensing region of $r$. The reader’s detecting cycle is $t_c$ which is synchronous among different readers. Suppose $R, O$ represent the sets of reader IDs and object IDs respectively. An RFID reading is modeled as a ternary tuple $p_i = < i \in O, k \in R, t >$ with the schema $< f_o, f_r, f_t >$, representing an object $o_i$ is detected by a reader $r_k$ at time stamp $t$. RFID readers are assumed to be independent, i.e., no sensing regions of two readers overlap with each other. Next, we will introduce the concept of the aware region.
Definition 1. (Aware Region) The sensing region of a reader \( r_k \) is called an aware region \( \Upsilon_k \), which can be approximated as a circle in two-dimension. \( (\Upsilon_k.c_x, \Upsilon_k.c_y) \) represents the circle center coordinate of \( \Upsilon_k \) and \( \Upsilon_k.c_r \) represents the radius of \( \Upsilon_k \).

Due to the consideration of cost and reader cover shape, readers are only arranged in the specific aware region to execute specified task instead of monitoring each inch of the working space. The space that is not covered by any aware region is called the vacant region, denoted as \( \bar{\Upsilon} \). We suppose the data have been cleaned if there are any missing readings. Therefore, those tagged objects in any aware region are supposed to be detected by a reader. While, those tagged objects will not be sensed by any readers if they are in \( \bar{\Upsilon} \).

The detected object set within \( \Upsilon_k \) at time stamp \( t \) is denoted as \( \Upsilon_k(t) = \{ o_i | \exists p = < i, k, t > \} \), and the tagged object set in the vacant region at time stamp \( t \) is represented as \( \bar{\Upsilon}(t) \). The RFID working space at time \( t \) denoted as \( \Delta(t) \) 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 aware region that an object belongs to, i.e., \( \mathcal{R}^t(o_i) = k \), iff \( o_i \in \Upsilon_k(t) \).

Based on the RFID-enabled smart museum scenario illustrated in Figure 1, we will further analyze the features of RFID spatio-temporal monitoring. Aware regions often correspond to some areas where specific activities or procedures take place while the vacant region offers the space for monitored objects to travel from one site to another. For example, the art exhibition area numbered 1-9 implicates RFID aware region 1-9. Specially, the sensing radius \( c_r \) of different readers are different. But because the positions of RFID readers are static, when the positions and types of readers are set in a specific scenario, the shapes of aware regions can be modeled. In the vacant region, we will further define the concept of physical path and virtual path. Physical path is constrained by the specific physical condition to form a natural path, such as passage, link and lane. The shape of the physical path will determine the model of the route directly. For example, the two links illustrated in Figure 1 can be separately modeled as straight line and poly line. Furthermore, the physical path can be one-way (the link above in Figure 1) or bidirectional (the link below in Figure 1), which should be reflected in the model. Between readers without physical path connected, we can construct an ideal virtual path according to the features of applications and sites. For the smart museum applications, the visitors aim to enjoy the amazing exhibits successively, so the most reasonable virtual path model should be the shortest route between two readers, namely the straight line connecting the centers of two aware regions for R5 and R6 in Figure 1. However, for R1 and R4, due to the space limitation of the room door, the virtual path has to be modeled as a broken line. Putting physical and virtual path together, we define the potential path as follows.

Definition 2 (Potential Path). The potential path between \( \Upsilon_{k_1} \) and \( \Upsilon_{k_2} \) is denoted as \( \rho(\Upsilon_{k_1}, \Upsilon_{k_2}) \), representing the physical path or virtual path between \( \Upsilon_{k_1} \) and \( \Upsilon_{k_2} \). The potential path can be modeled as a directed straight line or a broken line.

If a monitored object can move between two aware regions along some potential path without passing through the third aware region, we say these two aware regions have the connectivity (or they are connected). Figure 3 illustrates the spatial information model based on Figure 1 and implicates the connectivity of different aware regions. Circle represents the aware region with the information of \( c_r, c_x \) and \( c_y \). The directed line between nodes represents the connectivity of aware regions. Due to the limitation of the door and corner, the potential path may be a broken line and thus inflection points are defined to represent the fold points which split the line. The dotted line marks some of such occasions. Note that the inflection point information should be recorded for path object position computation. Besides, information about rooms (or a block of visiting places) can be contained in this concept model, which is useful for the route probability evaluation. The spatial model can be represented by a logic graph \( G = (V, E) \), where \( V \) denotes the aware region and
E denotes the potential path. In the next section, we will present the probability model for RFID-based range queries.

![Spatial information model illustration](image)

**Fig. 3.** Spatial information model illustration

### 4 Probability Modeling for RFID-based Range Query

In this section, we propose how to effectively evaluate the positions of the tagged objects and model the probability for some tagged object in the given range.

#### 4.1 Probability Evaluation for Object Position

In aware regions, monitored objects will execute some tasks in a relatively static manner. For example, a visitor will roam in an exhibition area to enjoy some artwork. Therefore, the positions of monitored objects can be reasonably assumed to follow the uniform distribution in the circle. On the other hand, monitored objects on the paths can be modeled as traveling from one aware region to another in the even pace. For some monitored object \( o_i \) along some potential path, following our propoal path shape modeling, the position can be estimated given the velocity \( v_i \). \( v_i \) can be offered as the experience value or evaluated as the average velocity by now and incrementally estimated according to historical readings. For example at \( t_0 \) a visitor left an RFID-monitored aware region (which can be detected by readings) and at \( t_1 \) this visitor entered the next aware region. We can obtain the path distance \( d \) and evaluate the velocity on this path as \( d/(t_1 - t_0) \). Therefore, incremental aggregating these history velocities will be reasonable to infer the current \( v_i \). For the straight line path shape, the position \( (x_i, y_i) \) of \( o_i \) on the path \( \rho(\Upsilon_{k_1}, \Upsilon_{k_2}) \) can be estimated as:

\[
\begin{align*}
  x_i &= \frac{\Upsilon_{k_2}.c_x + v_i t}{\sqrt{(\Upsilon_{k_1}.c_x - \Upsilon_{k_2}.c_x)^2 + (\Upsilon_{k_1}.c_y - \Upsilon_{k_2}.c_y)^2}}(\Upsilon_{k_2}.c_x - \Upsilon_{k_1}.c_x) \\
  &+ \Upsilon_{k_1}.c_x \\
  y_i &= \frac{\Upsilon_{k_2}.c_y + v_i t}{\sqrt{(\Upsilon_{k_1}.c_x - \Upsilon_{k_2}.c_x)^2 + (\Upsilon_{k_1}.c_y - \Upsilon_{k_2}.c_y)^2}}(\Upsilon_{k_2}.c_y - \Upsilon_{k_1}.c_y) \\
  &+ \Upsilon_{k_1}.c_y
\end{align*}
\]

For the broken line situation with inflection points, it is easy to calculate the position by simple geometry computation (get the position of moving objects on a broken line given \( v_i, t \) and inflection points). First, we need calculate the destination \( (x_1, y_1) \) using the above equation. If \( (x_i, y_i) \) have passed over the first inflection point \( (x_1, y_1) \), we need incrementally calculate the position by starting from \( (x_1, y_1) \). By iteratively conducting this process, the position in \( t \) can be defined. To better focus on core ideas, we only discuss the straight line situation in the proposed algorithm when calculating the object position on the path.

Note that from some aware region \( \Upsilon_{k_1} \), there may be more than one possible path. For the real-time query based on current status, we must offer efficient methods to evaluate the probability of heading along different potential paths. Suppose the candidate destination aware region set according to connectivity is \( CS^t(o_i, \Upsilon_k) \). We will calculate the estimated \( \widehat{CS}^t(o_i, \Upsilon_k) \) and the probability \( p_{kk'} \) representing the probability to walk along \( \rho(\Upsilon_k, \Upsilon_{k'}) \) at time stamp \( t \) for any \( \Upsilon_{k'} \in \widehat{CS} \). Obviously, \( \sum_{k'} p_{kk'} = 1 \). One strategy is based on the historical route statistics of different monitored objects to estimate the probability of the path. Suppose the historical aware region sequence of \( o_i \) until \( t \) is \( HS^t_i \). If \( p_{kk'} \) is independent of \( HS^t_i \), we can infer \( CS = \widehat{CS} \) and the probability can be directly marked along the edge of connectivity graph. However, \( p_{kk'} \) has relationship with \( HS^t_i \). For example, if the last path is \( \rho(\Upsilon_{k_1}, \Upsilon_{k_2}) \), there is a quite small probability that \( o_i \) may come back to \( \Upsilon_{k_2} \) and \( p_{kk_2} \) should be estimated as 0. Therefore, what we want to get is the condition probability given \( HS^t_i \). If building an intact possible route tree, given a starting aware region, the route number can reach \( N_c^{N_s} \), where \( N_c \) is the average number one aware region is connected to others and \( N_s \) is the average number of aware regions in a route.
sequence. Obviously, the evaluation workload is quite huge. Also, the statistics information may not reflect the individual features. This statistics-based possible route tree model is only suitable for the applications implicating a few specified routes where all objects routes follow some statistical regularity such as some scenarios in supply chain management [19]. For the general monitoring such as in a smart museum, the realtime path probability evaluation strategy needs to be proposed.

In this paper, we propose an implicated rule-based path probability evaluation strategy. For the smart museum scenario, the implicated rules include: (1) Visitors will avoid visiting the same exhibit twice if possible. (2) Visitors will be more inclined to visit all the exhibits from one room to another instead of turning back. Both rules are reasonable for our proposed application. Based on the two rules, we give a probability ratio factor $c$ to reflect the probability relationship of destinations in different rooms as follows, where $\perp$ denotes two objects are in the same room and $\top$ means not. $\alpha$ can be given according to experience value.

$$c = \frac{p_{kk_1}}{p_{kk_2}} = \{ \alpha \text{ iff } \forall Y_{k_1}, Y_{k_2} \in \tilde{CS}(o_i, Y_k) \wedge Y_{k_1} \perp Y_{k_2} \wedge Y_{k_2} \top Y_k$$

$$\top \text{ iff otherwise.}$$

Besides, an additional array is needed to store the historical records for each object. This array for $o_i$ is denoted as $\mathcal{H}_t^i$ at time $t$. In detail,

$$\mathcal{H}_t^i[k] = \{ 1 \text{ if } \exists \tau, o_i \in Y_k(\tau) \wedge \tau \leq t$$

$$0 \text{ otherwise.}$$

Specifically, we use adjacency list structure $AL_R$ to represent the spatial information graph illustrated in Figure 3. $A_R$ is an array and the ith entry stores the position $(c_x, c_y)$ and radius $c_r$ of aware region $Y_k$, also, the entry of a linked list $AL_k$. The nodes of $AL_k$ represent all the aware region pointers having connectivity with $Y_k$ and $f_1$ represents the corresponding aware region ID. Room and inflection point information is stored if there is any. Specially, the aware regions in the same room will be stored in the adjacent place for evaluation convenience. Combined with $AL_R$, the key steps of the path probability evaluation strategy PPE based on the implicated rules is given in Algorithm 1. Note that $n_1$ records the number of the connected aware regions which are in the same room as the current aware region. $n_2$ represents the number of the connected aware regions which are not in the same room as the current aware region. Based on rule(1), we first consider the unvisited aware regions (computation by $n_{in}$ and $n_{out}$). Only when every connected aware regions are visited, $1/(n_1 + n_2)$ can be represented as the possibility for a monitored object to move to each connected region.

<table>
<thead>
<tr>
<th>Algorithm 1: Path Object Probability Evaluation Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input:</strong> $o_i, Y_k, \mathcal{H}_t^i, AL_R$</td>
</tr>
<tr>
<td><strong>Output:</strong> $\tilde{CS}(o_i, Y_k), p_{kk_1}(\forall Y_k' \in \tilde{CS})$</td>
</tr>
<tr>
<td>foreach item $e_m$ in $AL_k$ do</td>
</tr>
<tr>
<td>if $Y_{e_m, f_1} \perp Y_k$ then</td>
</tr>
<tr>
<td>if $\mathcal{H}_t^i[e_m, f_1] = 0$ then</td>
</tr>
<tr>
<td>$l \leftarrow l + 1$</td>
</tr>
<tr>
<td>end</td>
</tr>
<tr>
<td>if $Y_{e_m, f_1} \top Y_k$ then</td>
</tr>
<tr>
<td>if $\mathcal{H}_t^i[e_m, f_1] = 0$ then</td>
</tr>
<tr>
<td>$l \leftarrow l + 1$</td>
</tr>
<tr>
<td>end</td>
</tr>
<tr>
<td>foreach item $e_m$ in $AL_k$ do</td>
</tr>
<tr>
<td>if $n_{out} \neq 0$ then</td>
</tr>
<tr>
<td>if $Y_{e_m, f_1} \perp Y_k \wedge \mathcal{H}_t^i[e_m, f_1] = 0$ then</td>
</tr>
<tr>
<td>$\tilde{Y}_{e_m, f_1} \in \tilde{CS}(o_i, Y_k)$;</td>
</tr>
<tr>
<td>$p_{kk_1, e_m} f_1 \leftarrow 1 \over n_{in} \alpha + n_{out}$;</td>
</tr>
<tr>
<td>end</td>
</tr>
<tr>
<td>if $n_{out} = 0 \wedge n_{in} \neq 0$ then</td>
</tr>
<tr>
<td>if $Y_{e_m, f_1} \perp Y_k \wedge \mathcal{H}_t^i[e_m, f_1] = 0$ then</td>
</tr>
<tr>
<td>$Y_{e_m, f_1} \in \tilde{CS}(o_i, Y_k)$;</td>
</tr>
<tr>
<td>$p_{kk_2, e_m} f_1 \leftarrow 1 \over n_{out}$;</td>
</tr>
<tr>
<td>end</td>
</tr>
<tr>
<td>if $n_{out} = 0 \wedge n_{in} = 0$ then</td>
</tr>
<tr>
<td>if $Y_{e_m, f_1} \in \tilde{CS}(o_i, Y_k)$;</td>
</tr>
<tr>
<td>$p_{kk_2, e_m} f_1 \leftarrow 1 \over n_{in} + n_{out}$;</td>
</tr>
<tr>
<td>end</td>
</tr>
</tbody>
</table>

Based on Algorithm 1, we can estimate the position of $o_i$ for $o_i \in \tilde{Y}(t_{now})$. Furthermore, the key steps of the position estimation is given in Algorithm 2. The output is the estimated candidate position $\tilde{CS}_p$ and the corresponding probability $p_{(x,y)}^P$ of each position $(x,y)$.
Algorithm 2: Path Position Probability Evaluation Algorithm

Input : $v_i,o_i, \overline{CS}_{1\text{now}}(o_i, \mathcal{T}_k), p_k; \forall \mathcal{T}_k \in \overline{CS}, t_{\text{last}}$
Output: $\overline{CS}_p, P'(x,y) \forall (x,y) \in \overline{CS}_p$

foreach $\mathcal{T}_k \in \overline{CS}$ do
    $x = \sqrt{(Y_k.x - Y_{k'}x)^2 + (Y_k.y - Y_{k'}y)^2}(Y_k.x - Y_{k'}x) + Y_{k'}x$;
    $y = \sqrt{(Y_k.x - Y_{k'}x)^2 + (Y_k.y - Y_{k'}y)^2}(Y_k.y - Y_{k'}y) + Y_{k'}y$;
    if $(x,y) \in P(\mathcal{T}_k, \mathcal{T}_k')$ then
        $(x,y) \rightarrow \overline{CS}_p$;
        $\gamma = \gamma + p_k$;
        $P'(x,y) = \frac{1}{\pi^2}P(x,y)$;
    end
end

4.2 Probability Evaluation for Range Query

In this subsection, we propose the probability evaluation for the RFID range query by two theorems.

Theorem 1. When the monitored object is in the aware region, the query range radius is $l$, the probability that the monitored object $o_i$ is in the range of focal object $o_f$ is

$$P(x,y) = \frac{\alpha l^2 - \beta}{\pi^2} \sin 2\alpha l^2 + \gamma \theta \gamma_{\mathcal{T}_R(o_i), o_f}^2 \sin 2\theta / 2,$$

where $\alpha l^2 - \beta > 0$ and the integration region is $\mathcal{T}_{\mathcal{R}(o_f)}$ and the notations are illustrated in Figure 4.

Proof. Suppose $O_1$ and $O_2$ represent $\mathcal{T}_{\mathcal{R}(o_i)}$ and $\mathcal{T}_{\mathcal{R}(o_f)}$ separately. Choose a point $Y(x,y)$ in the circle $O_2$ randomly and $Y(x,y)$ follows the uniform distribution in $O_2$. The distance $d$ from $Y$ to the center of $O_1$ can be calculated as

$$d = \sqrt{(x - \mathcal{T}_{\mathcal{R}(o_i), o_f})^2 + (y - \mathcal{T}_{\mathcal{R}(o_i), o_f})^2}.$$

Suppose the probability of the distance from $Y$ to any point in $O_1$ smaller than $l$ is $P(x,y)$. Shown in Figure 4, an arc can be drawn with the center $Y$ and radius $l$ then $P(x,y)$ can be inferred to be proportional to the intersection area of the two circles. According to cosine theorem, $\mathcal{T}_{\mathcal{R}(o_i), o_f}^2 + d^2 - 2\mathcal{T}_{\mathcal{R}(o_i), o_f}d\cos \theta = l^2$, and thus $\theta = \arccos(\mathcal{T}_{\mathcal{R}(o_i), o_f}^2 + d^2 - l^2)/2\mathcal{T}_{\mathcal{R}(o_i), o_f}d$. In the same way, $\alpha = \arccos(l^2 + d^2 - \mathcal{T}_{\mathcal{R}(o_i), o_f}^2)/2dl$. The right bow area of the intersection is $\theta \mathcal{T}_{\mathcal{R}(o_i), o_f}^2 + d^2 - 2\mathcal{T}_{\mathcal{R}(o_i), o_f}d\sin 2\theta / 2$ and the left one is $\alpha l^2 - l^2\sin 2\alpha l^2 / 2$, so the total area of the intersection is $\alpha l^2 - l^2\sin 2\alpha l^2 / 2 + \theta \mathcal{T}_{\mathcal{R}(o_i), o_f}^2 + d^2 - 2\mathcal{T}_{\mathcal{R}(o_i), o_f}d\sin 2\theta / 2$. Therefore, $P(x,y) = \frac{\alpha l^2 - l^2\sin 2\alpha l^2 + \theta \mathcal{T}_{\mathcal{R}(o_i), o_f}^2 + d^2 - 2\mathcal{T}_{\mathcal{R}(o_i), o_f}d\sin 2\theta / 2}{\pi^2}$.

Furthermore, the probability that the monitored object $o_i$ is in the range of focal object $o_f$ is $\int \mathcal{T}_{\mathcal{R}(o_i), o_f}^2 + d^2 - 2\mathcal{T}_{\mathcal{R}(o_i), o_f}d\sin 2\theta / 2$ where the integration region is $O_2$ and the query result set composed of all the $<o_i,p_i>$ is denoted as $\overline{Q}_S$. The theorem is proven.

![Fig. 4. Range query probability modeling for target object in the aware region](image)

For simplicity, we choose $\mathcal{T}_{\mathcal{R}(o_i), o_f}$ as the estimated result for the approximate evaluation instead of evaluating integration. The error bound analysis is omitted here. Obviously, when $l < d - \mathcal{T}_{\mathcal{R}(o_i), o_f} - \mathcal{T}_{\mathcal{R}(o_f), o_f}$, the probability is 0 and when $l > d + \mathcal{T}_{\mathcal{R}(o_i), o_f} + \mathcal{T}_{\mathcal{R}(o_f), o_f}$, the probability is 1.

Theorem 2. When the monitored object $o_i$ is on the potential path and its position $Y(x,y)$ is in $\overline{CS}_p$. Given the query range radius $l$, the probability that $o_i$ is in the range of focal object $o_f$ is $\sum_{(x,y)\in \overline{CS}_p} P'(x,y)$, where $P'(x,y) = \frac{\alpha l^2 - \beta}{\pi^2} \sin 2\alpha l^2 + \gamma \theta \gamma_{\mathcal{T}_R(o_i), o_f}^2 \sin 2\theta / 2$.
an object is in some aware region or on some potential path. The corresponding estimated positions and probabilities will be evaluated according to Theorem 1 or Theorem 2. Obviously, a high traversal cost will be incurred for the basic query method. Enhanced strategies need to be designed to improve the efficiency.

Algorithm 3: Basic Moving Range Query Algorithm

Input: \( \alpha_f, \mathcal{A}_k^{t_{now}} \)
Output: \( \mathcal{A}_k^{t_{now}} \cap \mathcal{Q}_S \)

foreach RFID reading \( p_1 \) of \( t_{now} \) do
\( \mathcal{A}_k^{t_{now}}[i].f_1 = t_{now} \);
if \( \mathcal{A}_k^{t_{now}}[i].f_1 \neq p_1.f_1 \) then
\( \mathcal{A}_k^{t_{now}}[i].f_1 = p_1.f_1 \);
end
end

if \( \mathcal{A}_k^{t_{now}}[i].f_1 \neq p.f_1 \) then
foreach \( i \) in \( \mathcal{A}_k^{t_{now}} \) do
if \( \mathcal{A}_k^{t_{now}}[i].f_1 = t_{now} \) then
\( p_1 = \frac{\alpha_f - l^2 \sin 2\alpha/2 + \theta \mathcal{Y}_{(o_j)}.c_2^2 - \mathcal{Y}_{(o_j)}.c_1^2 \sin 2\theta/2}{\pi \mathcal{Y}_{(o_j)}.c_1^2} \); \( p_1 > 0 \) then
\( < o_i, p_i > \in \mathcal{Q}_S \);
end
end
if \( \mathcal{A}_k^{t_{now}}[i].f_1 < t_{now} \) then
\( p_1 = \sum_{(x,y) \in \mathcal{C}_S} P(x,y) ; \)
if \( p_1 > 0 \) then
\( < o_i, p_i > \in \mathcal{Q}_S \);
end
end

5.2 Query Optimization based on Aware Region Pre-ordering

According to solution-indexing techniques, we first introduce a static ordering list for each aware region \( \mathcal{Y}_k \) to store all other aware regions such as \( \mathcal{Y}_k' \) according to the value of \( d(\mathcal{Y}_k, \mathcal{Y}_k') - \mathcal{Y}_k.c_r - \mathcal{Y}_k'.c_r \). All the ordering lists of different aware regions institute a matrix \( \mathcal{M} \), i.e., \( \forall k, j_1, j_2, \mathcal{M}[k][j_1] = k_1 \land \mathcal{M}[k][j_2] = k_2 \land j_1 < j_2 \Rightarrow d(\mathcal{Y}_k, \mathcal{Y}_k_1) - \mathcal{Y}_k.c_r - \mathcal{Y}_k_1.c_r \leq d(\mathcal{Y}_k, \mathcal{Y}_k_2) - \mathcal{Y}_k.c_r - \mathcal{Y}_k_2.c_r \). Besides, we construct a dynamic additional structure \( \mathcal{Y}_k'(t_{now}) \) to store the current monitored objects in each aware region and the monitored objects on the path with the departure place \( \mathcal{Y}_k' \), i.e., \( o_i \in \mathcal{Y}_k'(t_{now}) \Leftrightarrow \mathcal{A}_k^{t_{now}}[i] = k \). Furthermore, we give the distance of an aware region and a potential path as:
d(Υk, ρ(Υk1, Υk2)) = d((Υk1, cX, Υk1, cY), (X′, Y′)) \Rightarrow (X′, Y′) \in \rho(Υk1, Υk2) ∧ (∀(x, y) \in \rho(Υk1, Υk2), d((Υk1, cX, Υk1, cY), (x, y)) \Rightarrow (d(Υk1, Υk2) = d(Υk1, Υk2)) ∧ \bigwedge_{x \in \rho(Υk1, Υk2)} (x, y) \neq \rho(Υk1, Υk2))

\begin{align*}
\end{align*}

\textbf{Definition 4 (Non-supervised Path). Given an aware region Υk, for a path }\rho(Υk1, Υk2)\text{, if } d(Υk, ρ(Υk1, Υk2)) < d(Υk, Υk1) - Υk1, cY ∧ d(Υk, ρ(Υk1, Υk2)) < d(Υk, Υk2) - Υk2, cY, \text{ we say } ρ(Υk1, Υk2) \text{ is the non-supervised path of } Υk. \text{ The non-supervised path set of } Υk \text{ is denoted as } NS^l_k.

\textbf{Definition 5 (l-Non-supervised Path). If } ρ(Υk1, Υk2) \in NS^l_k \land d(Υk, ρ(Υk1, Υk2)) < Υk, cY < l, \text{ we say } ρ(Υk1, Υk2) \text{ is the l-non-supervised path of } Υk. \text{ The l-non-supervised path set of } Υk \text{ is denoted as } NS^l_k.

\textbf{Theorem 3. For a triggered moving range query, given the focal object }o_f, \text{ and the range radius }l, \text{ if } K = R^{t_{now}}(o_f), \text{ when } o_i \text{ is the query result and } o_i \in ρ(Υk1, Υk2)(t_{now}), \text{ we can infer } d(Υk, Υk1) - Υk1, cY - Υk2, cY < l \land d(Υk, Υk2) - Υk2, cY - Υk1, cY < l \land \bigwedge_{x \in ρ(Υk1, Υk2)} (x, y) \neq ρ(Υk1, Υk2) \text{ according to the non-supervised path definition. The theorem is proven.}

\textbf{Algorithm 4: Range Query based on aware region Pre-ordering}

\begin{algorithm}
\caption{Algorithm 4: Range Query based on aware region Pre-ordering}
\begin{algorithmic}
\State \textbf{Input}: \text{ } o_f, A_{\text{now}}^{l\text{-}}
\State \textbf{Output}: QS_f
\ForEach {RFID reading } p_i \text{ of } t_{\text{now}}
\If {A_{\text{now}}^{l\text{-}}[i].f_r \neq p_i, f_r}
\State \text{remove}(o_i);
\State T_{p_i, f_r}.add(o_i);
\EndIf
\EndFor
\If {A_{\text{now}}^{l\text{-}}[i].f_r \neq p_i, f_r}
\ForEach {k' \in M[k]}
\State check \text{item } e_m \text{ in } AL_{k'}
\If {x', y' \in S'}
\State remove \text{ all } o_i \text{ in } Y_{k'}(t_{\text{now}})
\If {A_{\text{now}}^{l\text{-}}[i].f_i < t_{\text{now}}}
\State \text{remove}(o_i);
\State T_{p_i, f_i}.add(o_i);
\EndIf
\EndIf
\EndFor
\EndIf
\EndFor
\EndAlgorithm
\end{algorithm}

\textbf{5.3 Query Optimization based on Monitored Object Dynamics}

In this subsection, we consider the features of the continuous query and the moving object’s position change, utilize the velocity information to improve the basic method, and propose a query optimization strategy MOD. Specifically, two additional fields \(< min_t, tag_p >\) are added to the
schema of $A_c$. $\min_t$ represents the minimum possible time for one monitored object to change the result status of some query. $\tag{p} = 1$, $\tag{p} = 2$ and $\tag{p} = 0$ separately represent the object is definitely in, possibly in and definitely out of the query range. The items with $\tag{p} = 0$ are neglected, while the items with $\tag{p} = 1$ are just added into the result set with $p_i = 1$. And the items with $\tag{p} = 2$ need calculating $p_i$ according to Theorem1 or Theorem2.

Fig. 6. Illustration of non-supervised path and re-maintenance estimation

The calculation of $\min_t$ for different $\tag{p}$ is illustrated in Figure4(c). When $o_5$ is out of the query range, the ultimate case for $o_5$ to quickly enter the query range is the imaginary $o_5'$ and $o_5''$ move along the straight line $p1$ face to face, where $\min_t = \frac{d(\gamma_{R(o_5')}, \gamma_{R(o_5'')})}{v_i + v_f}$. Also, for $o_5$ to travel out of the query range, $o_5'$ and $o_5''$ move in the opposite direction along $p2$ will be the bound case, where $\min_t = \frac{1-d(\gamma_{R(o_5')}, \gamma_{R(o_5'')})}{v_i + v_f}$. For the situation of the objects which are on the paths, the equation for calculating $\min_t$ are respectively $\min_t = \min(\forall(x', y') \in CS_p, \frac{d((x', y'), \gamma_{R(o_5')})}{v_i + v_f} + \frac{d((x', y'), \gamma_{R(o_5'')})}{v_i + v_f})$. But for the situation of $\tag{p} = 2$, we have to judge at the next time point because the status may be changed, so $\min_t = \min_t + 1$. Because the moving features of monitored objects are considered, some objects could be directly filtered from the result set. For a continuous query, the method can gain good response time because of scanning fewer objects in $A_c$.

MOD utilizes the velocity and the Euclidean distance to estimate $\min_t$, which is a lower bound. Furthermore, we propose an alternative method called path-based Monitored Object Dynamics(pbMOD), which considers the features of the potential path of moving objects to enlarge $\min_t$ under the premise of ensuring the accuracy.

Theorem 4. If $d(o_i, o_f) > l$, we can prove $\max(\min_t) \geq \sin (\min(\alpha/2))D_p(o_i, o_f) - l \geq \frac{d(o_i, o_f) - l}{v_i + v_f}$, where $D_p$ denotes the path between two objects and $\alpha$ denotes the intersection angle of any two sub-path of $D_p$.

Proof. Suppose $o_a$ and $o_b$ are separately on the two sub-paths with the intersection point $c$ and intersection angle $\alpha$, according to cosine theorem, $d(o_a, o_b)^2 = d(o_a, c)^2 + d(o_b, c)^2 - 2d(o_a, c)d(o_b, c)\cos(\alpha)$, where $D_p(o_a, o_b) = d(o_a, c) + d(o_b, c)$. By computing the optimum solution, we can prove $\min(d(o_a, o_b)) = \sin(\alpha/2)D_p(o_a, o_b)$. Furthermore, by expanding the equation to the global path area, we can infer $l \geq \sin(\min(\alpha/2))(D_p(o_i, o_f) - v_i * \max(\min_t) - v_f * \max(\min_t))$. Also, by using the straight line distance, $\frac{d(o_i, o_f) - l}{v_i + v_f}$ can be certified as the lower bound of $\min_t$ according to MOD. And thus, the theorem is proven.

According to theorem 4, pbMOD utilizes the new bound based on the observation that $\sin(\min(\alpha/2))D_p(o_i, o_f) - l \geq \frac{d(o_i, o_f) - l}{v_i + v_f}$, which may be a better estimation for $\min_t$ compared to $\frac{d(o_i, o_f) - l}{v_i + v_f}$. Especially, by utilizing solution-based indexing methods, for any two aware region center $c_1$ and $c_2$, $D_p(c_1, c_2)$ as well as corresponding $\min(\alpha)$ can be pre-computed off-line. When a query is triggered, $D_p(o_i, o_f)$ can be inferred according to $D_p(c_1, c_2)$ which will save a lot of on-line maintenance cost compared to computing exact $\max(\min_t)$. Furthermore, in the case $d(o_i, o_f) < l$, following theorem4, we can infer $\min_t = \frac{l - \sin(\min(\alpha/2))D_p(o_i, o_f)}{(v_i + v_f)\sin(\max(\alpha/2))}$ by using pbMOD.

Although the efficiency of pbMOD is higher, by using Dijkstra algorithm, the pre-computation cost may be too high to be suitable for the scenarios with a large number of readers.
6 Experiments

In this section, we evaluate the efficiency and accuracy of our proposed models and methods by generating data from a simulated scenario.

6.1 Experiment Settings

We simulated a real-world smart museum scenario deployed with RFID readers. The environment is modeled as a 200m x 100m area (default case) which includes some rooms. A glance of the layout for rooms and RFID readers is shown in Figure 7. The circles denote aware regions, and the solid lines connecting them represent potential paths. The dashed lines separate the regions into different rooms. The mobility model needs to reflect the most visitors’ potential behaviors. For a visitor moving from an aware region to the next, an aware region that has connection with the current aware region needs to be chosen. The unvisited aware regions will be first considered, among which the aware regions in the same room as the current region will be assigned a higher probability. If all the candidate regions are visited, the visitor will move to next region randomly.

Since we intend to investigate the impacts of the imbalance and changes in the workloads, we have randomly selected some points in the space as hotspots. Each hotspot uses a uniform distribution to generate objects around it. The object distributions with different number of hotspots are shown in Figure 7. We will evaluate its effect in the following experiments. When \( N_{hp} = 2N_r \), object can be taken as following uniform distribution. In order to study other influencing factors respectively, we use uniform workloads in most experiments unless explicitly stated. So we set \( 2N_r \) as the default value for \( N_{hp} \). The hotspots have different initial velocities with \( v_{obj} \) on average. The velocity of each object varies in the range of twice of their initial ones in simulation.

The performance of our proposed model and methods will be analyzed from three aspects: time efficiency, result accuracy and storage cost. The parameter settings in the experiments are summarized in Table 1. In each experiment, we vary a single parameter, and set the remaining ones to their default values based on the real world situation. Note that a threshold is introduced to filter the query result. Only when the result probability is above it, the object can be output. All experiments were conducted on a PC with Intel Core 2 Duo 2.00 GHz processor, 2.0GB memory and 160GB SATA Disk, running Window XP. Our simulation is written in j2sdk1.5.0_11.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>( l(m) )</td>
<td>20</td>
<td>range radius of query</td>
</tr>
<tr>
<td>( N_{obj} )</td>
<td>1000</td>
<td>number of objects</td>
</tr>
<tr>
<td>( v_{obj}(m/s) )</td>
<td>0.4</td>
<td>average velocity of objects</td>
</tr>
<tr>
<td>( N_{hp} )</td>
<td>2( N_r )</td>
<td>number of hotspots</td>
</tr>
<tr>
<td>( N_r )</td>
<td>20</td>
<td>number of regions</td>
</tr>
<tr>
<td>( R_c(m) )</td>
<td>6</td>
<td>average radius of regions</td>
</tr>
<tr>
<td>threshold</td>
<td>0.7</td>
<td>result probability threshold</td>
</tr>
<tr>
<td>( \alpha )</td>
<td>2</td>
<td>probability ratio factor</td>
</tr>
<tr>
<td>( L_{path}(m) )</td>
<td>20</td>
<td>average length of paths</td>
</tr>
<tr>
<td>( S(km^2) )</td>
<td>( 20*10^{-3} )</td>
<td>average area of the scenario</td>
</tr>
</tbody>
</table>

6.2 Efficiency

The efficiency of our proposed model and methods can be illustrated by the average response time and average maintenance cost per query result. The maintenance cost means the cost to maintain and update our proposed data structure because of incoming RFID readings which can be incurred without any query. The response time can be evaluated by the query processing cost and the maintain cost at the time point when a query is triggered.

Figure 8 measures the effect of \( N_{obj} \). Under all conditions, the response time related to LRP and MOD can be improved compared with the basic method. That is because LRP and MOD only process a portion of the objects which are possibly in the scope of query. When \( N_{obj} > 1600 \), the response time of LRP begins to exceed MOD. For the maintenance cost, the basic method will outperform other alternatives because fewer data structures are needed. Also, because MOD needs to maintain the \( max_t \) of some objects frequently, a higher maintenance cost can be incurred when a large number of objects are involved.
As shown in Figure 11(a), the response time tends to increase with the expansion of \( l \). Because more objects are likely in the query range, and
Thus more processing time is incurred. In the default situation where \( R_r = 6m \), we can infer the average center distance between regions is 30m. In this case, for the basic method and LRP, the increased part of the response time is caused by the objects on the paths. The response time of MOD displays a steady growth trend. Because as \( l \) enlarges, some objects turn to be correlative with the query, resulting in a larger increase in processing time. Figure 11(b) shows that \( l \) has no impact on the maintenance cost of the basic method and LRP, while there is a slight impact on MOD. Because MOD is based on \( v_{obj} \) and the query range to estimate \( \max_l \).

Figure 12 reflects the effect of \( N_{hp} \) on time efficiency. With the increase of \( N_{hp} \), the response time of the basic method keeps stable after increasing and the maintenance cost is still available. That is because as the number of objects belonging to the same hotspot with the focal object drops, objects which need calculation and judgement get more. The response time of LRP displays the trend of reducing at first and increasing again. And the maintenance cost of LRP decreases, which will stabilize when \( N_{hp} > 40 \). Because fewer entries of some indexes are involved as \( N_{hp} \) increases. So the indexes which need maintenance and traverse get fewer. Although the factors that affect the basic method also exist, the response time still follows an overall decreasing trend. When \( N_{hp} > 40(2N_r) \) (Figure 7(c)), the factors that affect the basic method turn to dominate. As a result, there is a rising trend. Both of MOD’s two curves tend to keep smooth, which shows that \( N_{hp} \) has no significant impact on them. It is because maintenance frequency and maintenance quantity are not affected. Even if the object distributions are different, there are only limited fluctuations for the maintenance cost.

Figure 13 reflects the efficiency differences between MOD and pbMOD. As shown in the figure, the response time and the maintenance time decrease when the size of the whole monitored area increases. That’s because the enlargement of the whole monitored area may lead to the longer distance between objects and also the increased \( \min_t \). The maintenance cost becomes lower because MOD and pbMOD don’t need to maintain \( \min_t \) of objects so frequently in the distance. For pbMOD uses shortest path instead of Euclidean distance to compute \( \min_t \), less maintenance cost and response time are needed. Therefore, pbMOD is better than MOD in time efficiency in this case.

6.3 Accuracy

In this paper, we use recall and precision to study the accuracy of the three methods. Recall is defined as \( \frac{tp}{tp+fn} \), and precision is defined as \( \frac{tp}{tp+fp} \), where \( tp \) denotes the number of objects appearing both in the estimated result and the accurate result; \( fn \) represents the number of objects which appear in the accurate result but not in estimated result; \( fp \) represents the number of objects which appear in the estimated result but not in the accurate result. The query will be triggered several times.
times, and we will use the average values as the test result.

![Graph](image1)

![Graph](image2)

(a) Recall vs. threshold  
(b) Precision vs. threshold

Fig. 14. Effect of threshold

As shown in Figure 14, the values of recall and precision of LRP and MOD are very close to each other, and recalls of all the methods can reach 78% at least while precisions can reach more than 70% (the basic method and LRP have the same values of recall and precision, which will not be mentioned repeatedly in the following parts). Recall and precision of LRP are slightly higher than those of MOD, because MOD is more dependent on $v_{obj}$. With the increase in threshold, the value of recall decreases and goes to stability, while the value of precision increases and goes to stability. It is clear that in the estimated results, the objects which should be in the accuracy results are often evaluated with high probability. When threshold increases, the likely results with low probability will be filtered, so recall decreases and precision increases. When threshold $\geq 0.7$, recall and precision approximately achieve stability.

![Graph](image3)

(a) Recall vs. $N_{obj}$  
(b) Precision vs. $N_{obj}$

Fig. 15. Effect of $N_{obj}$

Figure 15 demonstrates the impact of varying $N_{obj}$. With the increase in $N_{obj}$, while $N_{hp}$ keeps unchanged, the number of objects in each hotspot increases. So the ratio of the objects which are certain in the query range increases and the ratio of errors in position estimation declines relatively. As a result, the values of recall and precision will increase, while the accuracy gap between MOD and LRP is getting smaller. When $N_{obj}$ is small, there are few objects composing the results. And the precision of MOD is slightly higher than that of LRP. But with the increasing number of objects composing the results, the precision of MOD will be lower than that of LRP. Because the objects whose positions are estimated with errors increase in MOD compared with LRP. When $N_{obj}$ reaches a certain value, recall and precision will be stabilized.

![Graph](image4)

(a) Recall vs. $N_r$  
(b) Precision vs. $N_r$

Fig. 16. Effect of $N_r$

As shown in Figure 16, we can discern that recall has started to level off with an up trend as $N_r$ increases. Precision shows a steady scenario with all above 80%. For $N_{obj}$ is fixed, the number of regions which objects could be within increases as $N_r$ rises. So the probability in the query range decreases. But when it decreases to a certain level, $N_r$’s influence can be neglected. As $N_r$ increases, the accuracy gap between LRP and MOD gets larger. Because larger $max_l$ will lead to longer maintenance intervals, and there will be higher uncertainty in $v_{obj}$. In general, both LRP and MOD perform well in accuracy stability on $N_r$.

![Graph](image5)

(a) Recall vs. $l$  
(b) Precision vs. $l$

Fig. 17. Effect of $l$

With the expansion of $l$, the values of recall and precision act as a cyclical change as shown in Figure 17. Because $L_{path}$ and $R_r$ are fixed, then the query scope include few new regions over a
period of time although \( l \) still expands. So recall will decrease gradually at this time, while precision will go stable. When \( l \) continues expanding to a certain value, the query range begins to include new regions, so that precision will decrease gradually then increase sharply, acting as such a cyclical change. In general, both recall and precision could reach 60\% at least, and the accuracy shows good stability on \( l \).

\[
\begin{align*}
\text{Recall} &
\end{align*}
\]

\[
\begin{align*}
\text{Number of Hot Points} &
\end{align*}
\]

Basic Method
LRP
MOD

Figure 18 illustrates that given \( L_{\text{path}} \), if \( v_{\text{obj}} \) is low leading to longer time spent on the paths, there will be a higher probability to estimate the objects’ positions when the query is triggered, so the accuracy is lower. But the accuracy increases with \( v_{\text{obj}} \) grows. When \( v_{\text{obj}} \) increases to a certain degree, the impact of the \( v_{\text{obj}} \) variation becomes predominant. Thus, the higher the \( v_{\text{obj}} \) is, the larger the \( v_{\text{obj}} \) variation is, so the lower the result accuracy is.

\[
\begin{align*}
\text{Recall} &
\end{align*}
\]

\[
\begin{align*}
\text{Number of Hot Points} &
\end{align*}
\]

(a) Recall vs. \( v_{\text{obj}} \)  
(b) Precision vs. \( v_{\text{obj}} \)

Fig. 18. Effect of \( v_{\text{obj}} \)

While \( N_{\text{hp}} \) increases, the velocity and the next region for objects to choose are more irrelevant with each other. So the accuracy curves decline with twists and turns as shown in Figure 19. When \( N_{\text{hp}} \) is small, the proportion of the objects whose locations do not need estimating gets larger. Therefore both values of recall and precision are higher. In the course of decline, there will be “a small pick-up”. Because the object distribution is not uniform when \( N_{\text{hp}} < 40. \) The increase in \( N_{\text{hp}} \) makes some objects get far away from the hotspot that the focal object is within. Therefore a part of objects which might appear in the results are not included.

\[
\begin{align*}
\text{Storage cost(KB)} &
\end{align*}
\]

Basic method
LRP
MOD

(a) Storage cost vs. \( N_{\text{obj}} \)  
(b) Storage cost vs. \( N_r \)

Fig. 20. Storage cost test

Finally, we conduct tests on the storage cost. The impacts of varying \( N_{\text{obj}} \) and \( N_r \) are separately demonstrated in Figure 19(a) and Figure 19(b). For the basic method, all the aware region information, an additional array \( \mathcal{A}_c \) recording all the monitored objects’ current status and \( \mathcal{H}_t \) recording all objects’ visiting histories are needed. Therefore, with the increase of \( N_{\text{obj}} \) and \( N_r \), the storage cost will increase correspondingly. For LRP, on the basis of the basic method, a matrix \( \mathcal{M} \) recording the ordering list of all the aware regions and \( \mathcal{T}_l(t_{\text{now}}) \) are constructed in addition. Because the size of \( \mathcal{M} \) will scale in the square number of aware regions, LRP is quite sensitive to the increase of \( N_r \) which can be observed in Figure 19(b). For MOD, because two additional fields \( \min_t \) and \( \max_{hp} \) are introduced for each monitored object, the storage cost could be relatively high when a large number of monitored objects exist.

All the figures above display good time and accuracy performance of the two optimization strategies. Under normal circumstances the accuracy of LRP is slightly higher than that of MOD with little poorer time performance than MOD. It is Because MOD is higher dependent on \( v_{\text{obj}} \) than LRP. In the aspect of storage cost, in general, all the proposed methods can scale to a data-intensive scenario with tolerant storage cost. When there are massive aware regions, LRP will incur much higher storage cost than other alternatives. If the memory space is quite limited, the basic method and MOD could be much more adaptive and efficient.
7 Conclusion

In this paper, we have proposed a framework for the continuous moving spatial range query over RFID monitored objects. The location uncertainty in the region and path has been considered. Based on the reasonable modeling for the RFID spatial information and probability evaluation, one basic method and three optimization strategies are proposed to cope with the probabilistic triggered moving range query in the realtime manner. Finally, we have compared the time and space efficiency as well as the accuracy of various methods and identified the effectiveness of the proposed model through extensive experimental studies. The proposed optimization methods can be adaptive for data-intensive RFID-monitored scenarios.

References


Yu Gu received Ph.D degree in Computer Software and Theory from Northeastern University of China, in 2010. He is now an associate professor in the Department of Computer Science and Technology of Northeastern University. He is a member of ACM and CCF. His area of research is RFID data management.

Ge Yu received his B.E. degree and M.E. degree in Computer Science from Northeastern University of China in 1982 and 1986, respectively, Ph.D degree in Computer Science from Kyushu University of Japan in 1996. He has been a professor at Northeastern University of China since 1996. He is a member of IEEE, ACM, and a senior member of
CCF. His research interesting includes data base theory and technology, distributed and parallel systems, embedded software, network information security.

**Na Guo** received the B.E. degree from Computer Science Northeastern University of China, in 2008. She is a M.E. Candidate in the Department of Computer Science and technology of Northeastern University, China. Her area of research is RFID data management.