Mining Frequent Trajectory Patterns for Activity Monitoring Using Radio Frequency Tag Arrays

Yunhao Liu[1], Lei Chen[1], Jian Pei[2], Qiuxia Chen[1], Yiyang Zhao[1]

[1]Hong Kong University of Science and Technology, Hong Kong
[2]Simon Fraser University, Canada

Abstract

Activity monitoring, a crucial task in many applications, is often conducted expensively using video cameras. Also, effectively monitoring a large field by analyzing images from multiple cameras remains a challenging problem. In this paper, we introduce a novel application of the recently developed RFID technology: using RF tag arrays for activity monitoring, where data mining techniques play a critical role. The RFID technology provides an economically attractive solution due to the low cost of RF tags and readers. Another novelty of this design is that the tracking objects do not need to attach any transmitters or receivers, such as tags or readers. By developing a practical fault-tolerant method, we offset the noise of RF tag data and mine frequent trajectory patterns as models of regular activities. Our empirical study using real RFID systems and data sets verifies the feasibility and the effectiveness of our design.

1. Introduction

In many applications, it is required to monitor activities in closed fields. For example, in chemical plants or large industrial workshops, it is important for security control staffs to monitor "suspicious" activities. Oftentimes, in these applications, the monitoring area is very large and activities (moving trajectories) are sparse. The normal trajectories of moving objects often follow regular patterns. Once we have these patterns, abnormal behaviors of moving objects can be easily detected through pattern matching [7].

Currently, activity monitoring is most often completed using video monitoring equipment such as digital cameras. However, digital cameras are very expensive and each camera can only cover a small area and specific trails. Moreover, it is hard to automatically analyze the activity patterns in a large field with images from multiple cameras.

Consider monitoring the activities in the field illustrated in Fig. 1. Supposing that we know there are two frequent trajectories in the field, we can deploy a set of video cameras to monitor the activities along the frequent trajectories. In order to monitor the activities outside the two trajectories, however, more cameras are needed. Even so, it is hard to automatically analyze the activities captured by many cameras. Typically, security staffs have to monitor the images in real time.

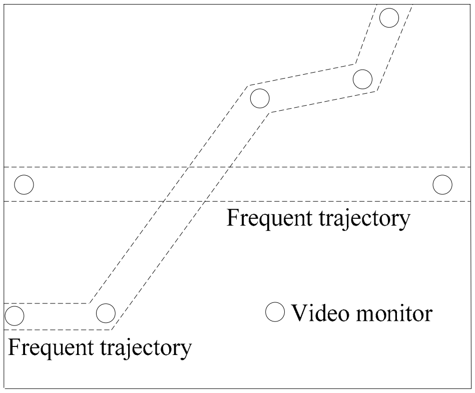


Figure 1: Monitoring activities using video equipment

Monitoring using video cameras has at least the following four limitations. First, *the target trajectories must be pre-defined*. Once the trajectories change, the cameras may need to be re-deployed. Indeed, in many situations, the frequent trajectories may not be known and frequently change over time. Second, *except for the target trajectories, monitoring other regions is difficult.* Third, *it is hard to automatically analyze the images from multiple cameras and detect irregular activities.* And last, *digital cameras are expensive.* It is often a financial concern to deploy a large number of cameras.

In this paper, we describe a novel application of the radio frequency identification (RFID) technology to provide an inexpensive and relatively accurate approach to activity monitoring. Instead of using a series of video cameras, we use an array of RF tags and a few RF readers. Our approach relies on data mining techniques to detect and analyze frequent trajectory patterns. We focus on extracting frequent patterns because these patterns can be used as domain knowledge to capture any anomalies.

Since RF tags and readers are much cheaper than cameras (in US dollars, an active RF tag is about 50 cents and an RF reader is several hundred dollars), and data mining techniques can detect frequent patterns online, our approach is more flexible and much cheaper than the video monitoring solutions.

1.1. RFID and location sensing

RFID is a means of storing and retrieving data through electromagnetic transmission to an RF compatible integrated circuit. It is now being seen as a radical means of enhancing data handling processes [10]. An RF reader can read data emitted from RF tags. RF readers and tags use a defined radio frequency and protocol to transmit and receive data. RF tags are categorized as either passive or active.

Passive RF tags operate without a battery. They reflect the RF signal transmitted to them from a reader and add information by modulating the reflected signal. Their read ranges are very limited.

Active tags contain both a radio transceiver and a button-cell battery to power the transceiver. Since the each tag has an onboard radio, active tags have lager ranges than passive tags. Active tags are ideally suited for the identification of high-unit-value products moving through a tough assembly process. They also offer the durability essential for permanent identification of captive product carriers.

We are interested in using commodity off-the-shelf products. The results of our comparative studies reveal that there are several advantages of the RFID technology. The no-contact and non-line-of-sight nature of this technology is the significant advantage common among all types of RFID systems. All RF tags can be read despite extreme environmental factors such as snow, fog, ice, paint, and other visually and environmentally challenging conditions.

The other advantages are their promising transmission ranges and cost-effectiveness. Indeed, if we deploy a video camera system to cover a 300m×300m factory surface, the cost could be up to a half million US dollars. On the other hand, to deploy an active RFID system merely needs four RF readers and thousands of tags, which would cost less than 10 thousand US dollars.

1.2. Our RFID configuration

After looking into the specifications of different available systems, we have chosen the Spider System manufactured by RF Code [1] to implement our activity tracking prototype.

The RF reader's operating frequency is 308 MHz. It also has an 802.11b interface to communicate with other machines. The detection range is set at 150 feet, and this range can be increased to 1000 feet with the addition of a special antenna. The readers provide digital control of read range via providing configuration software and API with 8 incremental read ranges. Each reader can detect tags within 2 seconds. Each RF tag is pre-programmed with a unique 7-character ID for identification by readers, and its battery life is 3-5 years. Tags send their unique ID signals at random with an average of 2 seconds. Based on the signal strength received by the RF reader, the reader will report or ignore the received ID.

1.3. Our contributions

The major contributions of this work are as follows.

First, *we introduce a novel RFID application that uses an array of stationary RF tags to monitor activities in large fields.* Differing from the traditional radio-based localization methods, our approach does not require the tracking objects to hold any transmitters or receivers, such as RF readers or tags.

Second, *we model a data mining problem that is critical for the activity monitoring application using RFID.* It is fault-tolerant sequential pattern mining from an array of time series generated by the RF tags. This problem has not been addressed in previous data mining studies.

Last, *we conduct an empirical study using real RFID systems and data sets to verify the feasibility and the effectiveness of our approach.*

The rest of the paper is organized as follows. In Section 2, we describe our design of activity monitoring using RF tag arrays. We discuss the data collection and the preprocessing in Section 3 and present the frequent trajectory mining in Section 4. An empirical study is reported in Section 5. Section 6 discusses the related work. The work is concluded in Section 7.

2. Activity Monitoring Using Tag Arrays

Most RFID applications attach RF tags to moving objects such as product items in a warehouse or customer carts in a store. In many applications, however, it is difficult to enforce an RF tag onto every object (e.g., people walking through the field).

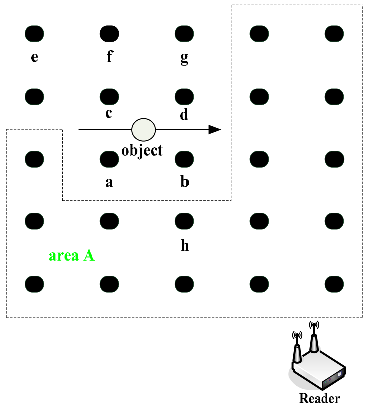


Figure 2: Activity monitoring using RF tag arrays

To tackle this problem, we propose a novel application of RFID. The general idea is as follows. Instead of attaching one RF tag to each object, we deploy an array of active RF tags onto the field. When an object moves through the field, the signals from some active tags will be affected and the RF readers will receive such signals. A database server collects the changes of signal strengths and uses the information to derive the activities in the field.

Figure 2 illustrates our approach, where each blackened box is an RF tag. A set of RF tags is deployed on the field to be monitored. When an object (the circle in the figure) moves in the array, the signal strengths from some RF tags may change. In this example, the strengths from tags *a*, *b*, *c*, and *d* are very likely affected, while the signal strengths from the tags in area *A*, such as *h*, may not be affected.

Figure 3 plots the signal strength changes of RF tags *c* and *h* on a real RF array deployment, as the one shown in Figure 2. The results indicate that when an object passes an RF tag such as *c* at time stamp 10, its signal strength is affected dramatically compared to an unaffected RF tag such as *h*.

By analyzing such changes, we want to derive the trajectories of the activities. Moreover, using the frequent trajectories, we can model the regular activities in a field. When an activity is detected, it can be compared with the frequent trajectories.

Due to the nature of RFID technology, we make the assumption that *the number of simultaneous activities in a field is not large***.** For example, our method can detect several frequent trails that people walk along through a workshop. However, activities such as large parties in a hall or a banquet where hundreds of people walk about randomly cannot be handled well with our current method. Such situations can hardly be handled well by video monitoring systems either. We argue that this assumption holds in many situations.

The novelty of our approach is that we use the interference on the RF tag signals caused by the activities to detect the activities themselves. However, it also poses the following two major challenges, which will be addressed in the remainder of this paper.

Challenge 1: How to detect the positions of objects accurately. RFID data is very noisy. RF tags may have very different characteristics. Some RF tags are very sensitive, i.e., their signal is not stable even when no activities exist. The magnitude of the RF tags also varies. Different RF tags may give very different signal changes even if they are under the same interference.

Challenge 2: How to detect the frequent trajectories of activities. Since the RF tags are not synchronized in sending their signals, some activities may escape from one or a few tags. Moreover, since signals are not synchronized, the order of the changes may not correspond to the spatial-temporal order that an activity happens. How to detect the frequent trajectories effectively and efficiently is far from trivial.

3. Data Collection and Preprocessing

As discussed in Section 2, RF tags might respond differently to interference. In order to identify the interference from moving objects accurately, we need to capture the sensitivity of RF tags.

To measure the sensitivity, we first monitor the signal strengths of tags when no activity is present in the field for a period of *t*. For each tag, we obtain a time series over the period. Let $s_1, \ldots, s_t$ denote the signal strengths collected. Then, we define the *neutral value* of the tag μ_s as the expected signal strength

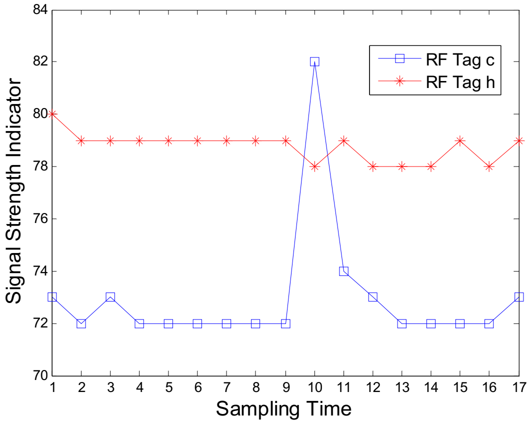


Figure 3: Signal strengths of affected and unaffected RF tags

when there is no interference, i.e., $\mu_s = \frac{\sum_{i=1}^{t} s_i}{t}$. The *sensitivity* of the RF tag is measured by the standard deviation of the time series, i.e., $\sigma_s = \sqrt{(\sum (s_i - \mu_s)^2)/t}$.

When an RF tag is used to detect activities and an object interferes with the signal of the tag, we call the activity an *interference activity* with respect to the tag. With the neutral value and the sensitivity of a tag, we can use a (small) number k ($k > 1$) as the threshold to detect whether interference is likely to happen to an RF tag. Technically, we have the result below following from the Chebyshev inequality.

Theorem 1 (Detection threshold) Let μ and σ be the neutral value and the sensitivity of an RF tag, respectively. During the activity monitoring, if the reader receives a signal from the RF tag of strength s, and $|s-\mu| \geq k\sigma$, *then the probability that an inference activity happens is at least* $(1-\frac{1}{k^2})$.

Proof: Directly derived from Chebyshev's inequality. ▪

An array of RF tags is deployed in a field. An RF tag sends a signal in every unit period (called a period hereafter). RF tags are not synchronized. Instead, they compete for the transmission window. Thus, a tag may send its signal at the end of the period, and its neighbor tag may send its signal at the beginning of the period.

A few readers are connected to a server to collect the signals. Thus, at the server side, a time series is accumulated for each tag and each reader. Using the sensitivity and the neutral value of each tag, we can transform the time series of a tag recorded by a reader R into a binary *tag signal sequence* (or *tag sequences* for short) s_i^R , where $s_i^R = 1$ if the tag is interfered in period i (i.e., $|s_i^R - \mu_s| \geq k\sigma$ according to Theorem 1), and $s_i^R = 0$ if the tag is not interfered in the period.

After the data collection and the preprocessing, we shall use the tag signal sequences instead of the raw signal data in our data analysis.

4. Frequent Trajectory Mining

In this section, we show how to mine frequent trajectories from the RF tag data. We first formulate the problem, and then introduce the algorithm.

4.1. Problem formulation

Since the RF tags deployed are stationary, their spatial locations are known to the server. The data mining task consists of two phases: the training phase and the monitoring phase.

In the *training phase*, we collect the RF tag signal sequences over n periods, where n is a user specified length of time. In practice, the training period can be a day or a week, depending on the nature of the application. The sequences in the training phase will be used to find frequent trajectories as the model of the normal activities in the field.

In the *monitoring phase*, activities are detected and compared with the frequent trajectories. If an activity matches a trajectory, then it is treated as normal. Otherwise, an alert will be issued.

Since the trajectory matching is very similar to the approximate sequence matching problem, many existing methods can be used [7]. In the rest of the paper, we focus on the frequent trajectory mining problem (i.e., the training phase) only.

For each tag μ, let $s(\mu)$ be the tag signal sequence, and $s(\mu)_i$ be the signal in period i.

Intuitively, an activity can be described as a trajectory in the field under monitoring. In a period, the location segment of the object can be determined by the tags that are closest to the segment. Ideally, an activity can be captured by a series of RF tag sets $V_1 \rightarrow \cdots \rightarrow V_l$ where $V_i (1 \leq i \leq l)$ is a set of RF tags describing the location segment of the object in period i, and the tag sets are interfered in consecutive periods.

If the tag sets can be detected accurately, the activity recognition problem is trivial. However, due to the nature of RFID systems, there are a few important obstacles in practice.

First, *not every RF tag along the trajectory may detect the activity*. For example, in Fig. 2, if the object moves fast, it is possible that the object interferes with tag c but not tag d. Moreover, the probability that a tag fails to detect an activity is low but is unknown.

Second, *the signals of tags may not accurately reflect the order of the activity*. For example, in Fig. 2, although the object passes tag c before tag d, the interference may happen in the signal sequence of d before that of c. The reason is that the object may pass c right after c sends a signal of period i, but pass d right before d sends the signal in the same period. Therefore, the interference to d is reflected in period i, but the interference to c is recorded in period (i+1).

Third, *an activity may interfere with multiple tags in a period*. In order to derive the trajectories, we have to infer the possible positions of the object based on the correlation of the interfered tags and the location of the readers.

In summary, the problem of *mining frequent trajectory patterns from RF tag sequences* is to find the trajectories happening at least *min_sup* times in the training phase, where *min_sup* is a user specified frequency threshold.

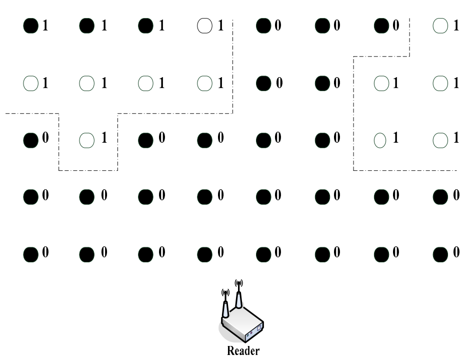


Figure 4: Detecting borders

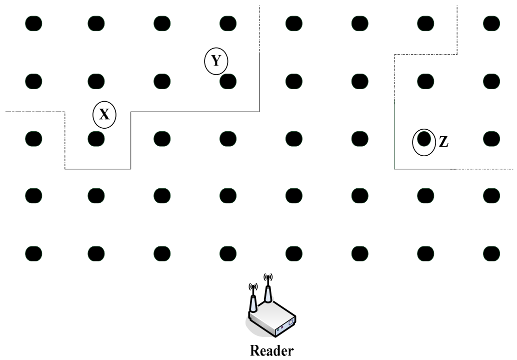


Figure 5: The positions of objects

4.2. Removing redundancy & detecting borders

The RF tag signal collection has the following property.

Property 1 *If a reader R detects that an RF tag u is interfered in a period i, then for any RF tag v behind u in space with respect to R, with high probability, R detects v being interfered in at least one of the following periods:* (*i-d*)*,* (*i-d+1*)*, i,* (*i+d-1*)*, and* (*i+d*)*, where d is a user specified time shifting factor.*

Rationale. The property is clear in geometry. However, note that the property only holds with high probability, since if the object moves fast, there could be a slim window such that the signal of *v* is not affected. The probability is unknown and hard to be estimated. Thus, the property can be used as a heuristic.

Using the above property, we can identify two types of redundancies among RF tag sequences. The first type is the *redundancy among non-interfered tags*. For example, in Fig. 2, all tags in area *A* are likely not interfered. We only need to know the area instead of individual values. The second type is the *redundancy among interfered tags*. For the same reason, the changes of tags *e*, *f* and *g* in Fig. 2 are redundant.

To capture the activity in a period, the border between the interfered tags and the non-interfered tags is good enough. Thus, in each period and for each reader, we derive a border. The border detection works as follows. In a period *i*, we check $s(u)_i^R$ for each RF tag u and reader ***R***. Recall that $s(u)_i^R$ is either 0 or 1. $s(u)_i^R$ is at the border if and only if there is at least one neighbor RF tag *v* such that $s(u)_i^R \neq s(v)_i^R$.
Figure 4 illustrates the snapshot in a period for a reader. The borders are given by the dash-dot lines. The whitened boxes denote the borders of the interfered RF tags. There might exist cases that very few "0"s or "1"s appear inside of an "1" or "0" zone, so that these "0"s or "1"s are treated as outliers and will not be considered during the border detection.

Clearly, when the snapshot in period *i* can be held into main memory, the border detection takes $O(m)$ time where *m* is the number of RF tags in the monitored field. Typically, *m* ranges from tens to thousands of RF tags, which can be easily accommodated in main memory.

4.3. Identifying possible object positions

Once we derive the borders between interfered and non-interfered tags, we identify the possible locations of objects using the spatial map of the stationary tags.

Intuitively, the locations of objects are the outstanding parts of the border that a reader can see. For example, consider the case in Fig. 5. From the reader, two segments (the solid segments in the border) are the possible locations where objects exist. Heuristically, an object may appear proximate to an RF tag u if the tag is at the border and there is no other interfered RF tag blocking the connection between *u* and the reader, such as RF tags x,, y, and z. By walking through the border once, we can identify the segments where an object may exist. We call such segments *object location segments* of the period w.r.t. the reader.

Please note that our location sensing is approximate. We only identify the ranges where objects may exist. Multiple objects may exist in the same range. In our trajectory mining algorithm, we shall use such ranges to assemble the possible trajectories. Another important issue is that some objects may hide behind other objects. For example, in Fig. 6, object *B* is hidden behind object *A*. Theoretically, we should be able to observe more degraded signals from the RF tags interfered by both *A* and *B*, such as the time shifting factor *d*. In the real

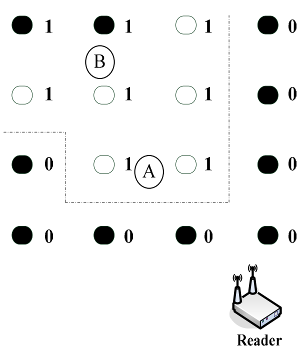


Figure 6: Objects may be hidden

system, however, the difference is often minor and not reliable for location detection.

To detect those hidden objects, we apply the following two methods.

First, *we employ multiple readers*. Multiple readers (e.g., 4-6) are deployed in a field so that the possibility that an object is hidden from all readers is reduced.

Second, *we conduct fault-tolerant mining.* As the objects are moving, one object hidden in one period may show up to some readers in other periods. As long as an object is not hidden at al times from all readers, our algorithm can detect it.

4.4. The mining algorithm

The frequent trajectories are mined in the following two steps.

4.4.1. Finding frequent positions of objects

Clearly,, a tag that is in an object location segment in a period is likely a part of the trajectory of an activity. The trajectory of a frequent activity may frequently trigger a tag in the object location segments. By scanning the object location segments in all periods once, we can find the tags that are in the segments in at least *min_sup* periods with respect to a reader.

Since an object can be occasionally hidden behind other objects, when counting the number of times a tag is in a segment, we also count the cases that tag is in the interfered side of the border. That is, if a tag is in the object location segments in some periods, and is interfered in some other periods, they are summed up together against the threshold *min_sup*.

We do not count the tags that are always hidden behind some tags in the object location segments. The rationale is that those tags are likely to be detected by other readers. On the other hand, if an activity is always hidden by some other activities, it is likely that either the activity is infrequent or it is a part of another activity. In many cases, the interfered tags not in the object location segments do not really capture the movement of objects. The method to find the frequent positions of objects is given in Fig. 7,which shows the cost of the algorithm is one scan of the tag sequences.

4.4.2 Finding frequent trajectory segments

As the second step, we find the frequent trajectory segments. The general idea is that we start with short segments and use them to derive.

Conceptually, a *l-segment* of trajectory is a sequence $V_1 \rightarrow \cdots \rightarrow V_l$ such that $V_j (1 \le j \le l)$ is a set of frequent positions of an object that are spatially adjacent, and V_q and $V_{q+1} (1 \le q \le l)$ are connected in space. In other words, the segment captures an activity in l periods such that V_j describes the trajectory of the activity in the *j*-th period.

We start with finding 2-segments. We check the combinations of frequent object positions and examine whether they happen consecutively in space and in time. To tolerate faults, we allow some appearances in the reverse order. For example, if we see that tag *a* and tag *b* are interfered in consecutive periods frequently, and in some cases, *b* is interfered right before *a*, then, all those cases should be counted together as the support of $a \rightarrow b$. Technically, we use a threshold γ to specify the degree of fault tolerance. In a window of γ periods, the frequent positions can appear in any order. For example, if γ*=2*, then $a \rightarrow b$ and $b \rightarrow a$ are considered matchable; if γ*=3*, then $a \rightarrow b \rightarrow c$ and $c \rightarrow a \rightarrow b$ are matchable.

Typically, γ is a small positive integer such as 2 or 3. The proper value of γ depends on the maximal speed objects can move. If an object moves fast, it may

```
Input: RF tag signal sequences $\{s(u)_i^R\}$, frequency threshold $min\_sup$
Output: the set of frequent positions of objects w,r,t, reader R;
Method:
FOR each tag $u$ DO
    create a counter $c_u = 0$ and a flag $f_u = 0$;
FOR each period $i$ DO
    FOR each tag $u$ DO
        IF $s(u)_i^R = 1$ THEN $c_u = c_u + 1$;
        IF $u$ is at the border of interfered tags
            THEN $f_u = 1$;
FOR each tag $u$ DO
    IF $c_u \ge min\_sup$ AND $f_u = 1$
        THEN output $u$ as a frequent position;
```

Figure 7: Algorithm to find frequent positions of objects

```
Input: RF tag signal sequences $\{s(u)_i^R\}$, frequency
       threshold $min\_sup$
Output: frequent trajectories;
Method:
find frequent positions of objects (Figure 7);
find frequent 2-segments;
FOR each 2-segment DO
    recursively, depth-first extend the segment to
    longer frequent segments, the tags closer to
    the reader should be considered before those
    behind, and once a frequent trajectory is
    found, all segments behind can be pruned;
```

Figure 8: The mining algorithm

have a better chance to cause more unsynchronized signals in more periods.

The space proximity is important here. It distinguishes the trajectories of consecutive movements from the spatial correlation of non-adjacent tags. Since a tag might be interfered by multiple moving objects, some tags non-adjacent in space may appear correlated. Those correlations should be filtered out in mining the frequent trajectories.

By scanning the tag signal sequences once, we can find all 2-segments and their counts (i.e., how many times a segment appears in the training phase). Only those segments appearing at least *min_sup* times are retained as the frequent 2-segments, where *min_sup* is the frequency threshold.

Once the frequent 2-segments are found, we extend them to longer segments and check their support in the data set. To extend a frequent *l*-segment, we check all occurrences of the segment in the data set, and find the frequent positions in the next period following the segment. Those frequent positions adjacent in space form possible extensions to an (l + 1)-segment. We check their frequency to identify the frequent (l + 1)-segments.

The extension of the frequent trajectory segments goes on until we cannot extend a frequent segment any more due to its frequency being lower than the threshold.

One important observation is that the same types of activities may not repeat their trajectories perfectly. For example, many people walk along a frequent trail, but each individual may have some variance. To handle such variance in the mining, we apply a fault tolerant strategy based on Property 1 as follows.

We adopt a depth-first search to extend the frequent segments. The segments closer to the reader have a higher priority to be extended. Once a length $(l+1)$ extension to V_{l+1} of a frequent *l*-segment $V_1 \to \cdots \to V_l$ is infrequent, before we abort the extension, we check whether other extensions of the frequent segment are frequent. Particularly, we check those RF tags behind the tags in V_{l+1}. Fig. 8 summarizes the mining method.

5. Empirical study

In this empirical study, we test our frequent activities mining algorithm on a real implementation of 100 RF tags and 1 reader. As shown in Fig. 9 (only two rows are shown due to space limitations), these RF tags are deployed in 10 rows and each row has 10 RF tags in a field of size $10\text{m} \times 10\text{m}$. The distance between neighboring RF tags within a row or a column is 1m.

We asked our graduate students to walk through this RF array following different routes and different speeds. The signal strength of each RF tag was recorded during the test period. We applied our mining algorithm on the readings of each RF tag received from the reader and reported the accuracy and efficiency of detecting trajectories of frequent activities.

Here, to measure the detection accuracy of our proposed algorithm, we use the ratio between the length of a correctly detected frequent activity's trajectory and the length of the real frequent route. We start the tests with simple activities, such as single or consecutive activities with only one direction and one route (Experiments 1 and 2), then we check the busy actives with two directions (Experiment 3). Finally, we examine the complex activities with multiple routes and directions (Experiment 4).

5.1. Experiment 1: single activity

The purpose of this experiment is to detect the trajectory of a single activity. We set up two routes (trails) in the RF array (as shown in Fig. 9). People walk through trails 1 and 2 independently three times with different speeds (slow—0.5m/sec, fast—1.0m/sec).

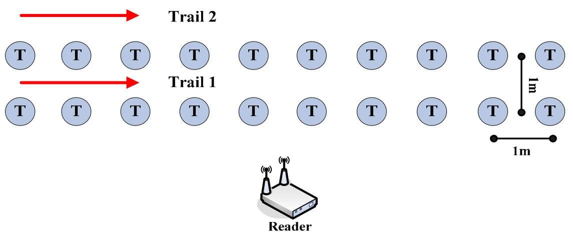
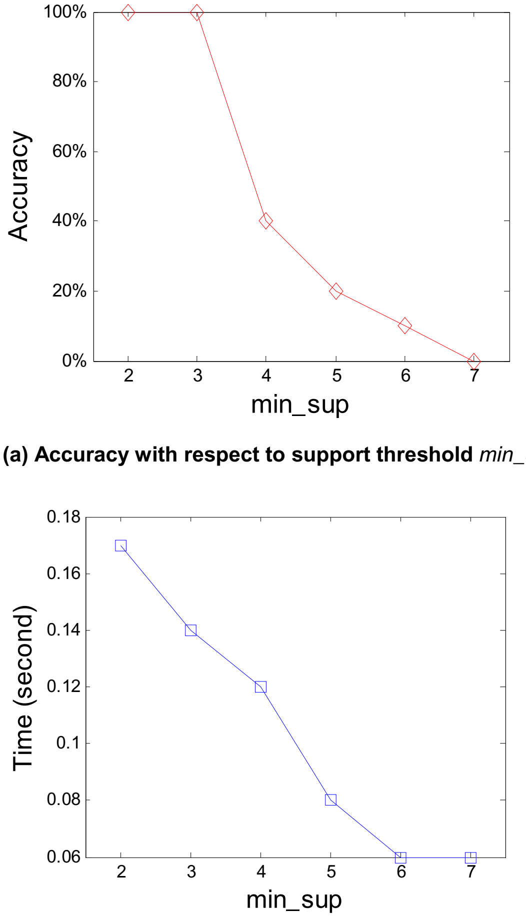


Figure 9: Setup of Experiment 1



(a) Accuracy with respect to support threshold *min_sup*

(b) Runtime with respect to support threshold *min_sup*

Figure 10: The Effect of Minimum Support

The experimental results show that we can get 100% accuracy if we set the threshold, *min_sup*=2, for detecting frequent positions of the objects, no matter what the walking speed of the people is. However, if we set the threshold *min_sup*=3, the accuracy drops down to 60%. Due to the physical setting of RF tags. An RF tag sends a signal within a two second time frame, and there exist cases when people block an RF tag but this RF tag does not transmit any signals during the blocking period. Thus, the reader cannot get the information about the RF tag was affected. As a consequence, this location may not be classified as a frequent one. Therefore, setting to a higher value may lead to a lower accuracy. However, setting *min_sup* to a lower value may result in a large number of frequent locations and the computation cost of detecting frequent trajectories becomes high. We will test the effect of *min_sup* on detecting accuracy and efficiency in Experiment 3, where people may pass an RF tag many times during a busy activity.

5.2. Experiment 2: group activities

The purpose of this experiment is to find the trajectory of a temporally consecutive, group activity. We use the same setting as Experiment 1 and only select trail 1 for testing. We test the following scenario: one person walks through trail 1 at various speeds and the second person starts when the first one arrives at the 8th tag. All walks are in the same direction. In total, five people walk through the trail. We vary the people's walking speeds to test the robustness of the algorithm.

Again, the results indicate that our algorithm can detect the trajectory of a consecutive activity, trail 1, with 100% accuracy when we set the threshold of detecting frequent positions, *min_sup*=2. We also test the case with *min_sup*=3, and we find out that we can still achieve 100% accuracy. This is because there are five consecutive objects passing the RF tags along the route. The results also show that the walking speed does not affect the detection accuracy as long as the activity is frequent.

Based on the results of Experiments 1 and 2, we conclude that our method can find the trajectories of simple frequent activities with a high accuracy.

5.3. Experiment 3: busy activity

In this experiment, we test the capability of our method in detecting the trajectory of a busy activity. The same experiment setting of Experiment 1 is used here. We let one person walk back and forth on trail 1 at various speeds for one minute. Since the person may pass an RF tag many times during the one minute time period, we test the effect of *min_sup* (the threshold of frequent locations) on detection accuracy and efficiency. The results are listed in Fig. 10.

The results confirm what we discussed in Experiment 1. That is, with the increasing support threshold (*min_sup* in the figure), both the accuracy and the time cost are reduced. An interesting fact is that when *min_sup*=3, we can achieve the best accuracy with the lowest time cost. Thus, how to set a proper value of support threshold for detecting frequent locations is an interesting work, which is left for our future investigation.

5.4. Experiment 4: complex activities

In this experiment, we test the activity with complex spatial trails. The setting is shown in Fig. 11.

We ask one person to walk through the trail (the solid line with an arrow) at various speed three times. The results of detected trajectories of frequent activities are reported in Fig. 12.

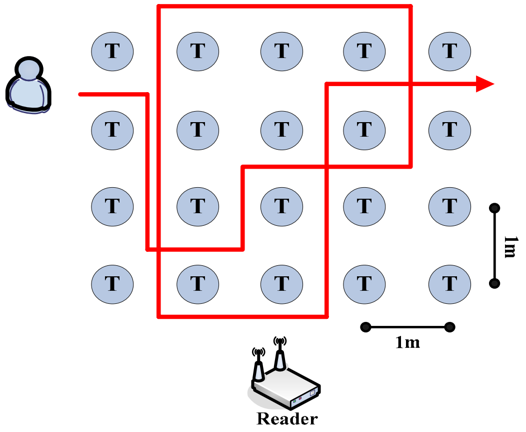


Figure 11: Setting of Experiment 4

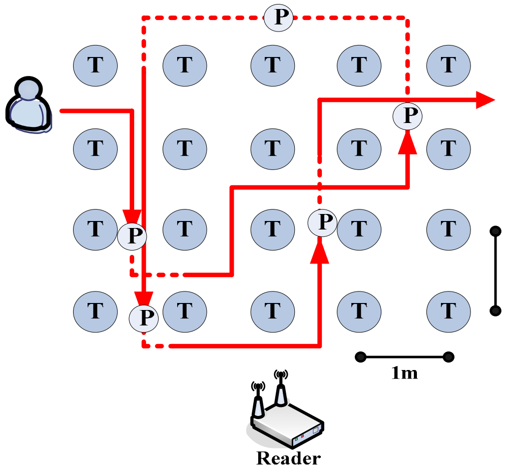


Figure 12: Detected Routes of Experiment 4

In the figure, we also plot the frequent object locations that ideally should be detected (the *P*-positions in Fig. 12). Comparing Figures 11 and 12, we can find that even for a frequent activity with a complex spatial trail, our algorithm can still detect most of the frequent trajectory segments (shown by connected solid line segments in Fig. 12).

However, our method may miss some segments. We observe that for routes outside of the RF array and the connection locations where multiple routes cross each other (shown by the dotted lines in Fig. 12), our algorithm has difficulties detecting them. However, by checking the timestamp of each possible appearance position and RF tag map, we can easily connect these separated segments into a continuous trajectory. Another possible solution for this problem is to add another RF reader at the opposite side of the current one and use cross validation to verify the results.

5.5. Summary

Our empirical study using the real RFID implementation confirms that using RF tags and readers to find trajectories of frequent activities is highly feasible. Our data mining techniques of mining fault tolerant frequent trajectories can detect frequent segments of activities. When the activities are not very complicated in space, the accuracy is high.

On the other hand, it remains a challenging task to improve the accuracy further for complex activities. We are working on using multiple readers for cross-validation as a promising solution.

6. Related Work and Discussion

Basically, there are two categories of studies highly related to our work, namely *sequential pattern mining* and *approximate frequent pattern mining*. We first review the related work and then discuss the differences between this study and the previous work.

6.1. Related work

Since it was first introduced [3], sequential pattern mining has been studied extensively. Conventional sequential pattern mining finds frequent subsequences in a sequence database based on exact match. There are two classes of algorithms. On one hand, the breadth-first search methods [10] are based on the a priori principle [2] and conduct level-by-level candidate-generation-and-tests. On the other hand, the depth-first search methods (e.g., PrefixSpan [9] and SPAM [4]) grow long patterns from short ones by constructing projected databases. Some variances of the depth-first search methods mine sequential patterns with vertical format [11]. Instead of recording sequences of items explicitly, they record item-lists, i.e., each item has a list of sequence-ids and positions where the item appears.

Recently, Guralnik and Karypis used sequential patterns as features to cluster sequential data [6]. They project the sequences onto a feature vector comprised of the sequential patterns, and then use a *k*-means like clustering method on the vector to cluster the sequential data. Approximate frequent itemset mining has also been studied [10]. Although the methods are quite different in techniques, they all explore approximate matching among itemsets.

Recently, Yang et al. presented a probabilistic model [11] to handle noise in mining strings. A compatibility matrix is introduced to represent the probabilistic connection from observed items to the underlying true items. Consequently, partial occurrence of an item is allowed and a new measure, *match*, is used to replace the commonly used support measure to represent the accumulated amount of occurrences. However, it cannot be easily generalized to apply on the sequential data targeted in this paper.

Chudova and Smyth used a Bayes error rate framework under a Markov assumption to analyze different factors that influence string pattern mining in computational biology[5]. Extending the theoretical framework to mining sequences of sets could shed more light to the future research in this direction.

6.2. How is the study different?

This study is different from the previous work in the following two aspects.

First, *trajectories are implicit in input data*. Most of the previous works assume that the input data is in the form of a set of sequences, and find the common subsequences shared by many input sequences. Although an RF tag data set is in the form of a set of time series（one time series for one tag per reader), the mining task is not to find frequent segments shared by the time series. In fact, such sharing does not make sense in this application. Instead, the trajectories are hidden across the time series and cannot be extracted as a set of sequences explicitly to feed an existing sequential pattern mining method. Therefore, the previous sequence mining methods cannot be applied.

Second, *the RF tag data is very noisy*. Most of the previous sequential pattern mining methods are not fault-tolerant. Here, the fault-tolerant mining strategies are needed to handle the noise and extract the representative patterns. Moreover, the redundant patterns should be pruned during the mining as well as in the final results. This requirement justifies the need in this paper to develop a new mining method, though the philosophy of sequential pattern mining is shared.

7. Conclusions

In this paper, we report a novel application of RFID technology: using RF tag arrays for activity monitoring. Frequent trajectory mining plays an important role in this application. We present the framework, formulate the frequent trajectory mining problem and develop a practical solution. Our empirical study using real RFID data sets verifies the effectiveness of the proposed method.

We are currently exploring the cross-validation method using multiple readers, and a more thorough test in real application fields. Moreover, it would be interesting to investigate the optimal deployment of RF tags and readers in a field. Again, data mining techniques will be very useful.

In the future, we are planning to explore more applications of RFID technology in ubiquitous computing. Since RFID applications often generate a large amount of data, we believe those applications will pose new challenges and opportunities for data mining research and development.

8. Acknowledgement

This work is supported in part by the Hong Kong RGC grant HKUST6152/06E, the HKUST Digital Life Research Center Grant, the National Basic Research Program of China (973 Program) under grant No. 2006CB303000, NSFC Key Project grant No. 60533110, the NSERC Discovery Grants, the NSERC CRD Program, and an IBM Eclipse Innovation Award.

9. References

[1] "RF Code, http://rfcode.com/ProductsFrame.asp,"

[2] R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules," in Proceedings of VLDB, 1994.

[3] R. Agrawal and R. Srikant, "Mining Sequential Patterns," in Proceedings of ICDE, 1995.

[4] J. Ayres, J. Flannick, J. Gehrke, and T. Yiu, "Sequential Pattern Mining using a Bitmap Representation," in Proceedings of ACM SIGKDD, 2002.

[5] D. Chudova and P. Smyth, "Pattern Discovery in Sequences under a Markov Assumption," in Proceedings of ACM SIGKDD, 2002.

[6] V. Guralnik and G. Karypis, "A Scalable Algorithm for Clustering Sequential Data," in Proceedings of ICDM, 2001.

[7] X. Lian, L. Chen, J. X. Yu, G. Wang, and G. Yu, "Similarity Match Over High Speed Time-Series Streams," in Proceedings of ICDE, 2007.

[8] L. M. Ni, Y. Liu, Y. C. Lau, and A. Patil, "Landmarc: Indoor Location Sensing using Active RFID," *Wireless Networks*, 2004.

[9] J. Pei, J. Han, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal, and M.-C. Hsu, "PrefixSpan: MiningSequential Patterns Efficiently by Prefix-projected Pattern Growth," in Proceedings of ICDE, 2001.

[10]J. K. Seppanen and H. Mannila, "Dense Itemsets," in Proceedings of ACM SIGKDD, 2004.

[11]J. Yang, P. S. Yu, W. Wang, and J. Han, "Mining Long Sequential Patterns in a Noisy Environment," in Proceedings of ACM SIGMOD, 2002.