A Highly Accurate Method for Managing Missing Reads in RFID Enabled Asset Tracking

Rengamathi Sankarkumar(✉), Damith Ranasinghe, and Thuraiappah Sathyan

Auto-ID Lab, The School of Computer Science, University of Adelaide, Adelaide, Australia
{rengamathi.sankarkumar,damith.ranasinghe, thuraiappah.sathyan}@adelaide.edu.au

Abstract. RFID based tracking systems have to overcome some significant challenges such as uncertainty to improve accuracy. We describe a highly accurate and scalable location tracking algorithm achieved by integrating an object compression technique with particle filtering.

Keywords: RFID · Tracking · Particle filter · Optimization · Supply-chain

1 Introduction

Radio Frequency Identification (RFID) is a promising unique identification technology widely adopted for tracking applications such as the location of goods in supply chains where the motivation behind using RFID based tracking is to improve the visibility of objects travelling though a supply chain [1]. But, RFID technology cannot be directly utilized because to obtain an accurate location tracking system, first, we must overcome uncertainty in RFID based tracking networks [2]. Uncertainty in RFID data is mainly caused by missed reads, where tagged objects exist in a readable zone but the RFID readers fail to read the tags due to factors such as interference, noise, distance between the reader and the tag [4]. The missed reads make RFID data incomplete [2] and when such noisy data is used in tracking applications the location of objects can become ambiguous.

Recent research such as [3] applied smoothing techniques to clean individual tag streams and estimate tag counts in a given location. However, these techniques are not suitable to identify anomalies such as missed reads. In [2] authors used a time varying graph model to capture packaging level information (containment relationships) and the location of an object is estimated using data inferred from the graphical model. But, this technique can only be applied for objects having a containment relationship such as cases on a pallet.

In this paper, we present a technique that addresses location uncertainty caused by missed reads in an RFID enabled returnable asset tracking application [1]. We propose a highly accurate method for real time tracking capable of estimating the most likely location of assets under uncertainty (missed reads) based

I. Stojmenovic et al. (Eds.): MOBIQUITOUS 2013, LNICST 131, pp. 683–687, 2014.
DOI: 10.1007/978-3-319-11569-6_53

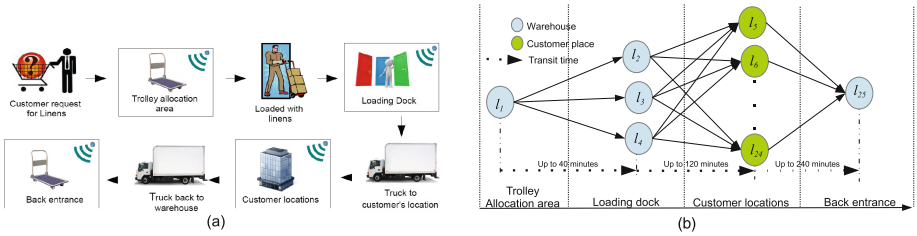

Fig. 1. (a) Business process (b) object flow model

on a sampling-based inference technique known as particle filters. Furthermore, we develop a real-time object compression technique by exploiting objects that travel together to optimize the tracking algorithm. Finally, we conduct extensive experiments and evaluate the performance of the proposed tracking algorithm. The rest of the paper is organized as follows Sect. 2 describes the problem we have considered; Sects. 3 and 4 explains the main contribution of the paper; and Sect. 5 concludes the paper.

2 Problem Statement

Before defining the problem, we define the notations used in this paper: (i) $L = \{l_p|p = 1, .., s\}$ denotes a set of locations where RFID tag reading capability is deployed; (ii) $T = \{t_i|i = 1, .., m\}$ gives a set of discrete time stamps; (iii) $O = \{o_j|j = 1, .., q\}$ is a set of tagged objects; and (iv) $transit = \{transit_{l_p}|p = 1, .., s\}$ denotes transit times between fixed locations. Then raw RFID reads from a reader can be represented by the schema {time(t_i), objectID(o_j), location(l_p)}.

Returnable asset management requirements are derived from a linen services company in South Australia. In this scenario, trolleys (objects) are the returnable assets which are attached with RFID tags to increase their visibility. The company's linen distribution scenario is depicted in Fig. 1, where RFID readers are installed in the trolley allocation area, loading docks, premises of customer locations and the back entrance of the company's warehouse. If there is a missed read in any of these locations then the location of the object is ambiguous. The particle filtering (PF) based tracking algorithm attempts to resolve this ambiguity by predicting the most likely location.

3 PF Based Tracking Algorithm with Object Compression

A particle filter can be applied to any non-linear recursive Bayesian filtering problem [4]. To define the problem of tracking, we consider the evolution of the state sequence from: (i) the **Motion Model** $x_t = f_t(x_{t-1}, v_{k-1})$, where f_t is a possible non-linear function, and v_{k-1} is i.i.d. process noise; and (ii) the **Measurement Model** $y_t = h_t(x_t, u_k)$, where h_t is a possible non-linear function,

Algorithm 1. *PF_based_tracking_with_object_compression*

Require: $raw\_reads^r = (T, O, L)$ where r = 1,2,... // (time, object ID, location) where T $=\{t_i | i = 1, .., m\}$, O $=\{o_j | j = 1, .., q\}$, L $= \{l_p | p = 1, .., 25\}$
Require: transit $= \{transit_k | k = 1, .., 25\}$, δt
$t \leftarrow r.t_1$// get the initial time from raw_reads
while $\forall$ r $\in raw\_reads$ **do**
  if $r.t_i >= t$ **and** $r.t_i <= t + \delta t$ **then**
    if new_object_found **then**
      $object\_list.add(r.t_i, r.o_j, r.l_p, probability \leftarrow 1, new\_object \leftarrow true)$
    else
      $object\_list.update(r, probability, new\_object \leftarrow false)$
    end if
  else
    while $\forall o_j \in object\_list$ **do**
      $group\_object^n \leftarrow$ grouped objects by location for time window [t and $t+\delta t$]
    end while
    for k = 1 **to** n **do**
      $hashtable.put(ID^k, group\_object^k)$
      $group \leftarrow$*(t, ID^k, location of k^{th}group_object)*
      if $group\_object^k$ contains $new\_object(true)$ **then**
        initialize group
      end if
      predict, update, normalize and resample group
      $object^d \leftarrow hashtable.get(ID^k)$ //get back the compressed objects
      $object\_list.update(\forall d \in object, probability)$
    end for
    $t \leftarrow r.t_i$ // next t from r
  end if
end while
current_time $\leftarrow$ current time from the system
while not_end_of($object\_list$) **do**
  $obj \leftarrow object\_list.get\_next$
  $max\_time \leftarrow obj.t_i + transit_{(obj.l_p)}$
  if $current\_time > max\_time$ **then**
    $pred\_loc \leftarrow$ the most likely location; $pred\_time \leftarrow obj.t_i + (transit_{(obj.l_p)})/2$
    $probability \leftarrow$estimated probability of o_j
    $obj.t_i \leftarrow pred\_time$, $obj.l_p \leftarrow pred\_loc$
    object_list.update(obj)
  end if
end while

and u_k is i.i.d. measurement noise. The PF consist of five steps: initialization, prediction, update, normalization and resampling [4].

The PF based tracking algorithm gives the most probable location l_p of the object o_j with N state particles, at every timestep t_i, as a probability distribution over l_p possible locations. The state space is defined along the x-axis. Every reader involved is represented as a sensor location along the x-axis.

The motion model is constructed with a transition matrix M. **Transition Matrix:** The transition probability is defined with a conditional probability, $p(L_{t_i}|L_{t_{i-1}})$ between locations L_{t_i} and $L_{t_{i-1}}$, where the current time step is t_i and previous time step is t_{i-1}. If L has p locations, then M can be represented by a $p \times p$ matrix. **Measurement Model:** The measurement model used to update particles is drawn from a Gaussian distribution, because it models the decreasing probability of a tag being read as tags moves further from a reader.

Even though PF can improve location tracking accuracy, it suffers from high computational cost when implemented in large scale tracking applications involving large number of objects. To overcome the computational cost, we optimize the particle filtering based tracking algorithm by compressing the object stream reported by readers. The real-time object compression technique exploits objects that travel together from one location to another (not containment relationships [3]). Our approach is based on compressing a group of trolleys (objects) that are travelling together within a time window to a single aggregated object as shown in Algorithm 1. This compression can be achieved in three steps: (i) create a dynamic time window δt and collect objects within that time window; (ii) group collected objects based on their location; and (iii) compress the group of trolleys to a single aggregated object. A hash table data structure is used to maintain a mapping between a grouped object and the trolleys contained within a group. A hash table also supports maintaining aggregation changes and disaggregation of trolleys from grouped objects. Then PF based tracking algorithm is applied to the aggregated object. The benefit of compressing the object stream is to minimize the need for using the PF based tracking algorithm for individual objects and to, consequently, achieve both greater scalability and accuracy.

In Algorithm 1, line 3–15 explains the object compression, resulting in one compressed trolley for each group. Line 16–24 explains the PF applied to the compressed trolley. Then, the location l_p with the highest probability is estimated as the most likely location. Line 27–36 checks for possible missed reads for objects currently being tracked by comparing the *current_time* with the *max_time* (t_i + *transit_time*).

4 Experiments and Results

We implemented the linen company's business process in Matlab (Fig. 1) and conducted extensive experiments to evaluate the performance of our tracking algorithm. In the simulation, the time duration from trolley allocation to its return to the back entrance is 2–6 h, read rate is defined as the probability that a tag is read successfully by a reader and the error rate is specified as the percentage of incorrect location predictions. The results reported were generated by averaging 10 repeated experimental runs using randomly generated trolley movement data.

Figure 2 concretely depicts that the object compression technique based PF outperforms the simple PF based tracking algorithm in terms of both accuracy and scalability. When the read rate is above 0.8, the error rate in object compression technique is below 5 %, which is a considerable improvement in accuracy

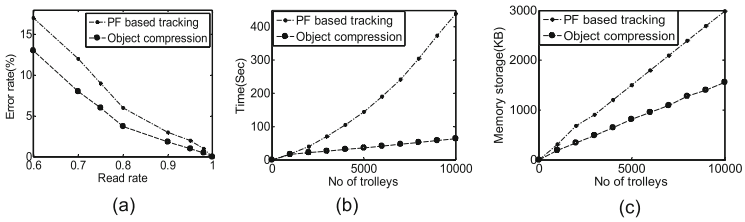

Fig. 2. Experimental results: (a) accuracy; (b) execution time; (c) memory usage

compared to PF based tracking algorithm. Figures 2(b) and (c) show the execution time and memory usage of the proposed algorithm, which clearly shows that the object compression technique significantly reduces the time taken and memory usage of the system.

5 Conclusion

In this paper, we presented an object location tracking algorithm that is robust under uncertainty and demonstrated its performance in an RFID enabled returnable asset management scenario. In practical deployments, the miss rate (1-read rate) is around 5 % and at this miss rate our system reduces the overall error rate to less than 2 % which is a significant improvement in performance. Comparing our algorithm with [2], where data compression is based on containment relationships, when read rate is 80 % and containment is 100 % their accuracy is around 94 % but with 0 % containment (similar to our scenario) the system accuracy falls to below 87 %. In contrast our tracking algorithm yields 96 % accuracy with a read rate of 80 % and outperforms the method in [2]. Furthermore, our technique can also be applied to contained objects, as in [2].

Acknowledgement. This work has been supported by ARC-Linkage grant LP100200114.

References

1. Wu, Y., Sheng, Q.Z., Ranasinghe, D., Yao, L.: PeerTrack: a platform for tracking and tracing objects in large-scale traceability networks. In: Proceedings of the 15th International Conference on EDBT, Berlin, Germany, pp. 586–589 (2012)
2. Nie, Y., Cocci, R., Cao, Z., Diao, Y., Shenoy, P.: Spire: efficient data inference and compression over RFID streams. IEEE TKDE **24**(1), 141–155 (2012)
3. Jeffery, S.R., Garofalakis, M., Franklin, M.: Adaptive cleaning for RFID data streams. In: Proceedings of the 32nd International Conference on VLDB Seol, Korea, pp. 163–174 (2006)
4. Yu, J., Ku, W.S., Sun, M.T., Lu, H.: An RFID and particle filter-based indoor spatial query evaluation system. In: Proceedings of the 16th International Conference on EDBT, pp. 263–274 (2013)