Manage the Data from Indoor Spaces: Models, Indexes & Query Processing

Huan Li

Database Laboratory, Zhejiang University

lihuancs@zju.edu.cn

March 22, 2016

Overview

About This Work...

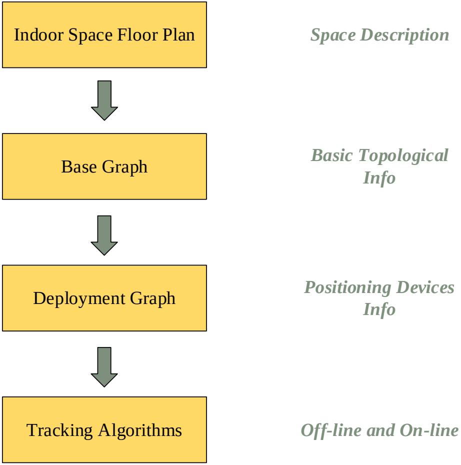
Graph Model Based Indoor Tracking. [1]
C. S. Jensen, H. Lu, and B. Yang.

- Published in year 2009, *MDM* conference.
- A pioneering work that introduces base graph model to indoor data management.
- Detailed tracking algorithms are designed for RFID-based positioning.
- Easy to understand, with comprehensive concepts.

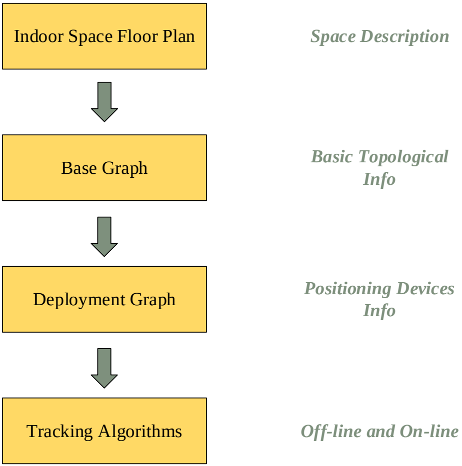
Motivation

- We are spending most of our time in indoor spaces
  - Office building, University, Shopping Centers, etc.
- We cannot use GPS-based tracking indoor movements
  - Indoor navigation and route guidance (museum)
  - Flow analysis
    - how do people use the indoor space $\rightarrow$ important in pricing of advertisement space in store rental
- We can use other technology...
  - Wi-Fi, Infrared, Bluetooth or RFID
  - This paper is focusing on RFID, since it is now mature and effortless
  - RFID tags are cheap and RFID reader are expensive

Idea


Indoor Space Floor Plan
Space Description
Base Graph
Basic Topological Info
Deployment Graph
Positioning Devices Info
Tracking Algorithms
Off-line and On-line

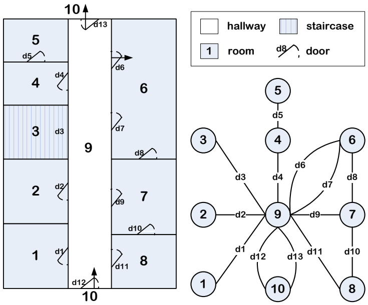
Idea



Goal: Improve indoor tracking accuracy from a data management perspective, to capture where a particular object can be at a particular time.

Base Graph Model

By capturing the essential connectivity and accessibility, **Base Graph** describes the topology of a floor plan of a possibly complex indoor space.



(a) Floor Plan (b) Connectivity Base Graph

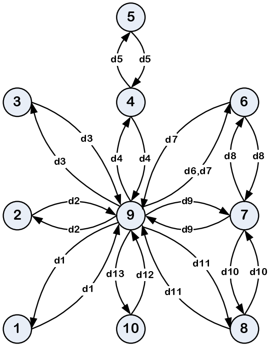
Connectivity Base Graph

a labeled, undirected graph.

- $G_{conn} = \{V, E_d, \Sigma_{door}\}$
- V: each separate partition is represented as a vertex
- E_d: each door is captured as an edge
- Σ_{door}: a set of edge labels that represent connections

Base Graph Model

Accessibility Graph is constructed to represent the movement permitted by doors or connections.



Accessibility Graph

a labeled, directed graph.

- $G_{accs} = \{V, E, \Sigma_{door}, l_e\}$
- V: the set of vertices
- E: the set of directed edges, i.e., $E = \{\langle v_i, v_j \rangle | v_i, v_j \in V \wedge v_i \neq v_j\}$
- l_e: a function that maps edges to subsets of the set of doors, i.e., $l_e : E \rightarrow 2^{\Sigma_{door}}$

Base Graph Model

In addition to the topological information of a floor plan, its geometrical information should also be captured.

Base Graph Model

In addition to the topological information of a floor plan, its geometrical information should also be captured.

The *Building Partitions Mapping* is defined as:

Base Graph Model

In addition to the topological information of a floor plan, its geometrical information should also be captured.

The $Building\ Partitions\ Mapping$ is defined as:

$$BuildingPartitions : V \rightarrow Ploygons \tag{1}$$

Base Graph Model

In addition to the topological information of a floor plan, its geometrical information should also be captured.

The *Building Partitions Mapping* is defined as:

$$BuildingPartitions : V \rightarrow Ploygons \tag{1}$$

The *Doors Mapping* is defined as:

Base Graph Model

In addition to the topological information of a floor plan, its geometrical information should also be captured.

The *Building Partitions Mapping* is defined as:

$$BuildingPartitions : V \rightarrow Ploygons \tag{1}$$

The *Doors Mapping* is defined as:

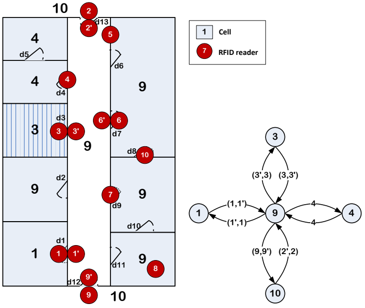
$$Doors : \Sigma_{door} \rightarrow Line\ Segments \tag{2}$$

RFID Deployment Graph Model

- RFID based proximity analysis
  - RFID readers deployment may cover only part of the space, or it may be capable of only detecting some movements in the space.
  - assume that all RFID readers have disjoint activation ranges.
- Types of RFID readers
  - **Partitioning Readers** partition the indoor space into cells in the sense that an object cannot move from one cell to another without being observed.
  - **Presence Readers** simply observe the presence(and non-presence) of tags in their activation ranges.

RFID Deployment Graph Model

Vertices represent cells. A directed edge indicates that one can move from one cell to another without entering other cells, which is detected by a corresponding partitioning reader.



(a) Readers Deployment (b) Readers Deployment Graph

RFID Deployment Graph

a labeled, directed graph.

- $G_{RFID} = \{C, E_r, \Sigma_{reader}, l_e\}$
- C: the set of the vertices
- E_r: An edge is an ordered pair $\langle c_i, c_j \rangle$ of distinct vertices from C
- l_e maps an edge to a partitioning reader (pair), i.e., $E_r \rightarrow 2^{\Sigma_{reader}} \cup 2^{\Sigma_{reader} \times \Sigma_{reader}}$

RFID Deployment Graph Construction

Each cell created by partitioning readers corresponds to one or more base graph partitions.

RFID Deployment Graph Construction

Each cell created by partitioning readers corresponds to one or more base graph partitions.

$$Cells : V \rightarrow C \tag{3}$$

RFID Deployment Graph Construction

Each cell created by partitioning readers corresponds to one or more base graph partitions.

$$Cells : V \rightarrow C \tag{3}$$

For each RFID reader, record its accurate deployment location and activation range.

RFID Deployment Graph Construction

Each cell created by partitioning readers corresponds to one or more base graph partitions.

$$Cells : V \rightarrow C \tag{3}$$

For each RFID reader, record its accurate deployment location and activation range.

$$\begin{aligned} Mapping\ 1 :& \Sigma_{reader} \rightarrow \{(loc, range, flag) \mid loc \in R^2 \wedge \\ & range \in (0, d_{max}] \wedge flag \in \{PAR, PRE\}\} \end{aligned} \tag{4}$$

RFID Deployment Graph Construction

Each cell created by partitioning readers corresponds to one or more base graph partitions.

$$Cells : V \to C \tag{3}$$

For each RFID reader, record its accurate deployment location and activation range.

$$\begin{aligned} Mapping\ 1 : &\Sigma_{reader} \to \{(loc, range, flag) \mid loc \in R^2 \wedge \\ &range \in (0, d_{max}] \wedge flag \in \{PAR, PRE\}\} \end{aligned} \tag{4}$$

A mapping of readers to the cells that their activation ranges intersect is introduced as:

RFID Deployment Graph Construction

Each cell created by partitioning readers corresponds to one or more base graph partitions.

$$Cells : V \rightarrow C \tag{3}$$

For each RFID reader, record its accurate deployment location and activation range.

$$\begin{aligned} Mapping\ 1 : &\Sigma_{reader} \rightarrow \{(loc, range, flag) \mid loc \in R^2 \wedge \\ & range \in (0, d_{max}] \wedge flag \in \{PAR, PRE\}\} \end{aligned} \tag{4}$$

A mapping of readers to the cells that their activation ranges intersect is introduced as:

$$Mapping\ 2 : \Sigma_{reader} \rightarrow 2^C \tag{5}$$

RFID Deployment Graph Construction

Algorithm 1 RFIDGraphConstruction (Readers R, ConnectivityBaseGraph G_{conn}, AccessibilityGraph G_{accs})

Readers $R' \leftarrow \emptyset$; $DR(\langle \Sigma_{door}\ k', Readers\ RSet \rangle) \leftarrow \emptyset$; Connected Component $CCs \leftarrow \emptyset$; int $m \leftarrow 0$;
$G_{RFID}(C, E_r, \Sigma_{reader}\ l_e) \leftarrow (\emptyset, \emptyset, R, NULL)$;
for each edge d_a in $G_{conn}.E_d$ **do**
    for each reader r_b in R **do**
        if Circle(Mapping1(r_b).loc, Mapping1(r_b).range) covers $Doors(d_a.k)$ **then**
            insert r_b into R';
    if $|R'| > 0$ **then**
        delete d_a from $G_{conn}.E_d$; insert $(d_a.k, R')$ into DR; $R' \leftarrow \emptyset$;
store all connected components of G_{conn} in CCs;
for each connected component cc_c in CCs **do**
    create a new vertex c_m and add it to $G_{RFID}.C$; m++;
    for each vertex v_x in the vertices of cc_c **do**
        add the mapping $(v_x \rightarrow c_m)$ to $Cells$;
for each dr_n in DR **do**
    for each e_l in $G_{accs}.l_e^{-1}(dr_n.k')$ **do**
        $c_p \leftarrow Cells(e_l.v_i)$; $c_q \leftarrow Cells(e_l.v_j)$;
        if $c_p \neq c_q$ **then**
            if $\langle c_p, c_q \rangle$ is not in $G_{RFID}.E_r$ **then**
                add $\langle c_p, c_q \rangle$ to $G_{RFID}.E_r$;
            add the mapping $(\langle c_p, c_q \rangle \rightarrow readersequence(dr_n.RSet))$ to $G_{RFID}.l_e$;

RFID Deployment Graph Construction

Algorithm 1 RFIDGraphConstruction (Readers R, ConnectivityBaseGraph G_{conn}, AccessibilityGraph G_{accs})

Readers $R' \leftarrow \emptyset$; $DR(\langle \Sigma_{door}\ k', Readers\ RSet \rangle) \leftarrow \emptyset$; Connected Component $CCs \leftarrow \emptyset$; int $m \leftarrow 0$;
$G_{RFID}(C, E_r, \Sigma_{reader}\ l_e) \leftarrow (\emptyset, \emptyset, R, NULL)$;
for each edge d_a in $G_{conn}.E_d$ **do**
  for each reader r_b in R **do**
    if Circle(Mapping1(r_b).loc, Mapping1(r_b).range) covers $Doors(d_a.k)$ **then**
      insert r_b into R';
  if $|R'| > 0$ **then**
    delete d_a from $G_{conn}.E_d$; insert $(d_a.k, R')$ into DR; $R' \leftarrow \emptyset$;
store all connected components of G_{conn} in CCs;
for each connected component cc_c in CCs **do**
  create a new vertex c_m and add it to $G_{RFID}.C$; m++;
  for each vertex v_x in the vertices of cc_c **do**
    add the mapping $(v_x \rightarrow c_m)$ to $Cells$;
for each dr_n in DR **do**
  for each e_l in $G_{accs}.l_e^{-1}(dr_n.k')$ **do**
    $c_p \leftarrow Cells(e_l.v_i)$; $c_q \leftarrow Cells(e_l.v_j)$;
    if $c_p \neq c_q$ **then**
      if $\langle c_p, c_q \rangle$ is not in $G_{RFID}.E_r$ **then**
        add $\langle c_p, c_q \rangle$ to $G_{RFID}.E_r$;
      add the mapping $(\langle c_p, c_q \rangle \rightarrow readersequence(dr_n.RSet))$ to $G_{RFID}.l_e$;

1. **Input:** the reader set R, the connectivity base graph G_{conn}, the accessibility graph G_{accs}

RFID Deployment Graph Construction

Algorithm 1 RFIDGraphConstruction (Readers R, ConnectivityBaseGraph G_{conn}, AccessibilityGraph G_{accs})

1: Readers $R' \leftarrow \emptyset$; $DR(\langle \Sigma_{door}\ k', Readers\ RSet \rangle) \leftarrow \emptyset$; Connected Component $CCs \leftarrow \emptyset$; int $m \leftarrow 0$;
2: $G_{RFID}(C, E_r, \Sigma_{reader}\ l_e) \leftarrow (\emptyset, \emptyset, R, NULL)$;
3: **for** each edge d_a in $G_{conn}.E_d$ **do**
4:   **for** each reader r_b in R **do**
5:     **if** Circle(Mapping1(r_b).loc, Mapping1(r_b).range) covers $Doors(d_a.k)$ **then**
6:       insert r_b into R';
7:   **if** $|R'| > 0$ **then**
8:     delete d_a from $G_{conn}.E_d$; insert $(d_a.k, R')$ into DR; $R' \leftarrow \emptyset$;
9: store all connected components of G_{conn} in CCs;
10: **for** each connected component cc_c in CCs **do**
11:   create a new vertex c_m and add it to $G_{RFID}.C$; $m{+}{+}$;
12:   **for** each vertex v_x in the vertices of cc_c **do**
13:     add the mapping $(v_x \rightarrow c_m)$ to $Cells$;
14: **for** each dr_n in DR **do**
15:   **for** each e_l in $G_{accs}.l_e^{-1}(dr_n.k')$ **do**
16:     $c_p \leftarrow Cells(e_l.v_i)$; $c_q \leftarrow Cells(e_l.v_j)$;
17:     **if** $c_p \neq c_q$ **then**
18:       **if** $\langle c_p, c_q \rangle$ is not in $G_{RFID}.E_r$ **then**
19:         add $\langle c_p, c_q \rangle$ to $G_{RFID}.E_r$;
20:       add the mapping $(\langle c_p, c_q \rangle \rightarrow readersequence(dr_n.RSet))$ to $G_{RFID}.l_e$;

1. Input: the reader set R, the connectivity base graph G_{conn}, the accessibility graph G_{accs}
2. Lines 1–2: Initialize G_{RFID}, DR, CCs

RFID Deployment Graph Construction

Algorithm 1 RFIDGraphConstruction (Readers R, ConnectivityBaseGraph G_{conn}, AccessibilityGraph G_{accs})

1: Readers $R' \leftarrow \emptyset$; $DR(\langle \Sigma_{door}\ k', Readers\ RSet\rangle) \leftarrow \emptyset$; Connected Component $CCs \leftarrow \emptyset$; int $m \leftarrow 0$;
2: $G_{RFID}(C, E_r, \Sigma_{reader}\ l_e) \leftarrow (\emptyset, \emptyset, R, NULL)$;
3: **for** each edge d_a in $G_{conn}.E_d$ **do**
4:   **for** each reader r_b in R **do**
5:     **if** Circle(Mapping1(r_b).loc, Mapping1(r_b).range) covers $Doors(d_a.k)$ **then**
6:       insert r_b into R';
7:   **if** $|R'| > 0$ **then**
8:     delete d_a from $G_{conn}.E_d$; insert $(d_a.k, R')$ into DR; $R' \leftarrow \emptyset$;
9: store all connected components of G_{conn} in CCs;
10: **for** each connected component cc_c in CCs **do**
11:   create a new vertex c_m and add it to $G_{RFID}.C$; m++;
12:   **for** each vertex v_x in the vertices of cc_c **do**
13:     add the mapping $(v_x \rightarrow c_m)$ to $Cells$;
14: **for** each dr_n in DR **do**
15:   **for** each e_l in $G_{accs}.l_e^{-1}(dr_n.k')$ **do**
16:     $c_p \leftarrow Cells(e_l.v_i)$; $c_q \leftarrow Cells(e_l.v_j)$;
17:     **if** $c_p \neq c_q$ **then**
18:       **if** $\langle c_p, c_q\rangle$ is not in $G_{RFID}.E_r$ **then**
19:         add $\langle c_p, c_q\rangle$ to $G_{RFID}.E_r$;
20:       add the mapping $(\langle c_p, c_q\rangle \rightarrow readersequence(dr_n.RSet))$ to $G_{RFID}.l_e$;

1. **Input:** the reader set R, the connectivity base graph G_{conn}, the accessibility graph G_{accs}
2. **Lines 1–2:** Initialize G_{RFID}, DR, CCs
3. **Lines 3–8:** the relationship of which door is covered by which readers is captured in DR

RFID Deployment Graph Construction

Algorithm 1 RFIDGraphConstruction (Readers R, ConnectivityBaseGraph G_{conn}, AccessibilityGraph G_{accs})

1: Readers $R' \leftarrow \emptyset$; $DR(\langle \Sigma_{door}\ k', Readers\ RSet \rangle) \leftarrow \emptyset$; Connected Component $CCs \leftarrow \emptyset$; int $m \leftarrow 0$;
2: $G_{RFID}(C, E_r, \Sigma_{reader}\ l_e) \leftarrow (\emptyset, \emptyset, R, NULL)$;
3: **for** each edge d_a in $G_{conn}.E_d$ **do**
4:     **for** each reader r_b in R **do**
5:         **if** Circle(Mapping1(r_b).loc, Mapping1(r_b).range) covers $Doors(d_a.k)$ **then**
6:             insert r_b into R';
7:     **if** $|R'| > 0$ **then**
8:         delete d_a from $G_{conn}.E_d$; insert $(d_a.k, R')$ into DR; $R' \leftarrow \emptyset$;
9: store all connected components of G_{conn} in CCs;
10: **for** each connected component cc_c in CCs **do**
11:     create a new vertex c_m and add it to $G_{RFID}.C$; m++;
12:     **for** each vertex v_x in the vertices of cc_c **do**
13:         add the mapping $(v_x \rightarrow c_m)$ to $Cells$;
14: **for** each dr_n in DR **do**
15:     **for** each e_l in $G_{accs}.l_e^{-1}(dr_n.k')$ **do**
16:         $c_p \leftarrow Cells(e_l.v_i)$; $c_q \leftarrow Cells(e_l.v_j)$;
17:         **if** $c_p \neq c_q$ **then**
18:             **if** $\langle c_p, c_q \rangle$ is not in $G_{RFID}.E_r$ **then**
19:                 add $\langle c_p, c_q \rangle$ to $G_{RFID}.E_r$;
20:             add the mapping $(\langle c_p, c_q \rangle \rightarrow readersequence(dr_n.RSet))$ to $G_{RFID}.l_e$;

1. Input: the reader set R, the connectivity base graph G_{conn}, the accessibility graph G_{accs}
2. Lines 1–2: Initialize G_{RFID}, DR, CCs
3. Lines 3–8: the relationship of which door is covered by which readers is captured in DR
4. Lines 9–13: a deployment graph vertex is created for each CC, mapping $Cells$ is also stored

RFID Deployment Graph Construction

Algorithm 1 RFIDGraphConstruction (Readers R, ConnectivityBaseGraph G_{conn}, AccessibilityGraph G_{accs})

1: Readers $R' \leftarrow \emptyset$; $DR(\langle \Sigma_{door}\ k', Readers\ RSet\rangle) \leftarrow \emptyset$; Connected Component $CCs \leftarrow \emptyset$; int $m \leftarrow 0$;
2: $G_{RFID}(C, E_r, \Sigma_{reader}\ l_e) \leftarrow (\emptyset, \emptyset, R, NULL)$;
3: **for** each edge d_a in $G_{conn}.E_d$ **do**
4:     **for** each reader r_b in R **do**
5:         **if** Circle(Mapping1(r_b).loc, Mapping1(r_b).range) covers $Doors(d_a.k)$ **then**
6:             insert r_b into R';
7:     **if** $|R'| > 0$ **then**
8:         delete d_a from $G_{conn}.E_d$; insert $(d_a.k, R')$ into DR; $R' \leftarrow \emptyset$;
9: store all connected components of G_{conn} in CCs;
10: **for** each connected component cc_c in CCs **do**
11:     create a new vertex c_m and add it to $G_{RFID}.C$; m++;
12:     **for** each vertex v_x in the vertices of cc_c **do**
13:         add the mapping $(v_x \rightarrow c_m)$ to $Cells$;
14: **for** each dr_n in DR **do**
15:     **for** each e_l in $G_{accs}.l_e^{-1}(dr_n.k')$ **do**
16:         $c_p \leftarrow Cells(e_l.v_i)$; $c_q \leftarrow Cells(e_l.v_j)$;
17:         **if** $c_p \neq c_q$ **then**
18:             **if** $\langle c_p, c_q\rangle$ is not in $G_{RFID}.E_r$ **then**
19:                 add $\langle c_p, c_q\rangle$ to $G_{RFID}.E_r$;
20:             add the mapping $(\langle c_p, c_q\rangle \rightarrow readersequence(dr_n.RSet))$ to $G_{RFID}.l_e$;

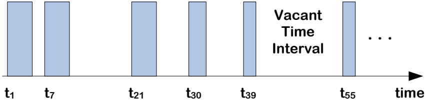
1. Input: the reader set R, the connectivity base graph G_{conn}, the accessibility graph G_{accs}
2. Lines 1–2: Initialize G_{RFID}, DR, CCs
3. Lines 3–8: the relationship of which door is covered by which readers is captured in DR
4. Lines 9–13: a deployment graph vertex is created for each CC, mapping $Cells$ is also stored
5. Lines 14–20: for each door in DR, determine if its edges' head and tail are mapped to different cells. If so, add an edge to deployment graph. Function $readersequence$ returns the possible reader sequence for that edge

RFID-based Indoor Tracking

Raw Trajectories: Sequences of RFID Tag Observation

Table 1. Raw Trajectory of tag_1

readerID	tagID	t	readerID	tagID	t
$reader_9$	tag_1	t_1	$reader_{10}$	tag_1	t_{40}
$reader_9$	tag_1	t_2	$reader_8$	tag_1	t_{55}
$reader_{9'}$	tag_1	t_3	$reader_8$	tag_1	t_{56}
$reader_{9'}$	tag_1	t_4	$reader_{3'}$	tag_1	t_{79}
$reader_{1'}$	tag_1	t_7	$reader_{3'}$	tag_1	t_{80}
$reader_{1'}$	tag_1	t_8	$reader_4$	tag_1	t_{85}
$reader_1$	tag_1	t_9	$reader_4$	tag_1	t_{86}
$reader_1$	tag_1	t_{10}	$reader_4$	tag_1	t_{87}
$reader_1$	tag_1	t_{21}	$reader_4$	tag_1	t_{100}
$reader_1$	tag_1	t_{22}	$reader_4$	tag_1	t_{101}
$reader_{1'}$	tag_1	t_{23}	$reader_4$	tag_1	t_{102}
$reader_{1'}$	tag_1	t_{24}	$reader_{6'}$	tag_1	t_{109}
$reader_7$	tag_1	t_{30}	$reader_{6'}$	tag_1	t_{110}
$reader_7$	tag_1	t_{31}	$reader_6$	tag_1	t_{111}
$reader_7$	tag_1	t_{32}	$reader_6$	tag_1	t_{112}
$reader_{10}$	tag_1	t_{39}	$reader_5$	tag_1	t_{125}



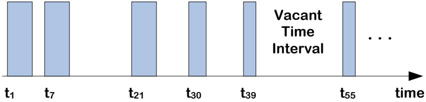
1. each reader detects and reports tags with a sampling rate
2. formatted as $\langle readerID, tagID, t\rangle$

RFID-based Indoor Tracking

Raw Trajectories: Sequences of RFID Tag Observation

Table 1. Raw Trajectory of tag_1

readerID	tagID	t	readerID	tagID	t
$reader_9$	tag_1	t_1	$reader_{10}$	tag_1	t_{40}
$reader_9$	tag_1	t_2	$reader_8$	tag_1	t_{55}
$reader_{9'}$	tag_1	t_3	$reader_8$	tag_1	t_{56}
$reader_{9'}$	tag_1	t_4	$reader_{3'}$	tag_1	t_{79}
$reader_{1'}$	tag_1	t_7	$reader_{3'}$	tag_1	t_{80}
$reader_{1'}$	tag_1	t_8	$reader_4$	tag_1	t_{85}
$reader_1$	tag_1	t_9	$reader_4$	tag_1	t_{86}
$reader_1$	tag_1	t_{10}	$reader_4$	tag_1	t_{87}
$reader_1$	tag_1	t_{21}	$reader_4$	tag_1	t_{100}
$reader_1$	tag_1	t_{22}	$reader_4$	tag_1	t_{101}
$reader_{1'}$	tag_1	t_{23}	$reader_4$	tag_1	t_{102}
$reader_{1'}$	tag_1	t_{24}	$reader_{6'}$	tag_1	t_{109}
$reader_7$	tag_1	t_{30}	$reader_{6'}$	tag_1	t_{110}
$reader_7$	tag_1	t_{31}	$reader_6$	tag_1	t_{111}
$reader_7$	tag_1	t_{32}	$reader_6$	tag_1	t_{112}
$reader_{10}$	tag_1	t_{39}	$reader_5$	tag_1	t_{125}



- *vacant time intervals*: unable to observe the moving objects

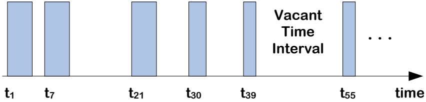
1. each reader detects and reports tags with a sampling rate
2. formatted as $\langle readerID, tagID, t\rangle$

RFID-based Indoor Tracking

Raw Trajectories: Sequences of RFID Tag Observation

Table 1. Raw Trajectory of tag_1

readerID	tagID	t	readerID	tagID	t
$reader_9$	tag_1	t_1	$reader_{10}$	tag_1	t_{40}
$reader_9$	tag_1	t_2	$reader_8$	tag_1	t_{55}
$reader_{9'}$	tag_1	t_3	$reader_8$	tag_1	t_{56}
$reader_{9'}$	tag_1	t_4	$reader_{3'}$	tag_1	t_{79}
$reader_{1'}$	tag_1	t_7	$reader_{3'}$	tag_1	t_{80}
$reader_{1'}$	tag_1	t_8	$reader_4$	tag_1	t_{85}
$reader_1$	tag_1	t_9	$reader_4$	tag_1	t_{86}
$reader_1$	tag_1	t_{10}	$reader_4$	tag_1	t_{87}
$reader_1$	tag_1	t_{21}	$reader_4$	tag_1	t_{100}
$reader_1$	tag_1	t_{22}	$reader_4$	tag_1	t_{101}
$reader_{1'}$	tag_1	t_{23}	$reader_4$	tag_1	t_{102}
$reader_{1'}$	tag_1	t_{24}	$reader_{6'}$	tag_1	t_{109}
$reader_7$	tag_1	t_{30}	$reader_{6'}$	tag_1	t_{110}
$reader_7$	tag_1	t_{31}	$reader_6$	tag_1	t_{111}
$reader_7$	tag_1	t_{32}	$reader_6$	tag_1	t_{112}
$reader_{10}$	tag_1	t_{39}	$reader_5$	tag_1	t_{125}



- *vacant time intervals*: unable to observe the moving objects
- to search RFID deployment graph to infer the possible regions of moving object

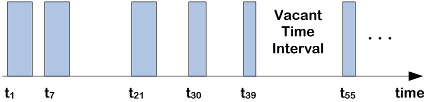
1. each reader detects and reports tags with a sampling rate
2. formatted as $\langle readerID, tagID, t \rangle$

RFID-based Indoor Tracking

Raw Trajectories: Sequences of RFID Tag Observation

Table 1. Raw Trajectory of tag_1

readerID	tagID	t	readerID	tagID	t
$reader_9$	tag_1	t_1	$reader_{10}$	tag_1	t_{40}
$reader_9$	tag_1	t_2	$reader_8$	tag_1	t_{55}
$reader_{9'}$	tag_1	t_3	$reader_8$	tag_1	t_{56}
$reader_{9'}$	tag_1	t_4	$reader_{3'}$	tag_1	t_{79}
$reader_{1'}$	tag_1	t_7	$reader_{3'}$	tag_1	t_{80}
$reader_{1'}$	tag_1	t_8	$reader_4$	tag_1	t_{85}
$reader_1$	tag_1	t_9	$reader_4$	tag_1	t_{86}
$reader_1$	tag_1	t_{10}	$reader_4$	tag_1	t_{87}
$reader_1$	tag_1	t_{21}	$reader_4$	tag_1	t_{100}
$reader_1$	tag_1	t_{22}	$reader_4$	tag_1	t_{101}
$reader_{1'}$	tag_1	t_{23}	$reader_4$	tag_1	t_{102}
$reader_{1'}$	tag_1	t_{24}	$reader_{6'}$	tag_1	t_{109}
$reader_7$	tag_1	t_{30}	$reader_{6'}$	tag_1	t_{110}
$reader_7$	tag_1	t_{31}	$reader_6$	tag_1	t_{111}
$reader_7$	tag_1	t_{32}	$reader_6$	tag_1	t_{112}
$reader_{10}$	tag_1	t_{39}	$reader_5$	tag_1	t_{125}

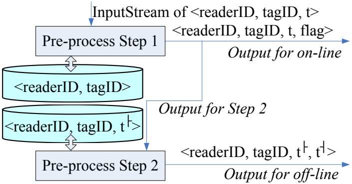


- *vacant time intervals*: unable to observe the moving objects
- to search RFID deployment graph to infer the possible regions of moving object
- to apply maximum speed position interpolation to further shrink the possible regions [2, 3]

1. each reader detects and reports tags with a sampling rate
2. formatted as $\langle readerID, tagID, t\rangle$

RFID Readings Pre-processing

Step-1's output is used in on-line tracking, while Step-2's is used in off-line tracking



- $Flag \in \{START, END\}$
- $START \rightarrow$ enters the range
- $END \rightarrow$ leaves the range

Off-line Tracking (Refinement Step 1)

Table 2. Off-Line Trajecotry of tag_1

readingID	readerID	tagID	$t^{\vdash}$	$t^{\dashv}$
$reading_1$	$reader_9$	tag_1	t_1	t_2
$reading_2$	$reader_{9'}$	tag_1	t_3	t_4
$reading_3$	$reader_{1'}$	tag_1	t_7	t_8
$reading_4$	$reader_1$	tag_1	t_9	t_{10}
$reading_5$	$reader_1$	tag_1	t_{21}	t_{22}
$reading_6$	$reader_{1'}$	tag_1	t_{23}	t_{24}
$reading_7$	$reader_7$	tag_1	t_{30}	t_{32}
$reading_8$	$reader_{10}$	tag_1	t_{39}	t_{40}
$reading_9$	$reader_8$	tag_1	t_{55}	t_{56}
$reading_{10}$	$reader_{3'}$	tag_1	t_{79}	t_{80}
$reading_{11}$	$reader_4$	tag_1	t_{85}	t_{87}
$reading_{12}$	$reader_4$	tag_1	t_{100}	t_{102}
$reading_{13}$	$reader_{6'}$	tag_1	t_{109}	t_{110}
$reading_{14}$	$reader_6$	tag_1	t_{111}	t_{112}
$reading_{15}$	$reader_5$	tag_1	t_{125}	t_{125}

Table 3. Trajectory After Step 1

tagID	$[t^{\vdash}, t^{\dashv}]$	Graph Element	Reader_1	Reader_2
tag_1	$[t_1, t_4]$	$e\langle 10, 9\rangle$	$reader_9$	$reader_{9'}$
tag_1	$[t_7, t_{10}]$	$e\langle 9, 1\rangle$	$reader_{1'}$	$reader_1$
tag_1	$[t_{21}, t_{24}]$	$e\langle 1, 9\rangle$	$reader_1$	$reader_{1'}$
tag_1	$[t_{30}, t_{32}]$	c_9	$reader_7$	$reader_7$
tag_1	$[t_{39}, t_{40}]$	c_9	$reader_{10}$	$reader_{10}$
tag_1	$[t_{55}, t_{56}]$	c_9	$reader_8$	$reader_8$
tag_1	$[t_{79}, t_{80}]$	c_9	$reader_{3'}$	$reader_{3'}$
tag_1	$[t_{85}, t_{87}]$	$e\langle 4, 9\rangle, e\langle 9, 4\rangle$	$reader_4$	$reader_4$
tag_1	$[t_{100}, t_{102}]$	$e\langle 4, 9\rangle, e\langle 9, 4\rangle$	$reader_4$	$reader_4$
tag_1	$[t_{109}, t_{112}]$	c_9	$reader_{6'}$	$reader_6$
tag_1	$[t_{125}, t_{125}]$	c_9	$reader_5$	$reader_5$

- Step 1 transforms an RFID reading sequence to corresponding vertices or edges in deployment graph
- If two consecutive reading sequences are *contiguous*, they should stem from a partitioning pair, which map to an edge
- Otherwise, should come from either a single PRE or a PAR reader
- A PAR is replaced by the set of corresponding edges according to $G_{RFID}.l_e{}^{-1}$
- A PRE always corresponds to one or several cells according to $Mapping\ 2$

Off-line Tracking (Refinement Step 2)

Table 3. Trajectory After Step 1

tagID	$[t^{\vdash}, t^{\dashv}]$	Graph Element	Reader$_1$	Reader$_2$
tag_1	$[t_1, t_4]$	$e\langle 10, 9\rangle$	$reader_9$	$reader_{9'}$
tag_1	$[t_7, t_{10}]$	$e\langle 9, 1\rangle$	$reader_{1'}$	$reader_1$
tag_1	$[t_{21}, t_{24}]$	$e\langle 1, 9\rangle$	$reader_1$	$reader_{1'}$
tag_1	$[t_{30}, t_{32}]$	c_9	$reader_7$	$reader_7$
tag_1	$[t_{39}, t_{40}]$	c_9	$reader_{10}$	$reader_{10}$
tag_1	$[t_{55}, t_{56}]$	c_9	$reader_8$	$reader_8$
tag_1	$[t_{79}, t_{80}]$	c_9	$reader_{3'}$	$reader_{3'}$
tag_1	$[t_{85}, t_{87}]$	$e\langle 4, 9\rangle, e\langle 9, 4\rangle$	$reader_4$	$reader_4$
tag_1	$[t_{100}, t_{102}]$	$e\langle 4, 9\rangle, e\langle 9, 4\rangle$	$reader_4$	$reader_4$
tag_1	$[t_{109}, t_{112}]$	c_9	$reader_{6'}$	$reader_6$
tag_1	$[t_{125}, t_{125}]$	c_9	$reader_5$	$reader_5$

Table 4. Trajectory After Steps 2 and 3

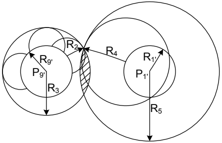
tagID	$[t^{\vdash}, t^{\dashv}]$	Step 2	Step 3
tag_1	$[t_5, t_6]$	c_9	$c_9 \cap \Theta(reader_{9'}, reader_{1'}, t_5, t_6)$
tag_1	$[t_{11}, t_{20}]$	c_1	$c_1 \cap \Theta(reader_1, reader_1, t_{11}, t_{20})$
tag_1	$[t_{25}, t_{29}]$	c_9	$c_9 \cap \Theta(reader_{1'}, reader_7, t_{25}, t_{29})$
tag_1	$[t_{33}, t_{38}]$	c_9	$c_9 \cap \Theta(reader_7, reader_{10}, t_{33}, t_{38})$
tag_1	$[t_{41}, t_{54}]$	c_9	$c_9 \cap \Theta(reader_{10}, reader_8, t_{41}, t_{54})$
tag_1	$[t_{57}, t_{78}]$	c_9	$c_9 \cap \Theta(reader_8, reader_{3'}, t_{57}, t_{78})$
tag_1	$[t_{81}, t_{84}]$	c_9	$c_9 \cap \Theta(reader_{3'}, reader_4, t_{81}, t_{84})$
tag_1	$[t_{88}, t_{99}]$	c_4, c_9	$(c_4, c_9) \cap \Theta(reader_4, reader_4, t_{88}, t_{89})$
tag_1	$[t_{103}, t_{108}]$	c_9	$c_9 \cap \Theta(reader_4, reader_{6'}, t_{103}, t_{108})$
tag_1	$[t_{113}, t_{114}]$	c_9	$c_9 \cap \Theta(reader_6, reader_5, t_{113}, t_{114})$

- The *graph elements* from Step 1 indicates some region(s) within which the object may be in during the vacant time interval
- Check its previous record's tail element and current record's head element, select their intersection as Step 2's candidate

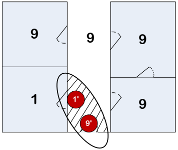
Off-line Tracking (Refinement Step 3)

- Calculate the *possible region* Θ according to maximum speed limit
- Circle based possible region
  - locations: $P_{9'}$, $P_{1'}$
  - activation ranges: $R_{9'}$, $R_{1'}$
  - for $t_x \in [t_5, t_6]$,
    $\Delta t_1 = t_x - reading_2.t^{\vdash}$,
    $\Delta t_2 = reading_3.t^{\dashv} - t_x$
  - $R_3 = R_{9'} + V_{max} * \Delta t_1$,
    $R_5 = R_{1'} + V_{max} * \Delta t_2$
- Ellipse based possible region
  - foci: two points belonging to the circle centered at $P_{9'}$, $P_{1'}$
  - length of major axis is:

$$2a = V_{max} * (\Delta t_1 + \Delta t_2)$$



(a) Circle based possible region

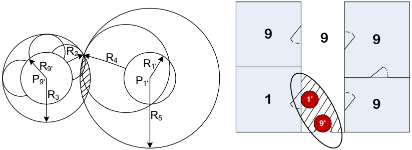


(b) Ellipse based possible region

Off-line Tracking (Refinement Step 3)

- Calculate the *possible region* Θ according to maximum speed limit
- Circle based possible region
  - locations: $P_{9'}, P_{1'}$
  - activation ranges: $R_{9'}, R_{1'}$
  - for $t_x \in [t_5, t_6]$,
    $\Delta t_1 = t_x - reading_2.t^{\vdash}$,
    $\Delta t_2 = reading_3.t^{\dashv} - t_x$
  - $R_3 = R_{9'} + V_{max} * \Delta t_1$,
    $R_5 = R_{1'} + V_{max} * \Delta t_2$
- Ellipse based possible region
  - foci: two points belonging to the circle centered at $P_{9'}, P_{1'}$
  - length of major axis is:

$$2a = V_{max} * (\Delta t_1 + \Delta t_2)$$



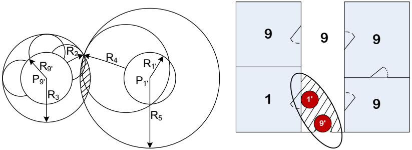
(a) Circle based possible region (b) Ellipse based possible region

$$\left.\begin{aligned}&(reading_1, reader_9, tag_1, t_1, t_2)\\&(reading_2, reader_{9'}, tag_1, t_3, t_4)\\&(reading_3, reader_{1'}, tag_1, t_7, t_8)\\&(reading_4, reader_1, tag_1, t_9, t_{10})\end{aligned}\right\} \xrightarrow{Step\ 1}$$

Off-line Tracking (Refinement Step 3)

- Calculate the *possible region* Θ according to maximum speed limit
- Circle based possible region
  - locations: $P_{9'}$, $P_{1'}$
  - activation ranges: $R_{9'}$, $R_{1'}$
  - for $t_x \in [t_5, t_6]$,
    $\Delta t_1 = t_x - reading_2.t^{\vdash}$,
    $\Delta t_2 = reading_3.t^{\dashv} - t_x$
  - $R_3 = R_{9'} + V_{max} * \Delta t_1$,
    $R_5 = R_{1'} + V_{max} * \Delta t_2$
- Ellipse based possible region
  - foci: two points belonging to the circle centered at $P_{9'}$, $P_{1'}$
  - length of major axis is:

$$2a = V_{max} * (\Delta t_1 + \Delta t_2)$$



(a) Circle based possible region (b) Ellipse based possible region

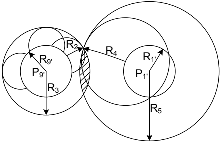
$$\left.\begin{array}{l}(reading_1, reader_9, tag_1, t_1, t_2)\\(reading_2, reader_{9'}, tag_1, t_3, t_4)\\(reading_3, reader_{1'}, tag_1, t_7, t_8)\\(reading_4, reader_1, tag_1, t_9, t_{10})\end{array}\right\}\xrightarrow{Step\ 1}$$

$$\left.\begin{array}{l}(tag_1, [t_1, t_4], e\langle 10, 9\rangle, reader_9, reader_{9'})\\(tag_1, [t_7, t_{10}], e\langle 9, 1\rangle, reader_{1'}, reader_1)\end{array}\right\}\xrightarrow{Step\ 2}$$

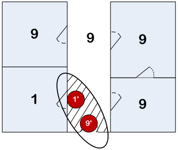
Off-line Tracking (Refinement Step 3)

- Calculate the *possible region* Θ according to maximum speed limit
- Circle based possible region
  - locations: $P_{9'}$, $P_{1'}$
  - activation ranges: $R_{9'}$, $R_{1'}$
  - for $t_x \in [t_5, t_6]$,
    $\Delta t_1 = t_x - reading_2.t^{\vdash}$,
    $\Delta t_2 = reading_3.t^{\dashv} - t_x$
  - $R_3 = R_{9'} + V_{max} * \Delta t_1$,
    $R_5 = R_{1'} + V_{max} * \Delta t_2$
- Ellipse based possible region
  - foci: two points belonging to the circle centered at $P_{9'}$, $P_{1'}$
  - length of major axis is:

$$2a = V_{max} * (\Delta t_1 + \Delta t_2)$$



(a) Circle based possible region



(b) Ellipse based possible region

$$\left.\begin{array}{l}(reading_1, reader_9, tag_1, t_1, t_2)\\(reading_2, reader_{9'}, tag_1, t_3, t_4)\\(reading_3, reader_{1'}, tag_1, t_7, t_8)\\(reading_4, reader_1, tag_1, t_9, t_{10})\end{array}\right\} \xrightarrow{Step\ 1}$$

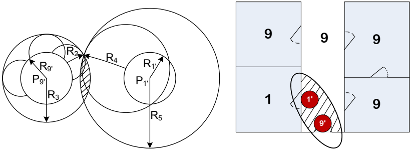
$$\left.\begin{array}{l}(tag_1, [t_1, t_4], e\langle 10, 9\rangle, reader_9, reader_{9'})\\(tag_1, [t_7, t_{10}], e\langle 9, 1\rangle, reader_{1'}, reader_1)\end{array}\right\} \xrightarrow{Step\ 2}$$

$$(tag_1, [t_5, t_6], c_9, reader_{9'}, reader_{1'}) \xrightarrow{Step\ 3}$$

Off-line Tracking (Refinement Step 3)

- Calculate the *possible region* Θ according to maximum speed limit
- Circle based possible region
  - locations: $P_{9'}, P_{1'}$
  - activation ranges: $R_{9'}, R_{1'}$
  - for $t_x \in [t_5, t_6]$,
    $\Delta t_1 = t_x - reading_2.t^{\vdash}$,
    $\Delta t_2 = reading_3.t^{\dashv} - t_x$
  - $R_3 = R_{9'} + V_{max} * \Delta t_1$,
    $R_5 = R_{1'} + V_{max} * \Delta t_2$
- Ellipse based possible region
  - foci: two points belonging to the circle centered at $P_{9'}, P_{1'}$
  - length of major axis is:

$$2a = V_{max} * (\Delta t_1 + \Delta t_2)$$



(a) Circle based possible region (b) Ellipse based possible region

$$\left.\begin{array}{l}(reading_1, reader_9, tag_1, t_1, t_2)\\(reading_2, reader_{9'}, tag_1, t_3, t_4)\\(reading_3, reader_{1'}, tag_1, t_7, t_8)\\(reading_4, reader_1, tag_1, t_9, t_{10})\end{array}\right\} \xrightarrow{Step\ 1}$$

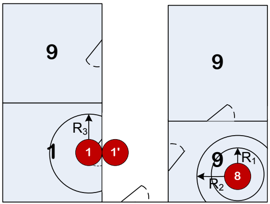
$$\left.\begin{array}{l}(tag_1, [t_1, t_4], e\langle 10, 9\rangle, reader_9, reader_{9'})\\(tag_1, [t_7, t_{10}], e\langle 9, 1\rangle, reader_{1'}, reader_1)\end{array}\right\} \xrightarrow{Step\ 2}$$

$$(tag_1, [t_5, t_6], c_9, reader_{9'}, reader_{1'}) \xrightarrow{Step\ 3}$$

$$(tag_1, [t_5, t_6], c_9 \cup \Theta(reader_{9'}, reader_{1'}, t_5, t_6))$$

On-line Tracking

Given $\langle readerID, tagID, t, flag \rangle$. **On-line tracking** is intended to infer the trajectory in the time interval between the last observation and the current time or even in the future.



- $flag = START$, object $tagID$ is in the activation range of $readerID$ at time t.
- $flag = END$, the object is beyond the activation range of $readerID$ and not in the range of any other readers.
  - constrains by a circle determined by the most recent observing reader's range.
  - further refined if an object has recently been detected by a partitioning reader pair.

Research Directions

- Extend the deployment graphs to accommodate RFID readers with large and overlapping activation ranges.
- Using multiple deployment graphs for several positioning technologies.
- To enhance on-line tracking. Historical data $\rightarrow$ association rules $\rightarrow$ better prediction.

References

C. S. Jensen, H. Lu, and B. Yang.
Graph model based indoor tracking.
In *MDM*, pp. 122–131, 2009.

Civilis, Alminas and Jensen, Christian S and Pakalnis, Stardas.
Techniques for efficient road-network-based tracking of moving objects.
In *TKDE*, pp. 698–712, 2005.

Pfoser, Dieter and Jensen, Christian S.
Capturing the uncertainty of moving-object representations.
In *Advances in Spatial Databases*, pp. 111–131, 1999.

The End. Thanks :)