32nd IEEE International Conference on Data Engineering
May 16-20, 2016 · Helsinki, Finland

Indoor Data Management

Hua Lu
Department of Computer Science
Aalborg University, Denmark
luhua@cs.aau.dk

AALBORG UNIVERSITET

Muhammad Aamir Cheema
Faculty of Information Technology
Monash University, Australia
aamir.cheema@monash.edu

MONASH University

Outline

- Introduction, Motivation and Challenges
- Existing Research
- Future Research Directions

Surprising 87%

- Californians spent on average 87% of their time indoors
  - California Air Resources Board survey, 1987-1988
- USA residents spent on average 87% of their time indoors
  - National Human Activity Pattern Survey 1992-1994
- Surveys conducted in other countries/regions disclosed the similar percent.
- Typical indoor spaces
  - Shopping malls, office buildings, airports, metro/railway stations, exhibition venues, conference venues…

Indoor Moving Objects (IMO)




Complex Indoor Space Examples

- Beijing Capital Airport
  - ~246,400 passengers daily in 2015
- New Town Plaza, Hong Kong
  - 200,000 m^2, 34 interconnected buildings
  - Weekend traffic 320,000 people (2004)

How to manage the spaces and objects?

- The New University Hospital in Aarhus, Denmark
  - The largest hospital project in the history of Denmark and as of 2011 the biggest building project in Northern Europe.
  - It needs to track 164,000 objects (persons, equipment, materials, etc.)
- Copenhagen Airport
  - 2.3 million passengers in March 2016

Indoor Positioning

- Assisted-GPS
- Cellular system
- Short-range wireless
  - Wi-Fi
  - Bluetooth, e.g., iBeacon
  - Infrared, RFID, NFC
- The Earth's magnetic field
  - E.g., IndoorAtlas, Finland
- Special sensors and instruments
  - Sextant, gyroscope

Smartphones!



- Many people and other indoor moving objects
- Smart hardware
- Appropriate positioning

Indoor Mobility Data

Indoor Trajectory Mining

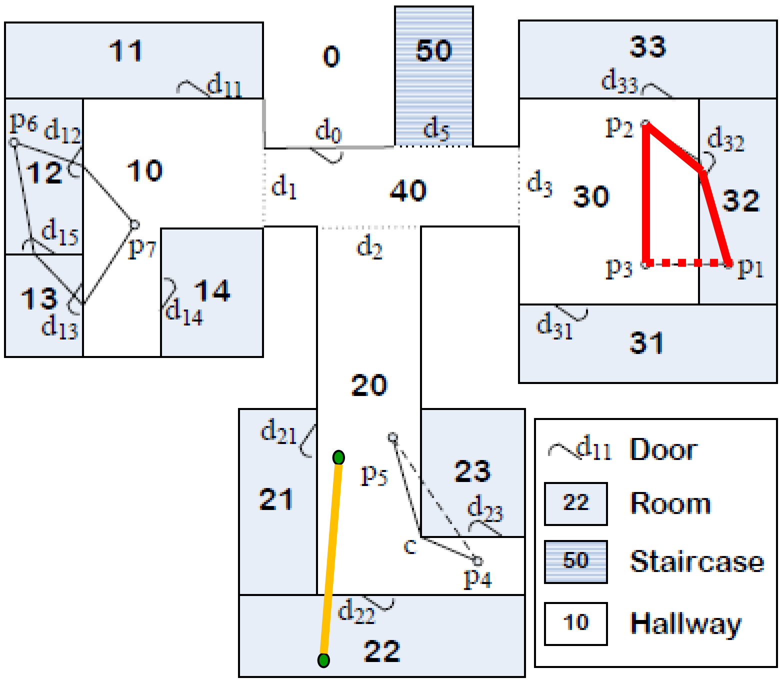
Indoor Venues: Next Frontier for LBS

- Make the physical world searchable down to the object level.
- Provide a new platform for in-store shopper engagement and experiences.
- Digitize the call for help.
- Make smart devices responsive to their environment.
- Enable universal tracking and monitoring of people and physical assets.
- Improve wayfinding to your actual destination.

- From http://www.forbes.com/sites/forrester/2013/01/23/indoor-venues-are-the-next-frontier-for-location-based-services/

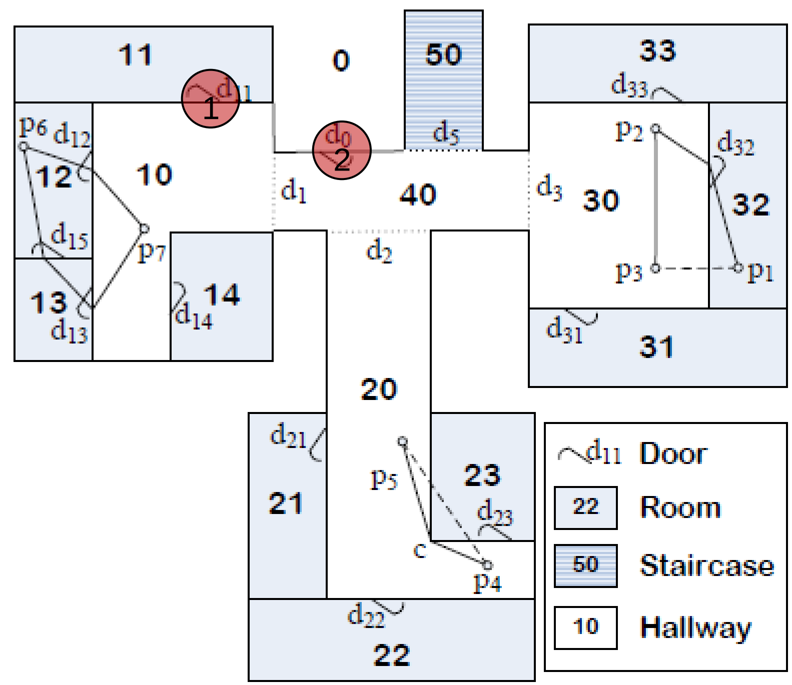
Technical Challenges: Space

- Indoor spaces are characterized by many unique entities like rooms, walls, doors, hallways, elevators, lifts, etc.
- Such entities enable as well as disable indoor movements.
- Consequently, indoor spaces cannot be modelled as Euclidean spaces or spatial (road) networks.
  - Euclidean distance metric may fall short in an indoor setting.
- Also, geometric movement representations are not suitable for describing indoor moving objects and their trajectories.



Technical Challenges: Positioning

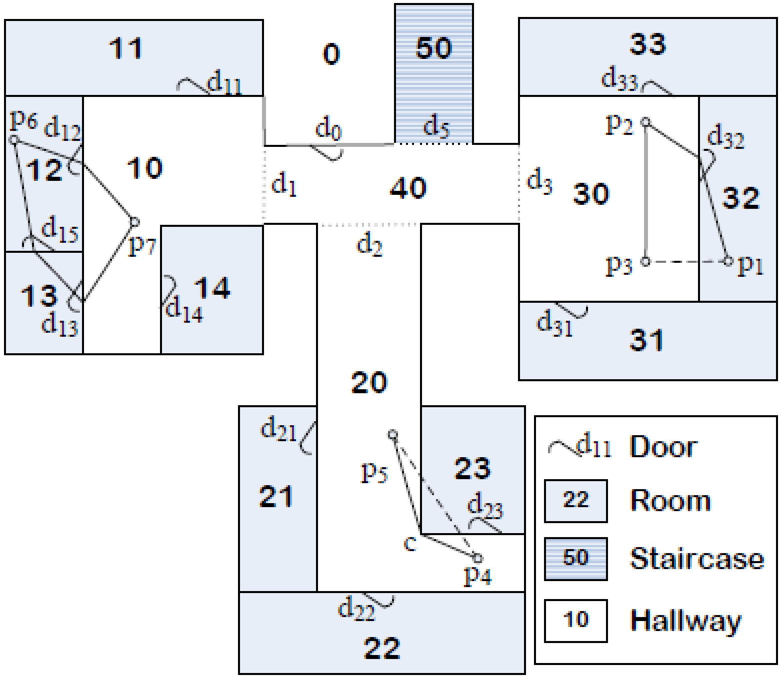
- Indoor positioning technologies differ from GPS
  - Fingerprinting, *proximity analysis, and hybrid*
  - *E.g., in* proximity analysis, RFID readers are deployed to detect moving objects with RFID tags.
    - Such technologies are unable to report velocities or accurate locations continuously.
    - They cover only part of rather than the whole space.
  - E.g., in *fingerprinting*, radio maps are created in the offline phase and used to estimate user location in the online phase.
- In general, state-of-the-art indoor positioning technologies offers considerably lower accuracy than outdoor GPS.



Technical Challenges: Data

- Indoor space
  - Entities: rooms, walls, doors, hallways, staircases, elevators, etc.
  - Complex topology
- Indoor POIs (point-of-interest)
  - Semantics associated to POIs
- Indoor moving objects
  - Low accuracy, uncertain indoor positioning data
  - Symbolic trajectories

Efficient and effective management of heterogeneous, raw data for indoor applications
•Indoor LBS (Location-based services)
•Security control
•Indoor space use analysis
•…



Outline

- Introduction, Motivation and Challenges
- Existing Research
  - Data Modeling for Indoor Space
  - Preprocessing Indoor Positioning Data
  - Indexing Indoor Space and Data
  - Querying Indoor Data
  - Other Topics
- Future Research Directions

Data Modeling for Indoor Space

- CityGML [10]
- IndoorGML [41]
  - Node-Relation Structure (NRS) [29]
- Distance-aware model [34]

CityGML

- CityGML models 3D cities
- Models relevant parts of the virtual city according to their semantics, geometry, topology and appearance
- Multi-scale Modeling (LOD – level of detail)

- LOD 0 – Regional model

- LOD 1 – City model

- LOD 2 – City model with explicit roof structure

- LOD 3 – Detailed architectural model

- LOD 4 – Interior Model

<Source: CityGML>

CityGML

- LOD 4 models indoor features
- Provides explicit relations between semantic objects and their geometrical representations

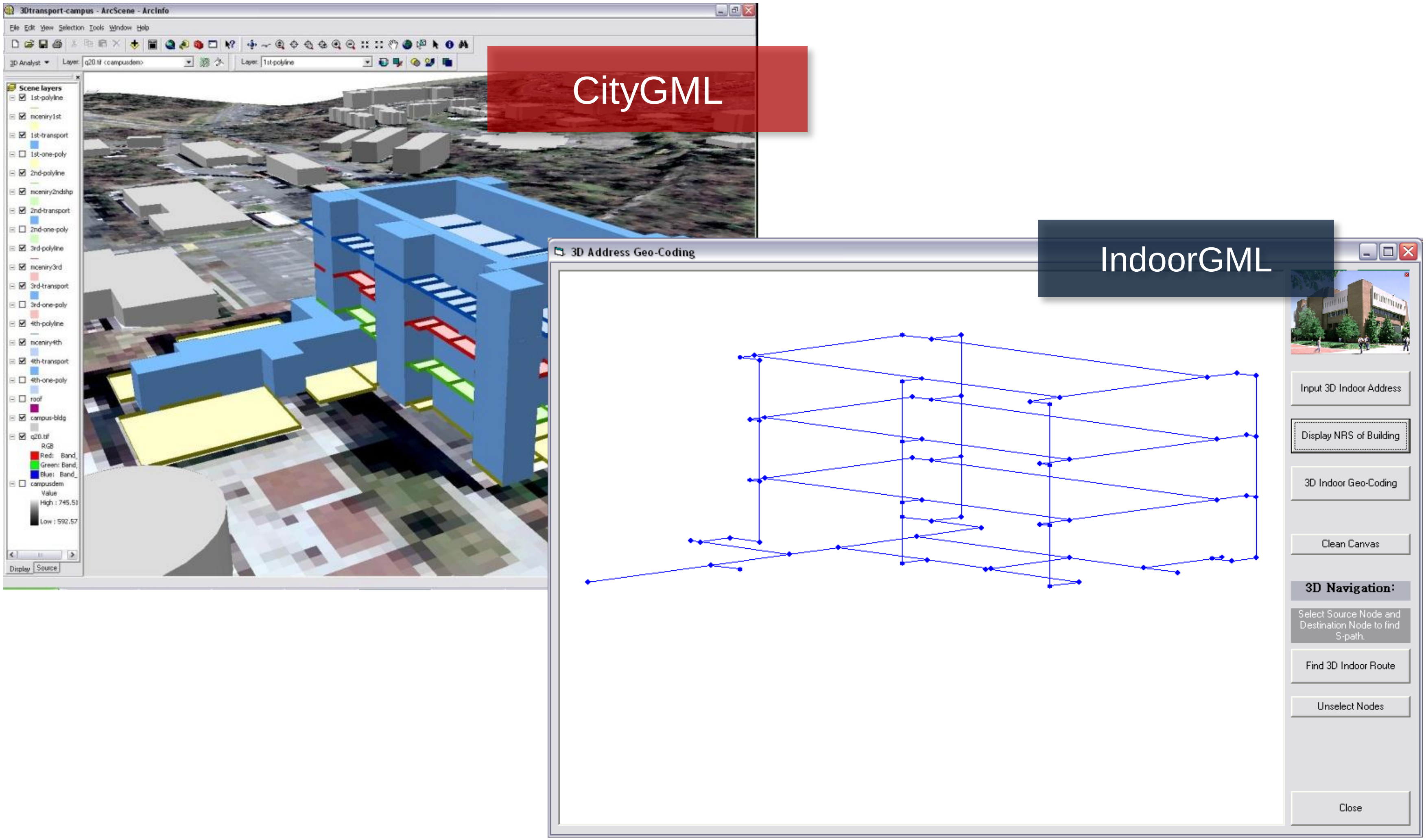
<Source: CityGML>

CityGML

- Pros: excellent visualization and geometric analysis
- Cons: not suitable for indoor location-base services, e.g.,
  - Navigation (how to go to the washroom)

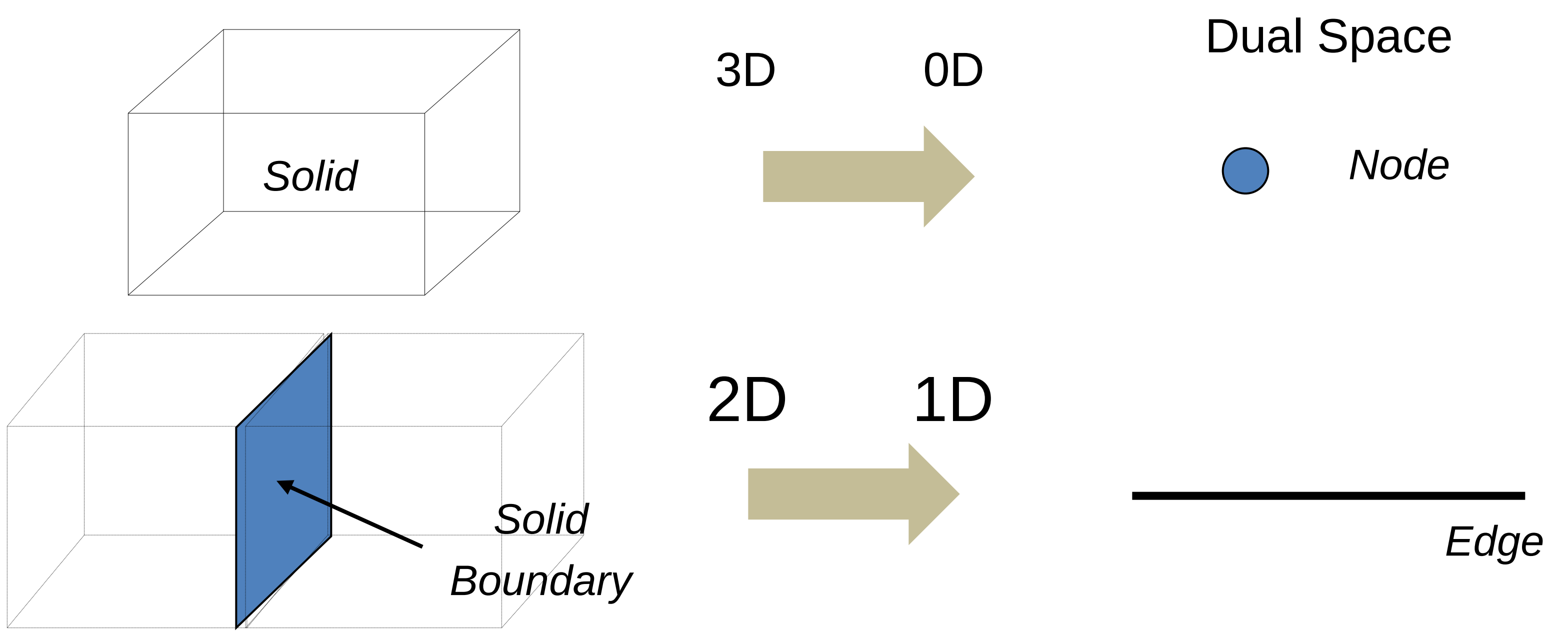
<Source: CityGML>

IndoorGML

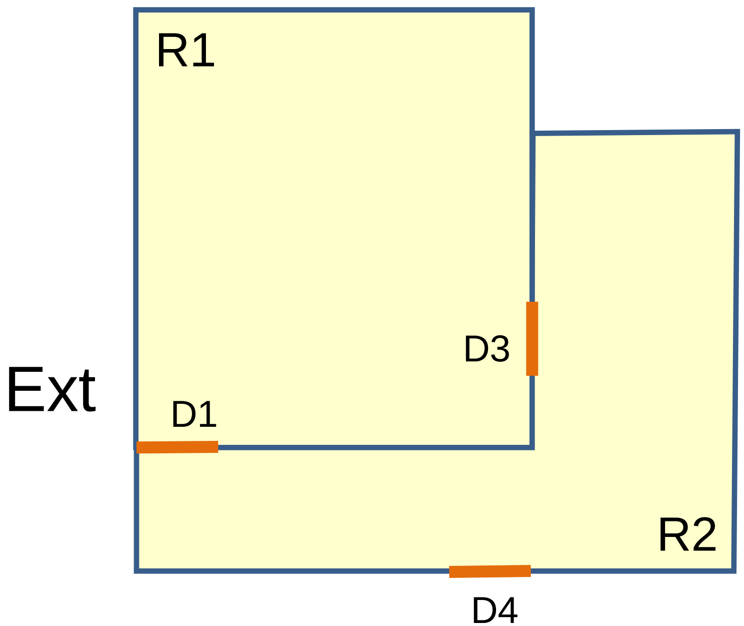

CityGML
IndoorGML
3D Address Geo-Coding
Input 3D Indoor Address
Display NRS of Building
3D Indoor Geo-Coding
Clean Canvas
3D Navigation:
Select Source Node and Destination Node to find S-path.
Find 3D Indoor Route
Unselect Nodes
Close

Node Relation Structure (NRS)

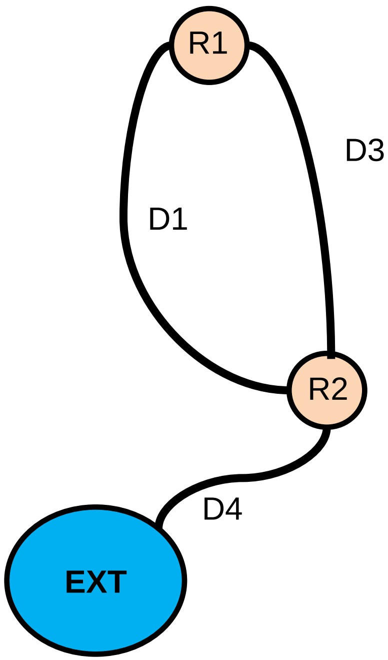
- Conversion from original (primal space) to dual space using Poincare Duality, e.g.,
  - Room → node
  - Door → relation between two nodes



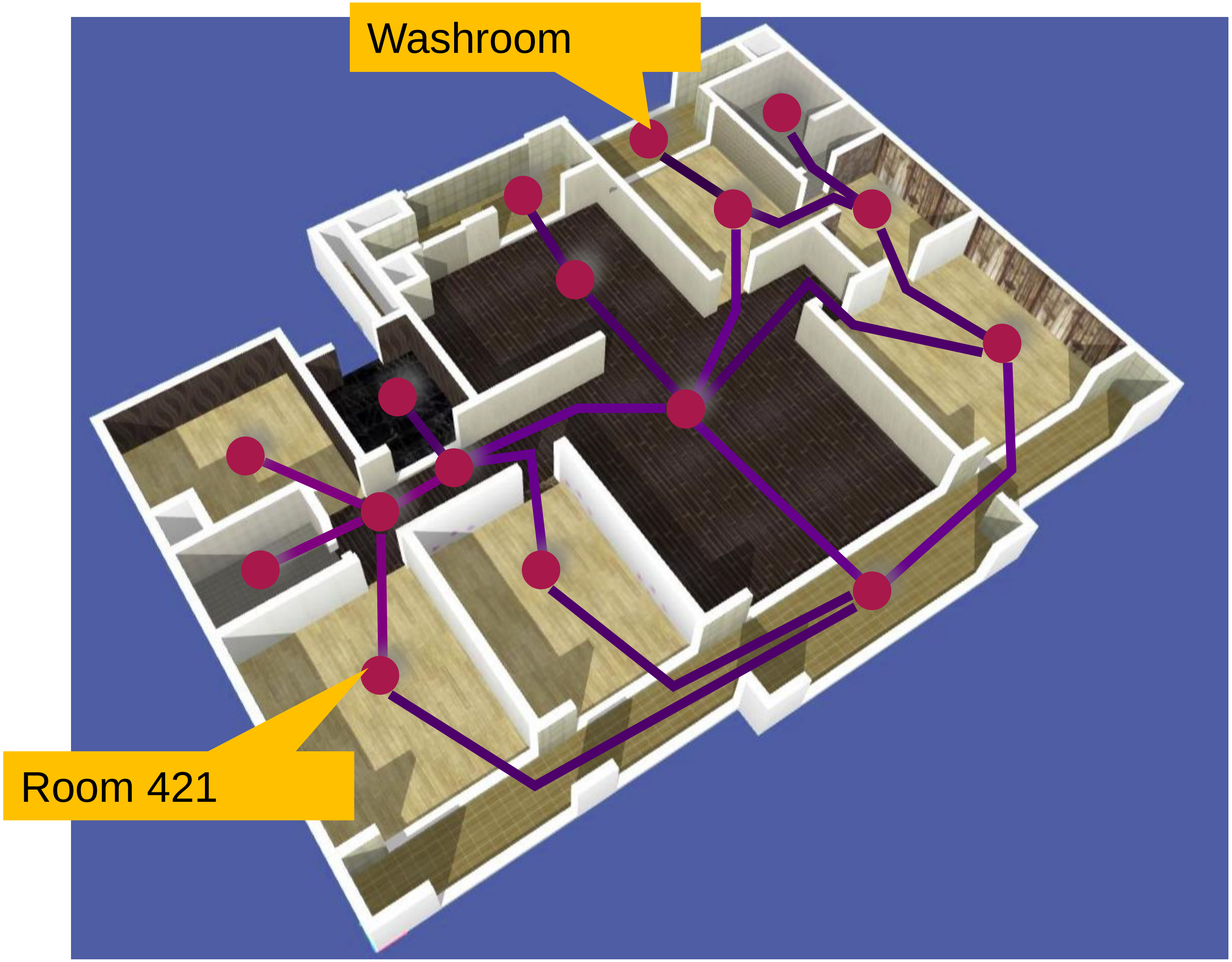
Example 1



Topographic Space

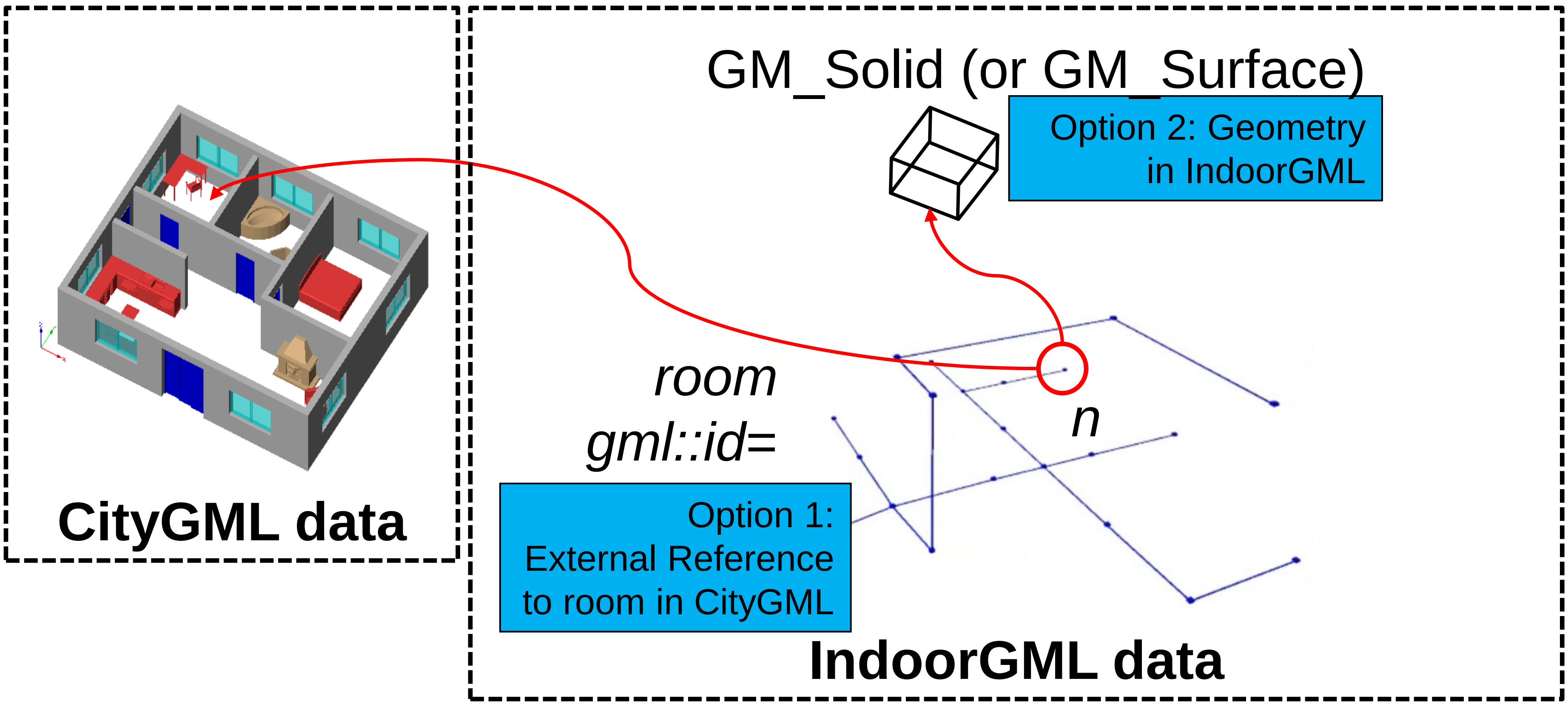


Example 2



Limitations

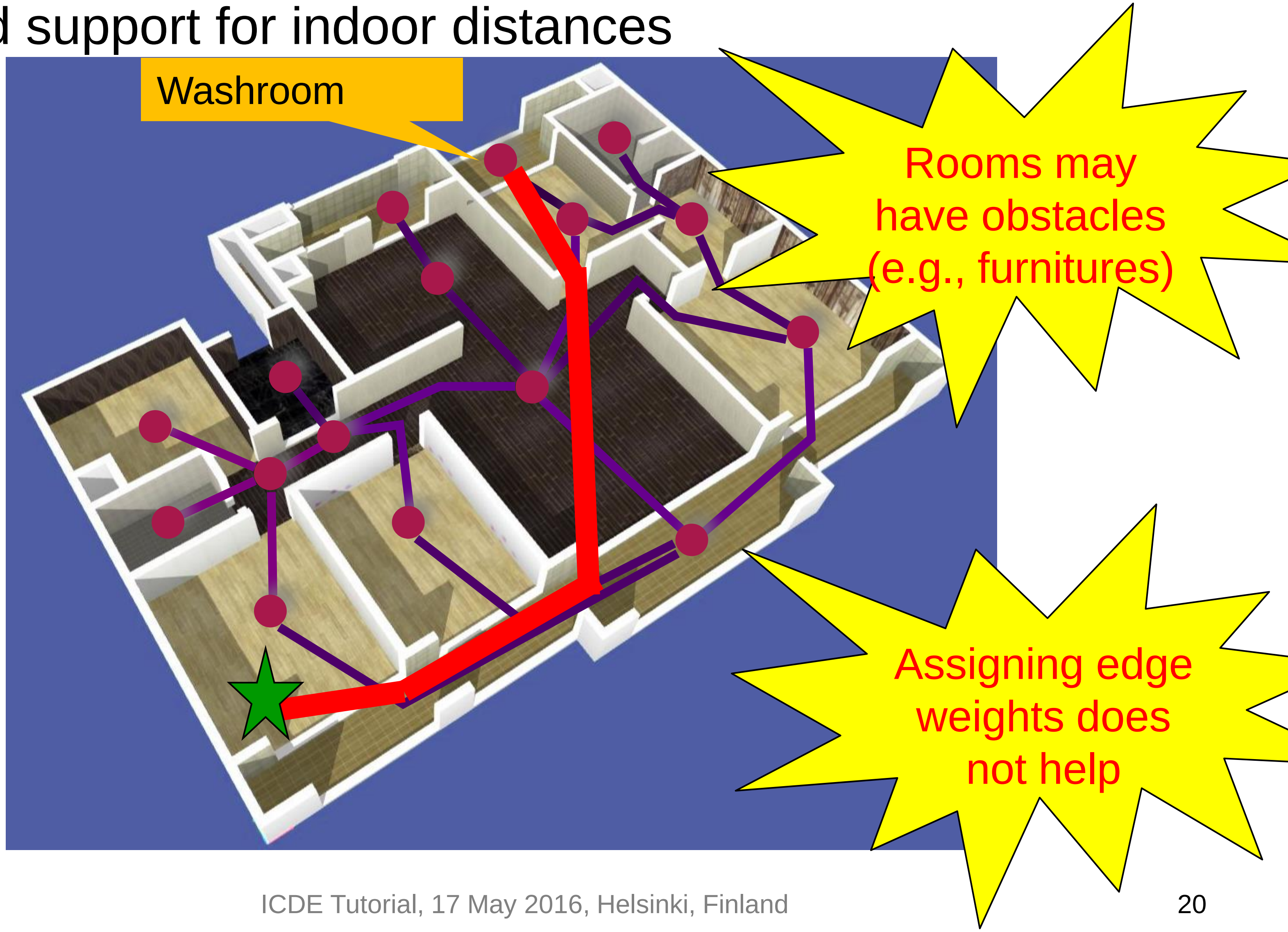
- Poor support for visualization, geometry analysis



Two options to represent geometry of each cell

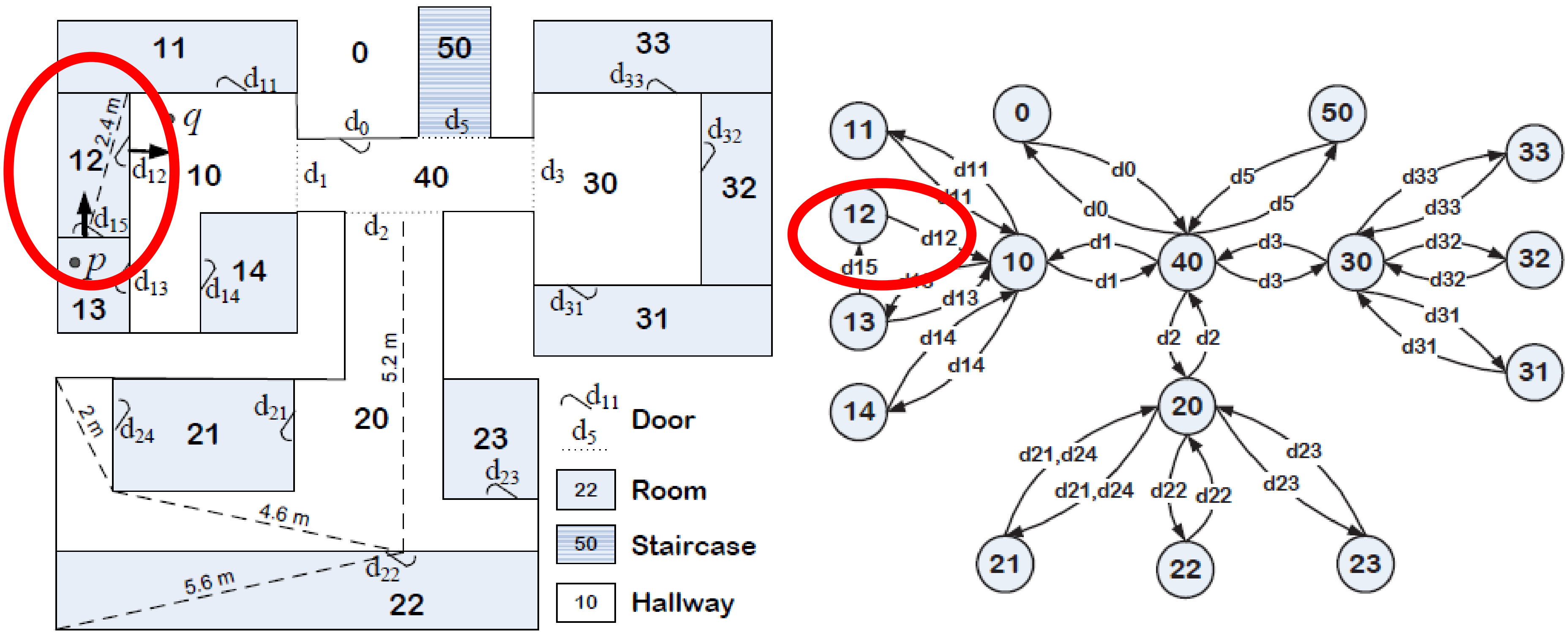
Limitations

- Poor support for visualization, geometry analysis
- Limited support for indoor distances



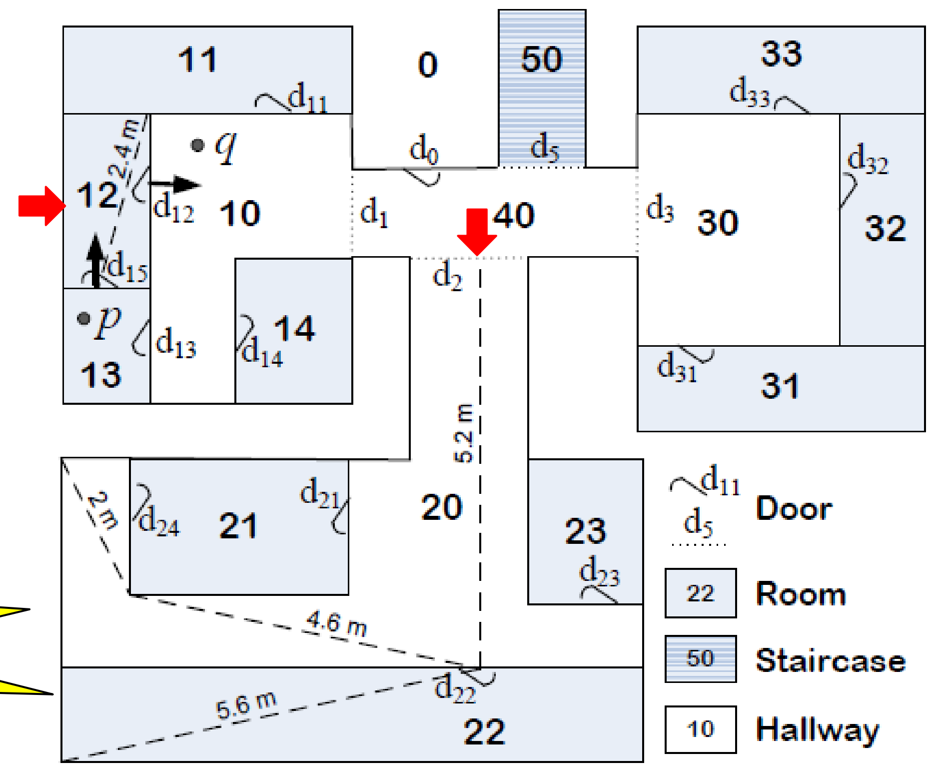
Distance-Aware Model

- Accessibility Base Graph
  - *Similar to Node-Relation Structure except that edges are directional*



Distance-Aware Model

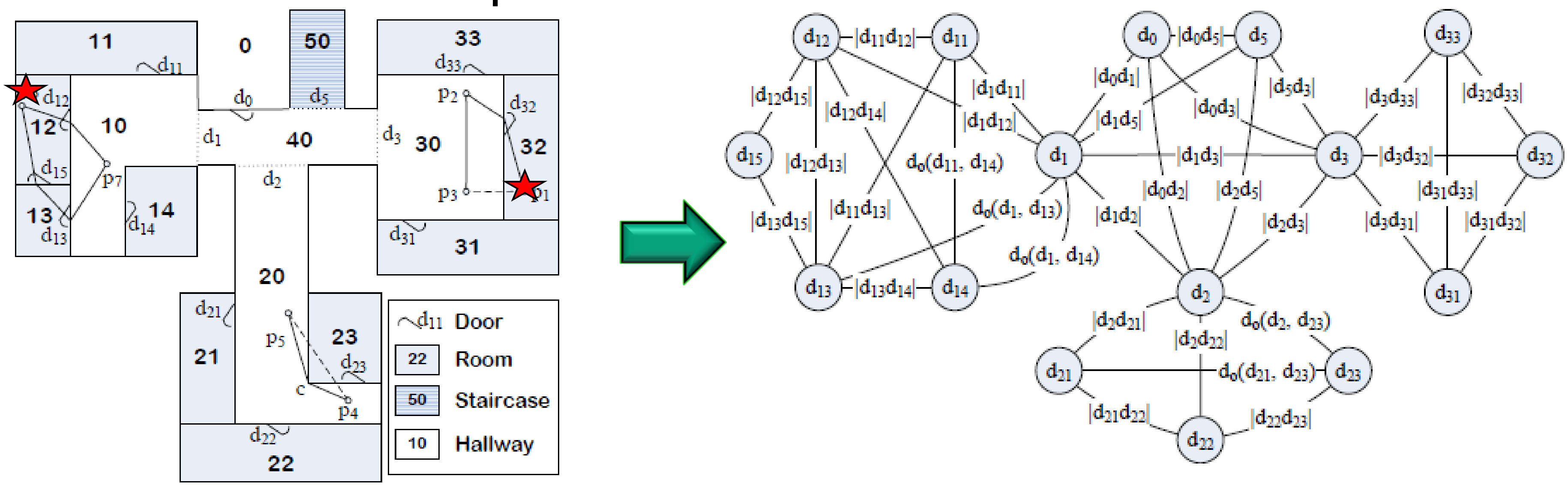
- Store distances between doors connected to the same partitions (e.g., d_{12} and d_{15})
- Given a partition v_k and its doors d_i and d_j
  - $f_{d2d}(v_k, d_i, d_j) = |d_i, d_j|_{v_k}$
  - $f_{d2d}(v_k, d_i, d_j) = \infty$ if d_i only allows exit
- Examples
  - $f_{d2d}(v_{12}, d_{15}, d_{12})$ = 2.2 meters
  - $f_{d2d}(v_{12}, d_{12}, d_{15}) = \infty$
  - $f_{d2d}(v_{20}, d_2, d_{22})$ = 5.2 meters
  - $f_{d2d}(v_{20}, d_{22}, d_2)$ = 5.2 meters



Distance considers obstacles in partitions

Distance-Aware Model

- Door-to-Door Graph



- Door-to-Door Distance Matrix

$$
\begin{array}{c}
\\ d_1 \\ d_{11} \\ d_{12} \\ d_{13} \\ d_{14} \\ d_{15}
\end{array}
\begin{pmatrix}
d_1 & d_{11} & d_{12} & d_{13} & d_{14} & d_{15} \\
0 & 1.7 & 2.7 & 3.2 & 2.6 & 4.3 \\
1.7 & 0 & 1.9 & 3.4 & 3 & 4.4 \\
2.7 & 1.9 & 0 & 2 & 2.2 & 3 \\
3.2 & 3.4 & 2 & 0 & 1.2 & 1 \\
2.6 & 3 & 2.2 & 1.2 & 0 & 2.2 \\
3.2 & 3.4 & 1.5 & 3.5 & 3.7 & 0
\end{pmatrix}
$$

Outline

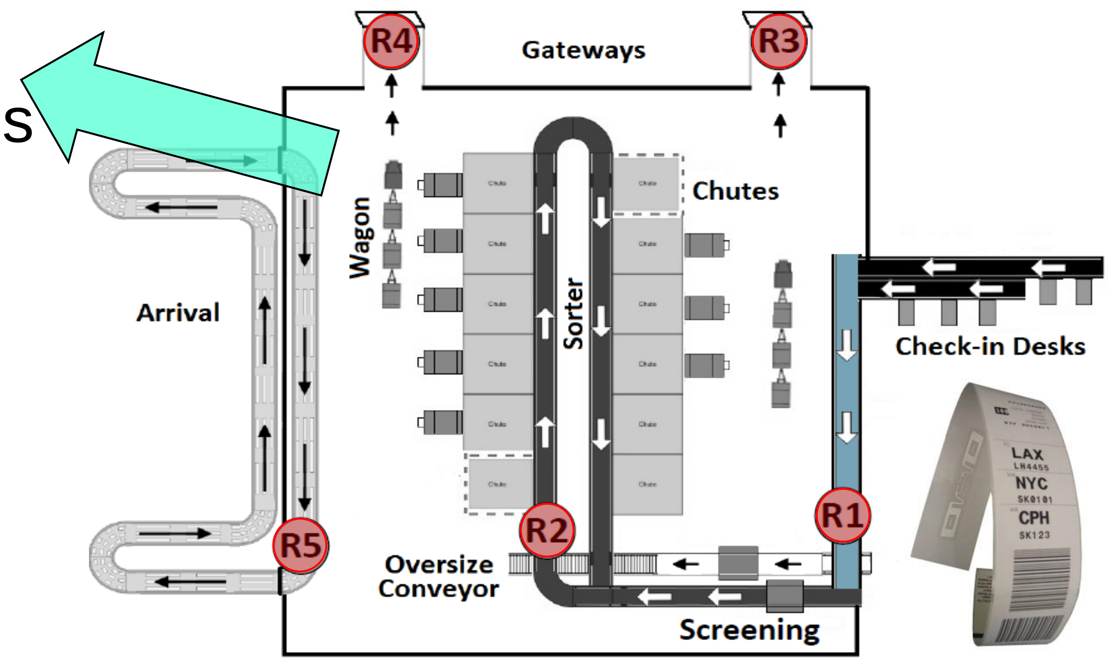
- Introduction, Motivation and Challenges
- Existing Research
  - Data Modeling for Indoor Space
  - Preprocessing Indoor Positioning Data
  - Indexing Indoor Space and Data
  - Querying Indoor Data
  - Other Topics
- Future Research Directions

Preprocessing Indoor Positioning Data

- Cleansing raw RFID data
  - False Positives [6]
  - False Negatives [5]
- Raw RFID data to probabilistic trajectory [11, 12]

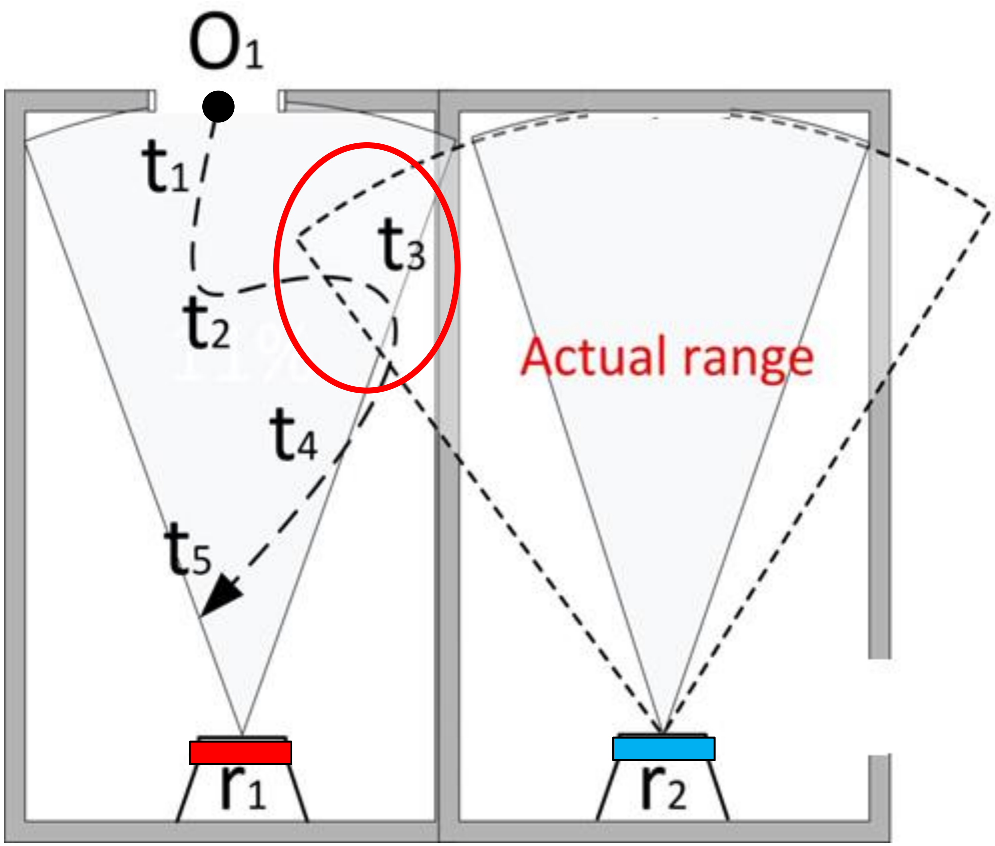
Cleansing Raw RFID Data

- RFID sensors
  - An RFID reader detects an RFID tag, when the tag (the object with the tag) enters the reader's detection range.
  - Deployment locations of RFID readers are recorded in advance.
- Raw reading format
  - (objectID, readerID, t)
- Such raw data contains two types of errors
  - False positive
    - Cross readings
  - False negative
    - Missing readings



False Positives

- A reader mistakenly reads out the tags which are outside its intended detection range.
  - Possible causes: metal reflection, antenna re-direction, etc.

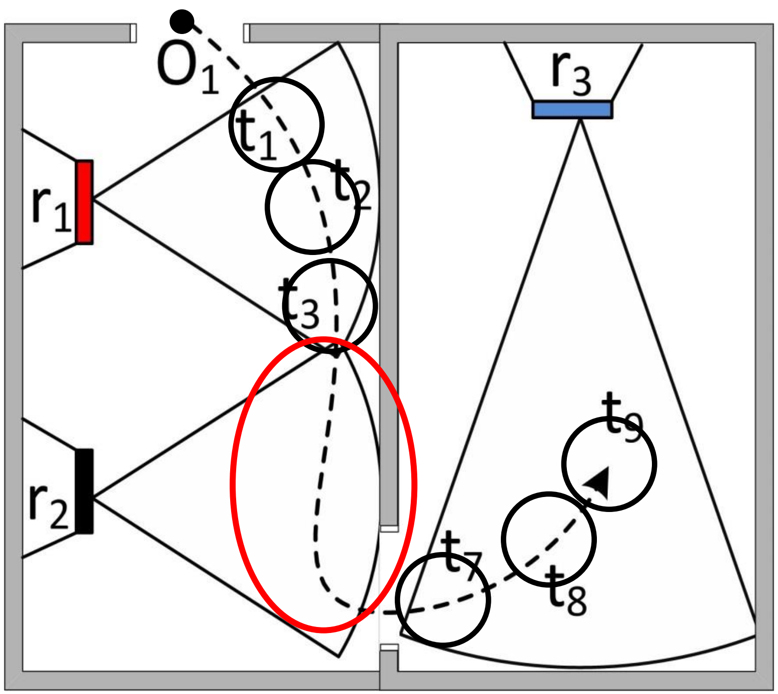


From r_1
$\langle O_1, r_1, t_1 \rangle$
$\langle O_1, r_1, t_2 \rangle$
$\langle O_1, r_1, t_3 \rangle$
$\langle O_1, r_1, t_4 \rangle$
$\langle O_1, r_1, t_5 \rangle$

From r_2
$\langle O_1, r_2, t_3 \rangle$

False Negatives

- A reader fails to read out a tag that is actually in its intended detection range.
  - Possible causes: out of battery, circuit failure, etc.



From r_1

<O_1, r_1, t_1>
<O_1, r_1, t_2>
<O_1, r_1, t_3>

From r_2

?

From r_3

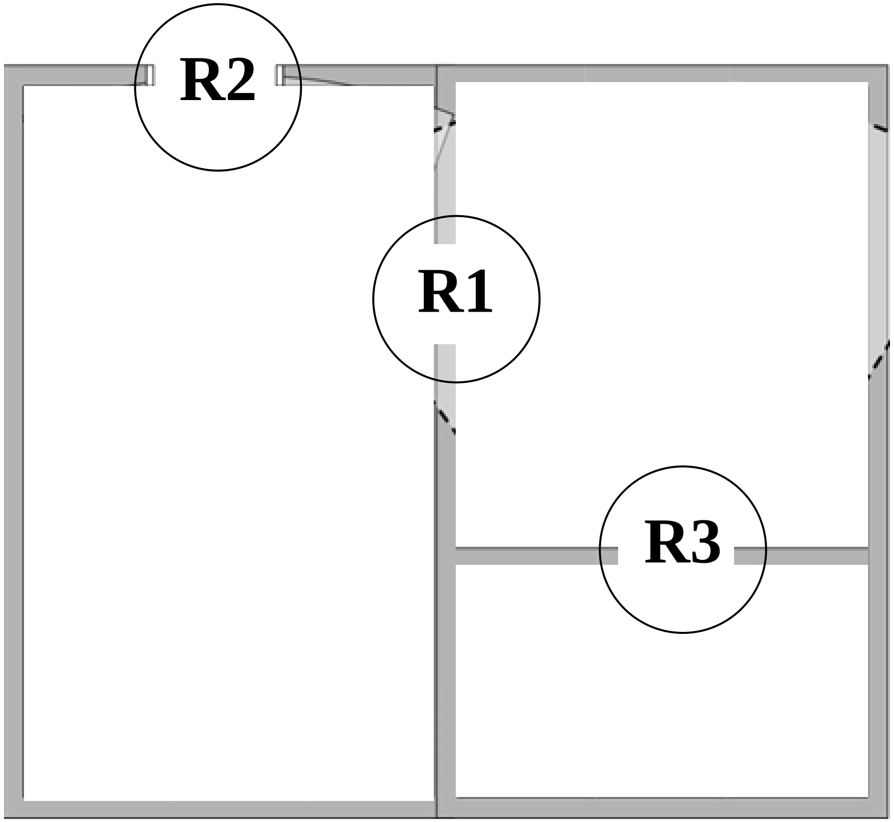
<O_1, r_3, t_7>
<O_1, r_3, t_8>
<O_1, r_3, t_9>

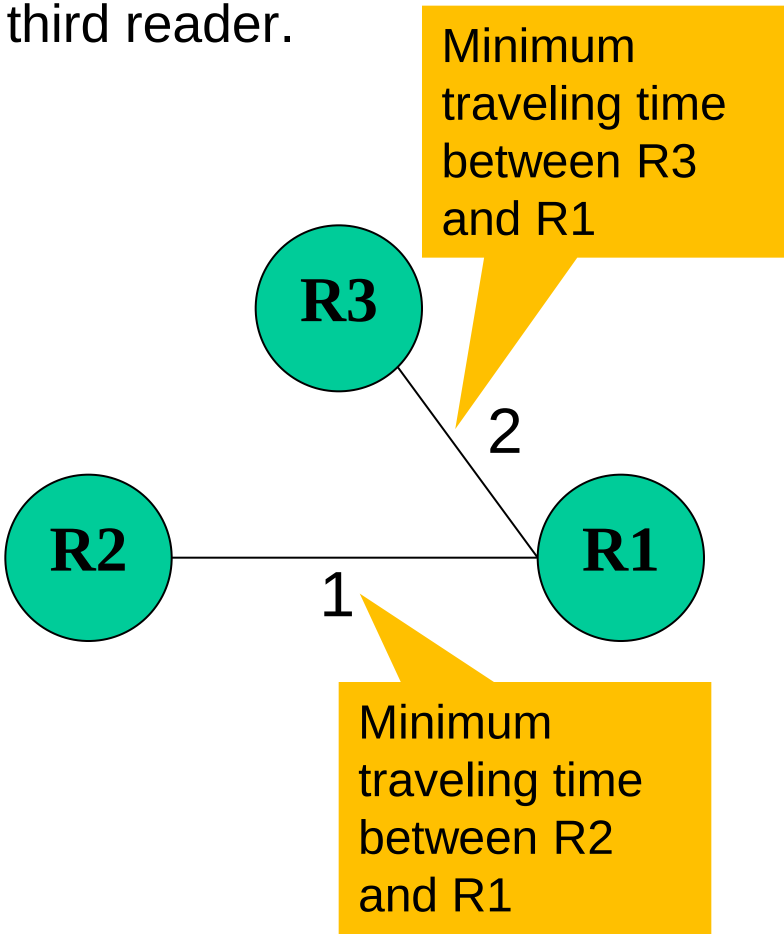
Problem

Cleansing Indoor RFID Tracking Data
- To *reduce* false positives
- To *recover* false negatives

False Positive Cleansing

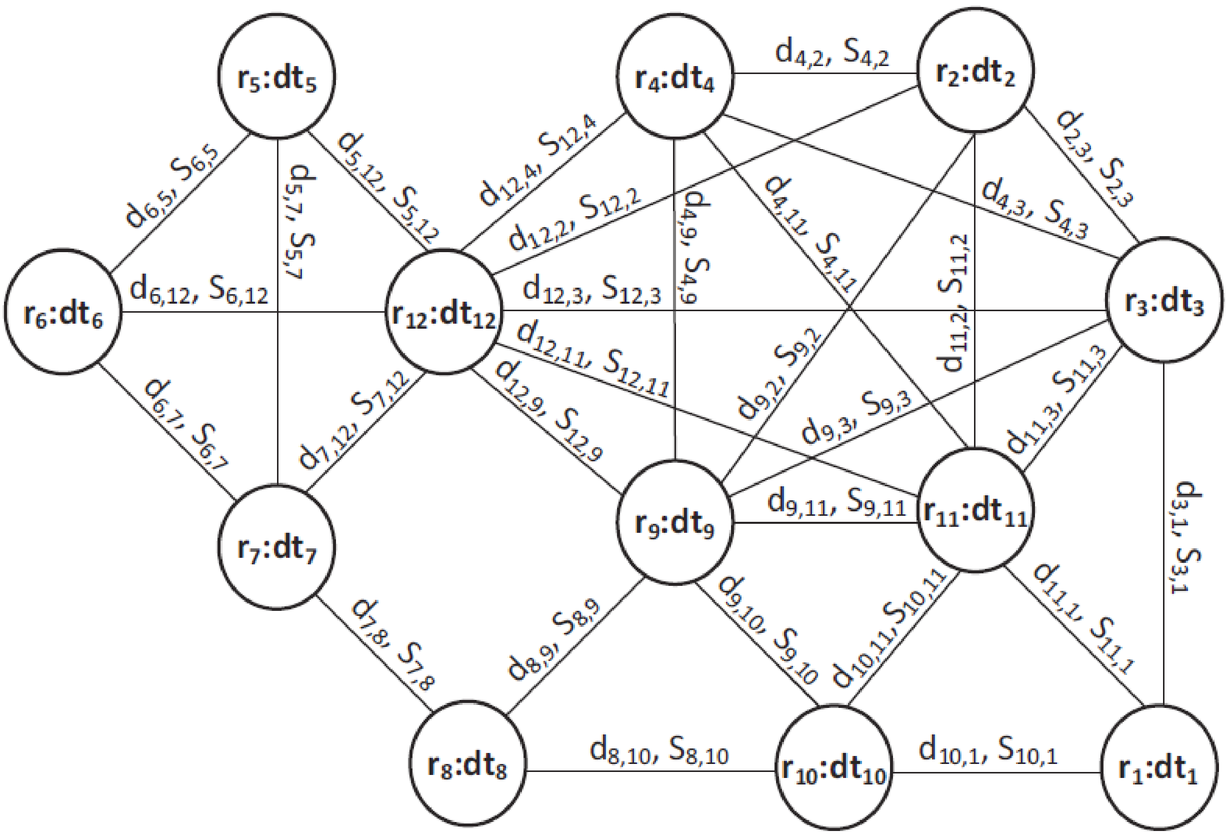
- Distance-Aware Deployment Graph of RFID readers
  - Each node represents a deployed reader.
  - An edge implies that an object can move from one reader to the other without involving a third reader.





False Positive Cleansing

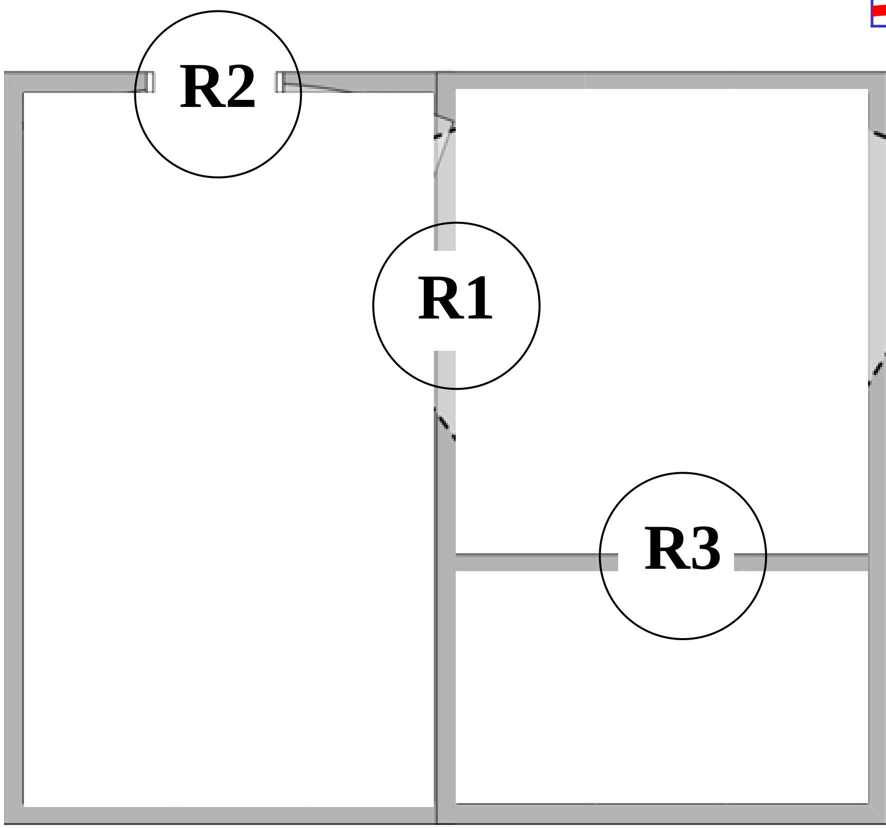
- Distance-Aware Deployment Graph of RFID readers
  - Each node represents a deployed reader.
  - An edge implies that an object can move from one reader to the other without involving a third reader.

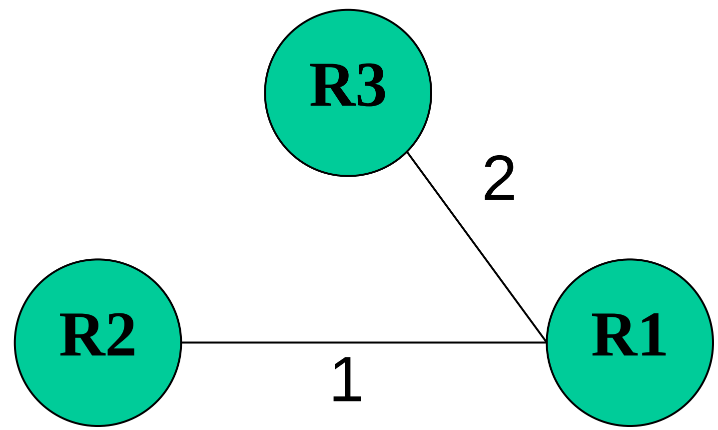


False Positive Cleansing

- Delete records that do not satisfy spatiotemporal constraints

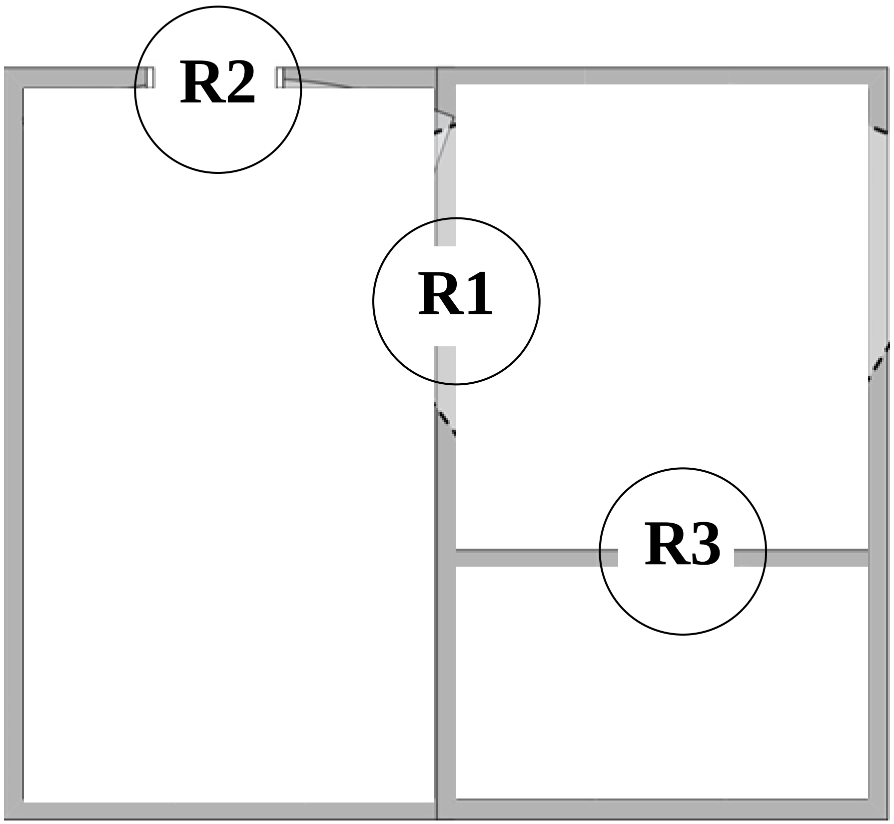
deviceID	objectID	t_s	t_e
R_2	*$object_1$*	*0*	*3*
~~***R_3***~~	~~*$object_1$*~~	~~*5*~~	~~*5*~~

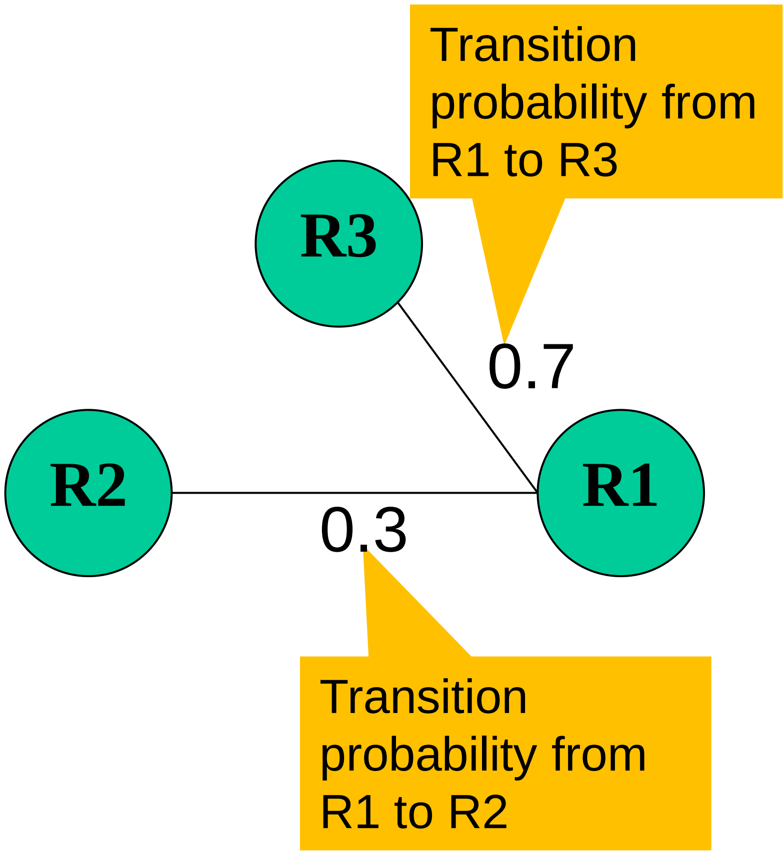




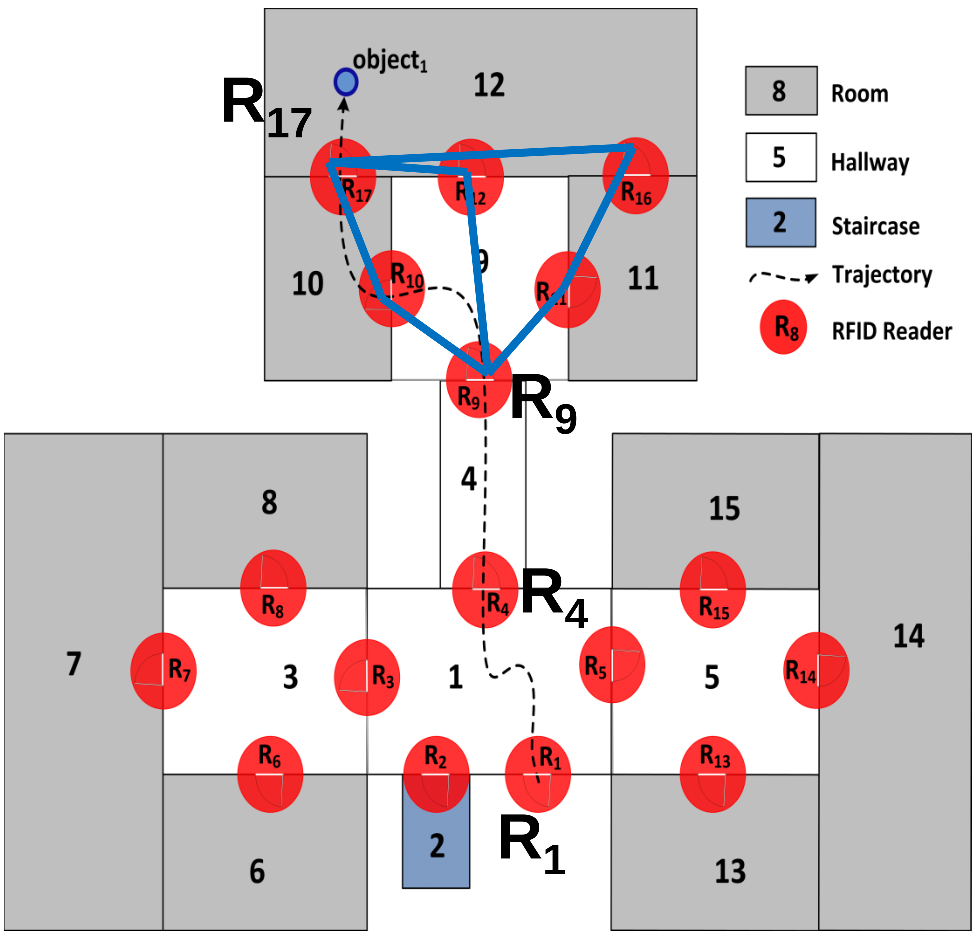
False Negative Recovery

- Augmenting the Distance-Aware Deployment Graph
  - Add a *transition probability* to each edge, i.e., to indicate the possibility that an object moves from reader r_i to reader r_j
  - Transition probability computed using historical data





False Negative Recovery



deviceID	t_s	t_e
R_1	0	3
R_4	7	8
R_9	12	16
R_{17}	20	23

1. Find all possible (non-cyclic) paths between R_9 and R_{17}
2. Delete paths that do not satisfy spatiotemporal constraints
3. Find most likely path (using transition probability of edges)
4. Insert missing readers using the most likely path

Raw Data to Probabilistic Trajectory

- For each tag, the result of the tracking task is a sequence of readings $R_1,\dots,R_T$
  - Each R_i is the set of readers that detected the tag at time point i

Time point	**Set of readers**
1	{r1, r2}
2	{r1, r2}
3	{r3}
4	{r3, r4}
5	{r1}
6	{r1, r2}

Raw Data to Probabilistic Trajectory

- For each tag, the result of the tracking task is a sequence of readings $R_1,\ldots,R_T$
  - Each R_i is the set of readers that detected the tag at time point i

- Convert raw data into a trajectory that records, for each timestamp, probabilsitic location of the object
  - location can be room names, cells over a grid, etc.

Time point	Set of readers	Position
1	{r1, r2}	*Corridor 80%* *Terrace 20%*
2	{r1, r2}	*Corridor 90%* *Terrace 10%*
3	{r3}	*Corridor 40%* *Office 60%*
4	{r3, r4}	*Office 100%*
5	{r1}	*Office 20%* *Corridor*
6	{r1, r2}	*Corridor 100%*

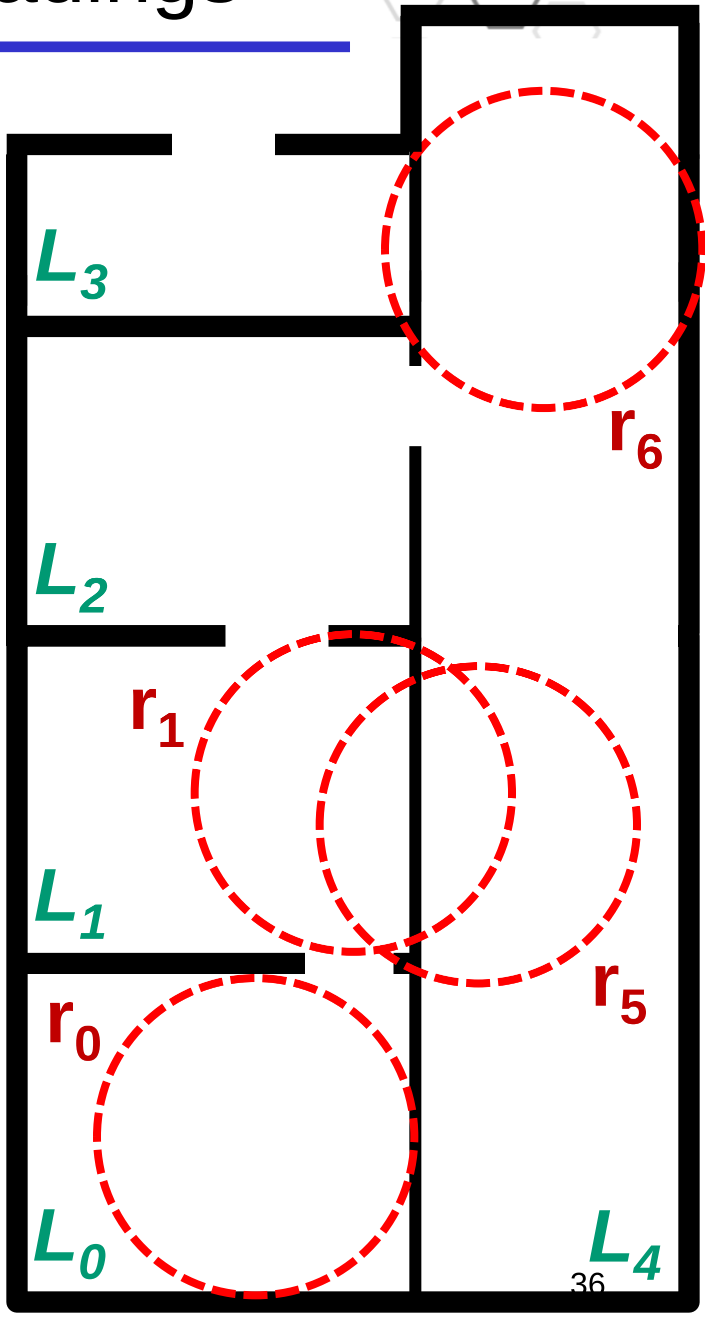
A Naive Interpretation of the Readings

Table of detections

Time	1s	2s	3s	4s
Set of readers	{r0}	{ r1, r5}	∅	{r6}

- Consider the time points separately

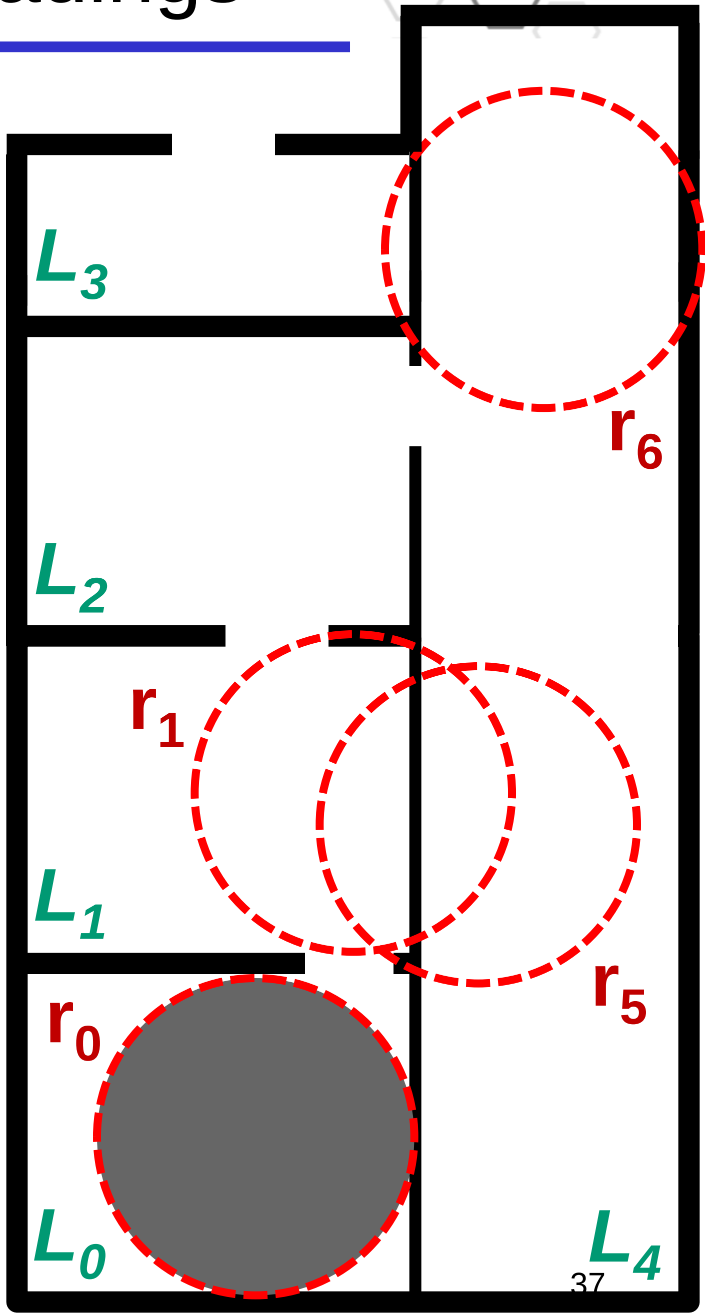
- ***For simplicity, disregard probabilities for now***



A Naive Interpretation of the Readings

Table of detections

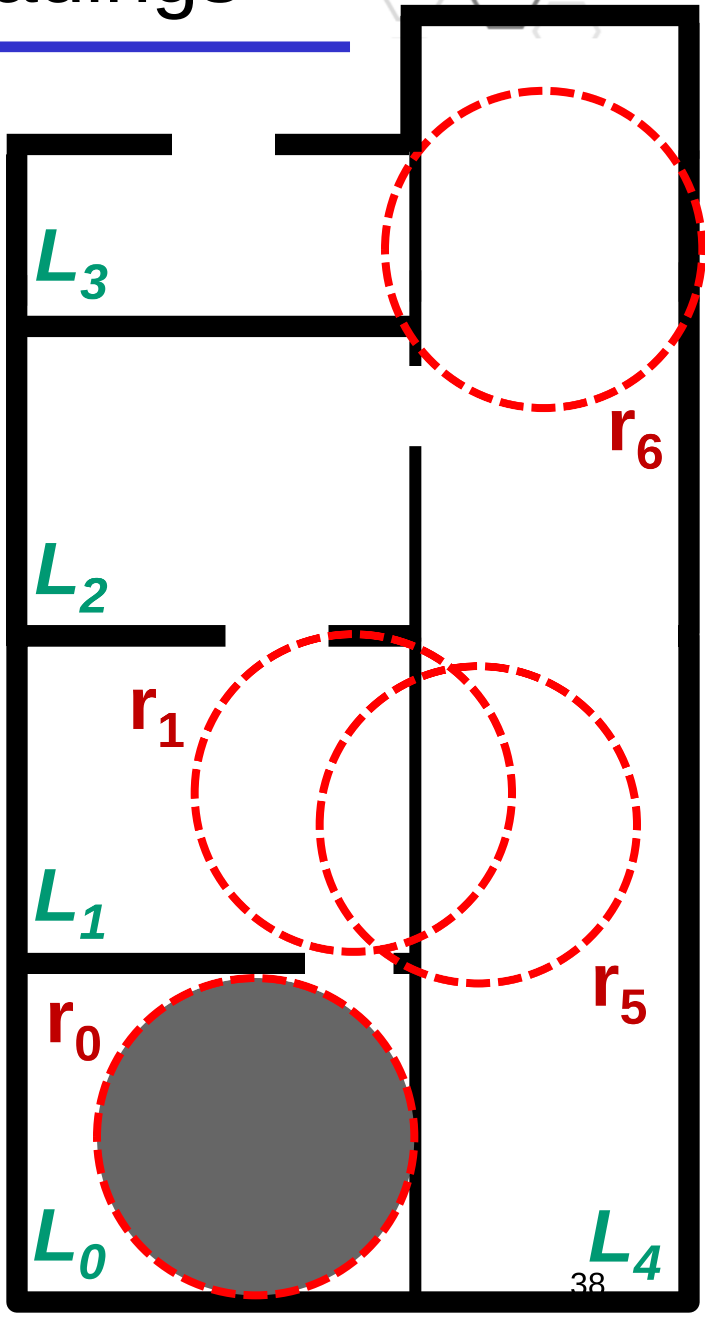
Time	**1s**	**2s**	**3s**	**4s**
Set of readers	{r0}	{ r1, r5}	∅	{r6}
Locations				



A Naive Interpretation of the Readings

Table of detections

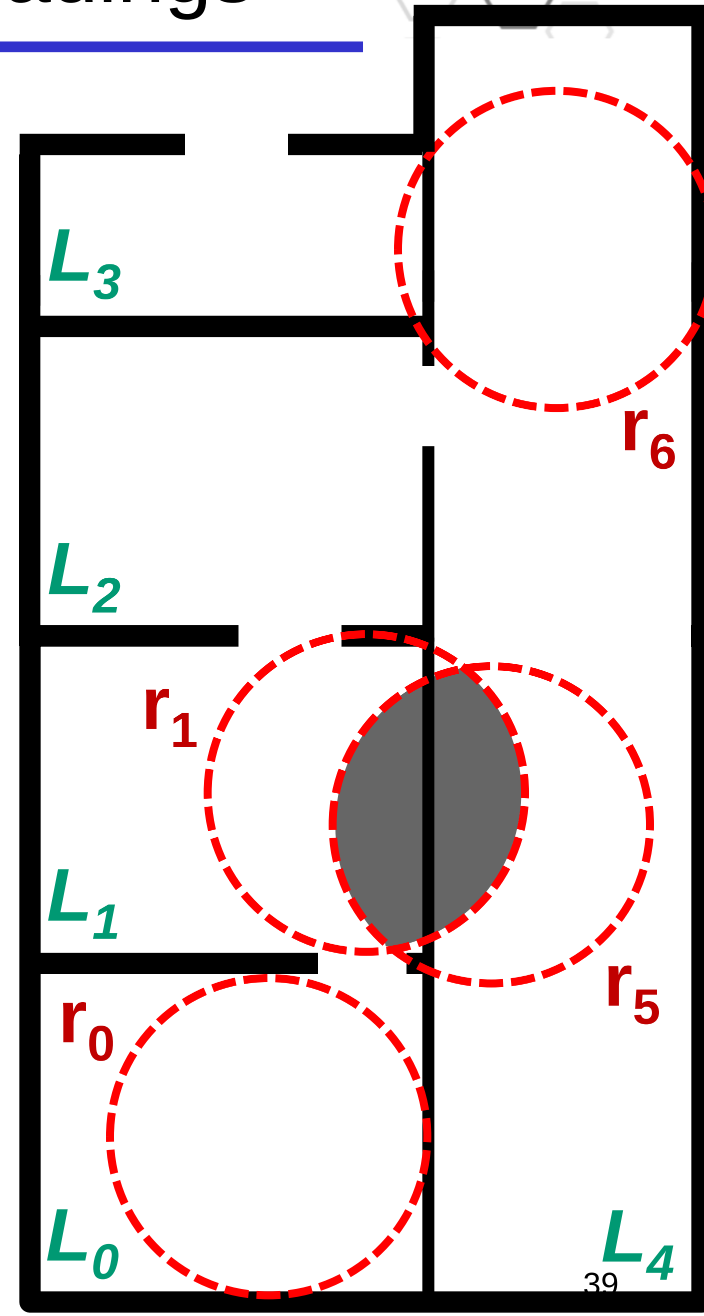
Time	1s	2s	3s	4s
Set of readers	{r0}	{ r1, r5}	∅	{r6}
Locations	L0			



A Naive Interpretation of the Readings

Table of detections

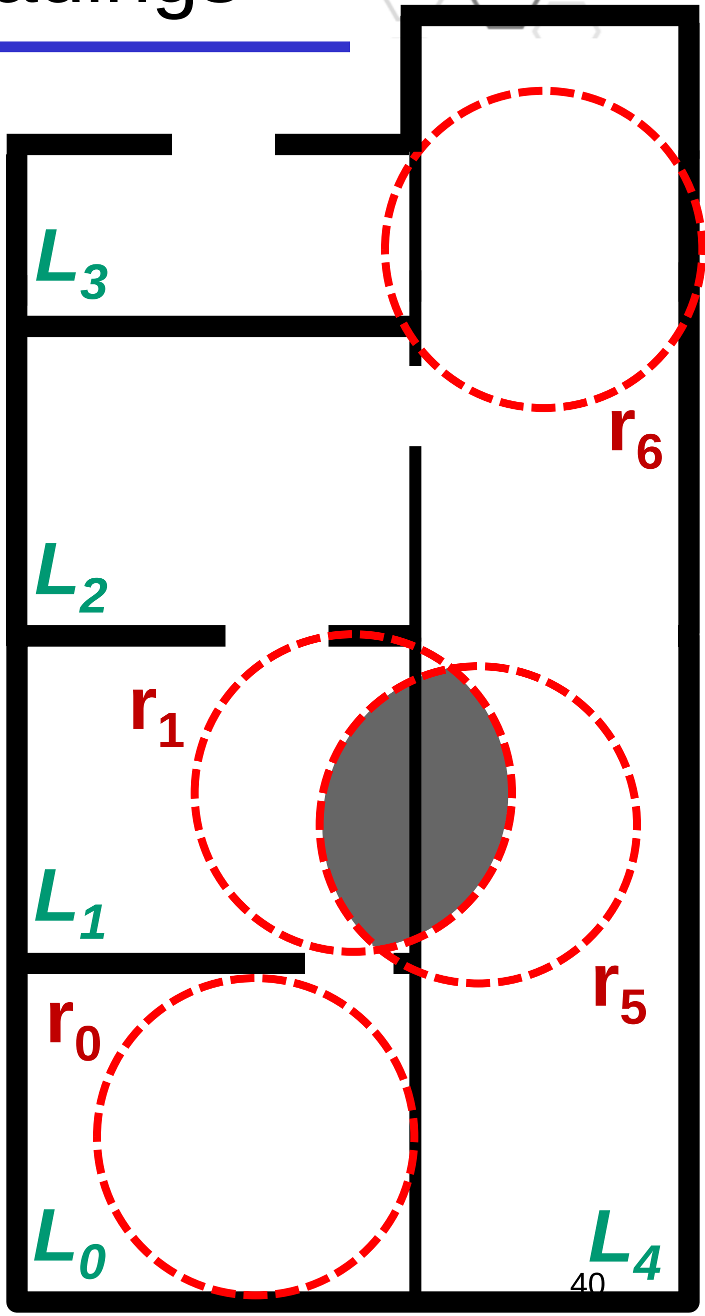
Time	**1s**	**2s**	**3s**	**4s**
Set of readers	{r0}	{ r1, r5}	∅	{r6}
Locations	L0			



A Naive Interpretation of the Readings

Table of detections

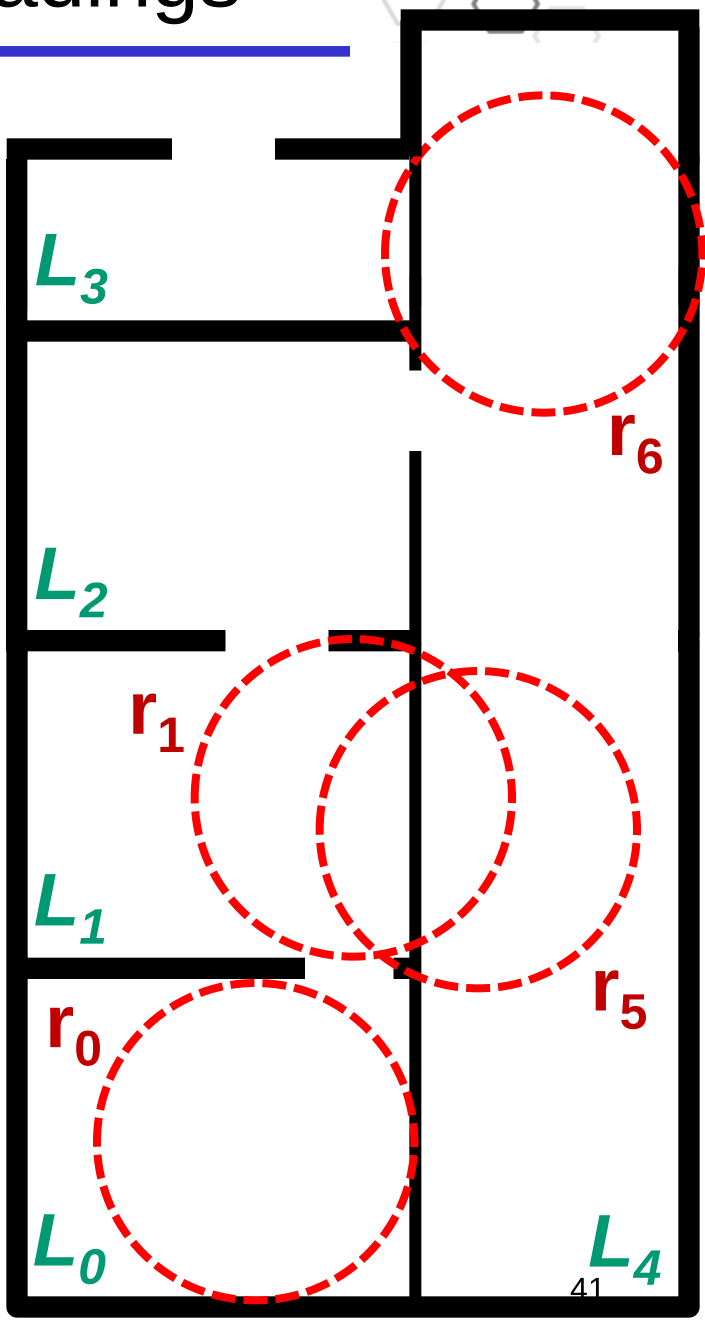
Time	1s	2s	3s	4s
Set of readers	{r0}	{ r1, r5}	∅	{r6}
Locations	L0	L1, L4		



A Naive Interpretation of the Readings

Table of detections

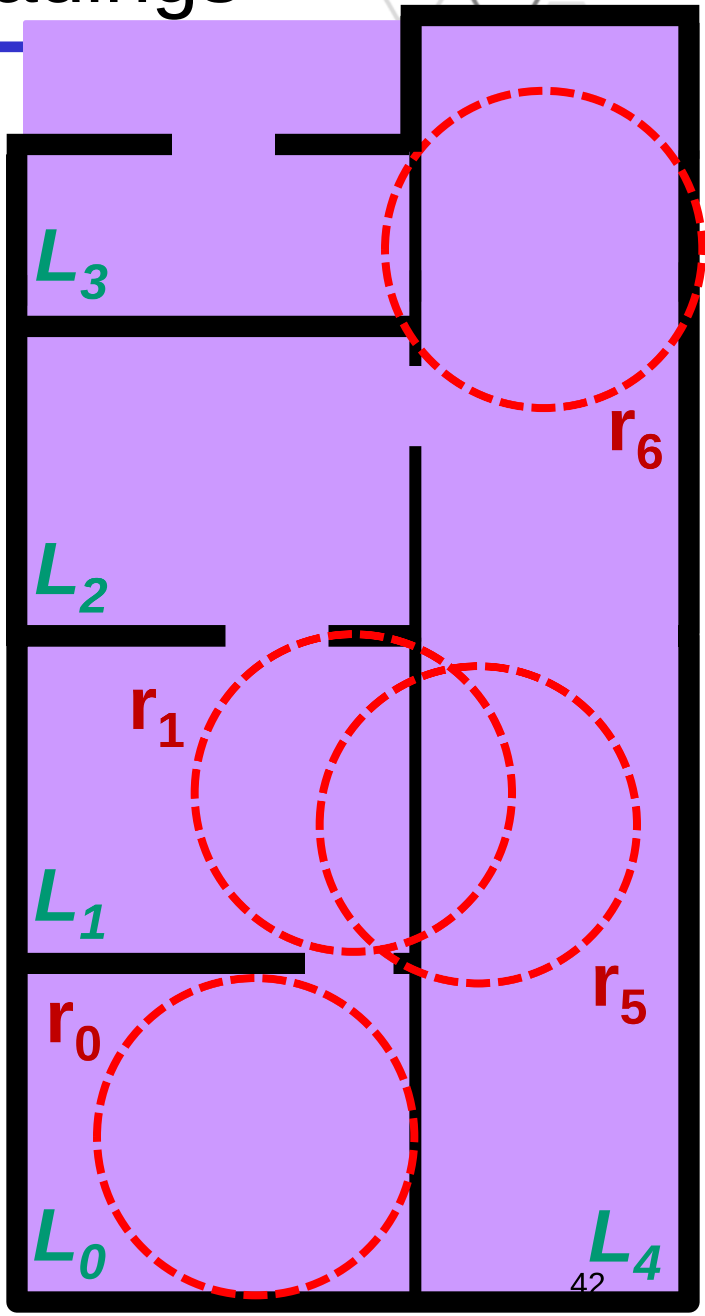
Time	1s	2s	3s	4s
Set of readers	{r0}	{ r1, r5}	∅	{r6}
Locations	L0	L1, L4		



A Naive Interpretation of the Readings

Table of detections

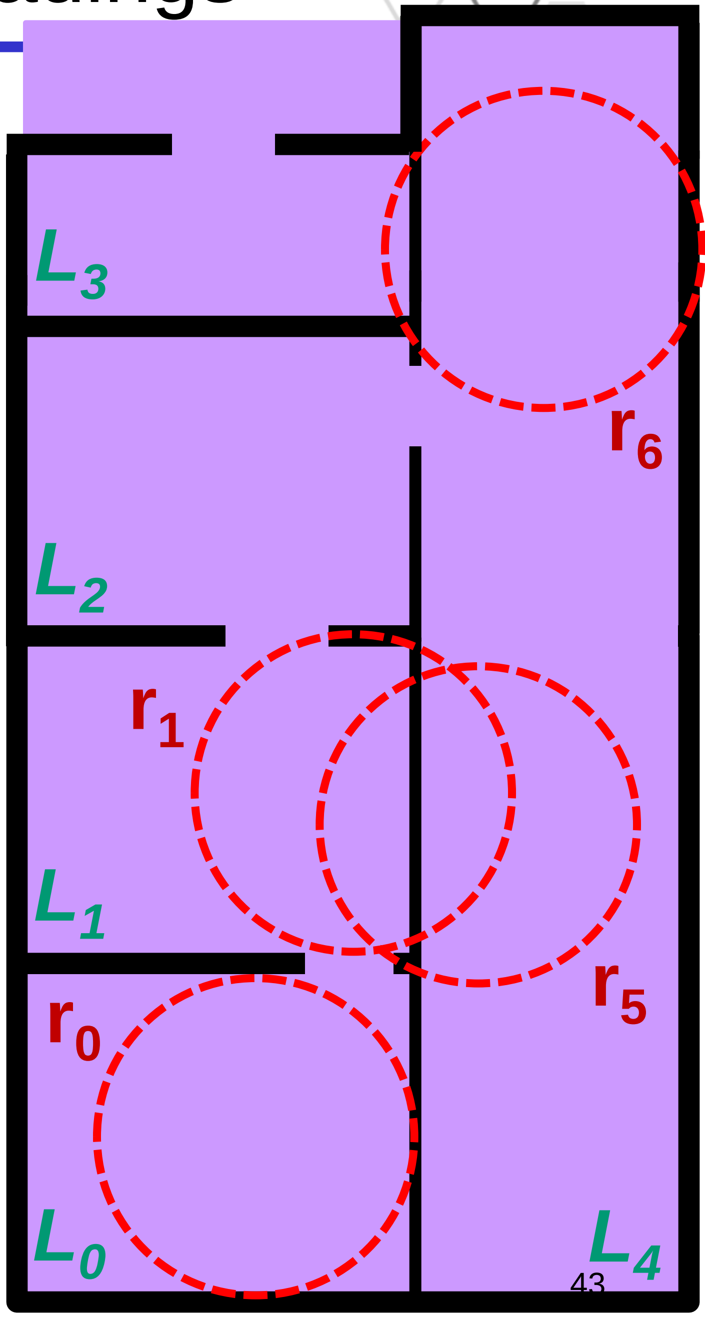
Time	**1s**	**2s**	**3s**	**4s**
Set of readers	{r0}	{ r1, r5}	∅	{r6}
Locations	L0	L1, L4		



A Naive Interpretation of the Readings

Table of detections

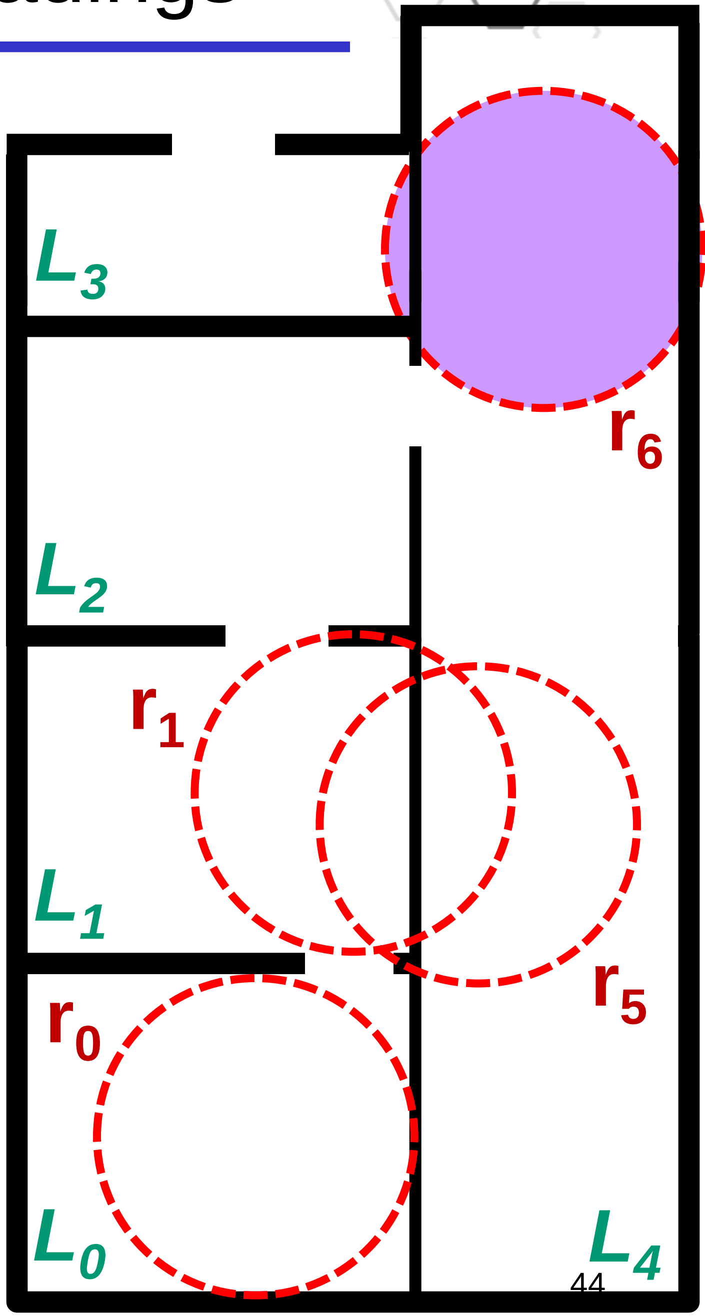
Time	1s	2s	3s	4s
Set of readers	{r0}	{ r1, r5}	∅	{r6}
Locations	L0	L1, L4	L0,L1,L2,L3,L4	



A Naive Interpretation of the Readings

Table of detections

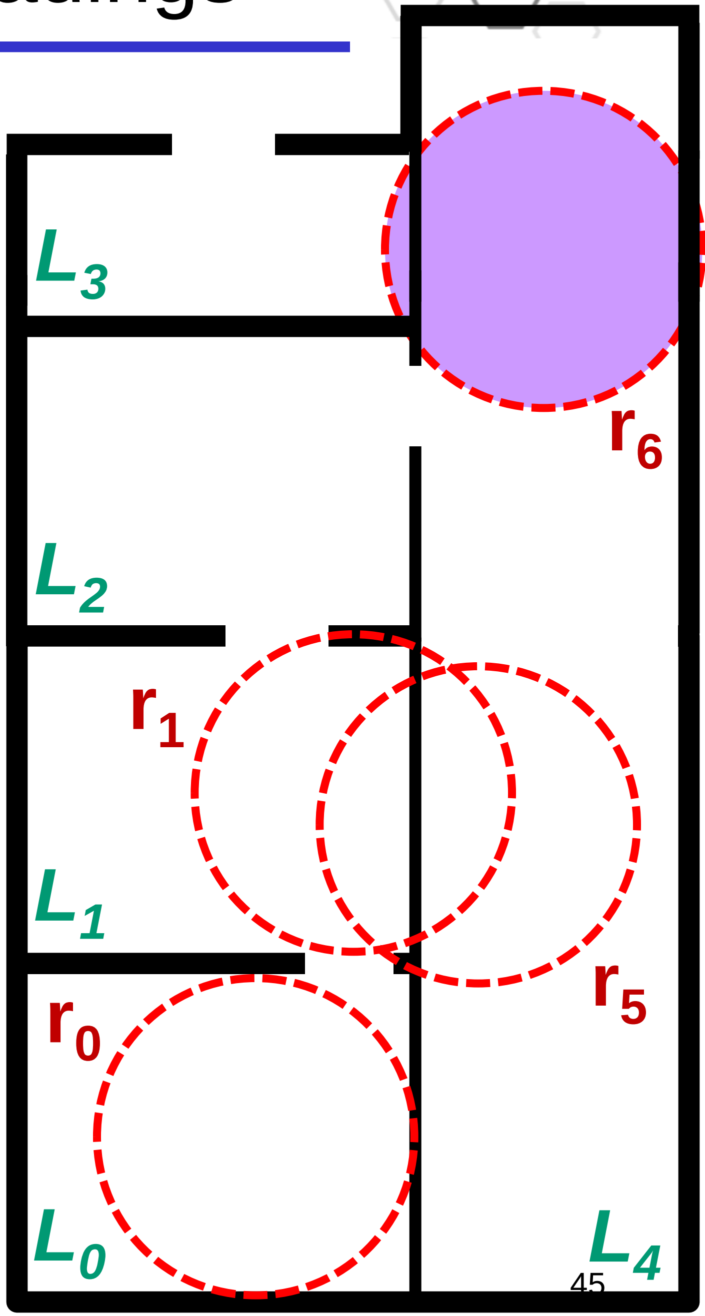
Time	1s	2s	3s	4s
Set of readers	{r0}	{ r1, r5}	∅	{r6}
Locations	L0	L1, L4	L0,L1,L2,L3,L4	



A Naive Interpretation of the Readings

Table of detections

Time	1s	2s	3s	4s
Set of readers	{r0}	{ r1, r5}	∅	{r6}
Locations	L0	L1, L4	L0,L1,L2,L3,L4	L3, L4



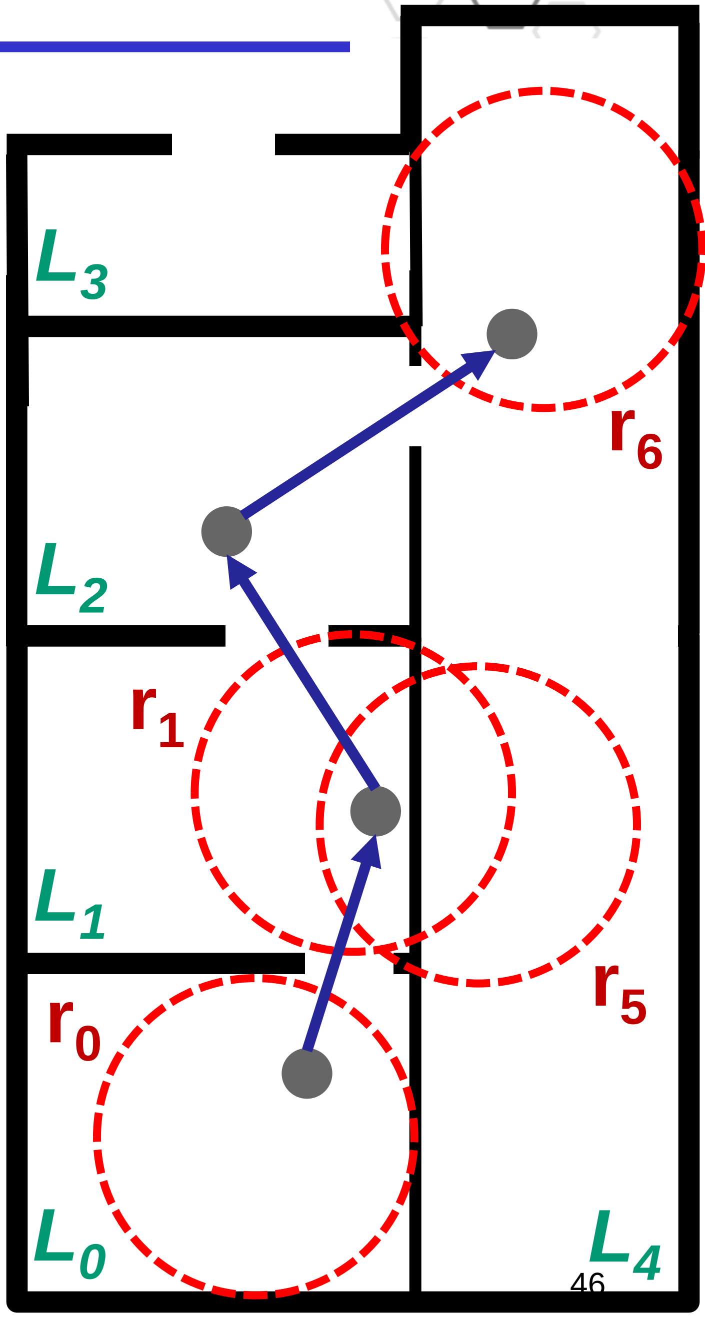
Candidate Trajectories

Table of detections

Time	1s	2s	3s	4s
Set of readers	{r0}	{ r1, r5}	∅	{r6}
Locations	L0	L1, L4	L0,L1,L2,L3,L4	L3, L4

Several candidate trajectories:

t1: L0–L1–L2-L4



Candidate Trajectories

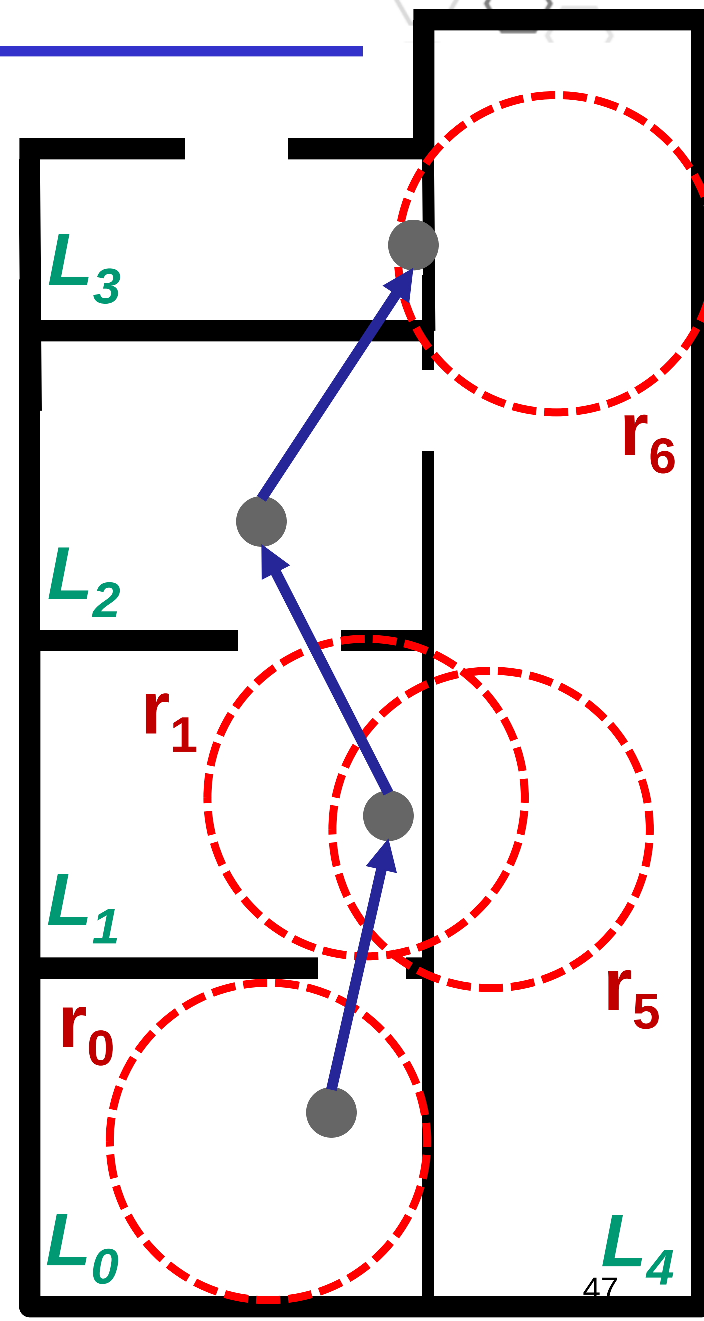
Table of detections

Time	1s	2s	3s	4s
Set of readers	{r0}	{ r1, r5}	∅	{r6}
Location s	L0	L1, L4	L0,L1,L2 ,L3,L4	L3, L4

Several candidate trajectories:

t1: L0–L1–L2-L4

t2: L0–L1–L2-L3



Candidate Trajectories

Table of detections

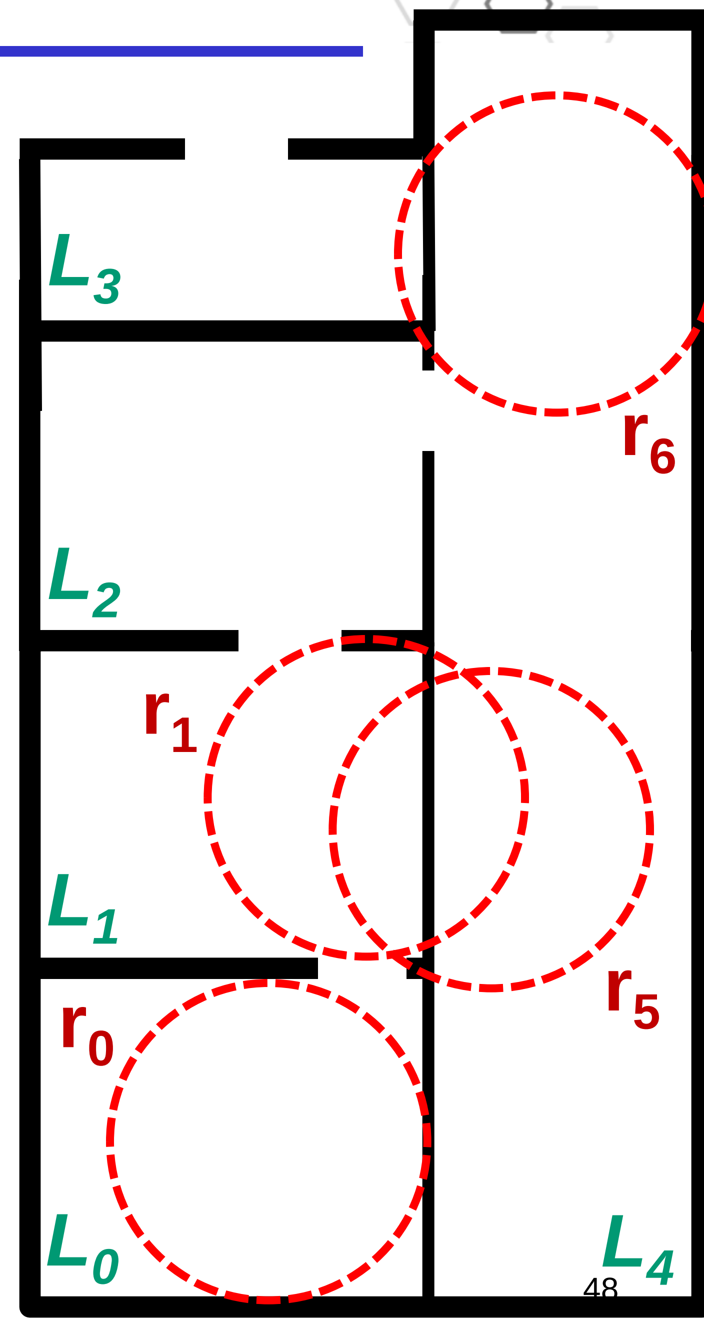
Time	**1s**	**2s**	**3s**	**4s**
Set of readers	{r0}	{ r1, r5}	∅	{r6}
Locations	L0	L1, L4	L0,L1,L2,L3,L4	L3, L4

Several candidate trajectories:

t1: L0–L1–L2-L4

t2: L0–L1–L2-L3

… and so on (1×2×5×2= 20 trajectories!)



Candidate Trajectories

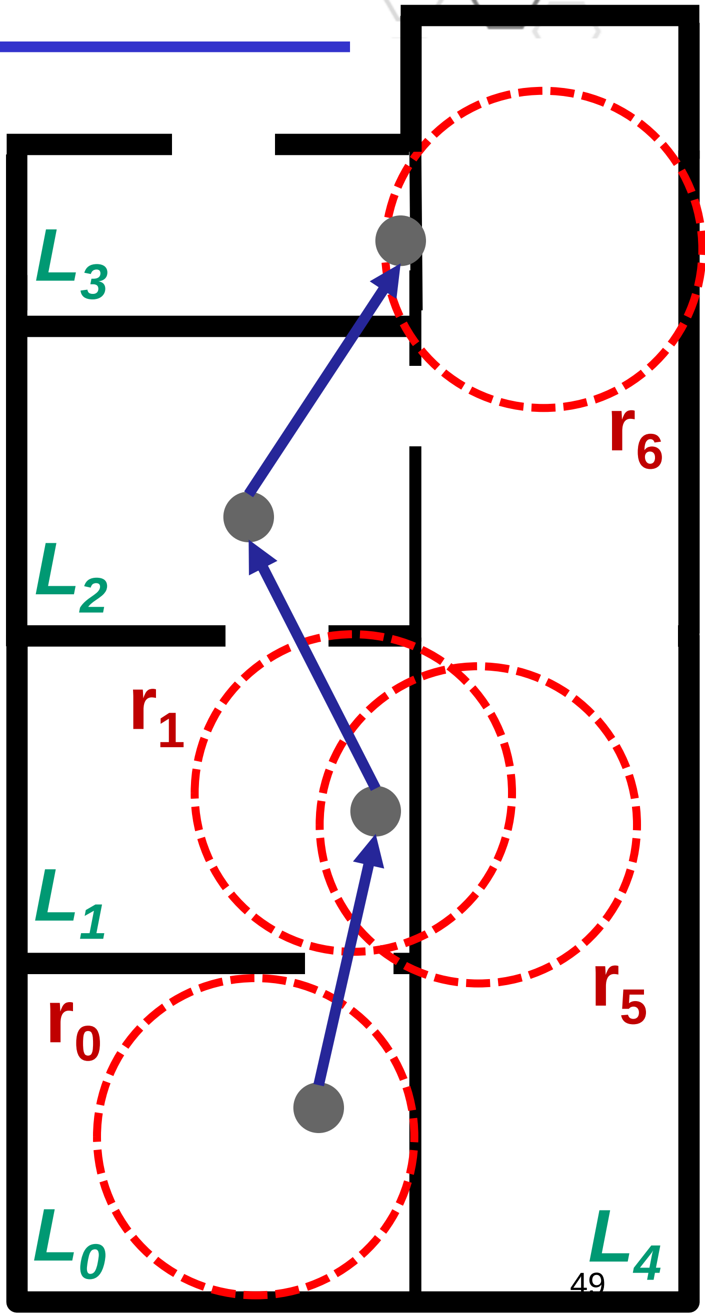
… but some trajectories do not satisfy spatiotemporal constraints

Several candidate trajectories:

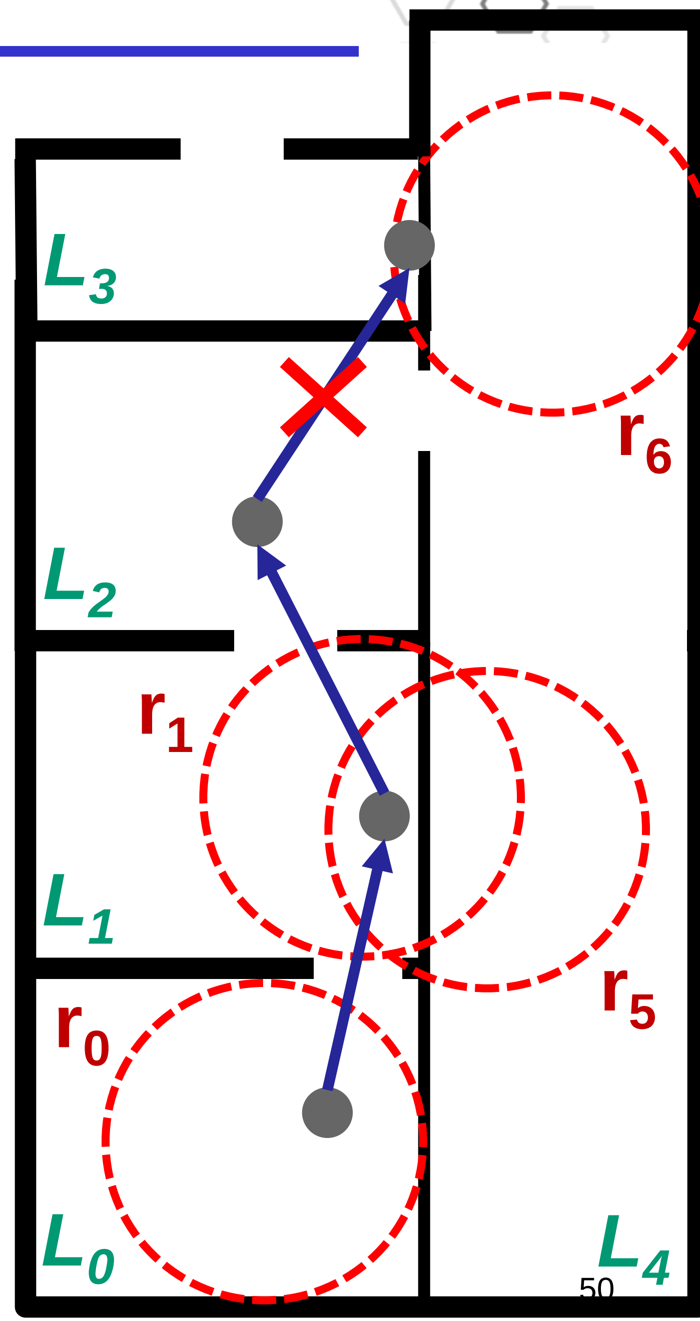
t1: L0–L1–L2-L4

t2: L0–L1–L2-L3

… and so on (1×2×5×2= 20 trajectories!)



Candidate Trajectories



Several candidate trajectories:

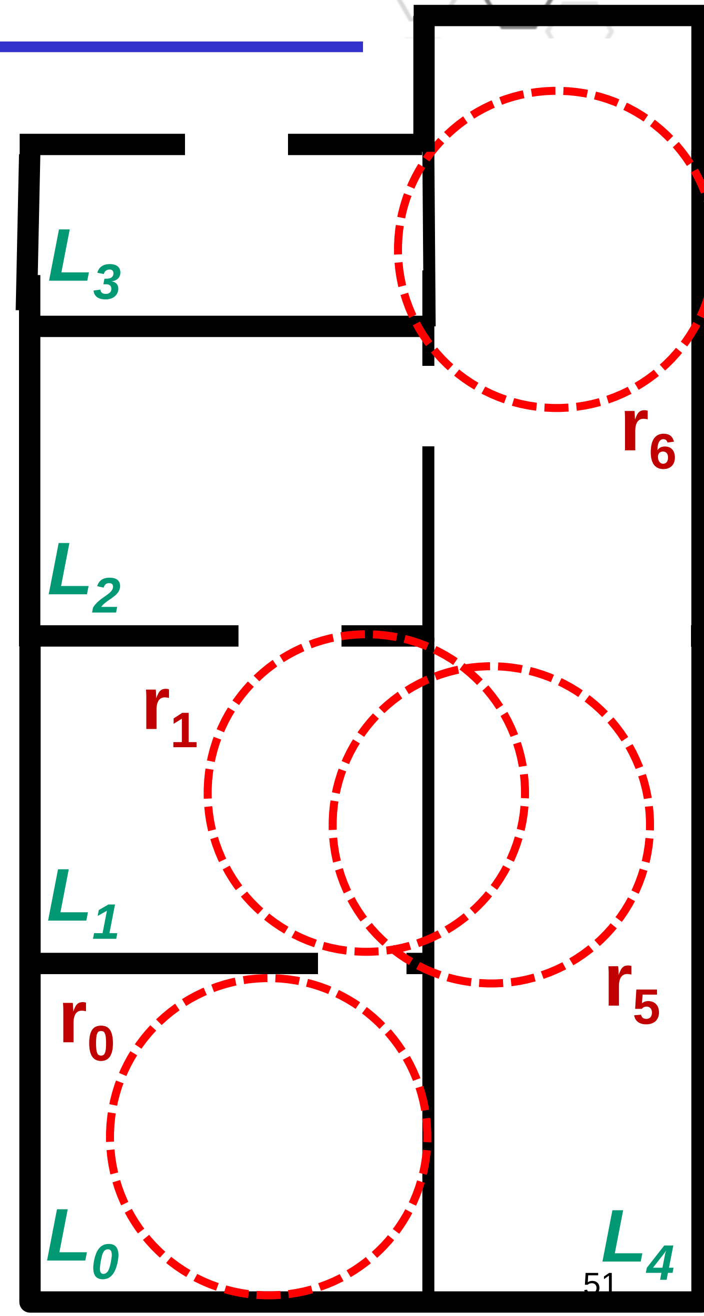
t1: L0–L1–L2-L4

~~***t2:*** L0–L1–L2-L3~~

… and so on (1×2×5×2= 20 trajectories!)

Candidate Trajectories

Disregarding spatio-temporal correlations yields a DIRTY SET of interpretations…



t1: L0–L1–L2-L4

~~***t2:*** L0–L1–L2-L3~~

… and so on (1×2×5×2= 20 trajectories!)

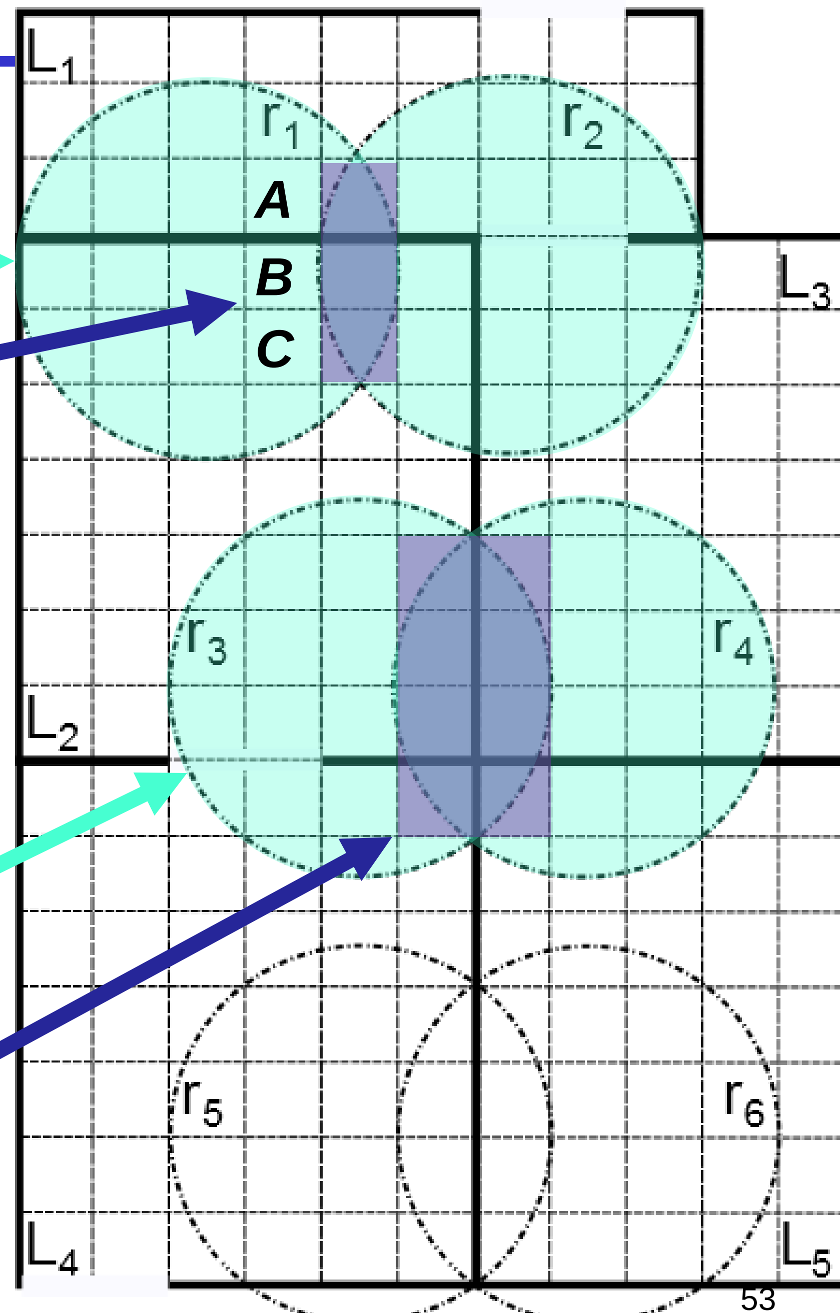
Raw Data to Probabilistic Trajectory

- *No independence assumption: exploits spatio-temporal correlations*
  - *Correlations implied by the map and the maximum speed of tags are considered*
- *Positions are considered at the granularity of cells of a grid over the map*
- *Offline computation (e.g., data for all timestamps is available)*

Candidate Cells

Reading at t1: *{r_1, r_2};*

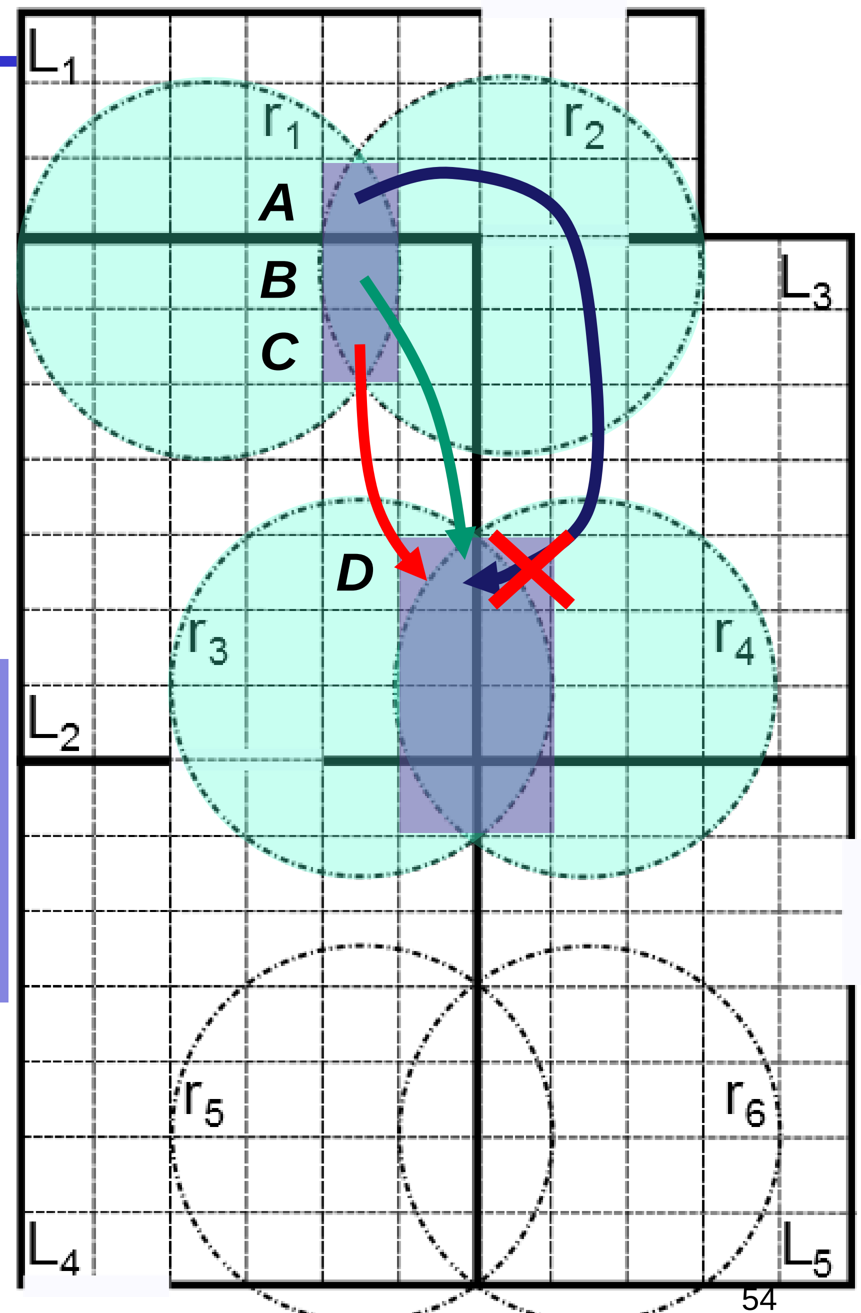
Candidate cells at t1:
C(t1)= {A, B, C}



Reading at time t2:*{ r3, r4};*

Candidate cells at t2:
C(t2)

Candidate Cells



for each candidate cell at time t

compute probability that it is compatible with candidate cells in (t-1) and (t+1)

Approach

TWO-WAY Compatibility Check

- ***Forward probability*** $p^{fw}(t,c)$***:*** *a measure of the compatibility of* ***c*** *with candidate cells of the* ***previous*** *timestamp;*
- ***Backward Probability*** $p^{bw}(t,c)$**:** *a measure of the compatibility of* ***c*** *with candidate cells of the* ***next*** *timestamp*

EXAMPLE

- $p^{fw}(t,c) = 1\%$: *c is hardly reachable from the candidate cells of time point t-1*
- $p^{bw}(t,c)=0$*: no candidate cell of time point t+1 is reachable from c*

Forward (Backward) Probabilities

Let C(i) denote the set of candidate cells at time i. Then:

$$p^{fw}(t,c) = \sum_{c' \in C(t-1)} p^{fw}(t-1, c') \times p_{mov}(c' \to c);$$

For each candidate cell c' at previous timestamp

Forward probability of c'

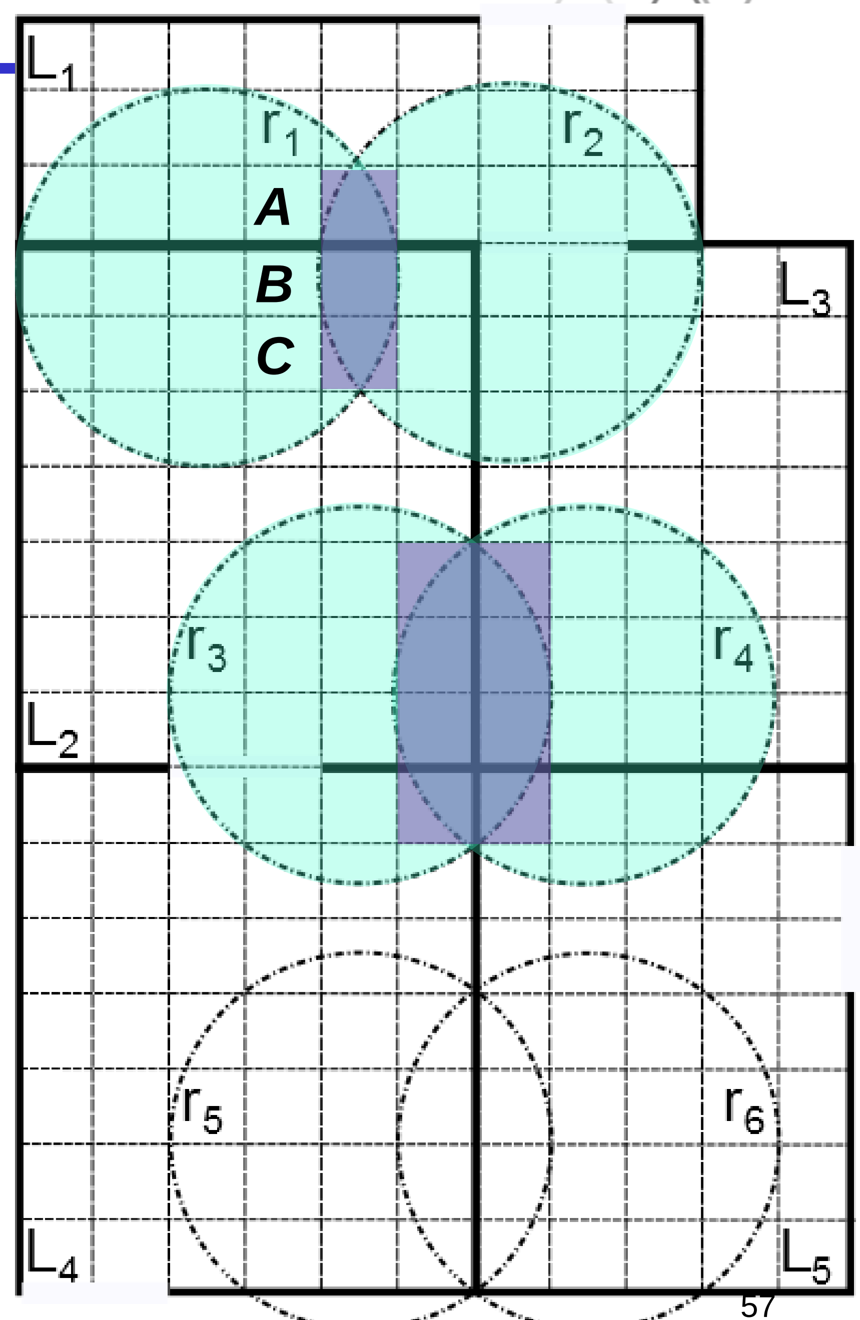
probability that ***o*** could reach ***c*** from **c'** in one time unit

Analogously:

$$p^{bw}(t,c) = \sum_{c' \in C(t+1)} p^{bw}(t+1, c') \times p_{mov}(c \to c');$$

Forward Probabilities

$p^{fw}(A) = p^{fw}(B) = p^{fw}(C) = 1/3$



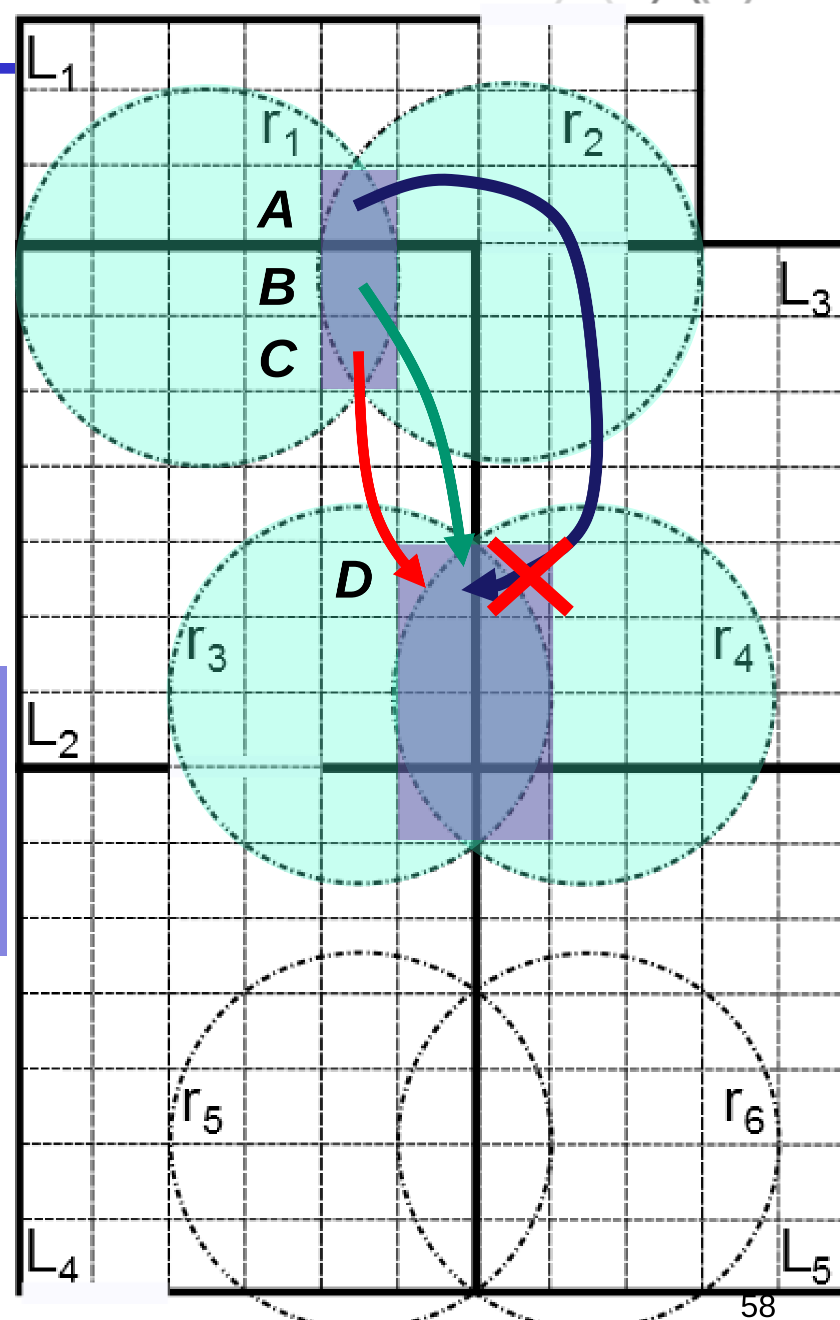
Forward Probabilities

$$p^{fw}(D) = p^{fw}(B) \times p_{mov}(B \rightarrow D) + p^{fw}(C) \times p_{mov}(C \rightarrow D)$$

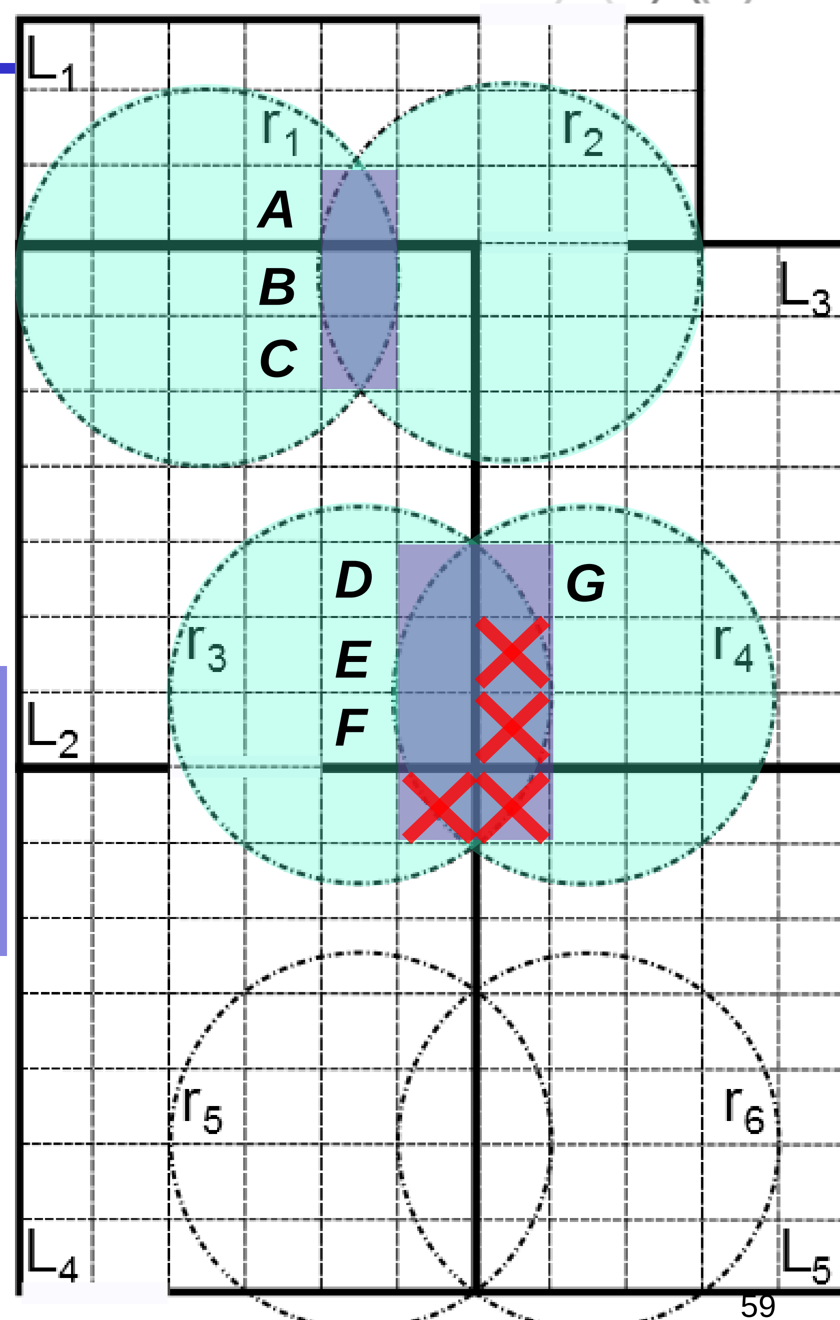
for each $c \in C(t)$

$$p^{fw}(t,c) = \sum_{c' \in C(t-1)} p^{fw}(t-1, c') \times p_{mov}(c' \rightarrow c);$$

$$p_{mov}(A \rightarrow D) = 0$$



Forward Probabilities



for each $c \in C(t)$

$$p^{fw}(t,c)=\sum_{c' \in C(t-1)} p^{fw}(t-1, c')\times p_{mov}(c' \rightarrow c);$$

Two-way Filtering Algorithm

INPUT: $R_1,\ldots,R_T$; **OUTPUT**: $p_1,\ldots,p_T$

1) Forward phase:

for each $t \in [1..T]$

compute $p^{fw}(t,c)$ of each candidate cell and filter the cells with $p^{fw}=0$;

2) Backward phase:

for each $t \in [T..1]$

compute $p^{bw}(t,c)$ of each candidate cell and filter the cells with $p^{bw}=0$;

3) Finale:

for each $t \in [1..T]$, $c \in C(t)$

$$\boxed{p_t(c)=p^{fw}(t,c) \times p^{bw}(t,c) \times h(R_t|c)}$$

Physical model and position of readers!

Outline

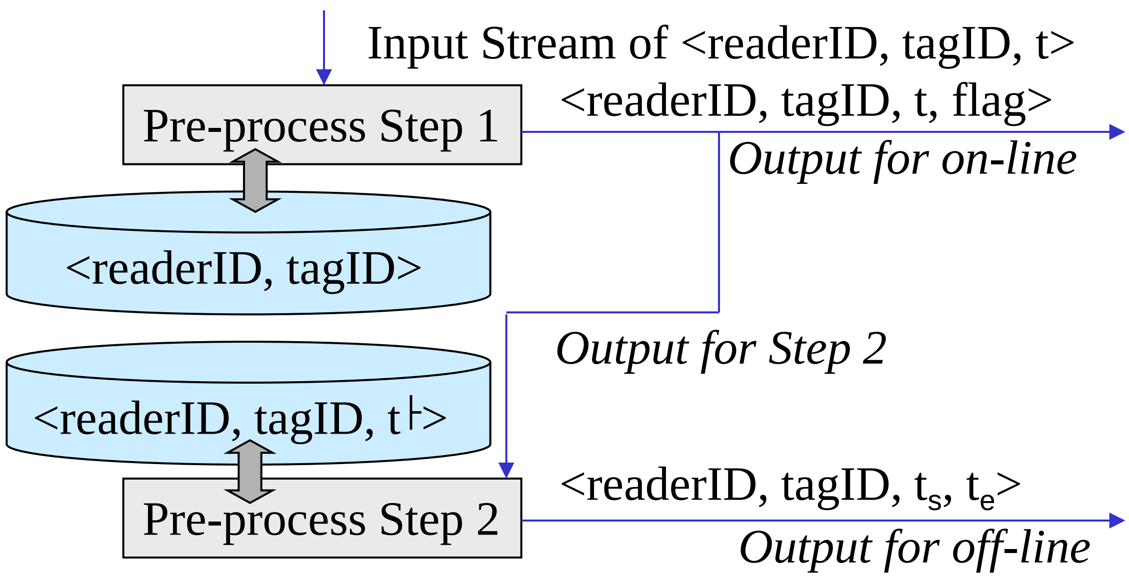
- Introduction, Motivation and Challenges
- Existing Research
  - Data Modeling for Indoor Space
  - Preprocessing Indoor Positioning Data
  - Indexing Indoor Space and Data
  - Querying Indoor Data
  - Other Topics
- Future Research Directions

Indexing Indoor Space and Data

- Indoor tracking [21]
- Indexing symbolic indoor trajectories [22]
- Hashing indoor moving objects [47, 48]
- A composite index (indR-tree) for indoor space and data [45]

Aggregating Raw Readings

- Raw readings
  - (readerID, tagID, t)
- Trajectory records
  - (recordID, tagID, readerID, t_s, t_e)



ENTER: enters the range;

LEAVE: leaves the the range

Graph Model Based Indoor Tracking

- A graph model based indoor tracking
  - A uniform data management infrastructure
  - Supporting a range of indoor positioning technologies like Bluetooth and RFID

- Roadmap

Indoor space floor plan — *Space description*

↓

Base graphs — *Basic topological info.*

Indoor positioning technology: Bluetooth, RFID, Wi-Fi… →

↓

Deployment graph — *Positioning devices info.*

↓

Tracking algorithms — *Both off-line and on-line*

↓

Our goal*: Where (a reduced indoor region) can a particular object be at a particular time?*

Indexing Symbolic Indoor Trajectories

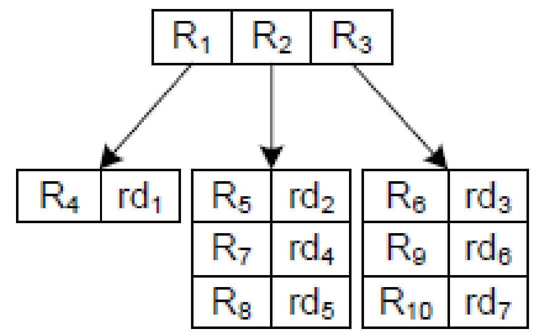
- Raw readings
  - (readerID, tagID, t)
- Trajectory records
  - (recordID, tagID, readerID, t_s, t_e)

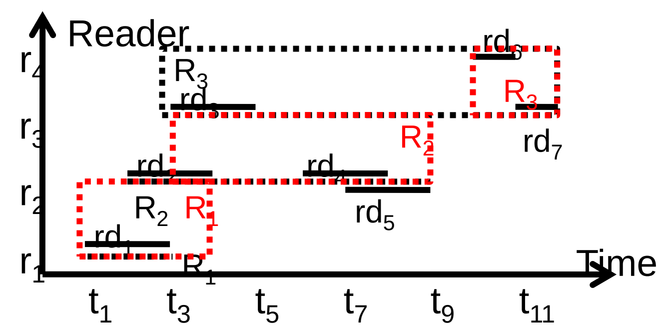
recordID	tagID	readerID	t_s	T_e
rd_1	tag_1	$reader_1$	t_1	t_3
rd_2	tag_3	$reader_2$	t_2	t_4
rd_3	tag_2	$reader_3$	t_3	t_5
rd_4	tag_3	$reader_2$	t_6	t_8
rd_5	tag_2	$reader_2$	t_7	t_9
rd_6	tag_1	$reader_4$	t_{10}	t_{11}
rd_7	tag_3	$reader_3$	t_{11}	t_{12}

- Two R-tree based indexes for processing the following two query types
  - Indoor Spatiotemporal Range Query
    - $Q(E_s, E_t) \rightarrow$ {trajectory records}
    - E.g., Q ($room_1$, [1:00 p.m., 1:15 p.m.])
  - Indoor Topological Query
    - $Q(E_s, E_t, P) \rightarrow$ {objectID}
    - P denotes a topological predicate, such as *enter*, *leave* and *cross*.
    - E.g., Q ($room_1$, [1:00 p.m., 1:15 p.m.], *enter*)

RTR-tree: Reader-Time R-tree

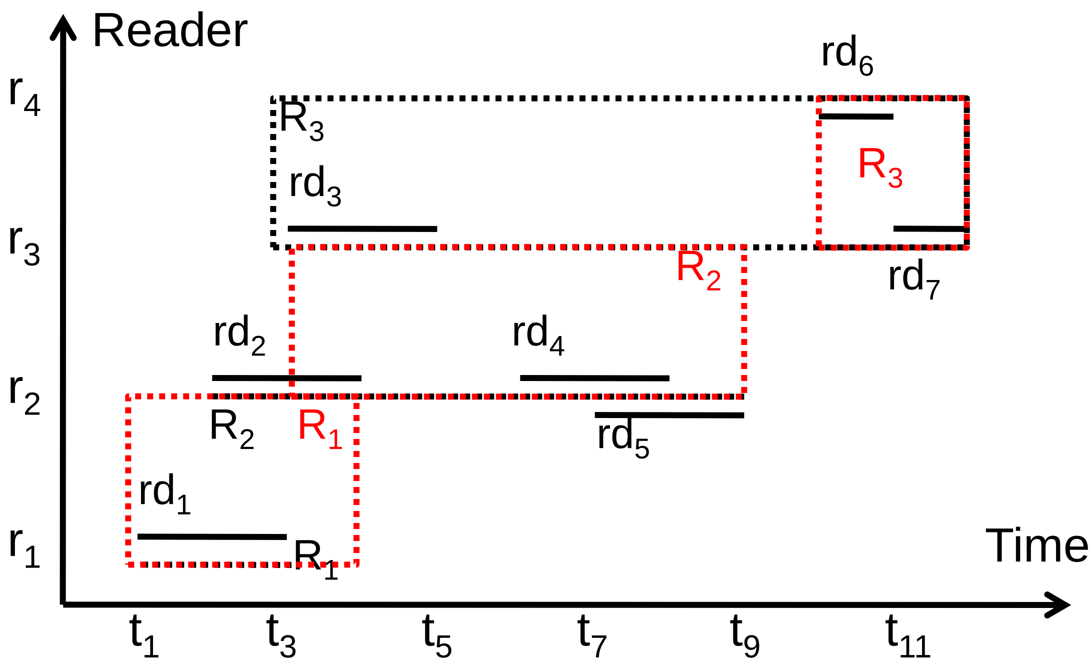
- Two dimensional R-tree in Reader-Time space
  - Vertical axis: reader IDs
  - Horizontal axis: timestamps
- Trajectory record representation
  - Horizontal line segment
- Leaf node entries:
  - (MBR, recordID)
  - MBR is a horizontal line segment: (readerID, t_s, t_e)
- Non-leaf node entries:
  - (MBR, cp)
  - MBR is a rectangle: ($readerID_{min}$, $readerID_{max}$, $t^{\vdash}$, $t^{\dashv}$)





Node Organization Strategies

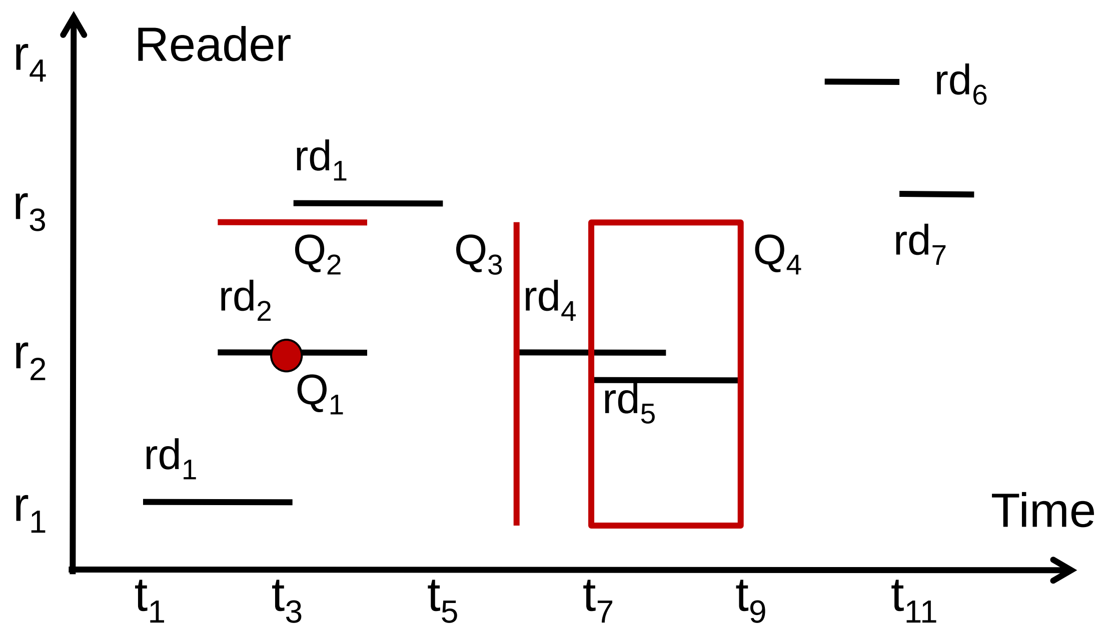
- Classic area formula:
  - Area = (readerID$_{max}$ - readerID$_{min}$) * (t┤ - t├)
  - E.g., Area(rd$_1$) = 0
- Area$^+$ formula: the number of possible raw readings
  - Area$^+$ = (readerID$_{max}$ - readerID$_{min}$ +1) * ((t┤ - t├)/T$_s$+1)
  - E.g., Area$^+$(rd$_1$) = 3



ID	tagID	readerID	t_s	t_e
rd_1	tag_1	$reader_1$	t_1	t_3
rd_2	tag_3	$reader_2$	t_2	t_4
rd_3	tag_2	$reader_3$	t_3	t_5
rd_4	tag_3	$reader_2$	t_6	t_8
rd_5	tag_2	$reader_2$	t_7	t_9
rd_6	tag_1	$reader_4$	t_{10}	t_{11}
rd_7	tag_3	$reader_3$	t_{11}	t_{12}

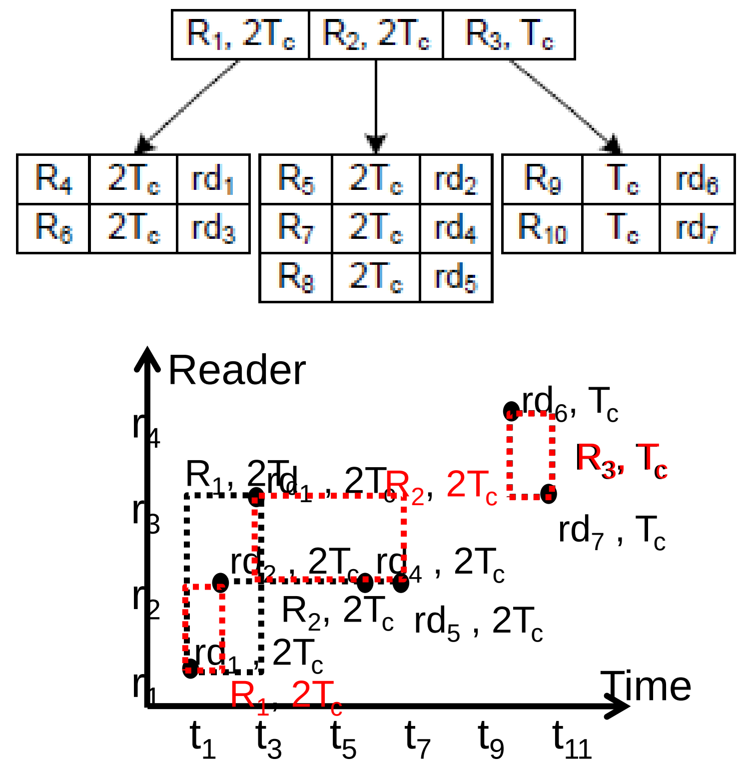
Query Processing on RTR-tree

Time		Single ReaderID	Continuous ReaderIDs
Instant	Query Format	$QT_1(readerID; t)$	$QT_3([readerID_m; readerID_n]; t)$
	Geometry Representation	Point	Vertical line segment
	Example	$Q_1(reader_2; t_3)$	$Q_3([reader_1; reader_3]; t_6)$
Interval	Query Format	$QT_2(readerID; [t_i; t_j])$	$QT_4([readerID_m; readerID_n]; [t_i; t_j])$
	Geometry Representation	Horizontal line segment	Rectangle
	Example	$Q_2(reader_3; [t_2; t_4])$	$Q_4([reader_1; reader_3]; [t_7; t_9])$



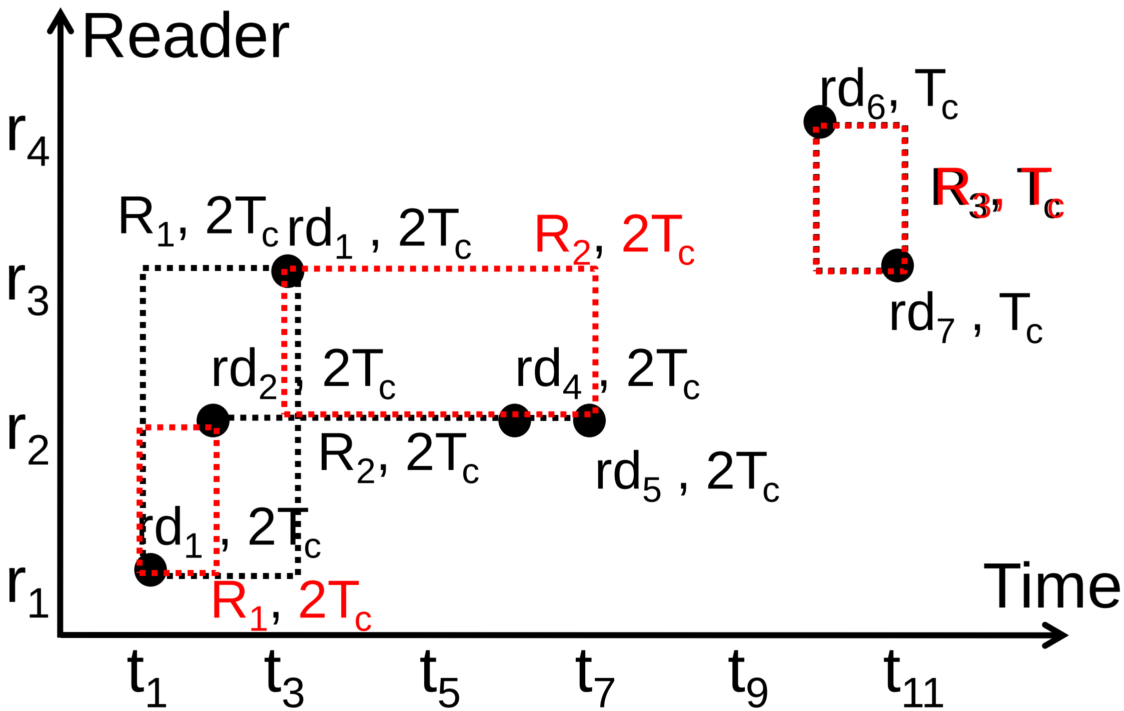
TP²R-tree: Time Parameter Point R-tree

- Trajectory record representation
  - Point + time parameter Δt
- Leaf node entries:
  - (MBR, Δt, recordID)
  - MBR is a point: (readerID, t_s)
  - $\Delta t = t_e - t_s$
- Non-leaf node entries:
  - (MBR, Δt, cp)
  - MBR is a rectangle: (readerID$_{min}$, readerID$_{max}$, $t^{\vdash}$, $t^{\dashv}$)
    - If cp points a leaf node:
      - $\Delta t = \max\limits_{\forall e_i \in N_l}(e_i.MBR.t_s + e_i.\Delta t) - \max\limits_{\forall e_j \in N_l}(e_j.MBR.t_s)$
    - If cp points a non leaf node:
      - $\Delta t = \max\limits_{\forall e_i \in N_n}(e_i.MBR..t^{\dashv} + e_i.\Delta t) - \max\limits_{\forall e_j \in N_n}(e_j.MBR..t^{\dashv})$



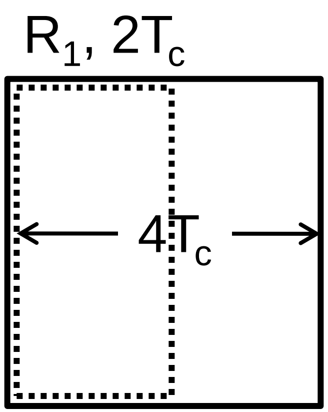
Node Organization Strategies

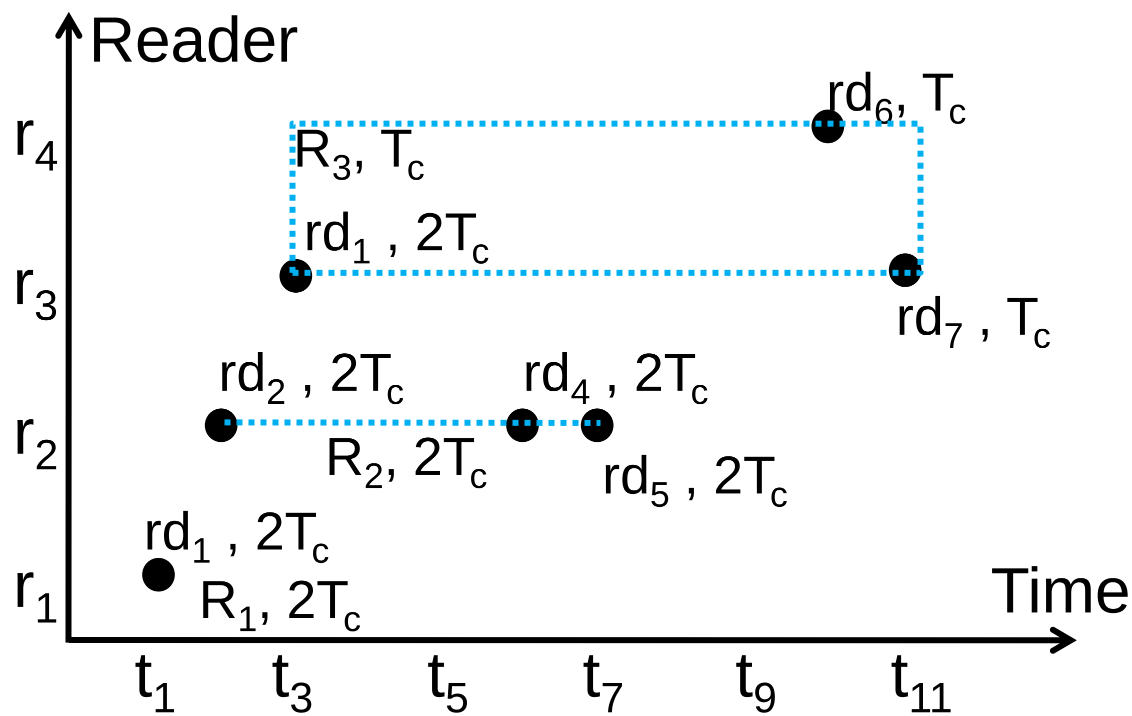
- Classical area formula
- Area$^+$ formula



Node Organization Strategies, cont.

- Split2 strategy
  - Least Reader dimension enlargement
  - Least enlargement of Virtual MBR (VMBR) Area+
    - VMBR is MBR extending with Δt on the time dimension

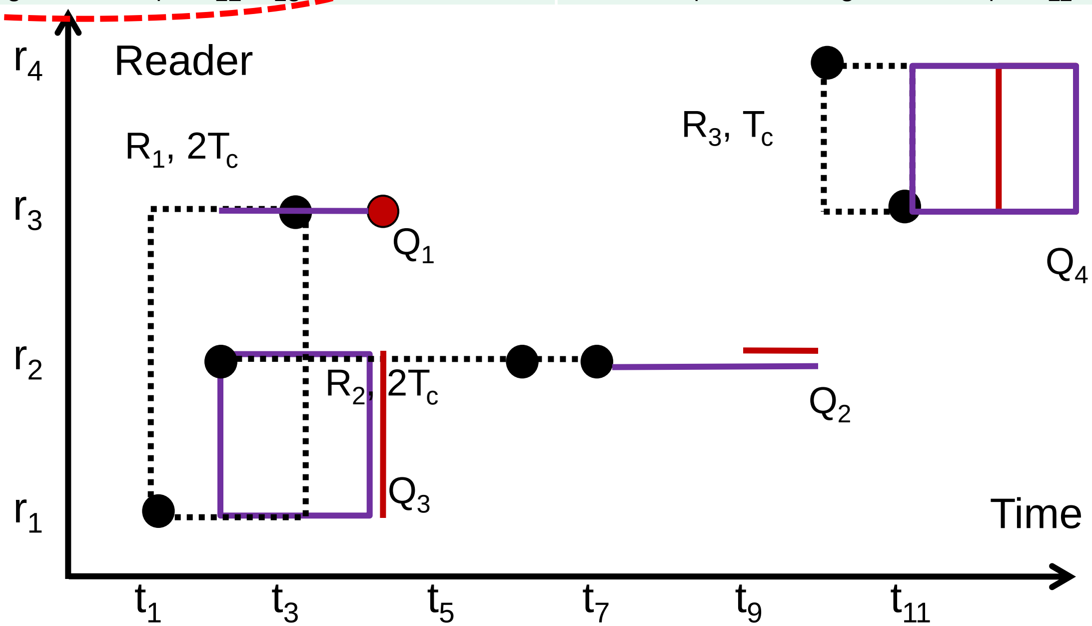




Query processing on TP^{2}R-tree

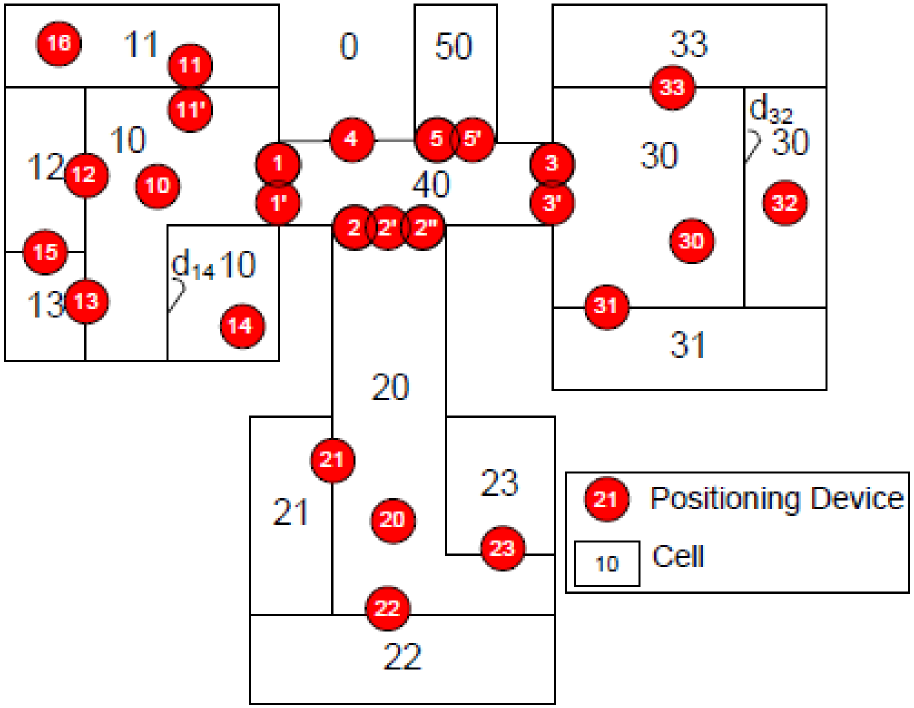
- Query geometry needs expansion in query processing

Original Query	Expanded Query
Type 1: QT_1(readerID; t) E.g., $Q_1(reader_3, t_4)$	*Type 2: QT_2(readerID;* [t-Δt, t]) E.g., $Q_1'(reader_3, [t_2, t_4])$
Type 2: QT_2(readerID; $[t_i; t_j]$) E.g., $Q_2(reader_2, [t_9, t_{10}])$	*Type 2: QT_2(readerID;* $[t_i - \Delta t; t_j]$) E.g., $Q_2'(reader_2, [t_7, t_{10}])$
Type 3: QT_3([readerID$_m$, readerID$_n$]; t) E.g., $Q_3([reader_1, reader_2], t_4)$	*Type 4: QT_4([readerID$_m$, readerID$_n$];* [t-Δt; *t*]) E.g., $Q_3'([reader_1, reader_2], [t_2, t_4])$
Type 4: QT_4([readerID$_m$, readerID$_n$]; $[t_i; t_j]$) E.g., $Q_4([reader_3, reader_4], [t_{12}, t_{13}])$	*Type 4: QT_4([readerID$_m$, readerID$_n$];* $[t_i - \Delta t; t_j]$) E.g., $Q_4'([reader_3, reader_4], [t_{11}, t_{13}])$



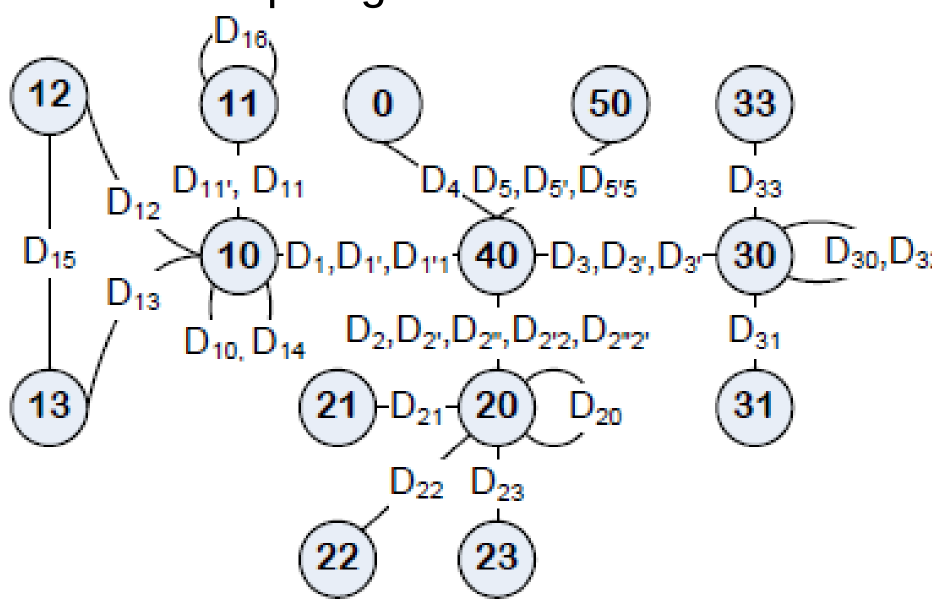
Deployment Graph

- Readers Deployment



- Partitioning Readers partition the indoor space into **cells**.
  - Directed partitioning reader (DP), e.g., readers 11 and 11'
  - Undirected partitioning reader (UP), e.g., reader 21
- Presence Readers (PR), e.g., reader 10

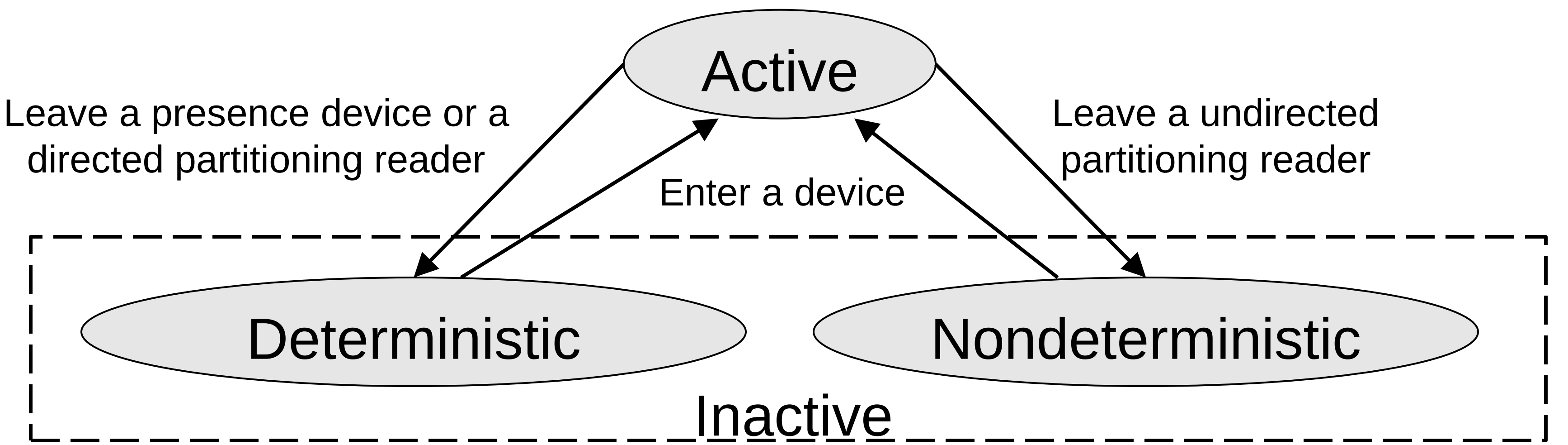
- Deployment Graph $G = \{C, E, \Sigma_{devicse}, l_E\}$
  - C: each cell is a vertex
  - E: an edge indicates that its two cells are connected.
  - l_E maps an edge to a subset of positioning readers in $\Sigma_{devices}$.
    - A non-loop edge: UPs and DPs
    - A loop edge: PRs



Devices: $\Sigma_{devices} \rightarrow \{(Range,\ 2^{\Sigma rooms},\ TYPE)\}$

Hashing Indoor Moving Objects

- We differentiate two states of indoor moving objects
  - Active objects: those objects that are currently seen by at least one positioning device.
  - Inactive objects: those objects that are currently not seen by any positioning device. They can be further differentiated
    - Deterministic objects: Must be in one specific cell.
    - Nondeterministic objects: May be in more than one cell.
- Accordingly, all objects are partitioned and indexed in different hash tables.

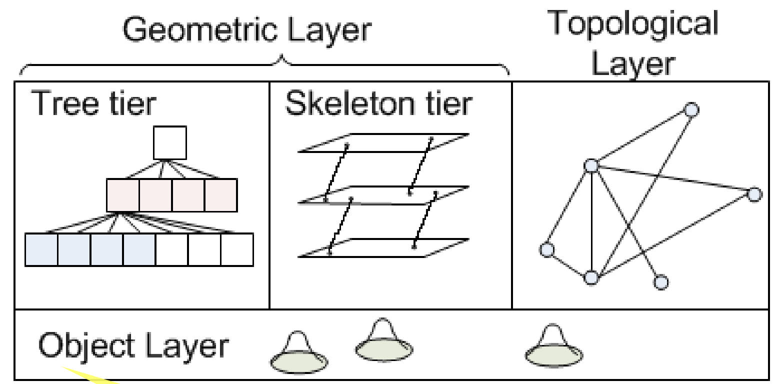


Hash Tables for Indoor Moving Objects

- Device Hash Table (DHT)
  - {deviceID} → {active objectID}
- Cell Deterministic Hash Table (CDHT)
  - {cellID} → {deterministic objectID}
- Cell Nondeterministic Hash Table (CNHT)
  - {cellID} → {nondeterministic objectID}
- Object Hash Table (OHT)
  - {objectID} → {(STATE, t, IDSet)}
    - STATE = active: IDSet contains relevant device identifiers
    - STATE = deterministic: IDSet contains one cell identifier
    - STATE = nondeterministic: IDSet contains a set of cell identifiers
- For each record from pre-processing output, these four hash tables need updating accordingly
  - The Deployment Graph is used to facilitate updates

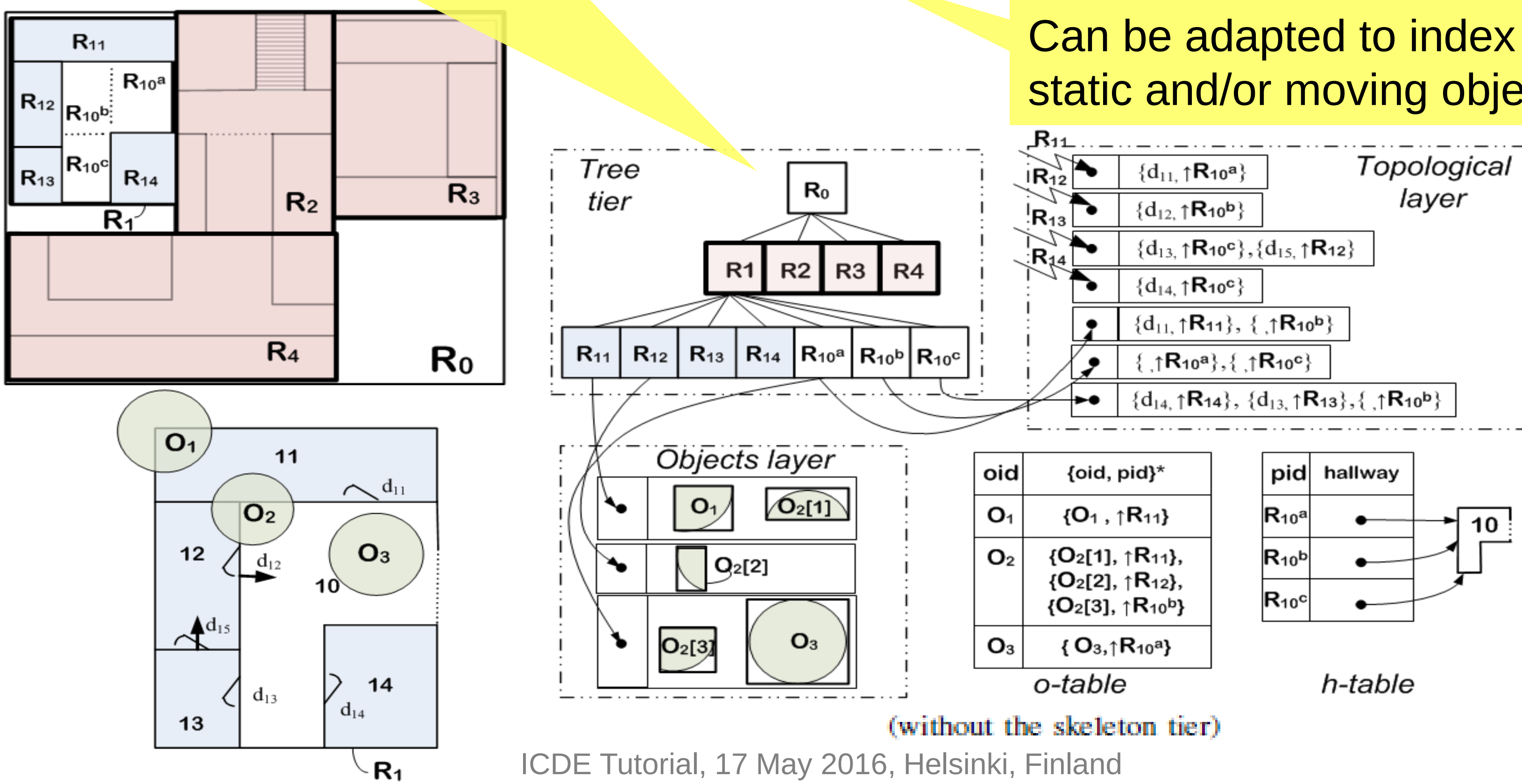
A Composite, Layered Index

- Layers and Tiers
  - divide the problem and localize changes



Query processing: Search the index with effective pruning bounds

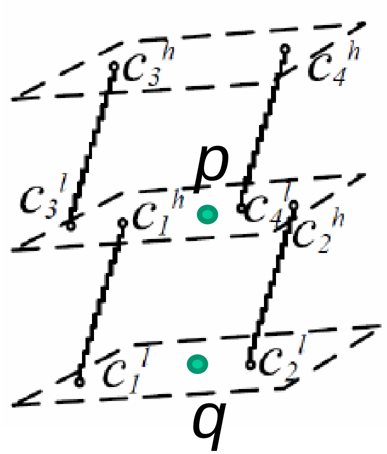
Can be adapted to index static and/or moving objects.

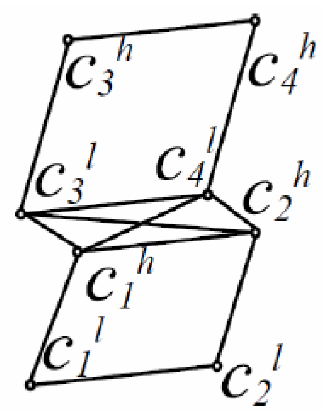


Skeleton Tier

- Skeleton tier handles the different floors
- It maintains a small number of distances between staircases.

- Skeleton distance: The distance between adjacent floors
- Geometric Lower Bound Property
  - Help to prune search space using distance in query processing
  - E.g., the indoor distance between p and q must be larger than the skeleton distance between their corresponding floors.





Outline

- Introduction, Motivation and Challenges
- Existing Research
  - Data Modeling for Indoor Space
  - Preprocessing Indoor Positioning Data
  - Indexing Indoor Space and Data
  - Querying Indoor Data
  - Other Topics
- Future Research Directions

Query Types on Indoor Data

- A non-exhaustive categorization

Data / Query	Static	Moving - Online	Moving - Historical
Static	Range [34], kNN [34]	Range [45], kNN [45, 48, 50], Continuous range [47]	Spatiotemporal range query [22], Topology query [22]
Moving	Continuous range [53]	Joins [46]	Joins [36]

Introduced already

- Indoor moving data type
  - RFID-like (proximity analysis): [22, 36, 47, 48, 50]
  - Probabilistic samples: [45, 46]
- Each of these queries requires a corresponding index.

Finding Static Indoor POIs

- Indoor spatial queries
  - Range query
    - Position *q*, distance range *r*
  - Nearest Neighbor query
    - Position *q*
- Indexing indoor objects
  - Store objects within each partition in a bucket
  - Door-to-Partition Table (DPT) maps a door to two relevant buckets
  - Indoor Distance-Aware Indexes (next slide)
- Query processing outline
  - Search relevant partitions via topology mappings, Distance Index Matrix, and DPT, giving priority to nearer doors and partitions
  - Stop when the distance from *q* to the current partition is larger than *r* or the current nearest neighbor distance

Indoor Distance-Aware Indexes

- Door-to-Door Distance Matrix

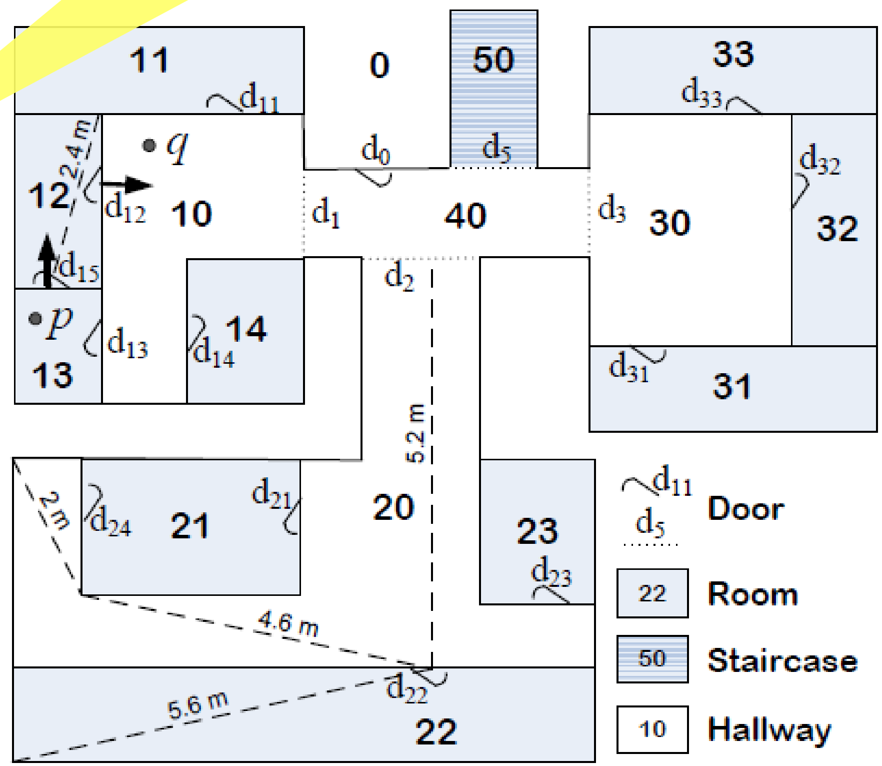
$$
\begin{array}{c}
 \\ d_1 \\ d_{11} \\ d_{12} \\ d_{13} \\ d_{14} \\ d_{15}
\end{array}
\begin{array}{cccccc}
d_1 & d_{11} & d_{12} & d_{13} & d_{14} & d_{15}
\end{array}
$$

$$
\begin{array}{c|cccccc}
 & d_1 & d_{11} & d_{12} & d_{13} & d_{14} & d_{15} \\
\hline
d_1 & 0 & 1.7 & 2.7 & 3.2 & 2.6 & 4.3 \\
d_{11} & 1.7 & 0 & 1.9 & 3.4 & 3 & 4.4 \\
d_{12} & 2.7 & 1.9 & 0 & 2 & 2.2 & 3 \\
d_{13} & 3.2 & 3.4 & 2 & 0 & 1.2 & 1 \\
d_{14} & 2.6 & 3 & 2.2 & 1.2 & 0 & 2.2 \\
d_{15} & 3.2 & 3.4 & 1.5 & 3.5 & 3.7 & 0
\end{array}
$$

Precomputed indoor distances, allowing pruning and prioritizing in search

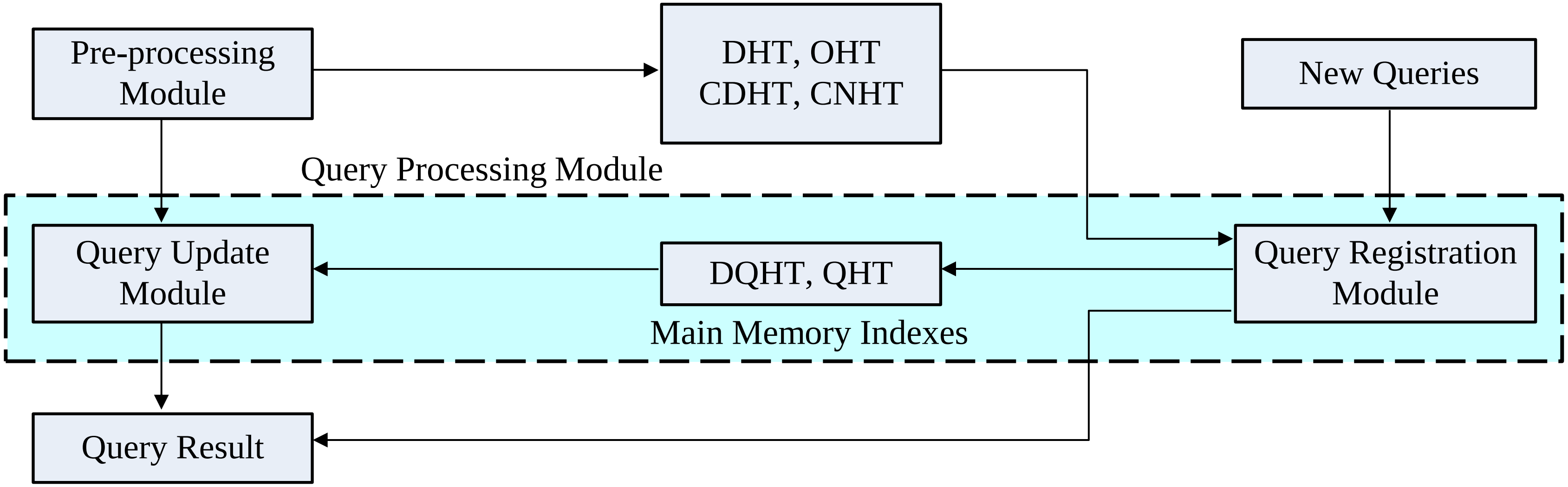
- Distance Index Matrix

$$
\begin{array}{c|cccccc}
 & 1 & 2 & 3 & 4 & 5 & 6 \\
\hline
d_1 & d_1 & d_{11} & d_{14} & d_{12} & d_{13} & d_{15} \\
d_{11} & d_{11} & d_1 & d_{12} & d_{14} & d_{13} & d_{15} \\
d_{12} & d_{12} & d_{11} & d_{13} & d_{14} & d_1 & d_{15} \\
d_{13} & d_{13} & d_{15} & d_{14} & d_{12} & d_1 & d_{11} \\
d_{14} & d_{14} & d_{13} & d_{12} & d_{15} & d_1 & d_{11} \\
d_{15} & d_{15} & d_{12} & d_1 & d_{11} & d_{13} & d_{14}
\end{array}
$$



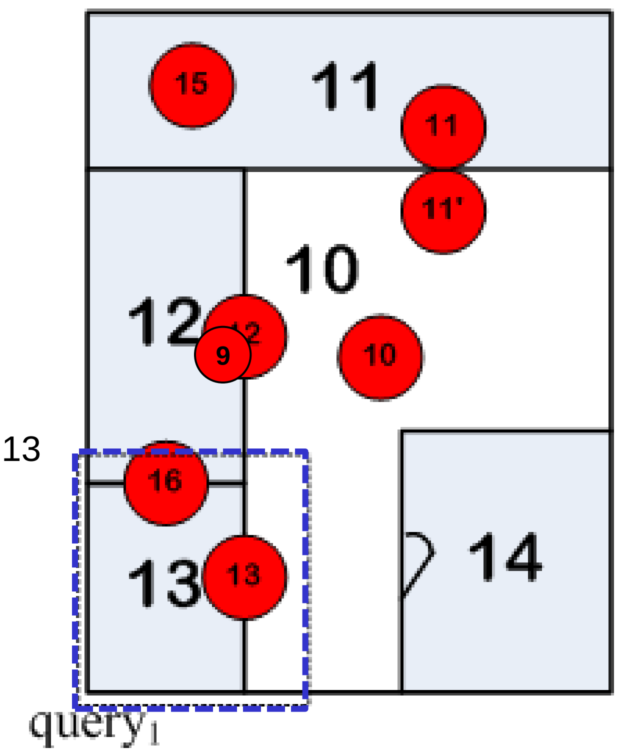
Continuous Range Monitoring Query (CRMQ)

- A CRMQ takes an indoor spatial range R as parameter, and monitors all the objects that are currently within R.
  - Symbolic representation of R: device/cell/room identifier
  - Geometrical representation of R: transform to symbolic identifiers
- Query-aware, incremental query processing approach
  - Identify the **critical devices**, from which new reading may potentially change the result of a given CRMQ.
  - Only *ENTER* and *LEAVE* readings from critical devices affect CRMQs



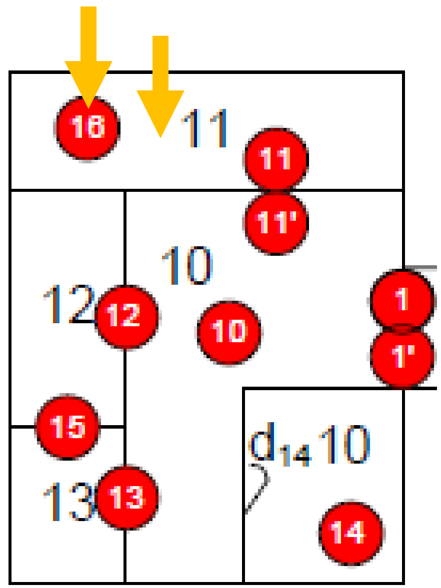
Query Result Accuracy

- RFID and like technology can only provide limited indoor positioning, and cannot report locations continuously.
- Certain Result
  - Those objects are definitely in the query range.
  - Active objects in device_{13}
  - Deterministic objects in cell_{13}
- Uncertain Result
  - Those objects may be in the query range.
  - Active objects in device_{16}
  - Nondeterministic objects that may be in cell_{13}
  - Both deterministic and nondeterministic objects in cell_{12} and cell_{10}
  - Use Uncertainty Region analysis to decide *how likely* (probability) an object is in a query range.



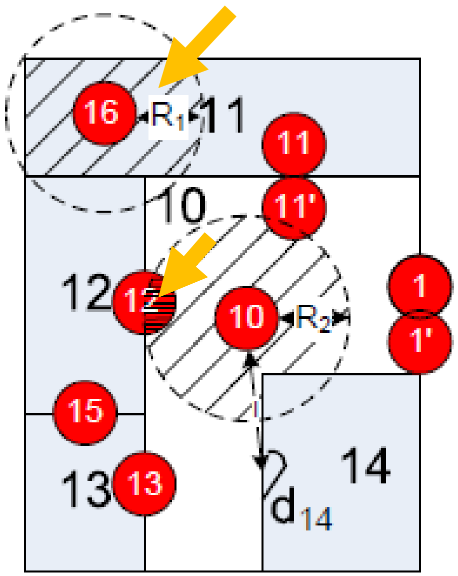
Uncertainty Regions for Indoor Moving Objects

- The **uncertainty region** of a moving object *o* at time *t*, denoted by $UR(o, t)$, is a region in which *o* must be at *t*.
- For an active object
  - $UR(o, t)$ is the detection range of the corresponding positioning device.
  - Suppose that object o_1 is seen by device dev_{16} at time t_{10}.
  - $UR(o_1, t_{10}) = Devices(dev_{16}).Range$
- For an inactive object
  - $UR(o, t)$ is the cell or cells that the object can belong to.
  - Suppose that object o_1 is seen LEAVING device dev_{16} at time t_{12}.
  - $UR(o_1, t_{13}) = C_{11}$



Refinement of Uncertainty Regions

- If we know an object's maximum speed V_{max}, we can refine its uncertainty region to a finer granularity.
- For a deterministic object
  - $UR(o, t)$ is the intersection of the object's cell and its maximum-speed constrained circle C_{MSC}
  - E.g., $UR(o_1, t_{13}) = C_{11} \cap C_{MSC}(\mathrm{R}_1)$, where $\mathrm{R}_1 = V_{max} \cdot (t_{13} - t_{12})$
- For a nondeterministic object
  - Do the intersection for every possible cell that may contains the object.
- An active object's UR may also be refined.
  - Suppose that object o left device dev_{10} at time t_{10} and then it is seen by device dev_{12}.
  - $UR(o, t_{now}) = Devices(dev_{12}).Range \cap C_{MSC}(\mathrm{R}_2)$, where $\mathrm{R}_2 = V_{max} \cdot (t_{now} - t_{10})$



Probabilistic Threshold *k*NN Query

- Given a set of indoor moving objects $O=\{o_1, o_2, \ldots, o_n\}$ and a threshold value T ($0 < T \leq 1$), a Probabilistic Threshold *k*NN Query (*PTkNN*) issued at time t with query location q returns a result set $R = \{A \mid A \subseteq O \wedge |A| = k \wedge prob(A) > T\}$, where $prob(A)$ is the probability that A contains the k nearest neighbors of the query location q at time t.

- Challenges
  - Given a large set O, the number of k-subsets (A in R) will increase exponentially.
  - For each k-subset A, computing $prob(A)$ can be expensive as it involves deciding URs and calculating probabilities.
  - Therefore, evaluating probabilities for all possible k-subsets is computationally prohibitive.

PTkNN Query Solution Overview

- Indoor Distance Based Pruning
  - Door-to-door distances are pre-computed from Doors Graph that is created based on the floor plan
  - Minimal Indoor Walking Distance (MIWD) is defined for any two positions in an indoor space
  - Combine URs MIWDs to prune unpromising objects
- Probability Threshold Based Pruning
  - Define relevant probability using the areas of URs

$$P_{o_i}(r) = \frac{Area(UR(o_i, t) \cap BR_q(r))}{Area(UR(o_i, t))}$$

  - Prune objects and *k*-subset by utilizing the probability threshold *T*
- Probability Evaluation

$$prob(A) = \sum_{o_z \in A} \int_0^{+\infty} p_{o_z}(r) \prod_{o_i \in A \setminus \{o_z\}} P_{o_i}(r) \prod_{o_j \in O' \setminus A} (1 - P_{o_j}(r))\, dr$$

  - Evaluate the continuous integral based probability in a more efficient discrete way.

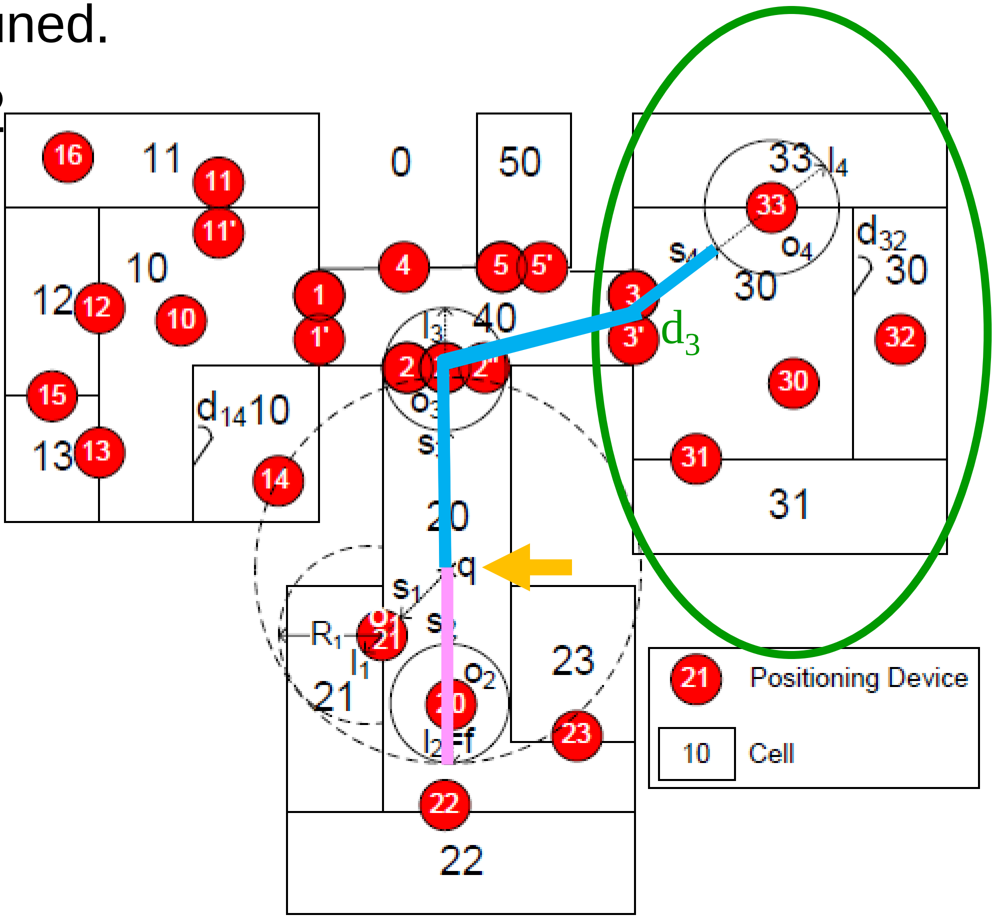
Indoor Distance Based Pruning

- The MIWD from query location q to object o_i's uncertainty region $UR(o_i, t)$
  - Lower bound: $s_i = \min_{p \in UR(o_i,t)} d_{MIW}(q,p)$
  - Upper bound: $l_i = \max_{p \in UR(o_i,t)} d_{MIW}(q,p)$
- k-bound f is the k'th minimal one of all objects' upper bounds (l_is).

- MIWD based pruning rule 1
  - If object o_i's $s_i \geq f$, o_i cannot be in any k-subset A of the result R because k objects are definitely colser to q than o_i.
- MIWD based pruning rule 2
  - Given a cell C, if $\min_{p \in C}\{d_{MIW}(q,p)\} \geq f$, all the objects in C can be safely pruned.

MIWD Based Pruning Examples

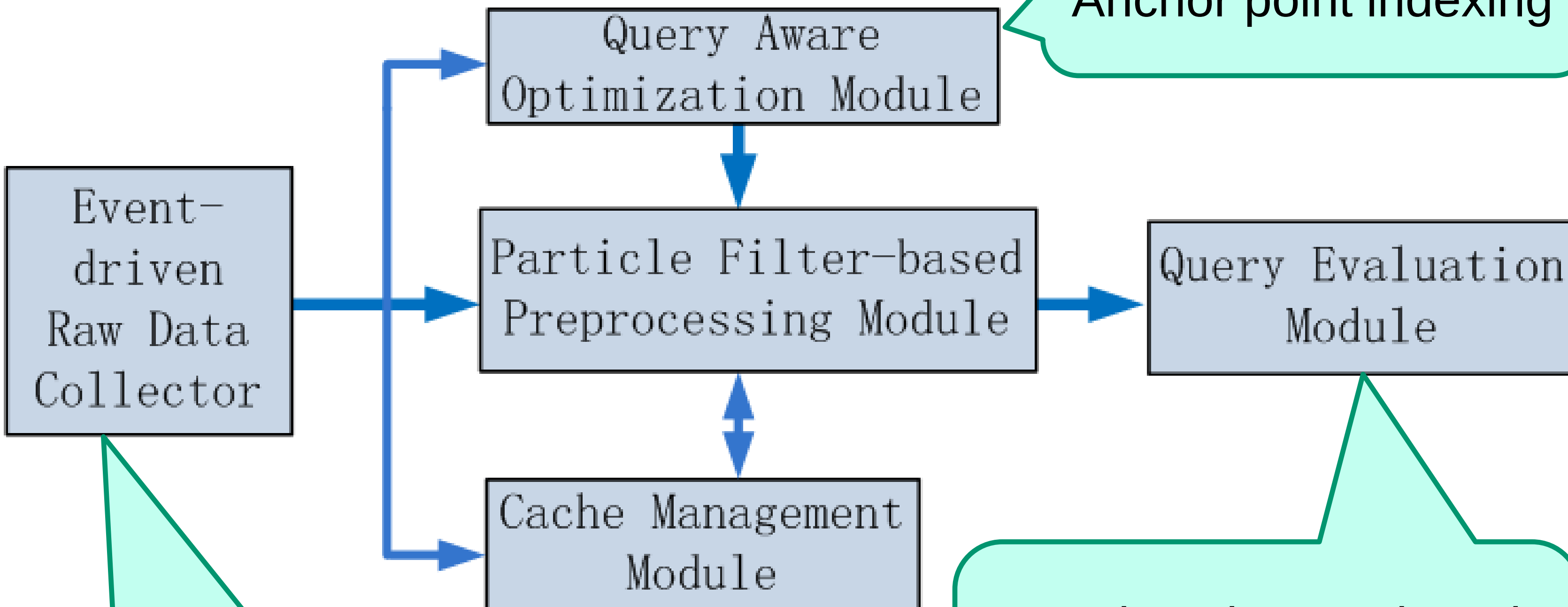
- $O=\{o_1, o_2, o_3, o_4\}$. Consider 2NN with query location q.
- MIWD based pruning rule 1
  - $l_1 < l_2 < l_3 < l_4$, so upper search bound $f = l_2$.
  - $s_4 > f$, so object o_4 can be pruned.
- MIWD based pruning rule 2
  - Cells 30, 31, and 33
  - $\min_{p\in C}\{d_{MIW}(q,p)\} = d_{MIW}(q,d_3) \geq f$, where C is one of these cells.
  - All objects in these cells can be pruned safely without computing their uncertainty regions.



Improving Query Accuracy with Particle Filter

Support snapshot range and kNN queries over online IMOs

Indoor walking graph
Anchor point indexing



Only stores readings from the most recent two detecting devices for each object.

Searches the graph and the index to determine the qualifying objects.

Probabilistic Threshold Indoor Spatio-Temporal Joins

- Probabilistic Threshold Indoor Spatio-Temporal Join (PTISSJ)
  - An Object Tracking Table *OTT*, a join predicate *P*, a time point *t*, and a threshold value $M \in (0, 1]$
  - *O* is the set of object identifiers
  - $\bowtie_{P,\, t,\, M}(OTT) = \{ (o_i, o_j) \mid o_i, o_j \in O \wedge o_i \neq o_j \wedge pr(P(o_i, o_j, t)) > M \}$

Join Probability

- Probabilistic Threshold *k* Indoor Spatio-Temporal Join (PTkISSJ)
  - An Object Tracking Table *OTT*, a join predicate *P*, a time interval $I = [t_m, t_n]$ $(m < n)$, an integer k $(0 < k < n\text{-}m)$, and a threshold value $M \in (0, 1]$
  - $$\bowtie_{P,\, I,\, k,\, M}(OTT) = \{ (o_i, o_j) \mid o_i, o_j \in O \wedge o_i \neq o_j \wedge \exists s \in m..n\text{-}k+1\ (\forall \delta \in 0..k\text{-}1 (pr(P(o_i, o_j, t_{s+\delta})) > M)) \}$$

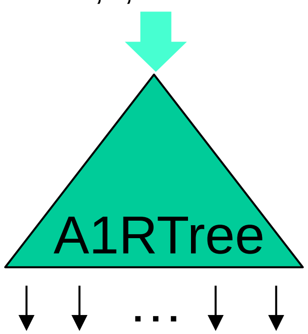
Join Processing

- Indexing Indoor tracking data
  - We create an augmented 1D R-tree (A1RTree) on the temporal attributes of object tracking table *OTT*.
  - Such that we can access relevant records (rd_{cov}, rd_{pre} and rd_{suc}) quickly for a given join time *t*.
- Object locations are basically bounded by device detection ranges or cells. It is beneficial to have the following mappings from a device or cell to X-region(s):
  - $CovD2X$: $D \rightarrow IR_X$ and $CovC2X$: $C \rightarrow IR_X$
    - Gets the X-region that fully covers the device/cell.
  - $IntD2X$: $D \rightarrow 2^{IR_X}$ and $IntC2X$: $C \rightarrow 2^{IR_X}$
    - Gets the set of X-regions that partially intersect the device/cell.
- A naïve join strategy
  - For each object pair, we get all relevant tracking records via the A1RTree, and evaluate the join probability.

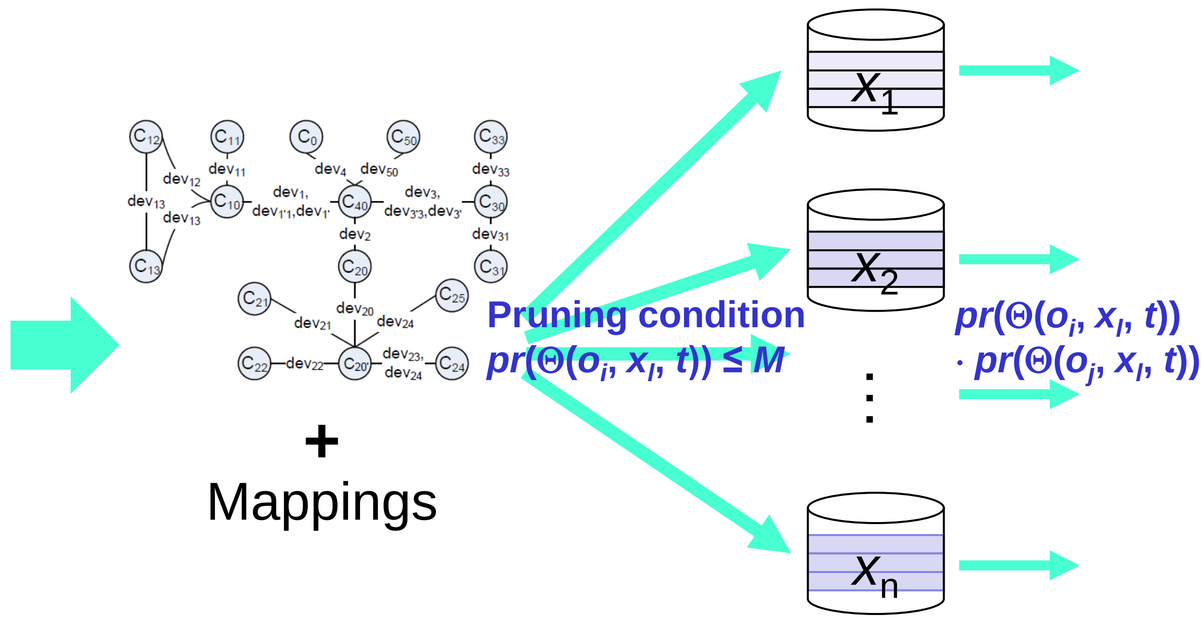
Two-Phase Hash-Based Join

- Motivation
  - In the naïve approach, it does not make sense to join two objects that are not in a same *X*-region.
- Join Processing

$\bowtie_{P,\, t,\, M}(OTT)$



ID	$objectID$	$deviceID$	t_s	t_e
rd_1	o_1	dev_4	t_1	t_2
rd_2	o_2	dev_4	t_1	t_2
rd_3	o_1	dev_2	t_5	t_6
rd_4	o_2	$dev_{1'}$	t_7	t_8
rd_5	o_1	dev_1	t_9	t_{10}
rd_6	o_1	dev_{12}	t_{15}	t_{16}
rd_7	o_2	dev_{13}	t_{20}	t_{21}
rd_8	o_1	dev_{13}	t_{21}	t_{22}
rd_9	o_2	dev_{13}	t_{29}	t_{30}
...	...	...	...	...

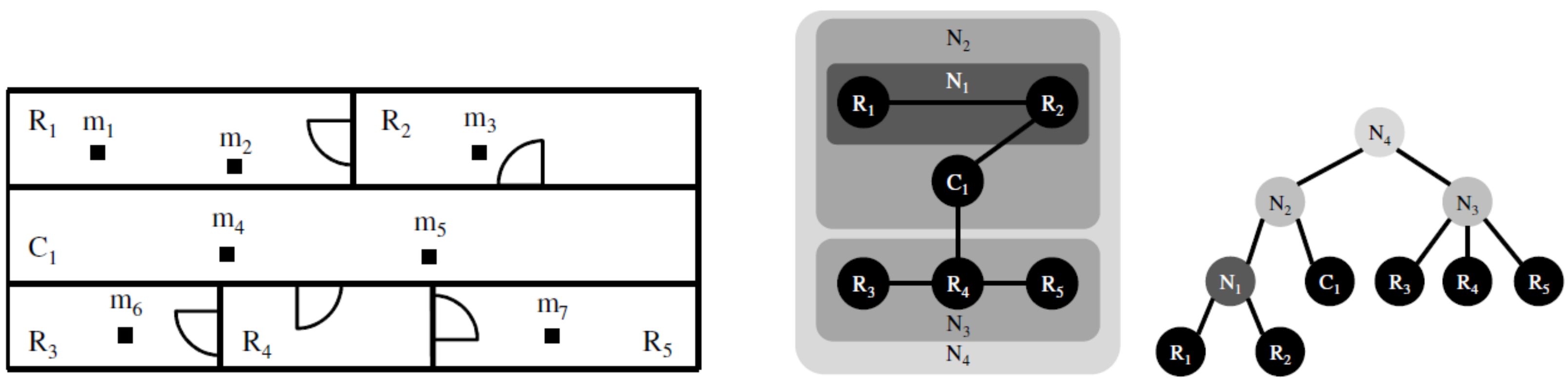


Outline

- Introduction, Motivation and Challenges
- Existing Research
  - Data Modeling for Indoor Space
  - Preprocessing Indoor Positioning Data
  - Indexing Indoor Space and Data
  - Querying Indoor Data
  - Other Topics
- Future Research Directions

Indoor Location Privacy

- Location k-anonymity in indoor spaces [24]
  - Using a hierarchical graph to organize a given indoor space.
    - A node corresponds to an indoor region, and an edge corresponds to the connectivity between two indoor regions
  - Indoor moving objects are managed on a region basis.
  - Bottom-up anonymizing indoor regions to achieve k-anonymity.
    - Start from the bottom region and goes up in the hierarchical graph until a region is found to have sufficient (>=k) objects.



(a) Floor plan and moving objects (b) Hierarchical graph (c) Hierarchy of nodes

Figures are from [24]

More Recent Works

- Joon-Seok Kim, Ki-Joune Li: Location K-anonymity in indoor spaces. GeoInformatica 20(3): 415-451 (2016)
  - An extension of [24]
- Andreas Konstantinidis, Georgios Chatzimilioudis, Demetrios Zeinalipour-Yazti, Paschalis Mpeis, Nikos Pelekis, Yannis Theodoridis: Privacy-Preserving Indoor Localization on Smartphones. IEEE Trans. Knowl. Data Eng. 27(11): 3042-3055 (2015)
  - Gives users location protection such that they are not tracked by localization services.
  - Exploits a k-Anonymity Bloom (kAB) filter and camouflaged localization requests.

Indoor Multimedia Data

- Geo-coding for indoor multimedia data [33]
  - Location information is explicitly or implicitly contained in multimedia data. Geo-coding for such data makes it easy to retrieve multimedia based on locations.
  - Requirement analysis for geo-coding of indoor multimedia
    - Indoor constraints, symbolic space, mobility, indoor positioning, etc.
  - Development of geo-coding scheme for indoor multimedia
    - Graph representation of indoor space
    - Stationary vs. mobile media
- Automatic geotagging and querying of indoor videos [26]
  - Wi-Fi fingerprinting indoor positioning at the room level
  - Smartphone based crowdsourcing to acquire locations for indoor spatial metadata

Analytics of Indoor Mobility Data

- Reasoning about RFID-tracked moving objects in symbolic indoor spaces [19]
  - A **model** for the indoor space and the RFID deployment
  - Techniques to **track** moving objects as symbolic routes
  - To **determine** the indoor locations of congestion
- Identifying typical movements among indoor objects [39]
  - Frequent indoor trajectory pattern mining
  - Candidate pattern generation, support computation
- Finding frequently visited indoor POIs from symbolic indoor tracking data [35]
  - Return the *k* POIs with the highest snapshot flows at time *t* or during interval $[t_s, t_e]$.
  - **Flow** is the probabilistic counting of objects whose uncertainty region overlaps an indoor POI's extent.

Outline

- Introduction, Motivation and Challenges
- Existing Research
- Future Research Directions
  - Keyword Search on Indoor Location Data
  - Integrating Indoor and Outdoor Space
  - Handling Uncertain Indoor Data
  - Indoor Trajectory Mining

Keyword search and beyond

- Indoor objects are associated with rich information, e.g.,
  - Textual information (e.g., nutritional information, price)
  - Social information (e.g., reviews, rating, recipe)
  - Multimedia (e.g., images, videos)
- Queries that search indoor space and exploit the associated information
- Existing outdoor techniques do not work
  - different indoor topology, distance measures, indexing etc.

Representative applications

- **Library:** Search a book by its title and navigate to it
- **Shopping:** Given a grocery list, find the ***optimal path*** to buy all items (e.g., minimize total price, or total walking distance etc.)
- **Shopping:** Find other people who will be interested in a “buy-one-get-one-free” deal (e.g., use shopping interests).
- **Airport:** Find nearest Emirates information centre

Integrating indoor and outdoor

- Many applications encompass both indoor and outdoor space (together called OI-space).
- Indexing and querying OI-space
- Trajectory mining in OI-space
- Representative applications
  - Navigate from multi-level car park to an office in a hospital
  - Find the nearest grocery shop from your hotel (considering multiple modes of transport, e.g., a combination of walk in OI-space and public transport)
  - Find the most popular/dense spots in a university campus

Handling uncertain indoor data

- Indoor locations/trajectories are uncertain
  - more serious than location errors in outdoor space
  - different sources of uncertainty (e.g, RFID, WiFi,. Bluetooth etc.)
- Indoor space may also be uncertain (e.g., unknown opening hours, door closing time, disability access etc.)
- Textual/social information associated with objects may also contain errors

- Model uncertainty from different types of positioning systems
- Queries to give probabilistic results

Indoor trajectory mining

- Indoor trajectory is similarly valuable as a user's clickstream
- Indoor trajectories are different from outdoor trajectories
  - Different topology
  - Different user behavior (e.g., walking speed, goal)
  - Different dimensionality and scale

Representative applications

- Flow analyses
  - How do people use the indoor space?
  - Waiting times in lines
    - At the ticket counter
    - At security
    - What can be done to improve the flow?
  - Travel times between zones
  - Heat map of the space
- Frequent visitor analysis
- Predict user's next location

Summary

- Existing research
  - Modelling of indoor space
  - Pre-processing of indoor positioning/tracking data
  - Indexing indoor data
  - Querying indoor data
  - Privacy, multimedia, etc.
- Future directions
  - Keyword search on indoor location data
  - Integrating indoor and outdoor space
  - Uncertainty in indoor data
  - Indoor trajectory mining
- Take-home message
  - A growing research field with high potential in practice
  - Open problems and direction for further research

ACM SIGSPATIAL Workshop: ISA 2016

- **What:** 8th International Workshop on Indoor Spatial Awareness (ISA) 2016
- **When:** Monday, October 31, 2016
- **Where:** San Francisco, USA
- **Submission deadline:** Early September 2016
- **Website:** coming soon (keep an eye)

References and Acknowledgments

- Reference list is given in the tutorial publication in the proceedings.
- Acknowledgements
  - Sponsors for our research
  - All our co-authors and collaborators
  - Ki-Joune Li for providing his slides for indoor data modeling
  - B. Fazzinga et al. [11, 12] for providing slides for their work

KLSG
Korean Land Spatialization Group

European Union
European Regional Development Fund
Investing in your future

The Danish National
Advanced Technology Foundation

daisy
Center for Data-intensive Systems

REGION NORDJYLLAND

Questions?

Thank you!

☺