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

Huan Li

Database Laboratory, Zhejiang University

lihuancs@zju.edu.cn

April 15, 2016

Overview

1. 1. Outlines
2. 2. Indoor Space Models & Applications
3. 3. Indoor Data Cleansing
4. 4. Indoor Movement Analysis
5. 5. Appendix

About This Work...

Efficient Distance-aware Query Evaluation on Indoor Moving Objects. [3]
X. Xie, H. Lu, and T. B. Pedersen.

- Published at *ICDE' 2013*.
- Study indoor distances and effective prunning bounds in relation to indoor moving objects.
- Design a composite index for indoor spaces and moving objects.
- Define and evaluate range queries as well as $k$nn queries on indoor moving objects.

Motivation

- In many indoor LBS scenarios, appropriate handling of indoor distances and relevant queries is of critical.
  - a cafe in a mall may send message to nearby shoppers to boost its business
  - in a large airport, it important to minitor individuals within a pre-defined range from a sensitive point
- Indoor spaces are characterized by many special entities and thus render distance calculation very complex.
- The limitations of indoor positioning technologies create inherent uncertainties in indoor moving objects data.

Notations

Notation	Meaning
$\mathbb{O}$	a set of uncertain objects
$\mathbb{I}$, $\mathbb{E}$	Indoor space, Euclidean space
$\|p, q\|_I$	Indoor distance between p and q
$\|p, q\|_E$	Euclidean distance between p and q
$\|p, q\|_K$	Skeleton distance between p and q
$a.l$ or $a.u$	lower or upper bound of the value a
$\uparrow A$	the link/pointer to the entity A
$[R_i^-, R_i^+]$	the range for R on dimension i
$len(R_i)$	$\|R_i^+ - R_i^-\|_E$
$D(p)$	doors of partition p
$P(d)$	partitions connected to door d
$P(q)$	the partition containing point q
$P(O)$	partitions overlapping with object O
$\|O\|$	the number of instances belonging to object O
$a \overset{*d}{\rightsquigarrow} b$	a path from a to b with d as the last door
$a \xrightarrow{*} b$	the shortest path from a to b
$\odot(c, r)$	a circle centered at c with radius r

Preliminaries: Indoor Space and Indoor Distance

Doors Graph has been proposed to represent the connectivity of indoor partitions as well as door-to-door distances. [1]

Preliminaries: Indoor Space and Indoor Distance

Doors Graph has been proposed to represent the connectivity of indoor partitions as well as door-to-door distances. [1]

Given two indoor positions p an q, we use $q \overset{\delta}{\rightsquigarrow} p$ to denote a path from q to p where δ is the sequence of doors on the path.

Preliminaries: Indoor Space and Indoor Distance

Doors Graph has been proposed to represent the connectivity of indoor partitions as well as door-to-door distances. [1]

Given two indoor positions p an q, we use $q \overset{\delta}{\rightsquigarrow} p$ to denote a path from q to p where δ is the sequence of doors on the path.

The length of the shortest path as *indoor distance* from q to p, and denote it formally as $|q,p|_I = min_{\delta}(|q \overset{\delta}{\rightsquigarrow} p|)$, also $q \overset{\delta}{\rightarrow} p$.

Preliminaries: Indoor Space and Indoor Distance

Doors Graph has been proposed to represent the connectivity of indoor partitions as well as door-to-door distances. [1]

Given two indoor positions p an q, we use $q \overset{\delta}{\rightsquigarrow} p$ to denote a path from q to p where δ is the sequence of doors on the path.

The length of the shortest path as *indoor distance* from q to p, and denote it formally as $|q, p|_I = min_\delta(|q \overset{\delta}{\rightsquigarrow} p|)$, also $q \xrightarrow{\delta} p$.

indoor distance consists of *door-door distance* and *intra-partition object-door distance*:

Preliminaries: Indoor Space and Indoor Distance

Doors Graph has been proposed to represent the connectivity of indoor partitions as well as door-to-door distances. [1]

Given two indoor positions p an q, we use $q \overset{\delta}{\rightsquigarrow} p$ to denote a path from q to p where δ is the sequence of doors on the path.

The length of the shortest path as *indoor distance* from q to p, and denote it formally as $|q, p|_I = min_\delta(|q \overset{\delta}{\rightsquigarrow} p|)$, also $q \overset{\delta}{\rightarrow} p$.

indoor distance consists of *door-door distance* and *intra-partition object-door distance*:

$$min_{d_q \in D(q), d_p \in D(p)}(|q, d_q|_E + |d_q, d_p|_I + |d_p, p|_E) \qquad (1)$$

Indoor Moving Objects

- Existing proposals [4, 1] model a moving object by an *uncertainty region*, where the exact location is considered as a random variable inside.
- The possibility of its appearance can be collected by object's velocities [1], parameters of positioning device [4], or analysis of historical records (represented by *pdf*).
- The *pdf* can be either a close form equation [5, 6] or a set of instance representation [7], as it is general for arbitrary distribution.
- Thus, an indoor moving object O is represented by a set (s_i, p_i), where s_i is an instance and p_i is its *existential probability*, satisfying $\sum_{s_i \in O} p_i = 1$.

Expected Indoor Distance

Definition (Expected Indoor Distance for Uncertain Object)

Given a fixed point $q \in \mathbb{I}$ and an uncertain object O, the indoor distance from q to O is

$$|q, O|_I = E_{s_i \in O}(|q, s_i|_I) = \sum_{s_i \in O} |q, s_i|_I \cdot p_i \tag{2}$$

an object O's uncertainty region may overlap with multiple partitions. Accordingly, all the instances in O are divided into subsets, i.e., $O = \cup_{1 \leq j \leq m} S[j] (1 \leq m \leq |O|)$ where each $S[j]$ corresponds to a different partition, it is called O's *uncertainty subregion*.

Case of Indoor Distance $|q, O|_I$ (I)

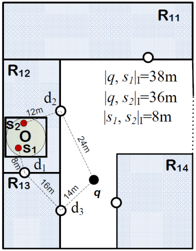
Single-Partition Single-Path Distance O's uncertainty region falls into one single partition P. For an arbitrary $s_i \in O$, the shortest path $q \overset{*d}{\rightarrow} s_i$ shares the path enters P to reach s_i.

$$|q, O|_I = |q, d|_I + \sum_{s_i \in O} |d, s_i|_E \cdot p_i \tag{3}$$

Case of Indoor Distance $|q, O|_I$ (II)

Single-Partition Multi-Path Distance O's uncertainty region still falls into one single partition P. However, for different instances s_i and s_j, the shortest path $q \xrightarrow{*} s_i$ and $q \xrightarrow{*} s_j$ do not share the same door sequence.

$$|q, O|_I = \sum_{s_i \in O} |q, s_i|_I \cdot p_i \tag{4}$$



Example

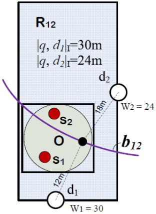
O has two instance s_1 and s_2, the shortest path from q to them are: $q \overset{d_3,d_1}{\rightsquigarrow} s_1$ and $q \overset{d_2}{\rightsquigarrow} s_2$.

Case of Indoor Distance $|q, O|_I$ (II)

The *solution space* of the single-partition multi-path distance is the **Additive Weighted Voronoi Diagram**.

Suppose partition P has doors $\{d_1, ..., d_m\}$, for each door d_i, a weight $w_i = |q, d_i|_I$ is assigned. Use *weighted bisectors* to represent the *Additive Weighted Voronoi Diagram*. Given two doors d_i and d_j, whose weights are w_i and w_j, respectively, the *weighted bisector* b_{ij} is a curve:

$$b_{ij} = \{p : |p, d_i|_E + w_i = |p, d_j|_E + w_j\} \qquad (5)$$



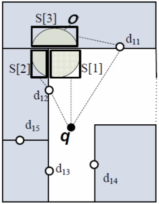
Shape of b_{ij}	Condition
straight line	$w_i = w_j$
hyperbola	$w_i \neq w_j$ *and* $w_i < \|d_j, P\|_{maxE}$ *and* $w_j < \|d_i, P\|_{maxE}$
null	$w_i > \|d_j, P\|_{maxE}$ *or* $w_j < \|d_i, P\|_{maxE}$

Case of Indoor Distance $|q, O|_I$ (III)

Multi-Partition Multi-Path Distance O's uncertainty region overlaps with more than one partition, and thus $O = \cup_{1 \leq j \leq m} S[j] (1 \leq m \leq |O|)$.

$$|q, O|_I = \sum_{1 \leq j \leq m} (|q, S[j]|_I \cdot \sum_{s_i \in S[j]} p_i) \quad (6)$$

$|q, S[j]|_I$ is calculated according to case I or case II, by substituting $S[j]$ for O.



Example

O has three uncertainty subregions S_1, S_2 and S_3. Accordingly, $|q, O|_I = E(\sum_{1 \leq j \leq 3}(|q, S[j]_I|))$.

Bounds for Indoor Distances

Euclidean Lower Bounds

Lemma (Euclidean Lower Bounds)

For point q and object O in an indoor space, the (virtual) Euclidean distance between them is the lower bound of their indoor space. Therefore, it has $|q, O|_{minE} \leq |q, O|_I$, where $|q, O|_{minE} = \min_{s_i \in O} |q, s_i|_E$.

it is impossible to derive the indoor upper bounds by using Euclidean distances only.

Bounds for Indoor Distances I

Indoor Toplogical ULBounds

Lemma (Toplogical Lower Bounds)

Let $t_{min}(S[i])$ *be:*

$$\min_{d_q \in D(P(q)), d_s \in D(P(S[i]))} |q, d_q|_E + |d_q \xrightarrow{*} d_s| + |d_s, S[i]|_{minE}$$

. *Then,* $|q, O|_I \geq min\{t_{min}(S[i])\}$.

Lemma (Toplogical Upper Bounds)

Let $t_{max}(S[i])$ *be:*

$$\min_{d_q \in D(P(q)), d_s \in D(P(S[i]))} |q, d_q|_E + |d_q \xrightarrow{*} d_s| + |d_s, S[i]|_{maxE}$$

. *Then,* $|q, O|_I \leq max\{t_{max}(S[i])\}$.

Bounds for Indoor Distances II

a looser topological upper bound is more economic to be derived, it also requires knowing some paths connecting point q and subregion $S[i]$:

Lemma (Toplogical Looser Upper Bounds, TLU)

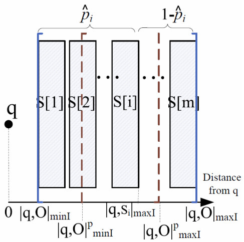
Let $t_{max}(S[i])$ *be:*

$$\min_{d_q \in D(P(q)), d_s \in D(P(S[i]))} |q, d_q|_E + |d_q \overset{*}{\rightsquigarrow} d_s| + |d_s, S[i]|_{maxE}$$

. *Then,* $|q, O|_I \leq max\{t_{max}(S[i])\}$.

Bounds for Indoor Distances

Indoor Probabilistic ULBounds



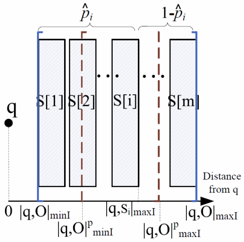
Lemma (Markov Lower Bounds)

Suppose object O overlaps with m partitions ($O = \cup_{i=1}^{m} S[i]$), and $S[i]$s are sorted according to the minimum distance to a given point q. Use $\widehat{p_i}$ to denote $\sum_{j=1}^{i} p_i$. As $S[i]$ and $S[j]$ do not overlap, using $Markov\ Inequality$, we have:

$$E(|q,O|_I) \geq |q,S[i]|_{maxI} \cdot (1 - \widehat{p_i})$$

Bounds for Indoor Distances

Indoor Probabilistic ULBounds



Lemma (Probabilistic ULBounds)

$$|q, S[i]|_{maxI} \cdot (1 - \widehat{p_i}) + |q, O|_{minI} \cdot \widehat{p_i}$$
$$\leq E(|q, O|_I) \leq$$
$$|q, O|_{maxI} \cdot (1 - \widehat{p_i}) + |q, S[i]|_{maxI} \cdot \widehat{p_i}$$

Proof: $E(|q, O|_I) = E(|q, \cup_{j \leq i} S[j]|_I) \cdot \widehat{p_i} + E(|q, \cup_{k > i} S[k]|_I) \cdot (1 - \widehat{p_i})$. *Since* $|q, S[i]|_{maxI} \geq E(|q, \cup_{j \leq i} S[j]|_I) \geq |q, O|_{minI}$, *and* $|q, O|_{maxI} \geq E(|q, \cup_{k > i} S[k]|_I) \geq |q, O|_{minI}$, *by substitution, the lemma is proved.*

Summary

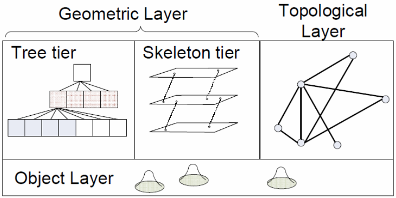
use *topological ULBounds* for the case that an object overlaps with a single partition;

use *probabilistic ULBounds* for the case that an object overlaps with multiple partitions.

Indoor Distance	Bounds
Single-partition single-path distance Single-partition multi-path distance	Indoor Topological Upper/ Lower Bounds (Equation 7)
Multi-partition path distance	Indoor Probabilistic Upper/ Lower Bounds (Equation 8)

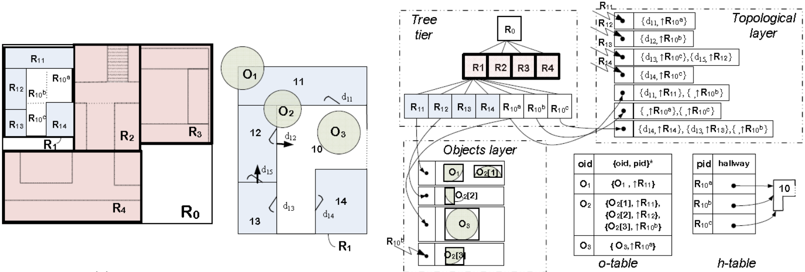
with the Upper and Lower Bounds, as well as the approximate indoor distance, one can avoid computing shortest paths for all existential instances of an uncertain objects.

Composite Index for Indoor Space



- **geometric layer** consists of a tree structure that adapts the R*-tree to index all partitions, as well as a skeleton tier that maintains a small number of distances between staircases.
- **topological layer** maintains the connectivity information between indoor partitions.
- **object layer** stores all indoor moving objects and is associated with the tree through partitions at its leaf level.

Composite Index: Overview



(a) (b) (c)

Composite Index: Tree Tier

- instead of 3D $MinimumBoundingRectangle$, when creating the tree, set the vertical length for one partition to 1 centimeter. Two advantage: 1) reduce the distance calculation workload; 2) makes the distance reflected in the tree more accurate without the disturbance from the vertical dimension.
- the imbalanced partition are decomposed to small but regular region, each is called an *index unit*.
- A hash table is used to map such an index unit to its original indoor partition.

Algorithm 3 Decompose

function DECOMPOSE(Region r, a set of turnning points P, threshold T_{shape})
  if r is concave **then**
    let $R(r)$ be the **MBR** of r;
    select a *turning point* $t \in P$ on r's boundary, such that t is closer to the middle of r;
    draw a splitting line perpendicular to the longer dimension d to divide r into two or more regions: $\{r_i\}$;
    for each r_i in $\{r_i\}$ **do**
      `Decompose(`r_i, $P - \{t\}$, T_{shape}`);`
  else
    if $\frac{len(R(r)_1)}{len(R(r)_2)} > T_{shape}$ or $\frac{len(R(r)_1)}{len(R(r)_2)} < T_{shape}$ **then**
      find the middle point m on r's longer dimension d;
      draw a splitting line perpendicular to d to divide r into two regions: r_1 and r_2;
      `Decompose(`r_1, P, T_{shape}`);`
      `Decompose(`r_2, P, T_{shape}`);`

Composite Index: Object Tier

A hash table $o - table$

$$o - table : \{O\} \rightarrow 2^{\{index\ unit\}}$$

$o - table$ maps an object to all the index units it overlaps, and it is tightly tie up with the tree tier.

When an object update occurs, $o - table$ needs to be updated accordingly.

Composite Index: Topological Tier

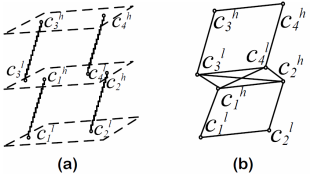
This layer maintains the connectivity between partitions. Each leaf node stores a (sub)partition.

For accessibility, the doors belonging to the partitions are also stored, as well as the the links to accessible partitions through each door.

Composite Index: Skeleton Tier

Skeleton Tier is a graph, each staircase entrance is captured as a graph node, and an edge connects two nodes if their entrances are on the same floor or their entrances belong to the same staircase.

The weight of an edge is the indoor distance between the two staircase entrances.



Definition (staircase distance matrix M_{s2s})

- $M_{s2s}[s_i, s_i] = 0$;
- $M_{s2s}[s_i, s_j] = |s_i, s_j|_E$ if s_i and s_j are on the same floor;
- if s_i and s_j are of a same staircase, $M_{s2s}[s_i, s_j]$ is the shortest distance from s_i to s_j within that staircase;
- $M_{s2s}[s_i, s_j]$ is calculated as the shortest path distance from s_i to s_j in the skeleton layer for other cases.

Skeleton Distance

Let q be a fixed indoor point, $q.f$ the floor of q, and $S(q.f)$ all the staircases on floor $q.f$.

Definition (Skeleton Distance)

Given two points p and q, their skeleton distance $|q,p|_K = |q,p|_E$ if they are on the same floor; otherwise,
$|q,p|_K = \min_{s_q \in S(q.f), s_p \in S(p.f)}(|q,s_q|_E + M_{s2s}[s_q,s_p] + |s_p,p|_E)$.

Define the skeleton distance as the alternative *Geometric Distance*.

Indoor Distance Bounds in the Geometric Layer

Lemma (Geometric Lower Bound Property)

Given two points p and q, their skeleton distance lower bounds their indoor distance, i.e., $|q,p|_K \leq |q,p|_I$.

Proof: *If q and p are on the same floor, $|q,p|_K = |q,p|_E \leq |q,p|_I$. Otherwise, suppose $s_q^* \in S(q.f)$ and $s_p^* \in S(p.f)$ are on the shortest path from q to p, denoted by $q \overset{*s_q^* * s_p^*}{\rightarrow} p$. Since* $|q,p|_K = \min_{s_q \in S(q.f), s_p \in S(p.f)}(|q,s_q|_E + M_{s2s}[s_q,s_p] + |s_p,p|_E) \leq |q,s_q^*|_E + M_{s2s}[s_q^*,s_p^*] + |s_p^*,p|_E = |q,p|_I$, *the lemma is proved.*

Indoor Distance Bounds in the Geometric Layer

Consider an entity e that is either an object or an indR-tree node. If e spans multiple floors, we use interval $[e.lf, e.uf]$ to represent all those floors. Note those floors must be consecutive. We define the minimum skeleton distance $|q, e|_{minK}$:

$$|q, e|_{minK} = \begin{cases} |q, e|_{minE}, \ if \ q.f \in [e.lf, e.uf]; \\ min \\ \{ \min\limits_{s_q \in S(q.f), s_e \in S(e.lf)} (|q, s_q|_E + M_{s2s}[s_q, s_e] + |s_e, e|_{minE}), \\ \min\limits_{s_q \in S(q.f), s_e \in S(e.uf)} (|q, s_q|_E + M_{s2s}[s_q, s_e] + |s_e, e|_{minE})\}, \\ otherwise. \end{cases}$$

With $|q, e|_{minK}$, one can constrain the search via the indR-tree to a much smaller range compared to if use the Euclidean distance bounds.

Dynamic Operations on the Topological Layer

Insertion. When the topological change leads to a new indoor partition P, P(or its sub-partitions due to decomposition) is inserted into the indR-tree, its leaf node is connected to the adjacent partitions, and the $h - table$ is updated if a decomposition is invovled.

Deletion. From the indR-tree to remove a partition P to be deleted, the links involving P are removed from the adjacent partitions, and P's entry in the $h - table$ is deleted if P is a decomposed sub-partition.

Dynamic Operations on the Object Layer

Insertion. To insert an object O, search the indR-tree to find the leaf nodes $\{P_i\}$ that overlap with O's uncertainty region. Also insert a new entry to $o - table$.

Deletion. To delete an object O, use the $o - table$ to find the indR-tree leaf nodes $\{P_i\}$ that overlap with O's uncertainty region. For each P_i, O is removed from its associated bucket. Also the entry for O is deleted from the $o - table$.

Query Semantics

Definition (Indoor Range Query, iRQ)

Given a query point $q \in \mathbb{I}$ and a distance value r, the iRQ returns objects whose indoor distances are smaller than r. Formally, $iRQ_{q,r}(\mathbb{O}) = \{O||q,O|_I \leq r, O \in \mathbb{O}\}$.

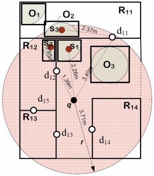
Definition (Indoor k Nearest Neighbor Query, ikNNQ)

Given a query point $q \in \mathbb{I}$ and a parameter k, the $ikNNQ$ returns k objects whose indoor distances to q are the smallest among all objects. Formally, $ikNN_{q,k}(\mathbb{O}) = \{O|O \in \mathbb{O}\}$, where $|ikNN_{q,k}(\mathbb{O})| = k, \forall O_i \in ikNN_{q,k}(\mathbb{O}), \forall O_j \in \mathbb{O} \setminus ikNN_{q,k}(\mathbb{O}), |q,O_i|_I \leq |q,O_j|_I$.

Efficient Query Evaluation

1. **Filtering Phase** locates the source partition that contains the query point and retrieves condidate partitions as well as candidate objects.
2. **Subgraph Phase** constructs a subgraph based on candidate partitions, and uses the doors of the source partition as sources to compute the shortest indoor paths that are to be used in the subsequent two phases.
3. **Prunning Phase**, upper/lower distance bounds for objects are calculated to further reduce the number of candidate objects.
4. **Refinement Phase**, the indoor distances for the remaining objects are computed and the qualifying objects are returned as the query results.

Indoor Range Query



Example

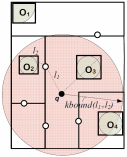
The circle $\odot(q, r)$ is the query region represented in the Euclidean space. Object O_1 is pruned away in filtering phase, since $|q, O_1|_{minE} > r$. After deriving the upper/lower bounds for the remaining objects in the pruning phase, O_3 is qualified. For the undetermined object O_2, the exact indoor distance is calculated and compared to r.

Algorithm 1 iRQ

1: **function** IRQ(query point q, distance r, indoor index $\mathcal{T}$)
2: result set R; candidate object set C;
3: $(R^o, R^p) \leftarrow RangeSearch(q, r, \mathcal{T})$; // **Phase1 : filtering**
4: Dijkstra(R^p); // **Phase 2: subgraph**
5: **for** each object O in R^o **do** // **Phase 3: pruning**
6: $[O.l, O.u] \leftarrow [|q, O|_{minI}, |q, O|_{maxI}]$; // (Table III)
7: **for** each $O \in R^o$ **do**
8: **if** $O.u \leq r$ **then** $R = R \cup \{O\}$
9: **else**
10: **if** $O.l \leq r$ **then** $C = C \cup \{O\}$
11: **for** each $O \in C$ **do** // **Phase 4: refinement**
12: Calculate $|q, O|_I$;
13: **if** $|q, O|_I \leq r$ **then** $R = R \cup \{O\}$;
14: **return** R.

- in the filtering, iRQ calls $RangeSearch$ to search the geometric layer.
- lines 5–10: iRQ makes use of the topological upper/lower bounds to approximate indoor distances and compare them to r.
- lines 11–13: the exact indoor distances are only computed for those objects whose bounds cover r.

Indoor k Nearest Neighbor Query



Example

$kSeedsSelection$ finds O_2 and O_3 as seeds. Because O_2's topological looser upper bound is longer, it is chosen as the kbound. Through the range search, O_1 is excluded since $|q, O_1|_K > kbound$.

Algorithm 2 ikNNQ

1: **function** IKNNQ(query point q, k, indoor index $\mathcal{T}$)
2: result set R; candidate object set C;
3: $(R_1^o, R_1^p) \leftarrow kSeedsSelection(q, k)$; // **Phase 1: filtering**
4: kbound $\leftarrow max_{O \in R_1^o}\{|q, O|_I.TLU\}$; // (Lemma 3)
5: $(R_2^o, R_2^p) \leftarrow RangeSearch(q, kbound, \mathcal{T})$;
6: Dijkstra(R_2^p); // **Phase 2: subgraph**
7: **for** each object O in R_2^o **do** // **Phase 3: pruning**
8: $[O.l, O.u] \leftarrow [|q, O|_{minI}, |q, O|_{maxI}]$; // (Table III)
9: Find object O_k which has the k-th shortest $O.u$; set $C = \emptyset$;
10: **for** each $O \in R_2^o$ **do**
11: **if** $O.u < O_k.l$ **then** $R = R \cup \{O\}$
12: **else**
13: **if** $O.l \leq O_k.u$ **then** $C = C \cup \{O\}$
14: **for** each $O \in C$ **do** // **Phase 4: refinement**
15: Calculate $|q, O|_I$;
16: Sort objects in C by $|q, O|_I$ in ascending order and add top $k - |R|$ objects to R;
17: **return** R.

- in the filtering, ikNNQ calls $kSeedsSelection$ to return an object R_1^o and a partition set R_1^p.
- R_1^o contains k objects taht are in query point q's partition or in the closet adjacent partitions. R_1^p is the set of all those involved partition.
- ikNNQ derives $Topological\ Looser\ Upper\ Bounds$ for the k objects and choose the longest one as $kounds = \max_{seed_i \in R_1^o}\{|q, seed_i|_I.TLU\}$.
- Line 4: a range search $\bigodot(q, kound)$ is done on the tree tier.

Research Directions

- it is of interest to study other query types using the distance bounds and the composite index proposed in this paper.
- it is useful to estimate the selectivity for indoor distance aware queries and make use of it in further optimizing queries over uncetain object.
- it is of benificial to reuse computational efforts on indoor distances when multiple, related queries are issued within a short period of time.

References I

[1] B. Yang, H. Lu, and C. S. Jensen.
Probabilistic threshold k nearest neighbor queries over moving objects in symbolic indoor space.
In *EDBT*, pp. 335–346, 2010.

[2] H. Lu, X. Cao, and C. S. Jensen.
A foundation for efficient indoor distance-aware query processing.
In *ICDE*, pp. 438–449, 2012.

[3] X. Xie, H. Lu, and T. B. Pedersen.
Efficient distance-aware query evaluation on indoor moving objects.
In *ICDE*, pp. 434–445, 2013.

[4] D. Pfoser and C. S. Jensen.
Capturing the uncertainty of moving-object representations
In *Advances in Spatial Databases*, pp. 111–131, 1999.

References II

[5] R. Cheng, D. V. Kalashnikov and S.Prabhakar.
Evaluating probabilistic queries over imprecise data.
In *SIGMOD*, pp. 551–562, 2003.

[6] R. Cheng, D. V. Kalashnikov and S.Prabhakar.
Querying imprecise data in moving object environments.
In *TKDE*, pp. 1112–1127, 2004.

[7] H.-P. Kriegel,P. Kunath and M. Renz.
Probabilistic nearest-neighbor query on uncertain objects.
In *DSAFAA*, pp. 337–348, 2007.

The End. Thanks :)