Indoor Spatial Queries: Modeling, Indexing, and Processing

Tiantian Liu, Huan Li, Hua Lu, Muhammad Aamir Cheema, Lidan Shou

An EDBT 2021 Work By
Aalborg University
Roskilde University
Monash University
Zhejiang University

Introduction

- Indoor location-based services (LBS) are becoming increasingly popular.

- To facilitate query processing for indoor LBS, space models, indexes and algorithms have been proposed.

- An experimental study on all these proposals is still missing.

- We conduct a comprehensive experimental study in this work.

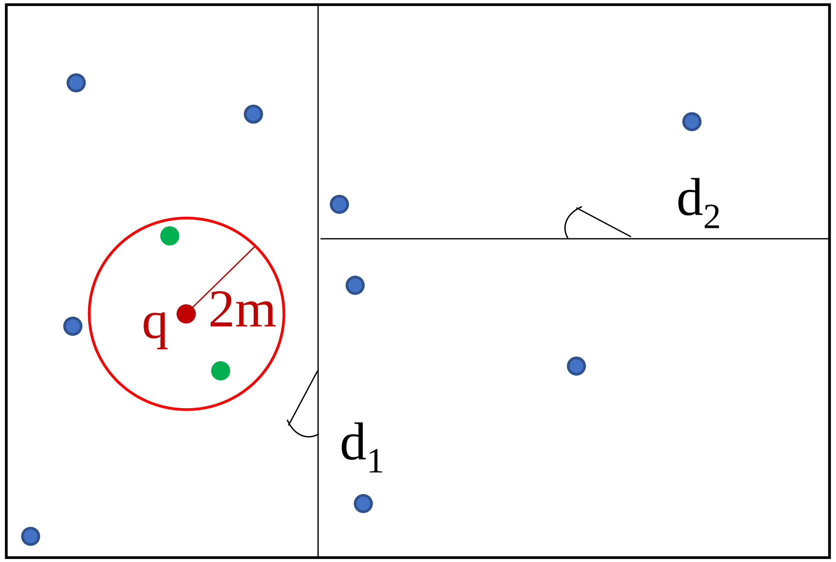
Contents

- Indoor Spatial Queries
- Model and Indexes
- Query Processing
- Benchmark
- Results Summary
- Future Work

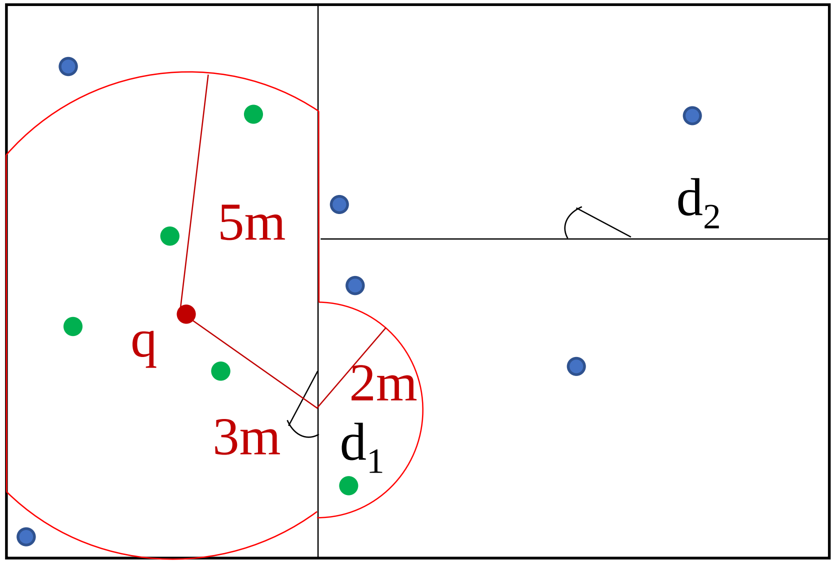
Contents

Indoor Spatial Queries

- Range Query (RQ)



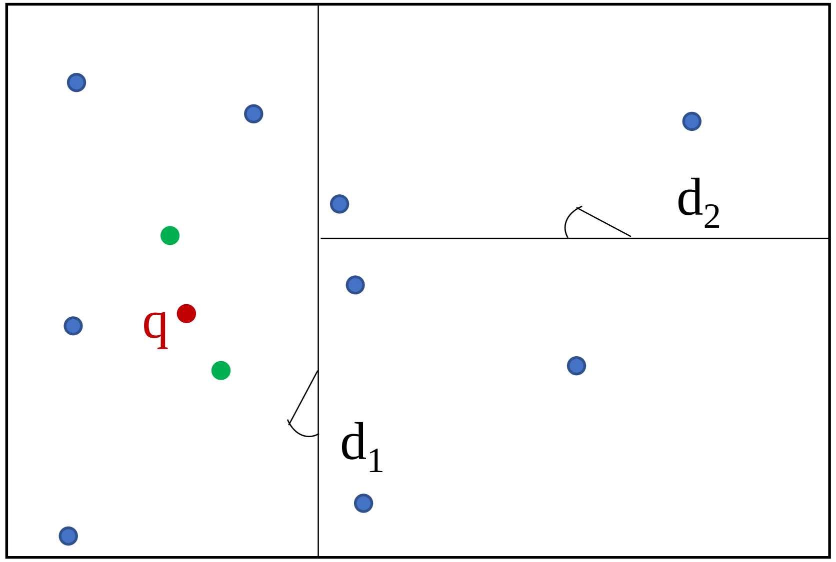
RQ (q, 2m)



RQ (q, 5m)

Indoor Spatial Queries

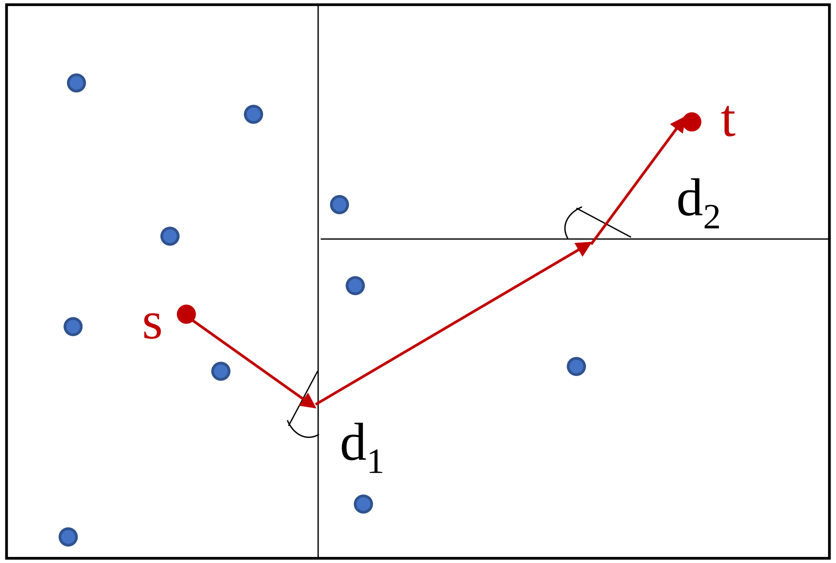
- k Nearest Neighbor Query (kNNQ)



kNNQ (q, 2)

Indoor Spatial Queries

- Shortest Path Query (SPQ)/Shortest Distance Query (SDQ)



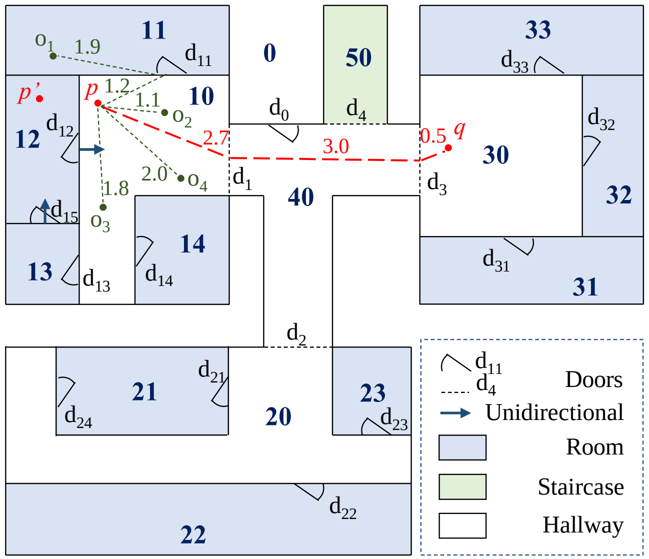
SPQ (s, t)

Contents

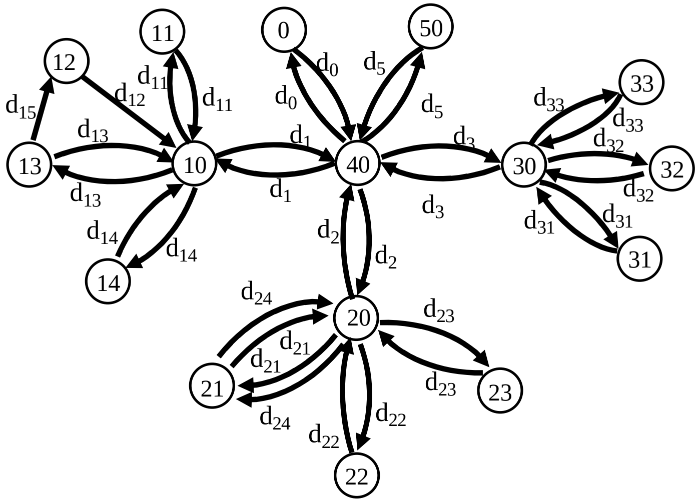
- Indoor Spatial Queries
- Model and Indexes
  - Indoor Distance-Aware Model (IDModel)
  - Indoor Distance-Aware Index (IDIndex)
  - Composite Indoor Index (CIndex)
  - Indoor Partitioning Tree (IP-Tree)
  - Vivid IP-Tree (VIP-Tree)
- Query Processing
- Benchmark
- Results Summary
- Future Work

Model and Indexes

- Indoor Distance-Aware Model (IDModel) *



An example of the floorplan



Door-to-partition distance map

Key	Value
(d_0, v_{40})	*3.5m*
(d_0, v_{30})	*6.1m*
…	

Door-to-door distance map

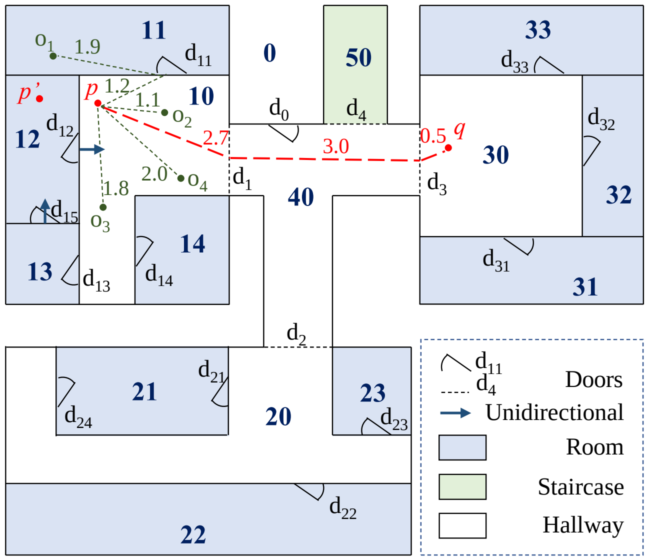
Key	Value
(v_{40}, d_0, d_1)	*1m*
(v_{40}, d_0, d_5)	*1.2m*
…	

An example of IDModel

* *Hua Lu, Xin Cao, and Christian S Jensen. 2012. A foundation for efficient indoor distance-aware query processing. In ICDE. 438–449.*

Model and Indexes

- Indoor Distance-Aware Index (IDIndex)*



An example of the floorplan

$$
\begin{array}{c}
\begin{array}{cccccc} d_1 & d_{11} & d_{12} & d_{13} & d_{14} & d_{15} \end{array} \\
\begin{array}{c} d_1 \\ d_{11} \\ d_{12} \\ d_{13} \\ d_{14} \\ d_{15} \end{array}
\begin{pmatrix}
0 & 1.7 & 2.7 & 3.6 & 2.8 & 4.6 \\
1.7 & 0 & 1.9 & 3.6 & 2.8 & 4.6 \\
2.7 & 1.9 & 0 & 2.6 & 1.8 & 1.6 \\
3.2 & 3.4 & 2 & 0 & 2 & 1 \\
2.8 & 2.8 & 1.8 & 1 & 0 & 2 \\
4.3 & 3.5 & 1.6 & 1 & 2 & 0
\end{pmatrix}
\end{array}
$$

(a) Distance Matrix M_{d2d}

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

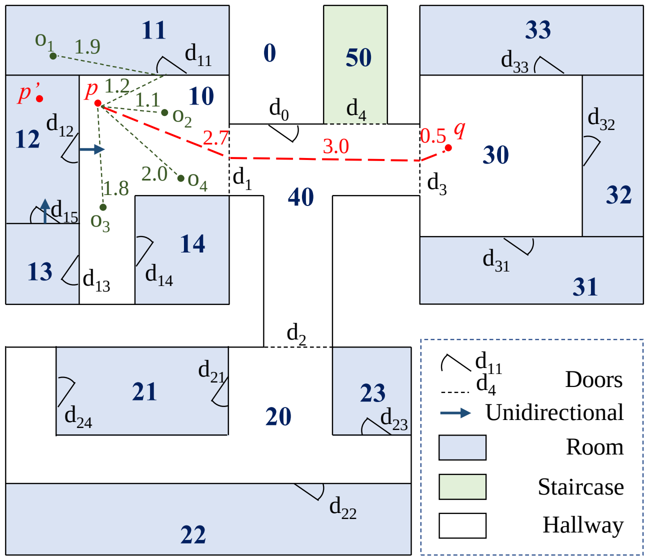
(b) Distance Index Matrix M_{idx}

An example of IDIndex

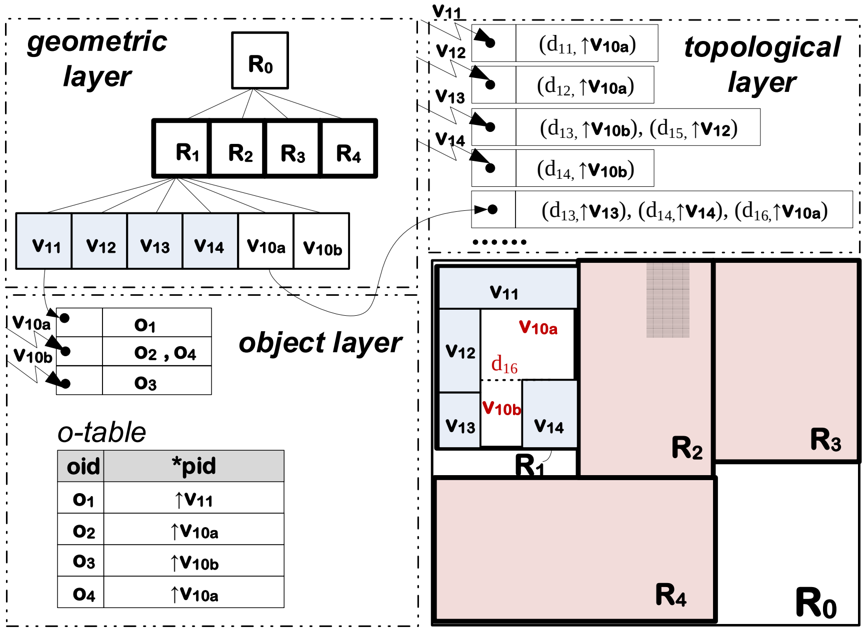
* *Hua Lu, Xin Cao, and Christian S Jensen. 2012. A foundation for efficient indoor distance-aware query processing. In ICDE. 438–449.*

Model and Indexes

- Composite Indoor Index (CIndex) *



An example of the floorplan

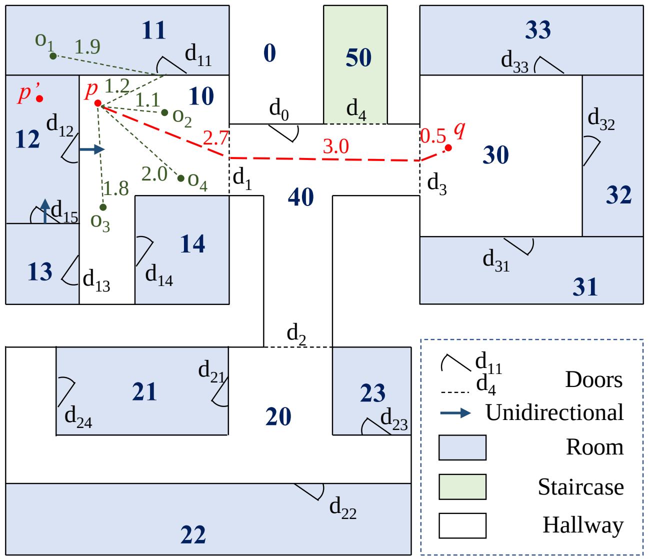


An example of CIndex

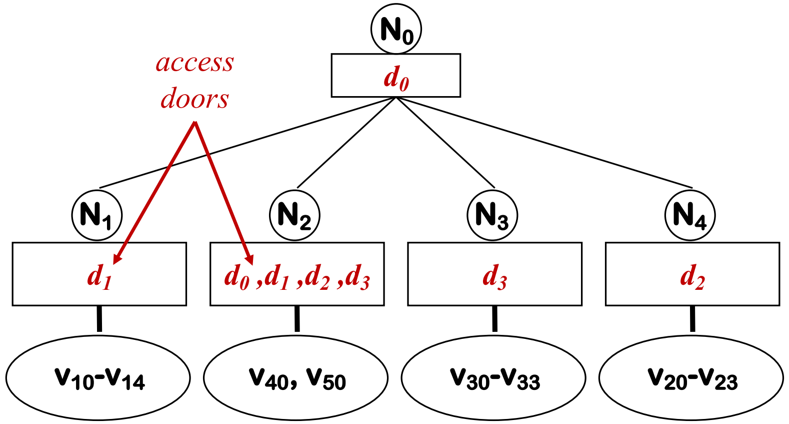
* *Xike Xie, Hua Lu, and Torben Bach Pedersen. 2013. Efficient distance-aware query evaluation on indoor moving objects. In ICDE. 434–445.*

Model and Indexes

- Indoor Partitioning Tree (IP-Tree) and Vivid IP-Tree(VIP-Tree)*



An example of the floorplan



	d_0	d_1	d_2	d_3
d_0	0	1.4	2	3.9
d_1	1.4	0	3	4
d_2	3.9	4	4.4	0
d_3	2	3	0	4.4

Distance Matrix for $\mathbf{N_0}$ (a non-leaf node)

	d_1	d_{11}	d_{12}	d_{13}	d_{14}	d_{15}
d_1	0	1.7	2.7	3.2	2.8	4.3, d_{12}

Distance Matrix for $\mathbf{N_1}$ (a leaf node)

An example of IP-Tree

* *Zhou Shao, Muhammad Aamir Cheema, David Taniar, and Hua Lu. 2016. Vip-tree: an effective index for indoor spatial queries. PVLDB 10, 4, 325–336.*

Contents

- Indoor Spatial Queries
- Model and Indexes
- Query Processing
  - Feature Comparison
  - Complexity Analysis
  - Extensibility Analysis
- Benchmark
- Results Summary
- Future Work

Query Processing

- Feature Comparison

Table 2: Feature Comparison

Models	Precompute	Structure	Initialization	Expansion	RQ	kNNQ	SPQ	SDQ
IDMODEL	No	Graph+ Mappings	Sequential scan	Dijkstra	△	△	✓	✓
IDINDEX	Yes	Matrix	Sequential scan	Loop	✓	✓	△	△
CINDEX	No	Tree+Links	R*-Tree pruning	Dijkstra	✓	✓	△	△
IP-TREE	Yes	Tree+Matrix	Sequential scan	LCA	✓	✓	✓	✓
VIP-TREE	Yes	Tree+Matrix	Sequential scan	LCA	✓	✓	✓	✓

Query Processing

- Complexity Analysis

Table 4: Complexity Analysis

	Space	RQ	kNNQ	SDQ	SPQ
IDMODEL	$\mathcal{O}(V + D + 2Vd + Vd^2)$	$\mathcal{O}(oV \log D)$	$\mathcal{O}(oV \log D)$	$\mathcal{O}(V \log D)$	$\mathcal{O}(V \log D + w)$
IDINDEX	$\mathcal{O}(2D^2)$	$\mathcal{O}(od \log D)$	$\mathcal{O}(od \log D)$	$\mathcal{O}(d^2)$	$\mathcal{O}(d^2 + w)$
CINDEX	$\mathcal{O}(V + Vd + 0)$	$\mathcal{O}(oV \log D)$	$\mathcal{O}(oV \log D)$	$\mathcal{O}(V \log D)$	$\mathcal{O}(V \log D + w)$
IP-TREE	$\mathcal{O}(\rho^2 f^2 L + \rho D)$	$\mathcal{O}((\rho \log_f L)^2 (Vo/L + \rho))$	$\mathcal{O}((\rho \log_f L)^2 (Vo/L + \rho))$	$\mathcal{O}(\rho^2 \log_f L)$	$\mathcal{O}((\rho^2 + w) \log_f L)$
VIP-TREE	$\mathcal{O}(\rho^2 f^2 L + \rho D \log_f L)$	$\mathcal{O}(\rho^2 \log_f L(Vo/L + \rho))$	$\mathcal{O}(\rho^2 \log_f L(Vo/L + \rho))$	$\mathcal{O}(\rho^2)$	$\mathcal{O}(\rho^2 + w)$

- Extensibility Analysis

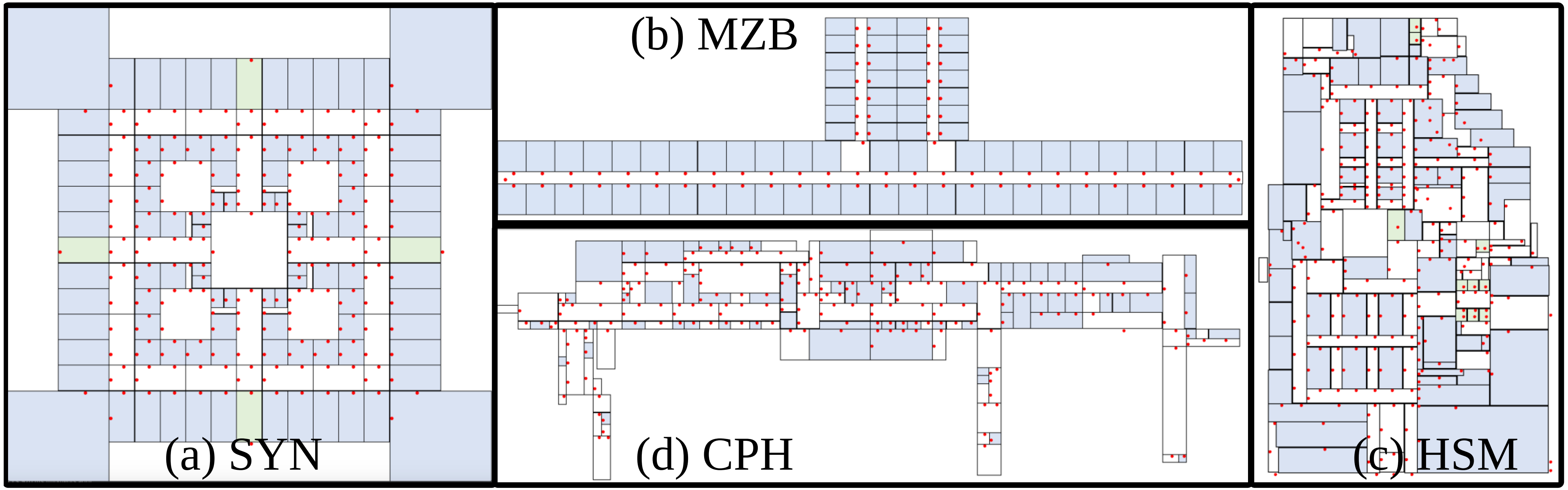
Table 3: Extensibility Analysis

	IDMODEL	IDINDEX	CINDEX	IP/VIP-TREE
Temporal Variation	✓	X	✓	X
Moving Objects	✓	✓	✓	✓
Uncertain Locations	X	X	✓	X
Keywords	✓	✓	✓	✓

Contents

Benchmark

- Datasets



Floorplan of Datasets

Benchmark

- Performance Evaluation Procedure

A. Model Construction.

(a1) model/index size and (a2) construction time.

B. Query Processing.

(b1) running time, (b2) memory use, and (b3) number of visited doors (NVD) for SPDQ.

B1. Effect of Floor Number n.

B2. Effect of Object Number |O|.

B3. Effect of Range Distance r .

B4. Effect of k.

B5. Effect of Source-Target Distance s2t.

B6. Effect of Topological Change.

B7. Effect of Hallway's Decomposition Method.

Benchmark

- Performance Evaluation Procedure

Table 6: Evaluation Settings (Default Parameters in Bold)

Symbol & Meaning		Task	Metrics	Queries	Dataset	Parameter Setting
n	floor number	A B1	a1, a2 b1, b2, b3 (only for SPDQ)	- RQ, kNNQ, SPDQ	SYN	3, **5**, 7, 9
$\|O\|$	object number	B2	b1, b2	RQ, kNNQ	all	500, 1000, **1500**, 2000, 2500
r	range value	B3	b1, b2	RQ	SYN5, HZM, CPH MZB	200, 400, **600**, 800, 1000 20, 40, **60**, 80, 100
k	-	B4	b1, b2	kNNQ	all	1, 5, **10**, 50, 100
s2t	source-target distance	B5	b1, b2, b3	SPDQ	SYN5, HZM, CPH MZB	1100, 1300, **1500**, 1700, 1900 30, 60, **90**, 120, 150
-	topological change	B6	b1, b2, b3 (only for SPDQ)	RQ, kNNQ, SPDQ	SYN	$SYN5^-$, SYN5, $SYN5^+$
-	decomposition method	B7	b1, b2, b3 (only for SPDQ)	RQ, kNNQ, SPDQ	SYN MZB	$SYN5^0$, SYN5 MZB^0, MZB, MZB^Δ

Contents

- Indoor Spatial Queries
- Model and Indexes
- Query Processing
- Benchmark
- Results Summary
- Future Work

Results Summary

Table 13: Summary of Findings

Model	**Construction Cost**		**RQ/kNNQ Search**		**SPQ/SDQ Search**	
	Model Size	Time	Memory	Time	Memory	Time
IDMODEL	★★★★★	★★★★★	★★★★★	★★★	★★★★★	★
IDINDEX	★	★	★	★★★★★	★	★★★★★
CINDEX	★★★★	★★★★	★★★★★	★★★	★★★★★	★
IP-TREE	★★★	★★★	★★★★	★	★★★★	★★★
VIP-TREE	★★	★★★	★★★	★★	★★★	★★★★

- IDIndex is preferred for small-scale spaces.
- VIP-Tree is recommended if routing is the task.
- IDModel is recommended for non-routing queries due to its low construction cost and good balance between storage and query time costs.

Contents

- Indoor Spatial Queries
- Model and Indexes
- Query Processing
- Benchmark
- Results Summary
- Future Work

Future Work

- Heuristics algorithms can replace the Dijkstra-based expansion in IDModel and CIndex.
- Strategies to select crucial doors/partitions can be developed to reduce the storage of door-to-door distances in IDIndex and IP-Tree/VIP-Tree while preserving their search efficiency.
- Indoor Moving objects.

Thanks!

Email: liutt@cs.aau.dk