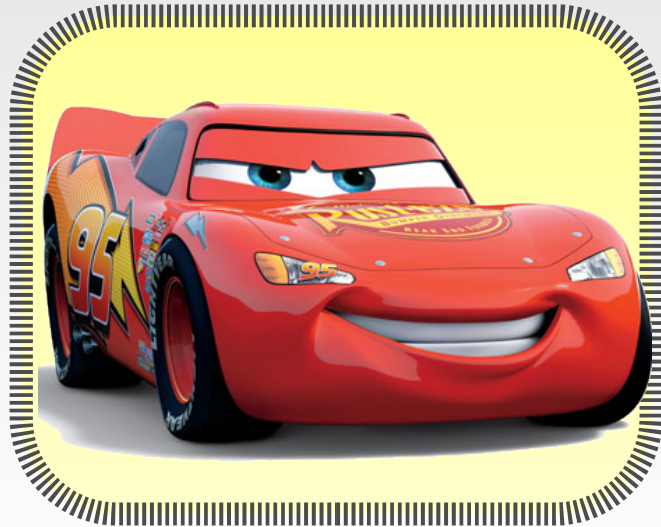
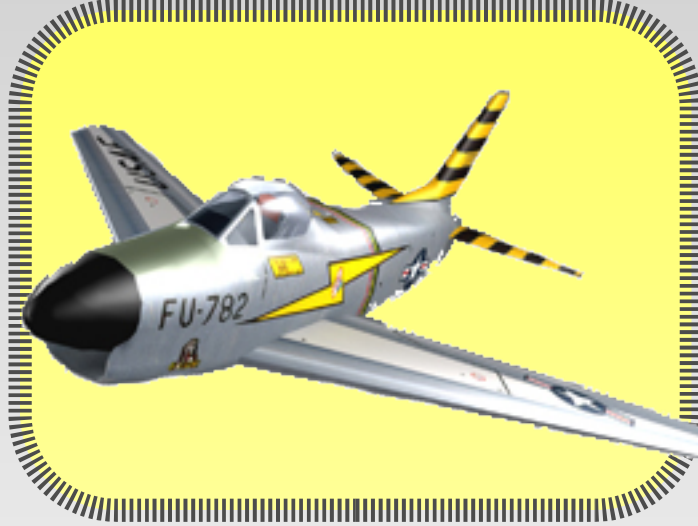
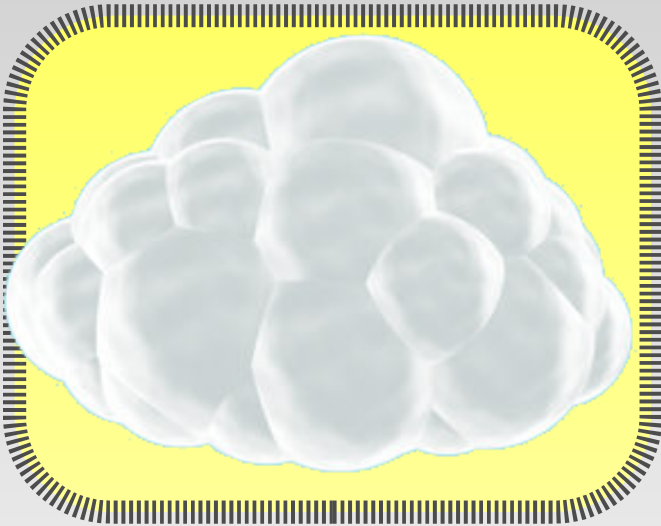


ST² B-tree...

... A Self-Tunable Spatio-Temporal B⁺-tree Index for Moving Objects



Authors:

Su Chen, Beng Chin Ooi, Kian-Lee Tan,
Mario A. Nascimento

Presenter:

Lucia Ambrošová

Supervisor:
Shepherd:

Prof. Dr. Peter Widmayer
Georgios Giannikis

Introduction

Moving Object Databases (MODs):

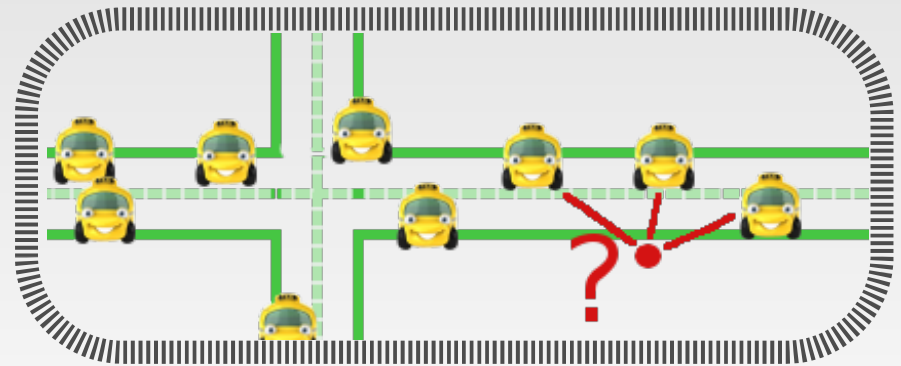
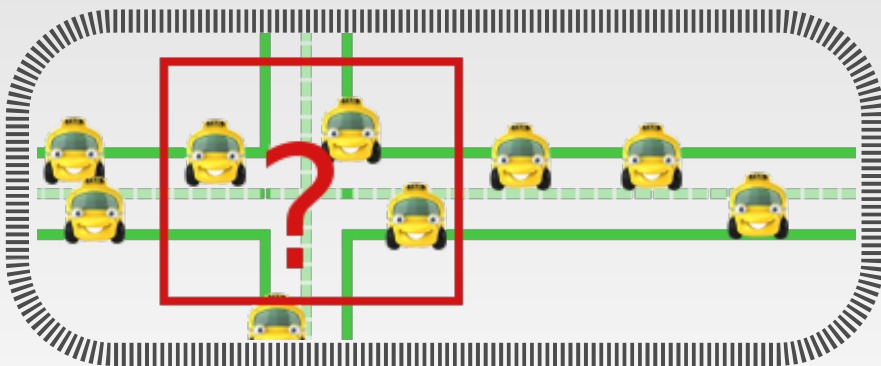
- moving objects in multidimensional space
- dynamic attribute values
- need special index structures
 - efficient for updates
 - adaptive to data distribution diversity



MOD Queries

Predictive spatial queries:

- Range queries
- kNN (k nearest neighbors) queries

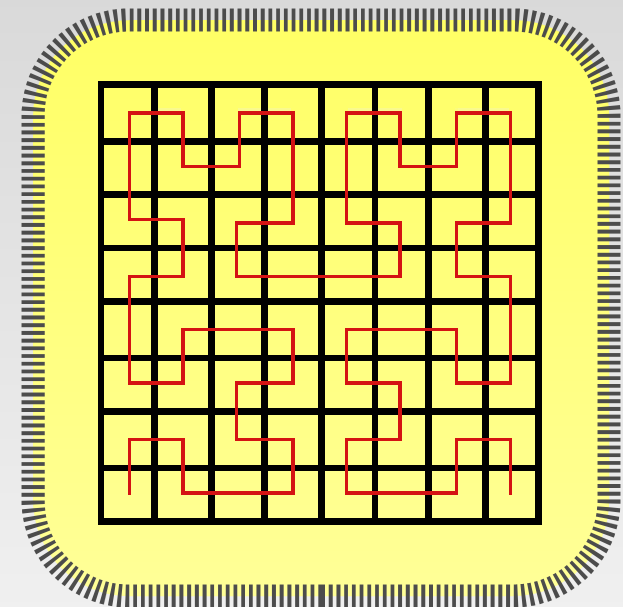
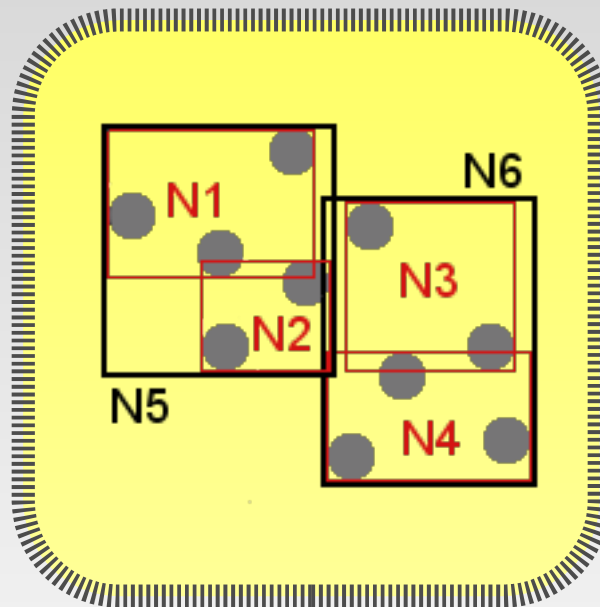


Space Representation

MOD queries require **spatial locality**

Existing representations:

- The TPR*-tree
- The B^x-tree



Good space representation → fast query execution

Dynamic Attributes

Object locations are dynamic

- Estimate locations between updates
- At query time:
 - TPR*-tree: bounding rectangle adjustment
 - B^x-tree: query window adjustment

Location updates have to be frequent

- update efficiency is crucial

TPR*-tree vs. B^x-tree

TPR*-tree

- Query efficiency.....
 - MBR enlargement**
 - + few false positives
 - performed many times, slower
- Update efficiency.....
 - Maintains MBR hierarchy**
 - inefficient
- Data diversity adjustment.....
 - Partitioning sensitive to data**
 - + diversity-friendly

B^x-tree

Query enlargement

- more false positives
- + simpler procedure, more efficient

Efficient B⁺-tree updates

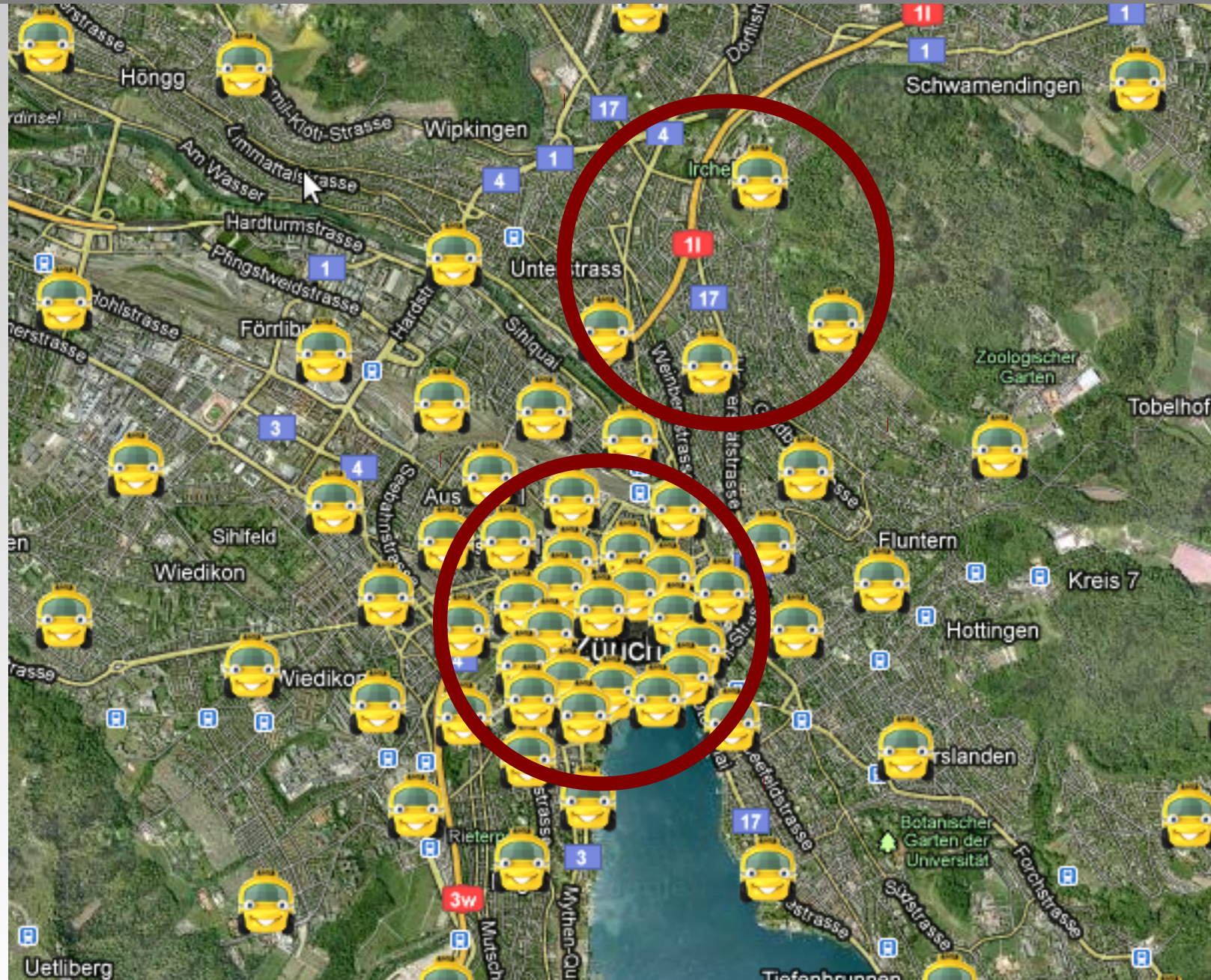
- + very fast

Uniform space partitioning

- likely to produce overflow pages

Spatial Data Diversity

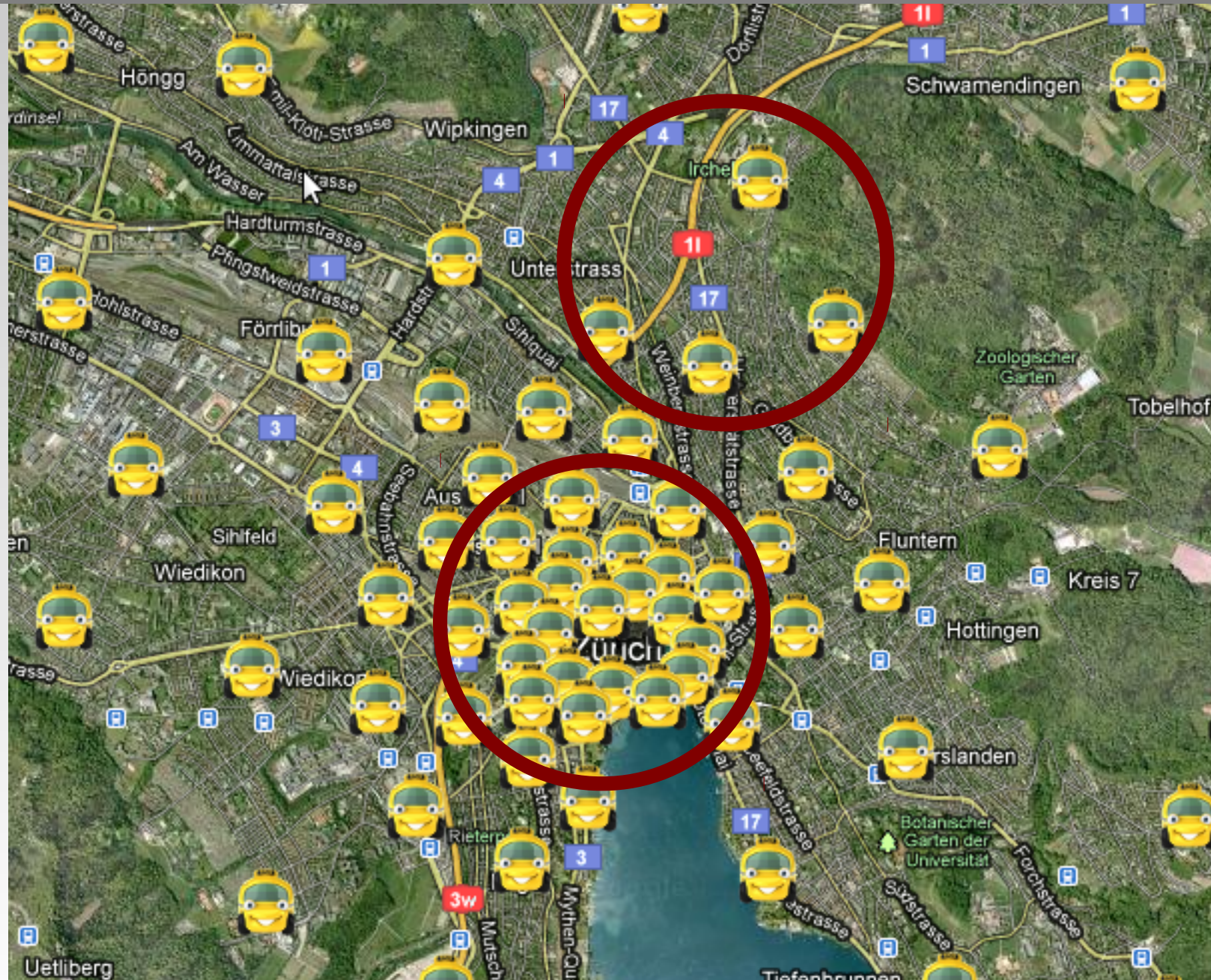
Zürich
- day time



Temporal Data Diversity

Zürich

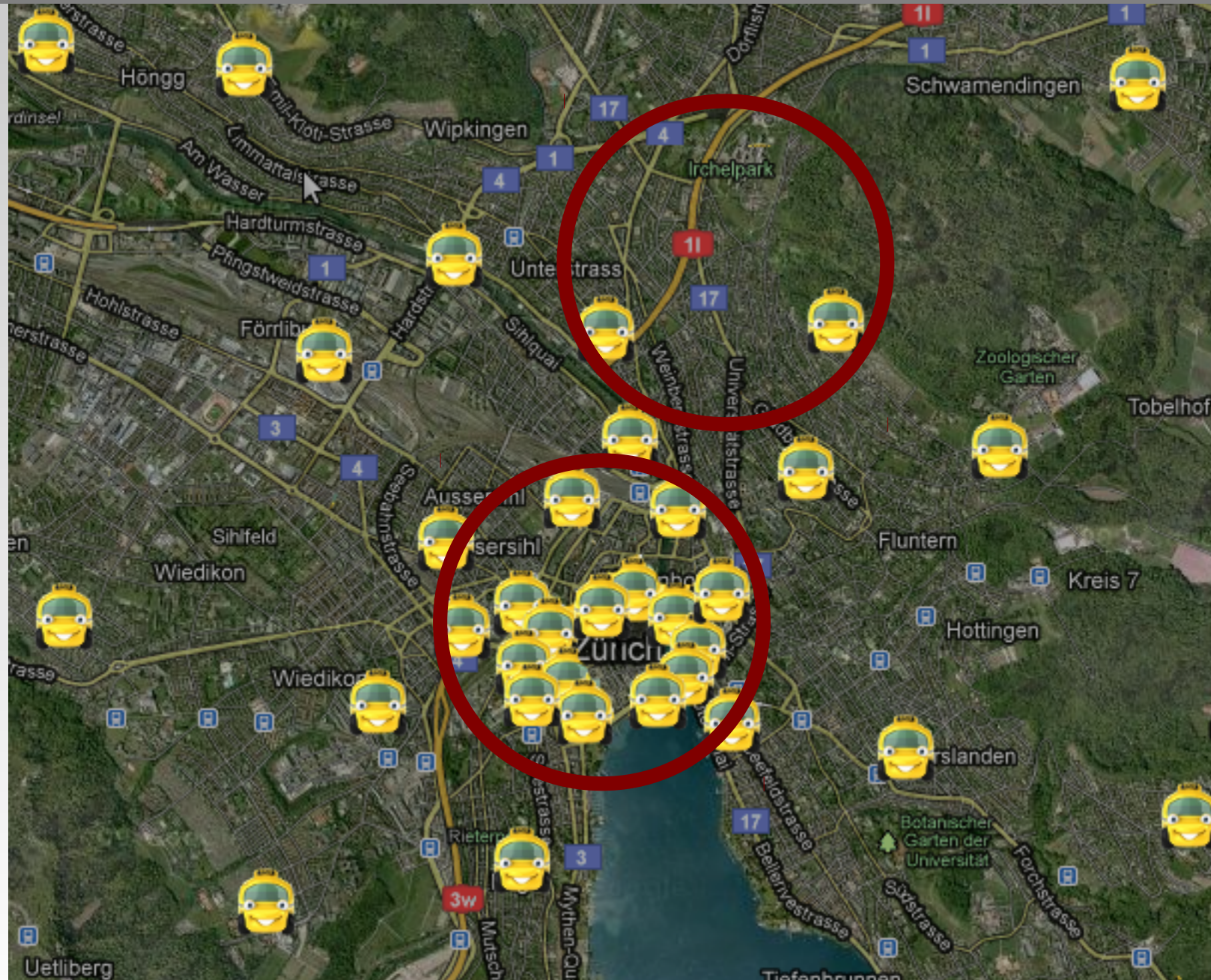
- day-night
transition



Temporal Data Diversity

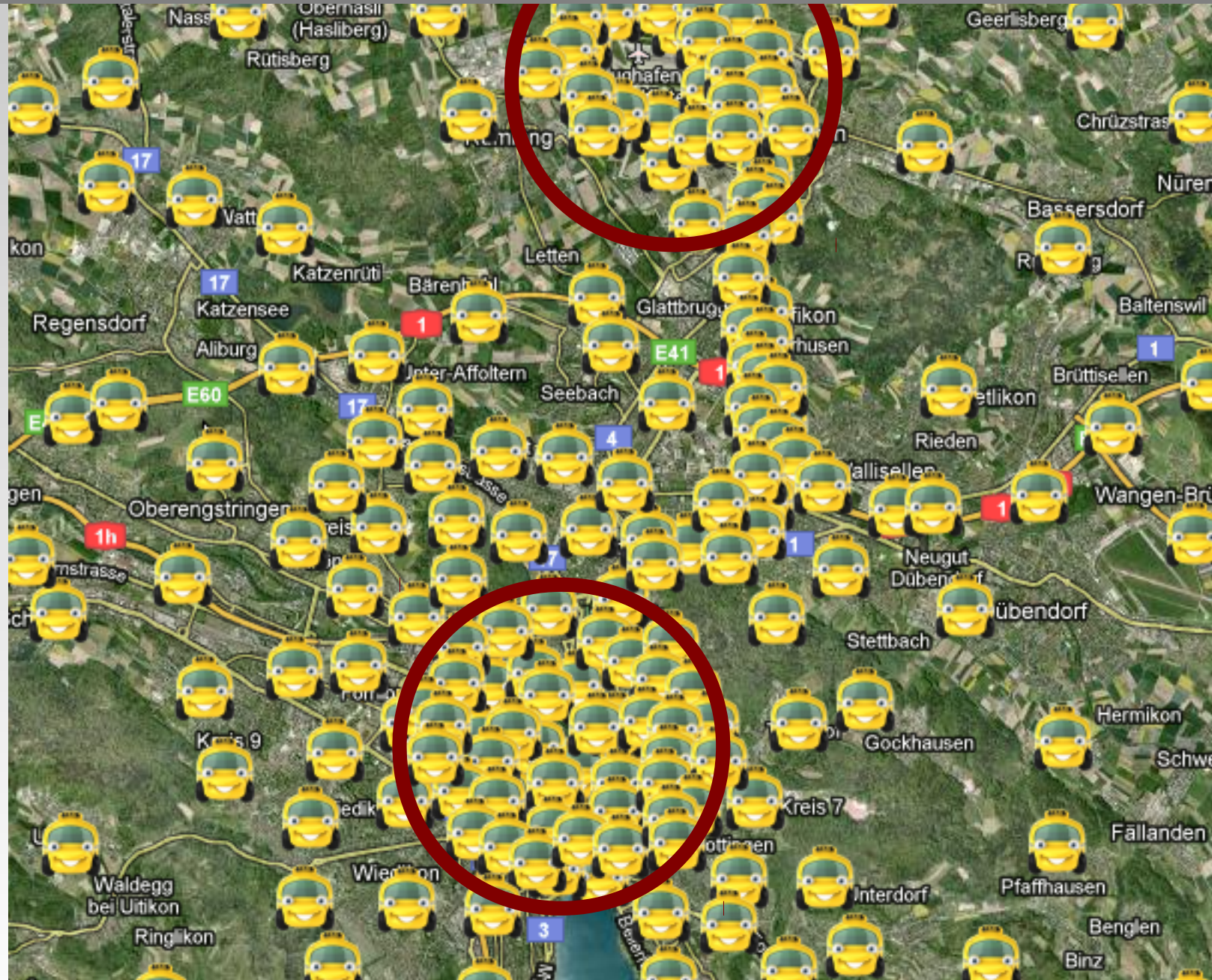
Zürich

- day-night
transition



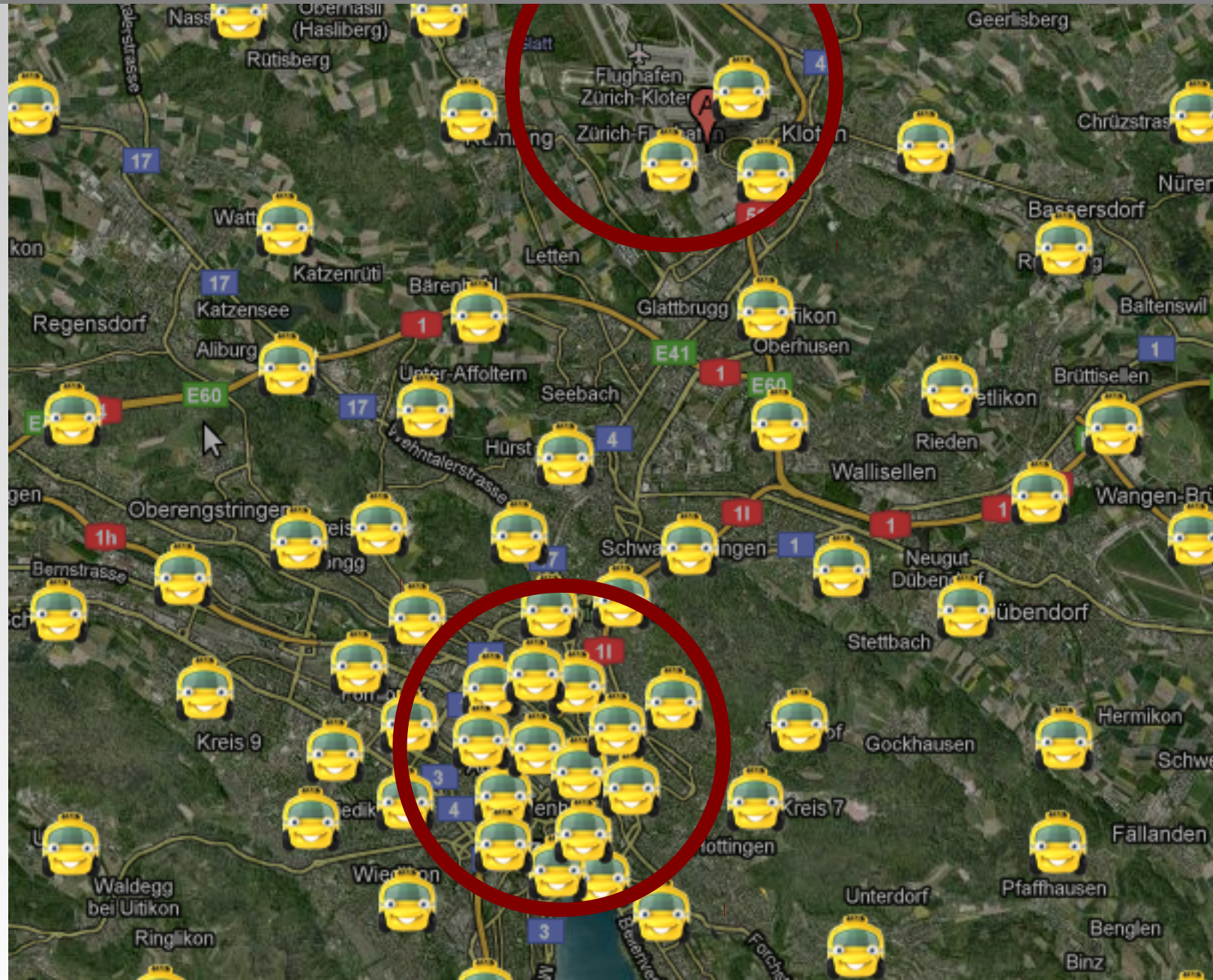
Spatio-Temporal Data Diversity

**Zürich +
Airport**
- day-night
transition



Spatio-Temporal Data Diversity

**Zürich +
Airport**
- day-night
transition



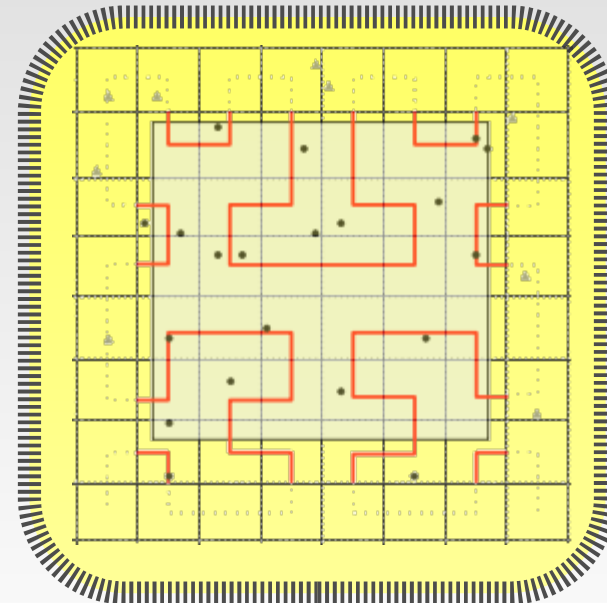
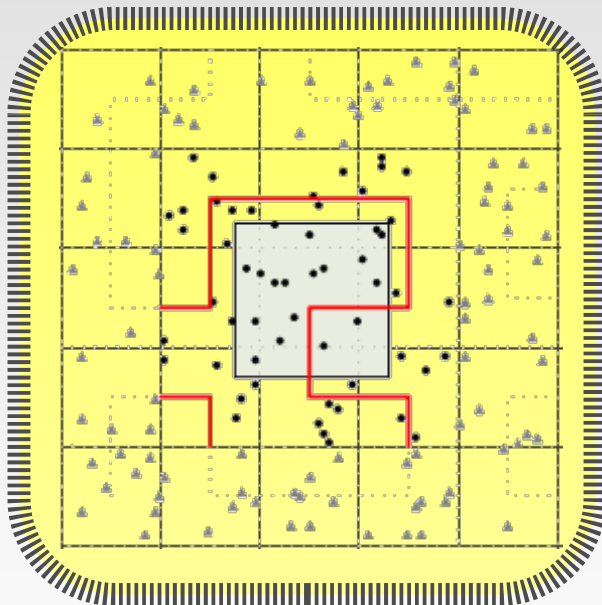
Grid Granularity Effects

Grid granularity impacts B^x-tree performance

• **Coarse grid**.....**Medium grid**.....**Fine grid**

Diversity-adaptable grid size is advantageous

• Uniform grid on uneven data → performance degradation



Goals

A good moving object index can:

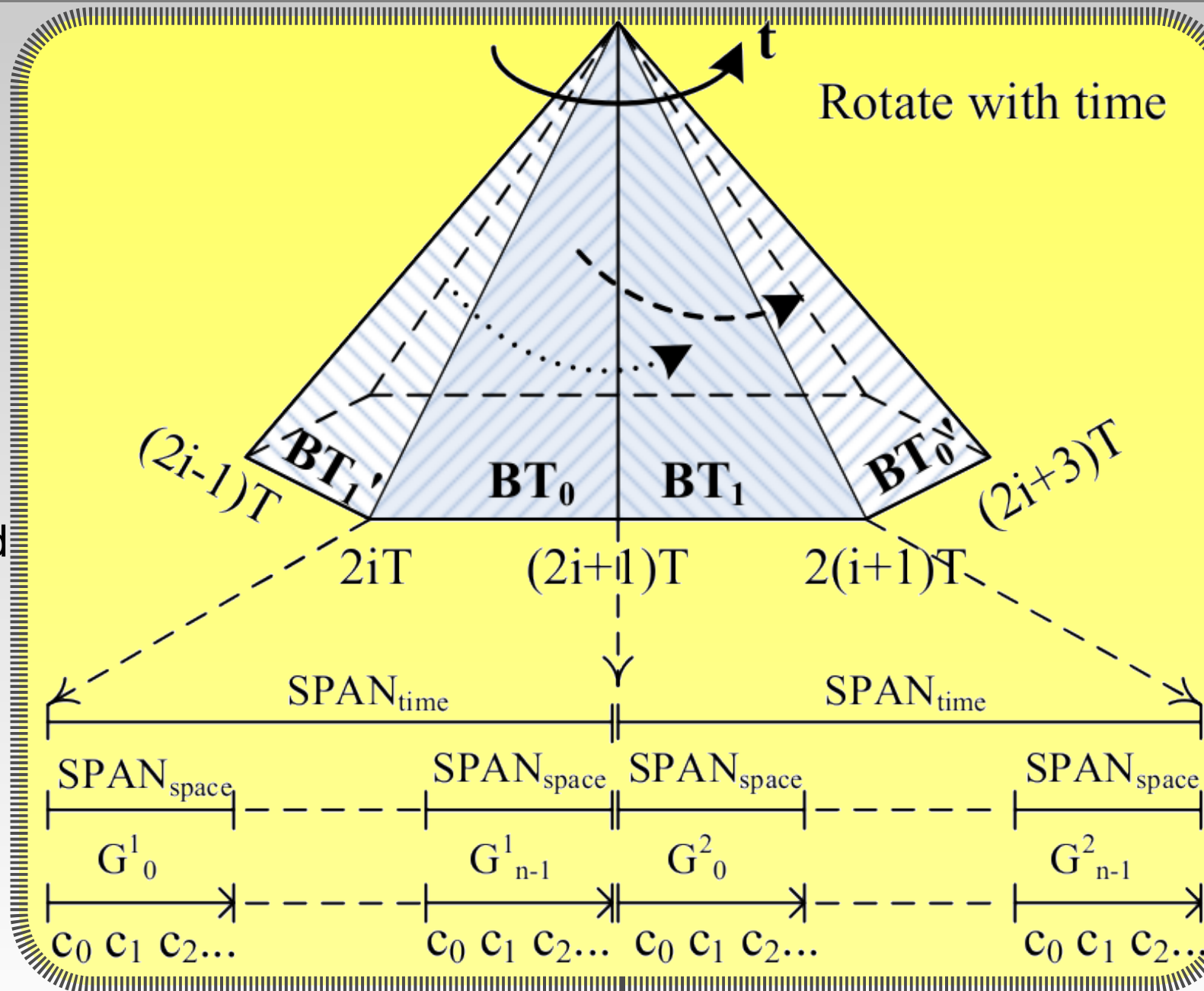
- discriminate between regions of different densities
- adapt to density and distribution changes with time

Additional benefit factors:

- ease of implementation
- support by many commercial databases
- fast concurrent insert/delete operations

The ST²B-Tree

- built on top of the B⁺-tree
- 2 subtrees: successive time periods
- space partitioned into regions of equal densities
- regions refined at rollover time



Space Partitioning

Space is partitioned by a set of reference points $RP_1 \dots RP_N$

- each point RP_j defines a Voronoi region VC_j

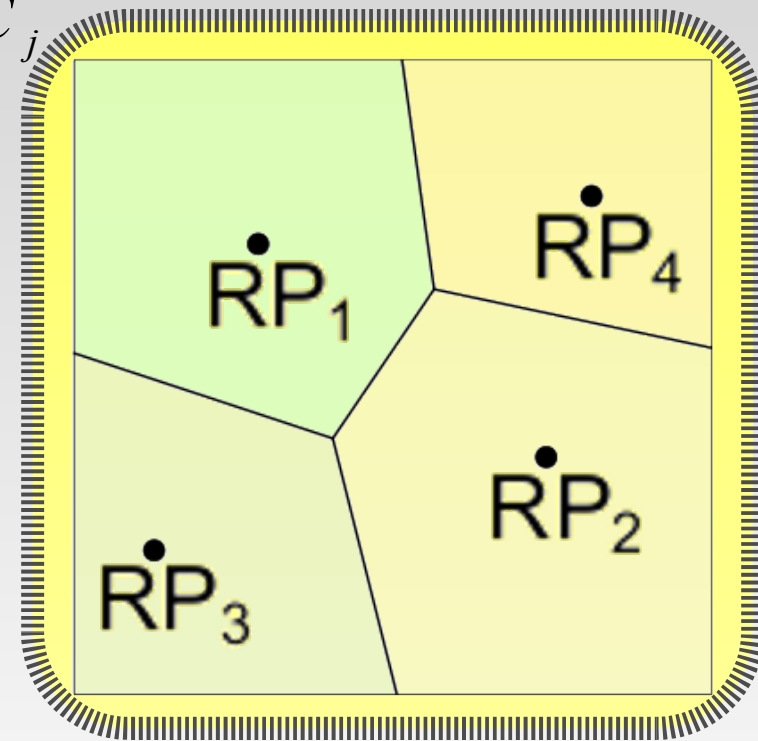
Regions only roughly “equidense”

- optimality is not required
- efficiency is a priority

Each region VC_j has its own grid G_j

- granularity tuned to local object density

→ Accounts for spatial data diversity!

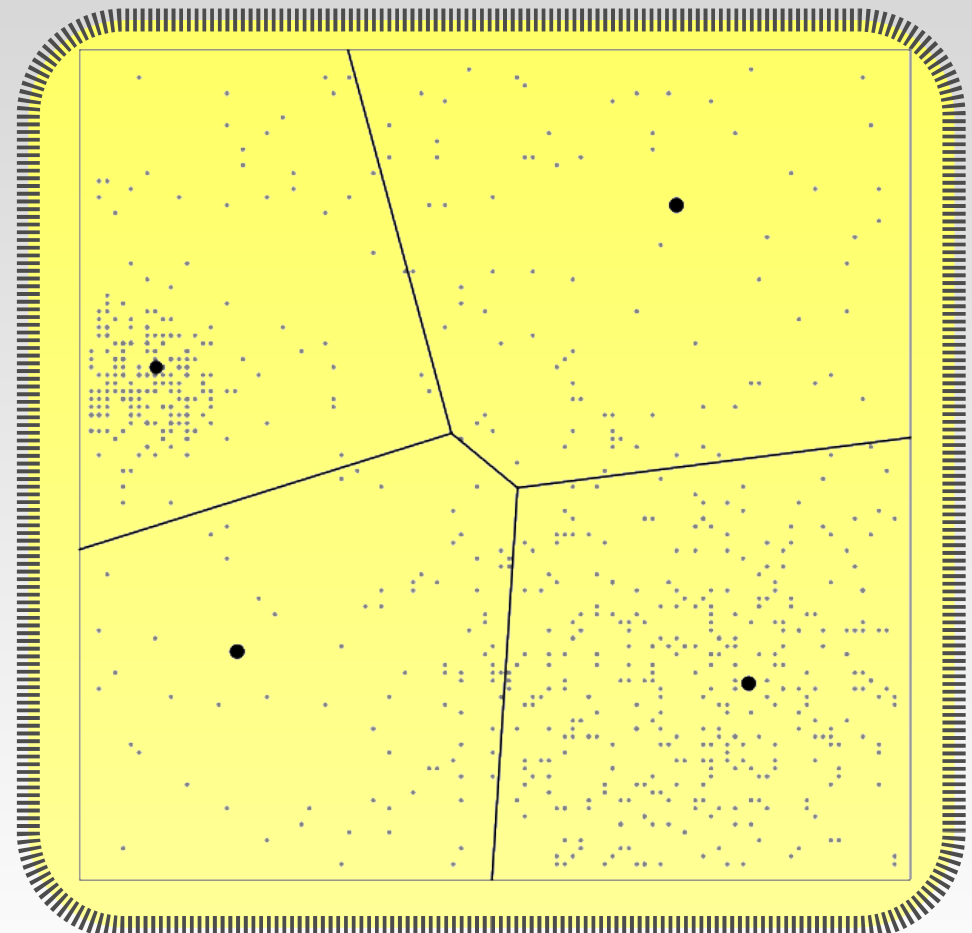


Region Growing

Find new reference points using a histogram of densities

- use old reference points are seeds
- grow regions based on cell similarity

1	2	3	3	4	3	2	3	2	2
6	4	4	3	3	4	1	1	1	2
5	3	0	2	1	2	1	1	2	2
68	30	4	1	3	4	3	1	3	3
51	35	1	3	4	3	10	5	1	3
4	3	0	3	4	5	9	11	6	7
1	1	2	3	8	9	14	15	18	7
4	2	2	3	7	22	15	28	23	18
2	1	0	2	6	14	16	24	15	16
1	1	3	4	3	8	16	14	12	3



Tuning Pop Quiz!

- (1) What are the arguments against a uniform grid?
- (2) What are the 3 types of data diversities?
- (3) Does the tuning procedure account for all 3 types?

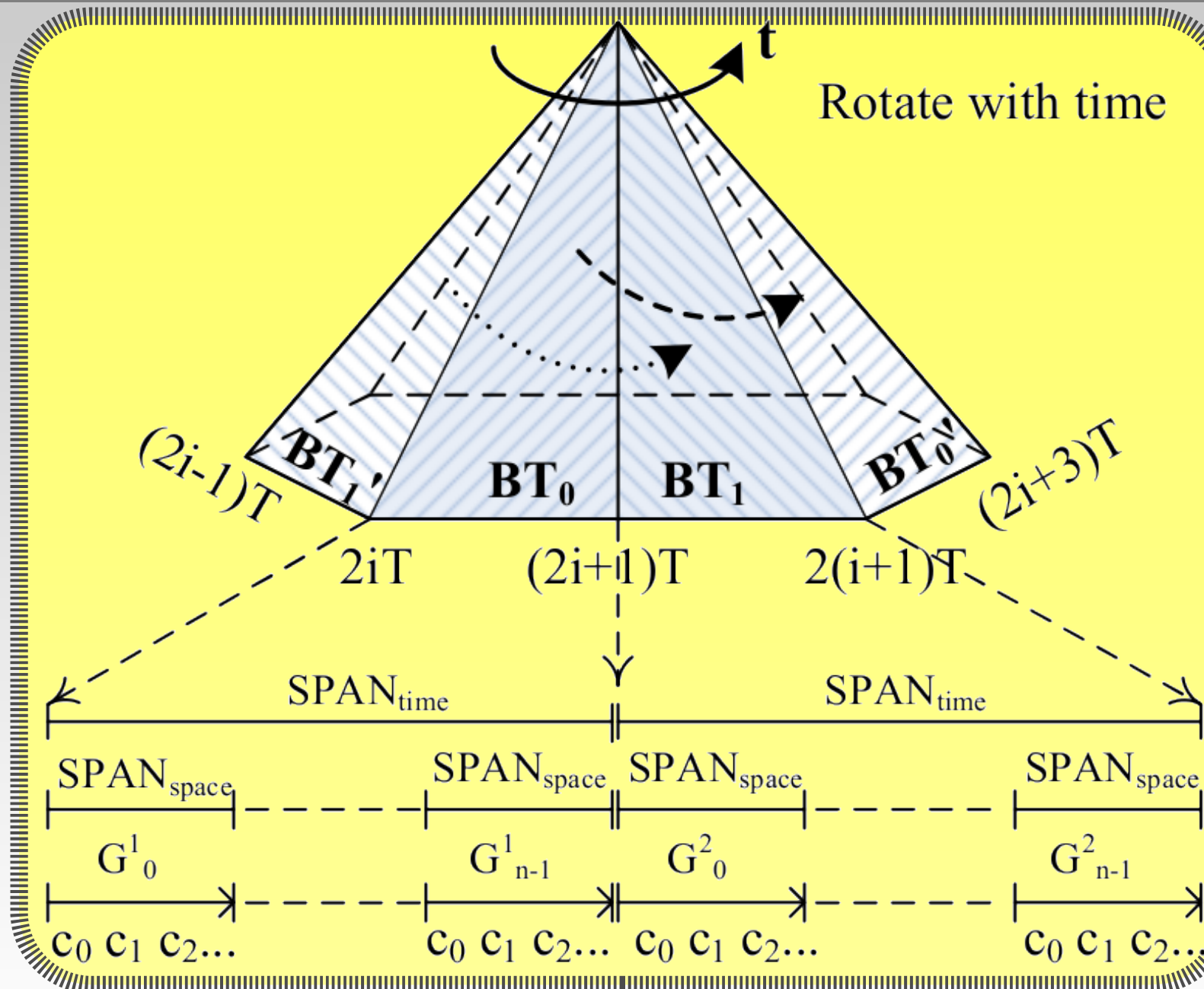
Query Evaluation

range queries

- region R_q
- time t_q

kNN queries

- iterative



Performance Evaluation

Comparison with the B^x-tree and the TPR*-tree

Randomly generated objects

- speeds {30, 60, 90, 150, 300} km/h
- experiments with non-uniform distribution: k hotspots

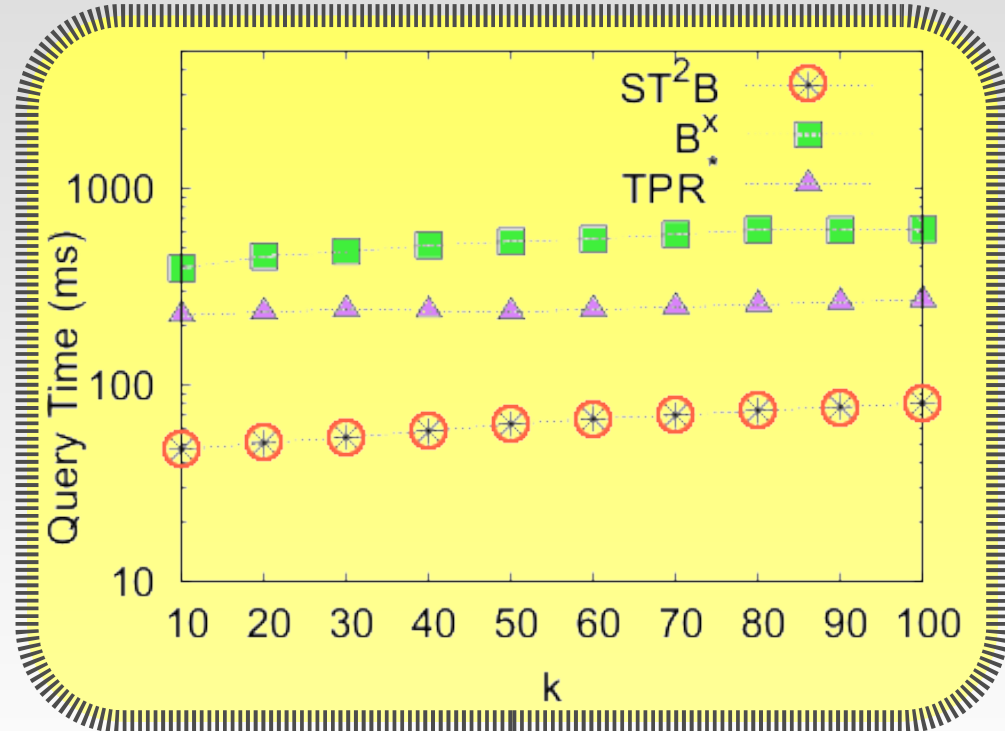
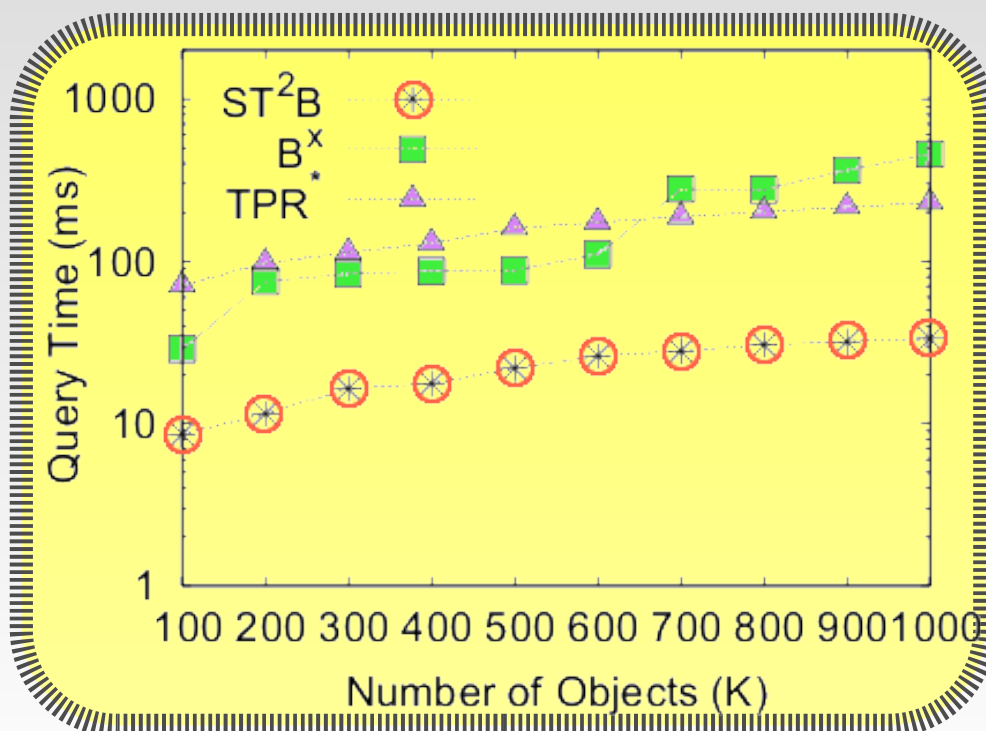
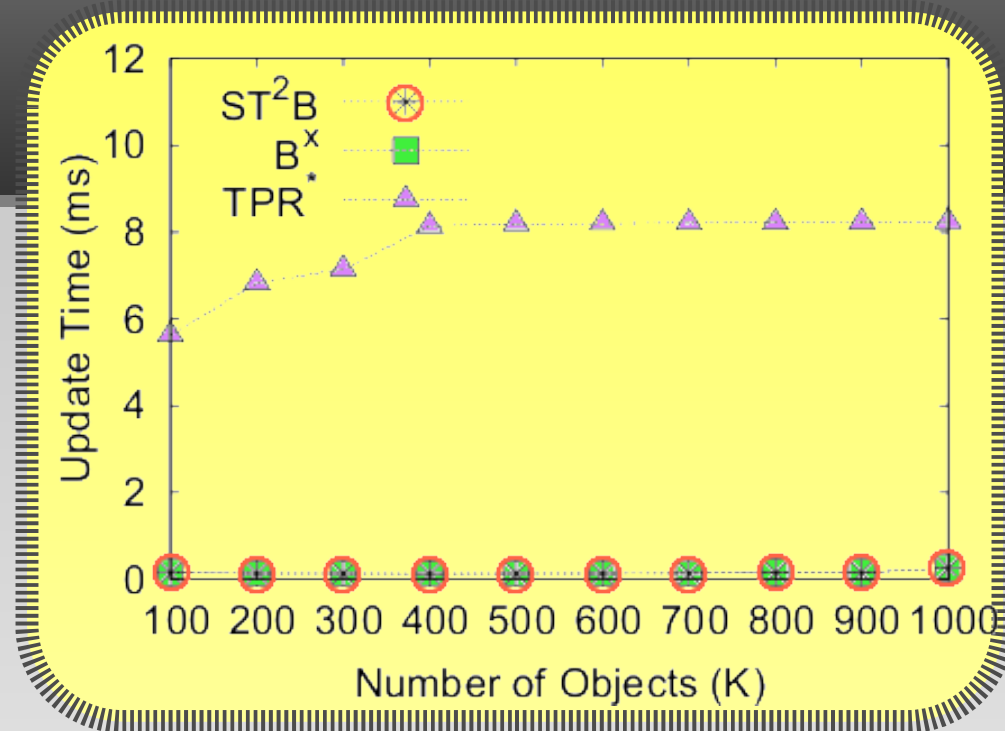
Update-to-query ratio: 100:1

PC specifications

- Pentium IV 3.0GHz processor, 1GB memory, 80G SATA Disk, Windows XP
- disk page size 4KB, LRU buffer of 50 pages

Spatial Diversity

- non-uniform object distribution
- $k = 10$ hotspots
- 1000 objects

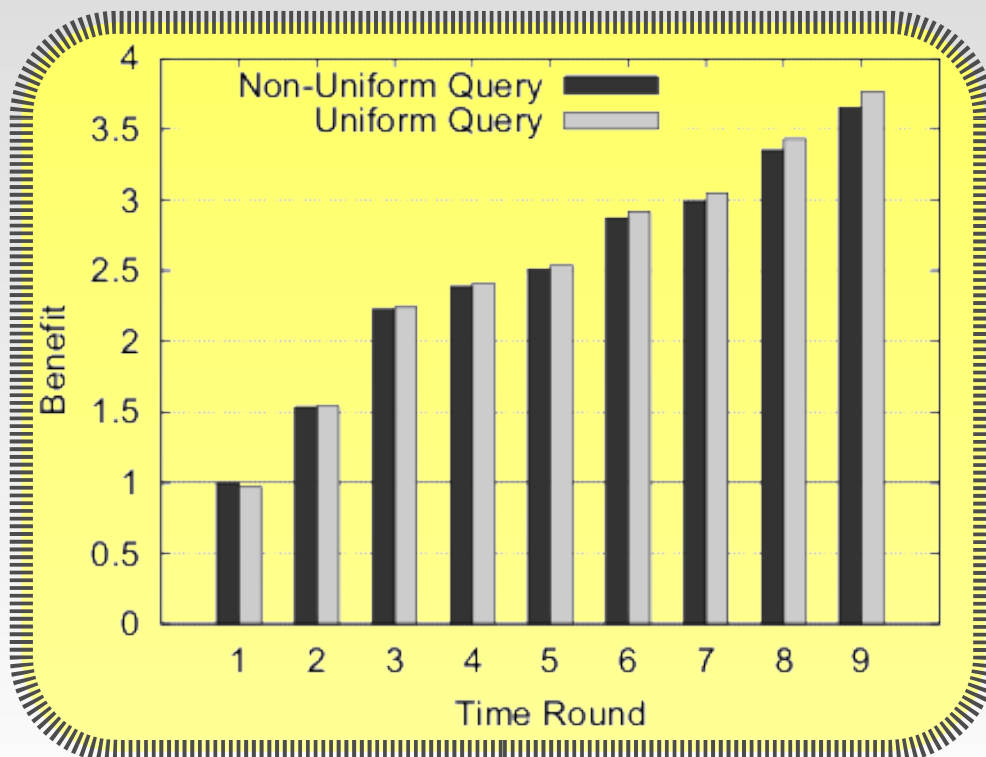


Temporal Diversity

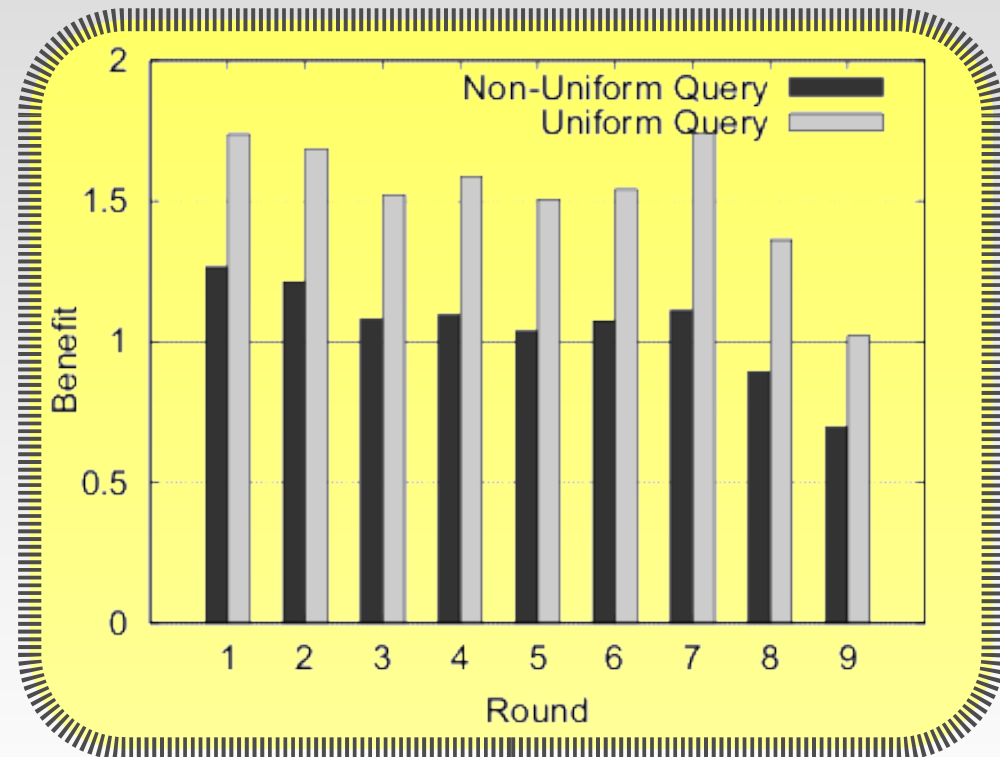
$$\text{benefit} = \frac{\text{total processing time of the } B^x\text{-tree}}{\text{total processing time of the } ST^2 B\text{-tree}}$$

- non-uniform distribution, $k = 10$ hotspots
- number of objects between 100K and 1M

1) Increasing number of objects



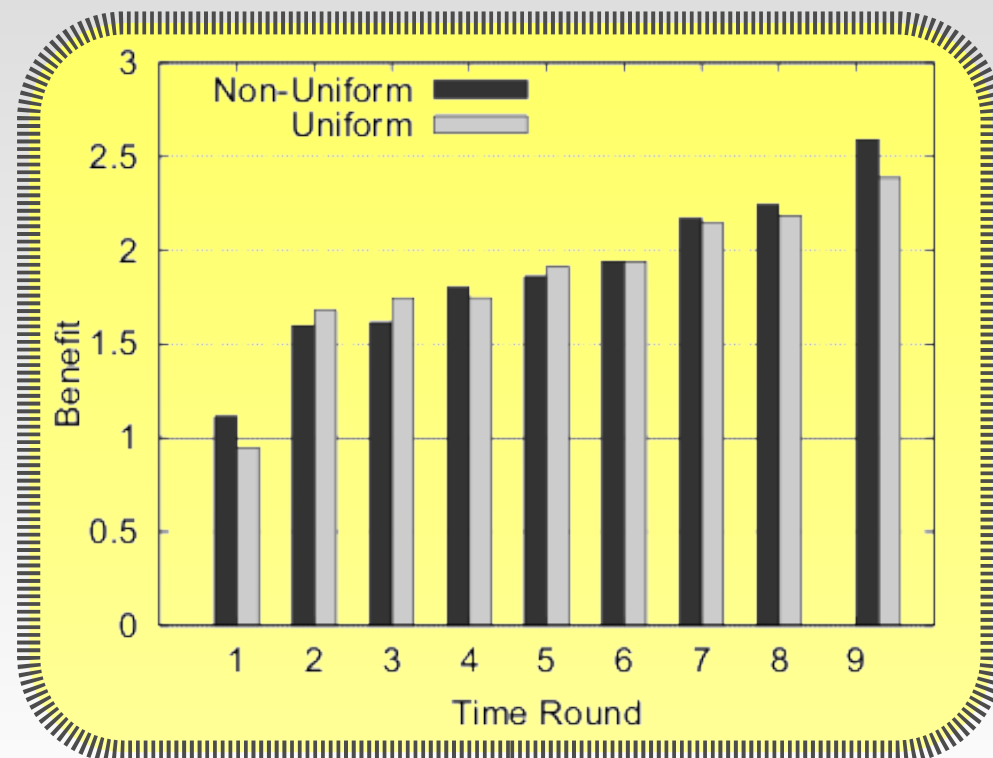
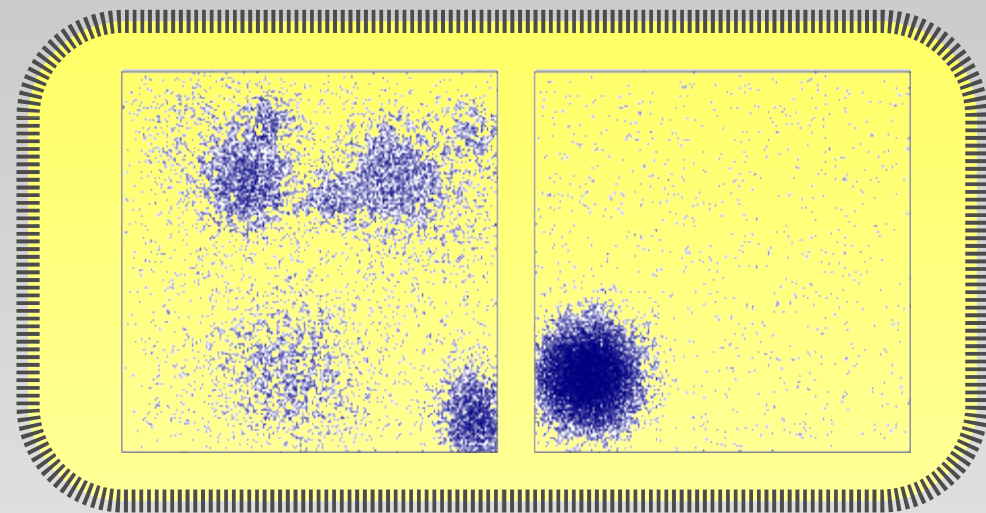
2) Decreasing number of objects



Spatio-Temporal Diversity

- non-uniform data distribution
- 1M objects

Increasing data skew

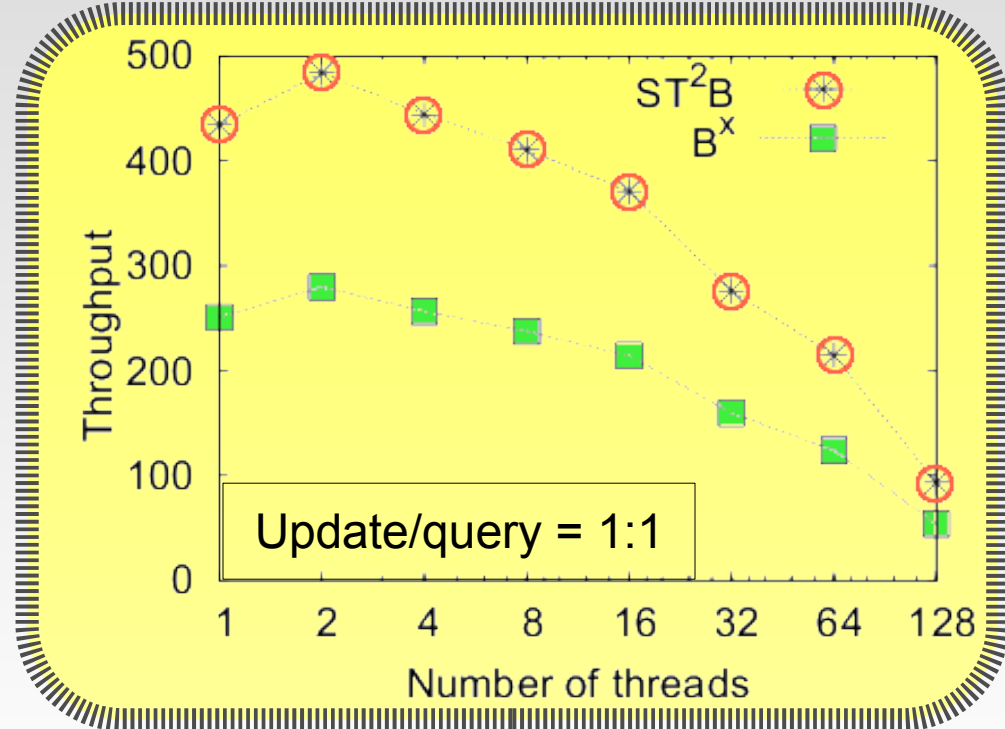
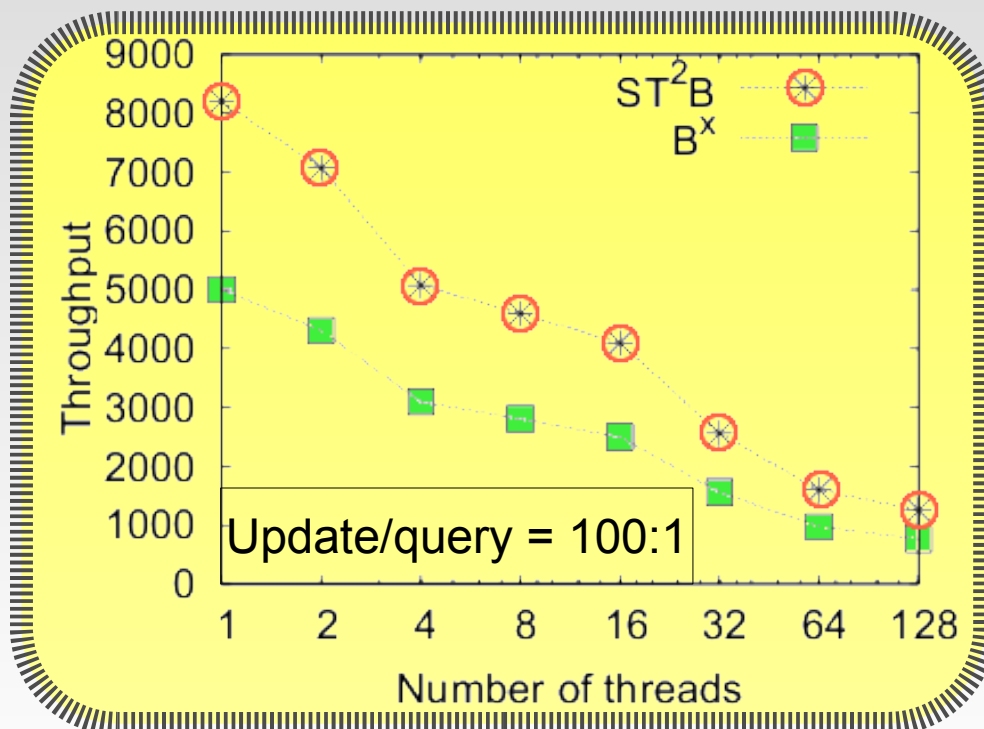
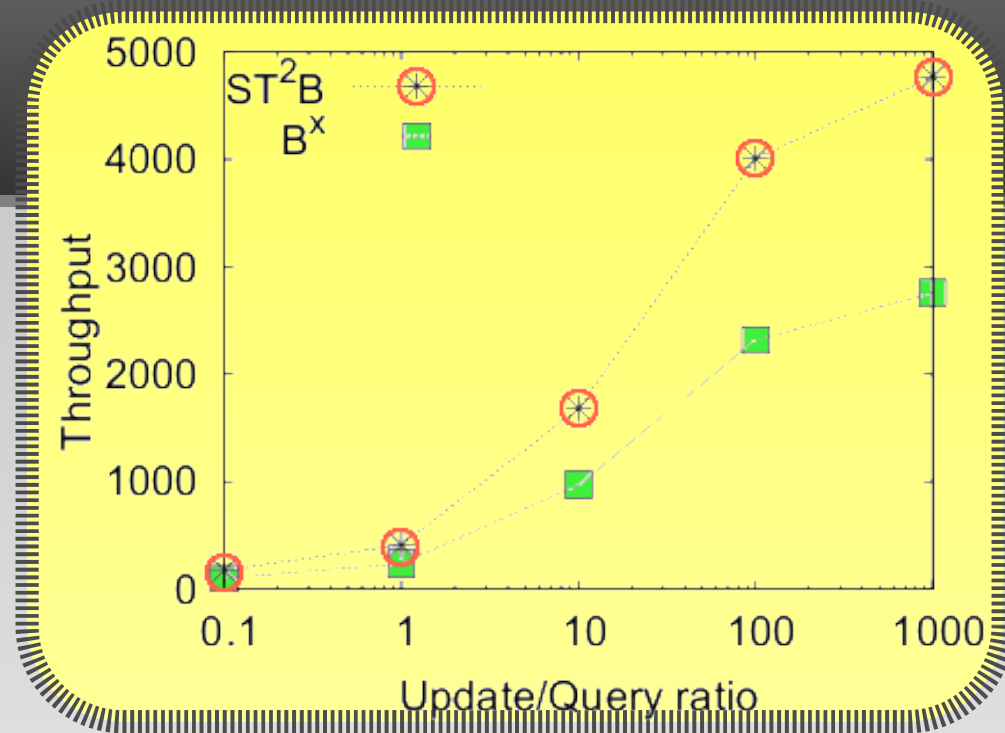


Throughput Test

Throughput

= number of operations per second

- 1M objects, non-uniform distribution
- thread workload: 200 non-uniform range queries



Goals Revisited

A good moving object index can:

- discriminate between regions of different densities
- adapt to density and distribution changes with time

Additional benefit factors:

- ease of implementation
- support by many commercial databases
- fast concurrent insert/delete operations

Conclusion

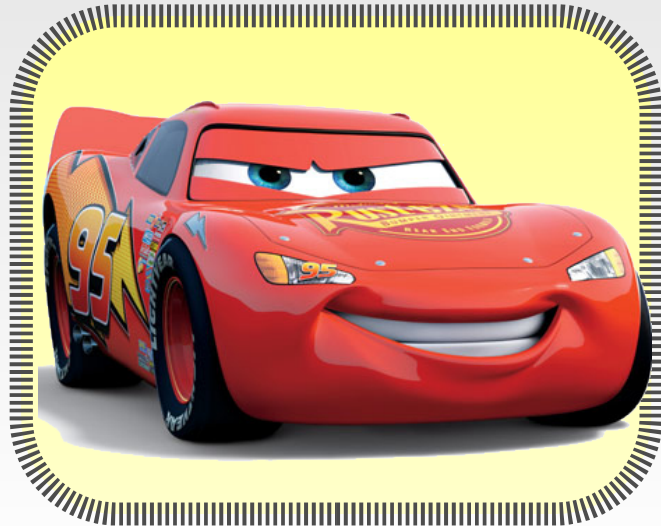
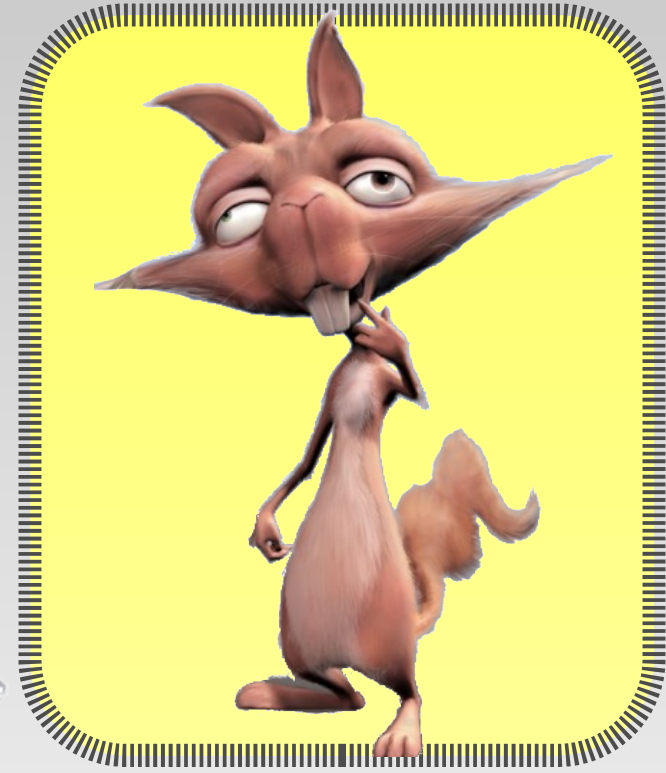
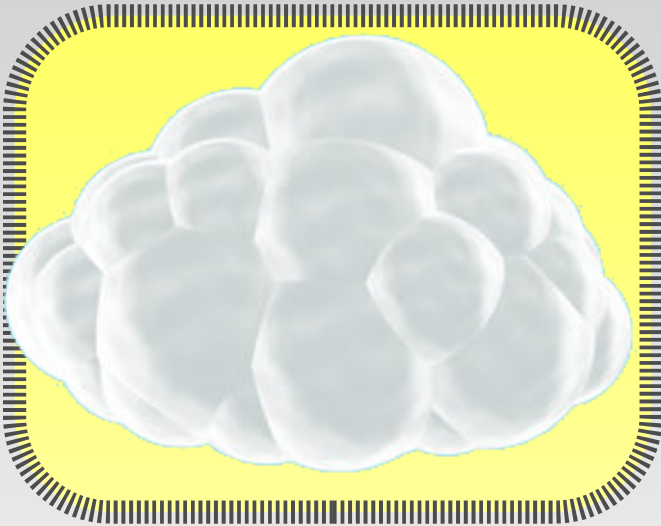
The ST²B-tree...

- achieves all goals
- efficient, robust, scalable
- outperforms its competitors in the experimental setting



Thank you...

... for your attention!



Questions?

References

- [1] S. Chen, B.C. Ooi, K.L. Tan, and M.A. Nascimento. ST 2 B-tree: a self-tunable spatio-temporal B+-tree index for moving objects. In Proceedings of the 2008 ACM SIGMOD international conference on Management of data, pages 29–42. ACM New York, NY, USA, 2008.
- [2] C.S. Jensen, D. Lin, and B.C. Ooi. Query and update efficient B+-tree based indexing of moving objects. In Proceedings of the Thirtieth international conference on Very large data bases-Volume 30, page 779. VLDB Endowment, 2004.
- [3] R. Ramakrishnan and J. Gehrke. Database management systems, 3rd edition, 2003.
- [4] Y. Tao, D. Papadias, and J. Sun. The TPR*-tree: An optimized spatio-temporal access method for predictive queries. In Proceedings of the 29th international conference on Very large data bases-Volume 29, page 801. VLDB Endowment, 2003.