A Disk-Based Index for Trajectories with In-Memory Compressed Cache

Daniela Campos, Adrián Gómez-Brandón, Gonzalo Navarro

IMFD, Dept. of Computer Science, University of Chile, Chile.
Laboratorio de Bases de Datos, Universidade da Coruña, Spain.
Context

GraCT
Compact Data Structure for representing trajectories

Large amount of data

- Car
- Hiker
- Motorcycle
- Truck
GraCT

- In-memory **spatio-temporal index** for the representation of trajectories.
- Efficient computation of trajectory, spatio-temporal and nearest neighbor queries.
- It uses a **raster** model, losing some precision.
GraCT-Disk

In-Memory Compressed Cached Index

GraCT

$\{t_0, t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8\}$

$\{(0,4), (1,5), (3,5), (3,4), (4,3), (5,3), (6,3), (7,4), (9,6)\}$

B-Tree

B-tree of an object
GraCT-Disk supports the following queries:

- Trajectory
- Time Slice
- Time Interval
- K-Nearest Neighbors
GraCT-Disk supports the following queries:

- **Trajectory**
- Time Slice
- Time Interval
- K-Nearest Neighbors

{ Spatio-temporal and KNN queries }
Trajectory queries

Trajectory of the object from $t_1$ to $t_5$?

GraCT

$\{ (0,4), (1,5), (3,5), (3,4), (4,3), (5,3), (6,3), (7,4), (9,6) \}$

B-Tree

$x = 0.09$
$y = 0.28$

$x = 0.17$
$y = 0.21$

$x = 0.22$
$y = 0.16$

$x = 0.27$
$y = 0.16$

$x = 0.48$
$y = 0.31$

$x = 0.34$
$y = 0.19$

$x = 0.39$
$y = 0.23$
GraCT-Disk supports the following queries:

- Trajectory
- Time Slice
- Time Interval
- K-Nearest Neighbors

Spatio-temporal and KNN queries
We reduce the number of I/O disk operations.
Time Interval: Which objects are within a region at any time instant from the time interval \([t_0, t_5]\)?
Spatio-temporal and KNN queries (II)

Time Interval: Which objects are within a region at any time instant from the time interval \([t_0, t_5]\)?
Spatio-temporal and KNN queries (III)

We take advantage of the MBRs stored in the B-tree

GraCT

B-Tree

{(0,4), (1,5), (3,5), (3,4), (4,3), (5,3), (6,3), (7,4), (9,6)}
Spatio-temporal and KNN queries (IV)

- **Time Slice** is solved as **Time Interval** but taking into account that the interval only lasts one time instant.

- **KNN queries** follow a similar procedure but considering:
  - $D$ is the Euclidean Distance of the $K$-th object obtained by the In-Memory Compressed Cache.
  - The objects with distance lower than $D-1$ cells are part of the solution.
  - We have to **check with the B-trees** those whose distance is in $[D-1, D+1]$. 
Results

- We compared GraCT-Disk with a classical spatio-temporal index: **MVR-Tree**.

- **GraCT-Disk**:
  - Requires *24 times less space* than MVR-Tree.
  - Obtains *competitive times* in Time Slice queries.
  - Outperforms MVR-Tree in Time Interval *(40x)*.
  - Is *slower in K-Nearest Neighbors* by around 32%.

- **GraCT-Disk** can solve *Trajectory* queries, that cannot be said about the MVR-Tree.
A Disk-Based Index for Trajectories with In-Memory Compressed Cache

Daniela Campos, Adrián Gómez-Brandón, Gonzalo Navarro

IMFD, Dept. of Computer Science, University of Chile, Chile.
Laboratorio de Bases de Datos, Universidade da Coruña, Spain.