Ludwig-Maximilians-Universität München
Institut für Informatik
Dr. Matthias Renz
Johannes Niedermayer
Gregor Jossé
Verena Link

München, 16.12.2011

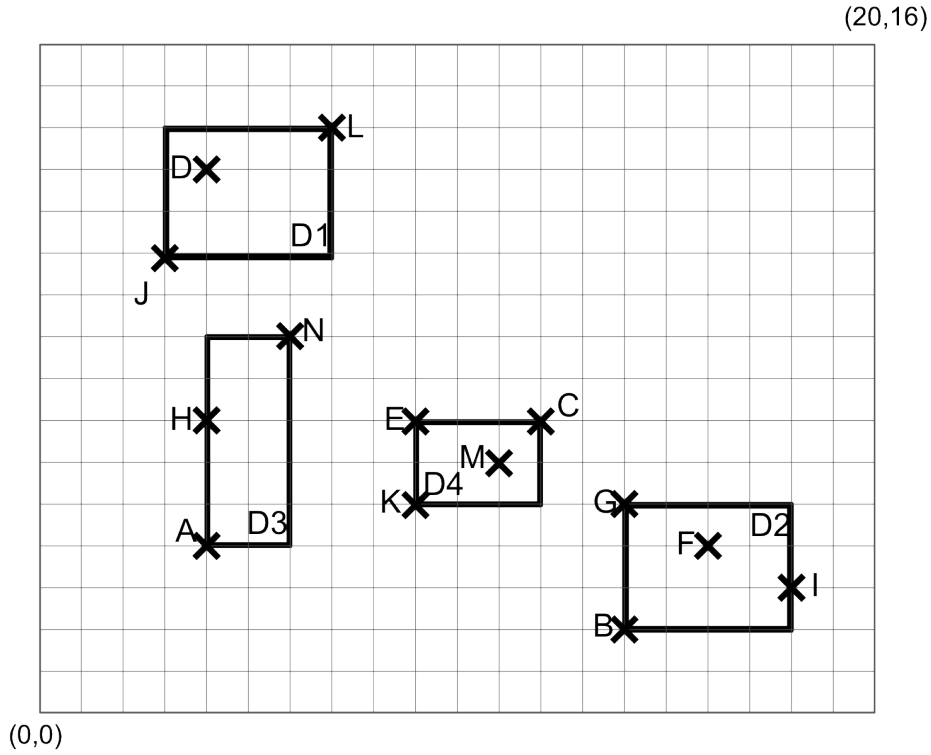
Spatial, Temporal and Multimedia Databases II

Wintersemester 2011/12

Übungsblatt 4: Skyline-Algorithmen, TPL

Besprechung: 19.12.2011

Aufgabe 4-1 Skyline-Queries



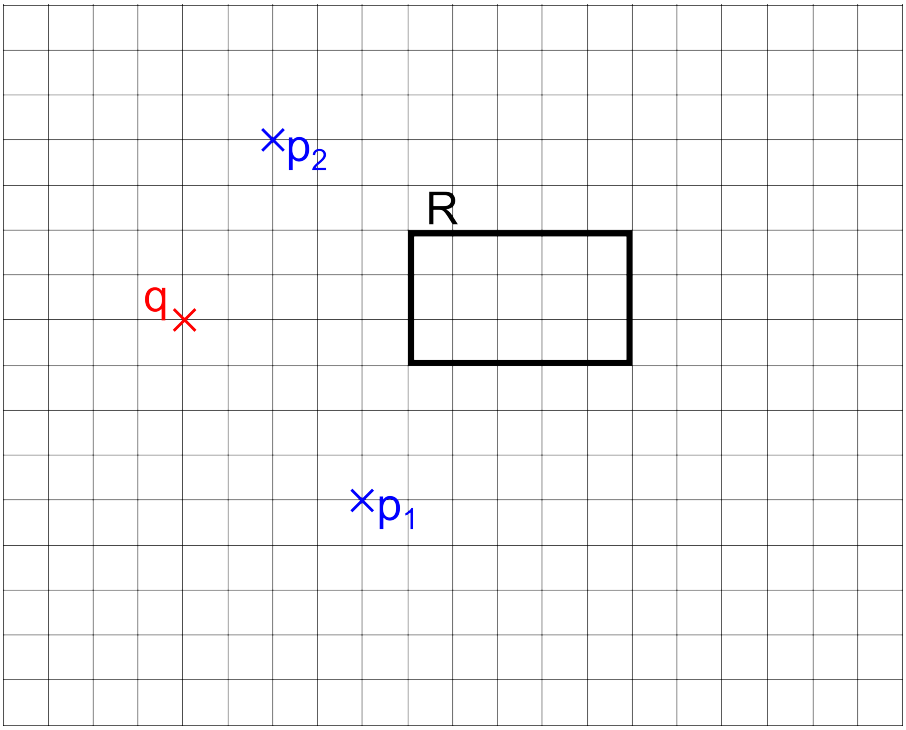
Führen Sie auf der unten dargestellten räumlichen Datenbank, indexiert durch einen R-Baum Skyline-Queries mittels der folgenden Algorithmen durch:

(a) NNS-Algorithmus (Folie 75 / 76): Zeichnen Sie die verschiedenen Suchraumpartitionen sowie die entstehende Skyline ein.

(b) BBS-Algorithmus (Folie 77 / 78): Hierbei seien je zwei Directory-Regionen wie folgt von in einer Metaregion zusammengefasst: D11 umfasse D1 und D3, D12 umfasse D2 und D4. Geben Sie in jedem wesentlichen Schritt die Prioritätsliste sowie die Skyline-Liste an. In der Prioritätsliste sollen Punkte Intermediate Nodes mit gleicher MINDIST vorgezogen werden.

Aufgabe 4-2 TPL: Trimming und Refinement

(a) Gegeben sei eine TPL-Kandidatenmenge $S_{cnd} = \{p_1, p_2\}$ für eine R1NN-Anfrage, ein Anfragepunkt q sowie eine Seitenregion R. Zeichnen Sie in die Grafik für jeden der Punkte aus S_{cnd} die geprunedte Fläche von R bezüglich q ein. Kann die Seitenregion R noch Kandidaten enthalten, oder kann die Seite verworfen werden? Zeichnen Sie auch verwendete Hilfsstrukturen in die Grafik ein.



(b) Gegeben sei die Situation aus der folgenden Grafik für eine R1NN-Query auf q. Es gelte $S_{cnd} = \{p_1, p_2, p_3, p_4\}$, $P_{rfn} = \{p_5\}$ sowie $N_{rfn} = \{R_1, R_2\}$. Führen Sie ein TPL-refinement entsprechend der unten liegenden Algorithmen aus der Originalpublikation[1] durch und geben Sie die Ergebnismenge an.

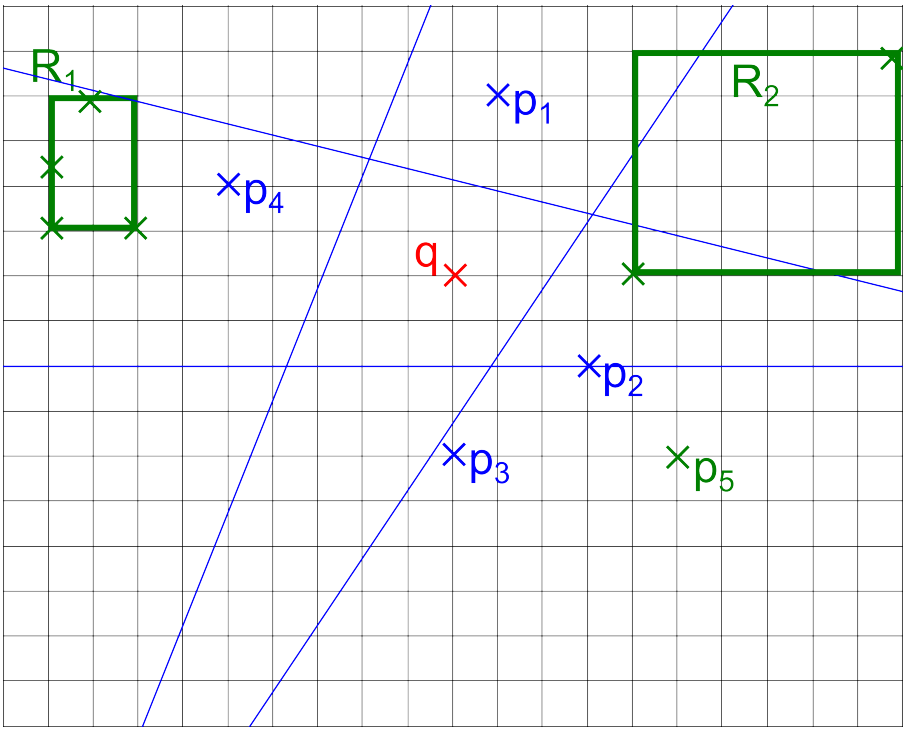
Algorithm TPL-refinement (q, S_{cnd}, P_{rfn}, N_{rfn})
1. for each point p in S_{cnd}
2. for each other point $p' \neq p$ in S_{cnd}
3. if $dist(p,p') < dist(p,q)$
4. $S_{cnd} = S_{cnd} - \{p\}$; goto 1
5. if p is not eliminated initialize $toVisit(p) = \emptyset$
6. repeat
7. **refinement_round**(q, S_{cnd}, P_{rfn}, N_{rfn})
8. if ($S_{cnd} = \emptyset$) return // terminate
9. $P_{rfn} = N_{rfn} = \emptyset$ //initialization of next round
10. Let N be the lowest level node that appears in the largest number of sets $toVisit(p)$, where $p \in S_{cnd}$
11. remove N from all $toVisit(p)$ and access N
12. if N is a leaf node
13. $P_{rfn} = \{p | p \in N\}$ //P_{rfn} contains only the points of N
14. if N is an intermediate node
15. $N_{rfn} = \{N' | N' \in N\}$//$N_{rfn}$ contains the child nodes of N

End TPL-refinement

Algorithm refinement_round(q, S_{cnd}, P_{rfn}, N_{rfn})
1. for each point p in S_{cnd}
2. for each point p' in P_{rfn}
3. if $dist(p,p') < dist(p,q)$
4. $S_{cnd} = S_{cnd} - \{p\}$ //false hit
5. goto 1 //test next candidate
6. for each node MBR N in N_{rfn}
7. if $minmaxdist(p,N) < dist(p,q)$
8. $S_{cnd} = S_{cnd} - \{p\}$ //false hit
9. goto 1 //test next candidate
10. for each node MBR N in N_{rfn}
11. if $mindist(p,N) < dist(p,q)$ add N in $toVisit(p)$
12. if ($toVisit(p) = \emptyset$)
13. $S_{cnd} = S_{cnd} - \{p\}$ and report p // actual result

End refinement_round

[1] Siehe Tao, Papadias, Lian. Reverse kNN Search in Arbitrary Dimensionality. In Proc. of VLDB, 2004.


R1
R2
p1
p2
p3
p4
p5
q