Dimitris Papadias
Donghui Zhang
George Kollios (Eds.)

LNCS 4605

Advances in Spatial and Temporal Databases

10th International Symposium, SSTD 2007
Boston, MA, USA, July 2007
Proceedings

Springer

Dimitris Papadias Donghui Zhang
George Kollios (Eds.)

Advances in Spatial and Temporal Databases

10th International Symposium, SSTD 2007
Boston, MA, USA, July 16-18, 2007
Proceedings

Springer

Volume Editors

Dimitris Papadias
Hong Kong University of Science and Technology
Department of Computer Science and Engineering
Clearwater Bay, Hong Kong, China
E-mail: dimitris@cs.ust.hk

Donghui Zhang
Northeastern University, College of Computer & Information Science
360 Huntington Avenue, #202WVH, Boston, MA 02115, USA
E-mail: donghui@ccs.neu.edu

George Kollios
Boston University, Computer Science Department
111 Cummington Street, Boston, MA 02215, USA
E-mail: gkollios@cs.bu.edu

Library of Congress Control Number: 2007929929

CR Subject Classification (1998): H.2, H.3, H.4, I.2.4

LNCS Sublibrary: SL 3 – Information Systems and Application, incl. Internet/Web and HCI

ISSN 0302-9743
ISBN-10 3-540-73539-9 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-73539-7 Springer Berlin Heidelberg New York

Springer is a part of Springer Science+Business Media

springer.com

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 12088263 06/3180 5 4 3 2 1 0

Lecture Notes in Computer Science 4605

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Preface

SSTD 2007 was the tenth in a series of biannual events that discuss new and exciting research in spatio-temporal data management and related technologies. Previous symposia were successfully held in Santa Barbara (1989), Zurich (1991), Singapore (1993), Portland (1995), Berlin (1997), Hong Kong (1999), Los Angeles (2001), Santorini, Greece (2003) and Angra dos Reis, Brazil (2005). Before 2001, the series was devoted solely to spatial database management, and called SSD. From 2001, the scope was extended in order to accommodate also temporal database management, in part due to the increasing importance of research that considers spatial and temporal aspects jointly.

SSTD 2007 received 76 submissions from 19 countries (based on the affiliation of the first author). A thorough review process led to the acceptance of 26 high-quality papers, geographically distributed as follows: USA 10, Germany 3, Denmark 2, Hong Kong 2, Singapore 2, Brazil 1, Canada 1, France 1, Greece 1, Israel 1, South Korea 1, and Taiwan 1. The papers are classified in the following categories, each corresponding to a conference session: (1) Continuous Monitoring, (2) Indexing and Query Processing, (3) Mining, Aggregation and Interpolation, (4) Semantics and Modeling, (5) Privacy, (6) Uncertainty and Approximation, (7) Streaming Data, (8) Distributed Systems, and (9) Spatial Networks.

The success of SSTD 2007 was the result of team effort. First, we would like to thank the authors for providing the content of the program. We would also like to apologize to the authors of rejected papers, as some good submissions had to be left out. Second, we are grateful to the members of the Program Committee (and the external reviewers) for their thorough and timely reviews. We were impressed by the fact that 95% of the reviews were submitted on time, despite a reviewing process that lasted less than a month. Third, we are grateful to Ellen Grady and the students at Boston University and Northeastern University for their help with organizing and running the conference. Finally, we would like to thank Oracle Spatial, ESRI, and Microsoft Research for their generous support.

We believe that SSTD 2007 continued the successful tradition of the series, providing an interesting program and lively discussions in a pleasant environment.

May 2007

Dimitris Papadias
Donghui Zhang
George Kollios

Organization

SSTD 2007 was organized by Boston University.

Executive Committee

General Chair	George Kollios (Boston University)
Program Chairs	Dimitris Papadias (Hong Kong University of Science and Technology)
	Donghui Zhang (Northeastern University)
Organizing Chair	Ellen Grady (Boston University)
Web Master	Ling Hu (Northeastern University)

Program Committee

Walid Aref	Purdue University (USA)
Lars Arge	University of Aarhus (Denmark)
Spiros Bakiras	John Jay College, CUNY (USA)
Elisa Bertino	Purdue University (USA)
Thomas Brinkhoff	Oldenburg University of Applied Sciences (Germany)
Reynold Cheng	Hong Kong Polytechnic University (Hong Kong)
Max Egenhofer	National Center for Geographic Information and Analysis, Maine (USA)
Amr El Abbadi	University of California, Santa Barbara (USA)
Dimitrios Gunopulos	University of California, Riverside (USA)
Ralf Hartmut Güting	Fernuniversität Hagen (Germany)
Marios Hadjieleftheriou	AT&T Labs Inc. (USA)
Erik Hoel	ESRI (USA)
Christian S. Jensen	Aalborg University (Denmark)
Panos Kalnis	National University of Singapore (Singapore)
George Kollios	Boston University (USA)
Ravi Kothuri	Oracle Spatial (USA)
Nick Koudas	University of Toronto (Canada)
Ki-Joune Li	Pousan National University (South Korea)
David Lomet	Microsoft Research (USA)
Nikos Mamoulis	Hong Kong University (Hong Kong)
Yannis Manolopoulos	Aristotle University (Greece)
Claudia Bauzer Medeiros	University of Campinas (Brazil)
Mohamed Mokbel	University of Minnesota (USA)
Kyriakos Mouratidis	Singapore Management University (Singapore)

External Reviewers

Sponsors

Oracle Spatial (Platinum Sponsor)
ESRI (Silver Sponsor)
Microsoft Research (Silver Sponsor)

Lecture Notes in Computer Science

For information about Vols. 1–4498

please contact your bookseller or Springer

Vol. 4605: D. Papadias, D. Zhang, G. Kollios (Eds.), Advances in Spatial and Temporal Databases. X, 479 pages. 2007.

Vol. 4602: S. Barker, G.-J. Ahn (Eds.), Data and Applications Security XXI. X, 291 pages. 2007.

Vol. 4600: H. Comon-Lundh, C. Kirchner, H. Kirchner, Rewriting, Computation and Proof. XVI, 273 pages. 2007.

Vol. 4597: P. Perner (Ed.), Advances in Data Mining. XI, 353 pages. 2007. (Sublibrary LNAI).

Vol. 4596: L. Arge, C. Cachin, T. Jurdzi, A. Tarlecki (Eds.), Automata, Languages and Programming. XVII, 953 pages. 2007.

Vol. 4595: D. Bošnački, S. Edelkamp (Eds.), Model Checking Software. X, 285 pages. 2007.

Vol. 4592: Z. Kedad, N. Lammari, E. Métais, F. Meziane, Y. Rezgui (Eds.), Natural Language Processing and Information Systems. XIV, 442 pages. 2007.

Vol. 4591: J. Davies, J. Gibbons (Eds.), Integrated Formal Methods. IX, 660 pages. 2007.

Vol. 4590: W. Damm, H. Hermanns (Eds.), Computer Aided Verification. XV, 562 pages. 2007.

Vol. 4589: J. Münch, P. Abrahamsson (Eds.), Product-Focused Software Process Improvement. XII, 414 pages. 2007.

Vol. 4588: T. Harju, J. Karhumäki, A. Lepistö (Eds.), Developments in Language Theory. XI, 423 pages. 2007.

Vol. 4587: R. Cooper, J. Kennedy (Eds.), Data Management. XIII, 259 pages. 2007.

Vol. 4586: J. Pieprzyk, H. Ghodosi, E. Dawson (Eds.), Information Security and Privacy. XIV, 476 pages. 2007.

Vol. 4585: M. Kryszkiewicz, J.F. Peters, H. Rybinski, A. Skowron (Eds.), Rough Sets and Intelligent Systems Paradigms. XIX, 836 pages. 2007. (Sublibrary LNAI).

Vol. 4584: N. Karssemeijer, B. Lelieveldt (Eds.), Information Processing in Medical Imaging. XX, 777 pages. 2007.

Vol. 4583: S.R. Della Rocca (Ed.), Typed Lambda Calculi and Applications. X, 397 pages. 2007.

Vol. 4582: J. Lopez, P. Samarati, J.L. Ferrer (Eds.), Public Key Infrastructure. XI, 375 pages. 2007.

Vol. 4581: A. Petrenko, M. Veanes, J. Tretmans, W. Grieskamp (Eds.), Testing of Software and Communicating Systems. XII, 379 pages. 2007.

Vol. 4580: B. Ma, K. Zhang (Eds.), Combinatorial Pattern Matching. XII, 366 pages. 2007.

Vol. 4578: F. Masulli, S. Mitra, G. Pasi (Eds.), Applications of Fuzzy Sets Theory. XVIII, 693 pages. 2007. (Sublibrary LNAI).

Vol. 4577: N. Sebe, Y. Liu, Y. Zhuang (Eds.), Multimedia Content Analysis and Mining. XIII, 513 pages. 2007.

Vol. 4576: D. Leivant, R. de Queiroz (Eds.), Logic, Language, Information, and Computation. X, 363 pages. 2007.

Vol. 4575: T. Takagi, T. Okamoto, E. Okamoto, T. Okamoto (Eds.), Pairing-Based Cryptography. XI, 408 pages. 2007.

Vol. 4574: J. Derrick, J. Vain (Eds.), Formal Techniques for Networked and Distributed Systems – FORTE 2007. XI, 375 pages. 2007.

Vol. 4573: M. Kauers, M. Kerber, R. Miner, W. Windsteiger (Eds.), Towards Mechanized Mathematical Assistants. XIII, 407 pages. 2007. (Sublibrary LNAI).

Vol. 4572: F. Stajano, C. Meadows, S. Capkun, T. Moore (Eds.), Security and Privacy in Ad-hoc and Sensor Networks. X, 247 pages. 2007.

Vol. 4570: H.G. Okuno, M. Ali (Eds.), New Trends in Applied Artificial Intelligence. XXI, 1194 pages. 2007. (Sublibrary LNAI).

Vol. 4569: A. Butz, B. Fisher, A. Krüger, P. Olivier, S. Owada (Eds.), Smart Graphics. IX, 237 pages. 2007.

Vol. 4566: M.J. Dainoff (Ed.), Ergonomics and Health Aspects of Work with Computers. XVIII, 390 pages. 2007.

Vol. 4565: D.D. Schmorrow, L.M. Reeves (Eds.), Foundations of Augmented Cognition. XIX, 450 pages. 2007. (Sublibrary LNAI).

Vol. 4564: D. Schuler (Ed.), Online Communities and Social Computing. XVII, 520 pages. 2007.

Vol. 4563: R. Shumaker (Ed.), Virtual Reality. XXII, 762 pages. 2007.

Vol. 4561: V.G. Duffy (Ed.), Digital Human Modeling. XXIII, 1068 pages. 2007.

Vol. 4560: N. Aykin (Ed.), Usability and Internationalization, Part II. XVIII, 576 pages. 2007.

Vol. 4559: N. Aykin (Ed.), Usability and Internationalization, Part I. XVIII, 661 pages. 2007.

Vol. 4551: J.A. Jacko (Ed.), Human-Computer Interaction, Part II. XXIII, 1253 pages. 2007.

Vol. 4550: J.A. Jacko (Ed.), Human-Computer Interaction, Part I. XXIII, 1240 pages. 2007.

Vol. 4549: J. Aspnes, C. Scheideler, A. Arora, S. Madden (Eds.), Distributed Computing in Sensor Systems. XIII, 417 pages. 2007.

Vol. 4548: N. Olivetti (Ed.), Automated Reasoning with Analytic Tableaux and Related Methods. X, 245 pages. 2007. (Sublibrary LNAI).

Vol. 4547: C. Carlet, B. Sunar (Eds.), Arithmetic of Finite Fields. XI, 355 pages. 2007.

Vol. 4546: J. Kleijn, A. Yakovlev (Eds.), Petri Nets and Other Models of Concurrency – ICATPN 2007. XI, 515 pages. 2007.

Vol. 4545: H. Anai, K. Horimoto, T. Kutsia (Eds.), Algebraic Biology. XIII, 379 pages. 2007.

Vol. 4544: S. Cohen-Boulakia, V. Tannen (Eds.), Data Integration in the Life Sciences. XI, 282 pages. 2007. (Sublibrary LNBI).

Vol. 4543: A.K. Bandara, M. Burgess (Eds.), Inter-Domain Management. XII, 237 pages. 2007.

Vol. 4542: P. Sawyer, B. Paech, P. Heymans (Eds.), Requirements Engineering: Foundation for Software Quality. IX, 384 pages. 2007.

Vol. 4541: T. Okadome, T. Yamazaki, M. Makhtari (Eds.), Pervasive Computing for Quality of Life Enhancement. IX, 248 pages. 2007.

Vol. 4539: N.H. Bshouty, C. Gentile (Eds.), Learning Theory. XII, 634 pages. 2007. (Sublibrary LNAI).

Vol. 4538: F. Escolano, M. Vento (Eds.), Graph-Based Representations in Pattern Recognition. XII, 416 pages. 2007.

Vol. 4537: K.C.-C. Chang, W. Wang, L. Chen, C.A. Ellis, C.-H. Hsu, A.C. Tsoi, H. Wang (Eds.), Advances in Web and Network Technologies, and Information Management. XXIII, 707 pages. 2007.

Vol. 4536: G. Concas, E. Damiani, M. Scotto, G. Succi (Eds.), Agile Processes in Software Engineering and Extreme Programming. XV, 276 pages. 2007.

Vol. 4534: I. Tomkos, F. Neri, J. Solé Pareta, X. Masip Bruin, S. Sánchez Lopez (Eds.), Optical Network Design and Modeling. XI, 460 pages. 2007.

Vol. 4533: F. Baader (Ed.), Term Rewriting and Applications. XII, 419 pages. 2007.

Vol. 4531: J. Indulska, K. Raymond (Eds.), Distributed Applications and Interoperable Systems. XI, 337 pages. 2007.

Vol. 4530: D.H. Akehurst, R. Vogel, R.F. Paige (Eds.), Model Driven Architecture- Foundations and Applications. X, 219 pages. 2007.

Vol. 4529: P. Melin, O. Castillo, L.T. Aguilar, J. Kacprzyk, W. Pedrycz (Eds.), Foundations of Fuzzy Logic and Soft Computing. XIX, 830 pages. 2007. (Sublibrary LNAI).

Vol. 4528: J. Mira, J.R. Álvarez (Eds.), Nature Inspired Problem-Solving Methods in Knowledge Engineering, Part II. XXII, 650 pages. 2007.

Vol. 4527: J. Mira, J.R. Álvarez (Eds.), Bio-inspired Modeling of Cognitive Tasks, Part I. XXII, 630 pages. 2007.

Vol. 4526: M. Malek, M. Reitenspieß, A. van Moorsel (Eds.), Service Availability. X, 155 pages. 2007.

Vol. 4525: C. Demetrescu (Ed.), Experimental Algorithms. XIII, 448 pages. 2007.

Vol. 4524: M. Marchiori, J.Z. Pan, C.d.S. Marie (Eds.), Web Reasoning and Rule Systems. XI, 382 pages. 2007.

Vol. 4523: Y.-H. Lee, H.-N. Kim, J. Kim, Y. Park, L.T. Yang, S.W. Kim (Eds.), Embedded Software and Systems. XIX, 829 pages. 2007.

Vol. 4522: B.K. Ersbøll, K.S. Pedersen (Eds.), Image Analysis. XVIII, 989 pages. 2007.

Vol. 4521: J. Katz, M. Yung (Eds.), Applied Cryptography and Network Security. XIII, 498 pages. 2007.

Vol. 4519: E. Franconi, M. Kifer, W. May (Eds.), The Semantic Web: Research and Applications. XVIII, 830 pages. 2007.

Vol. 4517: F. Boavida, E. Monteiro, S. Mascolo, Y. Koucheryavy (Eds.), Wired/Wireless Internet Communications. XIV, 382 pages. 2007.

Vol. 4516: L. Mason, T. Drwiega, J. Yan (Eds.), Managing Traffic Performance in Converged Networks. XXIII, 1191 pages. 2007.

Vol. 4515: M. Naor (Ed.), Advances in Cryptology - EUROCRYPT 2007. XIII, 591 pages. 2007.

Vol. 4514: S.N. Artemov, A. Nerode (Eds.), Logical Foundations of Computer Science. XI, 513 pages. 2007.

Vol. 4513: M. Fischetti, D.P. Williamson (Eds.), Integer Programming and Combinatorial Optimization. IX, 500 pages. 2007.

Vol. 4511: C. Conati, K. McCoy, G. Paliouras (Eds.), User Modeling 2007. XVI, 487 pages. 2007. (Sublibrary LNAI).

Vol. 4510: P. Van Hentenryck, L. Wolsey (Eds.), Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems. X, 391 pages. 2007.

Vol. 4509: Z. Kobti, D. Wu (Eds.), Advances in Artificial Intelligence. XII, 552 pages. 2007. (Sublibrary LNAI).

Vol. 4508: M.-Y. Kao, X.-Y. Li (Eds.), Algorithmic Aspects in Information and Management. VIII, 428 pages. 2007.

Vol. 4507: F. Sandoval, A. Prieto, J. Cabestany, M. Graña (Eds.), Computational and Ambient Intelligence. XXVI, 1167 pages. 2007.

Vol. 4506: D. Zeng, I. Gotham, K. Komatsu, C. Lynch, M. Thurmond, D. Madigan, B. Lober, J. Kvach, H. Chen (Eds.), Intelligence and Security Informatics: Biosurveillance. XI, 234 pages. 2007.

Vol. 4505: G. Dong, X. Lin, W. Wang, Y. Yang, J.X. Yu (Eds.), Advances in Data and Web Management. XXII, 896 pages. 2007.

Vol. 4504: J. Huang, R. Kowalczyk, Z. Maamar, D. Martin, I. Müller, S. Stoutenburg, K.P. Sycara (Eds.), Service-Oriented Computing: Agents, Semantics, and Engineering. X, 175 pages. 2007.

Vol. 4501: J. Marques-Silva, K.A. Sakallah (Eds.), Theory and Applications of Satisfiability Testing – SAT 2007. XI, 384 pages. 2007.

Vol. 4500: N. Streitz, A. Kameas, I. Mavrommati (Eds.), The Disappearing Computer. XVIII, 304 pages. 2007.

Vol. 4499: Y.Q. Shi (Ed.), Transactions on Data Hiding and Multimedia Security II. IX, 117 pages. 2007.

Table of Contents

Continuous Monitoring of Exclusive Closest Pairs*

Leong Hou U[1], Nikos Mamoulis[1], and Man Lung Yiu[2]

[1] Department of Computer Science, University of Hong Kong, Pokfulam Road, Hong Kong
{hleongu,nikos}@cs.hku.hk
[2] Department of Computer Science, Aalborg University, DK-9220 Aalborg, Denmark
mly@cs.aau.dk

Abstract. Given two datasets A and B, their exclusive closest pairs (ECP) join is a one-to-one assignment of objects from the two datasets, such that (i) the closest pair (a, b) in $A \times B$ is in the result and (ii) the remaining pairs are determined by removing objects a, b from A, B respectively, and recursively searching for the next closest pair. An application of exclusive closest pairs is the computation of (car, parking slot) assignments. In this paper, we propose algorithms for the computation and continuous monitoring of ECP joins in memory, given a stream of events that indicate dynamic assignment requests and releases of pairs. Experimental results on a system prototype demonstrate the efficiency of our solutions in practice.

1 Introduction

Due to the increasing popularity of location-based services, continuous monitoring of spatial queries emerges as an important research topic. Existing work [18, 11, 21, 13, 16, 17] focuses on *range* or k *nearest neighbor* (kNN) queries on moving objects. These problems can also be viewed as continuous joins between queries and data objects, according to their spatial relationship. However, there has not been much research done related to the continuous monitoring of spatial join results. Several variants of spatial join queries exist, such as the intersection join [2], the distance (or similarity) join [14], the all k nearest neighbors join [24], and the k (inclusive) closest pairs query (kICP) [9, 4].

In this paper, we study an interesting type of spatial joins that has received little attention in the past. We call this operation the k *exclusive closest pairs* join (kECP). kECP produces k one-to-one assignments of objects between two datasets A and B, such that (i) the closest pair (a, b) in $A \times B$ belongs to the result and (ii) the remaining pairs are determined by removing objects a, b from A, B respectively, and recursively searching for the next closest pair. Thus, each object appears only once in the result.

A real-life application of a kECP query is the car-parking assignment problem. Consider a set A of car drivers that request for a parking slot and another set B of available slots. The well-known assignment problem [19] searches for the 1-to-1 assignment of cars to parking spaces, such that the sum of travel distances is minimized. However, in a world of selfish users, it is more reasonable to assign each car $c \in A$ to the parking

* Supported by grant HKU 7160/05E from Hong Kong RGC.

D. Papadias, D. Zhang, and G. Kollios (Eds.): SSTD 2007, LNCS 4605, pp. 1–19, 2007.

space $p \in B$ that may not be taken by another driver c', which happens to be closer to p than c is. Therefore, our formulation of the kECP query (assuming that k is the minimum of cardinalities $|A|$ and $|B|$) searches for a practical solution to the problem.

We propose a technique for computing the ECP pairs efficiently given a set of cars and a set of parking slots. In addition, we extend it to monitor the ECP results, in a dynamic environment, where parking requests from cars and availability events from parking slots arrive from a data stream. Due to such events, ECP assignments must be deleted (i.e., when a car un-parks), new assignments must be added (i.e., when a new car requests parking), and current assignments may have to be changed. For instance, assume that pair (c, p) is in the current assignment and a new parking slot p' becomes available which is closer to c than p is. In this case, c must be re-assigned to p' and p should become available for other cars. This change may trigger a "chaining" effect which could alter the whole assignment. Our method processes incoming events in an appropriate order, such that the correct ECP results are maintained correctly and efficiently.

We assume that a centralized server monitors the locations of objects. When an object moves to another location, it informs the server about its new location. Since the frequent updates render disk-based management techniques inefficient, our solution is based on a memory grid-based indexing approach [16, 18].

Our contributions can be summarized as follows:

- We identify ECP as a new type of spatial join that finds application in real-life dynamic allocation problems (e.g., car/parking assignment).
- We show that the kECP for $k = \min\{|A|, |B|\}$ is equivalent to a special case of the stable marriage problem [7], where assignment preferences are derived from the distance function. Based on this observation, we adapt the Gale-Shapley algorithm [6] to solve kECP queries by computing only a small fraction of the distances dynamically and on-demand.
- We define a dynamic version of ECP for moving objects and streaming events that indicate (i) availability of slots and (ii) demand for new kECP pairs. We propose an appropriate extension of our static kECP query evaluation algorithm that solves this continuous kECP query.
- We conduct a set of experiments to verify the efficiency of the proposed methods for a wide range of problem parameters.

The rest of the paper is organized as follows. Section 2 surveys related work on closest pair queries in spatial data, continuous monitoring problems, and the stable marriage problem. Section 3 formally defines ECP and presents our solution to it for a static input. Section 4 presents an update framework for ECP calculation with two optimizations to improve its performance. Our solutions are evaluated in Section 5. Finally, Section 6 concludes the paper, giving directions for future work.

2 Background and Related Work

2.1 Closest Pairs Queries in Spatial Databases

Computation of closest pairs queries have been studied for several decades. Main-memory algorithms, such as the Neighbor Heuristic [1] and Fast Pair [3, 5], focus on

1CP problems. Fast Pair was shown to have the best overall performance. However, this method is not directly applicable to: (i) kCP queries for arbitrary values of k, and (ii) other variants of CP queries.

Some previous work [4, 9, 22] employ spatial indexes to solve kICP queries in secondary memory. [4, 9] assume that the datasets are indexed by R-trees [8]. On the other hand, Yang et al. [22] extended the R-tree to a b-Rdnn tree, by augmenting each non-leaf entry with the maximum nearest neighbor distance (with respect to the other dataset) of points in its subtree. During query evaluation, such distances are utilized for reducing the search space. [22] showed that their approach outperforms previous R-tree based methods. Since these methods operate on indexed data they may not be applied in a dynamic environment. A high rate of streaming events imposes a high burden to the update of the indexes, which in combination with the expensive refreshing of the query results, renders the overall approach inefficient or impossible. In addition, although an kICP algorithm can be tuned to process the kECP query (i.e., by remembering assigned points and avoiding their re-assignment), such an approach would require a large amount of memory (for $k=\min\{|A|, |B|\}$, as large as the size of a dataset).

2.2 Continuous Monitoring of Spatial Queries

Various spatial applications, like the car-parking problem of the Introduction, handle large amounts of information at fast arrival rate. Several extensions of R-trees have been developed for supporting frequent updates of spatial data. Lee et al. [15] proposed the FUR-tree (Frequent Update R-tree), which uses localized bottom-up update strategies into the traditional R-tree. Recently, Xiong et al. [20] developed the RUM-tree (R-tree with Update Memo), which was shown to have better update performance than FUR-tree. [12] applied an event-driven approach to maintain query results for kNN and spatial join queries, with the assumption that moving objects can be modeled by linear motion functions.

Continuous monitoring of *multiple* spatial queries (e.g., range [16, 17] and kNN [21, 23, 18]) adopt the *shared execution paradigm* to reduce the processing cost. Instead of monitoring the results for different queries separately, the problem is viewed as a large spatial join between the query objects and data objects. As illustrated in Figure 1, grid cells (of cell length δ) are employed for indexing the objects. In practice,

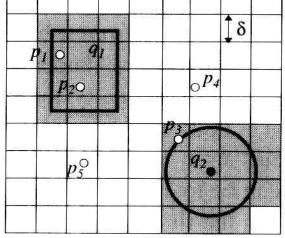

Fig. 1. Monitoring spatial queries

memory grid cells [23, 18] are used (instead of disk-based structures) in order to handle very high update rate. q_1 corresponds to a range query (shown in bold rectangle) and its *influence region* consists of the (gray) cells that intersect with q_1. Since data object updates outside the influence region cannot affect the query result, the processing cost is significantly reduced. As another example, q_2 represents a NN query (shown in bold circle). Its difference from q_1 is that its influence region is a circular region centered at q_2 with dynamic radius equal its NN distance. For example, when the NN of q_2 moves closer to (further from) q_2, then the influence region of q_2 shrinks (grows).

Observe that continuous monitoring of range/kNN queries is different from that of kECP queries. For range and kNN queries, only query results near a change triggered by a streaming event (i.e., appearance, disappearing, or movement of an object) need to be updated (i.e., only for queries whose influence region intersects the location of the change). On the other hand, as we will discuss in Section 4, a streaming event can generate a sequence of changes in the kECP result. Thus, the idea of influence regions is not appropriate for kECP monitoring, which calls for novel techniques.

2.3 The Stable Marriage Problem

The kECP join is closely related to the classic stable marriage problem [6, 7]. Given two set of objects A and B, M is said to be a *matching* between A and B if (i) M is a set of $\min\{|A|, |B|\}$ pairs of objects (a, b) where $a \in A$, $b \in B$, and (ii) each object $a \in A$ ($b \in B$) appears in at most one pair in M. A matching M is *stable* if there are no pairs (a, b) and (a', b') in M such that a prefers b' to b and b' prefers a to a'. Given the preference lists of all objects $a \in A$ and $b \in B$, the stable marriage problem seeks for a stable matching. In our context, the preference list of an object a is implicitly defined by the total order defined by the Euclidean distance; if a is closer to b than to b', then a prefers b to b'.

[7] is a nice reference text that introduces the stable marriage problem and presents solutions to it, for special cases of the input. For the generic problem, Gale and Shapley [6] proved that, if $|A| = |B|$, it is always possible to find a solution and provided an algorithm for this. For the ease of discussion, we call the objects in A and B as senders and receivers, respectively. In the first round, each sender (in A) calls its most preferred receiver (in B). If a receiver hears from at least one sender, then the receiver matches with the best sender (according to the receiver's preference) and the corresponding sender is removed from A. The above procedure is applied iteratively in subsequent rounds, but with an additional rule: if a receiver has been assigned a sender a_{old} (in previous rounds) and now it hears from a better sender a_{new} (in the current round), then the receiver matches with the new sender and the remaining set of senders becomes $A := \{a_{old}\} \cup A - \{a_{new}\}$. Eventually, the stable matching between A and B is obtained after all objects in A or B have been removed.

For example, Table 1 illustrates a set A of three jobs and a set B of three applicants, such that the applicants (jobs) can be totally ordered based on their qualification (preference) for the job (applicant). In the first round of the Gale-Shapley algorithm, both jobs a_1 and a_2 call the applicant b_1, who prefers a_1 to a_2. Thus, b_1 matches with a_1 and a_1 is removed from A. Also, a_3 calls b_2, b_2 matches with a_3 and a_3 is removed from A. In the second round, a_2 calls b_2. Since b_2 prefers the new job a_2 to its old

Table 1. Example of stable marriage

Job	Preference	Applicant	Preference
a_1	$b_1 \succ b_3 \succ b_2$	b_1	$a_1 \succ a_3 \succ a_2$
a_2	$b_1 \succ b_2 \succ b_3$	b_2	$a_2 \succ a_3 \succ a_1$
a_3	$b_2 \succ b_3 \succ b_1$	b_3	$a_2 \succ a_1 \succ a_3$

job a_3, b_2 now matches with a_2 instead and the job a_3 is added back to A. In the third round, a_3 calls b_3 and b_3 matches with a_3. Thus, the stable matching contains the pairs (a_1, b_1), (a_2, b_2), (a_3, b_3). Note that at least one pair is finalized at each round, thus the worst-case time complexity of the algorithm is $O(|A| \times |B|)$.

The stable marriage algorithm is asymmetric; if the roles of A and B are reversed, a different solution may be found. Furthermore, it has been shown that it is sender-optimal (i.e., A-optimal if A is the sender dataset); its execution will derive the optimal pair in B for any $a \in A$, for any order of examined objects from A. Thus, there is a unique solution when taking A as the sender input and another unique solution when taking B as the sender. We now prove that if the preference list is derived by a symmetric weight function w (e.g., Euclidean distance), such that $w(a, b) = w(b, a), \forall a \in A, b \in B$, then these two solutions are identical.

Theorem 1. *If preferences are defined by a weight function w, such that a prefers b to b' if and only if $w(a, b) < w(a, b')$ and b prefers a to a' if and only if $w(b, a) < w(b, a')$, and $w(a, b) = w(b, a)$, for any $a, a' \in A$, $b, b' \in B$ then the optimal stable marriage result is unique independently on whether A or B is the sender set.*

Proof. Without loss of generality, assume that $|A| = |B| = n$. Let M_A=$\{\{(a_{(1)}, b_{(1)})\}, \{(a_{(2)}, b_{(2)})\}, \ldots, \{(a_{(n)}, b_{(n)})\}\}$ be the A-optimal matching, such that $(a_{(i)}, b_{(i)})$ models the pair which is finalized at the i-th loop of the Gale-Shapley algorithm.[1] Let the B-optimal matching, generated by the Gale-Shapley algorithm, be $M_B = \{\{(b'_{(1)}, a'_{(1)})\}, \{(b'_{(2)}, a'_{(2)})\}, \ldots, \{(b'_{(n)}, a'_{(n)})\}\}$. We will first prove that $a_{(1)} = a'_{(1)}$ and $b_{(1)} = b'_{(1)}$, i.e., the first assignments output by the two runs of the algorithm are identical. Since $(a_{(1)}, b_{(1)})$ is the first finalized pair of the A-sender run, $w(a_{(1)}, b_{(1)})$ should be the smallest $w(a, b)$, for any $a \in A$, $b \in B$. Similarly, $w(b'_{(1)}, a'_{(1)})$ should be the smallest $w(b, a)$, for any $a \in A$, $b \in B$. Since $w(a, b) = w(b, a)$, it must be $a_{(1)} = a'_{(1)}$ and $b_{(1)} = b'_{(1)}$. By induction, we can prove that $a_{(i)} = a'_{(i)}$ and $b_{(i)} = b'_{(i)}$, for $1 \leq i \leq n$, since by removing pairs $\{(a_{(1)}, b_{(1)}), (a_{(2)}, b_{(2)}), \ldots, (a_{(i)}, b_{(i)})\}$ from the problem we showed that the first pair $(a_{(i+1)}, b_{(i+1)})$ in the resulting subproblem is identical for both A-sender and B-sender runs. □

A subtle issue to note is that the uniqueness argument for the A-sender (or B-sender) Gale-Shapley's output and Theorem 1 holds only for cases where the preference lists are *unique*, strictly total orders. Non-unique orders can be derived from weight functions

[1] Without loss of generality, we assume that only one pair is finalized at each loop. If there are multiple such pairs we could modify the algorithm to output only the one with the smallest $w(a, b)$, without affecting the correctness of the result.

w, for which there exist pairs (a, b) and (a', b'), such that $w(a, b) = w(a', b')$ and $(a = a' \wedge b \neq b')$ or $(a \neq a' \wedge b = b')$. In such cases, e.g., $a = a' \wedge b \neq b'$, object a has the same preference to b and b', therefore the stable marriage result may not be unique; there could be a stable solution that includes (a, b) and another that includes (a, b').

3 The Static *k*ECP Query

In this section, we define and solve the *static* case of the kECP query, where the kECP result is requested for two sets of static points. For completeness, we also provide the definition of the k *inclusive* closest pairs (kICP) query.

Definition 1. *Given two set of points A, B and a $k < |A \times B|$, the k* inclusive closest pairs *$kICP(A, B)$ is defined as the set $S \subset A \times B$, such that $|S| = k$ and $\forall (a, b) \in S, (a', b') \in (A \times B) - S, d(a, b) \leq d(a', b')$.*

Definition 2. *Given two set of points A and B, the k* exclusive closest pairs *$kECP(A, B)$ is recursively defined as:*

$kECP(A, B) = kICP(A, B)$, for $k = 1$, and
$kECP(A, B) = 1ECP(A, B) \cup (k-1)ECP(A - \{a\}, B - \{b\})$, otherwise.

Note that the maximum possible value for k is $\min\{|A|, |B|\}$ in kECP and $k \leq |A| \cdot |B|$ in kICP. It is easy to prove that kECP, for $k = \min\{|A|, |B|\}$ is a special case of the stable marriage problem, where the preference order is derived by the weight function $w(a, b) = d(a, b)$ (d denotes Euclidean distance). Therefore the Gale-Shapley stable marriage algorithm (SMA) can be applied to solve kECP queries; the preference list of a point $a \in A$ is constructed by placing points in B in ascending order of their distances to a. The preference lists of points $b \in B$ are generated symmetrically. After running SMA on the preference lists, the obtained results correspond to the results of ECP. Since $d(a, b) \equiv d(b, a)$, and assuming that the distances between a point $a \in A$ and the points in B are distinct (and vice versa)[2], SMA will derive the unique ECP result according to Theorem 1, no matter whether we take A of B as the sender set.

Nevertheless, the direct application of SMA requires the computation of a large number of distances and large space to store them (for $|A| \cdot |B|$ distances), thus it does not scale well for large problems. We conjecture that the spatial properties of the query, in combination with appropriate indexes can be utilized to accelerate SMA. For example, we need not compute the distance of a point $a \in A$ to *all* in B before running SMA; instead, we can applying spatial ranking techniques [10, 18] to generate the preference list of a incrementally and on-demand.

We adopt CPM; the grid-based technique of [18] for indexing data points in our problem, due to its good performance in environments with frequent updates. CPM is the state-of-the-art grid-based index for monitoring NN queries. Each query point is associated with a heap such that the objects and grid cells are visited in ascending order of their distances from the query point. In this way, query results can be computed fast and unnecessary accesses to other points are avoided. In particular, [18] propose a

[2] This is a realistic assumption since distances are real numbers and they are unlikely to coincide.