Theoretical Computer Science 526 (2014) 58–74

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Dynamic 3-sided planar range queries with expected doubly-logarithmic time ☆

Gerth Stølting Brodal [a], Alexis C. Kaporis [b], Apostolos N. Papadopoulos [c], Spyros Sioutas [d,*], Konstantinos Tsakalidis [e], Kostas Tsichlas [c]

[a] MADALGO, Department of Computer Science, Aarhus University, Denmark
[b] Computer Engineering and Informatics Department, University of Patras, Greece
[c] Department of Informatics, Aristotle University of Thessaloniki, Greece
[d] Department of Informatics, Ionian University, Corfu, Greece
[e] Computer Science and Engineering Department, CUHK, Hong Kong

ARTICLE INFO

Article history:
Received 12 January 2012
Received in revised form 24 November 2013
Accepted 14 January 2014
Communicated by S. Sen

Keywords:
3-Sided range reporting
Doubly logarithmic
Average case analysis
Dynamic data structures
Computational geometry

ABSTRACT

The Priority Search Tree is the classic solution for the problem of dynamic 2-dimensional searching for the orthogonal query range of the form $[a, b] \times (-\infty, c]$ (3-sided rectangle). It supports all operations in logarithmic worst case complexity in both main and external memory. In this work we show that the update and query complexities can be improved to expected doubly-logarithmic, when the input coordinates are being continuously drawn from specific probability distributions. We present three pairs of linear space solutions for the problem, i.e. a RAM and a corresponding I/O model variant:

(1) First, we improve the update complexity to doubly-logarithmic expected with high probability, under the most general assumption that both the x- and y-coordinates of the input points are continuously being drawn from a distribution whose density function is unknown but fixed.

(2) Next, we improve both the query complexity to doubly-logarithmic expected with high probability and the update complexity to doubly-logarithmic amortized expected, by assuming that only the x-coordinates are being drawn from a class of *smooth* distributions, and that the deleted points are selected uniformly at random among the currently stored points. In fact, the y-coordinates are allowed to be arbitrarily distributed.

(3) Finally, we improve both the query and the update complexity to doubly-logarithmic expected with high probability by moreover assuming the y-coordinates to be continuously drawn from a more restricted class of realistic distributions.

All data structures are deterministic and their complexity's expectation is with respect to the assumed distributions. They comprise combinations of known data structures and of two new data structures introduced here, namely the *Weight Balanced Exponential Tree* and the *External Modified Priority Search Tree*.

☆ This work is based on a combination of two conference papers that appeared in the Proceedings of the 21st International Symposium on Algorithms and Computation (ISAAC), 2010, pp. 1–12 (by all authors except the third one) and the Proceedings of the 13th International Conference on Database Theory (ICDT), 2010, pp. 34–43 (by all authors except the first one).

* Corresponding author.
E-mail addresses: gerth@madalgo.au.dk (G. Stølting Brodal), kaporis@ceid.upatras.gr (A.C. Kaporis), papadopo@csd.auth.gr (A.N. Papadopoulos), sioutas@ionio.gr (S. Sioutas), tsakalid@cse.cuhk.edu.hk (K. Tsakalidis), tsichlas@csd.auth.gr (K. Tsichlas).

0304-3975/$ – see front matter
http://dx.doi.org/10.1016/j.tcs.2014.01.014

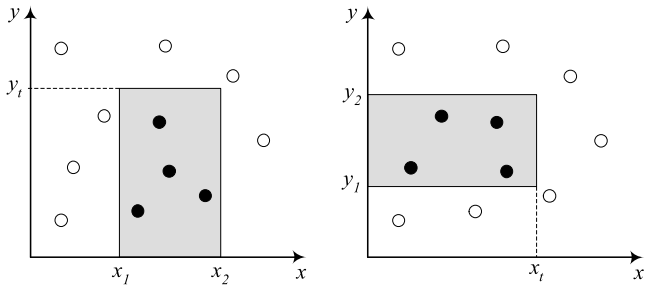

Fig. 1. Examples of 3-sided queries.

1. Introduction

Recently, a significant effort has been made towards developing worst-case efficient data structures for range searching in two dimensions [37]. In their pioneering work, Kanellakis et al. [20,21] illustrated that the problem of indexing in new data models, such as *constraint*, *temporal* and *object* models, can be reduced to special cases of two-dimensional indexing. In particular, they identified that *3-sided range searching* is of major importance.

The *3-sided range reporting query* in the 2-dimensional space is defined by an orthogonal region of the form $R = [a, b] \times (-\infty, c]$, i.e., a rectangular region with one side "open", and returns all points contained in R. Fig. 1 depicts examples of possible 3-sided queries, defined by the shaded regions. Black dots represent the points comprising the result. In many applications, only positive coordinates are used and therefore, the region defining the 3-sided query always touches one of the two axes, according to application semantics.

We note that there is a plethora of applications that can benefit by efficient indexing schemes for 3-sided queries. Consider for example a spatio-temporal database that monitors the locations of moving objects to enable *location-based services*. The query asking for the set of objects that reached a specific point of interest during a time interval may be expressed as a 3-sided query where the time interval is represented by the values t_1 and t_2, whereas the other dimension limits the distance from the point of interest. In the sequel we provide a more detailed application example based on a *sensor network*.

Consider a time evolving database storing measurements collected from a sensor network. Assume that each measurement is represented by a multi-attribute tuple of the form $\langle id, a_1, a_2, \ldots, a_d, time\rangle$, where *id* is the sensor identifier that produced the measurement, d is the total number of attributes, each a_i, $1 \leqslant i \leqslant d$, denotes the value of the specific attribute and finally *time* records the timestamp of the specific measurement. These values may relate to measurements regarding temperature, pressure, humidity, and so on. Therefore, each tuple may be seen as a *point* in $\mathbb{R}^d$ space. Let $F(p)$ be a real-valued *scoring function* that scores each point p based on the values of (a subset of) the attributes. Usually, the scoring function F is monotone and, without loss of generality, we assume that the lower the score the "better" the measurement (the other case is symmetric). Popular scoring functions are the aggregates SUM, MIN, AVG or other more complex combinations of the attributes. Now consider the query: "search for all measurements taken between the time instances t_1 and t_2 such that the score is below s". Notice that this is essentially a 2-dimensional 3-sided query with *time* as the x-axis and *score* as the y-axis, which may be expressed in SQL as follows:

```
SELECT id, score, time
FROM SENSOR_DATA
WHERE time >= t1 AND time <= t2 AND score <= s;
```

To support the previous application we need to provide efficient insertions and queries. Consider now the case where the measurements of interest belong to a *sliding window* [8] such that new measurements enter the window and old ones are evicted. This is a typical scenario of a stream-based application, where usually we need to process the data as they arrive since we can only perform online processing. In this case, deletions must be also handled efficiently, meaning that the data structure that is used to store the points contained in the sliding window must be a fully dynamic data structure. These techniques are ubiquitous in settings that require continuous query processing (e.g., [8,31]). Search efficiency directly impacts query response time as well as the general system performance, whereas update efficiency guarantees that incoming data are stored and organized quickly, thus, preventing delays due to excessive resource consumption.

Another important issue in such data intensive applications is memory consumption. The best practice is to keep data in main memory if this is possible. However, external memory solutions must also be available to cope with large data volumes and enable the operation of large databases. For this reason, in this work we study both cases, offering efficient solutions both in the RAM and I/O models of computation.

Related work The usefulness of 3-sided queries has been underlined many times in the literature [11,21]. Apart from the significance of this query in multi-dimensional data intensive applications [12,21], 3-sided queries appear in probabilistic