Chapter 3

A Linked Neighboring Leaves N–Tree to Support Distance Range Search

Faïza Najjar
National School of Comp. Science, Tunisia

Hassenet Slimani
Faculty of Sciences, Tunisia

ABSTRACT

Mobile query processing is, actually, a very active research field. Range and nearest neighbor queries are common types of queries in spatial databases and location based services (LBS). In this paper, we focus on finding nearest neighbors of a query point within a certain distance range. An example of query, frequently met in LBS, is “Find all the nearest gas stations within 2 miles neighborhood of his/her current location”. We propose two approaches for answering such queries. Both are based on a recent indexing technique called N-tree. The first one is a branch and bound approach, whereas the second, called ‘neighborhoods scanning’, is based on a variant of N-tree, Leaves-Linked N-tree (LLN-tree). LLN-tree is an index tree structure that avoids visiting multiple paths during range search. Both techniques are presented, illustrated and evaluated. Experiments show that the latter approach outperforms the former in response time and disk access as well.

INTRODUCTION

Given a set of objects \( S = \{s_1, s_2, \ldots s_n\} \), having certain properties/attributes, if the properties’ domain supports comparison operations, the range search problem is to find all objects in \( S \) that satisfy any range query.

It’s obvious that such a type of a query allows investigating correlation between attributes of the considered system and events occurring within it. Hence, it’s met in a wide range of research areas such as computational geometry (de Berg, 2000), database applications, geographical information systems and computer graphics (Hjaltason, 2003; Samet, 2007).
Yet, the problem of multidimensional/multivariate/orthogonal range search has gained a lot of investigation from researchers in centralized databases (Papadopolous, 2005; Ind, Iss; Alstrup, 2000) and distributed databases as well (Andrzejak, 2002; Li, 2003; Ratnasamy, 2003; Marzolla, 2006; Bharambe, 2004). An example of an orthogonal search query in a centralized database context is: “Select ID, NAME from STUDENTS where HOURS BETWEEN 75 AND 90 AND GPA BETWEEN 2.5 AND 3.0”. For a distributed context like a sensor network, an example of query is: “List all events that have temperatures between 50 Fa and 60 Fa, and light levels between 10 and 20 in a monitoring system for the growth of marine micro-organisms” (Li, 2003).

Proposed solutions include balanced binary trees for the one dimensional form, kd-tree for higher dimensional form, its embedding for specific systems, multi attribute trees, doubly chained trees, distributed Hash Table... etc.

The temporal aspect of range queries is dealt with in (Shi, 2005). In this work, the focus was on designing efficient indexing structures allowing fast retrieval of objects whose attributes fall within a set of ranges during a given period of time.

In spatial databases, a range may reduce to a distance range/interval as in the query ‘Find data objects situated within the range/ interval [5k,10km] from a certain query point q’. It, also, may reduce to a window/a 2-d rectangle as in ‘Find data objects situated in the box [x1, x2]*[y1, y2]’. Solutions are, also, essentially based on tree-like indexing structures (Jim).

In this article, we are interested, within a spatial context, in combining the following two issues: a distance range problem and nearest neighbors search. ‘Find nearest neighbors situated within the distance range [md, Md] from a query point’ is the query model which expresses our issue.

In the literature, we find a lot of work concerning nearest neighbors search, objects ranking and range search in spatial databases (Brabec, 1999; Hjaltason, 1995; Xu, 2004). Most of the proposed solutions are based on branch and bound algorithms based on hierarchical indexing structures. When used for nearest neighbor search, indices like R-trees (Papadopolous, 2005), kd-trees and quad-trees (de Berg, 2000) do not need to pre-compute the solution (e.g. Voronoi Diagram), hence they are called object-based indexing structures whereas solution based indexing structure like D-tree (Xu, 2004). Actually, R-tree had shown many limitations in nearest neighbor search, compared to solution based indices, because of overlapping between MBRs and subsequent backtracking in the search algorithm. This was shown in (Xu, 2004) and in a previous work (Najjar, 2006). The quad-tree has a fixed partitioning scheme which splits a 2d space into four equal sub squares. This splitting makes a deficiency in indexing sparse clouds of points and non-uniformly distributed datasets.

In (Najjar, 2006), we proposed an efficient solution based indexing structure, called N-tree, to efficiently process nearest neighbor queries. We focus on extending it, in this article, to support distance range search in the context of nearest neighbor queries. We develop two approaches for answering range nearest neighbor queries with different properties and advantages. The first approach, called branch and bound, visits index nodes which are expected to be candidate by the pruning strategy. The second approach, called jumping neighborhoods scan, explores neighborhoods of the query point’s nearest neighbor to find sites falling within the query range. This exploration is done due to a leaves-linked version of the index.

The main contributions of this article can be summarized as follows:

- We proposed in (Slimani, 2008) an adapted branch and bound approach for a solution based index to process distance range queries in a spatial context;
- We propose, as an extension to (Slimani, 2008), a linked leaves index tree that allows...
15 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the product's webpage: 
www.igi-global.com/chapter/linked-neighboring-leaves-tree-support/53052?camid=4v1

www.igi-global.com/e-resources/library-recommendation/?id=19

Related Content

Service-Driven Computing with APIs: Concepts, Frameworks, and Emerging Trends
www.igi-global.com/chapter/service-driven-computing-with-apis/140793?camid=4v1a

Sizing Web Applications
www.igi-global.com/chapter/sizing-web-applications/7163?camid=4v1a

Utilisation of Case-Based Reasoning for Semantic Web Services Composition
www.igi-global.com/chapter/utilisation-case-based-reasoning-semantic/37653?camid=4v1a

Specification of Transactional Requirements for Web Services using Recoverability
www.igi-global.com/article/specification-of-transactional-requirements-for-web-services-using-recoverability/85322?camid=4v1a