Efficient query processing on large spatial databases: A performance study

George Roumelis[1], Michael Vassilakopoulos[2,⋆], Antonio Corral[3,⋆], and Yannis Manolopoulos[1,⋆]

[1] Dept. of Informatics, Aristotle University of Thessaloniki, Greece. {groumeli,manolopo}@csd.auth.gr
[2] Dept. of Electrical and Computer Engineering, University of Thessaly, Volos, Greece. mvasilako@uth.gr
[3] Dept. of Informatics, University of Almeria, Spain. acorral@ual.es

Abstract. Processing of spatial queries has been studied extensively in the literature. In most cases, it is accomplished by indexing spatial data using spatial access methods. Spatial indexes, such as those based on the Quadtree, are important in spatial databases for efficient execution of queries involving spatial constraints and objects. In this paper, we study a recent balanced disk-based index structure for point data, called xBR$^+$-tree, that belongs to the Quadtree family and hierarchically decomposes space in a regular manner. For the most common spatial queries, like *Point Location*, *Window*, *Distance Range*, *Nearest Neighbor* and *Distance-based Join*, the R-tree family is a very popular choice of spatial index, due to its excellent query performance. For this reason, we compare the performance of the xBR$^+$-tree with respect to the R*-tree and the R$^+$-tree for tree building and processing the most studied spatial queries. To perform this comparison, we utilize existing algorithms and present new ones. We demonstrate through extensive experimental performance results (I/O efficiency and execution time), based on medium and large real and synthetic datasets, that the xBR$^+$-tree is a big winner in execution time in all cases and a winner in I/O in most cases. The excellent building performance of the xBR$^+$-tree is due to the regular subdivision of space that leads to much fewer and simpler calculations. And the higher query performance of the xBR$^+$-tree, with respect to the indexes belonging to the R-tree family, is due to the combination of the regular subdivision of space, the additional representation of the minimum rectangles bounding the actual data objects and the extra termination condition applicable in certain queries and the storage order of the entries of internal nodes gave.

Keywords: SAMs, xBR-trees, R-trees, Query Processing, Performance Evaluation

Journal of Systems and Software 132, 165-185, 2017
(DOI: 10.1016/j.jss.2017.07.005)

⋆ Work funded by the MINECO research project [TIN2013-41576-R]