A 3D X-tree optimized for Complex Indoor Building Models

Lei Niu [1,2], Zhiyong Wang [3]

[1] Faculty of Geometics, Lanzhou Jiaotong University, NO. 88 Anningxi Road, Lanzhou, China, PO 730070, l.niu@lzjtu.edu.cn;
[2] Geospatial Research Innovation Development, School of Built and Environment, Faculty of Art and Design, University of New South Wales, Australia, Kensington 2030, lei.niu@unsw.edu.au;
[3] Department of Transportation at the South China University of Technology, Wushan RD NO. 381, Guangzhou, Guangdong, China, PO 510641, zwang1984@scut.edu.cn.

Keywords: 3D, Xtree, complex indoor building, models.

Abstract

Studying complex indoor buildings is crucial for population evacuation research in our modern world, where indoor spaces are the primary living environment. However, the existing spatial indexes do not fully align with the unique characteristics of indoor space, leading to a significant bottleneck in the efficiency of searching many space areas. This article introduces a novel model that can efficiently retrieve and update compact three-dimensional indoor space by introducing space X-trees. Our experiments, which have shown that the introduction of X-trees, which can dynamically adjust the size of space nodes, has significantly improved the retrieval speed during the query operation process, underscore the reliability and potential of our new model.

1. Introduction

The current spatial data is growing exponentially, so efficient organizations can avoid additional costs in filtering the required data. The current mainstream spatial data tree structure includes binary trees, quadtrees, KD trees, R trees, and R * trees (Bozhi 2020, Ma Wubin 2020, Wang Huifang 2020, You, Wang et al. 2022). These trees each have different characteristics. Below, we divide the paper into three groups for analysis based on the X-tree origin and characteristics.

Because spatial partitioning has always been related to basic human digital cognition, the use of binary trees with a minimum composite of 2 and quadtrees with a minimum composite of 4 and octrees derived from them in three-dimensional space has been widely applied. Due to their simple construction logic, these digital-based spatial indexes can be quickly applied to various applications requiring basic index structures. However, due to its simple symmetric partitioning of space, this spatial index formed by basic numbers can easily lead to unstructured partitioning of indoor spatial objects (Bozhi 2020, Ma Wubin 2020, Wang Huifang 2020, Fulin 2021, You, Wang et al. 2022).

Corresponding to binary trees, quadtrees, and octrees based on naive mathematical intuition, researchers have constructed tree heaps, binomial trees, B-trees, red-black trees, etc., based on pure computer storage efficiency optimization principles.

The third type of tree data organization structure focuses on the spatial distribution of objects. This includes KD trees, R-trees, R * trees, R+ trees, and X-trees, among others. The most significant difference between this type of structure and the previous types of trees is that its objects are mainly geographical features that comply with the first law of spatial geography (Tobler's first law) (Liming 2023, Wu Yao 2023).

There are also differences in the distribution of indoor space between buildings and ordinary space, which are manifested in two aspects: 1. the space is more compact. 2. Three-dimensional distribution is more common. These two characteristics determine that the tree index used in indoor spaces must be able to handle compact three-dimensional distributed features. Due to their characteristics, spatial trees are more suitable for application to the interior space of buildings than two types of tree structures: mathematical intuition based on trees and optimization based solely on computer science data structures(Fu Zhongliang 2012, Gan Zhaobin 2012, TAK S 2013, Silver, Huang et al. 2016, Fadli, Kutty et al. 2018, Wang and Niu 2018, Bozhi 2020). Therefore, R-trees and their derivative trees have been widely used in this space.

The R-tree was proposed by Guttman in 1984 and is characterized by using the outer bounding boxes of spatial objects as the contours of corresponding nodes. This feature is inherited based on its derivative R+ trees and R * trees, and a more optimized approach is adopted when handling boundaries. The R+ tree achieves better retrieval (Fu Zhongliang 2012) performance by redundantly inserting spatial objects into their neighboring parent nodes; The R * tree achieves a better node distribution by reinserting and deleting spatial objects multiple times (GONG Jun 2011, WEI 2019, Donghai 2020, Liu Yongshan 2020, Naizhou 2020). However, due to its fixed upper limit on node size, the R family tree cannot efficiently describe entirely nodes with colossal space that exceeds this limit, resulting in lower efficiency in processing objects in the corresponding node space. Berchtold et al. proposed the(WEI 2019, YU Anbin 2019; Donghai 2020, Liu Yongshan 2020, Naizhou 2020) X-tree in 1996 to address this issue. Strictly speaking, X-trees (Fadli, Kutty et al. 2018) also belong to the R-tree family. Their difference from R-trees lies in their flexible upper limit on node size, thus possessing the ability to handle objects in vast node spaces(Zhou and Xie 2006, Zhu, Gong and Zhang 2007, Arge, Berg et al. 2008, Gan Zhaobin 2012, Wu Yao 2023).

2. A 3D X-tree suitable for indoor building complex models

To address the challenges posed by the compact aggregation of three-dimensional objects within indoor architectural complexes and the focused layout of buildings in specific functional zones (ZLATANOVA S 2020), this article introduces a novel data structure, the 3D X-tree, tailored for modeling indoor building complexes. In the subsequent sections, we will delve into this index structure's construction principles and optimization techniques.

This contribution has been peer-reviewed.
https://doi.org/10.5194/isprs-archives-XLVIII-4-2024-353-2024

① Principle of X-Tree Construction

Constructing an X-tree involves several key steps, beginning with clustering all spatial objects. This clustering process can be achieved through a method called K-nearest neighbor clustering. In K-nearest neighbor clustering, categories are predefined with 'K' as the expected value, and cyclic clustering operations are performed on existing spatial objects based on this value until the final spatial categories do not exceed the preset value of 'K'.

The implementation process entails randomly selecting 'K' class center points and adding corresponding categories and nodes based on the nearest neighbor principle. Once all nodes are added to their respective clusters, the center of each cluster is calculated using a weighted average of the coordinates of all points within the cluster. This process is repeated iteratively until the total number of clusters remains unchanged.

After conducting K-means clustering, the resulting clusters are organized into nodes of a red-black tree. Red-black trees are binary trees that are self-balancing internally, with additional color indicators compared to standard binary trees. These trees exhibit specific characteristics, such as the root node being black, the child nodes of the root node being red, and all leaf nodes being black. The most critical operations for red-black trees include node insertion and deletion.

Insertion and Deletion Operations in Red-Black Trees

Insertion Operation:

1. If the root node is inserted, it is directly colored black.
2. If the parent node of the inserted node is black, no color adjustment is needed.
3. If the parent node of the inserted node is red and its sibling nodes are also red, the color of the parent node and its sibling nodes is adjusted to black, while the color of the grandfather node is red.
4. The red-black tree's properties are rechecked to ensure compliance.
5. If the parent node of the inserted node is red and its sibling nodes are black, and the insertion node, parent node, and grandfather node are on the same side, the parent node is changed to black, the grandfather node to red, and a reverse rotation is performed on the grandfather node.
6. If the parent node's red violates property four, and its sibling nodes are black, the insertion node, parent node, and grandfather node are on the same side, a rotation is performed on the parent node, and then the situation is handled as in case 4.
7. If the parent node's red violates property four, and its sibling nodes are black, the insertion node, parent node, and grandfather node are not on the same side. A rotation is performed on the parent node to align it with the insertion node, and then the situation is handled as in case 4.

Deletion Operation:

1. Deletion operations are classified into deleting leaf nodes and identifying leaf nodes for deletion.
2. The successor or predecessor node of the node to be deleted is found, and if it exists, it replaces the current node.
3. Replacement of subsequent or predecessor nodes continues until no successor or predecessor nodes remain.
4. The red-black tree is adjusted to maintain its properties.
5. Finally, the nodes are truly deleted.

Optimization of X-Tree for Three-Dimensional Indoor Architectural Space

During the clustering process, the X-tree can accommodate super-large spatial nodes, resulting in clusters of super nodes containing numerous functional areas based on the spatial distribution trends within the studied building area. This optimization ensures that the X-tree efficiently represents the intricate spatial relationships within indoor architectural complexes, facilitating various analytical tasks and simulations tailored to the indoor environment's specific characteristics.:

Insert node:

1. If the root node is inserted, directly turn the point black;
2. If the parent node of the inserted node is black, the color will not be adjusted;
3. If the red node of the parent node of the inserted node is a red node, and the sibling nodes of the parent node are red,
4. Turn the parent node and its sibling nodes black, and turn the grandfather node-red;
5. Check again whether the grandfather node violates the properties of the red and black tree;
6. If the red node of the parent node is inserted, and the parent node's sibling nodes are black, the insertion node, parent node, and grandfather node are on the same side. Turn the parent node into a black node, turn the grandfather node into a red node, and rotate the grandfather node in reverse (if left, rotate right; if right, rotate left).
7. If the red node of the parent node of the inserted node violates property four, and the sibling nodes of the parent node are black. Moreover, insert nodes, parent nodes, and grandfather nodes on the same side.
8. Rotate the parent node to make it on the same side (case 4), then handle it according to situation 4.

Delete node:

1. Classify all deletion operations as deleting leaf nodes and obtain the leaf nodes that need to be genuinely deleted;
2. Find the successor node or predecessor node of the node to be deleted, and if it exists, replace the current node;
3. Replace the subsequent or predecessor nodes with the replaced nodes. Replace until there are no successor or predecessor nodes;
4. Adjust the red and black trees to ensure that the deleted nodes do not violate the properties of the red and black trees;
5. Truly delete nodes.

② Optimization of X-tree for three-dimensional indoor architectural space

During the clustering process, due to the ability of the X-tree to accommodate super large spatial nodes, there will be individual clusters of super nodes containing a large number of functional areas based on the spatial distribution trend of the studied building area. This attribute is indispensable for regions containing several super large areas with intensively more complex inner relationships than normal nodes on the same level. Thus, our model will allow the integration of various sizes and complexity nodes on the same level. However, this integration should be carefully supervised by a reflexible size and spatial complexity. We implement this function in our approach by setting a multiplier for the upper size and complexity range limitations. The lower limit of the nodes on the same level is 0.8 * the normal nodes range settings compared to the standard R tree, and the upper limit caps should be the typical ratio * 1.2 ~ more large figure. This cannot easily be pre-defined before the actual applications and should be testified by a general growing-up process test. This growing-up process is implemented by fusing the same nodes on this level, which is covered by a super large X node covering an area as 80% of the total sub-nodes on the neighboring child level.

However, a rigid theory has not yet mathematically proved our X-node theory. However, each level should have some super large node, which is an intuitive product of applying the Pareto principle to the node settings; this means some super nodes should occupy 80% of the total area or the number of chlld levels. This naive Pareto principle may also sometimes fail, under these situation, we should drop to a backup plan when needed. This plan describes that a super large or complex X ode could not be generated under extreme cases. Then, we could only allow the current research-level nodes to contain up to 80% of the whole node in the research area.

3. Experiments and Results

This experimental design aims to demonstrate the flexible setting of the upper limit of leaf node space size by introducing the X-tree index relative to the R-tree index and to optimize the operational performance of peer nodes and edges in the interior space of buildings. The selected experimental area is Shanghai Meiluo City, a famous landmark building in Shanghai, characterized by its spherical external features, as shown in Figure 1. Some regular parts and spherical-shaped parts of this building have different heights and internal space settings. Thus, this model is complex enough. Given this characteristic, this building can be optimized for spatial operations using the X-tree proposed in this article.

Figure 1. Demonstration of the Meiluocheng experiment building.

The comparative operations include two typical spatial application scenarios: querying and adding nodes. To suppress other uncontrollable external interference factors, 1000 operations were taken to compare the performance of R-trees and X-trees.

① Experimental data and platform introduction

The experimental data used in this article is from Meiluo City, located in the bustling urban area of Shanghai. A complex internal peer network and a multi-story building structure characterize the building. Due to the characteristics of this building, conducting experiments and analysis on it can reflect the applicability and performance characteristics of the model proposed in this article. The experimental platform uses the Ryzen 3950X CPU platform, which has

② Analysis of experimental results

This article provides a comparative analysis of the selected operations. Due to the different periods required for other operations, only random points were chosen for multiple cumulative operations for query operations. Then, the average time for timing and unit operations was calculated. The absolute values of the experiment are 0.0339369 seconds for constructing an R-tree index, 0.282185 seconds for constructing an X-tree index, 0.00297083 seconds for querying an R-tree index, 0.0021 seconds for querying an X-tree index, 2.4008 seconds for inserting an R-tree index, and 5.4569 seconds for inserting an X-tree index. The comparison results between the obtained data are shown in the following figure 2.

③ Analysis of experimental results

From the experimental results, it can be seen that during the process of constructing an index, the X-tree requires significantly more time than the R-tree. This additional time consumption is mainly due to the dynamic consideration of the size limit of its parent node while adding its child nodes. On the other hand, R-trees do not need to consider this aspect. However, the speed advantage of constructing R-tree indexes comes at the cost of sacrificing the efficiency of subsequent query operations, so this phenomenon must be carefully considered when selecting these two types of indexes. The X-tree showed significantly more time consumption during inserting indexes than the R-tree. After in-depth analysis, a possible explanation is that the X-tree needs to dynamically consider the space size occupied by the maximum bounding rectangle of each node's child nodes, resulting in significantly more insertion time consumption than the R-tree.

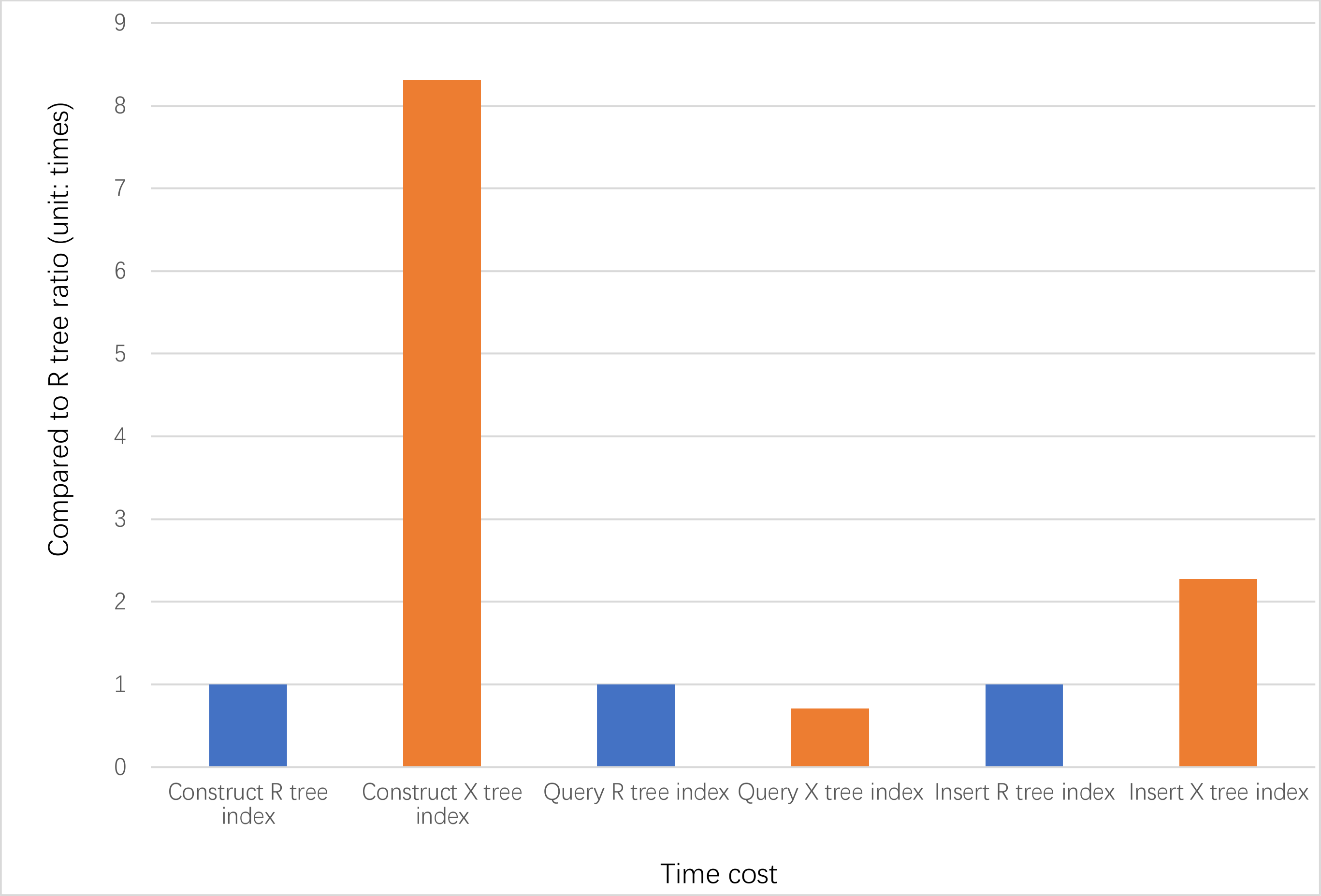

Figure 2. Comparison of time consumption for constructing, querying, and inserting R-trees and X-trees

4. Conclusion and Outlook

The X-tree index proposed in this article can significantly improve query efficiency compared to the standard R-tree index, and this performance improvement comes at the cost of initializing the construction of the overall index and increasing the number of index objects. This characteristic should be kept in mind in practical use. Although the principle of X-tree is only a naive Pareto principle applied in indoor GIS research, the function of scanning and fusing child nodes to fuse their outer shells to form a super node called ‘X’ is still an innovative point of our research. However, the negative side of introducing should also draw enough attention, which could cause an unstable operation efficiency.

The next step is to add more experimental scenarios, such as selecting more types of indoor building spaces, including incorporating test data from libraries, sports venues, and other large venues. In addition, it is necessary to include more spatial operations, including deleting and updating nodes. Furthermore, continuously optimizing the efficiency of the X-tree, such as incorporating Hilbert curves into index values, is also a beneficial exploration path for the future. The reason is also simple: the spatial traversing order of the nodes on the same level is crucial because the X nodes of the current trees on a specific level are quite different. Because the various orders of applying the traversing process at the same level could form different directions and shapes of an X node.

Acknowledgment

This paper is funded by the NSFC No. 42171401 and the Central Universities"(NO.2023ZYGXZR056). "the Fundamental Research Funds for the Central Universities"(NO.2023ZYGXZR056). The authors thank all the sincere support from the relevant researchers.

References

Arge, L., et al. (2008). "The priority R-tree: A practically efficient and worst-case optimal R-tree." *ACM Transactions on Algorithms (TALG)* **4**(1): 1-30.

Bozhi, Q. (2020). Research on Linkage Spatial Indexing Combining LSM Trees.

Donghai, S. (2020). "Distributed Mobile Object Spatial Indexing for Spatiotemporal Data Streams."

Fadli, F., et al. (2018). Extending indoor open street mapping environments to navigable 3D CityGML building models: Emergency response assessment. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*.

Fu Zhongliang, L. S., Tian Zongshun, etc. (2012). "Research on Distributed Spatial Index and Query Validation Method Based on Multi level R-tree.". *Surveying and Mapping Bulletin*, (11): 42-46.

Fulin, Z. (2021). "Research on Secure Queries Based on Location Privacy in Space Crowdsourcing Environment ".

Gan Zhaobin, Y. Y., Zhao Yizhu, etc. (2012). "Research on indoor moving object indexing based on DR tree." *Computer Science* **39**(10): 177-181.

GONG Jun, Z. Q., ZHANG Han, et al. (2011). "Adaptive Control Method of Three-dimensional Scene Detail Level Based on R-tree Index." *Acta Geodaetica et Cartographica Sinica* **40**(4): 531-534.

Liming, W. (2023). Design and Implementation of Trajectory Query Method Based on Distributed Index.

Liu Yongshan, G. X., Kong Dehan, etc. (2020). "Research on Hybrid Indexing Method Based on 3D Grid-R Tree." *Journal of Yanshan University* **44**(2): 138-149.

Ma Wubin, W. R., Wu Yahui, etc. (2020). "A Text Space Tree Clustering Algorithm Based on Improved NSGA-III." *Journal of Huazhong University of Science and Technology (Natural Science Edition)* **48**(5): 86-92.

Naizhou, H. (2020). R-tree construction based on improved Chameleon algorithm.

Silver, D., et al. (2016). "Mastering the game of Go with deep neural networks and tree search." *Nature* **529**(7587): 484.

TAK S, C. A. (2013). "Enhanced spatial stability with hilbert and more trees." *IEEE Transactions on Visualization and Computer Graphics* **19**(1): 141-148.

Wang Huifang, G. Y., Gao Peng (2020). "A minimum polygon construction algorithm based on quadtree index." *Geospatial Information* **18**(11): 118-120+118.

Wang, Z. and L. Niu (2018). "A Data Model for Using OpenStreetMap to Integrate Indoor and Outdoor Route Planning." *Sensors* **18**(7): 2100.

WEI, Y. (2019). Research on Point Cloud Compression Technology Based on Spatial Index.

Wu Yao, Z. R., Wu Jie (2023). "A hybrid deployment R-tree for remote persistence and write merging." *Small Micro Computer Systems* **44**(12): 2858-2864.

You, D., et al. (2022). "EfficientUNet plus : A Building Extraction Method for Emergency Shelters Based on Deep Learning." *Remote Sensing* **14**(9).

YU Anbin, M. W. (2019). "A Massive Subway Tunnel Point Cloud Management Method Combining R-tree with Grid." *Journal of Wuhan University (Information Science Edition)* **44**(10): 1553-1559.

Zhou, G. and W. Xie (2006). Accuracy improvement of urban true orthoimage generation using 3D R-tree-based urban model. *Proceedings of the 2006 international conference on Digital government research*, ACM.

Zhu, Q., et al. (2007). "An efficient 3D R-tree spatial index method for virtual geographic environments." *ISPRS Journal of Photogrammetry and Remote Sensing* **62**(3): 217-224.

ZLATANOVA S, Y. J., WANG Y, et al. (2020). "Spaces in Spatial Science and Urban Applications—State of the Art Review." *ISPRS International Journal of Geo-Information* **9**(1): 58.