

# STAR-TREE: AN INDEX STRUCTURE FOR EFFICIENT EVALUATION OF SPATIOTEMPORAL AUTHORIZATIONS<sup>1</sup>

Vijayalakshmi Atluri and Qi Guo

**Abstract** Typically, *geospatial* data include satellite imagery, digital orthophoto quads, maps, digital line graphs, census data, voter registration, land ownership data, and land use data. This data is considered sensitive based on its location (characterized by its longitude and latitude), resolution, and the time of capture, among others. Since both objects and authorizations are associated with spatial and temporal attributes, in order to process access requests efficiently, it is essential that they both be accessed using some sort of index structures. However, processing an access request under this approach requires searching two indexes – one the object index and the other the authorization index. In this paper, we propose a novel index called STAR-Tree, a Spatio Temporal Authorization-driven R-Tree, that can uniformly index both spatiotemporal objects and the authorizations that govern access to them. STAR-Tree is an extension of R-tree that allows objects of different resolutions be indexed based on their spatial and temporal attributes, as well as allows layering of spatiotemporal authorizations on the tree itself. Compared to the previously proposed RMX-Quadtree, STAR-Tree enjoys several advantages. First, the 3 dimensional nature of the STAR-Tree accommodates the temporal dimension. Second, the STAR-Tree imposes no restrictions on the region covered by the geospatial objects. Third, in the STAR-Tree images of the same resolution may overlap with one another. We demonstrate how such a tree can be constructed and maintained, and show how access requests can be processed in an efficient manner.

## 1. INTRODUCTION

Geospatial data can either be digital raster images that store image as a number of pixels, or digital vector data that store image as points, lines and polygons. Typically, satellite images, digital orthophoto quads and scanned

<sup>1</sup> The work is supported in part by the National Science Foundation under grant number IIS-0242415.

maps are raster images, while maps of vector type (e.g. Shape file), digital line graphs, or census TIGER data are vector images. Other non-image geospatial data sets are data with locational information, such as census data, voter registration, land ownership data, and land use data. Typically, this data is associated with the time of capture and/or the time of its validity. This data is considered sensitive based on its location (characterized by its longitude and latitude), resolution, and the time of capture, among others. For example, a high-resolution satellite image<sup>2</sup> overlaid with a vector road map when coupled with publicly available data can pose a significant threat to both privacy as well as national security [8]. Policies for prohibiting the release of imagery beyond a certain resolution (such as the guidelines provided by the Department of Commerce), notifying when an image crosses an international boundary, or when such a request is made, etc., are beginning to emerge [1,2]. Currently, commercial organizations (e.g., Space Imaging [3]) enforce several security policies while distributing images beyond a certain resolution covering a specific region. As an example, no American remote sensing company can collect or disseminate imagery of Israel at a resolution higher than two meters.

In other words, both objects as well as authorizations associated with these objects are multidimensional comprising of the x and y spatial coordinates, the time, and the resolution. In addition, the subject credentials also have a valid time interval associated with them [7]. To facilitate such controlled access and dissemination of satellite imagery and the information products generated from it, a suitable access control system must be in place. Recently, a GeoSpatial Authorization Model (GSAM) has been proposed [4] that allows specification of authorizations on geospatial objects based on their resolutions. GSAM supports, in addition to read, insert, delete and modify, other privilege modes such as view, zoom-in, overlay and identify. Since user credentials as well as the image objects have temporal attributes associated with them, a temporal extension of GSAM has been proposed in [5], in which authorizations can be specified based on the temporal characteristics of user credentials and objects, such as the time at which the object was created and the time during which it is valid. Enhancement with a temporal dimension makes it possible to provide access only during a certain predefined time interval [6], as long as a user holds a credential, or entitled to play a role [7]. In addition, this temporal validity of authorizations may allow the support of more articulated access control policies. For example, if

<sup>2</sup> There are now more than 15 commercial satellites with resolutions from 1 - 30 meters (e.g., IKONOS, ORBVUE, EROS and QUICKBIRD are privately owned and provide images with resolution of 1 meter or better.) One-meter imagery enables viewing of roads, houses, automobiles and aircrafts, and will make it possible to create highly precise digital maps and three-dimensional fly-through scenes. These are available at prices as low as \$10.

a subject rents a property during a certain time interval, he should be allowed access to the images of this region captured within this time interval.

Although implementation of authorizations as access control list, capability list or access matrix is suitable for traditional data, it is not suitable for spatiotemporal data. When a user issues a request to access an object(s), specified either as an area or an object identifier, in a specific privilege mode, the system must first verify whether such an authorization exists in the authorization base. Then it must retrieve the relevant objects from the image database. As such, serving an access request requires searching for authorizations from the authorization base, as well as searching for relevant images. Since both objects and authorizations are associated with spatial and temporal attributes, in order to process access requests efficiently, it is essential that both the geospatial object database as well as the authorization base be accessed using some sort of index structures. However, processing an access request under this approach requires searching two indexes – one the object index and the other the authorization index. In this paper, we propose a novel index called STAR-Tree, a Spatio Temporal Authorization-driven R-Tree [12], that can uniformly index both spatiotemporal objects and the authorizations that govern access to them. STAR-Tree is an extension of R-tree that allows objects of different resolutions be indexed based on their spatial and temporal attributes, as well as allows layering of spatiotemporal authorizations on the tree itself. As a result, processing of access requests can be done more efficiently as it is necessary to search only one index structure. STAR-Tree is a three dimensional R-tree that is capable of storing multiple resolution spatiotemporal objects as well as allows layering of spatiotemporal authorizations over the index.

When compared to the RMX-Quadtree [8], STAR-Tree enjoys several advantages. The notable ones are: (i) The 3 dimensional nature of the tree allows objects as well as authorizations to be indexed based on their temporal attributes (in addition to indexing them based on their spatial attributes). As a result, access requests based on the temporal as well as spatial attributes can be evaluated efficiently. (ii) Since STAR-Tree is based on the R-tree, unlike the RMX-Quadtree, it imposes no restrictions on the region covered by the geospatial images. In case of the RMX-Quadtree, it requires that the region covered by the parent is a perfect square and divided into exactly four equal quadrants represented by its four children. Since each level in the tree represents a different resolution, the above limitation implies that a low-resolution image must be divided into equal squares of higher resolution images, which is a severe limitation. (iii) In the STAR-Tree, images of the same resolution may overlap with one another, whereas they were not allowed in the RMX-Quadtree.

This paper is organized as follows. Section 2 reviews the GSAM. Section 3 first presents the building blocks that are required to construct the STAR-Tree and demonstrates how it can be constructed from these building blocks to accommodate spatiotemporal objects of different resolutions. It then discusses the challenges involved in inserting and deleting objects and briefly discuss the costs associated with them. Section 4 demonstrates how authorizations can be layered on top of the STAR-Tree. Section 5 presents our approach to process an access request that involves both searching for an image and evaluation of an authorization simultaneously. Section 6 provides some insight into our current and future research in this area.

## 2. GEOSPATIAL AUTHORIZATION MODEL

In this section, we review the GeoSpatial Authorization Model (GSAM) proposed earlier in [4] and later extended with temporal attributes in [5].

*Objects* in GSAM include geospatial *raster* images with multiple resolutions that represent a geographical region. We assume that each geospatial image object is associated with a unique identifier, *id*, and comprises of spatial features including *latitude(l)*, *longitude(g)*, *height(h)*, *width(w)*, *resolution(r)*, *timestamp(t)*, (either image download time or last update time), and a *link(k)*<sup>3</sup>. In addition to the raster images that store image as a number of pixels, objects include digital *vector* data that store image as points, lines and polygons, In addition, objects include *tabular* data linked to the images, which includes thematic layer information such as census data, voter registration, land ownership data, and land use data.

GSAM supports three types of *privilege modes* -- *viewing*, *copying* and *maintenance*. The viewing modes include *static* and *dynamic* types. Static viewing modes in turn include *view-thumbnail*, *view-annotation* and *view*. Dynamic viewing modes include *zoom-in*, *overlay*, *identify*, *animate* and *fly-by*. The view privilege allows a user to see an image object covering a certain geographic area, the zoom-in privilege allows a user to view an image covering a certain geographic area at a specific higher resolution, the overlay privilege allows users to generate composite images, where a composite image is constructed by overlaying one image on top of another, and finally, the identify privilege allows the user to view the tabular data linked to an image. The copying modes, *download* and *download-data*, allow source files to be downloaded. The maintenance modes include *insert*, *delete*, *update* and *compose*. Unlike the text data where the display privilege

<sup>3</sup> If objects are referred by canonical landmarks such as the name of the city or street, a geographic information system, such as a gazetteer service, can automatically convert this into the corresponding coordinates.

implies the copying privilege, the viewing and copying are distinguished as separate privileges with geospatial data since the objects displayed on the web browser often are image gif files, but not the original source files.

We assume that a subject is associated with a set *credentials*. Each subject credential belongs to a *credential type* [9]. These credential types can be organized into a hierarchical structure. The set of credentials each subject holds specifies to which credential type that subject belongs, and the values for attributes in each credential type. *Subjects* belonging to a credential type inherit privileges from those at upper levels. Let  $S = \{s_1, s_2, \dots\}$  denote a set of subjects,  $C = \{c_1, c_2, \dots\}$ , a set of credential types,  $O = \{o_1, o_2, \dots\}$  a set of objects, and  $M$  is a finite set of privilege modes.

Definition 1: [Authorization] [5] An authorization  $a$  is a quadruple  $\langle ce, ge, pr, \tau \rangle$ , where  $ce$  is a credential expression that denotes a group of authorized subjects,  $ge$  is a geotemporal object expression that denotes a set of authorized geospatial objects or geospatial extents,  $pr$  is a set of privilege modes in  $M$ , and  $\tau$  is a temporal term, which is an expression denoting the time interval during which the authorization is allowed.<sup>4</sup>

An example of the credential expression is as follows:  $ce_1$ : (Property owner(x)  $\wedge$  (home-address = '123 James Street, Newark, NJ')  $\wedge$  (owner-period before '1999')): This expression denotes all property owners of '123 James Street, Newark, NJ' before 1999.

An example of geotemporal object expression is as follows:  $ge_1$ : (type(x)=image  $\wedge$  rectangle(x) contains (10, 20, 10, 10)  $\wedge$  timestamp(x) within [Feb2, 2002, Mar1, 2002]  $\wedge$  resolution(x) > 10m). This specifies a set of images whose spatial extent contains the area of (10, 20, 10, 10) downloaded between February 2, 2002 and March 1, 2002, and whose resolutions are higher than 10 meter. Note that objects (as well as access requests) can be specified using its coordinates (longitude, latitude width and length), a region expressed by ZIP codes or highway mile markers, or by canonical landmarks such as city or street name. We assume that there exists a gazetteer service (<http://www.geocode.com/eagle.html-ssi>) that converts place names to coordinates.

Examples of policies that can be specified using the authorization specification  $\langle ce, ge, pr, \tau \rangle$  include: (i) A parcel located in "123 James Street, Newark, NJ" can be accessed by the current owner of this parcel, and (ii) 1 meter resolution images over Afghanistan after September 11, 2001 should not be made available to non-military personnel. The policies are both spatial and temporal based.

<sup>4</sup> Due to space limitations, we do not present the formalism [5] adopted in formulating these expressions; we simply present the formal definition and explain it with examples.

### 3. THE STAR-TREE

This section is organized as follows. In section 3.1, we develop the necessary formalism to construct the STAR-Tree. In section 3.2, we present the **STAR<sup>r</sup>-Tree**, which is a simplified version of the STAR-Tree comprising of images of uniform resolution  $r$ . In section 3.3, we demonstrate how the STAR-Tree can be constructed. Our work is influenced by the index structure for spatiotemporal data, proposed by Theodoridis et al. [10].

#### 3.1 Preliminaries

**3-Dimensional Space:** Essentially, STAR-Tree is a 3-Dimensional variant of the R-tree when applied to 3-dimensional spaces [14]. Each of the  $x$ ,  $y$  and  $t$  axes of the *3-Dimensional space* corresponds to the horizontal extent, vertical extent and the timestamp of the image, respectively. As can be seen from Figure 1, by intersecting the  $t$ -axis at a certain point, the polygon representing the covered region of an image can be viewed as a plane parallel to both  $x$  and  $y$  axes in the 3D space shown.

**Minimum-Bounding Rectangle:** Typically, the regions covered by our authorization objects can assume any shape since they represent geographical regions (e.g., city, county). For the sake of simplicity and efficiency, we approximate the region of an image to a *minimum bounding rectangle* (MBR). An MBR is the tightest rectangle among all the rectangles that entirely encloses the image. For example, a, b, c, d, e and f in Figure 1 are MBRs.

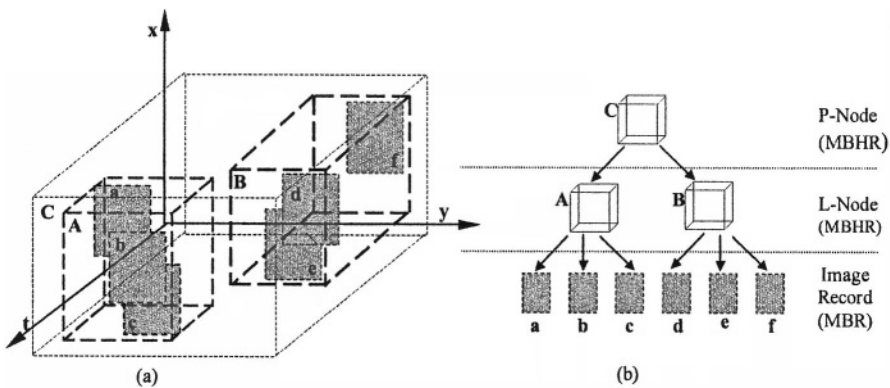


Figure 1. Minimum bounding rectangles and the hierarchical structure

**Minimum Bounding Hyper Rectangle:** Given a set of MBRs, we define a *Minimum Bounding Hyper Rectangle* (MBHR) as the tightest rectangle prism in the 3D space that encloses the set of all MBRs. An

MBHR may also be constructed recursively, as the minimum bounding hyper rectangle of a set of MBHRs. For example, in Figure 1, A is the MBHR for MBRs a, b and c, B is the MBHR for MBRs d, e and f, and C is the MBHR for MBHRs A and B. With any changes to the set of MBRs due to the insertion and deletion operations, in order to hold the minimum bounding property, the MBHR should change accordingly to tightly enclose the constituent MBRs. In other words, the MBHR must be sensitive to any spatial and/or temporal alternations of its element MBRs and must be adjusted.

Multiple levels of MBHRs can form a hierarchical structure, as can be seen from Figure 1(b). First, at the bottom most level of the hierarchy, several sets of MBRs are clustered together to form a set of MBHRs. The MBHRs of the next higher level are the bounding hyper rectangles of the first level MBHRs. The root of the tree is thus the bounding hyper rectangle covering the lower level layers in a recursive manner. Each MBHR has exactly the same attributes as its children MBHRs except being larger in terms of its magnitude. Several strategies exist to compute the minimum boundedness, such as the smallest sum of volumes of the MBHRs, the smallest sum of the overlapping regions among the MBHRs [13], etc. One may adopt an appropriate strategy based on the application under consideration. It is important to note that the cost of computing the MBR and MBHR are non trivial irrespective of the strategy adopted[11,12]. However, this discussion is beyond the scope of this paper.

**L-nodes and P-nodes:** Leaf nodes, called L-nodes, are the set of nodes at the lowest level of the hierarchy, whose MBHRs are composed of the MBRs representing the image records. On the other hand, propagation nodes, called P-nodes, are the non-leaf nodes. The MBHRs of the P-nodes are composed from the MBHRs of their children nodes. In other words, P-nodes include the parents as well as the ancestors of L-nodes. We use L-node.MBHR and P-node.MBHR to denote the MBHRs covered by the L-node and P-node, respectively.

**$L_r$ -node and  $P_r$ -node:** Given a  $STAR^r$ -Tree, we denote the L-node and the P-node of it as  $L_r$ -node and  $P_r$ -node. In other words, both the  $L_r$ -node and  $P_r$ -node comprise of the MBHRs resulted from the images of the same resolution  $r$ .

**$P_R^m$ -node:** A merged P - node, called  $P_R^m$ -node, represents a P-node merged with an L-node, a P-node merged with another P-node. The following five possible cases of merging will result in a  $P_R^m$ -node:

1. L-node merged with a P-node: An  $L_{r_1}$ -node merged with a  $P_{r_2}$ -node such that  $r_1$  and  $r_2$  are two resolutions where  $r_1 \neq r_2$ , results in a  $P_R^m$ -node where  $R = \{ r_1, r_2 \}$

2. An L-node merged with a Merged P-node: An  $L_r$ -node merged with a  $P_{R_1}^m$ -node where  $r \notin R_1$ , results in a  $P_R^m$ -node where  $R=R_1 \cup \{r\}$
3. A P-node merged with another P-node:  $P_{r_1}$ -node and  $P_{r_2}$ -node: A  $P_{r_1}$ -node merged with a  $P_{r_2}$ -node such that  $r_1$  and  $r_2$  are two resolutions where  $r_1 \neq r_2$ , results in a  $P_R^m$ -node where  $R=\{r_1, r_2\}$
4. A P-node merged with a Merged P-node:  $P_r$ -node and  $P_R^m$ -node: A  $P_r$ -node merged with a  $P_{R_1}^m$ -node, results in a  $P_R^m$ -node where  $R=R_1 \cup \{r\}$
5. A Merged P-node merged with another merged P-node:  $P_R^m$ -node and  $P_{R_2}^m$ -node: A  $P_{R_1}^m$ -node merged with a  $P_{R_2}^m$ -node, results in a  $P_R^m$ -node where  $R=R_1 \cup R_2$

It is important to note that, while a  $P_r$ -node is comprised of MBHRs constructed from images of the same resolution  $r$ , a  $P_R^m$ -node comprises of MBHRs constructed from images of different resolutions. Moreover, while a  $P_r$ -node may transform into a  $P_R^m$ -node, an  $L_r$ -node would never become a  $P_R^m$ -node. Furthermore, an L-node is always an  $L_r$ -node for some  $r$ .

In the image record, the  $\text{width}(w) = |x_b^I - x_e^I|$ , and  $\text{height}(h) = |y_b^I - y_e^I|$ ,  $\text{latitude}(l) = x_b^I$  and  $\text{longitude}(g) = y_b^I$ ,  $\text{latitude} + \text{width} = x_e^I$ , and  $\text{longitude} + \text{height} = y_e^I$ , where  $x_b^I, x_e^I, y_b^I$  and  $y_e^I$  stand for the beginning and ending points of spatial coordinates of MBHR on X-axis and Y-axis respectively. Here the superscript I refer to the Image Record. We use  $t^I$  to denote the image timestamp, and  $r$  to denote its resolution level. We also assume an image record comprises of the metadata of the image. The node structure of  $P_R^m$ -node is same as that of a  $P_r$ -node except that it represents multiple resolutions. In an L-node, we use  $\text{IR\_Count}$ ,  $\text{IR\_Ptr}$ ,  $\text{A}_s\text{-Ptr}$ ,  $\text{A}_O\text{-Ptr}$ ,  $[x_b^L, x_e^L], [y_b^L, y_e^L], [t_b^L, t_e^L]$  and  $r$  to denote pointer to each image record, pointer to authorization set S-Auth, pointer to set of overlaid authorizations, spatial coordinates of MBHR on X-axis, spatial coordinates of MBHR on Y-axis, time interval of MBHR and resolution level(s), respectively. S-Auth and overlaid authorizations will be discussed in section 4. Similarly, in case of M-node, we use  $\text{M\_Count}$ ,  $\text{M\_Ptr}$ ,  $\text{A}_O\text{-Ptr}$ ,  $[x_b^P, x_e^P], [y_b^P, y_e^P], [t_b^P, t_e^P]$  and  $r$  to denote the number of MBHRs stored per node, pointer to each L/P-node MBHR, pointer to set of overlaid authorizations, spatial coordinates of MBHR on X-axis, spatial coordinates of MBHR on Y-axis, time interval of MBHR and resolution level (s), respectively.

**Branching Interval:** We define *Branching Interval*  $[B_l, B_u]$ , where  $B_l$  and  $B_u$  indicate the lower and upper limit of the number of  $\text{IR\_Ptr}$  ( $\text{M\_Ptr}$ ) (essentially the number of children) accommodated in each L-node (P-node), except the root.

**Spatial (temporal) overlap, enclose and disjoint relationships:** The spatial and temporal relationships of MBHRs are a little complicated due to



the inclusion of the concept of time. If we consider the spatial and temporal features separately, the relationship between two MBHRs can be classified into 3 categories, overlap, enclose and disjoint, which are represented by  $\cap_S$ ,  $\cap_T$ ,  $\supset_S$ ,  $\supset_T$ ,  $\otimes_S$  and  $\otimes_T$ , respectively, where subscripts S and T denote spatial and temporal relationships. In fact, there exist 9 possible combinations of spatial and temporal relations between two MBHRs. The following three of these 9 cases,  $\cap_{S,T}$ ,  $\supset_{S,T}$ ,  $\otimes_{S,T}$ , represent overlap, enclose and disjoint in spatial AND temporal dimensions, respectively.

### 3.2 The STAR<sup>r</sup>-Tree

A **STAR<sup>r</sup>-Tree** is a special case of the STAR-Tree, which is constructed with the images of uniform resolution  $r$ . It can simply be viewed as the hierarchical structure (shown in Figure 1(b)), comprising of **L<sub>r</sub>-nodes** and **P<sub>r</sub>-nodes**, where **L<sub>r</sub>-nodes** point to image records corresponding to the MBRs of regions covered by images with resolution  $r$ . **P<sub>r</sub>-nodes**, on the other hand are the ancestors of the **L<sub>r</sub>-nodes**. Conceptually, we use several **STAR<sup>r</sup>-Trees** representing images at different resolution levels, in order to build the STAR-Tree. In the **STAR<sup>r</sup>-Tree**, only the leaf nodes contain the pointers to the image records, however, the propagation nodes do not contain any pointers to the image records.

The height of the **STAR<sup>r</sup>-Tree** can be calculated as follows. Let the branching interval be  $[B_l, B_u]$ , and  $N$  be the total number of images (MBRs) to be stored. In order to guarantee that the number of children of each leaf fits into the branching interval, the minimum level of granularity,  $H$  (height of the tree), can be formulated as:

$$H \in \left[ \left\lceil \log_{B_u} N \right\rceil, \left\lceil \log_{B_l} N \right\rceil \right]$$

For images of each resolution level  $r$ , we construct a corresponding **STAR<sup>r</sup>-Tree** in which the number of children in each non- root **P<sub>r</sub>-node** falls within the predefined Branching Interval.

### 3.3 Construction of the STAR-Tree

A STAR-Tree is constructed by systematically merging multiple **STAR<sup>r</sup>-Trees** by taking into account their resolutions as well as the spatiotemporal regions covered by the MBHRs. Essentially, we exploit some of the following properties of the geospatial data and the corresponding

authorization specifications in constructing the STAR-Tree, thereby providing efficient search and evaluation of access requests. (1) Geospatial images of the same resolution have the same extent of geographical coverage, and therefore, lower resolution images cover larger geospatial region than their higher resolution counterparts. In other words, the width and length of the spatial rectangle covered by an image are functions of its resolution. (2) If a subject is allowed to view a region with a certain level of resolution, he is always allowed to view the same region with a lower resolution. Moreover, subject classes inherit the privileges of their super-classes. As a result, if lower resolution images are placed at higher levels of the tree, the authorizations pertaining to these can be placed at those levels, whereas those pertaining to the higher resolutions can be placed at lower levels of the tree. With this organization, access requests can be evaluated by traversing the tree to a certain depth as opposed to the entire tree.

In contrast to the **STAR<sup>r</sup>-Tree** that contains **L<sub>r</sub>-nodes** only at the leaf level pointing to images with single resolution  $r$ , the STAR-Tree accommodates **L<sub>r</sub>-nodes** pointing to images with different resolutions, being at multiple levels on the tree. Specifically, the **L<sub>r</sub>-nodes** with different resolution levels reside at separate levels on the STAR-Tree. Supposing that we have  $n$  distinct resolutions, and then the last  $n$  levels of the STAR-Tree consist of **L<sub>r</sub>-nodes** of different resolutions, such that the **L<sub>r</sub>-nodes** representing the highest resolution are placed at the bottommost level and those consisting of the lower resolution are placed at the higher levels. In other words, the higher the resolution of the **L<sub>r</sub>-node**, the lower its level in the STAR-Tree. We denote the level at which **L<sub>r</sub>-nodes** with resolution  $r$  are placed as **H<sub>r</sub>**.

We denote  $O$  as the set of images with various resolutions. In Figure 2(a),  $A$  and  $B$  are two **STAR<sup>r</sup>-Trees**, constructed from images in  $O$ , with resolutions being 1 and 2 respectively. We assume that the resolution level of 2 is higher than that of 1. Therefore, we choose  $B$  as the base tree. Then we insert the 5 **L<sub>1</sub>-nodes** at the leaf level of  $A$  into the nodes at one level higher than the level of **L<sub>2</sub>-nodes**. Nodes at higher levels of the  $B$  are adjusted accordingly, resulting in  $C$ , the STAR-Tree as shown Figure 2(b).

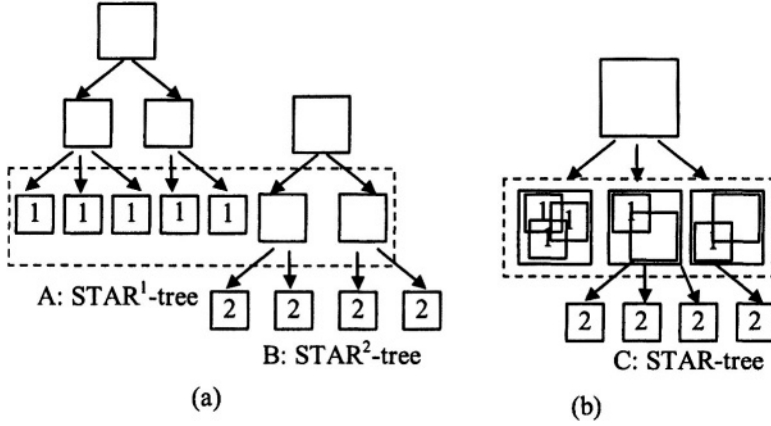


Figure 2. Construction of the STAR-Tree

The detailed steps of constructing the STAR-Tree are as follows:

- 1) Assort the images of the same resolution into groups.
- 2) Construct a **STAR<sup>r</sup>-Tree** for each of the image groups clustered by the resolution level  $r$ .
- 3) Pick the **STAR<sup>n</sup>-Tree** as the base tree, such that  $n$  is the highest resolution. The base tree ultimately grows into the STAR-Tree by systematically merging the **L<sub>r</sub>-nodes** from all the remaining **STAR<sup>r</sup>-Trees** of resolutions lower than  $n$ .
- 4) Pick the **STAR<sup>n-1</sup>-Tree** with the second highest resolution such that there exists no resolution  $m$  where  $n-1 < m < n$ . Merge all the **L<sub>n-1</sub>-nodes** into the **P<sub>n</sub>-nodes** at level higher than that of the **L<sub>n</sub>-node**. Let this level be **H<sub>n-1</sub>**. Such merging would result in **P<sub>n</sub>-node**, **P<sub>n-1</sub>-node**, **P<sub>{n,n-1}</sub><sup>m</sup>-node**, where each **P<sub>{n,n-1}</sub><sup>m</sup>-node** comprises of regions represented by images of resolution  $n$  as well as  $n-1$ . Pick the next higher resolution tree, i.e., **STAR<sup>n-2</sup>-Tree**, and insert all its **L<sub>n-2</sub> - nodes** at level **H<sub>n-2</sub>** into **P<sub>n</sub>-nodes** **P<sub>n</sub>-node**, **P<sub>n-1</sub>-node**, **P<sub>{n,n-1}</sub><sup>m</sup>-node** that are at **H<sub>n-1</sub>**.
- 5) Repeat Step 5 until the **L<sub>r</sub>-nodes** of all the remaining **STAR<sup>r</sup>-Trees** are inserted into the nodes at the appropriate level of the base tree.

As a result, **L<sub>r</sub>-nodes** with highest  $r$  serve as the lowest level of the STAR-Tree.

**Search, Insertion and Deletion operations on the STAR-Tree:** The process for search follows similar steps except that it does not involve reorganization of the tree. In order to accomplish this, we identify an important component of the tree, called **V<sub>r</sub>-Tree**.

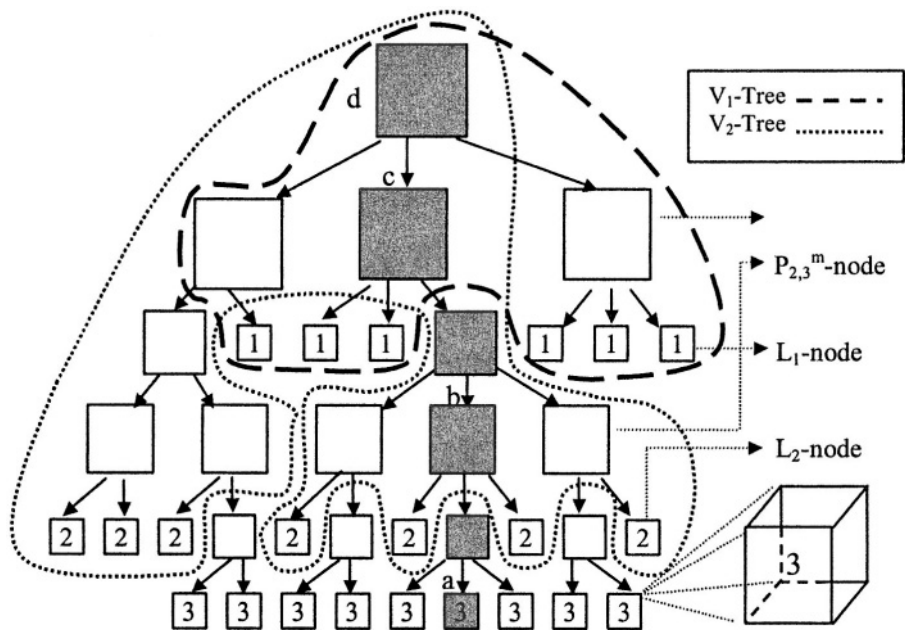


Figure 3. The V-Tree

**V<sub>r</sub>-Tree:** A V<sub>r</sub>-Tree is part of the STAR-Tree, comprised of all the L<sub>r</sub>-nodes, P<sub>r</sub>-nodes and P<sub>R</sub><sup>m</sup>-nodes where  $r \in \mathbb{R}$  for resolution  $r$ . We identify a V<sub>1</sub>-Tree and a V<sub>2</sub>-Tree in Figure 3. The STAR-Tree is no longer balanced from the view of the whole tree. Due to the fact that lower resolution L<sub>r</sub>-nodes are merged and included into the STAR-Tree, the height of each V<sub>r</sub>-Tree may be elongated in contrast to the STAR<sup>+</sup>-Tree. However, the STAR-Tree can conduct operations of both authorization evaluations and image searches simultaneously through a uniform indexing scheme. If  $N$  represents the number of images stored, and  $M$  the number of authorization rules overlaid on top of the STAR-Tree, the complexity of processing user requests is  $O(\log N)$  in the best case, and  $O(M+N)$  in the worst case.

There is another reason why the skewness of the STAR-Tree does not incur any additional cost in to perform operations such as searching, insertion and deletion of images, as well as evaluation of user requests. This is because, the STAR-Tree is a loose combination of a set of V<sub>r</sub>-Trees with different resolutions, and therefore, all operations are entirely dealt within an appropriate V<sub>r</sub>-Tree, rather than the STAR-Tree.

When inserting an image with resolution  $r$ , we first choose a candidate L<sub>r</sub>-node to place the image. We traverse the V<sub>r</sub>-Tree from the root to the L<sub>r</sub>-nodes on the leaf level with height of  $H_r$ . In each intermediate level of the V<sub>r</sub>-Tree, we pick the node with associated MBHR incurring the minimum spatiotemporal volume enlargement. Then we insert the image into the

selected **L<sub>r</sub>-node** and propagate the node split upward along the **V<sub>r</sub>-Tree**, if necessary.

The deletion operation is performed as follows: we first find the **L<sub>r</sub>-node** containing the MBR of the target image. Then we delete the image and adjust the volume of that **L<sub>r</sub>-node**, and its ancestors including the **P<sub>r</sub>-nodes** and **P<sub>R</sub><sup>m</sup>-nodes** upward along the **V<sub>r</sub>-Tree**. Merging of MBHRs when under-flow condition occurs is handled if necessary.

#### 4. OVERLAYING OF AUTHORIZATIONS ON THE STAR-TREE

In this section, we present our approach to identify the most relevant nodes in the STAR-Tree to overlay authorizations, so that it facilitates efficient search of authorizations.

Let the authorization be  $a = \langle ce, ge, pr, \tau \rangle$ . We use  $ge(a)$  to denote the set of authorized geospatial objects or geospatial extents, and  $ce(a)$  to denote the set of authorized subjects,  $pr(a)$  to denote the set of privileges in  $a$ , and  $\tau(a)$  to denote the time interval specified in  $a$ .<sup>5</sup> Since these objects are specified with the x and y extents, one can compute MBR for each object in  $ge(a)$ . Then based on the timestamp associated with each object and using the MBRs, one can construct the MBHR. Let **A<sub>r</sub>.MBHR** be the MBHR computed from all the MBRs of the objects in  $ge(a)$ . Since an authorization pertains to a set of objects pertaining to a certain resolution  $r$ , we use  $r(a)$  to denote that resolution level. Essentially **A<sub>r</sub>.MBHR** represents the MBHR covering the spatiotemporal region of the objects on which the authorization is specified.

We use *host node* to denote the node of the STAR-Tree on which an authorization is overlaid. Given an authorization  $a$ , we observe the following principles to identify the host node and overlay authorizations.

1. A node in the STAR-Tree is a candidate to be a host for  $a$  only if **A<sub>r</sub>.MBHR**  $\supset_{s,T}$  **host.MBHR**.
2.  $a$  can be overlaid on a node if it is a candidate host node and there exists no other candidate  $n$  such that **n.MBHR**  $\supset_{s,T}$  **host.MBHR**. In other words, an authorization is overlaid as high as possible on the STAR-Tree.
3. Authorization  $a$  cannot be overlaid on a host, whenever, (a)  $a.r < \min \{ \text{host} . r \}$ , if the host is a **P<sub>R</sub><sup>m</sup>-node**, and (b)  $a.r < \text{host} . r$ , if the host is a **L<sub>r</sub>-node**.

<sup>5</sup> Note that the  $ce$ ,  $ge$  and  $\tau$  are in fact expressions, which must be evaluated to obtain the set of objects, subjects and the time interval associated with the authorization. The details of these evaluation steps are outside the scope of this paper.

One may encounter the following three cases while overlaying:

- *Overlaying Authorizations (O-Auth)* is the set of authorizations such that for every  $A_r \in \text{O-Auth}$ , there exists some node  $x$  in the STAR-Tree such that  $A_r.MBHR \supset_{s,T} x.MBHR$ .
- *Overlapped Authorization (S-Auth)* is the set of authorizations such that for every  $A_r \in \text{S-Auth}$ , there exists some node  $x$  in the STAR-Tree such that  $x.MBHR \cap_{s,T} A_r.MBHR$ . Note that this includes the case where  $x.MBHR \supset_{s,T} A_r.MBHR$ . These are the set of authorizations that are associated with **L<sub>r</sub>-nodes**. Actually each **L<sub>r</sub>-node** has an affiliated S-Auth set that stores all authorizations overlapping with the **L<sub>r</sub>-node**.
- *Pending Authorizations (P-Auth)* is the set of authorizations such that every  $A_r \in \text{P-Auth}$  the following property is true. For every node  $x$  in the STAR-Tree  $x.MBHR \otimes_{s,T} A_r.MBHR$ . Essentially these are the set of authorizations not overlapping with any nodes.

We classify the overlaying operation into two major categories: *batch (static) overlaying* and *dynamic overlaying*. Batch overlaying refers to the process of overlaying of authorizations when the tree remains still. The internal structure of the whole tree is constantly updated due to insertion and deletion operations. The overlaying of authorizations performed in such a case is referred to as dynamical overlay. This requires that to adaptively adjust the positions of the overlaying authorizations on the modified tree. Dynamic overlay can be broken down into two subcategories based on whether it is an insertion or deletion operation. In reality, overlaying of both types of operations may occur randomly due to the nature of routine maintenance work to the database. As a result, insertion of an image may happen either before or after the insertion of an authorization.

**Batch Overlaying:** Batch overlaying traverses the tree in a top down manner. In each node, the  $A_r.MBHR$  of the overlaying authorization is recursively compared with the MBHR of the node. If it encloses the MBHR of the node, it will stop searching the subtree of that node and stores the authorization on that node. Otherwise, the comparison continues until the leaf level  $H_r$  of that resolution is reached.

**Dynamic Overlaying:** We introduce some preliminary terms crucial to the dynamic overlay process.

- *Dirty Path (DP)* is the set of nodes along the path chosen to either insert or delete a new object. It includes the root node, the candidate leaf node and all the propagating nodes in between. Due to the insertion of an image in the STAR-Tree, it may be required to update the position of the authorizations overlaid upon these nodes along this path. That is the reason why it is essential to identify a dirty path.

- *Split Twins* are the resulting pair of nodes when one MBHR is split due to overflow. When a node is split, some of the authorizations overlaid on the original node or its ancestors may need to be overlaid on the newly created twin node. This is the reason why one has to keep track of the split twins.
- *Degraded Authorizations (D-Auth)*: A degraded authorization is the one overlaid on the nodes of the *DP* that no longer fits in its position due to the enlarging of their hosts' MBHR during insertion.
- *Contingent Authorizations (C-Auth)*: A contingent authorization is one that is removed from the tree due to the elimination of its host node during deletion of an image(s). This will be reinserted as a new authorization into the STAR-Tree.

When a node in *DP* is split into split twins, we assume that the node itself is the parent or the ancestor of the MBR of the inserted image. It implies that 1) the parent of a node in *DP* is also in *DP*. 2) The parent of a node in *DP* will be kept unchanged all the time once it is determined even when a node could possibly split. Dynamic Overlay can be either *Insertion Overlay* or *Deletion Overlay*. Insertion Overlay checks each nodes on the *DP*, and finds out all *Degraded Authorizations* and repositions them to lower layers in a recursive manner. The fitting authorizations are assigned to the respective split twins of all splitting nodes on the *DP*. Deletion Overlay just collects all *Contingent Authorizations* (C-Auth) from the *DP* and reinsert them using the batch overlay algorithm. *Pending Authorizations* (P-Auth) are finally collected and treated as similar to C-Auth. Mix of insertion and batch overlay always follow the deletion overlay because of its operational nature.

## 5. PROCESSING OF ACCESS REQUESTS

In this section, we present our approach to processing the access requests, which is detailed in algorithm RequestEvaluation-STAR-Tree ( $N, U, r, h$ ). The algorithm traverses the  **$V_r$ -Tree** (since all access requests are processed in the  **$V_r$ -Tree**, but not in the whole STAR-Tree) from the root down to the  **$L_r$ -node**, where  $r$  is the resolution indicated by the access request. For each node, the MBHR associated with it is compared with that of the access request. If they are at least partially overlapped, the authorizations overlaid on the node are searched for potential access privilege. The result of the comparison falls into one of the three cases: 1) Evaluation process on the subtree rooted in the node halts when either the MBHR pertaining to the access request does not overlap with the associated MBHR of this node, or the tree is completely traversed, irrespective of the search results. 2) The MBHR pertaining to the access request is overlapping with that of the node.

In this case, the subject can access all images under the subtree of this node so there is no need to further search for the presence of authorizations in the subtree. However, the searching should continue till the leaf level containing the images is reached. 3) No relevant authorizations are found; therefore, more authorizations on the children of the node need to be checked for further information.

Retrieved images usually need to be post-processed to satisfy the access requests. Some portions of retrieved images need to be cropped because they are either beyond the access privilege of the user or irrelevant to the request. On the other hand, multiple images may need to be merged into one image, depending on the specifications of the user requests. The detailed discussion of them is beyond the scope of this paper.

## 6. CONCLUSIONS AND FUTURE WORK

In this paper, we have proposed a novel index called STAR-Tree, a Spatio Temporal Authorization-driven R-Tree that can uniformly index both spatiotemporal objects and the authorizations that govern access to them. When compared to the previously proposed RMX-Quadtree [18], STAR-Tree is capable of indexing the objects in 3 dimensions representing its location and timestamp. While RMX-Quadtree requires that all images are perfect squares and higher resolution images are always appear as the four quadrants of a lower resolution image, STAR-Tree does not impose no restrictions on the region covered by the geospatial objects. This resulted in relaxing several restrictive assumptions made in the RMX-Quadtree.

Although an authorization is specified with subject credentials associated with their temporal validity, STAR-Tree does not index them based on the temporal dimension of the subject credentials. Currently, we are extending the STAR-Tree, which is only 3-dimensional, to accommodate this 4<sup>th</sup> dimension. Support for negative authorizations require significant changes to the overlaying of authorizations as well as process of access request. In this paper, we do not consider negative authorizations; we will extend our work to support negative authorizations. Similar to geospatial data, moving object data as well as the authorizations governing access to them are also associated with spatiotemporal attributes and the temporal attributes; we are currently developing uniform indexing schemes for such databases. Our current work also includes conducting a performance evaluation to demonstrate that our uniform indexing scheme indeed has significant impact on the response time.



## References

- [1] John C. Baker, Kevin M. O'Connell, and Ray A. Williamson. *Commercial Observation Satellites*. Rand Edition, 2001.
- [2] Robert Wright. Private Eyes. *The New York Times Magazine*, 1999.
- [3] <http://www.spaceimaging.com>.
- [4] S. Chun and V. Atluri. Protecting Privacy from Continuous High-resolution Satellite Surveillance. In *Proc. Of the 14th IFIP WG 11.3 Working Conference on Database Security*, pages 399–420, August 2000.
- [5] V. Atluri and S. Chun, An Authorization Model for Geospatial Data, submitted for publication, 2004.
- [6] Elisa Bertino, Claudio Bettini, Elena Ferrari, and Pierangela Samarati. A temporal access control mechanism for database systems. *IEEE Transactions on Knowledge and Data Engineering*, 8(1):67–80, 1996.
- [7] Elisa Bertino, Piero Andrea Bonatti, and Elena Ferrari. TRBAC: A temporal role-based access control model. *ACM Transactions on Information and System Security (TISSEC)*, 4(3):191–233, 2001.
- [8] V. Atluri and P. Mazzoleni. A uniform indexing scheme for geo-spatial data and authorizations. In *16th IFIPWG11.3 Working Conference on Database Security*, July 2002.
- [9] N.R. Adam, V. Atluri, E. Bertino and E. Ferrari, “A Content-based Authorization Model for Digital Libraries,” *IEEE Transactions Knowledge and Data Engineering*, Volume 13, Number 4, 2002, pages 705-716.
- [10] Yannis Theodoridis, Michalis Vazirgiannis, Timos K. Sellis: Spatio-Temporal Indexing for Large Multimedia Applications. *ICMCS 1996*: 441-448.
- [11] C. H. Ang and T. C. Tan, New linear node splitting algorithm for R-trees, in M. Scholl and A. Voisard, editors, *Advances in Spatial Databases - Fifth International Symposium, SSD'97*, Berlin, July 1997, pages 339-349.
- [12] A. Guttman, R-trees: a dynamic index structure for spatial searching, *Proceedings of the SIGMOD Conference*, Boston, MA, June 1984, pages 47-57.
- [13] Simonas Saltenis, Christian S. Jensen: Indexing of now-relative spatio-bitemporal data. *VLDB J.* 11(1): 1-16(2002)
- [14] Dimitris Papadias, Yufei Tao, Panos Kalnis, and Jun Zhang, “*Indexing SpatioTemporal Data Warehouses*”. *ICDE2002*.