

Protecting Privacy from Continuous High-resolution Satellite Surveillance

Soon Ae Chun and Vijayalakshmi Atluri

MSIS Department, Rutgers University and
Center for Information Management, Integration and Connectivity
180 University Avenue, Newark NJ 07102
`soon@cimic.rutgers.edu,atluri@andromeda.rutgers.edu`

Abstract

Privacy refers to controlling the dissemination and use of personal data, including information that is knowingly disclosed, as well as data that are unintentionally revealed as a byproduct of the use of information technologies. This paper argues that the high resolution geospatial images of our earth's surface, produced from the earth observing satellites, can make a person visually *exposed*, resulting in a technological invasion of personal privacy. We propose a suitable *authorization model for geospatial data* (GSAM) where controlled access can be specified based on the region covered by an image with privilege modes that include `view`, `zoom-in`, `overlay` and `identify`. We demonstrate how access control can be efficiently enforced using a spatial indexing structure, called MX-RS quadtree, a variant of the MX-CIF quadtree.

1 Introduction

In the year 2000 and the new millennium, there will be 31 satellites capable of providing land cover data at resolutions of 1 to 30 meters in orbit; 14 of these will be privately funded by US corporations, all with resolutions of 10 meters or better. As low-cost, highly responsive commercial satellite systems become operational, high resolution imagery is expected to become a regular input to consumer products and information services.* Remote sensing data sales and services are predicted to grow into a \$2 billion dollar market by the beginning of the 21st century [1].

There are numerous benefits to society in the constructive use of low cost satellite imagery. Examples include environmental monitoring, map making, disaster relief, infrastructure planning, national security, pin-pointing of prospective sites to aid miners and drillers in planning access to natural resources, and detecting distressed crops early before such stress is visible to the human eye. Up-to-date satellite images can assist businesses in planning the placement of consumer outlets and manufacturing facilities, and help demographic analysts locate their target markets. Images can be used to aid police and fire

*In fact, the Ikonos 1 satellite system, launched on September 24, 1999, is capable of providing 1 meter resolution panchromatic images at 2 meter horizontal positional accuracy, provides a combination of very high resolution, high accuracy and high frequency revisit capability. <http://www.ssc.se/rst/rss/satinfo/ikonos.html>

crews to respond more quickly to distress calls, and to direct vehicle flows depending on observed traffic situations.

Motivation: While high resolution low cost satellite imagery enjoy many benefits, there are significant threats to privacy due to the commercial availability of high-resolution imagery in near real-time fashion. Public entities, such as local governments or public utility companies, collect, use and disseminate large amounts of personal information. Combination of this publicly available personal data pools with high resolution image data coupled with the integration and analysis capabilities of modern GIS systems providing geographic keys such as longitude and latitude, can result in a technological invasion of personal privacy. A person can, not only be identified by name or address, but can be *visually exposed*. Therefore, in the near future, it may be technically feasible for anyone to observe, record and measure the outdoor activities of anyone, at any place in the world (from backyard pools to nuclear plants), almost at any time. For example, one can clearly identify the objects in the high-resolution image shown in figure 1. Many scenarios can be envisioned that may threaten the privacy of individuals or organizations; some are listed below.

- Observation of military operations or movements of agents of foreign countries can be achieved by the click of a mouse [18].
- Unauthorized surveillance of a person's outdoor activities by a stalker or a burglar may help planning a break-in of a home. Tracking of residents entering and leaving the house through observing high resolution images over a period of time can simply be done on his computer.
- Tracking of the shipping volumes and patterns of a company by observing the number of trucks being loaded and unloaded can be valuable for a competing business enterprise.

These are some scenarios that depict the need for access control for high resolution geospatial image data. Although there are no policies or laws in place yet, they appear to be inevitable [18].

Aside from protecting privacy of individuals from near real-time high-resolution satellite surveillance, the need for controlled access to images arises because of different reasons:

- Concept based filtering: Filtering of images is needed, for example, to prevent children from accessing objectionable images available on the web. While traditionally access control is provided at the server, filtering requires access control at the client.
- Controlled access to images: Prevention of unauthorized access may be needed for providing controlled distribution of images to subscribers.
- Content based access control: Prevention of access may be needed for certain images based on their content, for example, to prevent public from accessing images of all vehicles with a color distribution used by the military.

Related Work: While there exist no work on providing access control for geospatial images, recently, a number efforts have been made to screen objectionable images using shape

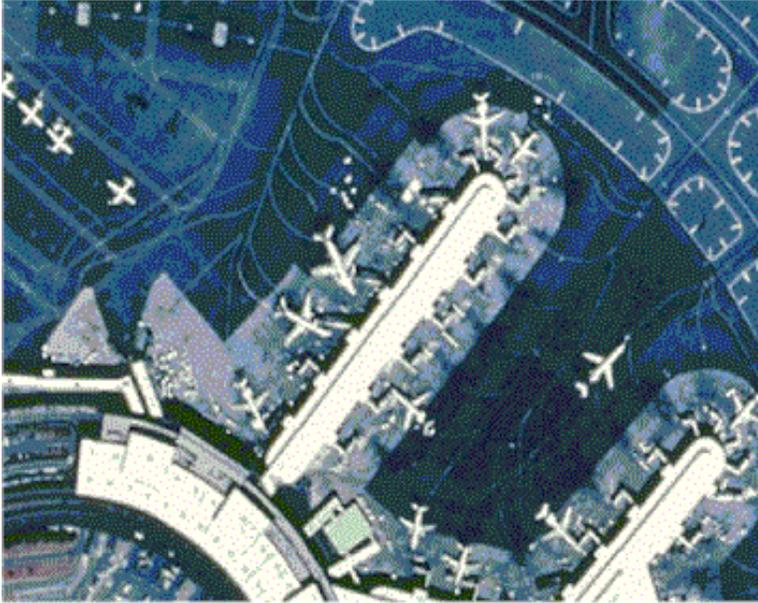


Figure 1: A high resolution image

detection, object recognition, people recognition, face recognition, and content-based image retrieval. They include (1) filtering of images of naked people using a skin filter and a human figure grouper [3, 4], and (2) using a content-based feature vector indexing where an image is matched against a small number of feature vectors obtained from a training database [16, 17]. However, these approaches filter all images that match a set of criteria, but do not provide controlled access that facilitates access to images for legitimate users.

Our Contribution: A suitable access control for protecting privacy due to unauthorized high-resolution surveillance should not only be based on the spatial extent of images but also be based on their resolution. This is because, while a low resolution image may be revealed to the user regardless of its location coordinates, a high resolution image may not be allowed access, except in the region where the user has access permission. For example, a factory owner may access every detail pertaining to his own operations, but should be prohibited from accessing the images that reveal the details of his competitor’s operations. To the best of our knowledge, there does not exist any authorization model suitable for geospatial images. In this paper, we propose an authorization model that can provide access control for geospatial images based on their spatial extent and resolution, called *Geo-Spatial Authorization Model* (GSAM). Our access control model will use publicly available user information, such as property ownership and voter registration records to determine the spatial extent that the user is allowed to access, which in turn is used to determine the appropriate image(s), or a portion of it, from the image database.

To accomplish this, GSAM supports, in addition to the conventional privilege modes

such as `read`, `insert`, `delete` and `modify`, privilege modes such as `view`, `zoom-in`, `overlay` and `identify` that can be defined based on the allowed resolution level for a given user/subject. To provide efficient processing of queries, images in a spatial database are typically stored using a spatial indexing structure. We have realized that an appropriate spatial indexing structure be based on image resolution to provide efficient processing of image operations, which will in turn effect the performance of evaluating an authorization in GSAM. Since there does not exist a suitable indexing structure, we first propose an enhancement to the spatial access method MX-CIF quadtree [13, 14], called MX-RS quadtree. Since MX-RS quadtree is capable of representing higher resolution images at lower levels of the tree, it facilitates efficient enforcement of privilege modes including the `zoom-in`, `view`, `overlay`, and `identify`, simply by controlling the traversal of the tree down towards higher resolution images.

We provide access control in two ways. (1) We *control the depth* a user can traverse, thereby controlling the resolution of the images (s)he can access. For example, anyone can access a low resolution image such as the New Jersey state map, but access to a 1 meter resolution image of an individual’s house is prohibited as it may infringe on the privacy of that individual. (2) We *control the extent* a user can view. That is, a user is given access to high resolution images (say 1 meter), only for certain regions (typically the property (s)he owns, public parks, etc.) but not to all.

Organization of the paper: This paper is organized as follows. Section 2 provides a brief introduction to satellite and other geospatial image data as well as image databases. While Section 3 presents GSAM, Section 4 shows how access control can be accomplished. Section 5 proposes an efficient access control approach that allows implementing it on top of the geospatial indexing structure. Finally Section 6 presents conclusions as well as future research we intend to pursue in this area.

2 Background on Geospatial Images

Geospatial images 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 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.

Since the main focus of this paper concerns protecting privacy from high-resolution satellite surveillance, we provide more details on satellite imagery. Satellite images are a product of Remote Sensing. Remote sensing is a technology for sampling radiation and force fields to acquire and interpret geospatial data. Geospatial data are used to develop information about features, objects, and classes on Earth’s land surface, oceans, and atmosphere. Remote sensing of the Earth traditionally has used reflected energy in the visible and infrared and emitted energy in the thermal infrared and microwave regions. It gathers radiation that can be analyzed numerically or used to generate images whose variations represent different intensities of photons associated with a range of wavelengths that are received at the sensor. Satellite images are pictorial representation of target objects and

features in different spectral regions. Each of different sensor (commonly with bandpass filters) is tuned to accept and process the wave frequencies (wavelengths) that characterize each region. Each region normally shows significant differences in the distribution (patterns) of color or gray tones. A chief use of satellite image data has been in classifying different features in a scene into meaningful categories or classes. The image then becomes a thematic map (the theme is selectable, e.g., land use; geology; vegetation types; rainfall).

Satellite data have the following characteristics:

- The satellite's orbital information is changing; hence it is hard to obtain images whose spatial coverages are exactly the same.
- There are variabilities of images coming from different satellites and sensors, even if they observe the same region. Typically different sensors capture different characteristics of earth surface, e.g. land coverage and weather.
- Different sensors provide images of different resolution levels, from low to high. For example, the Advanced Very High Resolution Radiometer (AVHRR) is a broad-band, four or five channel (depending on the model) scanner, sensing the visible (red, green, blue), near-infrared, and thermal infrared portions of the electro-magnetic spectrum. It produces 1km resolution images. Landsat Thematic Mapper (TM) provides multi-spectral imagery at 25m ground resolution. Radar sensors can transmit 5 to 10 meter resolution images. Sensors from the IKONOS satellite launched by Space Imaging/EOSAT promises to provide 1m Panchromatic and 4m Multispectral (blue, green, red, near-IR) data.
- For any remotely sensed image, there is a trade-off between spatial resolution, area of extent, and data volume. If the data volume is to be held constant, a high-resolution image will cover a small area, while a low-resolution image will cover a large area. The systems intended for the identification of land cover and land use have focused on moderate resolutions between 5 and 30 meters and swaths of 100 to 200 kilometers, while the high resolution satellites are designed with 1 to 3 meters resolution and 4 to 40 kilometer swaths.
- Each satellite image undergoes the process of georectification which involves two steps: georegistration and geocorrection. Geocorrection of the image is needed since the distances and directions in satellite images do not correspond to true distances and directions on the ground due to the variability of satellite position. Georegistration process registers each image with a known coordinate system (e.g. longitude, latitude), and reference units (e.g. degrees) and coordinates of left, right, top and bottom edges of the image.

In addition to these characteristics specific to remotely sensed data, satellite images also share characteristics of spatial data such as maps, since they cover spatial areas of the earth surface. In order to provide efficient retrieval of geospatial images based on spatial locations, and to support approximate matching (similarity based retrieval), range and nearest neighbor queries, various *spatial access methods* (SAMs) have been developed. They include approaches that transform rectangles into points in a space of higher dimensionality,

Id	Area	Town	Acre	Landuse	Owner	Address	State	Land_Val
1	1216235.75	Moonachie	27.92000	vacant	RALPH ZISA	18 GARFIELD ST	NJ	65200
2	7474.30	Carlstadt	0.17000	residential	WILLIAM LUBERTO	18 RIVERSIDE AVE	NJ	129000
3	3077350.50	Teterboro	70.65000	commercial	JOAN VALENTINE	120 EGG HARBOR RD	NJ	60000
4	44063.28	Little Ferry	1.01000	residential	HANNA Y. KARROUM	12 SCENIC VISTA DR	NY	65600
5	14740839.00	Teterboro	338.40000	office	ANTONIO TORRES	33 WASHINGTON AVE	NJ	60000
6	363254.62	Ridgefield	8.34000	warehouse	LAWRENCE P. MEDICH	37 WASHINGTON AVE	NJ	76000

Table 1: Tabular Data Linked to vector objects

e.g. grid files [8]; approaches that use linear quadtrees [15, 5], z-ordering [11] or other space filling curves [9]; and approaches based on trees (R-tree [7], MX quadtrees [13, 14], k-d-trees [2], k-d-B trees [12], hB trees [10] and cell trees [6]). In this paper, we propose extension to the MX-CIF quadtree, called MX-RS quadtree, that enables efficient implementation of various privilege modes of GSAM.

3 Authorization Model for Geospatial Data (GSAM)

In this section, we formally present GSAM, an authorization model suitable for providing controlled access to geospatial data. Let $S = \{s_1, s_2 \dots\}$ denote a set of subjects, $O = \{o_1, o_2 \dots\}$ a set of objects, and $M = \{view, zoom-in \dots\}$ a finite set of privilege modes. In the following, we describe in detail the image objects and privilege modes, and present the formalism for authorization specification.

3.1 Image Objects

Image objects can either be raster or vector images. Vector objects describe geographic map features such as roads, parcels, soil units, or forest stands. It can contain several feature classes, such as arc, node, polygon, label point, annotation, tic, and coverage extent.

Each raster image object O_i is represented as a tuple, $\langle id, l, g, h, w, r, t \rangle$, where id is a unique identifier and l, g, h , and w are *latitude, longitude, height, width*, respectively, that represent the spatial extent of the image. r is for *resolution* of O_i , while t represents the *download timestamp*.

There is a set of access functions associated with each object. Given an image object, O_i , the function $rectangle(id)$ would retrieve the rectangular region (l, g, h, w) of the object. Similarly $resolution(id)$ would return r , and $tabular(id)$ would return l .

Each vector object, O_v , is represented as a tuple, $\langle id, l, g, h, w, t, k \rangle$, where id is a unique identifier and l, g, h , and w are *latitude, longitude, height, width*, respectively, that represent the spatial extent of the vector file. t represents the *last update timestamp*. k represents a *link* that links tabular data of geographic features contained in the vector object, O_v . Table 1 shows an example of tabular data records that can be linked to vector objects.

3.2 Privilege Modes

In our model, we support two types of privilege modes – *viewing* and *maintenance*. The viewing modes include *view*, *zoom-in*, *overlay*, and *identify*, and the maintenance

modes are `insert`, `delete` and `update`. The `view` privilege allows a user to see an image object covering a certain geographic area within a permitted resolution level.

The `zoom-in` privilege allows a user to view an image covering a certain geographic area at a higher resolution. Unlike conventional privilege modes that allow or deny access, this privilege specifies the level of zoom-in allowed, and is therefore expressed with an associated value, called `zoom level` (for example, `zoom-in:10`). The access control algorithm interprets this value and determines the level of resolution of the image that is allowed to be viewed by the user. Note that given an image, zooming-in can also be achieved using zoom-in algorithms, but the quality of the image decreases so that the result becomes useless, if zooming is done beyond a certain level. (For example, Figure 3 shows an image after the zoom-in operation to the rectangular region shown in Figure 2.) Thus the level of zoom-in a user is allowed should be determined based on the level (s)he can attain after applying the `zoom-in` algorithm. That is, if a user is allowed a zoom-in level of l_z , the access control algorithm must make sure that the user is given an image with a resolution of at most r that can not be zoomed-in to a resolution higher than l_z without losing its content. The functionality of providing the desired level of zoom-in is achieved by storing multiple images with different levels of resolution. Thus, if a user is allowed to access a region at a certain level of resolution, zooming-in is accomplished by retrieving a higher resolution image.



Figure 2: Landsat Image with 28.5 meter resolution

The `overlay` privilege allows users to generate composite images, where a composite image is constructed by *overlaying* one image on top of another. Although each individual



Figure 3: Landsat Image with zoom-in level of 10

image in isolation can be viewed by a user, sometimes an overlaid image may reveal more information than the user is allowed to access. For example, Figure 4 shows a 1m resolution aerial orthophoto image overlaid with a zoning vector coverage, represented in white lines. While a user can view both images individually, overlaying the street map on a high resolution image may help pin-pointing a person's private property and viewing it in realtime.

The **identify** privilege allows the user to view the tabular data linked to an image. As can be seen from Table 1, some of the data linked to the image, for example the ownership information, when shown with a high resolution image may provide visual exposure of a person's private property.

While the **insert** privilege allows a user to insert an image objects into the database, the **delete** allows her to remove images. The **update** privilege allows a user to replace one image with another as well as modify the attributes of the image, such as latitude, longitude, resolution, and link. In addition, it allows the user to update the tabular data linked to the image.

3.3 Authorization

An authorization in GSAM is specified as follows:



Figure 4: Orthophoto image overlaid with a zoning vector map

Definition 1 An authorization a is a triple $\langle sub, obj, pr \rangle$, where sub is a subject $s \in S$,
 obj is (i) an object id of an object $o \in O$,
(ii) a region represented as a rectangle with (latitude, longitude, height, width), or
(iii) a set of object ids , and
 pr is (i) a single privilege mode $m \in M$ or
(ii) a set of privilege modes $\{m_1, m_2, \dots\} \subseteq M$.

An object in our authorization specification can be a single image, a set of images, or a region. Although the region could be any polygon, for the sake of simplicity, in this paper, we limit it to represent only rectangles.

The privilege pr in an authorization triple may be composite, that is, may contain more than one privilege mode, which is especially useful when used with `overlay`. That is the case because, a subject may be allowed to overlay an image over another low resolution image, but not over a high resolution image. In order to specify such access control policies, we need a combination of both `zoom-in` and `overlay`.

In our model, as can be seen from the above definition, authorizations will allow one to specify that a subject is allowed to view a specific image or region with a specific resolution, or allowed to overlay a set of images with a specific resolution.

Following are some examples of authorizations.

$$a_1 = \langle John, (50, 60, 10, 10), (zoom - in : 8) \rangle$$

$$a_2 = \langle Mary, 123, view \rangle$$

$$a_3 = \langle Ann, \{123, 456\}, overlay \rangle$$

$$a_4 = \langle Tom, \{123, 456\}, (overlay, *, 8) \rangle$$

Above authorizations can be interpreted as follows: a_1 specifies that John is allowed to access a region centered at point (50,60) with width and height of 10, with a zoom-in level of 8. a_2 specifies that Mary can view the object with the object id 123. a_3 specifies that Ann is allowed to overlay objects 123 and 456. Finally, a_4 specifies that Tom is allowed to overlay images 123 and 456 where the highest resolution level of object 456 is 8.

We use $a(sub)$, $a(obj)$ and $a(pr)$ to denote the subject, object and privilege of a , respectively. Moreover, to denote the attributes of each component in a , we use the notation $component_{attribute}$. For example, $a(pr_{zoomin})$ represents the zoom-in level specified in the privilege mode of a . We denote the set of all authorizations as *geo-spatial authorization base*, *GSAB*.

4 Access Control

When a subject requests to access images covering a specific geographic region at a specific resolution level, the access control mechanism must evaluate whether such request can be granted. We define the Access Request by a user, ur , as follows:

Definition 2 [Access Request] An *access request* is a triple $ur = \langle s, o, pr \rangle$, where s is the subject, pr is the privilege mode, and o is the object which can be either of the following two: (i) a tuple (l, g, h, w, r) where (l, g, h, w) represents the requested rectangle represented with latitude, longitude, height and width, and r represents the level of resolution, and (ii) a set of object *ids*.

According to the above definition, a user may request to access an object by specifying its object id, or may request to access a rectangular region by specifying its latitude, longitude, height and width. We use $ur(s)$, $ur(o)$ and $ur(pr)$ to denote the subject, object and privilege mode specified in ur , respectively.

When a subject requests to access images covering a specific geographic region at a certain resolution level, the access control module (refer to figure 5) verifies whether there exists an authorization such that the object region specified in the authorization *overlaps* with (or contains) the requested object area. As a first step, it determines all the authorizations relevant to the access request. Access is denied if no relevant authorization exists. Then the authorization evaluation module determines either a set of object ids or a rectangular region that is allowed to be viewed by the subject and sends a request to the image database.

Since the region allowed to be viewed by the subject may not match exactly with the image(s) returned, the images returned from the image database need to be edited, namely assembled and/or cropped. This function is performed by the image processing module.

Given an authorization base *GSAB*, the following algorithm describes how an access request ur with view, zoom-in and overlay modes can be evaluated.

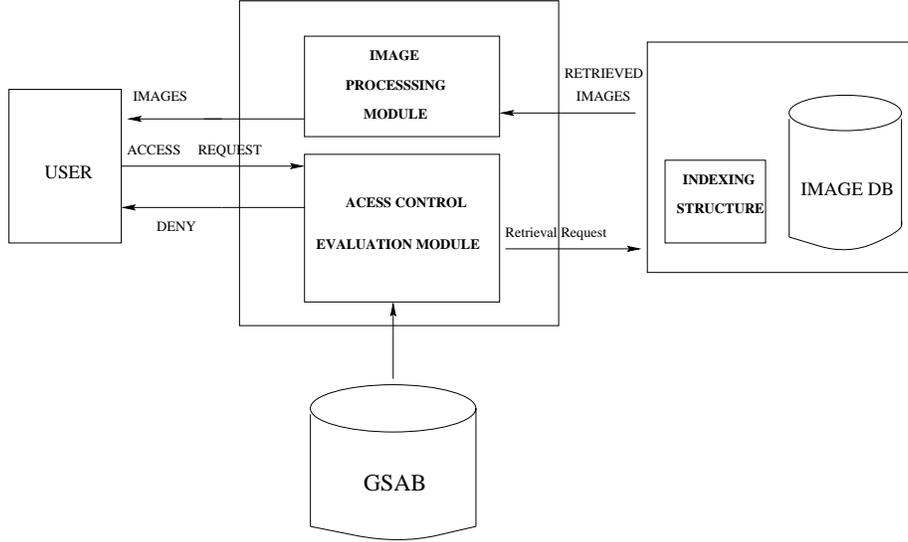


Figure 5: The System Architecture

Algorithm 1 [Authorization Evaluation]

input: ur

output: set of images

begin

1. Find the set of authorizations $A(ur)$ in $GSAB$ such that
 - foreach** $a \in A(ur)$
 - $(a(s) = ur(s)) \wedge (a(pr) = ur(pr))$
2. **if** $A(ur) = \emptyset$
 - then** return (“Access denied”)
 - else**{
 - while** $A(ur) \neq \emptyset$ {
 - foreach** $a \in A(ur)$
 - case** $ur(pr) = \text{‘view’}$: {
 - if** $((a(o) \text{ is } id) \wedge (ur(o) \text{ is } id))$
 - then**{**if** $(a(o) = ur(o))$
 - then** RETRIEVE-IMAGE-WITH-ID FROM ImageDB
 - WHERE $imageid = a(o)$ }
 - if** $((a(o) \text{ is } id) \wedge (ur(o) \text{ is not } id))$
 - then**{**if** $(\text{overlap}(\text{rectangle}(a(o)), \text{rectangle}(ur(o))) \neq \emptyset)$

```

    then RETRIEVE-IMAGE-WITH-ID FROM ImageDB
      WHERE  $imageid = a(o)$  }
  if (( $a(o)$  is not  $id$ )  $\wedge$  ( $ur(o)$  is not  $id$ ))
  then {
     $area = overlap(rectangle(a(o)), rectangle(ur(o)))$ 
    RETRIEVE-IMAGES-WITH-AREA FROM ImageDB
    WHERE  $overlap(area(image), area) \neq \emptyset \wedge$ 
       $resolution(image) \geq resolution(a(o))$ 
    PROCESS-IMAGES (area, images) }
   $A(ur) = A(ur) - a$ 
}
case  $ur(pr) = 'zoom-in'$ : {
   $resolution(ur(o)) = ur(pr_{zoom-in})$ 
  if (( $a(o)$  is  $id$ )  $\wedge$  ( $ur(o)$  is  $id$ ))
  then { if ( $a(o) = ur(o)$ )
    then RETRIEVE-IMAGE-WITH-ID FROM ImageDB
      WHERE  $imageid = a(o)$  }
  if (( $a(o)$  is  $id$ )  $\wedge$  ( $ur(o)$  is not  $id$ ))
  then { if (( $resolution(a(o)) \leq resolution(ur(o))$ )  $\wedge$ 
     $overlap(rectangle(a(o)), rectangle(ur(o))) \neq \emptyset$ )
    then RETRIEVE-IMAGE-WITH-ID FROM ImageDB
      WHERE  $imageid = a(o)$  }
  if (( $a(o)$  is not  $id$ )  $\wedge$  ( $ur(o)$  is not  $id$ ))
  then { if ( $resolution(a(o)) \leq resolution(ur(o))$ )
    then {  $area = overlap(rectangle(a(o)), rectangle(ur(o)))$ 
    RETRIEVE-IMAGES-WITH-AREA FROM ImageDB
    WHERE ( $overlap(rectangle(image), area) \neq \emptyset$ )  $\wedge$ 
      ( $resolution(image) = resolution(ur(o))$ ) }
    PROCESS-IMAGES (area, images) }
     $A(ur) = A(ur) - a$ 
  }
}
case  $ur(pr) = 'overlay'$ : {
  if ((( $a(o_i)$  is  $id$ )  $\wedge$  ( $ur(o_i)$  is  $id$ ))  $\wedge$  (( $a(o_j)$  is  $id$ )  $\wedge$  ( $ur(o_j)$  is  $id$ )))
  then { RETRIEVE-IMAGE-WITH-ID FROM ImageDB
    WHERE  $imageid = a(o_i) \cup imageid = a(o_j)$  }
  if (( $a(o_i)$  is  $id$ )  $\wedge$  ( $a(o_j)$  is  $id$ ))  $\wedge$  (( $ur(o_i)$  is not  $id$ )  $\wedge$  ( $ur(o_j)$  is not  $id$ )))
  then {
    if ( $overlap(rectangle(a(o_i)), rectangle(ur(o_i))) \neq \emptyset$ )
    then { RETRIEVE-IMAGE-WITH-ID FROM ImageDB
      WHERE  $imageid = a(o_i) \wedge resolution(image) \geq resolution(a(o_i))$  }
    if ( $overlap(rectangle(a(o_j)), rectangle(ur(o_j))) \neq \emptyset$ )
    then { RETRIEVE-IMAGE-WITH-ID FROM ImageDB
      WHERE  $imageid = a(o_j) \wedge resolution(image) \geq resolution(a(o_j))$  }
     $area = overlap(rectangle(ur(o_i)), rectangle(ur(o_j)))$ 
    PROCESS-IMAGES (area, images) }
  if (( $a(o_i)$  is not  $id$ )  $\wedge$  ( $a(o_j)$  is not  $id$ ))  $\wedge$  (( $ur(o_i)$  is not  $id$ )  $\wedge$  ( $ur(o_j)$  is not  $id$ )))
  then {  $R_i = overlap(rectangle(ur(o_i)), rectangle(a(o_i)))$ 
     $R_j = overlap(rectangle(ur(o_j)), rectangle(a(o_j)))$ 
    if ( $overlap(R_i, R_j) \neq \emptyset$ )
    then {
      RETRIEVE-IMAGES-WITH-AREA from IMAGEDB
    }
  }
}

```

```

WHERE  $overlap(rectangle(image), R_i) \neq \emptyset \wedge$ 
       $resolution(image) \geq resolution(a(o_i))$ 
       $resolution(image) \leq resolution(ur(o_i))$ 
RETRIEVE-IMAGES-WITH-AREA from IMAGEDB
WHERE  $overlap(rectangle(image), R_j) \neq \emptyset \wedge$ 
       $resolution(image) \geq resolution(a(o_j))$ 
       $resolution(image) \leq resolution(ur(o_j))$ 
PROCESS-IMAGES ( $overlap(R_i, R_j)$ , images) }}
A(ur) = A(ur) - a }
}
}
end

```

Procedure PROCESS-IMAGES

input: area, retrieved-images

output: images covering only area

begin

foreach image $i \in images$

 chop (area, i)

for each imageset $I \in$ same resolution level {

 images = assemble-area (area, I)

 return(images) }

end

This algorithm considers three cases for evaluating each privilege mode. In the first case, both the access request and authorization are specified with image ids. In this case, evaluation of access request is to simply test whether the ids are the same. In the second case, the access request is specified as a rectangular region, but the authorization is specified with an image id. In this case, evaluation involves determining the overlapping region of the image specified in the authorization with the requested region. If the overlapping region is empty, access is denied. Otherwise, appropriate request is sent to the image database to retrieve the image. The case where authorization is specified with a region and the access request is specified as an id can be dealt with in a similar manner. Therefore, this is not included in the algorithm. In the third case, both the access request and the authorization are specified as rectangular regions. In this case, the overlapped region must be determined first. The area is then used to retrieve the relevant images.

Further processing is done by the procedure PROCESS-IMAGES if the area covered by the retrieved images does not coincide with the region authorized to be viewed by the subject. In this case the image is cropped. If more than one image are retrieved, they are first assembled together before cropping.

5 Efficient Evaluation of Access Requests

Enforcement of access control for different types of privilege modes is closely dependent on how images are physically organized. Since neither the area covered by the access request nor the area specified in the authorization may correspond to a single image, one needs to

understand how the image objects are physically organized. Moreover, the data structure impacts greatly on the efficient enforcement of authorizations, especially for access control with privilege modes such as `zoom-in` and `view` at a certain level of resolution. In other words, referring to Figure 5, we propose to implement the access control module on top of the index structure.

To provide efficient search and retrieval of images, the image database must be organized by employing a spatial access method (SAM). Although there exist many SAMs in the literature, none of them is completely suitable for representing image data with different resolution levels that allows efficient implementation of `view`, `zoom-in`, etc, operations. While we do not develop such an indexing structure in this paper (will be reported in another paper), we identify the required features of such a structure.

1. The index structure must be able to represent regions.
2. In addition to the leaf nodes, nodes at the root as well as at the intermediate levels of the tree should be able to store images.
3. The index structure must allow representation of overlapping regions.
4. Nodes at the same level contain images at the same level of resolution.

The intention of the tree structure is that images be organized in such a way that those covering larger geographical regions are placed at higher levels of the tree. That is, satellite images with a larger swath are stored at higher levels of the tree. Since it is typical that the larger the swath, the lower the resolution, this way of organization results in placing low resolution images at higher levels of the tree. In other words, during a traversal down the tree, higher resolution images are encountered.

For the purposes of this paper, it is not important to detail the actual tree structure but it is necessary to demonstrate how access control can be enforced. Therefore, we consider a simpler tree structure that does not meet all the requirements enumerated above. More specifically, we relax requirement 2 above and assume that all the images in the database represent non-overlapping rectangles. The proposed SAM is a variant of the MX-quadtrees, called MX-RS quadtree. In the following, we will demonstrate how `view`, `zoom-in` and `overlay` privilege modes can be enforced when the image database is indexed using the MX-RS quadtree. Although MX-RS quadtree does not permit images with overlapping regions, the user may request a region (rectangles[†]) that overlaps several stored images.

5.1 MX-RS Quadtree

An MX-RS quadtree is a variant of an MX quadtree, representing region data in two dimensional spaces. As in MX quadtrees, an area represented in the region is split into cells of size $(2^k \times 2^k)$ for some fixed k . Since there is a limited number of resolution levels in satellite images, we can set k to correspond to the number of resolution levels available. Therefore, *level* k implicitly contains the resolution level of an image. For instance, if there are 5 resolution levels (1m, 10m, 20m, 75m, and 1km), and images at each resolution level

[†]We assume that the requested regions are rectangles, although in general they may be any polygon.

cover a geographical area of fixed size (i.e. spatial extent of an image), then the size implies the resolution information and vice versa.

Each cell is organized hierarchically according to geographic areas. The entire region (the largest cell with width and height $2^k \times 2^k$) is represented by the root node. The region covered by the root is split into four equal size quadrants represented by four child nodes (NE, NW, SE, SW). This process is repeated recursively for all children. For example, region A in Figure 6 is split into 4 equal size regions B, C, D, and E. Similarly, region E is split into F, G, H, and I, and I into J, K, L and M. The dotted rectangle in Figure 6 may be ignored for the moment.

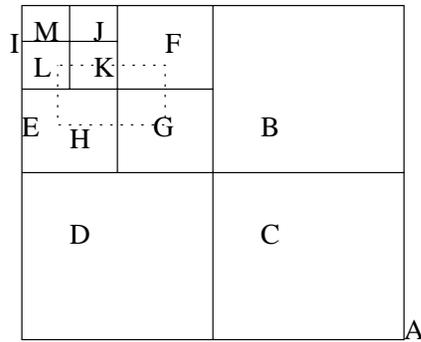


Figure 6: An example showing different regions

Since all splits are done in the middle, the regions represented by the four children of a node N can be calculated as shown in Table 2 below. The width, w , of the region represented by node N is given by $w=N.XUB-N.XLB$, where XUB (YUB) and XLB (YLB) represent the lower and upper bounds in the horizontal (vertical) direction, respectively.

child	XLB	XUB	YLB	YUB
NW	N.XLB	N.XLB+w/2	N.YLB+w/2	N.YLB+w
SW	N.XLB	N.XLB+w/2	N.YLB	N.YLB+w/2
NE	N.XLB+w/2	N.XLB+w	N.YLB+w/2	N.YLB+w
SE	N.XLB+w/2	N.XLB+w	N.YLB	N.YLB+w

Table 2: Computing rectangle regions of children nodes

Each node in the MX-RS tree has the following structure:

```
node = record
    XLB, YLB, XUB, YUB : real;
    NW, SW, NE, SE : *node
    image-list: *image
end
```

Each node in an MX-RS quadtree has a field `image-list` which holds a pointer to image list. This distinguishes an MX-RS quadtree from an MX quadtree in which a non-leaf node does not hold a pointer to any image. The images associated with the same node in the

MX-RS quadtree cover the same spatial extent (i.e., region), thus implying that they have the same resolution. The images in the same level of the tree can be represented as a linked list. Each image has the following record structure:

```

image = record
  id: integer
  latitude, longitude, width, height: real
  resolution: real
  link: *tabular
  next: *image
end

```

The `link` field is a pointer to tabular data attached to the image.

When using the MX-RS tree to represent geospatial images, we assume the region covered by each satellite image corresponds to the region covered by a cell. This can be realized by pre-processing images to correspond to the areas represented by appropriate cells. Since the lower resolution satellite images have wider swath than those with higher resolution, a larger grid cell can represent a lower resolution image, and a smaller cell can represent a higher resolution image. In other words, a parent node corresponds to a region covered by lower resolution images and each of its child node corresponds to a region covered by a higher resolution image. The latitude and longitude of an image represent the x and y coordinates of the center point of the region represented by a node. Figure 7 depicts the MX-RS tree of the different regions shown in Figure 6. The images covering the region represented by each node are also shown in the figure. For example, there exist three images `i1`, `i2` and `i3` covering region A, etc.

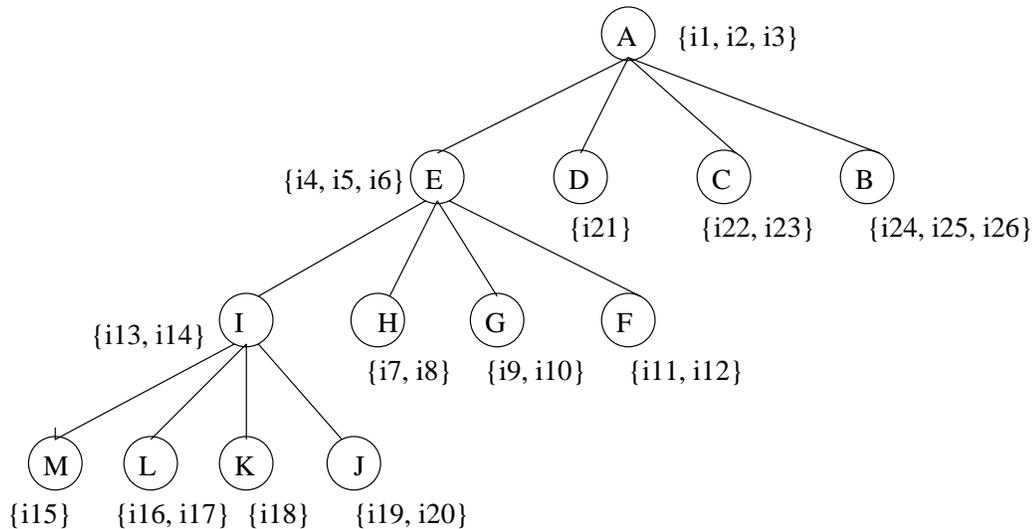


Figure 7: MX-RS tree of the regions in figure 4

5.2 Authorization Evaluation

In section, we illustrate how access control can be enforced when the image database uses the MX-RS quadtree for indexing. Specifically, algorithms for enforcing view, zoom-in, overlay and identify privileges are described.

The following algorithm illustrates how each access control can be implemented.

```
Procedure authorize (ur)
/* retrieves the authorization from database
 * see if it matches with the user request
 * request to view a region will return a set of images allowed which
 * contains the region; if no images are returned, then access is denied */
begin
  find A, the set of all a in GSAB
    such that a(sub) = ur(s) and a(pr)=ur(pr);
  if ur(pr) not equal to a(pr), for any a in A,
    then return ("Access Not allowed")
  else switch (ur(pr)) {
    case "view":      images = authorize_viewing(ur,A)
    case "zoom-in" :  images = authorize_viewing(ur,A)
    case "overlay" :  (images1, images2) = authorize_overlay (ur,A)
  }
end;

Procedure authorize_viewing (ur,A)
/* traverse index tree to retrieve images containing ur(o), the rectangle
 * whose resolution levels are authorized to view or zoom-in. */
begin
  list = NULL;
  T = root of the MX-RS tree;
  h = min(a(r)) for all a(r) in A; /* highest resolution allowed to view */
  list = retrieve_images(h, ur, A);
  if (list==NULL)
    then return ("Access is denied")
    else return(list);
end

Procedure retrieve_images (h, ur, A)
begin
  if (ur(pr) == 'zoom-in')
    then z = true
    else z = false
  while (h <= T(r)) and (T<>NULL) do
    begin
      if (overlap(ur(o), T))
        then {
          if ((z) /* if zoom-in, only retrieve on the zoom-level */
            then if (ur(pr_zoom) == T(r))
              then {
                foreach image in T(image_list)
                  if image is in a(o)
                    then list = append(list,image)
              }
        }
    }
end
```

```

    }
    else {
        foreach image in T(image_list)
            if image is in a(o)
                then list = append(list,image)
        }
        foreach T.child in (T.NE, T.NW, T.SW, T.SE){
            if overlap(ur(o),T.child)
                then T = T.child
            }
        } /* if overlap */
    end
    return(list);
end

```

```

Procedure authorize_overlay (ur: request, a: authorization)
/* Check if o1, o2 has permitted resolution level in authorization.
* If so, fetch images containing o1,o2 in the index and
* return imagelist1 and imagelist2 where any image in imagelist1
* can be overlaid with any image in imagelist2 */
begin
    list = NULL;
    T = root of the MX-RS tree;

    foreach a in A do:
        foreach o1, o2 in ur(o) and obj1, obj2 in a(o)
            if ((o1(r) < obj1(r)) OR (o2(r) < obj2(r)))
                then return ("Overlay is not allowed")
            else {
                while (o1(r) >= T(r)) AND (T <> NULL) do
                    begin
                        if overlap(rectangle(o1), T) and (o1(r) == T(r))
                            then image1 = image1 + (T(image_list) in obj1(o))
                        else if ((o1(r) < obj1(r))
                            then return("Not permitted overlay resolution)
                            else T = T.child
                    end
                while (o2(r) >= T(r)) AND (T <> NULL) do
                    begin
                        if overlap(rectangle(o2), T) and (o2(r) == T(r))
                            then image2 = image2 + (T(image_list) in obj2(o))
                        else if ((o2(r) < obj2(r))
                            then return("Not permitted overlay resolution)
                            else T = T.child
                    end

                return(image1, image2)
            }
        end
end

```

```

Procedure overlap (o, node)
/* returns true if the rectangle in o is contained in or overlapping */
/* with the rectangle in node */

```

```

begin
  OXLB = o.lat - o.width/2      /* figure out upper and lower boundary */
  OXUB = o.lat + o.width/2      /* coordinates of image in o */
  OYLB = o.long - o.height/2
  OYUB = o.long + o.height/2
  if (OXLB > node.XUB) or (OXUB < node.XLB)
    then return (false)
  else if (OYLB > node.YUB) or (OYUB < node.YLB)
    then return (false)
  else return (true)
end

```

The `view` privilege can be enforced by searching down the MX-RS quadtree and retrieving all the images that have a resolution lower or equal to the allowed resolution level and at the same time contain (or intersect with) the requested rectangular region. $T(\mathbf{r})$ represents the resolution level in the node T . If higher resolution images may be viewed, then it checks breadthwise if the requested region is contained in any of the four children, and recursively collects the images.

For example, assume we have a collection of images with 1km, 20m, 10m, and 1m resolution levels arranged in the MX-RS quadtree shown in Figure 7, where each area (i.e. each node in the quadtree) is covered by a set of images. We have *GSAB* (geo-spatial authorization base) where a user `Alice` is allowed to access region `E` at up to 10m resolution. Now `Alice` wishes to view areas specified in the dotted rectangle region shown in Figure 6. The view algorithm checks her access control information and returns the images on nodes `A`, `E`, `F`, `G`, `H`, and `I`, which are `i1`, `i2`, `i3`, `i4`, `i5`, `i6`, ... `i13`, and `i14`. On the other hand, the user `Bob` is allowed to view the same region `E` but only up to 20m resolution. Therefore, the view algorithm will allow access to images at nodes `A` and `E` which will yield images `i1`, `i2`, `i3`, `i4`, `i5`, and `i6`.

Once the list of images that may be viewed and that contain region o is returned, each image in the list needs to be post-processed to render only the region specified in o . For instance, an image that contains more than o should be either cut or blurred to show only region o . This postprocessing step is not discussed in this paper.

The `zoom-in` privilege allows a user to view a region at a zoomed-in resolution level, $ur(pr_{zoom})$. The `zoom-in` algorithm takes a user's request to zoom-in which specifies information about the requested region $ur(o)$ and the zoom level $ur(pr_{zoom})$. It is just like view request, but the `zoom-in` algorithm returns only the images at the requested zoom-in resolution level.

Considering once again the same scenario as in the example above, `Alice` is allowed to zoom in to 10m resolution for the image in the dotted rectangular area in Figure 6. Thus, images in the nodes `F`, `G`, `H`, and `I` are returned, i.e. `i7`, `i8` ... , `i13`, and `i14`. `Bob`, who is allowed to zoom in to 20m resolution images in the dotted line region, will be shown images from the node `E`, i.e. `i4`, `i5`, and `i6`.

The overlay algorithm above takes a user request to overlay two image objects, $o1, o2$, and check each object with objects allowed to overlay in *GSAB*, $obj1, obj2$ in terms of resolution. It retrieves $images1$, all images that are allowed in $a(obj1)$ that overlaps with $rectangle(o1)$, and $images2$, all images that are allowed in $a(obj2)$ that overlaps with $rectangle(o2)$. The above algorithm allows overlaying of only two objects, but it can be

easily generalized to allow the overlay of n objects.

As shown in the procedure above, access control is enforced using authorization information and the spatial extent of the images which are used to derive resolution levels.

6 Conclusions and Future Research

In this paper, we have argued that near-continuous surveillance through high resolution satellite images when combined with geographic information could be a threat to privacy. In order to address this issue, we presented a suitable access control model, called Geospatial authorization model (GSAM). GSAM supports privilege modes including `view`, `zoom-in`, `overlay` and `identify` that are essential for providing constrained access to geospatial data based on the region covered by an image. We have demonstrated how access control can be efficiently enforced using a spatial indexing structure called MX-RS quadtree.

Our future research spans a number of directions. We intend to extend the authorization specification GSAM with temporal attributes. Unlike conventional authorizations that can be implemented as lists, authorizations in GSAM involve spatial attributes. In such a case, managing the authorization base and searching for authorizations based on the spatial extent is not trivial. Therefore, we intend to investigate techniques to maintain the authorization base. We plan to devise methodologies to verify the consistency of the authorization specification, analyze conflicts occurring due to simultaneous presence of contains, overlap and other operations, and strategies to resolve these conflicts. Another direction of future research is to build an indexing structure suitable for image access control in more general cases, where images at the same resolution level do not have fixed spatial extents. In addition, we intend to consider including the temporal aspects into the indexing structure. We also plan to investigate methods for providing refined access control where different geospatial information sets, such as health data and income data are integrated with image and map data.

7 Acknowledgment

The concept of access control for high resolution satellite imagery was conceived through discussions with Dr. Geoff Henebry, Biological Science Department, Rutgers University. We acknowledge Dr. Francisco Artigas, CIMIC, Rutgers University, for the information on geo-spatial images, their analysis and processing. The work was partially supported by the National Science Foundation under grant IRI-9624222 and the Meadowlands Environmental Research Institute as a grant from the Hackenack Meadowlands Development Commission.

References

- [1] Jonathan Ball. Satellite remote sensing. *TCS Remote Sensing and GIS web page*.
- [2] J Bentley. Multidimensional binary search trees used for associative searching. *CACM*, 18(9):509–517, September 1975.

- [3] M. Fleck, D. Forsyth, and C. Bregler. Finding naked people. In *Proceedings of 4th European Conference on Computer Vision*, pages 593–602, 1996.
- [4] D. Forsyth et al. Finding pictures of objects in large collections of images. In *Proceedings of International Workshop on Object Recognition*, pages 69 – 142, 1996.
- [5] I Gargantini. An effective way to represent quadtrees. *Comm. of ACM (CACM)*, 25(12):905–910, December 1982.
- [6] O. Gunther. The cell tree: an index for geometric data. Technical report, University of California, Berkeley, December 1986.
- [7] A. Guttman. R-trees: a dynamic index structure for spatial searching. In *Proceedings of ACM SIGMOD*, pages 47–57, June 1984.
- [8] K. Hinrichs and J. Nievergelt. The grid file: a data structure to support proximity queries on spatial objects. In *Proceedings of the WG'83 (International Workshop on Graph Theoretic Concepts in Computer Science)*, pages 100–113, 1983.
- [9] H. Jagadish. Linear clustering of objects with multiple attributes. In *Proceedings of ACM SIGMOD*, pages 332–342, May 1990.
- [10] D. Lomet and B. Salzberg. The HB-tree: a multiattribute indexing method with good guaranteed performance. *ACM TODS*, 15(4):625–658, December 1990.
- [11] J. Orenstein. Spatial query processing in an object-oriented database system. In *Proceedings of ACM SIGMOD*, pages 326–336, May 1986.
- [12] J. Robinson. k-d-b tree a search structure for large multidimensional dynamic indexes. In *Proceedings of ACM SIGMOD*, pages 10–18, April 1981.
- [13] H. Samet. *Applications of Spatial Data Structures: Computer Graphics, Image Processing, and GIS*. Addison-Wesley, 1990.
- [14] H. Samet. *The Design and Analysis of Spatial Data Structures*. Addison-Wesley, 1990.
- [15] Aref W. and H. Samet. Optimization strategies for spatial query processing. In *Proceedings of VLDB (Very Large Data Bases)*, pages 81–90, September 1991.
- [16] James Ze Wang, Jia Li, Gio Wiederhold, and Oscar Firschein. System for Classifying Objectionable Websites. In *Proceedings of the 5th International Workshop on Interactive Distributed Multimedia Systems and Telecommunication Services (IDMS '98)*, volume LNCS 1483, pages 113–124. Springer Verlag, September 1998.
- [17] James Ze Wang, Gio Wiederhold, and Oscar Firschein. System for Screening Objectionable Images Using Daubechies' Wavelets and Color Histograms. In *Proceedings of the 4th European Workshop on Interactive Distributed Multimedia Systems and Telecommunication Services (IDMS '97)*, volume LNCS 1309. Springer Verlag, September 1997.

[18] Robert Wright. Private Eyes. *The New York Times Magazine*, September 1999.