

GeoIRIS: Geospatial Information Retrieval and Indexing System—Content Mining, Semantics Modeling, and Complex Queries

Chi-Ren Shyu, *Member, IEEE*, Matt Klaric, *Student Member, IEEE*, Grant J. Scott, *Student Member, IEEE*, Adrian S. Barb, *Student Member, IEEE*, Curt H. Davis, *Senior Member, IEEE*, and Kannappan Palaniappan, *Senior Member, IEEE*

Abstract—Searching for relevant knowledge across heterogeneous geospatial databases requires an extensive knowledge of the semantic meaning of images, a keen eye for visual patterns, and efficient strategies for collecting and analyzing data with minimal human intervention. In this paper, we present our recently developed content-based multimodal Geospatial Information Retrieval and Indexing System (GeoIRIS) which includes automatic feature extraction, visual content mining from large-scale image databases, and high-dimensional database indexing for fast retrieval. Using these underpinnings, we have developed techniques for complex queries that merge information from heterogeneous geospatial databases, retrievals of objects based on shape and visual characteristics, analysis of multiobject relationships for the retrieval of objects in specific spatial configurations, and semantic models to link low-level image features with high-level visual descriptors. GeoIRIS brings this diverse set of technologies together into a coherent system with an aim of allowing image analysts to more rapidly identify relevant imagery. GeoIRIS is able to answer analysts' questions in seconds, such as "given a query image, show me database satellite images that have similar objects and spatial relationship that are within a certain radius of a landmark."

Index Terms—Geospatial intelligence, image database, information mining.

I. INTRODUCTION

GEOSPATIAL information mining is essential for coping with the tidal wave of multimodal geospatial intelligence data that is now routinely collected. Traditional textual metadata such as geographic coverage, time of acquisition, sensor parameters, manual annotation, etc., are now insufficient to retrieve images of interest when the visual content of the scene contains the primary relevant information. New methodologies and prototype systems for dynamically incorporating automatic feature extraction, visual selection, and knowledge-rich semantics for content-based image database management and retrieval are needed to assist image analysts. Such content-based mining tools will enable a more rapid geospatial intelligence analysis by letting the user focus his or her attention on the most critical and relevant portions of the data. The goal of such

systems is to provide an online analysis of multimodal datasets in which the imagery requiring manual intervention and analyst attention is flagged and then queued by the information mining system for human analysts. Using such a system enables analysts to be more productive in targeted search and imagery interpretation.

Conventional image retrieval in the intelligence community is performed either by using a keyword search or by browsing through a hierarchical information retrieval structure. One could build a traditional information search engine [3] to locate images by matching a text-based query with the terms in the descriptions using relational database methods. This text-based query approach is highly user- and context-dependent and does not provide much descriptive power. On the other hand, content-based image retrieval (CBIR) provides image-based query methods for a more visually descriptive matching of images than can be accomplished with text alone. Given a query image, a CBIR system extracts image features—ranging from pixels, regions, and objects to higher level descriptors of objects in the image—and retrieves database images that share similar visual patterns with the query. This is quite powerful, but low-level features do not have adequate specificity to reliably map the complex notions that analysts have about patterns common to imagery intelligence.

Therefore, how do we connect the images and their low-level features with an analyst's expert knowledge for retrieval of relevant imagery? This question lies at the heart of achieving semantic exchange, visual content management, and associations between visual patterns and text-based annotations.

Many CBIR systems have been proposed and implemented with varying degrees of success during the last decade. Early prominent systems include the IBM QBIC [10], Chabot [22], VisualSEEK [34], MARS [28], FIDS [5], and PHOTOBOK [23]. Smeulders *et al.* [32] provide a survey of CBIR systems prior to 2000, and more recent reviews are provided by Sebe *et al.* [27] and Smeulders *et al.* [33]. Significant progress has been made, but no truly successful system has emerged in terms of practical usefulness and integration with other non-image databases. One main reason for the lack of success is that the domains these systems address are too broad. The images used in most of the systems range anywhere from trees to fabric and even to people and faces.

Success has been significantly greater with several systems developed specifically for remote sensing imagery [1], [9],

Manuscript received March 1, 2006; revised October 17, 2006. This work was supported by the National Geospatial-Intelligence Agency University Research Initiatives (NURI) under Grant HM1582-04-1-2028.

The authors are with the College of Engineering, University of Missouri—Columbia, Columbia, MO 65211 USA (e-mail: shyuc@missouri.edu).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TGRS.2006.890579

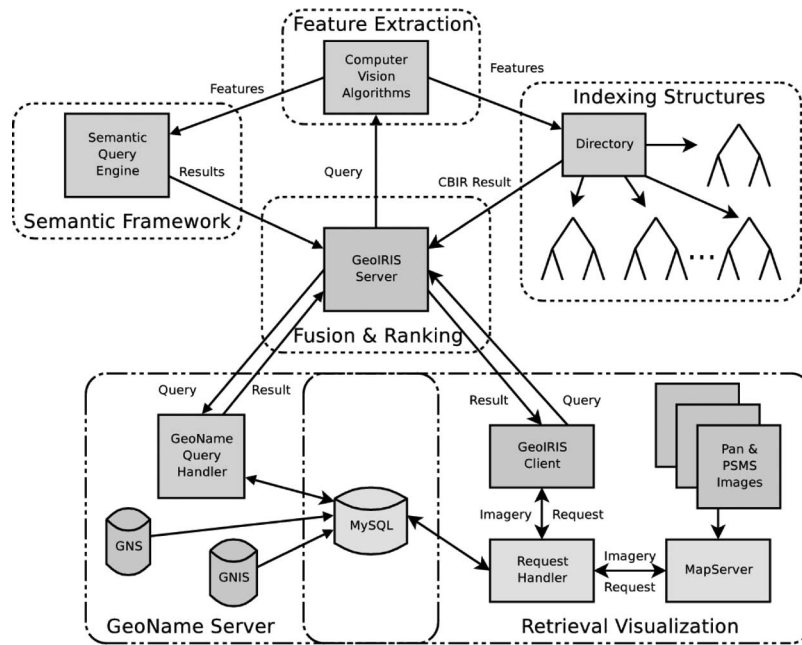


Fig. 1. GeoIRIS system architecture. Each of the major modules of our GeoIRIS system is shown above. The arrows linking the components demonstrate the interaction between the various subsystems. Our dataset consists of panchromatic (PAN) and PSMS imagery.

[17]. The large domain knowledge of intelligence tradecrafts related to geospatial data and information offers a unique opportunity to make further progress in image information mining and retrieval. These systems have made significant contributions in the design of system frameworks for geospatial CBIR techniques with large-scale spatial databases. However, these prominent systems may only partially fulfill intelligence analysts' needs. Further research is needed in utilizing indexing structures for efficient retrievals and in addressing the challenge of linking the analyst's semantics with low-level features. To make an image mining and retrieval system useful for image analysts, a more comprehensive approach is needed to build a multimodal database retrieval system.

This paper is organized as follows. In Section II, an overall system architecture is outlined, and each main component in the system is discussed. Section III lists tile-based and object-based feature extraction algorithms used in The Geospatial Information Retrieval and Indexing System (GeoIRIS). Section IV discusses two high-dimensional tree structures for object-based and tile-based indexing. To summarize the retrieval results from multiple feature sets, a multi-index ranking mechanism is introduced in Section V. Section VI provides a framework to link low-level features with high-level semantics. A suite of complex query methods is described in Section VII. Finally, Section VIII offers concluding remarks and the discussion of future work.

II. SYSTEM ARCHITECTURE

GeoIRIS can be best described by its architecture as shown in Fig. 1. There are six modules: feature extraction (FE), indexing structures (IS), semantic framework (SF), GeoName server (GS), fusion and ranking (FR), and retrieval visualization (RV). For offline indexing, all database images are processed in

the FE module and then indexed in the IS module. Linkages between a database image and its textual information are built using the GS module. Utilizing the extracted features and image annotation information, the SF module mimics the image analysts' domain expertise in describing high-level visual semantics by low-level features for semantics query and training of inexperienced image analysts.

When a query image is available, a user can submit the image and look for similar images from the database. This online process feeds the image into the FE module and sends the extracted features to the IS module for retrieving similar feature vectors from multiple indexing trees. If a semantic query is inputted by the user, the system calls the SF module to retrieve relevant images that contain the underlying semantics of interest provided by the user.

A key feature of GeoIRIS is the ability to merge information from heterogeneous sources. One such example of a data source useful in retrieval applications is a database containing geographic and cultural features of interest. For example, in the GS module, the locations of schools, churches, hospitals, lakes, and harbors are stored and linked to images indexed in the IS module. This information is acquired from the Geographic Names Information System provided by the U.S. Geological Survey [12] and from the GEOnet Names Server produced by the National Geospatial-Intelligence Agency [13].

The FR module is called to rank the results from heterogeneous intelligence sources based on user's preference. Top-ranked results are displayed to the user using the RV module. To provide an efficient handling of image requests, a mechanism for cataloging imagery for retrieval was developed. Underlying the visualization engine is MapServer [19] developed by the University of Minnesota. This open-source Geographic Information System (GIS) tool allows for on-the-fly creation of maps and imagery from full-size images.

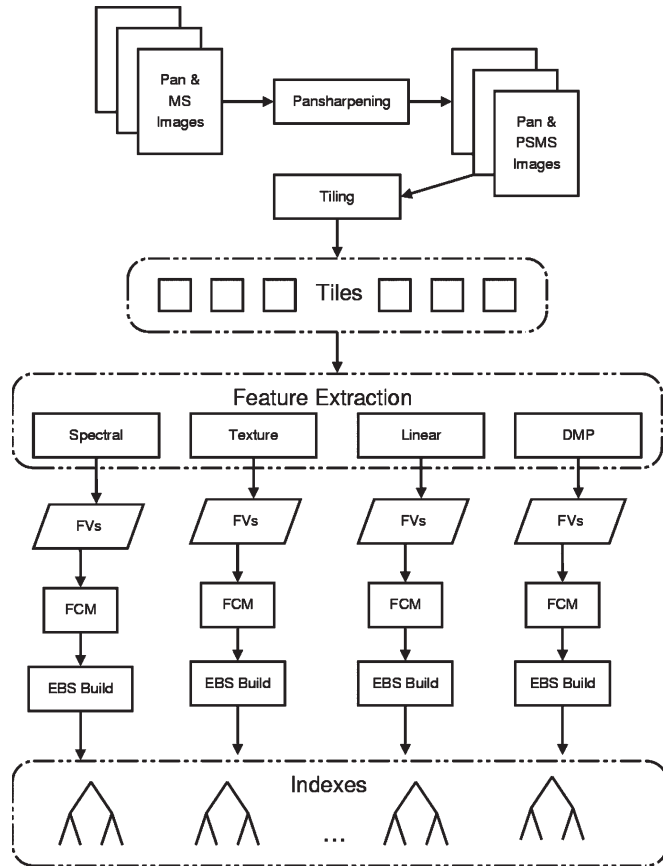


Fig. 2. Stages of image preprocessing, feature extraction and index building are shown in this figure. Following feature extraction, each set of feature vectors is clustered typically using the fuzzy c-means (FCM) algorithm prior to building EBS k -D Tree indexes. Note that only a sample of the feature extraction algorithms is shown here for simplicity.

III. FEATURE EXTRACTION

The first major consideration in the development of GeoIRIS is the determination of what relevant content will be used when retrieving images. We have developed our system using the features extracted from the high-resolution 0.6–1.0-m panchromatic and 2.4–4.0-m multispectral satellite imagery that we fuse to create a 0.6–1.0-m pan-sharpened multispectral (PSMS) imagery at the same spatial resolution for all channels including infrared.

A. Tile-Based Feature Extraction

Each ingested multispectral image is subdivided into image tiles of size $256 \text{ m} \times 256 \text{ m}$. The tile-based feature extraction is then performed on these individual tiles as shown in Fig. 2. The choice of tile size ensures that the extracted features capture the local characteristics within the tile and not the global features across the entire image. Each image tile has four quadrants that overlap with four neighboring tiles to capture information of objects across the boundaries.

Our features fall into two general categories: general image features and anthropogenic features (i.e., those associated with human modification of the landscape). For example, classic image processing and computer vision features such as spectral and texture measures are used. In addition, more specialized

features have been developed to represent characteristics of linear features (roads) and the scale of objects (buildings).

1) *General Feature Extraction*: The general feature extraction employs widely used computer vision algorithms for creating textural and spectral feature vectors. These features are important for discriminating between land-cover and land-use patterns such as urban, residential, cropland, etc. These features are the foundation of the content-based retrieval system that has been expanded with novel feature extraction algorithms.

Spectral Features: This group of features uses the spectral histograms of a given tile. The satellite imagery which is being used has an effective range spanning 11 bits. This range is divided into eight coarse bins to capture the general distribution pixel intensity in each tile. For each tile analyzed, three different types of spectral features are calculated: histograms for panchromatic, grayscale RGB, and near-infrared (NIR) data. The grayscale RGB is a monochrome intensity image computed from the visible channels using $G_{\text{RGB}} = 0.3 * R + 0.59 * G + 0.11 * B$.

Texture Features: Texture measures are essential for characterizing global visual patterns in high-resolution satellite images, such as landcover or sizable homogeneous regions. We apply the grayscale cooccurrence matrices described by Haralick *et al.* [14]. The six texture measures used are uniformity of energy, entropy, homogeneity, contrast, correlation, and cluster tendency. By averaging the responses from different angular values of the cooccurrence matrices, these features are rotationally invariant. To capture the textures across different scales, the texture measures are calculated for three different distances. The set of texture measures is generated for each channel—panchromatic, grayscale RGB, and NIR imagery.

2) *Anthropogenic Feature Extraction*: Anthropogenic image content is typically man-made elements detectable in the satellite imagery such as roads, buildings, and other structures. We are especially interested in anthropogenic elements due to their relevance in intelligence applications. The automatic extraction and retrieval of roadways and objects are important and necessary elements of GeoIRIS. Our tile-based anthropogenic features include linear features extracted from pixel correlation run-length and scale-based descriptors of object content generated from differential morphological profiles (DMPs) [24].

Linear Features: To capture the features specific to high-resolution satellite imagery, specialized feature detectors are used based on the study in [29]. One such feature identifies linear structures in imagery and measures characteristics of such structures. In satellite imagery, linear structures such as roads and paths are of particular interest. Images are first filtered using a vegetation mask based on thresholding the normalized difference vegetation index (NDVI) values; areas with high NDVI values correspond to vegetation and are excluded from the linear feature processing.

There are two components which comprise the set of linear features. The first set—termed linearity—characterizes the length of linear features present in the tile. In addition, these features capture the information regarding the ratio of the maximum length of a feature to its minimum length; in other words, is the feature long and skinny, or is it more compact?

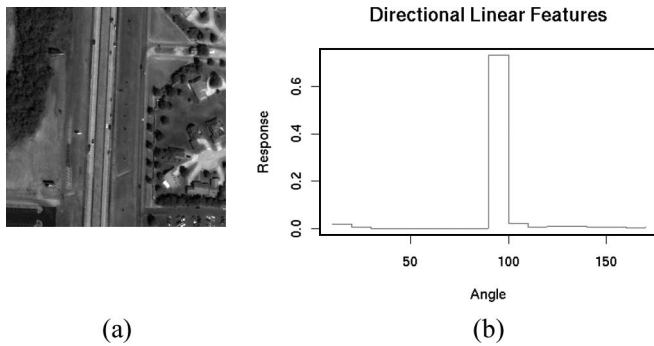


Fig. 3. Linear features can be used to capture information about linear structures present in an image. The tile shown in (a) contains a pronounced linear structure in the highway that runs from north to south in the image. The directional component of the linear features captures the presence of this highway as seen in the spike around 90° in (b). (a) Image tile. (b) Directional features.

The second set of features, directionality, captures the information about the angles at which linear features may be present. These features store values corresponding to the amount of linear structures from a tile that extends in the direction given by each angle and are binned into 18 ten-degree buckets. An example of a directional feature can be seen in Fig. 3.

Aggregate DMP Measure: The second class of anthropogenic features for which specialized techniques have been developed are objects. The term object is used in a general sense to describe significant structures found within an image. Objects include things such as buildings, water towers, airplanes, and open fields. A distinction is made between vegetative objects and nonvegetative objects by using an NDVI threshold at each pixel. Thus, two different object measures are calculated: one corresponding to the areas of vegetation and another corresponding to the nonvegetative objects.

A multiscale approach is taken for the problem of collecting object features by using the DMP originally developed by Pesaresi and Benediktsson [24]. This technique detects candidate structures at various spatial scales from airplanes to large building complexes; we aggregate the results of the structures detected at each scale (i.e., structuring element size) to arrive at one number for each scale in the profile and call this feature an aggregate DMP measure. These features allow the amount of response at each scale and, thus, a measure of the number of objects at each scale to be captured. An example of such features at varying scales can be seen in Fig. 4.

B. Object-Based Feature Extraction

In addition to the tile-based methods for CBIR, we have developed various methods that exploit the image content at the object level. We initially extract objects from the image content using the DMP. These objects are individually processed to extract shape and spectral features. In addition, configurations of objects are extracted into multiobject spatial relationships. Object-based query capabilities are an important extension to the tile-based image retrievals.

1) *Single-Object Characterization:* In order to index the objects for efficient retrieval, a set of features is needed to encode the shape properties of each object. The technique

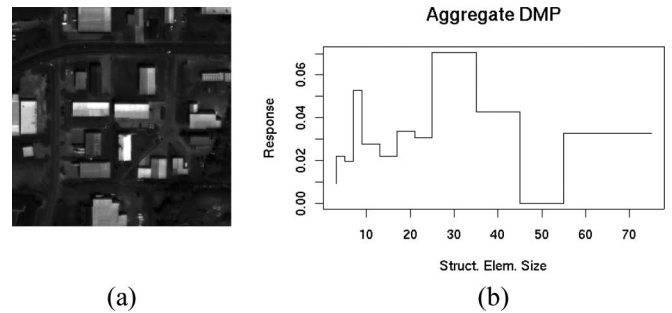


Fig. 4. Aggregate DMP features are used to encode information about the presence of objects at various spatial scales. To illustrate this, a tile is shown in (a) and the resulting features are depicted in (b). The features show a high response centered around the 30-m radius point; this corresponds to the size of many of the buildings present in the image. (a) Image tile. (b) Object features.

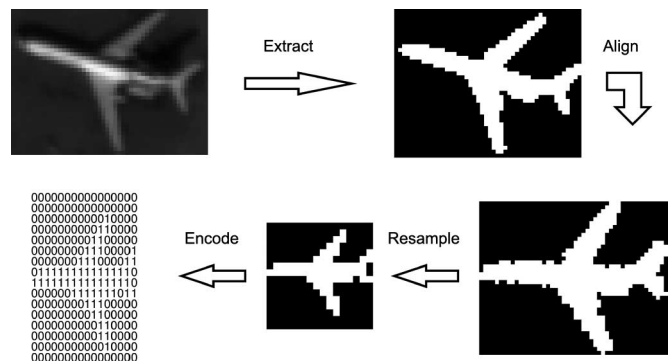


Fig. 5. Object shape characterization using object extraction from the DMP, principal axis alignment, resampling (downsampling in this case), and bitmap encoding.

chosen to represent the shapes of our objects is known as a grid descriptor [36]. This bitmap representation results in a coarse representation of the object that captures its general shape at the expense of fine detail. Efficient representation of the object shape is critical due to the fact that even relatively small amounts of imagery will result in millions of objects being extracted.

The process of automatically analyzing the objects for efficient retrieval in imagery can be broken down into several phases after obtaining candidate structures from the DMP. The output of the DMP is thresholded to limit our pool of objects to those which have a very high level of response at a given scale; from this process, we arrive at our set of objects to be indexed. We developed an additional approach for object extraction detailed in [15]. The airplane and its binarized image are depicted in the top row of Fig. 5 as the results obtained at this stage of the process from an image tile. To encode the shape of each object using the grid descriptor method, each of the objects is aligned to prevent sensitivity to translation and rotation. The object is then resampled to a fixed size of 32×32 bits to represent an object's shape regardless of its size. A binary feature vector corresponding to the shape of the object is finally encoded based on this bitmap.

In addition to storing the shape of the object, a second vector is generated for each object that stores the average intensity in each band of imagery within the object mask. This information can be used to limit retrievals to objects with similar spectral characteristics.

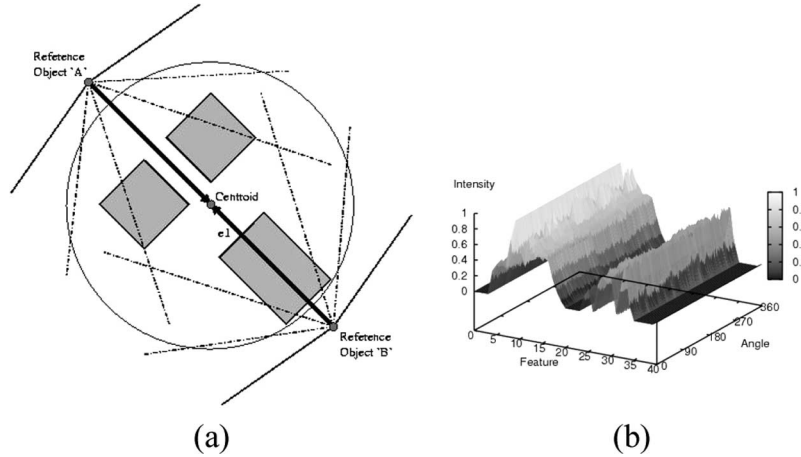


Fig. 6. Multiobject spatial modeling. (a) Three objects are shown with their centroid and principal axis e_1 . Two reference objects are placed outside the bounding circle along e_1 , equidistant from the centroid. (b) A surface plot of the multiobject spatial relationship signature of a three-object configuration rotated through 360° . The uniform shape along the axis labeled “Angle” indicates the insensitivity to rotation in our features. (a) Three objects. (b) Signatures.

2) *Multiobject Spatial Relationships*: An extremely challenging task in satellite imagery is the retrieval of similar spatial configurations of man-made objects found within a large imagery collection. Toward this goal, we have developed a multiobject spatial relationship feature extraction algorithm that is highly invariant to scale, translation, and rotation [26]. This feature extraction method produces multidimensional spatial relationship signatures, which are then indexed for similarity search. There are two main tasks accomplished for multiobject spatial relationship indexing: grouping of nearby objects into spatial configurations and development of an appropriate invariant signature with automatic feature extraction.

This subsection describes the approach used to generate a spatial signature of an object configuration by extending the pairwise determination of spatial relationships using the histogram of forces [20]. Given an n -tuple configuration, there are $n(n-1)/2$ symmetric pairwise spatial relationships. Attempting to correlate two configurations is equivalent to graph matching—an intractable problem.

Our solution is to treat the n -tuple configuration of objects as a single disjoint object. We then generate two synthetic reference objects outside of the configuration. For each synthetic object, we compute the histogram of forces against the entire configuration treated as a single disjoint object. Fig. 6 provides a visual depiction of the dual reference object placement. Each synthetic-reference object pair is used to generate rotation, translation, and scaling invariant signatures using the histogram of forces. It has been shown that rotations of a two-object configuration result in an angular shift of the histogram of forces [21]. Therefore, to achieve rotationally invariant signatures of a configuration, we must ensure that the synthetic object is always placed in the same position relative to a rotated configuration of the same group of objects.

To accomplish this, we find the principal eigenvector of a spatial configuration and always place the synthetic objects outside the configuration along the principal axis. The synthetic objects are circular, and their position along the principal axis is always axis length plus the radius plus one pixel.

TABLE I
AVERAGE RECALL OF ROTATED CONFIGURATIONS. TEN OBJECT CONFIGURATIONS WERE ROTATED BETWEEN 0° AND 360° AT 5° INCREMENTS FOR A TOTAL OF 720 SETS. THE VALUES SHOWN INDICATE THE AVERAGE RECALL AT RANK n IN THE RESULTS. RECALL AT RANK n IS CALCULATED AS THE PERCENT OF THE EXPECTED CONFIGURATIONS CORRECTLY RETURNED IN THE TOP n RESULTS.
RECALL OF SCALED CONFIGURATIONS: TEN OBJECT CONFIGURATIONS WERE SCALED TO TEN DIFFERENT IMAGE SIZES FOR A TOTAL OF 100 SETS. THE VALUES SHOWN INDICATE THE RECALL AT RANK n IN THE RESULTS

Recall %	Rank	1	2	3	4	5	10
	Rotation	82	91	97	98	98	100
	Scale	82	97	99	100	-	-

We construct our spatial signatures by calculating the histogram of forces, H_{+y} and H_{-y} , for the positive and negative y reference objects against the object configuration. The histogram generated from each reference object is then aligned to the principal axis of the configuration. After this alignment, two windows up to 180° , W_{+y} and W_{-y} , centered at the principal axis, are constructed from H_{+y} and H_{-y} . Each W is partitioned into F bins, and each bin generates a feature value which is the average response from H over that bin. In GeoIRIS, we chose an F value of 20. This results in each feature being an average response across 8° . From both reference objects, we compute $2 * F$ histogram response features for a spatial configuration. To ensure that the features are rotationally insensitive, we order each bin, $i \in [1, F]$, from W_{+y} and W_{-y} , such that

$$S[i] = \max \{W_{+y}[i], W_{-y}[i]\} \quad (1)$$

$$S[i + F] = \min \{W_{+y}[i], W_{-y}[i]\} \quad (2)$$

where S is the spatial signature from the object configuration and $W_{+y}[i]$ and $W_{-y}[i]$ represent bin i from W_{+y} and W_{-y} , respectively. As a final step, the spatial signature $S[i]$ is normalized to $[0, 1]$ to provide scale invariance, and a final feature is added to represent density of the objects within our earlier defined bounding circle. A brief explanation of the experimental methods and results is provided in Table I.

TABLE II
 FEATURES AND THEIR COMPUTATIONAL COMPLEXITIES FOR GEOIRIS. SPECTRAL AND TEXTURE FEATURE SETS ARE COMPUTED IN THREE SETS: PANCHROMATIC, VISIBLE SPECTRUM (RGB), AND NEAR-INFRARED (NIR). AGGREGATE DMP FEATURES ARE FROM VEGETATIVE AND NONVEGETATIVE FEATURE SETS

Feature	Dimension	Complexity
Spectral	8 each	$O(H * W)$
Texture	21 each	$O(H * W + B^2)$
Linear	37	$O(H * W * D * L)$
Aggregate DMP	26 each	$O(H * W * S * U)$
Object Spectral	5	$O(N)$
Object Shape	32x32	$O(N^2)$
Multi-Object Spatial Relationship	41	$O(H * W + N * A)$

In GeoIRIS, we generate configurations of 3-, 4-, and 5-tuples using all extracted objects within a 0.5-km radius of the seed object. Each configuration is then processed to create a multiobject spatial relationship signature.

C. Feature Extraction Complexity

We have evaluated the algorithm complexity of each of the feature extraction techniques employed. A summary of these results can be found in Table II. Calculating the spectral features of an image simply involves iterating through all pixels in the image; thus, its complexity is bound by the height (H) and width (W) of the image. Texture contains an additional term B which is the number of bins in the gray level cooccurrence matrix. More complicated are the complexities of the anthropogenic features. Extracting the linear features from imagery requires a complexity dependence upon the height, width, number of directions scanned (D), and max length scanned (L). The final tile-based feature, the DMP, does not easily lend itself specifying an analytical complexity. The number of times which this algorithm must iterate is dependent upon the image characteristics; thus, we define a complexity which involves the number of scales (S) and an unknown term (U) which is image dependent related to the convergence of the DMP algorithm.

Analytical complexities are also provided in the table for the object-based features. Object spectral characteristics can be extracted by examining each of the N pixels in the object. To produce object shape features in the form of our bitmap, the limiting factor is the axis-alignment process. This can be completed in a time on the order of N^2 , where N is once again the number of pixels in the object.

Lastly, the complexity of object spatial relationship feature extraction can be defined. The multiobject features require that the objects be written into a raster image; thus, the height and width of the bounding box of the objects can be used to define the complexity of that step. Added to that is the cost of actually performing the feature extraction; this can be defined in terms of the number of pixels that make up the objects (N) and the number of angles (A) at which features are computed. The overall complexity can be seen in the table.

IV. HIGH-DIMENSIONAL INDEXING FOR FAST RETRIEVALS

In this section, we present two indexing mechanisms for tile-based and object-based query methods. Indexing of continuous

valued features is currently done using the entropy balanced statistical (EBS) k -dimensional (k -D) tree [25], and indexing the binary-valued features is performed with the entropy balanced bitmap (EBB) tree. Tile-based indexing provides access into localized areas of similar features. For instance, given a query using an image of an urban area with a particular characteristic building type, the EBS k -D tree over the aggregate nonvegetative DMP feature space allows queries to find similarly scaled distributions, such as numerous small or a few large buildings. Object-based indexing includes both individual objects and spatial configurations of multiple objects. We are able to query for individual objects, such as airplanes or water towers, as well as finding configurations of objects such as building complexes.

Evaluations performed in this section are based on the following setting. The image database contains 70 824 image tiles and 531 208 extracted objects. Each machine runs the Linux operating system with hardware up to dual 2.8-GHz Xeon processors with Hyperthreading and up to 6 GB of RAM.

A. EBS k -D Tree for Continuous Features

As discussed in Section III, most feature sets in GeoIRIS are collections of multidimensional feature vectors. Currently, we use the feature sets described in Section III-A to accomplish the tile-based indexing. Each feature space, such as panchromatic spectral response, is built into its own EBS k -D tree index. The collection of EBS k -D tree indexes belongs to the IS module of GeoIRIS. In the context of the following discussion, $Class_i$ is a grouping of similar content (i.e., image tiles); in some cases, it may be from automatic clustering, or it may be from human analysts' annotations. In other cases, we define a partial knowledge as the labeling by experts of a subset of the data. This partial knowledge is grown through clustering techniques to generate labels for the remaining data.

The EBS k -D tree is designed to index large collections of high-dimensional feature vectors. It is particularly powerful when the domain knowledge, such as image analysts' semantic labelings or classes, is available for a small portion of the database. Examples include "harbor with ships," "under construction area," and "highway intersection." The algorithm first applies clustering to roughly assign a preliminary label to each image in the database. It then grows the EBS k -D tree, using top-down decision tree induction, by reducing the entropy at each successive node splitting while observing certain entropy

balancing factors. The decision criteria for the EBS k -D tree is to find the feature and threshold which maximizes

$$\gamma = H_{\text{parent}} - \sigma_{H_R} - \sigma_{H_L} - |\sigma_{H_R} - \sigma_{H_L}| \quad (3)$$

where H_{parent} is the entropy of the parent node, σ_{H_R} and σ_{H_L} are the weighted sum components of the right and left child nodes, respectively, and $\sigma_H = P(\text{Leaf}_j)H(\text{Leaf}_j)$. In (3), the first three terms on the right-hand side represent the reduction of entropy. The later two terms represent the balancing factor. We also constrain the node splitting by requiring the percentage reduction in entropy to meet a threshold. We calculate the entropy of a leaf using

$$H(\text{Leaf}_j) = -\sum_{i=1}^L P(\text{Class}_i|\text{Leaf}_j) \log P(\text{Class}_i|\text{Leaf}_j) \quad (4)$$

where L is the number of classes which exist in Leaf_j . Since the entropy is a measure of uncertainty, our tree building process should seek to minimize the entropy in the leaves; thereby increasing the accuracy of searches that reach Leaf_j . From Bayes' Theorem, we have

$$P(\text{Class}_i|\text{Leaf}_j) = \frac{P(\text{Leaf}_j|\text{Class}_i)P(\text{Class}_i)}{P(\text{Leaf}_j)} \quad (5)$$

where $P(\text{Class}_i)$ is the *a priori* probability of Class_i and we calculate $P(\text{Leaf}_j)$ as the sum of joint probabilities

$$P(\text{Leaf}_j) = \sum_{i=1}^C P(\text{Leaf}_j, \text{Class}_i). \quad (6)$$

If we examine a single feature k of Class_i using the bounds of the feature in a Leaf_j , we can determine the normal distribution values from a simple lookup table. Letting $C_{i,\min}^k$ and $C_{i,\max}^k$ represent the bounds of feature k for Class_i , and $a_{j,\min}^k$ and $a_{j,\max}^k$ represent the range of data on feature k for Leaf_j ; we calculate the probability of Leaf_j given Class_i using

$$P(\text{Leaf}_j|\text{Class}_i) = \prod_{k=1}^K \int_{\max(a_{j,\min}^k, C_{i,\min}^k)}^{\min(a_{j,\max}^k, C_{i,\max}^k)} f_i^k(a^k) da^k. \quad (7)$$

The integral value is evaluated numerically or using tables of the normal distribution integral values with $f_i^k(a^k)$ representing a normal probability density function of feature k conditioned on Class_i .

All feature vectors are fed into the root node of the tree, and the algorithm recursively finds the best feature to use for data splitting based on (3). The feature space is then carved into a number of hypercubes which are expected to group visually similar images and form leaf nodes of the EBS k -D tree. The induction time of the EBS k -D tree is dependent on the dimensionality and size of the data set. For our current database of imagery, build times vary from half an hour to around 1 h. Build times scale linearly with increases in database size and dimensionality.

B. EBB Tree for Binary Features

Here, we present an extension of the EBS k -D tree designed to index large numbers of bitmaps in a multidimensional binary feature space. The need for a specialized index for bitmaps arises due to the extracted object shape representations in Section III-B1. Because our shape features are encoded in grid descriptors or bitmaps, we developed the EBB tree. The size of the EBB is much smaller than an equivalent full traditional bitmap index and more accurately captures the tendencies of the clustered objects in the bitmap space than a traditional continuous valued index.

Although the clustered bitmaps could be represented as vectors of floating-point numbers, their mean values have no significance in the binary domain. Therefore, parameterized statistical distributions—as used in EBS k -D tree induction—have little relevance to the data. Furthermore, entropy reduction methods typically examine the data in a leaf using the class range for each feature in that leaf. By definition, the area under the distribution curve is the probability of the class given by the feature in that node. Given a node with Class_i that has all of feature k set to one value, the area under the distribution does not accurately represent the class (e.g., the minimum and the maximum are equal, which equates to zero area and zero probability in a statistical distribution). If Class_i has a bit both set and unset in some node, the minimum is 0 and the maximum is 1, which spans the binary space and which by definition implies probability of 1.0. What is desired is the point-based probability, not the area under statistical distribution.

There is a set of probabilities which describe a cluster of bitmaps. Precisely, for each feature k or bit of Class_i

$$P(\text{Class}_i, k=0) = \frac{|\text{off bits}|}{|\text{Class}_i|} \quad (8)$$

$$P(\text{Class}_i, k=1) = \frac{|\text{on bits}|}{|\text{Class}_i|} \quad (9)$$

represent the probability of bit being off or on, respectively. Similar to the EBS, we use (3), (4), and (5); for objective function and related leaf entropy, we define the probabilities of Class_i and Leaf_j as

$$P(\text{Class}_i) = \frac{|\text{Class}_i|}{|X|} \quad (10)$$

$$P(\text{Leaf}_j) = \frac{|\text{Leaf}_j|}{|X|} \quad (11)$$

where X is the total database population set. We calculate the remaining probability needed for (5), Leaf_j given Class_i as

$$P(\text{Leaf}_j|\text{Class}_i) = \prod_{k=1}^K P(\text{Class}_i, k, j). \quad (12)$$

Equation (12) is the binary equivalent to the continuous valued statistical version in (7). If a particular Leaf_{*j*} contains only off bits for feature *k* of Class_{*i*}, then $P(\text{Class}_{i,k,j}) = P(\text{Class}_{i,k=0})$. Given only on bits, then $P(\text{Class}_{i,k,j}) = P(\text{Class}_{i,k=1})$. If Leaf_{*j*} contains both on and off bits for bit *k* of Class_{*i*} in Leaf_{*j*}, then $P(\text{Class}_{i,k,j}) = 1.0$.

We consider this approach superior to working with 1024-dimensional continuous valued feature vectors that do not accurately capture the meaning of the bitmaps. The alternative of a pure bitmap index, that does not exploit entropy or clustered bitmaps, is impractical as our 1024 bits would lead to 2^{1024} leaves supported by $2^{1024} - 1$ internal nodes. Given our current object database, the EBB tree builds in 105 min. The build times of the EBB scale linearly with the increase in the database size.

C. Searching the EBB and EBS *k*-D Trees

Searching the EBB and EBS *k*-D trees requires a query search vector representing a position in feature space. For the EBS *k*-D tree, the query is a member of the appropriate *k*-D feature space, such as those corresponding to texture or vegetative aggregate DMP. During the induction of the EBS *k*-D tree, the decision feature and threshold are stored in each internal node. The values of the various query attributes allow the query to navigate to the appropriate leaf through a series of binary decisions. In EBS nonleaf nodes, if the leaf's decision feature has a threshold value above the search vector's feature value, the navigation continues left—otherwise right. When the leaves of the tree are reached, the first leaf has its population added to the ranked result set. Often, we encounter the situation where a leaf node does not contain the number of samples required by a search. At this point, searches must continue spiraling outward in *k*-dimensional space from the initial leaf. To accommodate the need to traverse the leaf structure in a nonlinear fashion, we build a secondary index over the set of leaves. This is accomplished by calculating the prototypes of the leaf clusters and using these prototypes to index the leaves in an M-tree [7] structure. A search in the feature space for *n* results is performed using Algorithm 1.

Searching the EBB tree is extremely efficient. During the induction of the index, each node stores a bitmap with the single decision bit set DB_{*j*}. Given a query bitmap QB, the navigation down the index is simple bitwise operations. When QB AND DB_{*j*} = DB_{*j*}, navigate right, else left. The result set of the EBB tree is ordered using a distance measure based on Tanimoto similarity [35]. In addition, the leaf linking M-tree depicted in Fig. 7 is generated using probabilistic bitmap prototypes to facilitate using the Tanimoto distance. Once the leaf nodes are reached, the results are ordered using the Tanimoto distance, and a leaf traversal is conducted identical to the EBS *k*-D tree.

The performance of both EBB and the EBS *k*-D tree is fast and consistent. When searching for the 100 most similar database items, from the 70 824 tiles in the database, the results are produced from 200 to 350 ms. This time is consistent across our different feature spaces. Search times depend on three properties of the database and search: the size of the desired

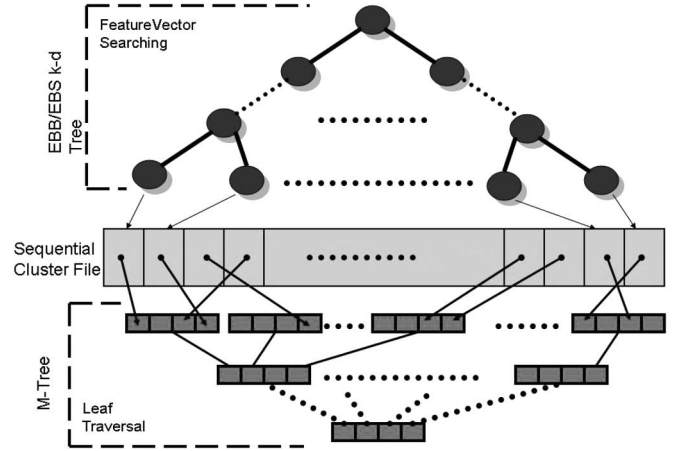


Fig. 7. File structure organization of an EBB/EBS *k*-D tree indexed content database. Searches are conducted using the EBB/EBS *k*-D tree, which partitions the feature space into relatively small clusters. Data are stored in a data file, organized in depth-first order of the tree leaves. When traversal through the leaf population is required, an M-tree structure is used for navigation through the data file.

result set, the dimensionality of the data, and to a lesser degree, the size of the database. The result size is the dominant cost of the query; as more results are requested, more processing is needed. The dimensionality of the feature space governs the cost of comparing any two database items for similarity. Finally, the larger the database, the more comparisons are required to reach the destination leaf. In our current system, our object data set is over seven times larger than the tile data set. The EBB tree is governed by the same properties for search times, but is typically faster, as most operations are bit operations instead of floating-point operations, despite the larger data set. Retrieving the top 100 ranked objects from our object database returns the results from 200 to 450 ms.

Algorithm 1 EBS *k*-D tree Searching for *n* results from population *P* partitioned into leaves *L*

- 1: Navigate to EBB/EBS *k*-D leaf from root, using each internal node's designated decision feature;
- 2: Order destination leaf's feature vectors in order of similarity into result *S*;
- 3: **if** $|S| < n$ **then**
- 4: $m \leftarrow n - |S|$
- 5: $leafCount \leftarrow (m / (|P| / |L|))$
- 6: **while** $|S| < n$ **do**
- 7: $leafCount \leftarrow leafCount + 0.5|L|$
- 8: $L' \leftarrow$ Search M-Tree for $leafCount$ closest leaves;
- 9: **for all** $L' \notin VisitedLeaf$ **do**
- 10: Add L' data to *S*
- 11: Add L' to *VisitedLeaf*
- 12: **end for**
- 13: **end while**
- 14: **end if**
- 15: return *S*;

V. RETRIEVAL FROM MULTIPLE INDEXES AND INFORMATION SOURCES

Section III described the many different types of features that have been extracted for use in GeoIRIS. It should be no surprise

that some classes of features are more useful in discriminating relevant results for certain types of queries, while a different set of features may be more useful for a second query. For example, linear features are particularly useful in identifying images relevant to a query image that includes road structures or a large highway. For this reason, a retrieval system should provide flexibility in determining how much each index contributes to the creation of the overall result list. To this end, users can weight indexes w_i corresponding to each index i . In our implementation, there were weights that corresponded to each of the classes of the tile-based features described in Section III. The vector $\vec{W} = \{w_i | 0 \leq w_i \leq 1\}$ represents the weight of each index. Once the weights of each index are established, the query can be performed against each index.

Let k_r be the image key name of the r th ranked result and d_r be the distance of this result from the query image q . The universe of image key names (i.e., all possible locations in the image database) is defined as the set \mathcal{K} . The result set of index i with query q is given by the following:

$$S_i = \{(k_r, d_r) | k_r \in \mathcal{K}, d_r = \text{dist}(q, k_r, i), d_r \leq d_{r+1}\} \quad (13)$$

where S_i is a set of image key and distance pairs in a sorted order. The image key must be a member of the universe of image keys \mathcal{K} , and the distance is defined by the chosen distance metric. In (13), the distance metric is a function of i the index number, because each index may have a unique distance metric chosen based on the type of features stored.

After searching each individual index, the final result set is constructed by aggregating the ranking information from each index. During this process, it is likely that a given result may not appear in the top- N result set returned by every index. For those indexes where a result does not appear, its distance will include a penalty that causes the distance to be larger than the maximum distance in the result set. A vector containing the maximum distance value in each result set can be constructed by

$$\vec{M} = \{m_i | m_i = \max \{S_i(d_r), \forall r \in [1, |S_i|]\}\}. \quad (14)$$

Once these maximal values have been identified, the final result set can be compiled.

By iterating over all possible keys and all possible indexes, the distance can be computed by summing the weighted distance score for results which occur in the result set for a given index and the weighted maximum distances for results which do not occur in the result set for a given index. The following equation outlines the method used to compute the final set of results:

$$F = \left\{ (k'_r, d'_r) \mid k'_r \in \mathcal{K}, d'_r = \sum_{i | k'_r \in S_i(k_r)} w_i * S_i(d_r) + \sum_{i | k'_r \notin S_i(k_r)} w_i * 1.1 * m_i \right\} \quad (15)$$

where the index weights come from the vector \vec{W} and are used to bias the results from specific indexes as per user request. In the calculation of d'_r , the first summation corresponds to

indexes in which the result is present, while the second summation refers to indexes in which the result is not present. These two values add to create the total distance value of a given result.

After computing the final result set, the elements of the set must be ordered in increasing order. This can be accomplished by constructing the set F_{sorted} , where

$$F_{\text{sorted}} = \{(k'_r, d'_r) | (k'_r, d'_r) \in F \wedge d'_r \leq d'_{r+1}\}. \quad (16)$$

Once the final result set has been sorted, a subset of the most highly ranked results is returned to the user.

In addition to the multi-index query by an example process previously discussed in this section, queries can also combine the information from multiple data sources, and these are termed hybrid queries. One example of such a query makes use of the GS module of our system. This database contains the location information about both geographic and anthropogenic features. By coupling the information found in this database and the results from a content-based query discussed above, the results can be restricted to a specific geographic region. For example, a hybrid query could be performed by providing an image of an area undergoing a construction with the geographic constraint that results must be within 3 km of a church. With these capabilities, image analysts are able to provide more specific constraints which allow more focused queries to be performed.

These hybrid queries involving data from multiple sources are performed by specifying a geographic constraint restriction. For example, a hybrid query might include a specific restriction that results should be returned only if they are within 2.5 km of a harbor. We can define the set of image keys H from our universe \mathcal{K} which satisfy the query constraint

$$H = \{k | k \in \mathcal{K} \wedge k \text{ satisfies the hybrid query}\}. \quad (17)$$

Once this half of the query is performed, the query-by-example portion can be executed according to (16). The result set from the hybrid query is generated by intersecting the results from (16) and (17) as follows:

$$F' = \{(k'_r, d'_r) | k'_r \in F_{\text{sorted}}(k_r) \wedge k'_r \in H\}. \quad (18)$$

Equation (18) shows the construction of this final set of results from a hybrid query as the intersection of data from multiple sources.

Evaluating the performance of queries in GeoIRIS must take into account each of the query mechanisms provided. First, the tile-based queries allow users to perform CBIR queries against the database. Our choice to index each group of features separately results in two important considerations. The queries against each index containing tile features can be performed in parallel; our system lends itself to distributing the indexes across the multiple machines—a feature which is exploited in GeoIRIS. Once all indexes have been queried, there is an additional price for combining the results from the multiple indexes. On the average, the tile-based queries in GeoIRIS take 5 to 7 s to search a database currently containing around 70 000 tiles. When these queries are performed in conjunction with

heterogeneous databases such as our GS, an additional computation is required leading to queries which take between 20 and 25 s; with further research, this process could be accelerated with additional indexing techniques. Lastly, despite containing over a half million objects, the object-based queries occur in around 5 to 7 s. The fact that these queries are performed in roughly the same amount of time as tile-based queries is an evidence of the use of successful bitmap indexing techniques.

VI. SEMANTIC MODELING

Textual metadata has proved to be insufficient [6] for articulating interesting content included in geospatial images. Semantic modeling has been shown to be useful in representing and integrating geographical information systems [11], [30]. To mimic the tacit knowledge of the image analysts for semantic retrievals, we define semantic terms (class labels) to model perceptual categories identified by domain experts and select an appropriate set of relevant features to describe the semantic terms. This semantic modeling process closes the gap between high-level text-based descriptions and low-level image features.

To achieve this goal, we have a three-step knowledge discovery process [31]: 1) data transformation: each continuous feature is partitioned into multiple discrete ranges that are meaningful for a chosen semantic; 2) mining associations: association rules [2] that map feature intervals into semantic categories are discovered; 3) semantic modeling: crisp intervals from association rules are replaced by sigmoid functions.

1) *Data Transformation*: Images with the same semantics are expected to be grouped together in the EBS k -D tree discussed previously in Section IV. For a semantic of interest such as Construction, the system visits all leaf nodes in the tree and selects those with a matched label. For each matched node, we determine the path in the tree and form a set of intervals for each feature which is normalized between 0 and 1. For example, let us consider a matched leaf node k containing a total of n_k labeled images and $n_k^{\text{Construction}}$ images with the semantic Construction. This node has the following path in the EBS k -D tree: $f_{15} \leq 0.5 \rightarrow f_{20} > 0.8 \rightarrow f_{15} > 0.2 \rightarrow f_{48} \leq 0.1$. The set of intervals for this node is $f_{15} \in (0.2, 0.5]$, $f_{20} \in (0.8, 1.0]$, and $f_{48} \in [0, 0.1]$. If we consider each labeled image from the node a data mining transaction, the intervals determined by the tree can approximate transaction items and be used for discovering association rules for semantic assignments. This process is iterated through all matched nodes and will generate sets of intervals for association rule mining.

2) *Mining Association Rules From Image Content*: We use the Classification based on Multiple Association Rules (CMAR) algorithm [8], [18] for discovering association rules using the transactions identified from the previous step. An example rule has the form $\{f_{15}(0.2, 0.5], f_{20}(0.8, 1.0], f_{48}[0, 0.1]\} \rightarrow \text{Construction}$. In this rule, the antecedent contains a set of discrete intervals, while the consequent is a semantic term. Each rule u has also specified support s , antecedent A , consequent C , and the confidence level c . The support of u is $s(u) = n_k^{\text{Construction}}/N$; the support of the antecedent is $s(A) = n_k/N$; and the support of the consequent is

$s(C) = \sum_{\forall k} n_k^{\text{Construction}}/N$, where N is the total number of images that were pre-labeled by domain experts; the confidence can be obtained by $c(u) = n_k^{\text{Construction}}/n_k$. A set of frequently cooccurring items from the selected rules will be used to model visual semantics for the predefined categories. Each item represents an interval of a specific feature and is mathematically modeled by a flexible possibility function.

3) *Semantic Modeling*: After the association rules are discovered, we refine them by replacing the crisp feature intervals in antecedents with possibility distributions as described in [4]. This approach has the advantage of capturing users' preferences in a computational way using an asymmetric property modeled by two halves of sigmoid functions (L —left and R —right). The following equation shows the possibility function $p(m)$ used to model the semantic assignment for an interval of a feature f . Each sigmoid function is controlled by three parameters: the center of the function (λ_1), the width factor (λ_2), and the exponential factor (λ_3). For each semantic term, there are two sets of λ values for the left sigmoid function, $(\lambda_1^L, \lambda_2^L, \lambda_3^L)$, and the right sigmoid, $(\lambda_1^R, \lambda_2^R, \lambda_3^R)$. The possibility function assigns a full degree of satisfaction to all the feature measures m in the regions between the two half sigmoids

$$p(m) = \begin{cases} \frac{2}{1+e^{((\lambda_1^L - m)/\lambda_2^L)^{\lambda_3^L}}}, & \text{for } m < \lambda_1^L \\ 1, & \text{for } m \in [\lambda_1^L, \lambda_1^R] \\ \frac{2}{1+e^{((m - \lambda_1^R)/\lambda_2^R)^{\lambda_3^R}}}, & \text{for } m > \lambda_1^R \end{cases} \quad (19)$$

This possibility distribution is shaped using the information in the training data set. First, a feature interval i is partitioned into a fixed number of evenly distributed subintervals. Then, the number of training images for each subinterval is counted and replaced with a normalized value. Using this information, the possibility function is computed using the algorithm described in [4]. These functions are then stored to be used for semantic queries.

4) *Semantic Search*: This query method searches the image database by semantics using the association rules described previously. For a given query such as "Retrieve images with Construction," the system first finds the rules that apply to the semantic term Construction, computes the relevance T of image j to each rule u for this semantic term, and then ranks the images in a descending order. The relevance of image j to the semantic term Construction is given by the following formula:

$$T_j^{\text{Construction}} = \sum_{\text{all } u} \frac{\min_{\text{all } A} (p_{\text{Construction}}^A(m)) \cdot c(u)}{e \cdot (\min(s(A), s(C)) - s(A) \cdot s(C))^2} \quad (20)$$

where c is the confidence, s is the support, and e is obtained from the following equation [18]:

$$e = \frac{1}{s(A) \cdot s(C)} + \frac{1}{(1-s(A)) \cdot s(C)} + \frac{1}{(1-s(A)) \cdot (1-s(C))}. \quad (21)$$

Equation (20) evaluates the minimum degree of satisfaction of image j 's features to the antecedents of rule u to ensure that the features satisfy the rule u . For image j , the system computes

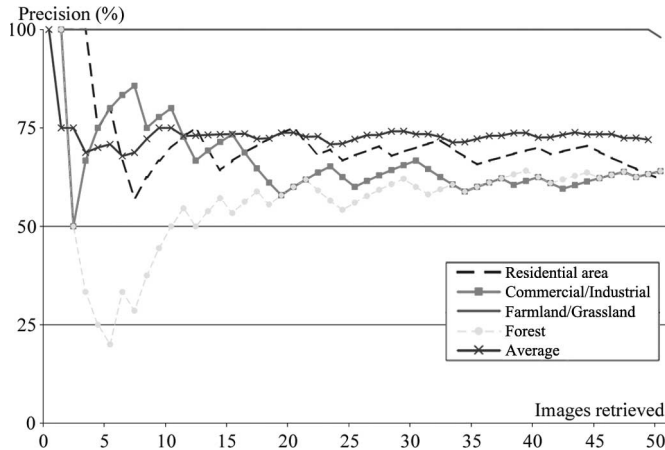


Fig. 8. Precision of semantic search methods. Images were retrieved using association rules that map image features to semantics. The relevance of each retrieved image was then evaluated by domain experts.

T_j^i for all semantics i available in our database and obtains the following normalized semantic significance vector:

$$\text{NSS}_j^i = \frac{T_j^i}{\sum_i T_j^i}. \quad (22)$$

The system then ranks images that have the highest NSS values for the semantic of interest and displays them to the users.

5) *Semantic Modeling Performance*: To evaluate the performance of the semantic modeling methods, we used an image database containing geospatial image tiles from three cities in Missouri. From these images, we selected 18% as training images labeled by experts to include Commercial & Industrial, Residential, Grassland & Cropland, and/or Forests. The performance of semantic modeling was evaluated by computing the precision of semantic search on a blind test using the remaining 82% of the image tiles. Each image was represented using a feature vector that includes features from the panchromatic, NIR, and visible spectra. These features were used for association rule mining and semantic search. Following the procedure described in this section 1055 association rules were extracted across the four semantics. These rules were then used to run the semantic search methods. The relevance of each retrieved images was evaluated by domain experts.

Fig. 8 shows the precision of the semantic search for the four semantics used in our search experiment. The average retrieval precision across the four semantics is 72%. The results also show that precisions vary among individual semantics.

VII. QUERY METHODS

A. Query by Example

The most basic method of query provided by our system is query by example. This mechanism allows an analyst to search for images that look visually similar to their query image. Underlying this type of query is the assumption that feature extraction algorithms can successfully extract features for which images that are visually similar are close in the feature space. As described in Section III, there are several different

categories of features used in our system. Each set of these features can be weighted by the user based on the usefulness of each set of features for the query being performed. For example, when performing a query containing a long stretch of highway, increasing the weight associated with the linear features will cause results from the corresponding indexes to exert more influence over the final ranked result set.

B. Hybrid Query

This query method is an extension of query by example described above. This technique provides an extension that allows the use of heterogeneous databases. Specifically, hybrid queries allow a user to perform a query by example and limit the results to a specific region surrounding the geographic and anthropogenic features from our GN module described in Section II. Users can choose from a list of features to select either a group of features—such as all schools—or a specific feature—such as Fairview School. In addition, the search radius surrounding these areas can be varied by the user. An example of this query method is “find an image that looks like this one and is within 3 km of Fairview School.” Fig. 9 shows the interface used to initiate a hybrid query, and Fig. 10 shows the results returned by a hybrid query.

C. Object Queries

The object query method differs from the two previously described. This technique allows users to query for specific objects of similar shape, size, and appearance within a database of objects that were automatically extracted from imagery using methods described in Section III-B1. Queries are posed to our system by manually annotating an object in an image. Using this selection, the object’s shape and appearance are used to query indexes to find similar objects. For example, by drawing an outline around an airplane, the system can identify other airplanes from the object database.

D. Multiobject Spatial Relationship Queries

Multiobject spatial relationship queries rely on the user to design the query spatial relationship. Users can either use a blank sketch area or an underlay image sample. For both methods, the user is provided with graphical user interface tools to draw ovals, rectangles, and free-hand polygons within the GeoIRIS client. Using an underlay image, users can trace the existing image objects, selecting the objects relative to the relationship. Once a set of objects has been designated by the user, the multiobject spatial relationship feature extraction is performed as described in Section III-B2. The signature generated is used to search in the spatial signature space, which is indexed using the EBS k -D tree. These queries are very powerful, as they can be incorporated with other query methods for robust retrievals. The range of query applications includes such applications as finding building complexes with particularly interesting spatial organization, or finding vehicles deployed in functional configurations (e.g., SAM sites).



Fig. 9. Example of a hybrid query being built. The query image is the region of interest within the white rectangle (i.e., the divided highway). Weights can be assigned to features of various indexes by adjusting the sliders, and the use of heterogeneous databases is demonstrated by specifying the constraint that results must be within 3 km of a school.



Fig. 10. Hybrid query results visualization of construction within 3 km of a school. The query image is shown in the upper left-hand corner. The current result is shown in the large center image, and the associated metadata are shown in the lower left-hand corner. A ranked list of the top relevant results is shown on the right.

E. Semantic Querying

Semantic queries rely on data mining rules to rank relevant images in the database. The relevance of each image is measured by the possibility functions shown in (19) of the decision rule antecedents for a particular semantic term. For example, given a semantic query, “retrieve images with Construction,”

the system will evaluate the possibilistic response of imagery to the antecedents of decision rules for construction. Semantic queries provide a higher level conceptualization of the underlying content-based features that support GeoIRIS. The semantic knowledge discovery brings the analysts and the system closer to a shared understanding of image content.

VIII. CONCLUSION AND FUTURE WORK

Mining relevant visual information that is valuable to the image intelligence community is a challenging task that is still in its infancy. We have presented the GeoIRIS system, which will enable scalable processing and retrieval of a large volume of data by automatically preprocessing and indexing satellite images. The main contribution of this paper is the design of the system framework as well as its underlying novel approaches in feature extraction, database indexing, information ranking, semantic modeling, and advanced queries. This framework has been developed using 0.6- and 1.0-m resolution imagery; although with minor modifications, it could be applied to higher resolution imagery. However, it is possible that some image characteristics, such as linear features, are unable to be extracted from images with lower resolutions. It is noteworthy to mention that many of the approaches, extended from our previous and current work [4], [15], [16], [25], [26], [29], [31], are unique and empirically proven to have reasonably good performance in both precision and efficiency.

We will continue to develop the GeoIRIS system by both extending the overall framework and adding novel algorithms for image mining, content indexing, and semantic development. We continually add imagery from all over the world into the database and expect to incorporate additional textual resources into the GN module. Based on the semantic modeling discussed in this paper, a knowledge sharing and exchange system will be built for the training of image analysts. We expect that the framework of GeoIRIS will be adapted by the image intelligence community to streamline information gathering and intelligence decision making from multimodal geospatial databases.

ACKNOWLEDGMENT

The authors would like to thank DigitalGlobe for providing QuickBird imagery from the RADII development dataset for use in this research.

REFERENCES

- [1] P. Agouris, J. Carswell, and A. Stefanidis, "An environment for content-based image retrieval from large spatial databases," *ISPRS J. Photogramm. Remote Sens.*, vol. 54, no. 4, pp. 263–272, Aug. 1999.
- [2] R. Agrawal, T. Imielinski, and A. Swami, "Mining association rules between sets of items in large databases," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, Washington, DC, May 1993, pp. 207–216.
- [3] R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval*. New York: ACM Press, 1999.
- [4] A. S. Barb, C.-R. Shyu, and Y. Sethi, "Knowledge representation and sharing using visual semantic modeling for diagnostic medical image databases," *IEEE Trans. Inf. Technol. Biomed.*, vol. 9, no. 4, pp. 538–553, Dec. 2005.
- [5] A. P. Berman and L. G. Shapiro, "A flexible image database system for content-based retrieval," *Comput. Vis. Image Underst.*, vol. 75, no. 1/2, pp. 175–195, Jul./Aug. 1999.
- [6] S. Chawla, S. Shekhar, W. L. Wu, and U. Ozesmi, "Modeling spatial dependencies for mining geospatial data: An introduction," in *Geographic Data Mining and Knowledge Discovery*, H. J. Miller and J. Han, Eds. New York: Taylor & Francis, 2001.
- [7] P. Ciaccia, M. Patella, and P. Zezula, "M-tree: An efficient access method for similarity search in metric spaces," in *Proc. Int. Conf. Very Large Databases*, 1997, pp. 426–435.
- [8] F. Coenen, "LUCS KDD implementation of CMAR (Classification Based on Multiple Association Rules)," Dept. Comput. Sci., Univ. Liverpool, Liverpool, U.K., 1958. [Online]. Available: <http://www.csc.liv.ac.uk/frans/KDD/Software/CMAR/cmar.html>
- [9] M. Datcu, H. Daschiel, A. Pelizzari, M. Quartulli, A. Galoppo, A. Colapicchioni, M. Pastori, K. Seidel, P. G. Marchetti, and S. D'Elia, "Information mining in remote sensing image archives: System concepts," *IEEE Trans. Geosci. Remote Sens.*, vol. 41, no. 12, pp. 2923–2936, Dec. 2003.
- [10] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, B. D. Q. Huang, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker, "Query by image and video content: The QBIC system," *Computer*, vol. 28, no. 9, pp. 23–32, Sep. 1995.
- [11] F. Fonseca, M. Egenhofer, P. Agouris, and G. Câmara, "Using ontologies for integrated geographic info," *Syst.—Trans. GIS*, vol. 6, no. 3, pp. 231–257, 2002.
- [12] *Geographic Names Information System*, Reston, VA: U.S. Geological Survey. Accessed Jan. 31, 2006. [Online]. Available: <http://geonames.usgs.gov>
- [13] *GEOnet Names Server*, Bethesda, MD: Nat. Geospatial-Intelligence Agency. Accessed: Jan. 31, 2006. [Online]. Available: <http://earth-info.nga.mil/gns/html/>
- [14] R. M. Haralick, K. Shanmugam, and I. Dinstein, "Textural features for image classification," *IEEE Trans. Syst., Man Cybern.*, vol. SMC-3, no. 6, pp. 610–621, Nov. 1973.
- [15] M. Klaric, C. R. Shyu, G. Scott, and C. H. Davis, "Automated object extraction through simplification of the differential morphological profile for high-resolution satellite imagery," in *Proc. Int. Geosci. and Remote Sens. Symp.*, Seoul, Korea, Jul. 25–29, 2005, pp. 1265–1268.
- [16] M. Klaric, C. R. Shyu, G. Scott, C. H. Davis, and K. Palaniappan, "A framework for geospatial satellite imagery retrieval systems," in *Proc. Int. Geosci. and Remote Sens. Symp.*, Denver, CO, Jul. 31–Aug. 4, 2006, pp. 2457–2460.
- [17] J. Li and R. Narayanan, "Integrated spectral and spatial information mining in remote sensing imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 42, no. 3, pp. 673–685, Mar. 2004.
- [18] W. Li, J. Han, and J. Pei, "CMAR: Accurate and efficient classification based on multiple class-association rules," in *Proc. ICDM*, 2001, pp. 369–376.
- [19] *MapServer*, Minneapolis, MN: Univ. Minnesota. Accessed: Dec. 13, 2005. [Online]. Available: <http://mapserver.gis.umn.edu/>
- [20] P. Matsakis, "Relations spatiales structurelles et interpretation d'images," Ph.D. dissertation, Inst. Recherche Inf. Toulouse, Toulouse, France, 1998.
- [21] P. Matsakis, J. M. Keller, O. Sjahputera, and J. Marjamaa, "The use of force histograms for affine-invariant relative position description," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 1, pp. 1–18, Jan. 2004.
- [22] V. E. Ogle and M. Stonebraker, "Chabot: Retrieval from a relational database of images," *Computer*, vol. 28, no. 9, pp. 40–48, Sep. 1995.
- [23] A. Pentland, R. W. Picard, and S. Sclaroff, "Photobook: Tools for content-based manipulation of image databases," in *Proc. SPIE—Conf. Storage and Retrieval Image and Video Databases II*, San Jose, CA, 1994, pp. 34–47. no. 2185.
- [24] M. Pesaresi and J. A. Benediktsson, "A new approach for the morphological segmentation of high-resolution satellite imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 39, no. 2, pp. 309–320, Feb. 2001.
- [25] G. Scott and C.-R. Shyu, "EBS k-d tree: An entropy balanced statistical k-d tree for image databases with ground-truth labels," in *Proc. Int. Conf. Image and Video Retrieval*, 2003, pp. 467–476.
- [26] G. Scott, M. Klaric, and C.-R. Shyu, "Modeling multi-object spatial relationships for satellite image database indexing and retrieval," in *Proc. Int. Conf. Image Video Retrieval*, 2005, vol. 3568, pp. 247–256.
- [27] N. Sebe, M. S. Lew, X. Zhou, T. S. Huang, and E. M. Bakker, "The state of the art in image and video retrieval," in *Proc. Int. Conf. Image Video Retrieval*, 2003, vol. 2728, pp. 1–8.
- [28] S. Seretto, Y. Rui, K. Ramchandran, and T. S. Huang, "A region-based representation of images in Mars," *J. VLSI Signal Process. Syst.*, vol. 20, no. 1/2, pp. 137–150, Oct. 1998.
- [29] A. K. Shackelford and C. H. Davis, "A hierarchical fuzzy classification approach for high-resolution multispectral data over urban areas," *IEEE Trans. Geosci. Remote Sens.*, vol. 41, no. 9, pp. 1920–1932, Sep. 2003.
- [30] A. Sheth, *Changing Focus on Interoperability in Information Systems: From System, Syntax, Structure to Semantics in Interoperating Geographic Information Systems*, M. F. Goodchild, M. J. Egenhofer, R. Fegeas, and C. A. Kottman, Eds. New York: Kluwer, 1999, pp. 5–30.
- [31] C. R. Shyu, A. Barb, and C. H. Davis, "Mining image content associations for visual semantic modeling in geospatial information retrieval and

indexing," in *Proc. Int. Geosci. and Remote Sens. Symp.*, Seoul, Korea, Jul. 25–29, 2005, pp. 5622–5625.

- [32] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, "Content-based image retrieval at the end of the early years," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 12, pp. 1349–1380, Dec. 2000.
- [33] A. W. M. Smeulders, T. S. Huang, and T. Gevers, "Special issue on content-based image retrieval," *Int. J. Comput. Vis.*, vol. 56, no. 1/2, pp. 5–6, 2004.
- [34] J. R. Smith and S.-F. Chang, "VisualSEEK: A fully automated content-based image query system," in *Proc. ACM Multimedia*, 1996, pp. 89–98.
- [35] T. Tanimoto, "An elementary mathematical theory of classification and prediction," IBM Corp., New York, 1958. Internal Rep.
- [36] D. Zhang and G. Lu, "Content-based shape retrieval using different shape descriptors: A comparative study," in *Proc. IEEE Int. Conf. Multimedia Expo.*, Aug. 2001, pp. 1139–1142.



Chi-Ren Shyu (S'89–M'99) received the M.S.E.E. and Ph.D. degrees from Purdue University, West Lafayette, IN, in 1994 and 1999, respectively.

Between 1999 and 2000, he was a Postdoctoral Research Associate with the Robot Vision Laboratory, Purdue University. In October 2000, he joined the Computer Science Department, University of Missouri, Columbia (MU), as an Assistant Professor. Currently, he is the Shumaker Associate Professor of Computer Science and an Adjunct Faculty with the Health Management and Informatics Department and

the School of Information Science and Learning Technologies at MU. He is the Founding Director of the Medical and Biological Digital Library Laboratory at MU and serves as a Chief Computer Engineer with the Family Physicians Inquiries Network. His research interests include geospatial information retrievals, biomedical informatics, medical imaging, computer vision, and pattern recognition.

Dr. Shyu received several awards including the National Science Foundation CAREER Award in 2005, MU College of Engineering Junior Faculty Research Award, and numerous teaching awards. Project sponsors for his researches include the National Geospatial-Intelligence Agency, National Science Foundation, National Institute of Health, U.S. Department of Education, and other organizations both for-profit and not-for-profit.



Matt Klaric (S'06) received the B.S. degree in computer science from the Saint Louis University, St. Louis, MO, in 2003. He is currently working toward the Ph.D. degree in computer science at the University of Missouri–Columbia.

In 2004, he joined the Medical and Biological Digital Library Research Laboratory and the Center for Geospatial Intelligence as a Research Assistant. His areas of research include geospatial content-based information retrieval, data mining, computer vision, and pattern recognition.



Grant J. Scott (S'02) received the B.S. and M.S. degrees in computer science from the University of Missouri–Columbia (MU), in 2001 and 2003, respectively. He is currently working toward the Ph.D. degree in computer science at MU.

He is currently a Database Programmer/Analyst with the University of Missouri System, in administrative enterprise IT applications. His research interests include high-dimensional indexing and content-based retrieval in biomedical and geospatial databases. Other research interests include computer

vision, pattern recognition, computational intelligence, databases, parallel/distributed systems, and information theory in support of media databases systems.

Mr. Scott is a member of the Medical and Biological Digital Library Research Laboratory and the Center for Geospatial Intelligence. He received an Outstanding Graduate Student Award from the MU College of Engineering in 2006.



Adrian S. Barb (S'04) received the Bachelor's degree in industrial engineering from the University of Bucharest, Bucharest, Romania, in 1990, and the Master's degree in business administration from the University of Missouri–Columbia (MU), in 2002. He is currently working toward the Ph.D. degree in computer science at MU.

He has worked as a Database Programmer Analyst with the Information Access and Technology Services, MU. His research interests include knowledge representation and exchange in content-based retrieval systems, semantic modeling and retrieval, capturing and predicting conceptual change in knowledge-based systems, ontology integration, and expert-in-the-loop knowledge exchange.



Curt H. Davis (S'90–M'92–SM'98) was born in Kansas City, MO, on October 16, 1964. He received the B.S. and Ph.D. degrees in electrical engineering from the University of Kansas, Lawrence, in 1988 and 1992, respectively.

Since 1987, he has been actively involved in the experimental and theoretical aspects of microwave remote sensing of ice sheets. He has participated in two field expeditions to the Antarctic continent and one to the Greenland ice sheet. From 1989 to 1992, he was a NASA Fellow with the Radar Systems

and Remote Sensing Laboratory, University of Kansas, where he conducted research on ice-sheet satellite altimetry. He is currently the Croft Distinguished Professor of Electrical and Computer Engineering and the Director with the Center for Geospatial Intelligence, University of Missouri–Columbia.

Dr. Davis received the Antarctica Service Medal from the National Science Foundation.



Kannappan Palaniappan (S'84–M'84–SM'99) received the B.A.Sc. and M.A.Sc. degrees in systems design engineering from the University of Waterloo, Waterloo, ON, Canada, and the Ph.D. degree in electrical and computer engineering from the University of Illinois, Urbana-Champaign, in 1991.

From 1991 to 1996, he was with the NASA Goddard Space Flight Center, working in the Laboratory for Atmospheres, where he coestablished the High-Performance Visualization and Analysis Laboratory. He developed the first massively parallel algorithm for satellite-based hurricane motion tracking using 64 K processors and invented the Interactive Image Spreadsheet system for visualization of extremely large image sequences and numerical model data over high-performance networks. Many visualization products created with colleagues at NASA have been widely used on television, magazines, museums, web sites, etc. With the University of Missouri, he helped establish the NSF vBNS high-speed research network, the NASA Center of Excellence in Remote Sensing, ICREST, and MCVL. He has been with UMIACS, University of Maryland, and worked in industry for Bell Northern Research, Bell Canada, Preussen Elektra Germany, the Canadian Ministry of Environment, and Ontario Hydro. His research interests include satellite image analysis, biomedical imaging, video tracking, level set-based segmentation, nonrigid motion analysis, scientific visualization, and content-based image retrieval.

Dr. Palaniappan received the highest teaching award given by the University of Missouri, the William T. Kemper Fellowship for Teaching Excellence in 2002, the Boeing Welliver Faculty Fellowship in 2004, the University Space Research Association Creativity and Innovation Science Award (1993), the NASA Outstanding Achievement Award (1993), the Natural Sciences and Engineering Research Council of Canada scholarship (1982–1988), and the NASA Public Service Medal (2001) for pioneering contributions to scientific visualization and analysis tools for understanding petabyte-sized archives of NASA datasets.