Searching satellite imagery with integrated measures

Ashok Samal\textsuperscript{a,}\textsuperscript{*}, Sanjiv Bhatia\textsuperscript{b}, Prasanth Vadlamani\textsuperscript{a}, David Marx\textsuperscript{c}

\textsuperscript{a}Department of Computer Science and Engineering, University of Nebraska-Lincoln, USA
\textsuperscript{b}Department of Mathematics and Computer Science, University of Missouri-St. Louis, USA
\textsuperscript{c}Department of Statistics, University of Nebraska-Lincoln, USA

\textbf{ARTICLE INFO}

Article history:
Received 28 July 2008
Received in revised form 18 January 2009
Accepted 27 January 2009

Keywords:
Image data mining
Remote sensing
Content based image retrieval
Geospatial analysis

\textbf{ABSTRACT}

Due to the advances in imaging and storage technologies, the number and size of images continue to grow at a rapid pace. This problem is particularly acute in the case of remotely sensed imagery. The continuous stream of sensory data from satellites poses major challenges in storage and retrieval of the satellite imagery. In the mean time, the ubiquity of Internet has resulted into an ever-growing population of users searching for various forms of information. In this paper, we describe the search engine SIMR—Satellite Image Matching and Retrieval system. SIMR provides an efficient means to match remotely sensed imagery. It computes spectral and spatial attributes of the images using a hierarchical representation. A unique aspect of our approach is the coupling of second-level spatial autocorrelation with quad tree structure. The efficiency of the web-based SIMR has been evaluated using a database of images with known characteristics: cities, towns, airports, lakes, and mountains. Results show that the integrated signature can be an effective basis for accurately searching databases of satellite based imagery.

© 2009 Elsevier Ltd. All rights reserved.

1. Introduction

An image has been aptly described as “worth a thousand words”. The extraction of information from an image in a precise, complete, and unambiguous manner is a complex and challenging task. The task becomes even more daunting due to the advances in imaging and storage technologies which have led to a rapid growth in the number and size of images. The problem of information extraction is particularly acute in the case of remotely sensed imagery. As more satellites go up in space and the spatial and temporal resolutions of the images increase, the amount of data being collected continuously is enormous. For example, the National Aeronautical and Space Administration (NASA) Earth Observing System (http://www.hq.nasa.gov/hpcc/insights/vol5/networks.htm) collects terabytes of data every day.

On one hand, the plethora of data makes the problem of extracting interesting and useful information from large repositories extremely difficult. On the other hand, the ubiquity of Internet has resulted into an ever-growing population of users searching for various forms of information on this type of data. Geospatial images are being produced and used in many applications, including hazard monitoring, drought management, commercial land use planning, agricultural productivity, forestry, tropical cyclone detection, and other intelligence and military applications. The range of users accessing these remotely sensed images varies from expert users such as meteorologists to novices such as farmers, trying to access and interpret these images, especially satellite images [1]. It remains a challenge to provide robust and effective techniques to mine information from these repositories and this challenge is the focus of this research.

The challenge of information mining in satellite images is harder compared to ordinary images due to the characteristics of satellite images. Typically, satellite images are available in different formats such as GeoTIFF and RGB. Some of these formats may contain georeferencing information that can be used as metadata for image annotation. The images may have varying number of bands or channels depending on the sensors used in the satellite. An important characteristic of these images is the source of data and the conditions under which the data has been acquired, in addition to the parameters of the acquisition sensor. The acquisition sensor parameters include band-specific radiometric maximum light level (or temperature range for IR bands), band-specific gain and level, and sensor altitude. The electromagnetic band range of data acquired by some satellites, such as LandsAT7, SPOT, QuickBird, and IKONOS, are well known. Some of the imagery may even come from high altitude aerial photography. In any case, the minimum data associated with the images includes meteorological and ephemeral parameters such as geodetic latitude/longitude, time of day, day of year, and observed or estimated general atmospheric conditions. The images may need

\textsuperscript{*}Corresponding author. Tel.: +1 402 472 2217.
E-mail address: samal@cse.unl.edu (A. Samal).

0031-3203/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2009.01.035
to be color-balanced across all channels and resolutions. Finally, images may have dead pixels to denote areas of no data in the image due to sensor aberrations.

In most of the current image repositories and retrieval systems, including “Google Images” (http://images.google.com), the semantics of an image are represented by human-supplied text annotations. This approach has several drawbacks. Since a large number of satellite images are being added to the repositories continuously, this process is both tedious and time consuming. This approach puts significant burden on the human annotator, which may result in inconsistent and incomplete indexing. A human annotator may inadvertently ignore or omit some relevant objects or latent features in an image during annotation leading to content mismatch. Similarly, an object in an image can be tokenized with many different names, but not all interpretations may be stored leading to language mismatch.

The limitations of manual annotation, coupled with the massive amount of data, create the need for an efficient retrieval system that is automated. The need is even more significant in the context of remotely-sensed imagery. The rate at which these images are being acquired from the satellites is extremely high. Many of these images may appear similar to the untrained eye and hence, are difficult to annotate well even by an expert. For example, Fig. 1 shows the aerial view of Brooksville, a town in Kansas. If someone looks at this image out of context, it is extremely hard to even point out the urban texture that is visible in the image (highlighted by red box). Obviously, it is difficult for non-experts to describe the image in terms of keywords or phrases. This underscores the need for an automated retrieval system in the domain of remotely-sensed imagery. In addition, it is important to exploit the inherent properties of the remotely sensed imagery. In particular, it would be beneficial to incorporate various domain specific or geospatial properties into the analysis.

The retrieval process is also highly dependent on the way a user specifies the query. The query can be specified in terms of some textual tokens if the images in the database are annotated. Another way to query the image repositories is to provide an example image and ask the system to retrieve information on the example image or retrieve similar images. The latter paradigm is known as content-based image retrieval or CBIR [2,3].

CBIR systems have found applications in many diverse domains including medicine, crime prevention, graphic design, publishing and advertising for targeted marketing, architectural and engineering design, and historical research. A geospatial CBIR system to assist in searching a large database of satellite images will be useful in spatial data mining or spatial surveillance. One can perform image information mining to discover potentially useful and hidden patterns from large geospatial imagery databases using such a system [4]. For example, in the context of military applications, one can search for specific patterns in the ground (e.g., airports or lakes). Agronomists and foresters can look for crop or tree diseases. Satellite-based images have proven to be useful in tracking natural calamities like droughts, cyclones, and tsunamis. The images are used to predict where and when natural hazards occur, and thus can be instrumental in mitigating their effects. Climatologists use satellite imagery to understand the short-term and long-term climate change and their effects [1].

In this paper, we describe the CBIR system named Satellite Image Matching and Retrieval System (SIMR) that is tuned specifically for satellite-based imagery. SIMR represents an image by a signature that consists of spectral and spatial attributes at different resolution levels using the quadtree data structure [5]. In addition, SIMR enhances the signature by computing spatial autocorrelation of the attributes at different levels. The spatial autocorrelation measure helps determine the spatial structures in an image at different levels, to provide an overall view of the image contents as well as the view of subareas in the image. SIMR uses the integrated signature in the retrieval process to match a query image to the images in the database. In addition to integrated signature, SIMR employs several distance measures to compute the similarity between two image signatures. It uses the computed similarity to derive a ranked list of images that are ordered on the basis of their distance from the query image.

We have evaluated SIMR using a large set of images with known image characteristics to verify its effectiveness. Furthermore, we have developed SIMR as a configurable search engine in which a user can choose the attribute set for matching. That is, any combination of spectral and spatial features can be selected by the user to compute distances between query and images in the database.

2. Background

The significance of image retrieval is well established and a large body of research already exists in this subject. Over the last several years, a number of CBIR systems have been developed. Some of these CBIR systems include SIMBA (Search IMages by Appearance) [6], WISE (Wavelet Image Search Engine) [7] and Blobworld [8]. Fu et al. have used image segmentation to identify objects in an image and then attempted to assign a weight of importance to those objects [9]. However, the segmentation is not perfect and may result in improper assignments, such as a yellow handle of a black-headed hammer being segmented into two objects. Qiu et al. have described a scheme to create clusters of images for retrieval based on color distribution along the visual gamut [10]. The main drawback of the technique is that despite using the visual gamut, the processing is done in RGB space. Torres et al. [11] have used a genetic programming framework to combine a number of similarity functions to perform content-based image retrieval. They have used shape similarity for comparison but report that the functions can use color and texture similarity as well.

Most of the existing image retrieval systems allow the specification of queries by using two techniques. The first of these techniques is known as query-by-example and forms the basis for CBIR systems. The second technique uses descriptive features in an image specified as text and is known as query-by-memory [12]. The CBIR systems are further classified into three categories based on the underlying search mechanism: histogram-based systems, color layout-based systems, and region-based systems [13].

A full discussion of all the CBIR systems is beyond the scope of this paper. However, since SIMR is based on Robust Image Search Engine (RISE), we briefly describe it here. RISE is a CBIR system that exploits the ideas from JPEG compression to build an index for images in the repository. The index contains the signature for each image that is computed by applying the Discrete Cosine Transform (DCT) to each image [14]. RISE computes the image signature by first dividing it
into blocks of 8x8 pixels. This division into 8x8 pixel blocks forms the basis for the quadtree structure where the leaf nodes contain the DC value from the DCT-encoded blocks. Each non-leaf node in the quadtree contains the aggregates of the values contained in its child nodes. The entire quadtree is stored as a single record in a relational database table.

Upon receiving a query in the form of an image, RISE partitions the query image into the quadtree structure. The signature from the query is then compared with the signature of images in the database to obtain the closest matches. RISE is a good general-purpose image search and retrieval engine using the color distribution in different parts of images to narrow down the matches. However, it does not perform as well for geospatial imagery because geospatial imagery is not very well characterized by color. In the next subsection, we examine the issues related to geospatial imagery.

2.1. Retrieval of geospatial imagery

Geospatial data, including remotely sensed imagery, are being assembled at a very rapid rate. The large volume of data can be analyzed to derive useful information if we know the properties of the regions represented in this data [15,16]. This analysis cannot be performed using the generic image retrieval techniques such as the ones used in RISE due to their reliance on color layout or distribution. Such limitation of generic image retrieval systems was the main motivation behind the development of the Image Characterization and Modeling System (ICAMS) [17].

ICAMS was developed to provide scientists with a tool to visualize, characterize, and analyze remote sensing images from NASA's Earth Observing System (EOS) [17]. It receives its input from the user in the form of a query image and a region of interest within this image using a graphical interface. In addition, ICAMS allows the user to characterize the retrieval with a wide range of spatial measures (fractal properties, spatial autocorrelation, and textural measures) as well as non-spatial measures (mean, mode, median, variance, and histogram). It computes these measures for six levels of quadtree decomposition of the image.

ICAMS identifies regions in an image that are of interest to the user and finds similar occurrence of those regions of interest in a database of remotely sensed imagery [18]. It achieves this by calculating the characteristics of different regions of an image. Then, it compares the calculated values to those in the database of image signatures.

The computation of image signature is a key step in both ICAMS as well as RISE even though the signature itself is completely different, both in terms of attributes and underlying computation methods. RISE builds the signature in spectral domain using color distribution while ICAMS does the same in spatial domain using statistical autocorrelation. Our search engine, SIMR, builds upon the strengths of both RISE and ICAMS. SIMR uses a hierarchical structure to represent the attributes of the images and uses spectral as well as spatial properties of images. In addition, it defines and uses second order spatial attributes to represent higher-level organization.

Another area that has influenced the image retrieval is the classification of remotely-sensed imagery. Blanzieri and Melgani have devised a variant of k-nearest neighbor classifier using the Support Vector Machine classification principles [19], specifically the maximal margin principle combined with locality. It exploits the geometric configuration of the k nearest training samples in the decision process using the effective maximal margin. It uses the locality information to tighten the bound on the generalization error. Camps-Valls et al. have described the multi-temporal classification of remote sensing imagery to integrate heterogeneous sources of information using kernel methods [20]. Kernel methods define a mapping function from the attribute space of input data set to the feature space. The mapping function is non-linear with respect to the attribute space. The problem to be solved is labeled as multitemporal classification and change detection. The classification is performed at the time of first observation and later, the pixels that have changed are determined. Mitrakis et al. have described a self-organizing neuro fuzzy multilayered classifier for land cover classification of multispectral images, [21]. The model structure is determined by a structure learning algorithm with feature selection capabilities. The parameters of the models are optimized through a genetic algorithm. The system is used to classify the landcover wetland and agricultural zones in a lake-wetland ecosystem in Greece using IKONOS imagery.

Myint used different fractal approaches to characterize features of land-cover data in high resolution imagery [22]. He explored the approaches based on isarithm, triangular prism, and variogram features for image classification, and compared them with spatial autocorrelation techniques (Moran's I and Geary's C), simple standard deviation, and mean of selected features. He concluded that the fractal approaches are not sufficient by themselves to extract texture features or to identify different land-use and land-cover classes in remotely sensed imagery. Myint and Lam [23] found the lacunarity approaches to be more effective than fractal triangular prism, spatial autocorrelation, and spectral band approaches to classify urban images.

2.2. Spatial statistics

The techniques in spatial statistics study spatial entities using their topological, geometric, or geographic properties. These techniques have been instrumental in a number of disciplines, including geography, biology, and environmental sciences. One of the characteristics of geospatial data (spatial data over the Earth's surface) is that they follow some laws of geography. Of particular interest in our context is Tobler's First Law of Geography which states that “Everything is related to everything else, but near things are more related than distant things” [24]. This law manifests in the form of spatial autocorrelation.

Formally, spatial autocorrelation is defined as a scale-dependent statistic that indicates the degree of clustering, randomness, or fragmentation of a pattern [25]. Spatial autocorrelation defines the correlation among values of a single variable that comes from the geographic arrangement of the areas in which these values occur. This helps in measuring the similarity of objects within an area, the degree to which a spatial phenomenon is correlated to itself in space, and the nature and strength of the interdependence. In this research, we use Moran's I and Geary's C indices, two of the most commonly used spatial autocorrelation measures.

Moran's I statistic: Moran introduced the first measure of spatial autocorrelation in order to study stochastic phenomena, which are distributed in space in two or more dimensions [17]. This particular statistic is designed to measure the strength of correlation between the observations as a function of the distance separating them. Moran's I parameter for a given distance $\delta$, given by $I(\delta)$, is computed by dividing spatial co-variation with total variation [18]:

$$I(\delta) = \frac{n \sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij} (x_i - x_{mean}) (x_j - x_{mean})}{\left( \sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij} \right) \left( \sum_{i=1}^{n} (x_i - x_{mean})^2 \right)}$$

where $n$ is the number of observations, $w_{ij} = 1$ if locations $i$ and $j$ (two different cells or points) are within a distance $\delta$ from each other and 0 for all other cells; $x_i$ is the $i$th observation or measured value, and $x_{mean}$ is the average value of the observations. The range of values for Moran's $I$ parameter is $+1.0$ to $-1.0$, where $+1$ means a strong or perfect positive spatial autocorrelation, a value of $0$ means a random pattern, and $-1$ indicates a strong negative spatial autocorrelation.
Geary’s C index: This spatial statistic is built on the concept of a paired comparison of juxtaposed map values [17, 18, 26]. It uses the sum of squared differences between pairs of data values as its measure of co-variation and is calculated as follows

\[ C = \frac{(n - 1) \sum_{i,j} w_{ij} (x_i - x_j)^2}{\sum_{j}(x_j - x_{\text{mean}}) \times \sum_{i}(x_i - x_{\text{mean}}) \times (2 \sum_{i,j} w_{ij})} \]  

Geary’s C values vary from 0 to +2, where a zero indicates positive spatial autocorrelation and value greater than one indicates a negative spatial autocorrelation. A Geary’s C value of one (+1) suggests that no spatial autocorrelation is present.

It has been reported that there is a strong relation between different spatial autocorrelation measures [22]. However, the two indices are not identical. Therefore, it is conceivable that in some situations, one index is able to capture the spatial structure better than the other and hence be more useful in matching. Therefore, we have chosen to use both Moran’s I and Geary’s C. We note that in a practical system, one is more likely to choose Moran’s I only since it is more widely used and can be generalized to higher dimensions.

### 2.3. Fractal statistics

Satellite imagery relates to surface of the Earth and many geological features that can be modeled using fractals. An important property of fractal geometry is that true fractals display self-similarity and the shape of a fractal object or surface is independent of the scale at which it is measured [27]. Also, fractals are considered as a potentially robust method for understanding landscape complexity because they conveniently describe many of the irregular and fragmented patterns found in nature [17]. The frequency distributions of fragments, faults, mineral deposits, oil fields, and earthquakes are some of the examples that are fractal. Fractal concepts can also be applied to continuous distributions like topography, which may constantly change due to erosion and sedimentation [28]. Further, fractal features such as fractal dimension and lacunarity have been shown to be useful in characterizing features in urban areas in satellite imagery [29].

We use fractal dimension and lacunarity in SIMR since they are central to characterizing remotely sensed imagery. For example, urban features that are characterized by their texture are more accurately retrieved using these indices [18]. The two selected features are explained below.

**Fractal dimension:** Fractal dimension is used to represent the spatial complexity of point patterns, lines, and surfaces formed by the pixels in an image. The fractal dimension varies from 0.0 to 1.0 for a point pattern, from 1.0 to less than 2.0 in case of lines, and from 2.0 to less than 3.0 for surfaces. Further, higher the spatial complexity of a line or surface, higher is its fractal dimension. An efficient method to compute the fractal dimension (d) of a spatial grid structure is described in [18, 25, 27, 29]. The fractal values are computed through surface areas, defined by triangular prisms, using various sizes of measuring grids. The triangular prism surface area (TPSA) method constructs triangles using sets of four adjacent pixels as corners for the base of the prism. The intensity values for corner pixels are used to establish height at each corner, and triangles are formed by connecting these corner values to the mean value of the four pixels at the center of the array. The average of the pixel values located at the four corners of the triangular prism defines the height of the apex of the prism.

To compute the fractal dimension, SIMR first computes the total surface area computed at different scales. It begins at scale = 2 and computes the areas of all triangles formed by 2×2 arrays of pixels. It computes the total surface area by taking the sum of these. Similarly, it computes the total surface area at scale = 3 by considering 3×3 arrays of pixels. In this manner, we can compute the total surface area at different scales. The fractal dimension (d) is computed as

\[ d = 2 - s \]  

where s is from the slope of the regression line between the surface area and the scale.

**Lacunarity:** Lacunarity is a scale-dependent measure that is related to the spatial distribution of magnitude of gaps or holes [18, 25, 30]. When gaps of same size occur at regular intervals, the lacunarity is said to be low. In this case, the geometric objects are deemed to be homogeneous. When the gap sizes are variable, the objects have high lacunarity and the objects are deemed heterogeneous [30]. Lacunarity is also considered to be a measure of texture uniformity. The level of lacunarity is estimated by the variance, or texture, of gap sizes within the spatial delineation of geometric objects [25]. Lacunarity is computed by using a sliding box approach [31]. Starting with the upper left corner, an r×r box is placed over the image. If the minimum and maximum pixel values in the gliding box are u and v, respectively, the relative height of the column is

\[ n_r(i, j) = v - u - 1 \]  

This is computed at each pixel, as we move the box over different parts of the image. The mass M is computed as the sum of the \( n_r \) over the entire image. The expression for measuring lacunarity from gray-scale images is described in [30] and is given below.

\[ l(r) = \frac{\sum M Q(M, r)}{\left(\sum M Q(M, r)^2\right)\frac{1}{2}} \]  

Here Q(M, r) is the probability function that is computed by dividing \( n(M, r) \), the number of sliding boxes with radius r and mass M by the total number of boxes for a given r. It should be noted that there are multiple ways to compute lacunarity; see [23] for a comparison of the gray-scale approach and the spatial autocorrelation approach.

In this section, we summarized the related work in the automated retrieval of remotely sensed imagery. We also briefly described some of the spatial and fractal statistics used in our research. The details of our approach are described next.

### 3. Research methodology

#### 3.1. Overview of approach

The main objective of our research is to develop an accurate measure to compute the similarity between two remotely sensed images. We use an integrated features approach in which we use spectral and spatial properties of the images at different levels of hierarchy. One of the unique features of our approach is the use of higher-level spatial autocorrelation properties. Our working hypothesis is that these images are representations over the geographic space and hence must follow the laws of geography. Fig. 2 shows the schematic of our approach.

Our database consists of a set of images and their pre-computed signatures. When a user presents a query image, we compute its signature and match it with the signatures of the images in the database to derive match scores. Based on the scores, we present the images that are most similar to the query image.

In the current implementation of SIMR, we have assumed that the images are 512×512 pixel gray scale images, at the same scale of resolution. If the image is an RGB image, we consider only the luminance component in the color space. Furthermore, we focus on the degree of match between the images as a whole. In the current implementation, we do not match an image to parts of another image.
3.2. Hierarchical representation

We organize the features in our images in the form of a quadtree. Quadtrees are hierarchical data structures that have been used in a variety of applications to compress the image. The motivation for using quadtrees is briefly explained below.

In the geospatial domain, the concept of scale is a significant factor. Many processes occur at different scales and hence, manifest in different levels of detail in the image. Thus organizing the properties in a hierarchical manner can lead to more efficient matching. Our long-term goal is to be able to match any part of a query image with any part of a database image. Using quadtree representation also facilitates this endeavor. Fig. 3 shows a remotely sensed image and a superimposed quadtree structure. For the ease of viewing, not all the levels of the quadtree are shown. The corresponding quadtree is also shown (Fig. 3(b)).

The top level of the quadtree is denoted as Level 0 and has only one block. The next level (Level 1) has four blocks of size 256×256 pixels. At the lowest level (Level 6) we have 64 blocks of size 8×8 pixels. We compute the properties of the image for each block at each level of the quadtree.

3.3. Image attributes

The features of the images form the basis for comparing them for their similarity. We use two types of features for our approach: (a) spectral and (b) spatial. We include fractal features as a part of spatial features in our research. All of the features are computed for each block at different levels of the quadtree.

Spectral attributes: Spectral attributes encode the reflectance of the corresponding areas of the Earth’s surface. A multispectral image has multiple bands corresponding to different wavelengths. In this research, we only deal with the panchromatic images in which there is only one band which represents the brightness of the area. Additional bands of image can be added seamlessly to our system directly. We compute the spectral property for the 8×8 pixel blocks first and then build the attributes for the higher levels in the quadtree, i.e., the average values of the four child nodes are stored in their parent node. Fig. 4 shows the average brightness ($L^*$) values for different levels of the quadtree for the image shown in Fig. 3.

Spatial attributes: As described in Section 2, we use two spatial autocorrelation measures: (a) Moran’s I and (b) Geary’s C indices. Apart from providing an indication of the type and degree of spatial
autocorrelation in the image, these second order statistics reflect the differing spatial structures of smooth and rough surfaces. We use a 4-neighbourhood for computing the spatial autocorrelation measures. Unlike the spectral attributes, one cannot propagate the autocorrelation measures from the values at lower levels. This is because these attributes are based on global statistics that do not average well over agglomerated areas. Hence, these are calculated at each level of the quadtree separately. Fig. 5 shows the Moran’s I values for different levels of the quadtree for the image shown in Fig. 3.

Fractal attributes: Fractal properties are the third set of attributes used in our approach. We use two fractal attributes: (a) fractal dimension and (b) lacunarity. Fractal dimension measures the geometric complexity of an image, while lacunarity is its counterpart dealing with gaps. As with spectral attributes, the fractal dimension and lacunarity are computed at each level of the quadtree. Fig. 6 shows the fractal dimensions for the image in Fig. 3 at first three levels of the quadtree computed using the approach described in Section 2.3.

3.4. Higher level spatial autocorrelation measures

As described above, we derive the spectral, spatial, and fractal attributes of the image at each node in each level of the quadtree. In order to get even greater understanding of the spatial structure, we compute the spatial autocorrelations in the attributes space. We compute Moran’s I and Geary’s C indices at each level of the quadtree for each of the hierarchical attribute structures. That is, we compute the autocorrelation of the five feature values (average luminance, Moran’s I index, Geary’s C index, fractal dimension and lacunarity) at each level of the hierarchy. We call this second level spatial autocorrelation (SLSA). The SLSA measures are computed for each level of the quadtree except the top two levels, since those levels do not have sufficient number of blocks (Level 0 has one and Level 1 has only four blocks). Thus at each level of the quadtree for each attribute, we compute the spatial autocorrelation.

For each level in the quadtree there is a vector of 10 properties (five each for Moran’s I and Geary’s C indices) that represent the SLSA
scores for that level. These provide a higher level characterization of the image that may prove to be useful to differentiate between the images. Table 1 shows the SLSA scores for the image shown in Fig. 3 for different levels of the quadtree using Moran’s I index.

### Table 1

<table>
<thead>
<tr>
<th>Level</th>
<th>Block size</th>
<th>No. of blocks</th>
<th>(L^*)</th>
<th>(I)</th>
<th>(C)</th>
<th>(d)</th>
<th>(l)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>512×512</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1</td>
<td>256×256</td>
<td>4</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>128×128</td>
<td>16</td>
<td>0.33655</td>
<td>0.16085</td>
<td>0.30942</td>
<td>0.34273</td>
<td>0.43259</td>
</tr>
<tr>
<td>3</td>
<td>64×64</td>
<td>64</td>
<td>0.31382</td>
<td>0.08709</td>
<td>0.37019</td>
<td>0.18387</td>
<td>0.07809</td>
</tr>
<tr>
<td>4</td>
<td>32×32</td>
<td>256</td>
<td>0.29244</td>
<td>0.18513</td>
<td>0.38110</td>
<td>0.10123</td>
<td>0.10594</td>
</tr>
<tr>
<td>5</td>
<td>16×16</td>
<td>1024</td>
<td>0.23435</td>
<td>0.14963</td>
<td>0.39861</td>
<td>0.02962</td>
<td>0.03176</td>
</tr>
<tr>
<td>6</td>
<td>8×8</td>
<td>4096</td>
<td>0.37365</td>
<td>0.06976</td>
<td>0.30545</td>
<td>0.02953</td>
<td>0.01205</td>
</tr>
</tbody>
</table>

\(L^*\): average brightness; \(I\): Moran’s I index; \(C\): Geary’s C index; \(d\): fractal dimension; \(l\): lacunarity.

### 3.5. Problem formulation

The central problem at hand is to determine the degree of similarity (or dissimilarity) between two remotely sensed images: one a query image and the other an image in the database. All our images are represented by a hierarchical structure (quadtree). A given geospatial image \(G\) can be represented using a level quadtree, where Level 0 corresponds to the whole image and Level \(r\) corresponds to a single pixel. For practical purposes, the smallest level of analysis is usually larger than a single pixel and often, it may be 8×8 or 16×16 pixels. In SIMR, we have limited the lowest level to 8×8 pixels. The number of blocks at a given level is \(4^r\).

At each level of the quadtree, we compute the spatial (including fractal) and spectral attributes corresponding to each block. In addition, we compute the higher level spatial statistics for the entire level. Thus, an image \(G_0\), with \(n\) levels, is represented as \(G_0 = [I_0, I_1, \ldots, I_n]\), where the representation at Level \(i\), \(I_i\), is given by \(I_i = [A_{i,k}, A_{i,j}, S_{i,k}, F_{i,k}]\). \(p_i\) is the set of attributes for the block \(b\) at Level \(i\) and \(A_{i,k}\) gives the SLSA (high level) features for Level \(i\). \(A_{i,k}\) is given by \(A_{i,k} = [P_{i,k}, S_{i,k}, F_{i,k}]\), where \(P_{i,k}\), \(S_{i,k}\), and \(F_{i,k}\) represent the spectral, spatial, and fractal attributes for the block, respectively.

Thus

\[
P_{i,j} = [p_{i,j,1}, p_{i,j,2}, \ldots, p_{i,j,m_p}] \tag{6}
\]

where \(p_{i,j,m}\) is the \(r\)th spectral attribute for block \(j\) at Level \(i\), and \(m_p\) is the total number of spectral features. Similarly,

\[
S_{i,k} = [s_{i,k,1}, s_{i,k,2}, \ldots, s_{i,k,m_s}] \quad \text{and} \quad F_{i,k} = [f_{i,k,1}, f_{i,k,2}, \ldots, f_{i,k,m_f}] \tag{7}
\]

where \(m_s\) and \(m_f\) represent the total number of spatial and fractal attributes, respectively.

The problem of matching now can be defined as follows. Given a query image \(G_q = [I_q, I_{q1}, \ldots, I_{qn}]\) and a database image \(G_d = [I_d, I_{d1}, \ldots, I_{dn}]\), the goal is to define a similarity function \(\psi(G_q, G_d)\) that will compute the similarity between the two images.

### 3.6. Similarity computation

The computation of similarity between two given images is the central step in the retrieval of the best matches for a given query image. The similarity between two images is determined by comparing their integrated signatures represented by a set of attributes (spectral, spatial, fractal, and SLSA) at multiple levels of hierarchy in the form of a quadtree. We propose two approaches to compute the similarity between two images: (a) a direct distance approach and (b) a progressive deepening approach.

**Direct distance approach:** In this approach, we compute the distance between the images as the Euclidean distance between the feature vectors of the two images at all levels combined. Since we use four different types of features, it is possible to define distances of each type separately and combining some, or all, of them. The spectral \((d_p)\), spatial \((d_s)\), fractal \((d_f)\), and SLSA \((d_m)\) distances between the images are defined as follows.

\[
d_p(l_q, l_d) = \sqrt{n \sum_{i=0}^{l_q} \sum_{j=0}^{l_d} \sum_{k=0}^{4^n} (p_{q,i,j} - p_{d,i,j})^2} \tag{9}
\]

\[
d_s(l_q, l_d) = \sqrt{n \sum_{i=0}^{l_q} \sum_{j=0}^{l_d} \sum_{k=0}^{4^n} (s_{q,i,j} - s_{d,i,j})^2} \tag{10}
\]

\[
d_f(l_q, l_d) = \sqrt{n \sum_{i=0}^{l_q} \sum_{j=0}^{l_d} \sum_{k=0}^{4^n} (f_{q,i,j} - f_{d,i,j})^2} \tag{11}
\]

\[
d_m(l_q, l_d) = \sqrt{n \sum_{i=0}^{l_q} \sum_{j=0}^{l_d} \sum_{k=0}^{4^n} (m_{q,i,j} - m_{d,i,j})^2} \tag{12}
\]

Note that the spectral, spatial, and fractal distances sum over the feature values for all blocks at each level of the quadtree. For the SLSA distance, we have only one value (for each spatial attribute) at each level of the quadtree. Various combinations of the attributes can be used to compute the distance between two images in a straightforward manner. For example, if we use spectral and spatial attributes together, we can compute the distance \((d_{ps})\) between the query image \(l_q\) and a database image \(l_d\) as

\[
d_{ps}(l_q, l_d) = \sqrt{n \sum_{i=0}^{l_q} \sum_{j=0}^{l_d} \sum_{k=0}^{4^n} (p_{q,i,j} - p_{d,i,j})^2 + \sum_{k=0}^{4^n} (s_{q,i,j} - s_{d,i,j})^2} \tag{13}
\]

**Progressive deepening approach:** In this approach, the lower levels of the image representations are considered only if the attributes at the higher levels are similar. We start by computing the distance between the two images at the top level. If the distance is greater than a predefined threshold, the database image is not considered any further. Only those images that survive this “cut” are used for similarity computation at the next level (Level 1). Again, the images whose distance is greater than a threshold at Level 1 are removed from consideration and we proceed to the next level. The process is repeated at each level in the quadtree structure until we reach the
Fig. 7. Two different images with same value of spatial autocorrelation.

Fig. 8. Sample images in our database.

lowest level, or the number of images considered at a level is zero. We normalize the distances (by the number of nodes) and hence, use the same threshold value at all levels. The value of the threshold is determined experimentally. However, it is possible to set it using a statistical approach; this is left for future work. The images that are eliminated at the same level are ranked by their Euclidean distance. The images with smaller Euclidean distance from the query image are ranked higher than the images with larger distance.

This approach is motivated by the nature of the quadtree representation, in which each level (except the lowest level) is an aggregation. The highest level of the quadtree represents an overall average for the whole image. In some cases, it is possible for two images which are not similar to each other in content but may have similar average values. Fig. 7 shows an example of two images with very similar values of spatial autocorrelation ($I = 0.92$). Comparing them at the highest level would give such images higher preference than images with greater similarity that manifests at lower levels. The progressive deepening approach provides a way to balance this consideration by using the signatures in different sub-regions of the images.

4. Implementation and results

In this section, we describe the implementation of our search engine and the results of its evaluation. We compiled a database of remotely sensed imagery with known characteristics for evaluating the performance of the system. The search engine is implemented as a web based system with a user friendly GUI for the application. The system is implemented in Java (Version 1.5.0) and is hosted on an Apache Tomcat web server. The performance of image operations was enhanced by the use of JAI (Java Advance Imaging) tools [32].

4.1. Image database

We developed a database of images for our research by selecting images from the TerraServer collection (http://www.terraserver.com). TerraServer contains a large collection of aerial photos and satellite images. In order to systematically evaluate the algorithms, we selected images with known characteristics. We compiled images from five different classes of terrain types: (a) airports, (b) lakes, (c) mountains, (d) large cities, and (e) small towns. We collected 100 images from each class at the same scale of resolution. We only used gray scale images and performed histogram equalization on all images to remove any inconsistencies. All the 500 images are manually classified prior to running the queries to enable the system to compute precision and recall values against the ground truth. Fig. 8 shows a sample image from each of the classes.

4.2. Features database

We store the features of each image in a relational database. The main advantage of using a relational database is that no conventional programming is required to design a data storage and retrieval system irrespective of the complexity of the database. As described in Section 3, each image in the features database is associated with a number of attributes of different types at different levels of the quadtree. Table 2 shows the number of features of each type that is stored for an image at different levels of the quadtree. To simplify the structure of the database, we do not represent each of the attributes as a separate column in the database. Instead, we combine and store values of each attribute of every level of quadtree in one separate column. However, we store each of the SLSA values in a separate column since there are only a few of them. In order to minimize the number of columns in the database, we concatenate
the features at a level into a string. During retrieval time, the string is retrieved from the database, and the feature values are extracted from the string prior to distance computation.

### 4.3. Evaluation

Evaluation of the CBIR systems is challenging since it is difficult to get an accurate description of the images in the database. This is particularly true in the case of satellite imagery since it is difficult to interpret an image without some level of expertise in the domain. We used 12 random images from each of the five classes in our database as query images for evaluation. We summarize the results by examining the top 10 retrieved images using standard measure of precision [13] defined as:

\[
\text{Precision} = \frac{|A \cap B|}{|B|}
\]  

(14)

where \(A\) is the set of relevant images in the database and \(B\) is the set of images retrieved in response to the query. Precision measures the degree of relevance of the retrieved images. Ideally, the system should retrieve all the relevant images and the value of precision in this case would be 1. Table 3 shows the summary of the results using total Euclidean and progressive deepening approach. In order to better understand the contributions of different types of attributes (spectral, spatial, and fractal), we ran the queries with several different combinations: (a) spectral, spatial and fractal (RSF) and (b) spectral, spatial, fractal, and hierarchical (RSFH). Table 3 also shows the performance of the progressive deepening approach.

Table 3 shows that using combinations of features result in better performance than using individual features in isolation. If only one set of features is used, spatial features worked the best and spectral features the worst. SLSA features performed better in precision scores. Overall, the system performed quite well with a high rate of precision. The best performance of the system was realized when spectral, spatial, and fractal, and SLSA features are used. The precision scores have been computed using the five datasets described earlier. They show patterns similar to the case of the total Euclidean approach. Combinations of features work better than individual ones. However, fractal features perform better than spectral and spatial. Overall, the results with the combination RSFH perform the best in most of the cases, except for the case of Lake and City imagery. This seems plausible as the Lake imagery is mostly flat in spatial as well as spectral domain and the addition of second level spatial attributes does not provide extra information. The City imagery contains textures to show houses, grassy areas, pools, trees, roads, and other features. This makes a comparison between images a very hard problem due to a large variation in both spatial and spectral domain.

Overall, Euclidean distance performs better than progressive deepening approach. The average precision of all classes with different combination of features for Euclidean distance is 0.703 while that of progressive deepening approach is 0.668. However, the progressive deepening approach works marginally better for airport images (0.588 vs. 0.573). Similarly for the feature combinations, the Euclidean distance approach worked better than progressive deepening approach in every case except when spectral only features are used (0.540 vs. 0.498).

Among the five classes of images, our system performed the best for lakes; Euclidean distance got a precision of 0.99. Somewhat surprisingly, airports were retrieved least successfully. Airport scenes have fairy unique structures (long and narrow elements, i.e. runways, surrounded by green areas and some buildings, i.e. terminal). One reason for this is that the airports have different structures depending on the size of the airport leading to imperfect matches. Human observers, even untrained ones, find airports relatively easily in a scene.

Comparison with ICAMS and RISE: As described before, SIMR is based on the principles developed in ICAMS and RISE. Therefore, a performance comparison between the systems would be logical. RISE uses color in perceptual space as the primary basis to match imagery. Since satellite imagery is devoid of color, or constituted of pseudocolor, RISE is not efficient in matching such imagery. Therefore, the precision and recall figures for RISE are significantly lower than SIMR. As for ICAMS, the authors do not present comprehensive performance measures as we have in this paper. The results are presented with respect to two queries: (a) a land/water contrast area and (b) a residential area. Actual precision and recall measures are not available, nor are the comparisons performed with different classes of satellite imagery. In the case of land–water contrast, of
the top five matches, in the best case four of them were matches. This was also the case for the residential area search. While performance summary with a larger number of images is not presented, we expect that in the best case, ICAMS would achieve the performance of RSF matching (shown in Table 3) that uses spectral, spatial and fractal features. This performance can be reached assuming that ICAMS
uses all levels of the quadtree. In summary, SIMR outperforms both ICAMS and SIMR, the two systems from which SIMR has drawn on.

4.4. Sample queries

Fig. 9 shows a sample query image and Figs. 10–12 show the results of queries with only spectral, fractal and spatial features, respectively. The results support the conclusions in Section 4.3. When only the spectral features are used, the results are inconsistent. Spatial features are generally more consistent. Fig. 13 shows the results of the query when all the features are used. It shows that eight of the retrieved images are city images. Overall, this combination of features works the best.

5. Summary and future work

In this paper, we have developed a retrieval engine to efficiently match remotely sensed imagery using a quadtree-based approach. We compute spectral ($L^*$), spatial (Moran’s I parameter and Geary’s C index) and fractal (lacunarity and fractal dimension) attributes of each image at different levels of hierarchy. The signature of an image consists of these attributes at all levels of the hierarchical representation. One of the unique aspects of our approach is that we use the hierarchical representation to derive second level spatial autocorrelation measures of the attributes. We have examined two different approaches to compute the similarity between two images using this representation. In the direct distance approach, we compute the Euclidean distance between two signatures. In the direct distance approach, we compute the Euclidean distance between two signatures. In the direct distance approach, we compute the Euclidean distance between two signatures. In the direct distance approach, we compute the Euclidean distance between two signatures. In the direct distance approach, we compute the Euclidean distance between two signatures. In the direct distance approach, we compute the Euclidean distance between two signatures. In the direct distance approach, we compute the Euclidean distance between two signatures. In the direct distance approach, we compute the Euclidean distance between two signatures.

While our system shows the effectiveness of the integrated signature approach, the research can be extended along many different directions. They relate both to the operational aspects of a system that deals with images in this domain as well as the conceptual aspects. Our current approach is to match a query image as a whole with whole images in the database. SIMR can easily be extended to match a whole image to a part of an image or match parts of two images. We can also extend the matching process to compare imagery at different resolutions. Currently, we assume that the images are aligned at the boundaries. In real-life, images are representations of the geographic space that are captured at arbitrary locations and may not be tied to any alignment grid. We would also like to add support for GeoTIFF format in the engine that can allow SIMR to resolve some queries based on specified locations. Use of other distance functions, e.g., Euclidean distance with a covariance matrix and normalization of the indices may also improve the performance of our approach.

References


About the Author—ASHOK SAMAL is an Associate Professor in the Department of Computer Science and Engineering at the University of Nebraska-Lincoln. His research interests include image analysis including biometrics, document image analysis and computer vision applications. He has published extensively in these areas.

About the Author—SANJIV BHATIA is an Associate Professor in the Department of Computer Science and Mathematics at the University of Missouri at St. Louis. His research interests include computer vision and image analysis.

About the Author—PRASANTH VADLAMANI has completed his MS in Computer Science at the University of Nebraska-Lincoln.

About the Author—DAVID MARX is a Professor in the Department of Statistics at the University of Nebraska-Lincoln. His research interests include image analysis and spatial statistics.