National Aeronautics and
Space Administration

Goddard Space Flight Center
Greenbelt, Maryland
20771

Reply to Attn of: 606.3

October 20, 2008

Dr. Russell G. Congalton
Editor-in-Chief, *PE&RS*
4 Ryan Way
Durham, NH 03824
E-mail: russ.congalton@unh.edu

Dear Dr. Congalton:

We are pleased to submit for consideration for publication in the *PE&RS* journal our paper "The Analysis of Image Segmentation Hierarchies with a Graph-Based Knowledge Discovery System." The authors are:

James C. Tilton, Computational & Information Sciences and Technology Office (606.3), NASA Goddard Space Flight Center, Greenbelt, MD 20771, James.C.Tilton@nasa.gov.
Diane J. Cook, Electrical Engineering and Computer Science, Washington State University, Pullman, WA 99164, cook@eecs.wsu.edu.
Nikhil Ketkar, Electrical Engineering and Computer Science, Washington State University, Pullman, WA 99164, nikhilketkar@mac.com.
Selim Aksoy, Department of Computer Engineering, Bilkent University, Ankara, Turkey, saksoy@cs.bilkent.edu.tr.

This paper reports on our original work and is currently not being considered for publication in any other journal.

Sincerely,

James C. Tilton
(Computational & Information Sciences and Technology Office)
Mail Code 606.3
NASA's GSFC
Greenbelt, MD 20771
http://cisto.gsfc.nasa.gov/TILTON/
James.C.Tilton@nasa.gov

THE ANALYSIS OF IMAGE SEGMENTATION HIERARCHIES WITH A GRAPH-BASED KNOWLEDGE DISCOVERY SYSTEM

This paper describes approaches to the integration of the Subdue graph-based knowledge discovery system with image segmentation hierarchies generated by the RHSEG algorithm and initial results from the application of the combined system to image data analysis.

Abstract: Currently available pixel-based analysis techniques do not effectively extract the information content from the increasingly available high spatial resolution remotely sensed imagery data. A general consensus is that object-based image analysis (OBIA) is required to effectively analyze this type of data. OBIA is usually a two-stage process; image segmentation followed by an analysis of the segmented objects. We are exploring an approach to OBIA in which hierarchical image segmentations provided by the Recursive Hierarchical Segmentation (RHSEG) software developed at NASA GSFC are analyzed by the Subdue graph-based knowledge-discovery system developed by a team at Washington State University. In this paper we discuss our initial approach to representing the RHSEG-produced hierarchical image segmentations in a graphical form understandable by Subdue, and provide results on real and simulated data. We also discuss planned improvements designed to more effectively and completely convey the hierarchical segmentation information to Subdue and to improve processing efficiency.

1.0 INTRODUCTION

Currently available image analysis techniques do not effectively extract the information content from the increasingly available high spatial resolution remotely sensed imagery data. High spatial resolution imagery can resolve individual objects such as man-made structures and even individual large trees. However, several studies have shown that most currently available pixel-based analysis techniques do not perform well on this type of data. The field of object-based image analysis (OBIA) has arisen in recent years to address the need to move beyond pixel-based analysis.

OBIA relies on image segmentation as a starting point. The quality of this initial image segmentation strongly influences the effectiveness of the ensuing object-based analysis. We have chosen to use the Recursive Hierarchical Segmentation (RHSEG) software (Tilton, 2008) developed internally at NASA GSFC as our image segmentation approach. RHSEG is an excellent choice because of three key factors: (i) RHSEG produces image segmentations with high spatial fidelity, (ii) RHSEG automatically groups spatially connected region objects into region classes, and (iii) RHSEG automatically produces a hierarchical set of image segmentations.

An approach capable of discovering patterns in the segmented image data is needed for analyzing a hierarchical set of image segmentations, such as the Subdue graph-based knowledge-discovery software (Cook and Holder, 2000). Subdue was developed by a team at Washington State University, and is designed to discover patterns in graph-based structural databases. Subdue has been successfully applied in a number of areas, including bioinformatics, web structure mining, counter-terrorism, social network

analysis, aviation and geology (Holder, *et al.*, 2005; Rakshan, *et al.*, 2004; Joyner, *et al.*, 2001). We hypothesize that the capabilities of RHSEG and Subdue can be combined to provide insightful analysis of remotely sensed data.

We have made some initial steps in translating image segmentations into relational graphs for analysis by Subdue, and achieved some limited data analysis success. The grouping of region objects into regions classes, as provided by RHSEG, has proved important in this translation. Our experience has also made clear the importance of enabling Subdue to utilize region object size and region object neighbor relationship information. In addition, full automation of the analysis also requires an image to be segmented at an appropriate level of spatial detail that can be selected from the RHSEG segmentation hierarchy. Towards this end, we discuss an approach designed to enable Subdue to directly utilize the RHSEG segmentation hierarchy information, obviating any need to preselect a level of segmentation spatial detail.

2.0 BACKGROUND

This section introduces RHSEG and Subdue, along with a preliminary approach for converting outputs from RHSEG to the graphical representation required by Subdue.

2.1 RHSEG

RHSEG is an approximation to the HSEG hierarchical image segmentation algorithm. HSEG is a hybrid of hierarchical step-wise optimization (HSWO) and constrained spectral clustering that produces a hierarchical set of image segmentations. HSWO is an iterative approach to region growing segmentation in which the optimal image

segmentation is found at N_R region objects, given a segmentation at N_R+1 region objects (Beaulieu and Goldberg, 1989).

HSWO produces hierarchical segmentations that are useful in many applications. However, with the addition of constrained spectral clustering, HSEG produces segmentations that capture with greater fidelity the spatial detail of the segmented images, while describing the image data compactly in terms of region classes, which are groups of region objects[†].

HSEG's addition of constrained spectral clustering makes it a computationally intensive algorithm for all but the smallest of images. To counteract this, a computationally efficient recursive approximation of HSEG (called RHSEG) was devised. Further improvements in processing speed are obtained through a parallel implementation of RHSEG. The HSEG and RHSEG algorithms, along with a coarse-grained parallel implementation of RHSEG on MIMD computer clusters, are described in detail in Tilton (2007).

RHSEG recursively subdivides two-dimensional image data into four equal-sized subsections until a processing window containing no more than 4000 pixels is obtained. (For three-dimensional image volumes the data is divided into eight equal-sized volumes.) The HSEG algorithm is executed on each processing window subsection until a preset number of regions is obtained and the results are passed up to the previous level of recursion. The HSEG algorithm is initialized with the results from the deeper level of recursion and again run until a preset number of regions is obtained. This continues until

[†] A region object is a set of spatially contiguous image pixels. A region class is a group of one or more spatially disjoint region objects.

the recursion is completed. For all but the deepest level of recursion, a blending algorithm is performed on the HSEG results in which certain pixels are split out from their original region assignment and remerged into a more appropriate region (Tilton, 2005). This prevents processing window artifacts in which region boundaries appear along processing window seams even though the image pixels across the seams are very similar. A hierarchical set of image segmentation is then produced from a final run of HSEG after the blending algorithm is executed on the initial run of HSEG (down to a preset number of regions) after the recursion is completed and the data set is being processed in its entirety.

HSEG controls the relative priority given to merges of spatially adjacent regions versus merges of spatially non-adjacent regions (spectral clustering) through the input parameter S_{wght}. This parameter, which can vary from 0.0 to 1.0, controls the relative importance of spatially adjacent and spatially non-adjacent region merges. When S_{wght} = 0.0, spatially non-adjacent region merges are not allowed and HSEG becomes equivalent to HSWO. With S_{wght} = 1.0, merges between spatially adjacent and spatially non-adjacent regions are given equal priority. For values of S_{wght} between 0.0 and 1.0, spatially adjacent merges are given priority over spatially non-adjacent merges by a factor of $1.0/S_{wght}$. Appropriate values for S_{wght} depend on the application, but usually range from 0.1 to 1.0.

A variety of region dissimilarity criteria may be used with HSWO and HSEG (see Tilton (2008) for a complete list). In our work with Subdue, we found the "square root of

band sum mean squared error" ($BSME^{1/2}$) criterion to be most appropriate. This criterion is defined as:

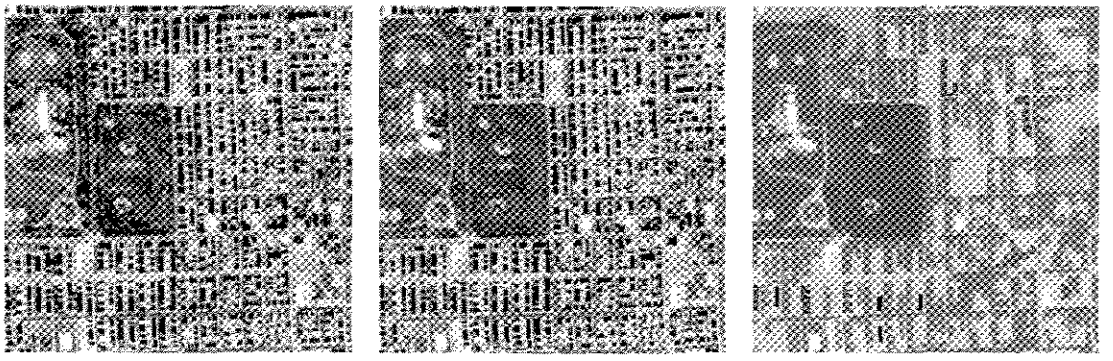
$$d^{1/2}_{BSMSE}(X_i, X_j) = \left[\frac{n_i n_j}{(n_i + n_j)} \sum_{b=1}^{B} (\mu_{ib} - \mu_{jb})^2 \right]^{1/2}, \tag{1}$$

where n_i (n_j) is the number of pixel in region X_i (X_j), and μ_{ib} (μ_{jb}) is the mean of region X_i (X_j) for spectral band b.

A key factor in selecting RHSEG for providing region objects for analysis by Subdue is the fidelity with which RHSEG renders region objects versus other available image segmentation programs. This fidelity stems from the manor in which spectral clustering is tightly bound with region growing in the HSEG algorithm. Other image segmentation programs may have a provision for performing spectral clustering on a final set of region objects, but HSEG is the only algorithm that tightly intertwines spectral clustering with region growing image segmentation. The difference that this makes can be demonstrated by comparing an image segmentation result from RHSEG with a result produced by HSWO. Figs. 1a, 1b and 1c show, respectively, a true color rendition of a 256x256 portion of an Ikonos image[‡], the region mean image from the RHSEG result, and the region mean image from the HSWO results. RHSEG and HSWO were both run with the $BSME^{1/2}$ dissimilarity criterion until a region merging threshold of 10.0 was reached. RHSEG used $S_{wght} = 0.25$.

The region object classification provided by RHSEG is also needed for input to Subdue because the region class provides a node label which Subdue needs to determine interesting associations between neighboring region objects.

[‡] This is a portion of an Ikonos image obtained May 17, 2000 from over Baltimore, MD, USA.



(a) (b) (c)

Fig. 1. (a) A true color rendition of a 256x256 pixel portion of an Ikonos data set. (b) The region mean image from an RHSEG segmentation with $S_{wght} = 0.25$ and maximum merge threshold = 10.0. (c) The region mean image from an HSWO segmentation with maximum merge threshold = 10.0.

Another important factor in selecting RHSEG for use with Subdue is that RHSEG is one of very few programs that readily produce a hierarchical set of image segmentations. Of course, HSWO does this. Also, the segmentation program provided with the commercially available Definiens Developer software product can also provide a hierarchical set of image segmentations (http://www.definiens.com), but this commercial product is expensive, is somewhat cumbersome to use, and includes a large amount of software for rule-based image classification that becomes just overhead when one is just interested in a hierarchical segmentation result. However, neither HSWO nor Definiens Developer provides image segmentation from tightly intertwined spectral clustering and region growing.

A segmentation program that produces a hierarchical set of segmentations is important for knowledge discovery because one generally does not know at what level of

segmentation detail a particular type of region object will be well delineated. While our initial approaches for combining RHSEG and Subdue do not effectively exploit the segmentation hierarchy, we will propose in a later section an approach to enabling Subdue to discover the appropriate level in the segmentation hierarchy at which to extract meaningful patterns and relationships.

2.2 SUBDUE

Numerous approaches have been developed for discovering concepts in linear, attribute-value databases (Frawley, *et al.*, 1992). Although much of the data collected today has an explicit or implicit structural component (e.g., spatial or temporal), only recently have discovery systems been designed to handle this type of data. Current data mining research focuses primarily on algorithms to discover sets of attributes that can discriminate data entities into classes, such as shopping or banking trends for a particular demographic group. These approaches experience difficulty when key concepts involve relationships between the data points. In contrast, we are developing data mining techniques to discover patterns consisting of complex relationships between entities.

Cook and Holder (2000) introduced a method for discovering substructures in structural databases implemented in the Subdue system. In contrast with alternative approaches, Subdue is devised for general-purpose automated discovery, concept learning, and hierarchical clustering. Hence, the method can be applied to many structural domains.

Subdue accepts as input directed or undirected graphs with labeled vertices (nodes) and edges (links), and outputs graphs representing the discovered pattern or learned concept. Formally, Subdue uses a labeled graph $G = (V,E,L)$ as both input and output, where $V = \{v_1, v_2, \ldots, v_n\}$ is a set of vertices, $E = \{(v_i, v_j) \mid v_i, v_j \in V\}$ is a set of edges, and L is a set of labels that can appear on vertices and edges. The graph G can contain directed edges, undirected edges, self-edges, and multi-edges. The input to Subdue can consist of one large graph or a collection of individual graphs, and in the case of supervised learning, the individual graphs are classified as positive or negative examples.

As an unsupervised algorithm, Subdue searches for a substructure, or subgraph of the input graph, that best compresses the input graph. Subdue uses a variant of beam search for its main search algorithm. A substructure in Subdue consists of a subgraph definition and all its occurrences throughout the graph.

Subdue uses a polynomial-time beam search for its discovery algorithm, as summarized in Fig. 2. The initial state of the search is the set of substructures consisting of all uniquely-labeled vertices. Search progresses by applying the ExtendSubstructure operator to each substructure in the current state. As its name suggests, it extends a substructure in all possible ways by a single edge and a vertex, or by only a single edge if both vertices are already in the subgraph. The resulting new substructures are ordered based on their compression (or sometimes referred to as value) as calculated using the MDL principle described below, and the top substructures (as determined by the beam) remain on the queue for further expansion.

Search terminates upon reaching a limit on the number of substructures extended, or upon exhaustion of the search space. Once the search terminates and Subdue returns the list of best substructures, the graph can be compressed using the best substructure. The compression procedure replaces all instances of the substructure in the input graph by single vertices, which represent the substructure definition. Incoming and outgoing edges to and from the replaced instances will point to or originate from the new vertex that represents the instance. The Subdue algorithm can be iterated invoked again on this compressed graph.

Subdue's search is guided by the Minimum Description Length (MDL) (Cook and Holder, 1994) principle formalized in (2), where $DL(S)$ is the description length of substructure S being evaluated, $DL(G|S)$ is the description length of the graph as compressed by the substructure, and $DL(G)$ is the description length of the original graph. The best substructure is the one that minimizes this compression ratio:

$$Compression = \frac{DL(S) + DL(G \mid S)}{DL(G)} \tag{2}$$

As an example, Fig. 3 shows patterns that Subdue discovers in an example input graph and a compressed version of the graph.

To allow slight variations between instances of a discovered pattern (as is the case in Fig. 3), Subdue applies an inexact graph match between the substructure definition and potential instances. Because instances of a substructure can appear in different forms throughout the database, Subdue computes the graph edit distance between two graphs

and considers the substructure instance to be a match if the distance is less than a pre-defined threshold (0 for exact matches) (Cook and Holder, 1994).

```
SUBDUE(graph G, int Beam, int Limit )
  queue Q = {v | v has a unique label in G}
  bestSub = first substructure in Q
  repeat
    newQ = {}
    for each S in Q
        newSubs = ExtendSubstruture(S)
        Evaluate(newSubs)
        newQ = newQ U newSubs mod Beam
        Limit = Limit - 1
        If best ∈ Q better than bestSub
        then bestSub = best substructure in Q
        Q = newQ
  until Q is empty or Limit <= 0
  return bestSub
```

Fig. 2. Subdue's discovery algorithm.

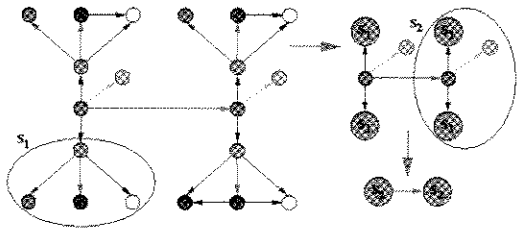


Fig. 3. An example of Subdue's substructure discovery capability. The figure shows the discovered pattern (S_1) from the original graph, the substructure found during the second iteration (S_2), and the final graph compressed using substructures S_1 and S_2.

2.3 Conversion of RHSEG Output to Graph Format

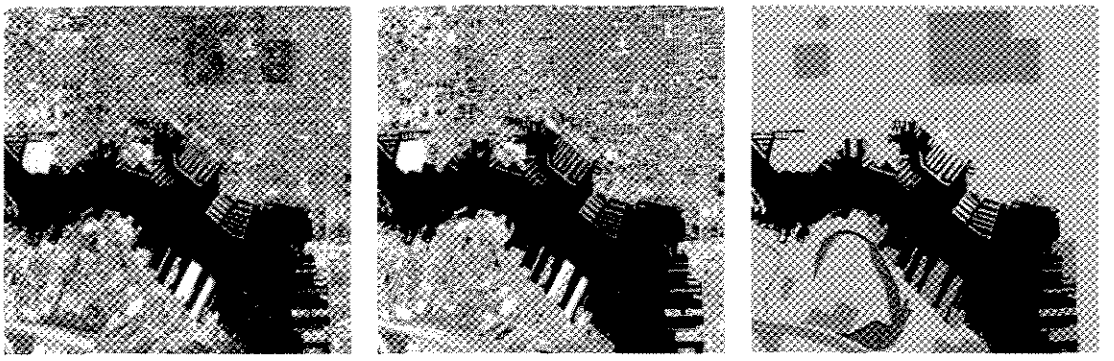
Before significant experiments could be performed on image data sets of any significant size, a computer program had to be devised to convert outputs from RHSEG to the graphical representation required by Subdue. An existing RHSEG utility program, called "feature_extract", was augmented to add an option to output a Subdue-compatible input graph. Eventually this program was further augmented to order the RHSEG region objects by size (largest to smallest) and drop out from consideration region objects smaller than a specified size (number of pixels). The only information conveyed from the RHSEG segmentation output to the Subdue input graph was the region class label for

each region object and whether or not a region object was spatially adjacent (linked) to another region object. This was done for just one selected level from the segmentation hierarchy. The Subdue input graph consisted of a list of graph vertices (region objects) labeled by the region class label and a list of undirected edges specifying which region objects were linked to what other region object.

3.0 Initial Results

Initial experiments combining RHSEG and Subdue were performed on a 768x768 pixel section of Ikonos data from over the center of Baltimore, MD. A true color rendition of this data set is displayed in Fig. 4a, and a hand labeling of the scene in terms of generalized land cover/land use classes is displayed in Fig. 4c. The RHSEG segmentation result selected to use for testing the RHSEG/Subdue interface provided by the feature_extract program is displayed in Fig. 4b. Our hope was that Subdue would find significant subgraphs out of the graphical representation of the RHSEG segmentation (Fig. 4b) corresponding to the hand labeled land cover/land use map (Fig. 4c).

Fig. 5 highlights in red the eight most significant subgraphs discovered by Subdue from the RHSEG segmentation displayed in Fig. 4b. While there is no clear correspondence between the first seven subgraphs and the sought-for generalized land cover/land use classes), the eighth most significant subgraph corresponds closely to the "Parks" class delineated in Fig. 4c. (NOTE: The RHSEG region classes mentioned in Fig. 5 do *not* have a one-to-one correspondence to the land cover/land use classes in Fig. 4c and Table I.)



(a) (b) (c)

Fig. 4. (a) A true color rendition of a 768x768 pixel section of Ikonos data from the harbor and Patterson Park area of Baltimore, MD. (b) The RHSEG segmentation result used for testing the RHSEG/Subdue interface. The $BSME^{1/2}$ dissimilarity criterion was employed with $S_{wght} = 0.25$. This segmentation has 11 region classes and 38,773 region objects. (c) A land use/land cover class hand labeling of the Ikonos scene in terms of generalized land use/ land cover labels (see Table I for color key).

Table I. Color key for land use/land cover class labeling of Baltimore, MD Ikonos scene. (Applies to Fig. 4c only.)

Class	Color	Class	Color	Class	Color
Harbor		Housing Project		Docks	
Commercial		Major Highways		Dredge Fill	
Residential		Rail Yard		Clouds	
Parks		Marina		Cloud Shadows	

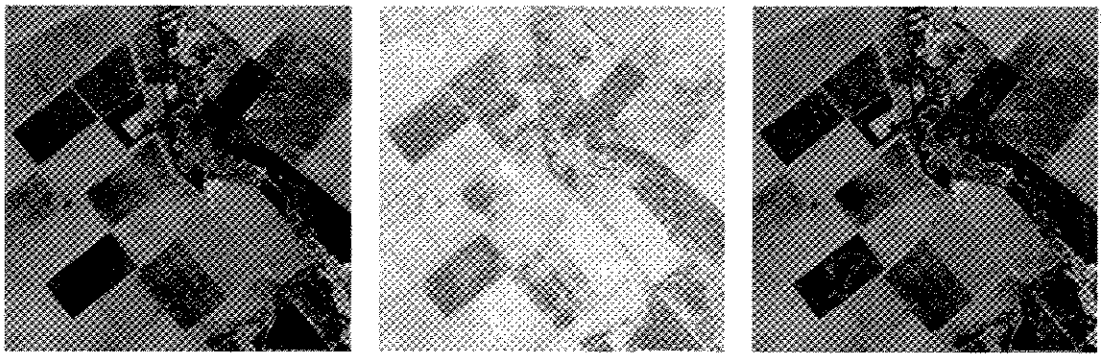
1: 4-5-9 2: 3-2-4 4: 6-7 8: 6-8

Fig. 5. A selection from the eight most significant subgraphs found by Subdue in analyzing the RHSEG segmentation displayed in Fig. 4b. The image area covered by the region objects participating in instances of these subgraphs are highlighted in red. Each subgraph is labeled by (order of significance):(region class relationship detected). (The 3: 4-5, 5:5-9, 6:4-9 and 7: 3-5 cases are not shown, but look similar to the 2nd most significant subgraph result.)

Subdue consumed a great deal of computer processing time producing this result: about 93 hours on a 2.33 GHz computer. The Subdue input graph is quite large, with 38,773 vertices. However, we discovered that Subdue produced nearly identical results with an input graph in which all region object less than four pixels in size were dropped out, resulting in an input graph of 17,531 vertices. Subdue consumed about 8.3 hours of computer time processing this input graph, a speed-up of over 11 times (we will later discuss how Subdue's processing time can be significantly further reduced).

In a completely different application of Subdue, we have found that Subdue can recognize certain noise patterns in one-look SAR imagery. Fig. 6a shows a 1024x1024 pixel subsection of a TerraSAR-X image from DLR (German Aerospace Center). The full image may be viewed at http://www.dlr.de/en/DesktopDefault.aspx/tabid-

4219/6774_read-9519/gallery-1/gallery_read-Image.1.3575/. This image is of a region in the south Russian Steppes about 500 kilometers northeast of the Black Sea and about 50 kilometers west of Volgograd. Fig. 6b shows a 10 region class segmentation of this image produced by RHSEG. Fig. 6c highlights the image areas covered by instances of the sixth most significant subgraph recognized by Subdue. These areas correspond to the darker colored fields in Fig. 6a represented by a SAR noise pattern represented in the RHSEG segmentation as turquoise regions next to purple regions.



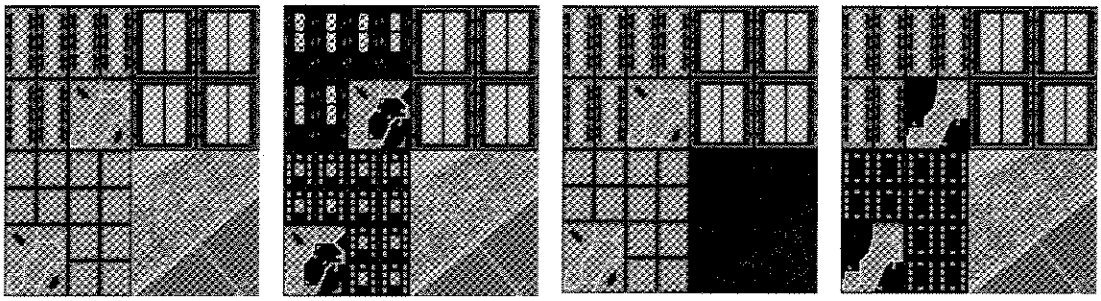
(a) (b) (c)

Fig. 6. (a) A section of TerraSAR-X imagery data. (b) Ten region class segmentation produced by RHSEG. (c) The image areas covered by instances of the sixth most significant subgraph discovered by Subdue.

The above tantalizing but inconclusive results led us to step back and think about how we could clarify the situation. We decided to put together a simulated segmentation result to test the behavior of Subdue run with different parameter settings. The simulated segmentation, displayed in Fig.7a, combines idealized segmentations of a residential area (most of the lower left quadrant), an apartment complex (most of the upper left quadrant), an industrial park (the upper right quadrant) and recreational parks (inserted in the

apartment complex and residential quadrants) with a section of an actual segmentation of SAR data (lower right quadrant).

We performed out initial run with Subdue with the same parameters used in our previous runs discussed above. The three most significant or highest-valued subgraphs discovered by Subdue are displayed as the blacked out areas in Figs. 7b, 7c and 7d. The most significant subgraph finds portions of three land use classes: apartment complex, residential and parks. The second most significant subgraph finds most of the SAR quadrant. The third most significant subgraph finds portions of two land use classes: residential and parks.



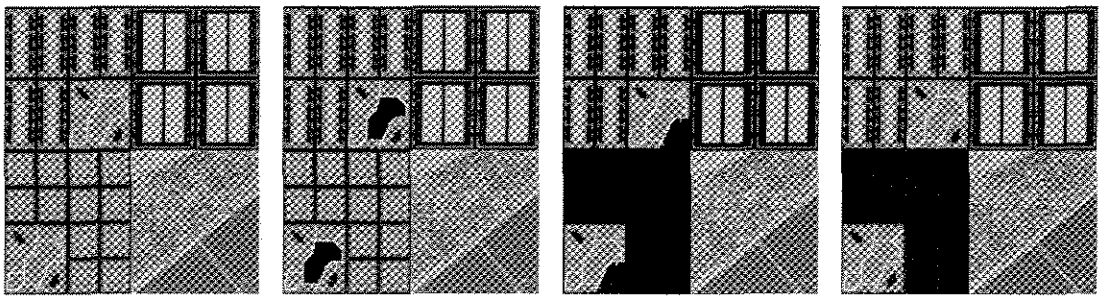
(a) (b) (c) (d)

Fig. 7. (a) A simulated image segmentation (see text). (b) Most significant sub-graph, (c) second most significant subgraph, and (d) third most significant subgraph from Subdue. The image area covered by the region objects participating in instances of these subgraphs is blacked out.

What we notice about these three most significant subgraphs is they are all two-vertex subgraphs. The first is a grass-roof subgraph, the second is a subgraph linking the two SAR classes, and the third is a grass-trees subgraph. The second thing we notice is that each region object is allowed to participate in only one subgraph instance. Thus, in Fig.

7b, for each one grass region object only one adjacent roof object is blacked out. This is also why all SAR region objects are not blacked out in Fig. 7c. The third thing we notice is that the two apparent SAR subregions (upper left vs. lower right) are not distinguished. We note that one SAR subregion (upper left) consists primarily of a large orange region object containing many small purple region objects. Conversely, the other SAR subregion (lower right) consists primarily of a large purple region object containing many small orange region objects. We will discuss in the following "improvements" section how the last problem can be addressed, but report on a simple remedy for the first two problems here.

Noting that the concepts of apartment complex, residential area, industrial park and recreational park involve more than a two-vertex graph, we decided to rerun Subdue on the same input graph with the requirement that only subgraphs with at least five vertices be considered. We also turned on the Subdue option to report all subgraph instances instead of limiting a region object to participate in only one subgraph instance. The results from this run of Subdue are displayed in Fig. 8.



(a) (b) (c) (d)

Fig. 8. (a) A simulated image segmentation (same as Fig. 7a). (b) Most significant subgraph, (c) second most significant subgraph, and (d) third most significant subgraph from Subdue with at least five vertices.

The most significant subgraph discovered by Subdue in this second analysis of the simulated data is a grass region object linked to a roof region object, and three bare soil region objects. The area covered by all instances of this subgraph is the center of both parks (Fig. 8b). Since all subgraph instances are taken into account, all three roof region objects in these park sections are blacked out instead of just one.

The second most significant subgraph discovered by Subdue in this second analysis is a grass region object linked to an asphalt region object and also linked to a tree region object and two roof region objects. This subgraph pattern occurs both in the residential area and in the lower right portion of the parks area. The area blacked out also extends to the asphalt areas in the apartment complex and industrial park because the asphalt area is just one large region object. Again, all related roof region objects are blacked out since all subgraph instances are taken into account.

The third most significant subgraph discovered by Subdue in this second analysis is a grass region object linked to a tree region object and also linked to three roof objects. The areas blacked out in Fig. 8d correspond precisely to the residential area sans the road/driveway network. Again, all roof objects in the residential area are blacked out because all subgraph instances are considered.

These encouraging results lead us to the next section, in which planned additional improvements to Subdue and the RHSEG/Subdue interface are discussed.

4.0 Planned Improvements to RHSEG, Subdue and Their Interface

The improvements to RHSEG, Subdue and their interface that we plan to perform can be categorized as follows: (i) modifications in how the RHSEG segmentation is translated into the Subdue input graph (section 4.1 below), (ii) improvements in Subdue processing efficiency (section 4.2 below), and (iii) augmentation of Subdue to input and utilize additional information describing the relationships between region objects in the RHSEG segmentations (section 4.3 below).

4.1 Enabling Subdue to Utilize the RHSEG Segmentation Hierarchy Information

We have noted that RHSEG produces a hierarchical set of image segmentations. So far we have not exploited this capability except for manually inspecting the hierarchical set of image segmentations and selecting the segmentation for further analysis from the hierarchical level that best distinguishes region objects of interest. In future work we plan to enable Subdue to utilize information from segmentations at several hierarchical levels in the process of discovering significant subgraphs. Enabling this capability will require relatively minor modifications in the RHSEG output and Subdue input. RHSEG would be modified to output a matrix describing the segmentation hierarchy, and Subdue would be modified to input this matrix and utilize this information in its inexact graph matching scheme. For example, region objects from region classes that merge one level up in the segmentation hierarchy would be considered to be closer to each other than region objects from region classes that merge two levels up in the segmentation hierarchy (here "going up the segmentation hierarchy" means going to a coarser segmentation level).

Successful completion of these modifications will be highly significant in adding the unique capability of automatically analyzing a segmentation hierarchy and automatically choosing the appropriate hierarchical level for land use / land cover identification. We are not aware of any other software package having this capability. However, a recent paper by Akcay and Aksoy (2008) describes an attempt for automatic selection of regions from a segmentation hierarchy based on spectral homogeneity and neighborhood connectivity.

4.2 Improving Subdue Processing Throughput with a Planar Graph Assumption

We noted in the background section that Subdue consumes a great deal of computer processing time in its analysis. It took about 93 hours to analyze an input graph with 38,773 vertices and about 8.3 hours to analyze an input graph with 17,531 vertices. Subdue assumes that the graphs being analyzed are general graphs. About 99% of Subdue's computation time is spent in identifying instances of a candidate pattern. At present there is no known polynomial-time algorithm for testing if two general graphs are isomorphic (Jenner, *et al.*, 2003). However, graph isomorphism for planar graphs is linear. Since we will be employing a region adjacency graph representation for the RHSEG segmentations, we will be working with planar graphs (Wang and Abe, 1995) and can make use of the graph isomorphism tests have been previously explored for this purpose (Kukluk, *et al.*, 2005; Boyer and Myrvold, 1999). We anticipate that the subsequent improvement in the graph match runtime brought about by giving Subdue an option to assume that the incoming graph is planar will greatly improve Subdue's ability to quickly process image data and find more complex discoveries.

4.3 General Approaches for Enabling Subdue to Utilize Region Object Size and Region Object Neighbor Relationship Information

The information provided to Subdue from the RHSEG generated image segmentations in our initial tests was very limited compared to what could potentially be provided. In particular, Subdue was provided with only the region class membership of each region object along with the connectivity between spatially adjacent region objects. In future work, we plan to explore techniques through which a richer set of information about the image segmentation is provided to Subdue, along with modifications to Subdue that will be necessary to enable Subdue to utilize this information.

Two key pieces of information that appear to be important are the size of the region objects and the nature of the region object neighbor relationship. While utilization of region size information is relatively straightforward, the region object neighbor relationship requires some further exploration. Given the image segmentation provided by RHSEG, the topological (bordering, invading, surrounding), distance-based (near, far) and directional (above, below, right, left) pair-wise region spatial relationships can be modeled based on relative overlaps, distances and orientations between region boundaries (Aksoy, *et al.*, 2005). Such relationships can be combined into an attributed relational graph structure where the regions are represented by graph nodes and their spatial relationships are represented by the edges between such nodes (Aksoy, 2006). Nodes can be labeled with the class (land cover/use) names and the corresponding confidence values (posterior probabilities) for these class assignments. Edges can be labeled with the spatial

relationship classes (pair-wise relationship names) and the corresponding degrees (fuzzy membership values) for these relationships.

An important issue is the trade-off between how much detail is modeled and how well the model can be generalized. For example, when the connectivity (spatial adjacency) relation is separated into the three cases of bordering, invading and surrounding, the number of instances of matching subgraphs can significantly decrease even if inexact graph matching techniques are used. An interesting problem is to find the important spatial relationships so that higher weights can be given to those relationships during substructure discovery using Subdue. For example, Kalaycilar, *et al.* (2008) examined, the summarization of the full graph structure using spatial relationship histograms and selection algorithms for identification of distinguishing region groups that are frequently found in a particular class of scenes but rarely exist in others. The multi-scale (hierarchical) abstraction of segmentation and classification using RHSEG provides a good basis for such selection algorithms. In order to effectively utilize region object size information and information from modeling the spatial relationships between region objects, Subdue will be modified to work on a weighted graph, i.e., a graph with weighted vertices and weighted links between vertices. One way to handle the weighted vertices and edges is to evaluate the graph patterns as though a vertex or edge with weight w was equivalent to w identical vertices or w edges between the neighboring vertices (which would be an alternative representation for these segmented images). Using the MDL principle, such an approach will tend to favor graph patterns which include heavier-weighted vertices and edges.

5.0 CONCLUDING REMARKS

With the increasing availability of high spatial resolution remote sensing imagery (< 5m), image analysis approaches must evolve to reflect the nature of this data. Pixel based image analysis techniques that were previously successfully applied to low spatial resolution remote sensing imagery have been shown to be inaccurate when applied to this high spatial resolution data (Marceau and Hay, 1999). This is because scale of the recognizable objects has become much smaller in this high spatial resolution data. For example, a modern housing development is captured in low spatial resolution data by image pixels consisting of mixtures of ground covers such as house roofs, driveways, grass, trees and streets. In contrast, high spatial resolution data captures such a scene as groups of relatively pure pixels of the different ground covers. That is, the discernable objects in the high spatial resolution are larger than the pixel spatial resolution, whereas these same objects are generally smaller than the pixel spatial resolution in low spatial resolution data.

The field of Object-Based Image Analysis (OBIA) has arisen in recent years to answer this need to move beyond pixel based analysis. Many of practitioners at the object-based image analysis conferences (OBIA 2006 and GEOBIA 2008) reported some limited successes in Geographic Object Based Image Analysis using software from Definiens (http://www.definiens.com/). The analysis approach taken by this software is to perform a segmentation of the imagery data, possibly at a couple different levels of detail, and then develop a set of rules based on region object features to identify the region objects. Certain rules can be developed based on region object neighbor relationships. In contrast

to the software we are developing, the Definiens software cannot implement rules concerning region object neighbor networks beyond a pair-wise relationship, and Definiens software has no provision for discovering significant spatial relationships between region objects. In this regard, the capabilities we are developing are significantly beyond those provided by any currently available software package.

ACKNOWLEDGMENT

This work was supported by NASA's Applied Information Systems Research program.

REFERENCES

Akcay, H.G. and S. Aksoy, 2008. Automatic detection of geospatial objects using multiple hierarchical segmentation, *IEEE Transactions on Geoscience and Remote Sensing*, 46(7):2097-2111.

Aksoy, S., K. Koperski, C. Tusk, G. Marchisio and J. C. Tilton, 2005. Learning Bayesian Classifiers for Scene Classification with a Visual Grammar, *IEEE Transactions on Geoscience and Remote Sensing*: 43(3):581-589.

Aksoy, S., 2006, Modeling of Remote Sensing Image Content using Attributed Relational Graphs, *Proceedings of IAPR International Workshop on Structural and Syntactic Pattern Recognition*, August 17-19, Hong Kong, China, pp. 475-483.

Beaulieu, J.-M. and M. Goldberg, 1989. Hierarchy in picture segmentation: A stepwise optimal approach, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(2): 150-163.

Boyer, J., and W. Myrvold, 1999. Stop minding your p's and q's: A simplified *o(n)* planar embedding algorithm. *Proceedings of the Tenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 140-146.

Cook, D.J. and L. B. Holder, 1994. Substructure discovery using minimum description length and background knowledge. *Journal of Artificial Intelligence Research*, 1:231-255.

Cook, D.J. and L. Holder, 2000. Graph-based data mining. *IEEE Intelligent Systems*, 15(2): 32-41.

Frawley, F., G. Piatetsky-Shapiro, and C. Matheus, 1992. Knowledge discovery in databases: An overview. AI Magazine, pp. 213-228.

Holder, L., D. Cook, J. Coble and M. Mukherjee, 2005. Graph-based relational learning with application to security, *Fundamenta Informaticae Special Issue on Mining Graphs, Trees and Sequences*, 66(1-2):83-101.

Jenner, B., J. Köbler, P. McKenzie, and J. Torán, 2003. Completeness results for graph isomorphism. *Journal of Computer and System Sciences*, 66(3):549-566.

Jonyer, I., D. Cook, and L. Holder, 2001. Discovery and evaluation of graph-based hierarchical conceptual clustering, *Journal of Machine Learning Research*, 2:19-43.

Kalaycilar, F., Asli Kale, Daniya Zamalieva, Selim Aksoy, 2008. Mining of Remote Sensing Image Archives Using Spatial Relationship Histograms, *Proceedings of the IEEE International Geoscience and Remote Sensing Symposium*, July 6-11, Boston, MA, unpaginated CD-ROM.

Kukluk, J., J., L. Holder, and D. Cook, 2005. Algorithm and experiments in testing planar graphs for isomorphism. *Journal of Graph Algorithms and Applications*, 8(2):313-356.

Marceau, D.J., and G. J. Hay, 1999. Remote sensing contributions to the scale issue, *Canadian Journal of Remote Sensing*, 25(4): 357-366.

Rakhshan, A., L. Holder, and D. Cook, 2004. Structural web search engine, *International Journal on Artificial Intelligence Tools*, 13(1):27-44.

Tilton, J.C., 2005. A Split-remerge Method for Eliminating Processing Window Artifacts in Recursive Hierarchical Segmentation, *Disclosure of Invention and New Technology: NASA Case No. 14,994-1.*

Tilton, J.C., 2007. Parallel Implementation of the Recursive Approximation of an Unsupervised Hierarchical Segmentation Algorithm, *High Performance Computing in Remote Sensing*, (C.-I. Chang and A. J. Plaza, editors), Chapman & Hall/CRC, Boca Raton, FL, pp. 97-107.

Tilton, J.C., 2008. *RHSEG, HSEGViewer and HSEGReader User's Manual, Version 1.40*, July 28, 2008, available from http://ipp.gsfc.nasa.gov/RHSEG.

Wang, C., and K. Abe, 1995. Region correspondence by inexact attributed planar graph matching. *Proceedings of the Fifth International Conference on Computer Vision*, pp. 440-447.