

# Automatic Detection of Geospatial Objects Using Multiple Hierarchical Segmentations

H. Gökhan Akçay and Selim Aksoy, *Member, IEEE*

**Abstract**—The object-based analysis of remotely sensed imagery provides valuable spatial and structural information that is complementary to pixel-based spectral information in classification. In this paper, we present novel methods for automatic object detection in high-resolution images by combining spectral information with structural information exploited by using image segmentation. The proposed segmentation algorithm uses morphological operations applied to individual spectral bands using structuring elements in increasing sizes. These operations produce a set of connected components forming a hierarchy of segments for each band. A generic algorithm is designed to select meaningful segments that maximize a measure consisting of spectral homogeneity and neighborhood connectivity. Given the observation that different structures appear more clearly at different scales in different spectral bands, we describe a new algorithm for unsupervised grouping of candidate segments belonging to multiple hierarchical segmentations to find coherent sets of segments that correspond to actual objects. The segments are modeled by using their spectral and textural content, and the grouping problem is solved by using the probabilistic latent semantic analysis algorithm that builds object models by learning the object-conditional probability distributions. The automatic labeling of a segment is done by computing the similarity of its feature distribution to the distribution of the learned object models using the Kullback–Leibler divergence. The performances of the unsupervised segmentation and object detection algorithms are evaluated qualitatively and quantitatively using three different data sets with comparative experiments, and the results show that the proposed methods are able to automatically detect, group, and label segments belonging to the same object classes.

**Index Terms**—Hierarchical segmentation, image segmentation, mathematical morphology, object-based analysis, unsupervised object detection.

## I. INTRODUCTION

**D**UE TO the constantly increasing coverage and availability of very high resolution remotely sensed data, automatic content extraction, object detection, and classification for urban applications have continued to be important research problems. There is an extensive literature on the classification of remotely sensed imagery, where pixel-level processing has been the common choice for remote sensing image analysis systems. These systems use a broad range of features, including multi- or hyperspectral information, texture features, edge detection, and

Manuscript received November 3, 2007; revised December 19, 2007. This work was supported in part by the TUBITAK CAREER Grant 104E074 and in part by the European Commission Sixth Framework Program Marie Curie International Reintegration Grant MIRG-CT-2005-017504.

The authors are with Department of Computer Engineering, Bilkent University, Ankara 06800, Turkey (e-mail: akçay@cs.bilkent.edu.tr; saksoy@cs.bilkent.edu.tr).

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

Digital Object Identifier 10.1109/TGRS.2008.916644

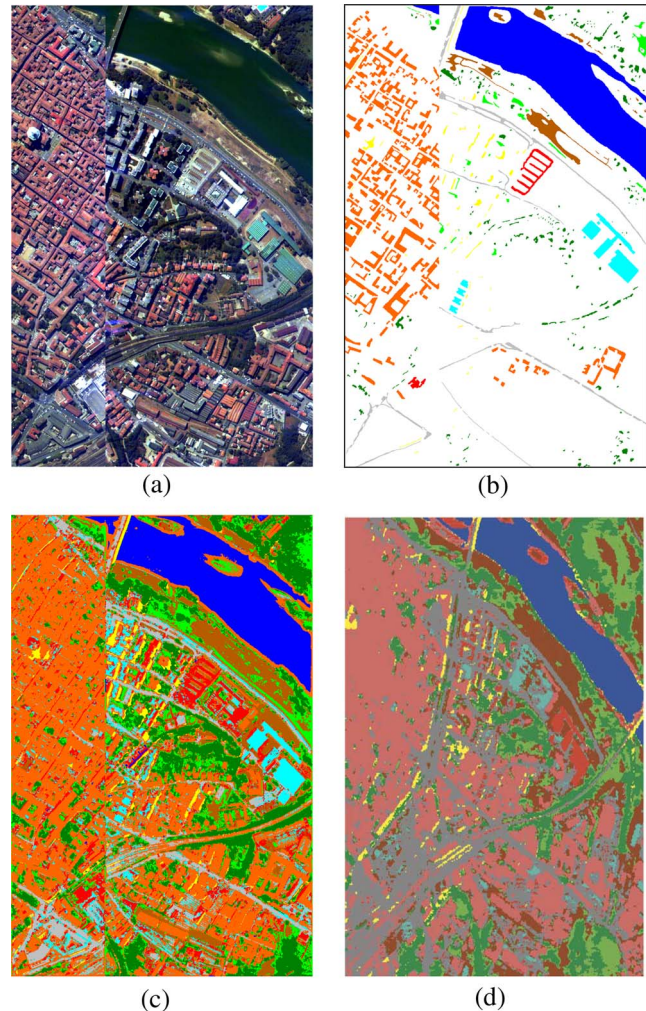


Fig. 1. Example classification results using a pixel-based quadratic Gaussian classifier with (c) PCA and Gabor features and (d) DAFE features. The classification maps for (c) and (d) are taken from [1] and [2], respectively. (a) False color. (b) Reference map. (c) Classification using PCA and Gabor features. (d) Classification using DAFE features.

linear or nonlinear transformations of these features. Such features are used with a wide range of classifiers, including probabilistic methods employing maximum-likelihood or Bayesian estimation techniques, neural networks, decision trees, support vector machines, and genetic algorithms for applications like land cover/land use classifications.

Despite the high success rates that have been published in the literature using limited ground truth data, visual inspection of the results shows that most of the urban structures still cannot be delineated as accurately as expected, particularly in high-resolution images. For example, Fig. 1(a) shows the false-color

representation of a hyperspectral image of Pavia, Italy. The classification map shown in Fig. 1(c) is obtained by using features extracted with principal components analysis (PCA) and Gabor texture filters with a quadratic Gaussian classifier [1]. Similarly, Fig. 1(d) shows the map obtained by using discriminant analysis feature extraction (DAFE) and a similar classifier [2]. Although the success rates obtained as 93.97% and 97.2%, according to the reference map shown in Fig. 1(b), can be considered quite high, none of the boundaries of the buildings, roads, and shadows on the left half of the image is explicit, and no structure can be seen in the results. In other words, the limitations of the pixel-based classification evaluated by using limited pixel-based ground truth are not necessarily reflected in the numerical accuracy. Therefore, this shows that there is still much work to be done, and more advanced classification methods must be designed for practically acceptable results.

We believe that in addition to pixel-based spectral data, spatial and structural information should also be used for a more intuitive and accurate classification. Common ways of incorporating spatial information into the classification involve the use of textural, morphological, and object-based features. Features extracted by using co-occurrence matrices, Gabor wavelets [3], morphological profiles [4], and Markov random fields [5] have been widely used in the literature to model spatial information in neighborhoods of pixels. However, problems, such as scale selection and the detailed content of very high resolution imagery, make the applicability of traditional fixed-window-based methods difficult for such data sets.

Another powerful method for exploiting structural information is to perform region-based classification rather than classifying individual pixels. This is also referred to as object-oriented classification in the remote sensing literature. For example, Bruzzone and Carlin [6] performed classification using the spatial context of each pixel according to a hierarchical multilevel representation of the scene. In a similar approach [7], we obtained a wavelet-based multiresolution representation, segmented images at each resolution, and used region-based spectral, textural, and shape features for classification. In [8], Katartzis *et al.* also modeled spatial information by segmenting images into regions and classifying these regions using a Markovian model, which is defined on the hierarchy of a multiscale region adjacency graph. In another study [9], Soh *et al.* presented a system for sea-ice image classification, which also segmented the images, generated descriptors for the segments, and then used expert system rules to classify the images.

Many popular segmentation algorithms in the computer vision literature assume that images have a moderate number of objects with relatively homogeneous features and cannot be directly applied to high-resolution remote sensing images that contain a large number of complex structures. Furthermore, another popular approach of edge-based segmentation is hard for such images because of the large amount of details. Moreover, watershed-based techniques are also not very useful because they often produce oversegmented results that are mostly because of irrelevant local extrema in the images. A common approach is to apply smoothing filters to suppress these extrema; however, lots of details in high-resolution images may be lost because the spatial support of these details are usually small. Therefore, most of the segmentation work in the remote sensing literature have been based on merging neighboring

pixels according to user-defined thresholds on their spectral similarity. Alternatively, proximity filtering and morphological operations can also be used as postprocessing techniques to pixel-based classification results for segmenting regions [10].

In a related work, Pesaresi and Benediktsson [4] successfully applied opening and closing operations with increasing structuring element (SE) sizes to an image to generate morphological profiles for all pixels and assigned a segment label to each pixel using the SE size corresponding to the largest derivative of these profiles. Although morphological profiles are sensitive to different pixel neighborhoods, the segmentation decision is performed by individually evaluating pixels without considering the neighborhood information, and the assumption that all pixels in a structure have only one significant derivative maximum occurring at the same SE size often does not hold for very high resolution images. Scale selection is also a very important problem in multiscale/hierarchical segmentation techniques. For example, Tilton [11] developed a hierarchical segmentation algorithm that combined spectral clustering with iterative region growing, where segments at coarser levels of detail were obtained by merging segments at finer levels of detail. The multiresolution segmentation implementation offered by the eCognition software also consists of bottom-up region merging, where each pixel is initially considered as a separate object and pairs of image objects are iteratively merged to form larger segments [12]. The main problems associated with both of these approaches are that the resulting segmentations depend on the thresholds used with local homogeneity criteria and that the manual interpretation of the hierarchy is needed because different objects may appear at different scales.

Our main contributions in this paper are twofold. We present a new segmentation algorithm for exploiting structural information and propose a novel method that uses the resulting regions for unsupervised object detection. Our first contribution, which is the segmentation algorithm, uses the neighborhood and spectral information as well as the morphological information. First, morphological opening and closing operations are applied to individual spectral bands using SEs in increasing sizes to generate morphological profiles. These operations produce a set of connected components, forming a hierarchy of segments for each band. Then, unlike [4], where only the scale with the maximum change in the profile is considered, each component at different levels of the hierarchy is evaluated as a candidate for meaningful structures using a measure that consists of two factors, namely: spectral homogeneity, which is calculated in terms of the variances of spectral features, and neighborhood connectivity, which is calculated by using the sizes of connected components. A novel two-pass algorithm is designed to select the segments that jointly optimize this combined measure and find the meaningful segments in a completely unsupervised process. The proposed selection algorithm is generic in the sense that other criteria for homogeneity and connectivity can also be directly incorporated.

An important observation is that different structures appear more clearly in different bands. For example, buildings can be detected accurately in one band, but roads, trees, fields, and paths can be detected accurately in other bands. With a similar observation, Benediktsson *et al.* [2] appended the morphological profiles that were independently extracted from multiple principal components into a single high-dimensional feature

vector, performed linear feature reduction, and classified the pixels using neural networks.

In this paper, as our second main contribution, we propose a novel unsupervised method for the automatic detection of objects from multiple hierarchical segmentations and the corresponding candidates for meaningful structures from individual bands. The goal is to find coherent groups of segments that correspond to actual objects. Considering multiple objects/structures of interest, this setting can also be seen as a grouping problem within the space of a large number of candidate segments obtained from multiple hierarchical segmentations. To solve the grouping problem, we use the probabilistic latent semantic analysis (PLSA) [13] technique by formulating a graphical model for the joint probability of the segments and their features in terms of the probability of observing a feature given an object and the probability of an object given the segment. The parameters of this graphical model are learned by using the expectation-maximization (EM) algorithm. Then, for a particular segment, the set of probabilities of objects/structures given this segment, can be used to assign an object label to this segment. The performances of the unsupervised segmentation and automatic object detection algorithms are evaluated qualitatively and quantitatively using three different data sets with comparative experiments.

The rest of this paper is organized as follows. The data sets and the features used for both segmentation and object detection are introduced in Section II. The segmentation algorithm for the extraction of candidate segments from individual bands in an image is described in Section III. The algorithm for grouping segments for object detection is presented in Section IV. Experiments are discussed in Section V, and conclusions are given in Section VI.

## II. FEATURE EXTRACTION

We illustrate the proposed algorithms using three data sets.

- 1) DC Mall: HYDICE image with  $1280 \times 307$  pixels, 3-m spatial resolution, and 191 spectral bands corresponding to an airborne data flightline over the Washington, DC Mall area. The false-color image is shown in Fig. 2(a).
- 2) Pavia: ROSIS data with  $1096 \times 715$  pixels, 2.6-m spatial resolution, and 102 spectral bands corresponding to the city center in Pavia, Italy. The false-color image is shown in Fig. 3(a).
- 3) Ankara: IKONOS data with  $500 \times 500$  pixels and 1-m spatial resolution pan-sharpened red, green, blue (RGB) bands corresponding to part of a university campus in downtown Ankara. The color image is shown in Fig. 19(a).

Considering that morphological operations have traditionally been defined for single-band binary or gray-scale images, we applied PCA to summarize the hyperspectral data because the PCA bands provide the optimal representation in the least-squares sense [14]. The resulting three bands corresponding to the top principal components representing the 99% variance of the whole data are shown in Figs. 2 and 3 for the DC Mall and Pavia data sets, respectively. Original RGB bands were used for the Ankara data set.

In addition to the PCA bands that give the best representation, we also applied linear discriminant analysis (LDA) that

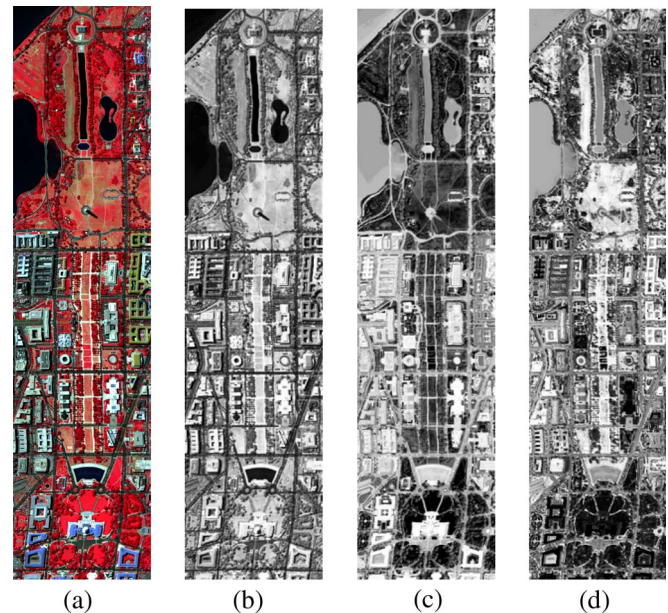


Fig. 2. False-color image (generated using the bands 63, 52, and 36) and the first three PCA bands of the DC Mall data set. (a) False color. (b) First PCA band. (c) Second PCA band. (d) Third PCA band.

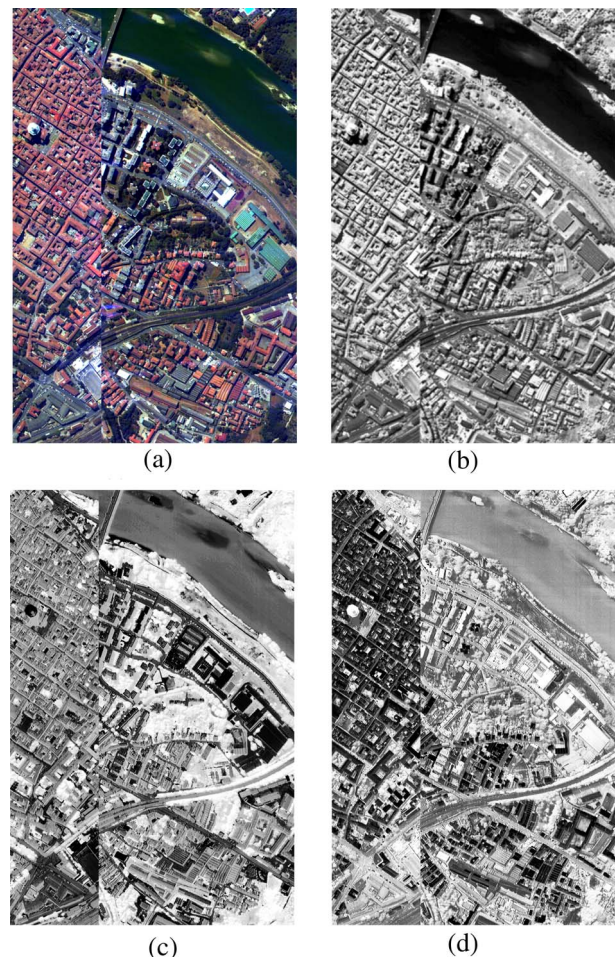


Fig. 3. False-color image (generated using the bands 68, 30, and 2) and the first three PCA bands of the Pavia data set (a missing vertical section in the middle was removed). (a) False color. (b) First PCA band. (c) Second PCA band. (d) Third PCA band.

projects the data onto a new set of bases that best separate the classes in the least-squares sense [14]. Six bands for the DC Mall and eight bands for the Pavia data sets were extracted by using the pixel level seven class and nine class ground truth available for these data sets, respectively. Finally, we extracted Gabor texture features [3] using kernels at two scales and four orientations, resulting in an additional feature vector of length eight for each pixel for a given band. The resulting PCA bands are used for image segmentation as the best representation for the spectral data in Section III, and the LDA and Gabor bands are used as alternative features, in addition to the PCA bands, for object detection in Section IV.

### III. IMAGE SEGMENTATION

The proposed segmentation algorithm combines spectral information from the original data with structural information extracted through morphological operations. These two complementary types of information are incorporated into a hierarchical structure, and a generic iterative algorithm is used to extract meaningful segments from this hierarchy by simultaneously optimizing spectral homogeneity and neighborhood connectivity. Considering the fact that different structures may appear more clearly in different bands, we analyze each band separately. The following sections describe the details of the algorithm. Parts of this section were presented in [15].

#### A. Morphological Profiles

We use mathematical morphology to exploit structural information. In particular, morphological opening and closing operations are used to model structural characteristics of pixel neighborhoods. These operations are known to isolate structures that are brighter and darker than their surroundings, respectively. Contrary to opening (respectively, closing), opening by reconstruction (respectively, closing by reconstruction) preserves the shape of the structures that are not removed by erosion (respectively, dilation). In other words, image structures that the SE cannot be contained are removed while others remain.

The opening and closing by reconstruction operations are applied by using increasing SE sizes to generate multiscale characteristics called morphological profiles. The derivative of the morphological profile (DMP) [4] is defined as a vector where the measure of the slope of the opening–closing profile is stored for every step of an increasing SE series. Pesaresi and Benediktsson [4] used the structural information encoded in the DMP for segmenting remote sensing images. They defined an image segment as a set of connected pixels showing the greatest value of the DMP for the same SE size. That is, the segment label of each pixel is assigned according to the scale corresponding to the largest derivative of its profile. Their scheme works well in images with moderate resolution, where the structures in the image are mostly flat so that all pixels in a structure have only one derivative maximum. A drawback of this scheme is that neighborhood information is not used while assigning segment labels to pixels. This often results in numerous small noisy segments in very high resolution images with nonflat structures where the scale with the largest value of the DMP may not correspond to the true structure (see Fig. 4 for an illustration). In our approach, we do not consider pixels

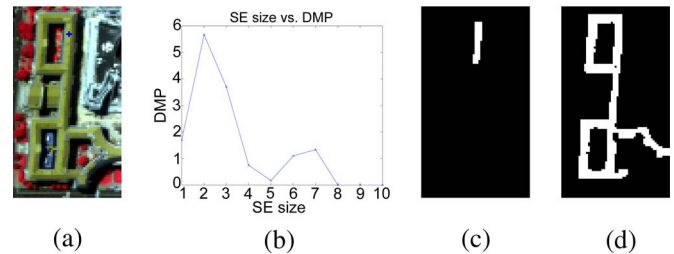


Fig. 4. Greatest value in the DMP of the pixel marked with a blue + in (a) is obtained for the SE size 2 [derivative of the opening profile is shown in (b)]. Subpanel (c) shows the segment that we would obtain if we label the pixels with the SE size corresponding to the greatest DMP. The segment in (d) that occurs with SE size 3 is more preferable as a complete structure, but it does not correspond to the scale of the greatest DMP for all pixels inside the segment. (a) Example pixel. (b) DMP of the pixel. (c) Segment for SE size 2. (d) Segment for SE size 3.

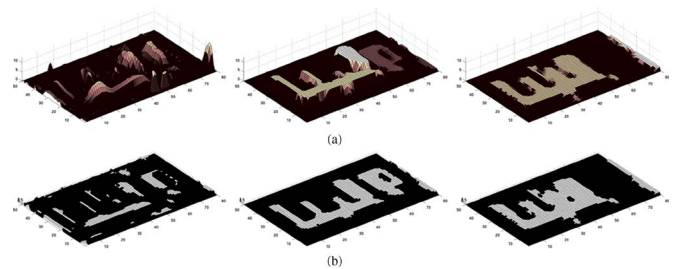


Fig. 5. (a) Example opening DMP at three scales. (b) Pixels whose DMP values are greater than zero (thresholding at  $DMP > 0$ ). Each connected component at each scale is a candidate segment for the final segmentation.

alone while assigning segment labels. Instead, we also take into account the behavior of the neighbors of the pixels.

#### B. Hierarchical Segment Extraction

In our segmentation approach, our aim is to determine the segments by applying opening and closing by reconstruction operations. We assume that pixels with a positive DMP value at a particular SE size face a change with respect to their neighborhoods at that scale. As opposed to [4], where only the scale corresponding to the greatest DMP is used, the main idea is that a neighboring group of pixels that have a similar change for any particular SE size is a candidate segment for the final segmentation. These groups can be found by applying connected components analysis to the DMP at each scale (see Fig. 5 for an illustration).

Considering the fact that different structures have different sizes, we apply the opening and closing by reconstruction using SEs in increasing sizes that are from 1 to  $m$  (radius of the disk). However, a connected component appearing for a small SE size may be appearing because of the heterogeneity and geometrical complexity of the scenes and of other external effects such as shadows producing texture effects in images and resulting in structures that can be one to two pixels wide [4]. In this case, there is, most probably, a larger connected component appearing at the scale of a larger SE to which the pixels of those noise components belong. On the other hand, a connected component that corresponds to a true structure in the final segmentation may also appear as part of another component at larger SE sizes. The reason is that a meaningful connected component may start merging with its surroundings and other

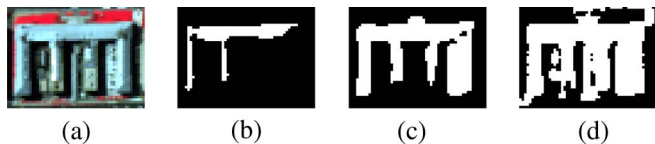


Fig. 6. Example connected components for a building structure. These components appear for SE sizes 3, 5, and 6, respectively, in the derivative of the opening profile of the second PCA band. (a) False-color image. (b) A small connected component that is part of (c). (c) The preferred connected component. (d) A large component where (c) started merging with others.

connected components after the SE size in which it appears is reached. Fig. 6 shows these cases.

For each opening and closing profile, through increasing SE sizes from 1 to  $m$ , each morphological operation reveals connected components that are contained within each other in a hierarchical manner, where a pixel may be assigned to more than one connected component appearing at different SE sizes as in Fig. 6. We treat each component as a candidate meaningful segment. Using these segments, a tree is constructed where each connected component is a node and there is an edge between two nodes corresponding to two consecutive scales if one node is contained within the other. Leaf nodes represent the components that appear for the SE size 1. Root nodes represent the components that exist for the SE size  $m$ . Fig. 7 shows a part of an example tree constructed by candidate meaningful segments appearing in five levels. Considering that we use a finite number of SE sizes, there may be more than one root node. In this case, there will be more than one tree, and the algorithms described in the next section are run on each tree separately.

### C. Segment Selection

After forming a tree for each opening and closing profile, our aim is to search for the most meaningful connected components among those appearing at different scales in the segmentation hierarchy. With a similar motivation in [11], Tilton analyzed hierarchical image segmentations and selected the meaningful segments manually. Then, Plaza and Tilton [16] investigated how the different spectral, spatial, and joint spectral/spatial features of segments change from one level to another in a segmentation hierarchy, with the goal of automating the selection process in the future. Alternatively, Klaric *et al.* [17] thresholded the DMP to obtain candidate objects and applied heuristics, such as thresholds, on the aspect ratio of the bounding box to accept a candidate as a building object. In this paper, each node in the tree is treated as a candidate segment in the final segmentation, and selection is done automatically as described in the following.

Ideally, we expect a meaningful segment to be as spectrally homogeneous as possible. However, in the extreme case, a single pixel is the most homogeneous. Hence, we also want a segment to be as large as possible. In general, a segment stays almost the same (both in spectral homogeneity and size) for some number of SEs and then faces a large change at a particular scale, which is either because it merges with its surroundings to make a new structure or because it is completely lost. Consequently, the size we are interested in corresponds to the scale right before this change. In other words, if the nodes on a path in the tree stay homogeneous until some node

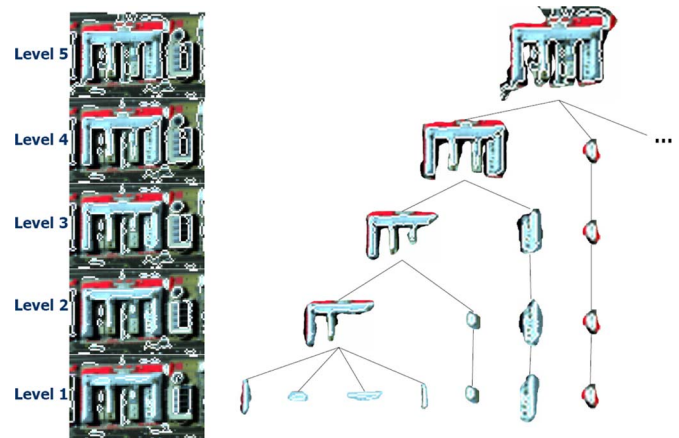


Fig. 7. Example tree where each candidate segment is a node.

$n$  and if the homogeneity is lost in the next level, we say that  $n$  corresponds to a meaningful segment in the hierarchy.

With this motivation, to check the meaningfulness of a node, we define a measure consisting of two factors, namely: spectral homogeneity, which is calculated in terms of variances of spectral features, and neighborhood connectivity, which is calculated by using the sizes of connected components. Then, starting from the leaf nodes (level 1) up to the root node (level  $m$ ), we compute this measure at each node and select a node as a meaningful segment if it is a highly homogeneous and large enough node on its path in the hierarchy (a path corresponds to the set of nodes from a leaf to the root).

In order to calculate the homogeneity factor in a node, we use the fact that pixels in a correct structure should not only have similar morphological profiles but also similar spectral features. Thus, we calculate the homogeneity of a node as the standard deviation of the spectral information of the pixels in the corresponding segment. The spectral information for the DC Mall and Pavia data sets consists of the PCA bands, whereas the RGB bands are used for the Ankara data set. The PCA components are used instead of the full hyperspectral data because they achieve dimensionality reduction and provide the best summarization of spectral data in the least-squares sense. The LDA bands are not used because their computation requires labeled data, and we want the segmentation step to be fully unsupervised. The rest of the algorithm is generic; thus, it is independent of which features are used to compute the spectral homogeneity.

In examining a node from the leaf up to the root in terms of homogeneity, we do not directly use the standard deviation of the node. Instead, we consider the difference of the standard deviation of that node and its parent. What we expect is a sudden increase in the standard deviation. When the standard deviation does not change much, it usually means that small sets of pixels are added to the segment or some noise pixels are cleaned. When there is a large change, it means that the structure merged with a larger structure or it merged with other irrelevant pixels, disturbing the homogeneity in the node. Hence, the difference of the standard deviation in the node's parent and the standard deviation in the node should be maximized while selecting the most meaningful nodes.

The computation of the standard deviation of the multispectral data of a node is done by projecting these data onto a 1-D

representation [18]. Let the number of spectral bands be  $d$ . The basis used for the 1-D representation is selected as the vector connecting the mean of the original  $d$ -dimensional data for the pixels of the current node and the mean of the data for its parent. The projection of the  $d$ -dimensional data onto this vector, which can be considered to separate the nodes in the spectral space, is computed using inner products, and the standard deviation of the resulting 1-D data is computed for each node. This formulation exploits the multivariate information contained in the multispectral bands while computing the standard deviation. We also tested a formulation using the average of the standard deviations computed from individual bands, but there was no visual difference in the results compared with the ones given in this paper.

As discussed previously, using only the spectral homogeneity factor will favor small structures. To overcome this problem, the number of pixels in the segment corresponding to the node is introduced as another factor to create a tradeoff. As a result, the goodness measure  $M$  for a node  $n$  is defined as

$$M(n) = D(n, \text{parent}(n)) \times C(n) \quad (1)$$

where the first term is the standard deviation difference between the node's parent and itself and the second term is the number of pixels in the node. The node that is relatively spectrally homogeneous and large enough will maximize this measure and will be selected as a meaningful segment. Other linear and nonlinear combinations of homogeneity and scale can be incorporated for calculating the goodness measure. However, we use the simplest combination in (1) to avoid the introduction of new parameters.

Given the value of the goodness measure for each node, we find the most meaningful segments as follows. Suppose that  $\mathcal{T} = (\mathcal{N}, \mathcal{E})$  is the tree, with  $\mathcal{N}$  as the set of nodes and  $\mathcal{E}$  as the set of edges. The leaf nodes are in level 1, and the root node is in level  $m$ . Let  $\mathcal{P}$  denote the set of all the paths from the leaves to the root,  $M(n)$  as the measure at node  $n$ , and  $\text{descendant}(n)$  as the descendant nodes of node  $n$ . We select  $\mathcal{N}^* \subseteq \mathcal{N}$  as the final segmentation with the following conditions:

- 1)  $\forall a \in \mathcal{N}^*, \forall b \in \text{descendant}(n),$   
 $M(a) \geq M(b);$
- 2)  $\forall a \in \mathcal{N} \setminus \mathcal{N}^*,$   
 $\exists b \in \text{descendant}(n) : M(a) < M(b);$
- 3)  $\forall a, b \in \mathcal{N}^*,$   
 $\forall p \in \mathcal{P} : a \in p \rightarrow b \notin p,$   
 $\forall p \in \mathcal{P} : b \in p \rightarrow a \notin p;$
- 4)  $\forall p \in \mathcal{P},$   
 $\exists a \in p : a \in \mathcal{N}^*.$

The first condition requires that any node in  $\mathcal{N}^*$  must have a measure greater than all of its descendants. The second condition requires that no node in  $\mathcal{N} \setminus \mathcal{N}^*$  has a measure greater than all of its descendants. The third condition requires that any two nodes in  $\mathcal{N}^*$  cannot be on the same path (i.e., the corresponding segments cannot overlap in the hierarchical segmentation). The fourth condition requires that every path must include a node that is in  $\mathcal{N}^*$ .

We use a two-pass algorithm for selecting the most meaningful nodes ( $\mathcal{N}^*$ ) in the tree. The bottom-up (first) pass aims

to find the nodes whose measure is greater than all of its descendants (condition 1). The algorithm first marks all nodes in level 1. Then, starting from level 2 up to the root level, it checks whether each node in each level has a measure greater than or equal to those of all of its children. The greatest measure, which is seen so far in each path, is propagated to upper levels so that it is enough to check only the immediate children rather than all descendants in order to find whether a node's measure is greater than or equal to that of all of its descendants'.

After the bottom-up pass marks all such nodes, the top-down (second) pass seeks to select the nodes that also satisfy the remaining conditions (2, 3, and 4). It starts by marking all nodes as "selected" in the root level if they are marked by the bottom-up pass. Then, in each level until the leaf level, the algorithm checks for each node whether it is marked in the bottom-up pass while none of its ancestors is marked. If this condition is satisfied, it marks the node as selected. Finally, the algorithm selects the nodes that are marked as selected in each level as meaningful segments. The pseudocode for the selection algorithm is shown in Algorithms 1–3.

#### Algorithm 1 Segment-Selection Algorithm

```

Run Bottom-Up algorithm
Run Top-Down algorithm
for each level  $l = 1$  to  $m$  do
  for each node  $n$  in level  $l$  do
    select  $n$  as a meaningful segment if it is marked as
    selected
  end for
end for

```

#### Algorithm 2 Bottom-Up Algorithm

```

Mark all nodes in level 1
for each level  $l = 2$  to  $m$  do
  for each node  $n$  in level  $l$  do
    if  $M(n) \geq \max\{M(a) | a \in \text{children}(n)\}$  then
      mark  $n$ 
    else
       $M(n) = \max\{M(a) | a \in \text{children}(n)\}$ 
      leave  $n$  unmarked
    end if
  end for
end for

```

#### Algorithm 3 Top-Down Algorithm

```

Mark all nodes in level  $m$  as selected if they are already
marked in Bottom-Up
for each level  $l = m - 1$  to 1 do
  for each node  $n$  in level  $l$  do
    if  $\text{parent}(n)$  is marked as selected or parent-selected
      then
        mark  $n$  as parent-selected
      else
        if  $\text{parent}(n)$  is not marked in Top-Down and  $n$  is
        not marked in Bottom-Up, then
          leave  $n$  unmarked
        else
          mark  $n$  as selected
        end if
      end if
    end for
  end for

```

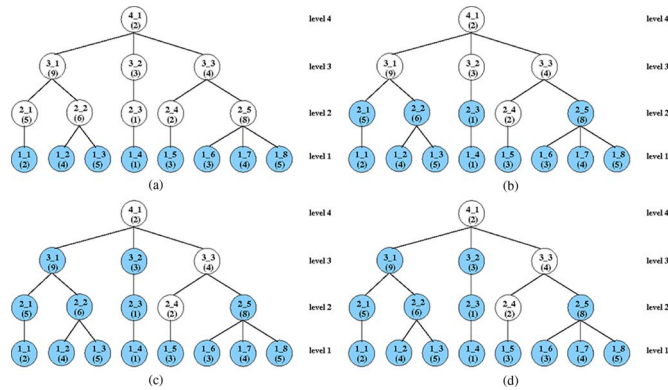


Fig. 8. Example run of the Bottom-Up algorithm on a sample tree. Beginning from the leaves until the root, the nodes whose measures are greater than all of the descendants (satisfying condition 1) are colored with blue in each step. (a) Step 1. (b) Step 2. (c) Step 3. (d) Step 4.

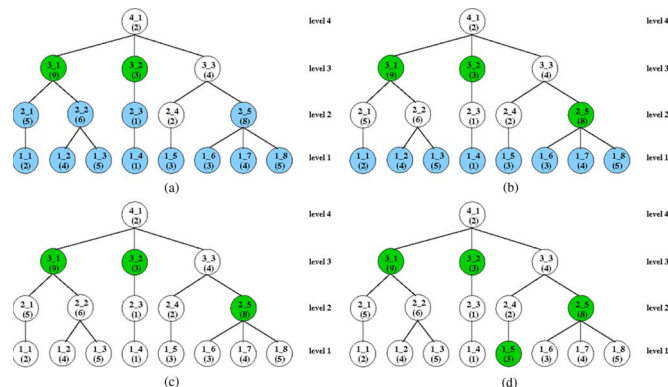


Fig. 9. Example run of the Top-Down algorithm on the tree in Fig. 8(d). Beginning from the root until the leaves, the nodes marked in the Bottom-Up algorithm which also satisfy the remaining conditions (2, 3, and 4) are marked with green in each step. When the algorithm ends, the green nodes are selected as the most meaningful nodes in the tree. (a) Step 1. (b) Step 2. (c) Step 3. (d) Step 4.

An example run of these algorithms is illustrated using a sample tree where the nodes are labeled as  $i_j$ , with  $i$  denoting the node's level and  $j$  denoting the number of the node from left to right in that level. A value for the goodness measure is given in parenthesis for each node. Figs. 8 and 9 show the marked nodes in each step of the Bottom-Up and the Top-Down algorithms, respectively. During the Bottom-Up algorithm, each node  $1_j$  ( $1 \leq j \leq 8$ ) is marked in the beginning. Then, as we move upwards, nodes  $2_1$ ,  $2_2$ ,  $2_3$ , and  $2_5$  in level 2 and nodes  $3_1$  and  $3_2$  in level 3 are marked because the measure of each of them is greater than or equal to those of all of its descendants. Then, we run the Top-Down algorithm and mark nodes  $3_1$ ,  $3_2$ ,  $2_5$ , and  $1_5$ , satisfying the four conditions defined previously, as selected.

After selecting the most meaningful connected components in each opening and closing tree separately, the next step is to integrate the resulting connected components. A problem may occur when two connected components, one being selected from the opening tree and the other being selected from the closing tree, intersect. In this case, the intersecting part is assigned to the connected component whose goodness measure is greater.

TABLE I  
TOTAL NUMBER OF SEGMENTS OBTAINED BY USING DIFFERENT SEGMENTATION ALGORITHMS FOR INDIVIDUAL BANDS. HS, PROPOSED HIERARCHICAL SEGMENTATION ALGORITHM; PB, PESARESI-BENEDIKTSSON ALGORITHM; AND WS, WATERSHED SEGMENTATION

	DC Mall			Pavia			Ankara		
	PCA1	PCA2	PCA3	PCA1	PCA2	PCA3	Red	Green	Blue
HS	647	710	713	1420	1432	1255	341	353	359
PB	30240	28097	280206	65296	63394	61965	16258	18057	15840
WS	14640	13653	1962	21817	14143	6537	7009	6335	4666

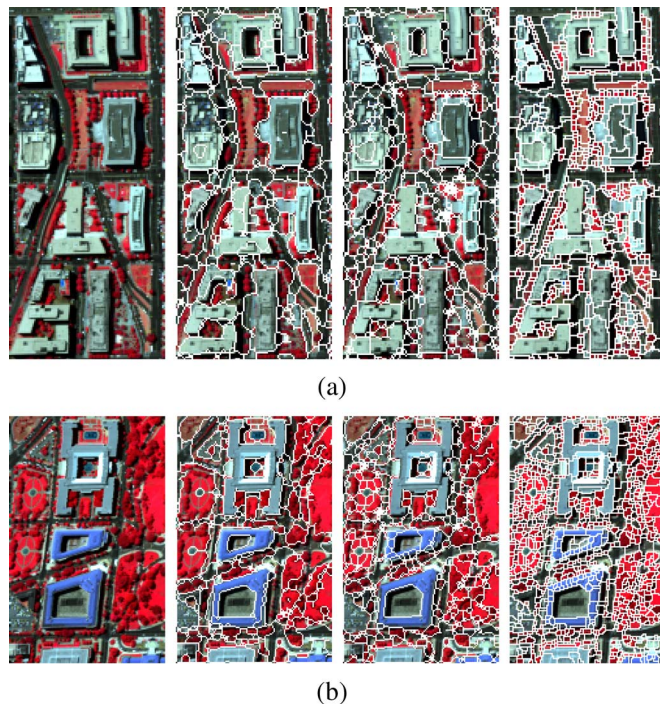


Fig. 10. Example segmentation results for the DC Mall data set. From left to right: false color, result of the proposed approach, result of Pesaresi-Benediktsson, and result of watershed segmentation.

#### D. Evaluation of Segmentation

We applied the proposed hierarchical segmentation algorithm to all three data sets described in Section II. Disk SEs, with radii from 3 to 15, were used for both opening and closing profiles constructed for each spectral band (3 PCA bands for DC Mall and Pavia and three RGB bands for Ankara). The tree structure was separately constructed for each band, and the segments were independently selected from each tree. The same bands were also segmented by using the Pesaresi-Benediktsson algorithm [4] that defines an image segment as a set of connected pixels showing the greatest value of the derivative morphological profile for the same SE size and the watershed segmentation [19] that uses the gradient of an image as input after suppressing small local extrema to avoid severe oversegmentation. The same parameters were used for all data sets for a given algorithm.

Table I shows the total number of segments obtained by using all three algorithms. Figs. 10–12 show example segmentations for the DC Mall, Pavia, and Ankara data sets, respectively. We present the zoomed versions of the results for several example areas to better illustrate the details for high-resolution imagery and for clarity of the presentation of this paper. Considering

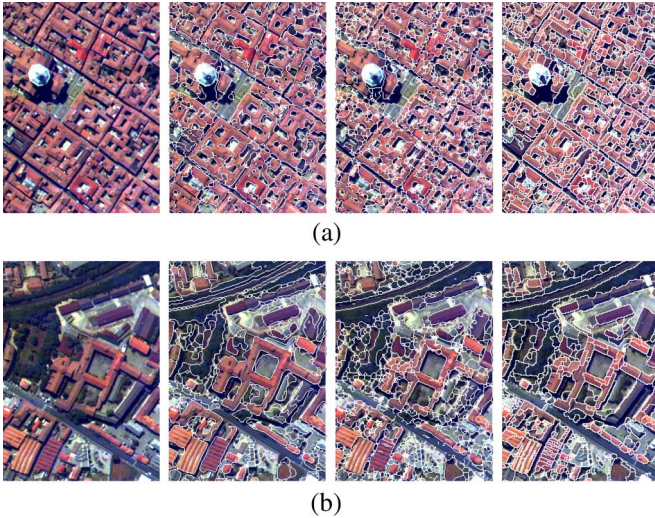


Fig. 11. Example segmentation results for the Pavia data set. From left to right: false color, result of the proposed approach, result of Pesaresi-Benediktsson, and result of watershed segmentation.

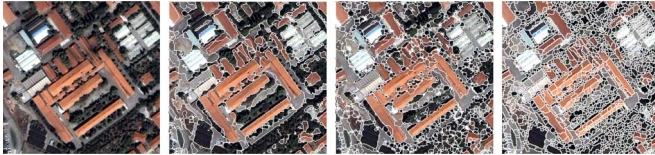


Fig. 12. Example segmentation results for the Ankara data set. From left to right: RGB color, result of the proposed approach, result of Pesaresi-Benediktsson, and result of watershed segmentation.

that there is no detailed object-level geographic information system (GIS) vector data available, only qualitative evaluation is done for segmentation. The results show that our segmentation algorithm is able to detect structures in the image that are more precise and more meaningful than the structures detected by the compared approaches. The oversegmentation produced by the Pesaresi-Benediktsson algorithm occurred because the segment label assignment is done individually for each pixel by only considering the greatest value in its DMP. Thus, noisy pixels that are different from their neighborhoods may produce small segments because they may have large values occurring at scales corresponding to small SE sizes. Similarly, although a prefiltering to suppress the small local extrema of the gradient is applied, the watershed segmentation algorithm still produces oversegmentation that is commonly observed in the literature. However, our algorithm considers both the morphological characteristics encoded in the DMP and the spectral homogeneity measured in terms of the standard deviation within contiguous groups of pixels. It also considers the consistency of these values within neighboring pixels forming large connected components. As a result, the combined measure that uses both spectral and structural information is both robust to noise and consistent within detailed structures in high-resolution images. In all of the examples, our algorithm is able to extract many meaningful structures as whole segments.

Note that it is possible to improve some of the segments by tuning the parameters of the Pesaresi-Benediktsson algorithm (e.g., specifying different set of scales) and the watershed segmentation algorithm (e.g., threshold for eliminating small local extrema; the same parameters were used for all algorithms

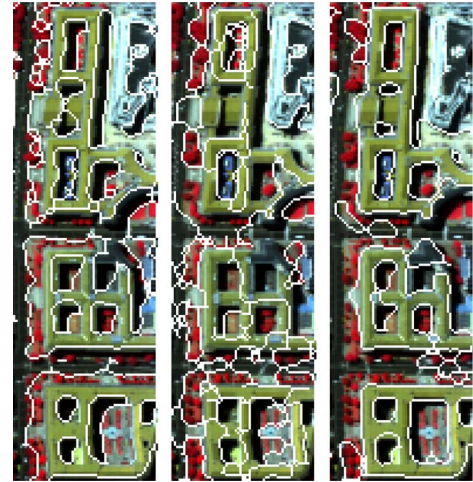


Fig. 13. Example segmentation results for different PCA bands of the DC Mall data set. The left, middle, and right images show the segments extracted in the first, second, and third PCA bands, where the roads/shadows, trees, and buildings are detected more clearly, respectively.

for all data sets in the experiments). However, we observed that different parameters needed to be manually selected for different bands of different data sets, and the parameters that performed well for one band of a data set could give very bad results for other bands and other data sets. On the other hand, the proposed segment extraction and segment selection algorithms are free from parameters (except for the number of scales  $m$  used to construct the range of SE sizes with fixed unit increments for the morphological profile for segment extraction) and can automatically select the meaningful segments at different scales and sizes in the hierarchy for different spectral bands and different data sets in a completely unsupervised process without any need for parameter tuning.

Another important observation is that different structures are extracted more clearly in different spectral bands. In particular, buildings can be detected accurately in one band, but roads, trees, fields, paths, and shadows can be detected accurately in other bands. For example, the structures in Fig. 10(a) and (b) are found in the second PCA band of the DC Mall data set. On the other hand, the structures in Fig. 11(a) and (b) are found in the third PCA band of the Pavia data set. Fig. 13 shows the extracted segments in different PCA bands of the DC Mall data set. The reason why a particular structure is better extracted in a particular band is that the pixels belonging to that structure are found lighter or darker than their surroundings on that band. This motivates the next step to automatically integrate the results from individual bands as a final segmentation with detected objects in an image.

#### IV. OBJECT DETECTION

In Section III, we described a method that used the neighborhood and spectral information, as well as the morphological information, for segmentation. In this section, we present an unsupervised algorithm for the automatic selection of segments from multiple segmentations and spectral bands (parts of this section were presented in [20]). The input to the algorithm is a set of hierarchical segmentations corresponding to different spectral bands. The goal is to find coherent groups of segments that correspond to meaningful structures. The assumption here



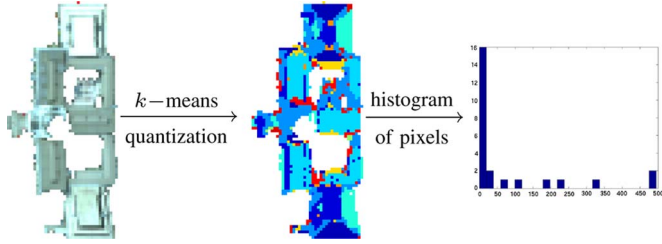


Fig. 14. Each segment is modeled by using the statistical summary of its pixel content. In the experiments, these summaries are obtained by quantizing the feature values using the  $k$ -means algorithm and representing the distribution of these quantized values in a histogram.

is that for a particular structure (e.g., building), the “good” segments (i.e., the ones containing a building) will all have similar features, whereas the “bad” segments (i.e., the ones containing multiple objects or corresponding to overlapping partial object boundaries) will be described by a random mixture of features. Therefore, considering multiple objects/structures of interest, this selection process can also be seen as a grouping problem within the space of a large number of candidate segments obtained from multiple hierarchical segmentations. The resulting groups correspond to different types of objects in the image.

#### A. Modeling Segments

The grouping algorithm consists of three steps, namely: extracting segment features, grouping segments, and detecting objects. In the first step, each segment is modeled by using the statistical summary of its pixel content. First, all pixels in the image are clustered by applying the  $k$ -means algorithm in a feature domain. This corresponds to the quantization of the feature values. Then, a histogram is constructed for each segment to approximate the distribution of these quantized values belonging to the pixels in that segment, as shown in Fig. 14. This histogram is used to represent the segment in the rest of the algorithm. Alternative representations include the use of the mean or the covariance of the feature values of the pixels within a segment. However, the mean is often not sufficient to distinguish complex objects, and the covariance estimation can have singularity problems for small sample sizes. The histogram model provides a tradeoff that contains more information than the mean while being easier to estimate than the covariance. Furthermore, the segment selection algorithm in Section III-C uses a goodness measure that selects the segments that are large enough so that the histograms can be reliably estimated. Note that the object detection algorithm is generic in the sense that any discrete model of the segment’s content can also be used by the grouping algorithm in the next section.

#### B. Grouping Segments

In this paper, we use the PLSA algorithm [13] to solve the grouping problem. PLSA was originally developed for statistical text analysis to discover topics in a collection of documents that are represented by using the frequencies of words from a vocabulary. In our case, the documents correspond to image segments, the word frequencies correspond to the histograms of pixel-level features, and the topics to be discovered correspond to the set of objects/structures of interest in the image.

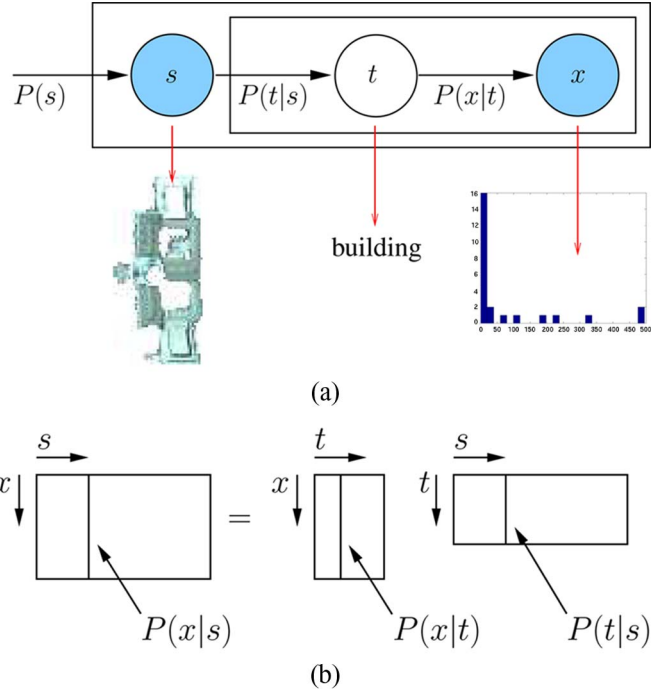


Fig. 15. (a) PLSA graphical model. The filled nodes indicate observed random variables, whereas the unfilled node is unobserved. The red arrows show examples for the measurements represented at each node. (b) In PLSA, the object-specific feature probability  $P(x_j|t_k)$  and the segment-specific object probability  $P(t_k|s_i)$  are used to compute the segment-specific feature probability  $P(x_j|s_i)$ .

Russell *et al.* [21] used a different graphical model in a similar setting where multiple segmentations of natural images were obtained by using the normalized cut algorithm by changing its parameters, and instances of segments corresponding to objects such as cars, bicycles, faces, sky, etc., were successfully grouped and retrieved from a large set of images.

The PLSA technique uses a graphical model for the joint probability of the segments and their features in terms of the probability of observing a feature given an object and the probability of an object given the segment. Suppose there are  $N$  segments (documents) having content coming from a distribution (vocabulary) with  $M$  discrete pixel feature values (words). The collection of segments is summarized in an  $N$ -by- $M$  co-occurrence table  $n$ , where  $n(s_i, x_j)$  stores the number of occurrences of the feature value  $x_j$  in segment  $s_i$ . In addition, there is a latent object type (topic) variable  $t_k$  associated with each observation; an observation being the occurrence of a feature in a particular segment.

The graphical model used by PLSA to model the joint probability  $P(x_j, s_i, t_k)$  is shown in Fig. 15. The generative model  $P(s_i, x_j) = P(s_i)P(x_j|s_i)$  for the feature content of segments can be computed by using the conditional probability

$$P(x_j|s_i) = \sum_{k=1}^K P(x_j|t_k)P(t_k|s_i) \quad (2)$$

where  $P(x_j|t_k)$  denotes the object-conditional probability of the feature  $x_j$  occurring in object  $t_k$ ,  $P(t_k|s_i)$  denotes the probability of the object  $t_k$  observed in segment  $s_i$ , and  $K$  is the number of object types. Then, the object-specific feature

distribution  $P(x_j|t_k)$  and the segment-specific feature distribution  $P(x_j|s_i)$  can be used to determine the similarities between object types and segments (as explained in the next section).

In PLSA, the goal is to identify the probabilities  $P(x_j|t_k)$  and  $P(t_k|s_i)$ . These probabilities are learned by using the EM algorithm [13]. In the E step, the posterior probability of the latent variables are computed based on the current estimates of the parameters as

$$P(t_k|s_i, x_j) = \frac{P(x_j|t_k)P(t_k|s_i)}{\sum_{l=1}^K P(x_j|t_l)P(t_l|s_i)}. \quad (3)$$

In the M step, the parameters are updated to maximize the expected complete data log-likelihood as

$$P(x_j|t_k) = \frac{\sum_{i=1}^N n(s_i, x_j)P(t_k|s_i, x_j)}{\sum_{m=1}^M \sum_{i=1}^N n(s_i, x_m)P(t_k|s_i, x_m)} \quad (4)$$

$$P(t_k|s_i) = \frac{\sum_{j=1}^M n(s_i, x_j)P(t_k|s_i, x_j)}{\sum_{j=1}^M n(s_i, x_j)}. \quad (5)$$

The E and the M steps are iterated until the difference between the consecutive expected complete data log-likelihoods is less than a threshold or the number of iterations exceeds a predetermined value.

### C. Detecting Objects

After learning the parameters of the model, we want to find good segments belonging to each object type learned. This is done by comparing the feature distribution within each segment  $p(x|s)$  and the feature distribution for a given object type  $p(x|t)$ . The similarity between two distributions can be measured using the Kullback–Leibler (KL) divergence  $D(p(x|s)||p(x|t))$ . Then, for each object type, the segments in an image can be sorted according to their KL divergence scores, and the most representative segments for that object type can be selected. However, if there are two segments that are extracted from different spectral bands grouped within the same object type and at least one of them overlaps with the other by a predetermined percent of its whole area, the less representative structure (the one with a larger KL divergence score) is removed from that object type to avoid having multiple segments of the same object.

### D. Evaluation of Object Detection

The performance of the PLSA-based segment grouping and object detection depends on the choice of the features that are used to model the segments and the number of object types that is given as the input to the grouping algorithm. Clustering evaluation measures can be used to study the effects of different settings of the parameters and provide an objective and quantitative evaluation of the unsupervised grouping/detection algorithms described in the previous sections.

In the literature, clustering is often used as an intermediate step of a classification/recognition system, where only the performance of the final system is analyzed or where, usually, only a qualitative visual inspection of the clustering results are performed. To evaluate the accuracy of object detection,

first, quality measures for both individual object types (clusters) and the overall detection (clustering) must be defined. One way of defining these measures involves the use of ground truth data, where the resulting groups are compared with the manually assigned labels for the segments. In other words, the quality measures should quantify how well the results of the unsupervised detection algorithm reflect the groupings in the ground truth.

In an optimal result, the segments with the same object class labels in the ground truth must be assigned to the same group (cluster), and the segments corresponding to different object class types must appear in different groups (clusters) at the end of the detection process. An information theoretic criterion that measures the homogeneity of the distribution of the segments with respect to different object types is the entropy [22]. Another measure is the Rand index [23] that is analogous to the Kappa coefficient and measures the agreement between two labelings. These two measures are described in the following.

1) *Entropy*: Let  $h_{ck}$  denote the number of segments assigned to the object type (cluster)  $k$  with a ground truth object class label  $c$ ,  $h_c = \sum_{k=1}^K h_{ck}$  denote the number of segments with a ground truth object class label  $c$ , and  $h_{.k} = \sum_{c=1}^C h_{ck}$  denote the number of segments assigned to object type (cluster)  $k$ , where  $K$  is the number of object types given as input to the grouping/detection algorithm and  $C$  is the true number of objects. The quality of individual clusters is measured in terms of the homogeneity of the true object class labels within each cluster. For each cluster  $k$ , the cluster entropy  $E_k$  is given by

$$E_k = - \sum_{c=1}^C \frac{h_{ck}}{h_{.k}} \log \frac{h_{ck}}{h_{.k}}. \quad (6)$$

Then, the overall cluster entropy  $E_{\text{cluster}}$  is given by a weighted sum of individual cluster entropies as

$$E_{\text{cluster}} = \frac{1}{\sum_{k=1}^K h_{.k}} \sum_{k=1}^K h_{.k} E_k. \quad (7)$$

A smaller cluster entropy value indicates a higher homogeneity. However, the cluster entropy continues to decrease as the number of clusters increases. To overcome this problem, another entropy criterion that measures how the segments of the same true object class are distributed among the clusters can be defined. For each true object class  $c$ , the class entropy  $E_c$  is given by

$$E_c = - \sum_{k=1}^K \frac{h_{ck}}{h_c} \log \frac{h_{ck}}{h_c}. \quad (8)$$

Then, the overall class entropy  $E_{\text{class}}$  is given by a weighted sum of individual class entropies as

$$E_{\text{class}} = \frac{1}{\sum_{c=1}^C h_c} \sum_{c=1}^C h_c E_c. \quad (9)$$

Unlike the cluster entropy, the class entropy increases when the number of clusters increases. Therefore, the two measures can

be combined for an overall entropy measure as

$$E = \beta E_{\text{cluster}} + (1 - \beta) E_{\text{class}} \quad (10)$$

where  $\beta \in [0, 1]$  is a weight that balances the two measures [22].

2) *Adjusted Rand Index*: When the number of true object classes and the number of detected object types are the same, the Kappa coefficient can be used to measure the amount of agreement between the two labelings. However, the number of detected clusters in an unsupervised classification is not always the same as the true number of objects. In such cases, the Rand index can be used to measure the agreement of every pair of segments according to both unsupervised and ground truth labelings [23]. The agreement occurs if two segments that belong to the same class are put into the same cluster or if two segments that belong to different classes are put into different clusters. The Rand index is computed as the proportion of all segment pairs that agree in their labels. The index has a value between 0 and 1, where 0 indicates that the two labelings do not agree on any pair of segments and 1 indicates that the two labelings are exactly the same.

However, the expected value of the Rand index of two random groupings does not take a constant value. The adjusted Rand index [23], which can be computed as

$$R = \frac{\sum_{c=1}^C \sum_{k=1}^K \binom{h_{ck}}{2} - \left[ \sum_{c=1}^C \binom{h_{c.}}{2} \sum_{k=1}^K \binom{h_{.k}}{2} \right] / \binom{N}{2}}{\left[ \sum_{c=1}^C \binom{h_{c.}}{2} + \sum_{k=1}^K \binom{h_{.k}}{2} \right] / 2 - \left[ \sum_{c=1}^C \binom{h_{c.}}{2} \sum_{k=1}^K \binom{h_{.k}}{2} \right] / \binom{N}{2}} \quad (11)$$

where  $N$  is the total number of segments, has a maximum value of 1 and an expected value of 0. Therefore, it has a wider range and more sensitivity than the original index. This index is also analogous to the Kappa coefficient because it measures the agreement over and above that expected by chance [23].

3) *Ground Truth for Object Detection*: Considering that suitable detailed GIS data and object-level ground truth are not available in the form of individual segments, we use the pixel-level ground truth to generate the object labels for evaluation. The pixel-level ground truth that we manually created and is shown in Fig. 16(a) is used for this purpose. Given all segments that are used for object detection, a segment is assigned an object class label if at least 20% of its pixels have an overlap with the pixel-level ground truth and at least 50% of those pixels have the same label. The first threshold handles the areas where the pixel-level ground truth is not available. The second threshold ensures that the majority of the segment belongs to the same object. These two thresholds are selected empirically to obtain an object-level ground truth, with a coverage as much as possible, by making use of the pixel-level ground truth as much as possible. These object labels for segments are used to perform a quantitative evaluation of the unsupervised object detection.

## V. EXPERIMENTS

The qualitative evaluation of the proposed image-segmentation algorithm with comparative experiments was presented in Section III-D. The input to the unsupervised

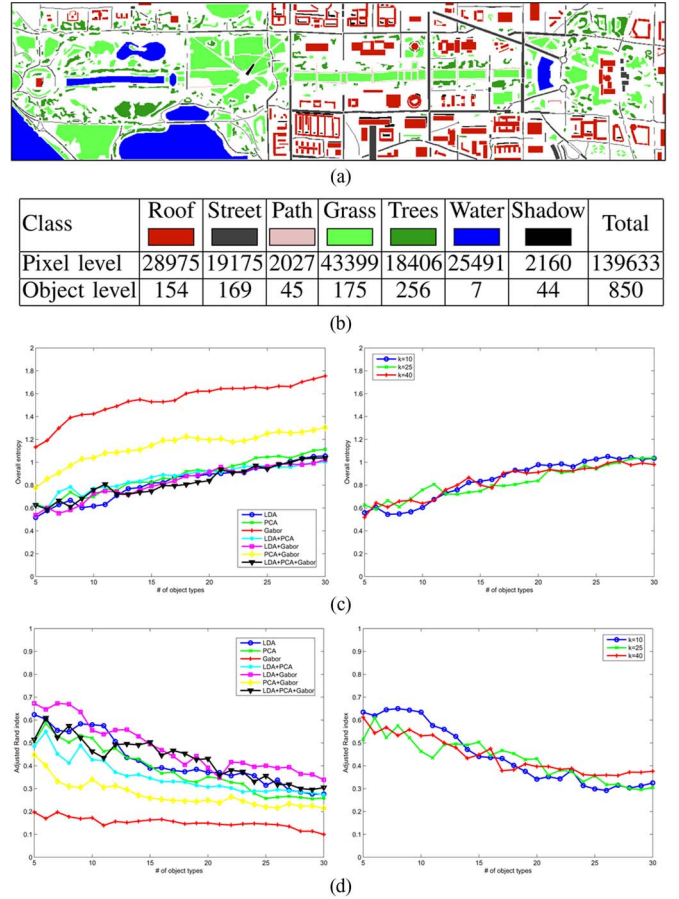


Fig. 16. Reference map and the statistics of the ground truth used to compute the object detection performance indexes. The plots in (c) and (d) show the indexes (left) for different settings of the features for  $k = 25$  and (right) for different number of quantization levels ( $k = 10, 25, 40$ ) when PCA, LDA, and Gabor features were used. The  $x$ -axes show the number of object types ( $K$ , number of clusters) given as input to the detection algorithm. (a) Reference map (which is rotated). (b) Ground truth statistics (number of pixels and segments in each class). (c) Entropy versus number of object types. (d) Adjusted Rand index versus number of object types.

object detection algorithm is the set of all segments extracted from individual bands of an image, where the goal is to find coherent groups of segments that correspond to different objects. The total number of segments that are automatically extracted was 2070, 4107, and 1053 for the DC Mall, Pavia, and Ankara data sets, respectively.

The next step is the modeling of the segments using histograms of quantized feature values. All seven possible combinations of three different types of features (PCA, LDA, and Gabor), for the DC Mall and Pavia data sets, and three possible combinations of two different types of features (RGB and Gabor), for the Ankara data set, were used as described in Section II. The pixels in each image were quantized using the  $k$ -means algorithm, where the number of quantization levels ( $k$ ) was set to three different values (10, 25, and 40) to study the effects of quantization. Then, for each segment, a histogram with  $k$  bins was constructed by counting the number of pixels belonging to each quantization level within that segment, as described in Section IV-A.

Next, the PLSA algorithm was used to learn the object-conditional feature distributions for all object types. In the

experiments, the number ( $K$ ) of latent object type variables ( $t_k$ ) was varied from 5 to 60 with increments of 1. The parameters of the distribution models were learned by using the EM algorithm for each setting, as described in Section IV-B.

In the final step, the KL divergence score between each segment and each object type was computed, and the segments were grouped as belonging to the object type where the KL score was the smallest. Considering that the segments were extracted from different bands, some of the segments could overlap. When the overlap between two segments belonging to the same group was more than 30% of the area of one of the segments, the one with a larger KL divergence score was removed, as described in Section IV-C.

Quantitative performance evaluation for object detection was performed for the DC Mall data set using the performance indexes described in Section IV-D. We extended the original pixel-level ground truth containing only 8079 pixels that we obtained with the hyperspectral data [24] to increase its coverage as much as possible [shown in Fig. 16(a)]. Fig. 16 shows the entropy and Rand indexes with respect to the different settings of the features, the number of quantization levels, and the number of object types (number of clusters). When individual cluster and class entropy values are analyzed in detail, we can see that the cluster entropy continued to decrease, as expected, as the number of clusters increased because purer clusters were obtained when the segments were divided into a larger number of groups. On the other hand, the class entropy tend to stay flat for a very small number of clusters and started to increase when the number of clusters became greater than the number of true object classes ( $C = 7$  for this data set) because the segments belonging to the same object class were divided into numerous groups, diversifying the distribution of the class over the clusters. The turning point for the overall entropy occurred when the number of clusters was approximately equal to the number of true object classes ( $K = 8$  or  $9$ ), after which it continued to increase because the increase in the class entropy was greater than the decrease in the cluster entropy.

The Rand index measures the agreement between the detected and true segment labels using two components, namely: the number of segment pairs that belong to the same class and put into the same cluster and the number of segment pairs that belong to different classes and put into different clusters. In the experiments, the former number tend to stay flat for a very small number of clusters and started to decrease, as expected, when the number of clusters became greater than the number of true object classes because the segments of the same object type were divided into different clusters. On the other hand, the latter number continued to increase, as expected, as the number of clusters increased. The overall Rand index followed the former number because its decrease was more significant than the increase in the latter. Note that the adjusted Rand index values above 30% correspond to an agreement of above 80% between the detected and ground truth labels of every pair of segments according to the definition of the original Rand index in [23].

When the effects of different feature combinations are analyzed, we can see that individual LDA features performed better than the PCA and Gabor features. This is expected because pixel-level class labels are used to extract these fea-

tures so that they maximize class separability, whereas the PCA and Gabor features are computed by using unsupervised techniques. The feature combinations that include the LDA features also performed better than other combinations for the same reasons. The Gabor features were not as effective as the others because such features are generally useful for large textured areas, such as vegetation, but many building segments did not gain additional information from texture because their support (area) were usually too small compared with the sizes of the texture filters. When the PCA (spectral) features are compared with the LDA features, the latter were very effective in distributing segments that belong to different object classes, such as buildings, vegetation, roads, etc., to different groups (clusters), whereas the former were powerful in distinguishing buildings with different types of roofs (all buildings were in the same ground truth class called roof). Therefore, they provide complementary information and, usually, the best results were obtained when they were used together; however, the results with only the PCA features were also acceptable and show that the grouping technique is very powerful for unsupervised object detection even when no manual label information is available.

When the effect of the number of quantization levels is analyzed, we did not observe any significant difference between  $k = 10, 25$ , or  $40$ . Finally, although there were some differences in the performance index values for different feature combinations, the visual inspection of the resulting groups showed that these differences occurred mostly because of different groupings of the small tree and grass segments. Although the major and relatively larger segments, such as buildings or roads, were similarly grouped with different feature combinations, there were some differences in the grouping of small vegetation segments (particularly when  $K$  was increased) that were larger in number compared with other object types, and this caused some differences in the performance indexes.

Note that the quantitative performance indexes actually provide a “pessimistic” estimate of the actual performance because, for example, two different kinds of buildings with different roof types that are correctly put into different clusters by the object detection algorithm will decrease the Rand index if those two visually different buildings have the same ground truth label as “roof.” Although the ground truth may not contain that much detail, visual evaluation confirms that such cases can be correctly handled by our algorithm. This also shows that there may not be a single best value for  $K$  but one can interpret the plots, such as in Fig. 16, by looking at the significant changes in the performance indexes that occur when a significant new cluster (group) is produced when the number of object types is increased. Therefore, a natural extension of this unsupervised grouping process is the selection and labeling of the detected groups by the user for a final classification with a desired level of detail.

In addition to the entropy and Rand indexes, we also evaluated the performance of the proposed object detection algorithm using precision and recall to measure how well the detected objects correspond to the ground truth objects. Given the segment groups (clusters) obtained using the unsupervised detection algorithm for a specific value of  $K$ , first, we manually identified the groups containing roads, buildings, and vegetation according to the content of the majority of the segments using visual

TABLE II  
NUMBER OF AUTOMATICALLY DETECTED SEGMENTS FOR ROADS,  
BUILDINGS, AND VEGETATION OBJECTS AND THE CORRESPONDING  
PRECISION AND RECALL VALUES (AS PERCENTAGES)  
FOR DIFFERENT NUMBER OF CLUSTERS ( $K$ )

$K$	Roads			Buildings			Vegetation		
	Detect	Prec	Recall	Detect	Prec	Recall	Detect	Prec	Recall
5	498	33.53	86.98	240	75.42	75.32	945	56.19	93.90
15	428	35.98	86.39	295	67.80	75.32	874	59.04	92.49
25	326	46.63	80.47	325	66.15	85.71	945	55.98	93.19

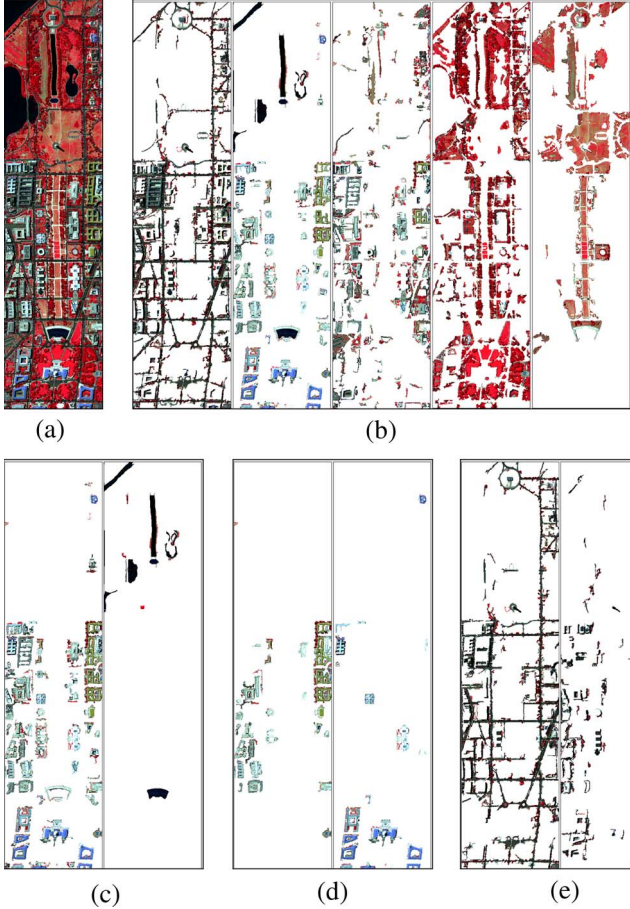


Fig. 17. Examples of object detection for the DC Mall data set. (a) False color. (b) Clusters when  $K = 5$ . The groupings are already meaningful for a small  $K$ . (c) Two new clusters introduced when  $K = 8$ . The second cluster in (b) is divided into building and water segments. (d) Two new clusters introduced when  $K = 10$ . The building segments in the first cluster in (c) are further divided according to their roof types. (e) Two new clusters introduced when  $K = 17$ . The first cluster in (b) is divided into road and shadow segments.

inspection. For a segment to be accepted as a correct detection, it must have a sufficient overlap with an object in the ground truth. Then, we used the object-level ground truth described in Section IV-D and, for each object type, computed precision and recall as

$$\text{precision} = \frac{\# \text{ of correctly detected objects (segments)}}{\# \text{ of all detected objects (segments)}} \quad (12)$$

$$\text{recall} = \frac{\# \text{ of correctly detected objects (segments)}}{\# \text{ of all objects in the ground truth}} \quad (13)$$

Recall can be interpreted as the number of true positive objects extracted by the algorithm, whereas precision evaluates the

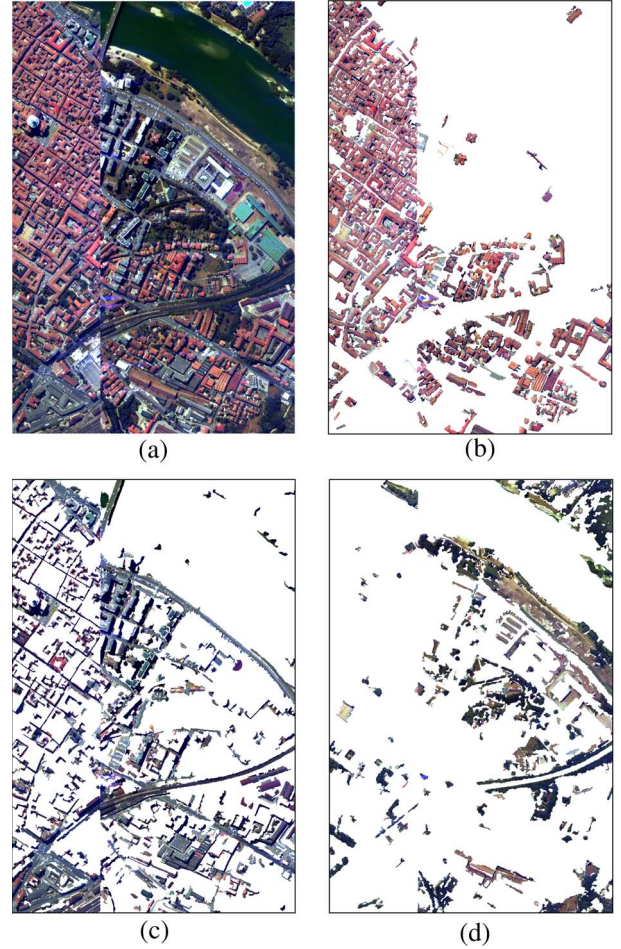


Fig. 18. Examples of object detection for the Pavia data set when  $K = 30$ . (a) False color. (b) Buildings. (c) Roads. (d) Vegetation.

tendency of the algorithm for false positives [17]. The results for three different values of  $K$  are shown in Table II. We believe that the results are quite satisfactory, given the complexity of the data and the unsupervised nature of the algorithm used.

Fig. 17 shows example groups obtained for the DC Mall data set when the PCA, LDA, and Gabor features were used with 25 quantization levels. The clusters for  $K = 5$  contained roads/shadow, buildings/water, buildings/soil, trees/grass, and grass and were already quite meaningful for a small  $K$ . When  $K$  was increased, the trees and grass segments started separating further. At  $K = 8$ , the water segments separated from the buildings.<sup>1</sup> At  $K = 10$ , the building segments started separating into different clusters according to their roof types. A further increase in  $K$  caused the grass segments to be divided into more clusters (e.g., greener segments versus browner segments). At  $K = 17$ , most of the shadow segments separated from the roads. Larger values of  $K$  produced small clusters that contained small tree or grass segments because there were more such segments compared with segments of other types.

<sup>1</sup>Some large water segments do not appear in the results because small structuring elements (maximum radius of 15) were used in the morphological profile because we are mainly interested in smaller structures such as buildings. All water segments can be extracted by using larger structuring elements. Alternatively, the simple thresholding of spectral bands can detect water segments before running our object detection algorithm.

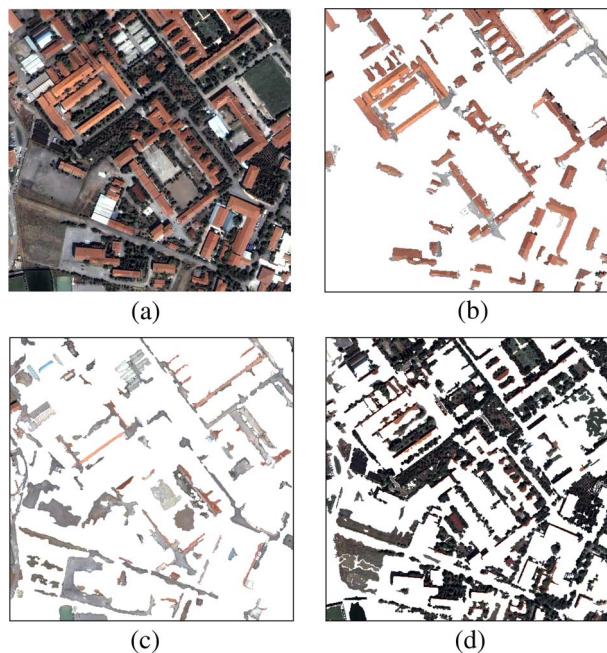


Fig. 19. Examples of object detection for the Ankara data set when  $K = 5$ . (a) RGB. (b) Buildings. (c) Roads. (d) Vegetation.

Similarly, Figs. 18 and 19 show example results for the Pavia and Ankara data sets, respectively. Due to space limitations, instead of individual clusters, the segments belonging to the groups that mostly contain buildings, roads, and vegetation are shown. For the Pavia image, all buildings with tile roofs were grouped together with almost no false alarms. There were some minor confusions between the roads and some shadow segments and buildings with very similar colors. For the Ankara image, the clusters were almost complete when  $K = 5$ . Overall, the quantitative evaluation using performance indexes and the qualitative visual inspection of the detection results for all data sets confirmed that the proposed algorithms were able to identify the segments corresponding to objects (i.e., “good” segments) by placing them into coherent groups in an unsupervised mode, where there is a strong correlation between the true object labels and the detected segment labels.

## VI. CONCLUSION

We presented novel methods for unsupervised image segmentation and automatic object detection in high-resolution remotely sensed imagery. Our segmentation algorithm exploited structural information using morphological operators. These operators were applied to each spectral band separately, where candidate segments were extracted by applying connected components analysis to the pixels selected according to their morphological profiles. These segments were hierarchically modeled by using a tree, and the most meaningful ones in this hierarchy were selected by optimizing a criterion that consisted of two factors, namely: spectral homogeneity and neighborhood connectivity. The segment selection algorithm is generic in the sense that not only can other criteria for a “good” (meaningful) segment be directly incorporated, but it can also be used with other hierarchical segmentation algorithms.

We evaluated the proposed approach qualitatively on three data sets. The results showed that our method that considers

morphological characteristics, spectral information, and their consistency within neighboring pixels is able to detect structures in the image which are more precise and more meaningful than the structures detected by two popular approaches that do not make strong use of neighborhood and spectral information jointly.

We also proposed an object detection algorithm that formulated the detection process as an unsupervised grouping problem for the automatic selection of coherent sets of segments corresponding to meaningful structures among a set of candidate segments from multiple hierarchical segmentations obtained from individual spectral bands. The grouping problem was solved by using the PLSA algorithm that built object models by learning the object-conditional feature probability distributions. The automatic labeling of a segment was done by comparing its spectral and textural content distributions with the distribution of the learned object models. The object detection algorithm is generic in the sense that any model for a segment’s content can be used by the grouping algorithm. Extensive performance evaluation showed that the proposed methods are able to automatically detect and group structures belonging to the same object classes.

## ACKNOWLEDGMENT

The authors thank Dr. D. A. Landgrebe and L. L. Biehl of Purdue University, for the DC Mall data set, Dr. P. Gamba of the University of Pavia, Italy, for the Pavia data set, and the Scientific and Technological Research Council of Turkey, for the Ankara data set.

## REFERENCES

- [1] S. Aksoy, “Spatial techniques for image classification,” in *Signal and Image Processing for Remote Sensing*, C. H. Chen, Ed. New York: Taylor & Francis, 2006, pp. 489–511.
- [2] J. A. Benediktsson, J. A. Palmason, and J. R. Sveinsson, “Classification of hyperspectral data from urban areas based on extended morphological profiles,” *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 3, pp. 480–491, Mar. 2005.
- [3] S. Bhagavathy and B. S. Manjunath, “Modeling and detection of geospatial objects using texture motifs,” *IEEE Trans. Geosci. Remote Sens.*, vol. 44, no. 12, pp. 3706–3715, Dec. 2006.
- [4] M. Pesaresi and J. A. Benediktsson, “A new approach for the morphological segmentation of high-resolution satellite imagery,” *IEEE Trans. Geosci. Remote Sens.*, vol. 39, no. 2, pp. 309–320, Feb. 2001.
- [5] F. Melgani and S. B. Serpico, “A Markov random field approach to spatio-temporal contextual image classification,” *IEEE Trans. Geosci. Remote Sens.*, vol. 41, no. 11, pp. 2478–2487, Nov. 2003.
- [6] L. Bruzzone and L. Carlin, “A multilevel context-based system for classification of very high spatial resolution images,” *IEEE Trans. Geosci. Remote Sens.*, vol. 44, no. 9, pp. 2587–2600, Sep. 2006.
- [7] S. Aksoy and H. G. Akçay, “Multi-resolution segmentation and shape analysis for remote sensing image classification,” in *Proc. 2nd Int. Conf. Recent Advances Space Technol.*, Istanbul, Turkey, Jun. 9–11, 2005, pp. 599–604.
- [8] A. Katartzis, I. Vanhamel, and H. Sahli, “A hierarchical Markovian model for multiscale region-based classification of vector-valued images,” *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 3, pp. 548–558, Mar. 2005.
- [9] L. K. Soh, C. Tsatsoulis, D. Gineris, and C. Bertoia, “ARKTOS: An intelligent system for SAR sea ice image classification,” *IEEE Trans. Geosci. Remote Sens.*, vol. 2, no. 1, pp. 229–248, Jan. 2004.
- [10] S. Aksoy, K. Koperski, C. Tusk, G. Marchisio, and J. C. Tilton, “Learning Bayesian classifiers for scene classification with a visual grammar,” *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 3, pp. 581–589, Mar. 2005.
- [11] J. C. Tilton, “Analysis of hierarchically related image segmentations,” in *Proc. IEEE GRSS Workshop Advances Tech. Anal. Remotely Sensed Data*, Washington DC, Oct. 27–28, 2003, pp. 60–69.

- [12] A. Darwish, K. Leukert, and W. Reinhardt, "Image segmentation for the purpose of object-based classification," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, Toulouse, France, Jul. 21–25, 2003, vol. 3, pp. 2039–2041.
- [13] T. Hofmann, "Unsupervised learning by probabilistic latent semantic analysis," *Mach. Learn.*, vol. 42, no. 1/2, pp. 177–196, Jan./Feb. 2001.
- [14] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. Hoboken, NJ: Wiley, 2000.
- [15] H. G. Akçay and S. Aksoy, "Morphological segmentation of urban structures," in *Proc. 4th IEEE GRSS/ISPRS Joint Workshop Remote Sens. Data Fusion Over Urban Areas*, Paris, France, Apr. 11–13, 2007, pp. 1–6.
- [16] A. J. Plaza and J. C. Tilton, "Automated selection of results in hierarchical segmentations of remotely sensed hyperspectral images," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, Seoul, Korea, Jul. 25–29, 2005, vol. 7, pp. 4946–4949.
- [17] M. Klaric, G. Scott, C.-R. Shyu, and C. Davis, "Automated object extraction through simplification of the differential morphological profile for high-resolution satellite imagery," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, Seoul, Korea, Jul. 25–29, 2005, vol. 2, pp. 1265–1268.
- [18] G. Hamerly and C. Elkan, "Learning the  $k$  in  $k$ -means," in *Proc. Neural Inf. Process. Syst.*, Vancouver, BC, Canada, Dec. 8–13, 2003, pp. 281–288.
- [19] P. Soille, *Morphological Image Analysis*, 2nd ed. New York: Springer-Verlag, 2002.
- [20] H. G. Akçay and S. Aksoy, "Automated detection of objects using multiple hierarchical segmentations," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, Barcelona, Spain, Jul. 23–27, 2007, pp. 1468–1471.
- [21] B. C. Russell, A. A. Efros, J. Sivic, W. T. Freeman, and A. Zisserman, "Using multiple segmentations to discover objects and their extent in image collections," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, New York, Jun. 17–22, 2006, vol. 2, pp. 1605–1614.
- [22] J. He, A.-H. Tan, C.-L. Tan, and S.-Y. Sung, "On quantitative evaluation of clustering systems," in *Information Retrieval and Clustering*, W. Wu, H. Xiong, and S. Shekhar, Eds. Norwell, MA: Kluwer, 2003.
- [23] B. S. Everitt, S. Landau, and M. Leese, *Cluster Analysis*, 4th ed. London, U.K.: Oxford Univ. Press, 2001.
- [24] D. A. Landgrebe, *Signal Theory Methods in Multispectral Remote Sensing*. Hoboken, NJ: Wiley, 2003.



**Selim Aksoy** (S'96–M'01) received the B.S. degree from the Middle East Technical University, Ankara, Turkey, in 1996 and the M.S. and Ph.D. degrees from the University of Washington, Seattle, in 1998 and 2001, respectively.

He has been an Assistant Professor at the Department of Computer Engineering, Bilkent University, Ankara, since 2004, where he is also the Co-Director of the RETINA Vision and Learning Group. Before joining Bilkent, he was a Research Scientist at Insightful Corporation, Seattle, where he

was involved in image understanding and data mining research sponsored by the National Aeronautics and Space Administration, the U.S. Army, and the National Institutes of Health. During 1996–2001, he was a Research Assistant at the University of Washington, where he developed algorithms for content-based image retrieval, statistical pattern recognition, object recognition, graph-theoretic clustering, user relevance feedback, and mathematical morphology. During the summers of 1998 and 1999, he was a Visiting Researcher at the Tampere International Center for Signal Processing, Tampere, Finland, collaborating in a content-based multimedia retrieval project. His research interests include computer vision, statistical and structural pattern recognition, machine learning and data mining with applications to remote sensing, medical imaging, and multimedia data analysis.

Dr. Aksoy is a member of the IEEE Geoscience and Remote Sensing Society, the IEEE Computer Society, and the International Association for Pattern Recognition (IAPR). He received the CAREER Award from the Scientific and Technological Research Council of Turkey in 2004. He was one of the Guest Editors of the special issue of IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING on Pattern Recognition in Remote Sensing in 2007. He served as the Vice Chair of the IAPR Technical Committee 7 on Remote Sensing during 2004–2006 and is currently the Chair of the same committee for 2006–2008.



**H. Gökhan Akçay** received the B.S. and M.S. degrees in computer engineering from Bilkent University, Ankara, Turkey, in 2004 and 2007, respectively. He is currently working toward the Ph.D. degree in computer engineering at Bilkent University.

Since September 2004, he has been a Teaching and Research Assistant at the Department of Computer Engineering, Bilkent University. His research interests include statistical and structural pattern recognition, computer vision, and remote sensing image analysis.