The Seventeenth Int. Conf. On Machine Learning (ICML-2000), Palo Alto, CA., 2000

Using Learning by Discovery to Segment Remotely Sensed Images

Leen-Kiat Soh
Information and Telecommunications
Technology Center (ITTC)
University of Kansas
Lawrence, KS 66044
lksoh@ittc.ukans.edu

Costas Tsatsoulis
Department of Electrical Engineering
and Computer Science (EECS)
University of Kansas
Lawrence, KS 66045
tsatsoul@ittc.ukans.edu

Abstract

In this paper, we describe our research in computer-aided image analysis. We have incorporated machine learning methodologies with traditional image processing to perform unsupervised image segmentation. First, we apply image processing techniques to extract from an image a set of training cases, which are histogram peaks described by their intensity ranges and spatial and textural attributes. Second, we use learning by discovery methodologies to cluster these cases. The first methodology we use is based on COBWEB/3, a conceptual clustering approach whose objective is to cluster the cases incrementally as the concept hierarchy is refined. The second methodology is based on an Aggregated Population Equalization (APE) strategy. This approach attempts to maintain similar strengths for all populations in its environment. The clustering result of either approach tells us the number of visually significant classes in the image (and what these classes are) and thus enables us to perform unsupervised segmentation, i.e., the labeling of all image pixels. Based on the results of the visual evaluation of the segmented images, we have built an unsupervised segmentation software tool called ASIS and have applied it to a range of remotely sensed images such as sea ice and vegetation index. In this paper, we present our machine learning approach to unsupervised image segmentation and discuss our experiments and their results.

1 INTRODUCTION

Image segmentation is a process of pixel classification where the image is segmented into subsets by assigning individual pixels into classes (Rosenfeld and Kak, 1982). An unsupervised technique implies automated operation independent of human intervention during the execution of the algorithm.

Remotely sensed images of natural scenes are inherently noisy, have a highly dynamic makeup, and lack homogeneous structures. In addition, remotely sensed data is typically voluminous. Hence, computer-aided analyses such as unsupervised segmentation are very important in improving the efficiency and consistency in image understanding.

We have designed an image segmentation methodology that automatically segments remotely sensed images into significant classes—determining the number of classes and describing what the classes are. One of the most important factors in unsupervised segmentation is the determination of the number of classes. Hence, we have turned to learning by discovery to achieve that objective. During the development phase, we have used two discovery learning methodologies: COBWEB/3 (Gennari *et al.*, 1990) and the Aggregated Population Equalization (APE) strategy (Soh, 1998). The former is an incremental conceptual clustering learning algorithm while the latter is a self-organization approach. Each of the above methodologies can determine the number of classes in the data and then cluster the data into the classes without human intervention.

Our approach is to first extract training cases from an image. The training cases are derived from a set of histogram peaks of the image. Each case consists of an intensity range, a set of spatial attributes, and a set of textural attributes. Note that these training cases are designed to capture the visual cues that human photo-interpreters use when they manually inspect and analyze images. Then, we feed these cases into a discovery mechanism (either COBWEB/3 or APE). The output of the mechanism is a clustering that groups the training cases into separate clusters. The clustering provides us with two important pieces of information on how to segment the image: (1) the number of classes, and (2) what the classes are. Finally, equipped with the information, we label all pixels in the image. Our approach therefore is able to learn how to segment an image by analyzing a set of derived training cases and by applying the learned clustering strategy to all pixels in the image.

We have built an unsupervised segmentation software tool called ASIS after the approach described above, and have applied it to a variety of remotely sensed images, particularly Synthetic Aperture Radar (SAR) sea ice images.

In the following, we first present some related work in unsupervised segmentation. In the third section, we describe our methodology, from the image processing techniques that extract and describe the training cases to the use of the learning-by-discovery techniques. In the fourth section, we discuss our experiments and results. In Section 5, we touch upon our ASIS system and show an example. Finally, we conclude the paper.

2 BACKGROUND

In general, the analysis of remotely sensed images of natural scenes differs from that of urban, commercial or agricultural areas, and from medical and industrial imagery taken in controlled environments. Natural scenes (forests, mountains, the seas, clouds, etc.) are not structured and cannot be represented easily by regular rules or grammars. In addition, the appearance of natural objects can vary greatly based on the geographic area, the season, and the past and current weather conditions. These factors complicate the unsupervised image segmentation task in remote sensing.

Several unsupervised image segmentation techniques have been proposed, such as iterative dominance clustering (Goldberg and Shlien, 1978), random field models (Panjwami and Healey, 1995), fuzzy clustering (Nguyen and Cohen, 1993), local Bayesian (Peng and Pieczynski, 1995), and maximum likelihood (Cohen and Fan, 1992). However, these techniques deal with either less complex scenes or highly textured regions. Hence, they are not readily extensible to our domain that works with highly complex, poorly structured, and poorly textured remote sensing imagery such as SAR sea ice images.

There are several noted discovery learning approaches applied to image segmentation. For example, ISODATA (Holt *et al*., 1989) and (fuzzy) K-means (Huntsberger *et al.,*1985, Bezdek and Trivedi, 1986) are based on numerical taxonomy; histogram smoothing (Smith, 1996) is based on speckle noise model of remotely-sensed imagery; AutoClass (Cheeseman *et al.,* 1990) is based on Bayesian probabilities; COBWEB/3 is based on conceptual clustering and incremental learning; non-linear regression (Acton, 1996) is based on the regularization theory; multi-thresholding (O'Gorman, 1994) is based on peaks in the imagery; SNOB (Wallace and Dowe, 1994) is based on minimum message length; and some are based on the self-organizing behavior of neural networks (Kohonen, 1989). Most of these techniques use a similar approach that allows for their automation. The algorithm first uses an initial number of classes to find clusters of data, then evaluates the clustering based on an optimization metric, and repeats with another number of classes. Finally, the algorithm selects the number of classes with the best score. Since some of these techniques are computationally expensive, several authors have introduced assumptions, reductions, and local optimizations. In our research, we use COBWEB/3 because it is fast (due to its incremental data learning), it does not determine an initial number of clusters, and it represents explicitly a concept hierarchy for data learning and understanding. We also developed our own clustering methodology called APE, which we compare to COBWEB/3 in this paper.

3 METHODOLOGY

The overall methodology of our approach is depicted in Figure 1. First, we extract a histogram from the original image. The histogram is based on regional bisectors of the image. Second, using a multiresolution approach, we obtain from the histogram a set of significant peaks, which become the basis of our training cases. We then describe these cases with spatial and textural attributes. Fourth, we feed the cases into the discovery mechanism. After obtaining the clustering result, we perform post-processing to resolve conflicts and refine clusters. The final clustering tells us of what the number of classes are and what the classes are in the image. That *discovered* knowledge enables us to finally label all pixels in the image.

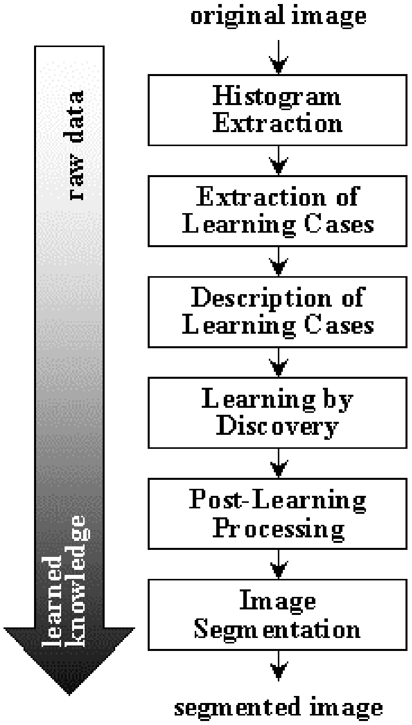

Figure 1. The block diagram of our unsupervised segmentation via learning by discovery

3.1 HISTOGRAM EXTRACTION

The objective of the histogram extraction phase is to transform the image data into a form from which training cases can be more accurately and easily derived. We use dynamic local thresholding (Haverkamp *et al.* 1995) to achieve this objective. Briefly, the input image is divided into smaller, overlapping regions for each a regional histogram is computed. For each region that has a high variance, a bimodal Gaussian curve approximation is performed to curve-fit the region's histogram. From the parameters of the curve, the valley-to-peak ratio can be computed. For each region that has a high ratio, a maximum likelihood method is used to compute the optimal bisector. The collection of all bisectors becomes the histogram from which peaks will be extracted later. We use dynamic local thresholding to combat inherent speckle noise in satellite images and to reduce range effects caused by angles of the radar at the near-end and the far-end of the image. A detailed treatment of our adaptation of dynamic local thresholding can be found in (Soh, 1998).

3.2 EXTRACTION OF TRAINING CASES

After obtaining the histogram, we use a multiresolution peak detection technique to extract significant peaks as the basis of training cases. First, we create a map of a number of cumulative distribution functions (*cdf*) (at different resolutions) of the histogram. At each resolution, we use the zero-crossings and local extrema to locate peaks—i.e., jumps in the *cdf* curve. At the end of the localization process, we have a multiresolution contour of the peaks, which we evaluate through a contour tracking process. The criteria we use are: (1) peaks found at a low-level resolution are more significant than the peaks found at a high-level resolution, (2) peaks found in high-level resolution are more accurate in terms of localization than the peaks found in low-level resolution, (3) a peak that is surrounded by neighboring peaks is a dominant peak, and (4) the significance of a peak is proportional to its height. After tracking, we identify peaks that have scores above a threshold as the significant peaks of the image.

Next we derive training cases from the peaks. Each training case is the intensity range between a pair of successive peaks. For example, suppose the system extracts four significant peaks: 35, 47, 55, and 60. Thus, we have the following five training cases: TC1, TC2, …, TC5, where TC1's intensity range is (0,35), TC2's intensity range is (35,47), TC3's is (47,55), TC4's is (55,60), and TC5's is (60,255), and 0 is the minimum intensity and 255 is the maximum intensity of an 8-bit image.

The *cdf*-based peak detection has been used to perform image segmentation (Sezan, 1990). By combining it with the multiresolution approach, we make the system noise-resistant and facilitate its automation. A detailed treatment of our multiresolution approach can be found in (Soh and Tsatsoulis, 1999b).

3.3 DESCRIPTION OF TRAINING CASES

After extraction, each training case is known only by its range along the intensity axis. We need to further describe the cases such that the discovery mechanism can learn from the training cases and their associated attributes to form clusters. We use two sets of attributes: spatial and textural.

3.2.1 Spatial Attributes

We use a spatial matrix to document the spatial relationships a training case has with all other training cases. To compute the matrix, we use a running 3x3 window on the image. The pixels in the window are tagged respectively to the range or training case along the intensity axis that they belong to. Then, we compute the number of times a pixel in the range of *TC1* has another *TC1*-tagged pixel as a spatial neighbor, *TC2* as a spatial neighbor, and so on. As a result, given *N* training cases, we build an *N*x*N* matrix in which each entry is the frequency of a case being a spatial neighbor to another case, including itself, as shwon in Table 1. By observing this matrix, one can visualize how the training cases *behave* in the image. A very compact training case will have a high frequency of having itself as a spatial neighbor (e.g., TCN). A parasitic training case will have a high frequency of neighboring another training case while having a weak core itself (e.g., TC2).

Table 1. A Spatial Matrix. TC1 neighbors with itself 86.17% of the time, with TC2 10.35%, and so on

	TC1	TC2	TC3	…	TCN
TC1	0.8617	0.1035	0.0044	…	0.0000
TC2	0.7924	0.1199	0.0343	…	0.0000
TC3	0.5871	0.3319	0.0550	…	0.0001
…	…	…	…	…	…
TCN	0.0000	0.0002	0.0002	…	0.9035

3.2.2 Textural Attributes

Textures have often been used to represent and analyze regions in remotely sensed images (Holmes *et al.*, 1984, Nystuen and Garcia, 1992, Chou *et al.*, 1994). In our research, we use the gray-level co-occurrence matrices (Haralick *et al.* 1973) to define textures such as energy, contrast, correlation, homogeneity, entropy, autocorrelation, dissimilarity, and maximum probability. Since textures can only be measured meaningfully over a sizeable region (e.g, 32x32), we use the overlapping regions outlined during the histogram extraction phase. First, we perform a bilinear interpolation to propagate regional bisectors to all regions. Second, we tag each region to a training case if its bisector falls into the intensity range of that training case. Third, we compute the aforementioned textural attributes for each region, and collect the measurements for each training case. Finally, we average each measurement for every training case to arrive at an *N*x8 textural matrix, as shown in Table 2.

Table 2. A Textural Matrix

	energy	contrast	corr.	…	max. pro.
LE1	0.023	412.058	-1.609	…	0.067
LE2	0.028	327.036	-2.186	…	0.081
LE3	0.033	349.111	-2.548	…	0.096
…	…	…	…	…	…
LEN	0.139	211.771	-10.022	…	0.327

3.3 LEARNING BY DISCOVERY

At the end of the description process, each training case is complete with an intensity range, a set of *N* spatial attributes, and a set of eight textural attributes. Now, we are ready to discover clusters from the set of training cases. To learn by discovery, we have experimented with two different approaches. The first approach is based on the incremental, conceptual clustering of COBWEB/3, while the second on the self-organization of the Aggregated Population Equalization or APE concept (Soh 1998).

3.3.1 Conceptual Clustering

According to (Gennari *et al.,* 1990), much of human learning can be viewed as a succession of events from which one induces a hierarchy of concepts that summarize and organize his or her experience. In conceptual clustering, the label (or class) of each instance (or training case) is not known *a priori* to the program. In order to cluster the cases into different groups or concepts, conceptual clustering observes their attributes and incrementally refines its concept hierarchy. In our research, we use COBWEB/3 (Thompson and McKusick, 1993) which deals with both nominal and normal features.

COBWEB/3 examines its cases sequentially and learns the concepts incrementally. Thus, the order of the training cases plays a role in the final structure of the concept hierarchy. Though COBWEB/3 uses merging and splitting operations to re-partition hierarchy upon receiving new cases, it is not able to fully eliminate the effects of early commitment of a case to a cluster, especially when the set of training cases is small. Our adaptation is to arrange the set of training cases in two exactly opposite orders, execute COBWEB/3 twice for each image, and perform conflict resolution during the post-learning processing.

To increase the role of the intensity range of a training case, we have also imposed a constraint on two operations in COBWEB/3: the placement of a case into an existing cluster and the merging procedure. A placement is considered detrimental to the concept hierarchy if the intensity range of a case does not fit in a sequence among the cases already accepted into the cluster. Likewise, a merging of two existing clusters with non-successive intensity ranges weakens the concept hierarchy. As a result, COBWEB/3's learning puts a higher weight on grouping cases with similar intensity ranges together than those with similar spatial or textural makeup.

3.3.2 Self-Organization

In our work, we also developed a clustering algorithm called the Aggregated Population Equalization (APE). In a set of populations, each population is related to another (including itself) in *K* dimensions. The dimensions could be the average spatial distance, correlation, entropy, etc. Each dimension describes how a population behaves in the set. In our imagery domain, we describe populations in terms of image pixels. A centralized population has large communities of pixels concentrated at various places in the image. A pixel at the core of this type of population is usually shielded and has no contact with pixels of other populations. The pixels of a scattered population, on the other hand, reside in the image in small, yet noticeable, groups, and these groups are distant from each other. If a population is interspersed, that means where the member of the population is present, there is usually another population in the mix, and thus it has a weak population core. Finally, a parasitic population does not scatter unrestrictedly; instead its pixels usually linger along the fringes of another population. These pixels do not have a strong population core, and they actually have more contacts with pixels of other populations than with themselves. The APE concept describes these different populations and decides which two populations to merge and which population to split.

The basic methodology of APE is straightforward. Populations that are not strong form alliances and unite to become a stronger *aggregated* population. On the other hand, a population can be subjected to population disintegration if its is overly dominant or diverse. The result is a group of smaller populations. The Aggregated Population Equalization (APE) is the process of obtaining an equilibrium of strong and weak populations such that every aggregated population is similarly strong. In this manner, the populations self-organize themselves into significant clusters. As a result, APE learns the number of clusters and what the clusters are through this form of discovery.

The APE algorithm has some analogues in the real world. For population aggregation, we see that business companies, striving to survive or eyeing a greater share of the market, collaborate through either joint ventures or mergers. Countries form economic unions and military alliances. Zebras and giraffes live together to form better defenses against predators. Insects, such as bees and ants, work in groups to build their colonies. Piranhas hunt together so that they can attack preys with size much larger than an individual piranha. On the other hand, we also notice that at times a human group becomes too dominant and diverse in its operations, opinions, or ideologies that a division, or a population disintegration, results.

When incorporating APE into our application, a training case is a population. We want to merge the weak cases to contend with the strong cases to achieve a clustering in which each cluster of training cases is more or less equally strong. We also impose a constraint, similar to that for COBWEB/3, on the process: only training cases (or populations) with neighboring intensity ranges are allowed to aggregate. In addition, we use only K = 1 dimension, using only the spatial attributes. The strength of each training case is thus the spatial relationship between the case and itself. Note also that we implement the methodology in a sequential fashion. Therefore, to avoid order-dependent aggregation, we examine the training cases in two exactly opposite orders by running our implementation of APE twice.

3.4 POST-LEARNING PROCESSING

After the learning by discovery phase, we perform post-processing to resolve conflicts and to refine clusters.

3.4.1 Conceptual Clustering

After further experiments and evaluations, we have decided to exclude the textural attributes from the learning phase since it causes COBWEB/3 to over-react to the fine details among training cases and eventually to fail to form meaningful clusters. COBWEB/3 is often not able to establish multi-instance clusters when textural attributes are involved—hinting that textural attributes might be too discriminative in the clustering process.

COBWEB/3 does not produce the same concept hierarchy given the same set but differently ordered training cases. Thus, we need to perform a conflict resolution. We first flatten the two resultant concept hierarchies generated by running COBWEB/3 twice. Then we locate and resolve any discrepancies between the two flattened hierarchies using the textural attributes. For example, if *hierarchy1* is TC1-TC2 and TC3-TC4-TC5, and *hierarchy2* is TC1-TC2-TC3 and TC4-TC5, then an inter-cluster difference based on the textural attributes is computed for each pair of clusters of each hierarchy. The pair of clusters with the larger difference wins and retains its status.

3.4.2 APE

Similarly, after running APE twice, we obtain two clusterings and we have to score each clustering to select the better one. Given a clustering, we take the difference in strength between each aggregated population and the strongest aggregated population. We then sum the differences, and select the clustering with the smaller sum as the better clustering. After the selection, we perform three refinement steps: population migration, population solidification, and population disintegration.

Population migration is used to move a training case from one aggregated population to another. A case can only move to another population when it is a neighbor to that population along the intensity axis, and only if the migration improves the overall equilibrium of the population set.

Population solidification is used to group a strong yet under-represented (in terms of the number of pixels in the image) aggregated population with its neighboring population.

Population disintegration is used to split an aggregated population into two if the diversity of the aggregated population is high. The diversity measure is similar to the spatial attribute: the probability of an aggregated population, i, having j as a spatial neighbor k times in a 3x3 window. An aggregated population is diverse if it has high probabilities of frequent contacts (high k values) with other populations.

3.5 IMAGE SEGMENTATION

After the post-learning processing stage, we have a consistent clustering. Suppose that, after histogram extraction, we obtain a set of 7 peaks = {25, 28, 33, 37, 45, 58, 67}. As a result, we have 8 training cases, TC1 to TC8, with TC1's intensity range = (0-25), TC2's = (25-28), …, and TC8's = (67-255). Then we compute for each training case its spatial and textural attributes. Suppose that, after the learning and refinement phases, we obtain the following clustering: TC1-TC2, TC3-TC4-TC5, and TC6-TC7. Hence, the number of clusters is 3. Our system then uses this acquired knowledge to label all image pixels, generalizing the knowledge from the histogram level to the pixel level. First, the system identifies a set of key thresholds. By combining the intensity ranges (according to the clusters), we have (0-28), (33-45), and (58-255). A key threshold is simply the upperbound of an intensity range: 28, 45, and 255. Since there are no pixels with a value greater than 255, we are left with 2 key thresholds: 28 and 45. Then the system labels the image pixels accordingly: pixels with intensity values less than or equal to 28 are labeled *class1*, those with values greater than 28 but less than or equal to 45 are labeled *class2*, and those with values greater than 45 are labeled *class3*.

Note that without the discovery mechanism, we would have identified 7 thresholds and subsequently 8 segmentation classes. By including the discovery approaches, however, the system learns how to cluster peaks based on their intensity ranges and spatial and textural attributes. As a result, we are able to segment the image into contextually meaningful classes.

4 DISCUSSION OF RESULTS

The domain and application of our studies are Synthetic Aperture Radar (SAR) sea ice image segmentation. The images were obtained from satellites ERS-1, ERS-2, and RADARSAT and each consists of water and different ice types. The evaluation was performed on nine images with distinctive characteristics. Table 3 shows the results of the experiments. We observe the following:

● The intensity and spatial attributes are sufficient for identifying different segmentation classes in SAR sea ice imagery.

● The APE-based discovery generates more coherent and meaningful sea ice classes, corresponding to human visual inspection. The COBWEB/3-based discovery, however, generates classes at a higher granularity.

Table 3. Discovery Results: X/Y/Z means the initial number of training cases/the final number of classes/and the visual evaluation score (0-5.0)

Image	COBWEB/3	APE
12146	14/6/3.0	14/4/5.0
14439	11/3/3.0	11/4/4.0
23816	8/2/4.0	8/3/4.5
25028	8/3/2.5	8/3/4.5
32007	19/7/2.0	19/5/4.0
60093	13/5/2.5	13/4/4.5
83282	18/8/2.0	18/6/3.5
85696	12/5/3.0	12/5/4.0
96895	9/3/4.0	9/4/4.0
Average	12.44/5.44/2.89	12.44/4.44/4.22

The table shows that COBWEB/3-based approach in general produces a higher number of classes than the APE-based approach. It also yields a significantly lower average score of visual evaluation. The clustering method of COBWEB/3 identifies conceptually different groups of training cases incrementally. It attempts to trade-off between generality and specificity for classification and prediction purposes. On the other hand, our implementation of the APE concept is an aggressive, spatially-based discovery technique. The decision to merge classes or split a class is not based on achieving balanced generality and specificity within the populations; instead, it is based on achieving a set of aggregated populations with similar strengths. The nature of this aggregation allows the classes to merge more freely, and thus form fewer clusters of training cases.

The visual evaluation is based on subjective inspection of the segmented images from the viewpoint of sea ice image analysis. Images with classes corresponding to sea ice types and regions are scored higher than those without. Over-segmented images are also scored higher than under-segmented images since over-segmented images can always be further refined while merged classes can no longer be split without substantial effort.

In addition to the above experiments, we have also incorporated AutoClass (Cheeseman *et al.,* 1990) and SNOB (Wallace and Dowe, 1994) as two alternative learning-by-discovery methodologies into the image segmentation system. We observed the following:

● The AutoClass-based discovery is less sporadic than COWBEB/3. It is able to cluster training cases without requiring additional emphasis on the intensity value.

● The AutoClass-based discovery is more aggressive than APE in merging. The average number of classes discovered by AutoClass was only 2.56, compared to 4.22 by APE. This is not good for our sea ice applications.

● The textural attributes are more influential in AutoClass than in APE, COBWEB/3 or SNOB, hinting that AutoClass might be more efficient in dealing with higher-resolution attributes.

● The SNOB-based discovery is less sporadic than COBWEB/3 but more sporadic than AutoClass.

● The SNOB-based discovery is also more aggressive than APE in merging. The average number of classes discovered by SNOB was only 2.31. This is not good for our sea ice applications.

From our experiments, we observe that the COBWEB/3-based discovery yields more sporadic clusters than the AutoClass-based and SNOB-based approaches. The COBWEB/3-based discovery also differentiates classes at a higher granularity, as it is the least aggressive among the four discovery techniques. Both the AutoClass-based and SNOB-based techniques suffer from initialization-dependency: the initial guess on the number of classes greatly influences the outcome of the discovery process. That is, given exactly the same set of data, in the same order, AutoClass discovers different clusters when it is run at different times; so does SNOB.

5 ASIS

We have built a fully automated image segmentation software tool called ASIS that implements the APE concept. The objective of this tool is to provide automated segmentation for SAR images for either image preprocessing or classification. ASIS has been tested on ERS and RADARSAT sea ice images, ERS-1 SAR images of mountains, Landsat TM images of urban and rural areas, NOAA AVHRR vegetation index images, and SAR images for roll vortices detection. Note that ASIS utilizes only the intensity and the spatial attributes.

Here we show an example of ASIS applied to a SAR sea ice image. Figure 2 shows an original SAR sea ice image that consists of packed ice (brightest regions) with very dark, cutting linear structures (ice leads) and grayish regions (new ice or open water). In addition, there are brighter, silky structures (possibly deformed first year ice) straining within the grayish regions. So there are essentially four classes in the image. ASIS first extracted a set of 14 peaks = {32, 39, 42, 44, 47, 52, 57, 61, 69, 79, 83, 86, 89, 91}. Figure 3 shows what the segmented image would be if the system were to use the 14 thresholds to obtain 15 segmentation classes, without clustering.

We first identify the 15 training cases, with TC1's intensity range as (0-32), that of TC2 as (32-39), and so on. Second, we describe the cases with their spatial attributes, as shown in Figure 4. Then, the APE module analyzed the training cases and discovered 4 clusters: TC1-TC2-TC3-TC4 (0-44), TC5-TC6-TC7-TC8 (47-61), TC9-TC10-TC11-TC12-TC13-TC14 (69-89), and TC15 (91-255). The key thresholds were 44, 61, and 89. ASIS then used these thresholds to segment the image into four classes, as shown in Figure 5. Pixels with intensity values lower than 44 are grouped into *class1* (black), those between 44 and 61 are grouped into *class2* (dark), those between 61 and 89 are grouped into *class3* (gray), and the rest are grouped into *class4* (white).

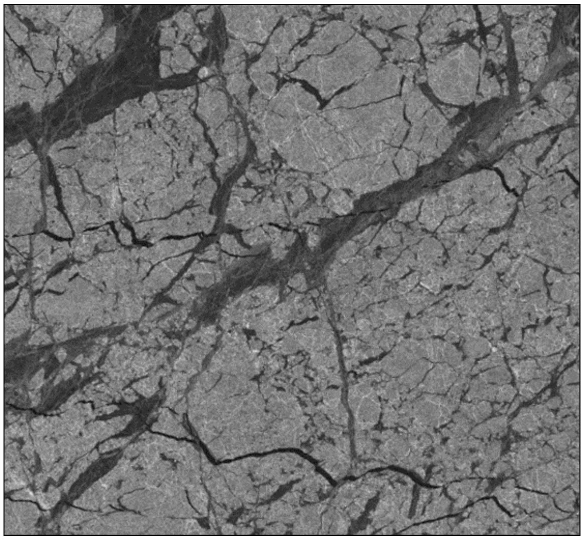

Figure 2. Original ERS-1 SAR sea ice image (portion) (March 27, 1992, 73.46N, 156.19E). © ESA

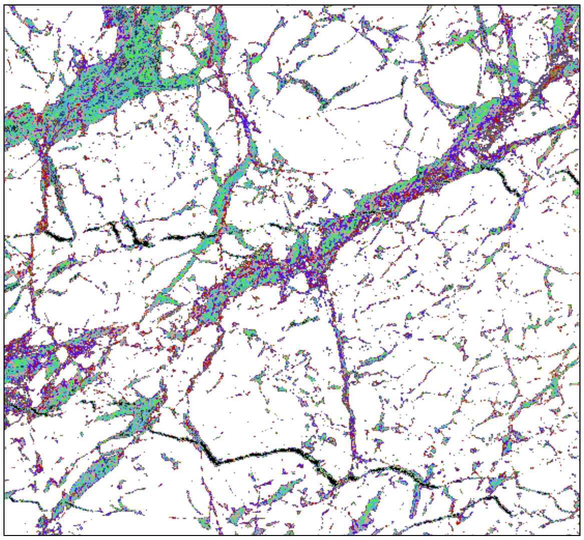

Figure 3. The image segmented with 14 thresholds, without going through learning by discovery.

TC1 = ((0,32) 0.45 0.13 0.06 0.04 0.04 … … 0.03)
TC2 = ((32,39) 0.19 0.13 0.10 0.08 0.08 … … 0.06)
…
TC14 = ((89,91) … … 0.09 0.08 0.07 0.05 0.14 0.48)
TC15 = ((91,255) … … 0.01 0.01 0.01 0.01 0.03 0.93)

Figure 4. The 15 training cases. Each training case has an intensity range and 15 spatial attributes.

6 CONCLUSIONS

We have outlined an unsupervised image segmentation approach based on machine learning by discovery. The approach uses image processing techniques to extract and describe a set of training cases, and then applies discovery mechanisms to group the cases into clusters. Based on the clusters, the approach finally labels all image pixels into meaningful segmentation classes. The emphasis in discovery allows the system to determine the number of classes and what the classes are in the image without any human intervention, which is important in dealing with highly dynamic images. We have described the important modules of our approach: histogram extraction, extraction and description of training cases, learning by discovery, post-learning processing, and image segmentation. We have defined the intensity range, and the spatial and textural attributes for our training cases. We have conducted analyses and comparison studies on two different discovery methodologies: COBWEB/3 and APE. We have observed that the textural attributes are discriminative while the spatial ones tend to help in better cluster formation.

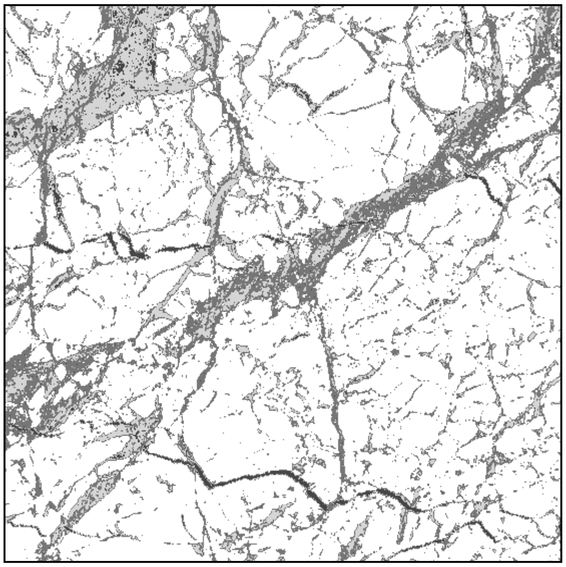

Figure 5. The final segmented image with four discovered classes: black, dark, gray, and white

We have adapted COBWEB/3 and APE for our remote sensing image segmentation. We impose a constraint where the intensity range is given a higher weight in the discovery process than the other attributes. Thus, in a way, the learning by discovery in our research can be viewed as guided by our knowledge of the imagery domain and application. In addition, to lessen order-dependency, we feed the training cases to the discovery mechanism twice, in exactly opposite orders. Then, the system resolves the discrepancies between the two clustering results and refines the final clustering.

We have also experimented with SNOB and AutoClass. Both discovery techniques are aggressive in merging training cases, rendering the segmentation useless in our application. Moreover, both techniques suffer from initialization-dependency that would have to be addressed if we incorporated them into our system.

Finally, we have built a software tool called ASIS based on the APE discovery concept and applied it to various remotely sensed images, especially SAR sea ice images.

Acknowledgements

This work was funded in part by Naval Research laboratory contract N00014-85-C-6038 and by The Research Development Fund of the University of Kansas. We would like to thank the reviewers for their suggestions that improved the paper.

References

Action, S. T. (1996). On Unsupervised Segmentation of Remotely Sensed Imagery Using Nonlinear Regression, *Int. J. Remote Sensing*, **17**(7):1407-1415.

Bezdek, J. C. and M. M. Trivedi (1986). Low Level Segmentation of Aerial Images with Fuzzy Clustering, *IEEE Trans. Systems, Man, and Cybernetics*, **16**:589-598.

Cheeseman, P., J. Kelly, M. Self, J. Stutz, W. Taylor, and D. Freeman (1990). AutoClass: A Bayesian Classification System, in *Readings in Machine Learning*, J. W. Shavlik and T. G. Dietterich (eds.), San Meteo, CA: Morgan Kaufmann, 296-306.

Chou, J., R. C. Weger, J. M. Ligtenberg, K.-S. Kuo, R. M. Welch, and P. Breeden (1994). Segmentation of Polar Scenes Using Multi-Spectral Texture Measures and Morphological Filtering, *Int. J. Remote Sensing*, **15**(5):1019-1036.

Cohen, F. S. and Z. G. Fan (1992). Maximum Likelihood Unsupervised Textured Image Segmentation, *CVGIP: Graphical Models and Image Processing*, **54**(3):239-251.

Gennari, J. H., P. Langley, and D. Fisher (1990). Models of Incremental Concept Formation, in *Machine Learning: Paradigms and Methods*, J. Carbonell (ed.), 11-61, MIT Press/Elsevier.

Goldberg, M. and S. Shlien (1978). A Cluster Scheme of Multispectral Images, *IEEE Trans. Systems, Man, and Cybernetics*, **8**:86-92.

Haralick, R. M., K. Shanmugan, and I. Dinstein (1973). Texture Features for Image Classification, *IEEE Trans. Systems, Man, and Cybernetics*, **3**:510-521.

Haverkamp, D., L.-K. Soh, and C. Tsatsoulis (1995). A Comprehensive, Automated Approach to Determining Sea Ice Thickness from SAR Data, *IEEE Trans. Geoscience and Remote Sensing*, **33**(1):46-57.

Holmes, Q. A., D. R. Nuesch, and R. A. Shuchman (1984). Textural Analysis and Real-Time Classification of Sea Ice Types Using Digital SAR Data, *IEEE Trans. Geoscience and Remote Sensing*, **22**(2):113-120.

Holt, B., R. Kwok, and E. Rignot (1989). Ice Classification Algorithm Development and Verification for the Alaska SAR Facility Using Aircraft Imagery, *Proc. Int. Geoscience and Remote Sensing Symposium*, 751-754.

Huntsberger, T. L., C. L. Jacobs, and R. L. Cannon (1985). Iterative Fuzzy Image Segmentation, *Pattern Recognition*, **18**:131-138.

Kohonrn, T. (1989). *Self-Organization and Associative Memory*, Berlin: Springer-Verlag.

Nguyen, H. H. and P. Cohen (1993). Gibbs Random Fields, Fuzzy Clustering, and the Unsupervised Segmentation of Textured Images, *CVGIP: Graphical Models and Image Processing*, **55**(1):1-19.

Nystuen, J. A. and F. W. Garcia, Jr. (1992). Sea Ice Classification Using SAR Backscatter Statistics, *IEEE Trans. Geoscience and Remote Sensing*, **30**(3):502-509.

O'Gorman, L. (1994). Binarization and Multi-thresholding of Document Images Using Connectivity, *CVGIP: Graphical Models and Image Proecssing*, **56**(6):494-506.

Panjwami, D. K. and G. Healey (1995). Markov Random Field Models for Unsupervised Segmentation of Textured Color Images, *IEEE Trans. Pattern Analysis and Machine Intelligence*, **17**(10):939-954.

Peng, A. and W. Pieczynski (1995). Adaptive Mixture Estimation and Unsupervised Local Bayesian Image Segmentation, *CVGIP: Graphical Models and Image Processing*, **57**(5):389-399.

Rosenfeld, A. and A. C. Kak (1982). *Digital Image Processing*, 2nd Ed., New York: Academic Press.

Sezan, M. I. (1990). A Peak Detection Algorithm and Its Application to Histogram-Based Image Data Reduction, *Computer Vision, Graphics, and Image Processing*, **49**:36-51.

Soh, L.-K. (1998). *Automated Image Segmentation: A Data Investigation Model and SAR Sea Ice Applications*, Ph.D. Dissertation, Department of EECS, University of Kansas, Lawrence, KS, USA.

Soh, L.-K. and C. Tsatsoulis (1999a). Segmentation of Satellite Imagery of Natural Scenes Using Data Mining, *IEEE Trans. Geoscience and Remote Sensing*, **37**(2):1086-1099.

Soh, L.-K. and C. Tsatsoulis (1999b). Unsupervised Segmentation of ERS and RADARSAT Sea Ice Images Using Multiresolution Peak Detection and Aggregated Population Equalization, *Int. J. Remote Sensing*, **20**(15/16):3087-3109

Smith, D. M. (1996). Speckle Reduction and Segmentation of Synthetic Aperture Radar Images, *Int. J. Remote Sensing*, **17**(11):2043-2057.

Thompson, K. and K. McKusick (1993). COBWEB/3: A Portable Implementation, Ames Res. Center*, Tech. Rep. FIA-90-6-18-2*, version 1.4.

Wallace, C. S. and D. L. Dowe (1994). Intrinsic Classification by MML—The Snob Program, in *Proc. 7th Australian Joint Conf. on Artificial Intelligence*, Armidale, New South Wales, Australia, 37-44.