

# Learning Methodologies and Discriminating Visual Cues for Unsupervised Image Segmentation

Leen-Kiat Soh and Costas Tsatsoulis

Information and Telecommunication Technology Center (ITTC)  
University of Kansas, 2291 Irving Hill Road, Lawrence, KS 66044 USA  
{lksoh,tsatsoul}@itcc.ukans.edu

## Abstract

In this paper, we describe our research in unsupervised image segmentation using machine learning techniques. 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 to compute spatial and textural attributes as visual cues. Second, we use learning by discovery methodologies to cluster these cases: COBWEB/3, SNOB, AutoClass, and APE. COBWEB/3 is based on incremental concept formation; AutoClass on Bayesian probabilities; and SNOB on minimum message length. APE is based on a new strategy called Aggregated Population Equalization that attempts to maintain similar strengths for all populations in its environment. Third, we obtain from the clustering results of the methodologies the number of visually significant classes in the image (and what these classes are) and finally segment the image. We conduct visual evaluation of the results to determine the best learning methodology and the set of discriminating visual cues for our remote sensing applications. Based on the findings, we have built an unsupervised image segmentation software tool called ASIS and have applied it to a range of remotely sensed images.

## Introduction

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 in this domain. Image segmentation is a process of pixel classification where the image is segmented into regions by assigning individual pixels into classes. An unsupervised technique implies automated operation independent of human intervention during the execution of the algorithm.

Our approach is to first identify training cases in the image. These cases are significant peaks extracted from an image histogram of regional bisectors. Then, we describe each training case with a set of visual cues that consists of an intensity range, a set of spatial attributes, and a set of textural attributes. Third, we feed the training cases, to-

gether with the visual cues, into a learning by discovery module. The output of the learning process is a clustering that groups the cases into separate clusters. The clustering gives us 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 methodology is thus able to segment an image automatically by first learning the significant clusters in the image and then applying the learned clustering strategy to all pixels in the image.

In our analysis, we experimented with four learning methodologies: COBWEB/3 (Gennari et al. 1990, Thompson and McKusick 1993), AutoClass (Cheeseman et al. 1990), SNOB (Wallace and Dowe 1994), and the Aggregated Population Equalization (APE) strategy (Soh 1998). COBWEB/3 is based on conceptual clustering and incremental learning; AutoClass is based on Bayesian probabilities; SNOB is based on minimum message length; and APE is based on population strengths. We also considered different sets of visual cues: intensity, spatial, and textural. The domain and application of our analysis are segmentation of remotely sensed images, particularly Synthetic Aperture Radar (SAR) sea ice images. We have found out that the APE strategy is the most suitable for our SAR sea ice application and the intensity and spatial attributes are the sufficient visual cues and have built a software tool called ASIS based on those findings.

## 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. Objects in these images also do not have textures that are observably distinctive. 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.

There have been several discovery learning approaches applied to image segmentation. For example, ISODATA

(Holt et al. 1989) and 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; non-linear regression (Acton 1996) is based on the regularization theory; and multi-thresholding (O’Gorman 1994) is based on peaks in the imagery. 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.

## 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 visual cues such as spatial and textural attributes. Fourth, we feed the cases into a learning by discovery module. Given the clustering result, we perform post-processing to resolve conflicts and refine clusters. The final clustering tells us what the number of classes are and what the classes are in the image. We use that *learned*

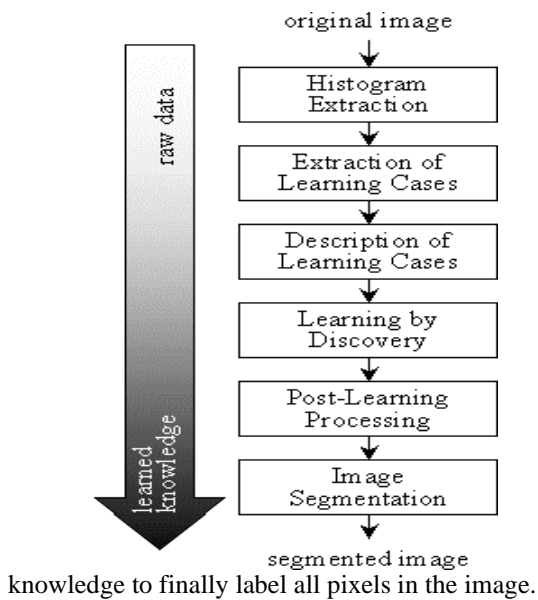


Figure 1. The block diagram of our unsupervised segmentation via learning by discovery.

## 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 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. Soh (1998) gives a detailed treatment of our implementation of dynamic local thresholding.

## 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).

## Visual Cues 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 visual cues (or attributes) to form clusters. We use two sets of visual cues: spatial and textural.

**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  $N \times N$  matrix in which each entry is the frequency of a case being a spatial neighbor to another case, including itself, as shown 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).

	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

Table 1. A spatial matrix in which TC1 neighbors with itself 86.17% of the time, with TC2 10.35%, and so on.

**Texture Attributes.** Textures have often been used to represent and analyze regions in remotely sensed images. 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, 32 x 32), 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  $N \times 8$  textural matrix, as shown in Table 2.

### 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 adapted four different approaches: COBWEB/3, AutoClass, SNOB, and APE.

	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

Table 2. A textural matrix.

**COBWEB/3.** 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—it is order-dependent. 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 resolve conflicts in the hierarchies later.

To increase the role of a case’s intensity range, we have 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, the biased learning technique puts a higher weight on grouping cases with similar intensity ranges together than those with similar spatial or textural makeup.

**AutoClass.** AutoClass suffers 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 its is run at different times. Hence, we run AutoClass  $N$  times and pick the best result. AutoClass does not suffer from order-dependency.

**SNOB.** Similarly, we also run SNOB  $N$  times and pick the best result since the design is initialization-dependent. SNOB does not suffer from order-dependency.

**APE.** 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 into smaller populations if its is overly diverse. 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. These aggregated populations are the clus-

ters. As a result, APE learns the number of clusters and what the clusters are through this form of discovery. See (Soh and Tsatsoulis 1999b) for a detailed treatment of the APE strategy.

When adapting APE to our application, each training case is a population. We want to merge the weak cases to achieve a clustering in which each cluster of training cases is more or less equally strong. We also only allow training cases (or populations) with neighboring intensity ranges to aggregate. In addition, we use only the spatial attributes as the measure of population strength. Note that since we implement the methodology in a sequential fashion, to avoid order-dependency, we examine the training cases in two exactly opposite orders by running our implementation of APE twice.

### Post-Learning Processing

After the learning by discovery phase, we perform post-processing to resolve conflicts and to refine clusters. Note that the post-learning processing on clusters generated by either SNOB or AutoClass is visual selection of the best clustering.

**COBWEB/3.** 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.

To resolve conflicts in the two resultant concept hierarchies, we flatten them, and then resolve any discrepancies between the two flattened hierarchies. 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.

**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 several refinement steps to move a training case from one population to a neighboring one and to split an overly-diverse population. 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  $3 \times 3$  window. An aggregated population is diverse if it has high probabilities of frequent contacts (high  $k$  values) with other populations.

### Image Segmentation

After the post-learning processing stage, we have a consistent clustering. Suppose that, after histogram extraction,

we obtain a set of seven peaks = {25, 28, 33, 37, 45, 58, 67}. As a result, we have eight 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 three. 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 two 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*.

### 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 using APE and COBWEB/3, the two fully automated designs. 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, on the other hand, generates classes at a higher granularity.

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.

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

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)

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.

We also experimented with AutoClass and SNOB and observed the following:

- ˆ 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.
- ˆ 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.
- ˆ 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.
- ˆ The SNOB-based discovery is less sporadic than COBWEB/3 but more sporadic than AutoClass.
- ˆ The SNOB-based discovery is also more aggressive than APE in merging. The average number of classes discovered by SNOB was only 2.31.

In addition, the AutoClass-based approach is highly stable, as it is able to cluster cases with neighboring intensity ranges together without additional constraints. Without the intensity emphasis, COBWEB/3 is the most sporadic learning mechanism. 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- and SNOB-based techniques suffer from initialization-dependency and thus are not suitable for unsupervised image segmentation unless we can find a way of evaluating and selecting the best clustering. Both the COBWEB/3- and APE-based designs suffer from order dependency and thus require post-learning processing to resolve conflicts and refine clusters. In conclusion, the APE-based approach is the most suitable learning mecha-

nism as it is able to generate visually good segmentation for SAR sea ice applications. Also, the intensity and spatial attributes are sufficient visual cues. The textural attributes, however, are highly discriminating visual cues.

## 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 pre-processing or classification. ASIS has been tested on ERS and RADARSAT sea ice images, ERS-1 SAR images of mountains, 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 extracted a set of 14 peaks = {32, 39, 42, 44, 47, 52, 57, 61, 69, 79, 83, 86, 89, 91}. Then, ASIS identified three key thresholds as 44, 61, and 89 and segmented the image into four correct classes, as shown in Figure 3.

## Conclusions

We have described an unsupervised image segmentation approach based on machine learning by discovery and an analysis of that approach using different learning methodologies and discriminating visual cues. The approach uses image processing techniques to extract and describe a set of training cases with visual cues, applies discovery mechanisms to group the cases into clusters, and ultimately segments the image based on the clustering. The utilization of learning by discovery techniques allows the approach 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 also defined the intensity range, and the spatial and textural attributes for our training cases. From our analysis, we concluded that the textural attributes are more influential and discriminative visual cues than the spatial ones, while the spatial ones tend to help in better cluster formation, and, without intensity values, most of the learning mechanisms (COBWEB/3, APE, and SNOB) failed to yield coherent clusters. We also concluded that APE is the most suitable learning methodology, and have built a software tool called ASIS for remote sensing applications, particularly in SAR sea ice imagery.

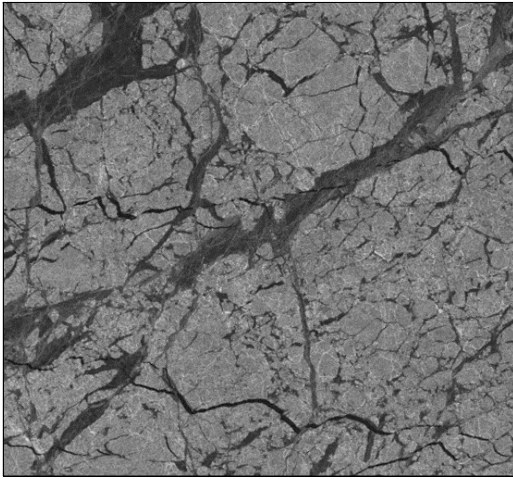


Figure 2. Original ERS-1 SAR sea ice image (portion) (March 27, 1992, 73.46N, 156.19E). © ESA

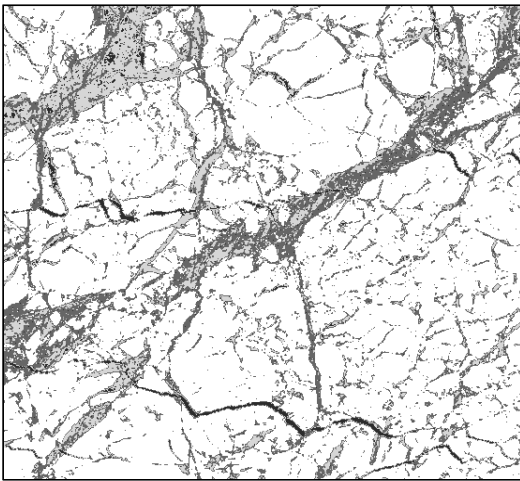


Figure 3. The final segmented image with four discovered classes: black, dark, gray, and white.

### 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.

### References

Acton, S. T. 1996. On Unsupervised Segmentation of Remotely Sensed Imagery Using Nonlinear Regression. *International Journal of Remote Sensing*, 17:1407-1415.

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

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

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

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

Holt, B.; Kwok, R.; and Rignot, E. 1989. Ice Classification Algorithm Development and Verification for the Alaska SAR Facility Using Aircraft Imagery. In Proceedings of the International Geoscience and Remote Sensing Symposium, 751-754.

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

O’Gorman, L. 1994. Binarization and Multi-Thresholding of Document Images Using Connectivity. *Graphical Models and Image Processing*, 56:494-506.

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. diss., Dept. of EECS, University of Kansas.

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

Soh, L.-K.; and Tsatsoulis, C. 1999b. Unsupervised Segmentation of ERS and RADARSAT Sea Ice Images Using Multiresolution Peak Detection and Aggregated Population Equalization. *International Journal of Remote Sensing*, 20:3087-3109

Smith, D. M. 1996. Speckle Reduction and Segmentation of Synthetic Aperture Radar Images. *International Journal of Remote Sensing*, 17:2043-2057.

Thompson, K.; and McKusick, K. 1993. COBWEB/3: A Portable Implementation, Technical Report FIA-90-6-18-2, Ames Research Center.

Wallace, C. S.; and Dowe, D. L. 1994. Intrinsic Classification by MML—the Snob Program. In Proceedings of the Seventh Australian Joint Conference on Artificial Intelligence, 37-44, Armidale, New South Wales, Australia.