

# Lossless Compression of Classification Map Images

H. Xie<sup>1</sup> and M. Klimesh<sup>1</sup>

*A classification map is an image that provides a low-level content description of a corresponding remote-sensing image. Each “pixel” in the map is the index of a class that represents some type of content, such as a type of vegetation or mineral, in the corresponding spatial location of the remote-sensing image. Classification maps generated onboard a spacecraft may be used as part of a region-of-interest image data compression scheme, and/or to provide low-data-volume summaries of the remote-sensing images. In either case, it is desirable to losslessly compress classification maps prior to transmission. In this article, we describe a technique for lossless compression of classification maps. Compression tests on sample classification maps indicate that our technique yields considerable improvement, e.g., a 15 to 40 percent bit-rate reduction, as compared to existing general-purpose lossless image compression methods.*

## I. Introduction

A classification map is an image that provides a low-level content description of a corresponding remote-sensing image. This description is in terms of a (typically small) number of classes, each of which is designed to represent some type of image content, such as a type of vegetation or mineral. Each “pixel” value in the map is the index of the class that (ideally) best describes the scene content in the corresponding spatial location in the remote-sensing image. A common method of producing a classification map is to use a support vector machine (SVM) classifier [3,8]; Fig. 1 shows an example of a classification map produced by applying an SVM classifier to an Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) scene.<sup>2</sup>

Two ways in which a classification map generated onboard a spacecraft may be used are (1) to provide a summary of scene content, perhaps in conjunction with transmission of a low-fidelity version of the scene itself or even without transmission of any further scene information, and (2) as part of a region-of-interest (ROI) image data compression scheme [2,4]. In the first case the classification map must be transmitted to the ground, and in the second case either the classification map or a derived priority map must be transmitted. Note that a priority map is itself a type of classification map where the classes indicate the relative importance value.

---

<sup>1</sup> Communications Architectures and Research Section.

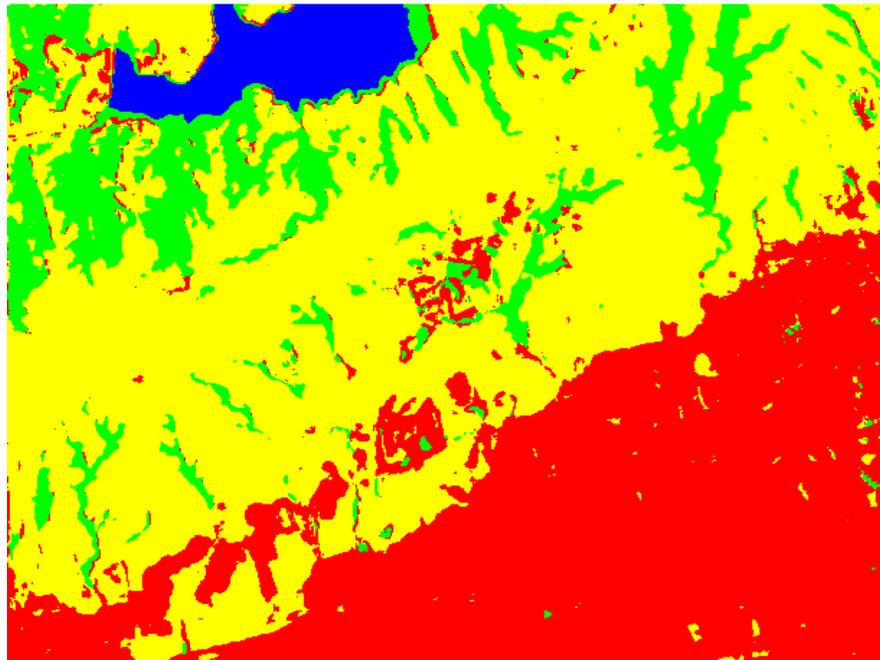
<sup>2</sup> AVIRIS data sets are available from the AVIRIS web site: <http://aviris.jpl.nasa.gov/html/aviris.freedata.html>.

The research described in this publication was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

(a)



(b)



■ WATER    ■ LAND    ■ FOREST    ■ HUMAN HABITAT

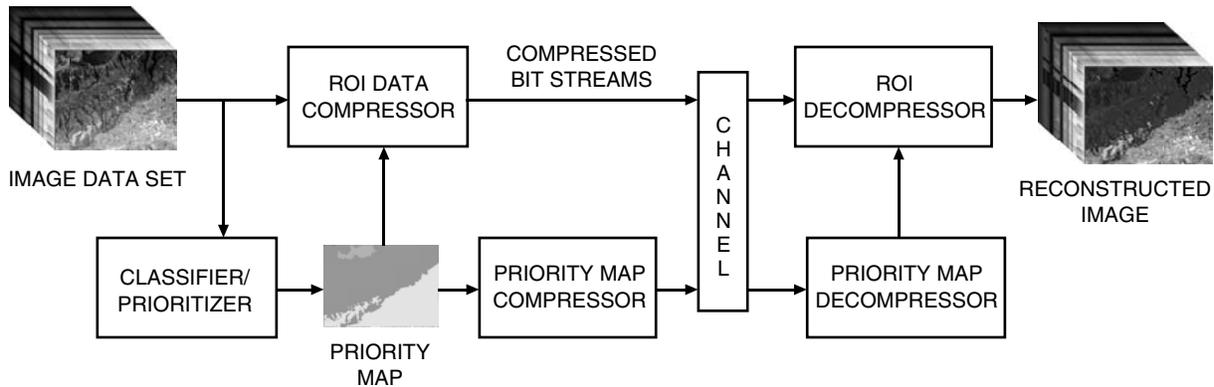
Fig. 1. AVIRIS Moffett Field Scene 1: (a) false color and (b) the classification map produced by an SVM pixel classifier. The scene is classified into four classes, roughly corresponding to water, land, forest, and human habitat.

Effective lossless compression of classification maps prior to transmission is important for reducing the transmission cost. In this article, we describe a technique for lossless compression of classification maps. We also present results of compression tests on some classification maps; these tests show that our technique yields considerable improvement over existing general-purpose lossless image compression methods.

In the case of ROI compression, each class in the classification map is associated with a priority level; thus, together the classification map and the associated priorities define a data prioritization map (or priority map), with higher priority levels assigned to classification regions that are more important to the scientist users. ROI compression [2,4] allows more compressed bits to be allocated to identified regions of scientific importance, resulting in higher fidelity reconstruction of such regions by the decompressor at the expense of some fidelity in regions indicated to be less important. ROI compression is thus potentially able to increase the net science value of the data returned from a spacecraft with limited downlink capacity. Figure 2 shows a block diagram of a region-of-interest image compression system. Both the compressor and decompressor require the same priority map as an input.

Each pixel value in a classification map represents a certain scientifically defined class. There are often large contiguous regions of the same class; thus, if a pixel's neighbors all belong to the same class, then it is highly likely that the pixel itself will belong to that class. In contrast to natural continuous-tone images, there are typically a relatively small number of different pixel values in a classification map. As an extreme example, if we are interested in finding rocks on Mars, then the classification map might consist of only two values, "rock" and "non-rock." Also, unlike natural images, because the pixel values in a classification map are indices, numerically close pixel values usually do not necessarily represent similar content. These properties make the classification map compression problem somewhat different from compression of more conventional images.

A common lossless compression method is predictive compression. In predictive compression, samples (e.g., pixels) are encoded sequentially based on a probability distribution that is estimated from previously encoded samples. When this method is applied to continuous-tone images, typically an estimate is calculated from previously encoded pixels, and the difference between the estimate and the actual pixel value is encoded in the compressed bitstream. In linear prediction, the estimate is a linear combination of nearby pixel values. Context modeling is a technique that is often used in conjunction with predictive compression. In context modeling, samples to be encoded are classified into one of several *contexts* based on previously encoded samples. The context modeler maintains separate statistics for each context and uses these statistics to encode the samples more effectively. Ideally, contexts are defined so that different contexts include pixels with substantially different statistics.



**Fig. 2. Block diagram of a region-of-interest image compression system. The image data are first classified using an onboard classifier/prioritizer. The ROI compressor then employs the priority map as side information to allocate bits accordingly to regions of different priorities. The classification map or the priority map needs to be losslessly transmitted to the receiver for proper reconstruction of the image.**

Many state-of-the-art lossless image compression schemes, such as JPEG-LS (Joint Photographic Experts Group-lossless image compression standard) [10] and CALIC (Context-based Adaptive Lossless Image Compression) [12], use predictive compression with predictors derived from linear predictors, along with context modeling. These techniques are appropriate for natural continuous-tone images. Context modeling and general predictive compression would also be expected to be useful for compression of classification maps. However, linear prediction or, more generally, encoding of the differences between an estimate and a pixel value, is not appropriate for classification map compression because these techniques rely on continuity of the meaning of pixel values, which is typically not present in classification maps.

For our purposes, context modeling can be described somewhat formally as follows. For each pixel, the compressor determines an estimate of the probability distribution of values for the pixel, conditioned on a neighborhood of previously encoded pixels. A context is a function of previously encoded pixel values in the neighborhood. The conditional probabilities  $\Pr(x|c)$ , where  $x$  is a possible pixel value and  $c$  is a context, are estimated and updated empirically from the previously encoded data.

If each distinct combination of pixel values in the context neighborhood is considered to be a distinct context, then the number of contexts grows exponentially with the number of pixels in the context neighborhood. For natural continuous-tone images, which typically have an alphabet size of 256 or larger, the contexts need to be defined appropriately (e.g., by quantization of pixel values in the context neighborhood) in order to avoid the “context dilution” problem, in which there are so many contexts that there are not enough samples to accurately estimate the conditional probabilities associated with the individual contexts. Classification map images typically contain a small number of values, so the potential context dilution issues are not as serious. However, the alphabet in a classification map image is still usually too large to use the context modeling approach of the Joint Bi-level Image Experts Group (JBIG) image compression standard [1], which defines contexts based on all possible pixel values in a reasonably sized neighborhood of the pixel to be encoded.

The Graphical Interchange Format (GIF) and Portable Network Graphics (PNG) format are commonly used file formats for image compression that might be expected to be reasonably appropriate for non-continuous-tone images such as classification maps. GIF uses Lempel–Ziv–Welch (LZW) compression [11], and PNG uses a combination of Lempel–Ziv–77 (LZ77) [13] and Huffman coding. LZ77 and the LZW algorithm are dictionary-based coding techniques that were designed to effectively encode sources consisting of recurring patterns (e.g., English text or recurring sequences of pixel values as might appear in computer-generated artwork). GIF treats image data as a one-dimensional sequence, while PNG does make some use of two-dimensional correlation with a prediction step.

In this article, we propose a classification map compression method that is a simple adaptive context modeler that feeds into a binary interleaved entropy coder [5,7]. A sequence of binary decision bits is produced for each pixel to indicate which, if any, neighboring pixel it matches. The encoder maintains the probability-of-zero estimates for these bits for each of the contexts. The interleaved entropy coder [7] is bit-wise adaptable, which allows the context modeler to quickly adapt to changing statistics in the image. Our results show that our proposed technique achieves about a 15 to 40 percent bit-rate reduction as compared to the JPEG-LS, GIF, and LZ77 compressors.

## II. The Algorithm

We compress the classification map pixels in raster scan order. The details of the pixel encoding are described in Section II.A. When producing the decision bits for a pixel, the context for each such decision bit is determined from information contained in neighboring pixels that have already been encoded. In Sections II.B.1 and II.B.2, we describe two different possible ways of defining the contexts; these yield two different versions of the algorithm. For all the pixels on the image boundary, i.e., the first row, the first column, or the last column, there are fewer adjacent pixels. One could use separate contexts for these pixels, but for simplicity, and because there are a small number of boundary pixels in the images, we simply define all pixels outside the image boundaries to have the value 0.

## A. Piecewise Pixel Prediction

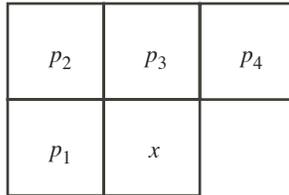
An important characteristic of classification map images is that they tend to contain large contiguous regions of pixels of the same class. Therefore, it is highly likely that  $x$  is the same as one (or more) of its immediate neighboring pixels,  $p_1$ ,  $p_2$ ,  $p_3$ , and  $p_4$ , as indicated in Fig. 3. Instead of directly encoding the pixel value  $x$ , we decompose the information into a sequence of binary decisions as follows:

- (1) Is  $x$  equal to  $p_1$ ?
- (2) If not, is  $x$  equal to  $p_3$ ?
- (3) If not, is  $x$  equal to  $p_2$ ?
- (4) If not, is  $x$  equal to  $p_4$ ?

If  $x = p_k$ , we represent an answer with a “0”; otherwise, it is represented with a “1.” When an affirmative answer is reached, we do not need to continue to the next question. Note that if the neighboring pixel values are not all different, then a question may be redundant, in which case it is skipped. For example, if  $p_1 = p_2 = p_3$  and  $x = p_4$  and  $p_3 \neq p_4$ , then the bit sequence for  $x$  is “10,” corresponding to “ $x$  does not equal  $p_1$ ” and “ $x$  equals  $p_4$ .” If  $x$  is different from all of its neighboring pixels, then the value of  $x$  is included in the compressed bitstream uncoded.

The length of the decision bit string varies from pixel to pixel. Table 1 shows some statistics for the number of decisions that need to be made based on four test images. The column labeled “One decision” lists the percentage of cases when the value of  $x$  is decided by one decision bit; the column labeled “Two decisions” lists the percentage of cases when the value of  $x$  is decided by two decision bits; and so on.

We can see that most of the time the value of  $x$  can be determined from one or two decision bits. The sequence of decision bits is encoded using context modeling and entropy coding, which are described in the following sections.



**Fig. 3. The neighboring pixels that are used to derive the sequence of binary decision bits for current pixel  $x$ . Questions are posed as to whether  $x$  is equal to its immediate neighboring pixels in the order  $p_1, p_3, p_2, p_4$ .**

**Table 1. Statistics for the number of decision bits needed for encoding a pixel. The statistics are shown for four of the test images described in Section III.**

Image	One decision, %	Two decisions, %	Three decisions, %	Four decisions, %	No match, %
Map 1	94.68	4.29	0.21	0	0.82
Map 2	91.96	6.44	0.31	0.00	1.29
Map 3	87.95	8.51	0.98	0.01	2.55
Map 4	78.98	12.69	2.57	0.14	5.62

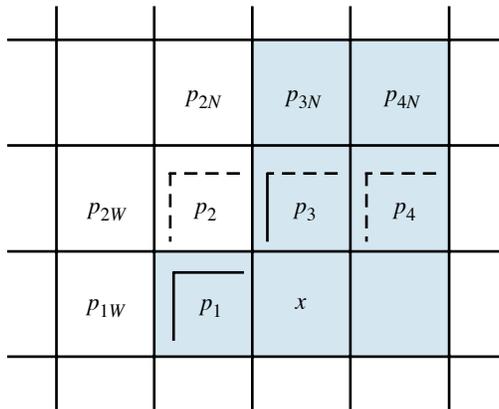
## B. Context Modeling

**1. Edge-Based Context Model.** We call our first context model the edge-based context model. In this context model, edge information in adjacent pixel locations is exploited. For each of the four adjacent previously encoded pixels,  $p_1$  through  $p_4$ , the context index includes two bits indicating the presence or absence of region boundaries in the north and west directions. Figure 4 provides an example.

In this example, the presence of a western edge (represented as a solid vertical line) in pixel location  $p_3$  means that  $p_3$  is in a different classification region from the neighboring pixel  $p_2$  immediately to its left; and the absence of a northern edge (represented as a dashed horizontal line) means that  $p_3$  is in the same classification region as the neighboring pixel  $p_{3N}$  immediately above it.

The number of parameters for this model is 8 (2 edges for each pixel location, and 4 pixel locations are used); therefore, this results in a total of 256 contexts for each type of decision bit. Not all combinations of contexts and decisions are needed due to constraints imposed by the pattern of edges. For example, if no edges are present in the template, which means that all the neighboring pixels are in the same classification region, then only one decision bit (whether  $x$  is equal to  $p_1$ ) needs to be encoded for pixel  $x$ .

**2. Pattern-Based Context Model.** We call our second context model the pattern-based context model. In this context model, the pattern formed by the four neighboring pixels (see Fig. 3) is determined and used as the basic context for coding of the current pixel. The pattern is a set of labels that indicate which of the adjacent pixels are equal to each other. Specifically, we always label the first pixel in the template,  $p_1$  in this case, by the letter A;  $p_2$  is labeled as either A, when  $p_2 = p_1$ , or B, when  $p_2 \neq p_1$ ; and so on. Each pattern is represented by a string of letters with each of them identifying the label of the corresponding pixel. For example, pattern “AABB” represents the case when  $p_1$  and  $p_2$  belong to the same classification region, while  $p_3$  and  $p_4$  are identical to each other but different from  $p_1$  and  $p_2$ . In total there are 15 possible patterns for this four-pixel template. We list in Table 2 all the possible patterns and the set of binary decisions for each type of context. The binary decisions listed in Table 2 are simply a customized version of the decision sequences described in Section II.A, taking into account the constraints imposed by each type of context pattern. From Table 2, we can see that the number of contexts for each of the four types of decision bits is 15, 6, 10, and 6, respectively. Therefore, 37 contexts are used for this context model.



**Fig. 4. An example of the edge-based context determined by the edge information in four adjacent pixel locations. In this example, each pixel location is filled with one of two colors (white or blue) representing its classification label.**

**Table 2. Pattern-based contexts for the four-pixel template and the binary decisions that need to be sent for each type of context. The pattern is formed in the order  $p_1$ ,  $p_2$ ,  $p_3$ , and  $p_4$ . For example, the pattern “AAAB” occurs when  $p_1$ ,  $p_2$ , and  $p_3$  all belong to the same classification region, while  $p_4$  belongs to a different one.**

Context pattern	Binary decisions for each context
AAAA	Does $x$ equal $p_1$ ?
AABA, AABB, ABBA, ABBB	Does $x$ equal $p_1$ ? If not, does $x$ equal $p_3$ ?
ABAA, ABAB	Does $x$ equal $p_1$ ? If not, does $x$ equal $p_2$ ?
AAAB	Does $x$ equal $p_1$ ? If not, does $x$ equal $p_4$ ?
AABC, ABBC, ABCB	Does $x$ equal $p_1$ ? If not, does $x$ equal $p_3$ ? If not, does $x$ equal $p_4$ ?
ABAC	Does $x$ equal $p_1$ ? If not, does $x$ equal $p_2$ ? If not, does $x$ equal $p_4$ ?
ABCA, ABCC	Does $x$ equal $p_1$ ? If not, does $x$ equal $p_3$ ? If not, does $x$ equal $p_2$ ?
ABCD	Does $x$ equal $p_1$ ? If not, does $x$ equal $p_3$ ? If not, does $x$ equal $p_2$ ? If not, does $x$ equal $p_4$ ?

### C. Interleaved Entropy Coder

We employ the non-recursive binary interleaved entropy coder [7], developed by Kiely and Klimesh and used in the ICER imager compressor [6], to encode the sequence of binary decisions. An interleaved entropy coder compresses a binary source by interleaving the output of several different variable-to-variable-length codes that each encode groups of bits with similar probability estimates. Design variations in the choice and number of component codes yield different coding complexities and compression efficiencies. The particular interleaved entropy coder design used in ICER has 17 component codes, details of which can be found in [6].

As in ICER, the context modeler maintains the nominal counts of the number of zero bits and the total number of bits that occur in each context. The probability of zero is estimated by the ratio of these counts.

## III. Experimental Results

As an experimental evaluation of the proposed scheme, we compare the compression performance of the proposed method with that of commonly used lossless compression methods for graphical imagery. Five classification map images were used. These images were produced by applying either the SVM-based pixel classifier [3,8] or a spectra-clustering-based classifier [9] to AVIRIS hyperspectral images. AVIRIS images include 224 spectral bands covering wavelengths from 370 nm to 2500 nm. In our experiments, we use three scenes from the calibrated 1997 Moffett Field radiance data set, each scene having dimensions of

614 pixels by 512 lines. Among the test images, Maps 1 through 4 are generated using the SVM classifier, and Map 5 is generated using the spectra-clustering-based classifier.

In the SVM classifier, we used feature vectors of size 288, consisting of the sample values in a  $3 \times 3$  region surrounding each pixel in 32 bands. To train the SVM classifier, we hand-labeled a small set of pixels belonging to each class. In the spectra-clustering-based classifier, we used feature vectors of size 224, consisting of the sample values of each pixel in all 224 bands. In general, the granularity of the classification map, i.e., the number of distinct classes identified, depends on the content of the scene and on choices reflecting the scientific interest of the user. The number of classes in our maps ranges from 4 to 32.

Table 3 shows the compressed bit rates (in bits per pixel) for each of the test images and the different compression methods. Map 1 (as shown in Fig. 1) is the classification map of the first 512-line scene of the Moffett field dataset. Map 2 is generated from the third 512-line scene of the dataset and has 7 different classes. Maps 3, 4, and 5 are produced from the second 512-line scene of the Moffett field dataset with increasing classification granularities (9, 17, and 32 classes for Maps 3, 4, and 5, respectively). The results for GIFs are produced using ImageJ software.<sup>3</sup> LZ77 [13] is a dictionary-based compression method that is evaluated using the standard UNIX gzip command. Results for JPEG-LS [10] are produced using the Jasper software.<sup>4</sup> The ICER results are for lossless compression with the ICER wavelet-based image compressor [6]. The column labeled “Edge-based” lists the results of the edge-based context modeling version of our method as described in Section II.B.1, and the “Pattern-based” column contains results for the pattern-based context modeling version described in Section II.B.2. For all the test images, both versions of our proposed method achieve better compression performance than all other methods evaluated. Compared to JPEG-LS, which is a state-of-the-art lossless image compressor, both proposed versions achieved more than a 30 percent bit-rate reduction. The two proposed versions of our method achieved comparable results, with the pattern-based version performing slightly better than the edge-based version.

The choice of index assignment for different classification regions will affect the compression performance of compression methods designed for continuous-tone images. In our experiments, we did not attempt to assign indexes to the regions in any particular order. To show that our region index assignments are not abnormally unsuitable for JPEG-LS or ICER, we measured the range of compressed bit rates for all the compression methods on 10 test images derived from Map 5 by randomly rearranging the

**Table 3. Comparison of different compression methods based on five test classification map images. Compressed bit rates (bits per pixel) are shown for each of the test images. For each image, the best result is shown in boldface.**

Image	No. of classes	GIF	LZ77	JPEG-LS	ICER	Edge-based	Pattern-based
Map 1	4	0.4517	0.4639	0.3546	1.9888	0.1809	<b>0.1764</b>
Map 2	7	0.6640	0.6159	0.5873	2.5625	0.2826	<b>0.2793</b>
Map 3	9	0.9637	0.9311	0.9104	3.5398	<b>0.4859</b>	0.4869
Map 4	17	1.6137	1.4616	1.6875	4.6313	0.9242	<b>0.9146</b>
Map 5	32	3.2197	2.8738	3.7374	6.1112	2.4281	<b>2.4147</b>

<sup>3</sup> Available online from the project home page: <http://rsb.info.nih.gov/ij/>.

<sup>4</sup> Available online from <http://www.ece.uvic.ca/~mdadams/jasper/>.

region index assignments. The results are given in Table 4. It can be seen that the range of variation is quite small. Of course, there may well be region index assignments that yield significant improvements for JPEG-LS and ICER, but searching for such assignments is beyond the scope of this work. Note that the tiny variation in the results from our proposed method arises from the way boundary pixels are handled. Also, we omitted the results for the GIF and LZ77 compression methods since region index rearranging does not affect their compressed bit rates.

Table 5 shows compression times of the proposed method. All test images are  $614 \times 512$ ; therefore, they contain 314368 pixels. The implementation was not optimized for benchmarking, so some improvement should be achievable by code optimization. In any case, we expect that the compression times will typically be small compared to the time needed to generate the classification map, and so they are not likely to be an issue.

**Table 4. Average, minimum, and maximum compressed bit rates for JPEG-LS, ICER, and the proposed method. The results are based on 10 test images that were derived from Map 5 by randomly rearranging the region index assignments.**

Compression	JPEG-LS	ICER	Pattern-based
Average bit rate	3.7576	6.1386	2.4173
Minimum bit rate	3.6510	6.0965	2.4127
Maximum bit rate	3.8865	6.2011	2.4192

**Table 5. Compression times (in seconds) using a research-grade implementation of the proposed context modelers and ICER’s implementation of the interleaved entropy coder. Times were obtained on an Intel Pentium-4 3.0-GHz processor running Linux.**

Image	No. of classes	Edge-based	Pattern-based
Map 1	4	0.05	0.07
Map 2	7	0.06	0.07
Map 3	9	0.07	0.08
Map 4	17	0.10	0.10
Map 5	32	0.16	0.17

## Acknowledgments

The SVM-based hyperspectral image classifier and the spectra-clustering-based classifier were provided by D. Mazzoni, K. Wagstaff, and R. Castano.

## References

- [1] “Progressive Bi-Level Image Compression,” International Standard ISO/IEC 11544, ITU-T Recommendation T.82, International Telecommunications Union, 1993.
- [2] E. Atsumi and N. Farvardin, “Lossy/Lossless Region-of-Interest Image Coding Based on Set Partitioning in Hierarchical Trees,” *Proc. 1998 International Conference on Image Processing (ICIP 98)*, vol. 1, Chicago, Illinois, pp. 87–91, October 4–7, 1998.
- [3] R. Castano, D. Mazzoni, N. Tang, T. Doggett, S. Chien, R. Greeley, B. Cichy, and A. Davies, “Learning Classifiers for Science Event Detection in Remote Sensing Imagery,” *iSAIRAS*, September 2005.
- [4] S. Dolinar, A. Kiely, M. Klimesh, R. Manduchi, A. Ortega, S. Lee, P. Sagetong, H. Xie, G. Chinn, J. Harel, S. Shambayati, and M. Vida, “Region of Interest Data Compression with Prioritized Buffer Management (III),” *Proceedings of NASA Earth Science Technology Conference*, College Park, Maryland, June 2003.
- [5] P. G. Howard, “Interleaving Entropy Codes,” *Proc. Compression and Complexity of Sequences 1997*, Salerno, Italy, pp. 45–55, June 11–13, 1998.
- [6] A. Kiely and M. Klimesh, “The ICER Progressive Wavelet Image Compressor,” *The Interplanetary Network Progress Report 42-155, July–September 2003*, Jet Propulsion Laboratory, Pasadena, California, pp. 1-46, November 15, 2003. [http://ipnpr/progress\\_report/42-155/155J.pdf](http://ipnpr/progress_report/42-155/155J.pdf)
- [7] A. B. Kiely and M. Klimesh, “A New Entropy Coding Technique for Data Compression,” *The Interplanetary Network Progress Report 42-146, April–June 2001*, Jet Propulsion Laboratory, Pasadena, California, pp. 1-46, August 15, 2001. [http://ipnpr/progress\\_report/42-146/146G.pdf](http://ipnpr/progress_report/42-146/146G.pdf)
- [8] D. Mazzoni, K. Wagstaff, and R. Castano, “Using Trained Pixel Classifiers to Select Images of Interest,” *The Interplanetary Network Progress Report*, vol. 42-158, Jet Propulsion Laboratory, Pasadena, California, pp. 1–8, August 15, 2004. [http://ipnpr/progress\\_report/42-158/158G.pdf](http://ipnpr/progress_report/42-158/158G.pdf)
- [9] K. L. Wagstaff, H. P. Shu, D. Mazzoni, and R. Castano, “Semi-Supervised Data Summarization: Using Spectral Libraries to Improve Hyperspectral Clustering,” *The Interplanetary Network Progress Report*, vol. 42-163, Jet Propulsion Laboratory, Pasadena, California, pp. 1–14, November 15, 2005. [http://ipnpr/progress\\_report/42-163/163J.pdf](http://ipnpr/progress_report/42-163/163J.pdf)
- [10] M. Weinberger, G. Seroussi, G. Sapiro, and M. W. Marcellin, “The LOCO-I Lossless Image Compression Algorithm: Principles and Standardization into JPEG-LS,” *IEEE Transactions on Image Processing*, vol. 9, pp. 1309–1324, 2000.
- [11] T. A. Welch, “A Technique for High-Performance Data Compression,” *Computer*, vol. 17, pp. 8–19, June 1984.
- [12] X. Wu and N. D. Memon, “Context-Based, Adaptive, Lossless Image Compression,” *IEEE Transactions on Communications*, vol. 45, pp. 437–444, April 2000.
- [13] J. Ziv and A. Lempel, “A Universal Algorithm for Sequential Data Compression,” *IEEE Transactions on Information Theory*, vol. 23, no. 3, pp. 337–343, 1977.