HYPERSPECTRAL IMAGE CLASSIFICATION USING A SELF-ORGANIZING MAP

P. Martínez,[1] J.A. Gualtieri,[2] P.L. Aguilar,[1] R.M. Pérez,[1] M. Linaje,[1] J.C. Preciado,[1] A. Plaza[1]

1. INTRODUCTION

The use of hyperspectral data to determine the abundance of constituents in a certain portion of the Earth's surface relies on the capability of imaging spectrometers to provide a large amount of information at each pixel of a certain scene. Today, hyperspectral imaging sensors are capable of generating unprecedented volumes of radiometric data. The Airborne Visible/Infrared Imaging Spectrometer (AVIRIS), for example, routinely produces image cubes with 224 spectral bands (Green, 1988-2000). This undoubtedly opens a wide range of new possibilities, but the analysis of such a massive amount of information is not an easy task. In fact, most of the existing algorithms devoted to analyzing multispectral images are not applicable in the hyperspectral domain, because of the size and high dimensionality of the images.

Hyperspectral unmixing or linear pixel unmixing is becoming increasingly popular for the analysis and interpretation of hyperspectral images (Martínez et al., 1999). The basic assumption is that the signal received from each pixel can be considered as a simple linear combination of the spectral contributions of all pixel components. The technique, therefore, decomposes the scene in such a way as to recover the fractional contributions of the fundamental components or "endmembers" (as abundance or fraction images). This provides a means of extracting sub-pixel information from the scenes, which is particularly advantageous when the size of the interesting ground elements is much smaller than the image resolution and there is dominance of "mixed" pixels.

Unsupervised clustering is a challenging problem in many areas of data analysis (Antonille & Gualtieri, 2000). It can be stated as follows: given a set of *N* data points in a feature space of *D* dimensions,

$$\{x_1, x_2, ..., x_N\} \in R^D, x_i = (x_{i1}, x_{i2}, ..., x_{iD}), i = 1, ..., N, \quad (1)$$

we wish to characterize K clusters for the data, where K is obtained from statistical information about the data using some distance metric,

$$d_{ij} = d(x_i, x_j). \quad (2)$$

The principal characteristic of unsupervised clustering is that it does not incorporate any previous knowledge about the data. Since ground truth data in remote sensing is expensive and hard to obtain, the use of unsupervised procedures has become more relevant in this field during recent years. In particular, several unsupervised procedures to process hyperspectral data are available in well-known commercial software systems as Research Systems ENVI^{TM}.

The application of neural networks to perform unsupervised classification of hyperspectral data has been tested by several authors (Jiménez et al., 1999) and also by us in some previous work (Aguilar et al., 1998; Martínez et al., 1999; Aguilar et al., 2000a, Aguilar et al., 2000b). We have also focused on analyzing the intrinsic capability of neural networks to parallelize the whole hyperspectral unmixing process (Pérez et al., 1999). The results shown in this work indicate that neural network models are able to find clusters of closely related hyperspectral signatures, and thus can be used as a powerful tool to achieve the desired classification.

[1] Departamento de Informática, Universidad de Extremadura, Avda. de la Universidad s/n, 10071 Cáceres, SPAIN
E-mail: pablomar@unex.es

[2] Global Science and Technology at Applied Information Sciences Branch, NASA/GSFC, Greenbelt, Maryland 20771, U.S.A. E-mail: gualt@peep.gsfc.nasa.gov

One of the most widely used unsupervised neural network algorithms is the Self-Organizing Map (SOM), proposed by Kohonen, 1995-1997. This approach has been successfully applied in many different fields, including image analysis and computer vision, handwritten text recognition, analysis and recognition of human voice and telecommunications. Some reasons for using SOM to perform hyperspectral analysis have been described by Bruske and Merenyi, 1999, who highlight the computational speed provided by this method when implemented by hardware in the form of a massively parallel algorithm, surpassing the performance of conventional classification algorithms. Since the Kohonen algorithm is simple and intuitive, highly parallelizable (which can lead to an easy VLSI implementation based on systolic arrays or FPGAs) and is easily extendable to a high number of dimensions, we have selected it as our starting point to deal with hyperspectral data.

The present work discusses the possibility of using a Self Organizing neural network to perform unsupervised classification of hyperspectral images. In sections 3 and 4, the topology of the proposed neural network and the training algorithm are respectively described. Section 5 provides the results we have obtained after applying the proposed methodology to real hyperspectral data, described in section 2. Different parameters in the learning stage have been modified in order to obtain a detailed description of their influence on the final results. Finally, in section 6 we provide the conclusions at which we have arrived.

2. DATA

The hyperspectral unmixing algorithms proposed in this work have been tested using the public domain Indian Pines hyperspectral dataset, which has been previously used in many different studies. This image was obtained from the AVIRIS imaging spectrometer at Northern Indiana on June 12, 1992 from a NASA ER2 flight at high altitude with ground pixel resolution of 17 meters. The dataset comprises 145x145 pixels and 220 bands of sensor radiance without atmospheric correction. It contains two thirds of agriculture (some of the crops are in early stages of growth with low coverage), and one third of forest, two highways, a rail lane and some houses. Ground truth determines sixteen different classes (not mutually exclusive). Water absorption bands (104-108, 150-163 and 220) were removed (Tadjudin and Landgrebe, 1998), obtaining a 200 band spectrum at each pixel. In order to reduce the time of training and testing, we have selected a subscene of the complete Indian Pines dataset (depicted in Figure 1) of size 68 samples x 86 lines at [27-94] x [31-116] in the original image, considering left in the full scene is at (1,1). In the selected subscene there are four known ground truth classes.

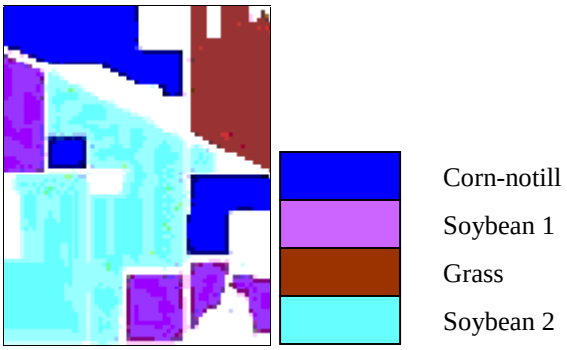


Figure 1. A subset of the Indian Pines hyperspectral dataset with ground truth.

3. TOPOLOGY OF THE PROPOSED NEURAL NETWORK

The Self-Organizing Map (SOM) is based on competitive learning that leads to the construction of topologic maps representing class prototypes. In order to understand the topology of the proposed neural network, we first need to define some basic concepts. A neuron is an information-processing unit. Neurons are connected

by synapses or connecting links, each of them characterized by a weight. Specifically, a signal x_j at the input of synapse j connected to neuron k is multiplied by the synaptic weight w_{kj}.

A neural network is a set of neurons organized in the form of layers. In the simplest form, an input layer projects onto an output layer of neurons. If the input layer has N units and the output layer has M units, each unit in the output layer owns N weights associated to the connections which come from the input layer, so that the set of neural weights is organized in the form of a two-dimensional lattice (W_{MxN}).

Our proposed network architecture is depicted in Figure 2 (Aguilar et al., 2000b). In our case, N corresponds to the number of channels of the hyperspectral image and M is the number of classes or prototypes to be extracted by the network. M must be carefully selected according to some metric (we will insist on this issue later on in the paper). There are feedforward connections from the input layer to the output layer and self-feedback and lateral feedback connections in the output layer. These two types of local connections serve two different purposes:

a) In the classification stage, the weighted sum $x \cdot W_i$ (scalar product) of the input signals x at each neuron i performs feature detection: each neuron produces a selective response to input signals.

b) In the learning stage, lateral and self-feedback connections produce excitatory or inhibitory effects depending on the distance from the corresponding output layer neuron to the winning neuron (Aguilar et al., 2000c). Their associated weights are used to determine the W_i classification prototype for each neuron.

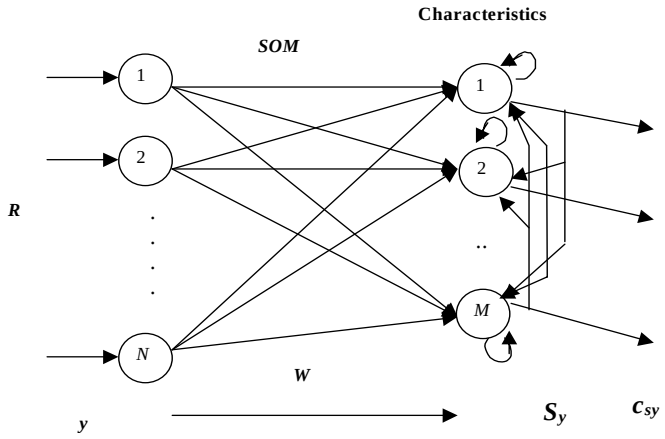


Figure 2. SOM neural network topology including weight matrix W and self-feedback and lateral feedback connections.

4. TRAINING ALGORITHM

There are five basic steps involved in the training algorithm. These steps are repeated until the topological map is completely formed:

a) Initialization: choose random values for the initial weight vectors, $w_i(0), i = 1,2,...M$. It is desirable to keep the magnitude of the weights small.

b) Sampling: choose an input pattern x(n) belonging to a set of learning patterns or references, R. The selection is done randomly.

c) Similarity Matching: find the best-matching (winning) neuron i* at time t, using the minimum-distance criterion as shown in the following equation, where dist is the euclidean distance:

$$i^*[x(n)] = \min_j dist\{x(n), w_j(t)\}, j = 1,2,...,M \quad (3)$$

d) Learning: adjust the synaptic weight vectors of all neurons, using the update formula (4), where $\eta(t)$ is a learning-rate parameter, and $\gamma\left(t,i,i^*[x(n)]\right)$ is a Gaussian neighborhood function centered around the winning neuron. The size of the neighborhood is determined by the reference distance $\sigma(t)$ (see equation 5).

$$w_i(t+1) = w_i(t) + \eta(t) \cdot \gamma\left(t,i,i^*[x(n)]\right)\left(x(n) - w_i(t)\right) \tag{4}$$

From the different options to select the previously mentioned parameters, taking into account the studies done in Aguilar, 2000, we have selected the following ones:

$$\eta(t) = \frac{1}{t}, \qquad \gamma\left(t,i,i^*[x(n)]\right) = \frac{e^{dist(i^*,i)^2}}{\sigma(t)}, \qquad \sigma(t) = \frac{\sigma_0}{t}. \tag{5}$$

e) Continue from step b) until no noticeable changes in the weight space are observed, or until the maximum convergence time is achieved.

In order to analyze a hyperspectral image using this algorithm, the network must be trained with hyperspectral signatures obtained directly from the image.

The weights initially associated with each output layer neuron contain the hyperspectral signatures of some carefully selected pixels on the image (according with their spatial distribution).

5. RESULTS AND DISCUSSION

We have applied our proposed neural network to real hyperspectral data, described in section 2. Since there are several parameters involved in the training algorithm (described in the previous section) in this section we analyze the influence of those parameters in the process of class prototype extraction. In particular, the parameters that we consider in the present study are the number of iterations until convergence of the neural network is reached, the size of neighborhood function γ centered around the winning neuron, which is determined by $\sigma(t)$, and the number of neurons in the output layer of the neural network.

The experiment is performed as follows. We train the network with all the hyperspectral signatures of the image. During the learning stage, we go through all the pixels of the image starting from a random pixel which is different in each of the iterations. Once class prototypes have been extracted, each pixel is classified and the confusion matrix (Chuvieco, 2000) is obtained. This matrix allows us to visualize winning neuron density for each class.

The characteristics of the confusion matrix provides us with a comprehensive visualization of distribution in N-dimensional space, and may indicate the accuracy of the classification. Since each column corresponds to an output neuron, if one column presents high values for different classes, the overall accuracy of the classification should be low. In order to measure the degree of accuracy of the classification, we propose the following metric based on the topology of the confusion matrix:

$$E_i = \frac{Xm_i}{Xs_i}, \qquad Xm_i = \underset{i}{Max}(X_{ij}), \qquad Xs_i = \sum_j X_{ij}. \tag{6}$$

Xm_i is the maximum value for a column of the confusion matrix and Xs_i is the sum of all the values in that column. E_i provides information about the capacity of each neuron to discriminate between the classes, and can be averaged for all the neurons in the network, providing a general measure about the accuracy of the classification.

The performance of the SOM Neural Network depends on a lot of adjusting parameters:

1. **Number of output neurons:** the ideal numbers of output neurons must be equal to the number of ground truth classes, associating exactly one neuron with one class. Usually this fact is not possible if we have a larger number of output neurons that ground truth classes. In this way, one correct classification

uses several neurons for each ground truth class. In the Indian Pines ground-truth image there are 16 classes (plus one class of unclassified pixels).

2. **Weigth initialization:** the weights w_{ij} initially associated with each neuron contain random values.
3. **Order to scan the image:** a random initial point of the image is selected at each iteration.
4. **Neighborhood function:** as mentioned before, our choice for the neighborhood function is the Gaussian function.
5. **Reference distance σ(t):** when the algorithm starts (t=1), the neural lattice is in a random state, the neighborhood function in these first iterations of the algorithm must have similar values for i to include a large number of neurons and obtain some average values (high σ(0) values). When t increases, $\gamma(t,i,i^*[x(n)])$ needs to be adjusted to reduce the number of neighbor neurons. Care must be taken into account to avoid a quickly reduction of the number of neighborhood neurons, this reduction can be accomplished by changing the reference distance $\sigma(t)$. Our choice for the reference distance evolution is $\sigma(t)=(1/t)^2$

The parameters that we analyze in the present study are the number of neurons in the output layer, the number of iterations of the neural network and the neighborhood starting value σ_0. Next, some results obtained for the hyperspectral data described in section 2 are provided. In our first experiment, we have considered 16 neurons in the output layer, 100 iterations and $\sigma_0 = 2$. Table 1 shows the resulting confusion matrix and Figure 3 shows the the resulting classification provided by the neural network using the previously mentioned parameters along with a greyscale representation of the confusion matrix shown in Table 1.

A favorable result would be obtained if neurons activate exclusively for a particular class, discriminating this class from the others. In the confusion matrix, this can be graphically expressed as a row for which several columns present high values. In Figure 3 we can appreciate this situation at four different rows (2,6,10,11 and 17). The fact that column values overlap indicates an inaccurate classification.

Another indicator of the quality of the classification is the continuity of high values in the rows of the confusion matrix (this fact should produce bright contiguous rows in the confusion matrix image). In this experiment, the topology of the resulting classes is not preserved since we can appreciate several discontinuities in the learnt classes. The overall accuracy of the classification obtained in this experiment was 60% according to the measure provided in equation 6. Nevertheless, we have to take into account that class 17 in the confusion matrix (see Table 1) corresponds to pixels which have not been classified during the process, and we are considering these pixels when calculating the overall accuracy (if we do not consider these pixels, accuracy increases to 80%).

		Neurons in the output layer															
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**
Classes	**1**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	CORN-NOTILL - 2	0	0	0	0	0	0	98	245	71	161	162	98	11	1	115	46
	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	GRASS - 6	242	9	253	160	52	9	0	0	0	0	0	0	1	1	5	0
	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	SOYBEAN 1 – 10	0	0	0	0	0	0	61	0	137	42	0	180	1	1	28	277
	SOYBEAN 2 – 11	12	0	8	0	0	0	471	3	521	136	8	471	66	6	17	207
	12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	UNCLASS. - 17	121	66	186	161	101	56	127	19	72	24	54	37	191	165	39	36

Table 1. Resulting confusion matrix for the Indian Pines dataset considering 100 iterations, 16 neurons in the output layer and $\sigma_0 = 2$. Non-zero cells in the matrix are highlighted.

In our second experiment, we consider a smaller number of neurons in the output layer (5), 80 iterations and $\sigma_0 = 4$. Figure 4 shows the resulting classification and the associated confusion matrix. We can appreciate an improvement in the topology of the confusion matrix (horizontal "lines" are more contiguous). In this case, the overall accuracy is 60%, without considering the unclassified pixels.

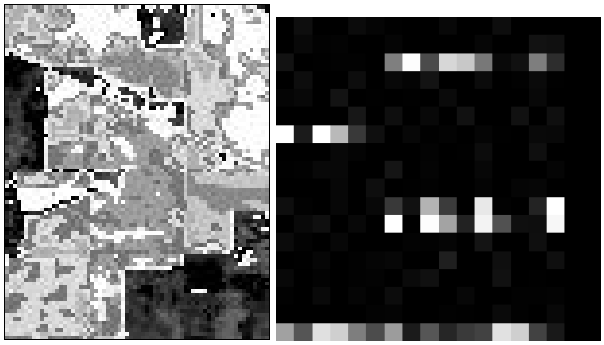
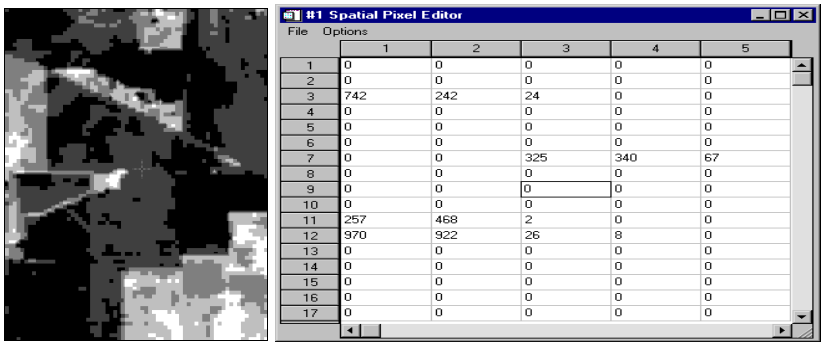


Figure 3. Resulting classification and greyscale representation of the confusion matrix for the Indian Pines dataset considering 100 iterations, 16 neurons in the output layer and $\sigma_0 = 2$. The time used for this computation was approximately 60 minutes in an AMD K-7 600 MHz Processor with 128 Mb of SDRAM memory and IDL 5.4.



	1	2	3	4	5
1	0	0	0	0	0
2	0	0	0	0	0
3	742	242	24	0	0
4	0	0	0	0	0
5	0	0	0	0	0
6	0	0	0	0	0
7	0	0	325	340	67
8	0	0	0	0	0
9	0	0	0	0	0
10	0	0	0	0	0
11	257	468	2	0	0
12	970	922	26	8	0
13	0	0	0	0	0
14	0	0	0	0	0
15	0	0	0	0	0
16	0	0	0	0	0
17	0	0	0	0	0

Figure 4. Resulting classification and associated confusion matrix for the Indian Pines dataset considering 80 iterations, 5 neurons in the output layer and $\sigma_0 = 4$. The time used for this computation was 20 minutes in an AMD K-7 600 MHz Processor with 128 Mb of SDRAM memory and IDL 5.4.

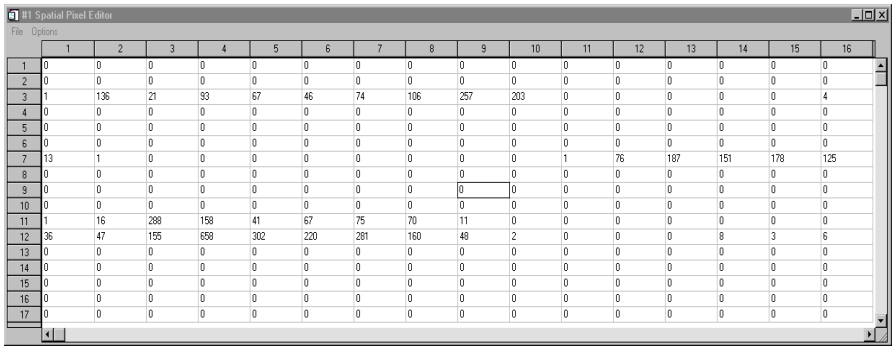
Finally, in our last experiment we increase the number of neurons in the output layer (16), we decrease the number of iterations (20) and consider $\sigma_0 = 12$. The results are addressed in Figure 5. A general improvement in the topology is achieved in this experiment, and the overall accuracy in this case is 72%, due to the fact that we compensate the increase in the number of neurons with a subsequent increase in the number of neurons that are considered in the competitive step ($\sigma_0 = 12$). Table 2 shows other experiments we have performed over the Indian Pines hyperspectral dataset.

As we can appreciate in Figure 5, the increase in the number of neurons produces some discontinuities in the topology, but the overall performance increases due to the reduction in the overlapping percentage between bright zones (high values) in the confusion matrix. These facts can be reduced increasing σ_0.

From the previously addressed results, we can conclude:

1. The number of iterations needed to obtain an acceptable accuracy is low compared to other SOM applications. This fact is probably related to the high dimensionality of hyperspectral signatures.

2. It is more efficient, in terms of accuracy, to increase the number of neurons and decrease the number of iterations. As we increase the number of neurons, the overall accuracy increases, but the topology of the classes is poor.
3. When the SOM neural network has a large number of neurons in the output layer, the initial reference distance must be increased in order to maintain similar performance values in the competitive process. In this sense, $\sigma(t)$ plays an important role.
4. Further work is still needed in order to achieve a reasonable compromise between topology preservation, $\sigma(t)$ function and overall accuracy.



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	1	136	21	93	67	46	74	106	257	203	0	0	0	0	0	4
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	13	1	0	0	0	0	0	0	0	0	1	76	187	151	178	125
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	1	16	288	158	41	67	75	70	11	0	0	0	0	0	0	0
12	36	47	155	658	302	220	281	160	48	2	0	0	0	8	3	6
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 5. Resulting classification and greyscale representation of the confusion matrix for the Indian Pines dataset considering 20 iterations, 16 neurons in the output layer and $\sigma_0 = 16$. The time used for this computation was 7 minutes in an AMD K-7 600 MHhz Processor with 128 Mb of SDRAM memory and IDL 5.4.

Number of neurons	Number of iterations	Size of neighborhood (σ_0)	Accuracy (%)
16	400	10	74
	150	6	73
	60	2	73
	40	14	73
	20	12	74
8	60	6	70
	60	4	70
6	150	2	66
	60	2	65
	60	2	63
5	80	4	60
	30	1	66

Table 2. Other experiments performed over the Indian Pines dataset.

6. CONCLUSIONS

We have presented a new approach to unsupervised classification of hyperspectral images using a Self Organizing Map. The overall performance of the method has been tested by its application to real hyperspectral data. The availability of ground truth allows us to introduce a new statistical measure to quantify the accuracy of the resulting classification. Since the training stage of the neural network incorporates several parameters, we have studied the influence of some of these parameters on the final result.

ACKNOWLEDGEMENTS

Fundings from Junta de Extremadura (PRI Program, IDUAP Grant) and European Community (LFR Program, TEITORS Grant) are also gratefully acknowledged.

REFERENCES

Aguilar, P.L , Martínez, P., Pérez R.M., Hormigo, A., "Abundance Extractions from AVIRIS Images Using a Self Organizing Neural Network," *Summaries of the IX JPL Airborne Earth Science Workshop*, pp. 281-286, JPL/NASA, 2000a.

Aguilar, P.L., *Cuantificación de Firmas Hiperespectrales Usando Mapas Autoorganizativos*, PhD Thesis, Escuela Politecnica de Cáceres, Universidad de Extremadura, 2000c.

Aguilar, P.L., Pérez, R.M., Martínez, P., Bachiller, P., Merchán, A., "Spectra Evaluation and Recognition in the Mixture Problem Using SOFM Algorithm," *Proc. International Symposium on Engineering of Intelligent Systems, (EIS'98)*, 1998, Vol. 2, pp. 118-124.

Aguilar, P.L., Plaza, A., Martínez, P., Pérez, R.M., "Endmember Extraction by a Self-Organizing Neural Network on Hyperspectral Images," *Proc. International Conference on Automation, Robotics and Computer Vision*, Nanyang Technological Institute, Singapore, 2000b.

Antonille, S. and Gualtieri, J.A., "Visualizing Clusters in High-Dimensional Data with a Kohonen Self Organizing Map," *Technical Report*, 2000.

Bruske, J. and Merényi, E., 1999, "Estimating the Intrinsic Dimensionality of Hyperspectral Images," *Proc. European Symposium on Artificial Neural Network, ESANN'99*, Brussels, Belgium, 21-23 April, 1999, pp. 105-110.

Chuvieco, E., *Fundamentos de Teledetección Espacial*, Ediciones Rialp, Spain, 2000.

Green, R.O., Editor, AVIRIS Earth Science Workshop Proceedings, 1988-2000. Available at http://makalu.jpl.nasa.gov/

Jimenez, L.O., Morales-Morell, A., Creus, A., "Classification of Hyperdimensional Data Based on Feature and Decision Fusion Approaches Using Projection Pursuit, Majority Voting, and Neural Networks," *IEEE Trans. Geoscience and Remote Sensing*, Vol. 37, Issue 3, Part 1, May 1999, pp. 1360 –1366.

Kohonen, T., *Self-Organizing Maps (2nd ed.)*, Springer Series in Information Science. 1.995-1.997

Martínez, P., Pérez, R.M., Aguilar, P.L., Bachiller, P. and Diaz, P., "A Neuronal Tool for AVIRIS Hyperspectral Unmixing," *Summaries of the IX JPL Airborne Earth Science Workshop*, pp. 281-286, JPL/NASA, 1999.

Pérez, R.M., Aguilar, P.L., Bachiller, P. and Martínez, P., "Neural Network Quantifier for Solving the Mixture Problem and its Implementation by Systolic Array," *Microelectronics Journal*, 30 (1), pp. 77-82, 1999.

Taudjin, S. and Landgrebe, D., *Classification of High Dimensional Data with Limited Training Samples*, Doctoral Thesis, School of Electrical Engineering and Computer Science, Purdue University, 1998.