

Self-Organizing Map for Hyperspectral Image Analysis

P. Martínez¹, P.L. Aguilar¹, R.M. Pérez¹, M. Linaje¹, J.C. Preciado¹, A. Plaza¹

¹Departamento de Informática, Universidad de Extremadura, Avda. de la Universidad s/n,
10071 Cáceres, SPAIN
{pablomar, paguilar, rosapere, mlinaje, jcpreciado, aplaza}@unex.es

Abstract. In this paper we present a neural network methodology used for classifying an hyperspectral image referenced as Indian Pines. The network parameters (learning and neighborhood function) are adjusted using a test battery generated from the image, selecting the values that give the best robustness and discrimination capacity. The availability of ground truth allows us to introduce a new statistical measure to quantify the resulting classification accuracy. The results of this methodology show an accuracy of 80% in the classification.

1. Introduction

The use of hyperspectral imaging sensor data to study the Earth's surface is based on the capability of such sensors to provide high resolution spectra, on a per pixels basis, along with the image data.

Hyperspectral sensor provides a large number of narrow bands that provides sensor capabilities to recognize narrow absorption band, like the laboratory measurements. This capability can be used to classify and determine Earth's surface constituents signatures from the hyperspectral information provided by the sensor. Conventional algorithms use these signatures to classify and/or determine the abundances of a composite pixel spectrum [1].

This problem is known as *Hyperspectral Image Analysis*. Highly successful results have been obtained following this approach [2],[3], particularly in the field of geology where exposed lithographies have been mapped based on specific mineral hyperspectral reflectance signatures [4].

The great size of the hyperspectral images is one of its principal disadvantages, usually 30 times larger than a Landsat TM image of the same spatial size.

The algorithms for conventional multi-spectral sensors can't be used with hyperspectral images by the following reasons:

- High dimensionality of the images.
- Great size of the image (hundred of bands, thousands of pixels)
- Sub pixel analysis possibilities.

- High discrimination to resolve classes
- Difficulties to use training data.

Un-supervised clustering is a problem with application in many areas. Given a set of N data points in a feature space R_D of D dimensions $(x_1, x_2, x_3, \dots, x_D)$, we wish to characterize the data as belonging to K cluster where K , must be obtained from the data.

Parra et al. [5] assume linear combinations of reflectance spectra with some additive normal sensor noise and they derive a probabilistic maximum a posteriori framework for analyzing hyperspectral data. The material reflectance characteristics are not know a priori, so this is the problem of unsupervised linear unmixing. The incorporation of different prior information (positivity and normalization of abundances) leads to a family of algoritms. In [5] the constrained independent component analisis (ICA) for the noise-free case is used.

The clustering is based on some distance metrics and one of the most usefully unsupervised algorithms is the Self-Organizing Neural Network or Self Organizing Map (S.O.M) proposed by T. Kohonen [6].

The present work exploits the possibility of using a Self-Organizing Neural Network to analyze the hyperspectral images. Some reason for using this neural network model, in hyperspectral analysis has been described by Bruske and Merényi [7]:

- To avoid the need to degrade the data
- To provide speed (when implemented in hardware as massively parallel algorithm.
- To surpass conventional classification algorithm performance.
- Good performance for large real life task.

The Self-Organizing neural network has the advantage that obtains by competitive procedures the class prototypes. This ability can be used for the classification of the hyperspectral images. We need tools that match up the intricacy of hyperspectral data. We propose a neural network for processing the hyperspectral information for each pixel. The neural model consists on Self-Organizing Neural Network. This net has an input neuron for each image channel. The output neurons number is related with the characteristics of the image and must be carefully optimized according some metric. Different distances and learning functions are used to obtain a better class prototypes extraction.

The result discussion also includes the influence of the following parameters in the Neural network performance:

- Neighborhood function
- Learning function
- Noise contamination in the spectra.
- Output layer geometry

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 whole 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 [8], 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, which is depicted in Fig 1.

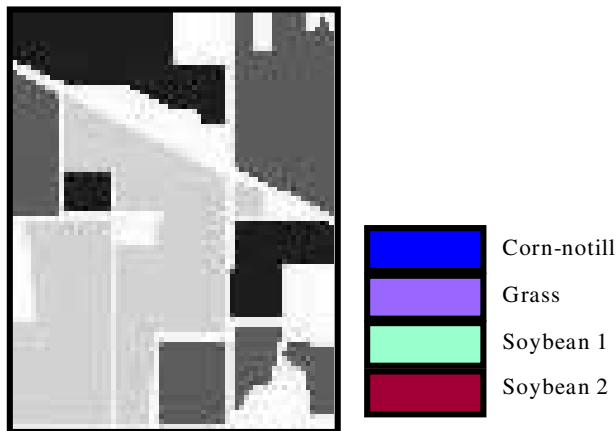


Fig. 1. A subset of the Indian Pines hyperspectral dataset with ground truth.

3. Topology of the Proposed Neural Network

The Self-Organizing Map (SOM) is based on a 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 firstly 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} [9]. 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 that come from the input layer, so that the set of neural weights are organized in the form of a two-dimensional

lattice ($W_{M \times N}$). Our proposed network architecture is depicted in Fig. 2 [10],[11]. 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 with 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. In the classification phase, the input signals x are projected on the feature space by the feedforward connections W , each neuron produces a selective response to input signals. In the learning phase, lateral and feedback output layer connections produce excitatory or inhibitory effects depending on the distance from the neuron to the winner neuron [12],[13]. These weights are used to determine the W_i classification prototype per each neuron.

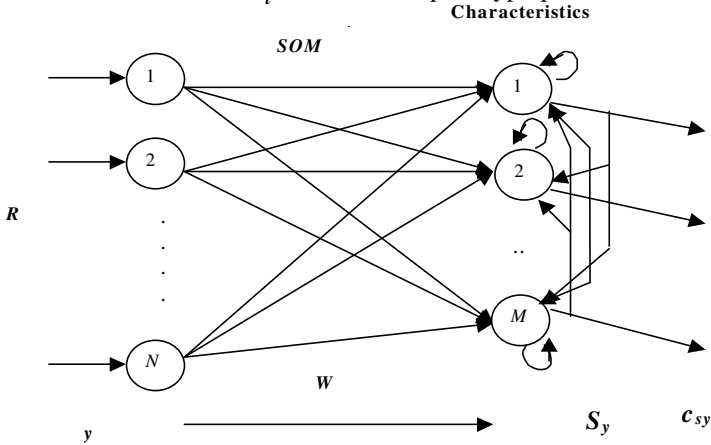


Fig. 2. SOM neural network topology including weight matrix W and learning lateral and feedback connections.

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) Initialisation. Choose random values for the initial weight vectors $w_i(0)$, $i = 1, 2, \dots, M$. 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:

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

where $\text{dist}(i^*, i)$ is the euclidean distance.

- d) Learning. Adjust the synaptic weight vectors of all neurons, using the update formula (2), where $\eta(t)$ is a learning-rate parameter, and γ is a Gaussian neighborhood function centred around the winning neuron. The size of the neighborhood is determined by a parameter $\sigma(t)$ (see equation 3).

$$w_i(t+1) = w_i(t) + \eta(t) \gamma(t, i, i^*[x(n)])(x(n) - w_i(t)) \quad (2)$$

From the different choice for the selection of the involved parameters, taking into account the studies made in [13], we have choose the following:

$$\eta(t) = \frac{1}{t} \quad \gamma(t, i, i^*[x(n)]) = e^{-\frac{\text{dist}(i^*, i)^2}{\sigma(t)}} \quad \sigma(t) = \left(\frac{\sigma_0}{t}\right)^2 \quad (3)$$

where σ_0 is the initial width that changes in the results that we present later.

- 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 Neighborhood and Learning Function Selection

We have realized a lot of experiments with a predetermined training set, to establish the optimum values to the neighborhood and the learning parameters for the rest of the studies.

Each training set has a fixed number of signatures with different sets generated as follows: the first set contains ten reference signatures free of noise; the second set has twenty signatures those ten and ten new signatures corresponding to noise version of them, one for each reference, the third contained thirty signatures (two noise signatures for each reference); the last set had a hundred of signatures (ten noise

signatures for each reference). These sets have been used to train ten different networks.

Hundred signatures that don't belong to the training set have formed the validation set.

5.1. Learning function

We have analyzed five different learning functions given in 4:

$$\begin{aligned}
 a) \quad \eta(t) &= \frac{1}{t} & b) \quad \eta(t) &= \frac{1}{\sqrt{t}} \\
 c) \quad \eta(t) &= \eta_1 \left(1 - \frac{t}{\eta_2} \right) & d) \quad \eta(t) &= \eta_1 e^{-\frac{t}{\eta_2}} \\
 e) \quad \eta(t) &= \eta_1 \left(\frac{\eta_1}{\eta_2} \right)^{\frac{t}{t_{\max}}}
 \end{aligned} \tag{4}$$

Where η_1 is the initial learning value (normally equal to 0,9 in c) and d) and 0,8 in e), η_2 is the final learning value (0,08 in e) or the number of iterations in c) and d)) and t_{\max} is the number of iterations.

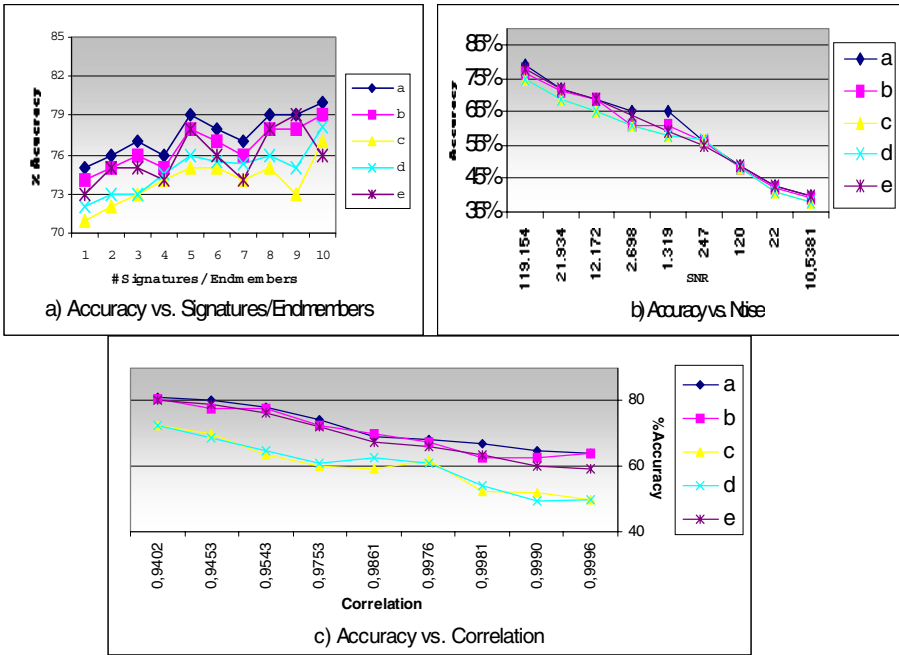


Fig. 3. Learning parameter study

The Fig. 3 shows the obtained results for three types of studies; it represents the accuracy versus the endmembers number (figure a.), the Signal to Noise Ratio defined as $SNR = E(\text{signal})/E(\text{noise})$, (figure b.) and the correlation between endmembers (figure c.). We can observe that the better results are obtained by the equation 4.a.

5.2. Neighborhood function

We have analyzed different neighborhood functions given in 5.

$$\sigma(t, i, i^* [x(n)]) = \begin{cases} a) & e^{\left[\frac{-d_L(i, i^* [x(n)])^2}{2\sigma(t)^2} \right]}, & \sigma(t) = \frac{\sigma_0}{t} \\ b) & \frac{\text{sen}\left(\left[\frac{-d_L(i, i^* [x(n)])}{\sigma(t)^2} \right]\right)}{d_L(i, i^* [x(n)])} \sigma(t), & \sigma(t) = \frac{\sigma_0}{t} \\ c) & e^{\left[\frac{-d_L(i, i^* [x(n)])^2}{\sigma^2(t)} \right]}, & \sigma(t) = \sigma_0 \left(\frac{\sigma_1}{\sigma_0} \right)^{\frac{t}{t_{\max}}} \end{cases} \quad (5)$$

Where σ_1 is the initial neighborhood value (normally equal to middle of map), σ_2 is the final neighborhood value (0,08 in c) and t_{\max} is the number of iterations.

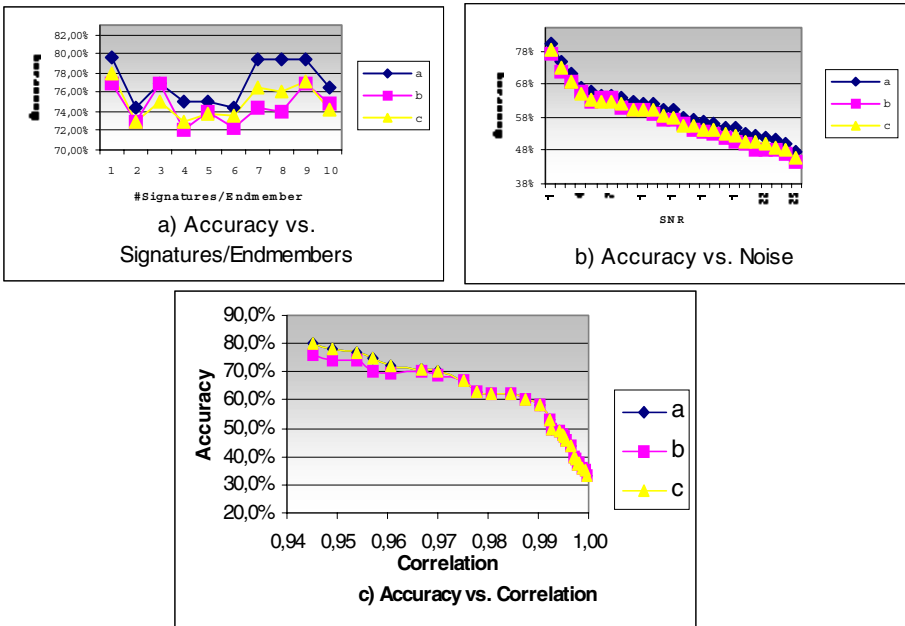


Fig. 4. Neighborhood parameter study

The figure 4 shows the obtained results for the same types of studies used in the learning function. We can observe in fig. 4 the obtained results for the same types of

studies used in the learning function; the better results are obtained by the equation 5.a.)

The previous studies justify the choice of the used functions in the next section.

6. 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 section 4); 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 that is different in each of the iterations. Once class prototypes have been extracted, each pixel is classified and the confusion matrix [14] is obtained; this matrix allows us visualize the winner neuron density for each class and can be used as one different approach of the Umatrix [15],[16].

The characteristics of the confusion matrix provide 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 in the same column, high values for different classes are presents, 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:

$$E_i = 100 \frac{X_{mi}}{X_{si}} \quad X_{mi} = \max_i(X_{ij}) \quad X_{si} = \sum_j X_{ij} \tag{6}$$

X_{mi} is the maximum value for a column of the confusion matrix and X_{si} 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.

Next, some results obtained for the hyperspectral data described in section 2 are provided. In the experiment, we have considered 16 neurons in the output layer, $\sigma_0 = 2$ and 100 iterations. Fig. 5 shows the resulting classification provided by the neural network along with a greyscale representation of the confusion matrix obtained.

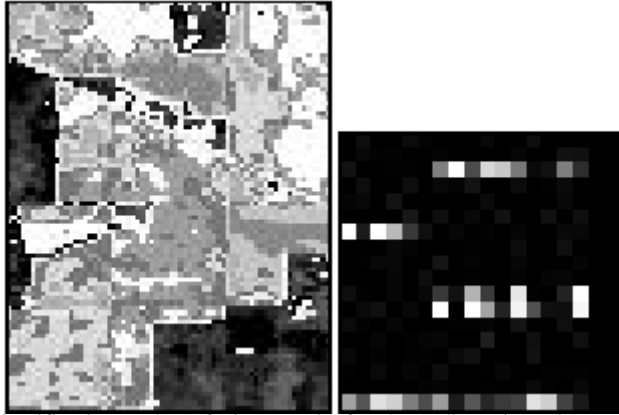


Fig. 5. Resulting classification and confusion matrix for the Indian Pines dataset considering 100 iterations, $\sigma_0 = 2$ and 16 neurons in the output layer.

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 Fig. 5 we can appreciate this situation at four different rows, indicating that there are four major classes in the image. The fact that column values overlap indicates an inaccurate classification. Another indicator of the quality of the classification is the continuity of bright zones in the confusion matrix. 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. These results are obtained when we have signatures that don't belong to any class, if these signatures are not included, the accuracy increases to 80%.

One more detailed discussion of these results can be obtained in [17], from these results we can conclude two suggestion to modify the SOM learning algorithm:

1. Use the confusion matrix in the learning phase to avoid the misclassification problems associated with several winner neurons for the same learning vector. The unsupervised confusion matrix CM will be one $M \times M$ matrix and can be filled increasing for each learning vector, whose winner neuron is I , the value of $CM(I,j) = CM(I,j) + S_j(X_k)$. The high values of $CM(I,j)$ should be for $I=j$, in the other hand must be avoided by one hard competition between I and j neurons.
2. Take into account for the hyperspectral classification the high order statistics [8] collected during the learning phase, one possibility is to store different SOM neural networks parameters for each pair of neurons I,j :
 - a. $\gamma_{I,j}$
 - b. $\eta_{I,j}$
 - c. $\sigma(t)_{I,j} \quad \sigma_{0,I,j}$

7 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

We would like to thank Applied Information Sciences Branch for their help and support during our visits to NASA/Goddard Space Flight Center. Fundings from Junta de Extremadura (PRI Program, IDUAP Grant) and European Community (LFR Program, TEITORS Grant) are also gratefully acknowledged.

References

- [1] Green, R.O., Editor, AVIRIS Earth Science Workshop Proceedings, 1988-2000. Available at <http://makalu.jpl.nasa.gov/>
- [2] Ifarraguerrri, A.; Chang, C.-I. "Multispectral and Hyperspectral Image Analysis with Convex Cones". *IEEE Trans. Geoscience and Remote Sensing*, Vol. 37 Issue 2 Part 1, March 1999, 756-770.
- [3] Jimenez, L.O., Morales-More ll, 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., 1360-1366.
- [4] Richardson L and Kruse F.A., "Identification and Classification of mixed phytoplankton assemblages using AVIRIS image derived spectra" *Summaries of the VIII JPL Airborne Earth Science Workshop* . (1999), 339-347
- [5] Parra L, Spence C., Sadja P, Ziehe A., Müller K-R, "Unmixing Hypersepectral Data" *Proc. Advances in Neural Information Processing System 12, Proc. Of the 1999 Conference*, MIT Press (1999) 942-948
- [6] Kohonen, T., *Self-Organizing Maps* (2 nd ed.) , Springer Series in Information Science. (1997)
- [7] Bruske, J and Merényi, E.: "Estimating the Intrinsic Dimensionality of Hyperspectral Images ". *Proc. European Symposium on Artificial Neural Network, ESANN'99* ,Bruges, Belgium, (1999) , 105-110 .
- [8] Taudjin S and Landgrebe D., Classification of High Dimensional Data with Limited Training Samples, Doctoral Thesis, School of Eleectrical Engineering and Computer Science, Purdue University. (1998).
- [9] Martínez P., Pérez R.M., Aguilar P.L., Bachiller, P. and Diaz, P., "A Neuronal Tool for AVIRIS Hyperspectral Unmixing", *Summaries of the VIII JPL Airborne Earth Science Workshop* , JPL/NASA, (1999), 281-286.
- [10] 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)*, Vol. 2 , (1998), 118-124.
- [11] 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, (2000).
- [12] 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*, JPL/NASA (2000), 281-286,.
- [13] Aguilar P.L., "Cuantificación de Firmas Hiperespectrales Usando Mapas Autoorganizativos", Ph. D. Thesis, Escuela Politécnica, Universidad de Extremadura, (Chapter 5), 2000.
- [14] Chuvieco E., *Elementos de Teledetección Espacial*, Ed Rialp (2000)
- [15] Ultsch, A. and Siemon, H.P., "Kohonen's Self Organizing Feature Maps for Exploratory Data Analysis", *Proc. ICNN'90 International Neural Network Conference* , (1990), 305-308.
- [16] Antonille, S. and Gualtieri, J.A., "Visualizing Clusters in High-Dimensional Data with a Kohonen Self Organizing Map". *Summaries of the IX JPL Airborne Earth Science Workshop*, JPL/NASA (2000) 281-286.
- [17] Martínez, P, Gualtieri, J.A., Aguilar, P.L., Pérez R, Linaje M, Preciado J.C. , Plaza A., "Hyperspectral image classification using self-organizing map", *Summaries of the X JPL Airborne Earth Science Workshop*, in press, JPL/NASA, 2001.