

# A combined measure for quantifying and qualifying the topology preservation of growing self-organizing maps

Soledad Delgado <sup>a,\*</sup>, Consuelo Gonzalo <sup>b</sup>, Estibaliz Martinez <sup>b</sup>, Agueda Arquero <sup>b</sup>

<sup>a</sup> Departamento de Organización y Estructura de la Información, Universidad Politécnica de Madrid, Ctra. Valencia, Km. 7, 28031 Madrid, Spain

<sup>b</sup> Departamento de Arquitectura y Tecnología de Sistemas Informáticos, Universidad Politécnica de Madrid, Campus de Montegancedo, 28660 Madrid, Spain

## A B S T R A C T

### Keywords:

Topology preserving  
Self-organizing map  
Growing cell structures  
Visualization methods  
Delaunay triangulation

The Self-Organizing Map (SOM) is a neural network model that performs an ordered projection of a high dimensional input space in a low-dimensional topological structure. The process in which such mapping is formed is defined by the SOM algorithm, which is a competitive, unsupervised and nonparametric method, since it does not make any assumption about the input data distribution. The feature maps provided by this algorithm have been successfully applied for vector quantization, clustering and high dimensional data visualization processes. However, the initialization of the network topology and the selection of the SOM training parameters are two difficult tasks caused by the unknown distribution of the input signals. A misconfiguration of these parameters can generate a feature map of low-quality, so it is necessary to have some measure of the degree of adaptation of the SOM network to the input data model. The topology preservation is the most common concept used to implement this measure. Several qualitative and quantitative methods have been proposed for measuring the degree of SOM topology preservation, particularly using Kohonen's model. In this work, two methods for measuring the topology preservation of the Growing Cell Structures (GCSs) model are proposed: the topographic function and the topology preserving map.

## 1. Introduction

Nowadays there are many research areas that generate large volumes of information of high dimensional nature, which require data mining tools to extract behavior models through the data analysis [1]. Data mining is defined as the process of extracting information that resides implicitly in the data. Several techniques are included under this discipline, their main purpose being extracting useful information and knowledge present in databases characterized by a high dimension, size or complexity, excluding, in this way, the use of manual analysis techniques. Usually, data mining first stage consists of building simplified global overviews of datasets, generally in a graphical form [2]. Neural networks are one of the models commonly used in data mining. These models stand out from the rest by their suitability for solving problems associated with datasets with *a priori* unknown statistical distribution. The neural network model most widely used in this area is the Self-Organizing Map (SOM). SOM fits well in the exploratory data analysis since its main purpose is the visualization and analysis of nonlinear relations implicit in multidimensional data [3]. The SOM

model generates similarity maps of the input data, transforming nonlinear statistical relationships of multidimensional patterns in simple geometric relationships on a low-dimensional graph, which usually looks like a two-dimensional mesh of nodes. In addition, SOM networks compress information while preserving the most important metric and topological relationships associated with the data set. In this way, the SOM network is a simplified model of the data space. These properties make SOM an ideal model for exploitation in several visualization techniques of multidimensional information [4–7]. All of them are based on exploring different features related to the knowledge learned by the network.

The SOM model most widely known is that proposed by Kohonen [8]. This network generates a mapping of the input space (dataset  $M$ ), which is usually multidimensional, in the two-dimensional output mesh of the network (lattice  $A$  of  $N$  neural units). Each output unit has an associated vector, usually known as synaptic vector, with the same dimension as  $M$ . The neurons in the output layer are organized in a discrete two-dimensional grid, which establishes the neighboring connection architecture between them. The SOM algorithm determines the method of creating the input space mapping. This is an iterative process of training that adapts the synaptic vectors values of the network to perform a particular task. During SOM learning phase, a set of input vectors  $\in M$  are presented to the network in an iterative way. For each input pattern, the best matching unit is computed (the

\* Corresponding author. Tel.: +34 91 336 78 79; fax: +34 91 336 75 20.

E-mail addresses: sole@eui.upm.es (S. Delgado), chelo@fi.upm.es (C. Gonzalo), emartinez@fi.upm.es (E. Martinez), aarquero@fi.upm.es (A. Arquero).

one with the synaptic vector closest to the input pattern) and the synaptic vector of the winner and all neurons that belong to its neighborhood area are modified slightly, making it closer to that input. The purpose of this adaptation process is to map similar input patterns into neighbor output units, as well as neighbor output units map only similar input patterns. Informally, this can be the definition of topology preservation [9], which shows the quality of the simplified model and the degree of success in the choice of the neighborhood connection architecture of the output layer neurons. A misconfiguration in the SOM output layer topology or an improper tuning of the network training parameters can result in erroneous feature maps. Therefore, it is necessary to create tools to evaluate the quality of trained networks in order to validate their subsequent exploitation.

Several qualitative and quantitative methods for measuring the topology preservation degree of SOM have been proposed in the literature, in particular using Kohonen's self-organizing feature map model, as the quantization error [3], the topographic error [10], or the topographic product [11]. None of these approaches use the dataset itself for measuring, so if the manifold is nonlinear, as in many practical applications of SOM, none of these methods can distinguish a correct folding in the network due to a fold in a nonlinear data set from one due to a topological error between  $M$  and  $A$  [12]. Some methods that explicitly take into account the structure of the dataset have been proposed for Kohonen's SOM, such as the topographic function or the connectivity strength matrix. Villmann et al. presented in [12] the topographic function to measure the topology preservation grade for Kohonen's model. While the information provided by this function quantifies the volume of topology preserving violations, it does not provide enough information about the seriousness of the same. Tasdemir and Merényi [13] introduced a new visualization scheme to represent data topology superimposed on Kohonen's SOM grid: the connectivity strength matrix. This map can help to analyze the weightiness of the topology violations of the network graphically but, a priori, it does not provide quantitative information about the volume of infractions.

In Kohonen's SOM the lattice of the output layer has to be pre-specified at the beginning of the training process and it remains static during all the adaptation steps. Sometimes it is not possible to decide *a priori* the neighborhood connection architecture that fits well with the structure of the dataset. In some cases even knowing the input data space, it is not viable to configure an appropriate architecture, e.g. when the input space consists of several separate regions of positive probability density. These situations can cause Kohonen's model to fail providing perfectly topology preserving maps. To overcome the static architecture of Kohonen's model, several incremental SOM have been proposed. The Incremental Grid Growing (IGG) [14], Grid Growing SOM (GGSOM) [15], Growing SOM (GSOM) [16], Growing Neural Gas (GNG) [17] and Growing Cell Structures (GCSs) [18] are some of the best known incremental SOM. IGG y GSOM models preserve a regular two-dimensional neighborhood grid at any moment during the self-organizing process, ensuring that topological relations can be easily examined by plotting the neurons and the connections of the map in 2-D. However, in both models the number of neighboring units associated with a neuron is limited (similar to Kohonen's model), which can lead to low degrees of topology preservation in certain cases. GGSOM architecture maintains a regular grid neighborhood connection in the output layer, with a fixed dimension, which may increase during the adaptation process of the network. As it does not ensure a dimension equal to two in the output layer, this model cannot be used to generate 2D maps to visually analyze the network. GGSOM and GSOM models do not implement any mechanism for eliminating neurons and connections in the output layer during

the training process, so it can generate low levels of topology preservation if the input space consists of separate clusters of data. On the other hand, GCS and GNG models incorporate insertion and removal of neurons and connections in the output layer of the network. GNG does not ensure a fixed size of the neighborhood connection topology of the output layer, allowing the existence of different dimensions according to the characteristics of the input data. This usually makes this model better suited to the properties of any input space, but has the disadvantage of not being able to generate two-dimensional output layer maps to analyze the intrinsic information of the network. The GCS model ensures a particular dimension in the neighborhood connection topology. For a neighborhood connection factor  $d=2$ , output layer of the network exhibits a two-dimensional nature that can be exploited in the visualization processes of multidimensional input spaces. Neuron insertion and removal, and the capability of projecting multidimensional information in two-dimensional maps have made GCS a model particularly interesting within the field of self-organizing maps.

GCS network architecture consists of connected units composing  $d$ -dimensional hypertetrahedron structures linked between them. The interconnection scheme defines the neighborhood relationships. Thus for example, for  $d=2$ , the GCS output layer exhibits a structure whose connections are established by a system of triangles. During the learning process, new neurons are added in those areas of the map where a heuristic measure determines that they inadequately represent their corresponding areas in the input space. Also, neurons located in areas of the input space with low or null probability density are periodically removed from the map. This can lead to the division of the output layer into two or more isolated meshes of neurons. In this way, the adaptation algorithm does explicitly represent clustering of the input data by removing connections between the clusters in the structure. Insertion and removal of units are performed in such way that the original architecture structure is maintained (i.e., every node must always be a member of at least one triangle). Using the method proposed in [19] by the authors of this paper, the GCS two-dimensional output layer can be plotted in a two-dimensional graph (the topographic map) giving a way to make use of this model in visualization techniques of high dimensional information. Usually, the flexibility of the output layer lattice in the GCS network provides better degrees of topology preservation than Kohonen's model.

In this work it is proposed to combine two methods to measure the topology preservation of the GCS model: the topographic function and the topology preserving map. Both techniques used separately do not give a complete vision of topology transgressions, but in combination they offer the possibility of quantifying the average number of neurons that generate violations and qualify their severity. In this paper, an adaptation for calculating the topographic function for the GCS model has been made. Moreover, the connectivity strength matrix map generation (topology preserving map) for the GCS model is proposed, in order to complement the information that the topographic function does not provide.

The remaining of the paper is organized as follows: Sections 2 and 3 describe the topographic function and the topology preserving map for GCS networks, respectively. The experiments and results are discussed in Section 4 and finally, conclusions are established in Section 5.

## 2. The topographic function in GCS

The output layer in GCS network is organized by neurons that exhibit neighborhood connections forming  $d$ -dimensional structures.

Normally, factor  $d$  takes value 2, thus these structures have the shape of a triangle. As in Kohonen's model, each output neuron  $i$  has associated a synaptic vector,  $w_i$ , with the same dimension that the input patterns ( $M \subset R^d$  and  $w_i \in R^d$ ). In a GCS trained network, the synaptic vectors of the  $N$  neurons of the output layer ( $A$ ) represent the prototype vectors of the training patterns. Each neuron  $i \in A$  has a receptive field for which the best matching unit is  $bm_u$ , known as the Voronoi region ( $V_i$ ), where the synaptic vector  $w_i$  is the kernel point. Each neuron represents those training patterns that are within its Voronoi region.

The neighborhood connections of the GCS output layer define the connectivity matrix  $C_{ij}^{(A)}$  of the  $A$  network

$$C_{ij}^{(A)} = \begin{cases} 1, & \text{if } i \text{ and } j \text{ are connected in } A \\ 0, & \text{otherwise} \end{cases} \quad \forall i, j \in A \ (i, j = 1, \dots, N). \quad (1)$$

$C_{ij}^{(A)}$  describes the lattice of  $A$  and is used to obtain the symmetric square matrix  $D_A$ . This matrix will keep the distance between each pair of neurons  $i$  and  $j$ , understood as the minimum number of neighborhood connections that separate  $i$  and  $j$  according to  $C_{ij}^{(A)}$ . As a result of the removal of neurons during the weight adaptation process of the network, the neighborhood lattice of the output layer in the GCS network can be divided into two or more isolated meshes. In this case, when calculating  $D_A$ , the distance between two neurons that belong to two separate meshes is theoretically  $\infty$ .

In addition, a new connectivity matrix can be defined based on the characteristics of the dataset. Each input pattern has an associated  $bm_u$  and a second  $bm_u$ , this being the last neuron of the output layer whose synaptic vector is the second one most similar to the pattern (by means of Euclidean distance). With the concept of  $bm_u$  and second  $bm_u$  a new connectivity matrix  $C_{ij}^{(M)}$  can be generated, which will reflect how the output neurons should be connected using the knowledge inherent to  $M$ :

$$C_{ij}^{(M)} = \begin{cases} 1, & \text{if } i \text{ is } bm_u \text{ and } j \text{ is second } bm_u \text{ for some input pattern} \\ 1, & \text{if } j \text{ is } bm_u \text{ and } i \text{ is second } bm_u \text{ for some input pattern} \\ 0, & \text{otherwise} \end{cases} \quad \forall i, j \in A \ (i, j = 1, \dots, N). \quad (2)$$

This new connectivity matrix reflects what is known as the induced Delaunay triangulation [12].  $C_{ij}^{(M)}$  matrix is used to calculate the symmetric square matrix  $D_M$ , which will keep the distance between each pair of neurons  $i$  and  $j$ , understood as the minimum number of neighborhood connections that separate  $i$  and  $j$  according to the lattice established by  $C_{ij}^{(M)}$ . Neighborhood connection established by the induced Delaunay triangulation ( $C_{ij}^{(M)}$ ) may result in a structure composed of several separate meshes. As in the case of the  $D_A$  matrix, when  $D_M$  matrix is calculated, the distance between two neurons located in two disconnected meshes is theoretically  $\infty$ .

Using  $D_M$  y  $D_A$  distance matrices, the topographic function  $\Phi_A^M$  [12] can be defined for GCS network as

$$\Phi_A^M(k) = \begin{cases} 1/N \sum_{i \in A} f_i(k) & k > 0 \\ \Phi_A^M(1) + \Phi_A^M(-1) & k = 0, \\ 1/N \sum_{i \in A} f_i(k) & k < 0 \end{cases} \quad (3)$$

where

$$\begin{aligned} f_i(k) &= \#\{j | D_A(i, j) > k; D_M(i, j) = 1\} \quad j = 1, \dots, N \\ f_i(-k) &= \#\{j | D_A(i, j) = 1; D_M(i, j) > k\} \quad j = 1, \dots, N. \end{aligned} \quad (4)$$

The expression  $\#\{.\}$  denotes the cardinality of the set, and  $k$  is defined in the range  $[1 \dots N-1]$ . A GCS network has a perfect topology preservation if  $\Phi_A^M(0) = 0$ . In the range of positive values of  $k$ ,  $\Phi_A^M(k)$  function calculates the mean number of neighboring units per neuron that should exhibit a direct neighbor connection in  $A$ , but present a higher distance than  $k$ . On the other hand, in the range of negative values of  $k$ ,  $\Phi_A^M(k)$  function determines the mean number of neighboring units per neuron that are direct neighbors in  $A$ , but should be at a distance greater than  $|k|$ . The topographic function can be drawn in the  $(-N \dots N)$  range, giving a graphical fast evaluation tool to measure the topology preserving quality in a trained GCS network.

The topographic function defined by Eq. (3) has the disadvantage that it does not accurately reflect the relevance of the topology violations with respect to the input data space, as the connectivity matrix  $C_{ij}^{(M)}$  does not weigh how many input vectors may be causing the infringement. For this reason, the topology preserving map is proposed in this work as a complementary measure to address this problem.

### 3. GCS topology preserving map

When  $d=2$  architecture factor is used, the GCS output layer neurons are organized in groups of interconnected triangles, forming a bi-dimensional mesh that can be projected in the plane. In [18], Fritzke presents a drawing method based on a physical force analogy that works reasonably well when the input space is low-dimensional (2D or 3D), but is not guaranteed to produce planar drawing. Delgado et al. [19] proposed a new approach for avoiding this restriction in order to embed the GCS output layer structure in the plane, independently of the dimension of the input space, for the case  $d=2$ . The resulting graph is known as the topographic map, which shows the output neurons and the neighborhood connections, ensuring that neighbor units

appear near in the map and neighborhood connections do not cross each other. Topographic map provides the basic support to implement different visualization methods of multidimensional information, similar to those generated using Kohonen's network [19].

The GCS topographic map and the dataset can be used to produce a graph that reflects the degree of topology preservation, equivalent to the one proposed in [13] for Kohonen's network. This map will reflect the neurons of the output layer that should be direct neighbors on  $A$  as well as the strength of these connections, according to the dataset structure  $M$ , i.e., how strong each pair of neurons should be connected on the basis of information provided by the input space. If the GCS network is perfectly topology preserving, only the immediate neighbors in  $A$  should be connected. Otherwise, connections may exist between neurons, which are not immediate neighbors in  $A$ . In order to be able to detect violations of the topology preservation, it is necessary to generate in parallel the topographic map of the network to visualize the connections that really exist in the lattice  $A$ .

First of all, the cumulative adjacent matrix  $CADJ$  is calculated, where  $CADJ(i, j)$  will keep the number of training patterns for which  $i$  is the  $bm_u$  and  $j$  is the second  $bm_u$ . In a second step, the connectivity strength matrix,  $CONN$ , can be obtained as  $CONN(i, j) = CADJ(i, j) + CADJ(j, i)$ , that shows how strongly two

neurons must be connected in the basis of the dataset structure.

In the topology preserving map, neurons are placed in the coordinates calculated for the topographic map of the network [19]. Between each pair of neurons  $i$  and  $j$  with  $CONN(i,j) > 0$ , a line connecting them is included. The strength of connections is reflected by the thickness of the line, and is calculated as

$$\text{line width } (i,j) = \begin{cases} 1 & \mu - \sigma > CONN(i,j) > 0 \\ 2 & \mu > CONN(i,j) > \mu - \sigma \\ 3 & \mu + \sigma > CONN(i,j) > \mu \\ 4 & CONN(i,j) > \mu + \sigma. \end{cases} \quad (5)$$

where  $\mu$  represents the mean value and  $\sigma$  the variance of the strengths of all connections in  $CONN$ . Value 1 is associated with the weakest line and 4 with the strongest one. In addition, 5 levels of gray are used to reflect the local strength of the connections for each neuron, associating high values with dark tones and low values with clear. A dark line between two neurons  $i$  and  $j$  will indicate that there exists a high volume of data for which  $i$  is  $bm_u$  and  $j$  the second  $bm_u$  or vice versa.

Lines connecting units in the topology preserving map, which are not direct neighbors in the lattice defined by  $A$  are considered topology violations. Usually local violations tend to arise, that is, fine and clear gray lines connecting units that belong to the same isolated cluster of neurons. Serious violations will be represented by strong dark lines connecting units of separate clusters according to  $A$ .

#### 4. Experiments and discussion

To evaluate the potential of combining the two methods exposed previously in this paper, this section presents four experiments conducted with two datasets. In both cases the input space,  $M$ , consists of vectors distributed in several clusters. These experiments show the usefulness of the two methods to determine the quality of a trained GCS network, thus providing an objective criterion to analyze the number of clusters present in the input space, and a robust base to select the GCS network that best fits with an input space from a set of networks trained with the same dataset.

The first dataset consists of eleven normally distributed clouds of two-dimensional data points (small light gray dots in Fig. 1). This dataset was generated artificially from eleven centroids, producing 178 points normally distributed around each one.

The second is a real-world dataset, formed by a group of four-dimensional pixels selected from a scene registered by the multi-spectral optical sensor onboard the QuickBird satellite, with a spectral resolution of four bands: blue (450–520 nm), green (520–600 nm), red (630–690 nm) and near infrared (760–900 nm). It consists of 1460 pixels distributed over seven land cover categories as follows: 61 vectors of dark-ways, 91 of dark vegetation, 299 of light vegetation, 187 of dark mixed, 411 of mixed, 96 of bright ways and 315 of bright soil. Fig. 2a shows a projection of this dataset using only blue and near infrared bands. Fig. 2b depicts in graphical format the mean value of the four spectral bands for each of the seven groups of land cover categories.

Two GCS networks have been trained with each data set: without and with removal of neurons. For the two GCS networks without removal of units, the ending of the training phase has occurred when a predetermined number of neurons in the output layer of the network has been achieved (100 units for the two-dimensional dataset and 200 for the multispectral one). On the other hand, obtaining a predetermined minimum number of

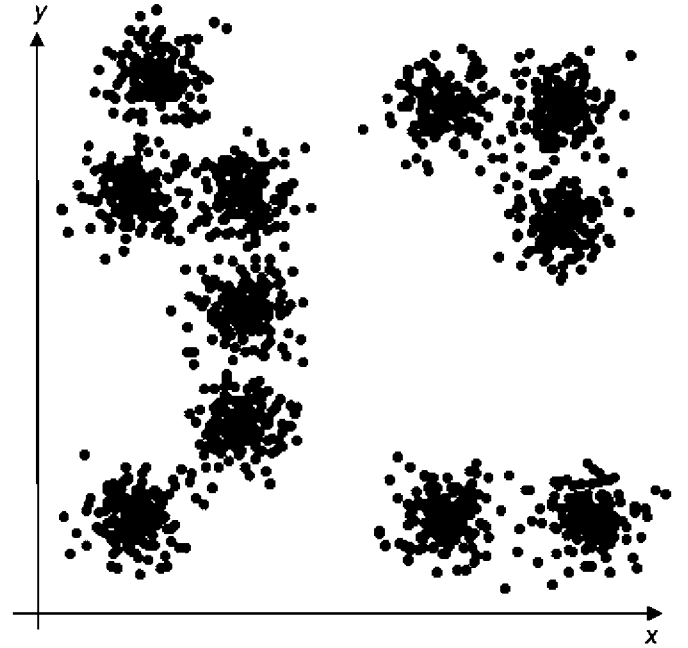


Fig. 1. Eleven separate regions in the 2-dimensional plane. Input data visualization.

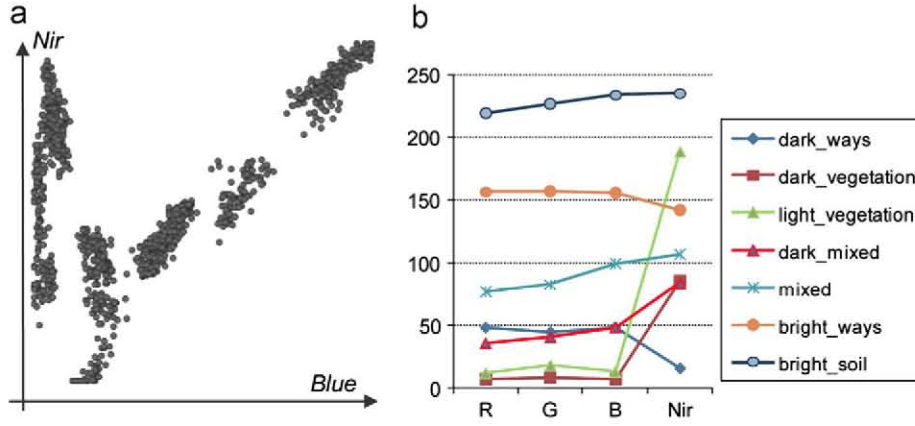
isolated clusters of neurons in the output layer has been the ending condition of the training phase for the two networks with removal of units (eleven for the two-dimensional dataset, and seven for the multispectral one).

In GCS networks without removal of units, the topology preserving map can be analyzed jointly with some of the topographic maps that make available the visualization of clusters in the dataset. In this work the well known U-map proposed for Kohonen networks [4–5], adapted to the GCS model [19] has been used. U-map shows an overview of the potential number of clusters present in the input space, and provides the necessary information to determine those connections on the topology preserving map, which correspond to topology violations in the network.

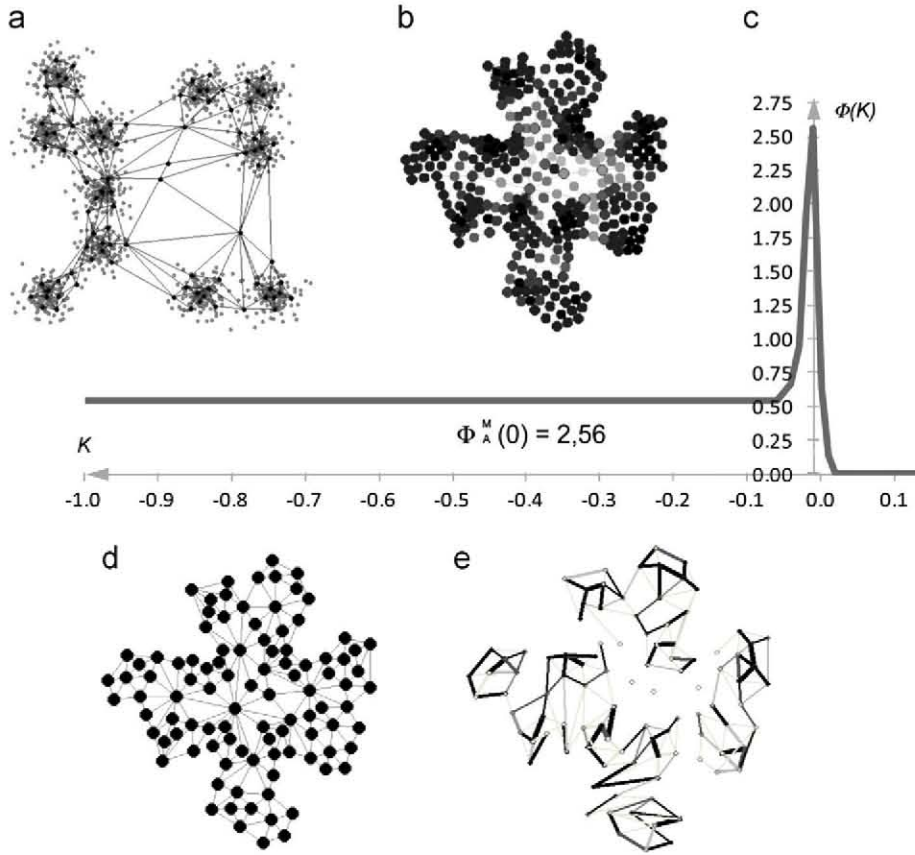
In all the experiments, the modification of the training algorithm for GCS network described in [20] has been used, in order to achieve a physical interpretation of the learning parameters related to the removal of output units. In addition, given that the two networks trained with the same dataset have different number of output neurons, with the aim of comparing both topographic functions, a normalization of X-axis coordinates has been carried out in these graphs, transforming values in the range  $(-N \dots N)$  to the range  $(-1 \dots 1)$ . The definition, description and values of the GCS training parameters used in all experiments are detailed in [19]. For the two GCS networks without removal of units, a neuron removal factor  $\mu=0$  (relative normalized probability density estimation value) has been used, and  $\mu=0.0006$  for the two networks trained with the removal of units.

##### 4.1. Two-dimensional data set

Using this data set, a GCS network without removal of units has been trained until 100 neurons have been obtained in the output layer. Given that this is a two-dimensional input space, the input vectors and the output neurons have been projected directly on the plane. Fig. 3a shows the input vectors (small light gray dots), the output neurons (small black dots) and the neighbor connections (lines connecting neurons). In general, neurons in the output layer have been distributed around the eleven clusters of data on the input space. However, there are



**Fig. 2.** QuickBird multispectral data set. (a) Input data visualization using the blue and near infrared spectral bands and (b) spectral signature mean values for the seven land cover categories present in data.

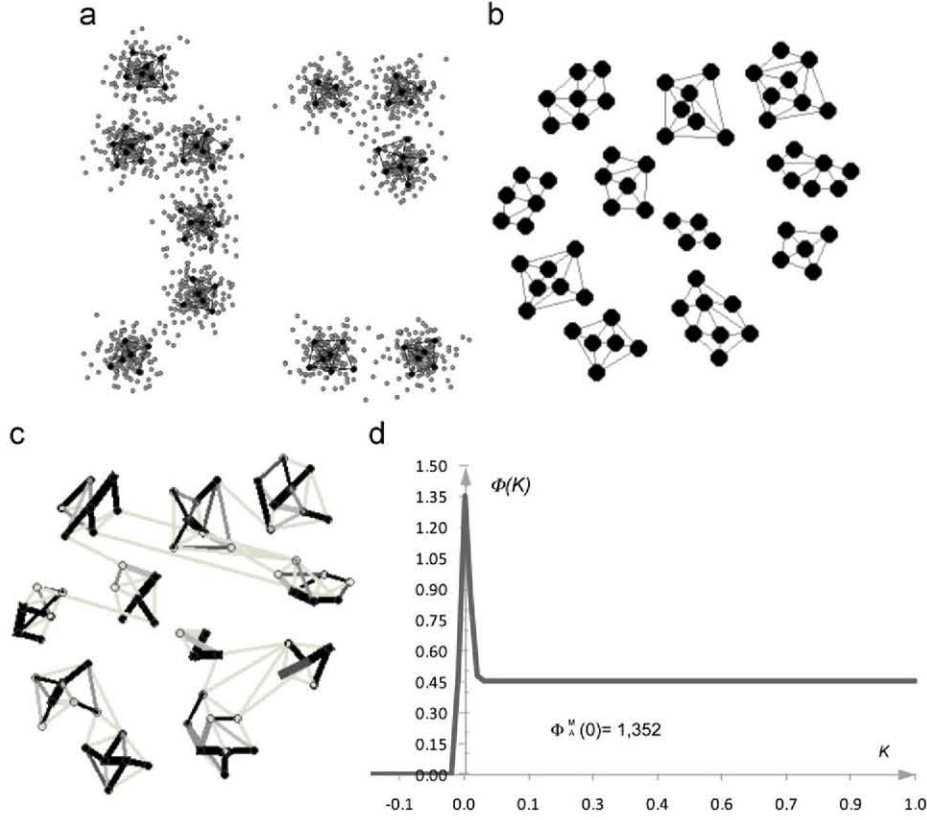


**Fig. 3.** Eleven separate regions in the 2-dimensional plane. GCS network without removal of units. (a) Visualization of the input data (light gray dots) and the network projection (black dots and neighbor connection lines) using the two component values of the synaptic vectors, (b) U-map, (c) topographic function and topology preserving value for  $k=0$ , (d) topographic map and (e) topology preserving map.

connections between neurons that represent different clusters of data as well as neurons placed in areas of the input space without any vector. The grouping of neurons around the clusters of data is validated by the U-map of the network (Fig. 3b). It clearly exhibits the presence of eleven clusters (the groups of neurons in dark tones). This network consists of a single mesh of neurons in the output layer, so the information displayed by the U-map advances the low adaptation quality of the network to the input space. The topographic function (Fig. 3c) reveals that a small number of units should be direct neighbors but have a slightly higher distance in  $A$  (it quickly falls to 0 for positive values of  $k$ ), whereas the asymptote that emerges for negative values of  $k$  indicates that

there are several units in  $A$  that should not belong to the same neighborhood mesh (those that do not identify any pattern and those that identify patterns belonging to different input clusters). In this network there exists an average of about 0.5 neighbors per neuron that should not be connected directly or indirectly, i.e., they should not belong to the same neighborhood mesh. The topology preserving map has been generated (Fig. 3e) for evaluating the relevance of the violations identified by the topographic function. On comparing the topology preserving map with the topographic map (Fig. 3d) it is illustrated that there exist neighborhood connections in  $A$  that should not exist (those that do not exist in the topology preserving map). Moreover,





**Fig. 4.** Eleven separate regions in the 2-dimensional plane. GCS network with removal of units (11 isolated clusters). (a) Visualization of the input data (gray dots) and the network projection (black dots and neighbor connection lines) using the two component values of the synaptic vectors, (b) topographic map, (c) topology preserving map and (d) topographic function and topology preserving value for  $k=0$ .

comparing the topology preserving map with the U-map, some strong dark lines connecting units that belong to separate clusters of neurons indicating serious topology preserving violations can be located. On this basis, it can be concluded that the transgressions observed in the topographic function for negative values of  $k$  correspond to serious violations of the topology preservation, so this network would not be an accurate model of the input space.

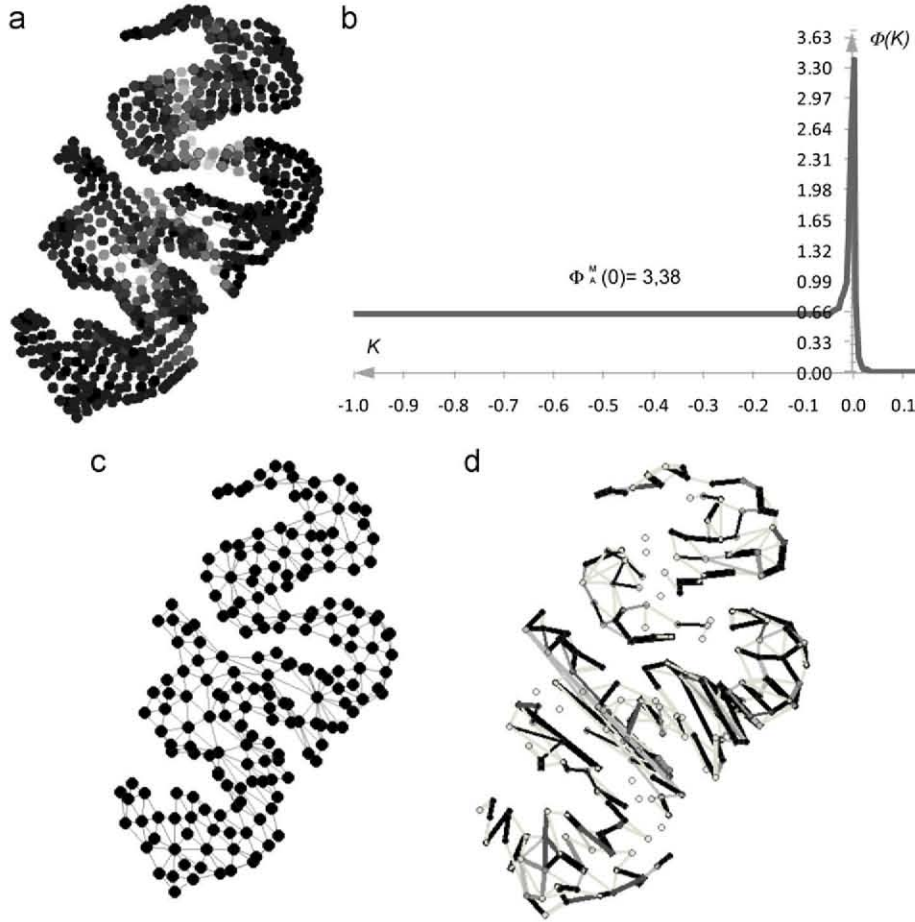
Using the same dataset, a GCS network with removal of units has been trained, using as ending adaptation condition to obtain at least eleven isolated clusters of neurons in the output layer. Finally, the trained network has 71 neurons distributed in eleven separate clusters. Fig. 4a shows the 2D training vectors represented by small light gray dots, the output neurons by small black dots and the neighbor connections by lines connecting neurons. This figure shows that this network is better suited to the input space than the network trained without removal of units. The topographic function (Fig. 4d) reveals that there are very few direct neighbors that should be to a slightly higher distance (it quickly falls to 0 for negative values of  $k$ ), suggesting that internally each cluster of neurons is well suited to the corresponding input subspace. Furthermore, the asymptote that emerges for positive values of  $k$  indicates that there exists an average of 0.45 neighbors per neuron that should be connected in  $A$ , but theoretically they distance  $\infty$ , i.e., there exist neurons belonging to separate clusters that should be connected according to intrinsic information of the input space. Analyzing the topology preserving map (Fig. 4c) it can be observed that there are only a few fine and clear lines (slight topology violations) connecting units that belong to a different cluster (Fig. 4b), suggesting that the transgressions identified by the topographic function are produced by a small number of input vectors. Therefore, this GCS network could be considered a good accurate model of the input space.

Evaluating the value of the topographic function ( $k=0$ ) obtained with both GCS networks shows the improvement in the network of eleven clusters in contrast with the network of 100 neurons. In addition, analysis of the relevance of violations using the topology preserving map has shown that the network of eleven clusters is better suited to the topology of the input space than the network of 100 neurons.

#### 4.2. QuickBird multispectral data set

When a GCS network is trained using a two-dimensional or three-dimensional input space, the information provided by the training vectors and the network can be projected in a two-dimensional or tree-dimensional graph (as corresponds). This map can be used to visually analyze the quality of adaptation of the network to the input space, as discussed in the previous subsection in Figs. 3a and 4a. However, when the input space consist of vectors with more than three components, it is not possible to generate a graph that includes all the information of  $M$  and  $A$ . There exists the possibility of projecting  $M$  and  $A$  using only two or three components, but in many cases, the resulting graph does not display correctly the data or the network. For this reason, it is necessary to use other techniques to facilitate the analysis of the input space and the adaptation quality of the network. This is the case of the real-world dataset discussed in this subsection. It is a four-dimensional input space consisting of 1460 pixels extracted from a multispectral image captured by the multispectral optical sensor onboard the QuickBird satellite. As was stated at the beginning of this section, this data set has been constructed selecting characteristic pixels of seven land cover categories (Fig. 2).

Using this input space, a GCS network has been trained without removal of units, until obtaining 200 neurons in the



**Fig. 5.** QuickBird multispectral data set. GCS network without removal of units. (a) U-map, (b) topographic function and topology preserving value for  $k=0$ , (c) topographic map and (d) topology preserving map.

output layer. To analyze the adaptation quality of the trained network, U-map (Fig. 5a), topographic function (Fig. 5b) and topology preserving map (Fig. 5d) have been generated. U-map displays the presence of several clusters (the groups of neurons in dark tones), indicating that this network has a poor degree of topology preservation since it is composed by a single mesh of neurons. This information is confirmed by the asymptote of the topographic function for negative values of  $k$  (in  $A$  there exists an average of about 0.5 neighbors per neuron that should not be connected directly or indirectly, i.e., they should not belong to the same neighborhood mesh). The topographic function falls quickly for positive values of  $k$ , indicating that a small number of units should exhibit a direct neighborhood connection but present a slightly higher distance in  $A$ . Comparing the topology preserving map (Fig. 5d), which represents the neighborhood connections that should exist in the network according to  $M$ , with the topographic map (Fig. 5c), which shows the real neighborhood connections in  $A$ , the presence of a high volume of neighboring neurons in  $A$  that should not belong to the same mesh can be observed, indicating serious topology preserving violations. As result of this analysis, it can be concluded that this network has a low topology preserving grade.

Using the same dataset, a GCS network with removal of units has been trained, until at least seven isolated clusters of neurons have been obtained. The trained network has 61 neurons distributed in seven separate clusters, as shown in the topographic map (Fig. 6a). The topographic function (Fig. 6c) shows that each cluster of neurons fits well with the cluster of data that it identifies (it quickly falls to 0 for negative values of  $k$ ). On the

other hand, the asymptote that emerges for positive values of  $k$  indicates that there are some neurons in  $A$  belonging to separate clusters that should be directly connected. The topology preserving map (Fig. 6b) shows the lightness of these violations, as they are caused by a tiny number of input vectors (only three thin and light gray lines appear connecting units that belong to separate clusters).

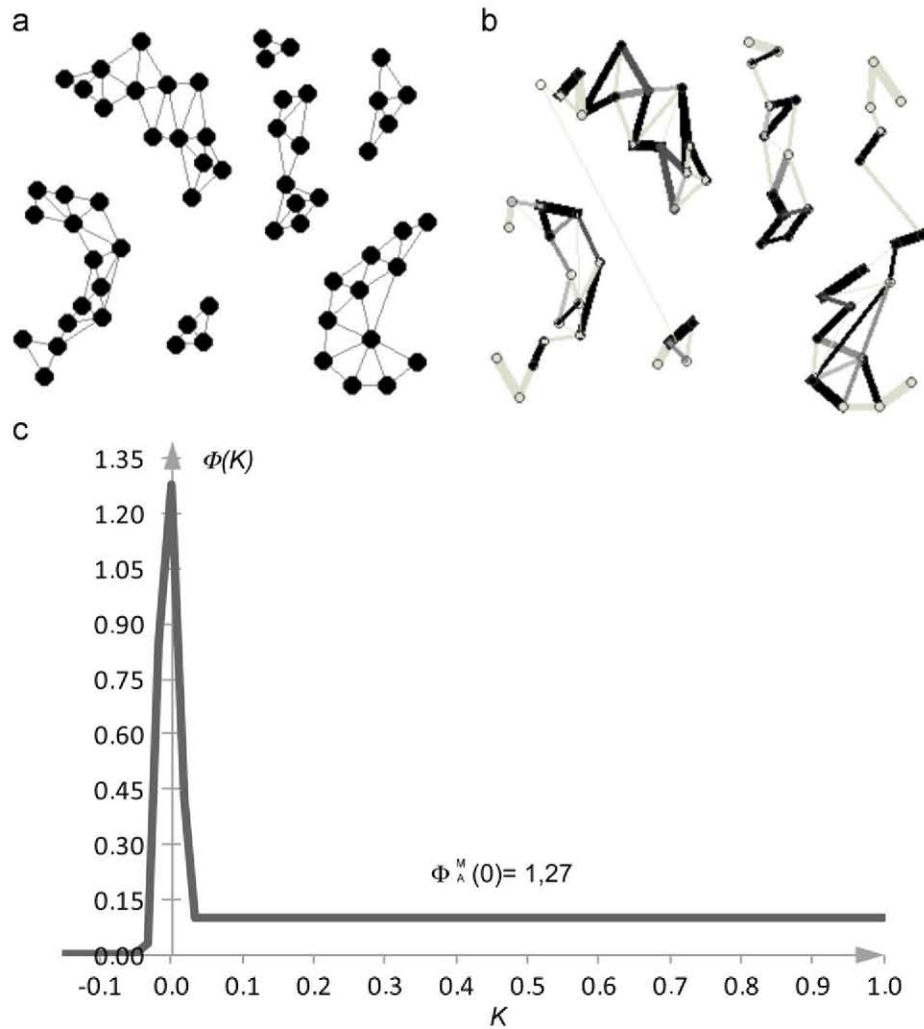
Quality results obtained with the GCS network trained with removal of units are clearly better than those achieved with the network formed by a single mesh of neurons. The topographic function value in  $k=0$  for the network of seven clusters significantly improves the value obtained for the network of 200 neurons. Furthermore, analyzing the topology preserving maps, the relevance of violations detected by the topographic functions has been qualified concluding that GCS network of seven clusters is better suited to the topology of the input space than the network of 200 neurons.

## 5. Conclusions

In this paper, a new technique based on the integration of information provided by two methods for quantifying and qualifying the topology preservation of the GCS network model has been proposed. Both measures used in conjunction provide a powerful validation tool for a quick evaluation of the trained network quality.

The topographic function offers a clear view of the quality of the trained network. For positive values of  $k$  it displays the mean





**Fig. 6.** QuickBird multispectral data set. GCS network with removal of units (7 isolated clusters). (a) Topographic map, (b) topology preserving map and (c) topographic function and topology preserving value for  $k=0$ .

number of neighboring units per neuron that should exhibit a direct connection in  $A$ , but present a distance greater than  $k$  in the network. For negative values of  $k$  it shows the mean number of neighboring units per neuron that should not be directly connected.

As stated throughout the paper, the topographic function detects and quantifies the topology preserving violations, but it does not reflect the seriousness of the transgressions. On the other hand, the topology preserving map used along with the topographic map shows clearly those neurons that are generating topology violations, as well as the relevance of the same, complementing the restrictions of the topographic function.

The two datasets used in the experiments have shown that when the input space consists of vectors grouped in separate clusters, GCS networks usually present better topology preservation value when removal of units is used, since it eliminates those neurons located in areas of the input space with low probability density. This is one of the most important advantages supplied by the GCS model compared with that of Kohonen's.

The volume of violations provided by the topographic function along with the detailed evaluation of each output neuron displayed in the topology preserving map makes both graphs present a complete view of the quality of a GCS trained network with regard to the topology preservation.

From the results obtained in this study, the automation of the ending condition of the training in a GCS network is planned as

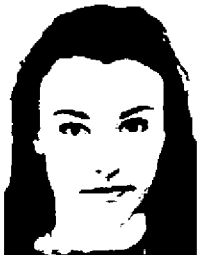
the future line of work. Based on the information provided by both measures of topology preservation, the adaptation process will terminate when an acceptable grade of quality is achieved in the network. This will simplify the selection of the best number of clusters of neurons that will result in the GCS network that best fits to the input space.

## References

- [1] V. Cherkassky, Filip Mulier, *Learning from Data: Concepts, Theory and Methods*, John Wiley & Sons, Inc, 1998.
- [2] J.W. Tukey, *Exploratory Data Analysis*, Addison-Wesley, Reading, MA, 1977.
- [3] T. Kohonen, *Self-Organizing Maps*, third ed., Springer, Berlin Heidelberg New York, 2001.
- [4] A. Ultsch, H.P. Siemon, Kohonen self-organizing feature maps for exploratory data analysis, in: *Proceedings of the International Neural Network*, Dordrecht, The Netherlands, 1990.
- [5] M.A. Kraaijveld, J. Mao, A.K. Jain, A non linear projection method based on Kohonen's topology preserving maps, *IEEE Transactions on Neural Networks* 6 (3) (1995) 548–559.
- [6] D. Merlk, A. Rauber, Alternative ways for cluster visualization in self-organizing maps, in: *Proceedings of the Workshop on Self-Organizing Maps (WSOM'97)*, Espoo, Finland, 1997, pp. 106–111.
- [7] M. Rubio, V. Giménez, New methods for self-organizing map visual analysis, *Neural Computation and Applications* 12 (2003) 142–152.
- [8] T. Kohonen, Self-organized formation of topologically correct feature maps, *Biological Cybernetics* 43 (1982) 59–69.
- [9] T. Martinez, K. Schulten, Topology representing networks, *Neural Networks* 7 (3) (1994) 507–522.



- [10] K. Kiviluoto, Topology preservation in self-organizing maps, in: Proceedings of the International Conference on Neural Networks (ICNN'96), vol. 1, 1996, pp. 294–299.
- [11] H. Bauer, K. Pawelzik, Quantifying the neighborhood preservation of self-organizing feature maps, *IEEE Transactions on Neural Networks* 3 (1992) 570–579.
- [12] T. Villmann, R. Der, M. Herrmann, T. Martinetz, Topology preservation in self-organizing feature maps: exact definition and measurement, *IEEE Transactions on Neural Networks* 8 (2) (1997) 256–266.
- [13] K. Tasdemir, E. Merényi, Data topology visualization for the self-organizing map, in: Proceedings of the European Symposium on Artificial Neural Networks (ESANN'06), Bruges, Belgium, 2006, pp. 277–282.
- [14] J. Blackmore, K. Miikkulainen, Incremental grid growing: encoding high-dimensional structure into a two-dimensional feature map, in: Proceedings of the IEEE International Conference on Neural Networks (ICNN-93), 1993, pp. 450–455.
- [15] H. Bauer, T. Villmann, Growing a hypercubical output space in a self-organizing feature map, Technical Report TR-95-030, ICSI, Berkeley, 1995.
- [16] A. Hsu, I. Saeed, S. Halgamuge, Dynamic self-organizing maps: theory, methods and applications, *Studies in Computational Intelligence* 1 (2009) 363–379.
- [17] B. Fritzke, A growing neural gas network learns topologies, in: *Advances in Neural Information Processing Systems*, vol. 7 (NIPS'94), MIT Press, Cambridge, 1995, pp. 625–632.
- [18] B. Fritzke, Growing cell structures—a self-organizing network for unsupervised and supervised learning, *Neural Networks* 7 (9) (1994) 1441–1460.
- [19] S. Delgado, C. Gonzalo, E. Martínez, A. Arquero, Visualizing high-dimensional input data with growing self-organizing maps, *Lecture Notes in Computer Science* 4507 (2007) 580–587.
- [20] S. Delgado, C. Gonzalo, E. Martínez, A. Arquero, Improvement of self-organizing maps with growing capability for goodness evaluation of multi-spectral training patterns, geoscience and remote sensing symposium (IGARSS'04), *IEEE International* 1 (2004) 564–567.



**Soledad Delgado** received the B.S Degree in computer science from the Technical University of Madrid, Spain, in 1990 and the M.S Degree in computer science from the Carlos III University of Madrid, Spain, in 2000, and the Ph.D. Degree in computer science from the Technical University of Madrid, Spain, in 2010.

She is currently Assistant Professor of Department of Organization and Structure of Information, Technical University of Madrid, Spain. Her research interests include neural networks, pattern recognition, exploratory data analysis, image classification and remote sensing.



**Consuelo Gonzalo** received the Sciences degree from the University of Salamanca, in 1986, and the Sciences Ph.D. Degree from the Complutense University of Madrid (Spain), in 1989.

She is currently an Associate Professor of Department of Architecture and Technology of Computer Systems, at the Computer School (Technical University of Madrid, Madrid, Spain). Her main research interests are in the area of Remote Sensing and include the fields of Fusion Images and Artificial Neural Networks for the processing, analysis and interpretation of remotely sensed data. Moreover she is working on the parallel computation of this kind of algorithms.



**Estibaliz Martínez** received the Ph.D. in Chemistry Sciences from the Complutense University, Madrid, Spain, in the July of 1989.

She is currently an Associate Professor of Department of Architecture and Technology of Computer Systems, Computer School, Polytechnical University of Madrid, Madrid, Spain. She teaches courses at the University in Digital Design, VLSI Technology and Remote Sensing. Her research interests concern the area of the Remote Sensing Image Processing and include the use of Neural Networks Algorithms, Genetic Algorithms and Fuzzy Logic for the analysis and interpretation of remotely sensed data. In addition,

her research areas are in the applications of remote sensed data management and their access through client-server architectures.



**Agueda Arquero** received the Ph.D. Degree in Chemistry Sciences from Autónoma University of Madrid (Spain) in the January of 1983.

She is currently an Associate Professor of Department of Architecture and Technology of Computer Systems, Computer School, Polytechnical University of Madrid, Madrid, Spain. Her current research interests in remote sensing image processing area include field targets radiometric measurements and later processing, analysis of remotely sensed data using Neural Networks Algorithms, Genetic Algorithms and Fuzzy Logic. Other research interests are synthesis and spectral characterization of new materials. She is AET's member.