

# The Growing Hierarchical Neural Gas Self-Organizing Neural Network

Esteban J. Palomo and Ezequiel López-Rubio

**Abstract**—The growing neural gas (GNG) self-organizing neural network stands as one of the most successful examples of unsupervised learning of a graph of processing units. Despite its success, little attention has been devoted to its extension to a hierarchical model, unlike other models such as the self-organizing map, which has many hierarchical versions. Here, a hierarchical GNG is presented, which is designed to learn a tree of graphs. Moreover, the original GNG algorithm is improved by a distinction between a growth phase where more units are added until no significant improvement in the quantization error is obtained, and a convergence phase where no unit creation is allowed. This means that a principled mechanism is established to control the growth of the structure. Experiments are reported, which demonstrate the self-organization and hierarchy learning abilities of our approach and its performance for vector quantization applications.

**Index Terms**—Hierarchical model, self-organization, unsupervised learning, vector quantization.

## I. INTRODUCTION

C LUSTER analysis is the organization of a collection of patterns (usually represented as a vector of measurements or a point in a multidimensional space) into clusters based on similarity, where patterns within a valid cluster are more similar to each other than they are to patterns belonging to a different cluster [1]. Clustering techniques are understood as unsupervised learning methods, where clusters are data driven, i.e., they are obtained uniquely from the data. They have been researched across multiple disciplines over several decades [2].

Manuscript received December 24, 2014; revised August 2, 2015 and January 31, 2016; accepted May 15, 2016. Date of publication June 2, 2016; date of current version August 15, 2017. This work was supported in part by the Ministry of Economy and Competitiveness of Spain within the European Regional Development Fund (ERDF) through the Project entitled Video Surveillance by Active Search of Anomalous Events under Grant TIN2014-53465-R, in part by the Autonomous Government of Andalusia, Spain, within ERDF through the Project entitled Development of Self-Organizing Neural Networks for Information Technologies under Grant TIC-6213, in part by ERDF through the Project entitled Eco-Neighborhoods Versus District Rehabilitation under Grant PI-57101, and in part by ERDF through the Project entitled Self-Organizing Systems and Robust Estimators for Video Surveillance under Grant TIC-657.

E. J. Palomo is with the Department of Computer Languages and Computer Science, University of Málaga, Málaga 29071, Spain, and also with the School of Mathematical Sciences and Information Technology, Yachay University, Urcuquí, Ecuador (e-mail: ejpalomo@lcc.uma.es).

E. López-Rubio is with the Department of Computer Languages and Computer Science, University of Málaga, Málaga 29071, Spain (e-mail: ezeqlr@lcc.uma.es).

This paper has supplementary downloadable material available at <http://ieeexplore.ieee.org/>, provided by the author.

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TNNLS.2016.2570124

Self-organizing maps (SOMs) have been widely used for data clustering, since they perform a mapping between high-dimensional data and a lower dimensional representation space, preserving the topology of input data. Many self-organizing neural networks are based on a fixed lattice topology among the neurons, after the pioneering SOM [3], [4]. This strategy is best suited to those applications where a computational map is required. However, if the aim is to learn the topology of the input distribution so as to obtain a closer adaptation, a fixed topology cannot be used. This calls for self-organizing neural models that learn topologies. Among them, the growing neural gas (GNG) has become a standard with myriad applications to complex data set modeling [5], [6], computer vision [7], [8], and robotics [9], [10]. It learns a dynamic graph with variable numbers of neurons and connections. This graph represents the input data in a more plastic and flexible way than a fixed-topology map, improving visualization capabilities and understanding of data.

One of the key features of SOMs is their ability to discover the structure of the input data set. This makes them particularly suitable to process data with a temporal structure [11]. More general complex discrete structures, such as sequences and trees, can also be processed by SOMs. This is possible because SOM algorithms can be adapted to use a distance metric among those discrete structures [12].

Another successful approach to self-organization is that of hierarchical models. They exploit the hierarchical structure present in many practical data sets, so that a tree of SOMs is learned. The growing hierarchical SOM (GHSOM) is one of the most popular models of this kind [13], although there are a wide variety of proposed hierarchical approaches [14]–[17]. It also has many applications to computer network traffic analysis [18], [19], finance [20], [21], manufacturing [22], and image analysis [23]. In general terms, all clustering methods that learn a hierarchical structure have advantages even when used for nonhierarchical data. The learned hierarchical structure can be pruned at several levels, which yields alternative representations of the input data set at different levels of detail. This can be used to visualize a data set in coarser or more detailed way. For vector quantization applications, the different pruning levels correspond to smaller or larger codebooks, so that a balance can be attained between the size of the codebook and the quantization error within the same hierarchical structure.

The possibility of considering hierarchical versions of the GNG has rarely been explored. The hierarchical GNG (HGNG) [24] has a bottom layer, which consists of a

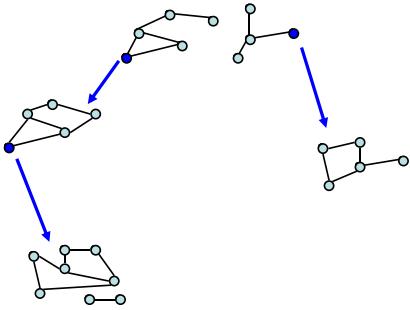


Fig. 1. Structure of a GHNG model with four graphs. The parent units are shown in a darker tone. Please note that the root graph and the graph in the lower left corner are not connected.

single GNG that learns from the input data. Then, each upper layer contains a single GNG, which learns from the prototypes of the lower layer. Consequently, a stack of GNGs rather than a tree is built, and the higher GNGs do not learn from the input data directly. This puts the HGNG model far away from the GHSOM, where a tree of SOMs is learned and all the SOMs learn from the input data. The TreeGNG [25] is aimed to study the connected subgraphs, which arise in the learning process of a standard GNG, so that a tree of connected subgraphs is built. Consequently, the original learning algorithm of the GNG is not modified, and no tree of GNGs is built.

Here, our goal is to propose a hierarchical self-organizing neural model, which learns a tree of GNGs, in the same way that the GHSOM is a hierarchical extension of the SOM. Hence, our aim is to analyze the data sets with a hierarchical structure without the limitation of the fixed lattice topology of the GHSOM. In addition to this, a principled procedure to control the growth of the individual GNGs is developed, so as to control the expansion of the tree.

The structure of this paper is as follows. Section II defines the proposed model, whose properties are studied in Section III. Experimental results are reported in Section IV. Finally, Sections V and VI are devoted to discussion and conclusions, respectively.

## II. MODEL

A growing hierarchical neural gas (GHNG) model is defined as a tree of self-organizing graphs. Each graph is made of a variable number of neurons or processing units, so that its size can grow or shrink during learning. In addition, each graph is the child of a unit in the upper level, except for the top level (root) graph. The definition of the model is divided in two sections. First, we present the basic model for a graph and the corresponding learning algorithm (Section II-A). Then, we explain how new graphs are created to yield a hierarchy of graphs (Section II-B). The overall structure of a GHNG model is exemplified in Fig. 1. It is worth noting that the graphs need not be connected, as shown.

### A. Graph Model

Each graph of the GHNG is made of  $H$  units ( $H \geq 2$ ) and one or more undirected connections among them. Both the units and the connections can be created and destroyed

during the learning process. It is not necessary that the graph is connected, as mentioned earlier. The training set for the graph will be noted  $\mathcal{S}$ , with  $\mathcal{S} \subset \mathbb{R}^D$ , where  $D$  is the dimension of the input space. Each unit  $i \in \{1, \dots, H\}$  has an associated prototype  $\mathbf{w}_i \in \mathbb{R}^D$  and an error variable  $e_i \in \mathbb{R}$ ,  $e_i \geq 0$ . Each connection has an associated age, which is a nonnegative integer. The set of connections will be noted as  $A \subseteq \{1, \dots, H\} \times \{1, \dots, H\}$ .

The learning mechanism for a graph of the GHNG is based on the original GNG [26], but it includes a novel procedure to control the growth of the graph. First, a growth phase is performed where the graph is allowed to enlarge, until a condition is fulfilled, which indicates that further growing would provide no significant improvements in the quantization error. After that, a convergence phase is executed where no unit creation is allowed in order to carry out a fine tuning of the graph. The learning algorithm is given by the following steps.

- 1) Start with two units ( $H = 2$ ) joined by a connection. Each prototype is initialized to a sample drawn at random from  $\mathcal{S}$ . The error variables are initialized to zero. The age of the connections is initialized to zero, too.
- 2) Draw a training sample  $\mathbf{x}_t \in \mathbb{R}^D$  at random from  $\mathcal{S}$ .
- 3) Find the nearest unit  $q$  and the second nearest unit  $s$  in terms of Euclidean distance

$$q = \arg \min_{i \in \{1, \dots, H\}} \|\mathbf{w}_i(t) - \mathbf{x}_t\| \quad (1)$$

$$s = \arg \min_{i \in \{1, \dots, H\} - \{q\}} \|\mathbf{w}_i(t) - \mathbf{x}_t\|. \quad (2)$$

- 4) Increment the age of all edges departing from  $q$ .
- 5) Add the squared Euclidean distance between  $\mathbf{x}_t$  and the nearest unit  $q$  to the error variable  $e_q$
- 6) Update  $q$  and all its direct topological neighbors with step size  $\epsilon_b$  for unit  $q$  and  $\epsilon_n$  for the neighbors, where  $\epsilon_b > \epsilon_n$

$$\epsilon(t, i) = \begin{cases} \epsilon_b, & \text{iff } i == q \\ \epsilon_n, & \text{iff } (i \neq q) \wedge (i, q) \in A \\ 0, & \text{iff } (i \neq q) \wedge (i, q) \notin A \end{cases} \quad (4)$$

$$\mathbf{w}_i(t+1) = (1 - \epsilon(t, i))\mathbf{w}_i(t) + \epsilon(t, i)\mathbf{x}_t. \quad (5)$$

- 7) If  $q$  and  $s$  are connected by an edge, then set the age of this edge to zero. Otherwise, create it.
- 8) Remove edges with an age larger than  $a_{\max}$ . Then, remove all units that have no outgoing edges.
- 9) If the current time step  $t$  is an integer multiple of a parameter  $\lambda$  and the graph is in the growth phase, then make a backup copy of the full graph and insert a new unit as follows. First, determine the unit  $r$  with the maximum error and the unit  $z$  with the largest error among all direct neighbors of  $r$ . Then, create a new unit  $k$ , insert edges connecting  $k$  with  $r$  and  $z$ , and remove the original edge between  $r$  and  $z$ . After that, decrease the error variables  $e_r$  and  $e_z$  by multiplying them with a constant  $\alpha$ , and initialize the error variable  $e_k$

to the new value of  $e_r$ . Finally, setup the prototype of  $k$  to be halfway between those of  $r$  and  $z$ , as follows:

$$\mathbf{w}_k(t) = \frac{1}{2}(\mathbf{w}_r(t) + \mathbf{w}_z(t)). \quad (6)$$

- 10) If the graph is in the growth phase and the current time step  $t$  satisfies

$$\text{mod}(t, 2\lambda) = \left\lfloor \frac{3}{2}\lambda \right\rfloor \quad (7)$$

where  $\lfloor \cdot \rfloor$  stands for rounding toward  $-\infty$ , then a check is done in order to see whether the graph growth has resulted in an improvement of the quantization error. The mean quantization error per neuron of the backup and current versions of the graph are computed as the sum of their error variables divided by their number of units  $H$ . Let  $\text{MQE}_{\text{old}}$  and  $\text{MQE}_{\text{new}}$  denote the mean quantization errors of the backup and current versions of the graph, respectively. **If the following condition holds, then the current version is destroyed, the backup copy is restored, and the graph enters the convergence phase:**

$$\frac{\text{MQE}_{\text{old}} - \text{MQE}_{\text{new}}}{\text{MQE}_{\text{old}}} < \tau \quad (8)$$

where  $\tau \in [0, 1]$  is a parameter, which controls the growth process. As shown in (8), the  $\tau$  parameter establishes how big the reduction of the mean quantization error must be in order to accept the insertion of an additional unit to an individual GNG. Thus, the higher  $\tau$ , the more significant the improvement in the quantization error must be in order to continue the growth phase. Hence, the higher values of  $\tau$  are associated with smaller graphs, and vice versa.

- 11) Decrease all error variables  $e_i$  by multiplying them by a constant  $d$ .  
 12) If the maximum number of time steps has been reached, then stop. Otherwise, go to step 2.

In order to select the parameters of the graph learning algorithm detailed above, the recommended values of the original GNG have been employed in the experiments [26]:  $\lambda = 100$ ,  $\epsilon_b = 0.2$ ,  $\epsilon_n = 0.006$ ,  $\alpha = 0.5$ ,  $a_{\max} = 50$ , and  $d = 0.995$ .

### B. Hierarchical Model

As mentioned before, the GHNG is defined as a tree of graphs. The procedure to learn such hierarchy is detailed in the following. The process starts by training the root graph with the overall set of training samples. Each time that a graph must be trained with training set  $\mathcal{S}$ , this is done according to the algorithm specified in Section II-A. If the resulting number of units is  $H = 2$ , then the graph is pruned, because it is too small to represent any important features of the input distribution. Otherwise, a new graph is created for each unit  $i$  and the training process is invoked recursively with the receptive field of unit  $i$  as the training set

$$\mathcal{S}_i = \left\{ \mathbf{x} \in \mathcal{S} \mid i = \arg \min_{j \in \{1, \dots, H\}} \|\mathbf{w}_j - \mathbf{x}\| \right\}. \quad (9)$$

---

**Algorithm 1** Train a GHNG Model
 

---

**Input:** 1)  $\mathcal{S}$  (training set)  
 2)  $\tau$  (growth control parameter)  
 3)  $\lambda, \epsilon, \epsilon_n, \alpha, a_{\max}, d$  (GNG parameters)  
 4) *level* (current level)  
**Output:** A GHNG model  
 1: **if** *level* > *MAX\_LEVEL* **then**  
 2:   **return**  
 3: **end if**  
 4: Train the modified GNG as given in Subsection II-A  
 5: **if**  $H == 2$  **then**  
 6:   **return**  
 7: **end if**  
 8: **for**  $i=1:H$  **do**  
 9:   Invoke recursively this algorithm with the receptive field  
     of unit  $i$  as the training set  $\mathcal{S}_i$  (Eq. 9) and *level* + 1  
10: **end for**

---

This recursive process continues until a prespecified number of levels are reached. The elimination of the graphs with fewer than three units and the split of the training set given by (9) work together in order to attain a parsimonious hierarchy, i.e., one with a reduced number of graphs and units. This is because lower graphs in the tree cannot have many units because their training sets are smaller. It is worth noting that many of the created graphs will eventually be pruned right after their training, so that the fact that a graph is created for each unit does not lead to uncontrolled growth. The pseudocode of our proposal is described in Algorithm 1.

### III. PROPERTIES

Here, some important properties of the model proposed in Section II are proved. The initialization and the learning rule for the graphs of the GHNG model (Section II-B) are invariant [27], [28] with respect to homogeneous scalings and rigid transformations on the input data. Hence, a graph is not affected by homogeneous scalings, translations, reflections, and rotations. This is a very desirable feature of any multidimensional data analysis method [29]–[31], and in particular of self-organizing neural networks [17]. This property is also fulfilled by the overall GHNG trees. In a color quantization application like that of Section IV-B, the properties mean that it does not matter whether the input color data are provided in the  $[0, 1]$  or the  $[0, 255]$  ranges. For a computer vision application where 3-D points are provided as input data, the properties mean that it does not matter if the world axes are rotated or the unit of length is changed.

First of all, the kind of transformations that will be studied is established (Definition 1). Then, the equivariance property for the graphs is proved (Theorem 2). Finally, the equivariance property for the trees is also derived (Theorem 3).

**Definition 1:** We say that  $\xi$  is an angle preserving affine transformation on the input data if and only if the following conditions hold:

$$\xi : \mathbb{R}^D \rightarrow \mathbb{R}^D \quad (10)$$

$$\xi(\mathbf{x}) = \beta \mathbf{A} \mathbf{x} + \mathbf{b} \quad (11)$$

where  $\beta > 0$ ,  $\mathbf{b}$  is a  $D \times 1$  column vector, and  $\mathbf{A}$  is a  $D \times D$  orthogonal matrix

$$\mathbf{A}^{-1} = \mathbf{A}^T. \quad (12)$$

**Theorem 2:** Let  $\xi$  be an angle preserving affine transformation on the input data. Let  $G$  be a graph trained with the original data  $S$ , and  $G'$  a graph trained with the transformed data  $S'$ , and the same set of learning parameters. In addition, let us assume that the samples are drawn from the training sets in the same order. Then, the following conditions hold.

- 1) The number of units, the connections, and their age are the same for  $G$  and  $G'$ .
- 2) For every unit  $i$ , the prototypes and error variables of both models are related as follows:

$$\mathbf{w}'_i = \beta \mathbf{A} \mathbf{w}_i + \mathbf{b} \quad (13)$$

$$e'_i = \beta^2 e_i. \quad (14)$$

*Proof:* See Supplementary Material.  $\square$

**Theorem 3:** Let  $\xi$  be an angle preserving affine transformation on the input data. Let  $M$  be a GHNG trained with the original data  $S$ , and  $M'$  a GHNG trained with the transformed data  $S'$ , and the same set of learning parameters. In addition, let us assume that the samples are drawn from the training sets in the same order. Then, the following conditions hold.

- 1) The tree structure and the number of units, connections, and connection ages of each graph are the same for  $M$  and  $M'$ .
- 2) For every unit  $i$  of each graph, the prototypes and error variables of both models are related as follows:

$$\mathbf{w}'_i = \beta \mathbf{A} \mathbf{w}_i + \mathbf{b} \quad (15)$$

$$e'_i = \beta^2 e_i. \quad (16)$$

*Proof:* See Supplementary Material.  $\square$

#### IV. EXPERIMENTAL RESULTS

This section is devoted to report the results of the experiments that have been carried out in this paper to show the performance of the GHNG model compared with the GHSOM, GNG, and SOM proposals.<sup>1</sup> These models have been chosen for comparison purposes, since they are commonly used artificial neural networks based on self-organization like our proposal. Besides, the GHNG can be considered a hierarchical extension of the GNG, in the same way that the GHSOM is a hierarchical extension of the SOM. We have designed three different sets of experiments, which correspond to the following three sections. First, self-organization experiments have been carried out to demonstrate the self-organization capabilities of the GHNG (Section IV-A). Then, our proposal is evaluated for color image quantization (Section IV-B). Finally, in Section IV-C, the clustering of video sequences is examined, which represents a typical unsupervised learning task.

<sup>1</sup>The source code and some demos of our proposal are available online at <http://www.lcc.uma.es/%7Eejpalomo/downloads/ghng.zip>.

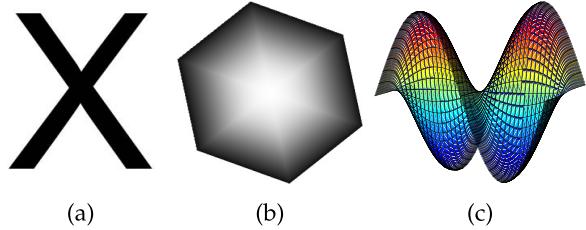


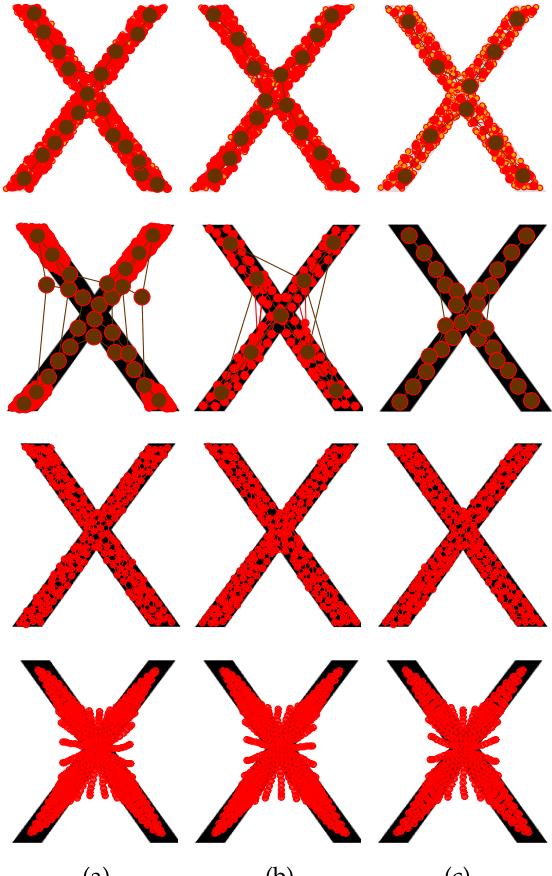
Fig. 2. Input distributions used for self-organization experiments. (a) 2-D X letter input distribution. (b) 2-D hexagon input distribution with different densities. (c) 3-D Twin Peaks input distribution.

#### A. Self-Organization Experiments

This first set of experiments is designed to show the self-organization capabilities of the GHNG model with respect to three input distributions representing different shapes in 2-D and 3-D. The first input distribution is a 2-D ( $D = 2$ ) input distribution with uniform density with the shape of an X letter. The second one is a 2-D ( $D = 2$ ) nonuniform input distribution with the shape of a hexagon, which features probability density gradients in its support. The last one is a 3-D ( $D = 3$ ) input distribution called Twin Peaks due to its shape. These input distributions are shown in Fig. 2.

The GHNG models were trained using  $M = 10\,000$  input samples and during  $N = 20\,000$  time steps for each input distribution. The maximum number of levels was set to three in order to ensure that the visualizations are not cluttered. Three different values of the  $\tau$  parameter (0, 0.1, and 0.2) were chosen to show the effect of  $\tau$  on the size of the obtained architecture, where the lower the value of  $\tau$ , the higher the number of neurons. In order to compare our results, we have also trained GHSOM, GNG, and SOM models using the same setup. For the GHSOM, the  $\tau_1$  parameter controls the size of the individual maps (the smaller  $\tau_1$ , the bigger the maps), and the  $\tau_2$  parameter controls the depth of the hierarchy (the smaller  $\tau_2$ , the deeper the hierarchy). In order to carry out a fair comparison, we have used different values of  $\tau_1$  and  $\tau_2$  to try to obtain a number of leaf neurons  $H$  as similar as possible to that obtained by the GHNG. We have considered the leaf neurons, and not all of the neurons, since only the leaf neurons of the GHNGs and GHSOMs are considered for the quantization error computation (the rest are just intermediate neurons). Similarly, the GNG has been trained using the same number of neurons  $H$  than the number of leaf neurons obtained by the GHNG, whereas the SOM was trained using the map size with the most similar number of neurons  $H$  to the GHNG.

The resulting GHNG, GHSOM, GNG, and SOM networks are shown in Figs. 3 and 4 in the first, second, third, and fourth rows, respectively. Please see Supplementary Material for the Twin Peaks input distribution results. In these plots, neurons are represented by circles, and connections among the mean vectors of the neurons are plotted as straight lines. For the hierarchical models (GHNG and GHSOM), neurons and connections are plotted in different colors depending on the layer of the hierarchy they belong to. Upper layers are painted darker. Thus, the hierarchical architecture can be better appreciated. However, in Fig. 3, the GHSOM and the SOM

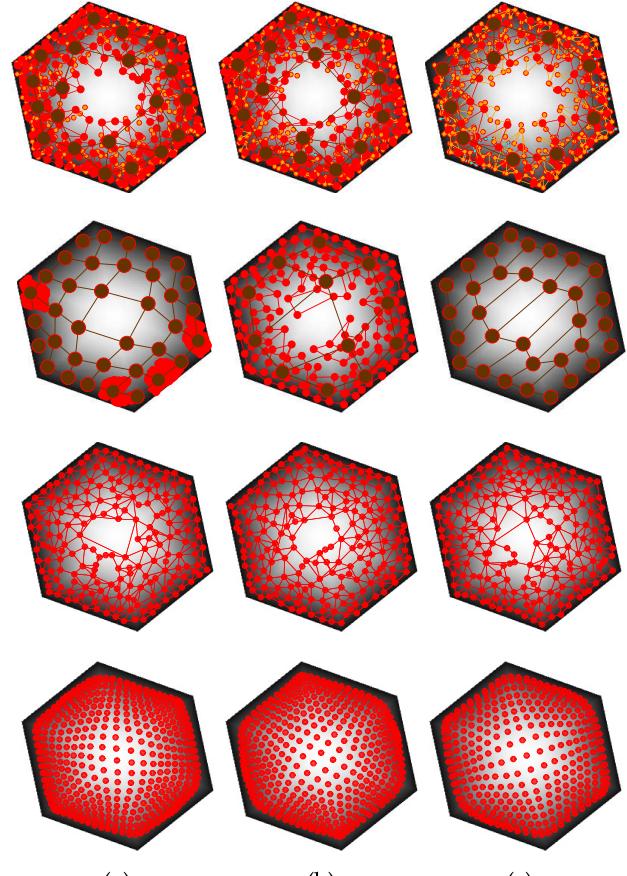


(a)

(b)

(c)

Fig. 3. GHNG (first row), GHSOM (second row), GNG (third row), and SOM (fourth row) results for the 2-D X letter input distribution. (a)  $\tau = 0$  ( $H = 459$ ),  $\tau_1 = 0.001$  and  $\tau_2 = 0.001$  ( $H = 378$ ),  $H = 459$ , and  $21 \times 22$  ( $H = 462$ ). (b)  $\tau = 0.1$  ( $H = 451$ ),  $\tau_1 = 0.01$  and  $\tau_2 = 0.001$  ( $H = 114$ ),  $H = 451$ , and  $21 \times 21$  ( $H = 441$ ). (c)  $\tau = 0.2$  ( $H = 347$ ),  $\tau_1 = 0.001$  and  $\tau_2 = 0.01$  ( $H = 28$ ),  $H = 347$ , and  $19 \times 19$  ( $H = 361$ ).



(a)

(b)

(c)

Fig. 4. GHNG (first row), GHSOM (second row), GNG (third row), and SOM (fourth row) results for the 2-D Hexagon input distribution with different densities. (a)  $\tau = 0$  ( $H = 446$ ),  $\tau_1 = 0.001$  and  $\tau_2 = 0.001$  ( $H = 317$ ),  $H = 446$ , and  $21 \times 22$  ( $H = 462$ ). (b)  $\tau = 0.1$  ( $H = 439$ ),  $\tau_1 = 0.01$  and  $\tau_2 = 0.001$  ( $H = 174$ ),  $H = 439$ , and  $21 \times 21$  ( $H = 441$ ). (c)  $\tau = 0.2$  ( $H = 355$ ),  $\tau_1 = 0.001$  and  $\tau_2 = 0.01$  ( $H = 32$ ),  $H = 355$ , and  $19 \times 19$  ( $H = 361$ ).

put neurons out of the X letter shape. In addition, in Fig. 4, since the hexagon distribution has not a homogeneous density, the GHSOM, GNG, and SOM make mistakes because they puts neurons on the low density (lighter tone) areas without samples near the center of the distribution, whereas the GHNG neurons remain further from the center.

In order to provide a quantitative comparison between the unfolding capabilities of the GHNG, GHSOM, GNG, and SOM models, the peak signal-to-noise ratio (PSNR) of the resulting networks has been computed. The PSNR is defined as follows (in decibel, higher is better):

$$\text{PSNR} = 10 \log_{10} \left( \frac{\text{MAX}_I^2}{\text{MSE}} \right) \quad (17)$$

where  $\text{MAX}_I^2$  is the squared Euclidean norm of the vector, which joins the two most distant points in the input distribution support. Likewise, the mean squared error (mse) is given by (lower is better)

$$\text{MSE} = \frac{1}{M} \sum_{i=1}^M \| \mathbf{w}_i - \hat{\mathbf{x}}_i \|^2 \quad (18)$$

where  $M$  is the number of samples in the distribution,  $\mathbf{x}_i$  is the  $i$ th input sample, and  $\mathbf{w}_i$  is the winning neuron corresponding to  $\mathbf{x}_i$ . The PSNR results for the four models are shown in Fig. 5(a). In these plots, each model is plotted in a different color and each configuration with a different symbol: ‘o’ stands for configurations with the highest number of neurons, ‘+’ for configurations with the second highest number of neurons, and ‘\*’ for configurations with the lowest number of neurons. Data points near the top left-hand corner represent the best results. Note that the GNG achieved the best results. This is due to the fact that the selected data sets are more suitable for planar neural networks since they contain nonhollow, connected, and compact data, so that a hierarchical structure is not required. By comparing the results of the two hierarchical models, the GHNG attains better results than the GHSOM in all tested input distributions, because it yields higher PSNRs with similar numbers of leaf neurons  $H$ .

### B. Color Quantization Experiments

This set of experiments is devoted to demonstrate the ability of the GHNG in color image quantization applications.

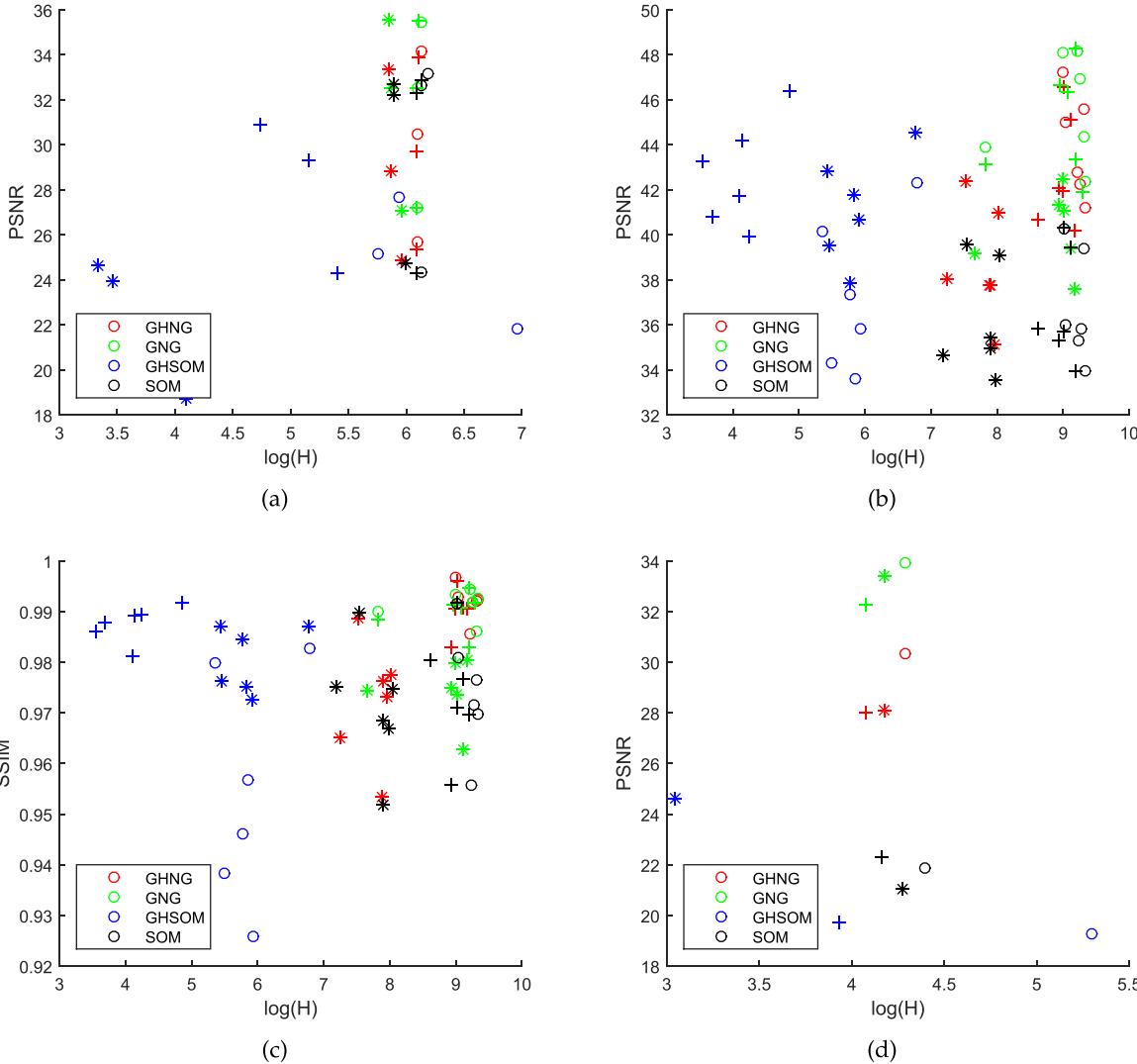


Fig. 5. Experimental results obtained by the GHNG, GNG, GHSOM, and SOM models using different configurations for different experiments. (a) PSNR for self-organization experiments. (b) PSNR for quantization experiments. (c) SSIM for quantization experiments. (d) PSNR for clustering experiments. Each model is plotted in a different color and each configuration with a different symbol: ‘o’ stands for configurations with the highest number of neurons, ‘+’ for configurations with the second highest number of neurons, and ‘\*’ for configurations with the lowest number of neurons. Data points near the top left-hand corner represent the best results.

Color quantization is a typical image-processing task, which consists of selecting a small number of code vectors from a set of available colors to represent a high color resolution image with minimum perceptual distortion. Self-organizing approaches have been extensively used to this end [32]–[36].

For our experiments, six color images broadly adopted in the related literature have been selected from the USC-SIPI Image Database [37]. Since all of them have a resolution of  $512 \times 512$  pixels, the GHNG, GHSOM, GNG, and SOM models are fed with the data sets of  $M = 262\,144$  input samples and during  $N = 524\,288$  time steps (two epochs). The maximum number of levels for GHNG was set to three, so that the size of the learned codebook was not too large. Three different values of the  $\tau$  parameter were used (0, 0.1, and 0.2) to compare the effects of the network size on the GHNG results. The resulting number of leaf neurons was used

to set the number of neurons of the GNG. For the GHSOM, we chose three different values of the  $\tau_1$  and  $\tau_2$  parameters to obtain similar numbers of leaf neurons  $H$  (0.0001/0.001, 0.001/0.0001, and 0.0001/0.0001), whereas for the SOM, we used map sizes with a number of neurons similar to those yielded by the GHNG.

In order to gain a better qualitative understanding of the resulting color image quantizations performed by the considered approaches, the absolute difference of the baboon original image and their corresponding quantized images are shown in Fig. 6. These images have been amplified 20 times for a better appreciation of these differences. This way the errors made by each neural network in representing a color image are highlighted. In these figures, we can see how the SOM and GHSOM yield worse errors with many more neurons. In addition, the GHNG attains a better adaptation than the GNG using the lowest number of neurons ( $H = 2877$ ).

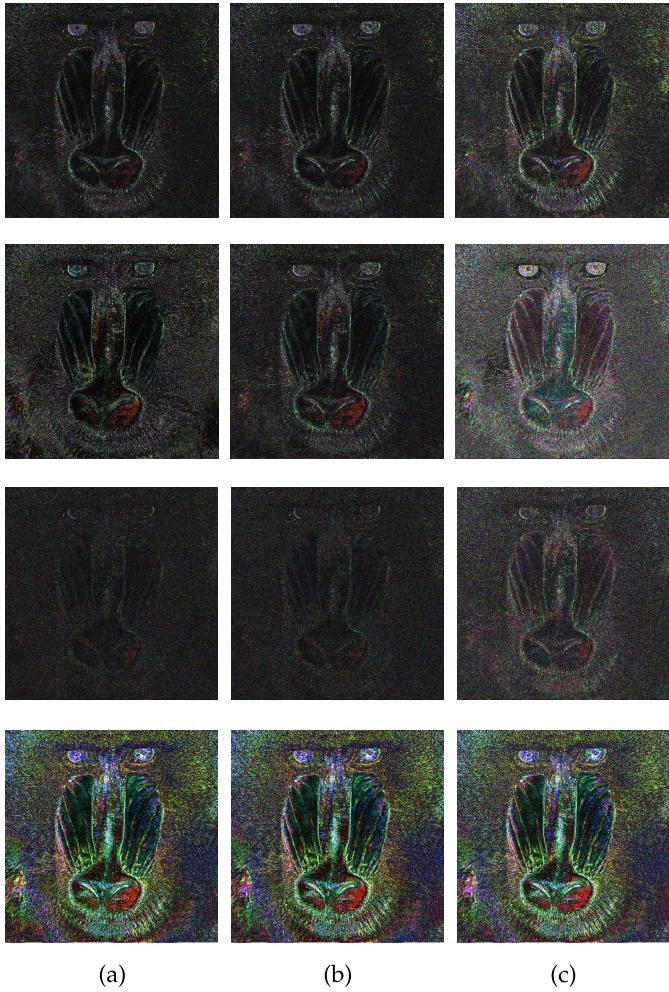


Fig. 6. GHNG (first row), GHSOM (second row), GNG (third row), and SOM (fourth row) results for the baboon color image. (a)  $\tau = 0$  ( $H = 11\,343$ ),  $\tau_1 = 0.0001$  and  $\tau_2 = 0.0001$  ( $H = 84\,361$ ),  $H = 11\,343$ , and  $106 \times 107$  map size ( $H = 11\,342$ ). (b)  $\tau = 0.1$  ( $H = 9742$ ),  $\tau_1 = 0.001$  and  $\tau_2 = 0.0001$  ( $H = 9089$ ),  $H = 9742$ , and  $99 \times 99$  ( $H = 9801$ ). (c)  $\tau = 0.2$  ( $H = 2877$ ),  $\tau_1 = 0.0001$  and  $\tau_2 = 0.001$  ( $H = 1385$ ),  $H = 2877$ , and  $54 \times 54$  ( $H = 2916$ ).

The performance of the neural network results has been quantitatively evaluated using the PSNR (17) and the structural similarity index measure (SSIM). The SSIM is given by (higher is better)

$$\text{SSIM} = \frac{1}{M} \sum_{j=1}^M \frac{(2\hat{\mu}_j \mu_j + c_1)(2\tilde{\sigma}_j + c_2)}{(\hat{\mu}_j^2 + \mu_j^2 + c_1)(\hat{\sigma}_j^2 + \sigma_j^2 + c_2)} \quad (19)$$

where  $M$  is the number of pixels,  $\mu_j$  and  $\sigma_j^2$  are the local mean and variance of the original image computed on a  $11 \times 11$  window centered on pixel  $j$ ,  $\hat{\mu}_j$  and  $\hat{\sigma}_j^2$  are their counterparts for the quantized image,  $\tilde{\sigma}_j$  is the covariance between the two images computed on the same window, and the constants  $c_1 = 0.01$  and  $c_2 = 0.03$  are chosen as recommended in [38]. The values obtained from (19) are averaged over the three RGB channels to obtain the performance measures for color images.

Note that the color image quantization is an unsupervised learning task; therefore, both the training set and the

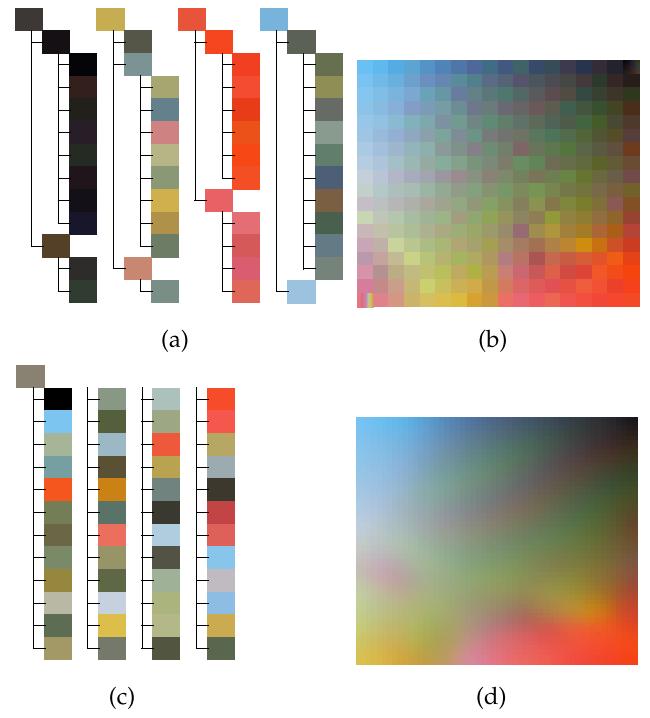


Fig. 7. Color quantization structure of the baboon color image obtained after training with the following models. (a) GHNG with  $\tau = 0.2$  ( $H = 2877$ ). (b) GHSOM with  $\tau_1 = 0.0001$  and  $\tau_1 = 0.001$  ( $H = 1385$ ). (c) GNG with  $H = 2877$ . (d) SOM with a map size of  $54 \times 54$  ( $H = 2916$ ).

performance measures consider all the pixels of the original image. The PSNR and SSIM results are shown in Fig. 5(b) and (c), respectively. By observing these plots, we can see how both the PSNR and SSIM results are quite similar to the GHNG and the GNG. However, the advantage of the GHNG is confirmed since it learns a color hierarchy, where each color can be represented in a higher level of detail. This GHNG capability is compared with the rest of models for  $H = 2877$  in Fig. 7. Note that the GNG and the SOM lack this hierarchical color structure, although the SOM has a grid topology. The hierarchical color structure of the GHSOM is represented in a flat way to be compared with the SOM.

### C. Clustering Experiments

The last set of experiments is designed to test the capabilities of the GHNG when performing clustering tasks and discovering their underlying hierarchical structure. For this experiment, all frames belonging to the *Akiyo*, *Carphone*, *Container*, *Foreman*, and *Suzie Yuv* video sequences [39] have been utilized. This data set has a training set of  $M = 1432$  input samples and a dimensionality of  $D = 25\,344$  ( $176 \times 144$  pixel frames with the three color channels). Since this data set represents high-dimensional data, a reduction of the dimensionality to  $D = 8$  has been considered by principal component analysis (PCA). This PCA has been performed by means of the eigenface method [40], given the high original dimension and the small number of input samples. The maximum number of levels for GHNG was set to three, so that the generated visualizations were clear enough. The training

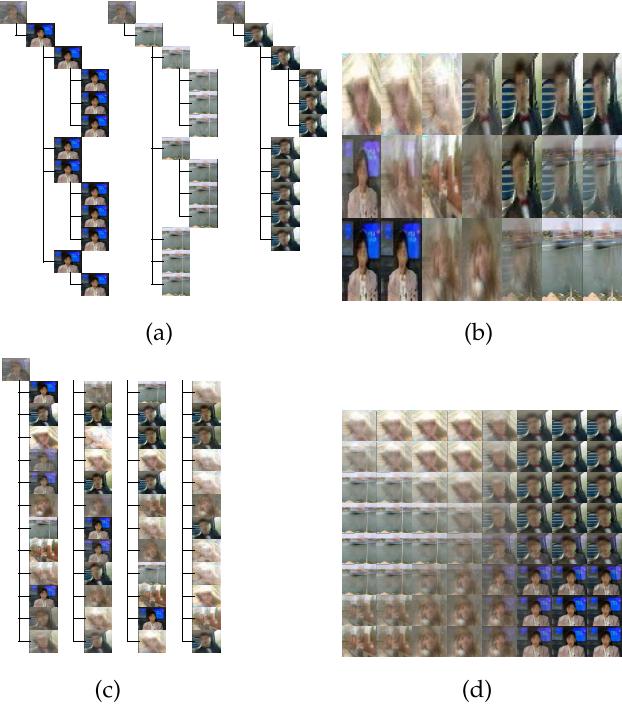


Fig. 8. Clustering structure for the video sequences obtained after training with the following models. (a) GHNG with  $\tau = 0.2$  ( $H = 65$ ). (b) GHSOM with  $\tau_1 = 0.001$  and  $\tau_2 = 0.01$  ( $H = 21$ ). (c) GNG with  $H = 65$ . (d) SOM with a map size of  $8 \times 8$  ( $H = 64$ ).

was done during two epochs, and we considered three different values of the  $\tau$  parameter (0, 0.1, and 0.2) and three different values of the  $\tau_1$  and  $\tau_2$  parameters (0.001/0.01, 0.01/0.001, and 0.001/0.001) for comparative purposes. In addition, the number of neurons of the GNG was set to the number of leaf neurons obtained by the GHNG, whereas the SOM map sizes were set in order to get a similar number of neurons.

The GHNG, GHSOM, GNG, and SOM clustering results are shown in Fig. 8 for the  $\tau = 0.2$  configuration. Please see Supplementary Material to see all the results. Note that in the GHSOM plots, we show the 2-D projection of the GHSOM hierarchy due to its grid topology. As seen in the GHNG plots, **similar frames belonging to the same video sequence are clustered into similar groups very early in the hierarchical structure**. This important aspect is completely missed in the rest of the models, where prototype video frames belonging to different video sequences can be seen close to each other.

Quantitative results for the clustering of frames from different video sequences can be found in Fig. 5(d). The PSNR (17) has been used here for the evaluation of the GHNG, GHSOM, GNG, and SOM results (higher is better). It can be seen that the GHNG overcomes its hierarchical competitor, although the GNG obtains the best results. Therefore, **by imposing a hierarchical structure, neurons have less freedom of movement. However, it allows the visualization of a hierarchy that can be easily understood by a human.**

## V. DISCUSSION

In this section, we outline the most outstanding features of the GHNG model.

- 1) The GHNG model is able to learn the topology of each graph, so that the structure is dynamic not only in terms of the creation of the graphs and units but also in the connections among units, as shown in Section II. This is a substantial enhancement with respect to many self-organizing models both hierarchical and nonhierarchical [3], [13], [41]–[43], which are based on maps with a fixed topology (usually rectangular or hexagonal).
- 2) Some parameters of the GHNG are inherited from the original GNG (Section II-A). The  $a_{\max}$  parameter is used to control the learning of the connections among units in each graph. **The smaller  $a_{\max}$ , the more likely that inactive connections among neurons are removed.** Hence, the small values of  $a_{\max}$  produce graphs with a lower connection density, and vice versa. The higher  $\lambda$ , the slower the neuron creation process. The update of the error variables is controlled by  $\alpha$  and  $d$  (the larger the parameters, the faster that the past errors are forgotten). Finally,  $\epsilon_b$  and  $\epsilon_n$  govern the update of the winning neuron and its neighbors, respectively (the larger the parameters, the more importance that is given to the new inputs). It was not necessary to tune them, since the recommended values from [26] worked correctly in all the experiments.
- 3) The growth is controlled by a user-defined parameter  $\tau$  [see (8)]. This improves the ease of use with respect to the classical GHSOM, where two parameters  $\tau_1$  and  $\tau_2$  must be specified. Moreover, the recursive nature of the growing algorithm of GHNG makes it simple to implement and understand by practitioners not specialized in neural network models.
- 4) The prespecified maximum number of levels (Section II-B) places an additional limit on the size of the model. As explained in Section II-B, the removal of the graphs with fewer than three units and the training set split given by (9) work together to yield a parsimonious hierarchy. However, **it is advantageous to define a maximum level in order to stop the growth at a certain depth.** This is useful to produce visualizations that are not cluttered by the excessive number of levels (see Section IV-C), and it can also be used to place bounds on the computation time required to learn the model.
- 5) The equivariance property of the GHNG (Section III) allows a significant freedom in the scaling and baseline of the input variables, since it is guaranteed that the behavior of the GHNG is not affected by any such transformations on the training set.
- 6) As shown in Section IV, the GHNG can provide better results than the GHSOM in terms of self-organization, and on typical unsupervised learning tasks, such as color quantization and clustering of video data. This is due to the fact that the GHNG contains the GNG, and therefore, it enhances the GNG capabilities to learn the structure of complex input distributions, whereas the GHSOM has a fixed topology as mentioned earlier. **It must be highlighted that a higher number of neurons is not always better, since two models can have similar mses**

with very different numbers of neurons, which would mean that the model with the fewest neurons has the best clustering quality. This clustering performance depends on the learning model, the model configuration, and the data set at hand.

## VI. CONCLUSION

We have presented a new hierarchical self-organizing neural model with dynamic topology learning that we call GHNG. It is based on a modified version of the GNG, which is equipped with a growth control procedure to limit the expansion of the model. A tree of these modified GNGs is learned from the input data set, so that the hierarchical data can be analyzed without the restriction of fixed lattice topologies among the neurons. Some formal properties of the model have been proved, and extensive experimental results have been reported for self-organization and typical unsupervised learning tasks, such as color quantization and clustering. These experimental results show that the proposed model can provide better results than the GHSOM for the aforementioned tasks.

## ACKNOWLEDGMENT

The authors would like to thank the computer resources, technical expertise, and assistance provided by the Supercomputing and Bioinformatics Center of the University of Málaga.

## REFERENCES

- [1] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: A review," *ACM Comput. Surv.*, vol. 31, no. 3, pp. 264–323, Sep. 1999.
- [2] A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data*. Upper Saddle River, NJ, USA: Prentice-Hall, 1988.
- [3] T. Kohonen, "The self-organizing map," *Proc. IEEE*, vol. 78, no. 9, pp. 1464–1480, Sep. 1990.
- [4] T. Kohonen, "Essentials of the self-organizing map," *Neural Netw.*, vol. 37, pp. 52–65, Jan. 2013.
- [5] M. C. Di Piazza, M. Pucci, and G. Vitale, "Intelligent power conversion system management for photovoltaic generation," *Sustain. Energy Technol. Assessments*, vol. 2, no. 1, pp. 19–30, Jun. 2013.
- [6] I. T. Podolak and S. Jastrzębski, "Density invariant detection of osteoporosis using growing neural gas," in *Proc. 8th Int. Conf. Comput. Recognit. Syst., Adv. Intell. Syst. Comput.*, vol. 226. 2013, pp. 629–638.
- [7] J. S. Cope, P. Remagnino, S. Mannan, K. Diaz, F. J. Ferri, and P. Wilkin, "Reverse engineering expert visual observations: From fixations to the learning of spatial filters with a neural-gas algorithm," *Expert Syst. Appl.*, vol. 40, no. 17, pp. 6707–6712, Dec. 2013.
- [8] H. Frezza-Buet, "Online computing of non-stationary distributions velocity fields by an accuracy controlled growing neural gas," *Neural Netw.*, vol. 60, pp. 203–221, Dec. 2014.
- [9] D. Viejo, J. Garcia-Rodríguez, and M. Cazorla, "Combining visual features and growing neural gas networks for robotic 3D SLAM," *Inf. Sci.*, vol. 276, pp. 174–185, Aug. 2014.
- [10] P. M. Yanik *et al.*, "A gesture learning interface for simulated robot path shaping with a human teacher," *IEEE Trans. Human-Mach. Syst.*, vol. 44, no. 1, pp. 41–54, Feb. 2014.
- [11] G. de A. Barreto, A. F. R. Araújo, and S. C. Kremer, "A taxonomy for spatiotemporal connectionist networks revisited: The unsupervised case," *Neural Comput.*, vol. 15, no. 6, pp. 1255–1320, 2003.
- [12] B. Hammer, A. Micheli, A. Sperduti, and M. Strickert, "A general framework for unsupervised processing of structured data," *Neurocomputing*, vol. 57, pp. 3–35, Mar. 2004.
- [13] A. Rauber, D. Merkl, and M. Dittenbach, "The growing hierarchical self-organizing map: Exploratory analysis of high-dimensional data," *IEEE Trans. Neural Netw.*, vol. 13, no. 6, pp. 1331–1341, Nov. 2002.
- [14] J. M. Barbalho, A. Duarte, D. Neto, J. A. F. Costa, and M. L. A. Netto, "Hierarchical SOM applied to image compression," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, vol. 1. Jul. 2001, pp. 442–447.
- [15] J. Lampinen and E. Oja, "Clustering properties of hierarchical self-organizing maps," *J. Math. Imag. Vis.*, vol. 2, no. 2, pp. 261–272, Nov. 1992.
- [16] X. Deng, P. Xu, and C.-H. Chang, "Self organizing topological tree for skin color detection," in *Proc. IEEE Asia-Pacific Conf. Circuits Syst. (APCCAS)*, vol. 2. Dec. 2004, pp. 1097–1100.
- [17] E. López-Rubio and E. J. Palomo, "Growing hierarchical probabilistic self-organizing graphs," *IEEE Trans. Neural Netw.*, vol. 22, no. 7, pp. 997–1008, Jul. 2011.
- [18] E. J. Palomo, J. North, D. Elizondo, R. M. Luque, and T. Watson, "Application of growing hierarchical SOM for visualisation of network forensics traffic data," *Neural Netw.*, vol. 32, pp. 275–284, Aug. 2012.
- [19] E. de la Hoz, E. de la Hoz, A. Ortiz, J. Ortega, and A. Martínez-Álvarez, "Feature selection by multi-objective optimisation: Application to network anomaly detection by hierarchical self-organising maps," *Knowl.-Based Syst.*, vol. 71, pp. 322–338, Nov. 2014.
- [20] Y.-C. Hsu and A.-P. Chen, "A clustering time series model for the optimal hedge ratio decision making," *Neurocomputing*, vol. 138, pp. 358–370, Aug. 2014.
- [21] S.-Y. Huang, R.-H. Tsaih, and F. Yu, "Topological pattern discovery and feature extraction for fraudulent financial reporting," *Expert Syst. Appl.*, vol. 41, no. 9, pp. 4360–4372, Jul. 2014.
- [22] M. Chattopadhyay, P. K. Dan, and S. Mazumdar, "Comparison of visualization of optimal clustering using self-organizing map and growing hierarchical self-organizing map in cellular manufacturing system," *Appl. Soft Comput.*, vol. 22, pp. 528–543, Sep. 2014.
- [23] A. Ortiz, J. M. Górriz, J. Ramírez, and D. Salas-González, "Improving MRI segmentation with probabilistic GHSOM and multiobjective optimization," *Neurocomputing*, vol. 114, pp. 118–131, Aug. 2013.
- [24] Y. Toda and N. Kubota, "Feature extraction based on hierarchical growing neural gas for informationally structured space," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Aug. 2013, pp. 1–7.
- [25] K. Doherty, R. Adams, and N. Davey, "TreeGNG—Hierarchical topological clustering," in *Proc. 13th Eur. Symp. Artif. Neural Netw. (ESANN)*, Apr. 2007, pp. 19–24.
- [26] B. Fritzke, "A growing neural gas network learns topologies," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 7. 1995, pp. 625–632.
- [27] W. Zhou and R. Serfling, "Multivariate spatial U-quantiles: A Bahadur-Kiefer representation, a Theil-Sen estimator for multiple regression, and a robust dispersion estimator," *J. Statist. Planning Inference*, vol. 138, no. 6, pp. 1660–1678, Jul. 2008.
- [28] B. Chakraborty, "On affine equivariant multivariate quantiles," *Ann. Inst. Statist. Math.*, vol. 53, no. 2, pp. 380–403, Jun. 2001.
- [29] X. Dang and R. Serfling, "Nonparametric depth-based multivariate outlier identifiers, and masking robustness properties," *J. Statist. Planning Inference*, vol. 140, no. 1, pp. 198–213, Jan. 2010.
- [30] K. Crimin, J. W. McKean, and S. Sheather, "Discriminant procedures based on efficient robust discriminant coordinates," *J. Nonparam. Statist.*, vol. 19, nos. 4–5, pp. 199–213, Dec. 2007.
- [31] K. Nordhausen, H. Oja, and D. Paindaveine, "Signed-rank tests for location in the symmetric independent component model," *J. Multivariate Anal.*, vol. 100, no. 5, pp. 821–834, May 2009.
- [32] A. H. Dekker, "Kohonen neural networks for optimal colour quantization," *Netw., Comput. Neural Syst.*, vol. 5, no. 3, pp. 351–367, 1994.
- [33] N. Papamarkos, "Color reduction using local features and a Kohonen self-organized feature map neural network," *J. Imag. Syst. Technol.*, vol. 10, no. 5, pp. 404–409, 1999.
- [34] N. Papamarkos, A. E. Atsalakis, and C. P. Strouthopoulos, "Adaptive color reduction," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 32, no. 1, pp. 44–56, Feb. 2002.
- [35] Y. Xiao, C.-S. Leung, P.-M. Lam, and T.-Y. Ho, "Self-organizing map-based color palette for high-dynamic range texture compression," *Neural Comput. Appl.*, vol. 21, no. 4, pp. 639–647, Jun. 2012.
- [36] E. J. Palomo and E. Domínguez, "Hierarchical color quantization based on self-organization," *J. Math. Imag. Vis.*, vol. 49, no. 1, pp. 1–19, May 2014.
- [37] A. Weber. (2010). *The USC-SIPI Image Database*. [Online]. Available: <http://sipi.usc.edu/database/>
- [38] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.
- [39] M. Reisslein, L. J. Karam, P. Seeling, F. H. Fitzek, and T. K. Madsen. (Dec. 2010). *YUV Video Sequences*. [Online]. Available: <http://trace.eas.asu.edu/yuv/index.html>

- [40] M. Turk and A. Pentland, "Eigenfaces for recognition," *J. Cognit. Neurosci.*, vol. 3, no. 1, pp. 71–86, 1991.
- [41] J. J. Verbeek, N. Vlassis, and B. J. A. Kröse, "Self-organizing mixture models," *Neurocomputing*, vol. 63, pp. 99–123, Jan. 2005.
- [42] M. M. Van Hulle, "Maximum likelihood topographic map formation," *Neural Comput.*, vol. 17, no. 3, pp. 503–513, 2005.
- [43] M. M. Van Hulle, "Joint entropy maximization in kernel-based topographic maps," *Neural Comput.*, vol. 14, no. 8, pp. 1887–1906, Aug. 2002.



**Esteban J. Palomo** was born in 1982. He received the M.Sc. and Ph.D. (Hons.) degrees in computer engineering from the University of Málaga, Málaga, Spain, in 2008 and 2013, respectively.

He joined the Department of Computer Languages and Computer Science, University of Málaga, in 2007. In 2015, he joined the School of Mathematical Sciences and Information Technology, Yachay University, Urcuquí, Ecuador, where he is currently a Lecturer. His current research interests include unsupervised learning, self-organization, data mining, image/video processing, and network security.



**Ezequiel López-Rubio** was born in 1976. He received the M.Sc. and Ph.D. (Hons.) degrees in computer engineering from the University of Málaga, Málaga, Spain, in 1999 and 2002, respectively.

He joined the Department of Computer Languages and Computer Science, University of Málaga, in 2000, where he is currently an Associate Professor of Computer Science and Artificial Intelligence. His current research interests include unsupervised learning, pattern recognition, and image processing.