Representation of 2D objects with a topology preserving network

Francisco Flórez, Juan Manuel García, José García, Antonio Hernández,
Departamento de Tecnología Informática y Computación. Universidad de Alicante.
Apdo 99, 03080, Alicante, Spain
{florez,juanma,jgarcia,ahernandez}@dtic.ua.es
http:://www.ua.es/i2rc

Abstract. We propose the use of a self-organizing neural network, the Growing Neural Gas, to represent bidimensional objects, due to its quality of topology preservation. As a result of an adaptative process, the object is represented by a Topology Preserving Graph, that constitutes an induced Delaunay triangulation of their shapes. Features that are extracted from this graph simplify the later operations of classification and recognition, avoiding the high complexity of comparisons between graphs. This model of object characterization allows refining the quality of the representation based on the time available to its calculation, so that it will be the basis for the design of high performance real-time vision architectures. This work opens a new research field, because it employs the topology of a self-organizing neural network as feature, not, as usual, as a classifier.

1. Introduction

Shape representation is expressed by people using terms as elongated, rounded, squared, fragmented,... However, computers find this information insufficient. So, a great deal of different representation methods have been developed [1], [2], [3]. These techniques have the goal of obtaining a set of qualitative and quantitative measures to describe the objects, based on their external characteristics (contours) or on their internal features (silhouette).

Many of the developed representations are approached from an algorithmic point of view, ignoring the aspects of implantation and the characteristics of the computers. This is the reason why, in this paper, we propose a silhouette representation, easy and fast to obtain. This representation model must be easily transferable to the design of high performance architectures, for its calculation under real-time restrictions.

Taking the idea from active representations [4], we have used a self-organizing neural network that adapts its interconnection network to the shape of the objects.

2. Topology preservation

2.1 Topology preserving networks

Researchers have usually considered self-organizing networks as topology preserving models, since it has been thought that as consequence of the competitive learning similar patterns are mapped onto adjacent neurons and, vice versa, neighbouring neurons activate or code similar patterns. In fact, in many cases they are also named as topology preserving feature maps.

However, this is not true in a great number of cases. Martinetz and Schulten [5] have formally defined what is a Topology Representing Network and its relationship with computational geometry structures as the Voronoi Diagram and the Delaunay Triangulation. So that, several models are outside this category, for instance, the highly used Self-Organizing Feature Maps [6].

From previous studies [7], we conclude that neural gases, as the Neural Gas [8] and the Growing Neural Gas [9], are the self-organizing networks that better preserve the topology of bidimensional objects. This is because its topology is not previously defined, but they adapt it during the learning. We use Growing Neural Gas, due to its smaller temporal and spatial complexity.

2.2 Growing Neural Gas

The Growing Neural Gas (GNG) is an incremental neural model able to learn, as the other self-organizing networks do, the topological relationships of a given set of input patterns by means of hebbian learning.

Unlike other methods, the incremental character of this model, avoids the necessity to previously specify the network size. On the contrary, from a minimal network size, a growth process takes place, continued until an ending condition is fulfilled. Also, learning parameters are constant in time, in contrast to other methods whose learning falls basically in decaying parameters.

Learning algorithm. We consider a neural network as:

- a set $\mathcal{N}$ of nodes (neurons). Each neuron $c \in \mathcal{N}$ has its associated reference vector $w_c \in \mathcal{R}^d$. The reference vectors can be regarded as positions in the input space of their corresponding neurons,
- a set C of edges (connections) between pairs of neurons. Those connections are not weighted and his purpose is to define the topological structure.

The GNG learning algorithm to approach the network to the input manifold is as follows:

1. Start with two neurons a and b at random positions w_a and w_b in $\mathcal{R}^d$.
2. Generate an input signal x according to a density function $\mathcal{P}(x)$.

3. Find the nearest neuron (winner neuron) s_1 and the second nearest s_2.
4. Increase the age of all the edges emanating from s_1.
5. Add the squared distance between the input signal and the winner neuron to a counter error of s_1:

$$Derror(s_1) = \left\| w_{s_1} - x \right\|^2 \tag{1}$$

1. Move the winner neuron s_1 and its topological neighbours (neurons connected to s_1) towards x by a learning step e_w and e_n, respectively, of the total distance:

$$Dw_{s_1} = e_w (x - w_{s_1}) \tag{2}$$

$$Dw_{s_n} = e_n (x - w_{s_n}) \tag{3}$$

1. If s_1 and s_2 are connected by an edge, set the age of this edge to 0. If it does not exist, create it.
2. Remove the edges larger than a_{max}. If this results in isolated neurons (without emanating edges), remove them as well.
3. Every certain number l of input signals generated, insert a new neuron as follows:
   - Determine the neuron q with the maximum accumulated error.
   - Insert a new neuron r between q and its further neighbour f:

$$w_r = 0.5\left(w_q + w_f\right) \tag{4}$$

   - Insert new edges connecting the neuron r with neurons q and f, removing the old edge between q and f.
   - Decrease the error variables of neurons q and f multiplying them with a constant a.
   - Initialize the error variable of r with the new value of the error variable of q and f.

1. Decrease all error neuron variables by multiplying then with a constant b.
11. If the stopping criterion is not yet achieved, go to step 2.

3 Representing 2D Objects

Given an image $I(x, y) \in \mathcal{R}$ we perform the transformation $y_{\mathcal{T}}(x, y) = \mathcal{T}(I(x, y))$ that associates to each one of the pixels its probability of belonging to the object, according to a property $\mathcal{T}$. For instance, in figure 1, this transformation is a threshold function.

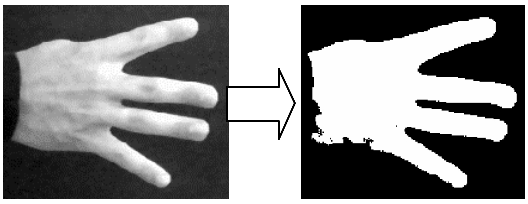

Fig. 1. Silhouette extraction

If we consider $x = (x, y)$ and $\mathcal{P}(x) = y_{\mathcal{T}}(x)$, we can apply the learning algorithm of the GNG to the image I, so that the network adapts its topology to the object. This adaptative process is iterative, so the GNG represents the object during all the learning, giving the opportunity to stop the process if necessary, obtaining a good answer (figure 2).

As a result of the GNG learning we obtain a graph, the Topology Preserving Graph $\mathcal{TPG} = \langle \mathcal{N}, \mathcal{C} \rangle$, with a vertex (neurons) set $\mathcal{N}$ and an edge set $\mathcal{C}$ that connect them (figure 3). This $\mathcal{TPG}$ establishes a Delaunay triangulation induced by the object [9].

Also, the model is able to characterize the diverse parts of an object, or several present objects in the scene that have the same values for the visual property $\mathcal{T}$, without having to initialize different data structures for each one of the objects (figure 4). This is due to the capacity of division of the GNG when removing neurons.

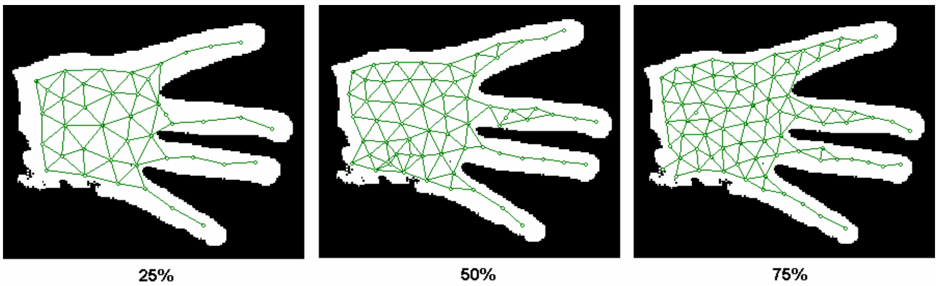

Fig. 2. Adaptation process of the GNG

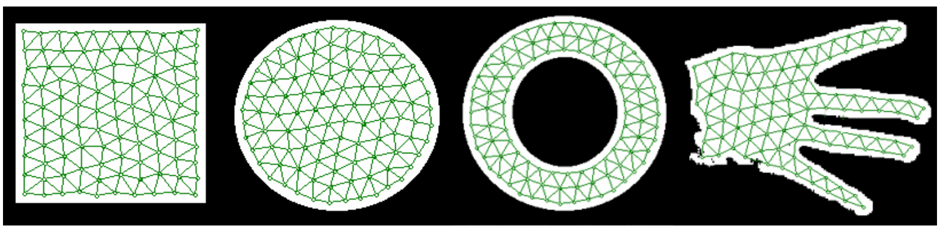

Fig. 3. Results of the GNG learning

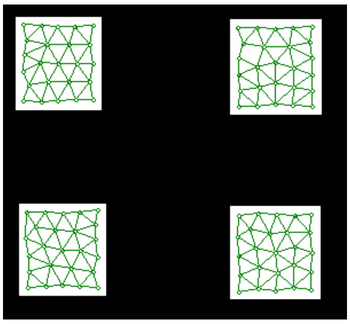

Fig. 4. Representation of different fragments of an object

4 Obtaining features from the $\mathcal{TPG}$

The Topology Preserving Graph is a simple data structure, easy to obtain, that performs a good characterization of the objects. Then, it can be directly used to recognize and classify them. Nevertheless, the comparison between graphs is a problem of high temporal complexity, reason why, we present some features that can be extracted from the $\mathcal{TPG}$. There is a lot of work made in this sense, since this graph is very related to diverse computational geometry structures.

4.1 $\mathcal{TPG}$ Subgraphs

In [10] the following relationship between graphs is established:

$$\mathcal{NNG} \subseteq \mathcal{MST} \subseteq \mathcal{RNG} \subseteq \mathcal{GG} \subseteq \mathcal{DT}$$

where $\mathcal{NNG}$ is the nearest neighbourhood graph, $\mathcal{MST}$ is the minimum spanning tree, $\mathcal{RNG}$ is the relative neighbourhood graph, $\mathcal{GG}$ is the Gabriel graph and $\mathcal{DT}$ if the Delaunay triangulation.

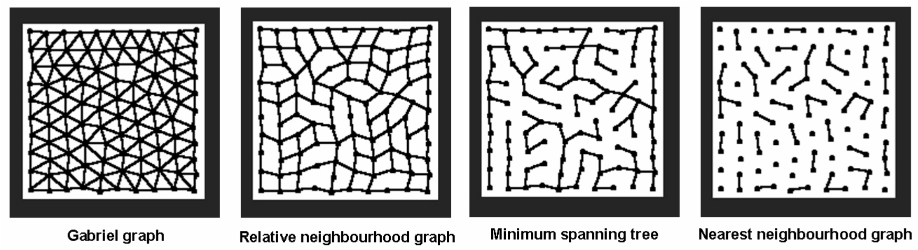

Fig. 5. $\mathcal{TPG}$ Subgraphs

This relation of subgraphs allows to simplify the original graph (figure 5) and, therefore, to reduce the cost of comparison.

4.2 Characterization of the topology of the $\mathcal{TPG}$

From the topology of the graph diverse measures can be obtained that give information on the morphologic characteristics of the objects. For instance:

- Number of edges: a greater number of edges suggests a more compact object, since there is a greater capacity of connection (Table 1).
- Neighbourhood histogram: calculated as the number of neurons that have each one of the neighbourhoods. In figure 6, it is observed that compact shapes have more nodes with high neighbourhoods. On the other hand, if the object has corners (square) or very narrow zones (fingers), there are neurons with few neighbours.
- Statistical measures of the neighbourhoods: the average neighbourhood, its median, its standard deviation,... gives information similar to the one of the neighbourhood histogram (Table 2).
- Statistical measures of the edge lengths: the existence in a $\mathcal{TPG}$ of edges of smaller length suggests the presence of more edges and, therefore, a compact shape (Table 1).
- Number and types of polygons: although the $\mathcal{TPG}$ would have to make a triangulation of the input space, when being induced by the object, causes that some of the edges of the Delaunay triangulation disappear and, therefore, appear lines, as well as others polygons.

Table 1. Information from the number and length of the edges

Object	Edges	Average length	Median	Standard deviation
Square	258	17.67	17.46	1.88
Círcle	256	16.71	16.76	2.02
Ring	226	17.17	17.00	2.06
Hand	228	19.07	18.87	2.25

Table 2. Information extracted from the neighbourhoods

Object	Neighb.	Average Neighb.	Median	Standard deviation
Square	516	5.16	6	1.20
Circle	512	5.12	6	1.25
Ring	454	4.54	4	1.27
Hand	456	4.56	4	1.18

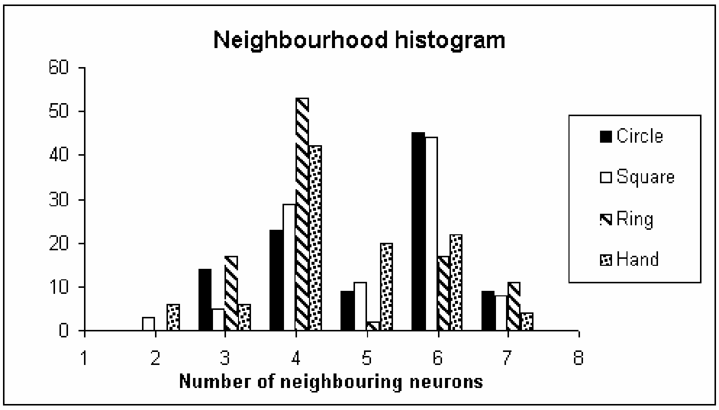

Fig. 6. Neighbourhood histogram

4.3 Moments

Moments are one of the most used methods to characterize the shape of an object. In this case, we calculate the scaled central moments and Hu moments [9]. The obtained results are not the same that calculating the moments directly from the object, but they allow its characterization, with a lower cost (Table 3).

4.4 Extracting the contour of the $\mathcal{TPG}$

Given a $\mathcal{TPG}$, we obtain its contour as the set of edges that belong to a single polygon (figure 7). This contour represents the convex hull minus the deficit of convexity of the $\mathcal{TPG}$.

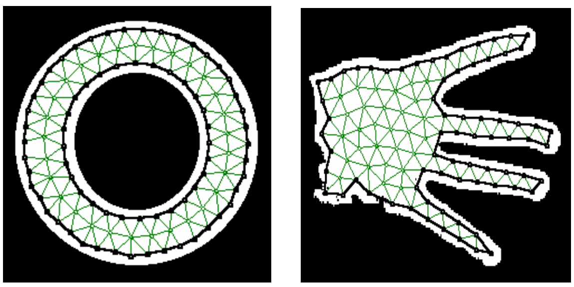

Fig. 7. Contour of the $\mathcal{TPG}$

Table 3. Moments of the $\mathcal{TPG}$.

	Square	Circle	Ring	Hand
h_{20}	23.18	20.71	38.13	45.72
h_{02}	21.70	18.35	36.24	23.92
h_{11}	-0.12	0.22	0.01	1.23
h_{30}	-0.32	-0.75	-0.81	21.21
h_{12}	-0.08	0.11	1.08	47.06
h_{21}	-0.80	0.32	-0.34	-15.29
h_{03}	0.94	1.15	-4.12	7.96
f_1	44.88	39.07	74.37	69.63
f_2	2.26	5.77	3.59	481.66
f_3	2.42	4.64	32.32	27807
f_4	0.18	2.57	19.98	4714.47
f_5	-0.22	-4.00	337.05	-31364862
f_6	0.23	-4.98	-37.62	97994
f_7	0.09	5.15	358.62	-32844276

4.5 Extracting the skeleton of the $\mathcal{TPG}$

Given the great number of works that use the medial axis transform to represent the objects, we have thought to extract it from the $\mathcal{TPG}$. For it, we calculate the Voronoi diagram of the edges of the contour of the graph (figure 8). We obtain the skeleton by joining those points that belong to the border of three or more Voronoi regions (figure 9).

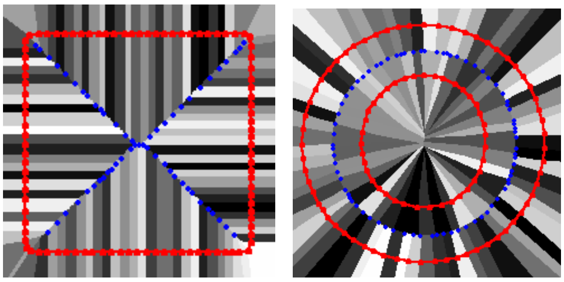

Fig. 8. Voronoi diagram of the contour of the $\mathcal{TPG}$

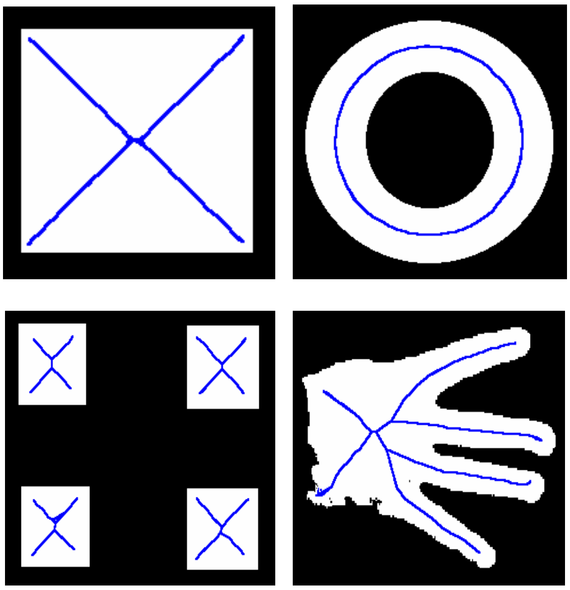

Fig. 9. Skeletons

5 Conclusion and future works

In this paper, we have demonstrated the capacity of representation of bidimensional objects by a self-organizing neural network. Establishing a suitable transformation function, the model is able to adapt its topology to the shape of an object. Then, a simple, but very rich representation of the objects is obtained.

The model, by its own adaptation process, is able to divide itself so that it can characterize different fragments from an object.

We have extracted several features from the $\mathcal{TPG}$, in order to facilitate tasks of more high level, like classifications and recognitions.

This work opens a new research line, in the development of neural networks whose topology adapts to the shape of the objects. In fact, we are already working in the development of models that adapt to the contours.

Finally, the iterative and parallel performance of the presented representation model is the departure point for the development of high performance architectures, that supply a characterization of an object depending on the time available.

References

1. Gonzalez, R.C., Woods, R.E.: Digital Image Processing, Addison-Wesley Publishing Company. Reading, Massachussets (1992)
2. Loncaric, S.: A Survey of Shape Analysis Techniques. Pattern Recognition, Vol. 31(8) (1998) 983-1001.
3. Sonka, M., Hlavac, V., Boyle, R.: Image Processing, Analysis, and Machine Vision, Brooks/Cole Publishing Company. 2nd edition (1998)
4. Blake, A., Yuille, A. (eds.): Active Vision, The M.I.T. Press. Cambridge, Massachussets (1992)
5. Martinetz, T., Schulten, K.: Topology Representing Networks. Neural Networks, Vol. 7(3) (1994) 507-522.
6. Kohonen, T.: Self-Organizing Maps, Springer-Verlag. Berlin Heidelberg (1995)
7. Flórez, F., García, J.M., García, J., Hernández, A.: Representing 2D objects. Comparison of several self-organizing networks. 3rd WSES Conference on Neural Networks and Applications, Interlaken (2002)
8. Martinetz, T., Schulten, K.: A "Neural-Gas" Network Learns Topologies. Artificial Neural Networks, T. Kohonen, K. Mäkisara, O. Simula and J. Kangas (eds.), Vol. 1 (1991) 397-402.
9. Fritzke, B.: A Growing Neural Gas Network Learns Topologies. In Advances in Neural Information Processing Systems 7, G. Tesauro, D.S. Touretzky and T.K. Leen (eds.), The M.I.T. Press. Cambridge, Massachussets (1995)
10. Fortune, S.: Voronoi Diagrams and Delaunay Triangulations. In Discrete and Computational Geometry, J.E. Goodman and J. O'Rourke (eds.), CRC Press LLC. Boca Raton, Florida (1997)