Abstract: In this paper we propose a neural network model to simplify and 2D meshes. This model is based on the Growing Neural Gas model and is able to simplify any mesh with different topologies and sizes. A triangulation process is included with the objective to reconstruct the mesh. This model is applied to some problems related to urban networks.

Key-Words: neural networks, self-organizing maps, mesh simplification, GNG3D algorithm, triangulations.

1. Introduction

Research on mesh simplification in Computer Graphics and Scientific Visualization has led to a substantial number of methods within the last two decades, one can refer to the many surveys available (see, for example, [3, 9]).

In the last years a tremendous amount of work has been done on mesh simplification, especially in three dimensions. Most of the techniques or algorithms proposed to accomplish this objective are based on reducing the mesh complexity either by merging/collapsing elements or by re-sampling vertices. Some techniques are well-known and have been widely studied, as for example those based on Vertex Decimation (see [12]), Re-Tiling Polygonal Surfaces (see [16]) or Edge Contraction (proposed by Hoppe et al., [6]).

There is a group of algorithms that offer a completely different technique to tackling this problem. They are algorithms that learn the shape of the object to simplify and are based in some particular artificial neural networks. The advantage of these methods with respect to the classical group is that they do not need to work with the whole object to reduce its complexity; they construct the simplifications using iterative processes or training processes.

Self-organizing networks are able to generate interesting low-dimensional representations of high dimensional input data. The most well-known of these models is the Kohonen’s Self-Organizing Feature Map (SOM) [7]. It has been used in the last decades to study a great variety of problems such as vector quantization, biological modeling, combinatorial optimization and so on.

A Self-organizing network (SOM) consists of a set of neurons arranged in some topological structure which induces neighborhood relations among the neurons. An n-dimensional vector is attached to every neuron, which determines the specific n-dimensional input signal to which the neuron is maximally sensitive. By assigning to every input signal the neuron with the nearest reference vector, a mapping is defined from the space of all possible input signals onto the neural structure.

Self-organizing networks [10] learn in an unsupervised manner from a stream of input signals. For each input signal, the neuron with the nearest reference vector is determined, called the winner unit. Then, the reference vectors of the winner unit and some of its topological neighbors are moved towards the input signal. The adaptation of topological neighbors distinguishes self-organization (winner takes most) from competitive learning, where only the winner is adapted (winner takes all).

The Growing Neural Gas (GNG) algorithm [4] is an unsupervised incremental clustering algorithm. Given some input distribution in \( \mathbb{R}^n \), it incrementally creates a graph, or network of nodes, where each node in the graph has a position in \( \mathbb{R}^n \). One of the primary characteristics of GNG is that it is an adaptive algorithm, in the sense that if the input distribution slowly changes over time, GNG is able to adapt, that is, to move the nodes so as to cover the new distribution.

Starting with two nodes the algorithm constructs a graph in which nodes are considered neighbors if they are connected by an edge. The neighbor information is maintained throughout execution by a variant of competitive Hebbian Learning.
2. The GNG3D model
The GNG model presents some drawbacks itself when it is applied to the problem of mesh optimization for three-dimensional objects in computer graphics. The Growing Neural Gas 3D (GNG3D) model [1] tries to solve the problems of the GNG model to reconstruct a 3D object. We can summarize GNG3D saying that consists on two phases. Phase 1 consists on a modification of the GNG model introducing the possibility to remove some nodes or neurons that do not provide relevant information about the original model. After this phase we obtain a cloud of points and edges connecting them, but we have no information about the faces of the simplified mesh. In Phase 2 we have developed a method to construct the faces of the optimized mesh from the information provided by Phase 1. As this information permits us to reconstruct the simplified mesh, we name it the reconstruction phase.

Some basic characteristics of this model can be summarized in the following items:
- The accumulated error of nodes in the algorithm is a quantity that allows us to determine the regions where there is a low density of nodes according to the vertices existing in the original 3D object. Regions where the accumulated error is high are suitable candidates for being covered with new nodes or neurons.
- The local activation counter variable is useful to eliminate nodes and avoid the problem of local minima.
- Parameters involved in the optimization algorithm are not fixed. They are obtained experimentally.
- The parameter \( \lambda \) is just used to determine the moment to insert a new node in the mesh.
- There exists a parameter \( \mu \) to determine when to eliminate a node that has not been referenced in the previous iterations.

For a more detail description of the model and some examples related with the simplification of 3D models, see [1] and [2]. This model has demonstrated its efficiency and ability to produce simplified 3D meshes with different topologies and has been compared with the classical and well-known Quadric Error Metrics model [5].

In two dimensions, most approaches to the problem of representing unstructured meshes are based on either triangles or edges. The simplest data structure is based on triangles. The early development of triangulation comes from surveying and the art of constructing maps. Surveyors and cartographers used triangles as the basic geometric feature for calculating distances between points on the Earth’s surface and a position’s elevation above sea level.

During recent decades, advances in computer hardware and software have also brought triangulation technology into many new areas of application. Triangulations are also used in geographical information systems (GIS), mainly for the purpose of representing parts of the Earth’s surface, more commonly referred to as terrain models or triangular irregular networks.

3. A 2D algorithm based on the GNG3D model
Our objective is to modify the GNG3D model to design a 2D model to simplify meshes in the plane. The basic modifications that we introduce are summarized in the following items:
- Initially we take M nodes randomly as a starting point, which represents a quite different initial approach to that proposed by the GNG3D model, where the number of nodes of the final mesh was totally controlled by the user throughout the training process of the construction of the mesh. This is a critical aspect because it eliminates the entire process of adding and removing nodes. Therefore, we focus on the process of self-learning the shape of the original mesh.
- Throughout this process we do not take into account the triangles or faces that form the initial mesh.
- The parameters involved in the training process are \( c_{\text{win}} \), \( c_n \) and \( \lambda \). The parameter \( c_{\text{win}} \) is related to the displacement of the winner node, while \( c_n \) is related to the displacement of the neighbor nodes in the plane. The parameter \( \lambda \) is introduced in order to be sure that any node of the initial set of random points will stay isolated during the entire execution of the algorithm.
- The triangulation phase is similar to the GNG3D triangulation process.

The 2D model is described by the following steps.
We consider, as a starting point, a 2D triangle mesh consisting of
- a set \( A = \{n_1, n_2, \ldots, n_N\} \) of vertices or nodes,
- a set \( T = \{t_1, t_2, \ldots, t_L\} \) of triangles among node pairs.
We can say that the set $T$ constitutes the set of triangles that make up the original 2D mesh. Consequently, each element of the triangle $T$ is generated by three vertices, that is, $t = \{n_i, n_k, n_l\}$, where $n_i, n_k, n_l$ are elements of the set $A$.

It is important to point out that we are going to fix the number of vertices of the simplified mesh; therefore, if we set the number of vertices of the simplified mesh as $M$, the objective is to find a set of vertices $K = \{k_1, k_2, \ldots, k_M\}$, where $k_i$, for $i=1,2,\ldots,M$, are the new nodes in the simplified triangle mesh. Note that always the set $M < N$ and we take $M$ fixed. The set $K$ may be computed running the following algorithm:

**INIT:** Start with $M$ points $k_1, k_2, \ldots, k_M$ at random positions $w_{k1}, w_{k2}, \ldots, w_{kM}$ in $\mathbb{R}^2$.

Initialize a local counter $l_i$ to zero for every point or node $k_i$, for $i=1,2,\ldots, M$. We set the maximum number of iterations as $I$

1. Generate an input signal $\xi$ that will be a random point $n_i \in A$ from the original mesh.
2. Find the nearest node $s_1$.
3. Find the second and third nearest nodes, $s_2$ and $s_3$ to the input signal.
4. Increment the local counter of the winner node $s_1$.
5. Move $s_1$ towards $\xi$ by fractions $e_{\text{win}}$ respect to the total distance

$$\Delta w_{s_1} = e_{\text{win}}(\xi - w_{s_1}).$$

6. Move $s_2$ and $s_3$ towards $\xi$ by fractions $e_{\text{win}}$ respect to the total distance:

$$\Delta w_{s_2} = e_{\text{win}}(\xi - w_{s_2}),$$
$$\Delta w_{s_3} = e_{\text{win}}(\xi - w_{s_3}).$$

7. Repeat steps 1 to 6 $\lambda$ times, with $\lambda$ an integer of order $O(10^3 \cdot N)$. We have the local counter vector

$$L_c = (l_c(k_1), l_c(k_2), \ldots, l_c(k_M)).$$

For $i = 1,2,\ldots,M$,

- If $l_c(k_i) = 0$ then
  - We choose randomly a node $n_i$ from the original mesh.
  - Make $k_i = n_i$.
  - Go to point 1.

- If $l_c(k_i) \neq 0$ then continue.

8. Stop when the maximum number of iterations $I_c$ has been reached.

This first part of the algorithm can be seen as a training process, where a set of nodes representing the new vertices of a planar mesh are obtained. So far, nothing about the triangles of the original planar mesh has been mentioned.

Now, we perform the triangulation process, as it is described in the GNG3D model.

**INIT:** Consider the set $A$ of the original nodes, $T$ the triangles of the original 2D triangle mesh, and $K$ the set of the nodes obtained by the above self-organizing algorithm.

1. Associate each node of the original mesh with a node of the set $K$.

For every $n_i$, for $i = 1,2,\ldots,N$, find $j \in \{1,2,\ldots,M\}$ such that

$$|w_{n_i} - k_j| \leq |w_{n_i} - w_{k_j}|, 1 \in \{1,2,\ldots,M\},$$

where $w_{n_i}$ represents the position of the node $n_i$.

Save $(n_i, k_j)$. We say that $k_j$ is the node associated to $n_i$.

2. Change the nodes of the original triangles by their associated nodes.

For every $t = \{n_{i1}, n_{i2}, n_{i3}\} \in T$, substitute

$$\{n_{i1}, n_{i2}, n_{i3}\} \rightarrow \{k_{j1}, k_{j2}, k_{j3}\},$$

where $k_{j1}, k_{j2}, k_{j3}$ are the associated nodes of $n_{i1}, n_{i2}, n_{i3}$, with $j_1, j_2, j_3 \in \{1,2,\ldots,M\}$.

- If $k_{j1} = k_{j2} = k_{j3}$, then save $t = \{n_{i1}, n_{i2}, n_{i3}\}$.

- If $k_{j1} = k_{j2}$, or $k_{j1} = k_{j3}$, or $k_{j2} = k_{j3}$, then continue.

3. Graph the set

$$C = \{t = \{n_{i1}, n_{i2}, n_{i3}\}, k_{j1} \neq k_{j2} \neq k_{j3}\}.$$
4. Applications of the 2D model to meshes in the plane
Modeling of cities as complex systems requires diverse information like socioeconomic and explicit demographic data as well as what we can call 'real-world' spatial data [2]. In the context of spatial planning and regional development, the increasing land consumption and the increase in impervious surfaces during the last decades are research topics of utmost interest.
Urban design concerns the arrangement, appearance and functionality of towns and cities, and in particular the shaping and uses of urban public space. Urban design theory deals primarily with the design and management of public space, and the way public places are experienced and used. One of the more relevant topics that urban design considers is the urban topology, density and sustainability.

In this section, our objective is to introduce some problems related to urban design and urban networks where may be interesting to apply some self-organizing algorithms. Some of these subjects are
- Urban density.
- Urban acupuncture.
- Urban transport networks.
- Wi-Fi networks design in urban areas.

4.1 Urban density.
Recently, in Spanish coast line has become the witness of one of these deep land transformation phenomena, in this case of unprecedented extension and speed. The south east Mediterranean coast, known as Costa Blanca is precisely the geo-economic context in which the favorable weather conditions and other attractive elements fostered a vertiginous development of the touristic activity in the last 30 years. And this fact led to an equally rapid urban development which has been profusely studied by architects.

The need to accommodate the huge number of people who came to this coast -- both from the rest of Spain and other European regions turned what used to be small fishing villages in the fifties into major tourist centers. In addition, vast portions of virgin territory were developed and occupied. Two types of solutions to these processes of urban growth were generally implemented: a model based on dispersion or spread, and a model based on densification (i.e., the concentration of density).

In opposition to the model of concentration, the spread model was planned as a way to provide to a not very high social class the American dream of being the owner of a single house with a garden by the sea: a kind of Florida on the Mediterranean that -- as its American counterpart -- was soon full of retired people from all over Europe. This illusion was behind hundreds of projects of housing developments which colonized this territory, like a gigantic carpet of small houses, small swimming pools and small private gardens.

The most obvious effects of this kind of urbanization are physical at a first level: a massive depredation of the territory, devastated landscapes that were turned into non-sustainable (in terms of energy, maintenance, network installations or safety, for example) anonymous places.

To focus the problem, what we try to do is to generate new nodes of concentration of density or new nodes to perform some actions. At this point is where we believe that the application of adapted neural network models may be a useful tool to change these urban concentrations.

For a more detailed description of the self-organizing model applied to the concept of urban density see [14].

4.2 Urban acupuncture.
Jaime Lerner, the three-time former mayor of Curitiba, Brazil, a city best known for its innovative approaches to urban planning, is calling for what he terms urban acupuncture to bring revitalization and sustainability to the worlds metropolitan areas. Lerner thinks that tackling urban problems at appropriate pressure points can cause positive ripple effects throughout entire communities (see [8]). He noted that even the poorest cities can boost their standards of living by using techniques like bus rapid transit, designing multiuser buildings, and encouraging residents to live closer to their workplaces. Although many cities spend decades building underground rail systems or other costly long-term projects, every city can improve its quality of life in 3 to 4 years.
For a more detailed description of the self-organizing model applied to the concept of urban acupuncture see [13]. In this paper the theoretical approach is applied to a concrete case of a real town (Elx, Spain). In the example developed, the urban acupuncture approach proposes to act with specific projects in a selection of hot points in the historic center of this town. The application of the self-organizing algorithm solves the problem of determining those points at which specific actions must be taken.

4.3 Urban transport networks.
The roadway network in an urban area includes intersections and streets through which traffic moves. These elements can be translated naturally into a structure of nodes and links (including impedance measures), respectively. However, all of the nodes or links in this structure do not have the same importance or weight in the whole structure. So, in a real transport network constructed in any town, we have main streets, main roads, main communication nodes, and so on.

For example, let us suppose that we must design a bike path in an urban area. If we design an initial mesh that represents this urban area, we might need to consider that all areas do not have the same importance in the whole area covered by the mesh. As a consequence, it will be necessary to establish some nodes with a greater weight or importance in the final design.

For a detailed description of this application and some real examples we can see [11], where we perform the design of some transport networks.

4.4 Wi-Fi networks design.
Actually, many of the larger municipalities have installed or are planning to install Wi-Fi networks for creating city-wide wireless coverage. In some cases, the coverage area is over a hundred square miles. These deployments have had ups and downs in terms of signal coverage and performance, and they are very costly. As an alternative to covering large expansive areas, some municipalities and private entities are building smaller-scale Wi-Fi networks (hot zones) offering wireless Internet connectivity to smaller groups of people.

In this process, one of the most important items to be considered is the problem of node locations. Here is where we apply our algorithm, since we are going to establish for each area separately two dimensional simplified meshes covering all inhabited areas.

We must decide where to place the nodes Wi-Fi, so we must establish a more streamlined network than the original where each point represents a possible location of the node. Moreover, the simplified network must take into account areas with higher density of houses; in such areas nodes must be closer one each other so that network traffic is not affected. In other words, the simplified network must learn the shape of the original network.

For a detailed description of this application and some real examples we can see [11], where we perform the design of some transport networks.

5. Conclusions

We propose a self-organizing algorithm for 2D triangle meshes which is able to generate a simplification mesh from an original mesh consisting of a set of vertices or nodes and triangles, with the main property of faithfully reproducing the original mesh topology.

This paper discusses the way in which this algorithm can be applied when we consider some problems related to urban design. Particularly, we think about some specific problems (urban density, urban acupuncture, urban transport networks and Wi-Fi networks design) where this algorithm could be applied.

The detailed study of real examples shows the effectiveness of the model to produce simplified networks with a similar topology as the original. Then, the characteristics of this model suggest that it may be very appropriate to design and develop different urban networks.

References:


