



HAL
open science

A neural network model to develop actions in urban complex systems represented by 2D meshes.

Leandro Tortosa, Jose Luis Oliver, Jose Francisco Vicent

► **To cite this version:**

Leandro Tortosa, Jose Luis Oliver, Jose Francisco Vicent. A neural network model to develop actions in urban complex systems represented by 2D meshes.. International Journal of Computer Mathematics, 2011, 88 (16), pp.3361-3379. 10.1080/00207160.2011.608843 . hal-00744820

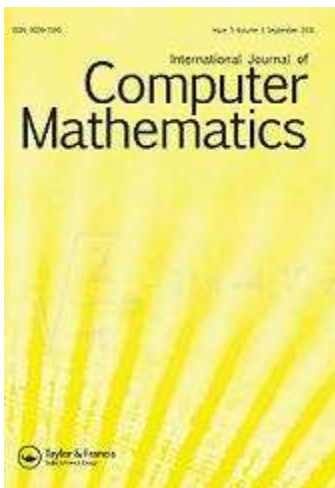
HAL Id: hal-00744820

<https://hal.science/hal-00744820>

Submitted on 24 Oct 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



A neural network model to develop actions in urban complex systems represented by 2D meshes.

Journal:	<i>International Journal of Computer Mathematics</i>
Manuscript ID:	GCOM-2010-0342-A.R1
Manuscript Type:	Original Article
Date Submitted by the Author:	29-Mar-2011
Complete List of Authors:	Tortosa, Leandro; University of Alicante, Ciencia de la Computación e Inteligencia Artificial Oliver, Jose; University of Alicante, Expresión Gráfica y Cartografía Vicent, Jose; University of Alicante, Ciencia de la Computacion e Inteligencia Artificial
Keywords:	Artificial neural networks, triangulation, mesh simplifications, urban systems, urban density, GNG3D model

SCHOLARONE™
Manuscripts

A neural network model to develop actions in urban complex systems represented by 2D meshes

Jose L. Oliver^a Leandro Tortosa^{a*} and Jose F. Vicent^a

^aUniversity of Alicante, Ap. Correos 99, E-03080, Alicante, Spain

(Received 00 Month 200x; in final form 00 Month 200x)

The main idea of this work is to present a tool which may be useful to generate a mesh of points where urban actions may be taken after analyzing and understanding complex urban situations. By the word complex we mean urban concentrations without precise limits and without a recognizable geometry pattern. In these situations, it is very hard for the architects to understand the system. Therefore, it is very difficult to define an action plan for this type of urban situations. What we propose is an adaptation of a neural network algorithm to work in the context of urban networks. Our objective is to develop an strategy to change this weakness of sparse urban development by activating public spaces with new meanings. A new 2D triangle mesh simplification model is introduced with the central property of preserving the shape of the original mesh. The mathematical model presented consists of a self-organizing algorithm which objective is to generate the positions of the nodes of the simplified mesh; afterwards, a triangulation algorithm is carried out to reconstruct the triangles of the new simplified mesh. With this algorithm it is possible to perform specific actions in an urban space, because of the urban territory can be considered as a complex mesh with nodes and edges. A real example of an urban action is shown with the introduction of a wireless network in a residential area.

Keywords: submission instructions; source file coding; Artificial neural networks; urban systems; mesh simplifications; triangulation; GNG3D model; urban density

AMS Subject Classification: 68T05; 68Q32; 68Q25

1. Introduction

Making architecture implies necessary disturbing an environment. Louis Kahn used to describe architecture as anything that nature doesn't know how to do. There is an extensive bibliography about Louis Kahn and many books have been devoted to him, as for example [14, 18]. These processes environmental change have taken place through history according to different dynamics, which imply different speeds and deepness of transformation. As an example, the Industrial Revolution led to a tremendous process of occupation and transformation of the landscape that took place in a extremely short period of time mostly due to a very fast demographic growth in urban areas. We can see [3] and [11] as general references to introduce general ideas and concepts related to urban development.

Modeling of cities as complex systems requires diverse information like socio-economic 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.

*Corresponding author. Email: tortosa@ua.es

1 Recently, in Spanish coast line has become the witness of one of these deep land
2 transformation phenomena, in this case of unprecedented extension and speed. The
3 south east Mediterranean coast, known as *Costa Blanca* ('white coast'), is precisely
4 the geo-economic context in which we place our study area.

5 The favorable weather conditions and other attractive elements fostered a ver-
6 tiginous development of the touristic activity in the last 30 years. And this fact
7 led to an equally rapid urban development which has been profusely studied by
8 architects world wide.

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

16 Recently, the authors have discussed several applications of the self-organizing
17 model we use in this paper to some problems related to urban design, (see [19, 20]).

18 In [19] the central objective is to present a 2D triangle mesh simplification al-
19 gorithm and demonstrate its ability to perform the task of simplifying an original
20 mesh with irregular shape. The main idea of that paper is to show the effectiveness
21 of the neural network model to simplify the 2D mesh. However, the main objective
22 of this paper is quite different, as it attempts to exploit the capabilities of the neu-
23 ral network algorithm in urban environments. In our paper we find fundamental
24 ideas that are not covered in [19, 20], such as the issue of land occupation and
25 the strategy we can follow by using this algorithm in order to to improve urban
26 sustainability. Another fundamental original idea is the development of specific
27 'actions' in urban environments. This paper takes a comprehensive approach to
28 two urban problems: the generation new nodes of concentration to improve the
29 density of some urban areas and the problem of performing some specific actions
30 on various areas with particular attention to sustainable development.

31 2. The problem of land occupation and our strategy

32 In opposition to the model of concentration which may be represented, for example,
33 by the city of Benidorm, the spread model was planned as a way to provide to a
34 not very high social class the *American dream* of being the owner of a single house
35 with a garden by the sea: a kind of Florida on the Mediterranean that – as its
36 American counterpart – was soon full of retired people from all over Europe. This
37 illusion was behind hundreds of projects of housing developments which colonized
38 this territory, like a gigantic carpet of small houses, small swimming pools and
39 small private gardens.

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

44 However, in addition to these obvious physical consequences, something perhaps
45 more serious is occurring in a deeper level. This diffuse city encourages a way of life
46 based on the individual, and therefore renounces an essential element of the social,
47 as it is the public domain (following Arnold Reijndorp's definition in [7]). There
48 are no real public spaces in these urban developments, and nowadays we can admit
49 without any doubt that this approach has both physical and social consequences.
50 We can easily confirm this phenomenon by taking a look at the green areas planned
51

1 in such urban developments: in a very short period of time, they become degraded
 2 and useless spaces. But this is only a small example. As we said before, this failure
 3 of the concept of the public seems to be related to the lack of identity and liveliness
 4 of these places, which yields a social disaster. Our proposal aims at tackling this
 5 situation: we want a strategy to change this weakness of sparse urban development
 6 by activating public spaces with new meanings.
 7

8 Our strategy is inspired by the concept of density. It is believed that one of the
 9 main factors for the success of public spaces e.g., green spaces, is the existence
 10 of a *critical mass* related to them. With the expression *critical mass* we mean
 11 the precise number of persons and uses for an area. In other words and using the
 12 example we just mentioned: a minimum urban density linked to each greenery is
 13 needed. Otherwise, they don't have any real sense and they become the residual
 14 and desolated spaces they are now. Based on this idea, we propose what we call
 15 here a process of re-densification.
 16

17 Nevertheless, the physical reality of these urban dispersions implies a level of
 18 geometrical complexity which makes it very difficult to know where to implement
 19 the actions. In addition, the urban tissues of this sort of housing developments has
 20 been created using different kinds of geometrical bases, with irregular patterns,
 21 diffuse limits and non hierarchical outlines. In this situation, the right choice of
 22 where to act becomes a relevant decision, because changing the density in the
 23 wrong places could have awful consequences. In fact, giving a new order to this
 24 dispersion – as we aim to by generating new nodes of concentration which will free
 25 some ground – may have the danger of unbalancing the system, creating favored
 26 spots versus others which may become ghettos. Therefore, the precise disposition
 27 of the new nodes of concentration of density becomes critical in this strategy. We
 28 will identify these nodes in accordance with the characteristics of the problem that
 29 we are studying. In the case of urban re-densification, each node will be associated
 30 with some type of architectural building. In the case of the wireless network each
 31 node will represent a hub or antenna.
 32

33 The idea of using an optimization algorithm based on Self-Organizing Maps,
 34 is based on the intrinsic characteristics of these networks. It is necessary for our
 35 meshes to group nodes so that we can establish relationships between them. This
 36 type of learning where there is no external factor to the network that determines its
 37 performance, it is an unsupervised learning. Because the network does not receive
 38 any information from the environment to indicate whether the generated output is
 39 correct or not, it is necessary a self-organization process.
 40

41 Additionally, in this type of self-organizing networks, the nodes move towards
 42 areas of greater density, which is an important advantage compared to structures
 43 based on supervised learning. This will be crucial for the type of problem we want
 44 to address.
 45

46 Summarizing, Self-Organizing maps are different from other artificial neural net-
 47 works in the sense that they use a neighborhood function to preserve the topologi-
 48 cal properties of the input space. This makes these networks useful for visualizing
 49 low-dimensional views of high-dimensional data.
 50

51
 52 **3. The mathematical model**
 53

54 To focus the problem, what we try to do is to generate new nodes of concentration of
 55 density or new nodes to perform some actions. At this point is where we believe that
 56 the application of adapted neural network models may be a useful tool to change
 57 these urban concentrations. Among the extensive literature on neural networks, we
 58 can cite [6, 8] as very good introductions to the mathematical theory of artificial
 59
 60

neural networks.

In our case, the network is a mesh (or grid or network) of elements that are laid on a topography according to a non hierarchical geometrical system. The objective is to give a new order to this system, always maintaining its limits, and finding a new critical level of density in certain places, while at the same time avoiding new unbalanced situations.

We expect that the new resulting mesh may promote a proper zoning of the public spaces generated by releasing the land which was occupied before. With this action, we hope that all the values associated with the collective dimension, which are present in successful urban models yet absent at these sites, will be encouraged.

In artificial neural networks theory, a self-organizing network [9, 10, 16] 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.

3.1 The mathematical description

The model we present in this section is able to simplify any 2D triangle mesh, with the primary characteristics summarized in the following:

- It allows to obtain a planar point set that will constitute the vertices of the simplified mesh. These points are obtained using a neural network algorithm; more exactly, an incremental clustering algorithm as it will be described in the following.
- It performs a triangulation process based on a comparison between the original triangle mesh and the reduced set of vertices. After this process, a new simplified two dimensional mesh is generated with a similar shape as the original one.
- It allows to approximate or simplify any 2D triangle mesh, even presenting irregular shapes.

The model may be subdivided into two different phases: an optimized algorithm and a triangulation process. The optimization algorithm is based on the idea of Self-Organizing Maps and consists of computing a set of vertices which constitutes a simplification of the original mesh, while the second phase consists on the triangulation of the vertices obtained in the previous step with the aim of generating the final simplified mesh. We give the name of optimization algorithm in the sense that we try to find an optimal allocation of the new nodes with respect to the original mesh.

In the first phase, the objective is to find a set of vertices with the basic property that the geometric distribution of the new vertices minimizes the distance between them and the original planar points in the original triangle mesh. For this purpose, an algorithm based on the concept of self-organizing maps introduced by Kohonen [9] is used. We describe now in detail the algorithm.

Let us consider, as a starting point, a 2D triangle mesh consisting of

- a set $A = \{n_1, n_2, \dots, n_N\}$ of vertices or nodes,
- a set $T = \{t_1, t_2, \dots, t_L\}$ of triangles among node triples.

1 We can say that the set T constitutes the set of triangles or faces that make up
 2 the original 2D mesh. Consequently, each element of the set T is given by
 3

$$4 \quad t_i = \{n_j, n_k, n_l\}, \quad n_j, n_k, n_l \in A.$$

5
 6 It is important to point out that we are going to fix the number of vertices of the
 7 simplified mesh; therefore, if we set the number of vertices of the simplified mesh
 8 as M , then the objective is to find a set of vertices
 9

$$10 \quad K = \{k_1, k_2, \dots, k_M\},$$

11 where k_i , for $i = 1, 2, \dots, M$ are the new nodes in the simplified triangle mesh.
 12 Note that always $M < N$ and M is fixed. The set K may be computed by means
 13 of the following algorithm.
 14

15 *Phase 1: the optimization algorithm.*

16 INIT: Start with M points k_1, k_2, k_M at random positions $w_{k_1}, w_{k_2}, \dots, w_{k_M}$ in
 17 \mathbb{R}^2 . Initialize a local counter to zero for every point.

- 18 (1) Generate an input signal ξ that will be a random point $n_i \in A$ from the
- 19 original mesh.
- 20 (2) Find the nearest node s_1 to the input signal.
- 21 (3) Find the second and third nearest nodes, s_2 and s_2 to the input signal.
- 22 (4) Increment the local counter of the winner node s_1 .
- 23 (5) Move s_1 towards ξ by fractions ϵ_{win} respect to the total distance

$$24 \quad \Delta w_{s_1} = \epsilon_{win}(\xi - w_{s_1}).$$

- 25 (6) Move s_2 and s_3 towards ξ by fractions ϵ_n ($\epsilon_{win} > \epsilon_n$) respect to the total
- 26 distance:

$$27 \quad \Delta w_{s_2} = \epsilon_n(\xi - w_{s_2}),$$

$$28 \quad \Delta w_{s_3} = \epsilon_n(\xi - w_{s_3}).$$

- 29 (7) Repeat steps 1 to 6 λ times, with λ an integer. We check the local counter
- 30 for every node.
 - 31 • If the local counter for any $k_i, i = 1, 2, \dots, M$ is zero, then remove this
 - 32 node and add a new node between the node with higher local counter
 - 33 value and any node adjacent to it.
 - 34 • If the local counter for every $k_i, i = 1, 2, \dots, M$ is greater than zero, then
 - 35 continue and repeat steps 1 to 6.
- 36 (8) Stop when the maximum number of iterations has been reached.

37 This first part of the algorithm can be seen as a training process, where a set of
 38 nodes representing the new vertices of a planar mesh are obtained. So far, nothing
 39 about the triangles or faces of the original planar mesh has been mentioned. The
 40 second part of the process of generation of the simplified mesh is ready to begin.
 41 Now, a triangulation process is developed with the aim of reconstructing the new
 42 mesh. Our proposal constitutes a post-process which uses the information provided
 43 by the self-organizing algorithm and the information about the nodes and trian-
 44 gles of the original mesh. This triangulation procedure can be summarized in the
 45 following algorithm.
 46

47 *Phase 2: the triangulation process.*

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 $i = 1, 2, \dots, N$, find $j \in \{1, 2, \dots, M\}$ such that

$$|w_{n_i} - w_{k_j}| \leq |w_{n_i} - w_{k_l}|, \quad l \in 1, 2, \dots, M,$$

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 $t_i = \{n_{i_1}, n_{i_2}, n_{i_3}\} \in T$, substitute

$$\{n_{i_1}, n_{i_2}, n_{i_3}\} \longrightarrow \{k_{j_1}, k_{j_2}, k_{j_3}\},$$

where $k_{j_1}, k_{j_2}, k_{j_3}$ are the associated nodes of $n_{i_1}, n_{i_2}, n_{i_3}$, with $j_1, j_2, j_3 \in \{1, 2, \dots, M\}$.

- If $\{k_{j_1}, k_{j_2}, k_{j_3}\}$ are different, then save $t_i = \{n_{i_1}, n_{i_2}, n_{i_3}\}$.
- If $k_{j_1} = k_{j_2}$, or $k_{j_1} = k_{j_3}$, or $k_{j_2} = k_{j_3}$, then continue.

- (3) Graph the set

$$C = \{t_i = \{n_{i_1}, n_{i_2}, n_{i_3}\}, \quad k_{j_1} \neq k_{j_2} \neq k_{j_3}\}.$$

- (4) If some node is isolated we add a new triangle, linking this node with their adjacent nodes.

We can summarize the process of triangulation, saying that we perform a comparison between the nodes of the original 2D triangle mesh with the nodes generated by the first algorithm, in order to determine the triangles that must be graphed in the new mesh, following a *learning shape* process.

3.2 Some highlights of the model

The basic features of the optimization algorithm may be summarized in the following items:

- How to choose M . The number of vertices of the simplified mesh is a value chosen by the user and depends on the characteristics of the problem. For example, if we consider the problem of designing a Wi-Fi network in an urban area, we can consider M as the number of routers or antennas that we have to extend the signal. If we consider the problem of designing an urban transport network in a city, we set M as the number of stops of the network. Summarizing, M is chosen by the user based on the conditions or restrictions imposed by the problem.
- Initially we take M nodes at random positions as a starting point. This represents a quite different initial approach to other methods based on neural networks, like those based on the Growing Neural Gas [5] or the Growing Cell Structure [4] models, where the initial number of nodes are two. 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 ϵ_{win} , ϵ_n and λ . The parameter ϵ_{win} is related to the displacement of the winner node, while ϵ_n is related to the displacement of the neighbor nodes in the plane. The parameter λ 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 idea of displacing the nodes s_1 and s_2 closest to the winner node (point 3 of the optimization algorithm), aims to ensure that not only the winner node approaches to the original mesh, but the closest nodes to the winner one also suffer a shift towards the original mesh. Thus, we get the final mesh quickly acquires its final configuration, reducing the number of global iterations in the training process.

Some clarification is necessary regarding the parameter λ and the point 7 of the algorithm. As we randomly chose the geometric positions of the initial M nodes (point 1 of the algorithm), it is possible that some of these initial points were generated in positions far away from the area covered by the original mesh. Accordingly, no signal will be near these points and, therefore, these nodes will never be *winners* in the training process. Consequently, its local counter will remain zero as the number of iterations increase. So, this is the way we have to detect when a node is outside the area of the initial triangle mesh. This is the reason to introduce the condition in the point 7 of the self-organizing algorithm. This feature will be studied in more detail later when we present a real example of triangle mesh simplifications. Then, we will visualize how isolated vertices can arise when we try to simplify networks with very irregular shapes and the initial vertices are chosen randomly. We will see an example is Section 3.3.

In the self-organizing algorithm the positions of the initial vertices are modified with the purpose of learning the shape of the original mesh. After this process we have no information about the triangles that make up the simplified mesh. We only have the geometric position of the new vertices; therefore, an efficient algorithm may be implemented to reconstruct the 2D mesh preserving the original shape.

We have added a condition at the end of the triangulation algorithm. The primary objective of this condition is to avoid the existence of holes or unconnected regions in the final mesh. When the shapes of the original meshes are not too complicated, we have no problem in the final result with holes or isolated regions; however, as it has been observed in the examples performed, when the shape of the original mesh is really complex, we can have problems with the appearance of unconnected regions or isolated vertices, specially when $M \ll N$.

Another aspect that we want to highlight is related to the computational cost of the triangulation process. The only part that requires a cost of computational operations is point 1 of the algorithm, where we associate each node of the original mesh A with a node of the set K . We need to compute, for a fixed n_i ,

$$d^2(n_i, k_1), d^2(n_i, k_2), \dots, d^2(n_i, k_M),$$

which represents 8 arithmetic operations each time we compute the square distance. As we repeat these computations for every k_j , with $j = 1, 2, \dots, M$, the cost will be $8M$ operations. In addition, as the number of nodes of A is N , we repeat this procedure N times, which represents a total cost of $O(MN)$, with $M \ll N$. If we consider that the direct implementation of the Delaunay method leads to algorithms whose computational complexity is $O(N^2)$, that is, quadratic, we can conclude that the computational cost of the triangulation proposed in this algorithm is efficient from the point of view of its computational complexity.

1 The optimization algorithm proposed to obtain the reduced set of nodes or ver-
2 tices that better agree with the topology of the original mesh, is quite different
3 from other similar algorithms based on self-learning processes. The Kohonen's *Self-*
4 *Organizing Feature Map* (SOM) [9] method considers a hyper-rectangular structure
5 on the graph, while Neural Gas [13] is a pure vector quantization method which
6 not define any topology among the units. The Growing Neural Gas model [5] does
7 not impose any explicit constraints on the graph topology; the graph is generated
8 and continuously updated by competitive Hebbian Learning.

9 The advantage of Self-Organizing Map model to other types of Artificial Neu-
10 ral Networks is their unsupervised aspect of not depending on knowing the input
11 space. In spite of its high popularity, SOM encounters several drawbacks, such
12 as being computationally expensive, needing much learning time, being memory
13 lavishing, and searching slowly. These disadvantages have motivated the develop-
14 ment of various improvements, each having its own advantages and disadvantages.
15 Our approach seeks to improve two key aspects: the computational complexity and
16 memory requirements (see [17]).

17 In our algorithm and model we do not impose any explicit constraint on the
18 graph; this is continuously updated by competitive Hebbian Learning; however,
19 our approach does not start from two initial neurons or nodes; moreover, k nodes
20 are initially distributed randomly, with the advantage that this implies, since it
21 eliminates the process of introducing new neurons or nodes. We can say that this
22 modification represents a great advantage from the computational point of view,
23 since the process of creating new neurons is highly expensive within the overall
24 learning process of the network. This simplicity is the greatest advantage of the
25 proposed algorithm over other methods that require much longer and complex
26 training process.

3.3 Some numerical aspects

34 To assess the performance of the algorithm proposed in Section 3, several exper-
35 iments were conducted using some original 2D triangle meshes. Both the opti-
36 mization algorithm and the triangulation process were implemented in Matlab.
37 The numerical examples performed with 2D triangle meshes of different sizes and
38 topologies show us, as a general characteristic, that our model can produce high
39 fidelity approximations of any original triangle mesh, no matter how complex the
40 shape of the mesh.

41 To carry out the experiments, it is necessary to specify a set of parameters to
42 run the optimization algorithm. The parameters that must be specified are the
43 following:

- 44 • ϵ_{win} is related to the displacement of the winner node.
- 45 • ϵ_n is related to the displacement of the neighbor nodes.
- 46 • λ was introduced in order to be sure that any node of the initial set of random
47 points will stay isolated during the entire execution of the algorithm.

48 There is no theoretical result that provides us a parameter set that can always
49 produce the best results for the simplified mesh. Consequently, the determination
50 of the set of parameters to run the algorithm is obtained experimentally.

51 The values of these parameters used in our examples have been $\epsilon_{win} = 0.8$, while
52 $\epsilon_n = 0.1$. The parameter λ is not fixed, as it depends on the number of nodes N ;
53 in many examples we used $\lambda = 10 \cdot N$.

54 We do not consider in the design of the algorithm to introduce a reduction
55 factor for the parameter ϵ_{win} . We have performed tests for several examples with
56

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

For Peer Review Only

Figure 1. Original 2D triangle mesh with the 25 nodes placed at random.

Figure 2. Final mesh after running the self-organizing algorithm and the triangulation process.

different values of ϵ_{win} , but there were no noteworthy differences in the outcome of the simplified meshes.

In the collection of examples studied it has been observed that when the topology of the mesh is very irregular and the initial nodes are generated completely at random, it may be the case that when performing a low number of iterations, the geometric positions of the nodes are not optimal for the reconstruction of the final simplified mesh. In other words, when the above conditions are given, we suspect that some gaps or isolated nodes may appear in the simplification mesh. This behavior can be visualized in Figures 1 and 2.

The behavior of the algorithm shown in the mesh of Figure 1 is not unusual and has a clear explanation. In the Figure 1, if we pay attention to the situation of the

Figure 3. Inhabited areas.

initial vertices, then it is noted that some of these points are far from the original 2D mesh. We can affirm that, for example, nodes 1, 11 and 21 are located outside the triangle structure of the mesh. As a consequence of this fact, when performing successive iterations, none of these nodes will be the winner in the training process (remember that the input signal is always a node of the original mesh). Therefore, they will never move towards the structure and, as a result of this, some nodes will become isolated and will remain static throughout the process, regardless of the number of iterations (see the gaps that appear in the resulting mesh shown in Figure 2).

The occurrence of these isolated vertices leads us to introduce the concept of local counter in the algorithm. It is easy to understand that the local counter of an isolated vertex remains zero. When the local counter is checked, really what we are doing is detecting the isolated vertices. This is the reason for introducing the point 7 in the algorithm. An example is shown in order to visualize this behavior.

4. An example of urban actions

To assess the performance of the model proposed in this paper, several experiments were conducted using some original 2D triangle meshes. We show one of them. In this section we want to analyze in detail an urban action over a residential area of houses, with geographical coordinates $(38.25, -0.7)$ (latitude and longitude). The residential area to study is shown in Figure 3.

Actually, many of the larger municipalities have installed or are planning to install wireless 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 wireless networks (hot zones) offering wireless Internet connectivity to smaller groups of people. This appears to be a

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60



Figure 4. The image (a) represents the original 2D triangle mesh with the 273 nodes (area red), while the image (b) represents the mesh after running the self-organizing algorithm to obtain the initial vertices or nodes.

more feasible approach.

Suppose that the municipality that owns this residential area wants to establish a wifi network that reaches all corners of the inhabited areas. As it is shown clearly in Figure 3, we observe two clearly inhabited zones in the whole area, that we label as *red area* (up left zone) and *blue area* (down right zone). Remark that the topology of the urban area studied is not regular, but rather the opposite. Its form is quite complex, making it difficult to generate a simplification that covers the entire area.

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 of the wireless network, 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.

We work with each of the inhabited areas independently.

First area: red color area.

In the residential area shown in Figure 3, we identify each of the houses or buildings with a node in the mesh and perform a triangulation process with these nodes, obtaining a two-dimensional grid made up of planar triangles, as we can see in Figure 4 (a). This initial mesh has 273 vertices or nodes (houses) and 352 triangles.

Note that our goal is to achieve, while maintaining the same topology of the mesh, an optimal distribution of the new nodes (houses) with the aim to create more space between them to establish new services or enhance green spaces. So, we consider, as an initial parameter, that the final mesh will be composed by 125 nodes (half of the initial nodes); that is, we have that $N = 273$ and $M = 125$. Therefore, the new simplified mesh of the residential area will have 125 nodes, with the same shape as the original one.

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

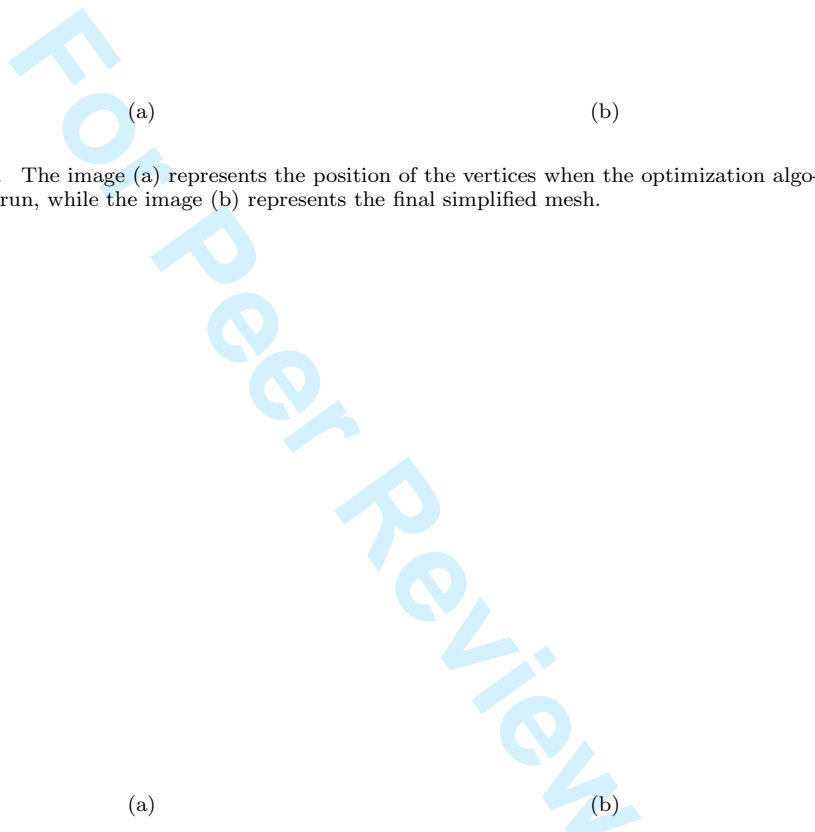


Figure 5. The image (a) represents the position of the vertices when the optimization algorithm is run, while the image (b) represents the final simplified mesh.

Figure 6. The image on the left represents the original 2D triangle mesh with the 156 nodes (area blue), while the image on the right represents the mesh after running the self-organizing algorithm to obtain the initial vertices or nodes.

To reach this objective we apply our optimization algorithm to the original mesh.

The starting point of the self-organizing algorithm is to place K points, k_1, k_2, \dots, k_K , at random positions in the graphic. We take, for this example, $K = 125$ nodes, but this is a parameter of the algorithm and must be chosen by the user. After generating, at random, the initial positions of the 125 nodes of the simplified mesh (Figure 4, (b)), we continue running the training process in the self-organizing algorithm. In this case, after $\lambda = 5000$ iterations, we check the local counters of the n_1, \dots, n_{293} nodes of the original network; then, we proceed to remove the vertices that have not been winners in any iteration. After this step, we follow with the iterations until the stopping criteria is reached, that is, when the number of iterations is reached.

After performing 750000 iterations, we stop the process, obtaining the final po-

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

(a) (b)

Figure 7. The image on the left represents the position of the vertices when the optimization algorithm is run, while the image on the right represents the final simplified mesh.

sitions of the vertices for the simplified network. The experimental result obtained is shown in Figure 5 (a). At this moment it is possible to carry out the triangulation process from the information provided by the self-organizing algorithm. The basis of triangulation process is the comparison between the original nodes of the network and the new nodes. The reconstruction of the simplified mesh is shown in Figure 5 (b).

As shown in Figure 5 (b), we have obtained a simplified two-dimensional grid that shows where we can place the network nodes. Then, a network covering all houses that we want to reach has been drawn. Furthermore, because the properties of the self-organizing algorithm proposed, this network has the property that takes into account the higher density of nodes in the original mesh.

Second area: blue color area.

Now, we carry out a process similar to that developed for the case of the red color area.

In the residential area shown in Figure 6 (a), we identify, as we did in the other area, each of the houses or buildings with a node in the mesh and perform a triangulation process with these nodes, obtaining a two-dimensional grid made up of planar triangles, as we can see in Figure 6 (b). This initial mesh has 156 vertices or nodes (houses) and 221 triangles.

To apply the optimization algorithm, we consider, as an initial parameter, that the final mesh will be composed by 30 nodes; that is, we have that $N = 156$ and $M = 30$. Therefore, the new simplified mesh covering the residential area will have 30 nodes, with the same shape as the original one. Note that in this case, the number of vertices we take to generate the simplified mesh is much less than the number of nodes used in the previous case (area red). In the previous case we construct a mesh composed by half of the vertices of the original one while in this case we generate a mesh composed by only one fifth of the nodes in the original node.

After generating, at random, the initial positions of the 30 nodes of the simplified mesh (Figure 7, (a)), we proceed running the training process in the self-organizing algorithm. In this case, after 4000 iterations, we check the local counters of the n_1, \dots, n_{293} nodes of the original network; then, we proceed to remove the vertices

1 that have not been winners in any iteration. After this step, we follow with the
2 iterations until the stopping criteria is reached.

3 After performing 500000 iterations, we stop the process, obtaining the final po-
4 sitions of the vertices for the simplified network. Then, the triangulation process is
5 performed using, as a starting point, the information provided by the self-organizing
6 algorithm. The reconstruction of the simplified mesh is shown in Figure 7 (b).

7 In other words, what is observed in Figure 7 (b) is that we have obtained a
8 simplified two-dimensional grid that shows us some locations where we can place
9 the network nodes. It is noteworthy that a network covering all houses that we want
10 to reach has been drawn. As it happened in the other area studied, because the
11 properties of the self-organizing algorithm proposed, this network has the property
12 that takes into account the higher density of nodes in the original mesh.

13 This example shows the efficiency of the model, especially the part related to the
14 self-organizing process; if we look at the figures with the initial nodes and the figures
15 obtained after running the algorithm, and compare them, it is clearly observed
16 how the initial points (randomly generated) move towards the original nodes and
17 remain in the area covered by the original network. This may be interpreted as a
18 self-organized process in which the shape of the original network is learned through
19 successive iterations.

20 Running the algorithm several times for this example, we found that it is not
21 necessary to run many iterations to obtain good approximations of the original
22 mesh. In addition, we also note that the algorithm is equally efficient regardless of
23 the proportion of nodes chosen for the simplified mesh.

24 5. An example of land occupation

25 As mentioned in Section 2, the indiscriminate occupation of territory in the area in
26 which we live has produced negative consequences for the sustainable development
27 of our territory. A proposal to control this disproportionate growth may be to pose
28 a lower densification of houses and buildings in cities and its surroundings, as well
29 as in the residential areas (very common in our environment).

30 In this section we consider a real example to reduce a density housing in a densely
31 populated urban area. Let us take the example of a residential area located in
32 the province of Alicante (Spain) with geographical coordinates (37.59, 0.40) NW
33 (latitude North and longitude West).

34 In Figure 8 we have a small piece of this area. Notice that we have a big carpet
35 of small houses, small swimming pools and small private gardens.

36 From the image in Figure 8, we design a two-dimensional mesh where the nodes
37 represent small private houses with private gardens. The two-dimensional mesh is
38 shown in Figure 9 and consists of 306 nodes.

39 It is well known by specialists in urban planning that this model of urbanism
40 does not encourage contact among the people. Rather the opposite, it encourages
41 a way of life based on the individual, renouncing to an essential element of the
42 social: the public spaces. But the only way to create common and public spaces
43 is by means of a re-densification of the territory. And that is our main purpose in
44 this example.

45 We propose in this example a re-densification of the area to create a greater
46 number of communal spaces such as parks, gardens, sports facilities, etc, that
47 encourages communication and relationship among the members of the community.
48 Consequently, we perform a simplification of the original mesh to keep the same
49 topology and optimize the location of the new nodes (housing) in order to create
50 a space between the different nodes of the new mesh. Let us assume that we want
51
52
53
54
55
56
57
58
59
60

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

For Peer Review

Figure 8. Inhabited area.

to reduce by half the mesh density (approximately). In this example, we are going to take the number of vertices or nodes of the simplified mesh as 135. This means that we reduce, approximately, by half the buildings of the whole area represented in the image.

Then, we have that our parameters are $M = 306$ and $N = 135$. We run the algorithm performing a million iterations and taking $\lambda = 100,000$.

The final locations of the nodes in the simplified mesh, as well as the triangulation of the points after running the simplification model proposed, is shown in Figure 10.

Probably looking at the figure in which we have drawn the original mesh and the simplified mesh we may not immediately realize the benefits of conducting a re-densification as proposed. Figure 11 shows the mesh simplified with some polygons between nodes, representing spaces generated from the re-densification process.

Of course, these spaces do not exist in the original mesh, where urban speculation does not allow spaces or areas of expansion. Now we have some spaces that offer us a huge potential in terms of urban planning. Therefore, in this case, by re-densification of this type we can propose the creation of green areas available to all residents, such as gardens, parks, tours for the development of physical activity, etc. We can also build sports facilities areas.

With all this, we aim to promote and enhance relations and contact between members of the community.

Figure 9. Mesh corresponding to inhabited area in Figure 8.

6. Conclusion

A new 2D triangle mesh simplification model has been introduced with the central property of preserving the shape of the original mesh. The model presented consists of a self-organizing algorithm which objective is to generate the positions of the nodes of the simplified mesh; afterwards, a triangulation algorithm is carried out to reconstruct the triangles of the new simplified mesh.

Among the various applications of the model we have chosen one related to urban density. An example using a real geographical area is shown to demonstrate that the model is able to reduce urban concentration, keeping the exact topology of the terrain and creating more space between buildings.

Some experimental examples have been performed using original meshes with irregular shapes. It is shown in detail an example which requires the introduction of a wifi network in a residential area.

In all examples considered, the proposed model generates simplified meshes where the topology of the simplified grid is similar to the original, regardless of the number of vertices used for the final grid. Then, the characteristics of this model suggest that it may be very appropriate to develop different urban actions in specific points regardless of complexity or topology of urban systems.

References

- [1] R. Alvarez, J. Noguera, L. Tortosa, and A. Zamora, *A mesh optimization algorithm based on neural networks*, Information Sciences 177 (2007), pp. 5347–5364.
- [2] C. Aubrecht, K. Steinnocher, M. Hollaus, and W. Wagner, *Integrating earth observation and GI-*

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

Figure 10. Simplified mesh corresponding to inhabited area in Figure 8.

Science for high resolution spatial and functional modeling of urban land use, Computers, Environment and Urban Systems 33(1) (2009), pp. 15–25.

[3] J. Barnett, *An introduction to urban design*, Harper and Row, New York, 1982.

[4] B. Fritzke, *Growing cell structures - a self-organizing network for unsupervised and supervised learning*, Neural Networks 7(9) (1994), pp. 1441–1460.

[5] B. Fritzke, *A growing neural gas network learns topologies*, Advances in Neural Information Processing Systems 7 (1995), pp. 625–632.

[6] D. Graupe, *Principles of Artificial Neural Networks*, World Scientific Publishing Company, 2007.

[7] M. Hajer and A. Reijndorp, *In Search of the New Public Domain*, NAi Publishers, 2002.

[8] M.H. Hassoun, *Fundamentals of Artificial Neural Networks*, MIT Press, Cambridge, MA 1995.

[9] T. Kohonen, *Self-Organizing formation of topologically correct feature maps*, Biological Cybernetics 43 (1982), pp. 59–69.

[10] T. Kohonen, *Self-Organizing Maps*, Springer Verlag, New York, 2001.

[11] M. Larice and E. MacDonald, editors, *The Urban Design Reader*, Routledge, New York, 2007.

[12] D.P. Luebke, *A developer’s survey of polygonal simplification algorithms*, IEEE Computer Graphics and Applications 21(3) (2001), pp. 24–35.

[13] T. Martinez and S. Schulten *A Neural gas network learn topologies*, In Proc. ICANN 1 (1991), pp. 397–402.

[14] R. McCarter, *Louis I. Kahn*, Phaldon Press, 2009.

[15] J. Noguera, L. Tortosa, and A. Zamora, *Analysis and efficiency of the GNG3D algorithm for mesh simplification*, Applied Mathematics and Computation 197 (2008), pp. 29–40.

[16] K. Obermayer and T. Sejnowski, editors, *Self-Organizing Map Formation*, MIT Press, Cambridge MA, 2007.

[17] Y. Prudent and Ab. Ennaji, *An incremental growing neural gas learns topologies*, Proceedings of International Joint Conference on Neural Networks, Montreal, Canada (2005), pp. 1211–1216.

[18] J. Rykwert, *Louis Kahn*, Harry N. Abrams publisher, 2001.

[19] L. Tortosa, J.F. Vicent and A. Zamora, *A model to simplify 2D triangle meshes with irregular shapes*, Applied Mathematics and Computation 216 (2010), pp. 2937–2946.

[20] L. Tortosa, J.F. Vicent, A. Zamora and J.L. Oliver, *Reducing Urban Concentration Using a Neural Network Model*, Communications in Computer and Information Sciences 43 (2009), pp. 143–152.

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

For Peer Review Only

Figure 11. Spaces in the simplified mesh for communal areas.