# We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900
Open access books available

185,000
International authors and editors

200M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities




WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

## Interested in publishing with us?
## Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



11

# THE SELF-ORGANZING APPROACH FOR SURFACE RECONSTRUCTION FROM UNSTRUCTURED POINT CLOUDS

Vilson L. DalleMole, Renata L. M. E. do Rego and Aluizio F. R. Araújo
*Federal Technologic University of Paraná, Federal Institute of Pernambuco, Federal University of Pernambuco*
*Brazil*

## 1. Introduction

Surface reconstruction from a point cloud is very useful in many different application areas, such as manufacturing (Bernardini *et al.*, 1999), cultural heritage (Bernardini *et al.*, 2002) and medicine (Satava & Jones, 1998). Surface reconstructions methods aim to create digital models to reproduce an object shape given a set of points sampled from its surface using 3D scanning technology.

Surface reconstruction starting from a cloud of points is a complex problem which raises a number of challenging issues: Firstly, the connectivity between the vertices must be constructed so that the reconstructed surface has the same topological features as the target surface. However no structural information is available in the input data. Instead the only items of information available are the 3D coordinates of a set of points sampled from the target surface. Secondly, meshes with different resolutions must be generated to fulfil the needs of different applications, otherwise additional processing is required to simplify (or refine) the mesh constructed. Another issue is that the meshes produced must be two dimensional manifolds. Finally, the triangular faces of the mesh should be approximately equilateral.

A lot of research effort has been expended to develop surface reconstruction methods. Some of these methods are based on geometric techniques (Amenta *et al.*, 2001), (Hoppe *et al.*, 1992). Another well known approach is that of dynamic methods (Miller *et al.*, 1991), (Qin *et al.*, 1998), based on the evaluation of energy or force functions. A more recent approach to the problem of surface reconstruction is that of learning-based methods. Learning algorithms are able to process very large and/or noisy data, such as point clouds obtained from 3D scanners and have been used to reconstruct surfaces. Following this approach, some studies (Brito *et al.*, 2008), (Hoffmann & Varady, 1998), (Yu, 1999), have employed Self-Organizing Maps (SOM) and their variants for surface reconstruction. SOM is suitable for the surface reconstruction problem because it can form topological maps to replicate the distribution of input data. In this chapter the authors present two Self-Organizing Maps based on what that they have proposed for surface reconstruction.

The rest of this chapter is structured as follows: Section 2 presents the necessary background for understanding the surface reconstruction methods presented in Sections 4 and 5. The self-organizing approach for surface reconstruction is presented in Section 3, where we discuss some of the self-organizing models useful for surface reconstructions. In Section 4 and 5 we present, respectively, the GSRM and GSOSM methods for surface reconstruction. Section 6 concludes this chapter.

## 2. Background

The geometric pipeline presented in (Saleem, 2003) describes the problem of reconstructing computational models of real object surfaces (called target objects). The first step of the pipeline consists of digitalizing the target object to get a set of points sampled from its surface. When no information about the connectivity among the points is available, or rather, the only items of information available for the reconstruction are the 3D coordinates of the sampled points, the output of the first step of the pipeline is called an unstructured point cloud. When additional information about the connectivity among the points is available, the point could is deemed to be structured. This study does not investigate the techniques available for digitalizing real objects, since the focus here is on the task of reconstruction itself (the second step of the pipeline).

The second step of the pipeline concerns the reconstruction itself, which is, producing a 3D model of a target surface. The input for this step is the point cloud produced in the first step. This study considers unstructured point clouds. Thus, the only information available for the reconstruction are the 3D coordinates of the points sampled from the object surface. No information about the connectivity among the points, which depends on how the object was digitalized, is required by the methods presented here. Thus, these methods do not depend on how the point cloud was acquired. This step outputs a model of the target object surface represented by a polygonal mesh, more specifically a mesh of triangles, because a triangle is the simplest polygon, consequently it is easier to draw a triangle on the computer screen than any other polygon. Section 2.1 presents the polygonal mesh representation.

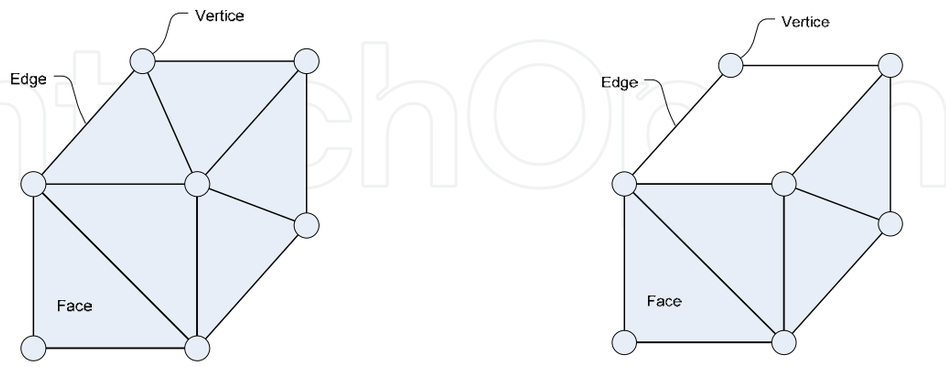
The last step of the pipeline is mesh simplification, that is, producing a simpler mesh (a mesh with less detail) from the mesh output in the second step. Some applications require meshes representing the shape of an object with less detail instead of quite dense meshes which a computer screen is slow to render. For this situation, a mesh simplification step is necessary.

Some desirable features of the reconstruction methods are: (a) To produce meshes of different resolutions so that the simplification step is not necessary; (b) To produce meshes with regular polygons, which in the case of triangles must be approximately equilateral; (c) To produce two-dimensional manifold meshes (2-manifolds); (d) To produce meshes that are a *Delaunay* Tesselation. A brief explanation about two-dimensional manifold and about the *Delaunay* Tesselation are presented in Sections 2.2 and 2.3 respectively.

### 2.1. Polygonal Meshes

Polygonal mesh representation is a particular case of boundary representation (b-rep), in which the faces delimiting the boundary of the solid are polygons, that is, plane figures bounded by a closed path, comprising a finite sequence of straight line segments (edges). A triangle is the simplest polygon, having only three edges. Most modeling and real-time

rendering software use the representation of triangular faces, because this requires less memory, less rendering time, and it adapts to any kind of contour. A surface representation in the form of a mesh of triangles consisting of a set of vertices, edges and triangular faces as illustrated in Fig. 1.




(a) (b)

Fig. 1. Object surface representation using a mesh of triangles. Note that the two upper faces are missing in (b), for this reason the box in (b) seems to be open.

The boundary representation (b-rep) is widely used as a solid modelling technique. However, many b-rep systems support only solids the *boundaries* of which are *2-manifolds*. The next Section gives a brief explanation of two-dimensional manifolds.

### 2.2. Two-Manifolds

A two-dimensional manifold (2-manifold) is a topological space the points of which all have a neighborhood homeomorphic to an open disk (Mäntylä, 1988). In a 2-manifold mesh an edge is regular if it has exactly two coincident faces, and a vertex is regular if it has the same number of edges and faces (Barhak, 2002). If the surface has boundaries, then the mesh also has boundary edges and boundary vertices. The boundary edges have only one coincident face, and at the boundary vertices the number of coincident faces is one less than the number of emanating edges (Barhak, 2002). Thus, in a polygonal mesh, at most two faces can share the same edge. If more than two faces share an edge, the resulting mesh is not a two-dimensional manifold.

### 2.3. Voronoi Diagrams and Delaunay Triangulations

The surface reconstruction problem boils down to creating a mesh establishing its vertices and the connections between them forming triangular faces. For this reason, a geometric construction defining a triangle as the simplex building block is a suitable tool. Many studies (Amenta *et al.*, 2001), (Edelsbrunner *et al.*, 1983) have shown that *Delaunay* triangulations and its dual graph, the Voronoi Diagram, are well suited for surface reconstruction. The *Delaunay* triangulation presents the property of maximizing the minimum angle of the triangles in the mesh, thus avoiding skinny triangles. The reasons why skinny triangles should be avoided are explained in (de Berg *et al.*, 2008). In this section we give a brief introduction to Voronoi diagrams and *Delaunay* triangulation. More details

about these constructions can be found in (de Berg *et al.*, 2008) and (Aurenhammer & Klein, 2000).

Given a space $\mathbb{R}^n$ and a set $S$ of nodes together with a notion of the influence that a node $s_i \in S$ with weight $\boldsymbol{\omega}_{s_i} \in \mathbb{R}^n$ exerts on a point $\boldsymbol{\omega}_k \in \mathbb{R}^n$, then the Voronoi region of $s_i$ consists of all points $\boldsymbol{\omega}_k$ for which the influence of $s_i$ is the strongest, over all $s_j \in S, j \neq i$ (Aurenhammer & Klein, 2000). Therefore, the Voronoi region of a node $s_i$ in a map $S \subset \mathbb{R}^3$ is formally defined by (1) and the Voronoi diagram of the map $S$, denoted by $V(S)$, is a cell decomposition of $\mathbb{R}^3$ in convex polyhedrons. Each Voronoi cell around a node $s_i$ contains all points of $\mathbb{R}^3$ that are closer to $s_i$ than to any other node $s_j$. The complete diagram of the nodes and their Voronoi regions is called Voronoi diagram, as instanced in Fig. 2 (a).

$$VR(s_i) = \{\boldsymbol{\xi} \in \mathbb{R}^3 \mid \|\boldsymbol{\omega}_{s_i} - \boldsymbol{\xi}\| \leq \|\boldsymbol{\omega}_{s_j} - \boldsymbol{\xi}\| \ \forall s_j \in S, i \neq j\} \quad (1)$$

A *Delaunay* Tessellation of $S$, $DT(S)$, is obtained by connecting any two nodes $s_i$ and $s_j$ of $S$ for which a circle $C$ exists that passes through $s_i$ and $s_j$ and does not contain any other node in its interior or boundary. And if there are not four cocircular nodes of $S$, then $DT(S)$ - the dual of the Voronoi diagram $V(S)$ - is a triangulation of $S$, called the *Delaunay* Triangulation (Aurenhammer & Klein, 2000).

It is known that there is a unique *Delaunay* triangulation for $S$, if $S$ is a set of points in a general position; that is, no three points are on the same line and no four are on the same circle, for a two dimensional set of points. For a set of points on the same line there is no *Delaunay* triangulation (in fact, the notion of triangulation is undefined for this case). For 4 points on the same circle (e.g., the vertices of a rectangle) the *Delaunay* triangulation is not unique: clearly, the two possible triangulations that split the quadrangle into two triangles satisfy the *Delaunay* condition. For a set of points $S$ in 3d space, a *Delaunay* triangulation is such that every tetrahedron in $DT(S)$ has an empty circum-sphere. A set of points in n-dimensional Euclidean space is in a general position if no n + 1 points are on the same hyperplane and no n + 2 points are on the same hyper-sphere.

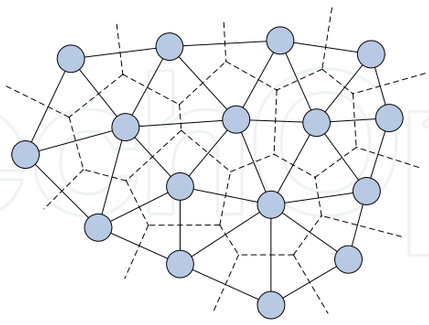


Fig. 2. Voronoi diagram and *Delaunay* triangulation. The dotted lines form the Voronoi diagram and the solid lines forms a *Delaunay* triangulation.

## 3. Using Self-Organizing Maps for Surface Reconstruction.

When self-organizing maps are employed for reconstruction, the mesh vertices correspond to the output nodes of the map. The weight vector of the nodes determines the positions of

the vertices in the mesh, whereas the connections between the nodes correspond to the edges of the mesh.

Self-Organizing Maps, a kind of Artificial Neural Network with unsupervised learning, consist of an input and an output layer. Each input node is connected to every output node. The output nodes may be connected to each other. When self-organizing maps are employed for reconstruction, the input message and the nodes in the output layer have a weight vector representing a 3D coordinate $[x, y, z]^T$. The output layer represents the polygonal mesh being reconstructed: The mesh vertices correspond to the output nodes of the map; the weight vector of the output nodes determines the positions of the vertices in the mesh; the connections between the output nodes correspond to the edges of the mesh. For this reason, the words node and vertex, connection and edge, are used interchangeably.

The faces of a polygonal mesh, however, do not have a direct representative in the Self-Organizing Map. If the connections between the nodes are fixed, as in the Self-Organizing Map (SOM) originally proposed in (Kohonen, 1998), then the faces can be established in advance. The faces of a triangular mesh can also be represented by the basic building block of a SOM variant called Growing Cell Structures (GCS) (Fritzke, 1994). Other SOM variants, such as the Topology Representing Network (TRN) (Martinez & Schulten, 1994) and Growing Neural Gas (GNG) (Fritzke, 1995), do not have a direct representative for the faces of a polygonal mesh.

In the original SOM (Kohonen, 1998) the connections among nodes are defined by their position in the lattice, so that a planar mesh is defined. The learning phase deforms and adjusts this initial mesh to couple with the objective surface. So, at the end of this phase, the distribution of the weight vectors of the nodes in the lattice comes with the surface folds. To obtain this, an important issue is the way in which the samples are presented to the SOM network. The learning phase produces a map that represents the probability density function (pdf) of the input data. For a 2D surface immersed into 3D space, this is one for all points on the surface and zero otherwise. However, this is not sufficient because the scanning phase could produce a point cloud where the occurrence of each sample is not equal. Some surface parts could be over sampled resulting in a point cloud in which some samples are repetitive, or which has a lot of very similar samples. Therefore a pre-processing step is need. Moreover, to avoid the effects of over-fitting nodes, the samples should be presented to the Network in a random sequence. Another relevant issue is the size of the lattice which has to be specified in advance. This implicitly specifies the level of detail preserved by the reconstructed surface model. To specify a suitable size, one should consider the total area of the surface and the size of the triangles in the mesh in order to preserve the richness of the details as required. Despite all these achievements, SOM has displayed difficulties in reconstructing concave regions. That is, to say, after the learning process has taken place, some vertices or triangles may be unstable, and dangle among regions densely populated by the data.

As in SOM the TRN model begin with a pre-defined number of nodes. However, the TRN initial map does not define an initial mesh as in SOM. TRN is a combination of Neural Gas (NG) (Martinez & Schulten, 1991) and Competitive Hebbian Learning (CHL) (Martinez, 1993). The former distributes the nodes according to the probability density function of input data, while the latter builds a mesh connecting these nodes in a fashion that reproduces the topology of the objective surface. As the connections between nodes are

defined later, after the nodes have been placed next to their final position, the learning process is improved.

A solution for the limitation represented by size of the of network, which in SOM and TRN needs to be specified in advance, is found in Growing models such as (GCS) and (GNG). These models build maps that grow incrementally as the samples are presented. The GCS generates maps consisting only of a basic building blocks type, triangles for surface reconstruction. As opposed to GCS, a map generated by GNG may have nodes with different connectivity and the topology may have different dimensionalities in different parts of the map. Like the TRN, the GNG is able to learn the topology of input data through Competitive Hebbian Learning. Hence, GNG can be seen as a GCS variant without its topological restrictions or as a growing version of TRN.

The Neural Mesh (NM) (Ivrissimtzis *et al.*, 2004a) is a growing model that starts from an initial mesh forming a tetrahedron and grows or diminishes by splitting the more active vertex or removing the least active vertex. This is a probabilistic algorithm in which the output is dependent on the presentation sequence of the samples. Thus, an ensemble combining several reconstructions into a single one, may be necessary in order to reach a quality reconstruction. The most undesirable NM limitation is the absence of any guarantee that a sufficiently good point cloud will result in a correct and faithful reconstruction (Ivrissimtzis *et al.*, 2004b). In particular, the algorithm has difficulties in distinguishing between two close sheets.

Growing Self-Reconstruction Meshes (GSRM) (do Rêgo *et al.*, 2007, 2009) extend Growing Neural Gas (Fritzke, 1995) by including the concept of triangular faces in the learning algorithm and additional conditions in order to include and remove connections, so that it can produce a triangular 2-manifold mesh representation of a target object given an unstructured point cloud of its surface. The main modifications concern Competitive Hebbian Learning, the vertex insertion operation and the edge removal mechanism. GSRM is able to reproduce the shape of the original object by learning the geometry and topology of the surface represented in the point cloud. Also, GSRM generates meshes with different resolutions during the learning process.

The Growing Self-Organizing Surface Map (GSOSM) (DalleMole & Araújo, 2008a, 2008b, 2009) aims to produce a mesh of equilateral triangles faithful to the object surface. The mesh is constructed in a growing fashion by a learning process starting from a cloud of points sampled on the object surface. The learning process inserts nodes and connections to represent the vertices and the edges of triangles of the mesh aimed at. The insertion of nodes considers an empty receptive field whereas the triangles edges are learned using a new learning rule called Competitive Connection Hebbian Learning (CCHL). This rule considers the three nodes closest to each input sample as candidate to form a triangle. Thus, for each input sample, CCHL inserts only one new edge. The approximate equilaterally of triangles is obtained as a consequence of the adaptation steps of the learning process. The final mesh is generally a two-manifold mesh without holes.

The last two models are novel and attempt to overcome several of the limitations of their predecessors, for example, dependence on cycle counters and error accumulators (GSOSM); topology learning and multi-resolution meshes (GSRM). These models are presented in detail below.

## 4. Growing Self-Organizing Meshes (GSRM)

Growing Self-Organizing Meshes is a surface reconstruction method based on growing self-organizing maps, which learns both the geometry and the topology of the input data set. GSRM produces 2-manifold meshes that approximate very well the shape of an object, including its concave regions and holes, if any. The maps produced with GSRM grow incrementally producing meshes of different resolutions, according to different application needs. Other important features of GSRM are that the triangular faces of the meshes are approximately equilateral and most of the edges satisfy the local criterion of the *Delaunay* triangulation of piecewise flat surfaces (Bobenko & Springborn, 2007), (Fisher *et al.*, 2007).
The GSRM has GNG as its starting point and modifies the following steps: node insertion, removal of nodes and faces, creation of nodes and faces. The modification aims at producing 2-manifold meshes of triangles, since standard GNG maps do not represent the faces of a polygonal mesh. The steps mentioned above are explained in Section 4.3 to 4.5. Section 4.1 presents the steps of the GSRM learning algorithm. Section 4.2 discusses the parameters of the algorithm. Section 4.6 presents the topological learning step performed to complete the topological learning after the desired mesh resolution is reached. Section 4.7 presents the post-processing step necessary to complete the triangulation of the mesh output by GSRM. Some experimental results of the GSRM reconstruction are presented in Section 4.8.

### 4.1. The GSRM Algorithm

The GSRM learning algorithm is briefly described below:

1. Initialize the map $A$ with three nodes with weight vectors randomly chosen from a point cloud $P$.
2. Present a sample $\boldsymbol{\xi}$, randomly chosen from $P$.
3. Find the two nodes ($s_1$ and $s_2$) of the map that are nearest to $\boldsymbol{\xi}$ according to the Euclidean distance.
4. Create connections and faces according to the ECHL (Section 4.3). Sometimes connections are reinforced instead of created. In this case, other edges are checked against a condition for edge removal based on the Thales Sphere concept (Section 4.4).
5. Update the error counter of node $s_1$:

$$\Delta E_{s_1} = \left\| \boldsymbol{\xi} - \boldsymbol{\omega}_{s_1} \right\| \tag{2}$$

   where, $\boldsymbol{\omega}_{s_1}$ is the weight vector of node $s_1$.
6. Adapt the weight vector of the winner node $s_1$ and its neighbors.

$$\Delta \boldsymbol{\omega}_{s_1} = \varepsilon_b \left\| \boldsymbol{\xi} - \boldsymbol{\omega}_{s_1} \right\| \tag{3}$$

$$\Delta \boldsymbol{\omega}_{s_n} = \varepsilon_n \left\| \boldsymbol{\xi} - \boldsymbol{\omega}_{s_n} \right\|, s_n \in N(s_1) \tag{4}$$

   where $\mathrm{N}(\mathrm{s}_1)$ is the neighborhood of $\mathrm{s}_1$.
7. Update the age of all edges $e$ emanating from $s_1$.

$$age_e = age_e + 1 \tag{5}$$

8. Remove the faces coincident to an old edge $e$ ($age_e > age_{max}$) and remove this edge (Section 4.4).
9. If the number of samples presented so far is greater than $\lambda$ , insert a new node in the map (Section 4.5).
10. Decrease the error variables of all nodes:

$$\Delta E_s = -\beta E_s \text{ , } \forall s \in A \tag{6}$$

11. If the map has achieved the desired resolution, complete the topological learning (Section 4.6).

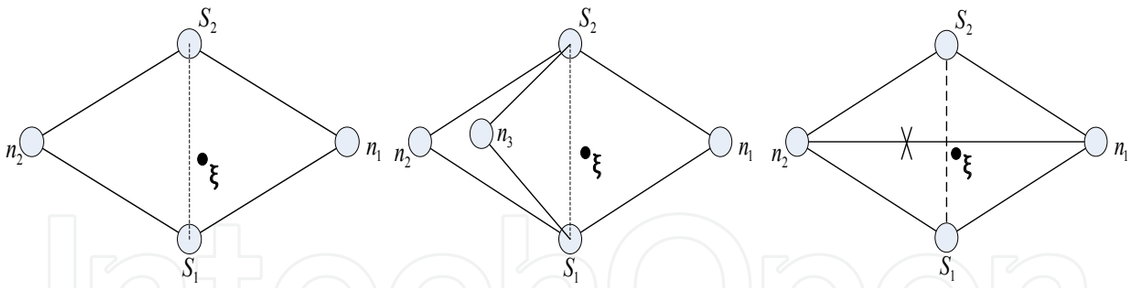
### 4.2. Parameters

The GSRM learning algorithm relies on six parameters: $\varepsilon_b$ and $\varepsilon_n$ – learning rate of the winner node and its neighbors, respectively ($\varepsilon_b > \varepsilon_n$); $\lambda$ – the frequency at which a new node is inserted; $\alpha$ – the error reduction rate of the nodes that are neighbors of a node that has just been inserted; $\beta$ – the error reduction rate that aims at stressing the impact of recently accumulated errors ($\beta < \alpha$); $age_{max}$ – the maximum age at which an edge can be removed.

### 4.3. Creation of Edges and Faces

The mechanism that GSRM uses for the creation of edges and faces is an extension of standard Competitive Hebbian Learning (CHL) that we call Extended Competitive Hebbian Learning (ECHL). The goal is to extend the CHL so that instead of defining only edges, it can also define the faces of a triangular mesh representation. Moreover, ECHL avoids the creation of overlapping edges and non-manifold meshes. The ECHL algorithm is described below:

1. Present a randomly chosen sample **ξ**.
2. Select the nodes of the map $s_1$ and $s_2$ that are nearest to **ξ** according to the Euclidian distance. See Fig. 3 (a) for an example.
3. If $s_1$ and $s_2$ are not connected by an edge, create such an edge, since $s_1$ and $s_2$ do not have more than two neighbors in common. This condition avoids more than two faces incident on the same edge, so that the mesh remains a 2-manifold Fig. 3 (b) illustrates what would happen without this condition).
   3.1. If $s_1$ and $s_2$ have two common neighbors ($n_1$ and $n_2$) connected by an edge, this edge is removed together with its coincident faces. This step avoids overlapping edges. In the example of Fig. 3 (b) the edge between $n_1$ and $n_2$ would be removed, so the it does not overlap that new edge $e_{s_1,s_2}$.
   3.2. Create new face(s) comprising $s_1$, $s_2$ and each of their common neighbors, if any. See Fig. 3 (c) for an example.
4. Otherwise
   4.1. Reinforce the existing edge $e_{s_1,s_2}$ by setting its age to zero. This reduces the possibility of the edge connecting $s_1$ and $s_2$ being removed.
   4.2. Check if some edge emanating from $s_1$ against the edge removal condition based on the Thales sphere concept (Section 4.4).




(a) (b) (c)

Fig. 3. Extended Competitive Hebbian Learning: a) The nearest nodes $s_1$ and $s_2$ for the sample $\boldsymbol{\xi}$ and its common neighbors $n_1$ and $n_2$; b) The manifoldness of the mesh is broken, if the connection between the nodes $s_1$ and $s_2$ is created; c) The sample $\boldsymbol{\xi}$ induces the establishment of a connection between the nodes $s_1$ and $s_2$ crossing the connection between the nodes $n_1$ and $n_2$ which is removed.

### 4.4. Removal of Edges and Faces

GSRM uses two schemes to decide whether an edge is obsolete and should be removed. The first is the edge ageing scheme used in the standard GNG. The second is based on the Thales sphere concept, and it is also used in GSOSM. The edge ageing scheme removes an edge when its age becomes greater than a given threshold ($age_{max}$). The age of an edge is zero when it is created, and increases as described in step 7 of the algorithm presented in Section 4.1. The age of an edge can be reset, as described in step 4.1. of the algorithm presented in Section 4.3.

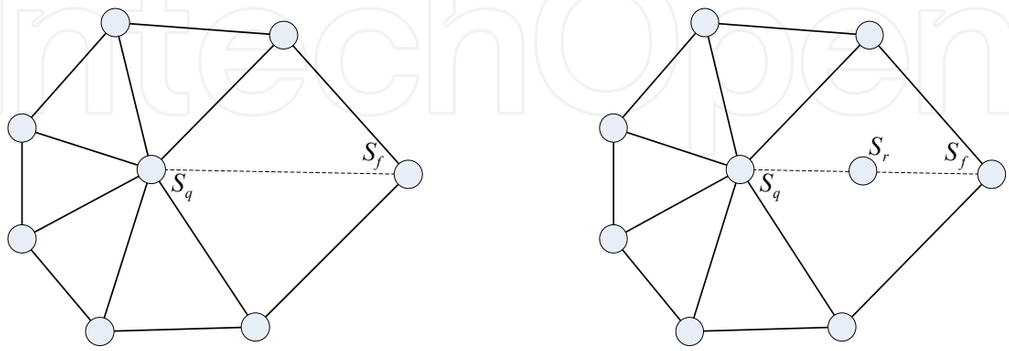
As the position of the nodes change during the learning process, two connected nodes may become distant from each other, implying that the mesh has long edges. These long edges should be removed; otherwise, the algorithm fails to meet the goal of having triangular faces which are approximately equilateral. To remove these long edges an edge removal scheme based on the ideas of (Jockusch & Ritter, 1993) is applied. According to this scheme, whenever a connection to be generated by a sample presentation already exists, the edges connecting the winner node $s_1$ to each of their neighbors $s_k$ are considered candidates for removal. A connection between $s_1$ and $s_k$ is removed when the second winner node $s_2$ is inside the Thales sphere with a diameter of $\|\boldsymbol{\omega}_{s_1} - \boldsymbol{\omega}_{s_k}\|$, where $\boldsymbol{\omega}_{s_1}$ and $\boldsymbol{\omega}_{s_k}$ are respectively the weight vectors of the nodes $s_1$ and $s_k$. To verify this condition the angle between the vectors $\mathbf{v} = \boldsymbol{\omega}_{s_1} - \boldsymbol{\omega}_{s_k}$ and $\mathbf{u} = \boldsymbol{\omega}_{s_k} - \boldsymbol{\omega}_{s_1}$ is determined. If this angle is greater than $\pi/2$, then the condition is satisfied and the edge between $s_1$ and $s_k$ is removed (Fig. 11 (a) illustrates this situation). Otherwise, the condition is false.

If an edge must be removed, either because of its age or because of the Thales sphere based condition, the removal of this edge is performed after the removal of every incident face. After the removal of the edge, the vertices of the newly removed edge may not have an edge emanating from it, and must be removed as well.

### 4.5. Node Insertion

A new node is always inserted in the map after a number $\lambda$ of adaptation steps. The new node $s_k$ is inserted between the node $s_q$ with the highest error counter and its neighbor $s_f$ with highest error counter, this means that the weight vector of $s_k$ is initialized as: $\boldsymbol{\omega}_{s_k} =$

$0.5(\boldsymbol{\omega}_{s_q} + \boldsymbol{\omega}_{s_f})$. The edge connecting $s_q$ to $s_f$ is removed together with its coincident faces. New edges connecting $s_k$ to $s_q$ and $s_k$ to $s_f$ are created. Fig. 4 illustrates this process: (a) it presents a mesh before insertion of the node, and (b) it presents the same mesh after the node has been inserted. The error counters of nodes $s_q$ and $s_f$ are decreased according to parameter $\beta$ ($\Delta E_s = -\beta E_s$). Finally, the error counter of the new node is interpolated from the error counter of $s_q$ and $s_f$ ($E_{s_r} = 0.5(E_{s_q} + E_{s_f})$).




(a) (b)

Fig. 4. Example of a mesh before (a) and after (b) a new node insertion.

### 4.6. Topology Learning

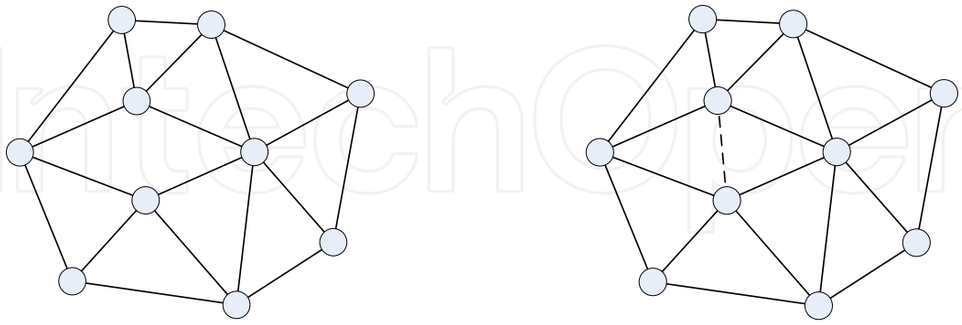
The GSRM learning algorithm behaves non-deterministically. Some edges needed for the reconstructed mesh may not have been created when the learning process finishes, because no sample that would trigger these edges was presented at the appropriate moment of the learning process. Also, some long edges may not have been removed since the Thales sphere based condition to verify if an edge is long enough and must be removed is not checked at each iteration, but only periodically. Thus, to complete the topological learning, a deterministic topological learning step is performed after the main learning processing is finished. At this moment, the vertices no longer change their position (weight vectors). First, every edge is checked against the Thales sphere based condition to verify if any edge is too long and must therefore be removed. Secondly, every sample is presented so that, all the necessary edges, that is, connections between nodes, are created. Note that since the vertices are no longer changing their positions, no edge becomes unseful during this final topological learning step.

### 4.7. Post Processing Step

After the topology learning step, some approximately regular polygons remain untriangulated because the two winner nodes for any sample internal to these polygons are already connected (see Fig. 8). In our experiments, the vast majority of the untriangulated polygons had four or five vertices, a few of them had six vertices, and none of them had more than six vertices. The post processing step aims at triangulating these polygons.

As to the quadrilaterals, triangulation is performed by creating one of their edges and replacing the quadrilateral with two triangles. See Fig. 5 for an example. The diagonal chosen is the one that makes the sum of the opposite angles in the adjacent triangles less than $\pi$. This choice leads to edges that satisfy the local *Delaunay* condition for piecewise flat

surfaces (Bobenko & Springborn, 2007), (Fisher *et al.*, 2007). The triangulation of polygons with more than four vertices is performed by inserting a new vertex at the geometric center of the polygon and connecting this new vertex to each of the vertices of the polygons. See Fig. 6 for an example.



(a) (b)

Fig. 5. Quadrilateral before (a) and after (b) triangulation.

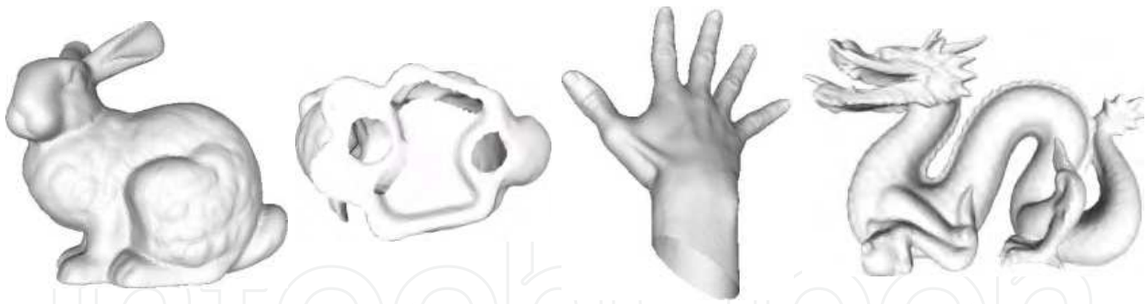



(a) (b)

Fig. 6. Pentagon before (a) and after (b) triangulation.

### 4.8. Experimental Results

This Section presents some reconstructions produced by GSRM. Fig. 7 shows pictures of three synthetic objects: Bunny and Dragon available at the Stanford Repository (graphics.stanford.edu/data/3Dscanrep), and Hand, donated by Ioannis Ivrissimtzis (Ivrissimtzis *et al.*, 2004). All the reconstructions presented in this Section have about 20,000 vertices. Note that the shape of the objects is represented in the GSRM reconstructions as are their concave regions and holes (Fig. 7 (b)).



(a) (b) (c) (d)

Fig. 7. Example of two reconstructions produced with GSRM: a) and b) are respectively a lateral view and a view of the underside of reconstructed version of the Bunny model; c) and d) lateral views of reconstructed versions of the Hand and the Dragon models.

The numerical results concern: distance from the target object, valence distribution, polygon conformity and the number of *Delaunay* edges. The Hausdorff distance between the target and the reconstructed objects was calculated with the Metro tool (Cignoni *et al.*, 1998). Valence distribution refers to the number of neighbors of the mesh vertices. Valences must be distributed as evenly as possible, i.e., ideally all vertices should have the same valence. Polygon conformity R(P) is measured as the ratio between the smallest and the largest distance of its vertices (s) to the polygon baricenter (bp). The *Delaunay* condition of piecewise flat surfaces presented in (Bobenko & Springborn, 2007) has been applied to verify the number of valid *Delunay* edges of the GSRM reconstructions. Table 1 shows the following measurements: the Hausdorff Distance, the average polygon conformity and the percentage of *Delaunay* edges of the meshes reconstructed with GSRM. The average percentage of the valences presented by the vertices of the meshes reconstructed with GSRM is shown in Table 2.

| Model | Hausdorff Distance | Polygon Conformity (avg) | Delaunay Edges (%) |
|---|---|---|---|
| Bunny | 0.001510 | 0.688785 | 99.86 |
| Dragon | 0.023644 | 0.684732 | 99.81 |
| Hand | 0.001364 | 0.689153 | 99.83 |

Table 1. The Hausdorff distance, the average polygon conformity and the percentage of *Delaunay* edges of GRSM reconstructions.

| Model | Valence Distribution (%) | | | | | |
|---|---|---|---|---|---|---|
| | 4 | 5 | 6 | 7 | 8 | others |
| Bunny | 4.32 | 25.76 | 42.49 | 21.87 | 4.76 | 0.8 |
| Dragon | 5.55 | 25.66 | 41.10 | 22.15 | 5.38 | 1.15 |
| Hand | 4.27 | 25.7 | 42.45 | 25.03 | 4.88 | 0.66 |

Table 2. Valence distribution of the vertices of GSRM reconstructions.

## 5. Growing Self-Organizing Surface Map (GSOSM).

This model overcomes several restrictions of its predecessors, the mapping process is incremental and there are no dependences on error accumulators or cycle counters. Moreover, the model is able to handle correlated samples and disjoint close surface parts and to recover a surface representation from multiple subsets of an entire point cloud blending the surface parts automatically, in response to the presentation of data. However, GSOSM is not suitable for producing surface representation with different levels of detail. The GSOSM learns the surface starting from an empty map and inserts nodes and connections between them, thus producing a mesh of equilateral triangles. This mesh is the surface representation and, therefore, the preservation of richness of detail is dependent only on the length of the edges of the triangles.

### 5.1. The GSOSM Algorithm

The GSOSM algorithm comprises nine steps:

1. Parameter setup;
2. Presentation of a new sample;
3. Determination of the three closest nodes to the current sample;
4. Insertion of a new node;
5. Weight vector update;
6. Node collapse;
7. Determination of the connection that captures the current sample;
8. Insertion or substitution of connection;
9. Removal or swap of connections.

### 5.2. Parameters

The GSOSM has four parameters: $e_{max}$ - defines the size of edges of triangles; $\alpha$ - the learning rate; $\theta_{min}$ - specifies the largest value allowed for an internal angle of a triangle; and $\beta_{min}$ – represents a constraint to connection insertion and is employed to distinguish between parallel close sheets of surface.

### 5.3. The Nodes Insertion Operator

A GSOSM reconstruction starts with an empty map $S$ and the first node is inserted with the first input signal. A spherical receptive field with a radius of $e_{max}$ is considered for each node and a new node is inserted only if the input signal $\boldsymbol{\xi}$ is out of any receptive field. When a new node $s_{new}$ is inserted its weight vector $\boldsymbol{\omega}_{s_{new}}$ is equal to the current sample $\boldsymbol{\xi}$, and the node does not have any initial connection.

### 5.4. The Adaptation Operator

The objective of the adaptation is to adjust the map to the surface and consists of two distinct sub-operations. The first is the original SOM adaptation step (7) and moves the winner node towards the input signal $\boldsymbol{\xi}$ according to the learning rate α. This step is applied only if the three nodes closest to the input signal are not connected and do not form a triangle. The second sub-operation (8) rotates the triangle about the axis defined by the second and third closest nodes, towards the input signal.

$$\boldsymbol{\omega}_{s_0} = \boldsymbol{\omega}_{s_0} + \alpha\left[\boldsymbol{\xi} - \boldsymbol{\omega}_{s_0}\right] \tag{7}$$

$$\boldsymbol{\omega}_{s_0} = \boldsymbol{\omega}_{s_0} + \alpha\, \mathbf{P}_n\, d(P, \boldsymbol{\xi}) \tag{8}$$

where: $\mathbf{P}_n$ is the normal vector of plane $P$ formed by the three closest nodes for $\boldsymbol{\xi}$, and $d(P, \boldsymbol{\xi})$ is the distance of $\boldsymbol{\xi}$ to the plane $P$.
The original SOM adaptation operator applied to correlated and repetitive samples pulls the winner node towards the second closest node undoing the equilateral triangles. However, this does not occur when the schema above is employed. Therefore, the GSOSM is able to handle repetitive samples.

### 5.5. The Merging Operator

The operator of insertion of new nodes ensures a minimal distance, greater than $e_{max}$, between any two nodes. However, all nodes are subject to being repositioned by the adaptation operator. As a consequence, a repositioned node can invade the receptive field of another node. Thus, the GSOSM applies the merging operator and transforms these nodes into one that is positioned midway between them.

### 5.6. The CCHL Learning Rule

The GSOSM introduces a very nice learning rule named Competitive Connection Hebbian Learning (CCHL). This new learning rule was proposed to substitute the CHL (Martinez & Schulten, 1991) as the rule to learn the node-neighborhood relationships among nodes. The new rule overcomes the failure of CHL to complete a triangulation inside polyhedrons such as a quadrilateral as illustrated in the Fig. 8.

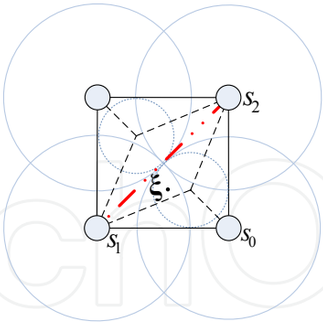
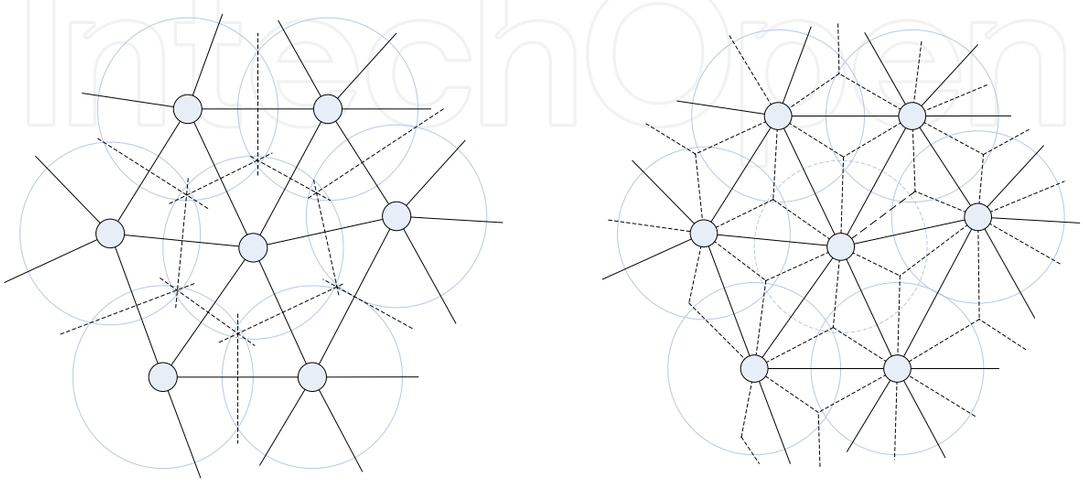



Fig. 8. A case where CHL Learning rule fail to complete the triangulation.

The CHL rule performs a competition among all nodes and the two nodes closest to the current sample are selected to be connected. Now consider four interconnected nodes forming a quadrilateral without any diagonal and an input signal $\boldsymbol{\xi}$ within it (Fig. 8). In this configuration, the most activated nodes are $s_0$ and $s_1$. Therefore, CHL would insert a connection between them. However, this is an existing link and to complete the local triangulation a new connection should be inserted between $s_1$ and $s_2$, to form two triangles. The pentagon and hexagon are similar cases.

The basis of the CCHL learning rule is Voronoi diagrams, that is a cell decomposition of the space into convex polyhedrons. Similar to the Voronoi cell of a node (Fig. 9 (a)), each Voronoi cell around a connection $c_{s_i s_j}$ contains all points that are closer to $c_{s_i s_j}$ than to any other $c_{s_k s_l}$. A sample of a diagram of connection Voronoi cells $V^c(S)$ in the $\mathbb{R}^2$ space is shown in Fig. 9 (b). If all triangles are equilateral, then $V^c(S)$ is equal to the second order Voronoi diagram.



(a) (b)

Fig. 9. $\mathbb{R}^2$ Voronoi Diagrams: a) Node receptive field Voronoi cells; b) Connection Voronoi cells.

In CCHL, the competition for the connection insertion has two distinct phases. The first one determines the three nodes closest to the current sample. The second phase considers the three nodes as vertices of an imaginary triangle and the closest edge to $\boldsymbol{\xi}$ is the connection that should be inserted, if it does not already exist. So, the CCHL can be stated as: given an input signal $\boldsymbol{\xi}$ and its three closest nodes $\{s_0, s_1, s_2\}$, the pair to be connected is that forming the edge closest to $\boldsymbol{\xi}$.

Return to Fig. 8, notice the stippled lines denoting the Connection Voronoi Cells and the winner connection linking the nodes $s_1$ and $s_2$. This connection split the lozenge into two triangles, thus completing the local triangulation. Nonetheless, CHL would recommend inserting the connection between nodes $s_0$and $s_1$, thus not completing the triangulation.

### 5.7. The Connection Insertion Operator

In the GSOSM, the mesh that represents the surface is implicitly defined by the nodes and its connections. The CCHL rule is used to decide the connection $c_{new}(s_i, s_j)$ that should be inserted. However, the GSOSM imposes three other conditions that need to be satisfied before creating a connection.

1) The coefficient of correlation $\beta$ (9) between the activation of $s_i$ and $s_j$ needs to satisfy (10). This condition is used to verify the continuity of the input space between the nodes $s_i$ and $s_j$.

$$\beta = \frac{(\boldsymbol{\xi} - \boldsymbol{\omega}_{s_i}).(\boldsymbol{\omega}_{s_j} - \boldsymbol{\omega}_{s_i})}{\left\|\boldsymbol{\omega}_{s_j} - \boldsymbol{\omega}_{s_i}\right\|^2} \quad , \quad \left\|\boldsymbol{\xi} - \boldsymbol{\omega}_{s_i}\right\| \leq \left\|\boldsymbol{\xi} - \boldsymbol{\omega}_{s_j}\right\| \tag{9}$$

$$\beta_{min} \leq \beta \leq 0.5 \qquad \beta_{min} \in [0, 0.5] \tag{10}$$

As the projection of ξ approximates to the connection midpoint (Fig. 10) the probability of inserting a connection crossing an empty area between two close sheets vanishes.

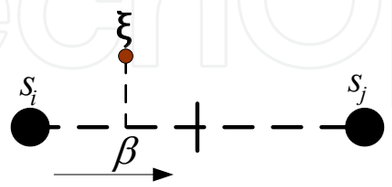



Fig. 10. Visualization of coefficient $\beta$ as a projection of the input signal over the winner connection.

2) The coefficient of similarity $\theta$ (11) between $s_i$ and $s_j$ with respect to node $s_k$ is greater than or equal to $\theta_{min}$. Geometrically, $\theta$ represents the cosine of the angle between vectors $\mathbf{v} = \boldsymbol{\omega}_{s_i} - \boldsymbol{\omega}_{s_k}$ and $\mathbf{u} = \boldsymbol{\omega}_{s_j} - \boldsymbol{\omega}_{s_k}$, with $s_i$ and $s_j$ being the nodes in the extremities of the winner connection and $s_k$ the other closest node. This condition assures the formation of triangles that are approximately equilateral.

$$\theta = \frac{\boldsymbol{\omega}_{s_i} - \boldsymbol{\omega}_{s_k}}{\left\|\boldsymbol{\omega}_{s_i} - \boldsymbol{\omega}_{s_k}\right\|} \cdot \frac{\boldsymbol{\omega}_{s_j} - \boldsymbol{\omega}_{s_k}}{\left\|\boldsymbol{\omega}_{s_j} - \boldsymbol{\omega}_{s_k}\right\|} \tag{11}$$

3) The winner connection $c_{win}(s_i, s_j)$ does not cross the Voronoi region of any other connection, thus avoiding the creation of overlapping triangle faces. The crossings can be found using the CCHL to verify if the sample midpoint of $c_{win}$ is within a Voronoi region of another connection $c_{s_k s_t}, t \neq i, j$. If an overlap is detected the value of the coefficient $\phi$ of each overlapping connection is used to decide which should be removed. The coefficient $\phi$ (12) is a measure of the adherence of the connection to the objective surface and is obtained as follows:

$$\phi = \left(1 + |\beta - 0.5| + \frac{d(c_{s_i s_j}, \boldsymbol{\xi})}{\left\|\boldsymbol{\omega}_{s_i} - \boldsymbol{\omega}_{s_j}\right\|}\right)^{-1} \tag{12}$$

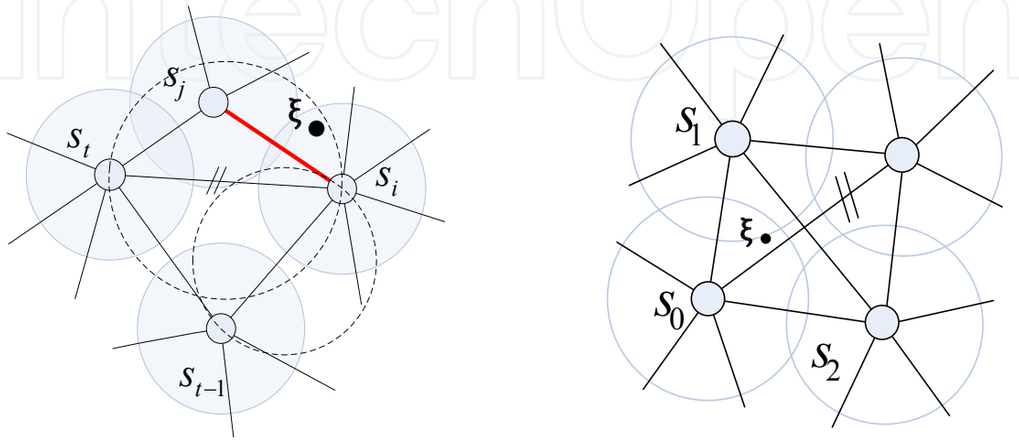
with $$d\left(c_{s_i s_j}, \boldsymbol{\xi}\right) = \left\|\left[\boldsymbol{\omega}_{s_i} + \beta\left[\boldsymbol{\omega}_{s_j} - \boldsymbol{\omega}_{s_i}\right]\right] - \boldsymbol{\xi}\right\|$$

where: $\boldsymbol{\omega}_{s_i}$ and $\boldsymbol{\omega}_{s_j}$ are the weight vectors of the nodes at the extremities of the connection $c_{s_i s_j}$.

### 5.8. The Connection Removal Operator

The connection removal operator in GSOSM consists of two steps. The first removes or swaps connections that could produce triangles with an internal $\theta$ angle the cosine (13) of which is lower than the parameter $\theta_{min}$ as shown in Fig. 11 (a).

$$\frac{\boldsymbol{\omega}_{s_i} - \boldsymbol{\omega}_{s_j}}{\left\|\boldsymbol{\omega}_{s_i} - \boldsymbol{\omega}_{s_j}\right\|} \cdot \frac{\boldsymbol{\omega}_{s_t} - \boldsymbol{\omega}_{s_j}}{\left\|\boldsymbol{\omega}_{s_t} - \boldsymbol{\omega}_{s_j}\right\|} < \theta_{min} \tag{13}$$




(a) (b)

Fig. 11. The process of detection and removal or swap of connections: a) long connections; b) crossing connections forming overlapping triangles.

The second step is used to eliminate overlapping triangle faces. The removal of cross connections is carried out using the CCHL to verify if the midpoint of $c_{win}$ is within another connection Voronoi region, as discussed earlier. If an intersection is detected (Fig. 11(b)), the operator removes the connection with the smallest $\phi$ value (12).

### 5.9. Experimental Results

In this Section, numerical and visual results of GSOSM reconstructions are reported. The models used come from the Stanford 3D Scanning Repository (graphics.stanford.edu/data /3Dscanrep) and the AIM@Shape Repository (www.aimatshape.net). The general quality of GSOSM reconstructions can be visually accessed in Fig. 12 which shows views of four reconstructed object models. Notice that GSOSM preserves the details of the original surface in the reconstructed version (Fig. 12 (a) and (d)) and the suavity of the curves of the surface (Fig. 12 (b)). Moreover, the ability of GSOSM in reproducing close sheets correctly is showed in the Fig. 12 (c). The equilaterality quality of the triangles in a mesh produced by GSOSM can be inspected visually at Fig. 13.

Table 3 shows measurements on the quality of the meshes produced by GSOSM. The columns under the caption "Valence" show that the meshes produced are very regular with each vertex being a centre of a hexagon. The next column shows that GSOSM is able to produce a mesh in which most of its edges are valid *Delaunay* edges. The last columns, under the caption "Equilaterality" show the measured quality of the triangles in terms of

their equilaterality. These values were calculated using the equilaterality quality measure defined by Rypl (2005), that is.

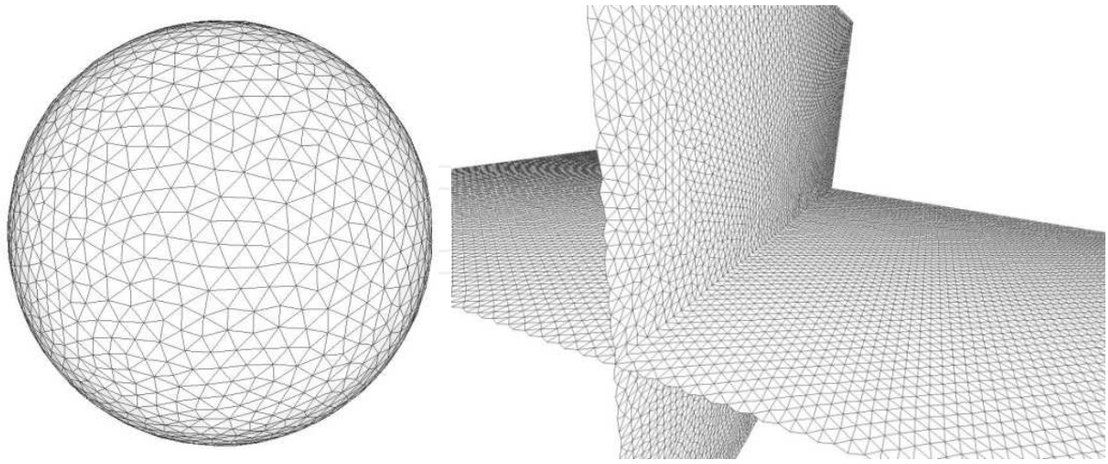
$$q = f\frac{A}{a^2 + b^2 + c^2} \tag{14}$$

where, A represents the area of the triangle, a, b and c are the lengths of its edges and $f = 4\sqrt{3}$ is a normalizing coefficient which justifies the quality of an equilateral triangle to one.



(a) (b) (c) (d)

Fig. 12. Views of GSOSM object surface reconstructions: a) the Happy Buddha; b) the Bust; c) the Gips Hand; d) the Raptor;



(a) (b)

Fig. 13. Visual inspection of the equilaterality quality of the triangles: a) a sphere; b) a non 2-manifold surface.

Notice that the column "min" indicates the existence of some skinny triangles, this is because of the coefficient $\phi$ which forces the permanence of long connections if and when they are fitted to the objective surface fold. However, the following three columns show that high quality triangles are the great majority and that the skinny triangles are exceptions.

| Object | Triangles | | | | | | |
|---|---|---|---|---|---|---|---|
| | Valence | | % invalid Delaunay Edges | Equilaterality (q) | | | |
| | avg. | std. dev. | | min. | max. | avg. | std. dev. |
| Bunny | 5.990 | 0.724 | 1.53 | 0.216 | 1.000 | 0.936 | 0.058 |
| Asian Dragon | 5.999 | 0.718 | 1.51 | 0.210 | 1.000 | 0.933 | 0.067 |
| Dragon | 5.999 | 0.766 | 2.50 | 0.181 | 1.000 | 0.931 | 0.073 |
| Happy Buddha | 5.999 | 0.759 | 2.30 | 0.208 | 1.000 | 0.933 | 0.071 |
| Armadillo | 5.999 | 0.772 | 2.66 | 0.232 | 1.000 | 0.930 | 0.074 |
| Lucy | 5.999 | 0.667 | 1.21 | 0.178 | 1.000 | 0.945 | 0.061 |
| Thai Statue | 5.998 | 0.712 | 1.62 | 0.160 | 1.000 | 0.940 | 0.065 |
| Filigree | 6.002 | 0.688 | 1.80 | 0.187 | 1.000 | 0.938 | 0.064 |
| Bust | 5.999 | 0.788 | 2.73 | 0.236 | 1.000 | 0.927 | 0.078 |
| Hand | 5.996 | 0.805 | 2,59 | 0.204 | 1.000 | 0.928 | 0.078 |
| Gips Hand | 5.995 | 0.770 | 2.34 | 0.219 | 1.000 | 0.933 | 0.070 |

Table 3. Statistics on triangles from GSOSM reconstructions.

The results reported in Table 4, were computed using the Metro tool (Cignoni *et al*., 1998) and are indicative of the fidelity of GSOSM reconstructions to the original object surfaces. In this table, $\mathcal{M}_1$ refers to the reconstructed mesh while $\mathcal{M}_2$ is the original mesh. The labels are: "max" for the maximum calculated value of the distance between the surfaces $\mathcal{M}_1$ and $\mathcal{M}_2$; "avg" for the mean distance; and "rms" for the root mean square. The column "H" is the Hausdorff distance, which is equal to the greatest maximum value.

| Model | Metro results (x 10-3) | | | | | | |
|---|---|---|---|---|---|---|---|
| | $\mathcal{M}_1 \rightarrow \mathcal{M}_2$ | | | $\mathcal{M}_2 \rightarrow \mathcal{M}_1$ | | | H |
| | max | avg | Rms | max | avg | rms | |
| Bunny | 1.419 | 0.040 | 0.064 | 1.490 | 0.043 | 0.078 | 1.490 |
| Dragon | 1.348 | 0.018 | 0.031 | 1.807 | 0.023 | 0.119 | 1.807 |
| Happy Buddha | 1.389 | 0.024 | 0.042 | 1.255 | 0.034 | 0.146 | 1.389 |
| Armadillo | 0.870 | 0.025 | 0.037 | 0.925 | 0.026 | 0.040 | 0.870 |
| Filigree | 1.180 | 0.019 | 0.043 | 1.827 | 0.021 | 0.050 | 1.827 |
| Bust | 0.788 | 0.019 | 0.031 | 1.135 | 0.020 | 0.034 | 1.135 |
| Gips Hand | 0.646 | 0.022 | 0.033 | 1.392 | 0.025 | 0.038 | 1.392 |
| Hand | 0.909 | 0.011 | 0.017 | 1.588 | 0.011 | 0.021 | 1.588 |

Table 4. Metro measures of the reconstructions of models using GSOSM.

## 6. Conclusions and Remarks

This chapter presented the self-organizing approach to solve the surface reconstruction problem. The features of the Self-Organizing Map and of some of its well known variants, such as the Topology Representing Networks, the Growing Cell Structures and the Growing Neural Gas, were presented. Also, two novel SOM based models were presented – GSRM and GSOSM - which have been proposed by the authors and overcome several drawbacks of their predecessors.

GSRM is a self-organizing based surface reconstruction method that profits from the topology learning ability and incremental growth of Growing Neural Gas, while, at the same time, it also modifies some of the standard GNG operations to meet some requirements of the surface reconstruction method, for example: to produce 2-manifold meshes of triangles and to have faces that are approximately equilateral. The main advantages of GSRM are: it can learn the topology of a given object from the input point cloud; it produces meshes with different resolutions during the learning process, avoiding the need for a mesh simplification step. The main limitations of GSRM are: a dense, random sample set of points is required for the reconstruction.

The GSOSM explores the concept of Voronoi region to define a new learning rule (CCHL) that is employed as the basis for establishing connections between nodes. The CCHL enables GSOSM to produce a complete local triangulation, thus overcoming the limitation of its predecessor (CHL). The two step adaptation operator of GSOSM qualifies it to handle any input sequence producing a mesh that is faithful to the objective surface. Moreover, the GSOSM algorithm overcomes the homeomorphism limitations and is able to produce representations of folded surfaces that may or may not be 2-manifold. However, the GSOSM is not able to produce a quality mesh if the input has areas with low sampling density.

Moreover, there was discussion of pre and post-processing steps and some of the other procedures that need to be observed when using the models cited as a tool for solving the surface reconstruction problem.

The methods presented here as solutions for the surface reconstruction problem present promising results. However, they still have some limitations such as to deal with sparse or not randomly sampled point clouds. Therefore this is an open area for future developments.

## 7. References.

Amenta, N., Choi, S. & Kolluri, R. K. (2001) The power crust, *In Proceedings of the sixth ACM symposium on Solid modeling and applications*, pp. 249-266.

Aurenhammer, F. & Klein, R. (2000) *Voronoi diagrams*, In J.-R. Sack, J. Urrutia (Eds.), Handbook of Computational Geometry, Elsevier Science/North-Holland, Amsterdam, pp. 201-290.

Barhak, J. (2002) *Freeform objects with arbitrary topology from multirange images,* PhD thesis, Technion - Israel Institute of Technology, Haifa, Israel.

de Berg, M., Cheong, O., van Kreveld, M. & Overmars, M. (2008) *Computational Geometry: Algorithms and Application,* 3rd Edition, Springer-Verlag.

Bernardini, F., Bajaj, C. L., Chen, J. & Schikore, D. (1999) Automatic Reconstruction of 3D CAD Models from Digital Scans, *International Journal of Computational Geometry and Applications*, Vol 9, No 4, pp. 327-369.

Bernardini, F., Martin, I., Mittleman, J., Rushmeier, H. & Taubin, G. (2002) Building a digital model of Michelangelo's Florentine Pieta, *IEEE Computer Graphics and Applications*, Vol 22, No 1, pp. 59–67.

Bobenko, A. I. & Springborn, B. (2007) A discrete Laplace-Beltrami operator for simplicial surfaces, *Discrete and Computational Geometry*, Vol 38, No 4, pp. 740-756.

Brito, A. D., Doria, A. D., de Melo, J. D. & Goncalves, L. M. G. (2008) An adaptive learning approach for 3D surface reconstruction from point clouds, *IEEE Transactions on Neural Networks*, Vol 19, No 6, pp. 1130-1140.

Cignoni, P., Rochini, P. & Scopigno, R. (1998) Metro: Measuring Error on Simplified Surfaces, *Computer Graphics Forum*, Blackwell Publishers, Vol 17, No 2, pp. 167-174.

DalleMole, V.L. & Araujo, A. F. R. (2008a) The growing self-organizing surface map. *In International Joint Conference on Neural Networks*, pp. 2061-2068.

DalleMole, V. L. & Araújo, A. F. R. (2008b) The Growing Self-Organizing Surface Map: Improvments, *In 10th Brazilian Symposium on Artificial Neural Networks (SBRN´08)*, Salvador.

DalleMole, V. L. & Araujo, A. F. R. (2009) Growing Self-Organizing Surface Map: learning a surface topology from a point cloud. To appear on Neural Computation.

Do Rêgo, R. L. M. E., Araujo, A.F.R. & de Lima Neto, F.B. (2007) Growing self-organizing maps for surface reconstruction from unstructured point clouds, *In International Joint Conference on Neural Networks*, pp. 1900–1905.

Do Rêgo, R. L. M. E., Bassani, H.F. & Filgueiras, D. (2009) Surface Reconstruction System Based on a Growing Self-organizing Map. *To appear in International Conference on Artificial Neural Networks*.

Edelsbrunner, H., Kirkpatrick, D.G. & Seidel, R. (1983) On the shape of a set of points in the plane, *IEEE Transactions on Information Theory*, Vol 29, pp. 551-559.

Fisher, M., Springborn, B., Bobenko, A. I. & Schroder, P. (2006) An algorithm for the construction of intrinsic Delaunay triangulations with applications to digital geometry processing. *In SIGGRAPH '06: ACM SIGGRAPH 2006 Courses*, pp. 69-74, New York, NY, USA, ACM.

Fritzke, B. (1996) *Unsupervised ontogenetic networks*. Handbook of Neural Computation.

Fritzke, B. (1995) A Growing Neural Gas Network Learns Topologies, *in Advances in Neural Information Processing Systems* 7, G.Tesauro, D.S. Touretsky and T. K. Leen, MIT Press, Cambridge MA.

Fritzke, B. (1994) Growing cell structures - A self-organizing network for unsupervised and supervised learning, *Neural Networks*, Vol 7, pp. 1441-1460.

Hoffmann, M. & Varady, L. (1998) Free-form modelling surfaces for scattered data by neural networks, *Journal for Geometry and Graphics*, Vol 2, No 1, pp. 16.

Hoppe, H., Derose, T., Duchamp, T., Mcdonald, J. & Stuetzle W. (1992) Surface reconstruction from unorganized points, *In Proceedings of Siggraph Conference*.

Ivrissimtzis, I., Jeong, W. K., Lee, S., Lee, Y. & Seidel, H. P. (2004) Neural Meshes: Surface reconstruction with a learning algorithm. *Eurographics Symposium on Point-Based Graphics*, pp. 1-10.

Ivrissimtzis, I., Jeong, W. K., Lee, S., Lee, Y. & Seidel, H. P. (2004) Surface Reconstruction Based on Neural Meshes, *Matematical methods for CAGD*, Nashoboro Press, Brentwood.

Jockusch, J. & Ritter, H (1993) An instantaneous topological mapping model for correlated stimuli. *In International Joint Conference on Neural Networks*, Vol 1, pp. 529-534.

Kohonen, T. (1998) The self-organizing map, *Neurocomputing*, Vol 21, pp. 1-6.

Mäntylä, M. (1988) *An Introduction to Solid Modeling*. Computer Science Press, Rockville.

Martinez, T., Schulten, K. (1994) Topology Representing Network, *Neural Networks*, Vol 7, No 3, pp. 507-522.

Martinez, T. (1993) Competitive Hebbian learning rule forms perfectly topology preserving maps. International Conference Artificial Neural Networks, pp. 427-434.

Martinez, T., Schulten, K. (1991) A "Neural-Gas" Network Learns Topologies, *in Artificial Neural Networks*, T. Kohonen, K. Makisara, O. Simula and J. Kangas, Elselvier Science Publishers. North-Holland.

Miller, J.V., Breen, D.E., Lorensent, W.E., O'Bara, R.M. & Wozny, M.J. (1991) Geometrically deformed models: A method for extracting closed geometric models from volume data. *In Proceedings of the International Conference on Computer Graphics and Interactive Techniques*, Vol 25, pp. 217-226.

Qin, H., Mandal, C. & Vemuri, B.C. (1998) Dynamic Catmull-Clark subdivision surfaces, *IEEE Transactions on Visualization and Computer Graphics*, Vol 4, No 3, pp. 215-229.

Saleem, W. (2003) *A Flexible framework for learning-based surface reconstruction,* Master's thesis, Computer Science Department, University of Saarland, Saabrucken, Germany.

Satava, R. M. & Jones, S. B. (1998) Current and future applications of virtual reality for medicine". Proceedings of the IEEE, Vol 86, No 3, pp. 484-489.

Yu, Y. (1999) Surface reconstruction from unorganized points using self-organizing neural networks, *In Proceedings of IEEE Visualization Conference*, pp. 61-64.

IntechOpen




**Self-Organizing Maps**

Edited by George K Matsopoulos

ISBN 978-953-307-074-2
Hard cover, 430 pages
**Publisher** InTech
**Published online** 01, April, 2010
**Published in print edition** April, 2010

The Self-Organizing Map (SOM) is a neural network algorithm, which uses a competitive learning technique to train itself in an unsupervised manner. SOMs are different from other artificial neural networks in the sense that they use a neighborhood function to preserve the topological properties of the input space and they have been used to create an ordered representation of multi-dimensional data which simplifies complexity and reveals meaningful relationships. Prof. T. Kohonen in the early 1980s first established the relevant theory and explored possible applications of SOMs. Since then, a number of theoretical and practical applications of SOMs have been reported including clustering, prediction, data representation, classification, visualization, etc. This book was prompted by the desire to bring together some of the more recent theoretical and practical developments on SOMs and to provide the background for future developments in promising directions. The book comprises of 25 Chapters which can be categorized into three broad areas: methodology, visualization and practical applications.

**How to reference**

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Vilson L. DalleMole, Renata L. M. E. do Rego and Aluizio F. R. Araujo (2010). The Self-Organizing Approach for Surface Reconstruction from Unstructured Point Clouds, Self-Organizing Maps, George K Matsopoulos (Ed.), ISBN: 978-953-307-074-2, InTech, Available from: http://www.intechopen.com/books/self-organizing-maps/the-self-organzing-approach-for-surface-reconstruction-from-unstructured-point-clouds

INTECH
open science | open minds

**InTech Europe**
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

**InTech China**
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2010 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License, which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen