

Growing Self-Organizing Maps for Surface Reconstruction from Unstructured Point Clouds

Renata L. M. E. do Rêgo, Aluizio F. R. Araújo, and Fernando B. de Lima Neto

Abstract— This work introduces a new method for surface reconstruction based on *Growing Self-organizing Maps*, which learn 3D coordinates of each vertex in a mesh as well as they learn the topology of the input data set. Each map grows incrementally producing meshes of different resolutions, according to the application needs. Another highlight of the presented algorithm refers to the reconstruction time, which is independent from the size of the input data. Experimental results show that the proposed method can produce models that approximate the shape of an object, including its concave regions and holes, if any.

I. INTRODUCTION

THE problem of reconstructing the surface of objects from a point cloud is quite common in many areas such as CAD design, virtual reality and medicine. The surface reconstruction methods aim to create a model to reproduce an object shape given a set of points of the surface of this object. The large number of points to be processed constitutes an ubiquitous difficulty, *i.e.* a challenge that reconstruction methods must face.

Depending on their purpose, some applications of surface reconstruction require accurate and detailed models while others require models with less vertices and polygons to be effective. Then, a good reconstruction method must be able to produce models according to different resolutions of the target object, depending on the problem; to avoid further processing such as simplification or refinement procedures.

The sample points used for reconstruction can be classified as structured or unstructured depending on the existence or absence of connectivity information, respectively. One of the most challenging stages of surface reconstruction methods applied to unstructured point cloud is to obtain the correct connectivity about the sample points.

Some reconstruction methods such as Deformable Models [1], establish the mesh topology beforehand. The main pitfall of this approach is the topologies of the produced meshes which are equivalent to the pre-defined one; sometimes, producing poor results. For instance, a pre-defined topology with no holes can produce only meshes without holes.

Many well established techniques, propose solutions to the surface reconstruction problem from a geometric point of view. These algorithms require long processing time for the input point cloud and do not scale up properly with very large point clouds [2], [3].

A different perspective for surface reconstruction is to look at it as a learning problem. Under this perspective some methods were proposed based on self-organization [4], [5], [6], [7]. These methods can learn the coordinates of the mesh vertices and can handle very large input point clouds. The methods draw the point cloud and use the samples as training data. During the learning process, the vertices of the mesh have their positions adapted to fit the input data. In this particular sense, learning-based surface reconstruction methods are equivalent to deformable models once they change the shape of an initial model toward the shape of a target object.

Previously proposed reconstruction methods based on self-organizing maps include Kohonen's SOM [4], [5], Growing Cell Structures [6], and Topology Representing Networks [7]. However, they can not simultaneously learn the input data topology and grow incrementally to produce meshes with different resolutions as Growing Neural Gas (GNG) [8] can do.

We propose a surface reconstruction method based on GNG to learn topology through self-organization. Furthermore, this alternative can produce meshes with different resolutions because the mesh grows incrementally during the learning process. The standard GNG learning algorithm, however, produces only wireframes, so the proposed method includes some changes in this algorithm in order to produce triangular meshes. These changes include an extension of the Competitive Hebbian Learning, the use of GCS vertex insertion operation and a different operation to remove edges from the meshes.

Experimental results show that the proposed method learn vertices coordinates and the connectivity among these vertices building meshes to reproduce the shape of target objects. The learning process successively generates meshes with increasing number of nodes and thus different resolutions.

This paper is organized as follows: Section II presents some previous works on surface reconstruction: self-organizing maps and reasons why they are suitable for solving surface reconstruction problem, and their actual use in reconstruction methods. Section III describes the proposed

Renata Rêgo and Aluizio F. R. Araújo* are with Center of Informatics, Federal University of Pernambuco, Recife/Brazil. *Contact phone: 55 81 21268430 #4321 and e-mail: aluizioa@cin.ufpe.br. Fernando Buarque de Lima Neto (IEEE M'2006) is with Department of Computing Systems at Polytechnic School of Engineering - Pernambuco State University, Recife/Brazil

method. The experiments and results are presented and discussed in Section IV. This paper conclusion, including achievements and limitations of the proposed method, is in Section V.

II. PREVIOUS WORK

An active research topic is the problem of surface reconstruction from unstructured point clouds. A number of approaches handle this problem simply by geometric perspective, an example of that is the approach proposed by Hoppe [2]. Another relevant approach to surface reconstruction is based on deformable models [9] that are slowly reshaped towards the target object. Such a reconstruction employs energy evaluation or force functions to assess each produced shape. The process constructs meshes to approximate the shape to the point cloud. Examples of this approach can be found in the review work of Montagnat *et al.*[1].

A third perspective to surface reconstruction is to understand it as a learning problem [10]. According to this approach all necessary information to reconstruct a surface can be learned from the available data. Different self-organizing maps are used as learning methods to solve a surface reconstruction problem.

A. Self-organizing Maps

Self-organizing maps (SOM) were originally proposed by Kohonen [11] for the visualization and abstraction of high-dimensional data. SOM basic structure is formed of an input and an output layer. Usually the topology of the output layer consists of a two-dimensional grid of nodes. Each output unit is connected with all input nodes through a weight vector. The learning algorithm of SOM is composed by three main steps: competition – to identify a winner, cooperation – to correlate activation of a winner and its neighbors, and adaptation – to update the weight vectors.

The ability of SOM to construct topological maps from an input data distribution can be employed in solving the surface reconstruction problem. Despite this ability, pre-defined number of nodes and the connections between them constraint the accuracy of the topological map produced by SOM [12]. Models proposed to overcome this limitation can be used for surface reconstruction.

The *Topological Representing Networks* (TRN) [12] has the network size pre-defined to construct topology preserving maps. TRN is seen as a combination of Neural Gas (NG), placing the nodes according to the probability distribution of input data, with Competitive Hebbian Learning (CHL), building a topology with these nodes, i.e. to perform topology learning [13]. Such ability is a suitable feature to be used as a solution for the problem of topology detection of a point cloud in surface reconstruction methods.

Despite the favorable characteristic, SOM and TRN share a relevant limitation: the previous choice of the number of nodes for the map. A solution to this drawback was to build

maps growing incrementally as those constructed by Growing Cell Structures (GCS) [14] and Growing Neural Gas (GNG) [8]. The former generates maps consisting only of basic building blocks, hypertetrahedrons of dimensionality k chosen in advance.

Unlike the 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. As the TRN, the GNG is able to learn the topology of input data through Competitive Hebbian Learning [13]. Hence, GNG can be seen as a GCS variant without its topological restrictions or as a growing version of TRN.

The use of SOM, TRN and GCS in the surface reconstruction problem has already been studied, a number of achievements were obtained, however some limitations still persist as they are discussed in the next sub-section.

B. Self-organizing Maps for Surface Reconstruction

Self-organizing maps are used in reconstruction methods because they can generate a map that closely matches the shape of a target object which has its shape represented by a point cloud. The self-organization process demands only the spatial coordinates of the input points in order to execute the reconstruction.

The reconstruction process of methods based on self-organizing maps use selected points within the input point set as training data for the learning algorithm. When self-organizing maps are used for reconstruction, the mesh vertices correspond to the output nodes of the map. The weight vector of the networks determines the vertices positions in the mesh whereas the connections between the nodes correspond to the edges of the mesh. So hereafter the terms map and mesh, node and vertex, and connection and edge, will be used interchangeably.

Surface reconstruction methods based on Kohonen's SOM ability to distribute the vertices in a polygonal mesh are presented in various works [4], [5], where (i) the topology of the mesh is predetermined and (ii) Kohonen's learning algorithm is carried out to obtain correct 3D coordinates of every vertex of the mesh.

Despite all achievements, SOM presented difficulties to reconstruct concave regions [4], [5]. That is, after the learning process some vertices or triangles may be unstable, dangling among regions densely populated by the data. To avoid such problems these methods employ some mesh operations: edge swap, edge collapse, vertex split and triangle subdivision. Furthermore, both works use mesh refinement algorithms to obtain a new mesh with more elements, once SOM does not insert new nodes during the learning process.

Ivrissimtzis *et al.* [6] has proposed a reconstruction method, called Neural Meshes, based on GCS. A straight forward consequence is that Neural Meshes pre-defines neither the number of nodes nor the exact topology; however, the generated meshes are always topologically

equivalent to the initial mesh, usually a tetrahedron. To overcome this limitation, a topology learning step in Neural Meshes algorithm was introduced [15] to enable the formation of handles and boundaries of a surface. Boundaries are then created by removing triangles and handles are created by merging these boundaries. The Neural Meshes grow incrementally, hence, such a method generates meshes of different resolutions at different stages of the learning process. So, differently from other SOM-based methods, Neural Meshes do not need mesh refinement algorithms for generating multiresolution meshes.

The Extended Neural Gas (ENG) [7] uses an extension of Competitive Hebbian Learning to discover the topology of the input point cloud. This extension allows ENG to generate not only the connections between nodes, as the standard CHL does, but also the triangular faces of a triangular mesh.

ENG creates a mesh that is not a 2-manifold surface [7] because the topology learning process does not prohibit the existence of some anomalies such as the occurrence of edges shared by more than two faces, of edges disconnected from any face, and of faces oriented towards different directions. To solve these anomalies, in ENG there is a manifold creation algorithm which is applied to the non-manifold meshes produced by ENG. As a TRN-based method, the number of nodes is fixed in ENG, so that the mesh resolution of the reconstructions generated with this method must be previously defined.

The promising results obtained with the neural-reconstruction methods show that self-organizing maps are suitable for the task of generating meshes to represent the shape of a particular object given only a set of spatial coordinate of points belonging to the object surfaces. However, the mentioned methods still have limitations, such as (i) to require previously defined number of nodes and (ii) incapacity to learn topology due restrictions for connection changes from the initial configuration.

III. PROPOSED SOLUTION

In this section, we introduce a new method to detect the topology of the point cloud, as ENG does, and to generate meshes with different resolutions during the learning process, as Neural Meshes does. The neural solution proposed here is an extension of the Growing Neural Gas and simultaneously uses the vertex insertion operation employed by Growing Cell Structures. We found that this combination of features in addition to other modifications carried on the standard GNG is able to generate polygonal meshes, composed by vertices, edges and faces, instead of the wireframes composed only by vertices and edges.

We refer to the proposed method as Growing Self-Reconstruction Meshes (GSRM). The pseudo-code of GSRM learning algorithm and parameters are introduced and discussed below.

The learning parameters of GSRM are ϵ_n (winner learning rate), ϵ_b (winner neighbors learning rate), λ (number of

adaptation steps until a new node insertion), β (decreasing rate for all error counters), α (decreasing rate for the nodes between which a new node have been inserted), age_{\max} (maximal age for an edge to be considered valid), nv (stop criterion – number of nodes in the map).

The main processing steps of GSRM are detailed below:

1. Start the map with a set A of three nodes with weight vectors randomly chosen from the input point set;
2. Draw an input point ξ from the point cloud P ;
3. Find the first (w_{s_1}), the second (w_{s_2}), and the third (w_{s_3}) best matching nodes (*i.e.* winners) of the map, that is, the three vertices (weight vectors) of the map with shortest distance from ξ ;

$$\|w_{s_1} - \xi\| \leq \|w_{s_i} - \xi\| \quad \forall s_i \in A \quad (1)$$

$$\|w_{s_2} - \xi\| \leq \|w_{s_i} - \xi\| \quad \forall s_i \in A - \{s_1\} \quad (2)$$

$$\|w_{s_3} - \xi\| \leq \|w_{s_i} - \xi\| \quad \forall s_i \in A - \{s_1, s_2\} \quad (3)$$

4. Create (or reinforce) connections between these nodes, and triangular faces of the mesh, according to the extended Competitive Hebbian Learning;
5. Update the winner error counter according to Equation (4):

$$\Delta E_{s_1} = \|w_{s_1} - \xi\|^2 \quad (4)$$

6. Move s_1 and its topological neighbors towards ξ with learning rates ϵ_b e ϵ_n , respectively:

$$\begin{aligned} \Delta w_{s_1} &= \epsilon_b (\xi - w_{s_1}) \\ \Delta w_{s_i} &= \epsilon_n (\xi - w_{s_i}) \quad \forall s_i \in N_{s_1} \end{aligned} \quad (5), (6)$$

where N_{s_1} is the set formed by the neighbors of s_1

7. Update the age of all edges emanating from s_1 according to Equation (7):

$$\text{age} = \text{age} + 1 \quad (7)$$

8. Remove invalid edges, *i.e.* those edges with age greater than age_{\max} , and their incident faces;
9. If the number of input signals presented so far to the network is an integer multiple of the parameter λ , then insert a new node.

A. Decrease the error variables of s_q e s_f :

$$\Delta E_{s_q} = -\alpha E_{s_q} \quad \Delta E_{s_f} = -\alpha E_{s_f} \quad (8), (9)$$

B. Interpolate the error variable of s_r from s_q e s_f :

$$E_{s_r} = 0.5(e_{s_q} + E_{s_f}) \quad (10)$$

where: s_q is the node with the highest error counter, s_f is the s_q neighbor with highest error counter and s_r is the newly inserted node.

10. Decrease the error variables of all nodes, according to Equation (11):

$$\Delta E_s = -\beta E_s \quad \forall (s \in A) \quad (11)$$

11. If a stopping criterion (e.g. net size or some performance measure) is not fulfilled, continue from step 2. GSRM considers the number of nodes in the map a stopping criterion as well.

The main differences between GSRM and the standard GNG are found in steps 4, 8, and 9. Such modifications involve extension of the Competitive Hebbian Learning, the procedure for edge removal and the vertex insertion operator. These differences are further discussed below.

A. Extended Competitive Hebbian Learning (ECHL)

In the Growing Self Reconstruction Meshes, CHL is extended to create the topological faces. The ECHL is described as:

- 1) For each sample presented to the map, three winner nodes are determined, i.e., nodes whose weight vectors are nearer to the samples than all the other nodes;
- 2) If there is not a synaptic connection between each pair of these nodes, this connection is then created. If any of these connections previously existed this fact is reinforced by the setting of its age to zero;
- 3) If the created or reinforced connections do not form a triangular face, such a face is then created.

B. Edges and Incident Faces Removal

During steps of weight vectors adaptation, some of the edges generated in the network may become invalid and are removed. As in GNG, an edge aging scheme is used here. Edges whose age becomes higher than a given threshold are automatically excluded from the mesh.

In standard GNG these edges are simply removed and so are the vertices without incident edges. In GSRM the elimination of edges implies in the removal of their incident faces as well. This because these faces can not exist without any of the incident edges. The exclusion occurs as follows:

- 1) Remove faces incident to the invalid edges, i.e. with age larger than age_{\max} .
- 2) Remove edges without incident faces. The invalid edges are removed here because their entire faces have been removed in the previous step. Some other edges without incident faces can be removed as well.
- 3) Remove vertices without incident edges.

C. Vertices Insertion

In the standard GNG, a new vertex (s_r) is inserted between

two existing vertices (s_q and s_f). The original edge connecting s_q and s_f is removed and two new edges, connecting s_q and s_f to s_r are then created.

This operation does not create new triangular faces when a new vertex is inserted. To overcome this restriction, GSRM inserts a new vertex in a similar way to GCS. Figure 1 illustrates this operation. The insertion of a new vertex in the mesh happens as follows:

- 1) Insert a new vertex (s_r) and initialize its weight vector with the average of s_q and s_f .
- 2) Create edges connecting s_r to s_q and to s_f and to their common neighbors (hatched edges in Figure 1(b)).
- 3) Replace each face incident to the edge between s_q and s_f by two new faces, one incident to the edge between s_r and s_q and the other incident to edge between s_r and s_f .
- 4) Remove the original edge connecting s_r e s_f (hatched edges in Figure 1(a)).

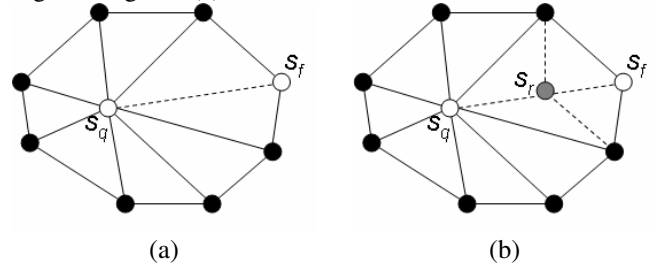


Figure 1 - (a) Original Mesh (b) Mesh after a new vertex insertion.

IV. EXPERIMENTS

This section presents the results of surface reconstructions carried out by the proposed neural reconstruction method applied to three synthetic objects: *the Hand*, *the Max-Planck face*, gently granted by Ionnais Ivrismitzis, and *the Stanford Bunny*, available on the internet (i.e. at Stanford repository¹).

The point cloud contains only the spatial coordinates of vertices in the original mesh, no additional informational about the connectivity information among these vertices are used for the reconstruction that are presented bellow. In all experiments presented here the meshes have five thousand vertices.

The experiments carried out in this work aim at showing the three major achievements of our new method:

- (1) The proposed learning algorithm for reconstruction can learn (a) vertices coordinates and (b) the connectivity among the vertices, as GNG does. This in addition to producing triangular meshes, as GCS does, while standard GNG only produces wireframes and GCS imposes restrictions on the mesh topology;

- (2) The proposed method is able to produce meshes with different resolutions throughout learning process;

- (3) The reconstruction time of the proposed approach does

¹ The Stanford 3D Scanning Repository can be reached on the internet at <http://graphics.stanford.edu/data/3Dscanrep>, accessed on October, 2006.

not depend on the point cloud size.

The Hausdorff distance [16] was used to compare the distance between original and generated meshes with: GSRM, GCS (standard), and Neural Meshes. The rationale is: the smaller the distance, the best a reconstructed mesh represents the original one. Meshes produces with ENG could not be compared because the comparison metrics used was unavailable. As GNG does not produced meshes (only wireframes), so the Hausdorff distance could not be calculated either.

Figure 2 shows the wireframes produced by GNG that approximates the shape of the reconstructed objects but are not polygonal meshes. Figure 3 illustrates the topological restriction of the GCS. Figure 3(a) shows that this map can not reconstruct concave regions (see false bridges between the fingers) and Figure 3(b) shows that GCS can not reconstruct holes, see the counter-example at the bottom of the hand.

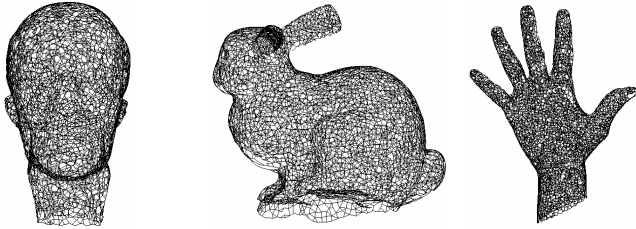


Figure 2 - Wireframes (5k vertices) produced with standard GNG

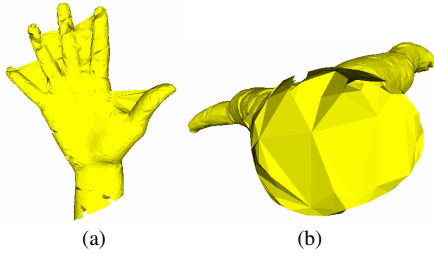


Figure 3 - Hand reconstructed with standard GCS [14].

Compare the two previous figures with results presented in Figure 4, which are reconstructions produced by GSRM. Note the remarkable learn ability of the presented method that can be evoked through the reconstruction of concave regions, in charts (a), (c) and (e), and, of holes as seen in the charts (b), (d) and (f) – these three latter are bottom views of images presented in charts (a), (c) and (e), respectively.

Due to its fractal growing, GSRM can generate meshes of different resolutions at distinct training stages. So it needs not necessarily to use any refinement or simplification method. Figure 5 shows an example of this feature.

The learning time is independent of the number of points from the point cloud, once sampling is the only step in which the point cloud is processed and it is not amount depend.

Table 1 shows the Hausdorff Distances between the original meshes and those reconstructed with GCS, GSRM (calculated with Metro Tool [16]), and Neural Meshes (taken from [10]).

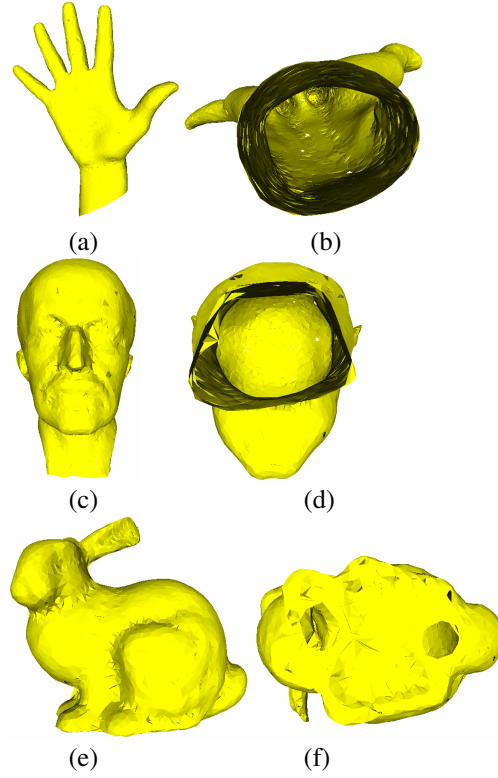


Figure 4 - Hand, Max Plunk, and Bunny reconstructions obtained with GSRM.

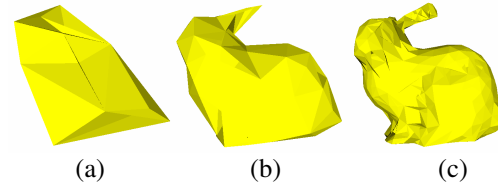


Figure 5 - GSRM Fractal Growing with different resolutions at distinct learning stages: (a) 20 vertices, (b) 100 vertices, (c) 500 vertices.

TABLE 1
HAUSDORFF DISTANCES BETWEEN THE ORIGINAL MODELS AND THE RECONSTRUCTION OBTAINED WITH GSRM AND NEURAL MESH.

	GCS	GSRM	Neural Mesh
Max Planck	35.35	7.687	5.23
Hand	20.891	9.292	6.60
Bunny	0.014258	0.005929	0.006

The results confirm that GSRM produce better results then standard GCS. This is because points filling concave regions and holes in GCS meshes do not have correspondence with points in the original meshes.

Although the overall numbers presents better results for the reconstructions of the Neural Mesh, there are some important remarks:

1. The number of vertices used in the Neural Mesh model was 20 thousand versus mere five thousand for the GCS,

GSRM models. 20k models wasn't generated for the GSRM due to time restrictions, as will be discussed later;

2. The results presented for the Neural Mesh does not relate to the original method proposed by Ivriissimtzis in 2003 [6]. It is related to another method also proposed by Ivriissimtzis in 2004 [15], when a topology learning step was introduced. Notice that the original Neural Mesh is based in GSC and, as said before, this network can not learn topology.

Although the proposed method could reconstruct the shape of target objects, the meshes generated have some limitations that could be solved by post-processing theses meshes in a future work. The limitations concern to meshes that are not 2 manifolds, and undesirable holes, that can be easily distinguished from the surfaces holes. The manifold mesh creation algorithm proposed by the author of ENG [7], could be applied to solve the limitations of GSRM meshes. There are also other algorithms that create manifolds from non-manifolds polygonal meshes, as the one proposed in [17]. Once the mesh becomes a manifold, the Hausdorff distances between the original mesh and the one generated with GSRM (shown in

Table 1) should decrease due the elimination of undesirable faces whose points does not have correspondent points in the original mesh.

Another limitation of the method is that the training time increases with the current number of vertices in the mesh. This is because of the winner search at step 4 of pseudo-code, the search for the highest error counter node at step 11 and the updating of each node error counter at step 12. The winner search time could be reduced with an octree based search [6]. Steps 11 and 12 could be faster, refer to [10].

V. CONCLUSION AND FUTURE WORK

The neural approach put forward in this article succeeded in reconstructing surface models of the 3D objects from point clouds representing their shape. The highlights of the proposed method are (i) topology learning, which is a challenging feature for reconstruction methods, (ii) time of reconstruction independent of the size of the point cloud and (iii) models generated at different resolutions.

The main problems detect for the method so-far are: (i) non 2-manifolds meshes, (ii) small amount of undesirable holes and (iii) prolonged learning time when dealing with many thousands of vertices. At this point we have devised possible solutions for the first two problems by means of post-processing of the mesh with manifolds creation algorithms as proposed at [7] and [17]. As for the sometimes elevated learning time problem, it could be solved with algorithms that make the winner search faster, as the octree-based search [6] and solutions that could fasten other steps of the algorithm, refer to Saleem [10].

VI. REFERENCES

- [1] J. Montagnat, H. Delingette, and N. Ayache. "A review of deformable surfaces: topology, geometry and deformation," *Image and Vision Computing*, vol.19, 2001.
- [2] H. Hoppe, T. Deroose, T. Duchamp, J. McDonald, W. Stuetzle, "Surface reconstruction from unorganized points," *In Siggraph 92, conference proceedings*, pp. 71–78, 1992.
- [3] N. Amenta, M. Bern, M. Kamvysselis, "A new voronoi-based surface reconstruction algorithm," *In Siggraph 98, Conference Proceedings*, pp. 415–422, 1998.
- [4] Y. YU, "Surface reconstruction from unorganized points using self-organizing neural networks.," *Proceedings of IEEE Visualization Conference*, pp. 61–64, 1999.
- [5] A.M.B. Júnior, A.D.D. Neto, and J.D. de Melo, "Surface reconstruction using neural networks and adaptive geometry meshes," *Proceedings of the International Joint Conference on Neural Networks*, vol. 1, pp.:25–29, 2004
- [6] I. Ivriissimtzis, W.-K. Jeong, and H.-P. Seidel, "Using growing cell structures for surface reconstruction," *Proceedings of Shape Modeling International Conference*, pp. 78–86, 2003.
- [7] J. Barhak, "Reconstruction of freeform objects with arbitrary topology from multi range images," *Ph.D. Thesis*, Mechanical Engineering Faculty, Technion, Haifa, Israel, 2003.
- [8] B. Fritzke, "A growing neural gas network learns topologies," *Advances in Neural Information Processing Systems*, vol. 7, pp.:625–632, 1995.
- [9] D. Terzopoulos, and A. Witkin, "Deformable models," *IEEE Computer Graphics and Applications*, vol. 8, no. 6, pp. 41–51, 1988.
- [10] W. Saleem, "A flexible framework for learning-based Surface Reconstruction," *Master Thesis*, Computer Science Department, University of Saarland, Saarbrücken, Germany, 2004.
- [11] T. Kohonen, "The self-organizing map," *Neurocomputing*, vol 21, pp. 1–6, 1998.
- [12] T.M. Martinetz, and K.J. Schulten, "Topology representing networks," *Neural Networks*, vol. 7, no 3, pp.:507–522, 1994.
- [13] B. Fritzke, "Unsupervised ontogenetic networks," *Handbook of Neural Computation*, IOP Publishing and Oxford University Press, 1996.
- [14] B. Fritzke, Growing cell structures - a self-organizing network for unsupervised and supervised learning," *Neural Networks*, vol. 7, no 9, pp.: 1441 – 1460, 1994.
- [15] I. Ivriissimtzis, W.-K. Jeong, S. Lee, Y. Lee, and H.-P. Seidel, "Neural meshes: surface reconstruction with a learning algorithm," *Research Report 2004-4-005*, Max-Planck-Institut für Informatik, Saarbrücken, Germany, 2004.
- [16] P. Cignoni, C. Rocchini, R. Scopigno, "Measuring error on simplified surfaces," *Computer Graphics Forum*, vol. 17, n° 2, pp.:167–174, 1998.
- [17] A. Guéziec, G. Taubin, F. Lazarus, W. Horn, "Converting sets of polygons to manifold surfaces by cutting and stitching," *Proceedings of the Ninth IEEE Visualization*, pp.:383 – 390, 1998.