Surface Reconstruction Method Based on a Growing Self-Organizing Map

Renata L.M.E. do Rego[1,2], Hansenclever F. Bassani[1], Daniel Filgueiras[1], and Aluizio F.R. Araujo[1]

[1] Center of Informatics at Federal University of Pernambuco, Recife/Brazil
[2] Federal Institute of Pernambuco, Recife/Brazil
{rlmer,hfb,dfg,aluizioa}@cin.ufpe.br

Abstract. This work introduces a method that produces triangular mesh representation of a target object surface. The new surface reconstruction method is based on Growing Self-organizing Maps, which learns both the geometry and the topology of the input data set. Each map grows incrementally producing meshes of different resolutions, according to different application needs. Experimental results show that the proposed method can produce triangular meshes having approximately equilateral faces, that approximate very well the shape of an object, including its concave regions and holes, if any.

Keywords: Surface reconstruction, self-organizing maps.

1 Introduction

Surface reconstruction has been an important research topic due to (a) the large variety of application areas, such as medicine, cultural artifacts and robotics; (b) the recent advances in scanning technology to capture massive amounts of geometric data; (c) the innovations on hardware technology of computers allowing the visualization and manipulation of large three-dimensional data.

Surface reconstruction aims at producing a digital representation of the shape of a real world object given a set of points from its surface. A surface reconstruction method can be classified as static methods [1,2], based on geometric techniques, and dynamic methods [3,4], based on the evaluation of energy or force functions. The limitation of the static methods is that they process the points directly, and thus, can not deal with large point sets. The limitation of the dynamic methods concerns those target shapes not reachable through the modification of the initial mesh. For example, when the target shape is a torus and the initial mesh is a sphere, i.e., objects non-topologically equivalent. Surface reconstruction can also use learning-based methods which can process very large and/or noisy data, such as point clouds obtained from 3D scanners. Following this approach, some researches [5,6,7,8] employed methods based on a self-organizing map for surface reconstruction.

In this paper we propose a learning-based surface reconstruction method that is an improved version of a previous work [8]. The main limitation of [8] that

C. Alippi et al. (Eds.): ICANN 2009, Part I, LNCS 5768, pp. 515–524, 2009.

is addressed in this work concerns edges with more than two incident faces, yielding non-manifold meshes. The method proposed in this paper produces two manifold meshes. Another improvement of the method proposed here in comparison with our previous work is the addition of a condition to avoid long edges, and thus, skinny triangles. Some positive aspects of our previous work present in the method proposed here are the ability to learn different topologies from unstructured point clouds and to produce meshes with different resolutions so that a posterior step for simplifying the mesh is not necessary.

The rest of this paper is organized as follows: Section 2 presents related works using self-organizing maps for surface reconstruction. The proposed method is presented in Section 3. The experiments and results are presented in Section 4 and Section 5 concludes this paper.

2 Surface Reconstruction Methods Based on Self-Organizing Maps

Self-organizing maps are in general able to drag an initial grid of nodes towards a set of input points. For this reason, some researchers use SOM and some of its variants as the basis of surface reconstruction methods adapting an initial polygonal mesh to a given point cloud. In these works, the SOM nodes represent the vertices of the polygonal mesh; the connections between the nodes represent the edges of the mesh; and the faces of the mesh, however, do not have a direct representant in the map structure. Hereafter, the terms nodes and vertices, edges and connections, will be used interchangeably.

Some reconstruction algorithms [6,7] are based on the original SOM [9]. These methods are suitable to represent the faces of a polygonal mesh since the SOM structure does not change during the learning process and thus, the faces can be established in advance. The main restrictions of SOM for surface reconstruction, addressed in [6,7], are their fixed topology and pre-defined number of vertices.

Some SOM variants [10] are also used as the starting point for surface reconstruction methods. The method proposed in [11], based on the Topology Representing Networks (TRN) [12], can learn the topology of the input data. As the TRN topological structure changes during the learning process, the faces of the desired polygonal mesh representation cannot be determined in advance as in the original SOM based methods. For this reason the surface reconstruction method proposed in [11] extends TRN to define faces. TRN has a pre-defined number of nodes, hence the meshes produced by TRN have a resolution dependent on the pre-defined number of vertices. Furthermore, the topology learning strategy of TRN does not produce a two-manifold mesh. To solve this limitation, Barhak [11] included a post-processing step responsible for creating manifold meshes.

The Neural Meshes [13], another learning based surface reconstruction method, employs the Growing Cell Structures (GCS) [14]. The structure of GCS map consists of k-dimensional simplices [14], for $k = 2$ triangles represent the faces of a triangle mesh. GCS maps grow incrementally producing meshes with different resolutions, avoiding mesh simplification. A relevant restriction of the standard

GCS for reconstructing surfaces is when the target shape is not topologically equivalent to the initial mesh.

The Growing Neural Gas (GNG) can learn the geometry and topology of an input point cloud, and it grows incrementally. Thus, we propose a method having GNG as its starting point.

3 Self-Organizing Solution Proposed for the Surface Reconstruction Problem

The proposed method receives as input the 3D coordinates of a set of points randomly sampled from the surface of a target object, learns its geometry and topology, and outputs triangular meshes representation of the target object surface with different resolutions. The neural solution proposed here addresses some challenges of the surface reconstruction task:

- Production of 2-manifold triangular mesh approximation of the target surface.
- Reconstruction of meshes according to different resolutions.
- Learning of the geometry (vertex coordinates) and topology of the mesh.
- Learning of the topology (vertex connectivity) of the mesh without any structural information.
- Reconstruction of surfaces with different topologies.

The proposed reconstruction method, called Growing Self-reconstruction Maps (GSRM), is a modified version of GNG aiming to support the representation of triangular meshes, to generate 2-manifold meshes, and to avoid long edges to produce only triangles which are approximately equilateral. The differences between GSRM and the standard GNG concern the Competitive Hebbian Learning algorithm (Section 3.2), the procedure for edge removal (Section 3.3), and the vertex insertion operator (Section 3.4). The step-by-step runthrough of the GSRM learning algorithm is presented in Section 3.1.

3.1 GSRM Learning Algorithm

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

The input for the GSRM learning algorithm is a point cloud P. The GSRM learning algorithm is described bellow:

1. Initialize the map (A) with three nodes. The weight vector of these nodes are randomly chosen from P.
2. Present a sample ξ, randomly chosen from P.

3. Find the two nodes (s_1, s_2) of the map that are nearest to ξ according to the Euclidian distance.
4. If a connection between s_1 and s_2 does not exists:
   – Then, create such a connection (e), and create faces incident to e, according to ECHL (Section 3.2).
   – Else, reinforce the edge e connecting s_1 and s_2 ($age_e = 0$) and check the other edges of the mesh against a condition for edge removal based on the Thales Sphere concept (Section 3.3).
5. Update the error counter of node s_1:

$$\Delta E = ||\mathbf{w}_{s_1} - \xi||^2 \tag{1}$$

   where $\mathbf{w}_{s_1}$ is the weight vector of node s_1.
6. Adapt the weight vector of node s_1 and its neighbors.

$$\Delta w_{s_1} = \varepsilon_b * (\xi - \mathbf{w}_{s_1}) \tag{2}$$

$$\Delta w_{s_n} = \varepsilon_n * (\xi - \mathbf{w}_{s_n}) \qquad \forall s_n \in N_i \tag{3}$$

   where N_i is the neighborhood of node s_1.
7. Update the age of all edges e emanating from s_1.

$$age_e = age_e + 1 \tag{4}$$

8. Remove the faces coincident to an old edge e ($age_e > age_{max}$) and remove this edge.
9. If the number of samples presented so far is greater then λ , insert a new node in the map according to the procedure presented in Section 3.4.
10. Decrease the error variables of all nodes:

$$\Delta E_s = -\beta E_s \qquad \forall s \in A \tag{5}$$

11. If the map achieved the desired resolution, complete the topological learning and perform the post processing step according to the procedure presented in Section 3.5.

3.2 Extended Competitive Hebbian Learning (ECHL)

An extended version of the Competitive Hebbian Learning rule (CHL) [15] is proposed for the creation of edges and faces. The two main modifications are: (a) before connecting two nodes (s_1 and s_2), the extended CHL checks if s_1 and s_2 have two connected common neighbors (n_1 and n_2), and if this is true, the edge connecting n_1 and n_2 is removed before creating the new edge connecting s_1 and s_2, to avoid overlapping edges; and (b) beyond edges, faces are also created which are formed by the nodes s_1 and s_2, and their common neighbors, if any.

Before creating a new face, the following conditions must be satisfied to avoid more than two faces sharing the same edge, and thus non 2-manifold meshes: (a) s_1 and s_2 must have at most two common neighbors, otherwise no face is created; (b) an edge e connecting s_1 or s_2 to a common neighbor must have at most one coincident face before creating a new face inciding on e, otherwise such a new face is not created.

The CHL extension just presented is based on the ideas of [11], however, we have introduced conditions to avoid overlapping edges and to produce 2-manifold meshes.

3.3 Edges and Incident Faces Removal

Some of the edges generated in the network may become invalid, then, they have to be removed. In this work, two mechanisms are used to identify such edges. The first one, employed by the standard GNG, consists of an edge ageing scheme. The second mechanism removes long edges to create triangular faces that are approximately equilateral. It is based on the Thales Sphere concept and has been presented in a related work [16]. The following procedure is used to identify long edges: If the angle σ between the vectors $\mathbf{v} = \mathbf{w}_{s_1} - \mathbf{w}_{s_2}$ and $\mathbf{u} = \mathbf{w}_{s_k} - \mathbf{w}_{s_2}$ ($\mathbf{w}_{s_i}$ is the weight vector of node s_i; s_1 is the winner node, s_2 is the second winner, and s_k is a neighbor of the winner node), is greater than $\frac{\pi}{2}$, then the edge connecting s_1 and s_k is removed.

This condition is verified whenever the connection to be created by a sample presentation already exists (step 4 of the algorithm presented in Section 3.1).

GNG simply removes an invalid edge and the vertices without any incident edges. The edge removal operation in GSRM takes face definition into account and removes the faces having an invalid edge before removing this edge. If the edge removal yields vertices without incident edges, these vertices are also removed.

3.4 Vertex Insertion

In standard GNG, a new vertex (s_r) is inserted between two existing vertices (s_q and s_f, where s_q is the vertex with highest accumulated error, and s_f is the neighbor of s_q with highest accumulated error). 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. GSRM considers face definition and, before removing the original edge, it removes each face inciding on this edge.

3.5 Topological Learning and Post-processing Step

At the end of the learning process the algorithm has learned the vertex coordinates and some of the edges and faces of the mesh. However some edges needed for the representation may not have been created and some long edges may not have been removed due to the non-deterministic behavior of the learning process. To overcome these consequences we complete the topological learning after the

mesh resolution desired is reached. The topological learning consists on removing long edges, according to the rules discussed in Subsection 3.3 and presenting all of the samples, to create all the necessary edges, according to the extended CHL presented in Subsection 3.2.

After the topological learning, some approximately regular polygons remain untriangulated after the learning process is finished. This occurs because any sample internal to those polygons has two connected nodes as winners, as illustrated in Figure 1(a), then CHL is unable to complete the triangulation internal to the polygon [16]. In order to complete the triangulation a new vertex is inserted in the geometric center of each polygon and the triangulation is possible by connecting this new vertex to each of the polygons vertices, see Figure 1(b) for an example. When the polygon is a quadrilateral, the triangulation is performed by creating its diagonal edge and replacing it by two triangles.

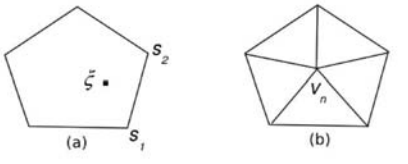

Fig. 1. Triangulating a polygon with five vertices

4 Experiments and Results

This section presents the experimental results carried out by the proposed surface reconstruction method. We first present visual results of surface reconstruction of synthetic objects. Then, we present numeric metrics to evaluate the reconstructed meshes and to compare our results with those of some related works.

The input point cloud has been acquired by randomly sampling points from the target synthetic object surface. The values of the parameters used in the experiments presented in this paper are: $\varepsilon_b = 0.05, \varepsilon_n = 0.0006, \lambda = 200, \alpha = 0.5, \beta = 0.0005, age_{max} = 30$. These values have been chosen by trial and error. Initial values were those presented in [10]. Then, these initial values have been gradually adjusted to better fit the proposed learning method.

The synthetic objects reconstructed are the hand and the Max-Planck, donated by Ioannis Ivrissmitzis [13], and the bunny, available at the Stanford repository (http://www-graphics.stanford.edu/data/3Dscanrep/). Figure 2 presents the reconstructions outputted by the proposed surface reconstruction method. All the reconstructions presented have about 20,000 vertices.

An important remark is the topology learning ability of the proposed surface reconstruction method. Note, for example, the hole boundaries reproduced in the bottom of the hand and the bunny (Figure 2). Note also that concave and detailed regions can be reproduced, such as the space between the fingers of the hand, and the ears of the bunny and the relief on its body.

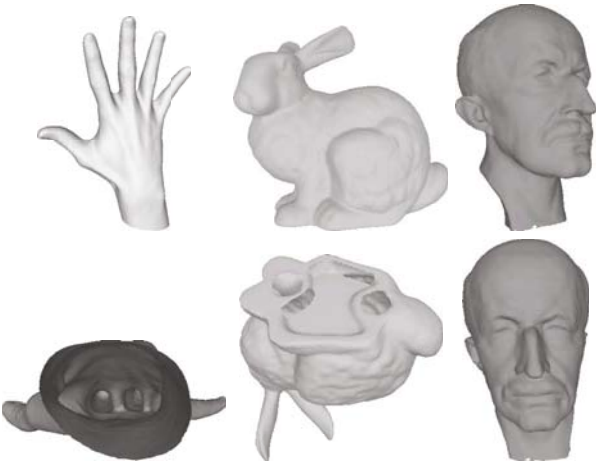

Fig. 2. Screenshots of GSRM Reconstructions

4.1 Evaluation Metrics of Reconstructed Meshes

The metrics used for evaluating the reconstructed meshes are: the distance between the target and the reconstructed surfaces, polygon conformity, and valence distribution. To get a more reliable estimate on the performance of the proposed algorithm, the values of the metrics presented here represents the average of the values observed in three runs, since the proposed algorithm is a stochastic one.

The distance between the target surface and the reconstructions are given in terms of Hausdorff distance, defined as follows: Given a point p and a surface S, the distance $e(p, S)$ is defined as:

$$e(p, S) = min\, d(p, p') \qquad \forall\, p' \in S \tag{6}$$

where $d()$ is the Euclidean distance between two points.

The one-sided distance between two surfaces S_1, S_2 is then defined as:

$$E(S_1, S_2) = max\, e(p, S_2) \qquad \forall\, p \in S_1. \tag{7}$$

The above definition of distance is not symmetric. There exist surfaces such that $E(S_1, S_2)$ is different from $E(S_2, S_1)$. Thus, a two-sided distance (Hausdorff distance) may be obtained by taking the maximum of $E(S_1, S_2)$ and $E(S_2, S_1)$. Note that, the smaller the Hausdorff distance, the better the reconstruction approximates the target shape.

Polygon conformity $R(P)$ is measured as the ratio between the smallest and the largest distance of the vertices of the polygon (s) to the polygon baricenter (b_p), see Equation (8). If the polygon tends to be elongated, $R(P)$ approximates to zero, but if the polygon tends to be regular, $R(P)$ is close to 1. Note that a triangle tends to be equilateral when its $R(P)$ value tends to 1.

$$R(P) = \frac{min||s - b_p||}{max||s - b_p||}, \tag{8}$$

Valence distribution refers to the number of neighbors of the mesh vertices. Valences must be distributed as evenly as possible [13], this means that the vertices of the mesh must have approximately the same number of neighbors.

It is important to compare the reconstructions of the proposed method with the reconstructions of other surface reconstruction methods: two versions of the Neural Meshes [17], the SOM based surface reconstruction [6], and the Power Crust [18], a traditional method. The values of the metrics related to the other methods, have been taken from their original papers. However, the original papers do not present values for all the metrics used here. For example, [13] presents the valence distribution of the meshes, but it does not present polygon conformity and Hausdorff distances. Thus, the comparisons presented here are according to the data available in the related papers.

Table 1 presents the Hausdorff distance from the original synthetic object to the reconstructions produced by GSRM and by two versions of the Neural Meshes algorithm [17]. Both the Neural Meshes and GSRM reconstructions have about 20,000 vertices. Neural Meshes performances were taken from [17]. GSRM reconstructions reproduce the shape of the original objects better then the Neural Meshes reconstructions. The GSRM reconstructions smaller distances from the original objects is due to the boundaries of the original objects (see the bottom of the bunny and the hand in Figure 2) that are reproduced in the GSRM reconstructions but not in the Neural Meshes reconstructions.

Table 2 compares the polygon conformity of the GSRM bunny (three options) with the Power Crust [18] and the SOM-based [6] reconstructions. The values for the reconstructions of the two last methods were extracted from [6]. From the values presented, GSRM reconstructions has better polygon conformity even with less elements (vertices and faces).

The valence distributions of GSRM and Neural Meshes reconstructions are shown in Table 3. The values for the Neural Meshes reconstructions were extracted from [13]. The first column of Table 3 refers to the name of the model and the approximate number of vertices. According to Table 3, most of the vertices have about five to seven neighbors both in the reconstructions produced by GSRM and Neural Meshes. Thus, both methods distribute valences almost evenly. The valence distribution of the meshes produced by the Neural Meshes, however, is slightly better.

Table 1. Hausdorff distances between original meshes and the reconstructions produced by two versions of Neural Meshes GSRM

	Max-Planck	Hand	Bunny
Neural Mesh I	5.23	6.60	0.006
Neural Mesh II	3.86	8.30	0.007
GSRM	0.002602	0.0013207	0.001513

Table 2. Polygon conformity comparison

Method	No. of vertices	No. of faces	$\overline{R(P)}$
Power Crust	277,242	191,981	0.519424
SOM	31,004	62,004	0.5676
GSRM	20,108	39,995	0.6886
GSRM	5,006	9,906	0.6893
GSRM	1,010	1,978	0.6855

Table 3. Valence distribution comparison

Model	Valence Distribution(%)											
	GSRM						Neural Meshes					
	4	5	6	7	8	others	4	5	6	7	8	others
Sphere 1k	4.13	24.90	43.81	22.67	3.96	0.53	0.20	29.60	47.00	18.40	4.40	0.50
Bunny 1k	5.89	26.48	39.60	21.20	5.17	2.09	0.70	27.40	49.00	18.60	4.20	0.20
Sphere 5k	4.12	26.15	41.61	22.68	4.87	0.56	0.20	29.02	46.60	19.98	3.54	0.68
Bunny 5k	5.00	26.28	40.87	22.24	4.60	1.00	0.44	28.32	47.28	19.74	3.54	0.70

5 Conclusions

The neural method put forward in this article succeeded in reconstructing surface models of 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) models generated at different resolutions, (iii) meshes with consistent polygon conformity and valence distribution.

At the moment, an important limitation of the proposed method concerns reconstruction time, which strongly depends on the number of nodes in the map. The main reason for this dependency is the linear search for winner nodes used in step 3, and the linear search for the node with highest accumulated error used in the step 9 of the proposed algorithm. To solve this limitation we propose, as a future work, implementing a more efficient search scheme in both situations.

References

1. Hoppe, H., Derose, T., Duchamp, T., Mcdonald, J., Stuetzle, W.: Surface reconstruction from unorganized points. In: Proceedings of Siggraph Conference, pp. 71–78 (1992)
2. Amenta, N., Bern, M., Kamvysselis, M.: A new voronoi-based surface reconstruction algorithm. In: Siggraph Conference Proceedings, pp. 415–422 (1998)
3. Miller, J., Breen, D., Lorensent, W., O'Bara, R., Wozny, M.: 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 (1991)
4. Qin, H., Mandal, C., Vemuri, B.: Dynamic catmull-clark subdivision surfaces. IEEE Transactions on Visualization and Computer Graphics 4(3), 215–229 (1998)

5. Hoffmann, M., Varady, L.: Free-form modelling surfaces for scattered data by neural networks. Journal for Geometry and Graphics 2(1), 1–6 (1998)
6. Brito, A., Doria, A., de Melo, J., Goncalves, L.: An adaptive learning approach for 3d surface reconstruction from point clouds. IEEE Transactions on Neural Networks 19(6), 1130–1140 (2008)
7. Yu, Y.: Surface reconstruction from unorganized points using self-organizing neural networks. In: Proceedings of IEEE Visualization Conference, pp. 61–64 (1999)
8. do Rego, R.L.M.E., Araujo, A.F.R., de Lima Neto, F.B.: Growing self-organizing maps for surface reconstruction from unstructured point clouds. In: International Joint Conference on Neural Networks, pp. 1900–1905 (2007)
9. Kohonen, T.: Self-Organizing Maps. Springer, Berlin (2001)
10. Fritzke, B.: Unsupervised ontogenetic networks. Handbook of Neural Computation (1996)
11. Barhak, J.: Freeform objects with arbitrary topology from multirange images. PhD thesis, Institute of Technology, Haifa, Israel (2002)
12. Martinetz, T., Schulten, K.: Topology representing networks. Neural Netw. 7(3), 507–522 (1994)
13. Ivrissimtzis, I., Jeong, W.K., Seidel, H.P.: Using growing cell structures for surface reconstruction. In: Proceedings of International Conference on Adaptive and Natural Computing Algorithms, pp. 78–86 (2003)
14. Fritzke, B.: Growing cell structures - a self-organizing network for unsupervised and supervied learning. Neural Networks 7, 9–13 (1994)
15. Martinetz, T.M., Schulten, K.J.: A neural-gas network learns topologies. Artificial Neural Networks, 397–402 (1991)
16. DalleMole, V., Araujo, A.F.R.: The growing self-organizing surface map. In: International Joint Conference on Neural Networks, pp. 2061–2068 (2008)
17. Saleem, W.: A flexible framework for learning-based surface reconstruction. Master's thesis, Computer Science Departament, University of Saarland, Saabrucken, Germany (2003)
18. Amenta, N., Choi, S., Kolluri, R.K.: The power crust. In: Proceedings of the sixth ACM symposium on Solid modeling and applications, pp. 249–266 (2001)