3D Object Surface Reconstruction Using Growing Self-organised Networks

Carmen Alonso-Montes and Manuel Francisco González Penedo

Dpto. de Computación, Universidade da Coruña, 15071 A Coruña, Spain
{carmen,cipenedo}@dc.fi.udc.es

Abstract. This paper studies the adaptation of growing self-organised neural networks for 3D object surface reconstruction. Nowadays, input devices and filtering techniques obtain 3D point positions from the object surface without connectivity information. Growing self-organised networks can obtain the implicit surface mesh by means of a clustering process over the input data space maintaining at the same time the spatial-topology relations. The influence of using additional point features (e.g. gradient direction) as well as the methodology characterized in this paper have been studied to improve the obtained surface mesh.

Keywords: Neural networks, Self-organised networks, Growing cell structures, Growing neural gas, 3D surface reconstruction, Gradient direction

1 Introduction

The study of 3D surface reconstruction techniques has become an important aspect of different applications in a variety of scientific areas, such as medicine, mechanical fluids, robotics or virtual reality. The main issue in 3D surface reconstruction is to obtain the mesh which defines the object surface, which is used as a basic 3D model by different applications. Input devices or filtering techniques obtain the 3D scene composed by isolated 3D points without connectivity information. This set of points is usually called unorganised cloud of points. The main step previous to the reconstruction process is to obtain the surface mesh which is defined by the connectivity among the points.

Several studies have focused on developing different algorithms to deal with point connectivity, like is shown in [1], where a new surface reconstruction algorithm is proposed. Obtaining the implicit mesh of the object surface has been tackled by different traditional techniques such as *Marching Cubes* [2] and *Voronoi filtering* [3]. The surface mesh is defined by representative elements and their spatial-topological relations. One disadvantage of using this kind of techniques is the impossibility of unifying the stages of obtaining the representative elements and their spatial-topological relations, which maintain a continuous surface. A further argument against these techniques is that only a point feature can be used. The results obtained are also dependent on the unorganised cloud of points. These techniques cannot deal properly with mesh surface continuity.

The research has tended to focus on algorithm improvement rather than studying the influence of using new point features [1]. However, few studies have considered the neural networks for the surface reconstruction problem [4]. The aim of this paper is to characterize a methodology to deal with the 3D surface reconstruction problem by means of growing self-organised networks in order to avoid the disadvantages of traditional techniques and improve previous results of self-organised networks using additional point features.

2 Growing Self-organised Networks

Self-organised networks are a flexible alternative to deal with the problem of obtaining the implicit surface mesh, since they perform a clustering of the input data space maintaining, at the same time, the spatial-topological relations by means of neurons. The most representative network in this area is the *Self-Organised Map (SOM)*, proposed by Kohonen [5]. This network, whose size and structure are fixed, adjusts itself the values associated with each neuron by means of a cyclic process of comparing input data patterns and weight vectors, learning gradually the input data space. Nevertheless, *SOM* are limited to obtain a true fitted surface mesh because of the fixed topology and size, being necessary the introduction of a *growing* stage [6, 7]. This issue, within this area, can be addressed by means of two main approaches: *Growing Neural Gas (GNG)* [8] and *Growing Cell Structures (GCS)* [9], whose behaviours are based on *SOM*.

GCS adds new neurons maintaining network topology. In the initialisation stage, it is established a k -dimensional structure of $k + 1$ neurons and $(k + 1)k/2$ connections, the learning rate for the winning neuron (ε_b), the learning rate for its direct neighbours (ε_n), and the resource values (τ_i) and weights (w_i) for each neuron i (n_i). In the training stage, P input patterns are presented to the net A in groups of λ , called adaptation step, until the stop criterion is satisfied. For each input pattern (ξ), a winning neuron ($\phi_w(\xi)$) is determined by means of the distance between its weight and the input pattern

$$\|w_{\phi_w(\xi)} - \xi\| = \min_{r \in A} \|w_r - \xi\| \quad (1)$$

where $\|\cdot\|$ is the normal Euclidean vector. The weights associated with the winning neuron j and with its direct neighbours (N_j) are updated by means of

$$\begin{aligned} w_j &= w_j + \varepsilon_b(\xi - w_j) \\ w_c &= w_c + \varepsilon_n(\xi - w_c) \quad \forall c \in N_j \end{aligned} \quad (2)$$

and the resource value for the winning neuron is updated as

$$\tau_j = \tau_j + \|\xi - w_j\|^2 \quad (3)$$

Every λ patterns a new neuron is added between the neuron (n_i) with the highest resource value and its direct neighbour (n_j) whose resource value is the highest among the direct neighbours (N_i). Their weights and their resource values are updated.

GNG not only allows insertion of new neurons, but also modification and elimination of neurons and connections. In the initialisation stage weights, their resource values, the age parameter for each connection and the learning rates (ε_b and ε_n) are established. In the training stage, P input patterns are presented to the net in groups of λ , until one of the stop criteria is satisfied. For each input pattern (ξ), two winning neurons (n_1, n_2) are determined in the net A

$$\begin{aligned} n_1 &= \min \|\xi - w_c\|, \forall c \in A \\ n_2 &= \min \|\xi - w_c\|, \forall c \in A - \{n_1\} \end{aligned} \quad (4)$$

If these neurons are connected, the resource value (τ_{n_1}) and the weights of both n_1 and its direct neighbours (N_{n_1}) are updated like in *GCS*. The age for each connection of n_1 is increased. Otherwise, a new connection is established between them, initialising its age to 0. After this, the connections whose age parameter is higher than a certain threshold T as well as isolated neurons are removed. Every λ patterns a new neuron is added in a similar way than *GCS*.

3 3D Object Surface Reconstruction Methodology

The methodology proposed in order to deal with 3D object surface reconstruction problem defines several stages (Fig. 1): obtaining the edge points, training the growing self-organised networks, obtaining the surface mesh and finally surface reconstruction.

Fig. 1. Methodology steps

In the first step, 3D images are processed by means of the edge filters. We have used both *Canny edge detector* [10] and *lineal multi-scale filter* [11, 12]. The aim of this step is to obtain the set of surface edges which defines the 3D object surface. Each edge point is defined both by its 3D position (v_x, v_y, v_z) and by its 3D weighted gradient direction (α', β'), which are obtained during the filtering process. The 3D weighted gradient direction is calculated as

$$\begin{aligned} \alpha' &= R \sin(\arctan(G_x/G_y)) + R, & \alpha' &\in [0, 2R] \\ \beta' &= R \sin(\arctan(G_z/G_y)) + R, & \beta' &\in [0, 2R] \end{aligned} \quad (5)$$

where G_i is the gradient in the axis i , $\arctan(G_i/G_j)$ is the direction of the gradient vector measured with respect to the j axis, $\sin$ function allows to obtain continuous gradient values and R is a scale factor which weights the influence of the feature with respect to the 3D position. This is the main feature to guide the following reconstruction stage. 3D weighted gradient direction feature is used to make a distinction between close point positions. Using this feature, neighbour points belong to different surfaces if they have opposite gradient directions.

The set of edge points, previously obtained, is used for making two different sets of input patterns, used to train *GCS* and *GNG* in order to study the influence of using 3D gradient direction feature. In the first set, each input pattern is only represented by its 3D position (v_x, v_y, v_z) , and in the second, each of them is represented both by its 3D position and by its weighted gradient direction $(v_x, v_y, v_z, \alpha', \beta')$.

GCS nets can only add new neurons. During the training stage, neighbour neurons can move to distant positions. In such case, connections become inappropriate due to the fact that deletion of connections is not allowed, which also prevents the proper processing of scenes with several objects. A heuristic way is proposed to avoid inappropriate connections. Our approach consists of splitting the training stage into two steps: first, a fast training in order to avoid inappropriate connections and second, a slow training to fit the object surface. The main parameters which have influence on *GCS* behaviour are ε_b , ε_n and λ . In the first step, a fast training is necessary to adjust the net to the object shape. This is achieved by means of high values for ε_b and ε_n which obtain a fast movement of the winning neuron and its direct neighbours to the input patterns, and a low value for λ which implies that new neurons are added quickly between far neurons breaking inappropriate connections. The object shape is fitted without regarding the surface adjustment. In the second step, the surface is fitted by the mesh by means of a slow training with low values for ε_b and ε_n and a high value for λ . The training stops when the final size of the network is reached.

GNG nets resolve the previous *GCS* problems by means of addition, modification and deletion of neurons and connections. Besides the number of neurons, also used in *GCS* training, a new stop criterion, the global error measure, is employed. The training process stops when the resource value of each neuron of the net is below the global error measure parameter. The influence of both ε_b and ε_n parameters is less outstanding in the training process than in *GCS* net. The obtained surface mesh, composed by neurons and spatial-topological relations, is used as a basic 3D model by different applications.

4 Results

Two kinds of 3D input images have been employed: synthetic tubular images built up from a serie of fifteen 2D slices and anatomical images based on 64 computed tomography (*CT*) slices from the *Visual Human Project*. The images used for the filtering process (Fig. 2(b) and 3(b)) belong to the 3D scenes in Fig. 2(a) and 3(a), respectively. The synthetic images have been filtered by means of the lineal multi-scale filter obtaining the edge points of the object surface (Fig. 2(c) and 3(c)), whereas anatomical images have been processed by means of the Canny edge detector. The most suitable filter was chosen in each case. The input pattern sets employed in the next stage to train each net have been obtained from these edge points.

Initially, *GCS* net has been trained using only 3D position feature. As it can be seen from Fig. 2(d), a lot of inappropriate connections appear linking

distant neurons due to the fact that the net has been trained in a classical way [9]. However, following the heuristic approach proposed in the previous section these connections have been avoided (Fig. 2(e)). During the fast training stage (Table 1), the 150-neuron net fits the object shape and the slow training, adjusts the mesh to the object surface (Fig. 2(e)), where the final number of neurons fitted to the surface are 905. *GNG* net has been trained with the same input pattern set as in *GCS* case and it has obtained a more accurate surface mesh and a better adjustment (Fig. 2(g)) than *GCS*, specially regarding the original surface continuity (see both mesh details in Fig. 2(f)). The object surface (Fig. 2(h)) has been obtained from the previous obtained mesh.

Fig. 2. 2(a) 3D scene. 2(b) 3D object. 2(c) Obtained edge points. 2(d) *GCS* Surface mesh obtained using only 3D positions trained in a classical way. 2(e) Like in 2(d) but with our heuristic approach. 2(f) mesh detail (left) from 2(e) and (right) from the *GNG* mesh surface 2(g). 2(h) Surface reconstruction from 2(g)

From Fig. 3(c) two different input pattern sets have been constructed, one considering only 3D point position feature, and the other considering both 3D position and 3D weighted gradient direction features. In order to guide neuron movement properly, 3D position is approximately four times as important as 3D weighted gradient direction, using a scale factor $R = 32$ (Eq. 5) for 256x256 slices. It has been experimental checked that this weighting maximises the adjustment of the net. However, although the use of our heuristic approach and both features improve the obtained mesh, *GCS* net cannot deal in a proper way with several objects (Fig. 3(d) and 3(e)). *GNG* net has not been able to distinguish close object surfaces employing only 3D position feature (Fig. 3(f) and 3(g)). This problem has been avoided using both 3D position and 3D weighted gradient direction features (Fig. 3(h)) without changing the main training parameters (Table 2).

Fig. 3. 3(a) 3D scene. 3(b) 3D object. 3(c) Obtained edge points. 3(d) and 3(e) *GCS* Surface mesh obtained using our heuristic considering respectively, only 3D position feature and both 3D position and 3D gradient direction features. 3(f) *GNG* Surface mesh obtained using only 3D positions. 3(g) Mesh detail from 3(f). 3(h) Like in 3(f) but considering both 3D position and 3D weighted gradient direction features

For anatomical images, *GNG* net has been trained using both 3D position and 3D weighted gradient direction features (Fig. 4(b) and 4(c)). Discarding gradients, surfaces cannot be accurately distinguished (Fig. 4(a)).

The runtime needed for the training stage for each net, measured in seconds (s.), and the main training parameters used for each net are shown in Tables 1 and 2. The training stage has been processed in an Athlon XP at 1.667 GHz processor running Linux.

5 Summary and Conclusions

In this paper, self-organised networks are shown as a good choice for 3D surface reconstruction. 3D scenes with several objects cannot be processed properly by

Fig. 4. 4(a) *GNG* mesh surface using only 3D position. 4(b) and 4(c) different views of *GNG* surface mesh using 3D gradient direction. 4(d) Surface obtained from 4(b)

Table 1. *GCS* training parameters

Figure	λ	ε_b	ε_n	Input Patterns	Neurons	Runtime
2(d)	150	0.05	0.008	16560	905	7231 s.
2(e) fast training	6	0.3	0.09	16560	150	1289 s.
slow training	150	0.05	0.008	16560	905	5923 s.
3(d) fast training	10	0.2	0.09	22200	50	583 s.
slow training	150	0.05	0.008	22200	3000	16806 s.
3(e) fast training	5	0.2	0.09	22200	50	452 s.
slow training	150	0.05	0.008	22200	3000	17105 s.

Table 2. *GNG* training parameters

Figure	λ	ε_b	ε_n	Error	Input Patterns	Neurons	Runtime
2(g)	150	0.05	0.005	0.995	16560	1700	2527 s.
3(f)	150	0.05	0.005	0.995	22200	700	978 s.
3(h)	150	0.05	0.005	0.995	22200	800	1239 s.
4(b) & 4(c)	150	0.05	0.008	0.05	20000	8000	345601 s.

GCS net due to the impossibility of deletion of connections. Distant neurons linked by connections which do not belong to the object surface, appear during early steps of the training process of *GCS* net. The heuristic approach proposed in this paper avoids these connections improving the obtained surface mesh. However, the obtained surface mesh is non-uniform with a lot of broken areas due to the inability to create new connections. *GNG* nets can deal with several objects in a more proper way than *GCS* because the deletion and the establishment of new connections is allowed. Surface mesh obtained by means of *GNG* is more accurate to the original object surface with less time effort than with *GCS*. However, when only 3D point position feature is involved, several close surfaces cannot be distinguished. In this sense, 3D weighted gradient direction feature allows the net to distinguish each surface using the same training parameters. Finally, it can be said that the runtime needed in order to complete the training process (Tables 1 and 2), is lower for *GNG* net than for *GCS* with the same input set. In conclusion, *GNG* net is more suitable than *GCS* for obtaining the surface mesh.

The obtained neurons and their neighbourhood relations are suitable to reconstruct the object surface, being needed far less neurons than input patterns used in other techniques [1]. Therefore, several point features can be used in an easy way in both *GCS* and *GNG* nets improving the obtained surface mesh. In conclusion it can be said that using the methodology characterized in this paper and taking into account new features related to the geometry of the object (like gradient), *GCS* and *GNG* improve the obtained surface mesh.

Acknowledgements

This work has been partly supported by Ministerio de Ciencia y Tecnología (TIC2003-04649-C02-01) and Xunta de Galicia (PGIDIT03TIC10503PR).

References

1. Hoppe, H., DeRose, T., Duchamp, T., McDonald, J., Stuetzle, W.: Surface Reconstruction from Unorganized Points. *ACM SIGGRAPH Computer Graphics* **26** (1992) 71–78
2. Lorensen, W.E., Cline, H.E.: Marching Cubes: A High Resolution 3D Surface Construction Algorithm. *ACM SIGGRAPH Computer Graphics* **21** (1987) 163–169
3. Amenta, N., Bern, M.: Surface Reconstruction by Voronoi filtering. In: *Proceedings of the 14th Annual Symposium on Computational Geometry*. (1998) 39–48
4. Ivriissintzis, I.P., Jeong, W.K., Seidel, H.P.: Using Growing Cell Structures for Surface Reconstruction. In: *Proceedings of the Shape Modeling International 2003*, IEEE Computer Society (2003) 78–86
5. Kohonen, T.: The Self-organizing Map. In: *Proceedings of the IEEE*. Volume 78(9). (1990) 1464–1480
6. Fritzke, B.: Kohonen Feature Maps and Growing Cell Structures—a Performance Comparison. In Hanson, S.J., Cowan, J.D., Giles, C.L., eds.: *Advances in Neural Information Processing Systems 5*, NIPS 1992, Denver. Volume 5. (1993) 115–122
7. Fritzke, B.: Growing Self-organizing Networks – Why? In: *ESANN’96*, European Symposium on Artificial Neural Networks. (1996) 61–72
8. Fritzke, B.: A Growing Neural Gas Network Learns Topology. In: *Advances in Neural Information Processing Systems*. Volume 7. (1995) 864–869
9. Fritzke, B.: Growing Cell Structures – a Self-organizing Network for Unsupervised and Supervised Learning. *Neural Networks* **7** (1994) 1441–1460
10. Trucco, E., Verri, A.: *Introductory Techniques for 3-D Computer Vision*. 1 edn. Prentice Hall (1998)
11. Sato, Y., Nakajima, S., Nobuyuki, S., Atsumi, H., Yoshida, S., Koller, T., Gerig, G., Kikinis, R.: Three-dimensional Multi-scale Line Filter for Segmentation and Visualization of Curvilinear Structures in Medical Images. *Medical Image Analysis* **2** (1998) 143–168
12. Koller, M., Gerig, G., Székely, G., Dettwiler, D.: Multiscale Detection of Curvilinear Structures in 2-D and 3-D Image Data. In: *5th International Conference on Computer Vision*. (1995) 864–869