

Neural Processing Letters manuscript No. (will be inserted by the editor)

3D surface reconstruction of noisy point clouds using Growing Neural Gas

3D Object/Scene reconstruction

Sergio Orts-Escolano · Jose Garcia-Rodriguez ·
Vicente Morell · Miguel Cazorla · Jose Antonio
Serra Perez · Alberto Garcia Garcia

Received: date / Accepted: date

Abstract With the advent of low-cost 3D sensors and 3D printers, scene and object 3D surface reconstruction has become an important research topic in the last years. In this work, we propose an automatic (unsupervised) method for 3D surface reconstruction from raw unorganized point clouds acquired using low-cost 3D sensors. We have modified the Growing Neural Gas (GNG) network, which is a suitable model because of its flexibility, rapid adaptation and excellent quality of representation, to perform 3D surface reconstruction of different real-world objects and scenes. Some improvements have been made on the original algorithm considering colour and surface normal information of input data during the learning stage and creating complete triangular meshes instead of basic wire-frame representations. The proposed method is able to successfully create 3D faces online, whereas existing 3D reconstruction methods based on Self-Organizing Maps (SOMs) required post-processing steps to close gaps and holes produced during the 3D reconstruction process. A set of quantitative and qualitative experiments were carried out to validate the proposed method. The method has been implemented and tested on real data, and has been found to be effective at reconstructing noisy point clouds obtained using low-cost 3D sensors.

Keywords GNG · 3D Reconstruction · low-cost 3D sensor · scene reconstruction · object reconstruction

Sergio Orts-Escolano, Jose Garcia-Rodriguez and Jose Antonio Serra Perez are with the Department of Computer Technology of the University of Alicante (email: {sorts, jgarcia, jserra, agarcia}@dtic.ua.es).

Vicente Morell and Miguel Cazorla are with the Computer Science and Artificial Intelligence Department of the University of Alicante (email: {vmorell, miguel}@dccia.ua.es).

Sergio Orts-Escolano
Carretera de San Vicente del Raspeig, s/n, 03690, Alicante
Tel.: +34-965903400
Fax: +34-965903464
E-mail: sorts@dtic.ua.es

S. Author
second address

1 Introduction

Many well established techniques proposed solutions to the 3D representation and surface reconstruction problem from a geometric point of view. However, these algorithms required long times to process the input point cloud and do not scale properly with very large data [1, 2]. Moreover, these traditional geometric approaches do not manage non-stationary distributions and do not deal with the lack of a priori information about the input space, e.g. the presence of multiple shapes in the point cloud and noise induced by the sensors. Even more recent methods that deal with point clouds fail to provide accurate solutions for some of the aforementioned constraints [3].

Considering the 3D representation problem from a computational intelligence approach and based on self-organizing maps, a different perspective to obtain 3D reconstructions is proposed. These methods could be defined as flexible and growing models considering if the structure of the selected map has a priori topology or otherwise it grows until a condition is fulfilled. Moreover, we can find some similarities or correspondences between the neural network map and the 3D representation obtained. Nodes of the neural network map correspond to vertices of a mesh and connection between nodes correspond to the edges. Therefore, in this work the terms node and vertex, and connection and edge are used interchangeably. From this perspective, some methods were proposed based on self-organizing maps.

In [4] it is proposed the use of Kohonen's self-organizing maps for surface reconstruction using as an input data unorganized point clouds. Moreover, since Kohonen's map does not produce regular faces, an edge collapse operation was introduced eliminating dangling faces. This approach presented some drawbacks as if the real object surface has concave structures, applying Kohonen's learning algorithm has some difficulties to correctly approximate those parts. In addition, as the Kohonen's algorithm has a high computational cost, the single thread CPU implementation presented in this work took more than one hour to represent the Stanford bunny model. This method was only tested with synthetic data and the bunny model, which is comprised of 34,834 points. Junior et al. [5], extended [4] introducing new mesh operators that allowed it to perform improvements on the surface geometry: edge swap, edge collapse, vertex split and triangle subdivision. Moreover, the method introduced a new step to remove unstable vertices using the mean distance and the standard deviation of the 3D representation regarding the sampled input space. Although this new approach improved the surface geometry, the method does not deal with concave or non-convex regions and the initial structure of the representation has to be pre-established considering the topology of the input space. The fixed structure of the SOM does not learn the spatial relationships between the vertices and therefore does not generate a model that accurately represents the shape of the input space. To overcome this problem, some methods based on Growing SOMs were proposed.

One of these SOM-based methods is the Growing Cell Structures (GCS) algorithm [6], which is a model formed incrementally. However, it constraints the connections between the nodes, so any model produced during the training stage is always topologically equivalent to the initial topology. In [7] it is used the GCS algorithm to reconstruct objects surface. Meshes operators are used to change the connectivity of the mesh and therefore final topology is always equivalent to the initial mesh.

The Topology Representing Networks (TRN), proposed by [8], does not have a fixed structure and also does not impose any constraint about the connection between the nodes. In contrast, this network has a pre-established number of nodes, and therefore, it is not able to generate models with different resolutions. The algorithm was also coined with the

term Neural Gas (NG) due to the dynamics of the feature vectors during the adaptation process, which distributes themselves like a gas within the data space. Barhak [9] proposed a NG-based surface reconstruction algorithm since this network has the ability to accurately represent the topology of a point cloud. However, as the NG has a fixed number of nodes, it is necessary to have some a priori information about the input space to pre-establish the size of the network. This model was extended by [10] proposing the Growing Neural Gas (GNG) network, which combined the flexible structure of the NG with a growing strategy. Moreover, the learning adaptation step was slightly modified. This extension enabled the neural network to use the already detected topological information while training in order to conform to the geometry. This approach has the capability to add neurons while preserving the topology of the input space.

As the original GNG algorithm does not produce faces and the generated map is a wire-frame representation model, some works extended the original algorithm to produce faces. In [11], the GNG network is employed to model a point cloud and those regions that need further sampling in order to obtain a more accurate model. Rescanning at higher resolution is performed for each identified region of interest and a multi-resolution model is built. In this work, only nodes of the generated map are used as the work is focused on sampling capabilities of the GNG. MGNG [12] applied some postprocessing steps in order to perform surface reconstruction once the map is generated using the original GNG algorithm. Most of these approaches were tested against CAD models or synthetic data and only few experiments were performed on objects acquired with 3D sensors. In [13], the GNG algorithm was modified in order to produce topological faces. The extended method was called Growing Self-Reconstruction Maps (GSRM) and some learning steps as the Competitive Hebbian Learning step (CHL) and the operations of vertex insertion and removal were also modified. Most experiments of this work were performed on the Stanford dataset, which had been previously filtered and therefore the surface reconstruction step does not have to deal with noisy input spaces produced by common 3D sensors. In [13, 9] the Competitive Hebbian Learning was extended considering the creation of 2-manifold meshes and face reconstruction. However, it was also required to apply some post-processing steps to create a complete model.

Although the use of the SOM-based techniques as NG, GCS or GNG for 3D input space representation and surface reconstruction has already been studied and successful results have been reported, there are some limitations that still persist. Most of these works assumed noise-free point clouds. Therefore, applying these methods on challenging real-world data obtained using noisy 3D sensors have not been object of study yet. Moreover, with the advent of low cost RGB-D cameras as the Microsoft Kinect ¹ partial point clouds have to be considered. Besides providing 3D information, these devices also provide colour information, feature that was not considered in the revised works.

In this work, we extended the method presented in [14] for 3D object reconstruction using the GNG algorithm. This work extends the already proposed method considering surface normal information during the learning process and performing an extensive evaluation on noisy point clouds acquired using low-cost sensors. Moreover, a detailed study is carried out analysing the quality of the 3D reconstructed model. We also compared the proposed method against the state-of-the-art Poisson surface reconstruction method [15], showing quantitative and qualitative results using different input data. Finally, more experiments were performed considering the reconstruction of 3D scenarios.

¹ Kinect for XBox 360: <http://www.xbox.com/kinect> Microsoft

The rest of the paper is organized as follows: first, a section describing briefly the GNG algorithm is presented. In Section 3 and Section 4 the modification of the GNG algorithm, considering colour and surface normal information during the learning process, is detailed. Section 5 presents and depicts the proposed algorithm for face creation during the network learning process. In Section 6 we present some experiments and discuss results obtained using our novel approach compared to existing methods. Finally, in Section 7 we give our conclusions and directions for future work.

2 GNG Algorithm

With Growing Neural Gas (GNG) [10] method a growth process takes place from minimal network size and new neurons are inserted successively using a particular type of vector quantization. To determine where to insert new neurons, local measures are gathered during the adaptation process and each new neuron is inserted close to the neuron with highest accumulated error. At each adaptation step a connection between the winner and the second-nearest neuron is created as dictated by the CHL algorithm. This is continued until an ending condition is fulfilled, as for example evaluation of the optimal network topology or a fixed number of neurons is reached. The network is specified as:

- A set N of nodes (neurons). Each neuron $c \in N$ has its associated reference vector $w_c \in \mathbb{R}^d$. The reference vectors can be regarded as positions in the input space of their corresponding neurons.
- A set of edges (connections) between pairs of neurons. These connections are not weighted and its purpose is to define the topological structure. An edge aging scheme is used to remove connections that are invalid due to the continuous update of neurons reference vector during the adaptation process.

The GNG learning algorithm is presented in Algorithm 1.

3 Colour interpolation

As modern 3D sensors provide colour information, the proposed method was modified regarding the original version considering also point cloud colour. Input space dimension is increased from 3 to 6 adding red, green and blue colour components. Now the input distribution is defined in \mathbb{R}^d where $d = 6$. Most SOM-based approaches already presented only considered spatial information as neuron’s weight vector w_c , so we modified the learning algorithm adding colour to the neuron’s weight vector w_c and considering it during the learning process, now the dimension of the neuron’s weight vector is 6 including spatial and colour information. Colour values were normalized ranging from 0.0 and 1.0. Colour information is considered during the weight adaptation process but it was not included in the CHL (winning neurons) stage as we still are focused on preserving the topology of the input space. Therefore, winning neuron stage only compute Euclidean distance using x, y, z components. Figure 1 shows how the GNG method generated a down-sampled version of captured coloured point clouds, interpolating the colour of original observations and achieving a good topological fitting for different objects and scenes. We called this version Colour-GNG.

In order to validate and compare the colour version of the GNG, we implemented a different strategy to consider point cloud colour information. Instead of adding colour information to the learning process, a post-processing step to compute colour information is


```

input : N-dimensional input data
output: N-dimensional map
1 Start with two neurons  $a$  and  $b$  at random positions  $w_a$  and  $w_b$  in  $R^d$ .
2 repeat
3   for  $patterns=0$  to  $\lambda$  do
4     Generate at random an input pattern  $\xi$  according to the data distribution  $P(\xi)$  of each input
       pattern.
5     Find the nearest neuron (winner neuron)  $s_1$  and the second nearest  $s_2$ .
6     Increase the age of all the edges emanating from  $s_1$ .
7     Add the squared distance between the input signal and the winner neuron to a counter error
       of  $s_1$  such as:
           
$$\Delta error(s_1) = \|w_{s_1} - \xi\|^2 \quad (1)$$

8     Move the winner neuron  $s_1$  and its topological neighbors (neurons connected to  $s_1$ )
       towards  $\xi$  by a learning step  $\epsilon_w$  and  $\epsilon_n$ , respectively, of the total distance:
           
$$\Delta w_{s_1} = \epsilon_w (\xi - w_{s_1}) \quad (2)$$

9     forall the direct neighbors  $n$  of  $s_1$  do
10      |
           
$$\Delta w_{s_n} = \epsilon_n (\xi - w_{s_n}) \quad (3)$$

11    end
12    if  $s_1$  and  $s_2$  are connected by an edge then
13      | Set the age of this edge to 0.
14    else
15      | Create the connection between  $s_1$  and  $s_2$ .
16    end
17    Remove the edges larger than  $a_{max}$ 
18    if any neuron is isolated (without emanating edges) then
19      | Remove those neurons as well.
20    end
21  end
22  Insert a new neuron as follows:
23    Determine the neuron  $q$  with the maximum accumulated error.
24    Insert a new neuron  $r$  between  $q$  and its further neighbor  $f$ :
           
$$w_r = 0.5(w_q + w_f) \quad (4)$$

25    Insert new edges connecting the neuron  $r$  with neurons  $q$  and  $f$ , removing the old edge
       between  $q$  and  $f$ .
26    Decrease the error variables of neurons  $q$  and  $f$  multiplying them with a constant  $\alpha$ . Initialize
       the error variable of  $r$  with the new value of the error variable of  $q$  and  $f$ .
27    Decrease all error variables by multiplying them with a constant  $\gamma$ .
28 until number of neurons reached;

```

Algorithm 1: Pseudo-code of the GNG algorithm.

added to the process. Once the network has been adapted to the input space (original GNG) and it has completed the learning process, each neuron of the network computes colour information from closest input patterns. Colour information of each neuron is calculated as the average of weighted values of the K-nearest input patterns, obtaining an interpolated value of the surrounding point. Colour values are weighted using Euclidean distance from input pattern to its closest neuron reference vector. k-Nearest Neighbours (kNN) are obtained using a radius search process, using as a radius the resolution of the generated map by the GNG algorithm. Therefore, RGB colour for each neuron is calculated using the following equation:

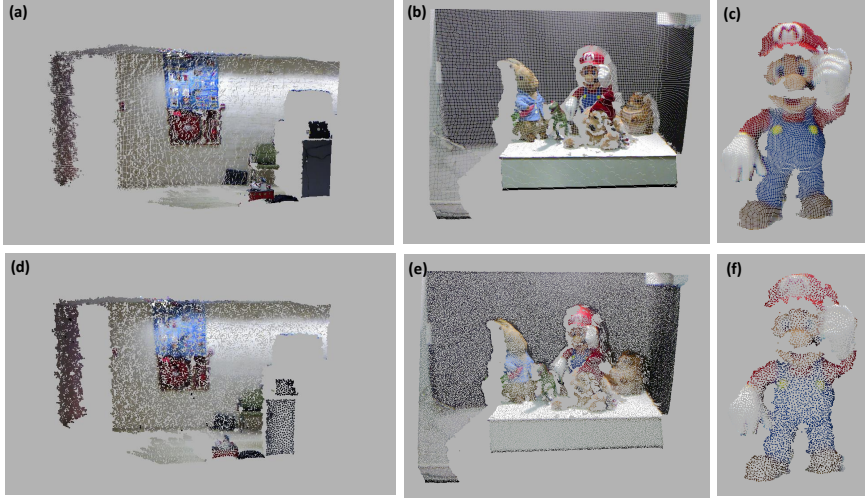


Fig. 1 Different objects and scenes are down-sampled using the Colour-GNG representation. (a),(b),(c) show original point clouds. (d),(e),(f) show down-sampled point clouds using the proposed method. Original point clouds: (a) 307,200 points; (b) 287,218 points; (c) 12,883 points. Down-sampled point clouds: (d) 25,000 points; (e) 50,000 points; (f) 5,000 points.

$$RGB_i = \psi \sum_{\forall j \in N_i} (RGB_j \cdot w(j - i)) \quad (5)$$

where N_i represents the nearest input patterns of the neuron i , i is the neuron being processed and $w(j - i)$ is the distance weighted function between the neighbouring pattern j and the neuron itself. Using that distance weighted function, the weight of the colour of the input pattern decays exponentially as the distance to the neuron increases. ψ is a normalization factor that makes RGB components range between 0.0 and 1.0.

$$w(j - i) = e^{-\|j - i\|} \quad (6)$$

Figure 2 visually shows this process. Although this search is accelerated using a Kd-tree structure, it is considerably slower than the colour version of the GNG. Colour-GNG is able to adapt its neurons' weights in the same learning process, fitting accurately to the input space.

Figures 3 and 4 show various observations that have been created using both approaches. Colour-GNG produces a map that successfully interpolates input colour information producing a useful down-sampled map. Moreover, results were similar to those obtained with the colour interpolation post-processing step.

Finally, some quantitative results are presented in Table 1 showing the mean error between estimated colours using the post-processing interpolation step and the proposed Colour-GNG method. The mean error is computed over the three components of the RGB model and it is truncated to the closest integer value. The maximum error obtained in a component between estimated colours using the post-processing step and the Colour-GNG is less than four units, considering that each component is represented using unsigned char format. In

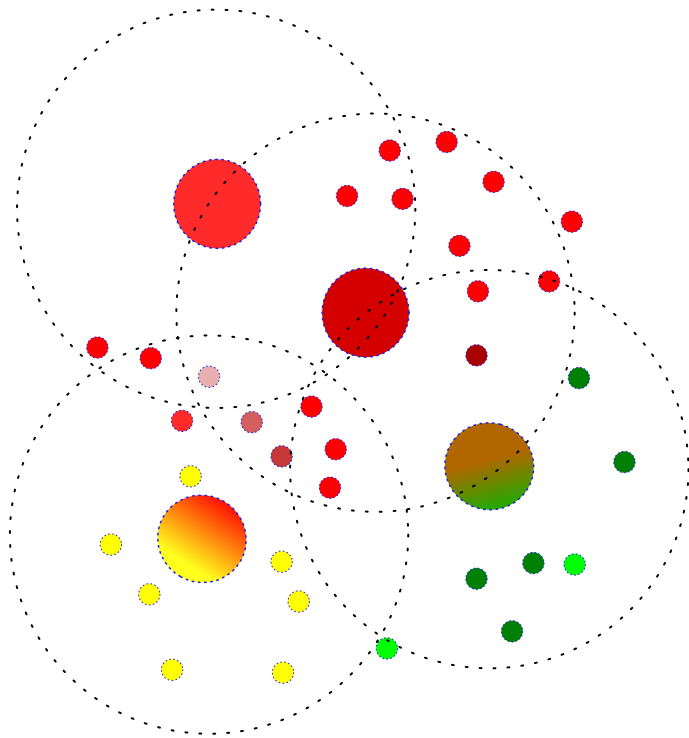


Fig. 2 Colour interpolation. Colour assigned to each neuron (large circles) is calculated as the averaged weighted sum of input space samples (small circles) within a search radius (dotted circle).

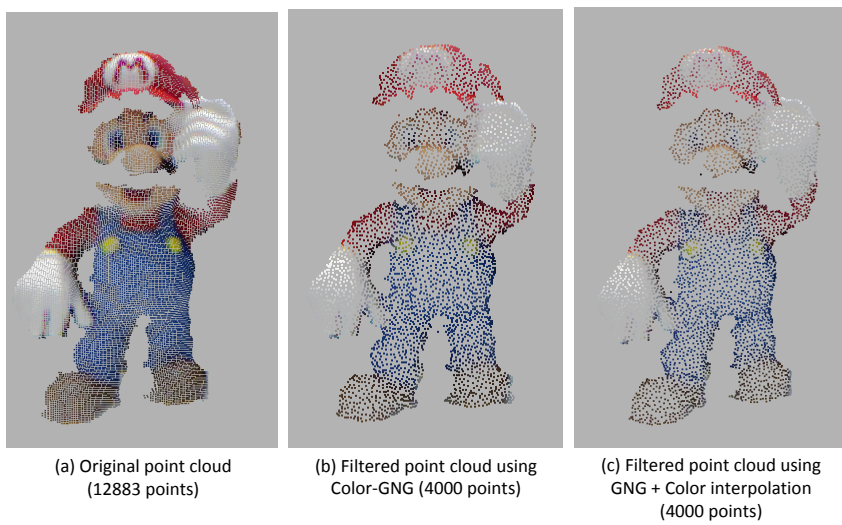


Fig. 3 Mario figure is down-sampled using the Colour-GNG method. Results are similar to those obtained with the colour interpolation post-processing step.



Fig. 4 Two different scenes captured using the Kinect sensor are represented using the Colour-GNG method. Results are similar to those obtained with the colour interpolation post-processing step.

this experiment we used values between 0 and 255 to facilitate understanding of the obtained results.

With these results we can conclude that Colour-GNG method is able to obtain similar results compared to complex post-processing steps and reducing the processing time.

	Mean error		
	Red	Green	Blue
Scene 1 - 20,000n 50λ	1	1	2
Scene 1 - 50,000n 100λ	1	1	1
Scene 2 - 20,000n 50λ	4	4	4
Scene 2 - 50,000n 100λ	2	2	2
Object 1 - 3,000n 50λ	2	3	3
Object 1 - 5,000n 100λ	1	2	2
Object 2 - 3,000m 50λ	3	3	3
Object 2 - 5,000n 100λ	2	2	2

Table 1 Colour mean error between computed colours using a post-processing interpolation step and the Colour-GNG.

4 Preserving surface normal information

In this work, we also modified the original GNG algorithm for considering surface normal information which is also present in the input data. We considered adding surface normal information to the reduced representation created by the GNG algorithm. Since common processing steps in 3D computer vision problems as 3D keypoint detection and 3D feature extraction algorithms use surface normal information [16, 17, 18]. As it was demonstrated in [19], the reduced representation created by the GNG algorithm has benefits for removing noise from data provided by low-cost sensors, improving the performance in some computer vision problems as 3D scene recognition. Normal information is included during the learning step of the GNG, and therefore, we are able to produce a reduced representation of the surface normal information. This new representation improves the performance of common 3D keypoint detection and feature extraction algorithms, since less data has to be processed.

The input space dimensionality has been increased from 6 to 9 adding surface normal information ($normal_x, normal_y, normal_z$). Finally the input distribution is defined in \mathbb{R}^d where $d = 9$. As we previously commented, most SOM-based approaches already proposed only considered spatial information as neuron’s weight vector w_c , so we modified the learning algorithm adding surface normal information to the neuron’s weight vector w_c and considering it during the learning process. Normal information is considered during the weight adaptation process but it is not included in the CHL (winning neurons) stage as we still are focused on preserving the topology of the input space.

Figure 5 shows how the GNG method generates down-sampled versions of different point clouds with surface normal information, interpolating normals and achieving a good topological fitting for different objects and scenes. In this figure we do not show colour information to emphasize how the GNG is able to learn surface normal information from the input data and create a reduced representation that contains both, the geometric structure and consistent surface normal information.

Figure 6 shows an example of an indoor scene obtained using the Kinect sensor (left). Normal information was computed on the original point cloud using Principal Component Analysis (PCA) [20] (middle). Then, a reduced representation was generated using the extended GNG algorithm considering surface normal information (right). We can appreciate on the right part of the figure that the reduced representation has consistent surface normal information regard the original input source (middle).

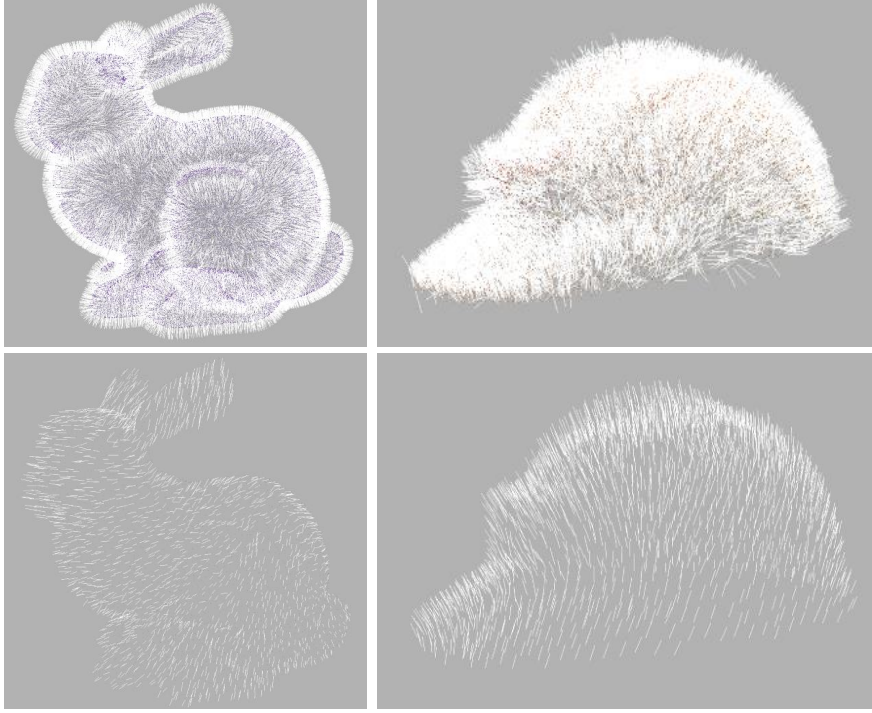


Fig. 5 Top: Normal information computed on the original point clouds using PCA. Bottom: reduced representations created using the extended version of the GNG considering surface normal information. GNG connections are not shown to emphasize surface normal information.



Fig. 6 Left: indoor scene obtained using the Kinect sensor. Middle: surface normal information computed on the original point cloud. Right: Down-sampled surface normal information using the GNG algorithm

5 3D surface reconstruction

Three-dimensional surface reconstruction is not considered in the original GNG algorithm as it only generates wire-frame models. As we presented in Section 1, some previous works as [12, 13, 9] have already considered the creation of 3D triangular faces modifying the original GNG algorithm. In these works, mesh operators as edge split, edge collapse, triangle subdivision, etc, were implemented to achieve that purpose. Although most of these works produced 3D meshes they have a common drawback or disadvantage, the need of post-

processing steps to finally close holes and missing faces that were not created during the learning stage.

Figure 7 shows the result of using an existing GNG-based method for surface reconstruction [13] without applying post-processing steps. The reconstructed model has a lot of gaps and holes that makes the model not suitable for computer vision applications.

In this section, we detail the proposed extension of the already described GNG method to generate full coloured 3D meshes without applying post-processing steps.

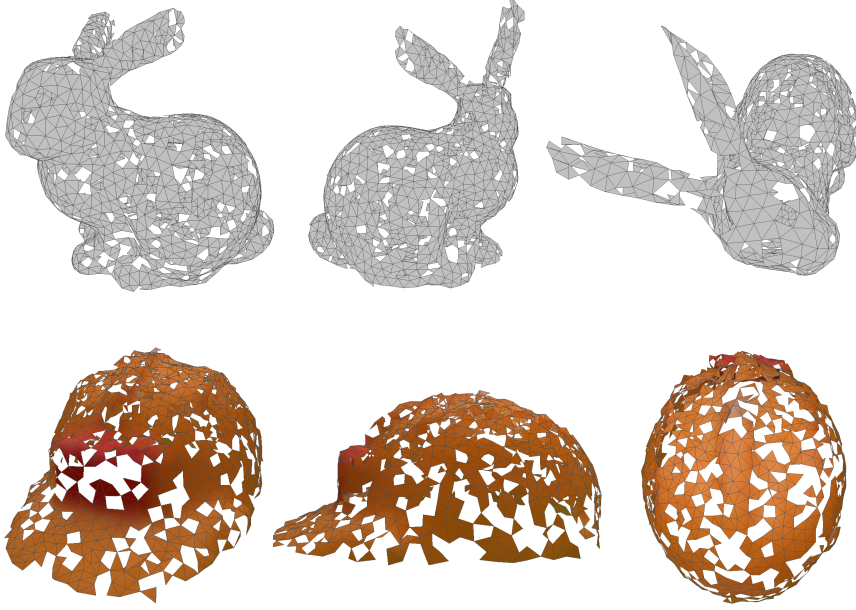


Fig. 7 Different views of reconstructed models using an existing GNG-based method for surface reconstruction. Post-processing steps were avoided causing gaps and holes in the final 3D reconstructed models.

5.1 Extended CHL

Original CHL, presented in Section 2, only considered the creation of edges between neurons producing wire-frame 3D representations. Therefore, it is necessary to modify this process in order to create triangular faces during the learning process. Based on [13] and [9] we extended the CHL rule developing a method able to produce full 3D meshes. In contrast to existing methods mentioned above, our extension does not need post-processing steps. The 3D mesh is created during the learning stage.

The edge creation stage, was also extended considering the creation of triangular faces during this process. Algorithm 2 describes our extended CHL to produce triangular faces.

In order to avoid non-manifold and overlapping edges, the edge creation step was modified restricting the creation of edges if the winning neurons s_1 and s_2 have already more than

```

input : A point cloud
output: 3D mesh
1 For each input pattern presented to the network, the two nearest neurons to the input pattern are
  selected as winning neurons  $s_1$  and  $s_2$ ;
2 if  $s_1$  and  $s_2$  are already connected by an edge then
3   Set edge age to 0 in order to "reinforce" it;
4   Check edge removal mechanism based on the Tales Sphere;
5   if  $s_1$  and  $s_2$  have one or two common neighbours then
6     foreach common neighbour  $n_i$  do
7       Create a face  $f$  using  $s_1$ ,  $s_2$  and  $n_i$ ;
8     end
9   end
10  if  $s_1$  and  $s_2$  have one or zero common neighbours then
11    if There exist two neighbours  $n_1$  and  $n_2$  of  $s_1$  and  $s_2$  respectively that are connected
      and are not common to  $s_1$  and  $s_2$  then
12      Triangulate rectangular hole (Figure 8d): Create two faces using  $s_1$ ,  $s_2$ ,  $n_1$  and  $s_2$ ,
         $n_1$ ,  $n_2$ ;
13    end
14    else if There exist two neighbours  $n_1$  and  $n_2$  of  $s_1$  and  $s_2$  respectively that are not
      connected between them and are not common to  $s_1$  and  $s_2$  and.  $n_1$  and  $n_2$  have a
      common neighbour  $n_3$  then
15      Triangulate pentagonal hole (Figure 8e): Create three faces using  $s_1$ ,  $s_2$ ,  $n_2$ ;  $s_1$ ,
         $n_2$ ,  $n_3$  and  $s_1$ ,  $n_1$ ,  $n_3$ ;
16    end
17  end
18 else
19   if  $s_1$  and  $s_2$  have two common neighbours  $n_1$  and  $n_2$  then
20     if  $n_1$  and  $n_2$  are already connected (Figure 8c) then
21       Edge between  $n_1$  and  $n_2$  is removed;
22       Create an edge between  $s_1$  and  $s_2$ ;
23       Faces coincident to  $n_1$  and  $n_2$  are removed;
24       Create two faces using  $n_1$ ,  $s_1$ ,  $s_2$  and  $n_2$ ,  $s_1$ ,  $s_2$ ;
25     else
26       Create an edge between  $s_1$  and  $s_2$ ;
27       Create two faces using  $n_1$ ,  $s_1$ ,  $s_2$  and  $n_2$ ,  $s_1$ ,  $s_2$ ;
28       (Figure 8b);
29     end
30   else
31     Create edge between  $s_1$  and  $s_2$ ;
32     if  $s_1$  and  $s_2$  have one common neighbour  $n_1$  (Figure 8a) then
33       Create a face  $f$  using  $s_1$ ,  $s_2$  and  $n_1$ ;
34     end
35   end
36 end

```

Algorithm 2: Pseudo-code of the extended CHL stage.

two common neighbours. This constraint helps avoiding edges with more than two incident triangles. Then, for every sampled point, a face is created whenever the already existing edges or the new ones form a triangle. Moreover, if the creation of faces would produce edges with more than two incident faces, then the face is not created avoiding overlapped triangles and non manifold meshes. During the creation of triangular faces it is checked if the face to be created already exist, in that case, the face is not created. Figure 8 shows common situations produced during the CHL and how our method create edges and triangular faces in those cases.

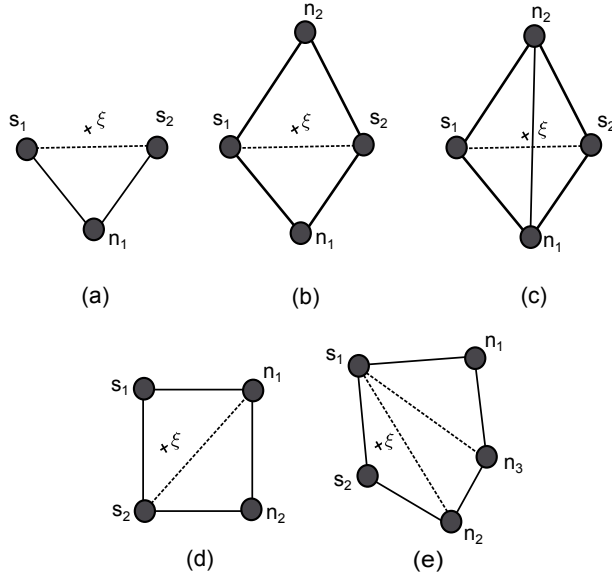


Fig. 8 Considered situations for edge and face creation during the extended CHL.

The age scheme presented in the original GNG algorithm was also considered to remove those edges that have an age higher than a given threshold age_{max} . This age scheme was extended including the removal of faces that shared this edge. Furthermore, to obtain regular triangular faces we included another constraint that was introduced in [21]. This constraint is based on the Thales sphere concept. For every edge that already existed in the CHL process, this mechanism computes the angle between the vectors formed by $s_1 - s_2$ and $s_1 - n_i$ where n_i is a common neighbour of s_1 and s_2 . If this angle $\theta > \theta_{max}$ the edge between s_1 and n_i is removed. Faces incident to this edge are also removed. Different values for θ_{max} were tested, obtaining regular triangles for θ_{max} values between $2/3\pi$ and $3/4\pi$. Figure 9 shows this process.

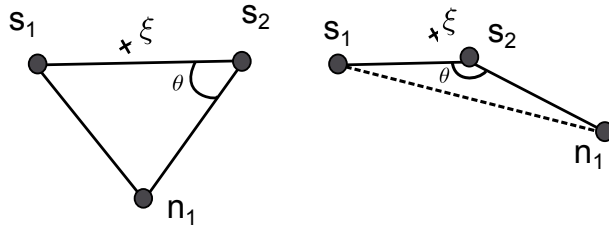


Fig. 9 Edge removal constraint based on the Thales sphere. Left: The triangle formed by these 3 neurons is close to a right triangle, therefore it is not removed. Right: The edge connecting s_1 and n_i is removed as the angle formed by vectors $s_2 - s_1$ and $n_i - s_1$ is larger than $3/4\pi$. Moreover, the triangle formed by these edges is also removed.

5.2 Inserting and deleting neurons

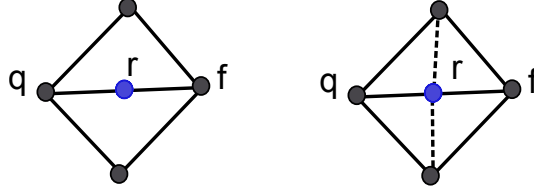


Fig. 10 Face creation process during the insertion of new neurons. Left: neuron insertion between the neuron q with highest error and its neighbour f with highest error. Right: four new triangles and two edges are created considering r , q and r .

The neuron insertion process was also modified. Every time a neuron is inserted in the network, an edge between the neurons with highest error is removed and therefore, triangles incident to this edge are also removed. If it is possible, new faces are created along with the new neuron (Figure 10).

Finally, if the given number of neurons is reached, all the input patterns are presented to the network in order to close possible gaps and holes that were generated during the learning process.

6 Experiments

In this section, different experiments are shown validating the capabilities of our extended GNG method to create 3D meshes. The proposed method is able to learn colour information and therefore to create coloured 3D meshes. 3D models were rendered using colour information stored in the neurons and a triangle smooth-shading technique. First, a quantitative study is performed using synthetic models and adding different levels of noise to the ground truth model (validation set). Using the ground truth models and the generated ones adding noise, we are able to measure the error produced by our 3D reconstruction method. In addition, our method is compared against the state-of-the-art Poisson surface reconstruction algorithm using the synthetic models with added noise. Second, data coming from different 3D sensors is reconstructed using the proposed method and it is visually compared against results obtained using Poisson surface reconstruction. In particular, we focused on data obtained from low-cost 3D sensors since these sensors provide us with noisy and unstable data.

All methods have been developed and tested on a desktop machine with an Intel Core i3 540 3.07Ghz. These methods have been implemented in C++. Moreover, Poisson surface reconstruction method, some metrics like the Hausdorff distance [22] and the visualization framework have been implemented using the PCL library [23] and the Meshlab tool.

6.1 Dataset

In order to compute the error produced by the proposed 3D surface reconstruction method, we need to manage models that provide us error-free information (ground truth). Scanning

real-objects do not allow us to have ground truth information about the real measure of these objects/scenes, therefore we decided to use synthetic models (Figure 11) which were generated using 3D design tools, in order to have ground truth data. As most 3D scanners produce noisy acquisitions, which is one of the concerns of this work, caused by the reflectance of the surface and other implicit factors, we added some Gaussian noise to the synthetic models to simulate this behaviour. The Stanford bunny from the Stanford 3D scanning repository [24] is considered as a noise-free model since it has been obtained using a high resolution sensor and some post-processing steps for removing noise and closing holes have been applied.

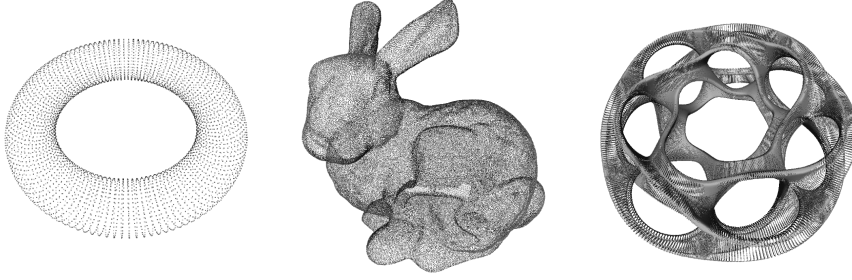


Fig. 11 Synthetic 3D models used in the experiments. From left to right: Torus (4,800 points), Stanford Bunny (72,027 points) and Heptoroid (286,678 points).

Moreover, we used other models from different sources for testing the proposed algorithm. We used some models from the Stanford 3D scanning repository [24], we also tested the dataset used in [25] obtained using a Kinect sensor and some models from [26], which were acquired using a Minolta 3D laser. Finally, we tested the proposed methods with data obtained at the University of Alicante (indoors) using low-cost sensors like the Microsoft Kinect and the Asus Xtion Pro.

We added different levels of Gaussian noise to the synthetic models to test our proposal and to see how the different methods are able to deal with this kind of noise, which is common in 3D sensors. The results of applying different levels of noise to the synthetic models are shown in Figure 12.

6.2 Surface reconstruction quality

To demonstrate the validity of our proposal, several experiments have been carried out comparing GNG 3D reconstruction results with Poisson surface reconstruction method. Different configurations for the GNG algorithm have been tested and compared using quality measures.

We first compared the proposed method against one of the state-of-the-art methods, the Poisson surface reconstruction algorithm. We have used Poisson algorithm for creating meshes from different noisy input clouds that were shown above (synthetic model). As in Poisson method, it is possible to define the level of depth (accuracy) we want to obtain in the final reconstruction we performed experiments with different levels of accuracy and therefore number of points.

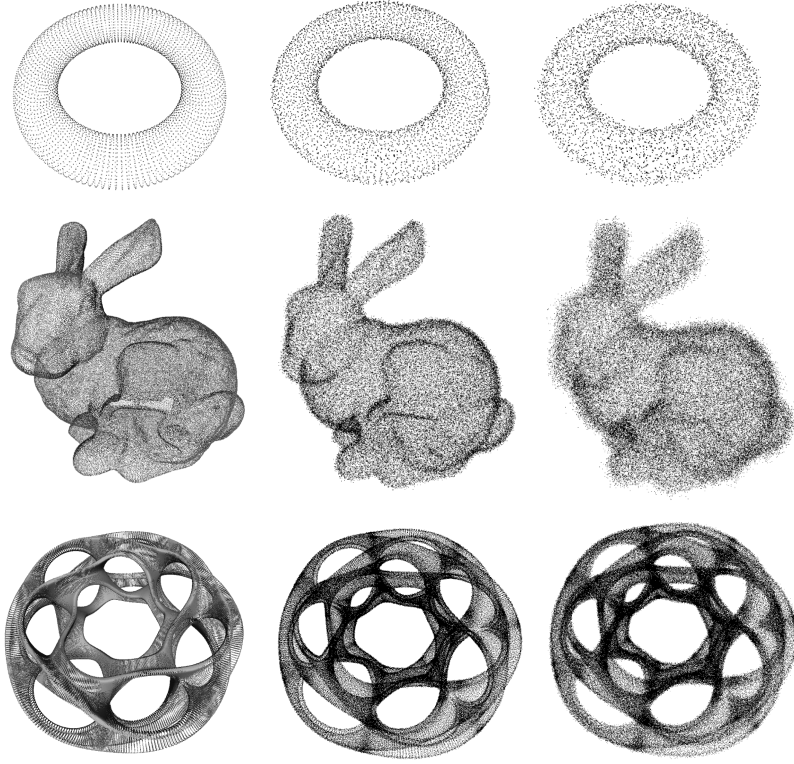


Fig. 12 Synthetic models with different levels of Gaussian noise. From left to right: $\sigma = 0$; $\sigma = 1$; $\sigma = 2.5$; (millimetres).

Figure 13 shows a colour map of the computed Hausdorff distance. It ranges from red to blue, using red for the areas with largest error and blue for the areas with lowest error. This figure shows the results of applying both reconstruction methods to noisy synthetic point clouds with different Gaussian error. In order to evaluate these reconstructions in a quantitative way, we computed Hausdorff distance (using the Metro tool [22]) between the synthetic mesh model and the reconstructed ones from the noisy data. In this way, we can measure the quality of the reconstruction obtained with the different methods using noisy data. Poisson algorithm performs surface reconstruction using surface normal information, and since normal information is very sensitive to the presence of noise, it is not able to correctly reconstruct some parts of the model.

In Figure 13 (bottom) we see a challenging high-genus model, the heptoroid. The genus² of this heptoroid surface is 22, and it has many hyperbolic regions. Despite the topological complexity of this surface, we are able to produce a well-shaped mesh whereas the Poisson method produces a reconstructed model with some errors and inconsistent geometry (red areas).

² Genus: A topologically invariant property of a surface defined as the largest number of nonintersecting simple closed curves that can be drawn on the surface without separating it. Roughly speaking, it is the number of holes in a surface. [27]

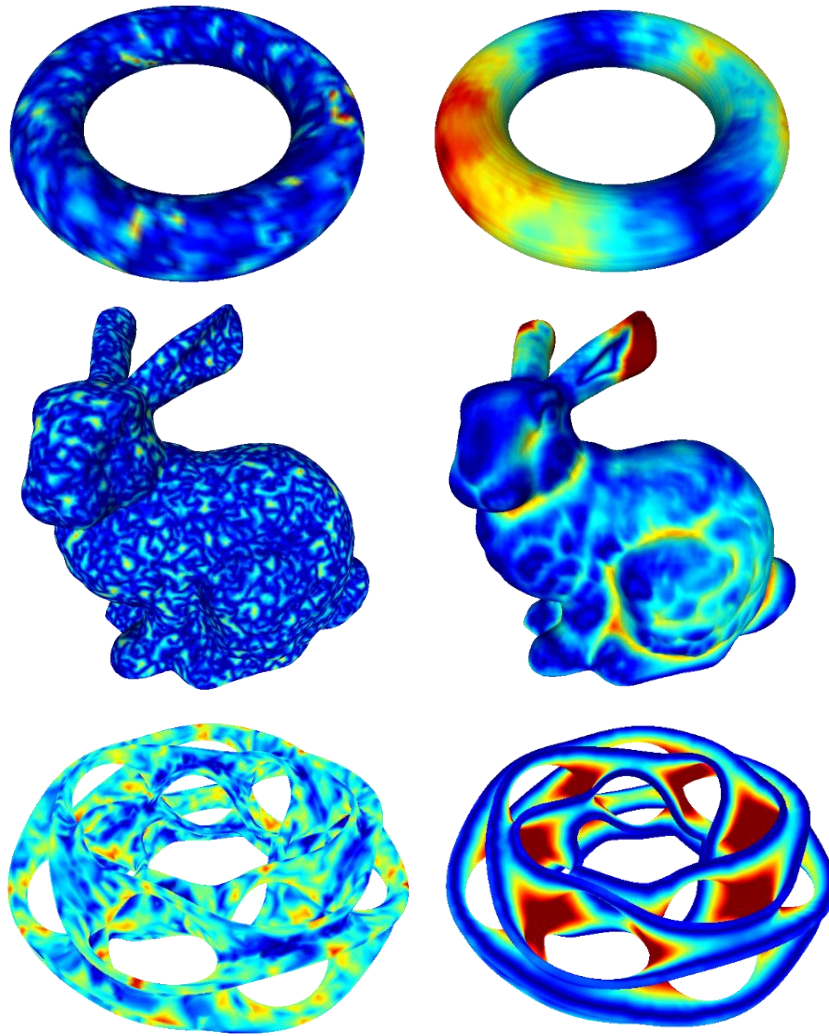


Fig. 13 Colour maps of the error distribution on the reconstructed models from Figure 12. Left: GNG. Right: Poisson mesh. It ranges from red to blue, using red for the areas with largest error and blue for the areas with lowest error.

Figure 14 shows the results of applying both reconstruction methods to noisy synthetic point clouds with Gaussian error equals to 2.5 millimetres. It can be seen how the Poisson algorithm (left) is not able to correctly reconstruct the ears of the bunny due to the amount of error in that area, creating a deformed shape. The rest of the bunny has been successfully reconstructed, but as we will see later, this reconstruction is not an accurate one since most generated surfaces are approximated, and therefore there exist error in terms of Eu-

clidean distance to the original model (error-free). The Growing Neural Gas (right) is able to generate a more accurate reconstruction of the original point cloud compared to Poisson.

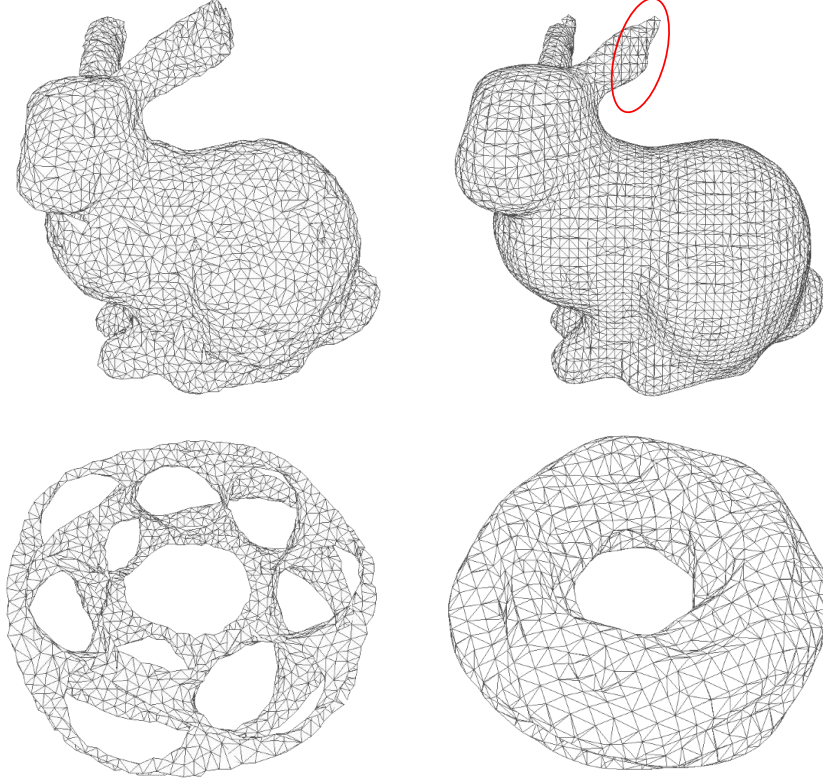


Fig. 14 Visual comparison of reconstructed models using GNG and Poisson methods. Left: GNG. Right: Poisson. Bunny (5,000 points), heptoroid (10,000 points). It is highlighted in red some areas of the bunny ears that are incorrectly reconstructed.

Finally, Table 2 shows all computed Hausdorff distances for all the tests performed using the synthetic models and different levels of noise and accuracy. It can be seen how the GNG algorithm outperforms the Poisson surface reconstruction algorithm for models with a certain amount of noise. In most cases, as we previously shown, the reconstructed models obtained using Poisson-based method are not able to correctly represent the noisy input data, and therefore in most of these cases the Hausdorff distance (error) is larger compared to the ones obtained using the GNG algorithm. In addition, we observed that not only the mean distance is larger, but also the final reconstructed model contains more outliers. This fact is extracted from the maximum error computed using the Hausdorff distance, which is shown in Table 2. Errors presented in Table 2 are in millimeters.

Method	Model	Error	Points	Max	Mean	RMS
Poisson	Torus	1	1500	5.8978	1.8325	1.2733
GNG	Torus	1	1500	4.4079	0.5151	0.9361
Poisson	Torus	2.5	1500	6.8978	2.8829	3.2788
GNG	Torus	2.5	1500	6.4079	0.9151	1.1871
Poisson	Bunny	1	7000	8.544	0.917	0.897
GNG	Bunny	1	7000	6.19	0.42	0.612
Poisson	Bunny	2.5	7000	12.544	1.117	1.697
GNG	Bunny	2.5	7000	3.19	0.67	0.814
Poisson	Heptoroid	1	10000	8.544	0.917	0.897
GNG	Heptoroid	1	10000	6.19	0.42	0.612
Poisson	Heptoroid	2.5	10000	12.544	1.117	1.697
GNG	Heptoroid	2.5	10000	3.19	0.67	0.814

Table 2 Hausdorff distances (millimeters) from the reconstructed models to the ground truth. The proposed method for 3D surface reconstruction using the GNG is compared against the Poisson surface reconstruction algorithm.

The proposed method allows to define the number of neurons (points) that will form the mesh, allowing to create meshes with different resolution. Since Poisson method does not allow to define this mesh attribute, to carry out these experiments and to establish a fair comparison between both methods we defined for the GNG method the same number of points that Poisson algorithm created for different levels of depth. This is the parameter that enable us to define the accuracy of the reconstructed model using Poisson method.

6.3 Real-world data experiments

In this section, we show the results of applying the proposed method to different real-world objects and scenes that have been acquired using low-cost 3D sensors. Figure 15 and 16 show the ability of the proposed method to create colour meshes of different types of models. It can be seen how most holes and gaps showed in Figure 7, generated by existing extensions of the GNG method for 3D surface reconstruction, were not generated using the proposed method.

The model of the person and the scene presented in Figure 16 and Figure 17 are not complete models, they are partial 3D views. The proposed method can work with partial views, creating coloured meshes of the input data. Additionally, as we can see on the bottom part of the Figure 16, the method was tested using a model of a foot obtained using a 3D foot digitizer.

Figure 17 shows the result of reconstructing two partial views of an office obtained using the Kinect sensor. We can appreciate how the proposed method is able to create a coloured mesh. However, the method is not able to manage the lack of information in some areas of the scene caused by occluded areas from the sensor’s point of view.

The proposed extension is also able to generate 3D meshes with different resolutions and therefore, detail level. Figure 18 shows the builder helmet and the dinosaur model reconstructed using different number of neurons, creating meshes with different level of detail.

However, the proposed method still produces some small gaps in the generated 3D reconstructions. Figure 19 shows small gaps and holes created in some of the experiments carried out. These are caused by the randomness of the network learning stage. Moreover, in some cases triangles removal is caused by the edge removal ageing scheme, which also is responsible of the good level of adaptation and interconnection between neurons. Despite

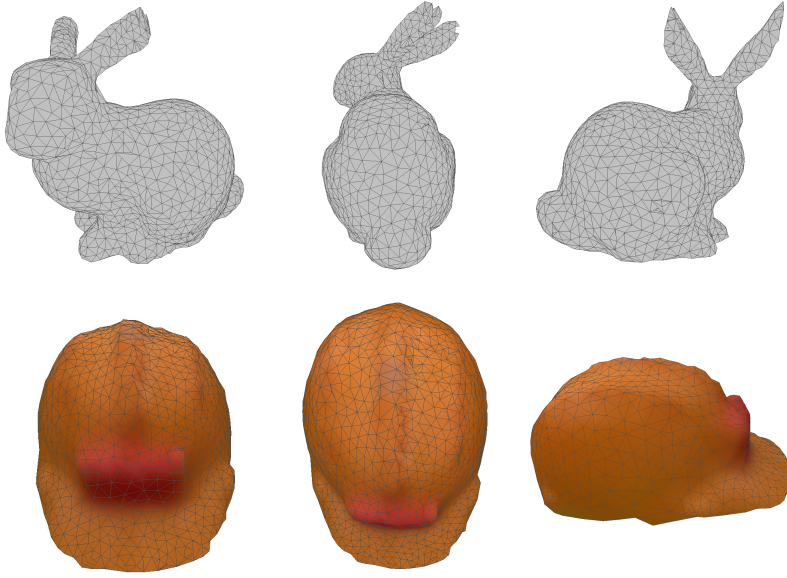


Fig. 15 Reconstructed models using our extended GNG method for face reconstruction and without applying post-processing steps. Top: Stanford bunny. Bottom: builder helmet.

this fact, the proposed method is valid for many computer vision applications. 3D triangular faces are used in many 3D descriptors which are often used in object and scene recognition applications.

7 Conclusions and future work

In this paper we have presented a novel unsupervised learning method to create complete meshes from unorganized raw noisy 3D data. Previous knowledge about the sensor is not necessary. It has been demonstrated how Growing Self-Organizing Maps (GSOM) are capable to represent noisy 3D data distributions. Neurons' codevector has also been modified adding input data colour and surface normal information to their weights. Due to this modification, the algorithm is able to adapt its structure to the input space topology during the learning step and also to learn and store colour from the observation. This eliminates the necessity to add post-processing steps to add colour information to the final representation. In addition, including surface normal information, the reduced representation can be used to directly compute 3D descriptors and 3D keypoints over the GNG structure, improving computing performance.

Furthermore, the GNG algorithm has been also extended considering the creation of triangular faces during the learning stage. In contrast with existing methods, our extension allows to create complete triangular meshes with colour information during the learning stage, not requiring any post-processing steps to close gaps and holes. The method was validated with several models ranging from scanned objects to body parts like a foot. We also performed an extensive quantitative and qualitative study comparing the proposed method



Fig. 16 Reconstruction of 3D human models with the proposed method. Top: 3D model of a person. Bottom: digitized human foot.

against the state-of-the-art Poisson surface reconstruction method. Results showed that the proposed method is able to deal with noisy information while the Poisson method fails reconstructing some parts of the tested models. Finally, real-world scenes have been tested, showing positive results in the reconstruction of indoor environments.

Future work includes the adaptation and integration of the proposed method for online 3D surface reconstruction in mobile robotics tasks.

Acknowledgements This work was partially funded by the Spanish Government DPI2013-40534-R grant.

References

1. Hoppe H, DeRose T, Duchamp T, McDonald J, Stuetzle W (1992) Surface reconstruction from unorganized points. SIGGRAPH Comput Graph 26(2):71–78, ISSN 0097-8930, doi:10.1145/142920.134011



Fig. 17 The proposed method for 3D reconstruction was also applied to partial 3D views of scenes. Left: Noisy point clouds captured using the Kinect sensor. Right: 3D reconstruction using the proposed method.

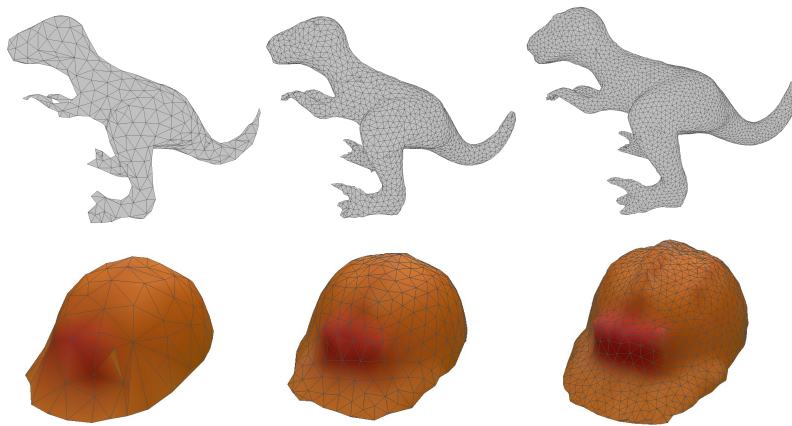


Fig. 18 Different 3D reconstructions of a the builder helmet model using various network sizes. Left: 3D reconstruction using 250 neurons and 200 input patterns. Middle: 3D reconstruction using 1000 neurons and 500 input patterns. Right: 3D reconstruction using 2500 neurons and 1000 input patterns.

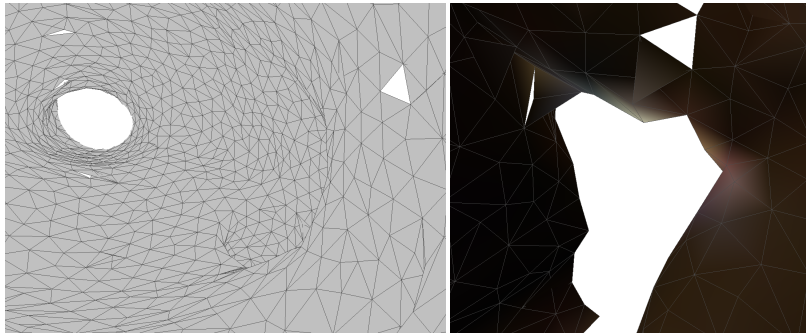


Fig. 19 Small gaps produced by our extended GNG method for 3D surface reconstruction.

2. Amenta N, Choi S, Kolluri RK (2001) The power crust. In Proceedings of the sixth ACM symposium on Solid modeling and applications, SMA '01, 249–266, ACM, New York, NY, USA, ISBN 1-58113-366-9, doi:10.1145/376957.376986
3. Berger M, Tagliasacchi A, Seversky LM, Alliez P, Levine JA, Sharf A, Silva C (2014) State of the Art in Surface Reconstruction from Point Clouds. Eurographics STAR (Proc of EG'14)
4. Yu Y (1999) Surface Reconstruction from Unorganized Points Using Self-Organizing Neural Networks Yizhou Yu. In In IEEE Visualization 99, Conference Proceedings, 61–64
5. Junior A, Neto ADD, de Melo J (2004) Surface reconstruction using neural networks and adaptive geometry meshes. In Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on, vol. 1, –807, ISSN 1098-7576, doi: 10.1109/IJCNN.2004.1380023
6. Fritzke B (1993) Growing Cell Structures - A Self-organizing Network for Unsupervised and Supervised Learning. Neural Networks 7:1441–1460
7. Ivriissimtzis I, Jeong WK, Seidel HP (2003) Using growing cell structures for surface reconstruction. In Shape Modeling International, 78 – 86
8. Martinetz T, Schulten K (1994) Topology Representing Networks. Neural Networks 7(3)
9. Barhak J (2002) Freeform objects with arbitrary topology from multirange images. Ph.D. thesis, Israel Institute of Technology, Haifa, Israel
10. Fritzke B (1995) A Growing Neural Gas Network Learns Topologies, vol. 7, 625–632. MIT Press
11. Cretu AM, Petriu EM, Payeur P (2008) Evaluation of growing neural gas networks for selective 3D scanning. In Proc. Int. Workshop Robotic and Sensors Environments ROSE 2008, 108–113
12. Holdstein Y, Fischer A (2008) Three-dimensional surface reconstruction using meshing growing neural gas (MGNG). Vis Comput 24:295–302
13. Do Rego RLME, Araujo AFR, De Lima Neto FB (2010) Growing Self-reconstruction Maps. Trans Neur Netw 21(2):211–223, ISSN 1045-9227, doi: 10.1109/TNN.2009.2035312
14. Orts-Escolano S, Garcia-Rodriguez J, Morell V, Cazorla M, Garcia-Chamizo JM (2014) 3D colour object reconstruction based on Growing Neural Gas. In 2014 International

- Joint Conference on Neural Networks, IJCNN 2014, Beijing, China, July 6-11, 2014, 1474–1481, doi:10.1109/IJCNN.2014.6889546
15. Kazhdan M, Bolitho M, Hoppe H (2006) Poisson surface reconstruction. In Proceedings of the fourth Eurographics symposium on Geometry processing, SGP '06, 61–70, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, ISBN 3-905673-36-3
 16. Rusu RB, Blodow N, Beetz M (2009) Fast Point Feature Histograms (FPFH) for 3D registration. In Robotics and Automation, 2009. ICRA '09. IEEE International Conference on, 3212–3217, ISSN 1050-4729, doi:10.1109/ROBOT.2009.5152473
 17. Tombari F, Salti (2011) A combined texture-shape descriptor for enhanced 3D feature matching. In Image Processing (ICIP), 2011 18th IEEE International Conference on, 809–812, ISSN 1522-4880, doi:10.1109/ICIP.2011.6116679
 18. Mian AS, Bennamoun M, Owens RA (2006) A Novel Representation and Feature Matching Algorithm for Automatic Pairwise Registration of Range Images. *Int J Comput Vision* 66(1):19–40, ISSN 0920-5691, doi:10.1007/s11263-005-3221-0
 19. Orts-Escolano S, Morell V, Garcia-Rodriguez J, Cazorla M (2013) Point cloud data filtering and downsampling using growing neural gas. In The 2013 International Joint Conference on Neural Networks, IJCNN 2013, Dallas, TX, USA, August 4-9, 2013, 1–8, doi:10.1109/IJCNN.2013.6706719
 20. Jolliffe I (1986) *Principal Component Analysis*. Springer Verlag
 21. Mole VLD, Araújo AFR (2010) Growing Self-organizing Surface Map: Learning a Surface Topology from a Point Cloud. *Neural Comput* 22(3):689–729, ISSN 0899-7667, doi:10.1162/neco.2009.08-08-842
 22. Cignoni P, Rocchini C, Scopigno R (1996) Metro: Measuring Error on Simplified Surfaces. Tech. rep., Paris, France, France
 23. Rusu RB, Cousins S (2011) 3D is here: Point Cloud Library (PCL). In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Shanghai, China
 24. (2013), The Stanford 3D Scanning Repository
 25. Tombari F, Salti S, Di Stefano L (2010) Unique Signatures of Histograms for Local Surface Description. In Proceedings of the 11th European Conference on Computer Vision Conference on Computer Vision: Part III, ECCV'10, 356–369, Springer-Verlag, Berlin, Heidelberg, ISBN 3-642-15557-X, 978-3-642-15557-4
 26. Mian AS, Bennamoun M, Owens R (2006) Three-Dimensional Model-Based Object Recognition and Segmentation in Cluttered Scenes. *IEEE Trans Pattern Anal Mach Intell* 28(10):1584–1601, ISSN 0162-8828, doi:10.1109/TPAMI.2006.213
 27. Gray A (1996) *Modern Differential Geometry of Curves and Surfaces with Mathematica*. CRC Press, Inc., Boca Raton, FL, USA, 1st edn., ISBN 0849371643