ARTÍCULO DE INVESTIGACIÓN ORIGINAL

REVISTA MEXICANA DE INGENIERÍA BIOMÉDICA

Vol. XXI, Núm. 3
Julio-Septiembre 2000
pp 73 - 81

A combined algorithm for image segmentation using neural networks and *3D* surface reconstruction using dynamic meshes

Reyes Aldasoro C C,*
Algorri ME*

* Departamento de Sistemas Digitales Instituto Tecnológico Autónomo de México. Río Hondo No. 1, Tizapán San Angel, 01000 México D.F
creyes@itam.mx, algorri@itam.mx

Artículo recibido 10/julio/ 2000
Artículo aceptado 28/agosto/ 2000

ABSTRACT

Reconstructing the surface from a set of unstructured points to build a 3D model is a problem that arises in many scientific and industrial fields as new 3D scanning technology is able to produce large databases of full 3D information. 3D surface reconstruction is also important after segmenting sets of 2D images to visualise the 3D surface represented by the segmentation. In this paper we propose an algorithmic methodology that obtains a series of segmentations of human head tomographies, produces a set of unstructured points in the 3D space, and then automatically produces a surface from the set of unstructured 3D points about which we have no topological knowledge. The methodology can be divided in two stages. First, tomographic images are segmented with a Neural Network algorithm based on Kohonen's[1,2] Self-Organising Maps (SOM). The output neurones that have adapted to the image, are a series of 3D points that will be fed to the second stage. Next, our method uses a spatial decomposition and surface tracking algorithm to produce a rough approximation **S'** of the unknown manifold **S**. The produced surface **S'** serves as initialisation for a dynamic mesh model that yields the details of S to improve the quality of the reconstruction.

Key words:
Neural networks, Image segmentation, Self-Organising maps, 3D Reconstruction, Dynamic meshes.

RESUMEN

La reconstrucción de superficies a partir de conjuntos de puntos no estructurados con el fin de crear modelos tridimensionales es un problema frecuente en muchas disciplinas científicas e industriales. La presencia creciente de escaners 3D que son capaces de producir grandes bases de datos de información parcial de objetos requiere de algoritmos robustos para completar la información faltante sobre los objetos y crear modelos completos de la información 3D. La reconstrucción de superficies es también importante después de la segmentación de imágenes, de modo que sea posible visualizar la superficie 3D representada por la segmentación. En este artículo proponemos una metodología algorítmica que obtiene segmentaciones de tomografías de la cabeza humana, produce conjuntos no estructurados de puntos a partir de las segmentaciones, y

después produce automáticamente una superficie a partir del conjunto de puntos sin importar la topología de la superficie que se esté reconstruyendo. La metodología se puede dividir en dos etapas. Primero, las imágenes tomográficas son segmentadas usando un algoritmo de redes neuronales basado en los mapas autoorganizables de Kohonen.[1,2] El resultado de este algoritmo son neuronas que se han adaptado a la imagen y que constituyen el conjunto de puntos 3D que se utilizará en la segunda etapa. La segunda etapa utiliza un algoritmo de descomposición espacial y seguimiento de superficies para producir una aproximación burda **S´** de la superficie desconocida **S**. La superficie **S´** sirve de inicialización para un modelo de malla dinámica que produce los detalles de S mejorando notablemente la calidad de la reconstrucción.

Palabras clave:
Redes neuronales, Segmentación de imágenes, Mapas autoorganizables, Reconstrucción 3D, Mallas dinámicas.

INTRODUCTION

We propose a solution to the following problem:

Given a set **Q** of images of human head tomographies, segment a certain structure of the images, and from the output obtain a series **Q'** of segmented images that can be interpreted as a set **P** of unstructured *3D* points that samples an unknown arbitrary surface **S**, reconstruct a surface **S'** that correctly approximates **S**.

The first stage of the algorithm consists of structure segmentation, an important and complex problem in image processing.[8,9,10] Segmentation, as defined by Kapur,[8] is "a labelling problem in which the goal is to assign to each vowel in an input gray-level image, a unique label that represents an anatomical structure." Therefore, the ultimate objective would be to properly identify structures such as a tumour, the brain tissue or the skull.

The segmentation of an image can be carried out by different techniques that are based mostly on the discontinuity and similarity of its grey levels. Gonzalez and Woods[11] propose several edge detection and segmentation techniques and Felzenszwalb and Huttenlocher[12] propose yet different methods. In this paper, a neural network approach based on Kohonen's Self-Organising Maps is used to segment medical images.

The Self-Organising Maps[1,2] consist of a series of nodes or "neurones" that will act upon a series of inputs. Each neurone is densely interconnected, receives a primary input and a great number of lateral interconnections from the outputs of other neurones. The lateral coupling of the neurones is thought of as a function of the distance in two ways: excitatory and inhibitory. The excitatory is a short-range area up to a certain radius, and the inhibitory area surrounds the excitatory area up to a bigger radius. Outside the inhibitory range, a weaker, and much bigger excitatory zone exists. A cluster or bubble around one particular node of the network is formed because of the lateral coupling around a given cell. The primary input determines a "winner" node, which will have a certain cluster, and then, following the input, the winner node with its surrounding cluster or neighbourhood will adapt to the input. The process continues for a number of interactions until a certain degree of adaptation is reached. When the input is an image, certain features are extracted by the final adaptation of the neurones. The neurones will then form a set **P** of unstructured *3D* points that will be the input for the reconstruction stage.

In the last years, a great deal of research has been dedicated to solving the general case of *3D* reconstruction where the only information available is the position of the points in **P**. Two categories of algorithm are efficient to deal with such reconstruction problems: 1) methods that reconstruct the surface by exploring the set **P** and imposing a structure on it (local techniques) such as modified marching cubes,[15] particle systems,[16] and 2) methods using deformable or physically based models to approximate the surface of the set **P** such as geometrically deformable models,[17] and adaptive meshes.[18]

Local techniques can reconstruct arbitrarily complex shapes to a good degree of accuracy. A drawback of these techniques is that they require additional information on the set of *3D* points other than the position (normal, neighbourhood information etc.), for which some sort of pre-processing is usually required.

Deformable models such as the *3D* adaptive meshes in[18,20] require no additional information on **P** to reconstruct

the underlying surface, but the type of surfaces they can reconstruct is usually limited to being homomorphic to the initial mesh: they start with a generic mesh (usually a sphere or cylinder) which is then deformed to approximate the shape of **S**. In a post-processing stage some deformable models incorporate holes and borders, or change the topology of the initial mesh,[22,21] but this post-processing requires the estimation of additional information on **P** (measures of curvature or distance maps from **P** to the deformable model). In addition, global models might develop auto-intersections when reconstructing surfaces that fold over themselves or have a complex topology, and they might fail to reconstruct the correct underlying surface when such a surface is very far away from the initial model.

The reconstruction algorithm in this paper combines the two categories of surface reconstruction methods mentioned before. We propose a simplified and robust reconstruction method that eliminates the hypotheses of additional information on P (in local techniques) and of homomorphism to the initial global mesh (in deformable models). We use a local technique to recuperate the initial topology of **P** and build a good initial model for a deformable adaptive mesh. The resulting algorithm is able to reconstruct arbitrary surfaces in a non-supervised fashion without the need to estimate additional information about **P**.

To solve the reconstruction problem we place ourselves in the situation where the set **P** satisfies the following density constraint: for any point **s** of **S**, its nearest point **x** Î **P** according to the point to point Euclidean distance in $\mathbf{R}^3$, is also its nearest point **x** Î **P** according to the geodesic distance r on the surface **S**. We will also assume that the maximal distance between any two nearest points of **P** is smaller than any hole, discontinuity or bending of **S**.

The remaining of the paper is organised as follows: Section 2 describes the implementation of the neural network interface for segmentation; Section 3 presents some segmentation results; Section 4 introduces the initial reconstruction model; Section 5 presents the Dynamic Model and Section 6 an overview of dynamic equations. Section 7 shows the reconstruction results and finally, conclusions are summarised in Section 8.

METHODOLOGY

Implementation of segmentation

The Self-Organising algorithm proposed by Kohonen follows two basic equations: matching and finding the winner node determined by the minimum Euclidean distance to the input[1] and the update of the position of neurones inside the cluster.[2]

$$\|x(t) - m_c(t)\| = \min_i \|x(t) - m_i(t)\| \quad (1)$$

$$\begin{aligned} m_i(t+1) &= m_i(t) + \alpha(t)[x(t) - m_i(t)] \quad i \in N_c \\ m_i(t+1) &= m_i(t) \quad i \notin N_c \end{aligned} \quad (2)$$

Where, for time *t*:

	x	is the input
	m_i	is any node,
	m_c	is the winner,
	a	is the gain sequence, and
	N_c	is the neighbourhood of the winner.

It should be noted that only the excitatory region would be considered for the update with satisfactory results. The decrease rate of the neighbourhood and the initial value of the parameters are studied in.[14]

The updating process is a variation of the location of the node, proportional to the Euclidean distance from the node to the input multiplied by the gain sequence if the node lies inside of the neighbourhood. If it is not inside the neighbourhood, its position remains unaltered.

The definition of the neighbourhood presents two different cases, one if the network of nodes accepts that the neighbourhood is limited by the edges of the network itself, and the other in case that the neighbourhood is not limited by the edges of the network. If the network consists of n*1 neurones, the first case results in a linear network and the second in an annular network. Figure 1 shows both cases for 10 neurones and a neighbourhood of size 2 around a winner node and the output to a given input. The neighbourhood for a linear network consists only of four neurones on one side of the network, contrary to the

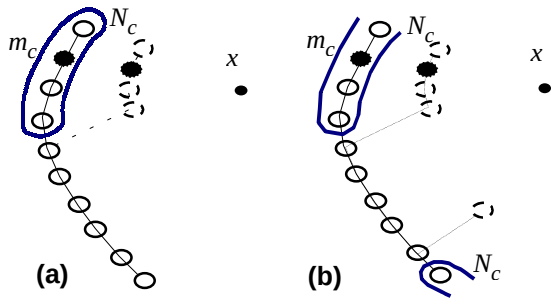

Figure 1. Differences in update regions: (a) Linear Network, (b) Annular Network.

annular network that includes a fifth neurone on the other side. If the network is a two-dimensional array of neurones, the network itself generally limits the neighbourhood.

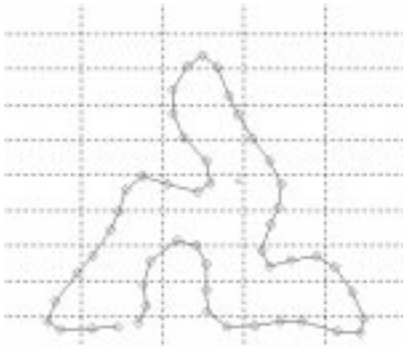

Figure 2. Output for triangular input and an Annular Network.

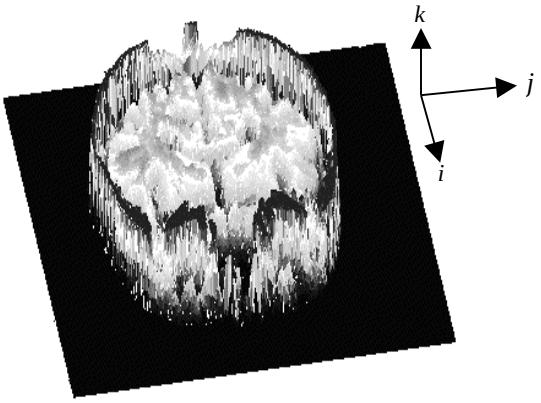

Figure 3. Matrix *3D* display of a brain slice image.

Figure 2 presents the results for neurones organised as an annular network, given as input a triangular region. The network consists of 50 neurones and the results were obtained after 10,000 interactions. It should be noted that no initial conditions for the network are required. In fact, the neurones start at random positions.

The input signal can be defined either as a test set (a geometrical region like a square, triangle, and cube), or it can be read from an image. The geometrical regions are widely used to test algorithms.

When an image is used as input, it is first transformed into an *i*j*l* matrix, where *i*j* is the dimension in pixels of the image and *l*, the number of layers of the image. For colour images *l* = 3 and for a grey scale images *l* = 1. For each (*i, j, l*) position there exists a $k_{i,j,l}$ value in the range 0-255 depending on the intensity of the colour or grey level. Let R_{ijl} be the region of existence of *k*.

Figure 3 shows a grey level Magnetic Resonance (MR) image of a transaxial slice of a human head transformed into a matrix. The *i* and *j* axis follow the original image, and the value of the *k*-axis is proportional to the intensity, darkness/brightness, of the pixel in the *x, y* position of the original image. The bones from the skull appear with a higher *k* value while the brain and other elements have different lower values.

This format of the image allows several transformations. Since the *k* value represents the grey level, selecting a certain range of *k* can segment the image. With two different thresholds one lower, *LT*, and one higher, *HT*, the values of *k* can be transformed according to:

$$\begin{aligned} &\forall i, j, l \qquad 0 \le LT < HT \le 255 \\ &0 < k_{i,j,l} \le LT \quad \rightarrow \quad k_{i,j,l} = 0 \\ &LT < k_{i,j,l} \le HT \quad \rightarrow \quad k_{i,j,l} = k_{i,j,l} \\ &HT < k_{i,j,l} < 255 \quad \rightarrow \quad k_{i,j,l} = 0 \end{aligned} \tag{3}$$

As an example, Figure 4 presents a segmentation with *LT* = 200 and *HT* = 255, of the higher *k* values which correspond to bone of the skull.

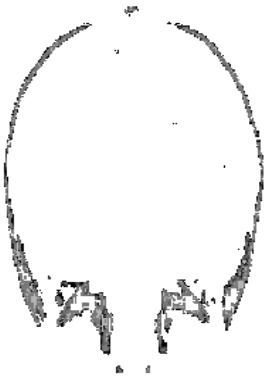

Figure 4. Segmentation of human head MR.

Once the image has been transformed into a matrix, and segmented if needed by grey level selection, the algorithm for the SOM will have as input region the non-zero values of the matrix.

$$\exists\ k_{i,j,l} \in R_{ijl} : [k_{i,j,l} \neq 0 \Leftrightarrow \exists\ x_{i,j,l}] \tag{4}$$

The *x* value is used for the Self-Organising algorithm that runs until a certain parameter of is reached. Then, the final network is compared with the original image to identify the segmented regions. The process is shown graphically in Figure 5.

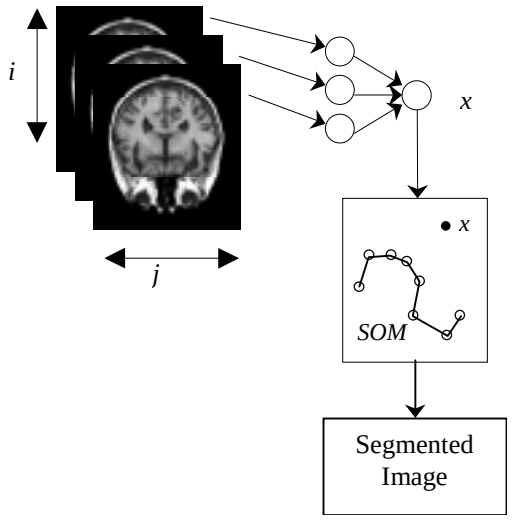

Figure 5. Segmentation algorithm.

RESULTS

Segmentation results

The Self-Organising Maps were run with different images as input signals. Figure 6 shows the result of an 80-neurone network after 10,000 interactions run over the MR slice previously segmented with grey levels. In this case a network with ring topology, instead of a grid or linear network can provide the best results in order to extract the contour and "thin" the bone as well as to give continuity to the shape of the skull. In Figure 6 (a) the black diamonds show the position of the neurones connected with virtual lines. In Figure 6 (b) only the neurones are presented with circles, note how the region of the nose is completed with a virtual line between neurones and also the occipital region. A slight discontinuity can be noted between the first and the last neurones in the parietal region on the left side of the network.

The previous procedure is repeated for the collection of images representing the contiguous slices in a MR study of a human head. In the end, an *n*-neurone network will represent points over the surface of the human skull. The position of the neurones is interpreted as a *3D* position that will be used to reconstruct the surface of the skull. Figure 7 shows the collection of points from 54 MR slices, each with 58 output neurones.

Construction of Initial reconstruction model

The surface reconstruction algorithm starts with the construction of an initial surface model. This stage consists of imposing a first structure on the set **P** in order to build a rough model **S'** of the same topology of **S**.

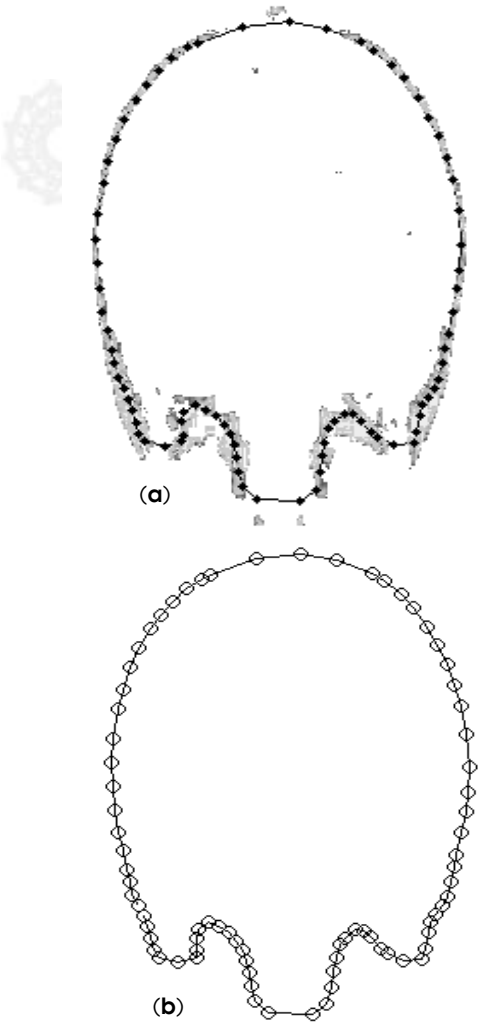

Figure 6. 80-neurone network adapted over figure 4. (a) Neurone positions (black) and skull (grey) (b) Neurones only, arrow points discontinuity between first and last neurones.

The rough model **S'** will then be used to initialise an adaptive mesh that will improve the quality of the reconstruction. The initialisation process takes as input the set **P** of points $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\} \hat{I} \mathbf{R}^3$, and the maximum distance between two adjacent neurones r. It performs a partitioning of the space where **P** is contained into cubes with edges of size *e* equal to r. If models of lower detail are sufficient a dilation factor can also be specified to increase the size *e* of the spatial cubes. Also, the partitioning of the space need not be regular in the three dimensions. The algorithm flags the cubes that are occupied by one or more points of **P**, and for each cube we register the information about what points are located inside it. The spatial partitioning

scheme automatically imposes an organisation on the data.

We build a cuberille from the set of occupied cubes, that is, a surface composed of all the exterior faces that are not shared by any two occupied cubes. The cuberille defines a rough surface **S'** approximating **S**. To build the cuberille we use the surface extraction algorithm by Gordon and Udupa[23] which tracks a closed surface from a set of connected cubes. Figure 8 a) shows the cuberille that was built upon the set **P** of figure 7 and that defines the surface **S'**.

We systematically triangulate the set of cuberille faces by splitting each square by one of its diagonals. We also integrate the information about the created triangles and their adjacencies into a triangulated mesh structure. The stepped surface **S'** of the triangulated cuberille can be directly used as an initialisation model for the adaptive mesh algorithm. However, to speed up the convergence of the adaptive mesh, we smooth the surface **S'** using a simple low pass filter. The low pass filter assigns to each triangle vertex a new position that is a weighted average of its old position and the position of its neighbours. Figure 8 b) shows the systematic triangulation constructed on the rough surface **S'**. Figure 8 c) shows the smoothed surface **S'** obtained after applying the low pass filter to the initial triangulation.

The initial model **S'** will incorporate the correct topology of **S** if the following conditions are verified: i) the data set **P** satisfies the density constraint, that is, r (the maximum distance between any two nearest neurones) is smaller than the characteristics of the surface **S**, and ii) the size *e* of the spatial cubes is equal to r. With these two conditions it can be verified that two adjacent sample points on **S** belong either to the same cube, or to two adjacent cubes (according to the 26-connexity).

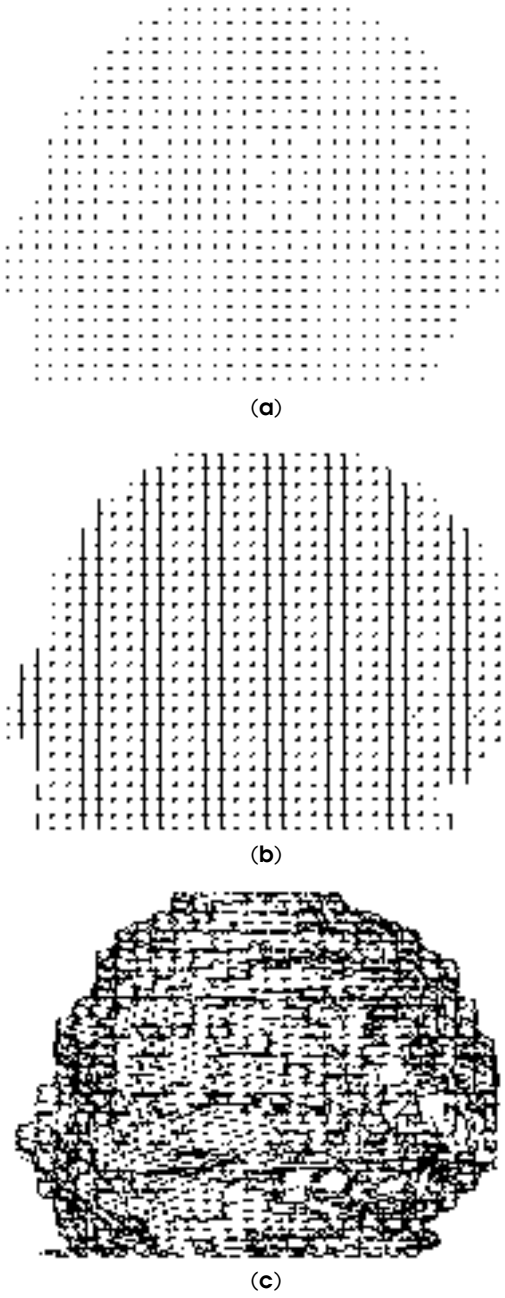

Figure 8. a) The rough surface **S'** (cuberille) built open the set **P** in Figure 7, b) The systematic triangulation constructed on **S'**, c) The smoothed surface **S'** after applying the low pass filter to the triangulation.

Figure 7. Collection of *3D* points of 54 MR slices, each segmented with 58 neurones: the set **P.**

Dynamic model

This stage of the reconstruction algorithm consists in deforming the initial model **S'** to better conform to the surface **S** and to recover the fine details not incorporated in the initialisation procedure. To deform

S' we will consider it as an adaptive mesh. We use the mass-spring adaptive mesh model described in.[18,20]

An adaptive mesh is a polyhedral structure and in our case a triangulated mesh. Adaptive meshes are flexible models because of their capacity to deform under the influence of attracting forces coming from the *3D* points in **P**. However, adaptive meshes alone are not effective to reconstruct complex shapes because of the difficulty of establishing a correct correspondence between the data points in **P** and the adaptive mesh. We overcome this limitation by initialising our adaptive mesh with the mesh **S'** that already has the topology of **S**. Each triangle vertex in **S'** is a nodal mass, and each triangle side an adjustable spring. Each nodal mass is attached to its closest point in **P** by an imaginary spring. From the space partitioning scheme we know the data points inside every cube and the triangles that were originated from it, thus we can establish a fast, local correspondence between **S'** and **P** as follows: for every vertex on **S'**, we search inside the cube from which it originated, and inside its 26-neighboring cubes, the point in **P** that is closest to the considered vertex. This correspondence is recalculated at every interaction by searching inside the cube containing the point of **P** that was last assigned to a given vertex and inside its 26-neighboring cubes. By controlling the interaction time step *dt* so that the motion of a vertex remains smaller that *e*, we keep a correct correspondence as the vertices of **S'** move in time through the spatial cubes to adapt to **S**.

Dynamic equations

In an adaptive mesh, the nodal masses are subject to structural forces coming from the springs that interconnect them, and to external forces coming from the data points in **P**. Each nodal mass has a dynamic equation of the form:

$$\mu_i \frac{d^2 x_i}{dt^2} + \gamma_i \frac{dx_i}{dt} + \sum_{j=1}^{n} k_j (x_i - x_j) + k_d (x_i - x_d) = 0, \qquad (5)$$

where x_i is the *3D* position $(x_i, y_i, z_i)^t$ of node *i* at time t, and μ_i and γ_i are the mass and damping values associated to node *i*. k_j is the stiffness coefficient of the spring connecting node *i* to neighbouring node *j*, and k_d is the stiffness coefficient of the spring connecting node *i* to its closest point in **P** x_d. We use springs of natural length equal to zero.

The dynamic equations of the nodal masses can be rewritten as:

$$\mu_i \frac{d^2 x_i}{dt^2} + \gamma_i \frac{dx_i}{dt} + \left(k_d + \sum_{j=1}^{n} k_j\right) x_i - \sum_{j=1}^{n} k_j x_j = x_d k_d, \qquad (6)$$

and in matrix form:

$$\mathbf{M}\ddot{\boldsymbol{x}} + \mathbf{G}\dot{\boldsymbol{x}} + \mathbf{K}\boldsymbol{x} = \mathbf{F}_{ext} \qquad (7)$$

where **M** is the mass matrix, **G** the damping matrix, **K** the stiffness matrix, and $\mathbf{F}_{ext}$ the external force vector. **M** and **G** are diagonal matrices; **K** contains off-diagonal elements and therefore the system is said to have stiffness coupling. An adaptive mesh can also be seen as a coupled oscillator with *N*-degrees of freedom (*N* = number of nodal masses in the system). We use a direct step-by-step explicit Euler integration scheme to solve the system of coupled equations.

An adaptive mesh will exhibit different types of behaviour (oscillatory, exponential) depending on the values of its dynamic parameters **M**, **G**, and **K**. To be able to use adaptive meshes efficiently for surface reconstruction, we need to perform an analysis of their dynamic characteristics in order to ensure a stable, non-oscillatory behaviour of the nodal masses and optimise the convergence time. For a detailed analysis on the determination of the dynamic parameters see.[24]

Reconstruction results

Figure 9 shows the results of the final reconstruction. We modelled the adaptive mesh using a critically damped coefficient value. Convergence of the mesh (i.e. the speed and the acceleration of the nodes falling below a given threshold) take in practice from 20 to 30 interactions.

Figure 10 show the final reconstruction as a shaded image. The presence of holes is noticeable in zones of the skull where neurones are very far away from each other.

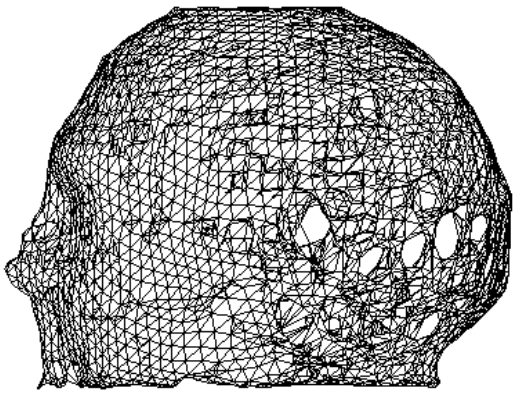

Figure 9. Final reconstruction.

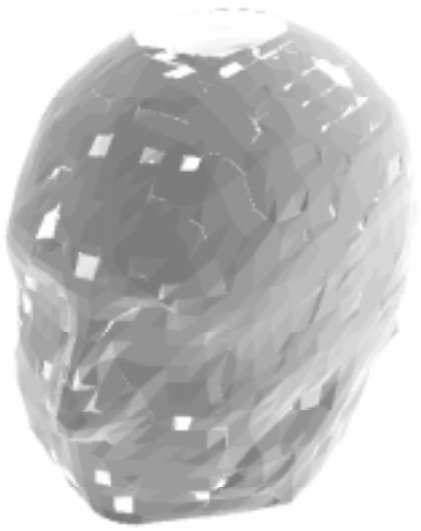

Figure 10. Final triangulation shown as shaded image.

CONCLUSIONS

A Self-Organising Map network was programmed to receive images, as input signal regions. For this work, medical images were used and different segmentations were obtained with annular and grid networks without the requirement of initial conditions. These types of results allow us to extract important features like the skull from a human head MR study and to represent its position through *i*, *j* coordinate points. The limitations of the algorithm depend now on the computational complexity of the images and the amount of neurones for the networks. Also, different kinds of networks can be programmed to improve segmentation results (i.e. an hexagonal grid).

Our current research focuses in applying the neural network algorithm to different types of images and in trying other segmentation problems with more complex shapes such as the ventricles or white matter in a human brain MR.

The segmentation results were input to an algorithm for reconstructing in two stages a surface **S** from a set **P** of dense but unstructured *3D* points. First, we recover the topology of the underlying object using a technique of exhaustive spatial decomposition and a surface tracking algorithm. This simple local technique allows us to build rough triangulations of arbitrary surfaces. In the second part of the algorithm we use the rough triangulation to initialise an adaptive mesh model. We dynamically deform the adaptive mesh to adapt it to the surface of the set **P**. By using the spatial partitioning information about the position of the *3D* points relative to the deformable mesh, we can establish a fast dynamic correspondence. Because the adaptive mesh is initialised very near the final surface, and because we optimise the dynamic parameters of the system, only a small number of interactions are needed for convergence.

By using only the position information of the dense set of *3D* points in **P** we are able to obtain accurate reconstruction that could not have been obtained by the use of local methods or deformable models alone, since both methods need to estimate additional information on **P** to perform successfully.

The two step reconstruction algorithm is fully automatic does not require additional pre-processing or post-processing of the data, and is capable of reconstructing arbitrary forms. Because of its flexibility to reconstruct different topologies, our method is a good tool for unsupervised reconstruction after segmentation of a set of images or for visualising the surface of acquired *3D* points.

ACKNOWLEDGEMENTS

Keith A. Johnson and J. Alex Becker from Brigham and Women's Hospital, Harvard Medical School, provided the Magnetic Resonance images, through *The Whole Brain Atlas*. The authors are grateful to them.

This work was supported by CONACYT.

REFERENCES

1. Kohonen T. *Self-Organisation and Associative Memory*, Heidelberg: Springer-Verlag 1988.
2. Kohonen T. *Self-Organising Maps: Optimisation and Approaches*, Finland: ICANN, Espoo 1991.
3. Hall EL. *Computer Image Processing and Recognition*, New York: Academic Press 1979.
4. Scudder HJ. *Introduction to Computed Aided Tomography*, Proc. IEEE 1978; 66: 628-637.
5. Mueller RK, Kaveh M and Wade G. *Reconstructive Tomography and Applications to Ultrasonic*, Proc IEEE 1979; 67: 567-587.
6. Hinshaw W and Lent AH. *An Introduction to NMR Imaging: From the Bloch Equation to the Imaging Equation*, Proc IEEE *1983;* 71: 338-350.
7. Warfield SK, Jolesz FA and Kikinis R. *Real-time Image Segmentation for Image-Guided Surgery*, Surgical Planning Laboratory, Department of Radiology, Harvard Medical School and Brigham and Women's Hospital: SPL Technical Report 1998; 106.
8. Kapur T. *Model based three dimensional Medical Image Segmentation*, Artificial Intelligence Laboratory, Massachusetts Institute of Technology: Ph.D. Thesis 1999.
9. Saiviroonporn P, Robatino A, Zahajszky J, Kikinis R and Jolesz FA. *Real-time Interactive 3D-Segmentation*. Acad. Radiol 1998; 5: 49-56
10. Rueckert D, Burger P, Forbat SM, Mohiaddin RD and Yang GZ. *Automatic Tracking of the Aorta in Cardiovascular*

MR images Using Deformable Models, IEEE Transactions on Medical Imaging 1997; 16(5): 581-590.
11. González RC and Woods RE. *Digital Image Processing*, Addison Wesley, Reading 1992.
12. Felzenszwalb PF and Huttenlocher DP. *Image Segmentation Using Local Variation*, Santa Barbara, California: CVPR98 1998.
13. Ramón-Cajal S. *Histology*, Baltimore: William Wood & Co 1933).
14. Reyes-Aldasoro CC. *A Non-linear Decrease Rate to Optimise the Convergence of the Kohonen Neural Network Self-Organising Algorithm*, ROCC99 Acapulco, Mexico, 1998: 11-16.
15. Lorensen WE, Cline HE. *Marching Cubes: A High Resolution 3D Surface Construction Algorithm*, *SIGGRAPH* '87 1987; 21(4): 163-169.
16. Szeliski R, Tonnesen D, Terzopoulos D. M*odelling Surfaces of Arbitrary Topology with dynamic particles*, Proc CVPR '93 *1993:* 82-87.
17. Miller JV, Breen DE, Lorensen WE, O'Bara RM, Wozny MJ. *Geometrically Deformed Models: A Method for Extract. Closed Geom. Models from Vol. Data*, SIGGRAPH '91 1991; 25(4): 217-225.
18. Vasilescu M, Terzopoulos D. *Adaptive Meshes and Shells: Irregular Triangulation, Discontinuities, and Hierarchical Subdivision*, Proc CVPR'92 1992: 829-832.
19. Hoppe H. *Surface Reconstruction from Unorganised Points*, Univ. of Washington, Seattle WA: Ph.D. Dissertation 1994.
20. Huang W-C, Goldgof DB. *Sampling and Surface Reconstruction with Adaptive-Size Meshes*, Proc SPIE Appl of Art. Intell. X 1992; 1798: 760-770.
21. Delinguette H. *Modelisation, Déformation et Reconnaissance d'Objets Tridimensionnels à l'aide de Maillages Simplexes*, Ecole Centrale Paris: Ph.D. thesis 1994.
22. Leitner F. *Segmentation Dynamique d'Images Tridimensionelles*, Institut National Polytechnique de Grenoble: Ph.D. thesis 1993.
23. Gordon D, Udupa J. *Fast Surface Tracking in 3-D Binary Images*, Computer Vision, Graphics, and Image Processing 1989; 45: 196-214.
24. Algorri MR, Schmitt F. *Surface Reconstruction from Unstructured 3D Data*, Computer Graphics Forum 1996; 15(1): 47-60.