

# Topological Active Volumes: a Topology-Adaptive Deformable Model for Volume Segmentation

N. Barreira<sup>a,\*,1</sup>, M. G. Penedo<sup>a</sup>, L. Cohen<sup>b</sup>, M. Ortega<sup>a</sup>

<sup>a</sup>University of A Coruña, Department of Computer Science, A Coruña (Spain)

<sup>b</sup>CEREMADE, Université Paris Dauphine (France)

---

## Abstract

This paper proposes a generic methodology for segmentation and reconstruction of volumetric datasets based on a deformable model, the Topological Active Volumes (TAV). This model, based on a polyhedral mesh, integrates features of region based and boundary based segmentation methods in order to fit the contours of the objects and model its inner topology. Moreover, it implements automatic procedures, the so called topological changes, that alter the mesh structure and allow the segmentation of complex features such as pronounced curvatures or holes, as well as the detection of several objects in the scene. This work presents the TAV model and the segmentation methodology and explains how the changes in the TAV structure can improve the adjustment process. In particular, it is focused on the increase of the mesh density in complex image areas in order to improve the adjustment to object surfaces. The suitability of the mesh structure and the segmentation methodology is analyzed and the accuracy of the proposed model is proved with both synthetic and real images.

*Key words:* 3D image segmentation, Topological Active Volumes, adaptive topology,

---

## 1. Introduction

The recent development of 3D acquisition technologies has emphasized the need of techniques for the understanding of volumetric datasets in several fields

---

\*Corresponding author

*Email addresses:* nbarreira@udc.es (N. Barreira), mgpenedo@udc.es (M. G. Penedo), cohen@ceremade.dauphine.fr (L. Cohen), mortega@udc.es (M. Ortega)

<sup>1</sup>Fax: +34 981 167 160

such as medical imaging [1, 2] or non-destructive testing of products [3, 4]. Object segmentation and reconstruction are important tasks in the image processing pipeline. Nevertheless, the detection of objects in volumetric datasets is a challenging problem due to the size of the data and the variability of the features of interest.

Deformable models are well-known tools in image analysis that have been applied successfully in several fields such as image segmentation and reconstruction, pattern recognition, surgery simulation or tracking. They were introduced by Kass et al. [5] and generalized to 3D by Terzopoulos et al. [6].

Deformable models can be broadly classified into explicit or implicit models. On one hand, the explicit techniques represent the object by means of meshes (curves, surfaces or solids) which shape evolves under the influence of external and internal forces. The shape deformation is the main advantage of this kind of models since it allows the adjustment to different objects. However, since the model topology is created before the deformations, deformable models are not generally able to segment complex shapes with genus higher than 0. In order to overcome this limitation, several works have been developed in past years. McInerney and Terzopoulos [7] have proposed the *T-snakes*, an extension of the classic snakes that enables topological flexibility by means of an iterative reparametrization of the contours. To this end, they decompose the image domain into a discrete grid and compute the intersections between the contour and the grid. Delingette and Montagnat [8], Duan and Qin [9] and Lachaud et al. [10, 11] have developed techniques based on detecting the self-intersections in the evolving mesh and merging the colliding regions using a set of remeshing rules. Moreover, Pons and Boissonat [12] have proposed an approach where the mesh is iteratively updated by computing the restricted Delaunay triangulation of the deformed objects. On the other hand, the geometric deformable models [13, 14] are based on the level set formulation [15] and they are represented implicitly. This kind of representation of contours and surfaces provides topological flexibility in the segmentation process so that the topological changes are automatically handled. However, it makes difficult the user interaction and increases the computational cost. Nevertheless, some optimizations have been proposed to increase the efficiency of the segmentation process [16, 17].

The Topological Active Volumes (TAV) [18] is a parametric 3D deformable model based on the active nets model. The active nets model was first proposed by Tsumiyama and Yamamoto [19] as a variant of the deformable models that integrates features of region-based and boundary-based segmentation techniques. To this end, this model has two different kinds of nodes: external nodes for surface adjustment, and internal nodes, for modelling the inner topology. The former uses boundary information whereas the latter is related to the region information. This duality is the main advantage of this model over other models as level sets since

it allows not only the surface adjustment but also the analysis of the inner object features. The nodes are organized in a polyhedral mesh that is deformed under the influence of energy functions.

Besides, another advantage of the TAV model is its ability to perform topological changes in its structure in order to adjust to concavities, detect holes and find separate objects in the scene. The topological adaptation in the TAV model is based on the detection of complex areas in the scene and the development of appropriate changes in the mesh topology to achieve the adjustment. The complex areas are detected by means of computing several variables related to the mesh features whereas the changes are related to the insertion, deletion or transformation of nodes. This way, the insertion of new nodes in areas with pronounced curvatures improves the mesh adjustment. Also, the deletion of nodes allows the adjustment to convex areas as well as the detection of several objects in the scene. Finally, the transformation of the type of node is used for detecting features such as holes inside the objects.

This paper reviews the model and segmentation methodology as well as describes the techniques to modify the TAV structure based on the removal of nodes in order to improve the segmentation results. In this sense, this paper is focused on the analysis of complex image areas using the structure tensor in order to detect object curvatures. The increase of the mesh density in these areas improves the adjustment to the object surfaces. To this end, this paper proposes several procedures to add new nodes to the existing mesh. Moreover, the features of the mesh topology as well as the proposed methodology are analyzed and compared with other techniques. Finally, several segmentation results on images with different complexity are presented.

This paper is organized as follows. Section 2 presents the features of the TAV model. Section 3 explains the steps followed in the segmentation process. Section 4 reviews how the TAV model can perform topological changes in order to detect complex structures in the scene. Section 5 shows some segmentation results that prove the model suitability and finally, section 6 presents the conclusions and the intended future work.

## **2. Model**

A Topological Active Volume (TAV) is a three-dimensional structure composed of interrelated nodes located at the vertices of a polyhedron [18]. This polyhedron is repeated throughout the mesh and defines the neighboring relationships among nodes. There are two types of nodes: internal, inside the mesh, and external, on the surfaces. Each type of node represents different object features. The external

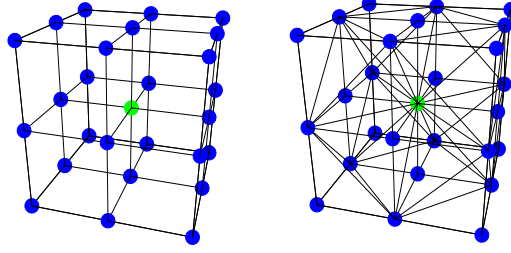


Figure 1: TAV meshes with  $3 \times 3 \times 3$  nodes. The dark nodes represent the external nodes whereas the light one is the internal node. Left: cubic mesh. Right: tetrahedral mesh.

nodes fit the surface of the object whereas the internal nodes model its inner topology. This fact allows the integration of information based on discontinuities and information based on regions. The former is associated to external nodes and the latter, to internal nodes. Figure 1 shows TAVs with cubic and tetrahedral topologies.

Parametrically, a TAV is defined as  $v(r, s, t) = (x(r, s, t), y(r, s, t), z(r, s, t))$ , where  $(r, s, t) \in ([0, 1] \times [0, 1] \times [0, 1])$ . The state of the model is governed by an energy function defined as follows:

$$E(v) = \int_0^1 \int_0^1 \int_0^1 E_{int}(v(r, s, t)) + E_{ext}(v(r, s, t)) dr ds dt \quad (1)$$

where  $E_{int}$  and  $E_{ext}$  are the internal and the external energy of the TAV, respectively. The former controls the shape and the structure of the net. Its calculus depends on first and second order derivatives which control contraction and bending, respectively. The internal energy term is defined by:

$$E_{int}(v(r, s, t)) = \alpha(|v_r(r, s, t)|^2 + |v_s(r, s, t)|^2 + |v_t(r, s, t)|^2) + \beta(|v_{rr}(r, s, t)|^2 + |v_{ss}(r, s, t)|^2 + |v_{tt}(r, s, t)|^2) + 2\gamma(|v_{rs}(r, s, t)|^2 + |v_{rt}(r, s, t)|^2 + |v_{st}(r, s, t)|^2) \quad (2)$$

where subscripts represent partial derivatives and  $\alpha$ ,  $\beta$  and  $\gamma$  are coefficients that control the smoothness of the net. In order to compute the energy, the parameter domain  $[0, 1] \times [0, 1] \times [0, 1]$  is discretized as a regular grid defined by the internode spacing  $(k, l, m)$  and the first and second derivatives are estimated using the finite differences technique in 3D.

$E_{ext}$  represents the features of the scene that guide the adjustment process and is defined as follows:

$$E_{ext}(v(r, s, t)) = \omega f[I(v(r, s, t))] + \frac{\rho}{|\mathbf{n}(r, s, t)|} \sum_{p \in \mathbf{n}(r, s, t)} \frac{1}{\|v(r, s, t) - v(p)\|} f[I(v(p))] \quad (3)$$

where  $\omega$  and  $\rho$  are weights,  $I(v(r, s, t))$  is the intensity value of the original image in the position  $v(r, s, t)$ ,  $f$  is a function related to the image intensity, and  $\mathfrak{N}(r, s, t)$  is the neighborhood of the node  $(r, s, t)$ . This way, given that the repeated polyhedron in the mesh defines the node neighborhoods, the shape of the polyhedron influences not only the flexibility of the mesh, but also the way the nodes are adjusted to the objects.

Since the internal and external nodes model different parts of the objects,  $f$  should be adapted for both types of nodes. On one hand, if the objects to detect are dark and the background is light, the energy of an internal node will be minimum when it is on a point with a low grey level. On the other hand, the energy of an external node will be minimum when it is on a discontinuity and on a light point outside the object. In this situation, function  $f$  is defined as:

$$f[I(v)] = \begin{cases} h[\overline{I(v)}_n] & \text{for internal nodes} \\ h[I_{max} - \overline{I(v)}_n + \xi(G_{max} - G(v))] + GD(v) & \text{for external nodes} \end{cases} \quad (4)$$

where  $\xi$  is a weighting term,  $I_{max}$  and  $G_{max}$  are the maximum intensity values of image  $I$  and the gradient image  $G$ , respectively,  $I(v)$  and  $G(v)$  are the intensity values of the original image and the gradient image in the node position  $v(r, s, t)$ ,  $\overline{I(v)}_n$  is the mean intensity in a  $n \times n \times n$  cube,  $h$  is an appropriate scaling function, and  $GD(v)$  is the gradient distance, this is, the distance from the node position  $v(r, s, t)$  to its nearest edge. This way, the combination of gradient terms and intensity values in an area allows the integration of both boundary and region information in the external energy term.

### 3. Segmentation methodology

The segmentation process consists of several stages as figure 2 shows. First, a mesh with an homogeneous distribution of nodes is created and located automatically over the whole image. This way, the mesh is able to detect objects located throughout the scene. After the initialization stage, the mesh energy is minimized iteratively by means of a greedy algorithm. This algorithm moves iteratively the mesh nodes to neighboring voxels where the node energy is lower. It stops when the energy functions reach a minimum, this is, when the mesh is located around the objects.

After the minimization stage, the number of nodes in each axis is recomputed to adapt the mesh size to the object size; for example, if the object is longer than wider, the number of nodes in the x-axis will be increased whereas the number of nodes in the y-axis will be decreased. The mesh is also centred over the detected objects in order to get a node distribution independent of the object location. Then, the mesh energy is minimized again.

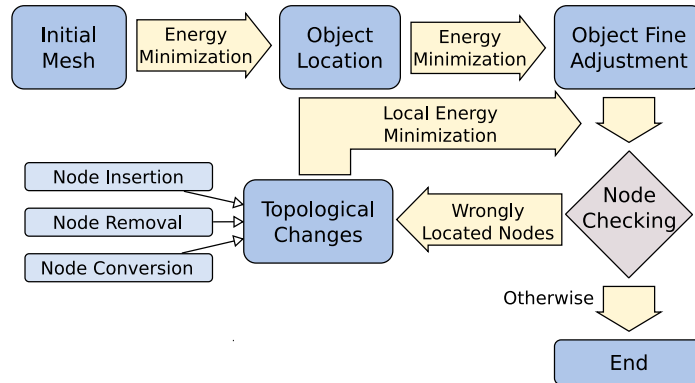


Figure 2: Stages in the segmentation process.

Finally, topological changes are performed to increase the flexibility of the model and, after a local energy minimization step, to adjust the mesh to concave surfaces, holes, or separate objects. There are three kinds of topological changes. First, the removal of nodes allows the adjustment to concave surfaces and the detection of several objects in the scene. Second, special features such as internal holes can be detected by changing the behavior of some nodes. Finally, the insertion of new nodes increases the accuracy of the adjustment to corners and sharp edges.

The TAV model has been developed using several base polyhedra. A topology based on cubes was implemented first. Even though the segmentation results using this configuration were good [18], the cubic topology has limitations in the adjustment to surfaces with pronounced curvatures as figure 3 shows. Hence, a new mesh topology based on tetrahedra was developed in order to improve the segmentation results [20]. Nevertheless, the tetrahedral meshes have an important drawback since their higher link density increases the cost of computing the node energies. For this reason, a mixed approach was developed. The mixed approach combines a cubic mesh in the early stages for a fast object detection and a tetrahedral mesh in the last stages for fine adjustment. Thus, the efficiency is increased whereas the accuracy is maintained.

#### 4. Topological changes

The TAV size and topology are defined at the beginning of the segmentation process as any other deformable model. Since the object shape is unknown, the final segmentation results could be inaccurate. Some authors have proposed to per-

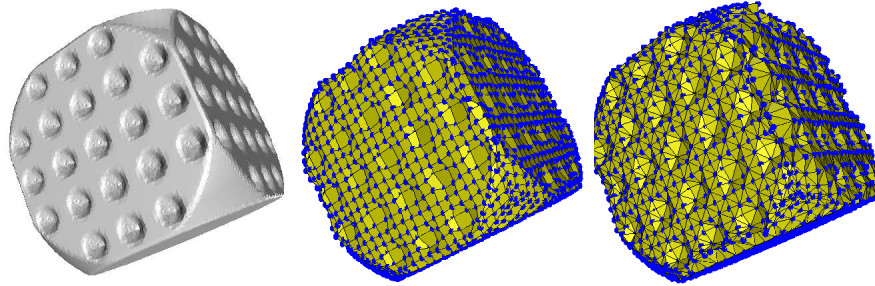


Figure 3: Adjustments of meshes based on cubes and tetrahedra. Left: original object. Center: results using a cubic mesh with  $20 \times 20 \times 20$  nodes. Right: results with a  $20 \times 20 \times 20$  tetrahedral mesh. The tetrahedral mesh is able to detect the small rounded holes in the surface of the object but the cubic mesh only detects them roughly. An increase of the mesh size does not improve the cubic results.

form topological changes in the mesh structure in order to improve the adjustment of the deformable models [7, 21, 8, 9, 10, 11].

Regarding the TAV model, three kinds of topological changes can be performed. On one hand, if nodes are removed, the mesh flexibility is increased and complex areas such as concave surfaces or external holes can be detected more accurately. On the other hand, the nature of the nodes can be changed. Since external nodes define surfaces and internal nodes represent the interior of the objects, inner holes can be segmented if some internal nodes inside the inner hole are turned into external nodes. Moreover, new nodes can be added to the original mesh. Thus, a higher mesh density can improve the adjustment to complex surfaces.

Each topological change involves several steps. The procedure starts with the identification of the complex areas where the mesh is not adjusted. After that, the mesh is reconfigured according to the kind of topological change and, finally, the energy is locally minimized to achieve the local adaptation.

This section describes the topological changes that have been developed in the TAV meshes in order to improve the adjustment to the object details.

#### 4.1. Removal of nodes

Due to its fixed topology, the TAV meshes are not able to segment complex areas such as concave regions, external holes or separate objects as figure 5 (a) shows. These areas can be pointed out by internal nodes over background as well as external nodes far from surfaces. Since an internal node over background can also point out an inner hole, the external nodes far from surfaces are chosen to detect the external complex areas [22].

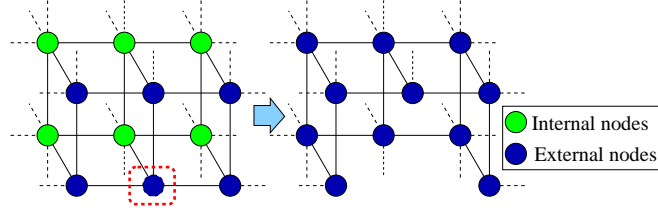


Figure 4: Removal of a node in a cubic mesh. After the topological change, the neighboring internal nodes become external since they are on the surface of the mesh.

The distance of each external node to its closest surface measures the level of adjustment. Since the Tchebycheff's inequality identifies the outliers in the set of external nodes, an external node is wrongly located if it verifies the following equation:

$$GD(r,s,t) > \mu_{GD} + 2\sigma_{GD} \quad (5)$$

where  $GD(r,s,t)$  stands for the gradient distance of the external node  $(r,s,t)$ , and  $\mu_{GD}$  and  $\sigma_{GD}$  are the average and standard deviation of the gradient distances of the set of external nodes, respectively. We are considering that the 25% of the external nodes can be outliers.

In the node removal procedure, there is a priority that weights the gradient distance. This priority forces the removal of nodes in the same area in order to avoid anarchical removals that sometimes cause a wrong segmentation result. At the beginning, the priority of each node is initialized to 1 and this value is doubled every time that a topological change is performed in the neighborhood of a node.

Once the set of external nodes far away from the object boundaries is identified, next node to remove is the node that fulfils

$$\begin{aligned} GD(r,s,t) * p(r,s,t) &> GD(r',s',t') * p(r',s',t'), \\ (r',s',t') &\in \{(x,y,z) | GD(x,y,z) > \mu_{GD} + 2\sigma_{GD}\} \end{aligned} \quad (6)$$

where  $p(r,s,t)$  is the priority of the node  $(r,s,t)$ . Thus, the choice of a node to remove relies on the node location in the image and on earlier node removals in the node neighborhood.

A node removal implies not only the elimination of a node but also the breaking of its links and the removal of the tetrahedra the node belong to. Figure 4 depicts the removal of a node in a cubic mesh (for simplicity). Note that the neighboring internal nodes become external after the node removal.

After each node removal, the mesh energy is locally minimized to accomplish the local adjustment. Figure 5 shows the results of the node removal procedure in several synthetic examples.



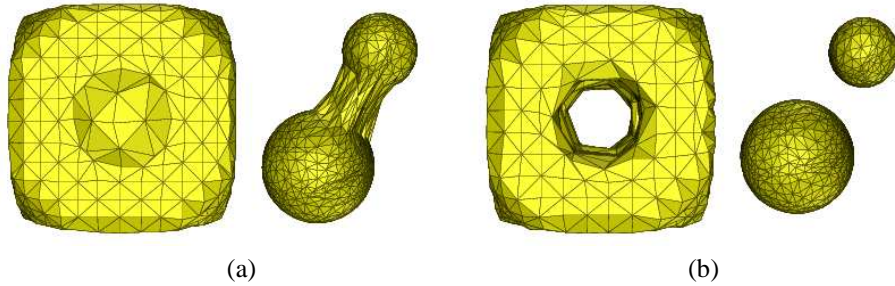


Figure 5: (a) TAV meshes before performing any topological change. (b) TAV meshes after removing nodes. The node removal procedure allows the detection of several objects on the scene as well as the segmentation of complex surfaces.

#### 4.2. Transformation of nodes

After the minimization stage, external nodes should be on the surfaces whereas internal nodes should stay inside the objects. This fact allows the reconstruction of the external surfaces as well as the analysis of the inner features of the objects. Moreover, the removal of nodes improves the surface definition. However, the initial mesh configuration is not able to detect some structures inside the objects, specifically inner holes, since internal nodes model the inside of the objects and avoid the boundaries and the background of the inner holes. Thus, some kind of mesh reconfiguration should be performed in order to segment these areas.

As the external nodes are able to detect surfaces, some external nodes should be generated inside the mesh in order to segment the inner holes. The straightforward solution is the transformation of several internal nodes into external nodes inside the mesh, the so called hole nodes. These internal nodes must be over background inside the inner holes, this is, they must be wrongly located.

Since the energy of an internal node reaches a minimum when the node is inside the objects, a high energy value represents an internal node not correctly located. Using the Tchebycheff's inequality, an internal node will be wrongly located if its energy value,  $E(r, s, t)$ , verifies that:

$$E(r, s, t) > \mu_E + 3\sigma_E \quad (7)$$

where  $\mu_E$  and  $\sigma_E$  represent the average and the standard deviation of the total energy, i.e., the sum of the internal and external energies of the set of internal nodes. Using this equation, we consider that the 89% of the internal nodes should be correctly located.

Once the outliers are identified, a tetrahedron that contains the worst located node and other three wrongly located nodes is chosen. All these nodes become

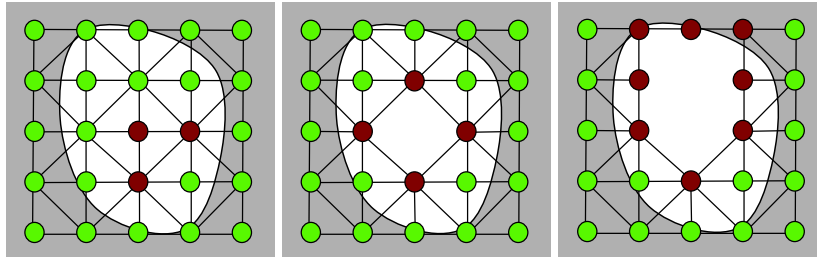


Figure 6: Transformation of nodes. Left: Internal nodes inside the hole become hole nodes. Center and right: Removals of nodes inside the hole. Light nodes represent internal nodes whereas dark nodes represent hole nodes.

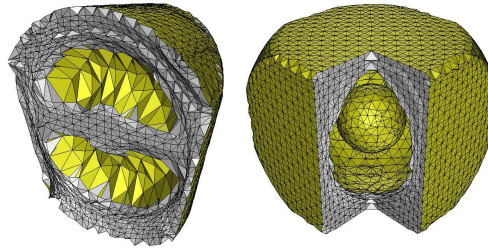


Figure 7: Segmentation results in synthetic objects with inner holes. The figures show a cross section of two synthetic objects.

*hole nodes*, this is, external nodes inside the mesh. Then, several node removals are performed from the hole nodes until no wrongly located hole node is found. This way, the hole nodes are able to detect the hole structure. These two steps, the transformation of internal nodes into hole nodes and the removal of hole nodes are repeated until no more inner holes are found. Figure 6 shows an example of this procedure in a 2D version of the tetrahedral mesh for simplicity.

Figure 7 shows the cross-sections of several segmented objects with inner holes.

#### 4.3. Insertion of new nodes

The mesh is defined initially as a discrete grid of a fixed size so the surface adjustment in complex areas such as pronounced curvatures or corners sometimes is not accurate even though the aforementioned topological changes have already been performed. A larger mesh size can improve the adjustment to complex surfaces at the expense of a higher computational effort. Moreover, since the object

complexity is a priori unknown, it is impossible to set the most appropriate mesh size. Hence, the addition of new nodes in complex areas can improve the results even with small meshes.

The development of a technique to insert new nodes in the mesh leads to two main issues. On one hand the definition of complex area and, on the other hand, the mesh reconfiguration task related to the insertion of a new node.

First of all, a metric is computed to point out complex areas such as pronounced curvatures or corners. The metric must take into account not only the object curvatures but also the reliability of the object contours. This way, the noise sensitivity is reduced and the true complex areas are enhanced. With this aim, we have used the 3D version of the metric proposed in [11]. The eigenvalues of this metric are chosen as follows:

$$\mu_1 = \left[ \frac{s^2}{s_{ref}^2} \frac{\kappa_{max}^2}{\kappa_{ref}^2} \right]_{1, \frac{\kappa_{max}}{\kappa_{ref}}}, \quad \mu_2 = \left[ \frac{\kappa_1^2}{\kappa_{max}^2} \mu_1 \right]_{1, \mu_1}, \quad \mu_3 = \left[ \frac{\kappa_2^2}{\kappa_{max}^2} \mu_1 \right]_{1, \mu_1} \quad (8)$$

where  $s$  is the intensity of the point whereas  $\kappa_1$  and  $\kappa_2$  measure the curvatures in the principal directions;  $\kappa_{max}$  is the maximum curvature value found in the image;  $s_{ref}$  and  $\kappa_{ref}$  are the intensity and curvature of reference, respectively, this is, the minimum value that is reliable. They were empirically set to 90% of the maximum values of intensity and curvature detected in the image. The notation  $[\cdot]_{a,b}$  constrains its arguments between the bounds  $a$  and  $b$ .

The curvature and, by extension, the node density of a deformable surface aligned with the image contours is determined by  $\mu_2$  and  $\mu_3$ . Note the influence of the voxel intensity in the computation of the curvatures.

The structure tensor is used for computing the intensity and the curvature from the image. It is defined as a positive symmetrical matrix which can be evaluated at each point of the image. Its spectral decomposition characterizes the way the image gradient changes in the neighborhood of a chosen voxel.

If  $\nabla_{\sigma_1} I$  is the gradient of the image  $I$  smoothed by a Gaussian filter of standard deviation  $\sigma_1$ , the gradient tensor  $J_{\sigma_1}$  is obtained by computing, at each point of the image, the Cartesian product of the gradient vector  $(I_x I_y I_z)^T$  with itself:

$$J_{\sigma_1} = \nabla_{\sigma_1} I \times \nabla_{\sigma_1} I = \nabla_{\sigma_1} I \times \nabla_{\sigma_1} I^T = \begin{pmatrix} I_x \\ I_y \\ I_z \end{pmatrix} \times (I_x I_y I_z) = \begin{pmatrix} I_x^2 & I_x I_y & I_x I_z \\ I_x I_y & I_y^2 & I_y I_z \\ I_x I_z & I_y I_z & I_z^2 \end{pmatrix} \quad (9)$$

The convolution of this tensor with a Gaussian filter of standard deviation  $\rho$  leads to the structure tensor:

$$J_{\sigma_2, \sigma_1} = \begin{pmatrix} g_{\sigma_2} * (I_x^2) & g_{\sigma_2} * (I_x I_y) & g_{\sigma_2} * (I_x I_z) \\ g_{\sigma_2} * (I_x I_y) & g_{\sigma_2} * (I_y^2) & g_{\sigma_2} * (I_y I_z) \\ g_{\sigma_2} * (I_x I_z) & g_{\sigma_2} * (I_y I_z) & g_{\sigma_2} * (I_z^2) \end{pmatrix} \quad (10)$$

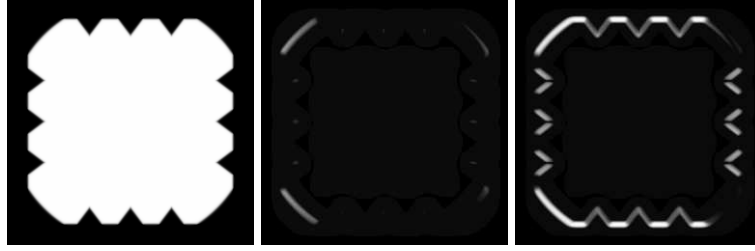


Figure 8: Curvature metric. Left: original slice of the 3D image. Center and right: curvature of the contours in the principal directions represented as  $\mu_2$  and  $\mu_3$ , respectively.

where  $\sigma_2$  defines the neighborhood in which the image structures are computed.

The eigen-decomposition of the structure tensor describes the local features. The matrix  $J_{\sigma_2, \sigma_1}$  has three positive eigenvalues,  $\xi_1$ ,  $\xi_2$  and  $\xi_3$ , and three corresponding eigenvectors  $v_1$ ,  $v_2$  and  $v_3$ . The first eigenvector  $v_1$  is aligned with the surface normal whereas the remaining two eigenvectors,  $v_2$  and  $v_3$ , lie in the principal directions of the surface. This way, the highest eigenvalue measures the intensity of the contours whereas the lowest eigenvalues are equivalent to the curvature of the contours.

According to [11], the intensity and the curvatures are computed from the eigenvalues of the structure tensor as follows

$$s = \sqrt{\xi_1}, \quad \kappa_1 = \frac{1}{\rho} \sqrt{\frac{\xi_2}{\xi_1 + \varepsilon}}, \quad \kappa_2 = \frac{1}{\rho} \sqrt{\frac{\xi_3}{\xi_1 + \varepsilon}} \quad (11)$$

where  $\varepsilon$  is a positive constant that assures the definition of the metric in regions where  $\xi_1 \approx 0$ . Experimentally, this constant was set to  $\varepsilon = 0.1e_1^{max}$ .

Figure 8 shows the proposed metric computed from a slice of the object in figure 3.

Once the curvature metric is computed, next task is the insertion of new nodes in complex areas. The remeshing implies the development of a technique to insert a node in a tetrahedron and a procedure to find the tetrahedra related to the complex areas pointed out by the metric.

Regarding the insertion of a node in a tetrahedron, two alternatives have been taken into account. The first one is the split of a link and the second one is the division of a face of a tetrahedron. The former splits each tetrahedra that shares the involved link into two tetrahedra whereas the later divides the tetrahedron into three tetrahedra. In both cases, the node is inserted in the middle point of the corresponding structure. Figure 9 depicts these two situations.

The procedure for the insertion of a node by means of the split of a link has several steps. First, the links between external nodes are analyzed. In order to

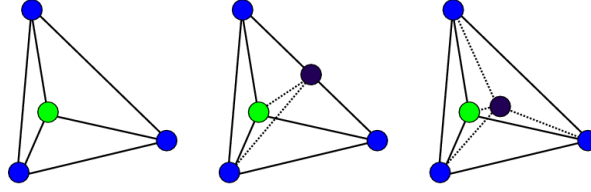


Figure 9: Methods for the insertion of a node. Left: original tetrahedron. Center: a connection split. Right: a face split.

produce uniform meshes and avoid an excessive remeshing in some areas, only external connections longer than a given threshold are taken into account. Typically, this threshold is set as follows

$$l_{threshold} = \mu_l + \sigma_l \quad (12)$$

where  $\mu_l$  is the media and  $\sigma_l$  is the standard deviation of the length of the links between external nodes. This threshold decreases linearly with the iterations.

The curvature of the set of links longer than  $l_{threshold}$  is checked in a restricted neighborhood of the image following a line defined by the middle point of the link and an inner point of the mesh. Since the object curvature affects regions and it is not an isolated feature, the analysis of the points that lie in a line is enough to find the closest curvatures. If the link belongs to tetrahedra with internal nodes, the inner point is computed as the average coordinates of the internal nodes. Otherwise, the inner point is computed as the average coordinates of the external nodes of the tetrahedra that share the link. The curvature is checked up to a distance equal to the previously computed threshold  $l_{threshold}$ . This way, only the closest curvatures are taken into account. Finally, a node is inserted in each link located near to curvature and the mesh energy is locally minimized. In the adjustment, the node generally follows the line defined in the curvature check.

The insertion of nodes can be also performed by means of the division of an external face of a tetrahedron. The procedure is similar to the split of links. First, only the tetrahedra composed by external nodes are analyzed. The same length threshold is applied to the links of the tetrahedra to prioritize the divisions and avoid an excessive remeshing. Once the set of candidate tetrahedra is obtained, the curvature is checked following a line defined by the middle point of a face composed by three external nodes and the coordinates of the other node in the tetrahedron. The curvature is checked up to a distance equal to the length threshold too. Finally, a node is inserted in the middle point of the face of each tetrahedron near to curvature and the TAV energy is locally minimized.

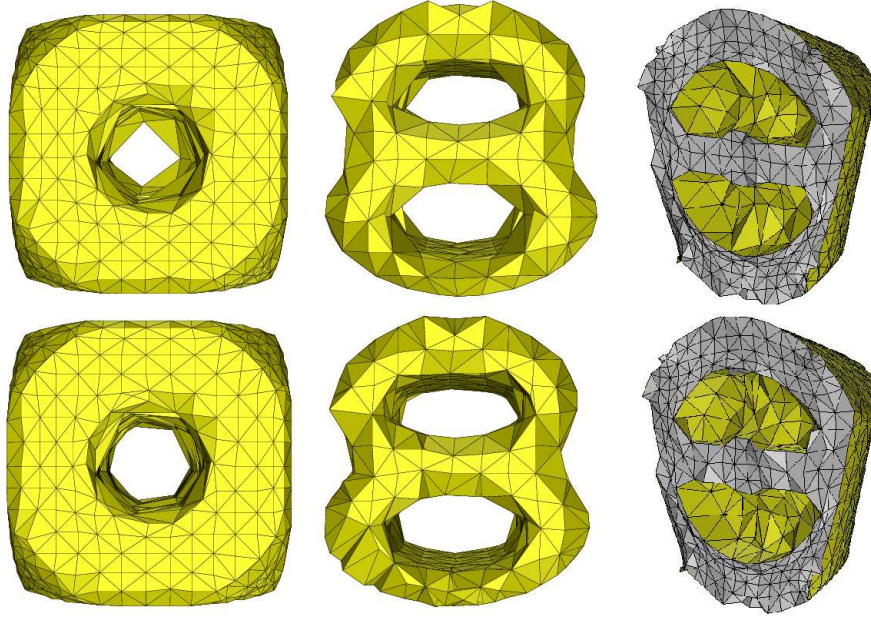


Figure 10: Insertion of nodes in synthetic images. First row: before the insertion of nodes. Second row: after the insertion of nodes.

Figure 10 shows the results of inserting nodes in TAV meshes. Note the improvement in the adjustment to curved surfaces in the first and second columns and the better definition of the internal hole in the third column. In these examples, the procedure to split links was used for the insertion of new nodes in the mesh. Next section analyzes the suitability of both methods.

## 5. Results

This section shows the segmentation results in both synthetic and real 3D images. In the examples, the input image was used in the external energy term for both internal and external nodes. The gradient images were computed using a 3D Canny detector. The model parameters as well as the initial mesh size of the examples presented in this section are summarized in table 1.

First of all, a statistical analysis has been performed in order to validate the segmentation results. To this purpose, the values of sensitivity and specificity have been computed in several segmentation examples. In this case, the sensitivity measures the proportion of object voxels that are segmented by the TAV whereas the specificity measures the proportion of background voxels not segmented by the

Table 1: Parameters used in the segmentation examples

<i>Figures</i>	<i>Mesh size</i>	$\alpha$	$\beta$	$\gamma$	$\rho$	$\omega$	$\xi$
10 (left)	$15 \times 15 \times 15$	3.0	0.00001	0.00001	3.0	3.0	5.0
10 (center)	$9 \times 9 \times 9$	1.5	0.00001	0.00001	3.0	3.0	5.0
10 (right)	$17 \times 17 \times 17$	2.5	0.00001	0.00001	3.0	3.0	5.0
11		2.5	0.00001	0.00001	3.0	3.0	5.0
12		3.0	0.00001	0.00001	3.0	3.0	5.0
16 (left)	$22 \times 22 \times 22$	3.5	0.00001	0.00001	3.0	3.5	5.0
16 (right)	$20 \times 20 \times 20$	2.5	0.00001	0.00001	3.0	3.0	5.0
17	$34 \times 34 \times 34$	2.5	0.00001	0.00001	3.0	3.0	5.0
18	$40 \times 40 \times 40$	2.5	0.00001	0.00001	3.0	3.0	5.0
19	$46 \times 46 \times 46$	2.5	0.00001	0.00001	3.0	3.0	5.0

TAV mesh. These measures are computed as follows

$$Sensitivity = \frac{TP}{TP+FN} \qquad Specificity = \frac{TN}{TN+FP} \qquad (13)$$

where  $TP$  is the number of true positives, i.e. the number of voxels correctly segmented by the mesh,  $TN$  is the number of true negatives, i.e. the number of background pixels not segmented,  $FP$  is the number of false positives, i.e. the number of background pixels segmented by the mesh, and  $FN$  is the number of false negatives, i.e. the number of object voxels not segmented.

The first experiment involves the analysis of the mesh features. To this end, the sensitivity and specificity measures were computed from the segmentation results obtained with the cubic topology [18] and the tetrahedral topology. Objects with concave and convex surfaces were studied. For the convex case, a 3D image with a sphere was selected to perform the statistical analysis. Figure 11 shows the segmentation results with both cubic and tetrahedral topologies using the greedy strategy as well as the evolution of the sensitivity and specificity values with respect to the mesh size. The sensitivity value, i.e., the proportion of object voxels correctly segmented, increases with the mesh size in both topologies since a large number of nodes improves the adjustment. Also, the tetrahedral adjustment is always better than the cubic adjustment. Regarding the specificity, the values slightly decrease with the mesh size because the segmentation of more object voxels can include some neighboring background voxels since the faces are not curved. However, the specificity values are always greater than 0.999, which is a good adjustment. The results of the tetrahedral approach are also better than the results of the cubic topology.

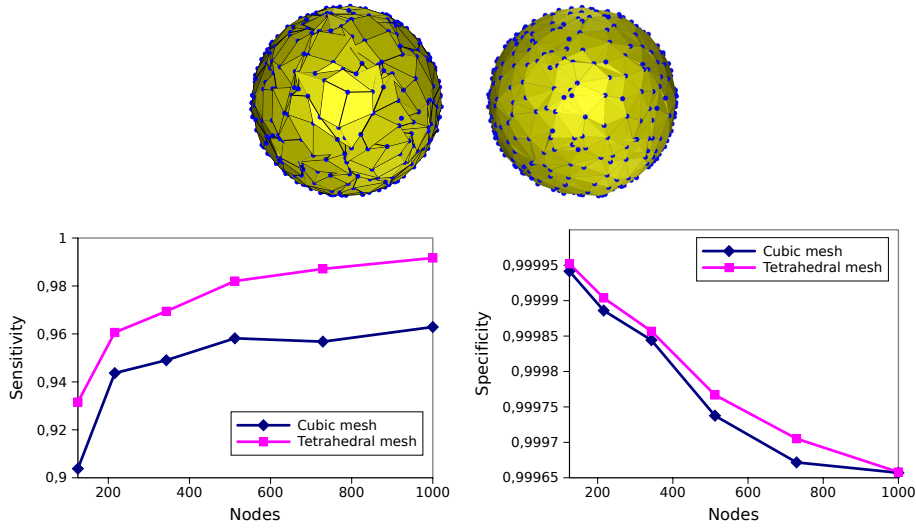


Figure 11: Sensitivity (left) and specificity (right) of a figure with convex surfaces.

In the concave case, an object with small holes was segmented with both cubic and tetrahedral topologies (figure 3). The holes are big enough for not requiring topological changes. Figure 12 shows the segmentation results using the greedy strategy and the evolution of the statistical measurements. The results of the concave case are different from the convex case. Since the cubic mesh is not adjusted correctly to the concave surfaces, the specificity values in the cubic mesh are lower, i.e., a high number of background voxels are considered as object areas. As a consequence, all the object voxels in the hole are also segmented so that the sensitivity is greater in the cubic meshes. The tetrahedral meshes are able to discard background voxels so their specificity is higher but their sensitivity is a bit lower because they fail in the detection of some object voxels placed at the concave surfaces. However, the adjustment is good since the worst segmentation results have a sensitivity higher than 0.95 and a specificity higher than 0.98.

The second experiment studies the suitability of the proposed methodology. Figure 13 shows the measures of sensitivity and specificity through the segmentation process of the images from figure 10. Since the mesh initially covers the whole image, the regions of segmented background are reduced as the segmentation process advances, so that the specificity is increased at each step of the segmentation procedure. On the contrary, the sensitivity is slightly reduced in each segmentation step since an improvement in the node adjustment sometimes reduces the seg-



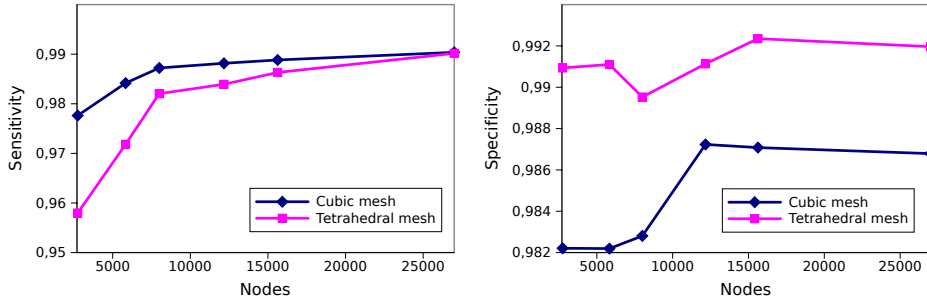


Figure 12: Sensitivity (left) and specificity (right) of a figure with concave surfaces (figure 3).

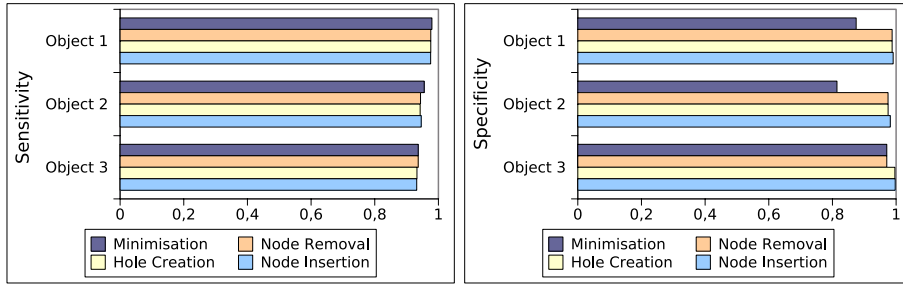


Figure 13: Sensitivity (left) and specificity (right) measures at each segmentation step of the images in figure 10

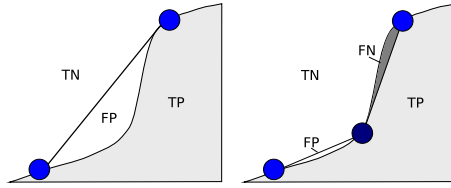


Figure 14: Example of mesh adjustment before (left) and after (right) the insertion of a node. The insertion of new nodes diminishes the false positive rate but sometimes increases the number of false negatives due to the object surface features.

mented object area as figure 14 shows. This situation arises when curved surfaces are approximated by polygonal meshes. However, at the end of the segmentation process, the specificity is near 1 and the sensitivity is greater than 0,92 in all the examples.

Moreover, our model was compared with the B-surface algorithm [23] and the Huang’s method [24] in order to test the adjustment to the object surfaces. A sphere with radius 40 was segmented and the distances from the TAV surface points to the sphere points were computed. Table 2 summarizes the results with different mesh sizes. The distance of the external nodes to the object surfaces is always 0 since the external nodes are adjusted to the object surfaces. Moreover, the distance from some interpolated points (the midpoint of each external face) to the sphere surface was also computed. The results show that this value depends, mainly, on the mesh density. The global mean values (external nodes + interpolated points) obtained with the TAV mesh improve the results of the previous methods. The B-surface algorithm obtains a 0.05 mean distance error whereas the Huang’s method obtains a 0.1 mean distance error. However, the maximum distance error of the B-surface algorithm is lower (1 voxel) because it has more surface model points (52 surface patches with 400 surface points each) than the TAV model (728 external nodes in the  $12 \times 12 \times 12$  mesh).

The proposed methods for the insertion of nodes have been statistically analyzed too. Figure 15 shows the evolution of the sensitivity and the specificity in several segmentation examples with different mesh sizes. Regarding the specificity, the split of links and the combination of both techniques produce better results than the division of faces. The sensitivity values are very close for all the proposed methods and only in the last example with inner holes, the measurements of the division of faces are slightly higher.

Moreover, figure 15 shows the behavior of the segmentation results with regard to the mesh size. On one hand, the sensitivity increases with the mesh size since a higher node density allows a better surface adjustment. On other hand,

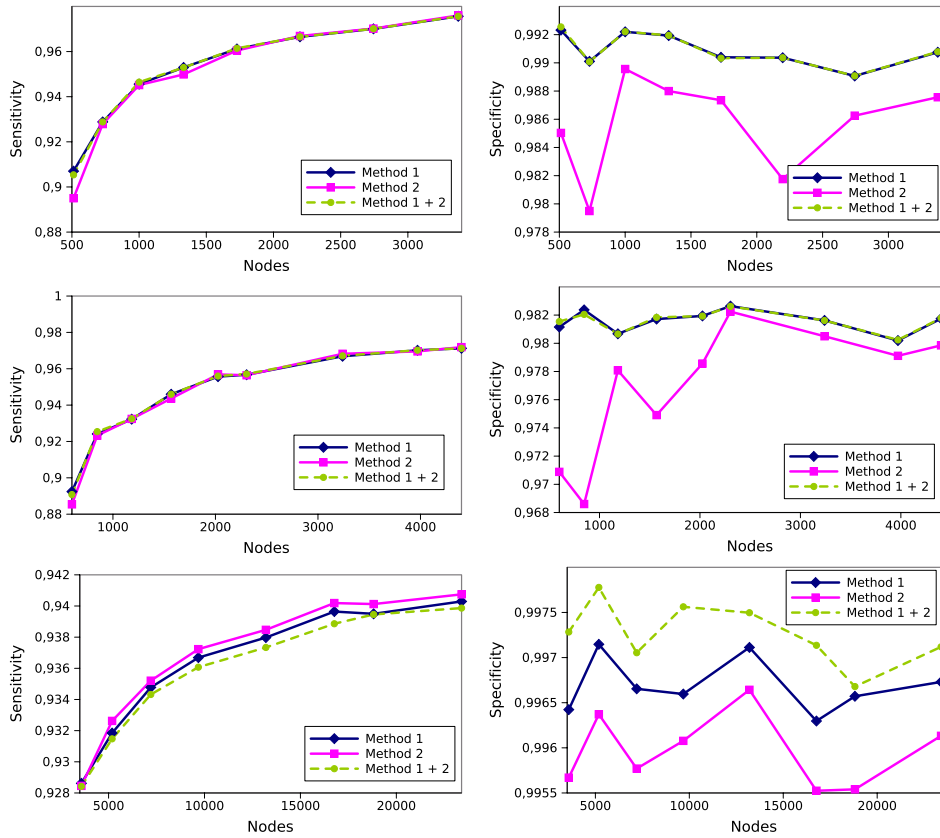


Figure 15: Evolution of the sensitivity (left column) and specificity (right column) in the methods for the insertion of nodes. The charts show the statistical segmentation results of the images shown in figure 10. Method 1 stands for the insertion of nodes by means of the split of links, method 2 is the division of faces and method 1 + 2 stands for the combination of both techniques.

Table 2: Evolution of the distance (in voxels) from TAV surface points to the actual object surface with respect to the mesh size in figure 11.

		<b>Mesh size</b>			
		$9 \times 9 \times 9$	$10 \times 10 \times 10$	$11 \times 11 \times 11$	$12 \times 12 \times 12$
<b>External Nodes</b>	<i>Mean distance</i>	0	0	0	0
	<i>Max. distance</i>	0	0	0	0
<b>Interpolated Points</b>	<i>Mean distance</i>	0.198	0.075	0.054	0.052
	<i>Max. distance</i>	2	2	2	2
<b>Global mean distance</b>		0.142	0.054	0.039	0.037

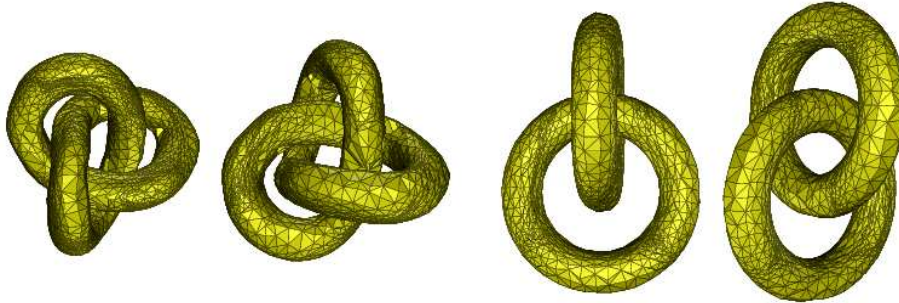


Figure 16: Segmentation results on synthetic images.

the specificity is very high, close to 1, and independent of the mesh size since the background area is bigger than the object area so a small increase of true negatives related to a higher mesh size does not affect the specificity value.

Figure 16 shows the segmentation results of some synthetic complex objects whereas figures 17, 18 and 19 shows the results in real CT images. In these cases, the surface reconstruction includes the node distribution in order to show how the nodes are adjusted to the object features. Note how the new nodes are inserted in areas with complex curvatures, how both internal and external holes are detected and how the noise is skipped in figure 19.

## 6. Conclusions

This paper presents the Topological Active Volumes (TAV) model, a 3D deformable model focused on segmentation and reconstruction tasks. This model allows the adjustment of complex features such as holes or pronounced curvatures as well

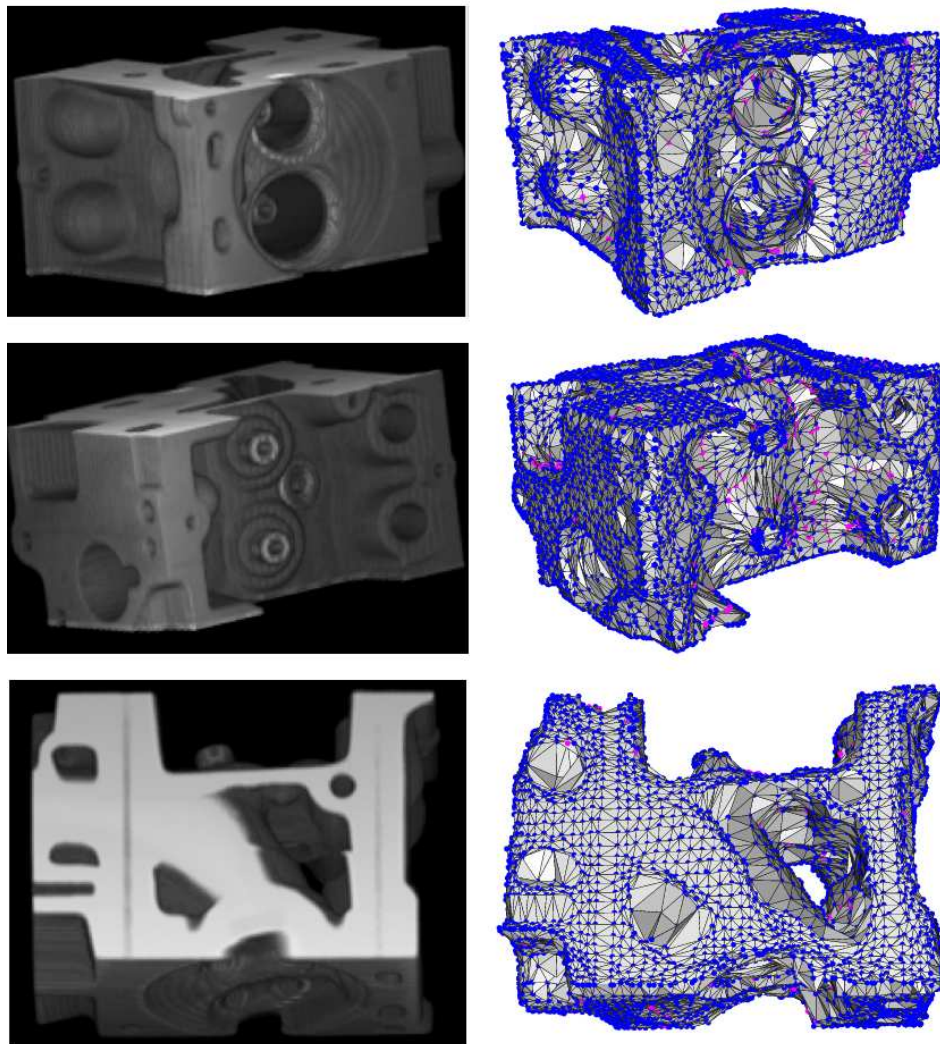


Figure 17: Segmentation of an engine from CT images. Left column: engine reconstruction using 3D Doctor software. Right column: TAV volume reconstruction. The blue nodes represent the external nodes whereas the purple nodes represent the new inserted nodes.

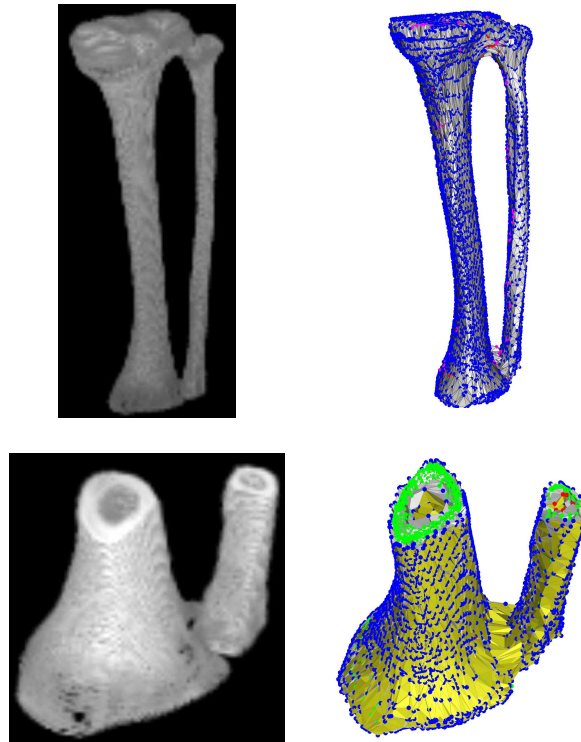


Figure 18: Segmentation of a tibia and a fibula from CT images. Left column shows the bone reconstruction using 3D Doctor software and the right column, the TAV surface reconstruction. The second row shows a cross section of both bones. The blue nodes represent the external nodes, the green nodes are the internal nodes, the purple nodes represent the new inserted nodes and the red nodes are the hole nodes used as a seed to create the inner holes.

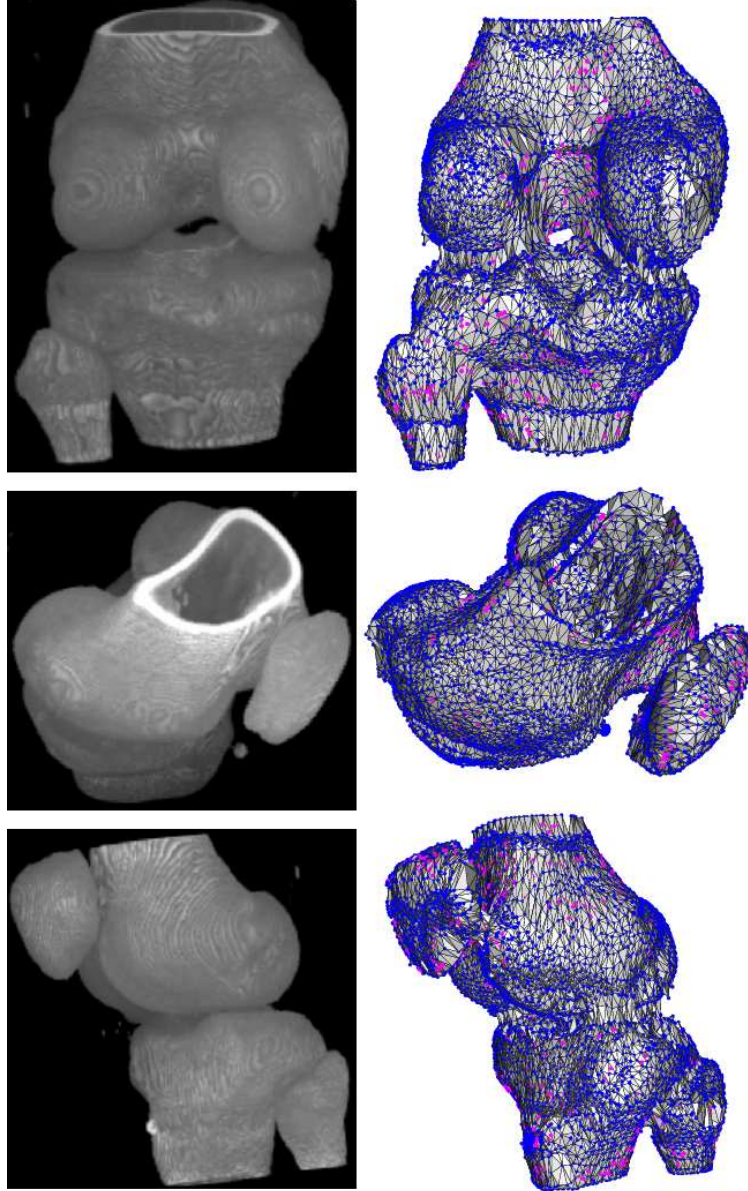


Figure 19: Segmentation of a knee from CT images. Left column shows the bone reconstruction using 3D Doctor software and the right column, the TAV surface reconstruction. The blue nodes represent the external nodes and the purple nodes represent the new inserted nodes.

as the detection of several objects in a scene by means of topological changes in its structure. Three kinds of topological changes have been proposed. First, the removal of nodes, that increases the flexibility of the mesh and allows the split of the mesh. Also, the change of the nature of the nodes allows the detection of inner holes inside the mesh. Finally, the insertion of new nodes increases the node density in some areas and improves the adjustment to complex surfaces. To this end, two methods have been proposed: the split of links and the division of faces. The statistical analysis of the segmentation results proves the superiority of the former method and the suitability of the overall segmentation process.

Future work includes the improvement of the efficiency and the accuracy of the model by means of the development of other minimization techniques as well as the addition of different energy terms.

## References

- [1] A. Falcao, F. Bergo, Interactive volume segmentation with differential image foresting transforms, *IEEE Trans. on Medical Imaging* 23 (9) (2004) 1100–1108.
- [2] M. del Fresno, M. Vénere, A. Clausse, A combined region growing and deformable model method for extraction of closed surfaces in 3d ct and mri scans, *Computerized Medical Imaging and Graphics* 33 (5) (2009) 369 – 376.
- [3] C. Heinzl, J. Kastner, E. Gröller, Surface extraction from multi-material components for metrology using dual energy ct, *IEEE Trans. on Visualization and Computer Graphics* 13 (6) (2007) 1520–1527.
- [4] M. Hadwiger, L. Fritz, C. Rezk-Salama, T. Höllt, G. Geier, T. Pabel, Interactive volume exploration for feature detection and quantification in industrial ct data, *IEEE Trans. on Visualization and Computer Graphics* 14 (6) (2008) 1507–1514.
- [5] M. Kass, A. Witkin, D. Terzopoulos, Active contour models, *International Journal of Computer Vision* 1 (2) (1988) 321–323.
- [6] D. Terzopoulos, A. Witkin, M. Kass, Constraints on deformable models: Recovering 3D shape and nonrigid motion, *Artificial Intelligence* 36 (1) (1988) 91–123.
- [7] T. McInerney, D. Terzopoulos, T-snakes: Topology adaptive snakes, *Medical Image Analysis* 4 (2) (2000) 73–91.



- [8] H. Delingette, J. Montagnat, Shape and topology constraints on parametric active contours, *Computer Vision and Image Understanding* 83 (2001) 140–171.
- [9] Y. Duan, H. Qin, A subdivision-based deformable model for surface reconstruction of unknown topology, *Graph. Models* 66 (4) (2004) 181–202.
- [10] J.-O. Lachaud, A. Montanvert, Deformable meshes with automated topology changes for coarse-to-fine 3D surface extraction, *Medical Image Analysis* 3 (2) (1999) 187–207.
- [11] J.-O. Lachaud, B. Taton, Deformable model with a complexity independent from image resolution, *Computer Vision and Image Understanding* 99 (2005) 453–475.
- [12] J.-P. Pons, J.-D. Boissonnat, Delaunay deformable models: Topology-adaptive meshes based on the restricted delaunay triangulation, in: *IEEE Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1–8.
- [13] V. Caselles, R. Kimmel, G. Sapiro, C. Sbert, Three dimensional object modeling via minimal surfaces, in: *Proc. Europ. Conf. Comput. Vis. (ECCV'06)*, Springer-Verlag, 1996, pp. 97–106.
- [14] A. Yezzi, S. Kichenassamy, A. Kumar, P. Olver, A. Tannenbaum, A geometric snake model for segmentation of medical imagery, *IEEE Trans. On Medical Imaging* 16 (2) (1997) 199–209.
- [15] S. Osher, J. A. Sethian, Fronts propagating with curvature dependent speed: algorithms based on hamilton-jacobi formulations, *Journal of Computational Physics* 79 (1988) 12–49.
- [16] D. Adalsteinsson, J. A. Sethian, A fast level set method for propagating interfaces, *Journal of Computational Physics* 118 (1995) 269–277.
- [17] J. Strain, Tree methods for moving interfaces, *Journal of Computational Physics* 15 (2) (1999) 616–648.
- [18] N. Barreira, M. G. Penedo, Topological Active Volumes, *EURASIP Journal on Applied Signal Processing* 13 (1) (2005) 1937–1947.
- [19] K. Tsumiyama, K. Yamamoto, Active net: Active net model for region extraction, *IPSI SIG notes* 89 (96) (1989) 1–8.

- [20] N. Barreira, M. G. Penedo, M. Ortega, J. Rouco, On the improvement of the topological active volumes model: A tetrahedral approach, in: *Third International Conference on Computer Vision Theory and Applications*, 2008, pp. 529–534.
- [21] T. Mcinerney, D. Terzopoulos, Topology adaptive deformable surfaces for medical image volume segmentation, *IEEE Trans. on Medical Imaging* 18 (1999) 840–850.
- [22] N. Barreira, M. G. Penedo, C. Alonso, J. Rouco, Handling topological changes in the topological active volumes model, *Lecture Notes in Computer Science: Image Analysis and Recognition* 5112 (2008) 122–131.
- [23] X. Chen, E. K. Teoh, 3d object segmentation using b-surface, *Image Vision Comput.* 23 (14) (2005) 1237–1249.
- [24] H. Jiantao, A. Amini, Anatomical object volumes from deformable b-spline surface models, *International Conference on Image Processing* 1 (1998) 732.