A Semi-Automatic Surface Reconstruction Framework Based on T-Surfaces and Isosurface Extraction Methods

EDILBERTO STRAUSS[2], WALTER JIMÉNEZ[1], GILSON GIRALDI[1], RODRIGO SILVA[1,2], ANTONIO A. F. OLIVEIRA[2]

[1]LNCC–National Laboratory for Scientific Computing -
Av. Getulio Vargas, 333, 25651-070 Rio de Janeiro, RJ, Brazil
{gilson,walter}@lncc.br
[2]LCG–Computer Graphics Laboratory, UFRJ-COPPE -
Mail Box 68511, 21945-970 Rio de Janeiro, RJ, Brazil
{strauss,rodrigo,oliveira}@lcg.ufrj.br

Abstract. In this paper we present a new approach which integrates the T-Surfaces model and isosurface generation methods in a general framework for surface reconstruction in 3D medical images. T-Surfaces is a deformable model based on a triangulation of the image domain, a discrete surface model and an image threshold. Two types of isosurface generation methods are considered: the continuation ones and the marching ones. The former is useful during the reparameterization of T-Surfaces while the later is suitable to initialize the model closer the boundary. Specifically, in a first stage, the T-Surfaces grid and the threshold are used to define a coarser image resolution. This field is thresholded to get a 0-1 function which is processed by a marching method to generate polygonal surfaces whose interior may contain the desired objects. If a polygonal surface involves more than one object, then the resolution is increased in that region and the marching applied again. Next, we apply T-Surfaces to improved the result. If the obtained topology remains incorrect, we enable the user to modify the topology by an interactive method based on the T-Surfaces framework. Finally, we demonstrate the utility of diffusion methods for our approach.

1 Introduction

Parametric Deformable Models, which includes the popular *snake models* [7] and deformable surfaces [10], are well known techniques for boundary extraction and tracking in 2D/3D images.

These models consist basically of an elastic curve (or surface) which can dynamically conform to object shapes in response to internal (elastic) forces and external forces (image and constraint ones).

However, for most of these methods the topology of the structures of interest must be known in advance since the mathematical model can not deal with topological changes without adding extra machinery [10].

Recently, McInerney and Terzopoulos [10] proposed the T-Snakes/T-Surfaces model to add topological capabilities (*splits and merges*) to a parametric model. The basic idea is to embed a discrete deformable model within the framework of a simplicial domain decomposition (*triangulation*) of the image domain. In this framework, the reparameterization is based on the projection of the curve/surface over the triangulation and on a *Characteristic Function* which distinguishes the grid nodes interior of the (closed) curve/surface from the other ones. The set of simplices in which the Characteristic Function changes value (*Combinatorial Manifold*) gives a simple way to reparameterize the model.

This reparameterization process allows to reconstruct surfaces with significant protrusions and objects with bifurcations. Furthermore, that process can deal easily with self-intersections of the surface during the model evolution. Also, T-Snakes/T-Surfaces depends on some threshold to define a normal force which is used to drive the model towards the targets [10].

Based on these elements (discrete surface model, simplicial decomposition framework and threshold) we propose in this paper a semi-automatic segmentation approach for $3D$ medical images based on isosurface extraction methods and the T-Surfaces model.

Among the isosurface methods [13], two types are considered in this paper: continuation and marching methods.

Continuation methods propagate the surface from a set of seed cells. The key idea is to use the spatial coherence of the isosurface during its extraction [13, 1].

As we have already stressed in previous works [4, 5], continuation methods are useful during the reparameterization and initialization of T-Surfaces model if some topological and scale restrictions for the targets are supposed.

In this paper we discard these restrictions. We show that continuation methods remain suitable in the reparameterization process. However, Marching methods should be used to initialize T-Surfaces closer the boundary. These are the starting points of this work.

Specifically, in a first stage, we use the T-Surfaces grid to define a coarser image resolution by sampling the image field over the grid nodes. The obtained low resolution image field is thresholded to get a binary field, which we call an *Object Characteristic Function*. This field will be searched for the isosurface generation method. The obtained result may be a rough approximation of the target. However, the surfaces obtained are in general not smooth and topological defects like holes may happens. Besides, due to inhomogeneities of the image field some components of the objects may be splited as well as merged due to the low image resolution used. We improve the result by using the T-Surfaces model.

The grid resolution is application dependent. However, an important point of our method is its multiresolution/multigrid nature: having resolved (segmented) the image in a coarser (grid) resolution we can detect regions where the grid has to be refined and then apply the method again only in these regions.

For noisy images, diffusion methods can improve both the isosurface extraction and the T-Surfaces result. If the extracted topology remains incorrect, even at the finest resolution, we propose an interactive procedure, based on T-Surfaces framework, to force merges/splits. This method is an extension of that one described in [4].

In the following section, T-Surfaces framework is developed. In section 3 we describe previous works. Sections 4,5 discuss the isosurface generation methods in the T-Surfaces context. In section 6 we present our method. The experimental results are shown on section 7. Finally, we compare our method with related ones (section 8) and present final conclusions.

2 T-Surfaces

The T-Surfaces approach is composed basically by three components [10]: (1) a tetrahedral decomposition (CF-Triangulation) of the image domain $D \subset \Re^3$; (2) a particle model of the deformable surface; (3) a *Characteristic Function* χ defined on the grid nodes which distinguishes the interior ($Int(S)$) from the exterior ($Ext(S)$) of a surface S:

$$\chi : D \subset \Re^3 \rightarrow \{0, 1\} \tag{1}$$

where $\chi(p) = 1$ if $p \in Int(S)$ and $\chi(p) = 0$, otherwise, where p is a node of the grid.

Following the classical nomenclature, a vertex of a tetrahedron is called a *node* and the collection of nodes and triangle edges is called the grid Γ_s. A tetrahedron (also called a simplex) σ is a *transverse* one if the characteristic function χ in equation (1) changes its value in σ. Analogously, for an edge.

In this framework, the reparameterization of a surface is done by [10]: (1)Taking the intersections points of the surface with the grid; (2)Find the set of transverse tetrahedra (*Combinatorial Manifold*); (3)For each transverse edge choose an intersection point belonging to it; (4) Connect these points properly.

In this reparameterization process, the transverse simplices play a central role. Given such a simplex, we choose in each transverse edge an intersection point to generate the new surface patch. In general, we will have three or four transverse edges in each transverse tetrahedron (Figure 4). The former gives a triangular patch and the later defines two triangles. So, at the end of the step (4) we have a triangular mesh. Each triangle is called a *triangular element* [10].

As an example for 2D, consider the characteristic functions (χ_1 and χ_2) relative to the two contours pictured in Figure 1. The functions are defined on the vertices of a CF-triangulation of the plane. The vertices marked are those where $\max\{\chi_1, \chi_2\} = 1$. Observe that they are enclosed by a merge of the contours. This merge can be approximated by a curve belonging to the region obtained by tracing the transverse triangles. The same would be true for more than two contours (and obviously for only one).

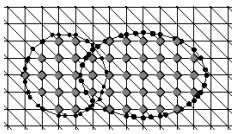


Figure 1: Two snakes colliding with the inside grid nodes and snaxels marked.

2.1 Discrete Model

A T-Surface, can be seen as a discrete form of the classical parametric deformable surfaces [10]. It is defined as a closed elastic mesh. Each node is called a *node element* and each pair of nodes v_i, v_{i+1} is called a *model element*.

The node elements are linked by springs, whose natural length we set to zero. Hence, a tensile force can be defined by:

$$\overrightarrow{\alpha_i} = \sum_j \overrightarrow{S}_{ij} \text{ where } \overrightarrow{S}_{ij} = c\,(r_{ij})\,, \tag{2}$$

where c is a scale factor. The model also has a normal force which can be weight as follows [10]:

$$F_i = k\,(sign_i)\,n_i, \tag{3}$$

where n_i is the normal vector at node i, k is a scale factor, and $sign_i = +1$ if $I(v_i) > T$ and $sign_i = -1$ otherwise (T is a threshold for image I). This force is used to push the model towards image edges until it is opposed by external image forces.

The forces given in (2)-(3) are internal forces. The external force is defined as a function of the image data, according to the features we seek. One possibility is:

$$Image :: Force :: f_i^t = -\gamma_i \nabla P; \quad P = \|\nabla I\|^2. \quad (4)$$

The evolution of the surface is governed by the following dynamical system:

$$v_i^{(t+\Delta t)} = v_i^t + h_i \left(\overrightarrow{\alpha_i}^t + \overrightarrow{F_i}^t + \overrightarrow{f_i}^t \right), \quad (5)$$

where h_i is an evolution step. During the T-Surfaces evolution some grid nodes become interior to a Surface. Such nodes are called *burnt nodes* and its identification is fundamental to update the characteristic function [10]. To deal with self-intersections of the surface the T-Surface model incorporates an entropy condition: *once a node is burnt it stays burnt*. A termination condition is obtained based on the number of deformations steps that a simplex has remained a transverse one.

3 Previous Works

The threshold T used in the normal force (3) plays an important role in the T-Surfaces model. If it was not chosen properly, the T-Surfaces can be frozen in a region far from the target(s) [9, 10].

The choice of T is more critical when two objects to be segmented are too close like in Figure 2. In this figure the grid nodes marked are the ones whose image intensity falls bellow the threshold T.

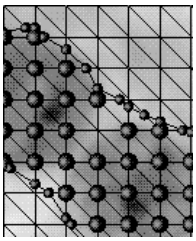


Figure 2: T-Snake and grid nodes marked.

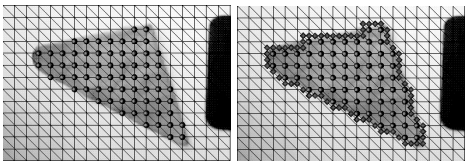
For T-Surfaces model separates the objects pictured, it has to burn the marked grid nodes. However, the normal force given by expression (3) changes its signal near these grid nodes. So, the force parameters in expressions (2)-(3) have to be chosen properly to advance the T-Snake over these grid nodes. However, parameters choice remains an open problem in snake models [6].

The grid resolution controls the flexibility of T-Surfaces. A possibility to address this problem is to use a finer grid resolution. However, this increases the computational cost of the method.

To address the trade-off between model flexibility and the computational cost, we propose in [4] to get a rough approximation of the target surfaces by isosurfaces generation methods and then apply T-Surfaces.

Specifically, a *Local Scale Property* for the targets was supposed: Given an object O and a point $p \in O$, let r_p be the radius of a hyperball B_p which contains p and lies entirely inside the object. We assume that $r_p > 1$ for all $p \in O$ and take the minimum of these radius (say r_{min}).

Thus, we can use r_{min} to reduce the resolution of the image without losing the objects of interest. This idea is pictured on Figure 3. In this simple example we have a threshold which identifies the object ($T < 150$). In the Figure 3.a we have a CF triangulation whose grid resolution is 10×10, and the field thresholded.



(a) (b)

Figure 3: (a) Original image and Characteristic Function. (b) Boundary approximation.

Now, we can define a simple function, called a *Object Characteristic Function*, similar to function (1):

$$\begin{aligned} \chi(p) &= 1, \quad if \quad I(p) < T, \\ \chi(p) &= 0, \quad otherwise, \end{aligned} \quad (6)$$

where p is a node of the triangulation (grid nodes marked on Figure 3.a).

We can do a step further, shown in Figure 3.b, where we present a curve which belongs to the triangles in which the characteristic function (marked nodes) changes its value. Observe that this curve approximates the boundary we seek. These curves (or surfaces for $3D$) can be obtained by isosurface extraction methods and can be used to initialize the T-Surfaces model. These are the key ideas of our previous works [4, 5].

If we take a grid resolution coarser then r_{min}, the isosurface method might split the objects. Also, in [4] it is supposed that the object boundaries are closed and connected. These topological restrictions imply that we do not need to search inside a generated connected component.

In this paper we discard these scale and topological constraints. Thus, we should be careful to preserve the target topology. An important point is the method used to generate the piecewise linear approximations (Figure 3.b). This will be discussed next.

4 Isosurface Extraction Methods

Isosurface extraction is one of the most used techniques for the visualization of volume data sets. Depending on

the data type (time-varying or stationary) and the data size, many works have been done to improve the basic methods in this area [13]. In this paper we consider two kinds of isosurface generation methods: the marching ones and continuation ones.

In Marching Cubes, each surface-finding phase visits all cells of the volume, normally by varying coordinate values in a triple "for" loop [8]. As each cell that intersects the isosurface is found, the necessary polygon(s) to represent the portion of the isosurface within the cell is generated. There is no attempt to trace the surface into neighboring cells. Space subdivision schemes (like Octree and k-d-tree) have been used to avoid the computational cost of visiting cells that the surface do not cut [3, 13].

Once the T-Surfaces grid is a CF one, the Tetra-Cubes is a natural choice for us [2]. Like in the marching cubes, its search is linear: each cell of the volume is visited and its simplices (tetrahedron) are searched to find surfaces patches. Following marching cubes, its implementation uses auxiliary structures based on the fact that the topology of the intersections between a plane and a tetrahedron can be reduced to three basic configurations pictured on Figure 4.

Figure 4: Basic types of intersections between a plane and a simplex in 3D.

Unlike marching methods, continuation algorithms attempt to trace the surface into neighboring simplices [1]. Thus, given a transverse simplex, the algorithm searches its neighbors to continue the surface reconstruction. The key idea is to generate the combinatorial manifold (set of transverse simplices) that holds the isosurface.

The following definition will be useful. Let's suppose two simplices σ_0,σ_1, which have a common face and the vertices, $v \in \sigma_0$ and $v' \in \sigma_1$ both opposite the common face. The process of obtain v' from v is called *pivoting.* Let us present the basic continuation algorithm [1].

PL Generation Algorithm:
Find a transverse triangle σ_0*;*
$\sum = \{\sigma_0\}$*;* $V(\sigma)$ = *set of vertices of* σ*;*
while $V(\sigma) \neq 0$ *for some* $\sigma \in \sum$
get $\sigma \in \sum$ *such that* $V(\sigma) \neq 0$*;*
. *get* $v \in V(\sigma)$*;*
. *obtain* σ' *from* σ *by pivoting* v *into* v'
if σ' *is not transverse*
. *then drop* v *from* $V(\sigma)$*;*
else
. *if* $\sigma' \in \sum$ *then*
. *drop* v *from* $V(\sigma)$, v' *from* $V(\sigma')$
. *else*
. $\sum \Longleftarrow \sum + \sigma'$*;*
. $V(\sigma') \Longleftarrow$ *set of vertices of* σ'*;*
. *drop* v *from* $V(\sigma)$, v' *from* $V(\sigma')$

Differently from Tetra-Cubes, once started the generation of a component, the algorithm runs until it is completed. However, the algorithm needs a set of seed simplices to be able to generate all the components of an isosurface. This is an important point when comparing continuation and marching methods.

If we do not have guesses about seeds, every simplex should be visited. Thus, the computational complexity of both methods are the same ($O(N)$ where N is the number of simplices). However, if we know in advance that the target boundary is connected we do not need to search for components inside it. Thus, the computational cost is reduced if compared with Tetra-Cubes. That is way we use continuation methods in [4] to get the initial surfaces.

5 Isosurfaces Methods and T-Surfaces

The reparameterization of T-Surfaces gives the link between the above isosurface generation methods and T-Surfaces model. To explain this, let us take an Object Characteristic Function defined in section 3.

If we apply tetra-cubes or continuation method to this field, we get a set of piecewise linear (PL) surfaces that involve the structures of interest. From the way the PL surfaces are generated, each connected component $\widehat{M}$ so obtained has the following properties: (1) The intersection $\sigma_1 \cap \sigma_2$ of two triangles $\sigma_1, \sigma_2 \in \widehat{M}$ is empty, a common vertex or edge of both triangles; (2) An edge $\tau \in \widehat{M}$ is common to at most two triangles of $\widehat{M}$; (3) $\widehat{M}$ is locally finite, that is, any compact subset of $\Re^3$ meets only finitely many cells of $\widehat{M}$.

A polygonal surface with such property is called a Piecewise Linear Manifold (*PL Manifold*). From the reparameterization process of section 2, we can see that a T-Surface is also a PL Manifold. Thus, the isosurface extraction methods can be straightforward used to initialize T-Surfaces with the Object Characteristic Function as the initial Characteristic Function.

But, what kind of isosurface method should be used? Based on the discussion of section 4 about tetra-cubes and PL generation we can conclude that, if we do not have topological and scale restrictions (section 3), marching methods are more appropriated to initialize the T-Surfaces. In this case, it is not worthwhile to attempt to reconstruct the surface into neighboring simplices because all simplices should be visited to find surface patches.

However, for the T-Surfaces reparameterization (steps (1)-(4) of section 2) the situation is different. Now, each connected component is evolved at a time. Thus it is interesting a method which generates only the connected component being evolved, that is, the PL Generation algorithm.

But, what about the seed points? Our implementation of T-Surfaces uses a hash whose elements are specified by Keys composed by two integers: the first one indicates a simplex and the second one the connected component that cuts the simplex. There is one entry for each simplex of the triangulation. This structure is simple to implement. There is no additional costs of insertion or removal operations and the cost of verifying if a transverse tetrahedron is or not in the hash is $O(1)$.

Let us take an intersection point obtained in step (1) of the reparameterization process. If it belongs to a simplex that is not on the hash (entry NULL), then it is consulted if that simplex is a transverse one. If true, the point is stored in the hash entry and the simplex becomes a new seed to find another connected component through the PL Generation Algorithm (section 4). Each simplex so obtained becomes a new entry on the hash. Following this procedure, when the queue of projected points is empty we can be sure that all the transverse simplices are on the hash. Then, the T-Surfaces components can be obtained by traversing the hash only once.

6 Reconstruction Method

The segmentation/surface reconstruction method that we propose in this paper is based on the following steps: (1) Extract region based statistics; (2) Coarser image resolution; (3) Define the *Object Characteristic Function*; (4) PL manifold extraction by the tetra-cubes; (5) If need, increase the resolution. Return to step (3). (6) Apply T-Surfaces model; (7) User interaction if need.

It is important to highlight that T-Surfaces model can deal naturally with the self-intersections that may happen during the evolution of the surfaces obtained by step (4). This is an important advantage of T-Surfaces.

Among the surfaces extracted in step (4), there may be open surfaces which starts and end in the image frontiers, small surfaces corresponding to artifacts or noise in the background. The former is discarded by a simple automatic inspection. To discard the later, we need a set of pre-defined features (volume, surface area, etc), and corresponding lower bounds. For instance, from the grid resolution used we can set the volume lower bound as $8\,(r_{\min})^3$.

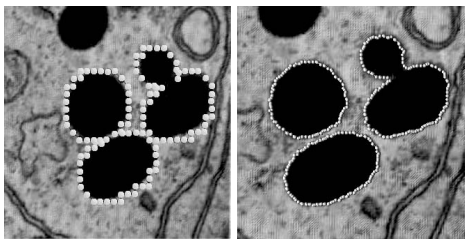
Besides, some polygonal surfaces may contain more than one object of interest (see Figure 5). Now, we can use upper bounds for the features. These upper bounds are application dependent (anatomical elements can be used).

The surfaces whose interior have volumes larger than the upper bound will be processed in a finer resolution. It is important to stress that the upper bound(s) is not an essential point for the method. It's role is only to avoid expending time computation in regions were the boundaries enclose only one object. When the grid resolution of T-Surfaces is increased we just reparameterize the model through the finer grid and evolve the corresponding T-Surfaces.

For instance, in images like in Figure 5, the *outer* scale corresponding to the separation between the objects may be finer than the scale inside the objects of interest. Hence, the coarsest resolution could not *separate* the objects. This happens for the bottom-left cells on Figure 5.a. To correct that result we increase the resolution in those regions to account for more details (Figure 5.b). Mathematical morphology operators could be used also [12].

However, for images like in Figure 5, some manual intervention may be required to split the upper-right cells.

Besides, due to inhomogeneities of the image field (supposed grey level), some objects may be splited in the step (4). Sometimes, T-Surfaces model can not merge them again.



(a) (b)

Figure 5: (a) PL manifolds for resolution 3×3. (b) Result with the highest (image) resolution.

To correct these problems, the user can manually burn some grid nodes to force merges or splits. From the entropy condition, these nodes remain burnt until the end of the process. This functionality can be implemented by selecting grid nodes with a pointer (mouse, for example), through implicitly defined surfaces (see section 7 bellow), or through a *virtual* scalpel.

7 Experimental Results

7.1 User Interaction

Firstly, we will demonstrate the potential of the user interaction procedure to force merges.

In this example, we have three spheres of intensity 50 placed in a $150 \times 150 \times 150$ image volume. The spheres were previously extracted by the proposed method with the following parameters: grid $5 \times 5 \times 5$, freezing point $= 10$, $\gamma = 0.01$, $k = 1.222$; $c = 0.750$. Every sphere has radius 30 (pixels).

In this case, the merge is forced through an implicit defined surface placed between the spheres (Figure 6.a). Grid nodes inside the surface are easily detected by its equation and then burnt. During the evolution, the 4 surfaces will merge and the final result is a connected surface (Figure 6.b).

Other possibility would be to burn manually a set of grid nodes linking the spheres. The idea in this case would be that the new set of burnt grid nodes generates a connected combinatorial manifold.

Now, let us demonstrate the manual split. Figure 6.c shows an example where the steps (1)-(6) where not able to complete the segmentation. The 3D image is composed by 2 ellipsoids of radius $30, 45$ and 60 and a sphere with radius 30, immersed in a $150 \times 150 \times 150$ noise volume.

The segmentation can be completed by user interaction based on the following steps: (a) Define a cutting plane; (b)Set to zero the grid nodes belonging to the triangles that the plane cuts and that are interior to the T-Surface; (c)Apply steps (4)-(6) above. The grid nodes set to zero become burnt nodes. Thus, the entropy condition will prevent intersections of the surfaces generated. Hence, they will not merge again.

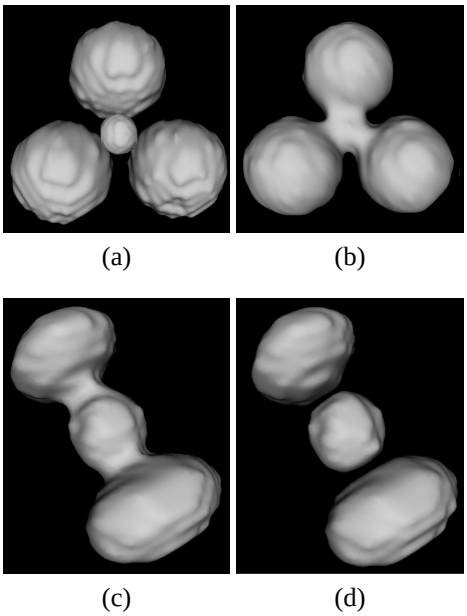


Figure 6: (a)Original objects. (b)Merge through the user interaction method. (c)Partial result. (d)Final solution after manual cut.

Figure 6.d shows the final result. The T-Surface parameters are: $c = 0.65$, $k = 1.32$ and $\gamma = 0.01$. The grid resolution is $5 \times 5 \times 5$,freezing point is set to 15 and threshold $T \in (120, 134)$ in equation (3).

7.2 Artery Segmentation

This section demonstrates the advantages of applying T-Surfaces plus isosurface methods. Firstly, we segment an artery from an $80 \times 86 \times 72$ image volume obtained from the Visible Human project. This in an interesting example because the intensity pattern inside the artery is not homogeneous.

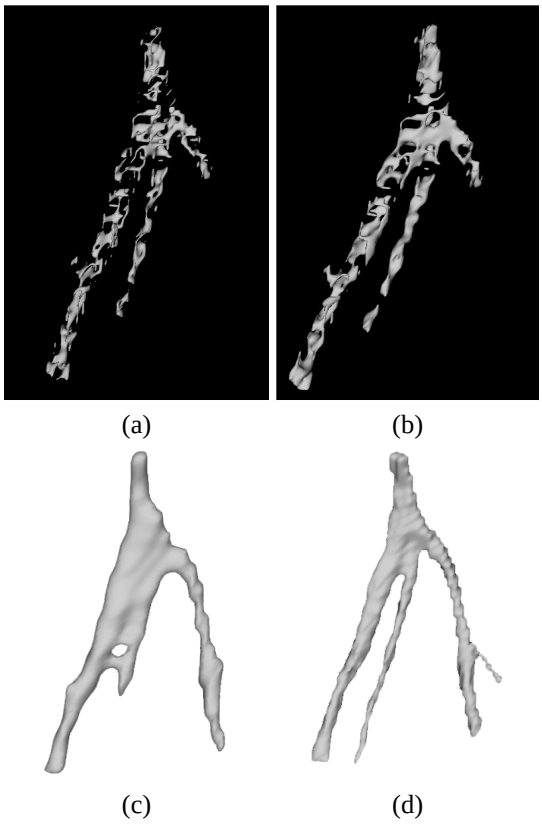


Figure 7: (a) Result of steps (1)-(4) with grid $3 \times 3 \times 3$. (b)T-Surfaces evolution (step 1). (c)Solution for initial grid. (d)Final solution for grid $1 \times 1 \times 1$.

Figure 7.a shows the result of steps (1)-(4) when using $T \in (28, 32)$ to define the object characteristic function (equation (6)). The topology extracted is too different from that one of the target. However, when applying T-Surfaces the situation gets better.

Figures 7.b shows the result after the first step of evolution. The merges among components improve the result. After 4 interactions of the T-Surfaces algorithm, the geometry extracted becomes closer to that one of the target (Figure 7.c).

However, the topology remains different. A problem in this case is that the separation between branches of the artery is coarser than the inner scale of the object in some regions. As a consequence, the used grid resolution do not

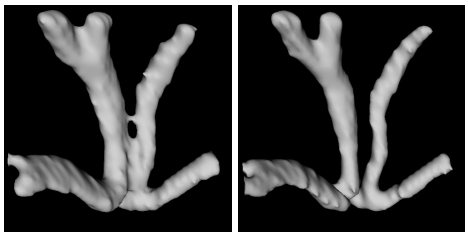
allow to complete the segmentation due to the problem discussed on section 3.

The solution is to increase the resolution. In this case, the correct result is obtained only with the finest grid resolution $(1 \times 1 \times 1)$. Figure 7.d shows the desired result obtained after 9 interactions. We also observe that new portions of the branches were reconstructed due to the increasing of flexibility given by the new grid to the T-Surfaces. We should emphasize that an advantage of the multiresolution approach is that at the lower resolution, small background artifacts become less significant relative to the object(s) of interest. Besides, it avoids the computational cost of using a finer grid resolution to get closer the target (see section 3).

The T-Surfaces parameters are $\gamma = 0.01$, $k = 1.222$; $c = 0.750$. The total number of evolution is 13. The number of triangular elements is 10104 for the highest resolution and the clock time of order of 3 minutes (see section 8).

Sometimes, even the finest resolution may not be enough to get the correct result. Figure 8.a pictures such an example. Among the proposal to address this problem (relax the threshold, mathematical morphology [12], etc), we tested the anisotropic diffusion [11]. This method enables to blur small discontinuities (improving the surface extraction) as well as to enhance edges (improving the T-Surface result). Appendix A discuss this technique.

Figure 8.b shows the correct result obtained when pre-processing the image with that method and then apply steps (1)-(6).



(a) (b)

Figure 8: (a)Example showing an incorrect result. (b)Anisotropic diffusion in a pre-processing phase improving final result.

8 Discussion

It is interesting to compare our approach to that one proposed in [9]. In this approach, a set of small spherical T-Snakes is uniformly distributed over the image. These curves progressively expand to recover the geometry of interest. The same can be done for 3D.

Our approach can be viewed as an improvement for that one. Our basic argument is that we should use the threshold to get *seeds* closer the objects of interest. Thus, we avoid expend time evolving surfaces far from the target geometry. Besides, we have observed an improvement in the performance of the segmentation process if compare with the traditional initialization of T-Surfaces (a implicit defined surface inside the object) [10].

We tested the precision for our approach when segmenting a sphere immersed on an uniform noise specified by the image intensity range 0-150. We found a mean error of 1.58 (pixels) with standard deviation of 2.49 for a $5 \times 5 \times 5$ grid resolution.

This error is due to the projection of T-Surfaces as well as the noise. Following [10], when T-Surfaces stops, we can discard the grid and evolve the model without it to avoid errors due to the projections. However, for noisy images the convergence of deformable models to the boundaries is poor due to the non-convexity of the image energy.

The Gradient Vector Flow [14] is a method that can be used to efficiently address this problem for 2D and 3D images. This method can be defined by the following equation:

$$\begin{aligned} \frac{\partial u}{\partial t} &= div\left(c\left(|\nabla f|\right)\nabla u\right) + h\left(|\nabla f|\right)\left(u - \nabla f\right), \\ u\left(x,0\right) &= \nabla f \end{aligned}$$

where f is a function of the image gradient (for example, P in equation 4). If u is the image intensity and $h = 0$ we get a version of the anisotropic diffusion equation discussed in Appendix A.

The field obtained by solving the above equation is a smoothing version of ∇f. When used as an external force for deformable models it makes the methods less sensitive to initialization. As the result of steps (1)-(6) is in general close to the target we could apply this method to push the model towards the boundary when the grid is turned-off. This is a further direction of our work.

9 Conclusions

In this paper we generalize our previous works [4, 5]. We demonstrate that steps (1)-(6) can be applied without scale and topological restrictions. Besides, we generalize the interactive procedure to change the topology of a surface.

Future directions for this work will be to generalize the user interaction method by using a scalpel and allowing the user to drag the scalpel and to apply the vector diffusion method described on section 8.

10 Appendix A

Let us take the non-linear diffusion equation as follows:

$$\frac{\partial I\left(x,y,t\right)}{\partial t} = div\left(c\left(x,y,t\right)\nabla I\right), \qquad (7)$$

where I is a gray level image. Following Perona-Malik [11], we suppose that the edge points are oriented in the x direction. Thus, equation (7) becomes:

$$\frac{\partial I(x,y,t)}{\partial t} = \frac{\partial}{\partial x}\left(c(x,y,t)\, I_x(x,y,t)\right). \qquad (8)$$

If c is a function of the image gradient: $c(x,y,t) = g(I_x(x,y,t))$, we can define $\phi(I_x) \equiv g(I_x) \cdot I_x$ and then write equation (7) as:

$$I_t = \frac{\partial I}{\partial t} = \frac{\partial}{\partial x}\left(\phi(I_x)\right) = \phi'(I_x) \cdot I_{xx}. \qquad (9)$$

We are interested in the time variation of the slope: $\frac{\partial I_x}{\partial t}$. If $c(x,y,t) > 0$ we can change the order of differentiation and with a simple algebra demonstrate that:

$$\frac{\partial I_x}{\partial t} = \frac{\partial I_t}{\partial x} = \phi'' \cdot I_{xx}^2 + \phi' \cdot I_{xxx}.$$

At edge points we have $I_{xx} = 0$ and $I_{xxx} << 0$ as these points are local maxima of the image gradient intensity. Thus, there is a neighborhood of the edge point in which the derivative $\partial I_x / \partial t$ has sign opposite to $\phi'(I_x)$. If $\phi'(I_x) > 0$ the slope of the edge point decrease in time. Otherwise it increases, that means, border becomes sharper. So, the diffusion scheme given by equation 7 allows to blur small discontinuities and to intensify the stronger ones. In this work, we have used ϕ as follows:

$$\phi = \left(\frac{\nabla I}{\left(1 + \left[\|\nabla I\| / K\right]^2\right)} \right), \qquad (10)$$

where the constant K can be determined by a histogram of the gradient magnitude.

References

[1] E. L. Allgower and K. Georg. *Numerical Continuation Methods: An Introduction*. Springer-Verlag Berlin Heidelberg, 1990.

[2] B. P. Carneiro, C. T. Silva, and A. E. Kaufman. Tetracubes: An algorithm to generate 3d isosurfaces based upon tetrahedra. In *International Symposium on Computer Graphics, Image Processing and Vision (SIBGRAPI'96)*, 1996.

[3] Yi-Jen Chiang, Cláudio T. Silva, and William J. Schroeder. Interactive out-of-core isosurface extraction. In David Ebert, Hans Hagen, and Holly Rushmeier, editors, *IEEE Visualization '98*, pages 167–174, 1998.

[4] G. Giraldi R. Silva E. Strauss, W. Jimnez and A. Oliveira. A surface extraction approach based on multi-resolution methods and t-surfaces framework. In *Proceedings of the 2002 International Conference on Imaging Science, Systems, and Technology (CISST'2002). To Appear. Also available in www.lncc.br/proj-pesq/relpesq-02.html*, 2002.

[5] G. Giraldi, E. Strauss, A. Apolinario, and A. F. Oliveira. An initialization method for deformable models. In *5th World Multiconference on Systemics, Cybernetics and Informatics (SCI 2001)*, 2001.

[6] G. A. Giraldi and A. F. Oliveira. Convexity analysis of snake models based on hamiltonian formulation. Technical report, Universidade Federal do Rio de Janeiro, Dep. Eng. Sistemas e Computação, http://www.cos.ufrj.br/relatorios/reltec99/, 1999.

[7] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1988.

[8] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. *Computer Graphics*, 21(4):163–169, July 1987.

[9] T. McInerney and D. Terzopoulos. Topologically adaptable snakes. In *Proc. Of the Fifth Int. Conf. On Computer Vision (ICCV'95), Cambridge, MA, USA*, pages 840–845, June 1995.

[10] T. McInerney and D. Terzopoulos. Topology adaptive deformable surfaces for medical image volume segmentation. *IEEE Trans. on Medical Imaging*, 18(10):840–850, October 1999.

[11] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Trans. on Patter Analusis and Mach. Intell.*, 12(7):629–639, July 1990.

[12] A. Sarti, C. Ortiz, S. Lockett, and R. Malladi. A unified geometric model for 3d confocal image analysis in cytology. In *Proc. International Symposium on Computer Graphics, Image Processing, and Vision (SIBGRAPI'98)*, pages 69–76, 1998.

[13] P. Sutton and C. Hansen. Accelerated isosurface extraction in time-varying fields. *IEEE Trans. Vis. and Comp. Graphics.*, 6(2):98–107, April-June 2000.

[14] C. Xu and J. Prince. Snakes, shapes, and gradient vector flow. *IEEE Trans. Image Proc.*, pages 359–369, March 1998.