Forschungsschwerpunkt S92

Industrial Geometry

http://www.ig.jku.at

FSP Report No. 33

# Segmenting Surfaces of Arbitrary Topology: A Two-Step Approach

J. Abhau, W. Hinterberger and O. Scherzer

September 2006

# Segmenting Surfaces of Arbitrary Topology: A Two-Step Approach

JOCHEN ABHAU[1], WALTER HINTERBERGER[2], OTMAR SCHERZER[1,2] *

1: Institut für Informatik, Technikerstraße 21a, A-6020 Innsbruck
2: IMCC Linz, Altenbergstraße 69, A-4040 Linz; current address is
dTech-Steyr GmbH, Steyrerweg 2, A-4400 Steyr
{jochen.abhau, otmar.scherzer}@uibk.ac.at, Walter@mathconsult.co.at

September 14, 2006

## Abstract

We propose a two-step approach to segment closed surfaces in 3D of arbitrary topology. First, a pre segmentation step with an active contour method is performed. This evolution process does not take into account topology adaptions. Topologically correct segmentations are derived with Kazhdan's algorithm in a second step. Kazhdan's algorithm requires information on the surface normals, which are obtained from the active contour method. We show that the two-step algorithm is computationally efficient. Moreover, we apply the algorithms for segmentation of 3D ultrasound data.

**Key Words**

image segmentation, active contour model, topology changes

## 1. Introduction

Starting with the pioneering work (WKT), *active contour models* have been extensively studied and applied in many applications such as *image segmentation*, *surface reconstruction* and *shape modeling*.

In this paper we investigate parametric active contour (AC) models for segmentation of three dimensional data which allow for topological changes. There have been several contributions on topology adaptive active contours in 2D, some of them are reviewed below. However, there has been done much less work in 3D. In 2D *discrete active contours* are curves given by a discrete number of vertices which are connected by lines. The curves are called *snakes*. In (MT2), snakes are implemented topology-adaptive using an additional Freudenthal triangulation of the image plane. With this additional simplicial structure, a re parametrization (and therefore a topology adaption) is performed cyclically after a fixed finite number of iterations of the active contour evolution. The re parametrization is performed by computing the intersections of the (discretized) snake with the triangles of the Freudenthal triangulation and the snake is locally adapted to the topology. After a re parametrization, the snake vertices are edges of the Freudenthal triangulation. A related approach is suggested in (BK) where the discretized snake is a-priori restricted to have its snaxels on edges on an underlying two-dimensional grid. If the number of snaxels is large, already in 2D, the evolution is rather time-consuming. This is due to the fact that a vertex is not allowed to move further than the next gridpoint, and therefore many iterations are required.

AC models evolve a contour over time. The evolution is driven by the propagation speed $v_S$, which is specified a-priori and usually image content specific. We discretize the AC model in time and space by a sequence of triangular meshes (see e.g. (LM)). Alternatively, one could discretize the surface by hexagons (see e.g. (Del)) or by spline surfaces (see e.g. (LC2)).

We suggest and investigate the following segmentation procedure:

Initialization: The user provides a starting point inside the object to be segmented. Automatically, a sphere inside the object to be segmented with center at the starting point is generated.

Segmentation: During the evolution of AC model the sphere expands towards the boundary of the object until boundaries or self intersection (for instance topology changes) are detected. We perform segmentation, that is the boundary detection, borrowing the ideas of *intelligent scissors* (see (SH)). Given the costs of moving between neighboring pixels (for instance the absolute value of the directional gradient between two neighboring pixels), the intelligent scissors calculate the minimal costs of moving from a user specified center to all other points. The segmentation is the closest

*Correspondence to: J. A.

contour to the center where all pixels exceed a cost threshold. Intelligent scissors are used to segment objects of different topology, for instance objects with inclusions. In our AC model we use a cost model, where the costs are evaluated along trajectories during the evalution. These costs are certainly higher than for intelligent scissors but it can be implemented more efficiently in the context of AC models. The advantage of intelligent scissors is that it can be implement more stable. Since the step length of the AC model is not restricted by the grid size of an underlying simplical structure, our algorithm is computational efficient.

Post Processing: The contour is re parametrized using Kazhdan's algorithm and topology adapted.

The paper is organized as follows: the active contour segmentation model with speed function motivated from intelligent scissors is derived in Section 2. Section 3 is concerned with the numerical implementation of the AC model. Kazhdan's algorithm for surface reconstruction is described in the Section 4. Moreover, we present some numerical experiments for segmentation of 3D clinical image data.

# 2. The Active Contour Model in 3D

Let $I \in C_0^1(\Omega, \mathbb{R})$ be a representation of intensities of an image with image domain $\Omega \subset \mathbb{R}^3$.

The continuous active contour model (see (TV)) consists in calculating a sequence of parametrized surfaces

$$(S_t : \Gamma \to \Omega)_{t \geq 0},\ \Gamma \subset \mathbb{R}^2$$

which for $t \to \infty$ approximates the segment of the object of interest. Following (CM) we use physical considerations and derive an evolutionary partial differential equation where a surface evolves like a rubber skin of a balloon which is blown up. We use the *momentum equation*

$$\frac{\partial S_t}{\partial t} = f_t \text{ on } \Gamma\ . \tag{1}$$

The forcing term is modelled as $f_t = k_t(f_t^{int} + f_t^{ext})$:

1. $f^{int}$ describes an *internal force*, which physically simulates a steady air flow into the balloon. Neglecting other forces this results in a steady expansion in normal direction to the surface

   $$f_t^{int} = \vec{n}_t\ .$$

   Here $\vec{n}_t$ denotes the outer unit normal vector to the surface $S_t$.

2. An *external force*, the surface tension, which depends on the shape of the balloon and is given by

   $$f_t^{int} = \Delta S_t.$$

   This simulates the forces on the rubber skin. For segmentation application this enforces a smooth surface.

3. The indicator function

   $$k_t : \Gamma \to \{0, 1\}\,,$$

   is used to ensure that the contour does not move across edges and corners. For modelling $k_t$ we use a *boundary* indicator $\Psi_t^{cost}$ for the image, a *smoothness* indicator of the surface $\Psi_t^{curv}$ and an indicator for *self intersections*. Let $(u, v) \in \Gamma$, then in case $\Psi_t^{cost}(u, v)$ and $\Psi_t^{curv}(u, v)$ do not exceed a certain threshold we assume that there do not occur self intersections, and we set $k_t(u, v) = 1$, otherwise we put $k_t(u, v) = 0$. The later enforces termination of the evolution at $(u, v) \in \Gamma$.

   We use the following *boundary indicator* function

   $$\Psi_t^{cost} : \Gamma \to \mathbb{R}\,,$$
   $$(u, v) \to \int_0^t \max\left\{0, \left\langle \nabla I(S_r), \frac{\partial S_r}{\partial t} \right\rangle (u, v)\right\} dr$$

   which is motivated from segmentation with *intelligent scissors* (SH). Intelligent scissors are hybrid indicator functions combining region based and gradient based segmentation. A typical examples of gradient based segmentation is Canny's edge detector (Can) and an example of a region based segmentation technique is Mumford-Shah segmentation (MS). The functional $\Psi^{cost}$ takes into account the sum of absolute values (costs) of all gradients along a trajectory. This is different to intelligent scissors where the path with minimal costs is selected, while we follow the path of evolution of the surface. We restrict our attention to the evolution along trajectories of the surface evolution since this allows 3D real time segmentation. We observed, that for our application of filtering ultrasound data (see below) this approach is more stable than pure gradient based segmentation.

   The smoothness control of the contour is achieved with

   $$\Psi_t^{curv} = \mathcal{H}_t,$$

   where $\mathcal{H}_t$ denotes the *mean curvature* of the surface $S_t$.

The contour moves outwards until internal and external forces balance, or $k_t$ becomes 0.

Discretizations of the forcing terms can be compared with established AC models in the literature, like for instance (CM) and (MT2); there the external forces are interpreted as spring forces (the forces serve as edge indicators and are related to the strength of an edge).

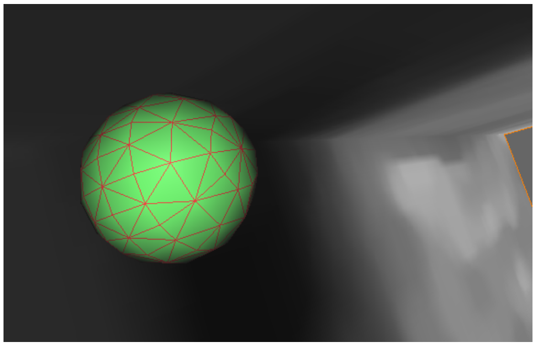

Figure 1: A small triangulated sphere is placed inside the object of interest.

# 3. Numerical implementation of the balloon model

Initially, we use a regularly triangulated sphere $S^0$ in $\mathbb{R}^3$ with 102 vertices and 200 triangles centered around a user supplied point, which is completely contained in the object of interest. Let $P$ be a vertex of the mesh of the contour $S_t$.

In the numerical implementation the derivatives in the active contour model (1) are approximated as follows:

- We use the following approximation of the time derivatives in the evolustion process:

$$\frac{\partial S_t}{\partial t}(u,v) = \frac{1}{h}(S^{n+1}(u,v) - S^n(u,v)))$$

if

$$S^n(u,v) = S_{hn}(u,v).$$

That is, we use an explicit Euler method. An explicit Euler method is considered to be inefficient in terms of numbers of iterations, however for this particular application where the cost functional is dependent on the trajectory, (semi-)implicit methods are inpractical.

- The Laplacian of the surface at $P = S^n(u,v)$ is approximated by the *umbrella* vector (see (KCVS)):

$$U(P) = \frac{1}{|\mathcal{N}_V(P)|} \sum_{Q \in \mathcal{N}_V(P)} (Q - P),$$

where $\mathcal{N}_V(P) = \{\text{neighbor vertices of } P\}$.

- The normal vector to the triangulated surface at $P$ is approximated by the normalized mean of normal vectors of the neighboring triangles:

$$\vec{n}^n(P) = \left( \left| \sum_{\Delta \in \mathcal{N}_T(P)} \vec{n}^n_\Delta \right| \right)^{-1} \sum_{\Delta \in \mathcal{N}_T(P)} \vec{n}^n_\Delta, \quad (2)$$

where $\vec{n}^n_\Delta$ is the outer normal vector of the triangle $\Delta$ and $\mathcal{N}_T(P)$ is the set of triangles with vertex $P$. The normal vector is unique up to a sign. If in addition the mesh is given an orientation at $t = 0$, the orientation is preserved as long as no topology changes occur, and the outer normal vectors are uniquely specified during the evolution.

- The boundary indicator function $\Psi^{cost}$ is approximated as follows:

$$\Psi^{n,cost} = \frac{1}{n}\left( \sum_{i=1}^{n} \max\left\{0, \left\langle \nabla_h I(S^n), \frac{\partial S^n}{\partial t} \right\rangle \right\} \right)$$

where $\nabla_h I$ is a finite difference approximation of gradient of $I$. Let $P$ and $Q$ be two vertices of the mesh, then for each point $R$ on the edge

$$e_{P,Q} = \{tP + (1-t)Q | 0 < t < 1\},$$

the mean curvature $\mathcal{H}_t$ can be approximated by (see (Sul))

$$\mathcal{H}_t(R) = |(Q-P) \times \vec{n}^{(n)}_{\Delta_1} - (Q-P) \times \vec{n}^{(n)}_{\Delta_2}|$$

Here, $\vec{n}^{(n)}_{\Delta_i}$ denote the outer unit normals on the two adjacent triangles to $e_{P,Q}$.

During the numerical evaluation of AC we store an array $A_t$, which marks all voxels inside the contour at time $t$. Using this array we can detect self intersections by checking, if the next evolution step will move a mesh point into a marked voxel.

Using the above numerical approximations the evolution of the nodes is given by

$$S^{(n+1)} = S^{(n)} + hk^{(n)}(\lambda U(S^{(n)}) + \mu \vec{n}^{(n)}) \text{ on } \Gamma.$$

In our numerical implementation, we additionally implemented a re meshing scheme (refinement and coarsening) to get regular triangles during the iteration. This ensures higher accuracy of the umbrella vector approximating the surface Laplacian and of the normal at the vertex points. Both approximations are inaccurate in case of obtuse triangles or highly varying triangle sizes at a vertex.

Refinement and coarsening is implemented as follows: If the length of an edge $e$ exceeds a certain user-defined parameter, a refinement of the two neighbor triangles is performed. Vice versa, it if the edge length is smaller that another threshold then coarsening is perormed. The two algorithms read as follows:

Algorithm: REFINEMENT

Given triangular mesh $(V, E, F)$ and lower length bound $\epsilon$.

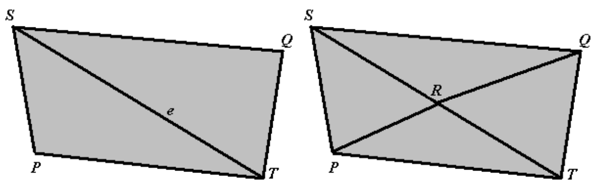


Figure 2: If the edge $e$ appears to be too long, vertex $R$ and lines $PR$, $RQ$ are inserted to subdivide the triangles.

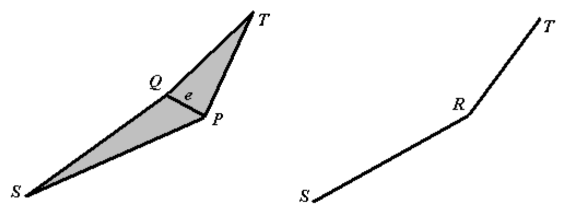


Figure 3: If two points are closed, they are melted.

WHILE $\{e \in E \mid \text{length}(e) > \epsilon\} \neq \emptyset$
  $S, T \leftarrow$ endpoints$(e)$
  $P, Q \leftarrow$ opposite vertices$(e)$
  $R \leftarrow \frac{1}{2}(S+T)$
  $V \leftarrow V \cup \{R\}$
  $E \leftarrow E \backslash \{(S,T)\}$
  $E \leftarrow E \cup \{(P,R),(R,Q),(S,R),(R,T)\}$
  $F \leftarrow F \backslash \{(P,S,T),(Q,T,S)\}$
  $F \leftarrow F \cup \{(P,R,S),(Q,S,R),(T,R,P),(Q,R,T)\}$
END WHILE

Algorithm COARSENING

Given a mesh $(V, E, F)$ and upper distance bound $\epsilon$.

WHILE $\{e \in E \mid \text{length}(e) < \epsilon\} \neq \emptyset$
  $P, Q \leftarrow$ endpoints$(e)$
  $S, T \leftarrow$ opposite vertices$(e)$
  $R \leftarrow \frac{1}{2}(P+Q)$
  $V \leftarrow V \backslash \{P, Q\}$
  $V \leftarrow V \cup \{R\}$
  $E \leftarrow E \backslash \{(S,P),(S,Q),(P,T),(Q,T),(P,Q)\}$
  $E \leftarrow E \cup \{(S,R),(R,T)\}$
  $F \leftarrow F \backslash \{(S,P,Q),(T,Q,P)\}$
  replace coordinates $P$ and $Q$ by $R$ in each triangle
END WHILE

Some results of the active contour model are presented in Figure 4.

# 4. Third step: Topology changes

The AC contour model described in the previous section did not allow for topology changes. In fact whenever topology changes are predicted, the iteration is locally terminated. Therefore the result of the AC model the balloon evolution is homeomorphic to a sphere. An example for the result of the AC model is shown in figure 4.

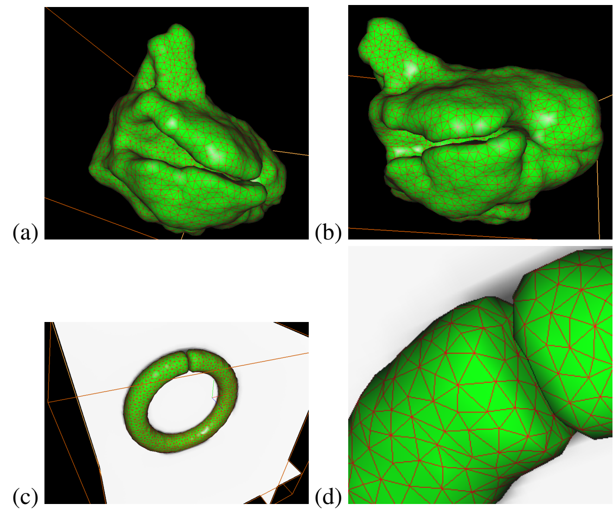


Figure 4: In (a) and (b), the end of the balloon evolution is shown for the cyst example. The evolution result for the torus can be seen in (c) and (d). All these four meshes are still homeomorphic to a sphere.

In the following we present an algorithm that allows to reconstruct the correct topology of surfaces of genus $g > 0$ from the output of the AC model.

Let $P_1, \ldots, P_{N_{term}}$ denote the mesh points of the stationary surface of the AC model. Moreover, we denote by $\vec{n}_1, \ldots, \vec{n}_{N_{term}}$ the weighted normals at the mesh points, computed with formula (2). In practice the informations on the mesh points and their weighted normals is sufficient to uniquely determine the mesh surface. However, there can occur situations when the mesh is not uniquely determined, which in a mathematical notation means that for each mesh point $P_i$ there exists a scalar $\mu_i$ such that

$$\sum_{\Delta \in \mathcal{N}_T^1(P_i)} \vec{n}_\Delta = \mu_i \sum_{\Delta \in \mathcal{N}_T^2(P_i)} \vec{n}_\Delta . \quad (3)$$

The set of configurations $\{P_1, \ldots, P_{N_{term}}\}$ admitting such a relation is a finite intersection of a finite union of codimension-2-subspaces of $\mathbb{R}^{3N_{term}}$ and hence has Lebesgue measure zero. If the mesh after the balloon evolution encloses a flat region, the upper mentioned ambiguities do not influence the shape of the mesh. Otherwise we can think of the vertices as a random sample of surface points, such that the upper relation is not satisfied. These considerations motivate to use as mesh information just points and normals.

Given the mesh after the balloon evolution, to approximate the surface in a topological correct form, we distinguish beetween two types of vertices:

1. The flow of a vertex has been stopped during the AC evolution since a self intersection has been detected. Such a vertex must be discarded as an interior point of the object.

2. In all the other cases, a vertex is considered to be part of the boundary of the object.

The following procedure shows how to remove vertices of type 1 which belong to the interior of the object. For this purpose the mesh is first opened and afterwards remeshed.

# Opening of mesh for topology adaption

A procedure for opening and updating of the normal is as follows:

- [1] Mark all triangles which contain a vertex of type 1.

- [2] For all vertices $P$ contained in a marked triangle, update the surface normals: Set $\mathcal{N}_T(P_i)) = \{$triangles incident to $P_i$, not marked$\}$, and

$$n_{P_i} = \left( \left| \sum_{\Delta \in \mathcal{N}_T(P_i)} \vec{n}_\Delta \right| \right)^{-1} \sum_{\Delta \in \mathcal{N}_T(P_i)} \vec{n}_\Delta$$

- [3] Delete all marked triangles and vertices of type 1 from the mesh.

We obtain a meshed surface with boundaries and update the normals of the vertices which had a neighboring triangle which has been cut out.

The result of the opening procedure is shown in figure 5.

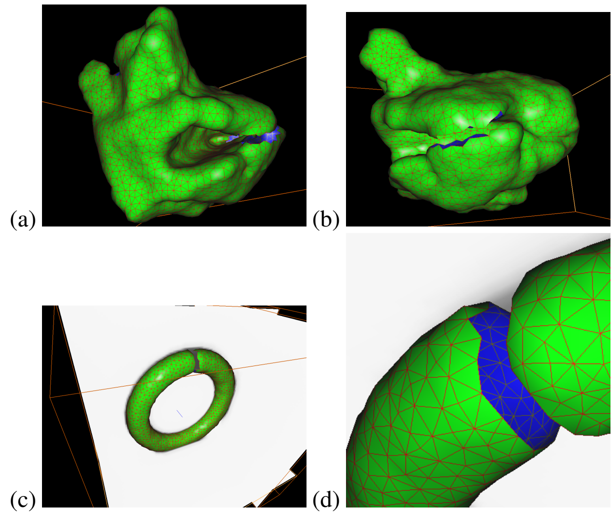


Figure 5: Opening Algorithm: vertices and triangles at places of self-intersection are discarded. The remaining vertices and triangles are remeshed.

# The remeshing algorithm

We discard all the edges of the mesh after the opening procedure, such that only surface points and their normals remain.

We compute the topology adapted mesh with vertices and outer unit normals following the mesh reconstruction method described in (Kaz). To be self-contained, we shortly summarize this algorithm, the features and adapt it to our purposes.

Let $P_1, \ldots, P_N$ denote the remaining points on the surface after the opening procedure, and $\vec{n}_1, \ldots, \vec{n}_N$ the corresponding outer unit normals. Moreover, let $M \subset \mathbb{R}^3$ denote the object of interest. We approximate the indicator (or characteristic) function $1_M$ of the set $M$ by computing the fourier series expansion of $1_M$. For this purpose, let

$$\hat{1}_M(k) = \int_M e^{-i\langle k,x\rangle} dx$$

be the k-th Fourier coefficient of $1_M$, $k \in \mathbb{Z}^3$.
Moreover, for $x \in \mathbb{R}^3$, let

$$F_k(x) = \begin{pmatrix} \frac{ik_1}{|k|^2} e^{-i\langle k,x\rangle} \\ \frac{ik_2}{|k|^2} e^{-i\langle k,x\rangle} \\ \frac{ik_3}{|k|^2} e^{-i\langle k,x\rangle} \end{pmatrix}.$$

This function satisfies

$$\mathrm{div} F_k(x) = e^{-i\langle k,x\rangle}, k \in \mathbb{Z}.$$

Therefore, by Stokes' Theorem, we have

$$\int_M e^{-i\langle k,x\rangle} dx = \int_{\partial M} \langle F_k(p), \vec{n}(p)\rangle \, dp.$$

Therefore, by using Monte-Carlo-Approximation we find

$$\int_{\partial M} \langle F_k(p), n(p)\rangle \, dp \approx \frac{c(M)}{N} \sum_{i=1}^{N} \langle F_k(P_i), \vec{n}_i\rangle .$$

Here $c(M)$ denotes the surface area of $M$, a constant which actually need not be computed for what follows.

To summarize, we have shown that for every $x \in \mathbb{R}^3$

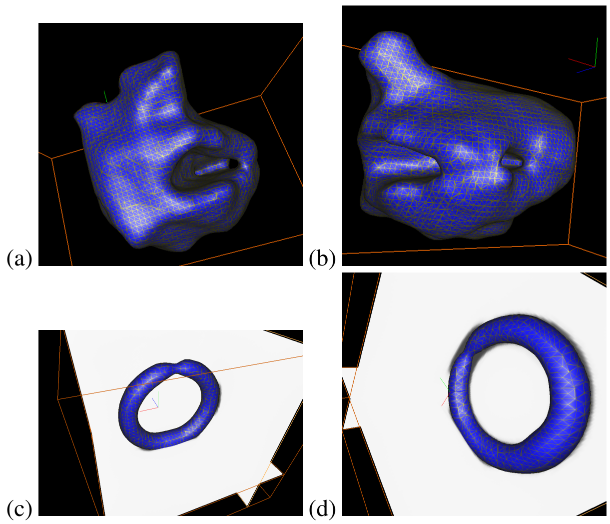


Figure 6: After remeshing, the mesh shown in (a) and (b) is obtained, (c) and (d) show the result for the torus. Note that both surfaces have genus 1, hence a topological change has taken place.

$$
\begin{aligned}
1_M(x) &= \sum_{k\in\mathbb{Z}^3} \hat{1}_M(k) e^{i\langle k,x\rangle} \\
&\approx \text{const} \underbrace{\sum_{k\in\mathbb{Z}^3} \sum_{i=1}^{N} \langle F_k(P_i), \vec{n}_i\rangle}_{=:\tilde{1}_M(x)} .
\end{aligned}
$$

Following (Kaz), an approximation of the characteristic function $1_M$ can be calculated using the mean value $\mu = E(\tilde{1}_M)$ and by appropriate thresholding

$$
1_M(x) \approx \begin{cases} 1, & \tilde{1}_M(x) \geq \mu \\ 0, & \tilde{1}_M(x) < \mu. \end{cases}
$$

Once the indicator function of the object has been computed, the surface mesh can be computed with the marching cubes algorithm, see (LC1).

Figure 6 illustrates the algorithm for topology adaption and the result obtained after applying the marching cube algorithm.

# 5. Results and Discussion

We tested our algorithm with artificial and real 3D voxel data. Wo have considered two examples showing a cyst in a kidney and an artificial torus shown in Figures 3 and 4-4. During biopsy a needle has been sticked into the cyst. Since the needle penetrates the object, mathematically we might say that this object has genus 1. The sequence of pictures shows

- the original voxel image,

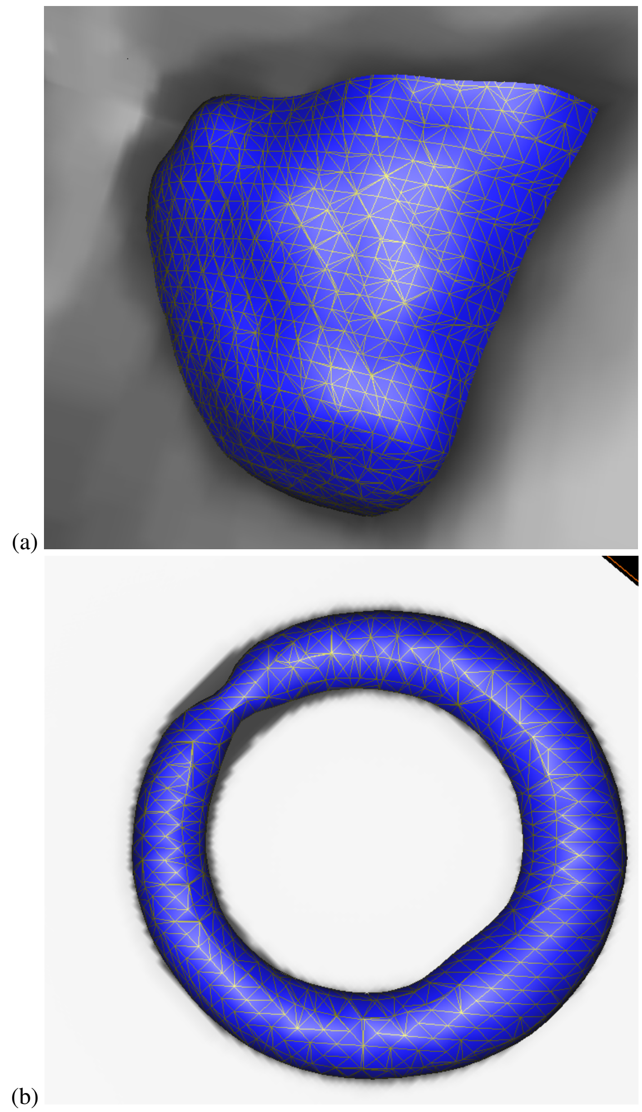


Figure 7: (a) Zoom into the cyst mesh. (b) The segmentation of the torus is equally good in regions where no topological change occured.

- the voxel image filtered (which is applied before segmentation to stabilize the algorithm),
- the small sphere is placed inside the object for initialization (is the single user interaction),
- the final result of the AC evolution (the surface is still homeomorphic to a sphere),
- the sphere points which were stopped because of threatening self-intersections are discarded,
- the surface mesh after reconstruction by normals and the marching cubes algorithm,
- the final segmentation result.

We computed the two examples on a Pentium 4 Computer with 2 GB RAM and 3,5 GHz CPU.

While Step 2 took approximately 8 seconds for the cyst and 5 seconds for the torus, the remeshing step 3 was done in ca. 0.1 seconds in both cases.

The drawback of our method can be seen in figure 4(b). In regions of topological change, the segmentation is a bit imprecise. This is a consequence of the curvature control. At the end of the evolution the deformed sphere fills the object up to small pieces at the boundary. Discarding intersection triangles, the normals of the resting triangles on the boundary of the opened mesh point slightly backwards, away from the boundary. Therefore, the torus is a bit thinner in this region. One possible solution to this problem might be a second usage of the first step, now blowing up the present (topologically correct) surface.

# 6. Summary and Future Work

In this paper we have proposed a new approach for active contours which allows for topology adaptations. The algorithms consists of two steps, and AC model which supresses topology changes and a topology adaptions using Kazhdans algorithm and a marching cube algorithm. The AC model is time efficient since it evolves a surface withour using an underlying grid structure. In practical applications we applies the segmentation algorithm to filtered data. The topic of adequate filtering has not been addressed in this paper but is important to achieve relyable results. As a future work we intend to study adaptions of the contour where the topological adaptation has taken place.

# Acknowledgments

The work of J.A. and O.S. is supported by the Austrian Science Foundation (FWF) Projects Y-123INF; O.S. is additionally supported by projects FSP 9203-N12 and FSP 9207-N12. Part of the work of W.H. has been supported by the Austrian Ministry for Economy and Labour and by the Government of Upper Austria within the framework "Industrial Competence Centers".

We thank GE - Medical Systems, Kretztechnik, for providing the voxel image of the cyst and Tobias Riser for his QT-Viewer, with which the example pictures of the cyst and the torus were visualized. The QT viewer has been developed within the TWF project Parallelisierte Datenauswertung am HPC.

# References

[BK] S. Bischoff and L. Kobbelt. *Snakes with topology control*. The Visual Computer. Vol 20, pp 217-228, 2004.

[Can] J. Canny. *A Computational Approach to Edge Detection.* IEEE Transactions on Pattern Analysis and Machine Intelligence. Vol 8, No 6, November 1986.

[CM] Y. Chen and G. Medioni. *Description of Complex Objects from Multiple Range Images Using an Inflating Balloon Model.* Computer Vision and Image Understanding. Vol 61, No 3, pp 325-334, 1995.

[Del] H. Delingette. *Adaptive and Deformable models based on Simplex Meshes.* IEEE Workshop of Non-Rigid and Articulated Objects. Austin, Texas, November 1994.

[Hin] W. Hinterberger. *Variational Methods and PDE's for Image(Feature-)Reconstruction*, 2003. PhD thesis, Universität Innsbruck, Austria.

[Kaz] M. Kazhdan. *Reconstruction of Solid Models from Oriented Point Sets.* Symposium on Geometry Processing. pp 73-82, 2005.

[KCVS] L. Kobbelt, S. Campagna, J. Vorsatz and H. P. Seidel. *Interactive multiresolution modeling on arbitrary meshes.* Computer Graphics (SIGGRAPH 98 Proceedings). pp 105-114, 1998.

[LM] J. O. Lachaud and A. Montanvert *Deformable Meshes with Automated Topology Changes for Coarse-to-fine 3D Surface Extraction.* http://dept-info.labri.fr/ lachaud/Articles/1999-mia/revueMIA.html

[LC1] W. E. Lorensen and H. E. Cline. *Marching cubes: a high resolution 3D surface construction algorithm.* Computer Graphics (SIGGRAPH 87 Proceedings). Vol 21,Nr. 4, pp 163-170, 1987.

[LC2] F. Leitner, P. Cinquin. *Complex topology 3D-objects segmentation.* SPIE Conference on Model Based Vision Development and Tools, Boston, MA (USA) pp. 16-26, 1992.

[MT1] T. McInerney and D. Terzopoulos. *A Finite Element Model for 3D Shape Reconstruction and Nonrigid Motion Tracking.* Proceedings of the International Conference on Computer Vision (ICCV). Berlin, Germany, pp 518-523, 1993.

[MT2] T. McInerney and D. Terzopoulos. *T-Snakes: Topology Adaptive Snakes.* Medical Image Analysis, 4(2). pp 73-91, June, 2000.

[MS] D. Mumford and J. Shah. *Optimal approximations by piecewise smooth functions and associated variational problems.* Commun. Pure Appl. Math. XLII. pp 527-685, June, 1989.

[PS] Combinatorial Optimization: Algorithms and Complexity C.H. Papadimitriou, K. Steiglitz. *Combinatorial Optimization: Algorithms and Complexity.* Prentice Hall, 1982.

[SH] D. Stalling and H.C. Hege. *Intelligent Scissors for Medical Image Segmentation.* Digitale Bildverarbeitung fr die Medizin, 4(2). Freiburg, Germany, pp 32-36, 1996.

[Sul] J.M. Sullivan. *Curvature Measures for Discrete Surfaces*. http://torus.math.uiuc.edu/jms/Papers/, 2002.

[TV] D. Terzopoulos and M. Vasilescu. *Sampling and Reconstruction with Adaptive Meshes.* Proceedings of the Conference on Computer Vision and Pattern Recognition. pp 70-75, 1991.

[WKT] A. Witkin, M. Kass and D. Terzopoulos. *Snakes: Active contour models*. International Journal of Computer Vision. Vol 1, Nr. 4, pp 321-331, 1987.