

Forschungsschwerpunkt S92

Industrial Geometry

<http://www.industrial-geometry.at>



FSP Report No. 72

A Combinatorial Method for Topology Adaptations in 3D Deformable Models

J. Abhau and O. Scherzer

July 2008

FWF

Der Wissenschaftsfonds.



A Combinatorial Method for Topology Adaptations in 3D Deformable Models

Jochen Abhau¹
Otmar Scherzer^{1,2}

¹Department of Mathematics
University of Innsbruck
Technikerstr. 21a
6020 Innsbruck, Austria

²Radon Institute of Computational
and Applied Mathematics
Altenberger Str. 69
4040 Linz, Austria

July 21, 2008

Abstract

In this paper we propose an efficient algorithm for topology adaptation of evolving surface meshes in 3D. This system has two novel features: First, a spatial hashing technique is used to detect self-colliding triangles of the evolving mesh. Secondly, for the topology adaptation itself, we use formulas which are derived from homology. In view of this the advantages of our algorithm are that it does not require global mesh reparameterizations and the topology adaptation can be performed in a stable way via a rather coarse mesh.

We apply our algorithm to segmentation of three dimensional synthetic and ultrasound data.

Keywords: deformable model, triangular mesh, topology adaptation, segmentation, homology

1 Introduction

Since the pioneering work [23] deformable contours have been used successfully in various areas of applications, such as image processing, medical imaging, cloth modeling and game development.

It is common to differ between *explicit* and *implicit* deformable contours – that are parametric and level set models. The later have been introduced in [17] and further achievements (especially on the theoretical side) have been made in [3]. One advantage of implicit methods is that topology adaptations are handled automatically during the evolution process. Nevertheless, for practical applications explicit models are often preferred since they can be implemented more efficiently. The reason for this is that an evolving parametric surface is a two dimensional manifold, while the evolving level set is three dimensional.

In this work we develop an explicit method which allows for topological adaptive segmentation. Such methods have already been subject to extensive

research. To our knowledge, explicit contour models with topological adaptiveness have been considered first in [12] and [20]. There, deformable contours are represented as *tensorial spline products* [12] and sets of *dynamic particles* [20], respectively. The basic *snake model*, introduced in [14], has been complemented with topology adaptivity in [15] utilizing a supplemental Freudenthal triangulation. With this additional simplicial structure a re-parametrization is performed periodically after a fixed finite number of iterations of the snake evolution. In each Freudenthal triangle mesh self-collisions are checked for and topology is adapted where collisions have been detected. Similar ideas have been presented in [5]. Basically, the algorithms of [15, 5] consist of three steps. First a grid is aligned on the two dimensional image domain containing the object to be segmented. Secondly, intersections of the contour edges with the grid edges are computed and stored as grid vertices. From the grid vertices new contour edges are computed, which are edges connecting the grid vertices. Thirdly, self-intersections of the re-parameterized contour are detected and the topology of the contour is adapted in all simplices composed of the grid edges. In [2] it is suggested to evolve a polygonal contour where the vertices are restricted to lie on a supplemental rectangular grid of the image domain. An advantage of this approach is that no re-parameterizations have to be performed and topology adaptations are along the lines of [15]. On the other hand, if the underlying grid is fine, small time stepping is required and thus the evolution becomes numerically expensive. In [7] a mesh transformation algorithm is proposed which discards overlapping mesh parts and performs a re-triangulation afterwards. This method only works if the mesh satisfies geometrical properties, which are controlled by a distance field evolution. According to [22, Sec. 4] distance field computations are numerically very expensive. A speed up of the algorithm of [7] has been obtained in [9, 10, 8] by relaxing the (global) geometrical constraints by local conditions.

Our proposed algorithm is designed for segmentation of multiple connected surfaces in **3D** and consists of the following steps:

Concept 1 (Segmentation concept).

1. *An active contour model is used to evolve a mesh until self-intersections are detected. Detection is performed by a spatial hashing algorithm described in Section 2. This algorithm is motivated from [21].*
2. *Neighboring vertices of colliding parts of the mesh M are removed to get an opened mesh M_b whose boundary consists of a number of simple closed polygons. The algorithm is described in Section 4.*
3. *The opened mesh M_b is completed by a handle, which consists of a mesh K , that is topologically equivalent (i.e. homeomorphic) to a sphere with holes. The completed mesh M_c consists of the union of M_b and K . The topology adaption is illustrated in Figures 1. To make the completion algorithm efficient we use a precomputed database of topologically equivalent meshes for the handles. The database is structured by the number of connected polygons and the numbers of faces, respectively.*
4. *Afterwards the active contour evolution is further continued.*

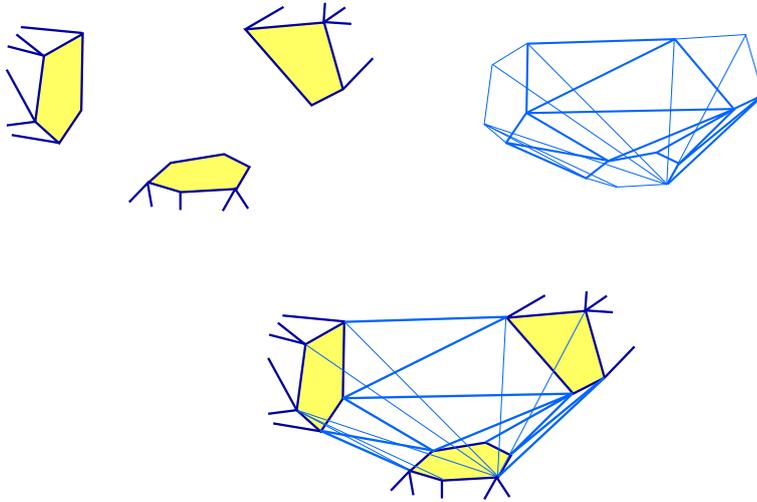


Figure 1: *Top Left*: Opened mesh M_b . *Top Right*: Handle K . *Bottom*: Closed mesh M_c .

In this paper we focus on algorithms for detection of self intersections and topology adaptations. Active contour models are not discussed here further, we refer to [4] for a standard reference on this topic.

2 (Self-)Collision Detection

For collision detection of the evolving surfaces we use a spatial *hashing algorithm* which is motivated from [21]. However, in comparison, our proposed algorithm has several additional features. For instance, for implementation it does not require complicated data structures and the running time is linear with respect to the number of vertices and the chosen hash table size.

We are given a triangular mesh $M = (V, E, F)$ in a bounded region $\Omega \subset \mathbb{R}^3$. The proposed hashing algorithm consists of the following two steps:

Concept 2 (Collision Detection Algorithm by Spatial Hashing).

1. For all mesh vertices v , hash functions $h_1(v)$ and $h_2(v)$ are computed by a subdivision of Ω into axes aligned bounding boxes.
2. Let $j = 1, 2$. For all hash values i let $V_i^j = \{\text{vertices with } h_j(v) = i\}$, the sets of vertices with hash value i . In this step it is checked whether triangles containing vertices of V_i^1 or V_i^2 intersect.

In the following we present some details of the spatial hashing algorithm. In the first step, for a definition of the hash functions, we use large prime numbers p_i , $i = 1, 2, 3$, choose a hash table size `htblSize`. Moreover, we denote by the real parameter l the size of the axes aligned bounding boxes (see Figure 2). By $\lceil a \rceil$ we denote the smallest integer greater than a , and $\lfloor a \rfloor$ denotes the greatest integer smaller than a . The two hash functions $h_j : V \rightarrow \{0, \dots, \text{htblSize}\}$,

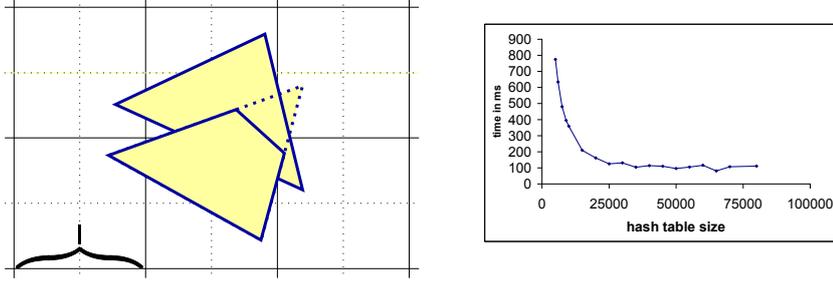


Figure 2: *Left:* Two systems of axes aligned bounding boxes of size l are shown. The intersection of the two triangles can only be detected by the boxes in dashed lines *Right:* Running times of the collision detection algorithm for triangular meshes of 10k vertices with different hash table sizes are given

$j = 1, 2$, are given by

$$\begin{aligned} h_1(v_x, v_y, v_z) &= \lfloor v_x/l \rfloor p_1 + \lfloor v_y/l \rfloor p_2 + \lfloor v_z/l \rfloor p_3 \bmod (\text{htblSize} + 1), \\ h_2(v_x, v_y, v_z) &= \lceil v_x/l \rceil p_1 + \lceil v_y/l \rceil p_2 + \lceil v_z/l \rceil p_3 \bmod (\text{htblSize} + 1). \end{aligned}$$

Note, that for each two vertices $v_1, v_2 \in V$ with $\|v_1 - v_2\| < l/2$, at least one of the following two equations holds:

$$h_1(v_1) = h_1(v_2), \quad h_2(v_1) = h_2(v_2).$$

We use vertices with euclidean distance smaller than $l/2$ as indicators for intersecting triangles. For an optimal choice of the box size l we use the following theorem. For a proof of the theorem we refer to the Appendix.

Theorem 3. *Assume that the length of every edge of a mesh $M = (V, E, F)$ is bounded by s . Moreover, we assume that the triangles $T = (T_1, T_2, T_3)$ and $S = (S_1, S_2, S_3)$ of the mesh intersect. Then there exist i, j such that*

$$\|T_i - S_j\| < \sqrt{\frac{2}{3}}s.$$

Here and in the following we identify the triangle with the triple of edge points.

According to the theorem we choose $l > 2\sqrt{2/3}s$, because this choice guarantees, that at least two vertices of intersecting mesh triangles are mapped to the same hash key, and thus in the sequel an intersection test is performed. For our applications we have chosen the hash table size to be twice the number of mesh vertices. This choice is based on numerical experiments with meshes of approximately 10k vertices and 20k triangles, respectively. The hash table sizes have been varied (see Figure 2). From this table we see that the running time for the collision detection is monotonously decreasing in hash table sizes smaller than twice the number of mesh vertices, and remains nearly constant for greater hash table sizes. Therefore, from a point of memory usage the suggested choice is optimal.

In the second step we iterate through the hash keys and check for each pair of non-neighboring vertices with the same hash key if they are contained in triangles which intersect. This is done with a fast triangle-triangle intersection test, see [16].

3 The Handle Database

In this section we show how to generate a data base of simplices which allows for completion of an opened mesh M_b to a closed mesh M_c . A set of such simplices will be called a *handle*. Here we make use of concepts from topology (see e.g. [6, 13, 19] for background material). We are mainly concerned with combinatorics and use simplicial complexes in a coordinate free abstract sense as in [19, p.141].

To start with, we make a basic definition of handles:

Definition 4. We call a simplicial complex $C = (C_0, C_1, C_2)$ a cap, if there exist $m \in \mathbb{N}$, $a_0, \dots, a_m \in \mathbb{Z}$, $a_0 < \dots < a_m$ such that:

$$\begin{aligned} C_0 &= \{a_0, \dots, a_m\}, \\ C_1 &= \{\{a_0, a_1\}, \dots, \{a_0, a_m\}, \{a_1, a_2\}, \{a_2, a_3\}, \dots, \{a_{m-1}, a_m\}, \{a_m, a_1\}\}, \\ C_2 &= \{\{a_0, a_1, a_2\}, \{a_0, a_2, a_3\}, \dots, \{a_0, a_{m-1}, a_m\}, \{a_0, a_m, a_1\}\}. \end{aligned}$$

The orientation of the complex is given by (a_0, a_1, a_2) , (a_0, a_2, a_3) , \dots , (a_0, a_{m-1}, a_m) , (a_0, a_m, a_1) . a_0 is called the vertex center.

Let C^1, \dots, C^k be caps. A simplicial complex $K = (K_0, K_1, K_2)$ is called handle if the geometric realization (see [19, p.142]) $|\bigcup_{i=1}^k C^i \cup K|$ of

$$\bigcup_{i=1}^k C^i \cup K := \left(\bigcup_{i=1}^k C_0^i \cup K_0, \bigcup_{i=1}^k C_1^i \cup K_1, \bigcup_{i=1}^k C_2^i \cup K_2 \right)$$

is homeomorphic to a 2-sphere (in signs $|\bigcup_{i=1}^k C^i \cup K| \approx \mathbb{S}^2$) and for all $j \in \{1, \dots, k\}$, the inclusion

$$C^j \hookrightarrow \bigcup_{i=1}^k C^i \cup K$$

is orientation preserving. We recall that homeomorphisms are defined to respect topological properties. A cap is visualized in Figure 3.

The following theorem characterizes topological properties of a handle and states that the a mesh M after opening at k locations and closing by a handle (this is the mesh M_c) constitutes a surface which has $k - 1$ tunnels more than M , that is the genus is increased by $k - 1$.

Theorem 5. Let $M = (V, E, F)$ be a simplicial complex. Furthermore, let M_b be the simplicial subcomplex ([19, p.144]) of M obtained by removing k connected components from M such that the geometric realization of M_b is a surface with boundary consisting of k simple closed polygons (see also Figure 1, top left). Let C^1, \dots, C^k be caps such that its boundaries consist of the boundary polygons of M_b (see also Figure 3). Moreover, let K be a handle for C^1, \dots, C^k , so that $|\bigcup_{i=1}^k C^i \cup K| \approx \mathbb{S}^2$. Then

$$|M_b \cup K| \approx T_{k-1+g(M)}, \quad (1)$$

where $T_{k-1+g(M)}$ is the closed, orientable surface of genus $k - 1 + g(M)$ and $g(M)$ is the genus of M .

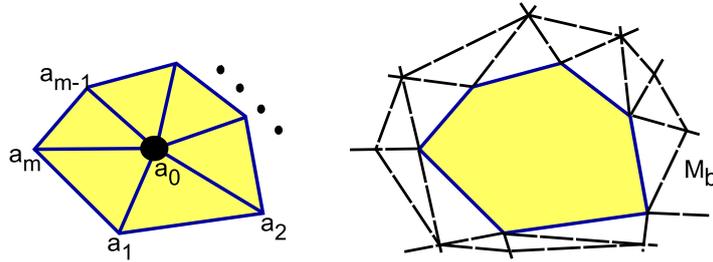


Figure 3: Left: A cap. Right: A simple closed boundary polygon of M_b .

This theorem can be proven by standard methods of algebraic topology.

Topological equivalent 2-spheres can be characterized by simplicial homology (see e.g. [19, p.144] for background on this topic).

Theorem 6. *Let $M = (V, E, F)$ be a simplicial complex such that every edge $e \in E$ is a face of some $f \in F$. If the homology conditions*

- $H_0(M) \cong \mathbb{Z}$
- $H_1(M) = 0$
- $H_2(M) \cong \mathbb{Z}$, generated by $\sum_{f \in F} \epsilon_f f$ with $\epsilon_f \in \{-1, 1\}$

hold, then the geometric realization of M is homeomorphic to \mathbb{S}^2 .

This theorem is similar to the well-known *Whitehead Theorem* (see [6, p.346]) which states that spaces with isomorphic homotopy groups are homotopy equivalent. Theorem 6 can be proven again by standard methods of algebraic topology.

Based on these basic definitions and Theorem 6 we can present an algorithm for computation of handles which contain given simple closed polygons as boundary polygons. With each simple closed polygon we can associate a cap by adding a center vertex and connecting the center with the vertices by edges. Therefore, we concentrate now on computation of handles given disjoint caps.

Let us assume that we have given k disjoint caps C^i , $i = 1, \dots, k$ with numbers of vertices m_i , $i = 1, \dots, k$, respectively.

We use the notation $m_s = \sum_{i=1}^k m_i$ and denote by v_c, e_c, f_c the numbers of vertices, edges, and faces of the mesh $\bigcup_{i=1}^k C^i \cup K$. Let us assume that $\bigcup_{i=1}^k C^i \cup K$ is homeomorphic to a sphere – that is the case if K is a handle. Then, from the Euler formula (see [19, p.146]) we know that $v_c - e_c + f_c = 2$. Moreover, by induction on the number of triangles, we can show that $2e_c = 3f_c$, and therefore $f_c = 2v_c - 4$. If k caps are connected by a handle, the number of vertices of the arising sphere is $v_c = m_s + k$, and therefore

$$f_c = 2m_s + 2k - 4.$$

Since the caps C^i contain m_s faces altogether, $m_s + 2k - 4$ faces have to be added to the faces of the set $\{C^i : i = 1, \dots, k\}$ to obtain a sphere. We differ between two different kinds of faces to be added (see Figure 4):

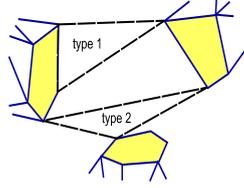


Figure 4: Different types of faces.

- When a face to be added has two vertices in common with some $C^i, i = 1, \dots, k$, then it is called of Type 1.
- Else it is called of Type 2. That is the case if the face has at most one vertex in common with every C^i .

There exist m_s faces of Type 1 and $2k - 4$ faces of Type 2.

Based on these considerations we are able to generate the handle database, which associates tuples consisting of the number of connected polygons k and the numbers of vertices $m_i, i = 1, \dots, k$, a set of handles, respectively. Without loss of generality we always assume in the sequel that $m_{i+1} \geq m_i, i = 1, \dots, k - 1$. For given k and $m_i, i = 1, \dots, k$, a handle of the database is determined as follows:

1. For each boundary edge of a cap choose a vertex in a different cap. The face determined by the edge and the chosen vertex is one of the m_s triangles of Type 1.
2. Locate two edges which share a common vertex such that all three vertices are contained in different caps. This determines the $2k - 4$ triangles of Type 2.
3. Check if the simplicial complex made of the caps and the added faces is a sphere, that is, it satisfies the conditions of Theorem 6. Computationally, one can check the homology criterion using the PARI software [18].

Given caps $C^i, i = 1, \dots, k$ with m_i vertices, respectively, it is useful for our purposes to associate a sequential enumeration to the vertices. To this end we use the notation $\mu_l = \sum_{i=1}^l m_i$. Vertices between $\mu_{l-1} + 1$ and μ_l (where we set $\mu_0 := 0$) correspond to the vertices in the cap C^l .

For $i \in \{1, \dots, \mu_k\}$, we set

$$i \oplus 1 = \begin{cases} \mu_{j-1} + 1 & \text{if } i = \mu_j \text{ for some } j \in \{1, \dots, k\} \\ i + 1 & \text{otherwise.} \end{cases}$$

Therefore, $i \oplus 1$ is the subsequent vertex of i in the cap C^i .

Example 7. *In this example we calculate the number of different elements of the handle database for some test cases of small k .*

$k = 2$: *Because we have $2k - 4 = 0$, only faces of Type 1 occur. For a function*

$$f : \{1, \dots, m_1\} \rightarrow \{m_1 + 1, \dots, m_1 + m_2\},$$

$k = 3:$					
m_1	m_2	m_3	handles	generators	
3	3	4	72	1	
3	4	5	120	2	
4	4	6	192	2	

$k = 4:$					
m_1	m_2	m_3	m_4	handles	generators
3	3	4	4	576	1

Table 1: The numbers of possible handles in the database, and the number of generators taking into account group actions.

which we assume to be monotonously decreasing and surjective we define

$$g : \{m_1 + 1, \dots, m_1 + m_2\} \rightarrow \{1, \dots, m_1\} .$$

$$j \mapsto \max\{i : f(i) = j\}$$

Note that f maps vertices of the first cap onto vertices of the second and g is a right inverse. These two functions define a handle with the face set

$$\{(1, 2, f(1)), \dots, (m_1 - 1, m_1, f(m_1 - 1)), (m_1, 1, f(m_1)),$$

$$(m_1 + 1, m_1 + 2, g(m_1 + 1)), \dots,$$

$$(m_1 + m_2 - 1, m_1 + m_2, g(m_1 + m_2 - 1)),$$

$$(m_1 + m_2, m_1 + 1, g(m_1 + m_2))\} .$$

For $k = 3, 4$ and some tuples of edge numbers the numbers of possible handles have been summarized in Table 1.

We consider symmetric group actions on the set of handles $\mathcal{K} = \mathcal{K}(k; m_1, \dots, m_k)$ for k caps of sizes m_1, \dots, m_k , respectively. For the theory of group actions, see [11, p.25]. By a group action on a set S , the set S is partitioned into disjoint orbits S_1, \dots, S_n (see [11, p.28]), and a set of generators of S is a choice of elements $s_i \in S_i$, $i = 1 \dots, n$. We identify generators of \mathcal{K} which completely determine \mathcal{K} modulo group actions.

Rotations: The topology of the caps remains unchanged if a rotation of the vertex numbers of a cap is performed. More precisely, let

$$\omega_{R,i}(l) = \begin{cases} l \oplus 1 & \mu_i < l \leq \mu_{i+1} \\ l & \text{otherwise,} \end{cases} ,$$

$$\Omega_R = \Omega_{R,1} \times \dots \times \Omega_{R,k} ,$$

$$\Omega_{R,i} = \text{subgroup of } \Sigma_{\mu_k} \text{ generated by } \omega_{R,i} ,$$

$$\Sigma_{\mu_k} \text{ denotes the symmetric group on the set } \{1, \dots, \mu_k\} .$$

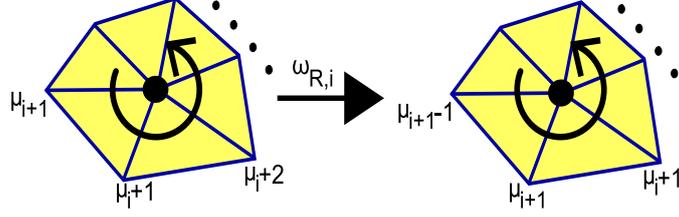


Figure 5: A rotation.

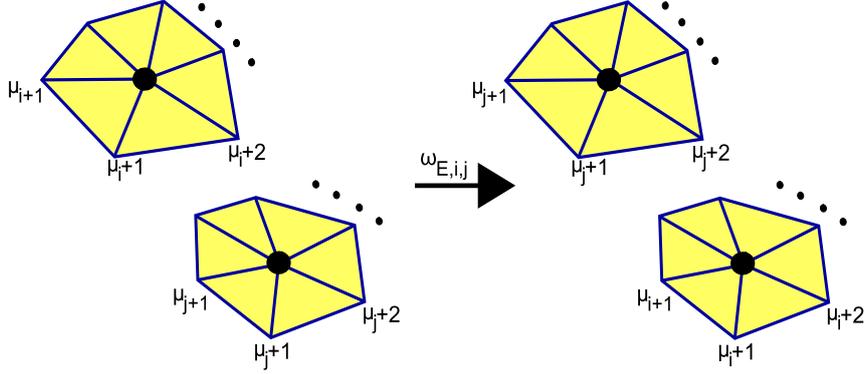


Figure 6: An exchange

We obtain a group action of Ω_R on \mathcal{K} by applying Ω_R to every vertex of every simplex of a handle K . A representant is Rotations are illustrated in Figure 5.

Exchanges: The order of two caps, consisting of the same number of vertices, can be exchanged. If

$$\omega_{E,i,j}(l) = \begin{cases} l - \mu_j + \mu_i & \mu_j < l < \mu_{j+1} \\ l + \mu_j - \mu_i & \mu_i < l < \mu_{i+1} \\ l & \text{otherwise} \end{cases}$$

and $\Omega_{E,i,j} = \{\text{id}, \omega_{E,i,j}\}$ the group of exchanges of cap i and cap j , then

$$\Omega_E = \prod_{i < j, m_i = m_j} \Omega_{E,i,j}$$

operates on \mathcal{K} . Exchanges are illustrated in Figure 6.

An easy computation shows that the two operations commute, i.e.

$$\omega\tau(K) = \tau\omega(K) \text{ for } \omega \in \Omega_R, \tau \in \Omega_E \text{ and } K \in \mathcal{K}.$$

As a consequence, we can apply rotations and exchanges in an arbitrary order to a handle. With these operations, only very few elements are required to generate elements of \mathcal{K} . This is illustrated by comparing the last two columns of Table 1.

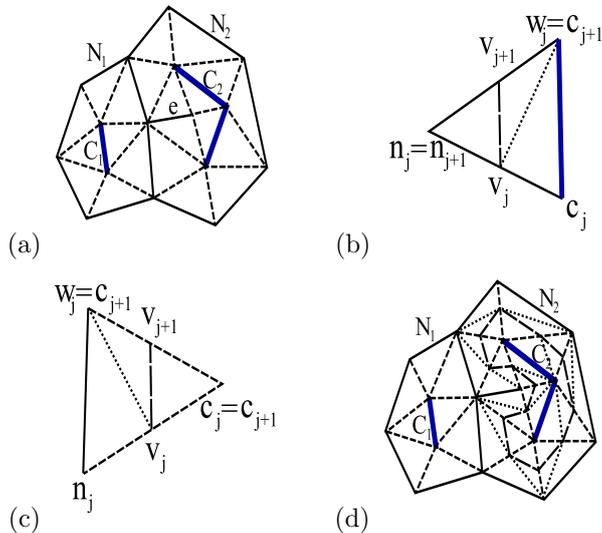


Figure 7: In (a), Λ^1 consists of two and Λ^2 of three vertices connected by a bold line. N_1 and N_2 have three common neighbor vertices, and N_2 forms no simple closed polygon because of edge e . The iterative refinement algorithm is demonstrated in (b) and (c). Bisector vertices v_j are inserted as well as edges between them (dashed line), and the arising quadrilaterals are triangulated (dotted line). The result of the refinement algorithm is shown in (d).

4 Implementation of the Topology Adaptation

For implementation of the topology adaption algorithm (compare the Segmentation Concept 1) the opening of the mesh has to be performed in such a way that the boundary of the opened mesh M_b consists of simple closed polygons.

For this purpose vertices of colliding mesh parts (which have been detected by the hashing algorithm described in Section 2) are grouped into disjoint connected sets $\Lambda^1, \dots, \Lambda^k$ such that for $i \neq j$, two arbitrary vertices $v \in \Lambda^i$, $w \in \Lambda^j$ have no common neighbor.

The set N_i of all neighboring vertices of vertices in Λ^i without Λ^i is a neighborhood of Λ^i , which consists of connected components $N_i^0, \dots, N_i^{l_i}$. We assume that the neighborhoods N_i^j are pairwise disjoint and that its edges form a simple, closed polygon, otherwise the following preprocessing routine is used: We insert new vertices on the bisectors of edges between vertices of Λ^i and of N_i^j , and connect these vertices by edges as shown in Figure 7. Arising quadrilaterals are triangulated. As a result, the edges connecting the bisectors form a simple closed polygon around Λ^i , and their neighborhoods are pairwise disjoint. The routine is illustrated in Figure 7(b)-(d).

In general, the neighborhood N_i consists of several components $N_i^0, \dots, N_i^{l_i}$. One component, say N_i^0 , encloses Λ^i , and the other components are enclosed by Λ^i , see Figure 8. The outside component N_i^0 can be computed from the orientation of the mesh. Components $N_i^1, \dots, N_i^{l_i}$ belong to enclosed parts of the mesh. There are two different kinds of inclusion neighborhood components

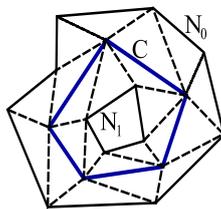


Figure 8: The neighborhood of cluster C consists of two components. N_0 is the component outside C , N_1 is an inclusion component.

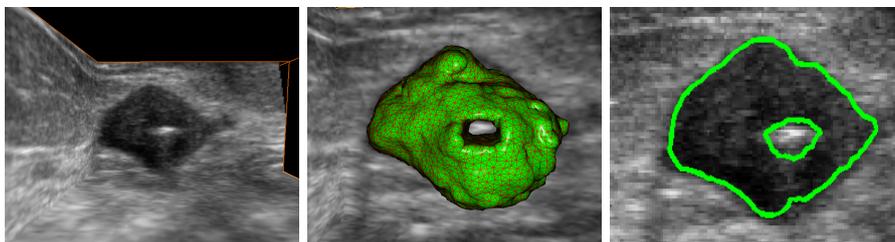


Figure 9: The original ultrasound image is shown on the left hand side. The final segmenting mesh is shown in the middle. A projection on the y - z plane is presented on the right hand side.

N_i^j for $j \geq 1$ and propose two different procedures:

- If N_i^j contains a triangle, i.e. there exist $v_1 \in N_i^j$, $v_2, v_3 \in V \setminus \Lambda^i$ neighbors of v_1 , s.t. $(v_1, v_2, v_3) \in F$. Then the neighborhood refinement algorithm is applied such that the boundary of Λ^i towards N_i^j as well as the boundary of N_i^j towards Λ^i are Jordan polygons. The connections between the two polygons are discarded. For each polygon, the barycenter of the vertices is inserted, and connected to the polygon. Thus, the mesh is split into two separate components.
- If N_i^j contains no triangle, no real inclusion has been detected, and N_i^j is added to Λ^i .

5 Results and Discussion

We tested our algorithm for topology adaptation in connection with the active contour algorithm published in [1] on artificial and medical test images. In both cases the dark part is regarded as the object to be segmented. To initialize the segmentation algorithm a small sphere is manually placed inside the dark part of the voxel image. The evolving surface moves towards the boundary of the object.

As far as possible we compare the numerical results to those given in [8].

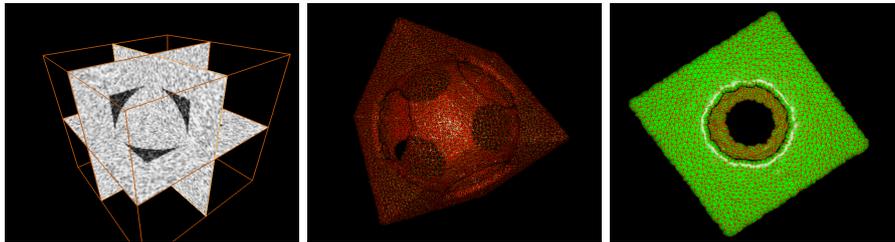


Figure 10: A cube with a spherical cavity and some Gaussian noise added. As a segmentation result we obtain the mesh shown on the middle and right. In the middle diagram only the edges are visualized.

Object	Voxel size	Iterations	Vertices	Sec.
Cyst	$199 \times 99 \times 171$	133	8687	9
Cube	$100 \times 100 \times 100$	709	13576	65
Genus 3	$100 \times 100 \times 100$	677	9680	28
Torus	$100 \times 100 \times 100$	215	5454	8

Table 2: For each test example, the number of iterations and vertices and the running time of the segmentation algorithm is given. Tests were performed on a 3.5 GHz computer with 2 GB RAM.

- An ultrasound image of a cyst is segmented. The white part inside the cyst, running from front to back, stems from a biopsy needle, see Figure 9. The segmentation is used to determine the shape of the cyst and the position of the biopsy needle. As the projection to the y - z plane shows the cyst and the needle are accurately segmented, also in regions where the topology of the mesh has been adapted during the evolution of contour.
- The next example concerns a computer generated voxel image of a cube with a spherical cavity (see Figure 10). Different from the example in [8], every side of the cube contains a hole such that the segmenting contour of the object has genus 5.
- The next example shows an object of genus 3, the starting ball chosen on one crossing of the four parts. Therefore, a topology change with four parts hitting at the same iteration step is performed. The result is shown in the left part of Figure 11.
- The last example shows a torus with 4 inclusions. As segmentation result a torus enclosing 4 spheres is obtained, see the right part of Figure 11.

The performance of our topology change algorithm tested on the four examples is summarized in Table 2.

The numerical experiments demonstrate the robustness and efficiency of the topology completion algorithm. Although no reparameterizations are performed during the mesh evolution, and no global restrictions are inferred, the algorithm works stable. As expected, the running times of the segmentation algorithm roughly depend linearly on the number of iterations resp. vertices. The running

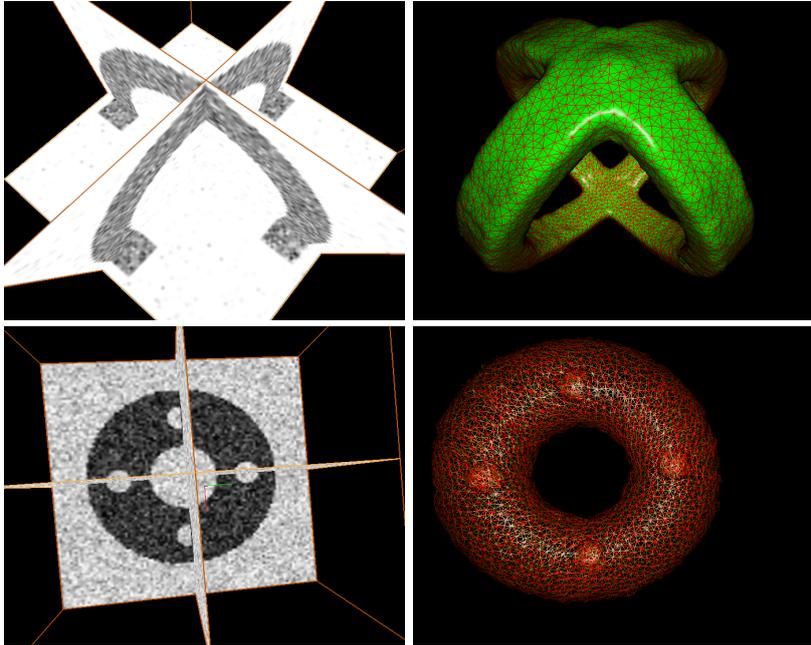


Figure 11: Voxel images for the last two examples and the segmentation results.

time for segmentation of the object of genus 3 is a bit shorter than expected, because many vertices reach the object boundary rather early, and only a comparably small number of vertices is actually updated during an evolution step. As a consequence, we obtain a speedup versus previous 3D topology adaptive segmentation routines. The cube with spherical cavity can be compared to the first example in [8]. There, only one face of the cube is penetrated by the ball, such that their object to recover has genus 0. The running time of our algorithm is much better, although the topology in our case is more complex, more faces are involved, and more iteration cycles are performed.

6 Conclusion and Outlook

We introduce a very efficient novel topology completion system which runs independently of the evolution, does not require any reparameterizations and runs stable, even if the mesh is not regularly sampled. We introduce a novel and efficient collision and self-collision detection algorithm for triangular meshes, which runs in linear time and does not require complex data structures or huge memory resources. The system is designed for interactive applications.

Due to the topological completion formulas obtained by the homology criterion, we were able to develop a very robust topological completion system, working with arbitrary mesh deformation algorithms. Since our (self-)collision detection algorithm works in linear expected time, the system is also very efficient resulting in significantly reduced computation times. For numerical experiments, we used a standard balloon model, thus losing overall efficiency for

segmentation a bit. As a future work, it seems to be interesting to combine the presented topological completion algorithm with a locally adaptive mesh evolution as presented in [8].

Acknowledgement

This work has been supported by the Austrian Science Fund (FWF) within the national research networks Industrial Geometry, project 9203-N12, and Photoacoustic Imaging in Biology and Medicine, project S10505-N20. 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, GZ:UNI-0404/460*.

References

- [1] J. Abhau, W. Hinterberger, and O. Scherzer. Segmenting surfaces of arbitrary topology: A two-step approach. In *Medical Imaging 2007: Ultrasonic Imaging and Signal Processing*. Proceedings of SPIE – Volume 6513, 2007.
- [2] S. Bischoff and L. Kobbelt. Cartoon extraction based on anisotropic image classification. In *Vision, Modeling, and Visualization Proceedings*, pages 293–300, 2006.
- [3] V. Caselles, F. Catté, T. Coll, and F. Dibos. A geometric model for active contours in image processing. *Numer. Math.*, 66(1):1–31, 1993.
- [4] Y. Chen and G. Medioni. Description of complex objects from multiple range images using an inflating balloon model. *Comput. Vision Image Understanding*, 61, No 3:325–334, 1995.
- [5] H. Delingette. Adaptive and deformable models based on simplex meshes. In *IEEE Workshop of Non-Rigid and Articulated Objects*. IEEE Computer Society Press, 1994.
- [6] A. Hatcher. *Algebraic Topology*. Cambridge University Press, 2002.
- [7] J. O. Lachaud and A. Montanvert. Deformable meshes with automated topology changes for coarse-to-fine three-dimensional surface extraction. *J. Med. Image Anal.*, 3, No 2:187–207, 1999.
- [8] J. O. Lachaud and B. Taton. Deformable model with a complexity independent from image resolution. *Comput. Vision Image Understanding*, 99(3):453–475, 2005.
- [9] J.O. Lachaud and B. Taton. Deformable model with adaptive mesh and automated topology changes. In *Proceedings of 4th International Conference on 3-D Digital Imaging and Modeling (3DIM'2003)*, 2003.
- [10] J.O. Lachaud and B. Taton. Resolution independent deformable model. In *International Conference on Pattern Recognition (ICPR'2004)*, pages 237–240, 2004.

- [11] S. Lang. *Algebra*. Springer Verlag, 2002.
- [12] F. Leitner and P. Cinquin. Complex topology 3d-objects segmentation. In *SPIE Conference on Model Based Vision Development and Tools*, pages 16–26, 1992.
- [13] W.S. Massey. *A Basic Course in Algebraic Topology*. Springer Verlag, 1991.
- [14] T. McInerney and D. Terzopoulos. A finite element model for 3d shape reconstruction and nonrigid motion tracking. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 518–523, 1993.
- [15] T. McInerney and D. Terzopoulos. T-snakes: Topology adaptive snakes. *Med. Image Anal.*, 4(2):73–91, 2000.
- [16] T. Moller. A fast triangle-triangle intersection test. *J. Graph. Tools*, 2/2:25–30, 1997.
- [17] S. Osher and J. A. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton–Jacobi formulations. *J. Comput. Phys.*, 79(1):12–49, 1988.
- [18] PARI. *PARI/GP, version 2.1.7*. The PARI Group, Bordeaux, 2005. available from <http://pari.math.u-bordeaux.fr/>.
- [19] J. J. Rotman. *An Introduction to Algebraic Topology*. Springer, New York, 1988.
- [20] R. Szeliski, D. Tonnesen, and D. Terzopoulos. Modeling surfaces of arbitrary topology with dynamic particles. In *Proc. Conf. Computer Vision and Pattern Recognition (CVPR’93)*. IEEE Computer Society Press, 1993.
- [21] M. Teschner, B. Heidelberger, M. Mueller, D. Pomeranets, and M. Gross. Optimized spatial hashing for collision detection of deformable objects. In *Proceedings of Vision, Modeling, Visualization*, pages 47–54, 2003.
- [22] M. Teschner, S. Kimmerle, B. Heidelberger, G. Zachmann, L. Raghupathi, A. Fuhrmann, M.P. Cani, F. Faure, N. Magnenat-Thalmann, W. Strasser, and P. Volino. Collision detection for deformable objects. *Comput. Graphics Forum*, 24:61–81, 2005.
- [23] A. Witkin, M. Kass, and D. Terzopoulos. Snakes: Active contour models. *Int. J. Comput. Vision*, 1, No 4:321–331, 1987.

Appendix

For a triangle T in \mathbb{R}^3 given by its vertices T_1, T_2, T_3 and a point P in \mathbb{R}^3 let

$$d(P, T) := \min \{ \|P - T_1\|, \|P - T_2\|, \|P - T_3\| \}.$$

We use the notation $T = (T_1, T_2, T_3)$ and denote by $\text{pr}_T(P)$ the orthogonal projection of P in the plane spanned by T - which is of course only well-defined if the triangle does not degenerate.

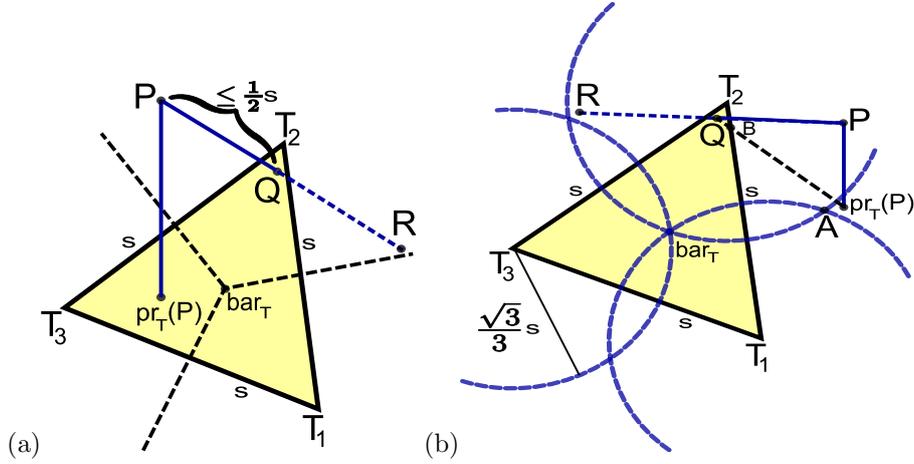


Figure 12: In (a), edge \overline{PR} intersects triangle (T_1, T_2, T_3) in Q . In (b), the projection $pr_T(P)$ of P onto the plane spanned by T lies outside the three circles.

Lemma 8. Assume that $T = (T_1, T_2, T_3)$ is a triangle in \mathbb{R}^3 , and $P \in \mathbb{R}^3$. Then

$$d(P, T) = \sqrt{\|P - pr_T(P)\|^2 + d(pr_T(P), T)^2}.$$

Proof. The situation is illustrated in Figure 12(a). Looking at the Voronoi diagram of the three points $T_1, T_2, T_3 \in \mathbb{R}^3$, we deduce that for some $i = 1, 2, 3$, $\|P - T_i\| \leq \|P - T_j\|$ for all $j = 1, 2, 3$ if and only if

$$\|pr_T(P) - T_i\| \leq \|pr_T(P) - T_j\| \text{ for all } j = 1, 2, 3. \quad (2)$$

Therefore, if $d(P, T) = \|P - T_i\|$ for some i , we have

$$\begin{aligned} d(P, T) &= \|P - T_i\| = \sqrt{\|P - pr_T(P)\|^2 + \|pr_T(P) - T_i\|^2} \\ &\stackrel{(2)}{=} \sqrt{\|P - pr_T(P)\|^2 + d(pr_T(P), T)^2} \quad \square \end{aligned}$$

Now we can give a proof of Theorem 3:

Proof. Without loss of generality we can assume that every edge of both triangles S and T has maximal edge length s , i.e. both triangles are equilateral. Moreover, we assume that an edge e of S intersects T (otherwise we interchange the role of S and T), and we denote the intersection point by Q .

Let P be an endpoint of e which fulfils $\|P - Q\| \leq \frac{1}{2}s$. Denote by bar_T the barycenter of T . (see Figure 12(a)). We consider two cases concerning the position of $pr_T(P)$:

- $d(pr_T(P), T) \leq d(bar_T, T)$: In this case, we have that

$$\|P - pr_T(P)\| \leq \|P - Q\| \leq \frac{s}{2}$$

and

$$d(pr_T(P), T) \leq d(bar_T, T) = \frac{1}{3}\sqrt{3}s.$$

Therefore, using Lemma 8 we deduce that

$$d(P, T) = \sqrt{\|P - pr_T(P)\|^2 + d(pr_T(P), T)^2} \leq \sqrt{\frac{1}{4} + \frac{1}{3}}s \leq \sqrt{\frac{2}{3}}s.$$

- $d(pr_T(P), T) > d(bar_T, T)$: In this case, $pr_T(P)$ lies in the complement of the discs around the points T_i with radius $\|bar_T - T_i\|$, as illustrated in Figure 12(b). Since P is projected to $pr_T(P)$ outside T , there exists a point B contained in an edge (T_i, T_j) of T , such that

$$\|P - B\| \leq \|P - Q\|$$

(namely the intersection point of the line $(pr_T(P), Q)$ with one of the triangle edges). Since $\|P - Q\| \leq \frac{s}{2}$, we can deduce that

$$\|P - B\| \leq \frac{s}{2} \quad (3)$$

From (3) it also follows that $\|pr_T(P) - B\| \leq \frac{s}{2}$, and moreover one of the norms $\|T_i - B\|$, $\|T_j - B\|$ is smaller than $\frac{s}{2}$. Therefore, we obtain

$$\begin{aligned} d(pr_T(P), T) &\leq \sqrt{\min\{\|T_i - B\|^2, \|T_j - B\|^2\} + \|pr_T(P) - B\|^2} \\ &\leq \frac{1}{2}\sqrt{2}s. \end{aligned} \quad (4)$$

Let A be the image of bar_T under reflection along the edge (T_i, T_j) . Since $pr_T(P)$ lies in the complement of the discs around the points T_i with radius $\|bar_T - T_i\|$, we have

$$\|B - pr_T(P)\| \geq \left\| A - \frac{1}{2}(T_i + T_j) \right\| = \left\| bar_T - \frac{1}{2}(T_i + T_j) \right\| = \frac{1}{6}\sqrt{3}s \quad (5)$$

Altogether, using Lemma 8, we obtain

$$\begin{aligned} d(P, T) &= \sqrt{\|P - pr_T(P)\|^2 + d(pr_T(P), T)^2} \\ &= \sqrt{\|P - B\|^2 - \|B - pr_T(P)\|^2 + d(pr_T(P), T)^2} \\ &\stackrel{(3)(5)(4)}{\leq} \sqrt{\left(\frac{1}{2}s\right)^2 - \left(\frac{1}{6}\sqrt{3}s\right)^2 + \left(\frac{1}{2}\sqrt{2}s\right)^2} \\ &= \sqrt{\frac{2}{3}}s. \end{aligned}$$

Altogether, we have found a point P of triangle S which is closer to T than $\sqrt{\frac{2}{3}}s$, and the assertion follows. \square