ABSTRACT
The following work contributes to the field of real-time simulation environments in which deformable objects are manipulated by rigid bodies. To model these manipulations, appearing collisions between bodies and objects have to be detected. Complex bodies are approximated by simple geometrical bounding volumes which enable the use of efficient collision detection algorithms. Depending on the consistency of the objects and the force which is transferred to them, deformable objects behave in different ways. The work at hand presents a method which triggers the corresponding collision response algorithm according to the depth the deformable objects have been penetrated by the bounding volumes. It introduces the "Line of No Return", which triggers topological changes instead of elastic or plastic deformations. Different parameters of this method lead to a variety of material consistencies. Results are presented for tissue handling with microsurgical forceps to decide whether tissue is dragged, pushed or teared. The computation time does not depend on the amount of collisions which were detected, thereby the real-time criteria for the simulation scene could thoroughly be met.

KEY WORDS
Virtual Reality, Physically Based Simulation, Collision Detection, Collision Response

1 Introduction
Physically based simulation models have become a common solution in different cases where simulations with a high level of realism are needed. Examples can be found in the game industry or in training simulation systems based on virtual reality. Such training systems have become a fast growing field in medical education. There already exist various apparatuses that teach and train different skills. One educational objective is the achievement of fine motor skills to proceed microsurgical tasks without risking the patients’ health.

A typical task in microsurgery is to grab and remove obsolete tissue. For example, connective tissue around a vessel’s lumen is removed before a surgeon can proceed suturing it. For this purpose micro-scissors and forceps are used. Grabbing tissue with forceps can result in tearing. This can happen by accident and injure the tissue, a surgeon is working on. In some cases though this is a typical way of dissecting connective tissue.

To model this behavior an algorithm was developed which is based on bounding capsules for dragging and pushing. Furthermore the "Line of No Return" inside the bounding volume is introduced which triggers topological changes of the tissue.

This setting can be used in different simulation environments or games where deformable objects are pushed, dragged or teared by tools depending on the force that is applied. The algorithm is time-efficient and achieves realistic results. Material is pushed aside after it has been topologically changed. Parameters of the model can be adjusted, leading to different behavior from sharp to stub tools.

The shown methods were developed for microsurgical procedures and implemented into the medical training simulator which was presented in [1]. Results are shown for tissue manipulation with forceps.

Figure 1. Tissue Manipulation With Forceps.

(a) Initial scene (b) Pushing
(c) Grabbing (d) Topological changes
2 Related Work

Physically based models have been well investigated in the last decades starting with the work of Terzopoulos et al. in 1987 [2]. A good overview can be found in [3].

Generally spoken, collision detection is not coupled to a specific simulation model, whereas collision response algorithms are dependent on the underlying model. For a comprehensive work about collision detection for real-time in general, an interested reader is referred to [4], whereas a summary of collision detection algorithms for deformable objects can be found in [5]. Collisions with deformable objects can cause different response behaviors. Depending on the force which is transferred, an object is being deformed plastically or elastically. Another collision scenario is the penetration by a sharp tool for cutting into the surface of a deformable object. In this case the object might be dissected exactly on the line where the tool collides with the object. Topological changes were introduced in [6] for the first time by removing elements from a tetrahedral mesh. [7] introduced splitting a mesh on existing element boundaries. Since then a lot of different algorithms have been introduced for topological changes, see [8] [9] [10] [11] [12] [13]. Most of the algorithms are based on remeshing and can also be used to simulate tearing or fracturing material.

3 Methods

In the presented work deformable objects are represented by tetrahedral meshes (see figure 2(b)) and are simulated by a physically based model. For the implementation the algorithms from [14] have been used. Triangles on the mesh surface are used to render the object, see figure 2(a).

For the manipulation of the mesh the tools were surrounded by bounding capsules. Figure 1(a) shows a pair of forceps and a simple deformable object. One can think of three tasks that can be completed with forceps: push tissue (see figure 1(b)), grab tissue (see figure 1(c)), dissect/tear tissue (see figure 1(d)). For the description of these tasks, the most simple simulation mesh \(X\) was considered, consisting of four nodes, six springs, one tetrahedron and four surface triangles. The mesh will be manipulated by two bounding capsules \(BC1\) and \(BC2\) which can be pushed against each other. This happens when the forceps are closed. The axis of each bounding capsule will be defined as the “Line Of NO Return (LONOR)”. The test case is illustrated in figure 3. Although the methods are based on three dimensional models, for clearness reasons the illustrations are simplified to two dimensions.

3.1 PUSH

The tetrahedron \(X\) is penetrated by the bounding volume but not by the LONOR at time step \(t\) and did also not pass the LONOR within the time step \(\Delta t\) (time from \(t - 1\) to \(t\)). The exact collision on the surface of the bounding capsules has to be calculated. The basic collision detection methods, describing collisions between triangles cylinders and other geometrical objects were adapted from [4]. Depending on the simulation model, e.g. linear penalty forces, have to be applied to the nodes, to "push" \(X\) out of the volume which leads to the fact, that the surface of \(X\) is lying on \(BC2\). In the implementation the algorithms to calculate response forces introduced in [15] were used. Figure 4 illustrates the task before (see figure 4(a)) and after (see figure 4(b)) the collision forces have been applied to \(X\).

3.2 GRAB

The tetrahedron \(X\) has penetrated both bounding capsules \(BC1\) and \(BC2\), but did not pass the LONOR within \(\Delta t\). To simulate dragging, friction forces have to be applied to certain nodes of \(X\). To find the nodes a bounding box is
built which is aligned to the forceps (see figure 5). All Nodes Inside the Bounding Box (NIBBs) are stored. Afterwards the nodes which have penetrated the bounding capsules are pushed out of them as described in the paragraph "PUSH". In the following time step \( t + 1 \) a force is applied to the NIBBs, so they follow the movement of the forceps. In this method an anchor point is used for all NIBBs, according to the position relative to the forceps, simulating static friction.

Figure 5. Grabbing.

### 3.3 DISSECT / TEAR

The tetrahedron \( X \) has penetrated the bounding capsules BC1 and has passed or is in contact with the LONOR in \( t \). Thus \( X \) has to be dissected.

Two cases must be distinguished: either \( X \) was pushed by another bounding capsule BC2 over the LONOR or the LONOR has penetrated \( X \) by its own movement. Usually, it is a mixture of both, as the mesh is manipulated between the two bounding capsules BC1 and BC2. It is assumed that the mesh can only be pushed over the LONOR of BC1 by a force BC2 applies to it or vice versa. Furthermore, the following three cases have to be considered (the illustrations below are in bird’s eye view):

1. \( X \) is in contact with the LONOR in \( t \), see figure 6(a)
2. \( X \) is not in contact with the LONOR in \( t \), but passed it in \( \Delta t \), see figure 7(a)
3. \( X \) collides between BC1 and BC2 which are also in contact with each other, see figure 8(a)

#### Case 1:

The LONOR collides with \( X \) in \( t \). A topological change has to be performed. Figure 6(a) illustrates the collision and figure 6(b) shows the mesh after the collision response depending on the algorithm which is used for topological changes. In this case the illustration shows \( X \) after a remeshing algorithm has been performed.

Before the mesh can be changed topologically, the collision surface has to be determined. A simple continuous collision detection algorithm, where the plane which is specified by the LONOR at \( t − 1 \) and \( t \), might not find all relevant collision information as the tetrahedron could also have done a movement in \( \Delta t \). The LONORs of BC1 and BC2 detect the collisions of \( X \) and the continuative part of the plane, but not with the part of the plane, between the LONORs themselves. The continuative planes are illustrated by the lines from the centers to the borders of the bounding capsules in figure 6(a).

#### Case 2:

At \( t \), \( X \) does not collide with the LONOR therefore the collision detection is not triggered, see figure 7(a). Considering the two things which could happen in \( \Delta t \), a continuous collision detection might not always work as expected: First, the forceps could have moved so fast, that it passed \( X \) without any movement of \( X \) itself. In this case the continuous collision detection as described in case 1 would be sufficient. Second, \( X \) is pushed against BC2, the force is strong enough and \( X \) “jumps” over the LONOR. It has to be checked, whether \( X \) passed the LONOR in \( \Delta t \). To cover this case the velocity of the tetrahedron is checked. If it is pointing to the LONOR, case 1 is triggered and \( X \) is pushed out. Otherwise \( X \) gets dissected, see 7(b). The collision surface is again detected by the continuative plane which is specified by the LONORs of BC1 and BC2.

Figure 7. Tetrahedron Passes Line Of No Return Within One Time Step.

#### Case 3:

Depending on the elasticity and sharpness of the tool the material can be completely dissected, when the forceps is closed. In order to consider this behavior, a third
case was added to the model. Here, all tetrahedrons are dis-
sected along the plane which is specified by the LONORs of BC1 and BC2 once the bounding capsules collide with each other. The distance of the bounding volumes which triggers case 3 is variable which allows the model to sim-
ulate material ranging from easy breakable to gummy like behavior. Case 3 is illustrated by figure 8 and shows the collision, see figure 8(a), and the response after the tetrahe-
dron has been subdivided and pushed out of the bounding volumes, see figure 8(b).

4 Results

To show the efficiency of the methods a mesh consisting of 960 tetrahedrons has been manipulated by six bound-
ings capsules which approximate the forceps. Figure 9(a) shows the initial test scene and fig 9(b) the same scene af-
fter the mesh has been pushed, dragged and dissected by the bounding capsules. The methods were tested on an Intel® Core™ i7 CPU with 2.67 GHz, 6GB Memory and a NVidia® GeForce GTX 285. Figure9 shows the time needed to run the simulation related to the collisions be-
tween the triangles of the mesh and the bounding capsules. Five milliseconds were never exceeded in the given sce-
nario. The brighter the colors in the chart, the more tetrahe-
drons were simulated. After the mesh has been subdivided a couple of times during simulation the mesh consisted of 2441 tetrahedrons. The test shows that the simulation is de-
pendent on the amount of collisions. Topological changes have an impact on the computation time. The more often the mesh is topologically changed, the more tetrahedrons have to be simulated.

The algorithms were used for soft tissue handling in a microsurgical training simulator where a surgeon can grab, hold and pull the tissue away until it tears. Furthermore he can push the outer mesh backwards over the inner mesh or hold the outer mesh an dissect it, by pushing the forceps together. Figure 10 shows some visual results of the application scenario and the different tasks.

5 Conclusion

By modeling the forceps out of ordinary bounding capsules and adding the "Line of No Return" an algorithm was intro-
duced which triggers different collision response options. Deformable objects based on tetrahedral meshes were used and the validity of the method has been shown for micro-
surgical tasks, where tissue is manipulated by forceps in real-time. The introduced contribution is not limited to this case. One can think of other scenarios where deformable objects have to be pushed or cut with blunt or sharp blades. Future work could focus on validation of the methods for different simulation models and bounding volumes. Also dynamic friction could be applied to the model, by using the depth the object has penetrated the bounding volume as indicator for the friction force.

Acknowledgements

Prof. Dr. Reinhard Männ er contributed valuable com-
ments to this work. The algorithms were developed within a project in cooperation with the VRmagic GmbH in Mannheim, Germany (www.vrmagic.com). The project was kindly supported by the German Bundesministerium für Wirtschaft und Technologie (BMWi) under grant ZIM (KF2351202SS9).
References


