Available at
www.ElsevierComputerScience.com
POWERED BY SCIENCE DIRECT

Medical Image Analysis 8 (2004) 23–33

MEDICAL IMAGE ANALYSIS

www.elsevier.com/locate/media

Modelling of interaction between a spatula and a human brain ☆

Kim V. Hansen *, Lars Brix, Christian F. Pedersen, Jens P. Haase, Ole V. Larsen

Department of Health Science and Technology, Aalborg University, Fredik Bajersvej 7, D1-203, Aalborg East 9220, Denmark

Received 9 September 2002; received in revised form 26 May 2003; accepted 7 July 2003

Abstract

This paper describes a method for surgery simulation, or more specifically a learning system of how to use a brain spatula. Improper use of brain spatulas can lead to brain tissue lesions such as tearing of brain tissue and ischemia. The idea is to provide surgeons with a tool which can teach them the correlation between deformation and applied force. The system includes a Finite Element based model of the brain in a Virtual Reality setup with haptic feedback. The physical model links the shape of the deformable model with the associated force. The interaction between the spatula and the brain model is handled by a collision response method which aims at smoothing the discrete haptic feedback. The experimental results are promising even though the used force feedback device is somewhat constraining the realism.

Keywords: Surgery simulation; Brain spatula; Dynamic finite element modelling; Collision detection; Collision response

1. Introduction

Surgery simulation has gained growing interest in recent years. The reasons are mainly reduced risk for patients and easy repeatability of surgical procedures. As both computer graphics and haptic interface devices have improved and the quality has become sufficient for many purposes (Bro-Nielsen, 1998) different kinds of surgery simulators have been developed. Examples of surgery simulators are: A liver model (Cotin et al., 2000) that allows cutting in pre-defined areas, an abdominal surgery training system (Kühnapfel et al., 2000) and vessel suturing which measures user performance (Brown et al., 2001).

The work presented in this paper is focusing on neurosurgery simulation and the interaction between a spatula and a human brain. A spatula is used to retract parts of the brain to gain access to the surgical target area (see Fig. 1). Spatulas used in neurosurgery are bendable and can be fitted into any desired shape needed for the specific task.

Brain retraction is necessary for the majority of intra-cranial procedures and injuries are unfortunately not uncommon (Andrews and Bringas, 1993; Yasargil, 1999). Injuries are caused by focal pressure of the retractor blade on the brain, which produces a local deformation of brain tissue for a finite period of time. Such deformations cause a reduction of the blood flow, ischemia. It may also cause direct injury to the brain tissue, cutting or tearing (Andrews and Bringas, 1993). These types of injuries are dependent on the shape and the number of retractors as well as the pressure and the duration of the retraction. Spatulas were mainly hand-held up till the late 1970s (Yasargil, 1999). Retraction with hand-held spatulas is unstable and in some cases produces an unnecessarily high brain pressure. Direct contusions and sliding of spatulas cutting into the brain tissue were important factors for complications (Rosenørn and Diemer, 1987).

Micro-surgical techniques led to the development of self-retaining spatulas in order to reduce brain lesions and ease manipulations (Yasargil, 1999). Self-retaining brain spatulas are now considered indispensable as they allow the neurosurgeon to work unhindered by an assistant's hand. Both in the case of hand-held and self-retaining spatulas, it is of utmost importance that the surgeon learns how to place the spatula and how much

☆ Supplementary data associated with this article can be found at doi: 10.1016/j.media.2003.07.001.

* Corresponding author. Tel.: +45-96358782.
E-mail address: kvh@hst.auc.dk (K.V. Hansen).

1361-8415/$ - see front matter
doi:10.1016/j.media.2003.07.001

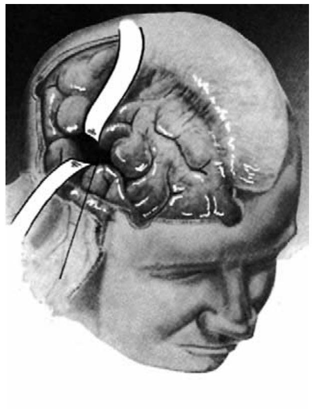


Fig. 1. Brain retraction using two spatulas (Poppen, 1960).

force to apply when retaining the tissue. To the best of our knowledge, the topic has never been introduced of how to learn to use a brain spatula. Whereas the necessity of careful positioning, monitoring of cerebral blood flow, and direct pressure of the brain have all been discussed (Yasargil, 1999), but nobody mentions the fact that the trainee only learns how to use a spatula by watching his mentor. It is not possible to communicate the sensing of the spatula on the brain from trainer to trainee in the present operation room situation.

We present a virtual brain model with haptic feedback, which interactively in a Virtual Reality (VR) environment can teach the correlation between brain deformation and applied pressure. In an VR environment, the trainee can rehearse the retraction numerous times without risking the health of a patient or worrying about ethical issues concerning animal studies. Direct pressure on the brain can easily be monitored and the trainee can be given direct feedback such as warnings or even a performance evaluation.

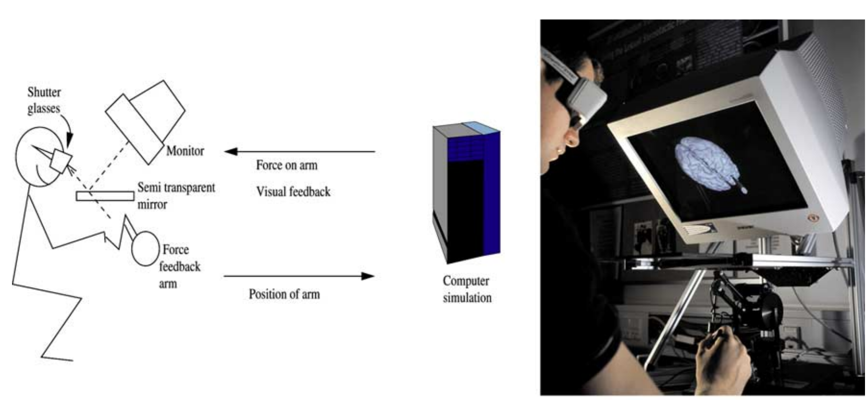
The outline of this paper is as follows. Section 2 describes the simulation set-up and the general functionality, and Section 3 how the virtual brain and spatula-models are built. The brain model has to be deformable and an interaction with the model is necessary. Details about the implementation of our approach are described in Sections 4–6. Experimental results are presented in Section 7, and conclusion and discussion in Section 8.

2. Simulator outline

One of the most important issues of learning how to use a brain spatula is to learn how much force to apply when retaining tissue. This issue is addressed in the simulator by enabling a user to apply a rigid virtual spatula onto a deformable virtual brain. When doing so, the user is provided with a suitable graphical and haptic feedback. In order for the visual feedback to appear convincing and smooth to the eye a refresh rate of 30 Hz is needed. The haptic feedback has to be updated with 500–1000 Hz (Cotin et al., 1996). Since 30 Hz for the visual feedback is a hard requirement 1000 Hz for the haptic feedback can seem almost impossible, fortunately it is possible to obtain this through interpolation.

From these feedbacks the user can train hand–eye coordination in different scenarios and learn how to correlate tissue deformation and force. Hopefully, by studying this correlation the user will learn how to quickly find the appropriate force to apply in a given situation. Even more important, the user will learn not to apply too much force which can lead to irreversible and severe brain damages. Several studies (Rosenørn and Diemer, 1987; Andrews and Bringas, 1993) indicate for how long a certain force can be applied without damaging the tissue. To apply this research, the simulator can map forces into a color so that the force distribution in the tissue can be monitored. Of course, assigning all kinds of colors to the brain tissue will reduce realism, so alternatively the tissue is just marked red when forces exceed a certain threshold.

Fig. 2 illustrates the hardware setup of the simulator. A pen, which is the physical representation of the virtual spatula, is attached to the force feedback arm (Phan-



(a) (b)

Fig. 2. (a) Drawing of the setup. (b) Real hardware setup. (This figure is available in colour, see the on-line version.)

tom™ 1.0). The position of the pen can be determined with 6 degrees of freedom (DOF), i.e., 3 translational degrees and 3 rotational degrees. Furthermore, the pen is able to provide the user with a translational 3 DOF force that is applied at the tip of the pen.

The graphical feedback is handled by a monitor, a semi-transparent mirror and a pair of cordless shutter glasses. The monitor shows stereo images, i.e. both left and right eye perspective images, which are reflected by the semi-transparent mirror into the shutter glasses. The shutter glasses are synchronized with the frequency of the stereo images to provide the user with stereo vision and depth perception of the virtual models.

In Fig. 2(a) the simulation algorithm is represented as a computer. The algorithm input is the position of the force feedback pen and the output is the updated graphical and haptic feedback. Basically, when the position of the pen is read, it is resolved whether the virtual spatula model has collided with the virtual brain model. If a collision has occurred, a deformation algorithm along with a finite element model (FEM) determine the correct deformation of the brain model. Hereafter, the user is provided with a graphical feedback showing the deformation and a haptic feedback rendering of the forces calculated by FEM and mapped onto the force feedback pen.

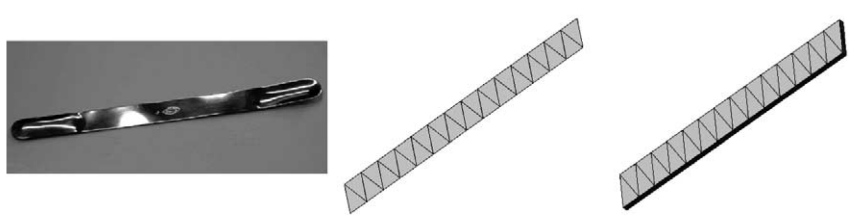
3. Model representation

The brain and spatula models have to be suited for an effective collision detection, collision response, physical modelling, geometric generality and graphical rendering. In the current implementation of the surgery simulator, the models are represented by triangles and tetrahedra.

Fig. 3 shows a photo of the spatula and how it is modelled physically and graphically. Spatulas are as shown in (a) flat and made of metal. Furthermore, it is possible to bend them into different forms making it possible to use them without blocking the surgeons view.

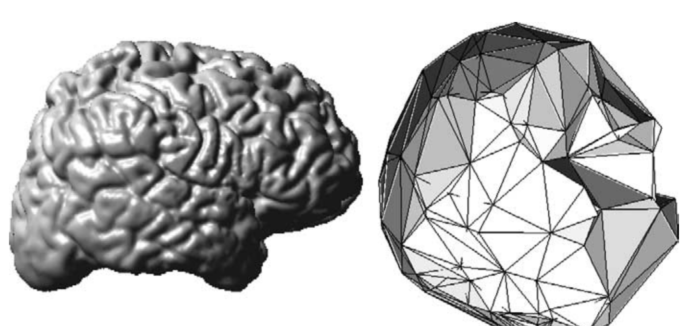
How the brain spatula is modelled physically by triangles is illustrated in Fig. 3(b). The spatula models used in the simulator are open surfaces; that is, the spatulas are not modelled volumetric as a closed surface around empty space. This approach is chosen to lower the total number of triangles in the physical modelling and thereby also the computational expense. Due to the fact that this approach can give graphical rendering problems with tissue hiding the spatula it has been decided to use another model for the graphical representation of the spatula as shown in (c). This is a closed surface model of the same spatula as shown in (b). The surface is made so that it brings a little thickness into the graphical representation of the spatula. None of the algorithms constrain the spatula to be modelled as a flat object. As mentioned earlier brain spatulas can be bent to suit the given situation. If the spatula has to be bent it is removed from the tissue bent and then re-applied, i.e. the spatula is not bent while retracting tissue. This bending property is addressed in the simulator by having pre-bent spatulas at hand. Brain spatulas are considered rigid compared to brain tissue.

The brain models are generated automatically from magnetic resonance imaging scans (MRI scans). This is



(a) (b) (c)

Fig. 3. The Spatula. (a) Photo of a spatula. (b) Physical model. (c) Graphical model.



(a) (b)

Fig. 4. The physical brain (a) and a virtual representation with 500 vertices (b).

done by segmenting the scans and then mapping them into a 3d tetrahedral mesh in such a way that the tetrahedra respect the surface topology and inner tissue type structures (Brix et al., 2001). Fig. 4 illustrates a model of the brain.

The tetrahedral mesh is used by the finite element models that model the physical properties of the tissue. The surface of the tetrahedral mesh is a triangulated surface which is used for collision detection, collision response and graphical rendering purposes.

4. Finite element modelling

In order to interact with the brain model it has to be deformable. A deformable model is a mathematical description of the geometry and the elasticity properties. We use a Finite Element Model of the brain to describe the relations between the displacement introduced by the spatula and the related deformations and haptical feedback.

The task of modelling a solid body of matter is referred to as a continuous problem (Cook et al., 1989). In a continuous problem the displacement variable contains an infinite number of values since it is a function of each generic point in the body. The finite element method reduces the problem to one of a finite number of unknowns by dividing the body into elements and by expressing the displacement field within the element in terms of assumed approximations. Fig. 5 shows the discretization of a solid body into a number of finite elements.

The tetrahedra are described by four nodes, where the displacement field is described by linear interpolation between nodes. In the finite element representation of a problem, the nodal displacement becomes the finite number of unknowns. Linear elasticity properties (Hooke's law) are used to define the relation between applied force $\boldsymbol{f}$ and the displacement $\boldsymbol{u}$. The elements must be assembled in order to find the properties of the overall system modelled by the mesh of elements.

The result of the assembly procedure is a large linear system of equations which normally is solved with respect to $\boldsymbol{u}$.

$$\boldsymbol{M}\ddot{\boldsymbol{u}} + \boldsymbol{C}\dot{\boldsymbol{u}} + \boldsymbol{K}\boldsymbol{u} = \boldsymbol{f}. \tag{1}$$

Eq. (1) represents a dynamic finite element model where $\boldsymbol{K}$ is the stiffness matrix defining the elasticity properties of the model, in the following only linear and isotropic elasticity will be considered. $\boldsymbol{M}$ is the mass matrix, $\boldsymbol{C}$ is the damping matrix, $\dot{\boldsymbol{u}}$ is the velocity and $\ddot{\boldsymbol{u}}$ is the acceleration. All matrices are of size $[3N \times 3N]$ and vectors are $[3N \times 1]$, where N is the number of nodes. Choosing the elasticity properties of the brain is important for the overall realism. Studies by Kyriacou et al. (2002) discuss finite element modelling and how to determine suitable elasticity properties mainly for accurate non-real-time modelling such as brain-shift. However, for this simulator we have chosen to let an experienced surgeon tune the elasticity properties. The dynamic model is solved for $\boldsymbol{u}$ using *the trapezoidal-rule method* to calculate the acceleration $\ddot{\boldsymbol{u}}$ and the velocity $\dot{\boldsymbol{u}}$. The *trapezoidal rule* is an implicit direct integration method which ensures that the solution will be unconditionally stable (Cook et al., 1989). Thus, the time step selection can be based on accuracy considerations alone.

The algorithm starts by determining an *effective stiffness matrix* and an *effective force vector* defined by Eqs. (2) and (3), respectively.

$$\boldsymbol{K}^{\mathrm{eff}} = \frac{4}{\triangle t^2}\boldsymbol{M} + \frac{2}{\triangle t}\boldsymbol{C} - \boldsymbol{K}, \tag{2}$$

$$\begin{aligned}\boldsymbol{f}^{\mathrm{eff}}_{n+1} = {} & \boldsymbol{f}_{n+1} + \boldsymbol{M}\left(\frac{4}{\triangle t^2}\boldsymbol{u}_n + \frac{4}{\triangle t}\dot{\boldsymbol{u}}_n + \ddot{\boldsymbol{u}}_n\right) \\ & + \boldsymbol{C}\left(\frac{2}{\triangle t}\boldsymbol{u}_n + \dot{\boldsymbol{u}}_n\right).\end{aligned} \tag{3}$$

Next step is to solve Eq. (4) with respect to $\boldsymbol{u}_{n+1}$.

$$\boldsymbol{K}^{\mathrm{eff}}\boldsymbol{u}_{n+1} = \boldsymbol{f}^{\mathrm{eff}}_{n+1}. \tag{4}$$

$\dot{\boldsymbol{u}}_{n+1}$ and $\ddot{\boldsymbol{u}}_{n+1}$ are updated for velocity and acceleration. $\dot{\boldsymbol{u}}_{n+1}$ and $\ddot{\boldsymbol{u}}_{n+1}$ are defined in Eqs. (5) and (6).

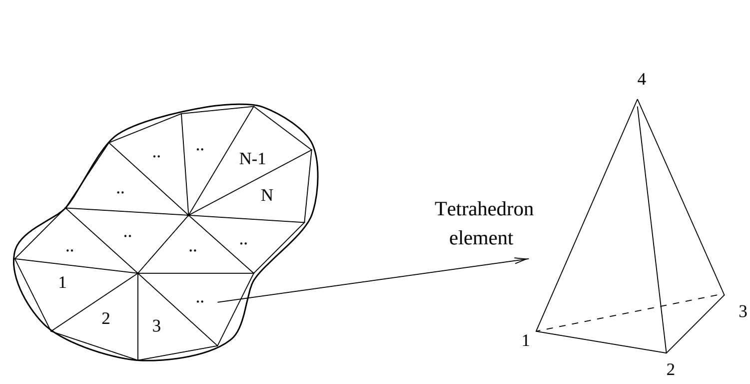


Fig. 5. Discretization of the continuum problem into discrete four-node tetrahedra.

$$\dot{\boldsymbol{u}}_{n+1} = \frac{2}{\Delta t}(\boldsymbol{u}_{n+1} - \boldsymbol{u}_n) - \dot{\boldsymbol{u}}_n, \tag{5}$$

$$\ddot{\boldsymbol{u}}_{n+1} = \frac{4}{\Delta t^2}(\boldsymbol{u}_{n+1} - \boldsymbol{u}_n) - \frac{4}{\Delta t}\dot{\boldsymbol{u}}_n - \ddot{\boldsymbol{u}}_n. \tag{6}$$

The challenge is to solve the equations in real-time and calculate both deformations and haptic feedback. The matrix equation system is solved using the iterative Gauss–Seidel Method (Barrett et al., 1994). This method allows for topological changes without new pre-calculation and it exploits the sparse structure of the effective stiffness matrix, hence is more memory efficient. The Gauss–Seidel method is shown as

$$\boldsymbol{u}_{n+1,i} = \left(\boldsymbol{f}^{\text{eff}}_{n+1} - \sum_{j<i} \boldsymbol{K}^{\text{eff}}_{i,j}\boldsymbol{u}_{n+1,j} - \sum_{j>i} \boldsymbol{K}^{\text{eff}}_{i,j}\boldsymbol{u}_{n,j}\right) \Big/ \boldsymbol{K}^{\text{eff}}_{i,j}. \tag{7}$$

Both haptic feedback and deformations are calculated. When the spatula collides with the brain model, some of the nodes are moved. The force needed to move these nodes is calculated, this is also the haptic-feedback that the user will feel during the simulation. These forces are then used to determine the remaining deformation of the model.

5. Collision detection

Basically, a collision detection algorithm has to perform a contact analysis where the aim is to resolve whether two or more objects in the scene have collided, and if so, where on the objects the collision has taken place. In the surgery simulator case, that is to find out whether the spatula has collided with the brain and if so which part of the spatula has collided with which part of the brain.

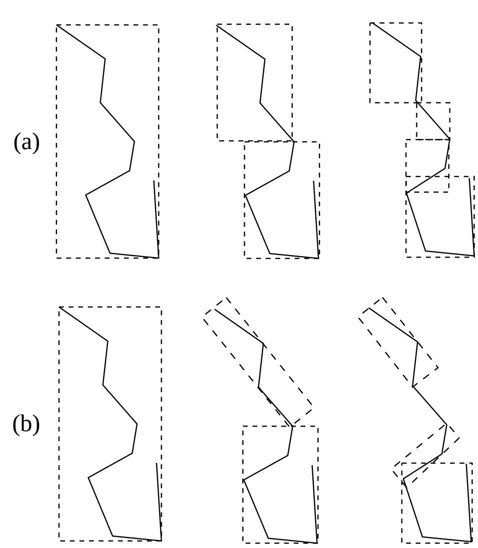


Fig. 6. Two types of BVHs. (a) The bounding volume geometry is axis-aligned bounding boxes. (b) The bounding volume geometry is object-oriented.

To meet the above-mentioned requirements and to obtain a collision detection algorithm that is computationally feasible the implemented detection algorithm is based on bounding volume hierarchies (BVHs) (Gottschalk et al., 1996; van den Bergen, 1997). The principle of BVHs is to build a simplified geometric representation of the objects that have to be checked for collision and then carry out the collision detection on these simple representations. Because of the simplicity of the BVHs, the collision detection can be carried out much faster than if collision between the original geometries should be detected.

Fig. 6 illustrates the two types of BVHs that are currently used in the surgery simulator. Each type of BVH is shown at three hierarchical stages: (a) A BVH where the bounding volumes are axis-aligned bounding boxes (AABBs). (b) A BVH where the bounding volumes are oriented bounding boxes (OBBs).

In the current implementation of BVHs, AABBs are used in connection with the brain geometry and OBBs are used in connection with the spatula geometry. The reason that AABBs are used for the brain is that BVHs based on AABBs are fast to build and more importantly they are fast to refit to geometry alterations (van den Bergen, 1997). This makes AABBs suitable for deformable geometries such as the brain. OBBs are used to model the rigid spatula because OBBs generally on an average fit any given geometry more tightly. Fitting a geometry tightly saves computational cost because it minimizes the number of occurrences where the bounding volumes have collided, but the original geometries have not collided (Gottschalk et al., 1996). This is illustrated in Fig. 7.

For the actual collision detection between the AABB-hierarchy and the OBB-hierarchy, Gottschalks separating axis test (SAT) has been implemented (Gottschalk, 2000). The collision test is carried out by recursively testing from the root of the hierarchy toward the leaves in both hierarchies. That is, first the roots are checked for collision and in the case of collision the children of the roots are checked. This procedure is carried out on each of the children until a collision cannot be found or until it is detected at leaf-level that a collision has occurred. So, the core problem in this way of collision detection is to find out whether two bounding volumes in each hierarchy have collided or not.

In the current implementation an exact collision detection at leaf-level is not performed. The consequence is that more collision detection triangles are passed on to the following algorithms of the surgery simulator. However, preliminary testing has revealed that passing on more collision triangles does not influence the overall computational time. The time spent

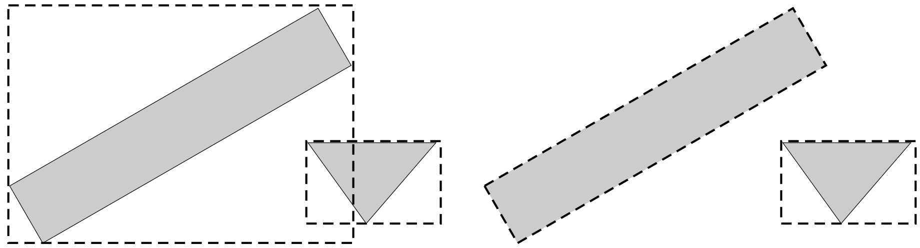


Fig. 7. Collision detection where dashed lines illustrate bounding boxes. Left: AABBs collide but there is no exact collision. Right: OBBs do not collide.

doing triangle–triangle test was about the same as just passing on all collisions and then letting the following algorithm address the problem.

6. Collision response

The collision response has the responsibility for using the information from the collision detection to generate a force vector to the force feedback arm and data to the visual feedback. To simplify the task it has been chosen to focus on a convex model of the brain. This makes it simpler to predict what tissue has been moved by the spatula and is not a restriction when it comes to training hand–eye coordination.

The collision response can be divided into three parts: Moving vertices locally, updating the model globally and calculating the force feedback arm response.

The first part of the collision response estimates which vertices on the brain surface have been hit and then move them according to the spatula position. Fig. 8(a) shows a fundamental problem in making this algorithm. Both the spatula and the brain surface are discrete and it is therefore not always possible to make a deformation of the brain that follows the spatula closely unless extra vertices are introduced as shown in Fig. 8(b). Extra vertices demand a restructuring of the matrices used by the finite element modelling and will therefore slow down the algorithm considerably.

The deformation used moves vertices that have been hit by the spatula but also partially moves the vertices close to the spatula as shown in Fig. 8(c). The distance they are moved is calculated according to the distance to the spatula plane. The new positions of the partially moved vertices are defined as

$$v_{\text{new}} = v_{\text{old}} + f(d) \cdot (v_{\text{spatula}} - v_{\text{old}}), \quad d < d_{\text{projection}}, \tag{8}$$

where v_{new} is the new position of the vertex, v_{old} is the position in the last timestep and v_{spatula} is the full projection of the vertex on the spatula plane, d is the distance to the spatula, $d_{\text{projection}}$ is the maximal distance to the spatula in which projections calculated and $f(d)$ is a factor that is dependent on the distance d.

It has been decided to use a simple linear relationship in this version of the system:

$$f(d) = 1 - (r \cdot d), \tag{9}$$

where r is a constant.

Obviously the value of r will influence the haptic feedback and thus the realism. Choosing a good value is not trivial and high realism will properly depend on the specific brain model and spatula. For this simulator, r was manually tuned with feedback from an experienced neurosurgeon. It is possible of course that a non-linear relationship could improve the realism.

To be able to use these formulas it is necessary to be able to define the following:

- which vertices have been hit by the spatula;
- which vertices are close to the spatula;
- the projection into the spatula plane.

The algorithm assumes that the spatula is moved from a position outside the brain. The translation vector is set to go from the last position of the spatula outside the brain to the new position as shown in Fig. 9. The position of the spatula in this context is given by a representative vertex on the spatula, in this case one near the tip. This will ensure that the used global translation

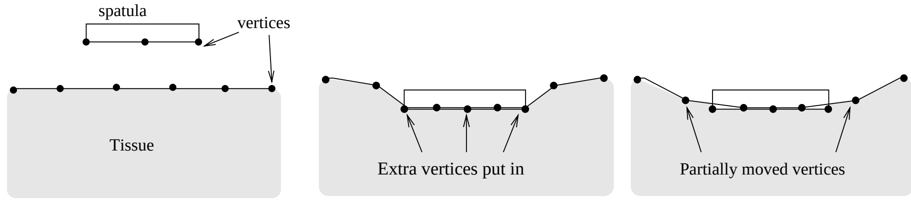


Fig. 8. Aspects of interaction between discrete surfaces (2d-representation).

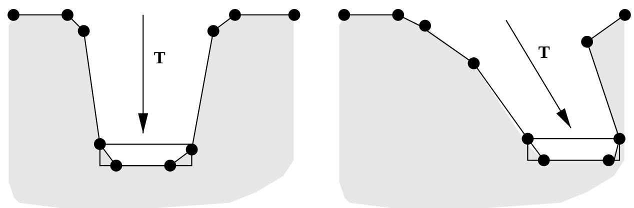


Fig. 9. The use of a global translation vector in deformations.

vector resembles the translation vector of the other vertices in the area closely.

Using a general translation vector makes it possible to map the spatula and the brain vertices into 2d before detecting whether a brain vertex is hit by a spatula triangle. As shown in Fig. 10 the space time movement of a triangle on the spatula defines a volume. If a brain vertex is inside this volume, it has been hit by the spatula.

Detecting whether a vertex is inside the volume is done by making a projection onto a 2d-plane translating the problem to whether a 2d-point is inside a triangle. The 2d-plane used is the plane that has the translation vector as normal. Each brain vertex visited is in this way tested up against the spatula triangles. If the vertex is not projected directly onto a spatula triangle, it has to be investigated whether the vertex is close to the spatula. This is done by calculating the distance in the 2d-plane to the spatula edge and using this as the value d in Eqs. (8) and (9).

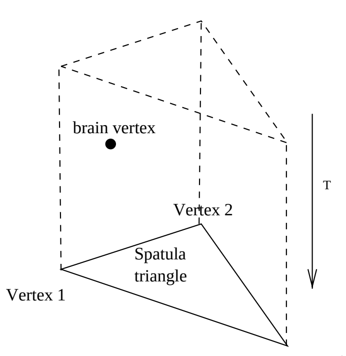


Fig. 10. The volume defined by the spatula triangle movement in space-time.

It is a time-consuming task to test all brain surface vertices up against all spatula triangles. Therefore, an iterative approach has been used to restrict the number of brain surface vertices visited. The approach starts from a triangle on the brain surface and tests the three corners up against the spatula triangles. Afterwards, it iteratively visits the neighbors of the triangle. The starting triangle is one of the triangles given by the collision detection. The iterations stop if the three corners of a given triangle are far away from the spatula defined by $d > d_{\mathrm{search}}$. To ensure that all hit areas are visited new iterations are begun from each triangle given by the collision detection that has not been visited in the last iterations.

Fig. 11(a) illustrates how $d_{\mathrm{projection}}$ and d_{search} define two areas around the 2d-projection of the spatula: the projection space in which vertices are partially projected and the search space in which triangles are iteratively visited.

The search space should be big enough to avoid triangles blocking the iterative algorithm. Fig. 11(b) shows an example of a triangle that is blocking the algorithm. The blocking triangle has three corners outside the search space and crosses the search space at the same time. If the iterations are started in the triangles in the bottom of the figure, the triangle will block the iteration which means that the triangles in the top are never visited. This should be avoided by ensuring that the search space is broader than the longest triangle edge on the brain surface. The forces given from the finite element model are the forces of every single vertex on the surface of the brain. The only forces needed are the forces of the vertices moved by the spatula. These are the points that the spatula will interact through. The force feedback device does not necessarily have the same form as the virtual spatula. This means that the tip of the spatula is not necessarily in the same position as the tip of the force feedback arm, i.e. the point upon which it is possible to use a force vector. What is common for the virtual and physical device is the handle. The user should feel like holding the virtual spatula while in reality he is holding the force feedback pen.

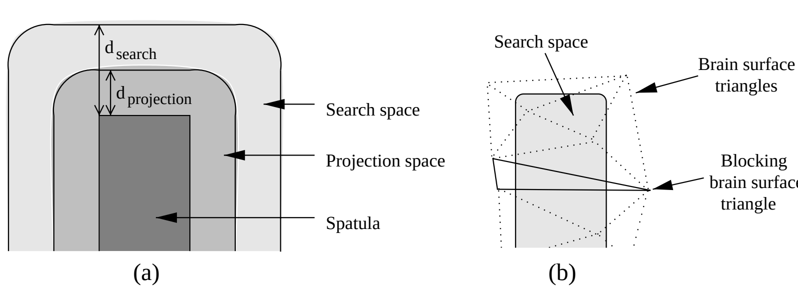


Fig. 11. Sketch of the spaces used by the local deformation.

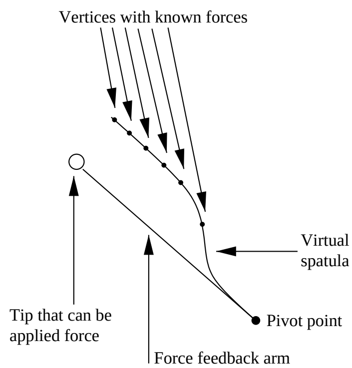


Fig. 12. The virtual spatula and the force feedback arm.

The individual forces of the moved vertices of the virtual spatula have to be transformed to a single force on the force feedback arm. The transformation is achieved by assuming that the user only holds the spatula in a single pivot point as shown in Fig. 12. In this way the force felt can be split into two parts: the rotational force felt when the tip of the pen is moved perpendicularly to the handle and the push/pull force felt when the tip of the pen is moved parallel with the handle as shown in Fig. 13. It is possible to address these forces individually.

7. Experimental results

The performance of the system to simulate spatula interaction with soft tissue was tested. We aimed at measuring the haptic response, the visual feedback and the execution time of the simulation which were important for both the haptic and the visual experience of the system.

7.1. Haptic response

The haptic feedback can only be calculated when a node is moved. This implies that larger spatulas are able to move more nodes and thereby produce further haptic feedback than the smaller spatulas, but as the deformable model is discrete, different sizes of spatulas might cover the same amount of nodes in some situations.

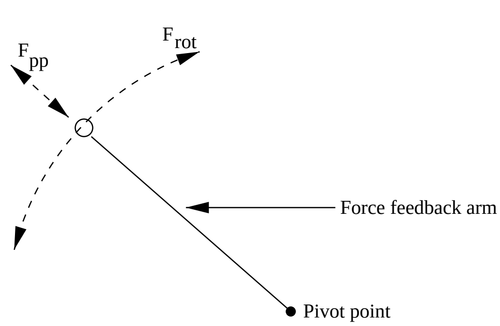


Fig. 13. Rotational and push/pull forces of force feedback arm.

When a spatula was pressed onto a deformable model the amount of haptic feedback only depended on how many nodes it touched and not the area of contact. The collision response algorithm could however compensate for this abnormal behavior. It is important that the haptic response is smooth without any sudden changes as they will be perceived as small jerks in the force-feedback arm and can significantly decrease the feeling of touching real tissue. In this test, the size of the spatula was increased which was supposed to have an increasing effect on the haptic feedback. Fig. 14 illustrates how the experimental test was carried out.

As shown in Fig. 14, the spatula was shaped as a square that had been applied orthogonally onto the surface of the soft tissue model, which was shaped as a cube. 50 different spatulas with the areas pf $0.005,\ 0.01,\ 0.015, \ldots, 0.25\ \mathrm{m}^2$ were used on the test. For comparison the cube surface, on which the spatula was applied, was $0.3\ \mathrm{m}^2$. The resolution of the surface was 15×15 squares each with an area of $0.02\ \mathrm{m}^2$. Fig. 15 illustrates the force feedback as a function of the spatula size. As the deformation depth was the same for each spatula size, the increasing force response seen in the graph was a result of the increasing spatula size. The force feedback shown was the sum of translational and rotational forces. As the spatula was applied orthogo-

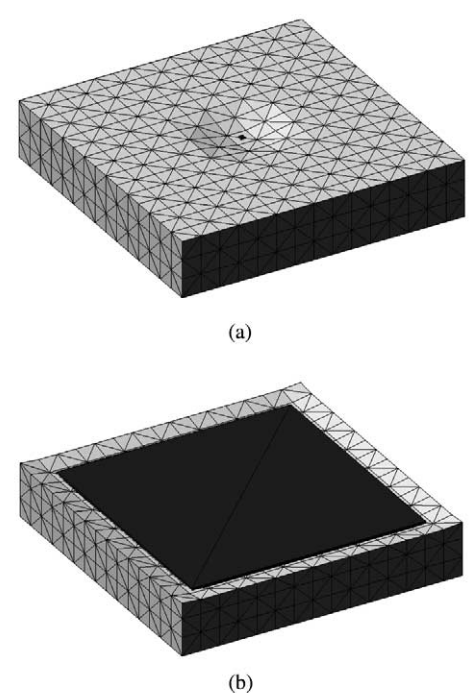


Fig. 14. Two of the 50 test cases carried out to examine the haptic response. (a) The smallest spatula used was applied orthogonally onto the synthetic brain model. (b) The largest spatula used was applied in the same way onto the model.

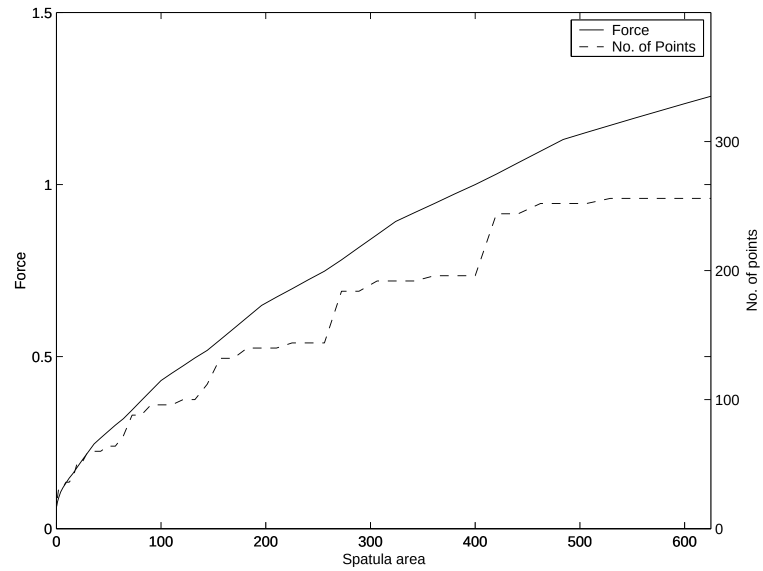


Fig. 15. Plot of the force and number of points moved as a function of the spatula size.

nally onto the surface, only forces along the z-axis could be measured. All other forces were zero.

The smooth appearance of the force feedback seen in Fig. 15 is worth noticing. Even though the resolution of the surface of the cube was rather low, no stepwise tendencies were found in the force feedback. Although the size of the spatula increased and in some situations suddenly covered more points on the surface, it could not be measured in the force feedback. This was a direct effect of the partial deformation which smoothened the discretization in this way. The dashed line is the sum of partial and full projected points. This line also had a stepwise tendency and the steps were defined by the steps in the partially projected points. The reason these steps could not be measured in the force feedback was that the partially projected points contributed with a force related to the distance to the spatula edge. When a step occurred, the points were relatively far away from the spatula edge and therefore contributed with a low force.

7.2. Algorithm performance

The execution time of the algorithm was important for the haptic and visual experience of the system. The purpose of this test was to find the execution time of

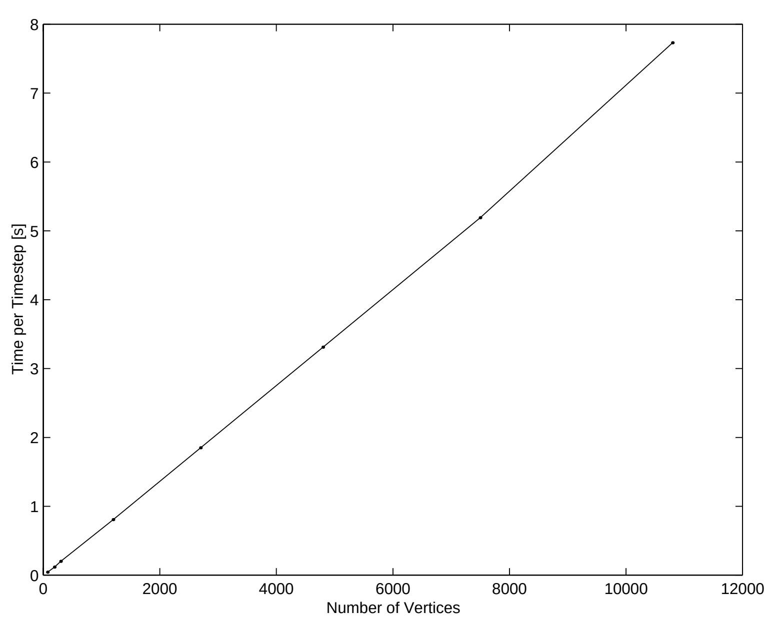


Fig. 16. Relation between the number of points in the brain model and the mean time per simulation loop.

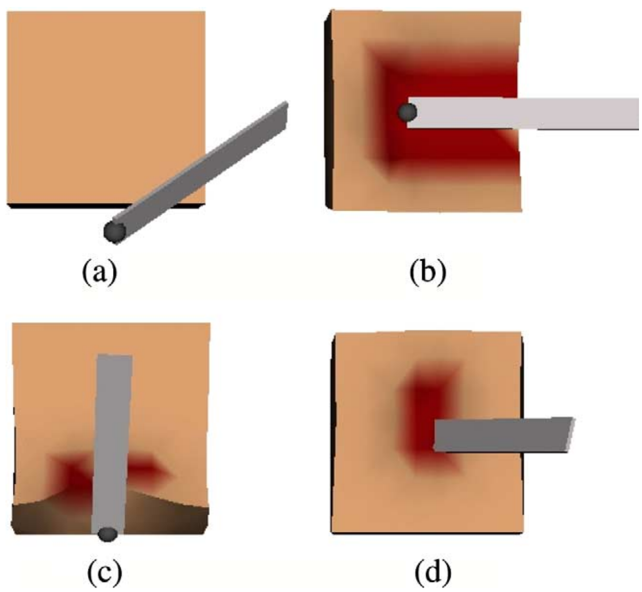


Fig. 17. Screen shots of the running simulator, model size was 192 vertices. (a) Undeformed cube and spatula. (b) Cube pushed by spatula. (c) Side moved by spatula. (d) Cube touched by spatula tip. (This figure is available in colour, see the on-line version.)

each simulation loop dependent on the number of vertices in the brain model. The information was useful when finding a trade-off between the detail of the model and the sampling interval of the algorithm.

The models used in the tests were cubes like the ones shown in Fig. 14. They were all of the same size, but the number of points were set differently to reflect the same cube with a different sampling interval. The cube sizes used were $5 \times 5 \times 3$, $8 \times 8 \times 3$, $10 \times 10 \times 3$, $20 \times 20 \times 3$, $30 \times 30 \times 3$, $40 \times 40 \times 3$, $50 \times 50 \times 3$ and $60 \times 60 \times 3$ ranging from 75 to 10,800 points.

In the test, a spatula was used to create a continuous push of the cube to provoke the worst-case situation for the algorithm. The total time used was divided by the number of loops to get the mean time for each loop. The machine used for the test was a Silicon Graphics Onyx2 dual CPU 195 MHz with 512 MB RAM.

In Fig. 16, the results showed a strong linear relation between the number of points and the time used for each loop. Furthermore, the results indicated that increased processing power would allow equally increased level of detail of the model.

7.3. Graphical representation

The purpose of this test was to show the visual result of the spatula/brain interaction and how the deformation of the brain looked when a spatula pressure was added. Furthermore, the distribution of the pressure added by the spatula was illustrated by a threshold color coding, obtained by coloring all vertices of the brain where the pressure was higher than the threshold.

The test had been split up into two parts: (1) One showing the screen shots from a running simulation, and (2) another one showing a higher resolution model calculated offline.

Fig. 17 shows screen shots of the running simulation with different situations.

When a cube was used it was easy to control both the number of total vertices and the number of vertices on the surface of the model. A better visual appearance was obtained by ensuring that the number of vertices on the surface was high in relation to the number of vertices inside the model. The color coding of the regions with the highest force illustrated the pressure of the spatula. With a proper threshold value, it should be possible to use the color coding for indicating where the pressure of the spatula is at risk of damaging the tissue. In this way, it could serve as a feedback to the person using the simulator helping him or her to get an intuitive feeling of

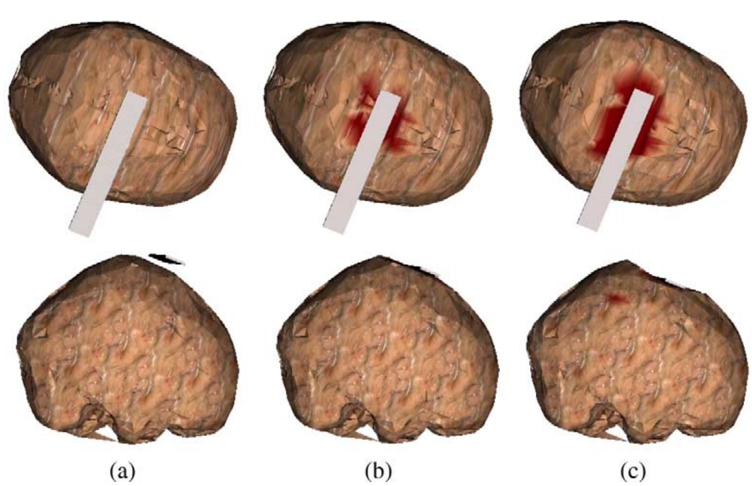


Fig. 18. High resolution model shown from top and side. (a) Undeformed model. (b) Model slightly touched. (c) Model firmly touched. (This figure is available in colour, see the on-line version.)

an adequate and safe amount of pressure. Fig. 18 shows a more detailed model in three situations all shown from two viewpoints.

The above model contained 10,000 vertices and was thereby a significantly more detailed model. This level of detail was shown to give an indication of the future capabilities of the system, but is not at the moment a realistic model in a real-time scenario. The model was autogenerated and therefore less regular than the cube. A texture was added to improve the impression of looking at a brain.

The three different levels of pressure showed that the red-colored area increased with a higher pressure. This was not unexpected as it reflected that the spatula added more pressure over a larger area of the brain model.

8. Conclusion

Currently, the development of the simulator is at a stage where the basic interaction between a virtual brain spatula and a virtual human brain is possible. A number of people have tested the simulator in order to evaluate qualitatively the overall simulator performance. The results are promising, showing the correlation between spatula size, force feedback and visual feedback feels and looks real.

However, the simulator is not yet at a level where it can be useful for surgical training on a larger scale. The details of the models and the refresh rate of the simulation are not sufficient enough. Further testing might indicate that more advanced modelling must be applied such as non-linear elasticity, more advanced collision response, or simply better hardware. The hardware utilized constrains to some extent the realism of the simulator and thus the overall impression in a qualitative test. It is of importance that the hardware used for the force feedback is limited to return a force vector with 3 degrees of freedom. If a torque could be induced on the force feedback pen, a more realistic haptic impression could be provided for.

Furthermore, the level of detail in the virtual brain model is maintained low to keep the refresh frequencies of the visual and haptic feedback at a level where the user is provided with a smooth simulation experience. Naturally, the freedom to choose a higher level of detail would give the simulator a higher degree of realism and the user an increased overall experience. The color coding of the high pressure areas seems suitable when teaching how spatula pressure should be applied. The results are preliminary and further developing and testing of the simulator will quantify its value as a surgical teaching device.

In spite of the constraints caused by the available hardware, the overall simulation experience is promising. The current design and models hold a potential and scalability to provide even better simulation results when increased processing power is applied.

References

Andrews, R.J., Bringas, J.R., 1993. A review of brain retraction and recommendations for minimizing intraoperative brain injury. Neurosurgery 33, 1052–1064.

Barrett, R., Berry, M., Chan, T.F., Demmel, J., Donato, J., Dongarra, J., Eijkhout, V., Pozo, R., Romine, C., der Vorst, H.V., 1994. Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods, second ed. SIAM, Philadelphia, PA.

Brix, L., Fischer, C., Hansen, K.V., Larsen, O.V., 2001. Automatic mesh generation of the human brain for use in a neurosurgical simulator. In: Proceedings of the International Federation for Medical and Biological Engineering, IFMBE, IX Mediterranean Conference on Medical and Biological Engineering and Computing, Medicon 2001, Pula, Croatia 2, pp. 978–982.

Bro-Nielsen, M., 1998. Finite element modeling in surgery simulation. Journal of the IEEE 86 (3), 490–503.

Brown, J., Montgomery, K., Latombe, J.-C., Stephanides, M., 2001. A microsurgery simulation system. In: Medical Image Computing and Computer-assisted Intervention, pp. 137–144.

Cook, R.D., Malkus, D.S., Plesha, M.E., 1989. Concepts and Applications of Finite Element Analysis. John Wiley & Sons.

Cotin, S., Delingette, H., Clement, J.-M., Tassetti, V., Marescaux, J., Ayache, N., 1996. Volumetric deformable models for simulation of laparoscopic surgery. In: Proceedings of the International Symposium on Computer and Communication Systems for Image Guided Diagnosis and Therapy, Computer Assisted Radiology (CAR'96). International Congress Series, Vol. 1124. Elsevier.

Cotin, S., Delingette, H., Ayache, N., 2000. A hybrid elastic model allowing real-time cutting, deformations and force-feedback for surgery training and simulation. The Visual Computer 16 (8), 437–452.

Gottschalk, S., 2000. Collision queries using oriented bounding boxes. Ph.D. thesis, Department of Computer Science, UNC Chapel Hill.

Gottschalk, S., Lin, M.C., Manocha, D., 1996. Obbtree: a hierarchical structure for rapid interference detection. In: Computer Graphics Proceedings, Siggraph.

Kühnapfel, U., Cakmak, H.K., Maass, H., 2000. Endoscopic surgery training using virtual reality and deformable tissue simulation. In: Computers & Graphics, vol. 24, pp. 671–682.

Kyriacou, S.K., Mohamed, A., Miller, K., Neff, S., 2002. Brain mechanics for neurosurgery: Modeling issues. Biomechanics and Modelling in Mechanology 1 (2), 151–164.

Poppen, J.L., 1960. An Atlas of Neurosurgical Techniques. W.B. Saunders Company, Philadelphia and London.

Rosenørn, J., Diemer, N., 1987. The influence of the profile of brain retractors on regional cerebral blood flow in the rat. Acta Neurochirurgica 87, 140–143.

van den Bergen, G., 1997. Efficient collision detection of complex deformable models using aabb trees. In: Journal of Graphics Tools, vol. 2, pp. 1–14.

Yasargil, M., 1999. A legacy of microneurosurgery: Memoirs, lessons and axioms. Neurosurgery 45, 1025–1091.