ROBOMECH Journal
a SpringerOpen Journal

RESEARCH ARTICLE Open Access

Proposition and evaluation of a collision detection method for real time surgery simulation of opening a brain fissure

Akira Fukuhara[1*], Teppei Tsujita[1,2], Kazuya Sase[3], Atsushi Konno[3], Xin Jiang[1], Satoko Abiko[1] and Masaru Uchiyama[1]

Abstract

In neurosurgery, opening a brain fissure allows an operator to access to the basis cranii and to treat a disease part safely, but the operator is required highly sophisticated techniques. In order to develop a neurosurgery simulator enabling surgeons to train themselves for opening a brain fissure procedure, it is required to detect collision between a thin instrument model and a deformable brain model and furthermore to compute feasible enforced displacements to deformable object. This paper proposes a new collision detection method which considers to the collision state of objects in previous loop of simulation. And simulation of opening a brain fissure is implemented using the proposed method which is accelerated by GPU (Graphics Processing Unit). Finally the performance of the proposed method is evaluated.

Keywords: Collision detection; Operation simulator; Software development; ROBOMEC2013

Background

Neurosurgery simulator

In recent years, due to the improvement of safety awareness and significant advances in medical technology, doctors have to spend more time in training themselves to learn up-to-date surgical techniques. In the preoperative training, surgical models and animals are used in spite of less reality or high cost. Surgical simulators based on Virtual Reality (VR) technology and haptic display offer high quality and cost-effective training. The surgical simulators can use a patient specific model and reproduce surgical situation repetitively. The surgical simulator benefits not only training but also preoperative efficient planning and have been actively studied in neurosurgery that requires particularly sophisticated techniques. NeuroTouch [1] provides training of tumor-debulking task and tumor caterization task. This simulator is introduced for neurosurgery training curriculums in Canada. By using this simulator, a novice doctor can learn surgery techniques for tumors in ensured view of the task-specific operating field. However, this simulator uses models of only the task-specific operating field to meet real-time computing requirement. ImmersiveTouch [2] is another surgical simulator which can be used for training to insert a neuroendoscope into a brain. This simulator supports the operator with a Computer Graphics (CG) view of internal objects in a skull. Using ImmersiveTouch surgeons can learn how to handle lesions located deep in the brain unlike in the case of NeuroTouch, but the surgical procedure which can be simulated by this simulator is limited to needle (neuroendoscope) insertion. These neurosurgery simulators to exist provide training environments which can display sight of operation or a perspective CG view. However, the operator needs to approach lesions located deep in the brain and ensure view of operation by opening a brain fissure using spatulas in some cases. During approaching, avoidance of any injury to a brain is essential for successful operation and difficult because internal tissues are composed intricately. It is necessary to develop neurosurgery simulator that realizes training to approach the depths of a brain by using a whole brain model.

*Correspondence: fukuhara@space.mech.tohoku.ac.jp
[1] Tohoku University, Sendai, Japan
Full list of author information is available at the end of the article

Opening a brain fissure

Brain fissures form deep valleys, and the Sylvian fissure and the inter-hemispheric fissure lead to the depths of the brain.

There are many combinations of brain fissures and the start position of the operation, but the standard flow of exposing lesion area is presented as Figure 1. There are layer structures between the skull bones to the brain, so some matter layers are partly removed in order to approach the depths of the brain.

Pterional approach is one of the techniques for approaching the depth through the Sylvian fissure [3]. By cutting arachnoid and connective tissues, the operator enters the shallow part of the brain fissure. To approach the deep part, the operator is required to assure states of internal structures by exclusion of brain tissues using instruments and to release restrictions by cutting connective tissues. Exclusion of brain fissures with excessive force and low visibility of internal structures may lead to damages to brain tissues. A novice doctor needs to acquire the suitable force for exclusion of brain tissues through training. Figure 2 shows a spatula inserted to the Sylvian fissure. To develop a simulator with which operators can learn how to open a brain fissure, it is required to detect collision between a non-convex deformable object and a thin rigid object and to give suitable enforced displacements for the deformable object. When a thin object penetrates a deformable object deeply, enforced displacements computed by a collision detection algorithm, based on only geometric relationship between colliding objects,

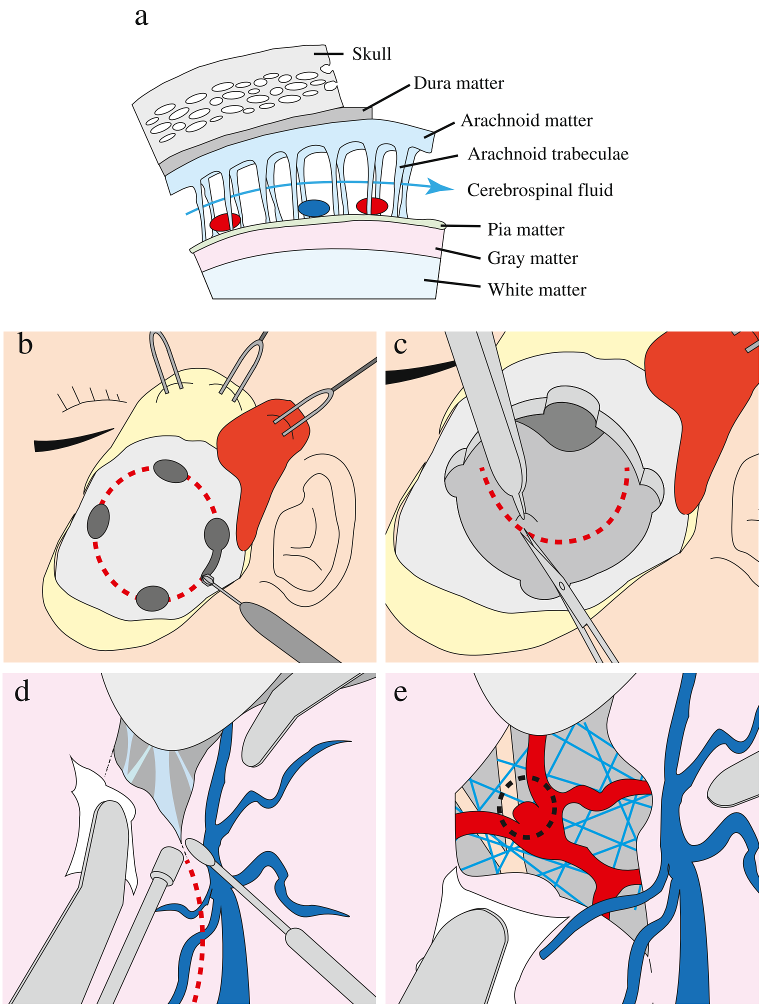

Figure 1 The surgical procedure of approaching to the depths of the brain [3]. (a) A layered structure from the skull bone to the brain. In order to expose a lesion area located in the brain, it is required to path through those layers. **(b)** Opening the skull. After dissection of soft tissues (skins and muscles), the skull bone layer is exposed. Some drill holes are made so that high-speed drill can cut the skull in curvilinear fashion. **(c)** Opening the dura matter. The arcuate opening can be made in the dura matter layer. **(d)** Cutting the arachoid matter. The arachoid matter and trabeculae fix brain and internal structures (nerves and veins), so those connective tissues should be removed to retract the brain. **(e)** Exposing the lesion area. Due to spatula pressure, Retraction the brain has potential dangers to injure to adjacent brain regions.

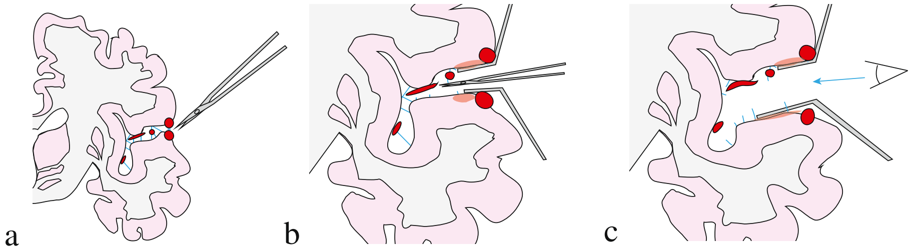

Figure 2 The procedure of opening a brain fissure. (a) A coronal section of a brain is shown. Connective tissues are cut to release restrictions. **(b)** A spatula is inserted into the Sylvian brain fissure. In this situation, the spatula contacts with complex inner structures of the fissure. **(c)** Deformation of the brain due to exclusion by the spatulas. Opening the brain fissure allows an operator to access hided parts of the brain and the depths.

are sometimes infeasible because the depth of penetration does not always match with suitable enforced displacements for realistic deformation.

Collision detection

Collision detection is the computational problem of detecting the intersection of objects and plays an important role in realizing interactive environments in physical simulation and video games [4]. In robotics, collision detection is also a major topic to calculate time of impact and generate a path to avoid collision [5]. In this paper, collision detection refers to detecting the intersection between a deformable object and a rigid object and handling responses of intersection of those objects. Collision detection for deformable models realizes many interactive simulations like a surgical simulation [1,2,6,7]. Many methods of collision detection for rigid and deformable bodies have been proposed. Collision detection can be categorized with Discrete Collision Detection (DCD) and Continuous Collision Detection (CCD). Here we describe related works of DCD for deformable objects with some different methods and CCD that is recently studied intensively.

Discrete collision detection

Surface-based method One of major methods is based on intersection between surface elements. To accelerate collision detection, some data structures (e.g., Axis Aligned Bounding Box (AABB) tree, hashing, and subdivision space) are used and decrease the number of intersection test of primitive shapes as following methods [8,9]. This method is based on intersection of surface elements. Elements which penetrate other objects cannot be searched by intersection surface elements as shown in Figure 3a. In this case, natural reactions for contacts are difficult.

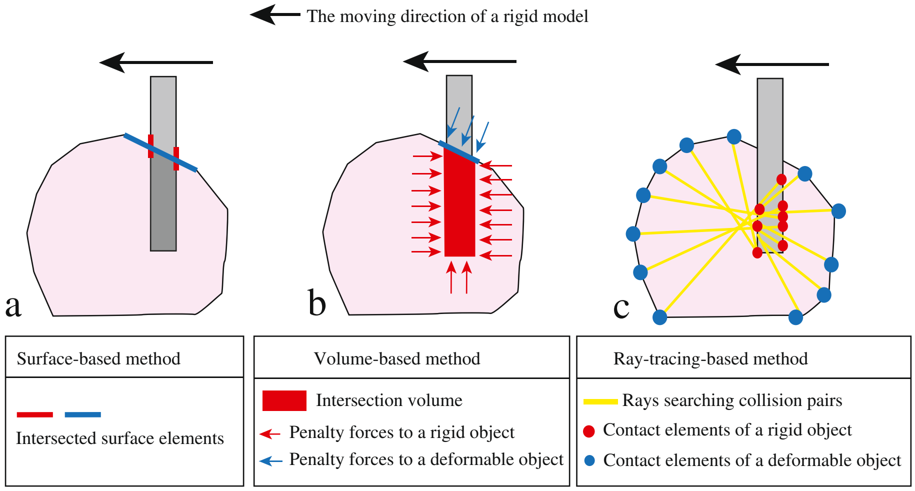
The moving direction of a rigid model
a
b
c
Surface-based method
Intersected surface elements
Volume-based method
Intersection volume
Penalty forces to a rigid object
Penalty forces to a deformable object
Ray-tracing-based method
Rays searching collision pairs
Contact elements of a rigid object
Contact elements of a deformable object

Figure 3 The previous collision detection methods. (a) Surface-based method can only detect intersection between surfaces. **(b)** Volume-based method can detect penetration between objects. The reaction algorithm based depth of the intersection is not suitable for opening a brain fissure. **(c)** Ray-tracing-method also can detect collision in deep penetration between objects. However this method cannot detect whether the collision pairs are feasible or not.

In the neurosurgery simulation, thin surgical instrument models penetrate a deformable model. Then the surface-based method is not suitable.

Volume-based method Other major methods are based on intersection of the internal meshes of objects [10,11]. Compared with surface-based method, this method can detect deep penetration between objects. The volume-based method can be classified roughly into image-based collision detection and intersection-based collision detection. Layered Depth Image (LDI) is a kind of image-based collision detection [12,13]. This method can be accelerated with graphic hardware and is adopted in real time interactive simulators [10]. LDI can detect fast collision between complex models. When objects penetrate each other deeply, LDI detects the contact volume as shown in Figure 3b. However the algorithm of response for collision detection is based on only geometrical relationship between models, therefore cannot handle correctly neither enforced displacements nor reaction forces. Other volume-based methods detect collision by intersection between internal meshes of objects [11,14]. In this method, depth of penetration between objects is calculated and used for displacement for intersected objects. In mesh-to-mesh collision detection, volume-based method sometimes lead to inconsistent direction of penetration depth. Heidelberger et al. cleared this problem with direction detected by intersection of surface [11]. However, in neurosurgery, direction of penetration depth does not always match with direction of ideal enforced displacements.

Ray-tracing-based method Ray-tracing-based method is one of volume-based methods. This method often uses surface data of an object and casts rays form a vertex of the surface meshes or internal meshes. This method also can detect intersection between objects with less tunneling [15,16]. However displacement is sometimes infeasible because of difficulty in choosing feasible collision pairs.

Continuous collision detection

While DCD detects collision between objects at discrete time, CCD targets collision during a time step, and can compute the first time of collision and the contact point. CCD was proposed for collision between rigid objects [17,18]. CCD is very high cost for deformable model because a large amount of elemental tests are needed. Recently many culling methods are studied and contribute to reduce computing time of CCD between deformable models [19,20]. CCD can detect self-collision of a deformable object and is applied in clothes-simulation [21,22]. Compared with DCD, CCD is robust for tunneling and can compute suitable enforced displacements for simulation of deformation [23]. However, computing cost of CCD is larger than DCD and high speed test are required in an interactive simulator like a surgery simulator. Tang et al. showed a result of simulation that a liver is cut during a surgical operation [23]. In this simulation, the model consists of 4338 tetrahedra, and the average CCD query time is 53.2 [ms] per frame. They achieved more than 10 times improvement over prior methods. This CCD query speed is not enough high because the scale of the brain model is larger and the required query time is less than 30 [ms] for real time simulation.

A real time simulator needs the contact state between objects for reactions and high speed DCD that performs test at small discrete time is suitable. DCD detects collision and enforced displacements using a geometrical relationship between objects. However enforced displacement as a result of collision detection sometimes cannot be detected with only the geometrical relationship like the depth of penetration. This paper proposes a DCD method realizing suitable deformation for opening a brain fissure.

Method

Overview of the proposed method

In this paper, collision detection problem between a deformable object and a rigid object is solved by performing two tests as shown in Figure 4. First test is to search pairs of elements which represent collision condition between objects by using geometrical relationship. Second test is to cull from the pairs to realize realistic collision response by using history of collisions and this part is one of the characteristics of the proposed method. Up to the first test, there is not major difference between the previous DCD method and the proposed method, with respect to detecting colliding elements. In the proposed method, the second test removes false positive collision pairs from the result of the first test by using history of collisions. Originality of the proposed method is removing false positive collision pairs by using history of collisions. However, the additional test leads to increasing computational costs.

Subsections from ‘Objects of collision detection’ to ‘Deformation with collision pairs’ explain how the proposed method detects collisions and how the results of the test are used for deformation of the brain. Then subsection ‘Preparation for culling collision pairs’ and ‘Culling collision pairs using history’ explain how removes false positive collision pairs by using history of collisions. Subsection ‘Merging the two intersection test’ and ‘Acceleration with GPGPU’ explain how the computational costs are decreased and how the computational speed is accelerated.

Objects of collision detection

Proposed method targets thin instrument models like a spatula (spatula model) and a deformable brain model.

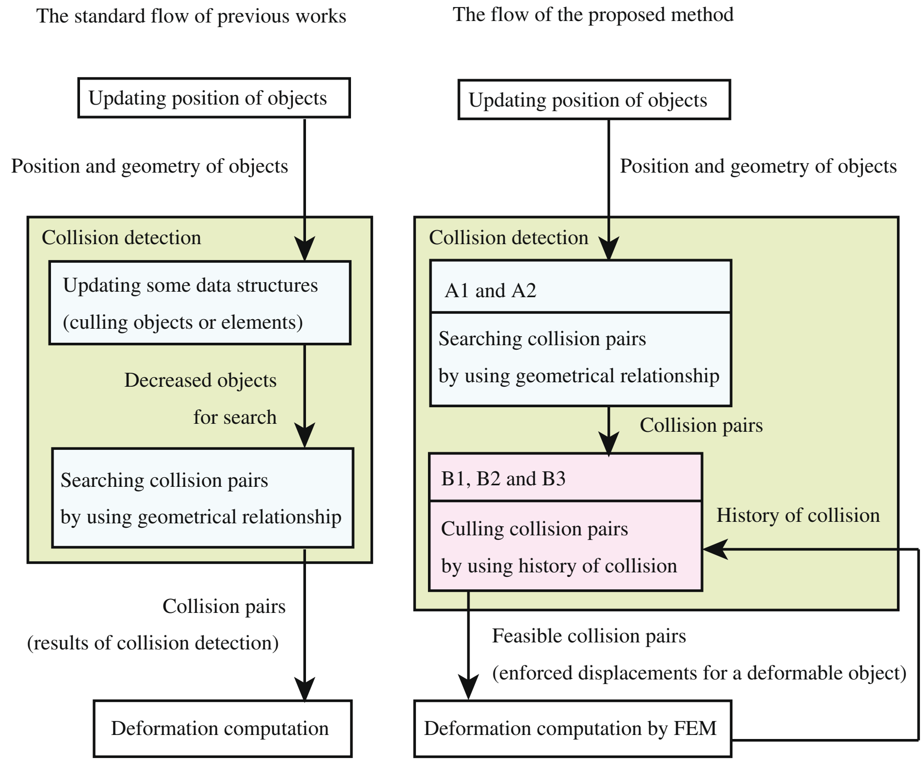
The standard flow of previous works
The flow of the proposed method
Updating position of objects
Position and geometry of objects
Collision detection
Updating some data structures
(culling objects or elements)
Decreased objects
for search
Searching collision pairs
by using geometrical relationship
Collision pairs
(results of collision detection)
Deformation computation
Updating position of objects
Position and geometry of objects
Collision detection
A1 and A2
Searching collision pairs
by using geometrical relationship
Collision pairs
B1, B2 and B3
Culling collision pairs
by using history of collision
History of collision
Feasible collision pairs
(enforced displacements for a deformable object)
Deformation computation by FEM

Figure 4 Overview of the proposed method. The proposed method uses geometrical relationship between objects and history of collisions. While previous works use culling to reduce tests of collision detection, the proposed method use culling to find feasible collision pairs for a deformable object.

In the process of opening a brain fissure, the brain is retracted with two large planes of a spatula. In order to perform the collision detection fast, side planes of the spatula model are excluded from colliding planes. Thus, the algorithm can only detect collision on the largest two planes of the spatula.

To detect collision between the spatula model and the brain model, nodes are discretely located on the two parallel planes of the spatula model as shown in Figure 5a. Distance between a node and the other is smaller than the length of surface triangles of the brain model. For a fundamental study of opening a brain fissure with spatulas, the shape of the model is similar to the shape of telencephalon having some brain fissures as shown in Figure 5c. This model is made by scanning a commonly-available anatomical brain model (Brain Model C20, 3B Scientific GmbH). The brain model consists of cerebral parenchyma and has no nerves and no blood vessels. To calculate deformation of the brain model with Finite Element Method (FEM), the inside of brain model is divided with tetrahedron and the surface of the brain model consists of triangle elements.

Collision pair

What is collision pair Collision pairs are results of collision detection and present which elements contact each other. The collision pair found by the proposed method is combination of a node of the spatula model and a surface triangle of the brain model. Collision pairs are used for detection of enforced displacements for the deformable brain model.

Feasible or infeasible collision pair Collision pairs are categorized as feasible pair or infeasible pair for realistic deformation. Whether a collision pair is feasible or not is detected the pair is suitable for deformation in the context of collision. In Figure 6a, for example, if the spatula model moves right and penetrate the brain model, the right plane of the spatula contacts firstly and gives enforced displacements to the brain model. Then the collision pairs on right side of the spatula are feasible and the pairs on left side are infeasible. In contrast, the collision pairs detected on the left plane are infeasible because the left plane will not contact. The property of collision pair is detected by not only geometrical relation but also the context of collision.

If all collision pairs are used as displacement for the brain model, infeasible deformation will occur especially in the case where the spatula model penetrates the brain model deeply as shown in Figure 6b.

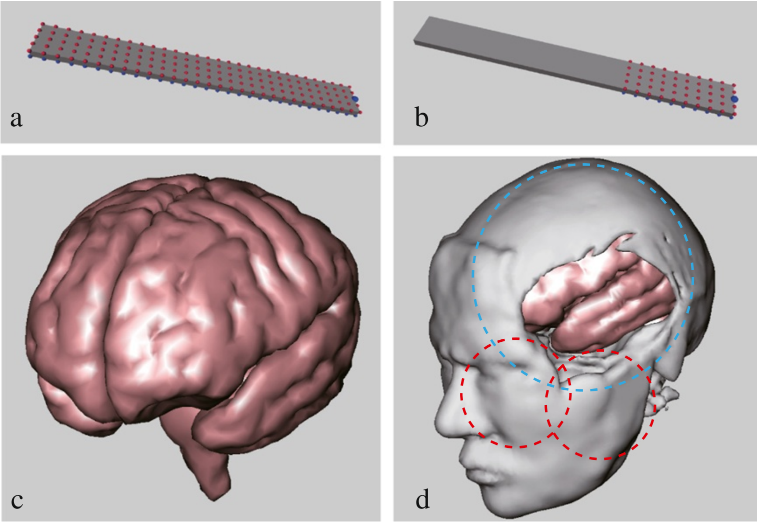

Figure 5 Geometrical Models in a brain surgery simulation. (a) A spatula model. 150 source nodes are located each collision plane. **(b)** 30 source nodes are located intensively on an anterior part of the spatula. **(c)** A brain model which is used in comparative simulation with-and-without the culling collision pairs. This model consists of 9,126 nodes, 34,401 tetrahedron elements and 12,608 surface triangles. **(d)** A brain model and a head model which are used in simulation of opening a brain fissure. This model consists of 12,331 nodes, 48,316 tetrahedron elements and 15,865 surface triangles. Collision detection between inside of the skull and the brain model are performed based on an approximation that a sphere represents a shape of the head. A blue sphere contains the brain model and red sphere excludes the brain model instead of a skull.

Handling pair To realize feasible deformation in deep penetration between models, the collision pair needs to be tested to see whether the intended pair is feasible or not. The geometric relationship between models and direction of moving model is not sufficient for testing which pair should be used for displacement in deep penetration. The proposed method can detect collision between deeply penetrated models and decide which pairs are feasible for deformation through two tests. In the first test, collision pairs are found with a ray-tracing-based method. In the second test, a history of the collision state finds infeasible pairs and displacement for the brain model is decided from feasible pairs.

Ray-triangle intersection

In the proposed method, a ray-triangle intersection test is used to find feasible and infeasible collision pairs. In the ray-triangle intersection, we adopt the Tomas Möller method for its quickness [24]. As a result of the intersection test, the distance t from the origin of the ray to the plane in which the triangle lies and barycentric coordinates (u,v) are given as shown in Figure 7. A ray $\boldsymbol{R}(t)$

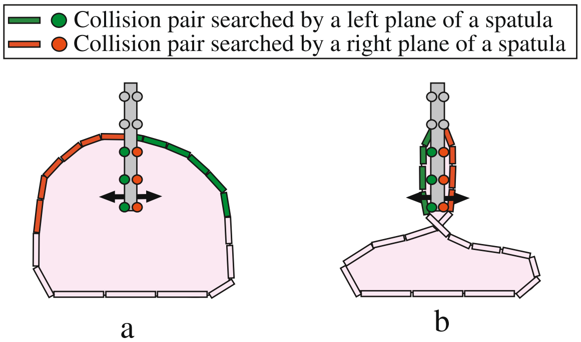
🟩 ● Collision pair searched by a left plane of a spatula
🟥 ● Collision pair searched by a right plane of a spatula
a
b

Figure 6 Unrealistic deformation [26]. (a) Two groups of collision pairs. If a spatula moves right and penetrate a brain, collision pairs on the right plane are feasible. From only geometrical information, it is difficult to decide which collision pairs should be used as enforced displacements. **(b)** Given all pairs as displacement, the brain deforms unrealistically.

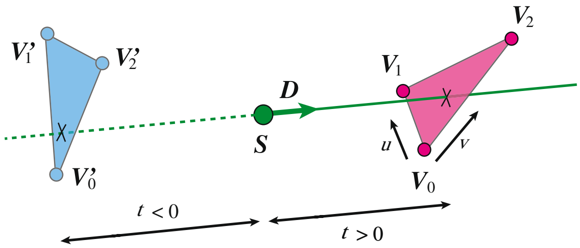

Figure 7 Ray-triangle intersection [24]. A ray is casted from a source node *S* with the direction of *D*. The right triangle in this figure intersects with ray and positive *t* is obtained. The left triangle also intersects with ray and *t* is obtained. The result of ray-intersection test *t* show relative positions between the source node of ray and the intersection point.

is denoted by source node position vector $\boldsymbol{S}$ (origin) and normalized direction vector $\boldsymbol{D}$ as follows:

$$\boldsymbol{R}(t) = \boldsymbol{S} + t\boldsymbol{D}. \tag{1}$$

A point, $\boldsymbol{T}(u,v)$, on a triangle is defined by three vertices $\boldsymbol{V}_0$, $\boldsymbol{V}_1$, $\boldsymbol{V}_2$ as follows:

$$\boldsymbol{T}(u,v) = (1-u-v)\boldsymbol{V}_0 + u\boldsymbol{V}_1 + v\boldsymbol{V}_2, \tag{2}$$

where u, v are barycentric coordinates. If the ray intersects the triangle, the equation between $\boldsymbol{R}(t)$ and $\boldsymbol{T}(u,v)$ is $\boldsymbol{R}(t) = \boldsymbol{T}(u,v)$. And it is arranged by matrix representation as Eq. (3):

$$\begin{bmatrix} -\boldsymbol{D} & \boldsymbol{V}_1 - \boldsymbol{V}_0 & \boldsymbol{V}_2 - \boldsymbol{V}_0 \end{bmatrix} \begin{bmatrix} t \\ u \\ v \end{bmatrix} = \boldsymbol{O} - \boldsymbol{V}_0. \tag{3}$$

The solution of Eq. (3) is obtained as Eq. (4) by Cramer's rule.

$$\begin{bmatrix} t \\ u \\ v \end{bmatrix} = \frac{1}{|-\boldsymbol{D}, \boldsymbol{E}_1, \boldsymbol{E}_2|} \begin{bmatrix} |\ \ \boldsymbol{T},\ \boldsymbol{E}_2,\ \boldsymbol{E}_2\ | \\ |-\boldsymbol{D},\ \boldsymbol{T},\ \boldsymbol{E}_2\ | \\ |-\boldsymbol{D},\ \boldsymbol{E}_1,\ \boldsymbol{T}\ \ | \end{bmatrix}, \tag{4}$$

where $\boldsymbol{E}_1 = \boldsymbol{V}_1 - \boldsymbol{V}_0, \boldsymbol{E}_2 = \boldsymbol{V}_2 - \boldsymbol{V}_0$ and $\boldsymbol{T} = \boldsymbol{O} - \boldsymbol{V}_0$. From linear algebra, Eq. (4) can be written as Eq. (5):

$$\begin{bmatrix} t \\ u \\ v \end{bmatrix} = \frac{1}{\boldsymbol{P} \cdot \boldsymbol{E}_1} \begin{bmatrix} \boldsymbol{Q} \cdot \boldsymbol{E}_2 \\ \boldsymbol{P} \cdot \boldsymbol{T} \\ \boldsymbol{Q} \cdot \boldsymbol{D} \end{bmatrix}, \tag{5}$$

where $\boldsymbol{P} = \boldsymbol{D} \times \boldsymbol{E}_2$ and $\boldsymbol{Q} = \boldsymbol{T} \times \boldsymbol{E}_1$. A determinant of the matrix in the left side is calculated by scalar triple product. When $0 < u < 1, 0 < v < 1$ and $u + v < 1$ are satisfied, the ray intersects the triangle. And when t is positive, the cross point between the ray and the triangle is in front of $\boldsymbol{V}_0$. When t is negative, the cross point between the ray and the triangle is behind $\boldsymbol{V}_0$. The positional relation between the source node and the triangle is important for detecting whether the intended collision pair is feasible or not.

Searching collision pairs using rays

To detect collision between a spatula and a brain fissure, a collision detection method based on the ray-triangle intersection test for a non-convex hull is proposed. The process of collision detection using rays is explained as follows:

> **A1. Search of nearest triangles**
> Source nodes on the two planes of the spatula model cast a ray in the reverse direction of the normal vector $\boldsymbol{N}_R$ of the plane, on which nodes are located. If the ray is casted in the reverse direction of the normal vector $\boldsymbol{N}_R$, the sign of t (the result of the ray-triangle intersection test) will change. It is not essential whether the direction of the ray is the reverse of $\boldsymbol{N}_R$ or not. The contact planes approach to surface triangles of the brain objects. When collision between the plane and the surface triangle occurs, the triangle is behind the back side the plane. So the ray for searching collision pairs is casted in the reverse direction of the normal vector $\boldsymbol{N}_R$ of the plane. Ray-triangle intersection test is pursued among rays from the source nodes and the surface triangle of the brain. The source node compares distances of intersected triangles and finds the nearest intersected triangle behind.
>
> **A2. Detection of collision pairs**
> The normal vector $\boldsymbol{N}_{Tri}$ of the nearest intersected triangle is calculated. Then, inner product $\boldsymbol{N}_R \cdot \boldsymbol{N}_{Tri}$ is calculated. When $\boldsymbol{N}_R \cdot \boldsymbol{N}_{Tri} > 0$, the source node and the nearest triangle do not compose any pairs as shown in Figure 8a. In contrast, when $\boldsymbol{N}_R \cdot \boldsymbol{N}_{Tri} < 0$, the source node and the nearest triangle compose a collision pair as shown in Figure 8b.

The proposed method can find a collision pair between a simple spatula model and a non-convex brain model.

Deformation with collision pairs

In order to apply the proposed method to the simulator [25], enforced displacement of the brain model is represented by enforced displacements of vetices which compose surface triangle of the brain model. The displacements are computed with collision pairs as shown in Figure 9. The triangle is detected to be paired with a source node. Then the triangle is projected onto the plane, on which the source nodes in located, in the direction of the normal vector $\boldsymbol{N}_R$. An enforced displacement $\boldsymbol{U}_{enf}$ of vertex which composes a surface triangle that intersects with a ray is defined as follows:

$$\boldsymbol{U}_{enf} = \boldsymbol{V}_{proj} - \boldsymbol{V}_{init}, \tag{6}$$

where $\boldsymbol{V}_{init}$ is the initial position vector of the vertex, and $\boldsymbol{V}_{proj}$ is the position of the vertex projected onto the col-

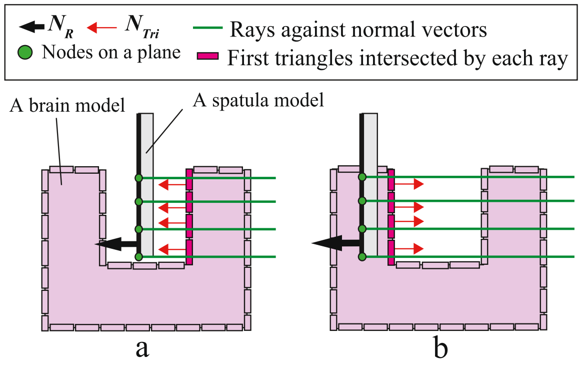

Figure 8 Searching collision pairs using ray [26]. (a) Without collision pairs. **(b)** With collision pairs.

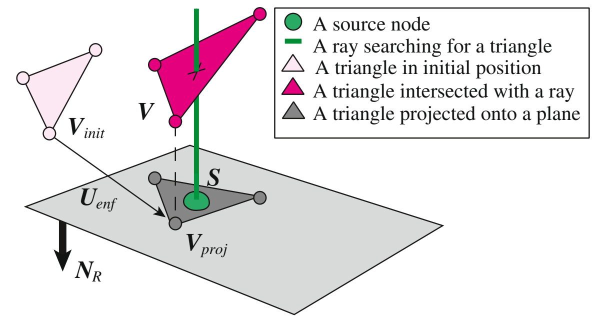
A source node
A ray searching for a triangle
A triangle in initial position
A triangle intersected with a ray
A triangle projected onto a plane
V_{init}
V
S
U_{enf}
V_{proj}
N_R

Figure 9 Enforced displacements computed with a collision pair. **V** is the current position of the vertex composes a surface triangle. $\boldsymbol{V}_{proj}$ is the position where *V* is projected in the same direction of the normal vector $\boldsymbol{N}_R$.

lision plane. When the spatula model moves very small distance during simulation period and collision pairs are detected on only one side of the spatula model, all collision pairs can be used for enforced displacements. However, when the spatula model moves fast and penetrates the brain model deeply within the simulation period, collision pairs are detected on both two planes. If all collision pairs give the brain model the enforced displacement, the brain model deforms unsuccessfully as shown in Figure 6. Volume-based collision detection can detect collision pair of each vertex of surface or each triangle, and enforced displacements of the deformable object are computed using the depth of the collision pair. However suitable enforced displacements cannot be selected when there are some collision pairs with only geometrical relationship of collision pairs. In Figure 6b, the spatula moves to right and penetrates the brain. As a result of collision detection, some collision pairs on both two sides of brain are detected. However, a feasible collision plane of the spatula model is only in right side because the first contact side is the right side of the spatula and the left side should not give the enforced displacement to the right side of the brain. Then collision pairs need to be culled with not only the geometrical relationship but also with contact state between objects for realistic deformation. This paper shows a method of culling collision pairs using history of collisions in the following subsection.

Preparation for culling collision pairs

A point of difference from other previous works is culling collision pairs with a history of collision state. To cull successfully, two preparations are needed.

History of collision state First preparation is management of collision state in previous loop of simulation. History of collision state manages with a flag whether or not each source node fixed on the spatula model is referred as displacement of surface triangle in the previous loop of simulation. Each source node has a collision history flag (His-flag). When a source node is referred to the displacement (calling the source node "collision node"), the His-flag of the source node is raised. In Figure 10a, some source nodes on the left side of a spatula model give enforced displacement and their His-flags are raised. In contrast, the His-flag will be turned off when the source node does not find a collision pair or give displacements to s surface triangle.

Arrangement of source nodes Second preparation is an arrangement of source nodes fixed on the two planes of spatula model. As previously mentioned, source nodes are discretely located on the two side of the spatula model. As additional assumption of the arrangement of source nodes, a source node on one side makes a pair with one on the other side. Pairs of source nodes are arranged so that pair source node is on same line parallel to the normal vector of the plane.

Culling collision pairs using history

Figure 10 shows an example of culling collision pair using His-flags. The process of culling collision pairs is as follows.

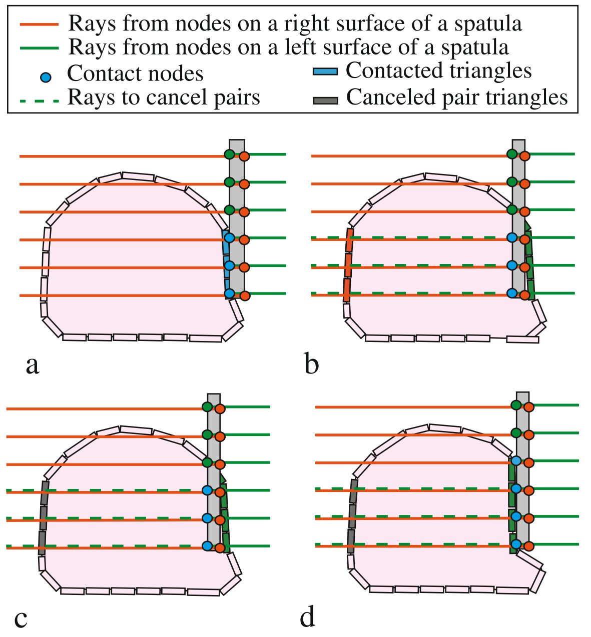
Rays from nodes on a right surface of a spatula
Rays from nodes on a left surface of a spatula
Contact nodes
Contacted triangles
Rays to cancel pairs
Canceled pair triangles
a
b
c
d

Figure 10 Deformation with culling collision pairs [26]. **(a)** Deformation in $n-1$ th loop. The left plane of a spatula contacts at first. Each contact source nodes changes its His-flag on. **(b)** Searching collision pairs in *n* th loop. **(c)** Culling collision pairs in *n* th loop. The source node whose His-flag is on searches triangles to cancel collision pairs. **(d)** Deformation in *n* th loop. Using residual collision pairs, the brain model deform naturally.

B1. Renewal of His-flags
In (n-1)th loop of the simulation, some source nodes find collision pairs and the collision pairs are referred as displacements of surface triangles. Flags of collision nodes are raised. In n th loop of simulation, collision pairs are found again as a result of collision detection. Each source node on the two planes of spatula model has a collision pair.

B2. Search of infeasible pair for deformation
If there are some source nodes whose His-flag is on, rays are cast from those nodes in the direction of the plane's normal vector $\boldsymbol{N}_R$. As a result of ray-triangle test, the source nodes find the nearest triangle in front. The nearest triangle in front of the source node composes infeasible pair for deformation of the brain model as shown in Figure 10c.

B3. Detection of enforced displacements
Triangles found in the process of B2 are infeasible for natural deformation. Then the collision pairs which contain infeasible triangle are canceled and there are only feasible collision pairs so that natural deformation is realized in deep penetration as shown in Figure 10d.

Merging the two intersection test

The proposed method carries out ray-triangle intersection test not only in collision detection but also culling collision pairs. If the intersection test in different process is carried out at once, computation time of collision detection will be decreased. Searching two kinds of triangle can be found in one ray-triangle intersection test. Triangles composing collision pairs are behind source nodes which cast rays. Triangles composing infeasible pairs are in front of the source nodes. In the previous subsection, the distance between a source node and a triangle can be measured with t in Eq. (5). So if rays which are cast with the same direction of $\boldsymbol{N}_R$ intersect with surface triangles in the test, the minimum of the absolute value $|t|$ is stored by each sign. As a result, source nodes find the nearest triangle in both front ($t > 0$) and backside ($t < 0$).

Acceleration with GPGPU

Bigger brain models and substantive source nodes increase computing time because the proposed method needs to try an intersection test for all combination of a source node and a surface triangle. Each ray-triangle intersection test has no data dependencies, so the test among all combination can be processed by parallel computing. As recent trends of parallel computing, General Purpose Graphic Processing Unit (GPGPU) contributes to acceleration using its many cores. GPGPU can be introduced into ray-triangle intersection during the proposed collision detection. For using GPGPU, we adopt Computing Unified Device Architecture (CUDA) that NVIDIA Inc. provides as an integrated development environment for Graphics Processing Unit (GPU). CUDA offers a hierarchical thread structure and various memory structures for parallel computing. The hierarchical thread structure consists of a grid, blocks and threads. Grid has some blocks and a block have some threads. Maximum number of block of each grid and maximum number of thread that each block has are limited. The source nodes are assigned to each block and the surface triangle to each thread so that the intersection test is performed in each combination of a block and a thread. When intersection test is performed by GPU, Central Processing Unit (CPU) sends the data sets of position of source nodes and all surface triangles to GPU by each side of the spatula. Using CPU, each result is sorted and the nearest surface triangle for a source node in both front and backside is determined.

Results and discussion

In this section, it is evaluated whether the proposed method can work for real-time surgical simulation and computing time of collision detection is measured. Deformation in all simulations is computed by FEM using enforced displacements as a result of collision detection [25]. Collision detection in the following simulations is accelerated with GPGPU. A specification of the computing system is shown in Table 1. An additional movie file shows the following simulation in more detail [see Additional file 1].

Effectiveness of culling collision pairs

One of features of the proposed method is culling collision pairs obtained as a result of collision detection. The effectiveness is evaluated by simulations with/without the culling scheme. Figure 11 shows overviews of the simulation results. A thin spatula model whose source nodes are located on both largest two planes moves in the same trajectory in with-and-without culling. In both conditions,

Table 1 Specifications of CPU and GPUs

Device	Parameters	Values
CPU	Core	intel Core i7-3960X
	clock frequency	4.8 [GHz]
	RAM	64 [GB]
Tesla K20c	Core	2,496 CUDA Cores
	Copy Engine	2
	RAM	5 [GB]
	Memory bandwidth	208 [GB/s]
GTX TITAN	Core	2,688 CUDA Cores
	Copy Engine	1
	RAM	6 [GB]
	Memory bandwidth	288 [GB/s]

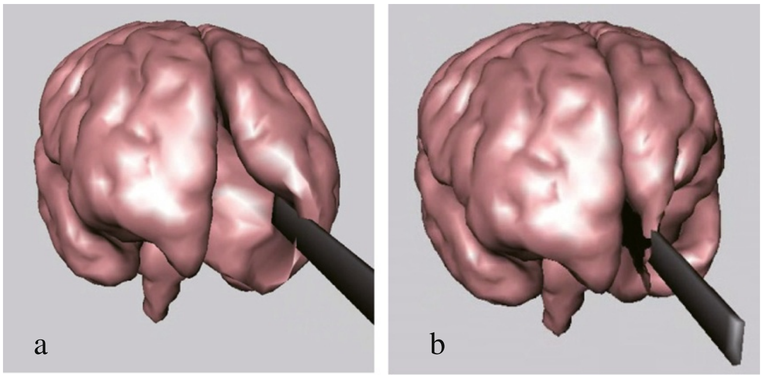

Figure 11 Comparing collision detection with/without the culling scheme. A spatula model is inserted into the inter-hemispheric fissure and moves. **(a)** Result of deformation with culling collision pairs. The brain deformed naturally. **(b)** Result of deformation without culling collision pairs. The brain failed deformation.

the spatula model penetrates the brain model deeply and each side of the spatula model searched collision pairs. While the computation without culling collision pairs cannot display realistic deformation in Figure 11b, the deformation computed with culling scheme is realistic as shown in Figure 11a. When a thin instrument excludes a brain fissure to ensure view of operation, travel distance of the instrument is larger than the thickness of the spatula as the trajectory in the simulation. From this result, culling collision pairs with His-flags works for giving enforced displacements so that natural deformation of the brain model is realized.

Simulation with collision on both sides of a spatula

If collision detection is performed with only one side of a spatula model, enforced displacement can be detected without culling collision pairs. However, both sides of spatula sometimes contacts with the soft tissue in operation of opening a brain fissure, and operation with one side collision detection leads to unrealistic deformation of the brain. Therefore, collision detections with both sides of the spatula model are required to opening the brain fissure. Through a simulation, it is evaluated that the spatula models can detect collision and give enforced displacements with the proposed method. In this simulation, source nodes located in whole of the largest side of the spatula model. Two simulations are performed to evaluate the algorithm in different situation. The first simulation evaluates whether both paired source nodes can give enforced displacement in spite of culling collision pairs. The spatula model is inserted into V-shape valley of a deformable model as shown in Figure 12a. While contact condition of the both planes are expected to be the same in the prior simulation, the second one evaluates whether the proposed method can handle a case that contact condition on a plane of the spatula differs from another plane. An U-shape deformable model as shown in Figure 12 is used in this simulation. Number of surface triangles of both models is less than 1,500, and number of tetrahedron elements is approximately 5,000. Figure 12 shows results of the simulation. In Figure 12b, the spatula model is inserted into a V-shape valley and deformed the deformable model with both sides of the spatula model. Triangles of infeasible pairs are located in front of a source node whose His-flag is on. The proposed culling does not cancel feasible pairs in error. In Figure 12d, each side of the spatula model can give enforced displacements to the U-shape deformable

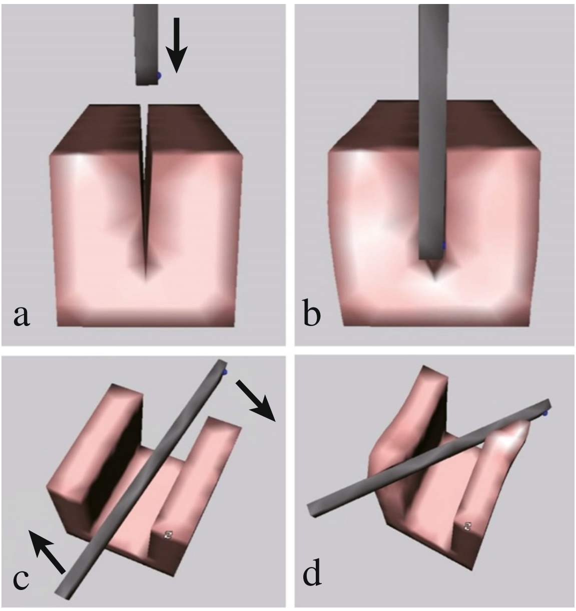

Figure 12 Collision detection in situation both sides of a spatula contact. (a) A spatula model is inserted into V-shape valley. **(b)** A soft tissue is deformed by both sides of the spatula. **(c)** A spatula model rotates in U-shape model. **(d)** Each side of the spatula model contacts the soft tissue model and gives enforced displacements.

model although the distribution of contact nodes is different. In the proposed method, His-flags are prepared for each source node not for each plane of spatula so that each source node can search infeasible collision pairs.

Measurement of computation time

The aim of this subsection is to measure the computing time of the proposed method. Computing time is critical term for DCD in a real time simulation. Relationships between time and number of nodes or computing hardware are measured. Specifications of the computing system are shown in Table 1. Collision detection is performed between two spatula models and brain model which has 15,865 surface triangles (Figure 5d). Collision detection between one spatula model and one brain model can be computed in parallel. The result of this measurement is shown in Figure 13. By using GPU, computing time of collision detection can decrease from 2000 [ms] to 160 [ms] when number of source node is 500. Lesser source nodes lead to acceleration of collision detection in all conditions. Comparing hardwares, GPGPU contributes to acceleration of collision detection especially in large number of source nodes. However, there is no major difference between GPUs. Computing task of collision detection consists of parallelized parts by GPU and sequential parts. In the proposed method, intersection test between a ray and a triangle is computed by GPU and the results are sorted by CPU for searching or culling of collision pairs. When computing time is near 30 [ms], effects of CPU's computing part cannot be negligible. As a matter of fact, GPU's computing time is about 10–15 [ms] and half of computing time of collision detection are occupied with sorting the result of intersection test and computing enforced displacements. This seems to be the reason of less difference between GPUs. When number of source nodes is small, computing time with GPU increases. This increment is due to communication between CPU and GPU. To decrease computing time of collision detection, it is easy and useful to decrease number of source nodes, however it may lead to false collision detection and increase effects of the overhead. One of other ways to accelerate the computing time is to introduce hierarchical data structures like Bounding Volume Hierarchies (VBHs). Our algorithm does not use some hierarchical data structures and there is room for improvement.

Simulation of opening a brain fissure

The condition of the simulation

In this simulation, an operator moves two spatula models using haptic devices and tries to open the Sylvian fissure of the brain model (Figure 14c). Connective tissues of the brain model are already removed and the model has neither blood vessels nor nerves. The brain model is fixed near to the brain stem. Collision detection between the spatula model and the rigid skull model is not considered. Collision detection between the skull model and the brain model is simplified by representing the shape of the skull in a sphere. This simplification enables the computation cost low. Previous works indicate computing time of collision detection depended on number of the

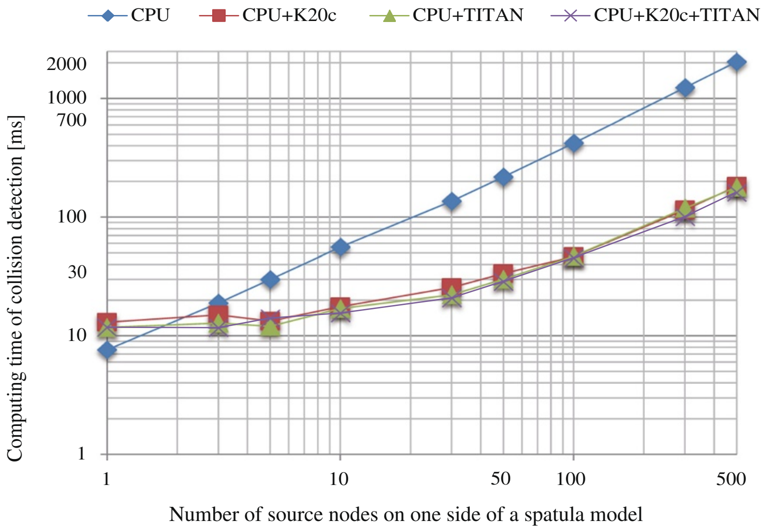

Figure 13 Computation time of the proposed method. Graph shows computation time of collision detection between two spatula models and a brain (Figure 5d). Real time simulation runs in less than 30 [ms/loop] [25]. However, computation time of deformation of a big brain model is about 700 [ms/loop].

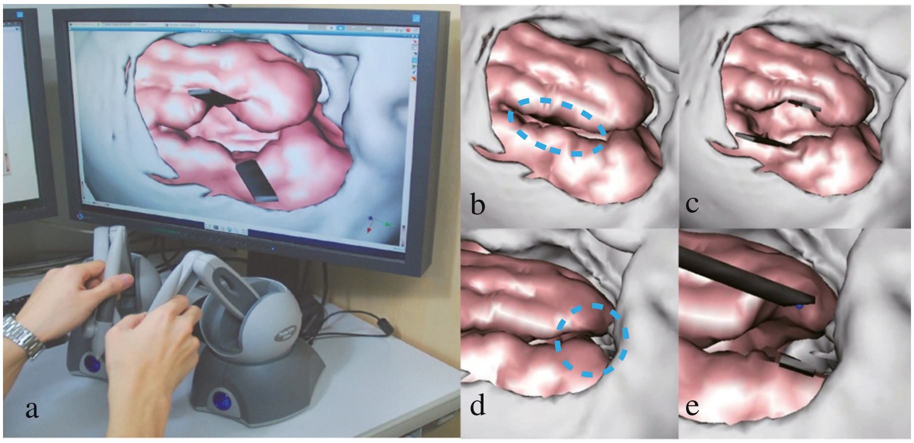

Figure 14 Simulation of opening the Sylvian brain fissure. (a) Overview of the simulation. Operator manipulates two spatula models with haptic devices. **(b)** Initial view of the Sylvian fissures. **(c)** The spatula models exposed the insula of the brain model. **(d)** Initial view from another angle. The depths of brain are not seen. **(e)** A part of the skull base is exposed.

source nodes and the surface triangles [26]. In this simulation, the brain model consists of 9,126 nodes, 34,401 tetrahedron elements and 12,608 surface triangles and it is expected computing time of collision detection is increasing. In order to decrease the computing time, the number of source nodes is reduced. In brain surgery, spatula rarely contacts with its whole body and source nodes are located intensively on an anterior part of the spatula to keep density of the nodes (Figure 5b). The number of source node located on one contact plane is 30.

The result of the simulation

Figure 14 shows the result of the simulation. Two spatula models exposed the insula of the brain model in Figure 14b and c. In the process of simulation, the spatula model was inserted and contacted the brain fissure with its both sides. In Figure 14d and e, a part of base of skull was exposed by excluding the brain. Proposed method can give realistic displacements to a deformable model in spite of long discrete time about 700 [ms] including computation of deformation.

Advantage of the simulation

By using this developing neurosurgery simulator, the operator can simulate opening a brain fissure with the reaction force from the deformed brain. This means that this simulator arrow surgeons to practice approaching to the lesion area in the depth of the brain. Comparing previous neurosurgery simulator [1], , this simulator provides training environment more similar to real neurosurgery in the variety of operative procedures.

Advantage of the proposed method

The advantage of the proposed method is to detect enforced displacements for deformation of the brain. Other deformation simulations adopt force as the

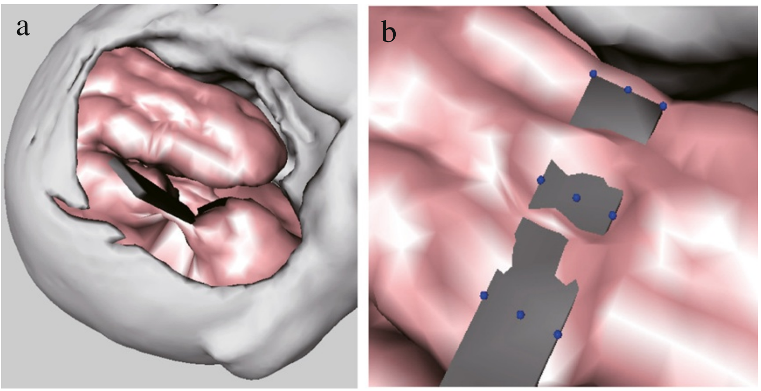

Figure 15 Limitation of the proposed method. (a) Approaching with side of a spatula, collision detection cannot work and unrealistic deformation occurs. **(b)** Low density of source nodes leads tunneling and the statement of contact is discrete.

result of collision to meet real-time computing requirement [10,13,14]. The model of contact force is detected with the intersection volume and some parameters, but validity of physics in the contact model is not clear. In this paper, the simulator adopts enforced displacements as the reaction of collision, so that reaction force are computed based on FEM. Other advantage is robustness against false negative collision pairs. While the previous DCD method works in less than 80 [ms] [10], the proposed method realizes to simulate realistic deformation of the brain in spite of long discrete time (700 [ms]). This simulator is still developing, so there are rooms for improvement in respect of CG rendering, computational speed, interface of simulator, and so on. After improvement those elements, a quantitative evaluation by surgeons is desirable whether the proposed method is adoptable for movement of their hands.

Limitations

The first limitation is the shape of instrument. Thickness of the spatula model is important to successful handling of the His-flags. When travel distance of the spatula is longer than the thickness of the spatula, His-flags cannot be updated in accordance with the change of collision states between the spatula and the brain. The thickness of spatula effects clerically in the first contact between the plane of the spatula and the surface of the brain as shown in Figure 10a. If the arrangement of source nodes that described in section 'Preparation for culling collision pairs' is satisfied, it is possible to extend the proposed method to curved or trapezoidal spatulas. However side planes of the spatulas are still not contactable.

The second limitation is length of a discrete time. While other real time simulation perform less than 30 [ms], the proposed method works in comparatively long discrete time about 700 [ms]. However, too long discrete times lead to false collision detection. And too thin spatula models cannot decide which side is make contact first.

The third limitation is direction of approaching. The proposed method considers only collision detection with the two largest planes of a spatula. Then collision detection with sides of the spatula cannot work and leads to unrealistic deformation (Figure 15a).

The last limitation is density of source nodes. Low source nodes contribute to decrease in computing time of collision detection. But low density leads to fail detect collision and cannot give enforced displacement as the plane (Figure 15b).

Conclusions

This paper proposes a new collision detection method which uses ray-triangle intersection test and history of collision state. Using history information, proposed method can search and cull collision pairs and realize interactive simulation of opening a brain fissure. In order to accelerate computing time of the proposed method, decreasing number of source node is easy and useful, but there is limitation with respect of precision. So it is required to introduce some hierarchical data structures to our algorithm. To develop more realistic brain surgery simulator, it is required to introduce blood veins, nervous tissue and connective tissue into the brain model and collision detection which can handle contact with those complex inner structures. After improvement the physical model of brain, computational speed, interface and so on, a quantitative evaluation by surgeons is desirable whether the proposed method is adoptable for movement of their hands.

Additional file

Additional file 1: Simulation of opening a brain fissure. This is a movie file of demonstration of brain surgery simulator using the proposed method collision detection method.

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

TT, AK and MU presented the problem of the collision detection in surgery simulation and AF proposed the novel collision detection method with in the course of discussion with coauthors. AF implemented the method in a neurosurgery simulator written by TT and KS. In order to accelerate the computation, TT and AK suggested the parallelization approach of the proposed collision detection, and they decided the computing system architecture for the collision detection based on prior performance evaluations conducted by TT using latest high performance computing systems. TT and AK set the simulation of opening a brain fissure as a target and AF evaluated the method by opening the Sylvian fissure. AF wrote this paper with the help of XJ and SA, TT and AK revised it, and KS and MU approved it. All authors read and approved the final manuscript.

Acknowledgements

This works was partly supported by JSPS Grant-in-Aid for Challenging Exploratory Research(24650288) and next generation world-leading researchers(LR003).

Author details

[1]Tohoku University, Sendai, Japan. [2]Carnegie Mellon University, Pittsburgh, USA. [3]Hokkaido University, Sapporo, Japan.

Received: 14 January 2014 Accepted: 12 June 2014
Published online: 26 September 2014

References

1. Delorme S, Laroche D, DiRaddo R, Del Maestro RF (2012) Neurotouch: a physics-based virtual simulator for cranial microneurosurgery training. Neurosurgery 71:32–42
2. (2014) Immersive Touch. http://www.immersivetouch.com/. Accessed 10 August 2014
3. Yasargi MG (1995) Microneurosurgery: operative treatment of CNS tumors, Vol. IV B. Thieme, New York
4. Ericson C (2004) Real-time collision detection. CRC Press, Boca Raton
5. Balan L, Bone GM (2006) Real-time 3D collision avoidance method for safe human and robot coexistence. In: Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on. IEEE, pp 276–282

6. Courtecuisse H, Jung H, Allard J, Duriez C, Lee DY, Cotin S (2010) Gpu-based real-time soft tissue deformation with cutting and haptic feedback. Prog Biophys Mol Biol 103(2):159–168
7. Malone HR, Syed ON, Downes MS, D'Ambrosio AL, Quest DO, Kaiser MG (2010) Simulation in neurosurgery: a review of computer-based simulation environments and their surgical applications. Neurosurgery 67(4):1105–1116
8. Teschner M, Kimmerle S, Heidelberger B, Zachmann G, Raghupathi L, Fuhrmann A, Cani M-P, Faure F, Magnenat-Thalmann N, Strasser W, Volino P (2005) Collision detection for deformable objects. In: Computer Graphics Forum, vol 24. Wiley Online Library, pp 61–81
9. Greß A, Guthe M, Klein R (2006) Gpu-based collision detection for deformable parameterized surfaces. In: Computer Graphics Forum, vol 25. Wiley Online Library, pp 497–506
10. Allard J, Faure F, Courtecuisse H, Falipou F, Duriez C, Kry PG (2010) Volume contact constraints at arbitrary resolution. ACM Trans Graph (TOG) 29(4):82
11. Heidelberger B, Teschner M, Keiser R, Müller M, Gross M (2004) Consistent penetration depth estimation for deformable collision response. In: Vision, Modeling, and Visualization 2004: Proceedings, November 16-18, 2004. IOS Press, Standford, USA, p 339
12. Heidelberger B, Teschner M, Gross M (2003) Real–time volumetric intersections of deforming objects. In: Vision, Modeling, and Visualization: Proceedings. AKA, Berlin, p 461
13. Faure F, Barbier S, Allard J, Falipou F (2008) Image-based collision detection and response between arbitrary volume objects. In: Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation. Eurographics Association, Switzerland, pp 155–162
14. Teschner M, Heidelberger B, Muller M, Gross M (2004) A versatile and robust model for geometrically complex deformable solids. In: Computer Graphics International, 2004. Proceedings. IEEE, pp 312–319
15. Hermann E, Faure F, Raffin B (2008) Ray-traced collision detection for deformable bodies. In: 3rd International Conference on Computer Graphics Theory and Applications, GRAPP 2008. INSTICC, Madeira, Portugal
16. Kim Y, Koo SO, Lee D, Kim L, Park S (2010) Mesh-to-mesh collision detection by ray tracing for medical simulation with deformable bodies. In: Cyberworlds (CW) 2010 International Conference On. IEEE, pp 60–66
17. Redon S, Kheddar A, Coquillart S (2002) Fast continuous collision detection between rigid bodies. In: Computer Graphics Forum, vol 21. Wiley Online Library, pp 279–287
18. Kim B, Rossignac J (2003) Collision prediction for polyhedra under screw motions. In: Proceedings of the Eighth ACM Symposium on Solid Modeling and Applications. ACM, New York, USA, pp 4–10
19. Shui Y, Zheng J, Ma X, Zhou H, Shen L (2012) Subspace culling for continuous collision detection In: Software Engineering (WCSE), 2012 Third World, Congress On. IEEE, pp 83–86
20. Wong S-K, Lin W-C, Hung C-H, Huang Y-J, Lii S-Y, (2013) Radial view based culling for continuous self-collision detection of skeletal models. ACM Trans Grap (TOG) 32(4):114
21. Govindaraju NK, Knott D, Jain N, Kabul I, Tamstorf R, Gayle R, Lin MC, Manocha D (2005) Interactive collision detection between deformable models using chromatic decomposition. In: ACM Transactions on Graphics (TOG), vol 24. ACM, New York, USA, pp 991–999
22. Brochu T, Edwards E, Bridson R (2012) Efficient geometrically exact continuous collision detection. ACM Trans Graph (TOG) 31(4):96
23. Tang M, Manocha D, Yoon S-E, Du P, Heo J-P, Tong R-F, (2011) Volccd: Fast continuous collision culling between deforming volume meshes. ACM Trans Graph (TOG) 30(5):111
24. Möller T, Trumbore B (1997) Fast, minimum storage ray-triangle intersection. J Graph Tools 2(1):21–28
25. Sase K, Nakayama M, Satake Y, Abiko S, Jiang X, Tsujita T, Konno A, Uchiyama M (2012) Evaluation of real-time preformance of the blunt dissection simulator using a real brain model. In: Proceedings of the 2012 JSME Conference on Robotics and Mechatronics. JSME, Japan, pp 2P-U02
26. Fukuhara A, Tsujita T, Sase K, Konno A, Jiang X, Abiko S, Uchiyama M (2013) Collision dtection for real time surgery simulation of opening a brain fissure In: Proceedings of the 2013 JSME Conference on Robotics and Mechatronics. JSME, Japan, pp 2A1-L02

doi:10.1186/s40648-014-0006-7
Cite this article as: Fukuhara *et al.*: **Proposition and evaluation of a collision detection method for real time surgery simulation of opening a brain fissure.** *ROBOMECH Journal* 2014 **1**:6.