Explicit contact modeling for surgical computer guidance and simulation

S. F. Johnsen, Z. A. Taylor, M. Clarkson, S. Thompson, M. Hu, et al.

S. F. Johnsen, Z. A. Taylor, M. Clarkson, S. Thompson, M. Hu, K. Gurusamy, B. Davidson, D. J. Hawkes, S. Ourselin, "Explicit contact modeling for surgical computer guidance and simulation," Proc. SPIE 8316, Medical Imaging 2012: Image-Guided Procedures, Robotic Interventions, and Modeling, 831623 (17 February 2012); doi: 10.1117/12.911787

SPIE.

Event: SPIE Medical Imaging, 2012, San Diego, California, United States

Explicit Contact Modelling for Surgical Computer Guidance and Simulation

S.F. Johnsen[a], Z.A. Taylor[b], M. Clarkson[a], S. Thompson[a], M. Hu[a], K. Gurusamy[c], B. Davidson[c], D.J. Hawkes[a] and S. Ourselin[a]

[a]University College London, Centre for Medical Image Computing, London, UK
[b]University of Sheffield, Dept. of Mechanical Engineering, Sheffield, UK
[c]Royal Free Hospital, Dept. of Surgery, London, UK

ABSTRACT

Realistic modelling of mechanical interactions between tissues is an important part of surgical simulation, and may become a valuable asset in surgical computer guidance. Unfortunately, it is also computationally very demanding. Explicit matrix-free FEM solvers have been shown to be a good choice for fast tissue simulation, however little work has been done on contact algorithms for such FEM solvers.

This work introduces such an algorithm that is capable of handling both deformable-deformable (soft-tissue interacting with soft-tissue) and deformable-rigid (e.g. soft-tissue interacting with surgical instruments) contacts. The proposed algorithm employs responses computed with a fully matrix-free, virtual node-based version of the model first used by Taylor and Flanagan in PRONTO3D. For contact detection, a bounding-volume hierarchy (BVH) capable of identifying self collisions is introduced. The proposed BVH generation and update strategies comprise novel heuristics to minimise the number of bounding volumes visited in hierarchy update and collision detection.

Aside from speed, stability was a major objective in the development of the algorithm, hence a novel method for computation of response forces from C0-continuous normals, and a gradual application of response forces from rate constraints has been devised and incorporated in the scheme. The continuity of the surface normals has advantages particularly in applications such as sliding over irregular surfaces, which occurs, e.g., in simulated breathing.

The effectiveness of the scheme is demonstrated on a number of meshes derived from medical image data and artificial test cases.

Keywords: Contact modelling, explicit solver, TLED, Lagrange-multiplier technique, bounding volume hierarchy, FEM, surgical simulation, surgical computer guidance

1. INTRODUCTION

FEM modelling has for some time now played an important role in surgical simulation, and is slowly finding its way into surgical guidance. In these applications the mechanical interaction between tissues plays an important role, and thus contact modelling is essential.

The discussion of FEM contact modelling in general has primarily been around implicit techniques, such as: Allard et al.'s LDI technique,[1] and the work on the Dellasus operator done by Duriez, Dequidt, and Cotin, amongst others.[2,3]

Little attention, though, has been given to dedicated techniques for explicit FEM solvers which have proven to be very suitable for high speed tissue simulation.[4,5]

The range of contact detection algorithms proposed for FEM contact modelling is as wide as that of methods for their solution. Spatial hashing,[6] implicit surface descriptions[2] and framebuffer-based techniques[1,7] have all been shown to be very efficient choices for detecting contacts between deformable objects. However, these techniques have in common that their extension to self-collisions is neither simple nor intuitive.

We propose a bounding volume hierarchy (BVH) based method with a self-collision detection based on the work of Volino and Thalmann,[8] and an intuitive way of assigning master and slave roles to surfaces involved in self collisions by means of primitive adjacency-based BVH construction. This class of methods is less commonly used in FEM contact modelling, but has yielded some impressive results in fracture and cloth modelling.[9,10]

Medical Imaging 2012: Image-Guided Procedures, Robotic Interventions, and Modeling,
edited by David R. Holmes III, Kenneth H. Wong, Proc. of SPIE Vol. 8316, 831623

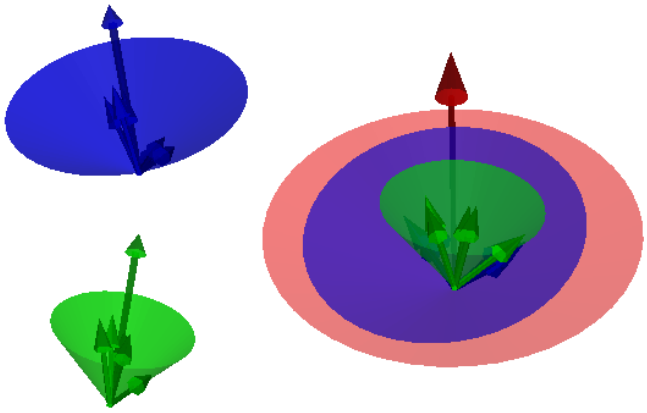


Figure 1. Left: Surface cones of child BVs with normals (small arrows) and axis (large arrow). Right: resulting parent surface cone

2. METHODS

2.1 Algorithm Overview

Algorithm 1 Pseudo-code for timestep

$t \leftarrow t + \Delta t$ {Beginning of time-step}
Compute new internal forces, update external forces
repeat
  `Start_of_loop`:
  Compute displacement predictors based on internal, external, and current contact forces
  Update contact surfaces, contact detectors with predictor displacement
  Switch role of master and slave surfaces
  Detect contacts
  Compute contact responses
  if Have only rate constraints **and** is first iteration **then**
    goto `Start_of_loop` {Have to try again with master/slave roles reversed since risk of having undetected intersections is high}
  end if
until No more mesh intersections detected
if had active rate constraints in last iteration **then**
  recompute displacements
end if
Save intersection free surfaces for narrow-phase contact detection in next time step

A pseudo-code overview of the contact algorithm is given in Alg. 1. The notion of rate constraints appearing in this description is explained in section 2.3.

2.2 Collision Detection

The proposed BVH consists of axis-aligned bounding boxes (AABB) and has a binary-tree structure. The extension of the algorithms described in this section to other types of bounding volumes and higher order trees, however, is straight-forward. At leaf level, the AABBs bound one primitive (surface triangle) each such that the primitive's vertices at the start of the time step as well as at the end of the predictor step are fully contained within it.

The key BVH property that allows for detection of self-collisions is that of surface cones. The surface cones were originally introduced by Volino and Thalmann[8] as a means of identifying BVH subtrees where self-collisions are possible, and are defined as the narrowest cone bounding the normals of a patch of connected primitives. See Fig. 1 for an illustration. Since in self-collision detection we only deal with solid elements, self-collisions

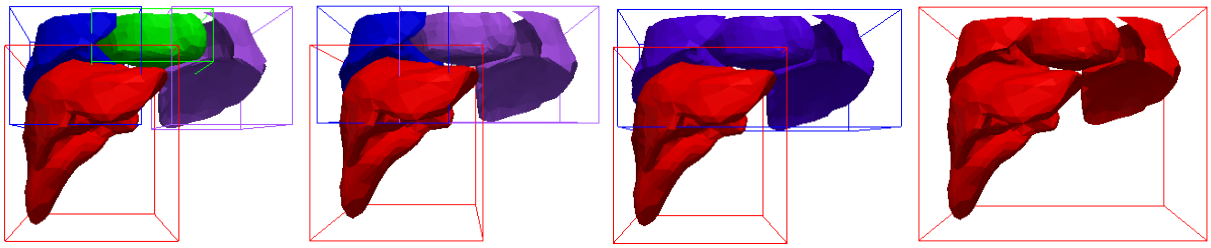


Figure 2. Illustration of the bottom-up AABB merging process. The left-most picture shows the initial state when the AABBs contain only connected geometry. The next picture shows the state after the two boxes yielding the smallest parent box have been merged. The last picture (right-most) shows the BVH root bounding all geometry. Meshes courtesy of IRCAD.

resulting from intersecting mesh boundaries as described by Mezger[10] are of no concern to us, and we can treat the surface cone self-collision criterion

$$\alpha_{VT} \geq \tau_{VT}, \quad \tau_{VT} \leq \pi \tag{1}$$

where α_{VT} is the cone opening angle and τ_{VT} the threshold above which self-collision tests are performed, as a necessary criterion for self-collision. The computation of this quantity α_{VT} is done recursively (Fig. 1) as part of the BVH update with the method proposed by Provot.[11]

Since the number of BVH nodes that need to be tested for self-collisions is determined by the surface cones, the same are used to influence how the BVH is generated and updated, so as to reduce the number of BV intersection tests and updates. The generation process comprises two main stages, in the first of which disconnected geometry is identified and the top part of the BVH created from the boxes bounding these clusters (Fig. 2) in bottom-up fashion. The second stage is the top-down division of the boxes bounding the connected primitives. In order to be able to apply the surface-cone self-collision criterion, the division process makes sure that at all stages the primitives bounded by the resulting child BVH nodes remain connected. The cost function governing the assignment of primitives to child nodes consists of two terms: the first one is the rather common minimal volume criterion, the second minimises the opening angle of the surface cone.

$$\mathcal{B}^{n+1}_{child_i} = \mathcal{B}^{n}_{child_i} \cup \left\{ \arg \min_{T \in \mathcal{B}_{parent}} V(\mathcal{B}^{n}_{child_i} \cup \{T\}) + c(\mathcal{B}^{n}_{child_i} \cup \{T\}) \cdot \alpha_{VT}(\mathcal{B}^{n}_{child_i} \cup \{T\}) \right\} \tag{2}$$

$\mathcal{B}_{child}$, $\mathcal{B}_{parent}$ denote the primitive sets bounded by a child and the parent BVH node being split, respectively. $T \in \mathcal{B}_{parent}$ is any primitive from the parent node set, $V(\mathcal{B})$ is the volume of the BV bounding the primitive set $\mathcal{B}$, $\alpha_{VT}(\mathcal{B})$ the opening angle of its surface cone. The latter is approximated with

$$\alpha_{VT}(\mathcal{B}) = 2 \max_{T \in \mathcal{B}} \angle \left(\boldsymbol{n}(T), \frac{\sum_{T' \in \mathcal{B}} \boldsymbol{n}(T')}{\| \sum_{T' \in \mathcal{B}} \boldsymbol{n}(T') \|} \right) \tag{3}$$

To be able to mix the two quantities being minimised we introduce $c(\mathcal{B})$ in (2) which is defined as

$$c(\mathcal{B}) = \frac{\overline{V}_{cluster}}{\pi} |\mathcal{B}| \tag{4}$$

where $\overline{V}_{cluster}$ is the volume of the AABB bounding the current cluster.
The child node sets $\mathcal{B}_{child_i}$, $i \in \{1, 2\}$, are initialised with the two primitives in the parent bounding whose centroids are the farthest apart.

The lazy updating strategy employed here was devised starting from the work of Larsson and Akine-Möller.[12] The central notion in this strategy is that of update nodes (UN) which we define as subtree roots in which the deformation undergone by the bounded geometry is quantified to assess the need for an update of the respective BVH subtree. The UN role is assigned to BVs at the time of BVH creation, and the number of these nodes N_{UN}

is fixed and dependent on the total tree size. Since subtrees containing self collisions have to be visited (and thus updated) anyway, it makes sense to place the update nodes below a point where $\alpha_{VT} > \tau_{VT}$.
The algorithm for the placing is a greedy one, initialising the set of UNs with the leafs of the BVH. In every step, it picks the two nodes whose parent has the narrowest surface-cone opening angle, and replaces them with their parent in the intermediate set of update nodes. This procedure continues until the set reaches the target size of N_{UN}.
The BVH updating algorithm begins by iterating over the UNs. If the self-collision criterion (1) was satisfied in the previous timestep, the subtree is unconditionally updated top-down. If not, the displacement of all nodes bounded by the respective UN is computed. If this displacement is such that it's magnitude allows us to exclude a satisfaction of (1), i.e.

$$\alpha_{VT}^{(t)} \leq \alpha_{VT}^{(t_U)} + 2 \arctan \frac{\max_{i \in \mathcal{B}_N} \| \boldsymbol{u}'_i - \frac{\sum_{j \in \mathcal{B}_N} \boldsymbol{u}'_j}{|\mathcal{B}_N|} \|}{\min_{T \in \mathcal{B}} h(T)^{(t_U)}} < \tau_{VT} \tag{5}$$

No update of the respective subtree is performed. t_U in (5) is the time of the last update of the subtree, $\mathcal{B}_N$ denotes the set of bounded mesh nodes, $\boldsymbol{u}'$ is the nodal displacement between t_U and t, $h(T)$ is the diameter of the primitive T.
A bottom-up update of the hierarchy starting at the UN is performed unconditionally.

The broad-phase collision detection is performed by recursively checking BVH (sub-) trees against each other.[10] For self-collision detection the two children of any BVH where (1) holds true need to be checked against each other. The primitive-primitive test consists of one test for vertices against facets and one for edges against edges. The vertex-primitive test starts with an initial projection of the slave vertex $\boldsymbol{x_s}$ onto the primitive T using a projection operator P_T computed with Möller and Trumbore's method[13] with the minor modification of employing normalised facet normals instead of unnormalised ones

$$\left(\tilde{\xi}, \tilde{\eta}, \tilde{g}\right) = P_T \boldsymbol{x_s} \tag{6}$$

If the initial projection $(\tilde{\xi}, \tilde{\eta})$ lies within the bounds of the master facet plus a narrow margin and $|\tilde{g}|$ is sufficiently close to the previously nearest projection, this initial projection is improved upon with an iterative procedrure that employs *C0-continuous* master facet normals, $\boldsymbol{n_m}(\xi, \eta)$, and yields the final penetration depth (gap function value), g, and projection $\boldsymbol{x_m}$

$$\begin{aligned} \boldsymbol{x_m} &= [(\boldsymbol{x_m})_1 - (\boldsymbol{x_m})_0]\,\xi + [(\boldsymbol{x_m})_2 - (\boldsymbol{x_m})_0]\,\eta + (\boldsymbol{x_m})_0 \\ g &= \boldsymbol{n_m}(\xi, \eta)(\boldsymbol{x_s} - \boldsymbol{x_m}) \\ \boldsymbol{n_m}(\xi, \eta) &= \textstyle\sum_i b_i(\xi, \eta)(\boldsymbol{n_m})_i \end{aligned} \tag{7}$$

Where $b_i(\xi, \eta)$ is a standard 2D linear shape function.
This projection $\boldsymbol{x_m}$ is the virtual master-surface node based on which the responses in section 2.3 are computed. An illustration of these quantities and their relationships is given in Fig. 3.
The continuous master surface-normal field is approximated with vertex normals computed from

$$(\boldsymbol{n_m})_i = \sum_{j \in \mathcal{T}_i} \alpha_{j,i} \, (\tilde{n}_m)_j \tag{8}$$

Where $\mathcal{T}_i$ denotes the set of primitives adjacent to the master-surface node i, $\alpha_{j,i}$ the opening angle of the primitive j at said node, and $(\tilde{n}_m)_j$ the unnormalised, constant facet normal of the primitive j.
For each slave node only the nearest projection onto a master facet is stored.

The edge-edge collision detection is performed with the standard continuous collision detection test[11,14] on the time interval $[t - \Delta t, t]$. The cubic polynomial equation arising in this test is solved with the interval Newton method employed by Redon.[14]

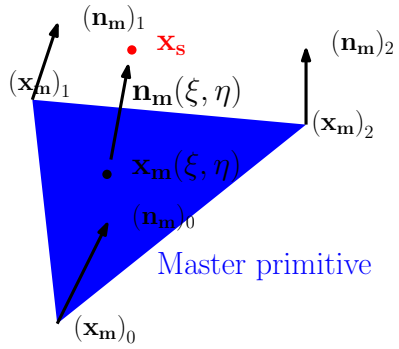


Figure 3. Illustration of the slave-to-master projection.

2.3 Collision Response Calculations

The contact modelling scheme presented in this section is derived from work of Heinstein et al.[6,15] and is only applicable to matrix-free FEM solvers that employ central-difference time integration, as is the case with the Total Lagrangian Explicit Dynamics algorithm[16] used in this work.

The collision response calculations employ the projections, and collision points computed in the narrow-phase of the collision detection. For vertex-primitive contacts we distinguish two types responses: penetration responses and rate responses. The prior come into effect when the gap function value g is negative, i.e. the slave node lies behind the master-surface facet. The forces are computed from

$$\begin{aligned} \boldsymbol{f_s} &= -\boldsymbol{n}(\xi,\eta)\beta\frac{m_s\overbrace{(\boldsymbol{x}_s-\boldsymbol{x}_m(\xi,\eta))\bullet\boldsymbol{n}(\xi,\eta)}^{g}}{\Delta t^2} \\ (\boldsymbol{f}_m)_i &= \boldsymbol{n}(\xi,\eta)\beta\frac{(m_m)_i g\gamma_i(\xi,\eta)}{\Delta t^2}, \quad \gamma_i(\xi,\eta) := \frac{b_i(\xi,\eta)}{\sum_j b_j(\xi,\eta)^2}, \quad i\in 0,1,2 \end{aligned} \tag{9}$$

$\boldsymbol{f_s}$ is the force applied to the penetrating slave node, $(\boldsymbol{f}_m)_i$ denotes the force applied to one of the three vertices of the penetrated master-surface primitive. $m_s, (m_m)_i$ are the masses of the slave node and master surface nodes, respectively.

The gap partioning factor β, appearing in (9), controls how the response is split between master and slave surfaces. For contacts between rigids and deformables this would be set to 0 and 1 respectively, as response forces applied to fixed, rigid objects have no effect and the penetration must be fully resolved by deformation of the deformable. For inter-penetration of deformables it is chosen to be 0.5 for both slave and master.

The factor γ_i distributes the gap function value over the master facet such that at the position $\boldsymbol{x_m}$ the fraction of the gap assigned to the master surface is recovered. Its derivation can be found in Ref. 17.

The latter type of response, rate responses, are used when g is positive but smaller than a constant Δ_{close} which is porportional to the average edge-length in the mesh and chosen such that any node at distance Δ_{close} from a master facet still lies within the safety margin of the BV. The rate responses serve to velocity-match surfaces at very close distance, and thus to improve the overall stability of the method. The force gradually increaseas as the distance between the surfaces decreases, and full velocity-matching is performed when there is zero distance between the slave node and its projection onto the master surface.

$$\begin{aligned} \boldsymbol{f_s} &= -\boldsymbol{n}(\xi,\eta)\dot{\lambda}\frac{g}{\Delta_{close}} \\ \boldsymbol{f_m} &= \boldsymbol{n}(\xi,\eta)\dot{\lambda}\frac{g}{\Delta_{close}} \end{aligned} \tag{10}$$

Where $\dot{\lambda}$ is the rate Lagrange multiplier computed with Heinstein et al.'s iterative method for node-node contact:[6]

$$\begin{aligned} \tfrac{1}{\Delta t}\tfrac{m_s m_m}{m_s+m_m}\lambda^{j+1} &= \boldsymbol{n_m}\bullet(\dot{\boldsymbol{x}}_{\boldsymbol{s}}^{\,j}-\dot{\boldsymbol{x}}_{\boldsymbol{m}}^{\,j}) \\ \dot{\boldsymbol{x}}_{\boldsymbol{s}}^{\,j+1} &= \dot{\boldsymbol{x}}_{\boldsymbol{s}}^{\,j}-\tfrac{\Delta t}{m_s}(\boldsymbol{f_s}^{j+1}-\boldsymbol{f_s}^{j}) \\ j &\leftarrow j+1 \end{aligned} \tag{11}$$

Test case	N. of BVs	Larsson, Akenine-Möller (Avg. BV updates/Time step)	Our Scheme (Avg. BV updates/Time step)
Ball, slab	2999	723	188
Liver, diaphragm	4799	2027.24	636.53

Table 1. Comparison in terms of number of updated BVs to Larsson and Akenine-Möller.

The $\dot{\boldsymbol{x}}$ denote velocities, initialised with

$$\dot{\boldsymbol{x}}^{(t)} = \frac{1}{\Delta t}(\dot{\boldsymbol{x}}^{(t)} - \dot{\boldsymbol{x}}^{(t-\Delta t)}) \tag{12}$$

The mass of the virtual master surface node m_m is computed with linear interpolation from the facet vertices' masses, so is the velocity of the virtual node $\dot{\boldsymbol{x}}_{\boldsymbol{m}}$. The application of forces to the real master surface nodes is done through the γ factor introduced in equation (9).

Edge-edge collision responses employ the same vertex-normals and rational that underlie equation (9), except that we have two slave nodes to deal with and only two master nodes, and the 2D shape functions of (9) are now 1D ones. The master normal is computed at the collision point on the master edge (its position on the edge is encoded in the parameter $r \in [0,1]$) by linear interpolation. The gap function value is given by the projection of the difference vector between the slave collision point (in turn given by a parameter $q \in [0,1]$ along the slave edge) and the master collision point at time t (end of timestep).

$$\begin{aligned}(\boldsymbol{f_s})_i &= -\boldsymbol{n_m}(r)\beta\frac{(m_s)_i\gamma(q)_i\overbrace{(\boldsymbol{x}_s(q)-\boldsymbol{x}_m(r))\bullet\boldsymbol{n_m}(r)}^{g}}{\Delta t^2} \\ (\boldsymbol{f_m})_i &= \boldsymbol{n_m}(r)\beta\frac{\gamma(r)_i(m_m)_i g}{\Delta t^2}\end{aligned} \tag{13}$$

3. EXPERIMENTS

3.1 Evaluation of BVH Generation and Update Strategy

The first experiment aims to demonstrate the benefits of having the UNs at positions determined by the self-collision criterion. To this end our proposed method is compared to that of Larsson and Akenine-Möller[12] in terms of the average number of updated BVs per timestep. The UNs in the reference code are assigned on a per geometry patch basis, i.e., we consider individual objects in the scene individually for BVH updating purposes. As recommended, we set the depth of the nodes to 1/2 the total depth of the subtree bounding the patch. The criterion based on which it is decided whether a subtree requires updating or not is the same for both methods, and given in equation (5).

The first test case is an artificial one consisting of a ball being pressed down against a slab through displacement boundary conditions at the top of the ball. The second one is based on CMIC volunteer data, and comprises a liver and a diaphragm in contact. The collision is in this case too achieved with displacement boundary conditions on the top of the diaphragm. In both experiments the constitutive model is Neohookean, the meshes are shown in Fig. 4. Table 1 shows the results from this experiment.

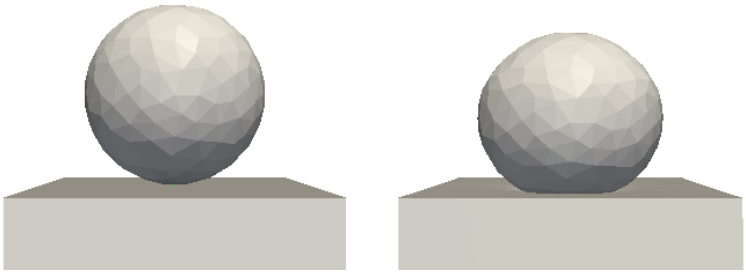
3.2 C0 and Constant Facet Normals

This experiment compares the continuous surface normal field underlying the projection and response-force computation to the standard constant facet normals typically employed in contact modelling with linear finite elements with the aim being to demonstrate the smoothness of the responses obtained with the proposed method.

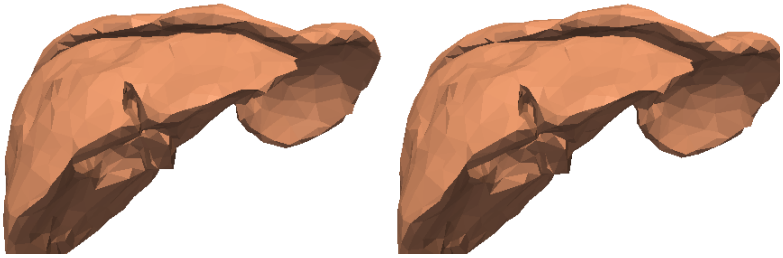
The experiment evaluates the normals along a line on a low resolution liver mesh, and computes the L2 difference between the two types of normals and with respect to the values (piece-wise constant) obtained along the same line on a higher resolution mesh. The lines over the mesh surface were created by raycasting along a straight line in the image plane. The line was chosen in an area of high curvature. The meshes and normals can be seen in figure 5. The meshes were derived from data obtained from IRCAD's 3DIrcaDB* with Meshlab's†

*http://www.ircad.fr/softwares/3Dircadb/3Dircadb.php

†http://meshlab.sourceforge.net/



(a) Ball/slab contact. Left: initial state. Right: final state



(b) Liver/diaphragm contact. Left: initial state. Right: final state

Figure 4. The meshes employed in the first experiment.

Figure 5. Left: C0 normals (red arrows) on a low-resolution liver mesh. Centre: Constant normals on low-resolution mesh. Right: Constant normals on high-resolution mesh.

quadratic edge-collapse feature. Nearest-neighbour interpolation was employed in the conversion from the high-resolution normal information to the lower resolution one. The resulting L2 differences (in mm space) are listed in Tab. 2.

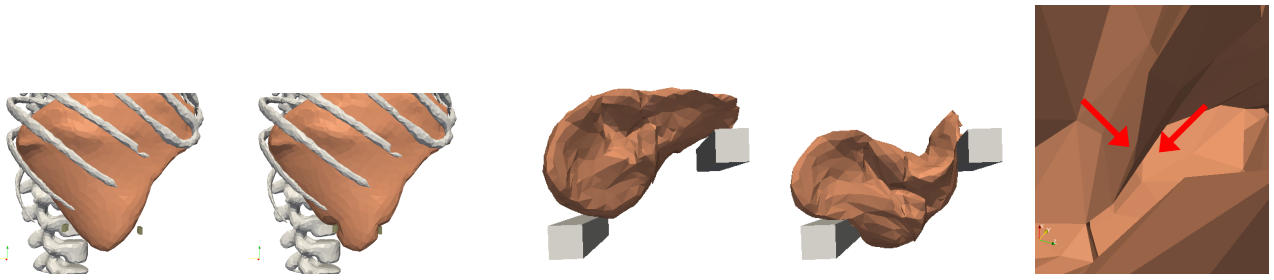
3.3 Full Simulations

The last two tests only serve to demonstrate the working of the entire piple. The deformables in both test cases use a Neohookean constitutive model. The first test case simulates the compression of the lower right lobe a liver surrounded by a rigid ribcage with surgical forceps. The anatomical meshes were obtained from IRCAD's 3DIrcadDB, down-sampled with Meshlab. The constitutive model for the liver is Neohookean. The forceps meshes were created in AutoCAD[‡]. The constitutive model for the forceps is also Neohookean, although with a higher stiffness. Initial and final state can be seen in Fig. 6 (a).
In the second test (Fig. 6 (b) and (c)), a liver mesh is placed on top of two rigid bars and subjected to large bodyforces, so as to force self collisions. The mesh was generated from CMIC volunteer image data. In Fig 6 (c)

[‡]http://usa.autodesk.com/autocad/

Const. normals/C0 normals	C0 normals/High-res	Const. normals/High-res
63.83	139.43	193.45

Table 2. Comparison between C0 and constant normals on a low-resolution mesh, and constant normals on a high-resolution mesh



(a) Liver manipulated with surgical forceps. Left: Initial state. Right: Final state

(b) Liver on two bars

(c) Interior structures

Figure 6. Full Simulations

it can be seen that structures collapse (indicated with red arrows), but there is no self-intersection.

4. CONCLUSIONS

We have introduced contact detection and modelling methods with a number of novel aspects such as BVH creation and updating strategies that lower the number of BVs that need to be visited in detection and updating by incorporating the surface-cone self-collision criterion of Volino and Thalmann in the BVH creation and updating.

The algorithms are still a work in progress, and at this time we only have a proof-of-concept implementation, hence no timing data is provided. However, we have demonstrated the efficacy of our proposed algorithms with a set of tests targeting individual aspects, as well as shown its applicability to anatomical data with two full simulations.

5. ACKNOWLEDGEMENT

This work was funded in part by the PASSPORT Liver (EU FP7) grant and the Intelligent Imaging Programme Grant (EPSRC Reference: EP/H046410/1).

REFERENCES

[1] Allard, J., Faure, F., Courtecuisse, H., Falipou, F., Duriez, C., and Kry, P. G., "Volume contact constraints at arbitrary resolution," *ACM Transactions on Graphics* **29**, 1 (July 2010).

[2] Dequidt, J., Lenoir, J., and Cotin, S., "Interactive contacts resolution using smooth surface representation," *Medical Image Computing and Computer-Assisted InterventionMICCAI 2007* , 850–857 (2007).

[3] Duriez, C., Andriot, C., and Kheddar, a., "A multi-threaded approach for deformable/rigid contacts with haptic feedback," *12th International Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2004. HAPTICS '04. Proceedings.* (section 2), 272–279 (2004).

[4] Taylor, Z. A., Cheng, M., and Ourselin, S., "High-speed nonlinear finite element analysis for surgical simulation using graphics processing units.," *IEEE transactions on medical imaging* **27**, 650–63 (May 2008).

[5] Taylor, Z. A., Comas, O., Cheng, M., Passenger, J., Hawkes, D. J., Atkinson, D., and Ourselin, S., "On modelling of anisotropic viscoelasticity for soft tissue simulation: numerical solution and GPU execution.," *Medical image analysis* **13**, 234–44 (Apr. 2009).

[6] Heinstein, M., "Contactimpact modeling in explicit transient dynamics," *Computer Methods in Applied Mechanics and Engineering* **187**, 621–640 (July 2000).

[7] Govindaraju, N. K., Knott, D., Jain, N., Kabul, I., Tamstorf, R., Gayle, R., Lin, M. C., and Manocha, D., "Interactive collision detection between deformable models using chromatic decomposition," *ACM SIGGRAPH 2005 Papers on - SIGGRAPH '05* **1**(212), 991 (2005).

[8] Volino, P. and Thalmann, N. M., "Efficient self-collision detection on smoothly discretized surface animations using geometrical shape regularity," *Computer Graphics Forum* **13**, 155–166 (Aug. 1994).

[9] Otaduy, M. A., Chassot, O., Steinemann, D., and Gross, M., "Balanced Hierarchies for Collision Detection between Fracturing Objects," *2007 IEEE Virtual Reality Conference* (1), 83–90 (2007).
[10] Mezger, J., Kimmerle, S., and Etzmuß, O., "Hierarchical techniques in collision detection for cloth animation," *Journal of WSCG* **11**(2), 322–329 (2003).
[11] Provot, X., "Collision and self-collision handling in cloth model dedicated to design garments," in [*Graphics interface*], **97**, 177–189, Citeseer (1997).
[12] Larsson, Thomas, and Akenine-Möller, T., "Collision detection for continuously deforming bodies," *Eurographics 2001* , 325–333 (2001).
[13] Möller, T. and Trumbore, B., "Fast, minimum storage ray/triangle intersection," in [*ACM SIGGRAPH 2005 Courses*], 7, ACM (2005).
[14] Redon, S., Lin, M., Manocha, D., and Kim, Y., "Fast continuous collision detection for articulated models," *Journal of Computing and Information Science in Engineering* **5**(2), 126 (2005).
[15] Heinstein, M. W. and Laursen, T. A., "An algorithm for the matrix-free solution of quasistatic frictional contact problems," *International Journal for Numerical Methods in Engineering* **44**, 1205–1226 (Mar. 1999).
[16] Miller, K., Joldes, G., and Lance, D., "Total Lagrangian explicit dynamics finite element algorithm for computing soft tissue deformation," *in numerical methods in biomedical engineering* **23**(2), 121–134 (2007).
[17] Lee, B., *Physically Based Modelling for Topology Modification and Deformation in Surgical Simulation*, PhD thesis, University of Sydney (2007).