Collision Algorithm for a Clamp Grabbing Deformable Materials: Implementation in an Open-Source Engine

F. Fazioli[1], F. Ficuciello[1], G. A. Fontanelli[1], B. Siciliano[1] and L. Villani [1]
[1] *Department of Electrical Engineering and Information Technology, University of Naples Federico II,*
f.fazioli@studenti.unina.it, fanny.ficuciello@unina.it

***Abstract*—The aim of this work is to integrate a soft-rigid collision algorithm in an open source physics engine, Bullet Physics, which simulates collision detection, soft and rigid body dynamics. In surgical applications this can be the case of a clamp grabbing deformable organic material or of a spatula opening a brain fissure. The goal is to realize a collision detection algorithm to handle the collision between the soft body (deformable organ) and the rigid body (spatula), assuming that the position of the spatula in the space is given by the Novint Falcon 3D Haptic Controller. The visual rendering has been realized using OpenGL, a cross-language for rendering 2D and 3D vector graphics.**

Keywords—**Surgical Simulation, Soft-Rigid Collision Detection, Real-Time Physics Simulation**.

I. Introduction

It is worth noting that the default soft-rigid collision algorithm proposed by Bullet Physics [1] is not very effective in the case of a thin tool interacting with a deformable object. In particular, if the rigid body (surgical tool) moves slowly, i.e. its displacement covers a small distance compared to the simulation step size, the collision is detected regularly. On the other hand, if the rigid object moves too fast, the Bullet default algorithm does not recognize the collision. As a consequence, the object penetrates into the soft body, and it appears located inside and not in collision. The implemented simulation scene is very essential. There are one rigid body, one soft body and a ground plane. The rigid body is a (1 m x 1 m x 0.125 m) parallelepiped while the soft body is a sphere of radius 3 m and composed by a 328 triangular mesh nodes. In the lack of interaction with the rigid object, the sphere rests on the ground plane.

II. Materials and Methods

To implement the soft-rigid collision algorithm [2], nodes are discretely located on the spatula surfaces. From these nodes rays are casted to detect the collision with the triangles of the soft body mesh. The ray casting algorithm used in this work is the one proposed in [3] by Möller and Trumbore. The rays are cast internally to the rigid body of the object (Figure 2), so as to find collisions as soon as a minimum intersection between the rigid and soft bodies is verified. Typically, for the scope of the work, the ray casting direction is the normal to the surface to which the node belongs. When the spatula (parallepiped) performs a very rapid displacement with respect to the simulation time, it will be detected by the algorithm as completely internal to the deformable organ (sphere, see Fig. 1) and the nodes on both parallel surfaces of the parallepiped would find a mesh on which collide. This problem also depends on the thickness of the rigid body. The thicker the more difficult to occur. To overcome this problem we can distinguish between feasible and non-feasible collisions with the flags method proposed by Fukuhara [2].
The algorithm, at each simulation step, executes a check for each node on the surfaces of the rigid body with the following steps.

A. First Step

To each node is associated a Boolean flag, that describes whether or not the relative node is colliding (flag equal to one means true). When a node is checked, at the beginning it is verified if the collision flag, relative to the opposite node on the parallel surface, is true. If this is true, we have to verify that the body is not fully contained in the deformable object. Thus, the node that presents flag true has to cast a ray, this time externally (see Fig. 1). If this ray finds a mesh, then the collision of the studied node is unfeasible, the collision flag (for the next simulation step) is turned to zero and the study goes to the next node of the rigid body. Otherwise, if the ray cast algorithm does not find a result, the algorithm continues checking the current node.

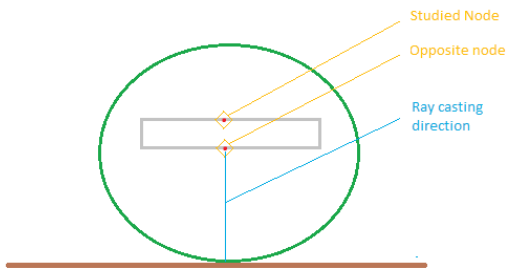

Figure. 1. First step of the algorithm, the opposite node casts the ray.

B. Second Step

The checked node casts a ray along the opposite direction of the normal to the surface which the node belongs to. If the ray casting algorithm gets a positive result, then an auxiliary condition has to be verified. Called N the normal to the triangle belonging to the soft body surface and N_r the normal to the surface of the rigid body to which the node belongs, the sign of the scalar product N·N_r must be evaluated. If negative, then the collision is taking place and

the collision flag relative to the node in exam is set equal to one (true), otherwise the collision flag is set equal to zero.

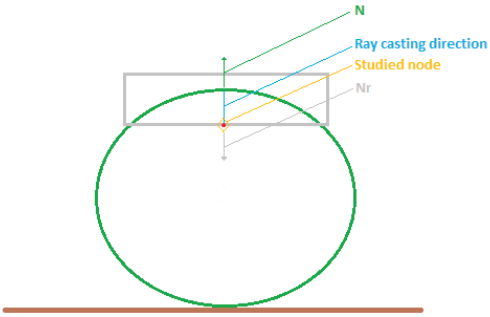

Figure. 2. Second step of the algorithm, the studied node casts the ray.

III. Results

By applying the default collision detection method proposed by Bullet Physics, it follows that the spatula collides with the soft body in a consistent way only for small displacements and slow motion. Fast displacement of the spatula can cause a complete penetration into the soft body surface without further collision (see Fig. 3).

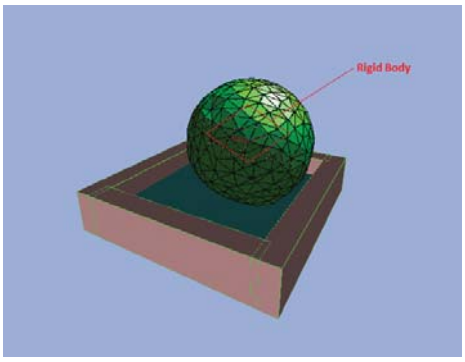

Figure. 3. Soft-rigid collision without the proposed algorithm. The rigid body is located inside

It is worth noticing that the application of only the second step of the algorithm causes drawbacks (see Fig. 4). In fact, the algorithm would work fine if the dimension of the rigid body along the collision direction is large enough to not allow entering, inside the soft body, to both the parallel surfaces (orthogonal to the colliding direction) during a single simulation step. If this is not verified, as in the case of Fig. 4, the collision appears unstable. This happens because both nodes belonging to the two opposite surfaces apply collision forces to the soft body even if some of these collisions are unfeasible. This drawback is overcome by applying also the first step of the algorithm, that detects the unfeasible collisions and excludes them from the study at the current simulation step (see Fig. 5).

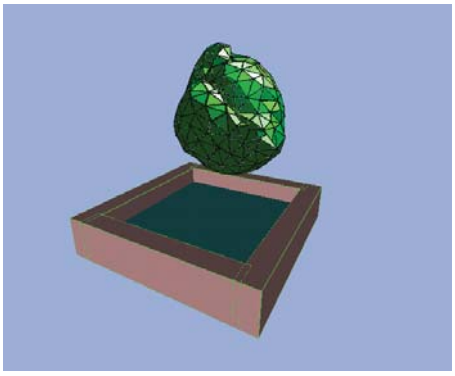

Figure. 4. The rigid body interacts badly with the soft object because of some unfeasible collisions.

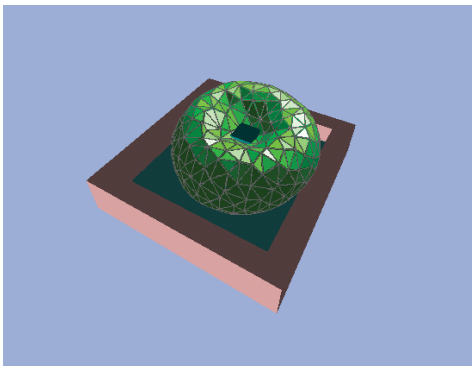

Figure. 5. Soft-rigid collision with the proposed algorithm.

IV. Conclusion

The simulation results of the collision detection method between a thin rigid body (spatula) and a soft object (organ), proposed in [1] and implemented in Bullet Physics, are promising if compared to the results obtained using the default algorithm of the physics engine. Despite the algorithm limitations related to shape, thickness and surfaces interested by the contact of the spatula (i.e. only collision detection with the two largest planes of the spatula are considered), it is worth continuing to investigate toward improved solutions, and future integration within different engines such as SOFA [4].

Acknowledgement

The research leading to these results has been partially supported by the RoDyMan project, funded by the European Communitys Seventh Framework Programme (FP7/2007 2013) under ERC AdG-320992.

References

[1] http://bulletphysics.org/wordpress/.
[2] Akira 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.
[3] T. Moller et al. "Fast, minimum storage ray/triangle intersection" Journal of Graphics Tools, 1997.
[4] https://www.sofa-framework.org/.