Implementation of a soft-rigid collision detection algorithm in an open-source engine for surgical realistic simulation

F. Fazioli, F. Ficuciello, G. A. Fontanelli, B. Siciliano, L. Villani

Abstract— **The aim of this work is to integrate a soft-rigid collision detection algorithm in the open source physics engine Bullet Physics. In surgical applications these types of collisions occur, for example, when a clamp grabs deformable organic material or a spatula opens a brain fissure. The goal is to handle the collision between the soft body (deformable organ) and the rigid body (surgical tool), assuming that the position of the rigid body in the space is given by the Novint Falcon 3D. The visual rendering has been realized using OpenGL, a cross-language for rendering 2D and 3D vector graphics. Moreover, the simulator allows to provide force feedback to the surgeon through the haptic device.**

I. INTRODUCTION

Minimally Invasive Robotic Surgery (MIRS) plays a fundamental role in modern medical procedures. Namely, a better sense of visual immersion and comfort for the surgeon, a lower post-operative pain and recovery time with respect to the classical open or laparoscopic surgery, are just some of the benefits of using robots in minimally invasive procedures.

Remarkably, the relationship between the surgeon and the trainee changes radically. In fact, the Halstedian concept of "see one, do one and teach one" is not valid anymore since the trainee can't see directly what the surgeon is doing during the surgical procedure. Therefore, it's necessary to develop new methods for training and this is the reason why, in recent years, there is a big interest in developing surgical simulators.

The aim of this work is to integrate a soft-rigid collision detection algorithm proposed by Fukuhara [3] in an open source physics engine, Bullet Physics [1]. In fact, the default soft-rigid collision detection algorithm proposed in Bullet is not very effective in the case of thin tools interacting with deformable objects. In surgical applications this can be the case of a clamp grabbing deformable organic materials, or of a spatula opening a brain fissure or a niddle interacting with soft tissues during a suturing procedure. 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, otherwise the default algorithm does not recognize the collision. As a consequence, the object penetrates into the soft body.

Besides the implementation in Bullet Physics [1] of the soft-rigid collision detection algorithm proposed in [3], the new contribution consists in the generalization of the algorithm to different shaped rigid objects such as convex rigid bodies that are thin along one of the three main directions. Moreover, haptic rendering has been realized by controlling the surgical tool in the 3D virtual space with the Novint Falcon 3D haptic device. The default linear elastic model of the interaction force has been replaced with a more realistic and physically consistent non-linear viscoelastic model [2]. Finally, the algorithm has been extended to a clamp constituted by two rigid colliding objects grabbing deformable materials.

The authors are with CREATE Consortium, via Claudio 21, 80125, Napoli, Italy, and with the Interdepartmental Center for Advances in Robotic Surgery of the University of Naples Federico II. Corresponding author's email: giuseppeandrea.fontanelli@unina.it

II. MOTIVATION

A. Use of simulators in robotic laparoscopic surgery

Surgical simulation has strongly widespread in the last decades, due to the progress of the robotic surgery. Simulation has gained acclaim in the teaching of robot assisted surgical procedures, as it allows solving various problems. For example, it enables the trainees to practice in any environments or for any amount of time, without occupying critical time and space of an operating room. The simulators, also, allow to train students in surgical robotic procedures for those hospitals that cannot afford purchasing surgical training robots.

A (non complete) list of simulators on the market is the following:

- Da Vinci Skills Simulator (DVSS)
- dV-Trainer
- Robotix Mentor
- Chiron
- Robotic Surgery Simulator (RoSS)
- Hands-on Surgical Training (HoST)
- SimSurgery Educational Platform (SEP)

Most of the commercial simulators offer to the surgeon just simple games aimed at training to the use of different kind of tools, i.e. simple pick and place of rings or simplified suturing, that are rather far to reproduce real situations in robotic surgery. To overcome these limitation, this work aims at developing a new simulator for visual and haptic rendering of soft-rigid contacts which is the basis for reproducing specific phases of important urological procedures. The simulator is conceived to provide a realistic haptic rendering to the surgeon.

B. Urological procedures

Urology is the field of surgery in which robotics plays the greatest role. Just consider that the da Vinci System has brought minimally invasive surgery to more than 3 million patients worldwide in the last years.

Robotics is used in urology for many disorders like, for example, kidney disorder or cancer, prostate cancer, bladder cancer and urinary blockage.

Radical prostatectomy is one of the urological procedure most commonly accomplished with a robotics systems. It consists in the complete removal of the prostate due to the presence of a tumor or an infection. The most critical phase of this procedure is the reconstruction called Vesicourethral Anastomosis (Fig. 1). During the reconstruction the surgeon has to link again, by suturing, the urethra with the bladder after the prostate removal. This procedure is very difficult and delicate due to the very little portion of tissue that the surgeon is available to re-create the link: an inexpert surgeon can break the suturing thread or create damage to the tissues.

This problem, as well as similar problems arising in other phases of surgery, has motivated the need for the development of simulators for critical parts of the surgical procedures, suitable for surgeon training.

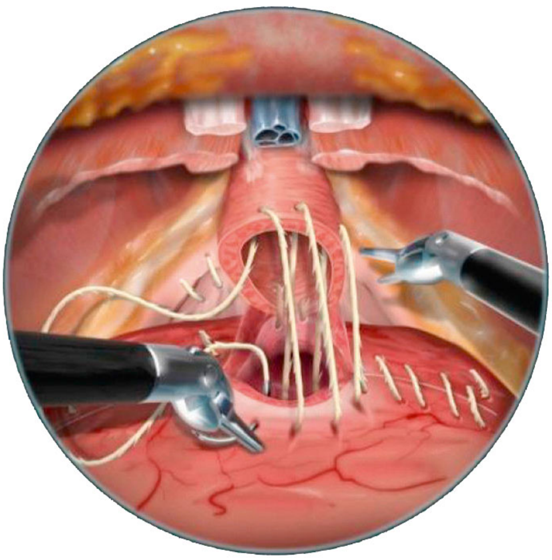

Fig. 1. Vesicourethral Anastomosis procedure.

III. MATERIALS AND METHODS

A. The physical engine

Bullet [1] is a 3D open source physics engine that supports rigid-body and soft-body dynamics and collision detection in 3D. It is a production physics engine that has wide support both in games and in movies. Moreover, it includes rigid-body simulation with both discrete and continuous collision detection, as well as support for soft bodies (such as clothes or other deformable objects).

B. The algorithm

In this section the problem of the collision between soft and rigid objects is considered.

A number of soft-rigid collision detection methods can be found in the literature (see, e.g., [4],[6],[5]):

- Bounding volumes (brute force method)
- Bounding volume hierarchies
- Sweep and prune
- Spatial partitioning
- Feature based
- Convexity based
- Position based
- Mixed

We have chosen the mixed algorithm proposed in [3] due to its fastness and effectiveness in evaluating the interaction between thin rigid objects and soft objects.

The algorithm uses ray casting in order to find a collision. In particular a ray-triangle intersection test, proposed by Moller and Trumbore [8], is computed at each ray casting step:

$$\begin{cases} R(t) = S + tD \\ t = -\dfrac{(S - V_0) \cdot (E_1 \times E_2)}{D \cdot (E_1 \times E_2} \\ E_1 = V_1 - V_0 \\ E_2 = V_2 - V_0 \end{cases} \tag{1}$$

where $R(t)$ is the Ray casting vector, t is the unit vector starting from a node of the rigid body along the normal to the triangle that have this node as a vertex, $V0$, $V1$ and $V2$ are the triangles vertex of the soft object.

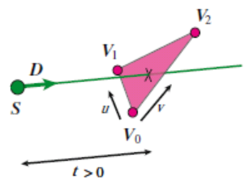

Fig. 2. Ray casting example.

Two checks are used in order to find and exclude the nodes that are in an unfeasible state of collision (see Fig. 3). In particular, this happens when a fast collision occurs in the direction in which the rigid object is thinner. In this case, it is possible that in one simulation step the rigid object is totally immersed in the soft object.

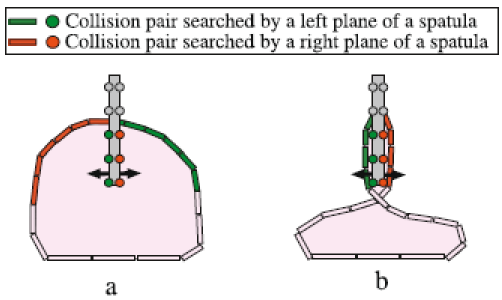

Fig. 3. Unfeasible collision.

In the first check all the nodes of the rigid mesh are evaluated and in particular if the opposite node of the studied one collides in the previous step, and the ray launched along the normal identifies a triangle of the soft object, then the studied node must be excluded in the current simulation step (see Fig. 4, step 3).

In the second check all nodes that pass the first check are studied: For each evaluated node a ray is launched along the normal opposite direction; if this intersects a triangle of the soft object and the scalar product between the normal of the two triangles involved in the collision is less then zero, then the current node is correctly colliding.

In detail, in Fig. 4 three steps of the algorithm are shown. In the first step, the two objects are not colliding. In the second step, only the bottom nodes of the rigid object are involved in the collision. The most critical case is evaluated in the third step in which the rigid object is immersed in the soft object and both the bottom and the top nodes are involved in the collision. In this latter case, the top nodes must be excluded from the collision computation.

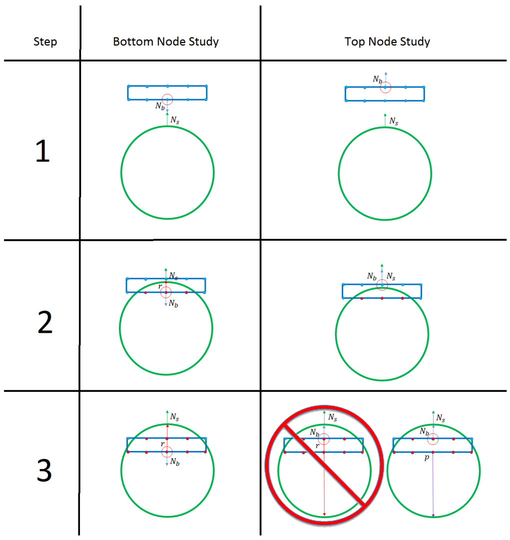

Fig. 4. Algorithm steps: (1) the two objects are not colliding, (2) only the rigid object bottom nodes are involved in the collision, (3) both bottom and top node of the rigid object are colliding.

Finally, in order to update the soft object shapes, the elastic force

$$f = kt \tag{2}$$

is applied to all the nodes of the soft object that are involved in the collision. In Eq. (2), t is the distance computed with the ray casting procedure. This force causes a deformation of the soft object in order to bring it outside the volume of the rigid body.

C. Extension to convex thin bodies

The rigid body used in [3] is a parallelepiped and the nodes used for collision detection are placed only on the two larger surfaces. The symmetrical shape of the body allows an easy manual placement of the nodes by defining the resolution along the planar directions of the two surfaces, see Fig. 5 (left).

In this work, the proposed algorithm has been extended from rigid boxes to generic convex objects modelled with triangular meshes. The algorithm in [3] requires that nodes on parallel surfaces are associated in pairs that belong to the same surface normal. For convex objects, as in Fig. 5 (middle), the nodes association is realized by placing a node in the centroid of each triangle mesh and by associating it to a node that belongs to the triangle intersected by the ray casting. The ray source is the node itself and the direction is the opposite of the normal to the triangle to which the nodes belongs. In this way, each triangle finds its opposite pair.

With respect to [3], where if a node x is associated to a node y also the vice-versa holds, in this case the nodes are not always mutually associated. This is not critical for the application of the algorithm , but some problems may arise.

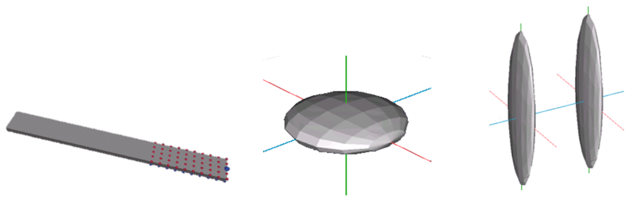

Fig. 5. Example of a spatula and nodes displacement in [3] (left). 3D Model of the convex rigid body (middle). The two collidable bodies of the clamp (right).

The rigid object represented in Fig. 6 (right) presents triangular faces in close proximity to the side edge, thus, very close to one another. Indeed, the distance between two triangles on the side edge may be of two orders of magnitude smaller than the thickness of the object, as in Fig. 6 (right).

Moreover, as an effect of triangle orientation, if a node on the top left, that belongs to a triangle in proximity to the edge side, casts a ray in the opposite direction of its normal, a node on the bottom right can be associated, as represented in Fig. 6 (left).

This can be critical for the proper application of the algorithm. Indeed, when the symmetrical arrangement of the associated nodes is lost, the wrong triangle associations can cause unfeasible deformations. To solve the problems arising for these critical nodes, the ray casted has no more infinite extension but is limited according to the dimensions of the scene and especially of the colliding soft bodies. In this way, infeasible collisions are highly reduced. Moreover, since there is no mutual association between the nodes, it is not possible to study the state of a node by verifying only the state of the opposite one. When a node belongs to an infeasible pair, the associated index is stored in a vector collecting all the indices of the nodes that cannot collide in the current simulation step. Thus, a cross-check is necessary at each step.

D. The interaction force model

In robotic surgical simulation, it is very important to have a precise control of the tools involved in the simulation but could be equally important to have the possibility to sense the scene through haptic force feedback. In fact, although in most

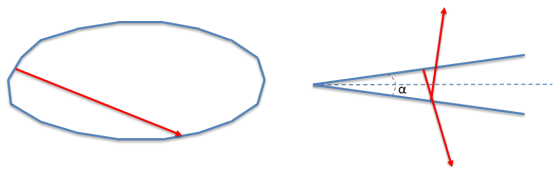

Fig. 6. Two associated triangles are in opposite position (left). Two associated triangles are adjacent to a sharp corner (right).

surgical robots the force feedback was not yet implemented, this technology could be helpful in the process of training in order to create force guided simulations, to provide a more realistic perception of the simulated scene and to help the surgeon to create a mental link between the applied forces and deformations perceived through the visual feedback.

In our work a Novint Falcon 3D haptic device is used in order to improve the sense of realism during the simulation. The contact force has been modeled using the Hunt Crossley model (Eq 3) presented in [2] incorporating a non linear spring in parallel with a non-linear damper to model the viscoelastic dynamics:

$$F(t) = \begin{cases} kx^n(t) + \lambda x^n(t)\dot{x}(t) & x(t) \geq 0 \\ 0 & x(t) < 0 \end{cases} \tag{3}$$

where x is the deformation, k and λ are the elastic and viscous parameters of the model respectively, and n is a real number (usually close to one), that takes into account the geometry of the contact surfaces.

At each time step, the deformation x is the distance between the current centroid of a triangle and the centroid of the same triangle at time zero. For this purpose, during the object initialization, the original position of each triangle vertices is stored. Obviously, the calculation of x is executed if and only if at least one of the nodes belonging to the triangle is in a colliding state. The total force is computed by adding the components related to each deformed triangle involved in the collision.

IV. EXPERIMENTAL RESULTS

A. The scene

In the first case study, a single rigid object (like a spatula) interacting with a deformable sphere has been considered. The sphere is composed by 128 triangular meshes, and is placed on a ground plane. The position of the spatula in the 3D space is given by the Novint Falcon 3D haptic device. The visual rendering has been realized using OpenGL.

In the second case study, the scene is constituted of a clamp realized with two collidable rigid objects and a handle not collidable but with solely aesthetic functions. The haptic interface allows three degrees of freedom (DoFs) for the controlled object motion. However, the clamp requires one DoF for closing and opening. Therefore, for simulation purposes, the motion of the clamp has been limited to a plane and the extra DoF has been used to control the closing/opening on different object shapes, such as a sphere and an elongated object shape, that simulate more realistically vessels and, in general, organic tissues.

B. 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 collisions, see Fig. 7 (left).

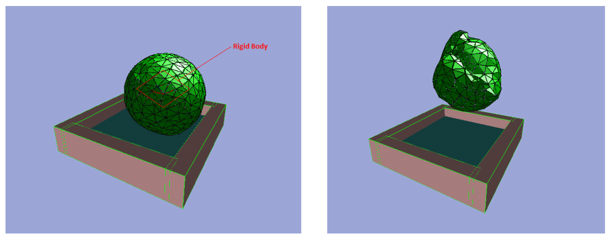

Fig. 7. Soft-rigid collision Bullet default algorithm.

Moreover, the application of only the ray casting method without node association (second step of the algorithm) causes drawbacks if the dimension of the rigid body along the collision direction is not large enough. The collision appears unstable, as in Fig. 7 (right), when both the nodes belonging to opposite surfaces apply collision forces to the soft body even if they are unfeasible.

This drawback is overcome by applying also the second step of the algorithm, that detects the unfeasible collisions and excludes them from the study at the current simulation step.

The algorithm has been tested with convex rigid objects and with a clamp grabbing deformable objects, see Figs. 8,9,10.

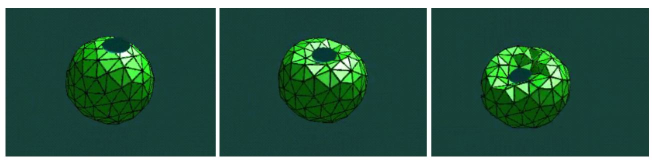

Fig. 8. Algorithm applied to a generic rigid convex object.

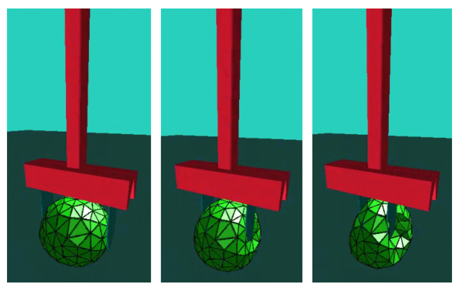

Fig. 9. Clamping simulation: simple spherical soft object.

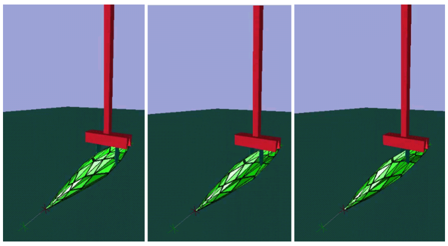

Fig. 10. Clamping simulation: simple organ simulation.

The results are realistic both in terms of visual rendering and force feedback, see Fig. 12.

In Fig. 11 the computation time of the algorithm as a function of the soft mesh triangles number is shown. A good performance is obtained also with a complex soft object (more than 4000 triangles) and using a basic hardware platform (*Intel I5 processor, Nvidia GTX750M VGA, 4GB of RAM*).

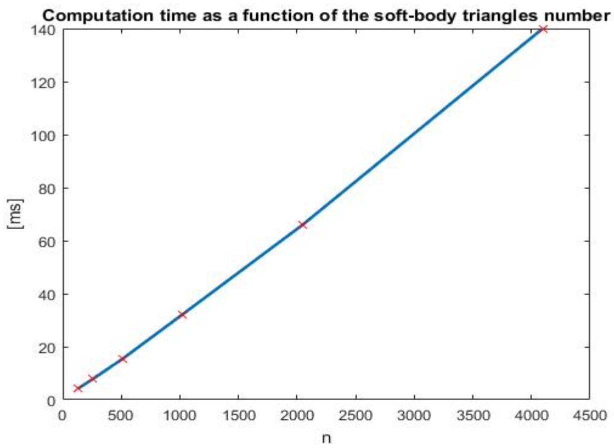

Fig. 11. Computation time as a function of the soft-body triangles number.

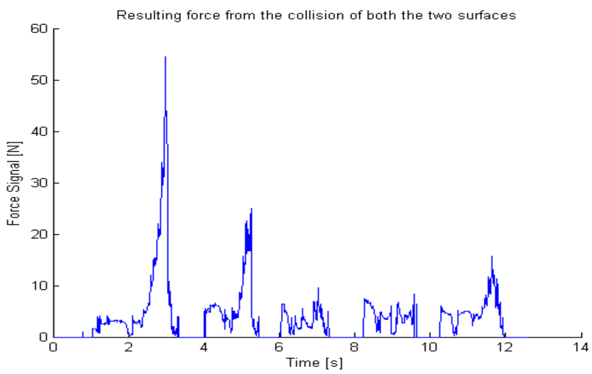

Fig. 12. Resultant force on the sphere grabbed by the clamp.

V. CONCLUSIONS AND FUTURE WORK

In the light of the promising results with respect to the Bullet default collision detection algorithm, future work will be devoted to develop further improved solutions, by including friction forces that have been neglected so far, as well as testing different physics engines, like SOFA [7].

VI. ACKNOWLEDGMENT

This research has been partially funded by the EC Seventh Framework Programme (FP7) within RoDyMan project 320992. The authors are solely responsible for its content. It does not represent the opinion of the European Community and the Community is not responsible for any use that might be made of the information contained therein.

REFERENCES

[1] *Bullet phisics Web page*. URL: `https://bulletphysics.org/`.

[2] F. Ficuciello et al. "Port-hamiltonian modeling for soft-finger manipulation". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2010.

[3] Akira Fukuhara et al. "Proposition and evaluation of a collision detection method for real time surgery simulation of opening a brain fissure". In: *ROBOMECH Journal* (2014).

[4] Morgan Kaufmann. "Real-Time Collision Detection". In: *The Morgan Kaufmann Series in Interactive 3d Technology*. 2005.

[5] S. Kockara et al. "Collision detection: A survey". In: *International Conference on Systems, Man and Cybernetics*. 2007.

[6] Timothy J. Purcell et al. "Ray Tracing on Programmable Graphics Hardware". In: *ACM Trans. Graph.* (2002).

[7] *SOFA Framework Web page*. URL: `https://www.sofa-framework.org/`.

[8] Moller Tomas and Trunbore Ben. "Fast, minimum storage ray-triangle intersection test". In: *Journal of Graphic Tools* (1997).