A multi-threaded approach for deformable/rigid contacts with haptic feedback

Christan Duriez, Claude Andriot
CEA SRSI
Route du Panorama, 92000 Fontenay-aux-Roses, France
contact author: christian.duriez@cea.fr
http://www.cea.fr

Abderrahmane Kheddar
Laboratoire Systèmes Complexes
University of Evry, Fre2494 CNRS
40, rue du Pelvoux, 91020 Evry, France

Abstract

This paper presents a physically based method to deal with haptic feedback of manipulated deformable virtual objects in contact with rigid or deformable environment. The haptic rendering algorithm is based on the Signorini's formulation of the contact as proposed in theoretical mechanics. Since haptic interaction requires high refresh rates, a multi-threaded methodology, sharing configuration values of the contact, ensures good stability and transparency performances. Thus, the interactive simulation gathers three synchronized process: the visualization process, the deformation/contact computation process, and the haptic rendering process.

1 Introduction

The use of physically based modeling for computing deformations in virtual environments is very often a compromise between "physical accuracy" (or plausibility) and real time performances. Many applications need interactive simulation of deformable virtual objects (ex. surgery simulators, CAD-based prototyping of compliant mechanical pieces, etc.) [3] [9] [6].

If the operator manipulates with haptic feedback the compliant virtual object, physically based computation (the collision detection and the deformation computation) must handle the real-time issue. Many researchers have studied truthful and fast models from finite element method, see for instance [3] [19] [11], but none of them used the contact formulation as proposed by Signorini [17].

After a brief review of previous works in terms of contact models (section 2), we show that, in the contact space, the Signorini's problem can be formulated as a linear complementarity problem (LCP) [12] (section 3). Under plausible contact hypotheses, efficient haptic rendering can be achieved by separating the haptic loop from the deformation computations (section 4). In order to assess the theoretical developments, the paper ends with an actual case study: an interactive virtual snap-in operation with haptic feedback.

2 Previous work

When a collision takes place between two virtual objects, penalty methods are mostly used to compute reaction forces. These obtained forces are used as inputs to the deformation calculus. In this case, the reaction forces are related to an *interpenetration measure* that could be either a volume [13] or a distance [4].

Reaction forces must ideally avoid deep interpenetration without adding energy to the system. In penalty-based methods, the force processing is straightforward, easy and fast. However, the non-penetration constraints are not guaranteed to hold (for the next simulation step). Moreover, the relation linking the interpenetration measure to the reaction contact forces is not true from the physical point-of-view (figure 1).

By using a linear FEM for virtual surgery simulation, Kühnapfel [8] imposed to the colliding virtual organ's vertices[1] a displacement law. The force imposed on each vertex (node) exactly achieves the desired displacement. These forces are the input of the deformation computation process. In another work, Picinbono [14] proposed to impose displacements of the colliding vertices to make them

[1] The ones that are involved in a collision with the surgeon's virtual tool (the rigid object).

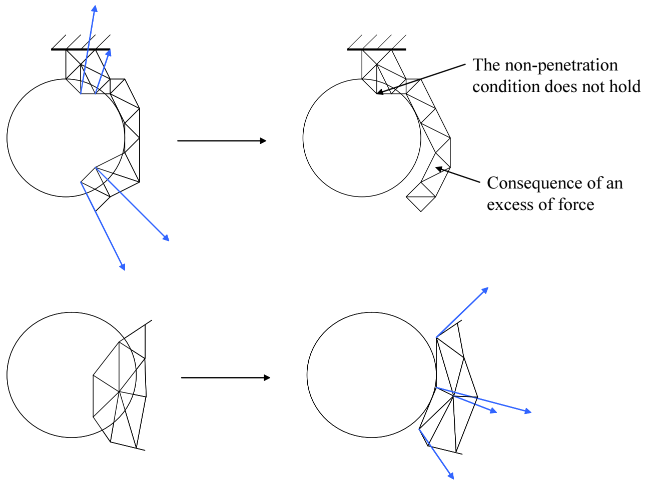

The non-penetration condition does not hold
Consequence of an excess of force

Figure 1. Problems that may arise using penalty methods: constraints may not hold (above), or artificial forces applies on nodes that are not in contact (below).

fit on the surface of the rigid body.

When using these methods, the non-penetration constraints hold perfectly for the nodes and show a good behaviour (namely for very deforming objects such as human organs) but the contact forces do not match the Signorini's problem conditions. Indeed additional "stick effect" appears during the displacement of colliding nodes.

In [10] and [1], the relative normal velocity of each colliding node is neutralized. This method requires however a continuous collision detection algorithm (i.e. the collision detection must be able to return the first collision time). In this case, the Signorini's conditions may not hold in some situations. For instance, the created forces are not necessarily normal to the surface (as it is the case for frictionless contact).

3 Rigid/deformable bodies contact

A simulation step begins with the motion integration without any contact constraint. This is called the free motion. During this motion, we want to detect if, when, and where a pair of objects collides. To deal with this computationally demanding problem, we are using a discrete collision detection algorithm. The AABB (axis aligned bounding box) is used as bounding volume hierarchy to speed up the collision detection process [18]. After a collision is detected, additional post-processing allows knowing the configuration of the first contact.

3.1 First collision between two triangles

Virtual objects are very often described by a collection of triangles that mesh their surface. Thus, body's geometric primitive that comes into contact is a point, a segment, or the interior of a triangle. Theoretically, when a pair of objects collides, nine configurations are denoted. They are described by three canonical pairs: point/triangle, segment/segment, and triangle/point.

The collision algorithm outputs actual colliding triangles and an estimation of the interpenetration vector. If the environment is deformable, we apply the algorithm of Provot [15] in postprocess. An adaptation in rigid environment case is presented here.

Triangles pairs from detection collision are tested. The coordinates of the deformable object's triangles are mapped into the rigid object's reference frame at times $T-1$ and T. For every possible triangle/triangle pair, we compute a linear interpolation of the deforming triangle points $[D_1 D_2 D_3]$ between their initial $(T-1)$ and final (T). Then three point/triangle collision tests, nine segment/segment collision tests, and three triangle/point collision tests are performed (see, the figure 2).

Indeed, the time-parameterized position of the each triangle point D_i is given by:

$$D_i^t = D_i^{T-1} + t(D_i^T - D_i^{T-1})$$

Each point D_i^t/triangle $[R_1 R_2 R_3]$ test 1 consists of a linear system resolution with t, α, $\beta \in [0,1]$ as variables and $\alpha + \beta \leq 1$, that is:

$$D_i^t = R_1 + \alpha(R_2 - R_1) + \beta(R_3 - R_1)$$

Each segment $[D_i^t D_j^t]$/segment $[R_k R_l]$ test 2 consists in finding the variables t, α, $\beta \in [0,1]$ by solving:

$$D_i^t + \alpha(D_j^t - D_i^t) = R_k + \beta(R_l - R_k)$$

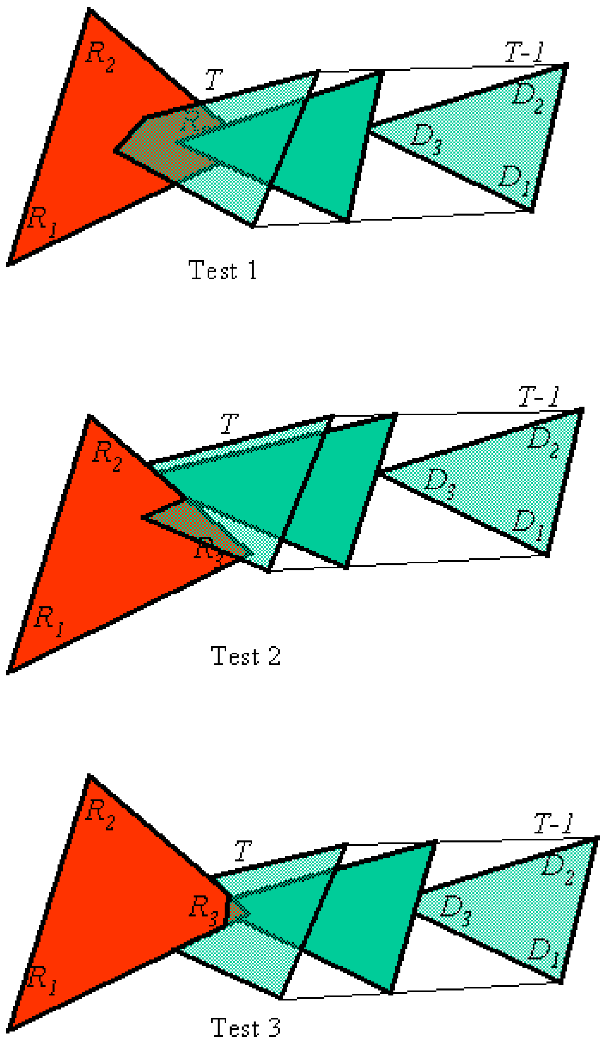
The two precedent systems are equivalent to the resolution of a polynomial equation of degree two in t.

Each triangle $[D_1^t D_2^t D_3^t]$/point R_i test 3 consists in finding the variables t, α, $\beta \in [0,1]$ with $\alpha + \beta \leq 1$ such that:

$$D_1^t + \alpha(D_2^t - D_1^t) + \beta(D_3^t - D_1^t) = R_i$$

This system leads to the resolution of a polynomial equation of degree three in t.

For the test 1, the contact normal vector n_i is given by the rigid triangle. It will constrain the only colliding vertex of the deformable object. For test 2, the contact normal vector is composed by two segments (the rigid and the deformable ones) at the collision time t. The contact will constrain the two segment's vertices of the deformable object by the FEM interpolation of the contact position. For the test 3, the deformable object's triangle gives the contact normal at the



Test 1

Test 2

Test 3

Figure 2. three different collision tests

collision time t. When a single deformable triangle is constrained by several contacts, the taken constraint contact is the one found with smallest time t.

For singular cases[2], the resolution is done on α and β for the final position $t = T$ of the triangle $[D_1^T D_2^T D_3^T]$ (In fact we are interested only in knowing what contact's configuration must be taken into account).

3.2 Contact space and FEM displacement space

To solve physically the deformations due to contacts, we have to take into account the interpolation chosen in Finite Element Method. Here, we use linear interpolation.

The collision detection's postprocess gives the time and the configuration when objects collide in free motion[3]. The whole of normal vectors n_i forms the contact space. To build a linear formulation, we assume that, during the resolution step, the contact space is constant. Barycentric coordinates are used to express δ in the contact space with FEM displacement.

3.2.1 Segment/Segment contact

Lets $D_1 D_3$ be the vertex of the colliding object's segment and $R_2 R_3$ the environment segment (Test 2). If a_i, b_i, c_i are the coordinates of the unit vector n_i, the final interpenetration distance δ_i will be related to displacement U_1 and U_3 of D_1 and D_3 and also to displacement V_2 and V_3 of R_2 and R_3 if the environment is deformable [4]:

$$n_i^T = \left[\begin{array}{ccc} a_i & b_i & c_i \end{array}\right]$$

$$\delta_i = n_i^T \left([\alpha_P, \beta_P] \begin{bmatrix} U_1 \\ U_3 \end{bmatrix} - [\alpha_R, \beta_R] \begin{bmatrix} V_2 \\ V_3 \end{bmatrix} \right) + \delta_i^{free}$$

Where $i = \left[\begin{array}{cc} 1 & n \end{array}\right]$ is the considered contact, n the number of contact, α_P, β_P and α_R, β_R are respectively segment $P_1 P_3$ and $R_2 R_3$ barycentric coordinates of the first collision point.

3.2.2 Triangle/Point contact

Lets D_1, D_2 and D_3 be the vertex of the colliding triangle and R_3 the colliding point (test 3). δ will be related to displacements U_1, U_2 and U_3 and also to displacement V_3 if the environment is deformable [5]:

$$\delta_i = n_i^T \left[+ \left[\begin{array}{ccc} \alpha & \beta & \gamma \end{array}\right] \begin{bmatrix} U_1 \\ U_2 \\ U_3 \end{bmatrix} - V_3 \right] + \delta_i^{free}$$

with α, β, γ, the barycentric coordinates of the first contact between triangle $D_1 D_2 D_3$ and R_3. The test 1 result is similar.

3.2.3 Transfer matrix

Gathering all relations from contact space to FEM displacement space allows building a matrix transfer H_P for the deformable object:

$$\delta = [H_D] \begin{bmatrix} \vdots \\ U_j \\ \vdots \end{bmatrix} + \delta^{free}$$

[2] Any t fits the equation for a single α and β, likely because of the point, the segment or the triangle of the deformable object moves on the contact plane.

[3] before movement correction due to contact resolution

[4] $V_2, V_3 = 0$ if the environment is rigid.

[5] $V_3 = 0$ if the environment is rigid.

The contact forces are also interpolated to FEM node's forces with this matrix transfer H_P. $j = \left[\begin{array}{cc} 1 & m \end{array} \right]$ with m the number of mesh points involved by the contact. H_P is an $n \times m$ matrix.

If the environment is deformable, a similar H_R matrix is build.

$$\delta = [H_P] \left[\begin{array}{c} \vdots \\ U_j \\ \vdots \end{array} \right] - [H_R] \left[\begin{array}{c} \vdots \\ U_r \\ \vdots \end{array} \right] + \delta^{free}$$

3.3 The Signorini's problem

After processing the collision detection and related contact computations, one needs to compute subsequent forces and constraints acting on the contact space. These forces allow deforming the compliant object (and optionally the deformable environment) in a 'realistic' manner[6]. To achieve this goal, we are using the characterization of the contact area as defined in [7] (it is also known as "the Signorini's problem" [17]). By using inequalities and complementarity relations, the Signorini's problem defines both:

- the forces distribution on the contact area, and
- the relation between these contact forces and the surface displacement of the deformable object (and environment).

This formulation resembles somehow what was performed by [2] to model non-penetration in rigid bodies simulation. In this work, the dynamic equations of motion are also formulated into a linear complementarity problem where the solution lies in the contact's space.

At the beginning of the loop, objects are moved in free motion. We want to correct all free interpenetration by putting its value positive or equal to zero:

$$\delta_i \geq 0$$

In this study, the only frictionless contact cases are considered. So the contact forces are projected along the contact normals. Moreover, these forces are directed towards the outside of the environment. Consequently, they are oriented:

$$\left\{ \begin{array}{l} F_i^n = f_i \geq 0 \\ F_i^t = 0 \end{array} \right\}$$

The same transfer is used from contact force to node's force.

$$\left[\begin{array}{c} \vdots \\ f_i \\ \vdots \end{array} \right] = [H_p] \left[\begin{array}{c} \vdots \\ F_j \\ \vdots \end{array} \right] = [H_p] F_c$$

Finally, there is a force only on the deformable vertices that are actually in contact after the deformation. This leads to the following complementarity condition:

$$(\delta_i \perp f_i)$$

These conditions are in accordance with the Signorini's problem. They are also re-written in a classical LCP form.

3.4 Linear relation between position and contact force

To build the LCP form, we need a linear relation between the contact forces on the concerned points and their position in the contact space. The contact space has been build by using free motion. So the deformation in free motion has integrated all forces in the simulation without the contact forces. We will describe here how to have a linear relation between constraint position of mesh nodes P_c, contact forces F_c and position obtained by free motion P_{free}. We will use U_c the displacement due to F_c:

$$U_c = P_c - P_{free}$$

If the deformation is static, the linear FEM gives a straightforward relation between displacements (U) and external forces (F), that is:

$$KU_c = F_c \Rightarrow P_c = K^{-1} F_c + P_{free}$$

where K is the object's stiffness matrix.

If the deformation model is linear while its dynamic is a Newtonian second order world, the relation is written in the form:

$$M\ddot{U} + D\dot{U} + KU = F$$

where M and D are the FEM mass and damping matrices.

By using the discrete decomposition of Tustin:

$$\dot{U}_t = \dot{U}_{t-1} + \frac{1}{2}(\ddot{U}_{t-1} + \ddot{U}_t)dt$$

and

$$U_t = U_{t-1} + \frac{1}{2}(\dot{U}_{t-1} + \dot{U}_t)dt$$

As the free position and the constraint position are defined in the same time step t, we just have to look the contribution of the contact force to the U_t terms.

$$P_c = \left(K + 2\frac{D}{dt} + 4\frac{M}{dt^2} \right)^{-1} F_c + P_{free}$$

If the deformation model is non-linear (as in the work of Zhuang [19]), the deformations are driven by:

$$M\ddot{U} + D\dot{U} + K(U) = F$$

[6] In a way that the none-penetration constraints (if any) hold, and the deformation is physically truthful

where $K(U)$ represents the non-linear stiffness of the deformable object.

The explicit scheme proposed by Zhuang leads to:

$$\dot{U}_t = \dot{U}_{t-1} + \frac{1}{2}(\ddot{U}_{t-1} + \ddot{U}_t)dt$$

$$U_t = U_{t-1} + \dot{U}_{t-1}dt + \frac{1}{2}\ddot{U}_{t-1}dt^2$$

Since, when using this scheme, it is not possible to constrain the position at time t, it will be done on the position at time $t+1$. This allows finding the contact forces that will induce an acceleration at time t in accordance with the Signorini's problem.

It is always possible to find vectors Q_{t-1}, R_{t-1}, S_{t-1}, gathering terms that can be computed at time $(t-1)$ to find the contribution of the contact force to U_{t+1}:

$$\ddot{U}_t = 2\frac{\dot{U}_t}{dt} + R_{t-1}$$

$$U_{t+1} = U_t + \dot{U}_t dt + \frac{1}{2}\left(2\frac{\dot{U}_t}{dt} + R_{t-1}\right)dt^2$$

$$2\frac{M}{dt}\dot{U}_t + D\dot{U}_t = F - K(U_t) + S_{t-1}$$

$$4\frac{M}{dt^2}U_{t+1} + 2\frac{D}{dt}U_{t+1} = F + Q_{t-1}$$

$$P_c = \left(2\frac{D}{dt} + 4\frac{M}{dt^2}\right)^{-1} F_c + P_{free}$$

To use implicit integration scheme with non-linear FEM, the system must be linearized at the current position. It is possible to compute it in an interactive simulation for virtual objects of low complexity (i.e. if the deformable model contains ∼20 nodes).

3.5 Resolution by classical LCP algorithms

First, we write the linear relation between position and contact forces. That is:

$$P_c = CF_c + P_{free}$$

where C can be considered as a compliance matrix.

As the contact forces apply to contact nodes, we can write a condensed system on theses nodes $(j, k \in [1, m])$:

$$\begin{bmatrix} \vdots \\ x_j \\ y_j \\ z_j \\ \vdots \end{bmatrix} = \begin{bmatrix} \ddots & & & & \\ & C^{jk}_{11} & C^{jk}_{12} & C^{jk}_{13} & \\ & C^{jk}_{21} & C^{jk}_{22} & C^{jk}_{23} & \\ & C^{jk}_{31} & C^{jk}_{32} & C^{jk}_{33} & \\ & & & & \ddots \end{bmatrix} \begin{bmatrix} \vdots \\ F^k_x \\ F^k_y \\ F^k_z \\ \vdots \end{bmatrix} + \begin{bmatrix} \vdots \\ x_i^{free} \\ y_i^{free} \\ z_i^{free} \\ \vdots \end{bmatrix}$$

where $C^{ij}_{\times\times}$ are coefficients of the C matrix (called C_{red} in its reduced form, and q^i_x, q^i_y, q^i_z are vector Q's coordinates.

The interpenetration distance δ_i of every contact can be express with all contact forces. Indeed:

$$\delta_i = (H_P C_{\text{red}} H_P^t)F_c + \delta_i^{free}$$

If the environment is deformable, in the same way, a reduced compliance matrix C_R of the environment can be found:

$$\delta_i = (H_P C_{\text{red}} H_P^t + H_R C_R H_R^t)F_c + \delta_i^{free}$$

This formulation leads to the following LCP form:

$$\begin{cases} F_c \geq 0 \\ \delta_i = (\Sigma H C_{\text{red}} H^T)F_c + \delta_i^{free} \\ F_c \perp \delta_i \end{cases}$$

If C_{red} is SPD, a principal pivoting method is used to solve the LCP and Lemke's algorithm for copositive matrix, see [12]. The obtained result are guaranteed to fits the Signorini's problem conditions.

4 Multi-threaded approach

In this section, we propose an efficient method to compute force feedback at required refresh rates (more than 500Hz). It allows to users manipulating deformable and virtual objects and experiencing haptic feedback during their interaction with rigid static objects of the virtual environment.

4.1 hypotheses

Collision detection, contact resolution and the FEM deformation may require important computational time. Using a single processor can compromise stability and transparency of the haptic control loop.

For most cases we experienced, the overall simulation runs at 100Hz. So, we make sure to fulfil time response requirements. The haptic feedback required frequency (500Hz) is obtained thanks to a buffering technique that consists in extracting the local contact's actual stiffness.

Nevertheless, this technique assumes two hypotheses in order to obtain intermediary right data between simulated states:

1. The really contact points and the contact configurations remain the same between two discrete simulation states.
2. The contact is frictionless (already mentioned). This condition will be eliminated in future works. Thorough study will take into account static and dynamic friction.

The deformable object is (virtually) attached to the haptic interface (since hold by a user) by mean of a set of nodes (see on figure 3, the O node's area). For such points, Dirichlet's conditions are applied. At each simulation step, the free motion configuration gives a contact space. The LCP resolution gives the non-nil contact forces F_i^0 and a constrained motion is deducted. These forces are transported to a point O, adequately positioned among the Direchlet's points, to create the force and the torque on the haptic interface. Both impedance or admittance control are available.

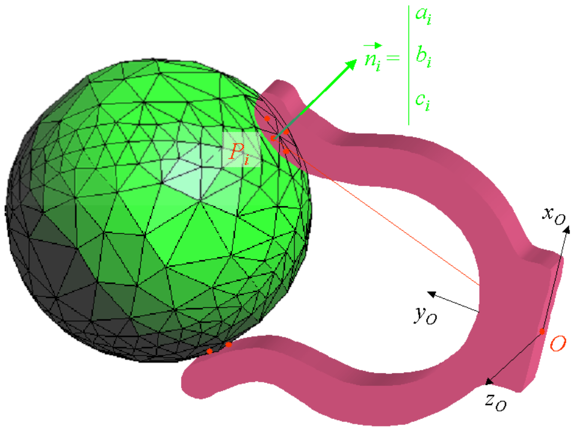


Figure 3. Deformable object in contact.

4.2 Approximation of the configuration between two simulation steps

There is no restriction concerning 3D motions of either the deformable virtual objects or the rigid ones. Consequently, moving the set of Direchlet's points and O modify the only initial position of the deformable object mesh points. Thus, value of δ_{free} is updated at each time step of the haptic loop. As this value depends on the contact configuration, the barycentric coordinates are shared to update transfer matrices H. The movement of the contact's normal vectors is updated in the haptic loop.

Since contact configuration will be the same between two discrete simulation steps, a condensed model can be extracted from the matrix of the deformation. This matrix model is in fact pre-computed in the two process loops. This allows reducing the communication bandwidth since only matrix indexes are sent from the simulation to the haptic loop.

In accordance with the cited hypothesis, every contact point remains in contact in the haptic loop. Indeed, $\delta_i = 0$ in this process[7]. The complementarity relation $(F_n \perp \delta)$ is always satisfied. This leads to a simple linear system resolution where contact normal forces are easily computed from:

$$\begin{cases} F_c > 0 \\ \delta = (\Sigma H C_{\text{red}} H^T) F_c + \delta_{free} = 0 \end{cases}$$

If F_c^i is found to be negative, we simply drop this force to zero since the contact is unilateral. This point i is eliminated from the contact set in the next simulation step.

5 Experimental results

This section describes actual implementation and obtained results of the proposed algorithm. First, the experimental setup and the virtual snap-in tasks are described. Then, obtained results from actual snap-in tasks achieved with haptic feedback are discussed. Only rigid/deformable contact simulations are presented here.

5.1 Snap-in tasks with haptic feedback

The experimental snap-in task case presented consists in deforming a simple clip on a tube (figure 5). The snap-in operation is a task where haptic feedback allows testing operator behaviour and studying performance and ergonomic aspects of such frequent maintenance, mounting and disassembling operations.

The originality of this simulation is that the user can hold the rigid tube as well as the deformable object with haptic feedback.

A FEM modal analysis of the clip (with PVC material characteristics) shows that the lower eigenfrequency is greater than 300Hz. As the deformable simulation is refreshed at 100Hz, we consider that the clip motions can be decomposed into a rigid dynamic motion of the global object surrounded with a static deformation part of the mesh. If necessary, to have better stability performance, the rigid motion created by the contact can be evaluated by transfer it in the contact space as in [16].

5.2 Experimental set-up

As haptic device, we are using a 6dof force feedback VIRTUOSE 6D [5] developed by CEA[8] and commercialized by HAPTION[9]. This haptic interface is linked to the dynamic simulation according to an admittance control

[7] Remark: If the maximum number of contact is reduced (less than ten in the same time) because of the geometry of the environment, it is possible to only share the configuration between the two loops and to solve the LCP in the haptic loop.

[8] Commissariat l'Energie Atomique (French Nuclear Authority).

[9] www.haption.com

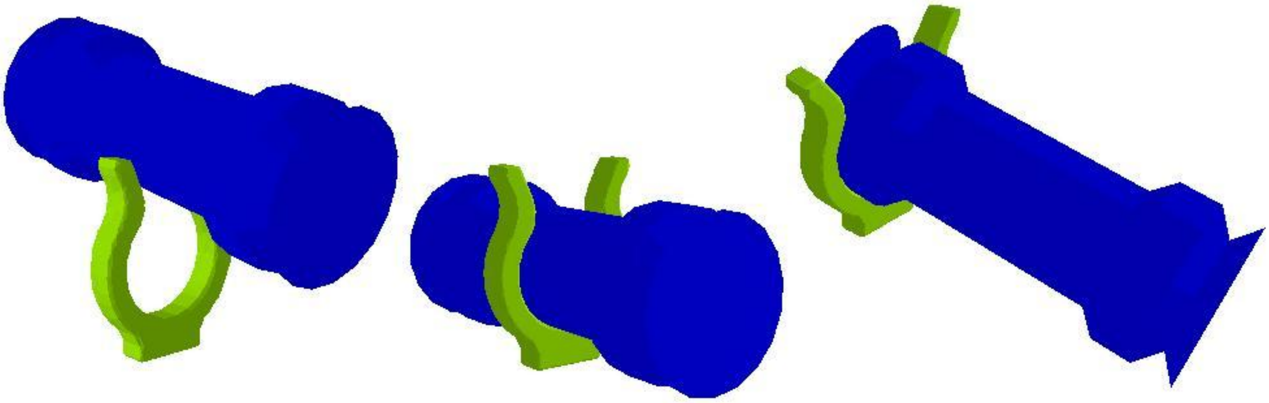


Figure 4. Snap-in operation

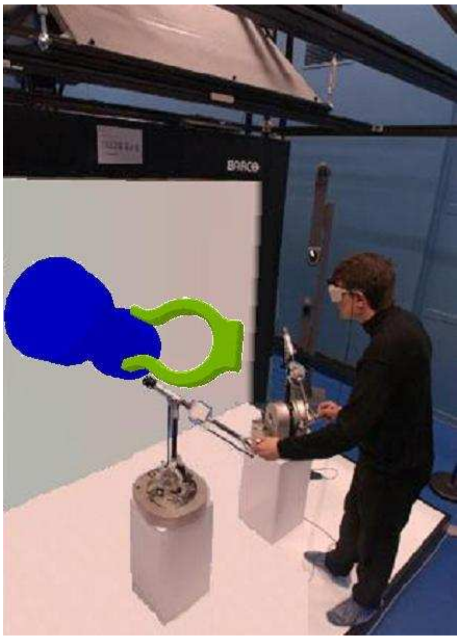


Figure 5. Experimental set-up

scheme. Consequently subsequent force feedback is computed based on the rigid motion part of the deformable clip. This simulation is updated every two milliseconds.

As explained before, the process that deals with the deformation computation is updated at slower rates. In theses experiments, the deformations are static and linear as proposed by the industrial characteristics of the task. In haptic loop process, a reduced model of the rigid/deformable contacts allow a good haptic perception. The virtual environment rendering is achieved in another parallel processing.

Visualization system is composed by two stereoscopic screen (horizontal and vertical). The graphic loop is also separated from the simulation and haptic loops. But the stereoscopic visualization is not implemented yet for deformable objects.

5.3 Results

Using a PC Pentium 4 computer at a 2.4GHz, the snap-in tool (the deformable object) is modelled by 639 nodes with 2395 linear thedraedron elements. The collision detection process takes 2.6ms; the first collision tests, less than 0.5 ms. The dynamic deformation and collision response computed by the LCP solver takes a mean time of 0.2ms and it never exceeds 1ms (even if the deformable object is highly constrained).

The contact reduced system, solved in the haptic loop, takes 0.1ms. The synchronization of the two loops is made respectively at 500Hz and 100Hz. It limits the communication time latency. The figure 6 shows that the contact force calculated by the reduced model and by the deformable simulation are nearly the same. That confirm our choices and the developed theoretical models.

6 Conclusion

This work shows a physically based approach that deals with the contact between deformable and rigid or deformable objects in an interactive haptic simulation. We have shown that the Signorini's problem can be expressed into a LCP formulation in the contact space. The multithreaded approach allows maintaining haptic good performance even if deformation algorithms are relatively slow.

Future work investigates contact with Coulomb's dry friction. Deformable/deformable haptic contact simulations will be presented soon.

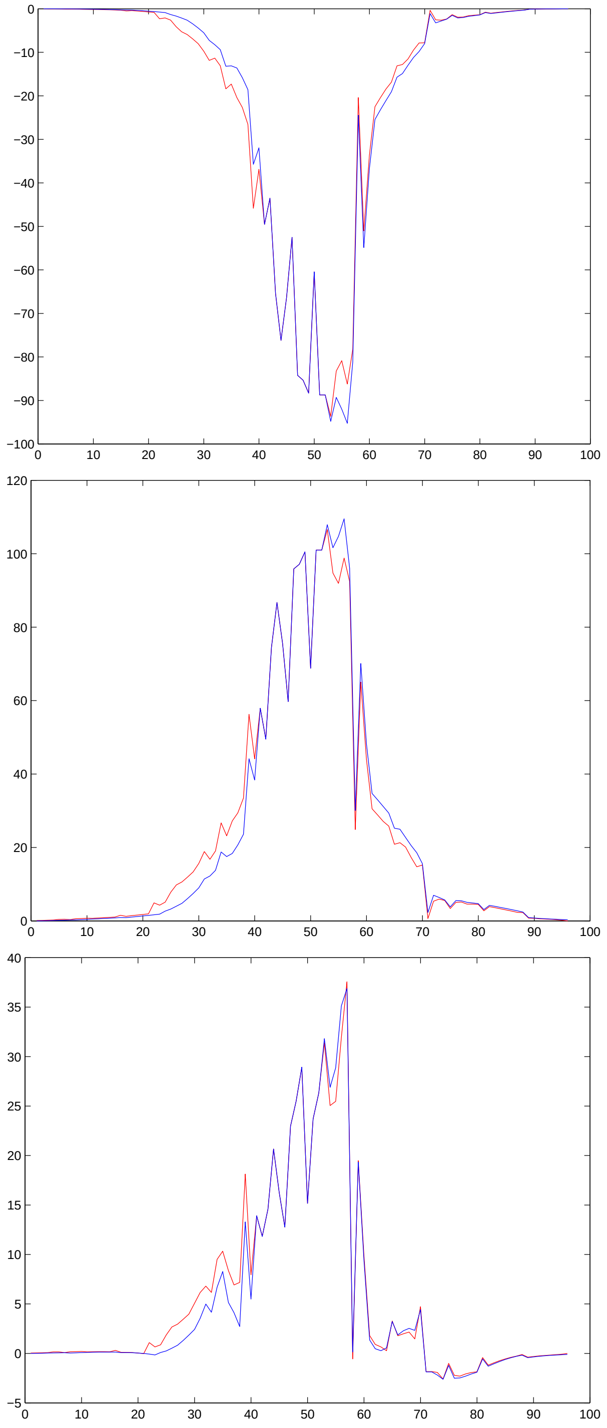


Figure 6. Haptic force along X, Y and Z (100 = 0.2N) calculated by the deformable loop (red) and by the latest haptic loop before the update (blue) during 100 steps of contact

References

[1] Bridson, R., Fedkiw, R., Anderson, J. : Robust Treatment of Collision, Contact and Friction for Cloth Animation. ACM SIGGRAPH 2002.

[2] Baraff, D. : Fast contact Force Computation for Nonpenetrating Rigid Bodies, ACM SIGGRAPH, Orlando, 1994.

[3] Debunne, G. : Animation multirésolution d'objets déformables en temps-réel, INPG PhD thesis, 2000.

[4] Fisher, S., Lin, M.C. : Fast Penetration Depth Estimation for Elastic Bodies Using Deformed Distance Fields, IEEE IROS, March 2001.

[5] Friconneau, J.P., Garrec, P., Gosselin, F., Riwan, A.: Satus and trends interface feedback master arms at CEA. IARP Nov 2002.

[6] James, D., Pai, D. ArtDefo : Accurate Real Time Deformable Objects. in Computer Graphics (SIGGRAPH 99 Conference Proceedings), 1999.

[7] Kikuchi, N., Oden, J.T. : Contact Problems in Elasticity : A Study of Variational Inequalitities and Finite Elements Methods. Ed. Siam, 1988

[8] Kühnapfel, U., akmak, H.K., Maa , H. : Endoscopic surgery training using virtual reality and deformable tissue simulation. Computer Graphics, No. 24, 2000.

[9] Laugier, C., Mendoza, C., Sundaraj, K. : Towards a realistic medical simulator using virtual environments and force feedback, International Symposium in Research Robotics ISRR-2001, Australia, 2001. In press in Advanced Robotics Series (ARS).

[10] Metzger, J., Kimmerle, S., Etzmu ,O. : Improved Collision Detection and Response Techniques for Cloth Animation. Proceedings of the Pacific Graphics 2002, p. 444.

[11] Müller, M., Dorsey, J., McMillan, L., Jagnow, R., Cutler, B., Stable Real-Time Deformations. ACM SIGGRAPH, pp 49-54, 2002.

[12] Murty, K.G., Linear Complementarity Programming. Internet Edition 1997.

[13] O'Brien, J.F. : Graphical Modeling and Animation of Fracture. Thesis, Georgia Institute of Technology, Juillet 2000

[14] Picinbono, G. : Modèles géométriques et physiques pour la simulation d'interventions chirurgicales, University of Nice Phd thesis, February, 2002.

[15] Provot, X. : Collision and self-collision handling in cloth model dedicated to design garment. Graphics Interface, 177-189, 1997.

[16] Ruspini, D.C., Kolarov, K., Khatib, O.: Haptic Interaction in Virtual Environments. Proc IROS 1997.

[17] Signorini, S.: Sopra akune questioni di elastostatica, Atti della Societa Italiana per il Progresso delle Scienze, 1933.

[18] van den Bergen, G. : Efficient Collision Detection of Complex Deformable Models using AABB Trees. Journal of Graphics Tools, 2(4):1-13 (1997),

[19] Zhuang, Y. : Real-time Simulation of Physically Realistic Global Deformations, UCB PhD thesis, Berkeley, 2000.