Interactive haptics for virtual prototyping of deformable objects: snap-in tasks case

Christan Duriez[1], Claude Andriot[1], and Abderrahmane Kheddar[2]

[1]CEA SRSI
Route du Panorama, 92000 Fontenay-aux-Roses, France
{christian.duriez, claude.andriot}@cea.fr
http://www.cea.fr
[2]Laboratoire Systèmes Complexes - University of Evry, Fre2494 CNRS
40, rue du Pelvoux, 91020 Evry, France
kheddar@iup.univ-evry.fr

Abstract. This work deals with computer haptics and virtual prototyping (VP) of complex deformable objects. This paper discusses the additional complexity induced when deformable virtual objects are considered. A preliminary proposed work concerns the pseudo-static small-deformation cases using FEM P-method combined with a LCP formulation of the Signorini's problem for contacts between deformable and rigid objects. As an implementation example of the proposed methodology, an interactive snap-in task is experienced.

1 Introduction

One of the most promising application of virtual reality techniques matters interactive virtual prototyping for industrial CAD-based process, digital mock-up evaluation and maintenance operation simulation and/or training (such as mounting/dismounting tasks). Many recent work addresses this issue for rigid bodies [12] including haptic feedback [20] [28].

The main purpose of this paper is to expand virtual prototyping to compliant virtual objects. Indeed, the first section focuses on a brief review of existing models to compute virtual deformations and handle contact between deforming objects.

A multi-resolution finite element method as applied to elasticity is considered and adapted to meet both accuracy in the contact space (P-method adaptation) and real-time computation (linear formulation). In this preliminary work, the only quasi-static case is considered. It is shown that, in the contact space Signorini's problem is formulated as a linear complementarity problem (LCP) [24].

The last part of the paper presents an actual virtual snap-in operation with haptic feedback to corroborate the theoretical developments.

2 A brief review

This section reviews briefly (i.e. in a not exhaustive way) two main computational aspects to deal with deformable virtual objects animation:

1. the deformation's formulation model that meets both fast haptics/vision rendering rates and actual physical law accuracy.
2. the contacts's formulation model when deformable objects interact (also with rigid or fluid objects).

2.1 Simulation of the deformation

In the computer graphics field, various methods were proposed in order to simulate (animate) virtual object deformations. These methods could be gathered in two categories:

- non-physically based methods;
- physically based methods.

The first class gathers the developed methods that do not have a conceivable physics science justification. Indeed, the main purpose of, theses models is to meet real-time calculation constraint, stability of the numerical integration and only a visual realism of the deformation. Geometrically-based deformation, also named free-form deformations [6], are instances of non-physically based models. In this case, object's deformation derive from object's related (exact or approximate, local or global) shape (for instance implicit equations or simple bounding volume), dynamically governed from external forces. In Faloutsos's deformation method [9], a physical meaning to this geometrical deformation is proposed. Typical other approaches considered to mesh virtual deformable objects as a collection of nodes connected by discrete mechanical components (e.g. mass-damper-springs methods [5]). A companion novel method considers object as collection of interacting particles governed by forces (similar to molecular interaction ones) [21].

In the computer graphics terminology, these methods are referred to as "physically-based methods". However, the only thing that fits physical laws accuracy is the core (or each local node) governing equations which do not adequately translate the appropriate physics laws of the continuum mechanics. Indeed, these methods present many drawbacks such as preserving anisotropy [4], setting the appropriate mechanical element's parameters, the global deformation behavior and stability is not guarantee…

In order to compute physically-realistic deformation behavior, the second class of method is considered. These methods are based on continuum mechanics governing laws (i.e. methods that drive links between the geometrical strain and the internal stress of the deformed material [15] [14]. However, modeling equations are not possible to solve analytically for the general object's shape. Nevertheless some approximate resolution methods, based on mathematical basis, exist. The Finite Element Method (FEM) is one of them. Its principle lies on meshing the deformable domain into a collection of simple-shaped interconnected elements (tetrahedron, hexahedron) for which deformation behavior is derived from actual continuum mechanics (i.e. from material physical parameters). Continuity between adjacent elements is made thanks to piecewise interpolation functions. The assembly of all elements leads to a global matrix differential system of the form:

$$M\ddot{U} + C\dot{U} + F_K(U) = F$$

Where M is the mass matrix, C is the global damping matrix, F_K the stiffness force corresponding to the internal stress, U is the nodal displacement vector and F is the external applied forces on each node. It is well known however that solving such a FEM obtained formulation is very time consuming because of the size of the mass, damping and stiffness matrices (the last must be computed and inverted each time step. In the linear case, when there is only 10% of relative deformation, $F_K(U)$ could be linearized to $F_K = KU$. In this case, off-line computing could be achieved to invert the system and to condensate the problem on surface nodes [2]. In [16], deformation of virtual objects is derived from the boundary element method (BEM) formulation.

If the relative deformation is over 10%, the problem could not be linearized. In this case, some solutions use an explicit finite element formulation of the problem (see [8] [31] or a slightly different approach using tensor-mass as in [26] which do not resolve the global matrix system using a local form. The stability of these methods depends however on the mass/stiffness ratio. A recent work shows that it is possible to solve the non-linear system thanks to a linearizing method around the each node position [23]. However, this last method is efficient only for systems having a small number of degrees of freedom.

Adaptive approaches may also be investigated to reduce the processing time. Indeed, the continuity of the formulation could be used to build a multi-resolution model by refining the object's mesh [8], or rather by refining the interpolation functions (i.e. putting more points in the element [174]. Finally, another FEM processing optimization, called the P-method, consists in adapting the basis interpolation functions order [7]. Our work derives from this last formulation.

There are indeed many "hybrid" modeling techniques that are intermediate between the first and the second categories. For instance, Laugier's team uses the so called Long Element Method to model living organ deformation. This model cross-assembles a set of `vertical' and `horizontal' parallelepiped rods that deformed on the basis of FEM formulation [19].

Although the referred modeling techniques for deformable objects are synthesized, this review is far from being exhaustive. The next section is dedicated to model formulation of contacts.

2.2 Contact’s characterization

The first step in any contact characterization is to detect if, when, and where objects collide. To deal with this computationally demanding problem in our simulations, we are using a discrete collision detection algorithm. The AABB (axis aligned bounding box) is used as bounding volume hierarchies to speed up the collision detection process [30]. After a collision is detected, it becomes necessary to compute subsequent forces and constraints acting on the system. These forces must deform the object in a realistic way (i.e. such that none-penetration constraints (if any) hold and that the deformation is physically truthful). To do this, we used the characterization of contact without friction between a deformable and a rigid bodies as defined in [17] (it is also known as "the Signorini's problem" [29]). By using inequalities and complementarily relations the non-penetration between objects, the Signorini's problem define both the forces distribution on the contact area and the relation between these contact forces and the surface displacement of the deformable object. This formulation is somehow similar to what was performed by [3] to model non-penetration rigid bodies simulation based on a linear complementarity problem formulation of the forces in the contact space.

When a contact occurs, penalty methods are often used as input to the deformation calculus preventing objects interpenetration. In this case, the forces calculation are related to the interpenetration volume [25] or to the interpenetration distance [10]. These obtained forces must ideally avoid deep interpenetration but without supplying additional energy to the system. In these methods, the force processing is straightforward. However, the non-penetration constraint could not be guaranteed (at the next step). Moreover the relation linking the penetration measure to the contact forces is not physically plausible (figure 1).

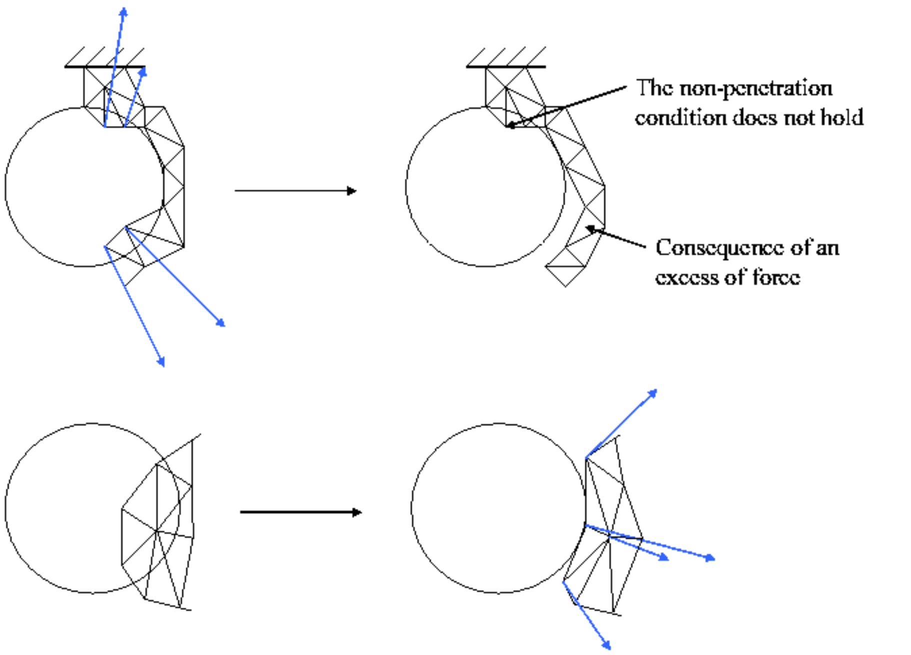

Fig. 1. Problems that may arise using penalty methods: constraints may not hold (above) or artificial forces applies on nodes that are not in contact (below).

Using a linear FEM for virtual surgery simulation, Kűhnapfel et al. [18] imposed on the node (involved in a collision with the surgeon's virtual tool) to follow its displacement. He derived the force to impose on the node in order to exactly achieve the imposed displacement. This computed force is used as an input of the deformation process of the whole model. In another work, Picinbono [26] proposed to impose a displacement of any node involved in a collision so that all nodes will be on the surface of the rigid body. In theses methods, the non-penetration constraint holds perfectly and show a good behaviour (namely for very deforming objects such as some human organs) but the contact forces do not match the Signorini's problem conditions. Indeed some "stick effect" appears during the displacement of colliding nodes.

In [22] [1], they propose to neutralize each colliding node's relative normal velocity. But such a method requires a continuous collision detection algorithm method (i.e. collision detection must be able to return the first collision time). In this case, the Signorini's problem conditions may not be guaranteed since the created forces are not necessarily normal to the surface (as in a frictionless contact case).

3 Multi-resolution FEM in linear elasticity

Our long term goal is to develop robust algorithms which allow haptic feedback for virtual prototyping of deformable objects. In this very first investigation, only the quasi-static case (small deformations) is considered. Moreover, we are convinced that computer haptics must be part of the simulation solver and not a set of “specific effects” (such as inertia effects, friction effects) that a user must include and set within a simulation. To do so however, it is important to guarantee real-time processing. In another hand, and since our application addresses industry, it is important to ensure physical coherence and physical realism in the haptic rendering of deformations. As a sample example, let us take a snap-in operation simulation (this operation is frequently used in automotive industry).

3.1 Process reduction by matrix analysis

Almost actual plastic materials, used in automotive snap-in assembly are structured materials. Then we can consider that the mass/inertia effects are minor relatively to the object internal stiffness. A quasi-static model is then adequate to this kind of manufacturing operations. After the meshing process and the stiffness calculus on every element, the system leads to the following matrix equation:

$$KU=F$$

where K is a $3n \times 3n$ matrix and n is the free-vertex number of the mesh. A vertex is said to be free if it is not subject to the Dirichlet's conditions. Let U be the displacement field vector of these nodes and F , the force vector. Using the matrix condensation method, a preference is given to the surface nodes "*s*" under the very reasonable hypothesis that the internal object nodes "*i*" are not subject to contact forces, that is:

$$\begin{bmatrix} K_{ss} & K_{si} \\ K_{is} & K_{ii} \end{bmatrix} \begin{bmatrix} U_s \\ U_i \end{bmatrix} = \begin{bmatrix} F_s \\ F_i = 0 \end{bmatrix}$$

As a consequence, a reduced matrix can be inverted off-line and the computation of the surface nodes displacement can be made using the following reduced system:

$$U_s = (K_{ss} - K_{si} K_{ii}^{-1} K_{is})^{-1} F_s$$

When the all matrix inversions are made, we compute in real-time the multiplication of the obtained matrix by the forces applied to the object's nodes:

$$K^{-1} F = U$$

In his work, Dan C. Popescu [27] proposes to use the propriety that the contact forces are nil on the points that are not in contact. Thus the computation could be restricted to:

$$K^{-1} F = \underbrace{\begin{bmatrix} C_{cc} & C_{cn} \\ C_{nc} & C_{nn} \end{bmatrix}}_{K^{-1}} \begin{bmatrix} f_c \\ 0 \end{bmatrix} = \begin{bmatrix} u_c \\ u_n \end{bmatrix}$$

where f_c represents the contact force (thus multiplications of C_{cn} and C_{nn} by $[0]$ are not performed).

3.2 P-Method adaptation

The FEM, as applied in elasticity problems, consists in the use of localized interpolation functions describing the displacement of any point behaving to the element by its node displacements. These functions can be either linear (in the case of a tetrahedral element it uses 4 nodes), quadratic (tetrahedral element with 10 nodes), cubic (tetrahedral element with 20 nodes) for a thorough description of the FEM method, see [7] and [15]. The higher is the interpolation function's degree the best is the accuracy of the results.
The so-called P-method consists in refining the interpolation displacement by increasing the order of the interpolation's functions especially in the areas where more accuracy is needed (for instance in the contact areas). The advantage of such an adaptation is in avoiding the meshing process that could be time consuming. To allow the continuity between the low and less degree interpolation frontiers, transition elements are introduced, see figure 2.

After a refinement, the system degree of freedom increases and the system's behavior governing equations must be built again. In most cases, one can not know at the early time-step where the contact will occur (i.e. where an increase of the interpolation degree is needed). In the other hand, the pre-processing time to compute the rigidity matrix of complex elements is made off-line and thus will not influence the simulation parameters (storage capabilities and integration time-step).

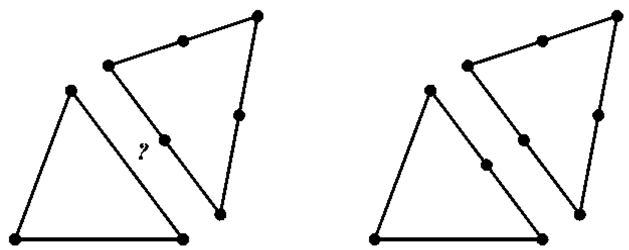

Fig. 2. Building a transition element to allow the continuity between a linear interpolation and a quadratic one.

Consequently all the virtual objects deformation parameters (such as the rigidity matrix) will be calculated using high order interpolation functions. So the system will have a very high number of degree of freedom. After the system inversion, we can decompose the system by separating the principal nodes p from the interface nodes i.

$$K^{-1}F = \begin{bmatrix} C_{pp} & C_{pi} \\ C_{ip} & C_{ii} \end{bmatrix} \begin{bmatrix} F_p \\ F_i \end{bmatrix} = \begin{bmatrix} U_p \\ U_i \end{bmatrix}$$

The principal nodes are the vertex of each element and the interface nodes are the ones which are on the edges (eventually on the faces) of each element. The proposed method consists in computing mainly the displacement of some privileged interface nodes. The displacements of the other interface nodes are calculated using those of the principal nodes. Thus, a separation is made between the free interface nodes i_f and the constraint interface nodes i_c which displacements depend on the principal nodes "*p*", $U_{i_c} = \vartheta(U_p)$. The i_c nodes are those which are not involved in the contact area (i.e. $F_{i_c} = 0$).

$$K^{-1}F = \begin{bmatrix} C_{pp} & C_{pi_f} & C_{pi_c} \\ C_{i_f p} & C_{i_f i_f} & C_{i_f i_c} \\ C_{i_c p} & C_{i_c i_f} & C_{i_c i_c} \end{bmatrix} \begin{bmatrix} F_p \\ F_{i_f} \\ F_{i_c} = 0 \end{bmatrix} = \begin{bmatrix} U_p \\ U_{i_f} \\ U_{i_c} = \vartheta(U_p) \end{bmatrix}$$

Consequently, on every element, where the displacement of all the interface nodes is calculated based on the principal nodes, the interpolation is linear. So far, our method is close to the P-method but still adapted to our application context that requires quick and real-time computations.

In fact, at each time-step, the algorithm chooses the free interface nodes i_f (actually the choice is only at the neighborhood of the contact computed in $o(n_i)$) and the displacements of the principal and free interface nodes are computed (at worst in $o((n_p + n_{i_f})^2)$) whereas the displacement of the other interface nodes are subsequently deduced in $o(n_{i_c})$.

Comparing to a linear interpolation, a quadratic interpolation of the entire object increases the accuracy of the deformation but is more time consuming. Consequently, the proposed method is an astute compromise between the quadratic interpolation accuracy on the contact zone with a near linear interpolation computation cost.

4 Rigid / deformable bodies contact

The section presents the Signorini's problem in a case of surface defined by planes and the algorithms used to solve it in a computing time compatible with haptic feedback requirements.

4.1 The pseudo-static problem definition

As previously stated, the stiffness mechanical part of the deformation behavior dominates the mass/inertia effects. This motivates a deformation model which takes into account only the static case. Nevertheless, since the collision detection and the contact characterization methods are using precedent time steps the formulation is said to be pseudo-static.

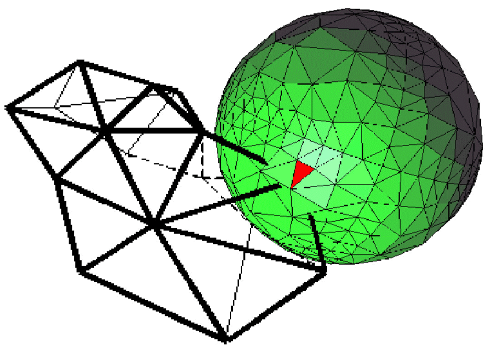

Fig. 3. One plane (triangle) is associated with the colliding node.

The detection collision algorithm [30] associates, to each vertex of the deformable mesh, an interpenetration distance (inside the rigid object) and an interpenetration direction These parameters give rise to a plane equation (see the figure 3). We want that the position of every colliding node passes above this plan after the collision response. This after collision state can be expressed using the following inequality:

$$a_i x_i + b_i y_i + c_i z_i + d_i \geq 0$$

The used detection algorithm is discrete[1], for the moment. To be sure that a colliding node is corrected associated to the appropriate colliding face, we should use a continuous collision detection method.

In this study, only the case of frictionless contact is considered. So the contact forces are projected along the surface normal. Moreover, these forces are directed towards the outside of the rigid object, so they are oriented:

$$\begin{pmatrix} F_n^i \geq 0 \\ F_t^i = 0 \end{pmatrix} \Leftrightarrow \begin{pmatrix} F_x^i \geq 0 \\ F_y^i = \dfrac{b_i}{a_i} F_x^i \\ F_z^i = \dfrac{c_i}{a_i} F_x^i \end{pmatrix}$$

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

$$(a_i x_i + b_i y_i + c_i z_i + d_i).F_n^i = 0$$

These conditions fits the Signorini's problem ones. We can re-write them in a classical LCP form.

4.2 Resolution by classical LCP algorithms

The collision detection algorithm gives a list of nodes actually interpenetrating the rigid body. Obviously, some of the them may be found to be not actually colliding (i.e. out of the contact) after computing the deformation (as a response to the collision). If m is the number of the mesh nodes that are in contact, we can write the following m-dimension vector X:

---

[1] Which simply means that collisions ae detected based on objects interpenetration. Discrete class algorithms make the well known hypothesis that if, for each time-step, no collision is detected then there was no collision between the considered time-steps.

$$X_i = a_i x_i + b_i y_i + c_i z_i + d_i = a_i x_i^0 + b_i y_i^0 + c_i z_i^0 + a_i u_i + b_i v_i + c_i w_i + d_i$$

where (x_i^0, y_i^0, z_i^0) is the position of the *i*-the node in the initial rest configuration and (u_i, v_i, w_i) are the displacement of the node referring to this configuration.

As the collision force firection is normal to the surface, the FEM system of equation is rewrite to a *m*-size form:

$$X = q + M.F_x$$

The obtained FEM system of equations gives the relation between the displacements (U) and the external forces (F) of all nodes. As the contact forces are only on the colliding nodes, we can write a condensed system on the colliding nodes $(i, j \in [1, m])$:

$$K^{-1}F \Rightarrow \begin{bmatrix} \vdots \\ u_i \\ v_i \\ w_i \\ \vdots \end{bmatrix} = \begin{bmatrix} \ddots & & & & \\ & C_{11}^{ij} & C_{12}^{ij} & C_{13}^{ij} & \\ & C_{21}^{ij} & C_{22}^{ij} & C_{23}^{ij} & \\ & C_{31}^{ij} & C_{32}^{ij} & C_{33}^{ij} & \\ & & & & \ddots \end{bmatrix} \begin{bmatrix} \vdots \\ F_x^j \\ F_y^j \\ F_z^j \\ \vdots \end{bmatrix}$$

The relation between the vector variables X and the force F_x is deducted by:

$$\begin{bmatrix} \vdots \\ X_i \\ \vdots \end{bmatrix} = \begin{bmatrix} \vdots \\ a_i x_i^0 + b_i y_i^0 + c_i z_i^0 + d_i \\ \vdots \end{bmatrix} \begin{bmatrix} \ddots & & \\ & \begin{matrix} a_i C_{11}^{ij} + b_i C_{12}^{ij} + c_i C_{13}^{ij} + \\ b_i C_{21}^{ij} + \frac{b_i^2}{a_i} C_{22}^{ij} + \frac{b_i c_i}{a_i} C_{23}^{ij} + \\ c_i C_{31}^{ij} + \frac{b_i c_i}{a_i} C_{32}^{ij} + \frac{c_i^2}{a_i} C_{33}^{ij} \end{matrix} & \\ & & \ddots \end{bmatrix} \begin{bmatrix} \vdots \\ F_x^j \\ \vdots \end{bmatrix}$$

In its compact form, this formulation leads to the following LCP form:

$$\begin{cases} X \geq 0 \\ F_x \geq 0 \\ X = q + M.F_x \\ (F_x)^t X = 0 \end{cases}$$

A principal pivoting method is used in order to solve the LCP, see [24]. And the result fits the SIgnorini's problem conditions.

5 Experimental results

This section describes the simulation and the implementation of the proposed algorithm. First, the experimental setup and its interfaces components used for interactive virtual prototyping are described. Then the results obtains from actual snap-in tasks experiments are discussed.

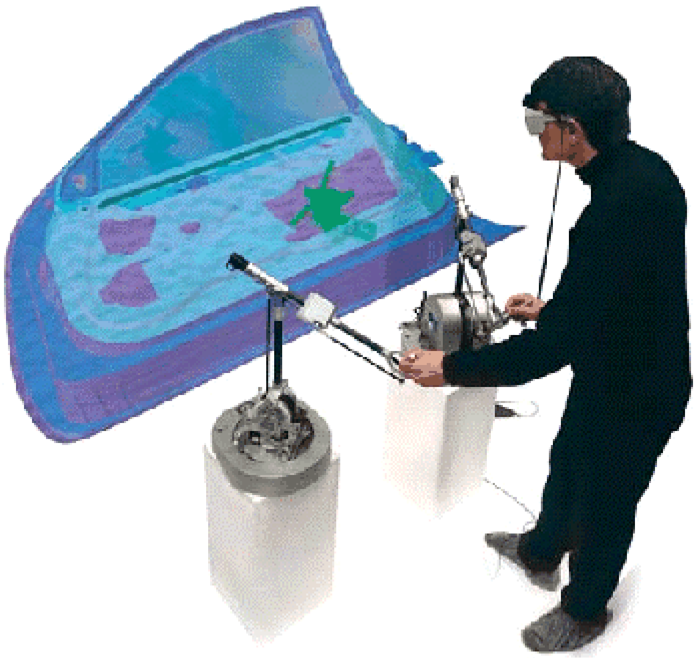

Fig. 4. The CEA PHARE virtual reality platform (rigid body prototyping instance): a stereoscopic workbench displays virtual environments, two haptic 6dof devices (VIRTUOSE 6D).

5.1 The CEA PHARE virtual reality platform

The CEA[2] platform is composed by a workbench visualization system with two stereoscopic screens (horizontal and vertical). The user interacts with the virtual objects thanks to two handed 6 dof haptic devices. The global system configuration is illustrated by the figure 4.

The used force feedback devices (VIRTUOSE 6D) [11] are also developed at the CEA and commercialized by HAPTION[3]. VIRTUOSE 6D is used as a 6dof motion input and generates 6dof force/torque as an output on the handle. The workspace is approximately a cube of 40cm side. The VIRTUOSE 6D can deliver a 10N continuous force (a 35N pick maximum force) with a resolution of 0.2N. The maximum torque is 3.1Nm with a resolution of $3x10^{-5}$rad. These performances allow a high fidelity display of virtual objects haptic characteristics.

5.2 Snap-in tests

The snap-in simulation example (see figure 5) consists in deforming simple clip on a tube. The snap-in operation is a task where the haptic feedback is important to allow to experience operator behavior and experience of such frequent maintenance, mounting and disassembling operations.

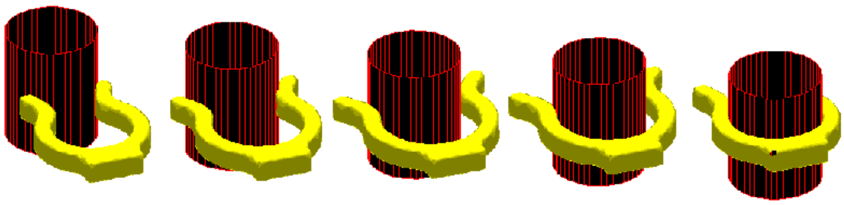

Fig. 5. Simulation frames of the snap-in simulation.

The user holds a virtual tube (cylinder or any other shaped object) linked to the haptic interface handle. The computing forces of the interaction contacts are applied on the dynamics of the controlled tube. The figure 5 shows picture snapshots successful simulations using Matlab. Afterwards, the software was ported on the PHARE

---

[2] Commissariat l’Energie Atomique (French Nuclear Authority)

[3] www.haption.com

platform in C++. Experimental results are very promising and confirm the proposed theoretical contribution of this paper.

Figure 6 shows screen snap-shot of an actual interactive snap-in experiments using the PHARE platform and the VIRTUOSE 6D haptic interface. The virtual interface terminal point is "linked" to the rigid object while the snap-in tool is static and vice-versa. Using a PC Pentium 4 computer at a 2.4GHz, the snap-in tool (the deformable object) is modeled by 686 nodes with 293 quadric elements. The collision detection process takes 2.5ms. The dynamic deformation and collision response takes 0.2ms for the LCP resolution and 0.01ms for the matrix multiplication. The virtual environment rendering is achieved in a parallel processing. Operator could experience intuitive and realistic interaction force feedback during snap-in operation.

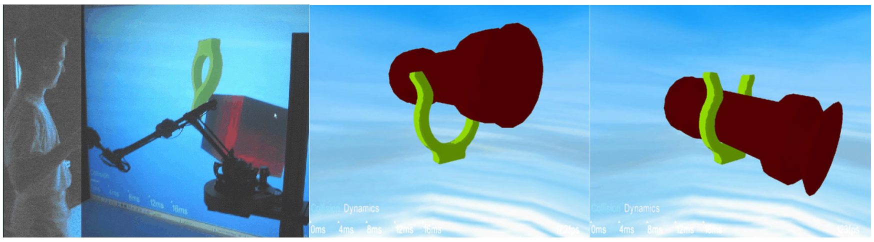

Fig. 6. Snap-shots from actual interactive snap-in operation with haptic feedback.

6 Conclusion

This preliminary work shows a possible approach to extend virtual prototyping to deforming objects. The algorithm adapts the quasi static finite element method to satisfy haptic feedback requirements. Future work will investigate a better definition of the P-method refinement localization and on a more general non-linear behavior. We have shown that the Signorini's problem can be treated thanks to a LCP formulation in the contact space. Still, a more physical definition of the contact must include friction and more robust collision detection algorithms.

References

1. Bridson, R., Fedkiw, R., Anderson, J. : Robust Treatment of Collision, Contact and Friction for Coth Animation. ACM SIGGRAPH 2002..
2. Bro-Nielsen, M., Cotin, S. : Real-time Volumentric Deformable Models for Surgery Simulation using Finite Elements and Condensation. Computer Graphics Forum 15 (1996), No. 3, C57-C66, C461.
3. Baraff, D. : Fast contact Force Computation for Nonpenetrating Rigid Bodies, ACM SIGGRAPH, Orlands, 1994.
4. Bourguignon, D., Cani, M.P. : Controlling Anisotroy in Mass-Spring Systems, Eurographics, Interlaken, Switzerland, August 2000.
5. Brown, J., Sorkin, S., Montgomery, K., Briyns, C., Stephanides, M.m Latombe, J-C. : Real-Time Simulation of Deformable Objects : Toola and Application. Proc. Computer Animation, Seoul, Korea, November 2001.
6. Chua, C., Neumann, U. : Hardware-Accelerated Free-Form Deformation. Euro-graphics, pp. 33-40, Interlaken, Switzerland, August 2000.
7. Cugnon, F. : Automatisation des calculs elements finis dans le cadre de la methode-P, these de doctorat, Université de Liège, mars 2000.
8. Debunne, G. : Animation multirésolution d'objets déformables en temps-réel, INPG PhD thesis, 2000.
9. Faloutsos, P., Van de Panne, M., Terzopoulos, D. : Dynamic Free-Form Deformations for Animation Synthesis. IEEE Transactions of visualization and computer graphics, July-September 1997.
10. Fisher, S., Lin, M.C. : Fast Penetration Depth Estimation for Elastic Bodies Using Deformed Distance Fields, IEEE IROS, March 2001.
11. Friconneau, J.P., Garrec, P., Gosselin, F., Riwan, A. : Satus and trends interface feedback master arms at CEA. IARP Nov 2002.
12. Gomes de Sa, A., Zachmann, G. : Virtual Reality as a Tool for Verification of Assembly and Maintenance Processes. In Computer and Graphics 23, 1999.
13. Grinspun, E., Krysl, P., Scröder, P. : CHARMS: A Simple Framework for Adaptive Simulation, ACM SIGGRAPH 2002.
14. Guarrugues, J. : Statique des solides élastique en petite deformations. www.esm2.imt-mrs.fr/gar/elas.html. Novembre 2001
15. Imbert, J.F. : Analyse des structures par elements finis. Cepadues Ed. 1984.

16. James, D., Pai, D., ArtDefo : Accurate Real Time Deformable Objects. In Computer Graphics (SIGGRAPH 99 Conference Proceedings), 1999.
17. Kikuchi, N., Oden, J.T. : Contact Problems in Elasticity : A Study of Variational Inequalities and Finite Elements Methods. Ed. Siam, 1988.
18. Kűhnapfel, U. Akmak, H.K., Maa, H. : Endoscopic surgery trainingusing virtual reality and deformable tissue simulation. Computer Graphics, No. 24, 2000.
19. 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).
20. Lécuyer, A., Kheddar, A., Coquillart, S., Graux, L., Coiffet, Ph. : A Haptic Prototype for the Simulations of Aeronautics Mounting/Unmounting Operations. IEEE ROMAN'2001, Bordeaux-Paris, France.
21. Luciani, A., Habibi, A., Vapillon, A., Duroc, Y. : A physical model of turbulent fluids. Eurographics, Netherlands, Septembre 1995.
22. Metzger, J., Kimmerle, S., Etzmu, O. : Improved Collision Detection and Response Techniques for Cloth Animation. Proceedings of the Pacific Graphics 2002, p. 444.
23. Műller, M., Dorsey, J., McMillan, L., Jagnow, R., Cutler, B., Stable Real-Time Deformations. ACM SIGGRAPH, pp49-54, 2002.
24. Murty, K.G., Linear Complementarity Programming. Internet Edition 1997.
25. O'Brien, J.F. : Graphical Modeling and Animation of Fracture. Thesis, Georgia Institute of Technology, Juillet 2000.
26. Picinbono, G. : Modèls géométriques et physiques pour la simulation d'interventions chirurgicales, University of Nice PhD Thesis, February 2002.
27. Popescu, D.C., Compton, M. : A Model for Efficient and Accurate Interaction with Elastic Objects in Haptic Virtual Environments. Graphite 2003, Melbourne.
28. Redon, S., Kheddar, A. and Coquillary, S. : Fast Continuous Collision Detection between Rigid Bodies. In proceedings of Eurographics. September 2002.
29. Signorini, S. : Sopra akune questioni di elastostatica, Atti della Societa Italiana per il Progresso delle Scienze, 1933.
30. van den Bergen, G. : Efficient Collision Detection of Complex Deformable Models using AABB Trees. Journal of Graphics Tools, 2(4):1-13 (1997).
31. Zhuang, Y. : Real-time Simulation of Physically Realistic Global Deformations, USB PhD thesis, Berkeley, 2000.