A Framework for the Haptic Rendering of the Human Hand

Panagiotis Stergiopoulos[1] Guillaume Moreau[2] Mehdi Ammi[1] Philippe Fuchs[1]

[1]Robotics Center-Ecole des Mines de Paris
60 bd St-Michel, 75272 Paris Cedex 06, France
e-mail : panagiotis.stergiopoulos@ensmp.fr

[2]CERMA UMR CNRS 1563
EAN - Rue Massenet BP 81931 44319 Nantes Cedex 3, France

Abstract

In this paper we present a framework that allows the haptic rendering of interactions between the human hand and a Virtual Environment consisting of polygonal objects. The proposed solution creates a representation of the human hand through simple geometric primitives (spheres, cylinders etc.) and allows the fast collision detection and calculation of forces of interaction. The forces are calculated with precision and smoothness, especially at the fingertips, where the human haptic perception is more sensitive. The area of each contacted region is taken into account to correctly weigh the importance of each force we calculate. The objective is to give a simulation tool that will allow the use of exoskeletons for the human hand with an improved level of precision, which is necessary for the correct evaluation of these haptic devices and their efficient use.

1. Introduction

Collision detection (CD) is a very important issue in the field of haptics. The last years there has been a considerable improvement on algorithms that treat with success the case of contacts between a virtual probe and the Virtual Environment (VE). Most of these solutions manage to maintain update rates for the haptic loop above the kHz threshold (which is considered essential for smooth and stable feedback [1]), even in models with tens of thousand polygons, [3], [7], [18]. There are also algorithms applied to models constructed with NURBS surfaces, which allows to directly use CAD designs, [12], [13], [20].

Contact between surfaces is a much more difficult problem and there is a small number of algorithms that run fast enough to be applicable in haptics [5], [9]. Some algorithms choose the use of an intermediate representation for the smaller objects [11], [17]. They propose the use of a shell of points that approximates the form of the original object and then contacts are checked individually for each point.

Although there are already a number of interfaces that offer haptic feedback to the human hand and the fingers, the algorithms for detecting the interactions of the hand with the VE are still not fully developed. This limits the utility of these interfaces as there is still no solution that allows to simulate with precision the actions one can make with his hand. Some solutions that take into account only the fingertips of two or more fingers as point probes have been proposed, [8], [10], [19]. However, these solutions limit the realism of the simulation as the orientation of the fingers and the hand is ignored and so the correlation between the manipulations in the VE and the ones in reality is not good.

Still, there have been some efforts for full hand CD. Popescu et al use a mesh of points that represent the main parts of the fingertip and they apply point-surface CD algorithms for each point of the mesh, [15]. Tremblay et al propose a framework in which the hand is separated into spheres that cover all its volume and he rapidly finds contacts between them and bounding boxes that approximate the form of the objects in the VE, [22]. Nelson et al work on the problem in the case where the finger is represented by parametric (NURBS) surfaces, [14].

In this paper we present a framework that proposes a simplified, yet relatively realistic model of the human hand. The CD algorithm we have developed for use with this model, can follow the interactions between the hand and a polygonal VE (consisting of solid non-deformable objects), at a very fast rate. The forces and torques are calculated by taking also into consideration the area of each contacted region for giving to each individual calculated force the correct weight. The framework allows different levels of precision for different parts of the hand, taking into account the sensitivity of human perception and its utility for the manipulation of the VE (more intense on the fingertips, less important along the phalanges or the back of the hand). Our objective is to use this algorithm for the haptic rendering of VR simulations employing the new force feedback hand exoskeleton that is currently under development in the Robotics Center of Ecole des Mines de Paris.

2. Theoretical Approach

2.1. Geometric representation of the human hand

Our objective is to detect contacts rapidly and to calculate the resulting forces with the necessary precision. In order to accelerate the CD, we create a simplified model of the hand. Each phalange of the finger is represented by a cylinder, while the fingertips and the joints are represented by spheres (Figure 1). In this first approach the palm is represented as a plane, but this could be easily substituted by a polygonal surface. Such a representation is not considered indispensable for the moment as the region of the palm is not used for delicate manipulations (for precision grips and for feeling the form of a surface, the fingertips are mostly used).

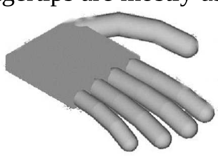

Figure 1. Model of the human hand

The use of spheres at the joints of the fingers is for covering gaps that occur when the fingers move and the angle between the phalanges changes (Figure 2). It is important to avoid these gaps as they could make us miss a collision of the finger with an object and cause unstable force rendering.

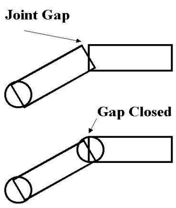

Figure 2. Closing joint gaps with a sphere

This approach allows to rapidly change the position and orientation of the hand and the fingers, as we just have to update the coordinates of a very small number of primitives. If the hand was constructed through a polygon mesh, the update of the position of each polygon would be necessary.

An advantage of this representation is that these primitives are closed forms and it is thus easy to determine if the triangles of the surface of an object are intersecting the hand and to calculate collisions and depths of penetration.

Another advantage is the easiness for using algorithms of different precision for the different parts of the hand if necessary. Precision of the intensity and direction of calculated forces is useful when we try to perceive the form of an object or when we make an action that demands high accuracy (e.g. a medical operation). Practically, the fingertips are used for probing the VE and we perceive most of the forms through them. On the other hand, the phalanges are mostly used either when we collide with objects in our path (during precise actions) or when we use a force grasp (e.g. when we hold a hammer or we pull a lever). In both cases, we don't need too much precision as our aim is to constrain an object in our hand or our movements to be constrained/blocked by the environment. We can also explain this with physiological arguments, as the hand is more sensitive at the fingertips and less along the phalanges or very few at the back part of the hand, so forces and textures are much better perceived by the fingertips. For the collision detection, we use only the part of the spheres that is not intersected with its neighboring cylinders (e.g. we take into account only one hemisphere for the fingertips). We do this to avoid defining 2 times the contact for the same triangle. In the current study we have used solutions of similar high precision both for the fingertips and the rest of the hand.

2.2. Collision detection

The collision detection algorithm is separated in 2 steps. The first step makes use of a hierarchical structure that allows the fast localization of the regions where there is a possibility of contact. In our case we propose the simultaneous use of a harsh grid partitioning and then, inside each box of the grid, the use of bounding sphere tree (Figure 3). Grid partitioning is useful when we have a large virtual environment (and as a consequence each grid box is approximately the same size as the hand). Otherwise, it is less efficient as it gets more complicated to find all the grid boxes that contain a part of the hand and for small environments it is preferable to use directly the bounding sphere tree.

We take care for the last leaf sphere of the sphere tree to contain one or just a few triangles. Although the spheres do not bound efficiently the triangles (the ratio of the sphere volume to the bounded triangle is generally higher than for other bounding volumes), they have the advantage of very rapid intersection testing, so a large number of individual volumes can be rapidly checked. The cylinders/phalanges are in their turn bounded by a big bounding sphere which contains a small number of leaf spheres (Figure 4).

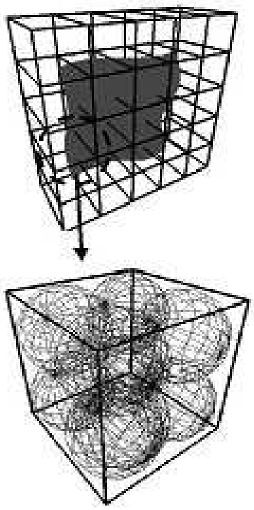

Figure 3. Grid partitioning and bounding sphere tree

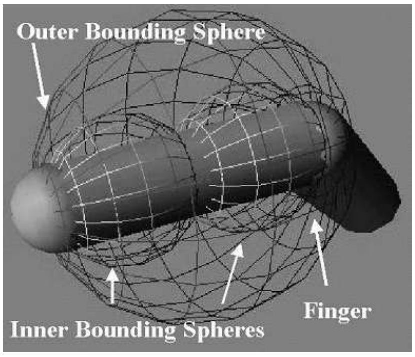

Figure 4. Bounding spheres for the finger phalange

During this first test, each of the spheres of the hand (fingertip, joint spheres and bounding spheres) is checked for intersection with the bounding sphere tree of the area of the VE that is in the neighborhood of the appropriate grid box (in which the hand lies). After an intersection with one of the leaf spheres of the phalange is detected, we continue the intersection test directly between the phalange cylinder and the bounding sphere tree of the virtual object. If an intersection is detected between the hand and a leaf sphere of the VE, we consider that the triangles contained in that sphere can be in contact with the hand.

The second step consists of the CD between the triangles contained in the previously mentioned leaf spheres and the possibly contacted part of the hand.

Finding contact with a part of a sphere is quite simple (the method is presented in [6]). For the cylinders, we find the distance D_{TC} between the triangle and each of the two bases of the cylinder by the Equation 1:

$$D_{TCi} = \overrightarrow{N_T} . \overrightarrow{O_i A} \qquad (1)$$

where N_T is the triangle normal vector and O_iA the vector formed by the cylinder base center O_i and the vertex A of the triangle. D_{TCi} is negative when the cylinder base center is above the triangle. We keep the smaller of the 2 values for the D_{TC}. There can be contact if:

$$-R \cdot \sin(f_1) < D_{TC\min} < H \cdot \cos(f_1) + R \cdot \sin(f_1) \qquad (2)$$

where f_1 is the angle between the triangle normal and the axis of the cylinder, R the radius of the cylinder and H the height of the cylinder. If the above conditions of Equation 2 are true, we search for an intersection between each edge of the triangle and the form of the section of the cylinder on the plane of the triangle. The shape of the section can vary depending on the angle f_1 and the distance D_{TC} (Figure 5). It can be a rectangle, a circle, an ellipse or part of an ellipse.

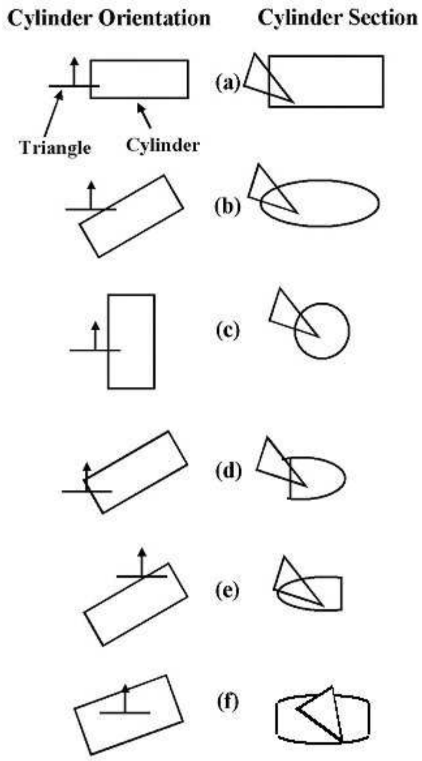

Figure 5. Shape of cylinder section depending on the orientation f_1 and height H of the cylinder with respect to the triangle

If there is no intersection between the triangle and the section of the cylinder, we check if the center of the section (i.e. center of the circle, ellipse or rectangle) is inside the triangle, in which case the triangle contacts the cylinder, otherwise there is no contact.

For each contact detected we need to determine 3 factors: a) The maximum depth of penetration dl_{max}, b) the area of the part of the contacted triangle that intersects the hand, which we call contacted surface A_C (Figure 6), and

c) the center of gravity (GC) of that part. The GC is calculated with respect to the contacted surface and not to the whole surface of the triangle.

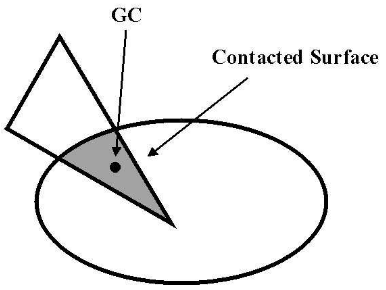

Figure 6. Contacted surface definition

2.3. Penetration depth and force calculation

For finding the dl_{max} we take into account not only the depth inside the specific primitive, but the whole distance in the direction of the normal of the triangle, till the exterior surface of the finger (Figure 7). Although it is more time consuming, it is necessary to use the whole volume of the finger for calculating the dl_{max}, otherwise force discontinuities could occur (e.g. when a triangle passes from the fingertip hemisphere into the distal phalange cylinder). With the dl_{max} we calculate the force per surface unit, with which it is related proportionally (Equation 3):

$$F_i = K_S \cdot dl_{\max} \quad (3)$$

where F_i is the force per surface unit for the ith triangle and K_s is the stiffness (unit of force per unit of volume) of the virtual object. The force vector has the direction of the triangle normal, although we could also use "force shading" methods, if such a solution is desired, [7].

The flow diagram of the algorithm is presented in Figure 8.

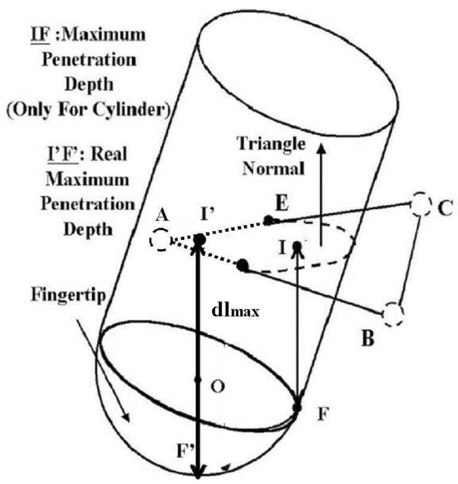

Figure 7. Projection depth calculation for a cylinder-triangle intersection

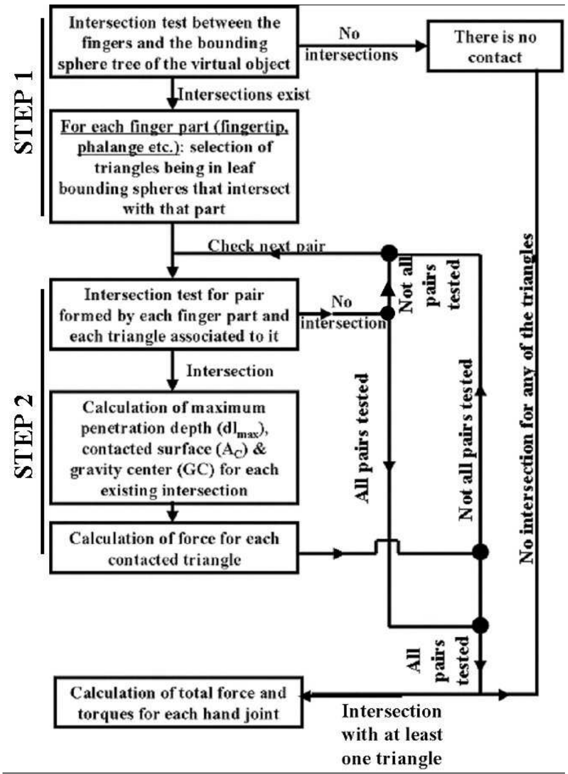

Figure 8. Algorithmic flow diagram

2.4. Total force calculation

It is very important to calculate the total force in an efficient way to avoid important deviations of the force value. This problem mainly raises for three reasons: a) a triangle that is colliding with the hand does not always fully penetrate in the hand, b) the size of the surface triangles can vary importantly in a VE and c) each finger can simultaneously be in contact with many triangles. For the same penetration depth, the resulting force should be more important when the contacted surface A_C is larger. This principle helps us avoid less realistic force calculations as the one presented in Figure 9a. If the individual forces for each contacted triangle were just added without taking A_C into account, a force resulting from a large contacted surface (e.g. a very big triangle) would be equally important with a force calculated from a smaller contacted surface.

We propose the following Equation 4:

$$\overrightarrow{F_T} = \sum_{i=1}^{k} A_{Ci} \cdot \overrightarrow{F_i} \quad (4)$$

where F_T is the resulting total force, A_{Ci} the contacted surface for the i^{th} triangle and k is the total number of contacted triangles. In this case the forces will be appropriately weighed and the vector of the total force will be calculated with more precision (Figure 9b).

In a similar manner we can calculate the total torque that is applied to each of the joints of the fingers and the hand. The total torque for a specific joint is given by Equation 5:

$$\overrightarrow{T_{Tj}} = \sum_{i=1}^{m_j} A_{Ci} \cdot (\overrightarrow{l_{iAJj}} x \overrightarrow{F_i}) \qquad (5)$$

where T_{Tj} is the resulting total torque for the j^{th} joint, l_{ij} is the vector formed by the center of gravity of the contacted surface and the j^{th} joint and m_j is the number of contacts that contribute to the creation of T_{Tj}.

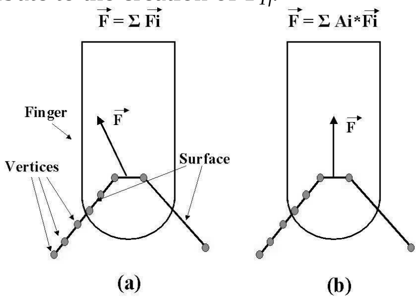

Figure 9. a) False and b) correct force calculation

3. Experimental results

Our experiments have the aim of proving 2 factors that are important for a haptic rendering algorithm:

a) The smoothness and continuity of calculated forces.

b) The speed of the CD algorithm.

For validating the first factor we've realized 2 tests. In general, we consider that the stability is high when it gives practically the same force for the same penetration depth and contacted surface, even when the number of contacts changes considerably. The first test consisted in sliding a virtual finger over a flat surface (Figure 10). During its movement the finger wasn't getting always in contact with the same number of triangles and for each triangle the dl_{max} differed. Nevertheless, as we can see in Figure 11, the variation of force was quite small, in the limits allowed by the human haptic perception. It is worthwhile to notice that even for these variations there was never a sudden change of force, which makes the difference even less perceptible.

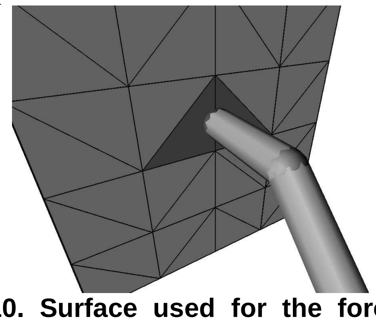

Figure 10. Surface used for the force stability experiment

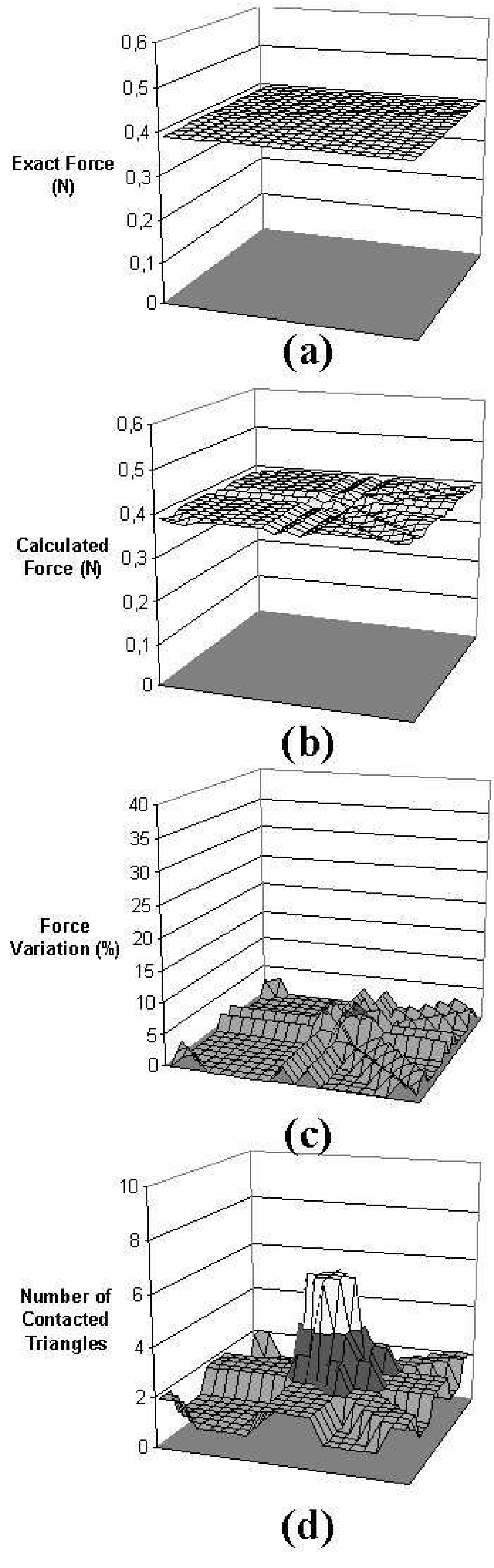

Figure 11. Results of force stability tests: a) theoretical force, b) calculated force, c) force variation and d) number of contacted triangles

Our second test aimed to validate the smoothness with which force size and direction changed when the finger moved on a surface that formed a slope (Figure 12). The resulting forces that are presented in Figure 13, show that here also there is a smooth transition of the direction of the force and there is no sudden change even when the number of contacted triangles varies.

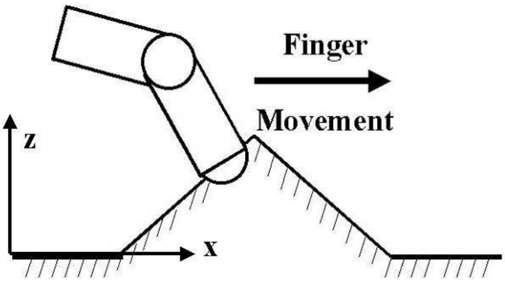

Figure 12. Edge contact test

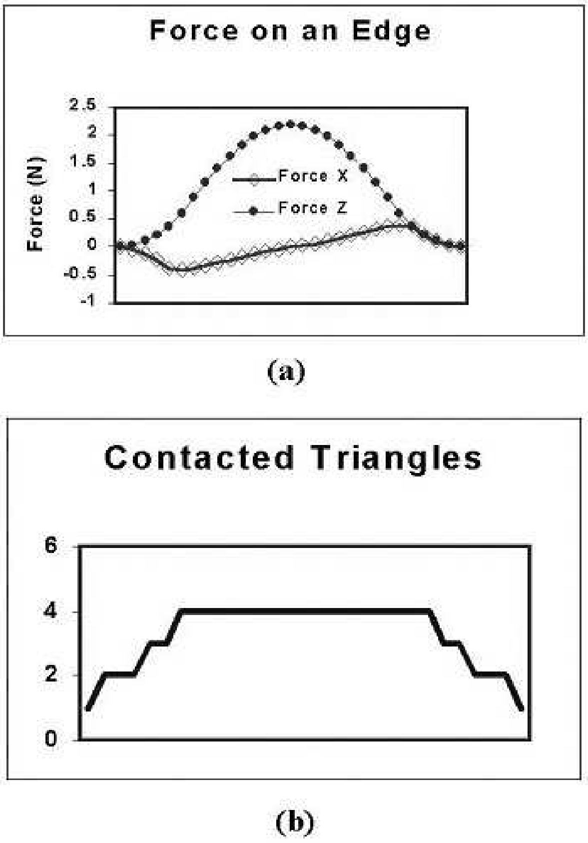

Figure 13. a) Resulting force for sliding on a slope and b) Number of triangles contacted during each calculation step

It is a fact that the haptic update rate is directly affected by the size of the VE and the number of simultaneous contacts. On the other hand, the grid partitioning and bounding sphere tree that we use, help to accelerate the haptic loop, by rapidly eliminating pairs obviously not colliding.

The time for individual intersection tests is given in Table 1. The results show that the procedure for determining individual contacts and defining their characteristics is very rapid, much below a millisecond, that is necessary for achieving a haptic rate of kHz level. We have also calculated the haptic loop rate for tests done on number of different surfaces. The tests were executed on a Pentium IV 1.7GHz PC with 512MB RAM. The results are presented in Table 2.

Table 1. Collision detection tests results (in μs)

Average Intersection Test Time for Sphere-Triangle Pair	6.2
Average Intersection Test Time for Cylinder-Triangle Pair	7.1
Average Intersection Test and Force Calculation Time for Sphere-Triangle Pair	9.7
Average Intersection Test and Force Calculation Time for Cylinder-Triangle Pair	16.3

Table 2. Haptic loop rate test

Test N°	Number of triangles	Levels of Bounding Spheres	Average Number of Contacted Triangles	Average haptic rate during contact (Hz)
1	162	0	2	650
2	162	6	2	3554
3	1680	0	9	66
4	1680	3	9	1263
5	1680	5	11	2034
6	4800	5	24	893
7	4800	6	23	2512
8	4800	6	48	909
9	9500	6	15	2086
10	9500	6	52	714

The results in Table 2 show two important facts. First of all, the correct use of the bounding sphere tree considerably accelerates the CD. We can notice also that more levels of bounding spheres increase the speed of the algorithm as the pairs of spheres/triangles and cylinders/triangles that need to be examined is reduced. There is of course a logical limit to the increase in efficiency when the levels of the sphere tree increase, as after a certain point there are too many leaf spheres that contain parts of the same triangle. Too many levels also increase the memory requirements (and slow down the computer performance). In any case, one can consider using other types of bounding volumes, depending on the application and the particularities of the VE.

The second fact is that when the hand is simultaneously in contact with an important number of triangles, the algorithm slows down considerably. This problem can not be avoided, as for all contacted triangles we have to calculate the force and torque. The only way to avoid this problem is to use a simplified force calculation method that needs less time per intersecting pair than the one shown in Table 1. We could also use the complete force calculation algorithm only at the fingertips and a less time-consuming method for the rest of the hand, as explained in Paragraph 2. Whichever the choice, using a simpler force calculation method will lead to a lack of precision and less smooth calculated forces, but that is a tradeoff that can not be avoided if we need to accelerate our application.

4. Conclusions-further work

The algorithm presented in this paper proposes a solution for the haptic rendering of the human hand. It approximates the form of the hand with a combination of geometric primitives. This simplified form allows the rapid detection of intersections and the efficient separation of zones that could be used with CD algorithms of different precision, depending on the sensitivity of the human hand. This enables the decoupling of CD for different parts of the finger when it is necessary to accelerate the force calculation procedure.

The first tests show that all the necessary calculations per pair (sphere-triangle or cylinder-triangle) are feasible in a few μs. Although there is an almost proportional relationship between the number of contacts and the speed of the algorithm, for scenarios that involve a logical number of contacts, the algorithm can give the desired update rate and relatively smooth force rendering (Figure 14). The algorithm however can still slow down for scenarios that involve too many simultaneous contacts (e.g. when all fingers are contacting an object and the triangles of its surface are very small compared to the size of the hand).

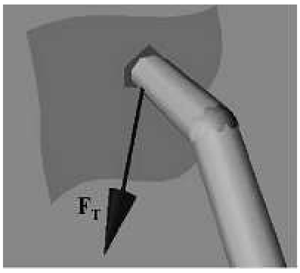

Figure 14. Smooth rendering of total force on the fingertip with multiple triangles contacted

In the future there are some issues to be considered and solved. When the total contacted surface is too large (as when we touch an object with the palm), the total force given by Equation 4 tends to be quite high. We are considering to also employ a contact clustering algorithm (as in [9]). This however demands further exploration. Till now we have also assumed that due to the speed of the haptic loop and the thickness of the fingers and hand, it is extremely difficult for the hand to pass through a surface without being restrained by it. In case such a problem exists (e.g. it can appear when the contact force is too small for constraining the finger on the object surface), a solution is to keep in the memory every pair of primitives that are in contact, so during the next iteration it should be tested if the finger is still in the surface of the virtual object, even if they don't intersect (Figure 15).

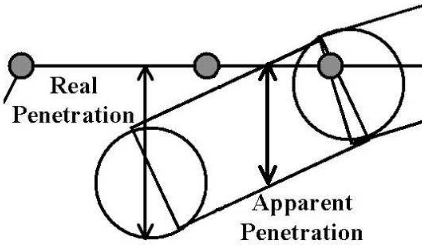

Figure 15. Fingertip passing through the surface

Nevertheless, this case is quite difficult to appear due to the important size of the fingers compared to the penetration depth for which a constraining force is being applied on the hand. So the necessity of a time-consuming solution needs further investigation. The most simple way to find that out is by implementing it with an existing exoskeleton for force feedback and eventually the algorithm will be tested with the new hand exoskeleton force feedback device designed in the Robotics Center of Ecole des Mines de Paris.

Finally, we must consider the use of algorithms that will allow the dynamic manipulation of objects (such as different types of grasps) and will take into account effects like friction. In this way, we could achieve the simulation of an environment with improved precision that will lead to the execution of tasks in a VE with more realism.

5. Acknowledgments

The work presented in this paper is a part of the research done for the PERF-RV Project, A5 Action, which is a subsidy of the French Ministry of Research and Technology. Its objective is the integration of force feedback in a VR simulation for the study of the ergonomics of vehicles. The first author was also supported by a Scholarship of the Alexander S. Onassis Public Benefit Foundation.

6. References

[1] Brooks, T.L., "Telerobotic response requirements", Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, Los Angeles, 1990, pp. 113-120.

[2] Gottschalk, S., Lin, M. C., Manocha, D. "OBBtree: A hierarchical structure for rapid interference detection", Proceedings of ACM SIGGRAPH 1996, pp. 171-180.

[3] Gregory, A., Lin, M. C., Gottschalk, S., Taylor, R., "Fast and accurate collision detection for haptic interaction using a three degree-of-freedom force-feedback device", Computational Geometry 15(1-3), 2000, pp. 69-89.

[4] Gregory, A., Lin, M. C., Gottschalk, S., Taylor, R., "A framework for fast and accurate collision detection for haptic interaction", Virtual Reality Annual International Symposium (VRAIS) '99, 1999, pp. 38-45.

[5] Gregory A., Mascarenhas A., Ehmann S., Lin M., Manocha D., "Six degree-of-freedom haptic display of polygonal models", in Proceedings of IEEE Visualization, 2000.
http://www.cs.unc.edu/~geom/papers/chron.shtml

[6] Held M., "ERIT-A collection of efficient and reliable intersection tests", Journal of Graphics Tools 1998.

[7] Ho, C.-H., Basdogan, C., Srinivasan, M. A., "Efficient point-based rendering techniques for haptic display of virtual objects", Presence, Vol. 8, No. 5, October 1999, MIT, pp. 477-491.

[8] Johnson D. E., Cohen E., "An improved method for haptic tracing of a sculptured surface", in Proceedings of the 7th Annual Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, (Anaheim, CA), ASME, November 1998, pp. 243-248.

[9] Kim Y. J., Otaduy M. A., Lin M. C., Manocha D., "Six-degree-of-freedom haptic display using localized contact computations", in the Proceedings of the 10th Symposium of Haptic Interfaces for Virtual Environment and Teleoperator Systems, March 24-25, 2002.

[10] Maekawa, H., Hollerbach, J. M., "Haptic display for object grasping and manipulating in virtual environment", Proceedings of the 1998 IEEE International Conference on Robotics & Automation, May 1998, pp. 2566-2573.

[11] McNeely W. A., Puterbaugh K. D., Troy J. J., "Six Degree-of-Freedom Haptic Rendering Using Voxel Sampling", Proceedings of ACM SIGGRAPH, 1999, pp. 401-408.

[12] Nahvi, A., Nelson, D. D., Hollerbach, J. M., Jonhson, D. E., "Haptic manipulation of virtual mechanisms from mechanical CAD designs", Proceedings of the 1998 IEEE International Conference on Robotics & Automation, May 1998, pp. 375-380.

[13] Nelson D. D., Cohen E., "Optimization-based virtual surface contact manipulation at force control rates", Proceedings of the Virtual Reality 2000 Conference (VR'00), 18-22 March, 2000, New Brunswick, New Jersey, USA, pp. 37-44.

[14] Nelson, D. D., Jonhson, D. E., Cohen, E., "Haptic rendering of surface-to-surface sculpted model interaction", Proceedings of the 8th Annual Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, (Nashville, TN), ASME, November 1999.
http://www.cs.utah.edu/gdc/publications/

[15] Popescu V., Burdea G., Bouzit M. "Virtual Reality Simulation Modeling for a Haptic Glove", Computer Animation'99 Conference, Geneva, Switzerland, pp.195-200.

[16] Redon, S., Kheddar, A., Coquillart, S., "CONTACT: arbitrary in-between motions for collision detection", IEEE ROMAN'2001, Bordeaux-Paris, France, 2001.
http://www-rocq.inria.fr/i3d/i3d/Publications_fr.htm

[17] Renz M., Preusche C., Pötke M., Kriegel H.-P., Hirzinger G., "Stable Haptic Interaction with Virtual Environments Using an Adapted Voxmap-PointShell Algorithm", Proceedings of the Eurohaptics Conference, Birmingham, UK, 2001.

[18] Stergiopoulos P., Moreau G., Lourdeaux D., Fuchs P. "Collision detection for haptic interaction with solid objects and virtual prototypes ", Proceedings of the Virtual Reality International Conference, Laval, June 2002, pp. 151-158.

[19] Thompson II, T. V., Cohen, E., "Direct haptic rendering of complex trimmed NURBS models", Proceedings of the 8th Annual Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, (Nashville, TN), ASME, November 1999.
http://www.cs.utah.edu/gdc/publications/

[20] Thompson II, T. V., Johnson, D. E., Cohen, E., "Direct haptic rendering of sculptured models", Proceedings of the Symposium on Interactive 3D Graphics, Providence, RI, April 27-30, 1997, pp. 167-176.

[21] Thompson II, T. V., Johnson, D. E., Cohen, E., Hollerbach, J. M., "Maneuverable NURBS models within a haptic virtual environment", Proceedings of the 6th annual Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, (Dallas, TX), ASME, November 1997, pp. 37-44.

[22] Tremblay M. R., Ullrich C., Gomez D. H., Carmel R., Tian J., "Whole-Hand Interaction with 3D Environments", Workshop on Perceptual User Interfaces, San Francisco, CA, 1998.