Virtual Grasping of Deformable Objects with Exact Contact Friction in Real Time

Jérémie Le Garrec Claude Andriot Xavier Merlhiot
CEA LIST SCRI
Fontenay-aux-Roses, France
(jeremie.legarrec , claude.andriot , xavier.merlhiot)@cea.fr

Philippe Bidaud
Laboratoire de Robotique de Paris LRP
Fontenay-aux-Roses, France
bidaud@robot.jussieu.fr

ABSTRACT

This paper describes a physically-based simulation for grasping tasks in an interactive environment. Fingertips and interacting objects are based on quasi-rigid models. The quasi-rigid model combines a rigid model for dynamic simulation and a deformable model for resolving local contact with friction and surface deformation. We simulate deformation by adding compliance on control points in the contact area based on point primitives. We use a Coulomb's law description to add friction phenomenon in the virtual environment. A linearized formulation of this model in the contact space and an iterative Gauss-Seidel like algorithm are able to solve complex multi-contacts problem between deformable objects in real time. Our method computes consistent and realistic contact surface in a stable way. This allows us to couple this system with an optical motion capture system for Virtual Reality applications.

Keywords
grasping, real-time simulation, deformable model, motion capture

1 INTRODUCTION

Deformable objects play an important role in many interactive virtual environments and this role has become crucial in grasping tasks. Indeed deformation of human fingertips is a key factor in stable manipulation, as they conform to the object's shape. Compliant contact with friction on a larger area than one point are able to resist more efficiently to perturbations in the contact space and prevent an object from slipping. Many approaches used to simulate real time deformations in interactive grasping task are based on mass-spring model or on finite element analysis. In this paper we introduce a compliant contact model adapted to a point-based representation to model the surface of the objects for this kind of tasks. Point primitives has been investigated in computer graphics in recent years. Using this representation seems to be the most adapted geometric modelling for fingertips to get objects exactly conform in the contact area. Indeed triangle meshes have irregular connectivity and objects have commonly incompatible connectivity graphs on the two sides of the contact area, which requires refinement with high computational cost.

Soft grasped objects and fingertips deform due to contact forces. So the collision detection algorithm as well as the collision response model play a major role in this kind of real time simulation. In this paper we present a robust collision detection for point-based simulation and a deformation model based on compliant contacts with friction. We use a constant time step integration called time-stepping method. The motion dynamic formulation solves for the contacts appearing between steps. This simulation can be coupled with an optical motion capture system for Virtual Reality applications.

Short Communications proceedings ISBN 80-86943-05-4
WSCG'2006, January 30-February 3, 2006,
Plzen, Czech Republic.

2 RELATED WORK

Various theoretical and experimental results have been presented by robotics communities for modelling soft finger contact or controlling robotic hands [1], [2]. In the context of real-time interaction, [3] uses a Finite Element approach and examines a method to compute a grasp quality metric which represents the ability of a grasp to maintain relative object position in the face of disturbances. To resolve contact forces for haptic feed-

foreach *time step t* **do**
Free Motion()
Collision Detection()
LCP Construction()
Gauss Seidel()
Constrained Motion()

Figure 1: Overview of our framework

back, [4] and [5] use Finite Element Methods. [6] propose a method based on a mass-spring model to compute repulsive forces for each finger of a user wearing a Cybergrasp force-feedback glove.
The surface deformation can be simulated with a fully deformable model, but it can be prohibitive since the size of the system to solve for contact resolution is large, even for small deformations. This method is not well adapted for modelling fingertips, because the surface around the deforming contact area stay intact. Instead, we use quasi-rigid object introduced by [9]. This model combines a rigid model for dynamic simulation and a deformable model for local contact resolution with friction and surface deformations. Resolving the contact problem exactly is a difficult problem because we do not know if a possible contact point on the surface, will stay in contact or not, depending on the boundary conditions and on the other points in the neighbourhood. We use a Linear Constraint Problem form (LCP) to solve the contact space [9], [4] and [10]. Point-primitives, which has become popular in recent years [7], [9], are adapted to conforming contact, as both interacting objects share contact points that have zero separation. In this context, we must also adapt the collision detection algorithm. [8] presents an approach with bounding volume hierarchy, but they do not adress the problem of contact response. [11] presents an efficient collision detection and a response based on penality force computation.

3 OUR METHOD

3.1 Overview of Our Approach

Figure 1 gives an overview of our collision detection and response framework. We use a constant time step integration called time-stepping method. The *Free Motion* will integrate the forces applied on fingertips by the user through the motion capture device, or forces from control laws for manipulation tasks. Other forces like gravity, but not contact forces, are also integrated. Then the *collision detection* algorithm gives contact normals and geometric configurations of the contact space (section 3.3). We use a *LCP formulation* and an algorithm close to the *Gauss Seidel* method [4] to solve the contact forces with friction such that the displacement of control points will deform the surfaces not to interpenetrate (sections 3.4 and 3.5). This is called the Signorini problem. The Gauss Seidel algorithm is easy to implement and robust to converge toward a solution even if several solutions are possible. Finally contact forces are integrated to find the *Constrained Motion*.

3.2 Surface Definition

Fingertips and grasped objects are modelled by an unstructured set of points. Given a point cloud with N primitives p, the surface is defined as the zero set S of an implicit function f using the Weighted Least Square (WLS) method which we breafly recap [7], [8]

$$\begin{aligned} S &= \{p \,|\, f(p) = 0\} \\ f(p) &= n(p) \cdot (a(p) - p) \end{aligned} \tag{1}$$

where $n(p)$ is the normal vector and $a(p)$ is the weighted average of all points at p :

$$a(p) = \frac{\sum_{k=0}^{N} \theta(\|p_k - p\|) p_k}{\sum_{k=0}^{N} \theta(\|p_k - p\|)} \tag{2}$$

We choose a truncated Gaussian as weighted function $\theta(x) = e^{-x^2/h^2}$ and h is the bandwidth of the kernel which tune the decay of the influence of points. The normal $n(p)$ is defined by moving last square and is exactly the smallest eigenvector of matrix B with :

$$b_{ij} = \sum_{k=0}^{N} \theta(\|p_k - p\|)(p_{k_i} - a(p)_i)(p_{k_j} - a(p)_j) \tag{3}$$

3.3 Collision Detection

Given two point clouds A and B, we need an approximation of the interpenetration distance for collision response. We use the same approach as in [11]. Given a point p_A on the surface S_A of the object A, we project this point onto S_B yielding the projected point $p^{proj}_{A,B}$. The projection operator is described in [7]. We then define p_A as intersecting with S_B if :

$$(p_A - p^{proj}_{A,B}) \cdot n_{p^{proj}_{A,B}} < 0 \tag{4}$$

To reduce the number of primitives to be tested, we use a bounding volume (BV) hierarchy for collision detection, which is a very efficient data structure for time critical application [8], [12]. We choose to implement a binary tree with a sphere as BV. Each node stores a sample of points underneath, which yields different levels of detail of the surface, and the leaf nodes must achieve a hole-free coverage of the surface, to avoid missed collisions. The traversal of the hierarchies is inspired by [8]. After this step, we obtain a set of coupled nodes which overlap. To accelerate

calculation versus accuracy, we project the center of each leaf node, not all the primitives contained in this node, on the surface in the node coupled. If the number of primitives in the coupling node is to small for correctly computing $a(p)$ and $n(p)$, we take into account points in a given border of this node (called r_ϵ-border in [8])(figure 2).

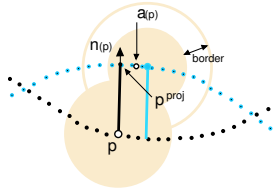

Figure 2: Estimation of the coupling interpenetration distances.

The bandwidth of the projection operator is chosen as the distance between the center of the two nodes. To avoid artefacts that can appear in high curvature region, we don't take into account cases when the projection point is not situated inside the border of the coupling node.
At each time step, the overlapping surfaces are computed after the Free Motion step, so in an undeformed state. However, because we deal with deformable models, collisions could be potentially missed even for small deformations. So we must update the BV hierarchy, whereas this operation is often very costly. We notice that the updated hierarchy is used to obtain a set of overlapping nodes, not to compute the interpenetration distance for collision response. This update is done by Bottom-Up strategy, but to accelerate this operation, we stop the process when we reach a given level in the hierarchy.

3.4 Linear Complementarity Formulation

Given the set of active points $\{(p,q)\,|p \in S_A, q \in S_B\}$ on the surface of two objects, we must find the contact forces acting on this points such that the displacement will deform the surfaces not to interpenetrate (figure 3).

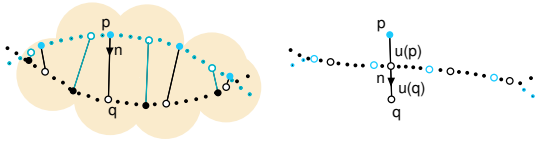

Figure 3: On the left, active points (p,q) obtained with collision detection algorithm, on the right, displacement (u_p,u_q) to obtain zero gap.

This problem is solved using the method introduced by [4]. We choose an arbitrary direction for the contact normal n and in this direction the force f_p^n is positive : $f_p = -f_q = f^n n$ and we define the gap : $\delta_p^n = (p-q)\cdot n$. Points in the contact space are chosen in the set of points determined by the collision detection with $\delta_p^n < \delta_\epsilon$ and $\delta_\epsilon > 0$. A positive threshold δ_ϵ permits to introduce in the contact space the neighborhood area of contact, which could penetrate during the step of contact resolution. To solve the contact forces, we use a linear complementarity problem LCP formulation [4], [9]. For each potential contact point, we can write :

$$0 \leq f^n \perp \delta^n \geq 0 \tag{5}$$

To linearize Signorini's problem, the contact space is frozen during the current time step, called the free motion. With the collision detection detailed above, we can compute the value of each gap, denoted δ^{nfree}, and the contact normal without contact forces. Deformation displacements u_p and u_q are defined between the free motion and the constrained motion, after solving Signorini's problem and the integration of the contact forces :

$$\delta_p^n = (u_p - u_q)\cdot n + \delta_p^{nfree} \tag{6}$$

Displacements are then first calculated at a coarser level (set of points (p,q)) and propagated to the surface using interpolation, allowing us to accelerate calculation versus accuracy. For visual accuracy, the normal vector of each contact point is recomputed using the method introduced in section 3.1.

3.5 Contact Resolution

3.5.1 Quasi-Rigid Model

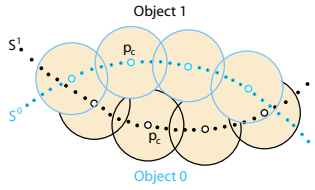

Figure 4: Control points p_c on the interpenetrating surfaces S^0 and S^1. They are chosen as the centers of the leaf nodes in the BV hierarchy.

We use the approach of quasi-rigid model introduced by [9], which splits the global motion, driven by a rigid model, from a local relative displacement driven by a linear deformable model. The deformation of the surface points can be linearized to $\widetilde{K}u = f$ where $\widetilde{K}$ is analogous to a stiffness and depends on the intrinsic properties of the object. To obtain deformation compliance $C = (\widetilde{K})^{-1}$, we can use a model based on continuum mechanics for mesh free [13], or a local model [9] [14]. We choose the second type

to speed up the calculation of compliance, but this contact resolution method is still valid for more complex deformation models. The deformation is simulated by adding compliance on N control points p_c on the surface. They are chosen as the centers of the leaf nodes in the BV hierarchy (figure 4). A linear relation links the constrained control node displacements $u_{p_c} = [u_{p_{c_0}} \cdots u_{p_{c_N}}]^T$ and the contact forces $f_{p_c} = [f_{p_{c_0}} \cdots f_{p_{c_N}}]^T$:

$$u_{p_c} = [C_{p_c}]\, f_{p_c} + u_{p_c}^{free} \tag{7}$$

The displacement of a contact point p is interpolated from the controlled points by :

$$u_p = \frac{\sum_{i=0}^{N} \theta(\|p - p_{c_i}\|) u_{p_{c_i}}}{\sum_{i=0}^{N} \theta(\|p - p_{c_i}\|)} \tag{8}$$

where $\theta(x)$ is a gaussian weighted function. We can equivalently interpolate the unknown contact force at p_{c_i}, with the force f_p^n obtained in the contact space. By stacking relations (6), (7) and (8) for each contact on the surface S_i for a couple of objects, we can write matrices $\left[H^{S_i}\right]$ such that :

$$\delta_p^n = (\underbrace{\sum_i \left[H^{S_i}\right] \left[C^{S_i}\right] \left[H^{S_i}\right]^T}_{W}) f_p^n + \delta_p^{nfree} \tag{9}$$

W is named Delassus operator [15]. See [4] for more information. To simulate friction, Coulomb's law describes the behaviour in tangent contact space :

$$\begin{aligned} \delta_p^t = 0 \Rightarrow \|f^t\| < \mu \|f^n\| \\ \delta_p^t \neq 0 \Rightarrow \|f^t\| = -\mu \|f^n\| \frac{\delta_p^t}{\|\delta_p^t\|} \end{aligned} \tag{10}$$

Then we linearize the system along two tangential directions, f^{t_1} and f^{t_2} :

$$\begin{bmatrix} \delta^n \\ \delta^t \end{bmatrix} = \begin{bmatrix} W_{nn} & (W_{nt})_{(1\times 2)} \\ (W_{tn})_{(2\times 1)} & [W_{tt}]_{(2\times 2)} \end{bmatrix} \begin{bmatrix} f^n \\ f^t \end{bmatrix} + \begin{bmatrix} \delta^{nfree} \\ \delta^{tfree} \end{bmatrix} \tag{11}$$

We use a Gauss-Seidel like algorithm to solve Signorini's and Coulomb's laws with a guaranteed convergence [4], [16].

As the quasi-rigid approach [9] splits the global motion from a local linear relative displacement, we can sum up these two models in compliance within the contact space. The equation (9) can be rewritten as

$$\delta_p = [W_{rigid} + W_{deform}]\, f_p + \delta_p^{free} \tag{12}$$

When the stiffness increases, the behavior tends to the rigid motion.

3.5.2 Deformable Model

A displacement $u_{p_{c_i}}$ of a control point on the surface at p_{c_i} induced by a force $f_{p_{c_j}}$ acting at p_{c_j} is given by

$$u_{p_{c_i}} = \phi_j(\left\|p_{c_i} - p_{c_j}\right\|) f_{p_{c_j}} \tag{13}$$

where ϕ a function of influence, which can be physically motivated as the Boussinesq approximation [9]. In our simulation, we choose to introduce a local coupling between points ensuring that the Gauss-Seidel algorithm yields coherent converging solutions in time,

$$\phi_j(x) = gauss(x) C_{p_{c_j}} \tag{14}$$

where $C_{p_{c_j}}$ is the compliance at point p_{c_j} and $gauss(x)$ a gaussian. Since we assume linear elasticity, the total displacement at p_{c_i} is the superposition of contribution of all neighboring points. Using an implicit euler scheme, the compliance include stiffness (K) and damping effects (D) : $C_{p_{c_i}} = (K + \frac{D}{\Delta T})^{-1}$. Then we can compute W_{deform} using the equation (9).

3.5.3 Rigid Model

We use the quasi-static generalized form for the rigid model [17] :

$$b(q, \dot{q})\dot{q} = \Gamma^{ext} + \Gamma^{contact} \tag{15}$$

where q is the vector of generalized degrees of freedom, b the viscosity, Γ^{ext} the resultant of external forces and $\Gamma^{contact}$ the contact forces. [17] introduced a methodology to map the rigid motion space into the contact space, using a jacobian J_c. We can now express the motion of the gaps in the contact space due to a rigid motion :

$$\dot{\delta}_p = \left[J_c B^{-1} J_c^T\right] f_p + \dot{\delta}_p^{free} \tag{16}$$

We can obtain the rigid form of the Delassus operator integrated with an implicit euler method :

$$W_{rigid} = \left[J_c \left(\frac{B}{\Delta t}\right)^{-1} J_c^T\right] \tag{17}$$

This model allows for a stable real time simulation with an arbitrary choice of time step.

4 DISCUSSION

We implement this method in C++. As of yet, the implementation is not fully optimized. In the following, all results have been obtained on a 2 Ghz Pentium-IV with 1 Go RAM.

An example for grasping task is shown in Figure 5. A box is maintained due to friction between three fingertips while the gravity force is pulling it down. The

virtual hand is composed of 3 fingers with 4 degrees of freedom for each finger. Each finger is controlled in position with generalized motor torques. The box is composed by 36K sampled points and 4800 control points and each fingertip is composed by 2400 sampled points and 320 control points. Figure 6 and 7 give performance mesurements for the detection collision algorithm and the contact resolution method with friction.

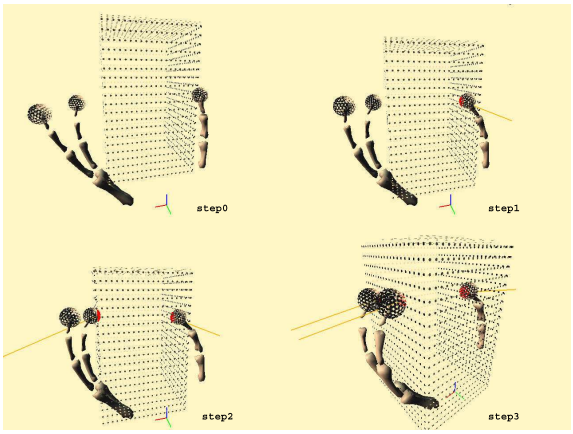

Figure 5: Grasping simulation of a box with gravity. The three fingers are controlled in position. Contact forces are drawn for each finger.

	step1	step2	step3
Traverse BV Hierarchy (ms)	1.170	1.904	2.973
Nb Contacts	60	108	180
Interpenetration Computing (ms)	1.394	2.376	4.100
Hierarchy Update (ms)	2.468	2.717	2.549
Total Time (ms)	5.279	7.382	10.200

Figure 6: Collision Detection computing time for each step of the simulation.

	step1	step2	step3
Nb Contacts	56	105	164
Fill W_{Rigid} (ms)	8.838	16.560	25.030
Fill W_{Deform} (ms)	1.159	3.140	7.350
Gauss Seidel Iterations	4	4	5
Gauss Seidel Resolution (ms)	0.409	2.061	5.410
Total Time(ms)	14.150	28.374	43.060

Figure 7: Contact Resolution computing time for each step of the simulation.

There are many factors influencing the performance. For large number of colliding points, the estimation of the different coupling distance has the more computational cost. Significant gain could be achieved exploiting the temporal coherence during collision detection. Using this particularity, different temporal resolution could be used for the different part of this method, which can be easily parallelized.
Like many discrete collision detection algorithm, our approach is susceptible to fail for deep penetrations, where bounding volumes of the hierarchy don't overlap. Furthermore the computation of the contact forces depends on the accurate estimation of the contact normal direction, especially for surface with high curvature like edges. Some artefacts can appear because our intersection resolution algorithm in the leaf nodes and the local deformation model are not well adapted in that case.
The Gauss Seidel algorithm is well suited for interactive application and robust to converge. Indeed convergence is reached for a hundred contacts in less than 2ms. To another part, the building of the Delassus operator dominates the computational overhead. We expect substantial speedups by reusing parts of the computation over multiple time steps.
There are many avenues for further work. First, we can use our contact resolution as a framework to test more advanced friction models. In fact, the geometry of the contacting surface and the friction model have an important influence on how the grasping simulation evoles. That's why more complex control laws are useful for computing non-sliding contact forces to move the object on a given consign position.
We expect that this system can be coupled with an optical motion capture system (ART [18] Advance Realtime Tracking). The fingertips are linked to trackers with damped springs, following the orientation of the hand and the position of the three fingers.

5 CONCLUSION

In this paper, we present a method to use point-primitives for modelling fingertips in interactive grasping tasks. We also present an exact contact resolution with friction between quasi-rigid objects. Thanks to Gauss Seidel algorithm and the use of Delassus operator, this approach is suitable for real time application. More advanced physical models could be integrated in our system such as advanced friction models. This model can be efficiently combined with an optical motion capture system for real time grasping task. The improvement of this kind of application is still in progress.

6 ACKNOWLEDGMENTS

We would thanks Christian Duriez for his help with the Delassus operator and the Gauss Seidel algorithm formulation.

References

[1] N. Xydas and I. Kao, Modeling of contact mechanics with experimental result for soft fingers. In *IEEE Intl. Conference on Intelligent Robots and Systems*, pp. 488-493, 1998.

[2] C. H. Xiong, M. Y. Wang, Y. Tang, and Y. L. Xiong, Compliant grasping with passive forces. In *Journal of Robotic Systems*, 22(5), pp. 271-285, May 2005.

[3] M. Ciocarlie, A. Miller and P. Allen, Grasp analysis using deformable fingers. In *IEEE Intl. Conference on Intelligent Robots and Systems*, Aug. 2005.

[4] C. Duriez and C. Andriot, A multi-threaded approach for deformable/rigid contacts with haptic feedback. *Intl. Symposium on Haptic Interfaces for Virtual Environments ans Teleoperator Systems*, 2004.

[5] G. Picinbono, J. Lombardo, H. Delingette and N. Ayache, Improving realism of a surgery simulator : Linear anisotropic elasticity, complex interactions and force extrapolation. In *J. Visualization and Computer Animation*, 2002.

[6] A. Maciel, S. Sarni, O. Buchwalder, R. Boulic and D. Thalmann, Multi-finger haptic rendering of deformable objects. In *Proc. of Eurographics Symposium on Virtual Environments*, 2005.

[7] A. Adamson and M. Alexa, Approximating and intersecting surfaces from points. In *Proc. of Eurographics Symposium on Geometry Processing SPG 03*, pp. 230-239, June 2003.

[8] J. Klein and G. Zachmann, Point Cloud Collision Detection. In *Computer Graphics Forum (Proc. of Eurographics 2004)*, 23, 3, Grenoble, France, pp. 567-576, Sep. 2004.

[9] M. Pauly, D. Pai and L. Guibas, Quasi-Rigid Objects in Contact. In *ACM Siggraph/Eurographics Symposium on Computer Animation* 2004.

[10] D. Baraff, Fast contact force computation for non penetrating rigig bodies. In *Proc. of Siggraph*, pp 23-34, 1994.

[11] R. Keiser, M. Müller, B. Heidelberger, M. Teschner and M. Gross, Contact Handling for Deformable Point-Based Objects. *Proc. of Vision, Modeling, Visualization VMV'04*, Stanford, CA, Nov. 2004.

[12] G. Van Den Bergen, Efficient collision detection of complex deformable models using AABB trees. *Journal of Graphics Tools 2*, pp. 1-14, 1997.

[13] M. Müller, R. Keiser, A. Nealen, M. Pauly, M. Gross and M. Alexa, Point-Based Animation of Elastic, Plastic, and Melting Objects. *ACM Siggraph/Eurographics Symposium on Computer Animation* 2004.

[14] P. Song, J.S. Pang and V. Kumar, A semi-implicit time-stepping model for frictionnal compliant contact problems. *Intl. Journal of Robotics Research*, 60, pp. 2231-2261, 2004.

[15] J-J. Moreau and M. Jean, Numerical Treatment of contact and friction : the contact dynamics method. *Engineering Systems Design and Analysis*, 4:201-208, 1996.

[16] F. Jourdan, P. Alart and M. Jean, A gauss-seidel like algorithm to solve frictional contact problems. *Computer Methods in Applied Mechanics and Engineering*, pp. 33-47, 1998.

[17] D. Ruspini and O. Khatib, Collision/Contact Models for the Dynamic Simulation of Complex Environments *IEEE/RSJ IROS'97*, Sept. 1997.

[18] www.ar-tracking.de