

Haptic manipulation of deformable objects in hybrid bilateral teleoperation system

doi:10.1080/11762320701848746

Juan Manuel Ibarra-Zannatha,^a Claudia Marmolejo-Rivas,^b
Manuel Ferre-Pérez,^c Rafael Aracil-Santonja^c
and Salvador Cobos-Guzmán^c

^a Centro de Investigación y de Estudios Avanzados del IPN, Laboratorio de Robótica y Visión Artificial del Departamento de Control Automático, Av. IPN N° 2508, San Pedro Zacatenco 07300 México, D.F. México

^b Universidad Autónoma de Sinaloa, Escuela de Informática de Mazatlán. Av. Universidad y Av. Leonismo Internacional s/n, 82017 Mazatlán, Sin., Mexico

^c Universidad Politécnica de Madrid, Escuela Técnica Superior de Ingenieros Industriales, Departamento de Automática, Ingeniería Electrónica e Informática Industrial c/ José Gutiérrez Abascal, 2; E-28006 Madrid, Spain

Abstract: The aim of this work is the integration of a virtual environment containing a deformable object, manipulated by an open kinematical chain virtual slave robot, to a bilateral teleoperation scheme based on a real haptic device. The virtual environment of this hybrid bilateral teleoperation system combines collision detection algorithms, dynamical, kinematical and geometrical models with a position–position and/or force–position bilateral control algorithm, to produce on the operator side the reflected forces corresponding to the virtual mechanical interactions, through a haptic device. Contact teleoperation task over the virtual environment with a flexible object is implemented and analysed.

Key words: Deformable objects, haptics, bilateral teleoperation, virtual environment, force reflection.

INTRODUCTION

Teleoperation extends user presence into remote or hazardous environments, for it allows users to control machines at large distances through a person–machine interface. Though three components are essential: a master robot operated by the user, a slave robot to perform contact tasks at a remote location and a computerised electronic interface to control and communicate mechanical parts. Force feedback provides useful information to the operator of such a system, improving safety for the user, slave robot and materials being handled. When force is directly provided to users by means of the master, teleoperation systems are bilaterally controlled (Hannaford 1989).

In a typical bilateral teleoperation system, the master robot is used to enhance the force and the reach of the user, the slave robot is therefore a big and powerful machine when compared with the former (Khatib *et al.* 2004, Turro *et al.* 2001, Kheddar *et al.* 2007). Substitution of the remote slave robot along with its environment for an interactive virtual world results in a more efficient way to

carry out experiments during teleoperation system development phase; the safety of both the user and equipment is a non-negligible additional advantage (Duriez *et al.* 2002, Marmolejo *et al.* 2005). The components of the hybrid teleoperation (HBT) system ought to be a haptic tool in place of the master robot, ranging from a joystick to a 6 DoF device (Otaduy and Lin 2006), whose duty is to reproduce the kinesthetic component of touch when rendering a force feedback, a virtual slave and a virtual environment to be manipulated and a person–machine interface. Virtual world inputs: master robot position changes or forces applied by the user, generate both: classical screen visual render and master robot force feedback to recreate the sense of touching a real environment through a bilateral teleoperation system and improving user's immersion feeling (Khatib *et al.* 2004, Marmolejo *et al.* 2005).

Hybrid teleoperation systems interactive models have important real-time restrictions, such as bandwidth and latency (Delingette and Ayache 2004). In a bilateral teleoperation system there are two main sources of stability problems: the transmission delay between the master and the slave and contact with the environment, especially when it is a deformable one. Stable haptic interactions with deformable objects is a challenging task (Barbagli *et al.* 2005), for they reach very high levels of computa-

Corresponding author:
Claudia Marmolejo-Rivas
Email: clamarmol@aol.com

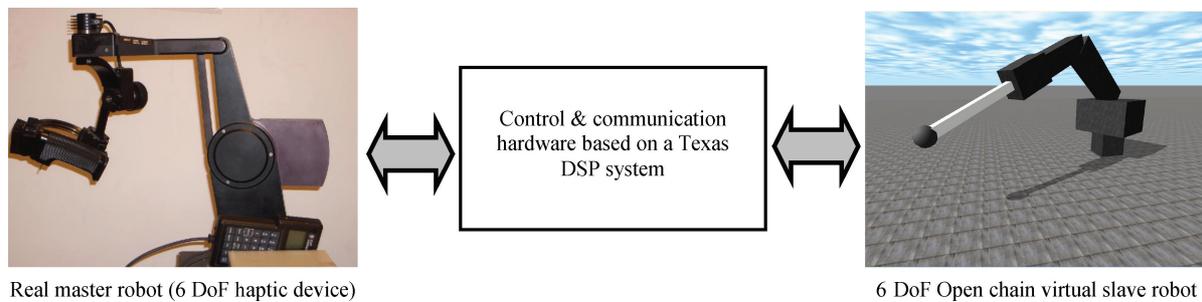


Figure 1 Hybrid bilateral teleoperation system.

tional complexity, producing low servo rates and supplementary computational delays that may produce some unstable behaviours. Solutions that reduce computation loop time include – to simplify deformable object algorithm (Cavusoglu and Tendick 2000, Cotin *et al.* 2000); to build an approximation of deformable object geometry that results in a haptic interaction computed at high servo rates (Mahvash and Hayward 2004, 2006), and to ensure the passivity of the whole hybrid system to obtain stable haptic interactions with a wide variety of unknown virtual environments (Colgate *et al.* 1995, Adams and Hannaford 1999). The latter two options are independent of the simulation method and may be used to represent any kind of deformable object model. Interactive virtual worlds, despite its inherent non-linearity, non-passivity and time delay, can be coupled with haptic devices in an HBT scheme in a stable way introducing additional damping in the system (Miller *et al.* 2000); therefore, when developing virtual environments one does not have to worry about stability. The main drawback of these algorithms is, however, that they rely on high servo rates, which often cannot be accomplished when interacting with deformable objects, and on damping, which compromises the overall transparency (Barbagli *et al.* 2005).

Today HBT systems may be found in computer animation, interactive games, surgical simulation, training systems, computer-aided design, interactive path planning, and in manufacture, HBT assembly and maintenance prototypes (Barbagli *et al.* 2005, Cotin *et al.* 2000, Marmolejo *et al.* 2005). In this inherently interdisciplinary field an application ought to combine computer graphics, Newtonian dynamics, continuum mechanics, numerical computation, differential geometry, vector calculus, approximation and control theories, among others, into a vast and powerful toolkit, which, in our case, is being further explored and extended.

Components of our HBT system, shown in Figure 2, are geometrical models of both the slave robot and its environment, containing a flexible or elastic object along with rigid walls, rendered to the screen; dynamical models to govern deformations of the flexible virtual object when manipulated by the virtual slave, and to track the virtual slave movement; collision detection module to notice contact between the virtual slave robot tip and scene objects, collision response in which deformation and pen-

etration depth are calculated in order to trigger object’s dynamical model and to obtain reaction forces and bilateral control ensuring stability of the 6 DoF master robot (Marmolejo *et al.* 2005).

The next section of this article is devoted to geometrical and dynamical models of 3-D deformable objects, including a comparison between linear models of continuum mechanics and inherently discrete mass-spring models, while the modelling of rigid bodies is the object of the third section. The fourth section presents the interaction between virtual objects including collision detection and force calculation algorithms. The fifth section shows the obtained experimental results and, finally, some conclusions are presented.

MODELLING DEFORMABLE OBJECTS

Geometrical properties and dynamical behaviour of deformable objects present in the environment ought to be represented in an HBT system by models, especially upon interaction between these models and the virtual manipulator. The field of physically based deformable models in computer graphics has expanded tremendously and is very active and fruitful with many visual stunning achievements to account for.

Significant contributions were made in many key areas, e.g. object modelling, fracture, plasticity, cloth animation, stable fluid simulation, time integration strategies, discretisation and numerical solution of partial differential equations, modal analysis, space-time adaptivity, multi-resolution modelling and real-time simulation (Cotin *et al.* 2000, Delingette and Ayache 2004).

The first model used to animate deformable objects, based on Hooke’s law, integrates energy-based Lagrange

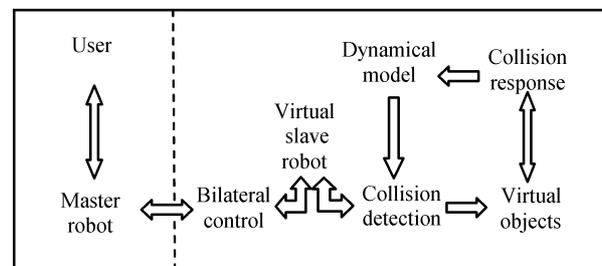


Figure 2 HBT system components.

equations with finite differences or finite elements and was proposed by Terzopoulos *et al.* (1987). After this seminal paper, substantial work has appeared on physically based deformable object modelling, simulation and animation in computer graphics and related scientific fields (Cavusoglu *et al.* 2001, 2002, Corso *et al.* 2003, Cotin *et al.* 2000, Debunne *et al.* 1999, 2001, Delingette and Ayache 2004, Duriez *et al.* 2002, Kuroda *et al.* 2002, and Mahvash and Hayward 2006); although not all are suited for interactive applications.

There are some different criteria in classifying existing modelling methods: regarding their nature, they are divided into physical models, based on classical dynamics, and geometrical models, such as parametric curves and surfaces, and free-form deformations. Physically based methods may be classified this time into Lagrangian, wherein models consist of a set of points with varying locations and properties, and Eulerian methods, wherein model properties are computed by a set of stationary points. When using Lagrangian mesh-based methods in order to model object deformation under the action of an external force, the system has to compute for each step and each object vertex, velocity v_i and position x_i . This may be done through a mass-spring model or with a continuum object model. Mass-spring model considers discrete mass elements concentrated in the nodes of a grid. Each node is connected to its neighbours by massless springs, thereby producing a discrete object representation. On the other hand, a numerical method should be applied to solve a continuum object model in order to obtain a representation of a force function with values found in predefined sample points.

Early animations were computationally intensive; Lagrangian models based on Hook's law were then applied (Terzopoulos *et al.* 1987). Other approaches showing fast representation of simple dynamic flexible objects but with restrictions that affected realism of animation have later on appeared. Multi-resolution models, together with an adaptive sampling, minimised calculations on this kind of physically based models (Debunne *et al.* 1999). Discrete linear elastostatic models (LEM) are important physically based elastic primitives that are used for haptic rendering because they admit a high degree of numeric compression. Intuitively, Green's functions became a basis for describing all possible deformations of LEM, which is a simple but stable linear 3-D mass-spring model. Green's functions have been suitable for force feedback rendering of stiff elastic objects during continuous contact. The fact that the model is linear is a crucial enabling factor. The implementation of LEM benefits from a significant pre-computation. Linear elastostatic model includes many global aspects of deformation that is appropriate in dealing with boundary discretisation. However, it is inaccurate for some materials and large deformations. The work done by researchers at l'Institut National de Recherche en Informatique et Automatique (INRIA) has been relevant. They use an elastostatic finite element method (FEM) model for liver surgical interactive application (Delingette and Ayache 2004), which

allows real-time interaction with deformable objects for force feedback and computer animation (Cotin *et al.* 2000).

To solve the same problem, at Johns Hopkins University the soft environment is modelled considering a fixed impedance, and then the deformation is a force function (Cavusoglu and Tendick 2000, Cavusoglu *et al.* 2002). The flexible objects can be defined also by a discretised medial axis transform, which consists of an ordered set of spheres whose centres are connected by a skeleton. Their algorithm automatically generates a high dimensional parametric surface that encapsulates the object shape, stiffness, damping, and mass, then, to compute the force upon deformation, it is used as a mass-spring-damper model that takes into account both normal and shear surface forces (Corso *et al.* 2003).

Visual rendering could be implemented by surface or volume techniques. Surface technique is the most commonly used as it takes basic polygonal elements (triangles, quads) to obtain the entire scene. Volume rendering based on polyhedral elements (cubes, tetrahedral) is more realistic and well suited to model objects that can be fractured (Cotin *et al.* 2000). Quality is proportional to the number of polygonal elements. However, the updating rate on the graph display is inversely proportional to this number.

Geometrical model: boundary representation

The first task in virtual object implementation is defining its geometry, as a teleoperation task nature function and robot slave type. Here the end effector of the robot slave is a little sphere used to touch an elastic sphere that is five times bigger, placed on the floor. In this case, the object in the virtual environment may be represented through its boundary. The penetration depth of the end effector tip into the non-deformed virtual object is required to obtain the corresponding reaction force to be rendered to the master robot.

Polygonal meshes are widely used to represent virtual objects in computer graphics. Among these meshes, triangles are preferred as the graphical primitive in the description of a virtual surface and, with appropriate constraints, are also used to build solid volumes (Lin and Gottshalk 1998). The triangular mesh representing the elastic sphere to be touched in this exercise was created over an initial coarse polygonal mesh and a recursive refinement method that allows progressive approximation error decrease of the actual grid. Platonic solids (octahedral, icosahedra) were used in the initial step. Each refinement level increased the number of triangles by a factor of 4 (Leech 1989).

The initial coarse mesh used was an origin-centred icosahedra. In each approximation step, every triangle in the actual mesh was subdivided by adding a new point in the middle of every edge. As the sphere had a unitary radius, every new point was normalised in order to move it to the sphere's surface. After five iterations, it contained a redundant set of 15,360 points, for almost every vertex belonged to six neighbouring triangles. An

additional list of non-redundant vertices with 2560 points was obtained. Comparing the triangles list with the point list, the non-redundant points were grouped by faces, getting a column called Vertices in Figure 3 example, of 5120 sets of three points, each one belonging to a triangle, in accordance with the number of triangles modelling the sphere derived from an icosahedra. In the triangular mesh shown in Figure 3, it is easy to find all the triangles to which vertex 7 belongs, by searching in each set of the column Vertices, vertex 7 and relating it to a triangles list getting that vertex 7 belongs to triangle B, C, D, H, I and J. Finally, by scanning the faces list, a new table of neighbouring vertices is created. In the graphical example shown in Figure 3, by following the previous method, vertex 1 has vertices 2 and 6 as its neighbours. One can get the redundant list of first neighbours for vertex 11 from the example by locating the set of faces containing vertex 11 in the column Vertices, which are I which includes vertices 7, 11 and 10; J with 7, 8 and 11; K has vertices 8, 12 and 11, M formed by 10, 11 and 13; N with 11, 14 and 7; O includes vertices 11, 12 and 14. Extracting from these points, vertex 11 the resulting neighbours are 7, 10, 7, 8, 8, 12, 12, 14, 14, 13, 10 and 13, after eliminating redundancies we now know, without reading it from the triangular mesh, that vertex 11 has as its first neighbours vertices 7, 10, 8, 12, 14 and 13.

△	Vertices	Point	Faces
A	1, 2, 6	1	A
B	2, 7, 6	2	A, B, C
C	2, 3, 7	3	C, D, E
D	3, 8, 7	4	E, F, G
E	3, 4, 8	5	G
F	4, 9, 8	6	A, B, H
G	4, 5, 9	7	B, C, D, H, I, J
H	6, 7, 10	8	D, E, F, L, K, J
I	7, 11, 10	9	F, G, L
J	7, 8, 11	10	H, I, M
K	8, 12, 11	11	I, J, K, O, N, M
L	8, 9, 12	12	K, L, O
M	10 11, 13	13	M, N, P
N	11, 14, 13	14	N, O, P
O	11, 12, 14	15	P
P	13, 14, 15		

The neighbours list informs which vertices are influenced by a change in position from any of them. If vertex 11 is taken out from its equilibrium position by the user, the dynamical model notices that it and its neighbours suffer an internal force and responds to it. This sphere may be further scaled without deformation, due to its symmetrical nature.

Teleoperation visual and haptic render of a deformable object needs, besides calculation of its geometrical evolution after deformation due to user interaction, computa-

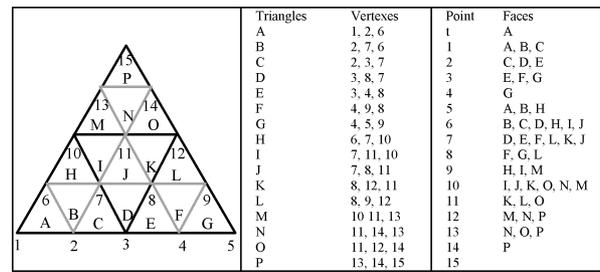


Figure 3 Data set creation example allowing simulation of a deformable object.

tion of resulting forces by means of a physical based model. Triangular meshes may be used to build complex virtual objects as well, but the attachment of a dynamical model requires a data set containing triangles, vertices, faces and neighbours.

Dynamical model

In the animation of deformable objects under the action of an external force, discrete systems like particle systems such as mass-spring systems are widely used in the animation of flexible virtual objects. Easier to implement and faster than finite element methods, these systems allow the animation of dynamic behaviours. The main advantages of mass-spring models are its easy implementation, fast calculation and inherently discrete nature; while its disadvantages are no triangular area preservation, useful only in small deformations and valid for small time steps. One of the main drawbacks of mass-spring systems is that neither isotropic nor anisotropic materials can be generated and controlled easily. Another problem is that most of the materials found in nature maintain a constant or quasi-constant volume during deformations. Mass-spring models do not have this property (van Gelder 1998, Bourguignon and Canni 2000), then, to overcome the volume conservation problem, a constraint, transformed into a force, was added by the inclusion of radial springs to our sphere, as described by Marmolejo *et al.* (2005).

In our application we considered a sphere whose geometrical model represents its surface through an adaptive triangular mesh as explained before. To rule the dynamical behaviour of the considered elastic sphere, the external force exerted over it was computed by calculating the penetration depth as will be shown in the fourth section. Forces exerted by the sphere deformed part over its neighbouring material elements were computed as well using the Navier equation:

$$\rho \mathbf{a} = \mu \Delta \mathbf{d} + (\lambda + \mu) \nabla (\nabla \cdot \mathbf{d}), \tag{1}$$

wherein λ and μ are Lamé coefficients, ρ represents object density, \mathbf{a} stands for acceleration, \mathbf{d} for deformation vector and $\Delta \mathbf{d}$ for the deformation of the Laplacian. Equation (1) has two terms: the deformation of the Laplacian and

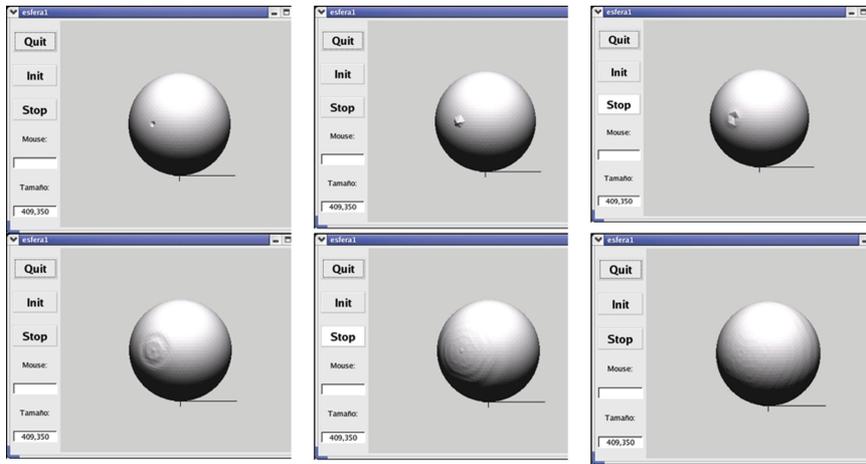


Figure 4 Evolution of the elastic sphere using Navier equation.

a term that limits the volume deformation. To solve the differential equation, among numerical methods, the finite difference method was applied to each term obtaining (Cotin *et al.* 2000):

$$\Delta \mathbf{d}_i = \frac{2}{\sum_j |\mathbf{x}_i - \mathbf{x}_j|} \sum_j \frac{\mathbf{d}_i - \mathbf{d}_j}{|\mathbf{x}_i - \mathbf{x}_j|}$$

$$\nabla(\nabla \cdot \mathbf{d})_i = \frac{2}{\sum_j |\mathbf{x}_i - \mathbf{x}_j|} \sum_j \frac{[(\mathbf{d}_i - \mathbf{d}_j) \cdot \mathbf{u}]}{|\mathbf{x}_i - \mathbf{x}_j|} \mathbf{u} \quad (2)$$

wherein \mathbf{d}_i stands for the deformation suffered by element i , and $|\mathbf{x}_i - \mathbf{x}_j|$ for the separation distance between neighbouring nodes, and $\mathbf{u} = (\mathbf{x}_i - \mathbf{x}_j)/|\mathbf{x}_i - \mathbf{x}_j|$ for the unit vector between them. Equation (2) may be used for simulating the small deformation of complex objects as long as the deformation of each vertex may be computed along with the distance separating neighbouring nodes. Advantages of the Navier equation are an available wider range of softness, preservation of the triangular area and linearity that leads to an easy implementation, whereas its disad-

vantages are the need of discretisation, useful only in small deformations and unable to rotate the object.

Experimental results using the Navier equation

The Navier equation along with a linear strain tensor was tested with the sphere, moving one of its vertices out of its equilibrium position, as if its surface was pushed inside using a single operation point, in order to observe its behaviour; in the first case without restricting the movement of any vertex and in the second one, keeping the deformed vertex out of its equilibrium position at all times.

In the first case, when all vertices are allowed to move freely after deformation, it is possible to watch in the graphical rendering (Figure 4) the deformed vertex moving around its equilibrium point. This same behaviour is observed through the phase graph presented in Figure 5, in which at first the particle moves along a spiral-like path, proper of a stable system, and then it goes back and forth as if it felt no frictional forces. In the second experiment, where the deformed vertex is forced to stay outside of its equilibrium position, the graphical rendering shows

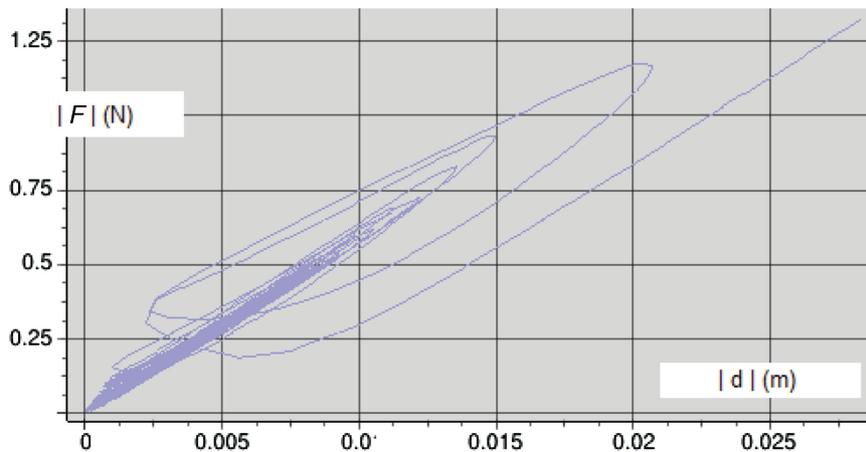


Figure 5 Phase graph of the deformed vertex moving freely.

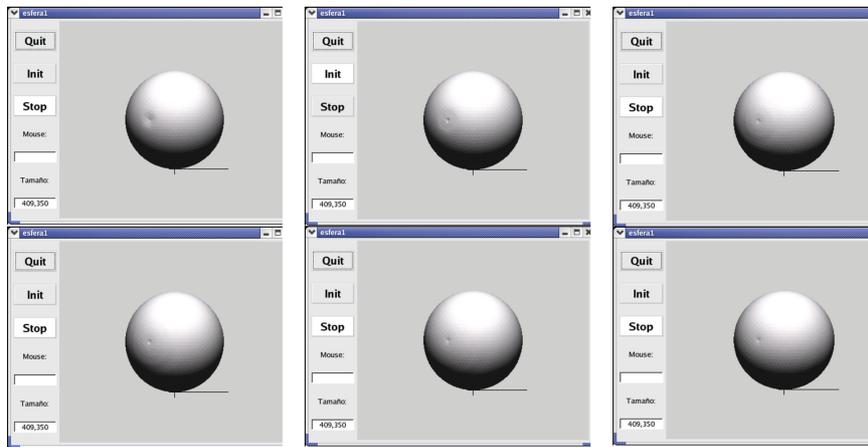


Figure 6 Evolution of the virtual sphere with a vertex out of its equilibrium position at all times.

(Figure 6) first that neighbouring particles move in order to be as near as possible of every other vertex next to it, and then it oscillates around its equilibrium position (Figure 7).

Simulation experiments showed that, when modelling non-moving objects, the Navier equation discretised model along with a linear strain tensor is to be chosen, but if the object is to suffer large deformations or to change position, the best option is to work with a linear Navier equation and a non-linear strain tensor, for under rotations of the object, the linear strain tensor does not remain invariant (Etmuss *et al.* 2003).

VIRTUAL SLAVE MODEL

The virtual slave robot is a typical open kinematic chain of 6 DoF with rigid links, whose end effector is a sphere. The dynamical behaviour of this manipulator is given by the classical equation representing the joint forces and torques τ acting in the mechanism:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) + F(q) = \tau \quad (3)$$

where $M(q)$ represents the inertia matrix, $C(q, \dot{q})$ is a matrix including Coriolis and centrifugal forces and vectors

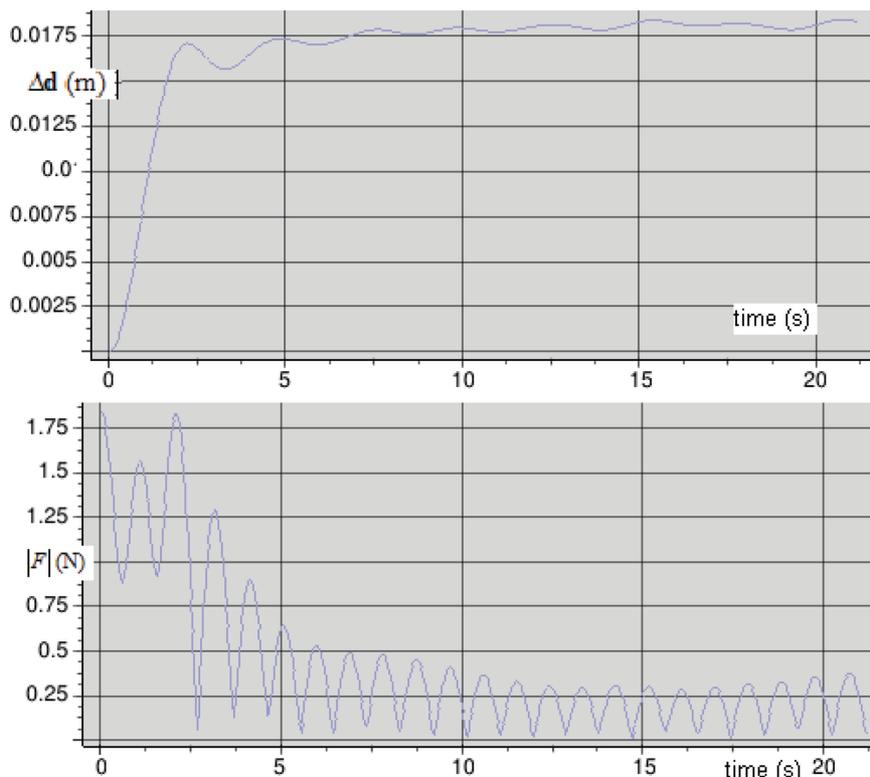


Figure 7 Distance to equilibrium and force magnitude of a neighbour of deformed vertex at all times.

$G(q)$ and $F(q)$ are gravitational and friction forces. The interaction between the slave robot and its environment generates forces that must be reflected to the user through the haptic device. The expression used to calculate these forces is

$$\tau = \mathcal{J}(q)^T \gamma, \quad (4)$$

where γ represents the resultant force at the end effector of the robot, relating the resultant force and torque felt by the end effector when a collision with the virtual environment is detected. The geometric Jacobian used in force transformation is given by the next expression (Siciliano 2000):

$$\mathcal{J}(q) = \begin{bmatrix} z_{i-1} \times (P - P_{i-1}) \\ z_{i-1} \end{bmatrix}. \quad (5)$$

Finally, the velocity of the end effector is computed with the following expression based on the geometric Jacobian:

$$v = \begin{bmatrix} \dot{p} \\ \omega \end{bmatrix} = \mathcal{J}(q) \dot{q}. \quad (6)$$

Differential equations representing dynamical behaviour of the slave robot, given by Equations (3)–(6), are solved using Open Dynamics Engine (ODE). For visualisation purposes, each link of the virtual robot slave is modelled as a simple 3-D box but it is characterised by its real dynamic parameters.

Hybrid teleoperation includes the kinematics of the slave robot and a classical algorithm for the Jacobian matrix calculation used in the force feedback loop.

INTERACTION BETWEEN VIRTUAL OBJECTS

The virtual environment in our HBT system includes deformable and rigid bodies. In a teleoperation task performance with the HBT system, the user moves the slave robot rigid tip and touches the boundary surface of an elastic polygonal structure placed on the floor, formed by a mesh of some thousands of triangular faces; the application reports a collision detection event and calls the 3-D deformable object dynamical model and the robot's slave tip penetration depth computation algorithms, which are the two things needed to be done when recreating interaction between objects in a virtual scene: detect its collision and respond to it. The former is a kinematical problem involving relative positions among objects, whereas the latter is a dynamical one requiring prediction of its behaviour in accordance to physical laws (Moore and Wilhelms 1988). In the simplest case, the question is if models associated to the virtual objects have collided; sometimes it is necessary to find contact points or distance between disconnected objects. When there is penetration between them, minimal displacement required to separate them has to be found. In systems based on physical models

with collision response, penetration distance between objects is used for computing interaction forces (Baraff 1997, Teschner *et al.* 2005).

There are two kinds of algorithms available in collision detection of convex virtual objects represented by B-Reps geometrical models: featured based and simplex based. Featured based collision detection algorithms perform some geometrical calculations on the object features (vertices, edges and faces forming object borders) and determine if objects are disjoint; if they are, then minimal distance between them is found; when distance is zero, the corresponding pair of convex polyhedra is in contact (Layer 2002). Simplex-based algorithms look for the existence of a separation plane in between a pair of convex polyhedral objects, as in the one used here known as the Gilbert–Johnson–Keerthi algorithm (1998), (see also Moore and Wilhelms 1988, Etmuss *et al.* 2003).

Collision detection algorithms are executed faster if they rely on hierarchical data structures such as boundary volume hierarchies. Among them axis aligned bounding boxes (AABB) are preferred when working with deformable objects (Terzopoulos *et al.* 1987). When performed in a brut force fashion, calculation of the collision between two polygonal objects has a complexity $O(n^2)$; therefore, the need to apply spatial subdivision methods to accelerate the execution of this task whose complexity then becomes $O(n \log n)$ (Marmolejo *et al.* 2005).

When performing collision detection tasks with deformable objects, among bounding volumes (BV), AABB along with binary tree spatial subdivision are preferred (van den Bergen 1997). Here the BV is a parallelepiped whose orientation coincides with those of the world coordinates and whose position and size is such as to enclose each of the objects present in the virtual scene.

Contact detection algorithm is to be done continuously. The first step is to detect if the BV attached to a pair of objects are touching; if they are not, a FALSE signal is produced, but if BVs are in contact, a TRUE signal triggers BV subdivision in two halves. In the second step, each new half is examined for collision, subdividing in two only the one that remains in contact with the part detected as touching the other object semi-box, obtaining a binary tree associated to each virtual object.

Tree depth depends on some specific criteria. If leaves of both binary trees are still in contact, collision test is to be executed for every pair of faces contained in each object tree leaf, identifying, if in touch, a pair of faces, each belonging to one of the pair of objects in question (Marmolejo *et al.* 2005).

Reaction force direction is taken as the normal component of the applied force and used to compute forces exerted between the two objects. Since the objects must not overlap, the set of points returned are constrained to remain on the surface of both objects. Upon penetration, forces are applied to each object so that they will separate quickly while maintaining the proper collision response behaviour.

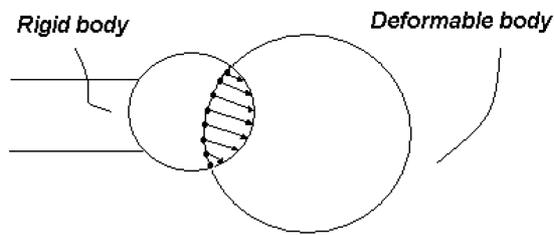


Figure 8 Detection of collision between a rigid and a deformable object.

Penetration depth for each point is defined as the minimum translational distance and direction for each point to escape penetration. It is difficult to describe what useful penetration information means; this is largely dependent on what the simulation is trying to accomplish. For rigid body and non-volume preserving deformable body simulation, the useful information is the identification of points inside the boundary of the object. A volume preserving deformable body simulation might require all points within the volume of penetration and their penetration depth. Figure 8 shows two bodies penetrating and the information returned by the collision detection. The dotted area around the deformable body is defined by the collision tolerance. When the rigid body is within the boundary, it is considered to be colliding.

For tasks in which virtual deformable objects are touched by a robot fingertip, the tool may be considered as a single point. When a collision is detected, the robot fingertip is located inside the object. In this case, the haptic feedback will be proportional to the distance between the actual fingertip position and the location on the object surface where the fingertip has initiated the contact. The corresponding deformation is calculated with this distance, while the reaction force is calculated with

$$F = kx. \tag{7}$$

Since computation steps are small, the supposition that the fingertip follows a linear path through time t is not restrictive. Locating the intersection point between this line and the touched object face, identified in the collision detection module, the following parametric equation can be used:

$$P(w) = (1 - w)P + wP_0 = P - w(P_0 - P), \tag{8}$$

wherein P and P_0 are the points in which the robot fingertip was at the beginning and at the end of the animation step. Consequently, in the next step, the corresponding deformation on the object surface is obtained by moving these triangle vertices until P_0 is in its surface. To detect collision between a moving point and a fix triangle (Figure 9), the value of the parameter w must be calculated using the next equation (Levey *et al.* 1999):

$$P + w(P_0 - P) = P_0 + (P_1 - P_0)u + (P_2 - P_0)v.$$

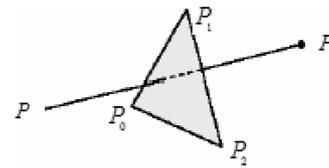


Figure 9 Intersection between a line and a planar triangle.

This vector equation contains three scalar equations with three unknowns; it can be solved by parametric variables u, v and w through a matrix inversion. First, verifying collision, for $0 < w < 1, u > 0, v > 0$ and $u + v = 1$; and second, finding the place where the line intersects the triangle by substituting the value of w in Equation (8). After calculating penetration distance through $x = |P(w) - P_0|$, the reaction force may be obtained by Equation (7).

EXPERIMENTAL RESULTS

Experiments were realised on the HBT system whose diagrams are shown in Figures 1 and 2. This system is constituted by the 6 DoF haptic device, the electronic system and the virtual environment. The electronic system used to control and communicate the haptic device is based on a DSP – TMS320F2812 from Texas Instruments, whose principal technical characteristic are

- a calculating power of 150 MIPS–150 MHz,
- 256 KB memory,
- communication ports SPI, CAN and USB.

USB port (v1.0) is used for processing control loop to 166.7 Hz. To obtain more sensibility in force feedback, control loop has been processed to 2 KHz using SPI port.

The virtual environment is simulated on a digital computer (CPU AMD Athlon, 2 GHz, 512 Mb on RAM and video card NVIDIA with 32 Mb). The characteristics of this video card are

- resolution: 640×480 ,
- vertical frequency: 60 Hz,
- horizontal frequency: 31.5 Hz,
- colour depth: 16 bpp = 65 K colours.

A position–position control scheme (Figure 10), wherein the operator closes the control loop, is used to control master and virtual robots in the hybrid bilateral teleoperation system. The master device and virtual robot dynamics are described by $M(s)$ and $S(s)$, respectively. The controllers are represented by the gains K_{pm} and K_{ps} . The position error e_p is calculated as the difference between master and slave position ($x_m - x_s$). Finally, gain K_e represents the remote environment.

Our interest is focused in applications where the slave robot is in contact with the environment. This physical restriction that makes the force commands demanded by an operator produces very slow movements. In this case, the robot dynamics is linear and decoupled, and we can use

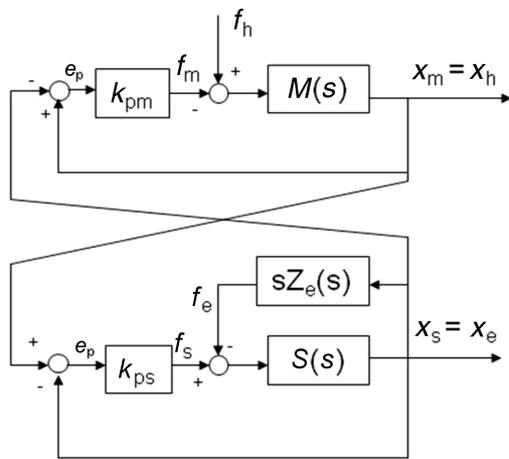


Figure 10 Position–position control scheme.

SISO controllers in each slave robot’s DoF (Hirche *et al.* 2007, Ferre *et al.* 2007).

A force–position control scheme is used in the force reflection loop (Figure 11). This figure shows the relation between forces applied by the operator (f_h) and the master (f_m). The force reflected to the operator should be proportional to the interaction force slave (f_s) and environment (f_e). The force reflection is manually scaled through a gain, to have soft or hard contacts. The control scheme guarantees stability, even in presence of transmission delays (Ferre *et al.* 2007).

The HBT virtual world was built by combining geometrical and dynamical models of the elastic sphere and the virtual slave robot, programmed with visual.net C++ with the simulation software Open Dynamics Engine (ODE) in Windows. The integration of the whole HBT system is a really hard task. In our prototype, the next three experiments were developed over this platform.

- (1) Simulation of a deformable sphere with zero gravity. The sphere has no interaction with the rest of the virtual environment. Evolution of the sphere shape is illustrated in Figure 4. Figure 15 shows the position evolution of a point on the sphere surface when a small force is applied

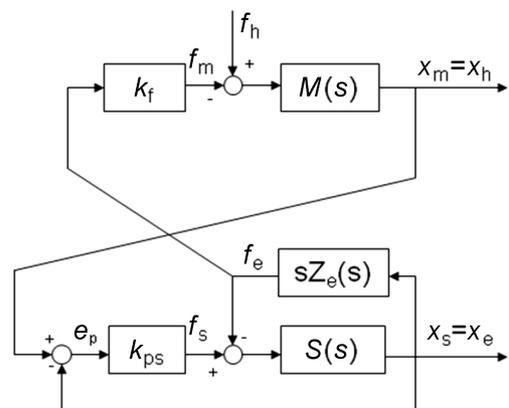


Figure 11 Force–position control scheme.

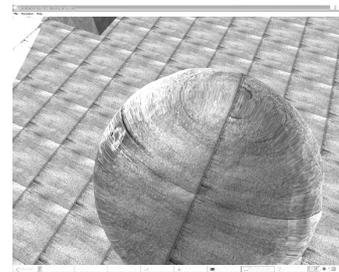


Figure 12 Deformable sphere interacting with rigid bodies, in presence of gravity.

- on a point of its surface, managing to deform the surface submerged in an environment with gravity zero and goes recovering its shape without being destroyed in a long time. The radius of the sphere is the steady state value of this point position, measured in metres (0.25 m).
- (2) Simulation of a deformable sphere with gravity equal to 9.81 m/s^2 , interacting within the rigid objects of the virtual environment. Figure 12 depicts the obtained results.
- (3) Collision between the virtual slave robot and the sphere producing some deformation, when the virtual robot is teleoperated with the haptic device. Figure 13 shows the results obtained in this experiment.

Measurement of positions and forces

Visual rendering shows that the virtual slave robot follows every master robot (haptic device) movement performed by the user. With every contact detection between virtual slave’s operation point and any environmental object surface present in the virtual scene, computation of a reaction force to be rendered triggers in response. The user feels the force reflected to the master robot whose magnitude is proportional to the interaction extent, noticing an increased compliant behaviour while touching the elastic sphere when compared to that felt when the floor or a wall is touched. Evolution of the elastic sphere is shown in Figure 13 through six images isolated from a video taken to the visual rendering of the HBT system while touching the ball. The sequence shows the slave’s tip approaching stage, first contact, pressure exertion over the sphere’s surface, tip slippery and departing stage. When the applied force is vanishing, the sphere again recovers its original shape.

Figures 13–16 show the evolution along some seconds of position and force measurements corresponding to two of the three experiments mentioned above.

Figure 14 represents the position of one point over the sphere surface, i.e. the radius variations of the sphere at this point, corresponding to the first experiment. The initial position of this point is 5 mm over the steady state surface of the sphere. The force variations at this point are shown in Figure 14. The relationship between deformation and force in this experiment is shown in Figure 16.

Finally, the results obtained in the third experiment are shown in Figure 17. The graph in this figure represents

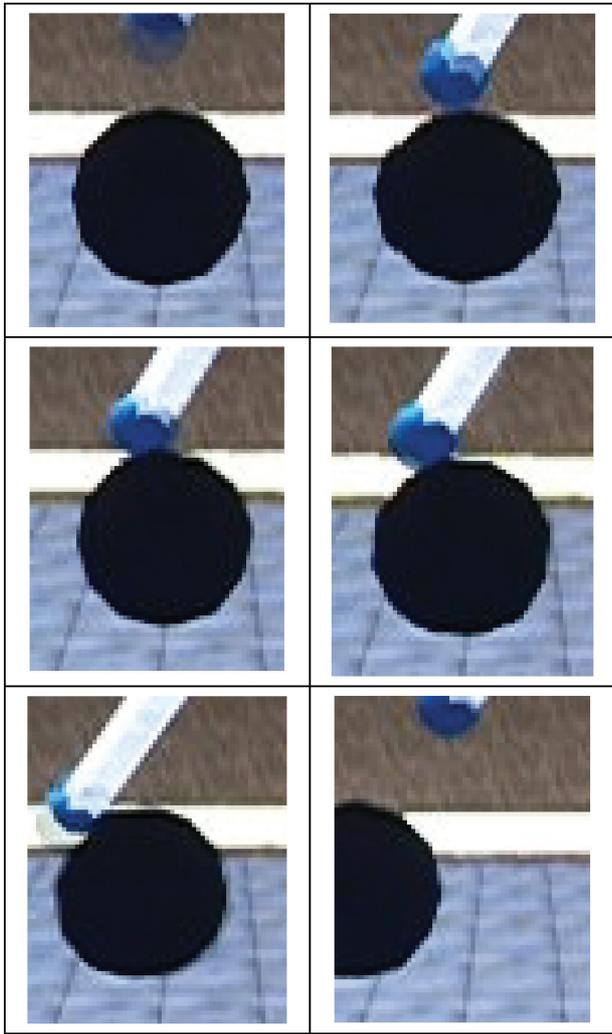


Figure 13 Collision between the virtual robot and the deformable sphere in a bilateral teleoperation task.

the reaction force of the sphere at the point touched by the virtual slave robot tip as a function of the deformation produced by the contact.

The graph in Figure 17 shows an erratic behaviour of the interaction force for small deformations, i.e. at the

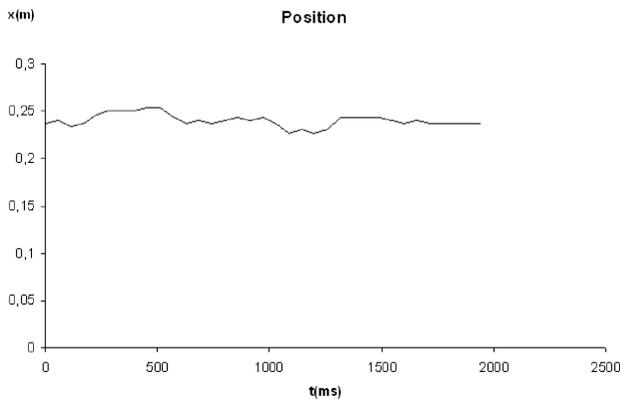


Figure 14 Position.

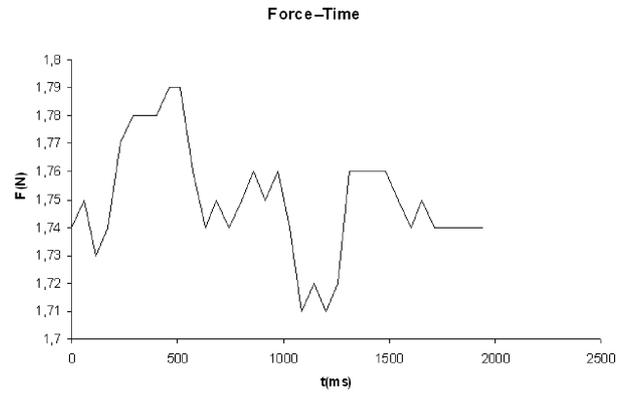


Figure 15 Force.

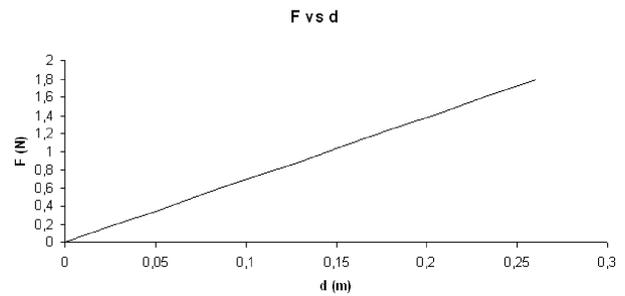


Figure 16 Relationship between position and force.

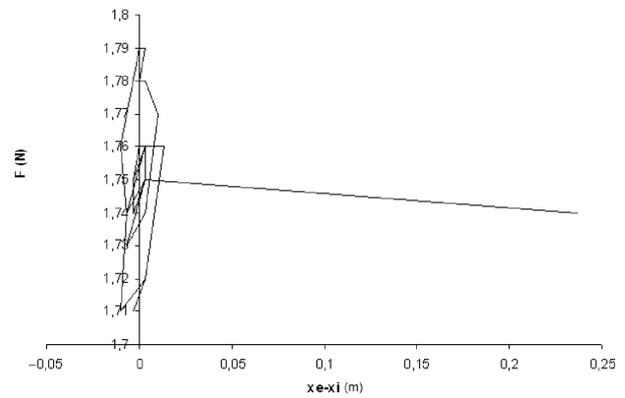


Figure 17 Evolution in the time.

beginning of the experiment, not only because it is not easy for the operator to take a stable contact with the virtual ball through the haptic device but the difficulty of the small force measurements.

CONCLUSIONS

The computational cost of the HBT is high since it includes the real time calculation of kinematics and collision detection; dynamics, position-position and force-position control loops, and visualisation of the virtual world.

Nevertheless, the structure of the developed algorithmic and the used video card, characterised by its high speed and big memory, permits to obtain a suitable response of the HBT in accord to the known visual and haptic restrictions. To appreciate correctly collisions and deformation

of elastic objects, the duty cycle of visual render must be in the range of 10–60 Hz. On the other hand, to have stable behaviours, the force feedback from slave robot to the haptic device must be, for soft contact forces, of 0.1–1 kHz., and for hard contact forces of 10–100 Hz.

The behaviour of the HBT is very close to the real bilateral teleoperation system; then it is possible to use the HBT in the analysis and tuning of the control loops of the real system. Moreover, the HBT will be useful in a lot of teleoperation applications like surgical training, spatial teleoperation, etc. Ergonomics of the haptic device is important in all kinds of bilateral teleoperation tasks but when telemanipulated objects are elastic this property is more relevant.

The obtained experimental results demonstrate that it is feasible to model the interaction between rigid and deformable bodies within the virtual environment to calculate the visual and haptic renders in an HBT system. Furthermore, the haptic interface of the HBT interacts with the virtual environment, in bilateral teleoperation tasks, respecting the above mentioned bandwidth restrictions.

Use of our hybrid bilateral teleoperation systems allows to carry out experiments for the development of the control laws in a safe way for both the users and equipment, mainly when the slave robot is used to amplify the human characteristics like dexterity, force or velocity.

ACKNOWLEDGEMENTS

The work of J.M. Ibarra-Zannatha was partially supported by grant SAB2003/0188 of the MECED (Spanish Ministry of Education) during his sabbatical stay at DISAM-UPM. Also we thank to CONACYT by the supports offered to Salvador Cobos-Guzmán and to Universidad Autónoma de Sinaloa and PROMEP for the support given to Claudia Marmolejo-Rivas.

REFERENCES

- Adams R, Hannaford B. 1999. Stable haptic interaction with virtual environments. *IEEE Trans Rob Autom*, 15(3):465–74.
- Baraff D. 1997. An Introduction to Physically Based Modeling: Rigid Body Simulation I-Unconstrained Rigid Body Dynamics. Robotics Institute, Carnegie Mellon University. Available from: <http://www.cs.cmu.edu/~baraff/sigcourse/notesd1.pdf>.
- Barbagli F, Prattichizzo D, Salisbury K. 2005. A multirate approach to haptic interaction with deformable objects single and multipoint contacts. *Int J Rob Res*, 24(9):703–15.
- Bourguignon D, Canni MP. 2000. Controlling anisotropy in mass-spring systems. In: Proceedings of the Eleventh Eurographics Workshop on Computer Animation and Simulation (EGCAS), Berlin: Springer Verlag, p. 113–23.
- Cavusoglu MC, Sherman A, Tendick F. 2001. Bilateral controller design for telemanipulation in soft environments. *Proc. of the ICRA 2001*, 1:1045–52.
- Cavusoglu MC, Sherman A, Tendick F. 2002. Design of bilateral teleoperation controllers for haptic exploration and telemanipulation of soft environments. *IEEE Trans Rob Autom*, 18(4):641–47.
- Cavusoglu MC, Tendick F. 2000. Multirate simulation for high fidelity haptic interaction with deformable objects in virtual environments. *Proc. of The IEEE ICRA 2000*, 3:2458–65.
- Colgate JE, Stanley MC, Brown JM. 1995. Issues in the haptic display of tool use. In: Proceedings of the International Conference on Intelligent Robots and Systems, Pittsburgh, PA, vol. 3, p. 3140.
- Corso JJ, Chhugani J, Okamura AM. 2003. Interactive haptic rendering of deformable surfaces based on the medial axis transform, *Eurohaptics 2003*. Available from: <http://www.eurohaptics.vision.ee.ethz.ch/2002/XCorso.pdf>.
- Cotin S, Delingette H, Ayache N. 2000. A hybrid elastic model allowing real-time cutting, deformation and force feedback for surgery training and simulation. *Vis Comput*, 16(8):437–52.
- Debunne G, Desbrun M, Barr A, Canni MP. 1999. Interactive multiresolution animation of deformable models. In: Proceedings of the Tenth Eurographics Workshop on Computer Animation and Simulation, '99; p. 133–44.
- Debunne G et al. 2001. Dynamic real-time deformations using space and time adaptive sampling. In: Computer Graphics Proceedings, Annual Conference Series, *ACM SIGGRAPH 2001*, New York: ACM, p. 31–6.
- Delingette H, Ayache N. 2004. Soft tissue modelling for surgery simulation. In: Computational Models for the Human Body, Handbook of Numerical Analysis, Ayache N, ed. (Ed: Ph. Ciarlet), Elsevier.
- Duriez C, Andriot C, Kheddar A. 2002. Interactive haptics for virtual prototyping of deformable objects: snap-in tasks case. *Eurohaptics 2002*. Available from: <http://www.eurohaptics.vision.ee.ethz.ch/2003/08.pdf>.
- Etmuss O, Groos J, Strasser W. 2003. Deriving a particle system form continuum mechanics for animation of deformable objects. *IEEE Trans Vis Comput Graph*, 9(4):538–50.
- Ferre M, Barrio J, Melchiorri C, Bogado JM, Custedo PL, Ibarra JM. 2007. Experimental results on bilateral control using an industrial telemanipulator, *Advances in Telerobotics*. In: Springer Tracts Advanced Robotics, vol. 31, Berlin-Heidelberg: Springer, p. 177–90.
- Gilbert EG, Johnson DW, Keerthi SS. 1998. A fast procedure for computing the distance between complex objects in three-dimensional space. *IEEE J Rob Autom*, 2(4):193–03.
- Hannaford B. 1989. Stability and performance tradeoffs in bi-lateral telemanipulation. In: Proceedings of the IEEE International Conference on Robotics and Automation. Scottsdale, AZ, USA, p. 1764–67.
- Hirche S, Ferre M, Barrio J, Melchiorri C. 2007. Bilateral control architectures for telerobotics. In: Advances in Telerobotics, Springer Tracts Advanced Robotics, vol. 31, Berlin-Heidelberg: Springer, p. 163–76.
- Khatib O, Brock O, Chang KS, Ruspini D, Sentis L, Viji S. 2004. Human centered robotics and interactive haptic simulation. *Int J Rob Res*, 23(2):167–78.
- Kheddar A, Neo ES, Tadakuma R, Yokoi K. 2007. Enhanced teleoperation through virtual reality techniques. In: Advances in Telerobotics. Springer Tracts Advanced Robotics, vol. 31, Berlin-Heidelberg: Springer, pp. 139–62.
- Kuroda Y et al. 2002. Haptic force feedback with an interaction model between multiple deformable objects for surgical simulations. *Eurohaptics*. Available from: <http://www.citeseer.ist.psu.edu/637961.html>

- Layer E. 2002. *Modelling of Simplified Dynamical Systems*. Berlin: Springer.
- Leech JP. 1989, www.neubert.net/Htmapp/SPHEmesh.htm
ljp@sgi.com.
- Levey E, Peters C, O'Sullivan C. 1999. New metrics for evaluation of collision detection techniques, Tech. Rep. TCD-CS-1999-55, Dublin 2, Computer Science Department, Trinity College Dublin, Ireland.
- Lin MC, Gottshalk S. 1998. Collision detection between geometric models: a survey. In: *Proceedings of IMA Conference on Mathematics of Surfaces*. Oxford: Oxford University Press.
- Mahvash M, Hayward V. 2004. High fidelity haptic synthesis of contact with deformable bodies. *IEEE Comput Graph Appl* (Special issue on haptic rendering), 24(2):48–55.
- Mahvash M, Hayward V. 2006. High-fidelity passive force-reflecting virtual environments. *IEEE Trans Rob*, 21(1):38–6.
- Marmolejo C, Ibarra-Zannatha JM, Basañez L. 2005. Simulation of the dynamical behavior of 3D deformable objects: Two case study. In: *Proceedings of Mexican Congress of Automatic Control AMCA 2005*. (in Spanish), Mexico City.
- Miller B, Colgate J, Freeman RA. 2000. Guaranteed stability of haptic systems with nonlinear virtual environments. *IEEE Trans Rob Autom*, 16(6):712–19.
- Moore M, Wilhelms J. 1988. Collision detection and response for computer animation. *Comput Graph*, 4(22):289–98.
- Otaduy MA, Lin MC. 2006. A modular haptic rendering algorithm for stable and transparent 6-DOF manipulation. *IEEE Trans Rob*, 22(4):751–62.
- Siciliano B. 2000. *Modelling and Control of Robot Manipulators*, 2nd ed. Texas: Springer Verlag.
- Terzopoulos D, Platt J, Barr A, Fleischer K. 1987. Elastically deformable models. *ACM SIGGRAPH Computer Graphics*, 21:205–14.
- Teschner M et al. 2005. Collision detection for deformable objects. *Comput Graph Forum*, 24 (1):61–81.
- Turro N, Khatib O, Coste-Maniere E. 2001. Haptically augmented teleoperation. *Proc. of the 2001 IEEE ICRA*, 1:386–92.
- van den Bergen G. 1997. Efficient collision detection of complex deformable models using AABB trees. *J Graph Tools*, 2(4):1–13.
- van Gelder A. 1998. Approximated simulation of elastic membranes by triangle meshes. *J Graph Tools*, 3(2):21–42.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

