Modelling and Haptic Interaction with non–rigid materials

U. Cugini, M. Bordegoni, C. Rizzi, F. De Angelis, M. Prati
Dipartimento di Ingegneria Industriale - Università di Parma - Italy
[cugini, mb, rizzi, fda, mprati]@ied.unipr.it

Abstract

This report presents an overview of main modelling and simulation techniques for non–rigid objects discussing advantages and disadvantages. Related techniques to face several problems rising when dealing with deformable objects simulation are also discussed. They concern numerical solvers, constraints management and collision detection. In the simulation of non-rigid objects another relevant issue is the provision of operational modality at higher and higher level and more and more user-oriented. The paradigm of haptic interaction recently developed seems promising to achieve this goal. Haptic interaction allows the users to feel several physical properties of the modelled objects through the manipulation of their virtual representation. Thus, second part of the report focuses on the state-of-the-art of haptic devices, technologies and applications concerning interaction with non-rigid models. A classification from different point of views is proposed. The specific work our group is performing in the area of non-rigid objects modelling and simulation and haptic interaction with non-rigid models is also illustrated.

Keywords: Deformable objects, physically based modelling, haptic devices, haptic interaction.

1. Introduction

Modelling non-rigid materials is an important issue not only for computer animation, but also for several industrial contexts concerned with the design and manufacturing of flexible products. Some examples are clothing, automotive and aerospace sectors. Different approaches can be found in literature to model and simulate the behaviour of non-rigid materials, as the finite element approach well known within engineering community or the particle-based approach mainly diffused within the computer graphics community for animation purposes. Textile industry is a typical sector where both approaches have been applied in order to simulate cloth behaviour but having in mind different goals. Computer graphics community puts most emphasis on visual realism, while textile scientists on physical accuracy studying the behaviour of the material according to its mechanical properties.

Many aspects have to be considered in order to make systems based on physically-based representation as much as possible practical everyday tools: material modelling, techniques to speed up calculation time, collision detection algorithms, and so on.

In the evolution trends of simulation, another main issue concerns the provision of operational modality at higher and higher level and more and more user-oriented. This can be achieved by means of introducing new interaction paradigms and new technology in the simulation phase. The paradigm of haptic interaction recently developed seems promising to achieve this goal. Haptic interaction allows the users to feel several physical properties of the modelled objects through the manipulation of their virtual representation. It is possible to virtually touch the object and feel the object shape, the object texture, the body properties, etc..

Haptic technologies are evolving quickly and applications of these technologies are growing exponentially.

The growing interest in both mentioned areas is underlined by the increasing number of workshops and conferences on topics related to physically based modelling, haptic interaction and their applications [1][2][3]. Besides, main conferences on computer graphics, like SIGGRAPH, have included courses and tutorials on haptics [4] [5] and physically based modelling [6] [7][8].

This report presents an overview on:

- techniques to model and simulate the behaviour of non-rigid materials;
- haptic devices, technologies and applications concerning interaction with non-rigid models, focused on the state-of-the-art of this research.

Published by the Eurographics Association ISSN 1017-4656

First part of the document describes main approaches to model/simulate deformable objects behaviour (section 2), and related techniques to face various problems, from numerical solvers to algorithms for collision detection (section 3). Second part (sections 4-6) concerns haptic devices classification from different point of views and haptic interaction with models of deformable objects. The specific work our group is performing in the area haptic interaction with non-rigid models is also illustrated.

2. Approaches to model and simulate non-rigid materials

Modelling and simulation of deformable objects are attracting more and more people from research and industrial communities. In this section a classification of main approaches found in literature to model and simulate the behaviour of non-rigid objects will be presented. Sources have been previous works presented in [9], mainly related to clothing simulation, [10], and courses and tutorials of SIGGRAPH conferences held in 90s.

2.1 Classification

Modelling and simulation techniques can be classified into three main categories [9]: *geometry-based*, *physically-based* and *hybrid*. A description of each category is presented, even if main focus has been on the physically-based approach that seems to be that one producing more interesting results.

2.1.1 Geometry-based approach

Models belonging to this category focus on the appearance of the non-rigid objects and do not include any information regarding the object physical properties in the representation. For example, considering a deformable object like a piece of fabric, it is possible to simulate folds and creases that look real but that do not have correspondence with the real physical behaviour of the object. Tools based on this approach can be considered as “drawing tools”, i.e., utilities but not independent modules to simulate and analyse the behaviour of deformable objects.
They use techniques such as catenary curve fitting [11], polygonisation [9] or interpolation [12].
The first important computer graphics model was proposed by Weil [11]. It was targeted exclusively for cloth drape simulation. His model was purely geometric and based on catenary and splines to simulate folds of a fabric suspended at constraint points. It didn’t simulate the material properties but only surfaces that look similar to draped cloth.

Another example could be the work described in [43] where authors propose an efficient algorithm for preserving the total volume of a deformable solid undergoing free-form deformation. The algorithm computes the new node positions of the deformation lattice, while minimising the elastic energy subject to the volume preserving criterion.

Main advantage of these techniques is execution speed since they don’t require complex equations to be solved. However they have relevant disadvantages [9]:

- results are not physically accurate; in fact, the representation is purely geometric and cannot simulate the mechanical behaviour of the material;
- they require very often the user’s interaction with the model in order to reach the desired configuration.

Fields of application are mainly related to computer graphics and animation, even if some works focused on the automation of design process, such as that one for garment manufacture [12].

2.1.2 Physically-based approach

Traditional geometric modelling techniques are not adequate to describe non-rigid objects. Therefore it is necessary to adopt a mathematical model that permits to describe, not only the geometry, but also physical properties, such as elasticity and viscosity.
Clearly the geometry does not play anymore the major role: the object shape depends on the forces applied and on its initial state. The geometry becomes time and forces dependent [13].
A physically based model is a mathematical representation of an object, and of its behaviour, that incorporates forces, torques, energies, and other attributes of Newtonian physics. This model is *active*, that is, it reacts in a natural way to external forces applied (such as gravity), to constraints or to impenetrable obstacles (such as a table) as one expect with a real object.
Physically based models can be categorised as follows [6] [10] [14]:

- *continuous models*;
- *discrete models*.

2.1.2.1 Continuous models

Continuous models employ continuum representation; for example, a piece of fabric is modelled as an elastic sheet.

The differential equations in continuos material are discretised using finite element or finite difference methods and then integrated through time [10]. Finite Element Method (FEM) is well known within the engineering community. Several textbooks can be found on finite element theory providing a rigorous theoretical treatment; herein we recall a basic definition, a detailed description can be found in [15] [16].

Finite element method is a numerical procedure that can be used to analyse structures and common applications including static, dynamic and thermal behaviour of physical systems and their components.

The finite element method [15] considers a deformable object to be an assembly of finite-sized particles, named *finite elements*. The points where the finite elements are connected are named *nodes* and the definition of nodes is called *discretisation* or *mesh.* The behaviour of the particles and the overall structure is obtained by formulating a system of algebraic equations that can be solved with a computer.

This method is used in different industrial contexts (automotive, aeronautics, textile, etc.) as well as in computer graphics and animation.

Continuos models have some disadvantages as follows [10] [17]:

- they are not easy to handle because they involve a large amount of computations;
- difficult handling of high discontinuity and non-linear cases;
- different behaviours and systems have to be represented by using different models.

First important model (in cloth modelling) of this category was introduced by Feynman [36], then more general continuous models have been proposed [17] [18] [21] [37].

In the following we briefly described some works that led to the development of continuos models adopting finite element method.

Terzopoulos et al.

This team introduced two formulations for deformable objects: a *primal* formulation, and a *hybrid* formulation.

Terzopoulos, Platt, Barr and Fleisher introduced first model, based on the elasticity theory, in 1987 [18]. Given ***u*** the material co-ordinates of points in a body Ω (Fig. 1), the position of point ***u*** on a body Ω is governed by

$$\mathbf{x}(\mathbf{u},t) = [\mathrm{x}_1(\mathbf{u},t), \mathrm{x}_2(\mathbf{u},t), \mathrm{x}_3(\mathbf{u},t)]$$

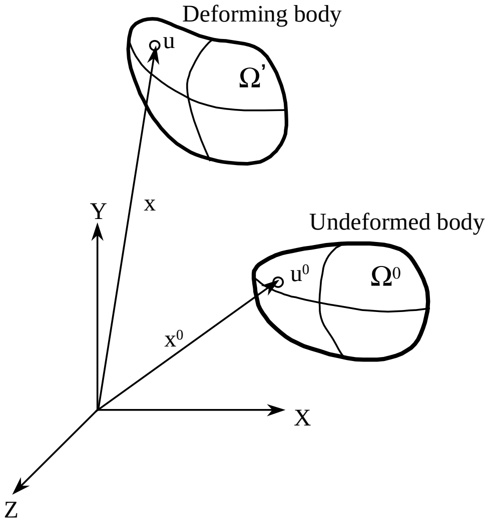

Fig. 1: *Geometric representation of primal model*

Using Lagrange's equation of motion, a deformable object motion is governed by

$$\frac{\partial}{\partial t}\left(\mu \frac{\partial \boldsymbol{x}}{\partial t}\right) + \gamma \frac{\partial \boldsymbol{x}}{\partial t} + \frac{\partial \varepsilon(\boldsymbol{x})}{\partial \boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x},t) \qquad (1)$$

where

$\boldsymbol{x}(\boldsymbol{u},t)$	is the position at time t
$\mu(\boldsymbol{u})$	is the mass density at ***u***
$\frac{\partial}{\partial t}\left(\mu \frac{\partial \boldsymbol{x}}{\partial t}\right)$	represents the inertial forces
$\gamma(\boldsymbol{u})$	is the damping density at ***u***
$\gamma \frac{\partial \boldsymbol{x}}{\partial t}$	represents damping
$\varepsilon(\boldsymbol{x})$	is a functional which measures the net instantaneous potential energy of the elastic deformation of the body
$\frac{\partial \varepsilon(\boldsymbol{x})}{\partial x}$	is the elastic force due to deformation
$\boldsymbol{f}(\boldsymbol{x}, t)$	represents the net externally applied forces

The simulation of the model dynamics is obtained discretising the equation (1) using finite element method (or finite difference method), then integrating resulting ordinary differential equations through time. The authors used a numerical step-by-step procedure that converts the system of non-linear ordinary differential equations into a sequence of linear algebraic systems.

This model was targeted for generalised deformable objects and was applied for realistic animation of cloth, piece of paper, metal bar or rubber.

Nevertheless this model had the drawback already mentioned and discrete equations become ill-conditioned when rigidity increases [17].
Thus, Terzopoulos and Witkin [17] [19] introduced the *hybrid* model as the sum of two components (Fig. 2):

$$\boldsymbol{q}(u,t) = \boldsymbol{r}(u,t) + \boldsymbol{e}(u,t)$$

where

- ***r*** *(u,t)* is the rigid *reference component* that evolves according to the laws of rigid body dynamics
- ***e*** *(u,t)* is the deformation component that models the difference between the actual shape of the model and its reference shape

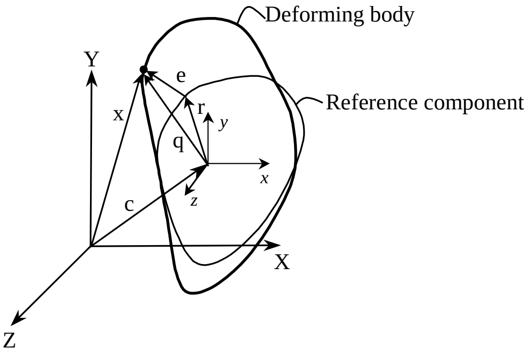

Fig. 2: *Geometric representation of hybrid model*

These components are expressed relative to a reference frame ϕ whose origin coincides with the body centre of mass ***c****(t)*.
e *(u,t)* is animated with a linear formula for the energy of deformation:

$$\varepsilon\,(e) = \int E(\omega(e, e_u, e_{uu} \ldots.)du$$

where *E* (density of elastic energy) is a linear combination of ***e*** *(u,t)* partial derivatives.
Some inelastic deformations, such as viscoelasticity, have been treated by the authors generalising the hybrid formulation [19] [20].

Thalmanns' team
This team has been developed a cloth modelling and animation software, still now evolving, at Geneva's MIRALAB and the Swiss Federal Institute of Technology's Computer Graphics Lab. (see web site: miralabwww.unige.ch).
They considered different models, and various solutions have been proposed through time to face all aspects related to this topic.
Their aim was to define a model specifically for cloth to be a part of an animation tool for clothing virtual actors. They addressed cloth visualisation with an approach similar to the manufacturing of a garment by a tailor [23] [24].
First model was derived from Terzopoulos et al.'s primal formulation introducing some improvements, such as more accurate damping using Raleigh's dissipative function, and the simulation of inelastic contacts.
Their model was formulated as [23]

$$\rho(\boldsymbol{a})\frac{d^2\boldsymbol{r}}{dt^2} + \frac{\delta}{\delta\boldsymbol{r}}\iint_{\Omega}\left\|\boldsymbol{E}\right\|^2 da_1 da_2 + \frac{\delta}{\delta\boldsymbol{v}}\iint_{\Omega}\left\|\dot{\boldsymbol{E}}\right\|^2 da_1 da_2 + \frac{\delta}{\delta\boldsymbol{r}}\iint_{\Omega}\left\|\boldsymbol{B}-\boldsymbol{B}_0\right\|^2 da_1 da_2 = \sum \boldsymbol{F}_{\mathrm{ex}}$$

where $\boldsymbol{E} = \boldsymbol{G} - \boldsymbol{G}^{\boldsymbol{0}}$ is the Lagrangian strain tensor.
In Terzopoulos et al.'s formulation all motion is subject to damping, with Thalmanns' approach only surface deformations are damped. Terzopoulos et al. adopted a penalty method for collision management. This technique is not appropriate for cloth simulation, so, to avoid this problem, they used the principle of conservation of momentum for collisions handling.

Eischen et al.
Eischen et al. proposed a model based on stress – resultant and geometrically exact non-linear shell theory [21]. They developed a system based on non-linear shell theory to simulate 3D motion related to real fabric manufacturing processes.
The kinematic description of the shell begins parameterising the position of points within the shell, both on and off the midsurface.
As shown in the Fig. 3 points off the midsurface are located by a position vector $\boldsymbol{\Phi}$

$$\boldsymbol{\Phi}(\xi^1, \xi^2, \xi) = \phi(\xi^1, \xi^2) + \xi\,\boldsymbol{t}(\xi^1, \xi^2)$$

where

- ϕ is a position vector locating points on the shell's midsurface (reference surface)
- $\boldsymbol{t}$ is a unit vector directed along shell fibres that are initially perpendicular to the reference surface
- ξ^1 and ξ^2 are dimensionless (non-arc-length) curvilinear co-ordinates
- ξ is the through – thickness co-ordinate

The problem consists in determining the evolution of ϕ and $\boldsymbol{t}$ as the shell deforms under its own weight or external forces.
Enforcing linear and angular momentum allows development of a variational form of the non linear shell theory.

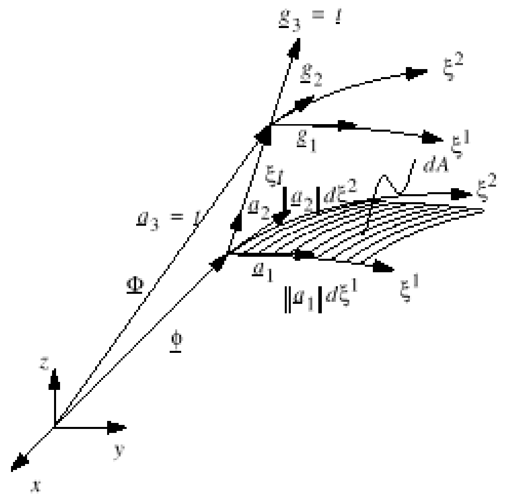

Fig. 3: *Shell configuration*

Standard finite element linearisation procedures brings to a matrix equation of the following form

$$\boldsymbol{K}(\phi,\boldsymbol{t})\begin{Bmatrix}\Delta\,\phi\\ \Delta\,\boldsymbol{t}\end{Bmatrix}=\boldsymbol{F}_{ext}-\boldsymbol{P}(\phi,\boldsymbol{t})$$

where

$\boldsymbol{K}(\phi,\boldsymbol{t})$ is the tangent stiffness matrix
$\boldsymbol{P}(\phi,\boldsymbol{t})$ is the internal force vector
$\boldsymbol{F}_{ext}$ is the external force vector

Eischen et al. provided their software with non linear material response based on Kawabata [22] bending test, fabric contact with rigid surface employing the penalty method, and adaptive arc – length control algorithm to accelerate computational speed without incurring in divergence problems.
As mentioned, the authors proposed their approach to fulfil manufacturing and design requirements in textile and clothing industry.

2.1.2.2 Discrete models

A discrete model describes a deformable object by combining very simple mechanical elements where all types of interactions (linear and non-linear) are managed in the same ways and the simulation is done computing the interaction laws among objects. In this approach, the object is discretised in small elements connected in some way.
Particle-based model is the most known and used discrete model [25-35][40]. By this model, also known as the mass-spring approach, an object is described by a set of particles with their mass, radius and other physical properties.
We can distinguish two techniques: *force-based,* and *energy based*.

Force-based [25]

The interaction laws among the particles are modelled by means of forces that determine the dynamic behaviour of the material.
It represents the forces among particles as differential equations and performs a numerical integration to obtain the position at each time step.
In a more rigorous way, given:

$S=\{P_1,P_2,\ldots\ldots,P_n\}$	a set of n particles
$m_i \quad 1\le i\le n$	the mass of the particle P_i
$\boldsymbol{p}_i(t)$ and $\boldsymbol{v}_i(t)$	the position and the velocity of the particle P_i at time t

the motion equation of the particle P_i is derived from Newton's second law:

$$\boldsymbol{f} = m\,\boldsymbol{a}(t) \qquad (2)$$

that can be written as a couple of first order ordinary differential equations:

$$\begin{cases}\dot{\boldsymbol{p}}=\boldsymbol{v}\\ \dot{\boldsymbol{v}}=\boldsymbol{f}/m\end{cases}$$

Both vectorial functions $\boldsymbol{v}(t)$ and $\boldsymbol{p}(t)$ can be written as three independent scalar functions (one for each co-ordinate). Therefore the particle system can be described by a set of $6n$ first order ordinary differential equations.
Determining velocity of each particle requires the solving of ordinary differential equations. Anyway force-based technique is quite simple and particularly right for dynamic analysis. The main drawback is the solution of differential equations system that can become stiff. A solution could be decreasing integration step, increasing, however, the calculation time.

Energy based [26]

This technique calculates the energy of the whole system from a set of equations and determines its shape by moving the points to achieve a minimum energy state.
Cope with the problem from this point of view permits a global approaches; moreover the definition of proper expressions defining various kinds of energy simplifies the optimisation of the model. Main drawbacks are the high computational time required by the algorithm that computes the minimum energy, and the fact that it

cannot produce the dynamic response required for animation.
In fact this model is mainly used for static simulation. An interesting work on this approach has been carried out by House and Breen [26]. Their work focused on a mechanical system made by fabric where bending and shear energies have been derived from data determined using the Kawabata Evaluation System [22]. Anyway they envisaged the need to move from an energy-based approach to a force-based approach to cope with dynamic analysis.

Main advantages of particle-based approaches are:

- semplicity and generality;
- they can be easily handled because global properties can be specified;
- they manage interactions with the world in a uniform way.

This model has been adopted to model and simulate different materials, such as clothes, rubber, wires, snake and worms, liquids, and others phenomena [26] [29] [31] [32] [35] [37] [53].
We recall here some works based on a discrete model.

Luciani et al.
Luciani et al. based their work on the following idea [10] [27] [28]:

… It is not always necessary to simulate the real physical structure of an object to reproduce its movement. The most important is to build a "mechanical representation" of the objects to animate.

They described deformable objects as some elementary masses linked together by *conditional connections*, such as damped springs, whose parameters can be changed through time.
In their model of deformable objects, parts that play an insignificant role, are neglected and a purely geometric model is adopted.
Interactions among objects can be considered a particular kind of connections. Each couple of mass points belonging to different objects is connected by a spring-like connection. When two points are far, the corresponding spring stiffness is equal to zero, when they gather, it increases simulating in this way the reactive force due to a collision.
The system they developed has been named CORDIS-ANIMA, and comprehends a modelling module and a simulation module.
It has been mainly used in the field of animation [28]. The system also featured a set of force feedback gestural transducers, a real-time visualisation processor and a real-time sound processor [27].

Breen et al.
Breen et al. propose a non-continuum interacting particle model right for simulating the draping behaviour of woven cloth. They base the work on the assumption [26]:

Cloth is a mechanism, and not a material

Their conviction is that cloth is best described as a mechanism of interacting mechanical parts rather than a substance. The model explicitly describes the micro-mechanical structure of cloth with a particle system [26] [30]. It represents the crossing points of the warp and weft threads as particles (Fig. 4).

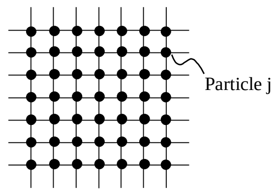

Fig. 4: *Particle representation of woven cloth*

The interaction between the particles are represented by energy functions. The strain energy of the crossing particle *j* is given by

$$U_j = U_{repel,j} + U_{stretch,j} + U_{bend,j} + U_{trellis,j}$$

where

$U_{repel,j}$ is the repulsion energy among particles

$U_{stretch,j}$ defines energy of tensile strain between each particle and its four-connected neighbours

$U_{bend,j}$ is the energy due to yarns bending out of the local plane of the cloth

$U_{trellis,j}$ is the energy due to bending around a yarn crossing in the plane

They derive an approximated expression for $U_{bend,j}$ and $U_{trellis,j}$ from the empirical data produced by the Kawabata Evaluation System [22]. They find the final equilibrium configuration of the cloth searching the minimum energy over the whole cloth. The adopted algorithm, called *stochastic gradient descent*, is quite slow and it cannot produce the dynamics response required for animation. To avoid this limitation they propose a method for the conversion of the particle draping model to a dynamics model. They simply differentiate the energy functions to produce a set of non-linear springs forces that are functions of particles displacement, and add inertia and damping terms. Work described in [26] is well suited for understanding differences between energy and force based approaches.

Eberhardt et al.
Eberhardt et al. extend the model of Breen et al.. They use Lagrangian dynamics reformulation of the basic energy equations and add kinetic and gravitational energy. They introduce [32] techniques to model experimental force data and the anisotropic behaviour and hysteresis of the cloth from Kawabata plots. Their system supports dynamic simulation considering air resistance, moving bodies and surface friction.
They describe the state of their system of n particles at time t by the sets x_0 , x_1 , x_2 , …, x_{n-1} and v_0 , v_1 , v_2 , …, v_{n-1} where the vector (x_0 , x_1 , x_2), (x_3, x_4 , x_5) and so forth are the locations of particles and (v_0 , v_1 , v_2), (v_3 , v_4 , v_5), … are the velocities. They calculate forces using the gradient of an energy function V. The Lagrange equation

$$\frac{d}{dt}\frac{\partial L}{\partial v_i} = \frac{\partial L}{\partial x_i}$$

fully describes the trajectory of the moving particle giving a second – order differential equation for each co-ordinate of the particle.
$L=E_{kin}-V$ is the Lagrange function

where
V is a cloth-specific potential in which they consider potential, tension, shearing and bending energy
E_{kin} is the kinetic energy of the system.

Their software uses a Runge–Kutta method with adaptive step – size control and despite its extra dimension of detail is faster than Breen et al.'s one.

Thalmanns' team
As already said Thalmanns' team first based their system on a continuous model that was an extension of Terzopoulos et al..
During system development new requirements arisen and they wanted to generalise the application range of their system moving towards a modified mechanical model able to handle non-linear deformations and discontinuous behaviours. They use elasticity theory and Newtonian dynamics. The animated deformable objects are represented as a particles system by sets of vertices forming irregular triangles (Fig. 5), allowing the simulation of a wide range of objects.

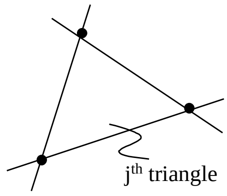

Fig. 5: *Particle representation with triangles*

Thus, using Newton's second law, the motion equation results in a system of equations resolved using the midpoint method [38].
They proceed in two steps:

1. first considering interparticle constraints and external ones;
2. then, collision are detected and discontinuous constraints (resulting from collisions with other objects) are handled.

They also introduce improved optimisation techniques for collisions detection and response [33] [39]. Some techniques are: proximity tracking and curvature-based techniques for self-collision.
Their system has been mainly used for animation and garment design.

Cugini et al.
Cugini et al. developed a system, named *SoftWorld*, specifically targeted to industrial applications [29] [41] taking into account that in industrial sectors it is not sufficient to produce simulations that look real, but both physical accuracy and visual realism are required.
SoftWorld adopts a particle model, force based.
Therefore a non-rigid material is discretised into a set of particles connected in some way by forces (internal forces) characterising the mechanical behaviour of the material. Fig. 6 shows the particle model of an elastomeric thick plate. The thick plate has been discretised as a set of cubic voxels placing particles as shown in Fig. 6b and viscoelastic forces characterising the viscoelastic behaviour of the material have been distributed as portrayed in Fig. 6c.
The main component of the system is the simulator that is the computational module comprising:

- an *ODE* numerical solver, whose aim is to compute the simulation step by step and based on Euler method;
- a *constraint manager*, whose aim is to compute additional forces and impulses to respect constraints and collisions with obstacles during

the simulation.

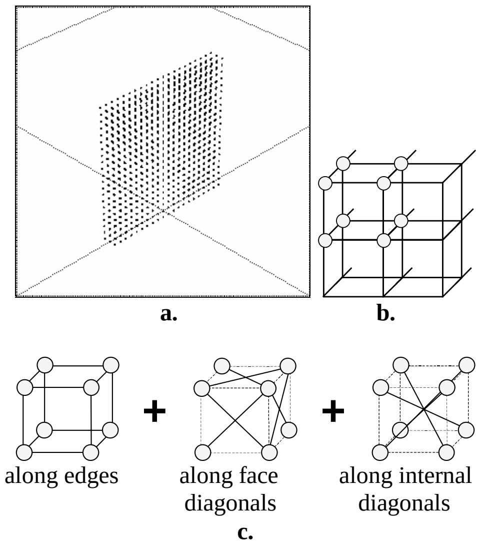

Fig. 6: *Particle-based model of an elastomeric thick plate*

The Dynamic constraint method (see section 3.2) has been adopted for constraints management since it permits to apply multiple constraints to the same particle and ensures the respect of all the constraints at each step of the simulation. This is done by solving a global linear equations system, representing all the active constraints, at each time step. Collisions management is based on bounding box hierarchy technique. (see section 3.3).
The prototype has been tested within the framework of international projects funded by the European Union, such as Brite/Euram projects NORMAH, SKILL-MART and MASCOT.
The test cases required the simulation of different types of deformable bodies: from flat objects, like fabrics for textile and clothing industry, to solid and wire-like objects for applications in automotive, aerospace and food sectors. The system has been partially integrated with a kinematic work-cell simulator [29], in order to obtain an integrated system to simulate manufacturing processes involving non-rigid products. Figures 7 and 8 portray two integrated simulations: first shows two co-operating robots manipulating three overlapped wires, the second one the grasping operation of a sponge.
The kinematic simulator computes the trajectories of the robot parts, the non-rigid material simulator calculates the effects of the gripper fingers on the shape of the flexible object and the result is a simulation showing the robotic system and the deformed material in the same space.

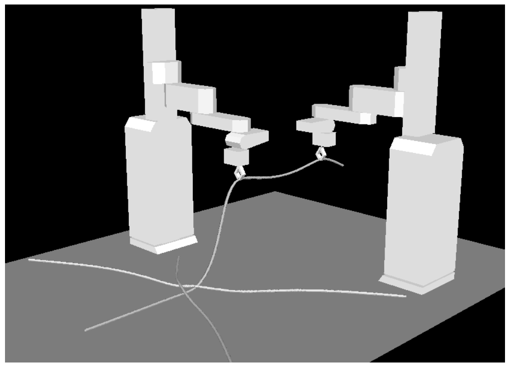

Fig. 7: *Two robots manipulating overlapped wires*

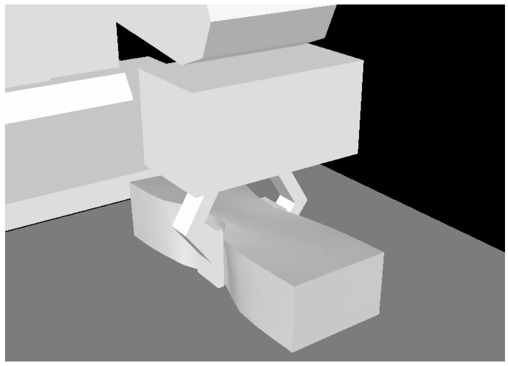

Fig. 8: *Grasping of a sponge*

2.3 Hybrid approach

Hybrid approach combines geometric and physical methods. Systems based on these techniques mainly rely on the geometric models and are oriented to interactive deformation of deformable object within the respect of certain physical laws.
An example is the work done by Taillefer [42] that can be considered an extension of Weil's work [11].
He subdivided folds in an hanging cloth into horizontal and vertical. The first are modelled using catenary, while the vertical using a relaxation process similar to Weil's one but with more external constraints.

3. Related issues

This section presents an overview of techniques for solving problems one can face when implementing systems to model and simulate non-rigid objects.
First part concerns methods for solving ordinary differential equations (ODE), then techniques to handle interactions of deformable objects with the environment are briefly described.

Usually the surrounding environment that is characterised by three types of interactions drives the behaviour of an object:

- *external forces,* such as gravity and wind;
- *constraints,* that restrict the allowed positions and movements of an object, that is, they are conditions that must be respected by the object during its motion. Various techniques can be found in literature.
- *Collisions with obstacles,* for example, when either a flexible object hits a rigid object or a flexible object penetrates itself (self-collision). Typically custom and optimisation techniques are implemented to detect and correct such situations.

3.1 Solving ordinary differential equations (ODE)

One main problem of system for physically based modelling is the computational time to solve ordinary differential equations. A small integration step provides more accuracy, but it increases excessively the computational time. At a parity of calculation step, a higher order integration method improves accuracy without raising too much the simulation time, but increasing the difficulties in the software development.

Numerical integration methods can be divided in *explicit* and *implicit* methods [38] [44].

Common explicit methods are:

- Euler method (I order);
- Midpoint method (II order);
- Runge – Kutta method (III, IV and V order).

Common implicit methods are:

- Implicit Euler method;
- Rosenbrook methods.

The explicit Euler method, based on the simple formula $y_{n+1} = y_n + h\dot{y}_n$ does not require difficult calculations or particular information in addition to the previous value of the unknown and its derivatives. Its real efficiency depends on the size of the step you can set to preserve accuracy and stability as well as the cost per step.

To improve performances the midpoint method can be used [33], in particular, the fourth-order version of the mid-point method, also known as fourth-order Runge-Kutta is especially widely used.

Eberhardt et al. [32], as already said, take a Lagrangian approach, adding kinetic and gravitational energy to the model, and simulate the resulting differential equations using an adaptive Runge-Kutta numerical integration algorithm.

More sophisticated methods can greatly outperform Euler's method because their higher cost per step is offsetted by the larger step sizes they allow.

To enhance numerical stability using explicit methods, Witkin [6] suggests applying at least a small amount of drag to each particle.

Implicit methods are very expensive to implement and cannot rival predictor – corrector or explicit methods such as Runge-Kutta in efficiency when the problem is not stiff. Their use is almost exclusively restricted to stiff systems, in which context their superior stability properties justify the high cost of implementation.

Witkin and Baraff [45] [46] use the implicit Euler method. This method, based on the formula $y_{n+1} = y_n + h\dot{y}_{n+1}$, requires the solution of a supplementary non linear algebraic equations system and, in dynamics, the evaluation of force derivatives.

3.2 Constraints management

Before describing the methods, we recall that there are two types of constraints: *equality* and *inequality* [47] that act in a completely different way during simulation. The former can be analytically expressed by an inequality, the latter by an equality.

There are different methods for constraints handling.They are [47] [48] [49]: *Penalty method, Lagrangian Constraint, Lagrange multipliers, Rate-controlled constraints,* and *Dynamics constraints.*

Penalty method

It is a technique from optimisation theory. Total energy of moving bodies is a decreasing function through time (for all dissipative models). In consequence dynamic animation can be viewed as an energy minimisation problem. This method consists in adding *penalty terms* to this energy to penalise the violations of constraints. To this end, "adding forces" are introduced in order to drive the system to a state compatible with the constraints. A new potential energy is introduced to calculate the adding force. This potential energy should be at the minimum when all constraints are fulfilled, or very close to the minimum when considering inequality constraints, and increasing when the constraints are violated. Main advantages of this method are ease of use and ease of implementation. Main drawbacks are: an intrinsic error due to the corrective force, constraints are never completely reached, and stiffness of equations.

Lagrangian Constraint

It retains many of the advantages of the previous method while avoiding many of the disadvantages. In Lagrangian constraints, forces that would cause the system to violate the constraints are cancelled independently of system state, and a force is substituted that gradually makes the system fulfil the constraints. Practically, for each constraint a new term of the potential energy is added and, the variables λ_i, named *Lagrange multipliers*, are introduced. With this method an inequality constraint is treated like an equality one. It retains the simplicity of the previous, but it augments the efficiency reducing the problem of differential equations stiffness. On the other hand, it is intrinsically inexact and the number of differential equations, that has to be solved, increases proportionally to the number of constraints.

Rate-controlled constraints

It is based on the hypothesis that all constraints are fulfilled at the initial state, and all terms of the force that cause the system to reach an illegal state are cancelled. The advantages are: the ease of implementation, the low computational cost, and the constraints are always respected. The main disadvantage relies on the fact than it not possible to apply multiple constraints to the particles, and therefore it is not general.

Dynamic constraints

This method is different depending on the type of constraint: equality or inequality. It is valid independently on the constraints number and reduces the problem of differential equation system stiffness. Main drawbacks are: the algorithm complexity, high computational cost, and problems when linear equation systems become ill-conditioned.

3.3 Collisions management

Collision detection is often the bottleneck of deformable objects simulation systems handling highly discretised objects.
The high number of objects and of surface elements forces the simulator to execute a continuous check of the possible situations of collision (collision detection) and takes a great part of the simulation time.
When a collision is detected, you must act directly on the vertices or on the forces to avoid collision (collision response). The selected method must provide realistic simulation leaving aside the complexity of the studied system. Therefore, two aspects should be considered*: collision detection* and *collision response*.
Instead of detecting collisions for each object (described by a set of particles) against all other objects, optimisation techniques have been introduced in order to reduce computational time.
Some techniques for collision detection optimisation are: *Voxel subdivision*, *Octree subdivision*, *Bounding box hierarchy*, *Proximity tracking*, and *Curvature – based*.

Voxel subdivision

The space is subdivided into an array of regular voxels (Fig. 9) and collision detection is performed only between objects sharing common voxels.

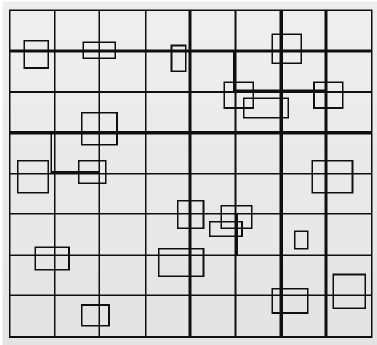

Fig. 9: *Voxel subdivision*

Octree subdivision [50]

It is a well known technique in geometric modelling. The original volume, say a cube, is partitioned into 8 cubes (octants) if it is non-empty. Recursively, each sub-cube is partitioned whenever non-empty, until some minimum size element is reached. Since empty cubes are not sub-divided, the storage space efficiency is increased. This process takes to a structure representing the shape of each object (Fig. 10). Detection is performed exploring this hierarchical structure.

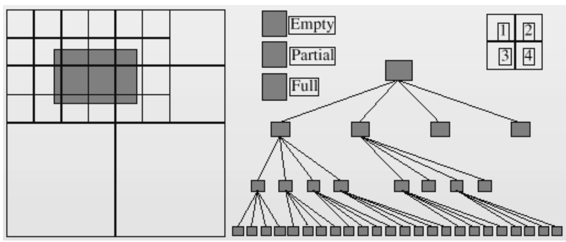

Fig. 10: *Octree subdivision*

Bounding box hierarchy

Objects are grouped hierarchically according to proximity rules (Fig. 11), and the detection is carried out by exploring bounding box intersections in the hierarchy.
Collision detection algorithm begins with a pre-processing step, in which a bounding box for every selected zone is computed [52]. The algorithm can be significantly improved by parsing the bounding box tree while eliminating rapidly collisions tests between elements that belong to two zones whose boxes do not intersect. In order to be accurate, the bounding box of each zone does not only bound the position of the zone at iteration $t_0+\Delta t$, but both its positions at t_0 and $t_0+\Delta t$ [51].

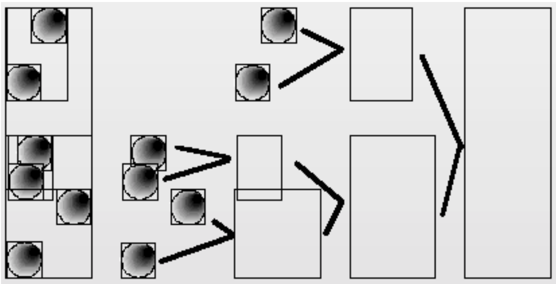

Fig. 11: Bounding box hierarchy

Proximity tracking

It keeps incremental information on the objects neighbourhood relations. Incremental algorithms update the existing proximity data at each animation step using some quick geometric computations, instead of using the whole global detection computation [33] [39]. This is a good approach when large surfaces collide but with small deformations and relative movements (Fig. 12).

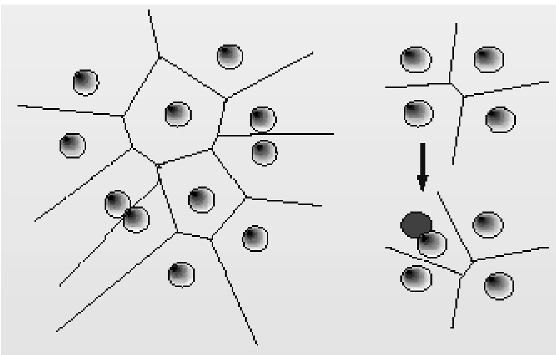

Fig. 12: *Proximity tracking*

Curvature – based

This technique uses curvature instead of bounding boxes. This optimisation is based on the property that if a given zone has a sufficiently "low curvature", it cannot self-intersect, and all the zones it includes do not intersect with each other (Fig. 13) [39] [51].

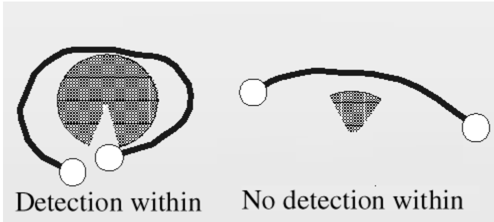

Fig. 13: *Curve-based Self collision detection*

Some techniques for the collision response are: well known *laws of friction*, "*temporary*" *force*, and *velocity computation*.

Friction

Provot [51] considers perfectly inelastic collision and applies general macroscopic laws of friction to describe what happens when two objects rest in contact after a collision. His approximation consists in considering that the forces implied are proportional to velocities, since it is obvious that the greater the impact velocity, the greater the force generated. He replaces in the laws of friction, the forces with the velocities and finally he finds the forces by calculating the velocity variation in the step involving the collision.

Temporary force

Dynamic simulation system must have a method for applying external forces to objects. Thus, when a collision is detected, a very stiff spring is temporarily inserted between the points of closest approach. This method is computationally expensive, stiffer springs mean stiffer equations, which require smaller time steps for accurate and stable numerical integration [54].

Velocity computation

Instead of calculating the reaction force or the new position after the collision, this method finds the new velocity applying the conservation of linear and angular momentum and determining how much kinetic energy is lost [54].

4. Haptic devices

In this section an attempt will be performed to classify the various existing haptic devices, and then a few examples of devices belonging to the various categories will be presented.
Before proceeding, we should define what we mean by "haptic device". The term "haptic device" is often used interchangeably with "force feedback device"; however, there are both examples of devices which are not force-feedback (e.g. "passive" devices) and force-feedback devices whose dominant interaction is not located at user fingertips, which are referred as "haptic".
In the following, we will use the term "haptic" accordingly to its wider acceptation, thus indicating any output device whose interface with the user occurs through its motor and/or tactile system.

4.1 Classification

Haptic devices may be classified by a number of different points of view. Here are some.

4.1.1 …. by how the haptic simulation is achieved

active devices

These are the so-called force-feedback devices that can exert random forces against the user. They can simulate quite well "any" situation but require a precise and timely accurate control of the exerted forces, which need to be updated at high frequency while user moves to keep the simulation consistent with user actions.

passive devices

The device can oppose to user movement (acting like a damper, or a brake) but can not push him back.
This kind of devices may be cheaper to build and may require slower refresh rates than previous ones, but are somehow limited in the set of situations which can be simulated (e.g. they can not simulate the sensation of releasing a spring after compressing it).

Please note that this doesn't mean that active devices are "bi-directional" and passive devices are "mono-directional"; that's an independent feature. Both active and passive devices can be either mono or bi-directional, and in fact examples for each combination do exist.

4.1.2 …. by number of degrees of freedom

input d.o.f.

The number of joints (or other features allowing user movement) whose position/angle is sensorised hence provide data which can be taken into account for simulation computation purpose.

output d.o.f.

The number of joints (or other features allowing user movement) which have actuators (whether active or passive) allowing the simulation application to perform the haptic rendering.

In general the number of input d.o.f. does not need to be the same as the number of output d.o.f. ; devices exist where they are different.

4.1.3 …. by how the high-frequency haptic loop is managed

software managed haptic loop

It is up to the application (or to the software driver) to supply data to the device at the required frequency. This usually imposes computational time constraints and arises real-time issues, as the application is generally required to be exactly synchronised with the haptic loop.
This also requires a communication channel between the host computer and the haptic device having a relatively high bandwidth, which cannot be provided by a standard serial/parallel port.
Devices of this category usually use dedicated controller cards to be installed in the computer, which manage the communication with the devices.

hardware managed haptic loop

The device has an on-board microcontroller that manages the haptic loop. Communication between the host computer and the microcontroller happens at a frequency low enough (10 or 100 times less) to be performed through a standard port (serial/parallel/game/etc.). The application doesn't need to be synchronised with the haptic loop.

Both approaches have their own advantages and disadvantages.
The first one (haptic loop software managed) is more expensive, as it requires controller cards and, being very computational demanding, fast computers, but it is also more flexible as it gives

the application a more free, direct and timely accurate control of the device behaviour.
The second one (haptic loop hardware managed) is cheaper, less computational demanding and free from real-time issues, but tends to be less precise (less timely accurate) and less flexible: often the on-board microcontroller accepts only a set of high-level commands defining built-in primitives, and doesn't allow direct access to the device (thus preventing to define new primitives).

One could say that the first approach is suitable for professional applications while the second one is for the consumer market, but this may be an overly simplified statement.
It is possible to imagine a version of the "hardware managed haptic loop" where the communication between the device and the application happens on a high bandwidth channel thus allowing the application complete and timely accurate control of the device while freeing it from the real-time issues.

4.1.4 …. by application domain

general-purpose devices

Devices in this category aim to be the haptic equivalent of a display for the sight or a speaker for the hearing: a generic tactile output device that can be used by any application.

application-specific devices

Devices in this category are built to simulate a specific task or operation better than a generic device could do, but can not do anything else.

The motor system is not as localised as eyes and ears are. A little head-mounted stereoscopic display projecting images toward the eyes and a couple of little headphones inserted in the ears can simulate almost perfectly any visual/audio stimulus of the real world, given that proper signals are sent to the devices. Motor skills and tactile perceptions are spread all over the body and that makes it difficult to implement "generic" haptic devices. Some attempts were made under assumptions/restrictions (e.g. "most of the haptic interaction happens through the right hand index fingertip") which do not always apply, and that's why many application-specific devices exist.

4.1.5 …. by user classes

academic

Prototypes of haptic devices built by researchers working in the area; many Universities have some.

professional

High-end haptic devices used by researchers, by professionals and in the industry to solve problems.

arcade rooms

Haptic devices used on the dedicated game machines found in arcade rooms. They were already in use since years, and they are now experiencing a rapid growth.

consumer

Low-cost haptic devices that have recently appeared on the consumer market. Used for home entertainment.

4.2 Examples

A few examples of devices belonging to the various categories follow in this section.

The CyberGrasp by Virtual Technologies (Fig. 14) [55] is a version of the popular CyberGlove augmented with force-feedback by a lightweight exoskeleton. It is a general-purpose mono-directional active device having 28 input d.o.f. (22 the glove, 6 the tracker) and 5 output d.o.f. The haptic loop is managed by the device controller and the application communicates high-level commands (only pre-defined primitives) at low frequency through RS-232 or Ethernet connection.

Figure 14: *Virtual Technologies CyberGrasp force-feedback glove*

The following is an application specific device designed for pre surgical endoscopy simulation

(Fig. 15) [56]. It is a passive device having 2-input/output d.o.f. No information is available on how the haptic loop is managed.

Figure 15: *HT Medical Systems PreOp*[TM] *Endoscopy Simulator*

Immersion Corporation laparoscopic impulse engine [57] is an application specific device for Laparoscopy (Minimally Invasive Surgery) (Fig. 16). Technical details of this device are currently not available to the author.

Figure 16: *Immersion Corporation Laparoscopic Impulse Engine*

The MouseCAT [58] is a general purpose active device targeted to be a plug-in replacement of a standard mouse; it has 2 input d.o.f. and 2 output d.o.f. (Fig. 17). The PenCAT is a similar device with one more input d.o.f. (the pressure on the pen along the vertical axis) (Fig. 17). For both, the controller manages the haptic loop and the application (the software driver) communicates at low frequency through a standard serial port.

The PHANToM "Premium" [59] is a commercial device very popular among researchers (Fig. 18). It is produced in three sizes, called 1.0, 1.5 and 3.0. It is a general purpose active device; the base version has 3 input d.o.f and 3 output d.o.f.; optionally 6 input d.o.f. versions are available and a prototype having 6 complete input/output d.o.f has been announced recently. The haptic loop is software managed: it requires a special interface card to manage communication with the host computer and it requires the application to supply data real-time at 1000 Hz.

Figure 17: *Haptic Technologies MouseCAT and PenCAT/Pro*

Figure 18: *SensAble Technologies PAHNToM line.*

Logitech force-feedback steering-wheel and joystick [60] are application-specific active devices for home entertainment (Fig. 19). The haptic technology used is actually from Immersion Corporation (I-FORCE product line). The steering wheel has 1 input/output d.o.f. and the joystick has

2 (excluding control buttons). The haptic loop is hardware managed and communication with the application occurs at a relatively low frequency through the USB interface or a standard serial port.

Figure 19: *Logitech force-feedback steering-wheel and joystick*

5. Applications

Since haptic devices were commercially available, the number of haptic-based applications has grown extremely rapidly. Among the others, we should mention a few fields where there is a great demand for haptic interfaces:

- medical;
- designing;
- psychophysics;
- applications for blind people;
- telemanipulation and micromanipulation.

5.1 Medical field

The strongest demand for haptic interfaces is probably coming from this area. Some conferences gathering technology providers and surgeons together have been organised recently [61].
Surgeons and other operators strive for haptic interfaces allowing them to do, among the other things:

- haptic rendering of volumetric data sets coming from computerised axial tomography (CAT) , nuclear magnetic resonance (NMR), ultrasound, etc.;
- training of specific procedures on virtual patients;
- pre-op simulation of the procedures to be executed on a specific patient;
- certification of students on virtual patients instead of real ones;
- teleoperation (remote surgery and/or minimally invasive surgery).

Most of these applications require a near-real simulation of soft objects (body tissues); the well-known problem is combining speed with accuracy

(simulation must be very realistic and performed real-time); people don't expect computers to be enough fast before 5-10 years.

In the while, very application-specific systems are being developed; they often consist of a physical model of the body part to be manipulated and/or of the tools to be used, with a certain number of sensors and actuators controlled by a computer.

5.2 Designing

CAD systems greatly benefit from haptic interfaces. Interacting with 3D objects using a 3D device is more intuitive and simplifies many operations (e.g. picking model features, placing parts in the space, etc.). Also being able to touch a virtual prototype reduces the need for making physical prototypes thus reducing development time and costs.
Haptic interfaces are currently being developed for some major CAD platforms and in the near future it is expected that haptic rendering will be a common option for a CAD platform.
Experimental applications are also under development for virtual hand sculpting of free-form surfaces.

5.3 Psychophysics

In the real world, the tactile perception of an object is always congruent with the visual image of that object. The haptic technology allows generating tactile stimuli that are independent from the image displayed on the screen. This makes it possible to make tests on the dominance of one of the two senses versus the other one when they are not congruent.
A number of such experiments are ongoing at different Universities.

5.4 Applications for blind people

Whether the sight dominates touch or not, when sight is missing touch becomes one of the primary interfaces between the individual and the world.
Haptic interfaces seem to be more robust than speech recognition; a "haptic mouse" has been demonstrated to enable a blind person to use MS-Windows just after a few minutes of training [58].

5.5 Telemanipulation and micromanipulation

There are different applications of telemanipulation, for example:

- dangerous environments;
- surgery (laparoscopy, minimally invasive surgery);
- micromanipulation (molecular manipulation, bio-engineering, etc.).

In most cases, having haptic feedback is very important, as it is not possible to control the manipulation accurately without "feeling" what is happening on the other side.

6. Haptic interaction with non-rigid models

Most of the applications mentioned in the previous section require the haptic rendering of both rigid and non-rigid objects. The haptic rendering of rigid objects is now a mature technology and not a research topic any more, as commercial libraries exist able to do it even on entry-level PCs [62].
The haptic rendering of non-rigid objects, instead, still has one main issue to be solved, that is the computational time required to do it.
Simulation of non-rigid materials is not an issue, per se; simulators exist able to accomplish this task satisfactorily [6] [63]. The problem is coupling the haptic rendering, which is an inherently real-time process, with non-rigid materials simulation, which is a computational demanding task. As it is well-known among haptic researchers, a simulation should run at least a few hundreds times per second to achieve satisfying results with haptic rendering, which is a very high frequency if we want to adopt complete physically-based models and simulate complex objects like body tissues.
As seen, there are two main approaches for non-rigid materials simulation, the particle based approach and the finite element approach. A complete implementation of either these approaches is definitely too slow for use in haptic rendering, at least on currently available hardware. Researchers in the haptic field are looking for alternatives to these approaches, which usually consist in partial implementations of them, less accurate but fast enough to allow the real-time haptic-rendering.
For example, at the Department of Industrial Engineering we have implemented an approximate particle-based simulator able to haptically render simple objects, which will be discussed in more details in the next section.
Several people believe that the particle-based approach is faster to compute than the finite element approach; however successful real-time implementations of the latter approach exist. For example, Berkley et al. implemented a "fast finite-elements" simulator able to simulate suturing of a wound [64]. Also, Shrinivasan suggests a method for reducing the order of magnitude of the number of elements to be used [65] to model certain classes of objects (mainly body organs).
Most of the applications developed in this area are prototypes. One exception is the suture simulator implemented by Boston Dynamics [66] that is a commercial product already in use for medical training.

In many cases the gap between the demand and current technology leads to resort to hybrid systems based on very application-specific devices which cover the inability to software simulate certain objects by providing physical (real) models of those objects. This is especially true in the medical field; a couple of examples of such devices were illustrated in a previous section.
The work we are doing is mainly focused on the base technologies for non-rigid materials simulation, but we are also looking for application-specific problems to validate the developed technologies on [67]. The technologies we are studying, or have studied in the recent past, can be roughly classified by the source of the haptic data used for rendering.

6.1 Off-line simulated data

This was the first attempt to integrate our high-fidelity simulator of non-rigid materials with haptic rendering [68]. Our system is particle-based and offers very good results in terms of fidelity of the simulation of the physical behaviour of the object, but it is too slow to use it directly for haptic rendering.
We observed that some "homogeneous" object geometry has the property to have the same behaviour in every point of them. Therefore, it makes sense to compute off-line the behaviour in one point, and just use the results run-time in whatever point happens the actual interaction between the user and the object. Examples of such object geometry are the sphere and the halfspace; the concept was validated on a sample implementation of the halfspace that gave good results (Fig. 20).

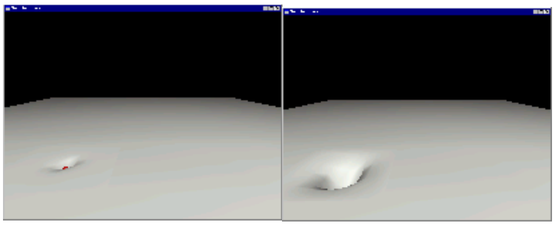

Figure 20:. *Soft Halfspace deformation simulated using pre-computed data*

This approach could be extended to other geometry under some assumptions and approximations, but most geometry couldn't be solved by this approach, especially when the deformations couldn't be thought as "local" and/or in the case of multi-finger manipulation where it would have been required to off-line pre-compute all the possible position combinations of the different fingers. So, though the results of the first experiments were very encouraging, due to the limitations intrinsic of this approach we are not developing this technology any more.

6.2 Run-time simulated data

As already stated, a simulation should run at least a few hundreds times per second to achieve satisfying results with haptic rendering; since a trade-off between the speed of a simulation and its accuracy exists, what we are currently working on is a version of the simulator which is approximated, but fast enough to supply the haptic data at the required frequency [69]. By adopting simplified physics we have obtained an "accelerated" simulator which still honours the physically-based particle model approach and exhibits the following performances:

- the computational time as a function of the input size n, where n is the number of particles, shows to be bound above by O (n) = n log (n);
- the speed-up on multi-processor computers appears to be equal, or even greater, than the number of CPUs (provided that the size of the model partition which is computed by each processor can fit in its cache memory);
- a 1000-particle model is computed at a rate of about 320 Hz on a 200 MHz Dual P-Pro PC.

Actually, the simulator can use one out of two different computational models: the classical spring-damper-mass model, and a simplified spring-only model.
The spring-damper-mass model represents each constraint as a spring-damper system, having separate constants for compression and extension; particle acceleration is derived from the forces exerted on them and their masses. The spring-only model ignores the damping values and the particle masses; updating of particle positions is based on heuristics.
The performances reported above are relative to the spring-only model; the spring-damper-mass model shows the same qualitative behaviour, but it is about three times slower.
Figure 21 shows two shots related to the real-time simulation of a soft box and a soft pipe. These examples have been developed within the Brite-Euram Project SKILL-MART.

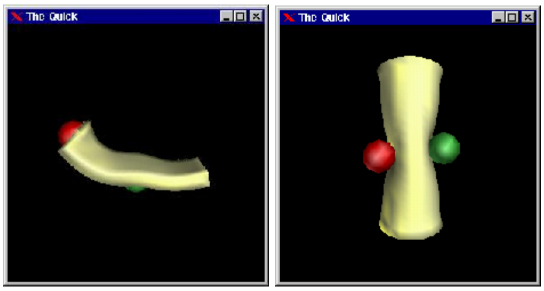

Figure 21:. *Real-time simulation of a soft box and a soft pipe*

6.3 Applications

We wish to validate on real problems the technologies we are developing; we are starting a few collaborations in various areas:

- medical
  pre-op simulation of the brain;
- psychophysics
  experiments about the dominance of sight versus the sense of touch;
- manufacturing
  Digital Mock-Ups of soft components.

7. Conclusions

An overview of techniques to model and simulate the dynamic behaviour of non-rigid objects has been presented. The physically based approach seems to be the most promising solution. In particular, particle-based systems are more flexible than approaches using continuum mechanics. Works done demonstrated that particle-based model could be interesting and effective for engineering problems currently faced and solved by finite element method.
Several efforts have been done in order to reduce computational time and face problems like collision detection and response.

Nowadays these models and related systems are mainly used for computer graphics or animation where main emphasis is on the production of images that look real. Some works, mainly for textile and clothing, have been carried out to support design and manufacturing processes.
Another relevant topic is the interaction with the non-rigid materials models. The haptic technology seems to be very promising and interesting for a variety of applications. At this regard, we can summarise that:

- haptic rendering of rigid objects is a mature technology; commercial libraries exist able to do it even on entry level PCs;
- haptic rendering of flexible objects is still an open issue; there is a strong demand for it but current technology can't satisfy it;
- the gap between demand and current technology is only partially covered by development of highly specialised systems.

References

1. *Proceedings of the Second Phantom Users Group Workshop*, edited by J.K. Salisbury and M.A. Srinivasan, October 1997.
2. *Proceedings of the Third Phantom Users Group Workshop*, edited by J.K. Salisbury and M.A. Srinivasan, December 1998.
3. *Proceedings of the First PHANToM Users Research Symposium*, Germany, May 1999.
4. *Physical Interaction: The Nuts and Bolts of Using Touch Interfaces with Computer Graphics Applications*, Course Note n.1, SIGGRAPH 1998.
5. *Haptics: From Basic Principles to Advanced Applications*, Course Note n. 38, SIGGRAPH 1999.
6. *Physically Based Modeling*, Course Note n. 13, SIGGRAPH 1998.
7. *Physically Based Modeling*, Course Note n. 36, SIGGRAPH 1999.
8. *Physically Based Modelling*, Tutorial T4, 5th Symposium on Solid Modelling and Applications, Ann Arbor, June 1999.
9. N. N.g. Hing, and L. Grimsdale, *Computer Graphics Techniques for Modeling cloth*, Computer Graphics in Textiles and apparel, IEEE Computer Graphics and Applications, vol. 16, n.5, September 1996, pp. 28-41.
10. M.P. Gascuel, and C. Puech, *Dynamic Animation of Deformable Objects*, State of the Art Reports, Eurographics 91, 1991, pp.187-207.
11. J. Weil, *The Synthesis of cloth objects*, *Computer Graphics*, vol.20, no.4, July 1987, pp. 333-334.
12. B.K. Hinds, and J. McCartney, *Interactive Garment Design*, Visual Computer, vol. 6, 1990, pp. 53-61.
13. *Development in Physically Based Modeling*, Course Note n. 27, SIGGRAPH 1988.
14. R. Barzel, *Physically-based Modelling for Computer Graphics-A Structured Approach*, Academic Press, 1992.
15. C.C. Spyrakos, *Finite Element Modeling in Engineering practice*, Algor Publishing Division, Pittsburgh, PA, USA, 1996.
16. O.C. Zienkiewicz, and R. L. Taylor, *The Finite Element Method*, 4th Edition, vol.1, Ed. Mc Graw Hill, 1991.
17. D. Terzopoulos, K. Fleisher, *Deformable Models*, The Visual Computer, vol. 4, 1988, pp.125-150.
18. D. Terzopoulos, J. Platt, A. Barr, K. Fleisher, *Elastically Deformable Models*, Computer Graphics (Proc. Siggraph), vol. 21, n. 4, July 1987, pp. 205-214.
19. D. Terzopoulos, and A. Witkin, *Physically based model with rigid and deformable components*, IEEE Computer Graphics and Applications, December 1988, pp. 41-51.
20. D. Terzopoulos, and K. Fleisher, *Modeling Inelastic Deformation: Viscoelasticity, Plasticity, Fracture*, Computer Graphics, vol. 22, n. 4, August 1988, pp.269-278.
21. J.W. Eischen, S. Deng., T.G. Clapp, *Finite-Element Modeling and Control of Flexible fabric Parts*, Computer Graphics in Textiles and apparel, IEEE Computer Graphics and Applications, vol. 16, n.5, September 1996, pp. 71-80.
22. S. Kawabata, *The Standardization and Analysis of Hand Evaluation*, The Textile Machinery Society of Japan, Osaka, 1980.
23. M. Carignan, Y. Yang, N. Magnenat Thalmann, and D. Thalmann, *Dressing animated synthetic actors with complex deformable clothes*, Computer Graphics (Proc. Siggraph), vol. 26, n. 2, July 1992, pp. 99-104.
24. P. Volino, N. M. Thalmann, S. Jianhua, and D. Thalmann, *An evolving system for simulating clothes on virtual actors*,Computer Graphics in Textiles and apparel, IEEE Computer Graphics and Applications, vol. 16, n.5, September 1996, pp. 42-51.
25. A. Witkin, *Particle System Dynamics*, Proc. 19th International Conference on Computer Graphics and Interactive Techniques, ACM Siggraph 92, Chicago, July, 1992.

26. D. H. House and D. E. Breen, *Representation of Woven Fabrics*, , Course Note n. 31, SIGGRAPH 98, B1-B17.
27. A. Luciani, S. Jimenez, J.L. Florens, C. Cadoz, and O. Roult, *Computational Physics: A modeler-simulator for Animated Physical Objects,* Proc. Eurographics 91, pp. 425-436.
28. A. Luciani, S. Jimenez, J.L. Florens, C. Cadoz, and O. Roult, *An unified view of multitude behaviour, Flexibility, Plasticity and Fractures: balls, bubbles and agglomerates,* Modeling in Computer Graphics, IFIP series on Computer Graphics, Ed. T.L. Kunii, Springer Verlag, 1991, pp. 55-74.
29. Cugini U., Dragoni P., Ippolito M., Rizzi C., *Modeling and simulation of handling machinery integrated with dynamic and static behavior of non-rigid materials***,** IEEE Robotics and Automation Magazine, March 1998, pp.48-56.
30. D. Breen, D. House, M. Wozny, *A Particle based Model for Simulating the Draping Behaviour of Woven Cloth*, Rensselaer Design and Manufacturing Institute, Rensselear Polythecnic Institute, Technical Report n. 93011, 1993.
31. D. Tonnesen, *Modeling Liquids and Solids using Thermal Particles*, Course Note n. 16, SIGGRAPH 92, pp. 4.22-4.29.
32. B. Eberhardt and A. Weber , *A Fast Flexible, Particle – System Model for Cloth Draping* , Computer Graphics in Textiles and apparel, IEEE Computer Graphics and Applications, vol. 16, n.5, September 1996, pp. 52-59.
33. P. Volino, M. Courchesne and N. Thalmann, *Versatile and Efficient Techniques for Simulating Cloth and Other Deformable Objects*, Computer Graphics, vol. 29, 1995, pp. 137-144.
34. X. Provot, *Animation realiste de vetements*, PhD Thesis, Université René Descartes, Paris V, Matematiques et Informatiques, 1997.
35. G.S.P. Miller, *The motion dynamics of snake and worms,* Computer Graphics, vol. 22, n. 4, August 1988, pp. 177-198.
36. C. R. Feynman, *Modeling the appearance of cloth,* MSc Thesis, Department of Mechanical Engineering and Computer Science, MIT, Cambridge, MA, 1986.
37. L. Gan et al., *A Finite Element Analisys of the Draping of Fabrics* , The 6th International Conference in Australia of Finite Element Methods, 1991.
38. J. D. Lambert , *Numerical Methods for Ordinary Differential System*, John Wiley & Sons, 1991.
39. P. Volino and N. M. Thalmann , *Efficient self – collision detection on smoothly discretized surface animations using geometrical shape regularity*, Computer Graphics Forum (Proc. Eurographics 94) vol. 13, n.3, Oslo, Norway, September 1994, pp. 155-166.
40. D. Terzopoulos, J. Platt, K. Fleisher, *Heating and melting deformable models,* Computer Interface 89, June 1989, pp. 219-226.
41. Frugoli C., Rizzi C., *Physically-based simulation with SoftWorld,* Proc. IASTED International Conference on Applied Modelling and Simulation, Honolulu, August 1998, pp. 373-377.
42. F. Taillefer, *Mixed modelling,* Proc. Compugraphics 1991, pp. 467-478.
43. G. Hirota, R. Maheshwari, M. C. Lin, *Fast volume preserving free-form deformation using multi-level optimisation,* Proc. , 5th Symposium on Solid Modelling and Applications, Ann Arbor, June 1999, pp. 234-245.
44. W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling. *Numerical Recipes in C,* Cambridge University Press, Cambridge England, 1988.
45. D. Baraff, I*mplicit Methods for Differential Equations*, Course Note n.13, SIGGRAPH 98, pp. E1-E5.
46. D. Baraff and A. Witkin, *Large Steps in Cloth Simulation*, Computer Graphics (Proc. Siggraph 98) pp. 43-54.
47. J. Platt, A. H. Barr, *Constraint Methods for Flexible Models*, Computer Graphics, vol. 22, n.4, 1988, pp. 279-288.
48. A. Witkin, *Interactive Dynamics*, Computer Graphics, vol. 24. Proc. Symposium on 3D Interactive Graphics, 1990, pp.11-20.
49. K. Fleisher, A. Witkin, and A. Barr, *Energy constraints on parametrized models*, Computer Graphics, vol. 21, 1987, pp. 225-232.
50. N. I. Badler and A. S. Glassner, *3D Object Modelling* , Course Note n. 12Introduction to Computer Graphics, SIGGRAPH 1998, pp. 1-14.
51. X. Provot, *Collision and self-collision handling in cloth model dedicated to design garments,* Proc. Graphics Interface 97, 1997, pp. 177-189.
52. D. Baraff , *Rigid Body Simulation*, Course Note n. 13, SIGGRAPH ’98, pp. D1-D68.
53. P. Volino and N. Thalmann, *Developing Simulation Techniques for an Interactive Clothing System*, Proc. VSMM 97, IEEE Computer Society 1997, pp. 109-118.

54. M. Moore and J. Wilhelms, *Collision detection and Response for Computer Animation*, Computer Graphics, ACM SIGGRAPH, Vol. 22, No. 4, 8/1988.
55. Virtual Technologies, Inc. http://www.virtex.com/
56. HT Medical Systems, Inc. http://www.ht.com/
57. Immersion Corporation http://www.immerse.com/
58. Haptic Technologies http://www.haptech.com/
59. Sensable Technologies, Inc. http://www.sensable.com/
60. Logitech http://www.logitech.com/
61. *Medicine Meets Virtual Reality*, edited by J.D. Westwood et al., IOS Press, 1999.
62. SensAble Technologies, Inc., Programmer's Guide, Version 2.1, *GHOST Software Developer's Toolkit.* SensAble Technologies. Manual, Cambridge 1999.
63. Denti P., Ursino M., Rizzi C., *Simulated behaviour of non rigid materials*, International Journal of CAD/CAM and Computer Graphics, vol. 10, n 1-2/1995.
64. Berkley J., Weghorst S., Gladstone H., Raugi G., Berg D., Ganter M., *Fast Finite Element Modeling for Surgical Simulation*, In Proceedings Medicine Meets Virtual Reality, 55-61, 1999.
65. De S., Srinivasan M.A., *Thin Walled Models for Haptic and Graphical Rendering of Soft Tissues in Surgical Simulations*, In Proceedings Medicine Meets Virtual Reality, 94-99, 1999.
66. O'Toole R., Playter R., Blank K., Cornelius N., Roberts W., Raiber M., *A novel virtual reality surgical trainer with force feedback: surgeon vs. medical student performance*, In Proceedings of the Second PHANToM users group workshop, December 1997.
67. Cugini U., De Angelis F., *Modeling, Simulation and Haptic Rendering of Non-Rigid Objects*, MTS, Rome, June 1999.
68. De Angelis F., Bordegoni M, Frugoli G., Rizzi C., *Realistic Haptic Rendering of Flexible Objects*, in Proceedings of The Second PHANToM User Group Workshop, October 19-22, 1997, MIT, Cambridge, MA.
69. Bordegoni M., Cugini U. and De Angelis F., *Evolution of Interaction in Physically-based Modeling*, Sixth IFIP WG 5.2 International Workshop on Geometric Modeling: Fundamentals and Applications (GEO-6) December 7-9, 1998.