Physically based modelling with particle systems

M. Amrani, M. Beuve, F. Jaillet and B. Shariat
LIRIS, FRE 2672-CNRS-University Lyon 1

In this paper, we describe a general methodology for the realistic modelling and simulation of deformable objects. The system of particles is used for this purpose. The first main contribution of this work is in the modelling field. Multi-layer particle systems are used to obtain a geometric description of an existing object. The second contribution is in the domain of the simulation. The particle systems are physically-based models that allow a realistic simulation of movements and deformations. This implies the representation of physical characteristics of existing objects, and thus the control of a wide variety of behaviours (stiffness, elasticity, control of volume variation) interesting in various areas such as medical applications, reverse-engineering, prototyping... The second contribution of this paper is to put these physical constraints into the model. Moreover, this paper proposes some techniques for the animation and the rendering of the particle systems.

1 Introduction

Computer systems are used in numerous industries to design and to create physical objects from geometric models. The inverse problem is the calculation of the geometric description of an existing physical object. In many applications like reverse engineering, rapid prototyping or inspection, the second point has become very important.

Important developments of the 3D digitising systems have permitted to achieve this goal. These systems produce an over-sampled cloud of points. In the medical area, digital imagery techniques provide Computer Tomography CT scan sections of internal and external structures of human anatomy. The calculation of the geometric description of the digitised parts is mainly based on the calculation of a surface model that approximates the whole digitised data. The most widespread way for this is an optimal triangulation. However, this method introduces a large number of triangles which may be inadequate for their manipulation or their. Therefore, some authors have proposed to fit the triangulation with a smooth surface to reduce the data and improve the visualisation (Hoppe 1994). In some cases, the data organised in parallel sections have been directly fit by a smooth surface (Park and Kim 1996; Jaillet, Shariat, and Vandorpe 2001). However, all these models have been developed only for the purpose of rigid modelling and visualisation.

If the modelling of rigid objects has been largely discussed in the literature, less works have been presented in the domain of automatic volumetric deformable model reconstruction. Some of these are based on geometrical approaches (NURBS, FFD, ...). However, one of the most important goals, when animating deformable object, is that the result have to be coherent with the reality. So, with geometric based models, this coherency depends on the animator's astuteness. In the real world, objects obey to physical laws. Thus, a heavy object is more difficult to move compared to a light one, and during deformations, many objects tend to preserve their volume.

With the increase of the calculation capabilities of computers, new simulation approaches, implementing physical principles, have been developed. One of the first physically based deformation model was proposed by Terzopoulos *et al.* (Terzopoulos 1989). Based on continuous physics, this model permitted to simulate a realistic behaviour.

An interesting approach has been proposed by Cotin et al. (Cotin, Delingette, and Ayache 1996) for a medical application, where the organ shapes are reconstructed with tetrahedra, using a simplex mesh, method, to produce deformable models for surgery simulation. Another interesting approach, based on a volume sampling of the object, was proposed by Debunne et al.(Debunne, Desbrun, Cani, and Barr 2001). To reduce the computation times, the authors use a multi-resolution mesh. Thus, at a time t, a mesh element has a fixed resolution, but different elements can have different levels of resolution, according to the accuracy needed in each part of the object. De-

formable superquadrics are also studied for non-rigid surfaces; since they can be deformed both locally and globally (Park, Metaxas, and Young 1994). Unfortunately, these methods are only adapted for the modelling of a restricted class of objects. The isopotential implicit surface is another alternative technique for the modelling of deformable objects. It is very simple and allows to model very complex shapes (Blanc and Schlick 1995). With this technique, an object is described by a skeleton and a field function. The object local deformation can easily be achieved by a local modification of the field function (Gascuel 1993). However, shapes obtained by union or blending between several implicit surfaces are not easily deformable. Some methods use articulated skeletons (Shariat and Vandorpe 2001). This allows global shape alterations. Another interesting technique for the modelling of deformable objects is the particle system. It has been used in the purpose of geometric reconstruction. In (Shimada and Gossard 1995), lines, surfaces and volumes are filled in with particles to obtain a spatial mesh, called bubble-mesh. Particle systems have also been widely used to model deformable surfaces in different applications. In (Szeliski and Tonnesen 1991), the authors propose a surface model with oriented particles, which allows joining, cutting and extending of deformable surfaces. Particle systems have also been used to model deformable volume objects. In (Lombardo and Puech 1995) and (Tonnesen 1998), authors describe a way to model deformable objects with oriented particles, and Meseure and Chaillou (Meseure and Chaillou 1997) apply the particle systems to the simulation of the dynamic behaviour of human organs. In (J. Jansson 2002) Jansson and Vergeest propose a particle systems in conceptual mechanical design. However, all these works do not permit to simulate a realistic behaviour of the deformable objects taking into account physical parameters (mass, young modulus) and geometric constraints (volume control) simultaneously. To find a solution to these requirements, in this paper, we propose a methodology based on multi layer particle systems. So, in section 2, our modelling approach is explained. Then, in section 3, we explain how the physical parameters of the object can be introduced into the model. Then in section 4 our reconstruction and animation approach for a multi-layer particle system is defined. The finally in section 5 we explain how objects modelled by particle systems can be visualized and rendered. Some results are given in section 6.

2 Particle systems and deformable object modelling

2.1 Definitions

Particle systems consist in a set of solid spheres whose movement follows physical laws. Particle systems were originally used to represent explosion, fire, cloud and any kind of objects that do not have any clear apparent bound. For deformable objects modelling, this simple model has been improved by the introduction of internal forces between particles to preserve the cohesion of the object. Tonnesen (Tonnesen 1998) applied a potential energy function (Lennard Jones potential). The corresponding forces have two terms: one short range repulsion to prevent particles to overlap and one long range attraction to ensure compactness and cohesion. Moreover, the potential energies are conservative. Therefore, the particle system may oscillate and become unstable. Thus damping forces are introduced to prevent system instability. The interaction of a particle system with its environment is represented by external forces (gravity, collisions, ...).

2.2 Interaction potentials

The behaviour of a particle system results from physical forces which manage the interactions between the particles. A first contribution of these forces consists in internal friction forces, which aim at dissipating energy avoiding a temperature increase. A second one includes the internal conservative forces, i.e. the interactions between the particles. Let us remind that for conservative forces, the relation between potentials and forces is given by:

$$\overrightarrow{f_i} = -\overrightarrow{Grad\, r_i}(\Phi(r_1, \ldots, r_k, \ldots, r_N)) \qquad (1)$$

where r_k is the position of the particle k, $\Phi(r_1, \ldots, r_k, \ldots, r_N))$ the internal potential and f_i the forces suffered by the particle i due to this potential. Internal potential is crucial and difficult to define. The choice is critical because of the following difficulties:

- the action of the interaction parameters have to be clear enough to give an idea of which parameter value have to be modified, and how, to produce a specified behaviour,
- the interaction computation time has to be as short as possible.

Physicists studying the molecular dynamics, have developed several kind of interactions to model the forces linking atoms. Even though these interactions have been developed for a microscopic scale, some of them can be used to describe the macroscopic behaviour of an object. These interactions allow us to produce physically realistic simulations.

Generally, interaction potential, for a system composed of N particles, is commonly described according to:

$$\Phi = \Phi^{(0)} + \sum_i \Phi_i^{(1)} + \sum_i \sum_j \Phi_{ij}^{(2)} + \sum_i \sum_j \sum_k \Phi_{ijk}^{(3)} + \ldots \qquad (2)$$

$\Phi^{(0)}$ and $\Phi_i^{(1)}$ are due to external conservatives forces ($\Phi^{(0)}$ represents the ground potential). $\Phi_{ij}^{(2)}$ are pair-potential contribution, $\Phi_{ijk}^{(3)}$ three-body potentials and generally speaking, $\Phi_{ij\dots}^{(k)}$ k-body potentials. A good approximation often consists in truncating the serial to the second order(pair potentials). Moreover we have studied the most commonly used potentials in physics in order to choose the most adapted one in our work context.

2.2.1 Spring potential

An approach to model an attractive/repulsive force between two masses consists in using a network of springs. Thus, when two particles get closer, it produces a repulsive force and an attractive force in the opposite direction. These forces are generated because the springs tend always to reach to their equilibrium state. The forces are computed using the following equation:

$$\overrightarrow{f} = -\left[k_r(r - r_0)\right]\frac{\vec{r}}{r} \tag{3}$$

where r_0 is the spring equilibrium distance, $r = \|\vec{r}\|$ is the distance between the two particles, k_r is the stiffness parameter. However, when using springs, an explicit neighbourhood relationship between the particles has to be defined.

2.2.2 Morse potential

Morse potential is a good approximation for molecular interactions. This potential consists in summing two parts, a Van der Walls attraction and a repulsive part, which avoid electron-cloud overlapping, called steric repulsion. It can be computed by the following equation:

$$\Phi(r) = D\left(e^{-2\mu(r-r_0)} - 2e^{-\mu(r-r_0)}\right) \tag{4}$$

At the equilibrium distance r_0, the potential reaches its minimal value. D represents the binding energy, i.e. the minimum energy to provide to a two-particle system in order to obtain a definitive separation. Changing the value of μ, changes the stiffness of the potential. One has also to note, that a modification of μ generates changes on the two parts of the potential, making the behaviour control difficult.

2.2.3 Lennard-Jones potential

Lennard-Jones potential is often used to model atom interactions. In a general way, this potential is given by the equation:

$$\Phi(r) = \frac{\epsilon}{m-n}\left(m\left(\frac{r_0}{r}\right)^n - n\left(\frac{r_0}{r}\right)^m\right) \tag{5}$$

If the equilibrium distance is set to r_0 and the potential value at r_0 to $\Phi(r_0) = \epsilon$. The force derivating from this potential is:

$$\overrightarrow{f}(r) = \frac{-mn\epsilon}{(n-m)r_0}\left(\left(\frac{r_0}{r}\right)^{n+1} - \left(\frac{r_0}{r}\right)^{m+1}\right)\frac{\vec{r}}{r} \tag{6}$$

In this equation, it is clear that n controls the repulsive part of the potential ($r < r_0$), when m controls the other part. ϵ is the binding energy. Numerical cost is very low for integer values of n and m. Nevertheless, the comutation time increases for non integer values of n and m.

2.2.4 Other potential functions

Some other potentials exists, but they are less interesting. However, we propose a brief description of two of them:

- Murrell potential

$$\Phi(r) = -D\left(1 + \sum_{i=1}^{3} a_i(r - r_0)^i\right)e^{-\mu(r-r_0)} \tag{7}$$

  The polynomial function, used for this potential, produces many unstable equilibrium states, corresponding to points of local minimal potential energy, making the simulation process more difficult.

- The Boscovich potential has many stable equilibrium states. Thus, when using this kind of interaction, the first deformations are elastic, but when reaching a new equilibrium state the deformation become plastic.

2.2.5 Friction

A damping force can also be added to the attractive/repulsive force between two particles. This damping component models the friction between the particles. Thus, when considering two particles, this force is given by the equation:

$$\vec{f_d} = -k_d\frac{\vec{v}.\vec{r}}{r}\frac{\vec{r}}{r} \tag{8}$$

where k_d is the friction parameter, $\vec{r}$ and $\vec{v}$ respectively the relative position and velocity between the particles.

2.3 Choosing the interaction potential

Without definitely excluding the other potentials, we focused on LJ (Lennard-Jones) potential. The action

of each parameter on the physical behaviour of the object is more clearly defined than for the other potentials. Moreover, in the case of integer exponents, the computation time of this potential is obviously shorter than any of the other potentials. This is crucial if the object is simulated with a large number of particles.

3 Lennard Jones parameters

When used in molecular dynamic, Lennard-Jones parameters have a precise meaning. The values are defined for each kind of atom and often stored in some physics tables. However, in our case, we have to define the parameter values according to the macroscopical properties of the real object we want to simulate. The choice must also take account for computing performances. In LJ-presentation section, we listed the four parameters to be set: ϵ, r_0, n and m. These quantities are not very direct and then not easy to set. So we need to arbitrarily define four new independent quantities expressed as a function of these parameters and closer to our preoccupations. For the sake of simplicity, we limited our present work to a system of identical particles and proposed the following quantities:

1. The number of particles to describe the object influences the accuracy of the simulation. As we shall see later, it is closely related to r_0.
2. The binding energy is the minimum energy to provide to the object in order to extract a particle. This value has to be set large enough to avoid any fragmentation of the object.
3. The cutoff distance of the interaction force describes the range of the interaction. Beyond this range, the force may be set to zero. A large cutoff distance means a large computing time within cell-linked algorithm.
4. One other interesting quantity is the Young modulus E of the object. Young modulus is an intensive quantity, which characterises the elasticity of a given, real, material.

Now, we have to establish a relation between these new quantities and the Lennard-Jones parameters.

3.1 Number of particles

The equilibrium radius r_0 is the distance at which Lennard-Jones force is null. For two particles of radius r_1 and r_2, this distance is, in general, equal to $r_0 = r_1 + r_2$, making the two particles in contact at the equilibrium state. It is also possible to take $r_0 < r_1 + r_2$, allowing the particles to overlap, or $r_0 > r_1 + r_2$ making the particles fill a largest volume. Finally, setting the equilibrium value r_0 permits to define the typical distance between two particles. According to the structure of the particle set (amorphous, fcc....), the occupied volume may be different. To conclude, for a fixed structure, the number of particles is directly controlled by the value of r_0. r_0 is therefore an appropriate parameter.

3.2 Binding energy

In principle, one can define two binding energies:

- a bulk binding energy, which corresponds to the extraction of a bulk particle;
- a surface binding energy, when the particle initially stands at the surface of the object.

These two quantities are, in fact, strongly related and, even if they should depends on the local structure, they are shown to be close to the minimum $\phi(r_0)$ of the two-particles potential, i.e. ϵ for Lennard-Jones potential. Finally, we shall keep $\phi(r_0)$ parameter to control the breakdown threshold.

3.3 Cutoff distance

First of all, we need to choose a criterion to properly define the cutoff distance r_c. In principle, this is the distance at which the interaction force can be neglected. So we proposed the following criterion:

$$F(r_c) = F_{min} * \eta_c \tag{9}$$

where η_c is small compared to 1 and F_{min} is the minimal value of $F(r)$, i.e. the value of the most attractive force ($F_{min} < 0$). F_{min} is obtained for a radius $r_{f_{min}}$:

$$r_{f_{min}} = r_0 \left(\frac{m+1}{n+1} \right)^{\frac{1}{m-n}} \tag{10}$$

Within a good approximation, it reads:

$$r_c \approx r_0 \left(\frac{1}{\eta_c (p-1)} p^{\frac{p}{p-1}} \right)^{\frac{1}{n+1}} \tag{11}$$

where $p = \frac{m+1}{n+1}$

3.4 Young modulus

The Young modulus of an object is a complex function of the interaction potential and of the structure of the particle system. It cannot possibly be expressed analytically and need therefore to be computed. At least, it remains easy to derive a two-particles Young modulus E_{LJ} for the Lennard-Jones potential. When two particles are bound by a spring, the Young modulus of the spring (E_{SP}) is related to its stiffness value (Tonnesen 1998):

$$k = \frac{E_{SP}}{r_0} \tag{12}$$

For Lennard-Jones interaction, we can see that, around the equilibrium distance r_0, the force can be approximated by a spring force. Writing Taylor expansion around r_0 gives:

$$F(r_0 + \delta r) = F(r_0) + F'(r_0)\delta r \quad (13)$$

Since $F(r_0) = 0$, the previous equation becomes:

$$F(r_0 + \delta r) = F'(r_0)(r - r_0) \quad (14)$$

We can consider $F'(r_0)$ as a stiffness parameter, deducing the Lennard-Jones force Young modulus (E_{LJ}):

$$E_{LJ} = r_0 F'(r_0) \quad (15)$$

By evaluating $F'(r_0)$ we finally have:

$$E_{LJ} = mn\epsilon \quad (16)$$

4 Object reconstruction and simulation

Our goal is to obtain a sampling of a volume defined by a closed surface. This surface can be of any type and any topology. The principle of our method that we have presented in (Jaillet, Shariat, and Vandorpe 1998) is to fill the inner volume with a set of particles. The reconstruction process can be divided in several steps. The first one is the initialization of particles inside the boundaries. They will act as seeds to create new particles that will progressively fill the whole object. Particles evolve to a stable state which is a minimal energy disposition.

Thus, we obtain easily a regular spatial sampling which corresponds to a maximal filling. The following algorithm summarizes our reconstruction method.

> Initial particles creation.
>
> **repeat** for all particles
>
> 1. collisions detection with the object boundaries.
> 2. internal and external forces computation.
> 3. computing velocities using physical laws.
> 4. let particles evolve to a stable state.
> 5. create new particles around the existing ones.
>
> **until** object volume is full by particles.

If we want to increase the precision of the resulting model, we need to decrease the radius of the particles that will dramatically increase the number of particles. To solve this problem, we have developed a multi-layer model. The key idea is to place small particles were details are required and big ones elsewhere:

- the centre is composed of particles of great radius. They are the kernel of the object.
- the centre is surrounded by one or more layers with decreasing radius as we get closer to the boundaries.

The coherency of each layer is preserved by virtual separations between layers. Then, the reconstruction process can be repeated for each layer (Jaillet, Shariat, and Vandorpe 1998), as shown on figure 1.

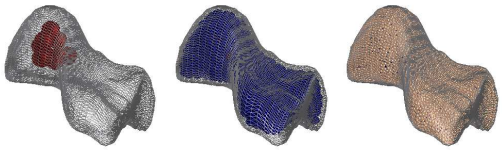


Figure 1: A 2D example of the reconstruction of an object with a multi-layer particle system

4.1 Animation and deformation of particle systems

Particle systems are well adapted for the animation of deformable objects subject to physical forces (gravity, collision with obstacles, ...).

As stated before, it is possible to handle a wide variety of behaviours, simply by changing force parameters. The objects can undergo very large deformations and change their topology. The animation of multi-layer particle systems and the respect of their volume constraints have necessitated the following considerations:

4.1.1 Dynamic layer structure

During a deformation, the big particles of the kernel could become to close to the object surface, altering the layer structure of the model. This modifies the physical properties of the object, and make the deformations incoherent (see figure 2).

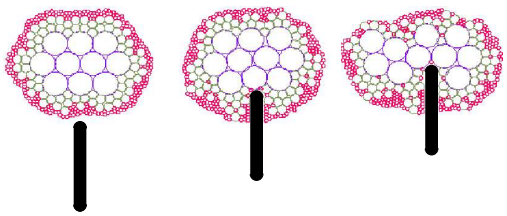


Figure 2: Layer structure alteration during the cutting of an object

To prevent this to occur, we have developed an adaptive deformation method. The idea is to subdivide big particles into smaller ones when they are

in contact with the external world. This permits to maintain the coherency of the multi-layer organisation. While subdividing, it is important not to introduce perturbation in the system: volume and energy must be preserved. This is achieved by replacing a particle P of radius R, mass M and velocity V by n particles of radius $R' = \frac{R}{n^{\frac{1}{D}}}$ (D is the dimension of the space, 2 or 3). For a minimal volume loss, we use the face-centred cubic (or hexagonal in 2D) lattice which is the most compact packing of spheres. With regards to the momentum and kinetic energy preservation, each new particle must have the same velocity V as the subdivided one, and a mass m equal to $\frac{M}{n}$:

$$\sum_1^n (mV) = \sum_1^n (\frac{M}{n}V) = MV \qquad (momentum)$$

$$\sum_1^n (\frac{1}{2}mV^2) = \sum_1^n (\frac{1}{2}\frac{M}{n}V^2) = \frac{1}{2}MV^2 \quad (kinetic\ energy)$$

The evolution of a bi-dimensional object cut by a tool is presented in figure 3. Note the large improvements in the shape preservation compared to figure 2.

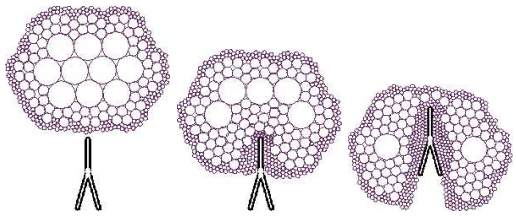


Figure 3: Evolution of the cutting of an object with layer structure preservation

4.1.2 Volume constrained deformation

One of the important points that should be handled when modelling deformable objects is the control of the object volume during the deformation. In some applications, we want to preserve volume during the deformation process. Consequently, we should be able to control the volume changes. Particle systems are really adapted to handle this constraint. Indeed, since this model is a volume sampling of the initial object, it gives an immediate approximation of its volume. During the deformation, particles tend to reach their most stable state, which corresponds to their equilibrium distances. In this case, the particles try to keep their respective distances in a manner to minimize the whole system energy. This avoid the creation of holes in the system. Thus, the object volume is naturally preserved.

On the contrary, in other applications we may ask for a change of object volume. For this, we can modify the equilibrium distance from r_0 to r'_0, its volume is consequently modified, and therefore the volume of the whole object changes. Of course, this approach can be used for decreasing or increasing the volume of the object.

5 Particle systems visualisation

If, the major advantage of using the particle system model, is its physical realism to produce simulations, its visualisation (by simply displaying a sphere for each particle) remains inaesthetic. Such a display generates a feeling of unrealism, even when the behaviour sticks to reality. A solution to this problem was proposed by Desbrun (Desbrun 1997). It consists in clothing the particle system by an implicit surface, which also acts as an interface handling interactions with the surrounding objects. But, in our model, this task is already taken by the external particles.

So, rather than using an implicit surface, we propose to compute a triangular mesh directly from the particles. The key idea is to extract, from the system, the external particles that defines the object surface. Then we compute a mesh from the centre points of these particles. Finally, we move each node of the computed mesh for distance equals to the particles radius, according to the direction given by the normal vector at this point (the average of the normal vectors to the triangles including this point).

5.1 Finding the external particles

The problem is to find, from the particles of the skin layer of the system, those which define the object boundaries. To do that, we have developed a new method, inspired by the marching cube algorithm (Bloomenthal 1994). Our method is based on the following three steps:

1. splitting the particle system bounding box into voxels,
2. finding the voxels intersecting the object surface. We propose a two-pass method:
   - in a first pass, we mark each voxel intersecting a particle. The non-marked voxels are outside the object and are removed;
   - in the second pass, we remove the voxels having all their neighbours (8 in 2D and 26 in 3D) marked.

   At the end of this process, the remaining voxels are those on the object boundaries.
3. deducing the particles of the object boundary, which are simply those intersecting these remaining voxels.

5.2 Computing a triangular mesh

In the previous step, the particles defining the object boundaries have been detected. We now want to compute a triangular mesh of the set points defined by the centre points of these particles. A lot of triangulation algorithms have been defined and could be used

for our problem. However, our set of points is results from a compact set of spheres. Thus, each sphere generates a triangle with two adjoining particles, which are also adjoining between them.

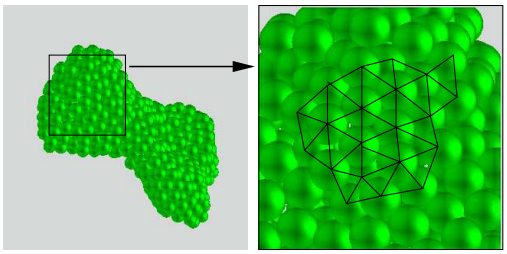
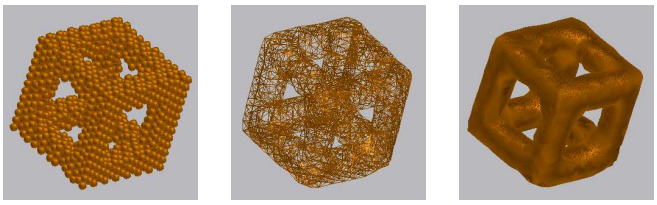


Figure 4: Meshing a particle system surface



(a) Displaying spheres (b) Meshing (c) Mesh with rendering

Figure 5: Particle system visualisation

Considering each set of three adjoining particles, we compute, triangle by triangle, a mesh covering all the object surface.

When the mesh is complete, and to ensure that all the particles will be inside the volume bounded by this mesh, a dilation is applied to the mesh.

The figure 5 shows an example of our visualisation approach. The object is a cube with a hole on each face. So, this example probes that our visualisation method can handle complex shapes.

6 Example

Figure 6 demonstrates the simulation of a very deformable object falling down because of the gravity. The collision deforms totaly the object. Figure 7shows the same object with an elastic behaviour. The collisions deforms partially the object. However, when the contact is removed the object goes back to its initial state.

7 Conclusion

In this paper we have proposed a 3D reconstruction methodology permitting to take into account the physical behaviour of the reconstructed object. Moreover, during the deformation, the volume of the object is controlled.

For the reconstruction, a method based on multilayer particle systems using Lennard-Jones potential

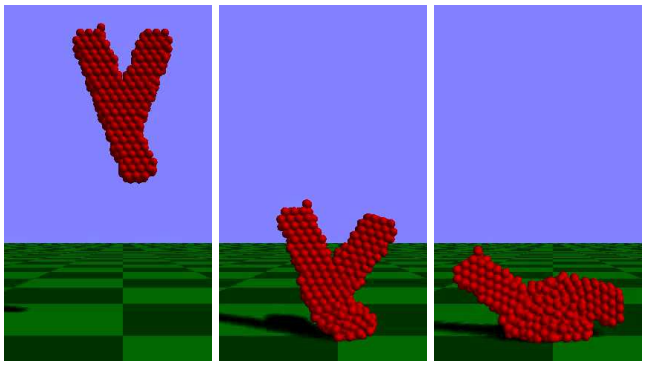


Figure 6: Simulation of the behaviour of a very deformable object

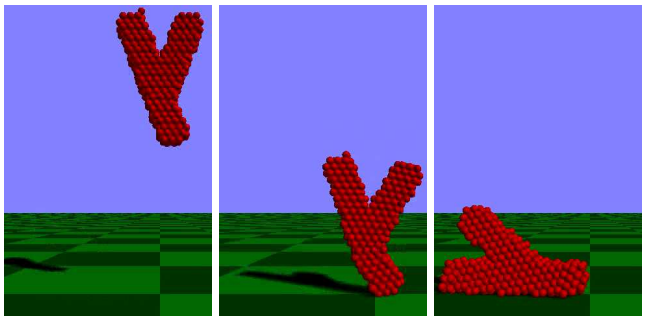


Figure 7: Simulation of the behaviour of a less deformable object

function has been introduced. The reconstructed objects can be of arbitrary topology. The Lennard-Jones parameters are computed explicitly, using the physical quantities (elasticity, precision, ...). Moreover, we have proposed a visualisation algorithm to render objects modelled by particle systems.

In the context of the european BIOMED2-ARROW project, the proposed modelling technique has been used to model and simulate human organs interactions to enhance the quality of the radiotherapy.

REFERENCES

Blanc, C. and C. Schlick (1995, April). Extended field functions for soft objects. In *Implicit Surfaces '95*, Grenoble, France, pp. 21–32.

Bloomenthal, J. (1994). An implicit surface polygonizer. In P. Heckbert (Ed.), *Graphics Gems IV*, pp. 324–349. Boston: Academic Press.

Cotin, S., H. Delingette, and N. Ayache (1996, January). Volumetric deformable models for surgery simulation of non-rigid organs. In 4^e *séminaire du groupe de travail "Animation et Simulation"*. GDR-PRC AMI (Algorithme, Modélisation et Infographie), Strasbourg.

Debunne, G., M. Desbrun, M.-P. Cani, and A. H. Barr (2001, August). Dynamic real-time deformations using space and time adaptive sampling. In *Computer Graphics Proceedings*, Annual Conference Series. ACM Press / ACM SIGGRAPH. Proceedings of SIGGRAPH'01.

Desbrun, M. (1997, December). *Modélisation et Animation de Matériaux Hautement Déformables en Synthèse d'Images*. Ph. D. thesis, INP Genoble.

Gascuel, M.-P. (1993, August). An implicit formulation for precise contact modeling between flexible solids. *Computer Graphics Proceedings 27*, 313–320.

Hoppe, H. (1994). *Surface reconstruction from unorganized points*. Ph. D. thesis, University of Washington.

J. Jansson, J. V. (2002). A discrete machanice model for deformable bodies. *Computer Aided Design 34*, 913–928.

Jaillet, F., B. Shariat, and D. Vandorpe (1998, March). Deformable object reconstruction with particle systems. *Computers & Graphics 22*(2-3), 189–194.

Jaillet, F., B. Shariat, and D. Vandorpe (2001). Smooth reconstruction of branching objects with b splines. *Revue de CFAO et d'Infographie 16*, 183–201.

Lombardo, J.-C. and C. Puech (1995, May). Oriented particles: A tool for shape memory objects modeling. In *Graphics Interface'95, Quebec City (CAN)*.

Meseure, P. and C. Chaillou (1997, February). Deformable body simulation with adaptive subdivision and cuttings. In 5^{th} *Int. Conf. in Central Europe on Comp. Graphics and Visualization WSCG'97, Plzen, CZ*, pp. 361–370. IFIP working group 5.10 on Computer Graphics and Virtual Worlds.

Park, H. and K. Kim (1996, December). Smooth surface approximation to serial cross-sections. *Computer Aided Design 28*(12), 995–1005.

Park, J., D. Metaxas, and A. Young (1994, June). Deformable models with parameter functions: application to heart-wall modeling. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), Seattle, WA*, pp. 437–442.

Shariat, B. and D. Vandorpe (2001, july). 3d reconstruction with implicit surfaces. In *8th ISPE International Conference on Concurrent Engineering: Research and Applications-CE2001*, pp. 137–144.

Shimada, K. and D. C. Gossard (1995, May). Bubble mesh: automated triangular meshing of non-manifold geometry by sphere packing. In *Third Symposium on Solid Modeling and Applications, Salt Lake City, Utah*, pp. 409–419.

Szeliski, R. and D. Tonnesen (1991, December). Surface modeling with oriented particle systems. Technical Report CRL 91/14, DEC, Cambridge Research Lab.

Terzopoulos, D. (1989). Heating and melting deformable models (from goop to glop). In *Graphics Interface '89*, pp. 219–226.

Tonnesen, D. (1998). *Dynamically coupled rticle systems for geometric modelling, reconstruction and animation*. Ph. D. thesis, Graduate department of computer science, University of Toronto, Toronto.