Simulation of deformable organs with a hybrid approach

Morade Amrani— Fabrice Jaillet— Mahmoud Melkemi— Behzad Shariat

LIGIM, Bat. 710 Université Lyon I, 43 bd du 11/11/1918, F-69622 Villeurbanne
{m-amra97,fjaillet, melkemi, bshariat}@ligim.univ-lyon1.fr

ABSTRACT. *In this paper, we describe two complementary methodologies for the reconstruction and animation of volumetric deformable objects from planar contours. The reconstruction is based on two different modelling techniques: implicit surfaces and particle systems. Although the implicit surfaces allow to easily model complex shapes, their deformation is troublesome. On the contrary, the reconstruction process based on particle systems requires an important computation time. Nevertheless, they permit to model different behaviours from rigid to fluid state. In the scope of a medical application, we want to simulate the motion and the form alteration of the internal anatomical shapes. An acceptable computation time with realistic simulation results requires to use a hybrid approach: implicit surfaces for organs that admit small deformations and particle systems for those subject to large deformations. In this paper, we have introduced the necessary tools to handle the reconstruction of different organs as well as their interaction within a hybrid approach.*

RÉSUMÉ. *Dans cet article, nous décrivons deux méthodes complémentaires pour reconstruire et animer des objets déformables volumiques, décrits par des contours planaires. La reconstruction est basée sur deux techniques de modélisation différentes : les surfaces implicites et les systèmes de particules. Les surfaces implicites permettent de modéliser des formes complexes, mais leur déformation est difficile. En revanche, la reconstruction avec des particules nécessite des calculs coûteux, mais elle permet de représenter de nombreux comportements, allant de l'état rigide à l'état liquide. Dans le milieu médical, nous voulons simuler le mouvement et la modification des formes anatomiques internes. La recherche d'un temps de calcul acceptable et de simulations réalistes nous a amenés à utiliser une approche hybride : utilisation des surfaces implicites pour les organes subissant des petites déformations et emploi des systèmes de particules pour les objets soumis à des déformations globales. Dans cet article, nous avons introduit les outils nécessaires à la reconstruction de différents organes ainsi qu'à leur interaction dans un modèle hybride.*

KEYWORDS: *physically based modelling, modelling and deformation of human organs, particle systems, implicit surfaces.*

MOTS-CLÉS : *modélisation basée sur la physique, modélisation et déformation d'organes humains, systèmes de particules, surfaces implicites.*

1. Introduction

Conformal Radiotherapy consists in administering a lethal dose of ionising radiation to cancerous regions causing a minimum of damage to surrounding healthy tissues. To achieve this goal, multi-leave collimators are used. They permit to have a precise control on the shape of the beam. In this technique the key problem is to choose the correct number of X-ray beams and their different shooting angles and intensities. Classically, to determine the therapy's ballistics, radiotherapy simulators and treatment planning systems are employed. However in all existing techniques, the patient is supposed to have a solid and static shape.

However, both the internal and external forms are dynamic and deformable, subject to internal and external forces, especially due to the natural rhythmical motions such as respiration. This can lead to an insufficient exposure of cancerous organs, and a useless or even a dangerous irradiation of healthy tissues. The use of physical devices fabricated to prevent the patient's movement, or the use of laser or stereo vision systems designed to survey the patient's external movements have shown their limits.

An alternative solution to this problem is to create geometrical models of the patient's body shape (external) and the treatment volumes (internal) and to correlate these two models in the same reference system. The animation of these models by considering patients' external and internal movements (detected by online external and internal sensors) as well as a good comprehension of human's anatomy and organs' degree of freedom can give a valuable information to evaluate the treatment error.

The choice of an appropriate geometrical model is critical. The chosen model should fulfill the requirements of this application :

– capacity to model complex shapes,

– possibility to simulate organs' shape alteration based on physical parameters.

Implicit surfaces and physically-based particle systems seem to be serious candidates for this application. Implicit surfaces can model very complex shapes at low computation cost and permit to take easily into account small deformations. Particle systems are powerful techniques to model plastic, elastic, viscoelastic and fracture behaviours. In this model it is possible to introduce several kinds of geometrical constraints such as volume conservation to reproduce the behaviour of cancerous tissues and organs. However they require important computation time.

The key idea is to use both representations to model human organs (hybrid approach). Particle systems could be used to model organs that are subject to important deformations and implicit surface for the others. This allows us to simulate interactions between organs in a body.

In this paper, after a short state of the art, in chapters 3 and 4, we present our methodology for automatic reconstruction and animation of organs, using implicit surfaces and particle systems. In chapter 5, we introduce our hybrid modelling approach.

2. Related works

In the area of modelling and simulation of deformable bodies many research works have been carried out. The most common way for elastic materials is to use finite difference or finite element techniques by approximating the body with a mesh of nodes of fixed topology [TER 88]. However, these techniques are not well adapted for the animation of materials able to accept large inelastic deformations.

In another approach, the shape of an unknown object is calculated by the deformation of a primitive to fit the data points. Pentland [PEN 86] and Bajcsy [BAJ 87] were the first authors who employed this technique for the reconstruction.

Later, the deformable superquadrics have been introduced by Terzopoulos [TER 91]. The reconstruction method proposed by Terzopoulos consists in applying some external forces to deform the superquadrics. The Bajcsy's and Terzopoulos' methods have been improved by several works [LEO 94, PAR 94, BAR 95]. Similarly, a methodology of 2D contour reconstruction using snakes [DER 92] and another using hyperquadrics [COH 94] have been proposed. Nevertheless, these methods are able to reconstruct shapes that are homeomorph to the chosen basic primitives.

An interesting approach is proposed by Cotin et al.[COT 96], where the organ is first reconstructed with tetrahedra, using a simplex mesh method, to produce deformable models for surgery simulation.

However, these methods suffer from the same handicaps that have been mentioned for the first ones.

Another class of reconstruction methodologies uses implicit functions. They allow to extract an iso-surface either by a procedural method, or skeletal implicit surfaces (surfaces generated by a field function and a skeleton).

The main problem of the methods of the first category is to find a field function that is only defined in the vicinity of the surface to be reconstructed. Moreover there should exist a potential for which the corresponding iso-surface passes through all the data points.

In [WAL 91], a method is proposed to reconstruct objects described by cross-sectional images, in which the pixels' value corresponds to the X-ray attenuation. This method is automatic and allows to reconstruct objects with complex shapes. However, it is difficult to reconstruct the objects for which the boundary does not correspond to an intensity change.

In [SAV 95], the model of an object, described by scattered points, is computed in two steps. First, the object is roughly approximated by a simple implicit object defined by a function f_c. Then, the value of f_c for each data point is interpolated by a spline volume U. The object's surface is given by the zero set of algebraic difference $f_c - U$.

There are also some research works concerning the skeletal implicit surface reconstruction. Most of them work with point skeletons, which generate spheres, and use an energy minimisation process.

Muraki was the first to employ skeletal implicit surfaces to reconstruct objects described by scattered data. In [MUR 91], Muraki proposes an automatic method to model the objects with a blended union of blobs. The reconstruction process converges extremely slowly (1 or 2 days of calculation).

Tsingos et al. [TSI 95] have improved Muraki's method. An object is modelled by a blended union of blobs. But, in order to reduce the computation complexity, a blob is defined by a local field function.

A non-iterative method has been introduced, in [FER 96] to reconstruct branching objects from a cloud of points. The skeleton is thread-like, composed of several B-spline curves which are surrounded with a non-uniform field. Each curve generates an implicit surface and the object is modelled with a blended implicit surface.

With this model, local deformation can easily be achieved by a local modification of the field function [GAS 93], and some methods use articulated skeletons allowing global shape alterations.

Despite of their very interesting capabilities, implicit surfaces are not very well adapted to large deformations and to non isotropic behaviour during a deformation.

Another interesting technique for the modelling of deformable objects is the particle system. They were first introduced by Reeves [REE 83], who used non-interacting particles for the modelling of physical phenomena and fuzzy objects like explosions, clouds and smoke. It has been also used in the purpose of geometric reconstruction. In [SHI 95], 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 [SZE 91], 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 [LOM 95], Lombardo and Puech describe a way to model deformable objects, and particularly muscles, with oriented particles and implicit surfaces, and Meseure and Chaillou apply the particle systems to the simulation of the dynamic behaviour of human organs [MES 97].

In the next sections we show how we used the last two models, implicit surfaces and particle systems, for the reconstruction and simulation of deformable organs.

3. Implicit surfaces

An implicit or an iso-potential surface $\mathcal{S}$ can be defined as the set of points p in 3D space that respect the following relationship :

$$\mathcal{S} = \{p \in \mathbb{R}^3 / f(p) = iso\}$$

The function $f(p)$ can be defined by a skeleton and a field function. The skeleton is composed of several geometric elements, namely skeletal elements, for which a distance is defined. The field function is a decreasing function depending on the distance between the skeletal elements and the points. It is used to assign a potential $f(p)$ to the points p. Generally, local potential functions $f_i(p)$ are associated with each skeletal element. Then the potential function is a blending of these local functions. The most used blendings are:

– The **Max** function : $f(p) = MAX_i f_i(p)$

– and the **Sum** function : $f(p) = \sum_i f_i(p)$

The simplest implicit model is defined by a uniform field function. In this case, the same field function is applied to all the skeletal elements and the iso-potential surface is the uniform offset of the skeleton.

3.1. *Our reconstruction methodology with implicit surfaces*

For the reconstruction, we are supplied with a set of points organised in parallel sections, that are generally obtained from the segmentation of CT scans. Initially, our approach requires the computation of a complex 3D skeleton of the object to be reconstructed. For this purpose, we propose to calculate a 3D skeleton based on Voronoi diagram concept. However, a direct calculation of a 3D Voronoi diagram is computationally expensive. To reduce the computation's cost, we firstly calculate the 2D Voronoi diagram of each section permitting to generate the 2D skeleton of each section [BOI 92]. The 3D skeleton is deduced from the 2D skeletons.

These 2D skeletons may contain too many branches. This is due to the sensitivity of Voronoï diagrams to noise and to small perturbations on the section's contour. These branches increase the complexity of the model. To simplify the skeletons, we have to suppress unnecessary branches by the deletion of the corresponding delaunay triangles. These are the triangles of the contour that have an area smaller than a threshold, predefined according to a precision criteria (see figure 1)[ATT 95, PON 00]. The computation of a 3D skeleton consists in establishing a correspondence between each pair of neighbouring 2D Voronoï diagrams. Therefore, the result is a 3D skeleton composed of triangles whose vertices are the nodes of the 2D Voronoï diagrams.

To explain the process of the 3D skeleton computation, let us consider two Voronoï skeletons of two neighbouring sections and define some terms (Figure 2):

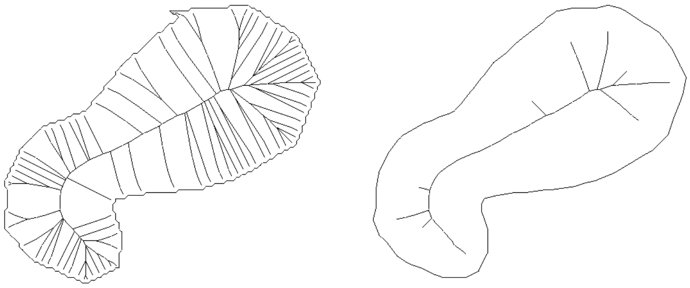

Figure 1. *2D skeleton : Non simplified 2D skeleton and simplified 2D skeleton of a 2D contour*

– A branching point is a 2D Voronoï vertex belonging to at least three 2D Voronoï skeleton branches. P_3*,* P_5 *and* Q_3 *are branching points.*

– An extreme point is a 2D Voronoï belonging to one skeleton branch. P_1*,* P_2*,* P_6*,* P_7*,* Q_1*,* Q_2 *and* Q_4 *are extreme points.*

– A corresponding point is a 2D Voronoï vertex which is in correspondence with at least one branching point or extreme point of a neighbouring skeleton. For example, Q_2 *is a corresponding point of* P_2*.*

– A corresponding branch is a skeleton branch in correspondence with a branch of a neighbouring skeleton. The branch $[Q_1; Q_3]$ *is a corresponding branch of the branch* $[P_1; P_3]$*.*

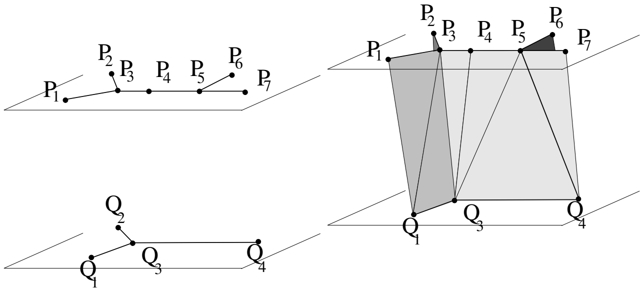

Figure 2. *2D skeleton of two neighbouring sections and the deduced 3D skeleton*

The 3D skeleton calculation process is done in three steps (Figure 2):

1) *Correspondence establishment for the branching points and the extreme points.* This is an automatic process based on the 3D Voronoï diagram of a cloud of data points organised in parallel sections. The correspondences of a point are the 2D Voronoï

vertices of the neighbouring sections to which the point is linked by paths composed of 3D Voronoï edges. For example in Figure 2, P_3 is linked to Q_1 and Q_3. To simplify the process, the branching points (respectively the extreme points) are in correspondence only with branching points (respectively extreme points).

2) *Correspondence establishment for two paths having their beginning and ending points in correspondence*. For example, the path (P_3, P_7) is in correspondence with the path (Q_3, Q_4) because P_3 and Q_3 as well as P_7 and Q_4 are in correspondence.

3) *Correspondence establishment for other branches*. At this stage, the processed branches belong to the paths linking an extreme point without correspondence to a branching point (P_B) belonging to at least one branch in correspondence. Either a correspondence is found in a set of candidate paths (paths containing one extreme branch without correspondence and one of the corresponding points of P_B) or no correspondence is found and the path is in correspondence with the nearest corresponding point of P_B. For example, in figure 2 the branch $[P_6, P_5]$ is in correspondence with Q_4.

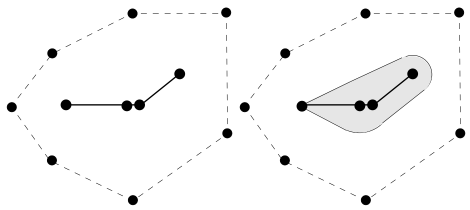

Figure 3. *2D Voronoï skeleton and 2D weighted skeleton of a section*

Finally, we have to associate a uniform potential function to this skeleton to generate the surface. However, the application of a uniform field function generates a surface that is a uniform offset of the skeleton. Therefore, the reconstructed surface will not interpolate the 2D contours. To avoid this problem, a weighting function is associated with the skeleton. In order to do this, the weight of all the Voronoï vertices of the 2D skeletons is computed first according to their distance to the contour (Figure 3). The 3D skeleton is composed of triangles whose vertices are 2D Voronoï vertices of the neighbouring sections. Thus, the vertices of different triangles of the 3D skeleton are already weighted. Weights of other skeleton points are obtained by an affine combination of triangle vertices weights [PON 00].

3.2. *Animation and deformation of implicit surfaces*

For a realistic simulation of the modelled objects, we use physically based animation techniques. This solution is based on the geometrical structure of the proposed reconstruction technique : an object is defined by a skeleton composed of triangles and a uniform field function. In our approach, for a deformable object, the vertices and the edges of skeleton's triangles are respectively transformed into masses and springs. In this method two levels of modelling are used, external (surface) and internal. When a collision occurs on a surface point, the reaction force is distributed on the skeleton's triangle that has contributed to the definition of this point.

The general process of the simulation is defined by successive iterations. Each iteration represents the transition of the object's state during the time dt. Figure 4 illustrates the evolution of two colliding objects (one rigid and one deformable) in the four following steps :

1) rigid movement of the object according to the applied forces to the skeleton (gravity for example),

2) collision detection,

3) deformation of the surface to approximate the exact contact surface,

4) deformation of the skeleton through the propagation of the constraints applied to the object's surface and the computation of the new shape.

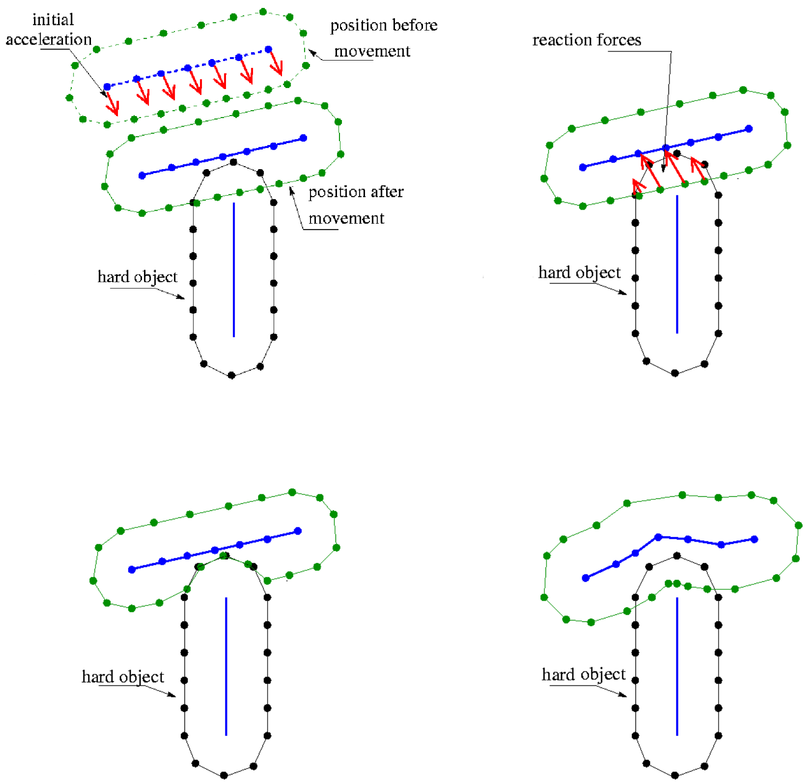

Figure 4. *Four steps of the animation*

The rigid displacement of the object consists in using the Newtonian laws of mechanics. For the collision detection of two implicit surfaces, we use a sampling method [DES 95a]. Therefore, we first sample the two surfaces. Then, using the potential functions, we test if sampled points of one surface are included in the other surface. When such points exist, a collision has occurred.

In case of collision, the surface of the object should be deformed (displacement of each sampled point P belonging to the contact zone) by applying a reaction force $\vec{F}_{R_P}$. The amplitude of this force depends on the degree of penetration of P in the other colliding object and its direction is obtained from the direction of the velocity of P when the collision occurs [PON 00].

It is possible to take into account the friction forces $\vec{F}_{f_P}$ on each point P (see [PON 00]). In this case the applied force to P is: $\vec{F_P} = \vec{F}_{R_P} + \vec{F}_{f_P}$.

Once the reaction force is computed, the contact surface can be calculated. In our approach, each point P is supposed to be linked to its corresponding point on the skeleton by a spring. This spring reacts to the force $\vec{F_P}$: in presence of this force the surface is deformed and when this constraint is removed the surface comes back to its initial state (for elastic objects).

Specially in case of elastic object, for this spring, two types of linear and angular forces should be defined (Figure 5). The linear forces $\vec{F_l}$ control the length of each spring and the angular forces $\vec{F_\gamma}$ control the orientation of the spring with respect to the skeleton. Thus, two elasticity parameters are introduced : k_l spring's linear elasticity and k_γ spring's angular elasticity. The movement of the point P is obtained as follows:

$$\vec{F_l} + \vec{F_\gamma} = \vec{F}$$

with :

$$\begin{aligned} \vec{F_l} &= k_l(l_0 - l) \\ \vec{F_\gamma} &= k_\gamma(\gamma_0 - \gamma). \end{aligned}$$

with l_0 and l the spring's initial and actual length, and γ_0 and γ the spring's initial and actual orientation respectively.

These reaction forces as well as other constraints like the gravity, are then propagated through the skeleton inducing the movement of different skeleton's nodes. We should recall that the skeleton is composed of triangles whose edges are springs.

Again, as stated for the surface/skeleton springs, three kinds of forces are used so that the skeleton could recover its initial state when external and internal constraints are removed [JOU 97]. These forces are (figure 6):

– linear forces $\vec{F_L}$, to conserve the length of the skeleton's edges,

– angular forces $\vec{F_\alpha}$, to conserve the angles between the skeleton's edges,

– co-planar forces $\vec{F_{\vec{n}}}$, to conserve the orientation of the skeletons' faces (triangles).

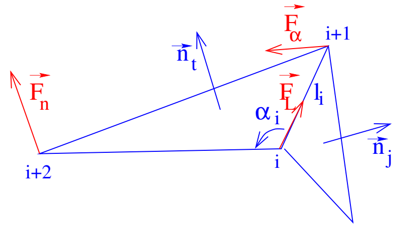

Figure 6. *Internal forces applied to object's skeleton*

The definition of these forces necessitates the definition of linear elasticity k_L, angular elasticity k_α and co-planar elasticity $k_{\vec{n}}$:

$$\vec{F}_{L_i} = k_L.(l_{i_0} - l_i),$$

with l_{i_0} and l_i spring's initial and actual length.

$$\vec{F}_{\alpha_i} = k_\alpha.(\alpha_{i_0} - \alpha_i),$$

with α_{i_0} and α_i initial and actual angle between springs.

$$\vec{F}_{\vec{n}_i} = k_{\vec{n}}.((\vec{n_{t_0}} - \vec{n_{j_0}}) - (\vec{n_t} - \vec{n_j})),$$

with $\vec{n_t}$ and $\vec{n_j}$ the normal vectors to two adjacent triangles t and j (Fig. 6).

The equilibrium condition can then be written as :

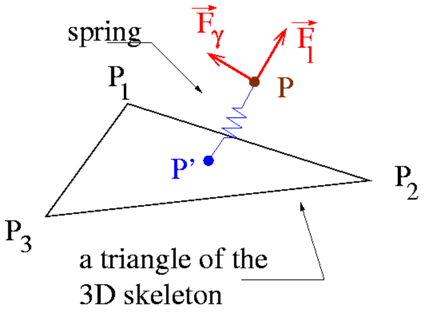

Figure 5. *Reaction of a surface/skeleton spring*

$$\sum_i \vec{F_L}_i + \vec{F_\alpha}_i + \vec{F_{\vec{n}}}_i = \vec{F}$$

Consequently, vertices' velocity is obtained by integrating these forces. The movement of the vertices deforms the skeleton, inducing the deformation of the object's surface. However, only small deformations can be easily handled. Indeed, a large deformation of the skeleton will make the corresponding surface incoherent. This type of deformation generally involves the re-computation of the skeleton. With this technique, the volume of the object can easily be controlled during the deformation as stated in [DES 95b].

Figure 7 shows an example of the application of our methodology.

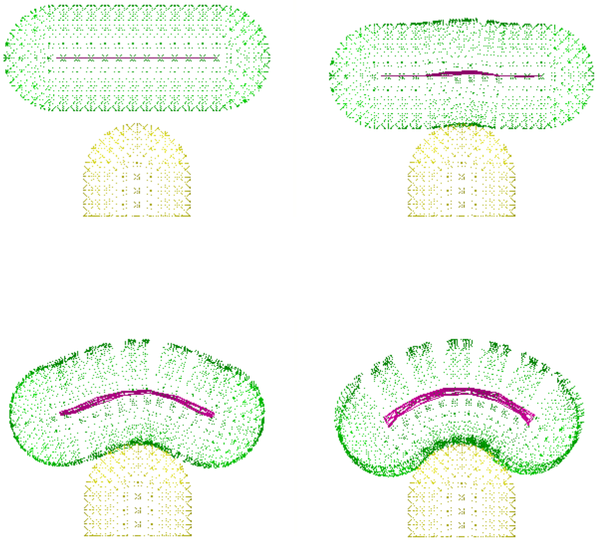

Figure 7. *Implicit surface animation and deformation*

4. Particle systems

Particle systems consist in a set of solid spheres whose movement follows physical laws. 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. Hence, particle systems model seems to be the most adapted for deformable objects modelling since many behaviours can be simulated. In this section we show how we have used this model to reconstruct and simulate deformable organs.

4.1. *Reconstruction with multi layer particle system*

Our reconstruction method requires to first compute a closed surface from the initial set of points. When the initial data is structured in parallel section, we can compute an implicit surface with the method described in previous sections. Then we fill the generated surface with particles.

However, the initial data can be an unorganised set of points. In this case, we have developed a method, based on the A-shape model, which permits to obtain a triangulated closed surface of the points set. This method is described in the following section.

4.1.1. *Surface calculation with A-shape*

In case of unorganised points set, surface reconstruction is realized using the three dimensional A-shape of three dimensional cloud of points and weighted set of points where each point's weight is the radius of a ball centered at this point [MEL 00]. Hereafter, we present the A-shape technique.

4.1.1.1. Notations

Let $S = \{p_1, \cdots, p_n\}$ be a set of n points in $\mathbb{R}^3$, $d(p, q)$ is the Euclidean distance between the points p and q. $R(S, p_i)$ denotes a Voronoï région of p_i. s_T "convex hull of T" is k-simplex of the Delaunay Triangulation of S, T being a subset of S of size $|T| = k + 1$.

4.1.1.2. Three-dimensional A-shapes

Definition 4.1. Let us recall that a ball b is said to be empty if and only if $b \cap S = \emptyset$. Let $A \subset \mathbb{R}^3$ be a finite set of points, T is a subset of S of size $|T| = k+1$, $0 \leq k \leq 2$. A k-simplex s_T is A-exposed if there exists an empty ball b such that the sphere ∂b goes through the points of T and at least a point of A. The A-shape of S -instead of use $F(S, A)$- is the polytope defined by the A-exposed simplices.

By varying the positions of the points of A, A-shapes of a set S describe a family of polytopes which presents S with different levels of details. The extreme levels of details reflected by A-shape of S are the convex hull of S and S itself. Between these two levels, A-shapes present "crude" shapes and "fine" ones which reflect cavities and holes.

4.1.1.3. Computing the A set

A-shape can be used to reconstruct curves and surfaces or more generally to identify clusters in trivariate data. In this last case, the points are not only on the surface

of an object but also in its interior. These points constitute a volumic pattern. The goal is to propose a method to compute A so that A-shape reflects an adequate shape. The idea of the A computation is based on the close relationship between an object and its skeleton. In other words, if an object is sampled and one places the points of A on its associated skeleton, A-shape corresponds then, to an adequate shape of the object. Knowing only the Voronoï diagram, we identify the faces which approximate the skeleton as those belonging to elongated polytopes.
Let $r_{\max}$ be the maximum distance from $x,\ x \in S$, to the boundary of a bounded polytope $R(S, x)$ and $r_{\min}$ denotes the minimal distance.
We said that $R(S,\ x)$ is an *elongated polytope* if and only if $r_{\max}/r_{\min} > t$ (t being a fixed threshold).
To determine the faces which belong to the skeleton, we refer to the center of gravity $g(x)$ of each polytope $R(S,\ x)$. Thus, for each polytope $R(S,\ x)$, we consider the set $A(x)$ of the Voronoï vertices $v \in R(S,\ x)$ which verify the relationship :

$$d(v,\ g(x)) \leq \sqrt{d(v,\ x)^2 + \gamma d(x,\ g(x))^2} \text{ with, } 0 \leq \gamma \leq 1.$$

In other words the vertices of $A(x)$ are located in the half-space containing the gravity center $g(x)$ and are defined by the radial axis of the weighted points ($g(x)$, $\sqrt{\gamma} d(x,\ g(x))$) and $(p,\ 0)$. The union of the sets $A(x)$, $x \in S$, defines the set A.

When γ is equal to zero, the radial axis defined previously corresponds to the bisector of x and $g(x)$. If the value of γ increases, the number of points in $A(x)$ increases as well. In our experiments, we have fixed γ to 0.5.

To this set A, we always add a set A' of four points which are sufficiently far from S so that the vertices of the convex hull of $S \cup A'$ are in A'. Adding these last points makes it possible to detect the faces of the convex hull of S shared with the A-shape of S.

Based on this construction of A, the A-shape of S can be seen as a one parametric family of shapes ranging from "fine" to "crude" shapes. Indeed, the unique parameter of A-shape is the positive real number $t \geq 1$. For $t = \infty$, the A-shape of S corresponds to the convex hull of S. However, as t decreases, the A-shape gradually reflects details like holes and cavities.

4.1.2. *Volume reconstruction with particle system*

Our reconstruction method consists in gradually filling this volume with particles, and let them evolve to a stable state according to physical laws. The algorithm 4.1 describes our reconstruction method. Thus, we easily obtain a regular sampling

which corresponds to a maximal filling. Unfortunately, if we want to increase the precision of the resulting model, we need to decrease the radius of the particles. This will dramatically increase their number. To solve this problem, we have developed a multi-layer particle model. The key idea is to place small particles were details are required and big ones elsewhere (Fig. 8):

Algorithm 4.1.

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.

– the center is composed of particles of great radius. They are the kernel of the object,

– this kernel is surrounded by one or more layers, according to the desired precision, with decreasing radius as we get closer to the boundaries.

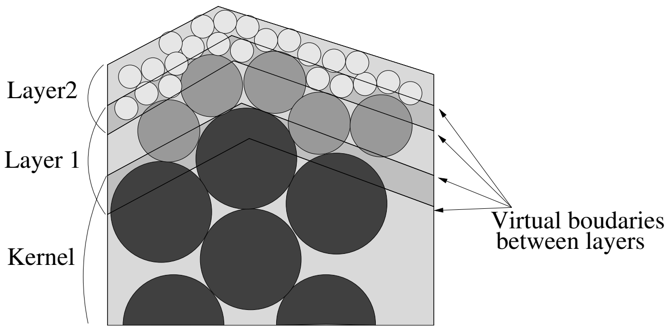

Figure 8. *Multi-layer structure*

The consistency of each layer is preserved by virtual separations between layers. Then, we can use the algorithm 4.1 to reconstruct each layer. However, when computing the internal forces, we have to include the interactions with the particles of the existing layers. The reconstruction process is finished when no particle can be added to any of the layers, and the system has evolved to a stable state. Figure 9 shows the reconstruction steps of a femoral head.

4.2. *Animation and deformation of multi layer particle systems*

Particle systems are well adapted for the simulation of deformable objects. The particles are subject to physical forces which induce their movement. Thus, these

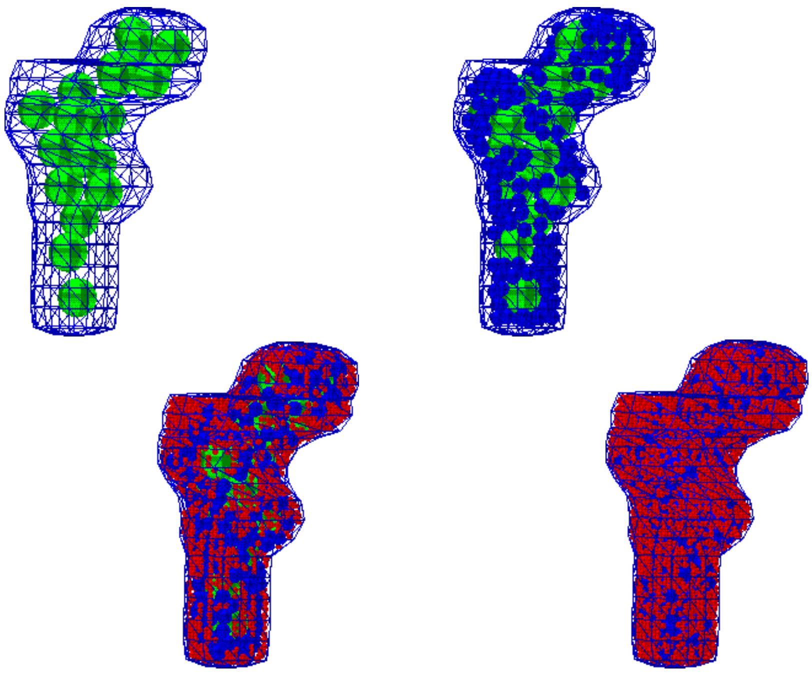

Figure 9. *Femoral head reconstruction with multi layer particle systems*

forces will determine the particles' displacement, and by extension the object's shape alteration. We can distinguish several types of forces:

– external forces, which define the environment in which evolve particles (gravity, collision with obstacles, damping...),

– internal forces that ensure shape coherence or volume preservation (attraction/repulsion, auto-collision).

It is possible to handle a wide variety of behaviours, simply by changing force formulation and its parameters.
If there is no change in the objects' topology (objects do not undergo large deformations), the internal forces could be represented by a network of mass/spring elements. This reduces greatly the complexity of the system. The interaction between particles is then simplified and modelled with the help of springs that can be written as follows :

$$\vec{f} = -\left[k_s(||\vec{r}|| - l_0) + k_d \frac{\vec{v}.\vec{r}}{||\vec{r}||}\right] \frac{\vec{r}}{r}$$

where k_s and k_d are the spring's elasticity and damping constants, l_0 is the initial length, $\vec{r}$ and $\vec{v}$ are the relative position and velocity of two adjacent particles. The constants define the characteristics of the organ: rigidity depends on k_s, while deformability depends on k_d. This permits to model different behaviours from rigid to elastic state. Elastic and inelastic deformations can also be handled by a combination of simple springs. Cutting may also be easily simulated by removing springs in contact with the cutting tool. This can be very interesting for surgery simulations. However, the fusion of objects is difficult to obtain.

For very deformable object, and in general case, the spring system is not sufficient. We use non coupled interaction forces like Lennard-Jones force (figure 10) to model internal interactions :

$$\vec{f}(r) = \frac{-mn\varepsilon}{(n-m)r_0} \left(\left(\frac{r_0}{||\vec{r}||} \right)^{(n+1)} - \left(\frac{r_0}{||\vec{r}||} \right)^{(m+1)} \right) \frac{\vec{r}}{||\vec{r}||}$$

where m and n are called the shape parameters, r_0 is the equilibrium distance, r the distance between two particles and ϵ the force amplitude.

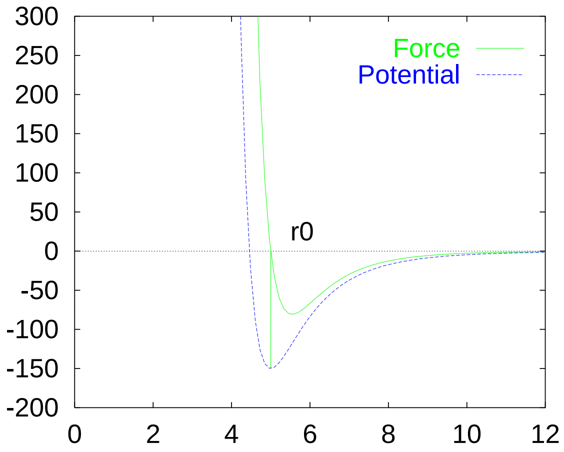

Figure 10. *The Lennard jones force*

4.2.1. *Dynamic layer structure*

During a deformation, the big particles of the kernel could get too close to the object surface, altering the layer structure of the model. This modifies the physical properties of the object, and makes the deformations inconsistent (see figure 11).

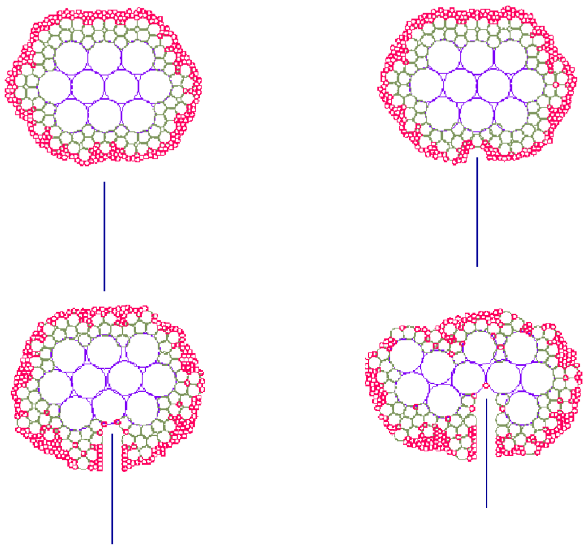

Figure 11. *Layer structure alteration*

In order 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 consistency 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 centered cubic (or hexagonal in 2D) packing of the new particles 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 $\frac{M}{n}$:

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

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

The evolution of a bi-dimensional object cut by a tool is presented in figure 12.

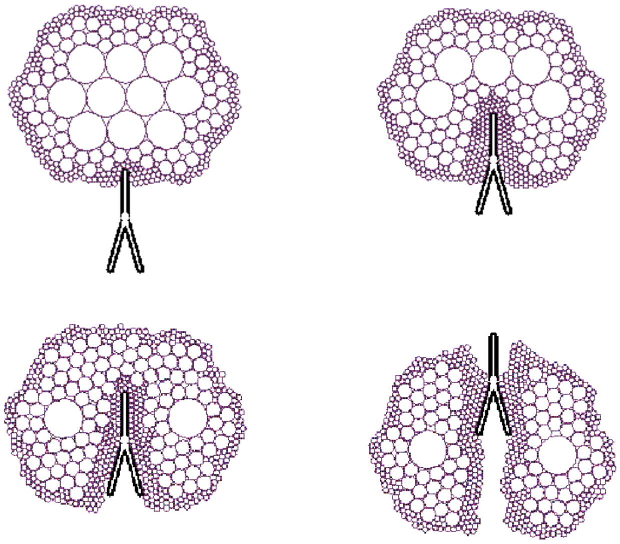

Figure 12. *Layer structure preservation*

4.2.2. *Volume control*

One of the important points that should be handled when modelling deformable objects is the control of the object's volume during the deformation. This problem is more important for organs' simulation, since we wish to obtain a more realistic behaviour. In some applications, we want to preserve object's volume during the deformation process. On the contrary, in other applications we may ask for a change of object's volume. This can happen, for example, during the simulation of some organs like the lungs whose volume increases or decreases during respiration, or the rectum's and bladder's filling/emptying process. Consequently, we should be able to control the volume changes.

Particle systems are very well adapted to handle this constraint. Indeed, since this model is a volume sampling of the initial object, an approximation of the volume can easily be found. For each particle P_i of radius R_i in the system, we define an equilibrium radius $Re_i = R_i$. The object's volume can then be computed as the sum of all the particles' volumes :

$$V = \sum_i (\frac{4}{3}\pi Re_i^3)$$

Thus, for the computation of the internal force between two particles P_i and P_j, the equilibrium distance R_0 (or l_0 for springs interaction) between these particles should be :

$$R_0 = Re_i + Re_j$$

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 minimise the whole system's energy. This avoids the creation of holes in the system. Thus, the object's volume is naturally preserved.

In the same way, in order to change the object's volume we just have to modify the equilibrium radius of some particles. Indeed, for a particle P_i, if we change Re_i to Re'_i, its volume is modified, therefore the object's volume changes. The problem is to know how to change the equilibrium distance to obtain a specific volume variation. In order to do so, we first choose the particles that will be modified. Generally, internal particles are preferable to avoid the creation of holes on the object's surface. Let N be the number of these particles. Then, the algorithm 4.2 gives the different steps to increase the object's volume of a quantity ΔV in a time Δt.

Obviously, the same algorithm can be used to decrease the object's volume.

Algorithm 4.2.

1) computing the number of steps between the times t and $t + \Delta t$:

$$n_{step} = \frac{\Delta t}{dt} \text{ where } dt \text{ is the time step}$$

2) computing the volume change at each step :

$$dV = \frac{\Delta V}{n_{step}} \text{ and then for each particle } dV_i = \frac{dV}{N}$$

inducing the following equilibrium radius change :

$$Re_{t_i+dt} = \sqrt[3]{\frac{3dV_i}{4\pi}} + Re_{t_i}$$

3) then, **for** $t_i = t$ **to** $t_i = t + \Delta t$ **by step of** dt

for each particle P to be increased, compute the new Re

let the system evolve.

The figure 13 shows an object which increases its volume by increasing the equilibrium radius of its kernel particles.

4.2.3. *Shape control*

The most significant problem of particle systems is the shape preservation during the deformation process. Since particles move according to physical laws, the system tries to minimise its energy. Often, this leads to a spherical clustering of the particles,

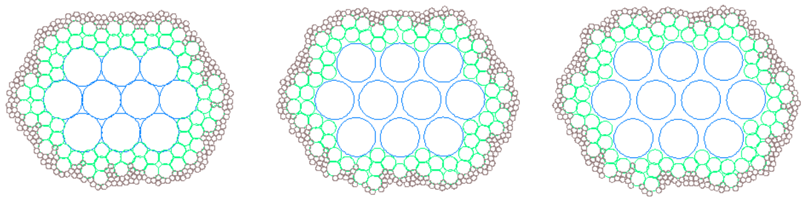

Figure 13. *Object's volume increasing*

leading to the loss of the modelled organs' shape (see figure 14). A solution to this

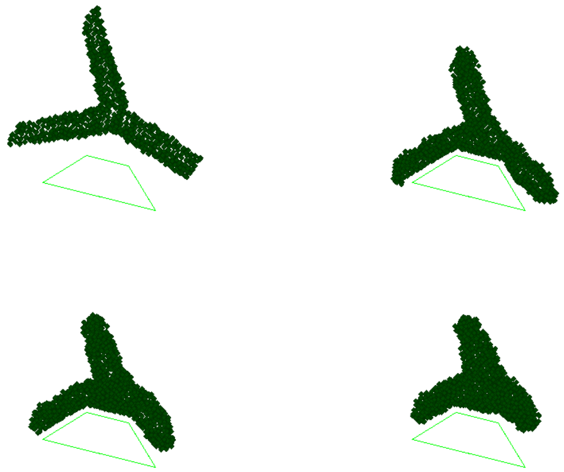

Figure 14. *A falling object : the system minimises its energy and the object's shape is lost*

problem is the introduction of a skeleton in the system, whose role is to preserve the shape of the object. In order to compute this skeleton, we use a subdivision of the object with a set of boxes. The organ's bounding box is first computed. This box is then subdivided into smaller ones according to the following criteria :

– boxes should be encompassed by a MIN and a MAX threshold,

– boxes should have a filling percentage of particles (density) greater than a given threshold.

At each subdivision step, sub-boxes are reduced to the bounding boxes of their included particles. When all boxes verify the previous criteria, the subdivision is complete. At the end of the process, each box generates a node of the skeleton. This node is associated with the particles included in the box. The mass center of these particles, and the sum of their masses, define the position and the mass of this node. After computing all the nodes, we define a neighbouring relationship between them. This can be done easily, considering that two nodes are neighbours if they are issued from two neighbouring boxes (see figure 15). In order to preserve the shape of the

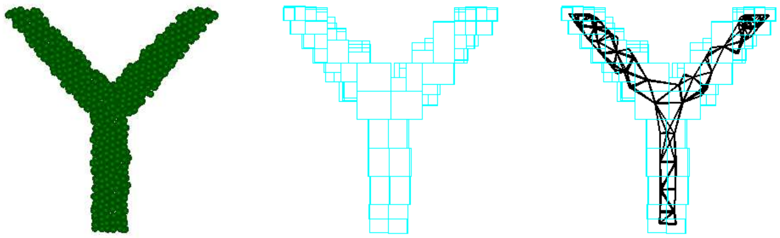

Figure 15. *Skeleton computation*

modelled organ, we introduce interaction forces (springs or Lennard Jones) between neighbouring nodes. So, at each time step, we compute for each node all the applied forces due to its interaction with its neighbours. Then, these forces are distributed among the particles associated with the node. The corresponding particles move under the action of the node, as well as other internal (inter particle) and external (collision, gravity, ...) forces. Then, after each step we update the node's position as the new mass center of its associated particles.

Thus, skeleton forces control global deformations of the organ, and preserve its shape. On the contrary, internal forces between particles control the local deformations (see figure 16). The use of a skeleton offers other advantages. First, by reducing inter particle interactions to the interaction between particles associated to the same, or to some neighbouring nodes, we can considerably reduce the computation time. Secondly, undesirable merges of organ's parts can also be avoided by introducing repulsion forces between non-interacting particles that get too close to each other. Finally, the skeleton can be used to simulate organ to organ, or organ to some external fixed points (on a bone for example) linkage. In order to do so, we only need to introduce an interaction force (spring or Lennard Jones) between a node of an organ, and a node of another organ or an external fixed point.

5. Hybrid scene

In previous sections, we have described how particle systems and implicit surfaces can be used to model deformable objects like organs. Using particle systems for modelling all the organs is computationally very expensive. On the other hand, implicit surfaces cannot simulate large deformations. This makes them undesirable

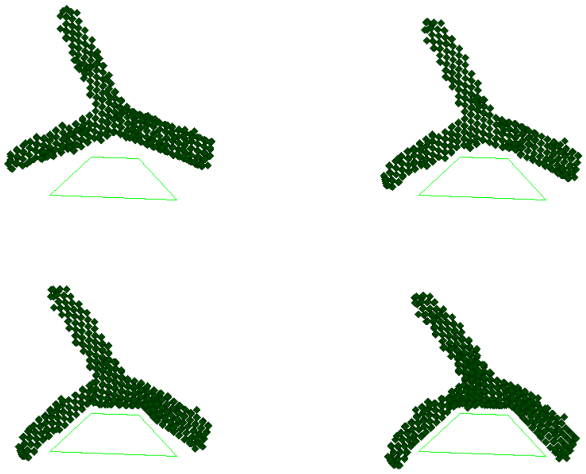

Figure 16. *A falling object : the skeleton preserves the shape during the deformation*

for the simulation of some behaviours. The key idea is to use both models. Therefore, particle systems will model organs that are subject to large deformations, and implicit surfaces for the others.

5.1. *Interaction between Particle Systems and Implicit Surfaces*

In order to use and animate two models simultaneously, we have to handle eventual interactions between them. To this end, we first define a collision detection method between a particle system and an implicit surface. A first problem is to quickly eliminate a particle which did not collide the surface. A first idea is to use the bounding box of the particle, and to test the potential of its nodes. But as we can see in figure 17, this method does not allow us to discriminate a particle since a collision can occur although no node of the particle's bounding box is included in the object. The other approach consists in using the classical sampling based method, which needs to test the potential of all sampled points on the particles' surfaces, and also the distances between the sampled points on the implicit surface and the particle center. However, this method is very expensive. Indeed, for a precise collision detection we need to increase the number of sampling points, and hence the computing time.

To avoid this problem, we propose a new collision detection method based on the potential function of the implicit surface.

Definition 5.1. Let $P(c, r)$ be a particle of center c and ray r. Let S be an implicit surface, $S_q = \{s_1, \cdots, s_m\}$ its skeleton which is composed from m skeletal ele-

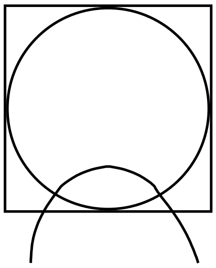

Figure 17. *Using particle bounding box*

ment, $f_1, \cdots, f_m$ the local potential functions associated with the skeleton elements, $F = \sum_{i=1}^{m} f_i$ the global potential function and ISO the iso potential value.

- Point of maximum local potential : it is the point p_i of the particle P which has the maximum potential according to the skeletal element s_i :

$$\forall p \in P(c, r), f_i(p) \leq f_i(p_i)$$

and it is also the closest particle's point to s_i :

$$\forall p \in P(c, r), d(p, s_i) \geq d(p_i, s_i)$$

- Local minimal distance : it is the distance $dmin_i$ between the point p_i and the skeletal element s_i :

$$dmin_i = \begin{cases} d - r & : \quad d > r \\ 0 & : \quad else \end{cases} \qquad where\ d = d(s_i, c)$$

Proposal 5.1. Let $B = \sum_{i=1}^{m} f_i(dmin_i)$ then

$$\forall p \in P(c, r), F(p) \leq B$$

Proof 5.1. $\forall p \in P(c, r)$,
$\forall i \in \{1, \cdots, m\} \quad d(p, s_i) < dmin_i$
$\Longrightarrow \forall i \in \{1, \cdots, m\} \quad f_i(d(p, s_i)) \leq f_i(dmin_i)$ (the f_i are decreasing functions),
$\Longrightarrow F(p) = \sum_{i=1}^{m} f_i(d(p, s_i)) \leq B$

The previous proposal means that B is the maximal potential value of a point of the particle. So, if $B < ISO$, we can deduce that there is no collision between the particle and the implicit surface.

In the other case ($B \geq ISO$), a collision can exist and we have to refine the detection. In order to do that, we first consider the surface implicit to be composed by two kinds of surface pieces :

1) surface areas resulting from the potential due to only one skeletal element. Although an implicit surface is defined as the result of the blending of potentials of all its skeletal elements, in practice the contribution of a skeletal element is restricted by an action ray beyond which the potential due to this element is negligible;

2) surface areas resulting from blending the potentials of the skeletal elements.

When we consider the first case, if a collision occurs in one of these areas it should be on one of the points of maximum local potential. Therefore, in order to detect the collision, we have just to test for each p_i if its local potential according to the skeletal element s_i is greater or equal to the iso potential value of the implicit surface ($f_i(p_i) \stackrel{?}{\geq} ISO$). Then, if this condition is true for one of these points, a collision has occurred on it.

Finally, to detect a collision in the areas defined from blending the potentials we have developed an heuristic which consists in computing the weighted affine combination (p_{aff}) of the points of maximal local potential. The weights are the normalised value of the local potentials associated to these points :

$$w_i = \frac{f_i(p_i)}{\sum_{j=1}^{m} f_j(p_j)} \quad p_{aff} = \sum_{i=1}^{m} w_i p_i$$

Using these weights allows us to take into account that the particle is closer to the part of the implicit surface defined by a certain skeletal element but also the contributions of the other elements. Then, we project the resulting point on the particle's surface, and we compute the potential value of the projection. The comparison of this value with the ISO-potential of the surface allows us to determine whether there is a collision or not (see Fig 18).

For collision detection of a particle system we just have to apply this method to each particle. We can reduce the number of particles that may collide the surface to those included in the intersection of the bounding boxes.

5.1.1. *Experimental results*

We have implemented our algorithm, and have compared it to a collision detection method based on the implicit surface sampling, which consists in testing the distance between the sampling points and the particle center. Therefore, we have applied the two algorithms for the collision detection of a single particle and an implicit surface. The case of a single particle seems to be more significant since the consideration of a complete particle system does not give us more information to compare the two algorithms.

We first test the sampling based method. Figure 19 shows the computing time needed by this method according to the number of sampling points. As we can expect, the time increases linearly with the number of sample points. Then, we have tested the method we have developed. We can see in figure 20 three curves showing the time evolution of our algorithm in its three cases. The worst case (curve #3) happens

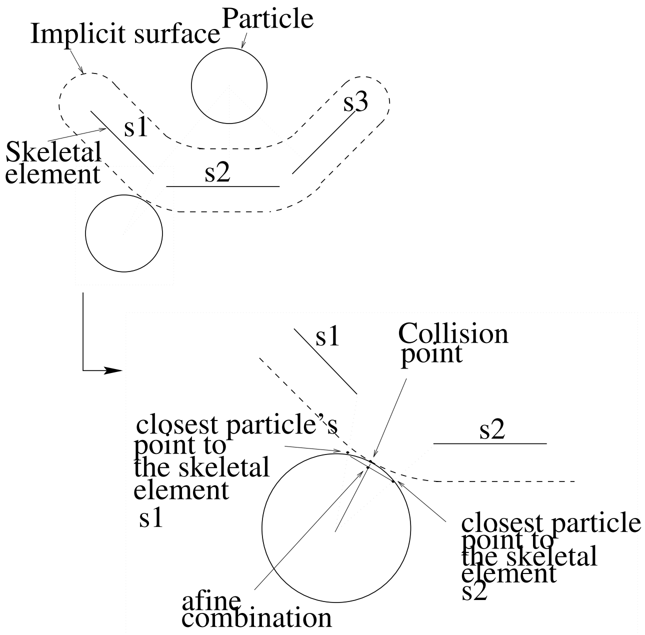

Figure 18. *Collision detection between a particle and implicit surface*

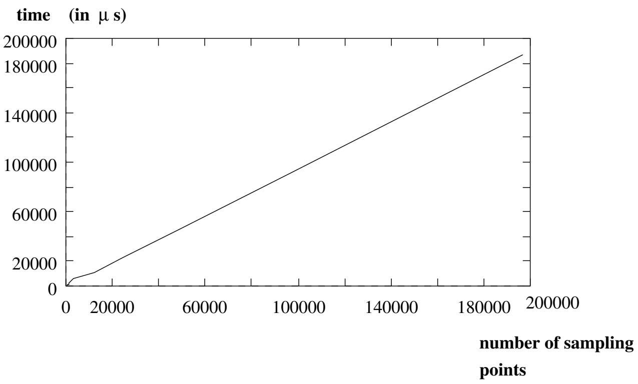

Figure 19. *Collision detection with a sampling based method*

when the collision occurs on an area of the implicit surface which results from the blending of the potentials of the skeletal elements. However, in practice the number of colliding particles is generally much smaller than the number of eliminated ones. Thus, our robust elimination criterion whose low cost is shown on the curve #1, seems to be a real advantage in making the collision detection process faster. Finally, we

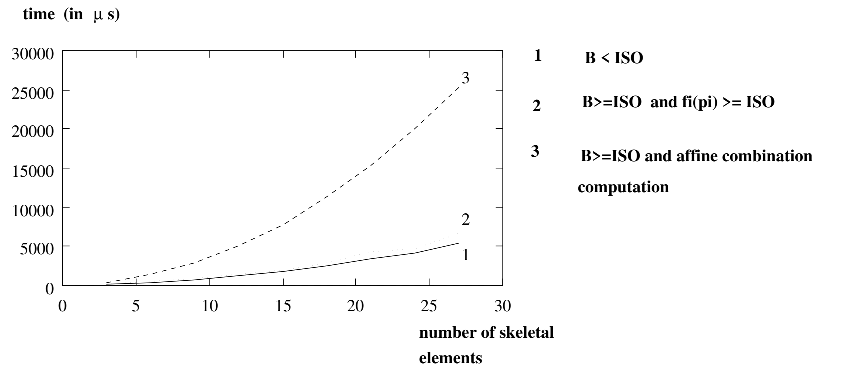

Figure 20. *Collision detection with our method*

tried to compare the two methods. In order to do so, we defined a sampling criterion which consists of choosing a number of sampling points per skeletal elements . Then, we applied the two methods on the same objects, increasing the number of skeletal element. The figure 21 shows that our method gives the best results.

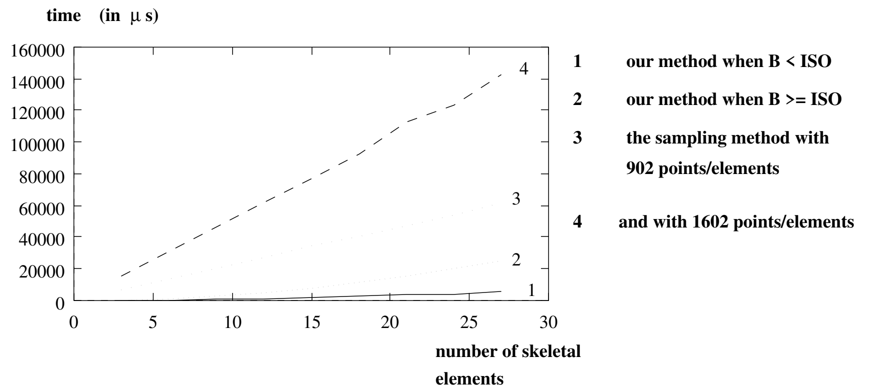

Figure 21. *Comparison of the two methods*

5.1.2. *Collision response*

When a collision is detected, a reaction force is computed and applied to the particles and to the implicit surface. Then, the two models are animated and deformed as described in previous sections. Figure 22 illustrates an example of an animated hybrid model. The lungs and the heart are modelled by particle systems and the mediastinum by an implicit surface. Thus we can modelise the organs interactions during the breathing process. In order to make the simulation more realistics the body, modelled by an implicit surface, has been added. Then, the movement of the lungs is constrained by the body which is deformed under the pressure of the lungs (see figure 23).

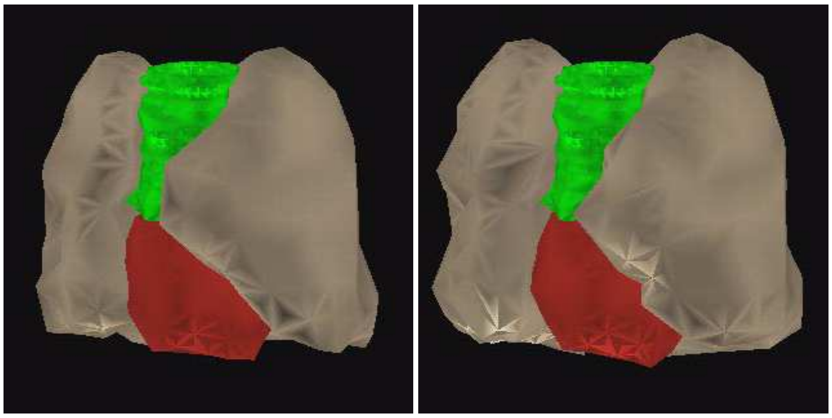

Figure 22. *Animation of a hybrid scene*

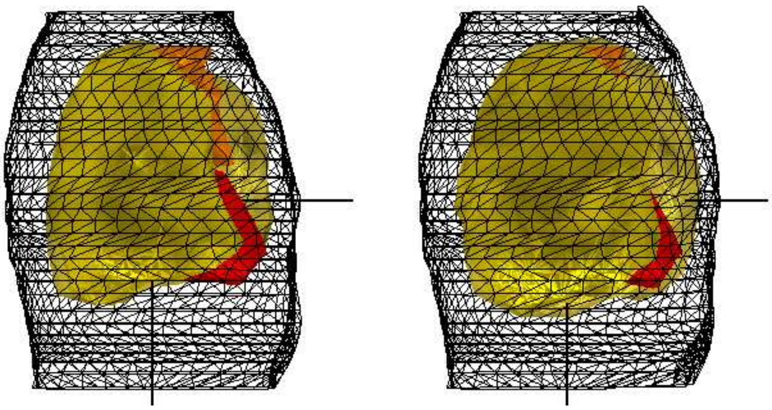

Figure 23. *Breathing simulation*

6. Conclusion

In this paper, we have presented a modelling technique for the reconstruction of deformable objects' shape with implicit surfaces and particle-based systems (hybrid approach). This technique seems to be adequate to model and simulate the interaction between the organs.

For implicit reconstruction, a method using the skeleton-based implicit surface has been introduced. The reconstructed objects can contain holes or can have several branches. In order to avoid the use of a blended field function, the objects are described with a uniform field function and a weighted skeleton. Rather than computing the skeleton from the whole 3D cloud of points, we deduce it automatically from the 2D Voronoï skeleton of each CT scan section.

Particle systems have also been used to reconstruct highly or globally deformable objects. The particle model is very simple because it is only composed of particles that are driven by spatial interaction forces. Nevertheless, their animation necessitates important computation time. To reduce the complexity of the model, multi-layer particles have been presented.

With this hybrid technique, most of anatomical shapes can be modelled. Our modelling algorithms have been integrated in a global treatment software. In our trials, the physical parameters that permit to define the organs' characteristics have been set intuitively. In cooperation with Christie Hospital in Manchester (UK) we are planning to find a better approximation of these parameters by long series of tests. Moreover, the computation time is actually rather prohibitive. Consequently, we are optimising our calculation algorithms.

Acknowledgment - This work was supported in part by EC Framework IV, Biomed II, contract BMH4-CT98-3660, project ARROW (Animated Realtime Radiation Oncology Worktools).

7. References

[ATT 95] ATTALI D., "Squelettes et Graphes de Voronoï 2D et 3D", PhD thesis, Université Joseph Fournier-Grenoble I, Oct. 1995.

[BAJ 87] BAJSCY R., SOLIN R., "Three-dimensional object representation revisited", *IEEE First Conference on Computer Vision*, , 1987, p. 231-240.

[BAR 95] BARDINET E., COHEN L. D., AYACHE N., "A parametric deformable model to fit unstructured 3D data", report num. 2617, July 1995, INRIA.

[BOI 92] BOISSONNAT J. D., GEIGER B., "Three dimensional reconstruction of complex Shapes based on the Delaunay", report num. 1697, April 1992, INRIA.

[COH 94] COHEN I., COHEN L., "A Hybrid Hyperquadric Model for 2-D and 3-D Data Fitting", report num. 2188, January 1994, INRIA.

[COT 96] COTIN S., DELINGETTE H., AYACHE N., "Volumetric deformable models for surgery simulation of non-rigid organs", *4e séminaire du groupe de travail "Animation et Simulation"*, GDR-PRC AMI (Algorithme, Modélisation et Infographie), Strasbourg, Jan. 1996.

[DER 92] DERDOURI B., NEVEU M., FAUDOT D., "Reconstruction d'objets 3D déformables", *Actes de GROPLAN 92, Nantes*, , 1992, p. 99-106.

[DES 95a] DESBRUN M., GASCUEL M.-P., "Animating Soft Substances with Implicit Surfaces", *Computer Graphics*, , 1995, Proceedings SIGGRAPH'95.

[DES 95b] DESBRUN M., TSINGOS N., GASCUEL M.-P., "Adaptive Sampling of Implicit Surfaces for Interactive Modelling and Animation", *Implicit Surfaces'95*, Grenoble, France, april 1995, p. 171-186.

[FER 96] FERLEY E., GASCUEL M.-P., ATTALI D., "Skeletal Reconstruction of Branching Shapes.", *Implicit Surfaces'96: 2nd International Workshop on Implicit Surface*, Eindhoven, The Netherlands, October 1996.

[GAS 93] GASCUEL M.-P., "An implicit formulation for precise contact modeling between flexible solids", *Proceedings of SIGGRAPH '93*, Computer Graphics, Aug. 1993, p. 313-320.

[JOU 97] JOUKHADAR A., GARAT F., LAUGIER C., "Parameter Identification for Dynamic Simulation", *IEEE Int. Conf. on Robotics and Automation*, New Mexico, US, 1997, p. 1928-1933.

[LEO 94] LEONARDIS A., SOLINA F., MACERL A., "A Direct Recovery of Superquadric Models in Range Images Using Recover-and-Select Paradigm", *Proceeding of third European Conference on Computer Vision*, Stockholm, Sweden, May 1994.

[LOM 95] LOMBARDO J.-C., PUECH C., "Oriented Particles: A Tool for Shape Memory Objects Modeling", *Graphics Interface'95, Quebec City (CAN)*, May 1995.

[MEL 00] MELKEMI M., SHARIAT B., "Reconstruction of three-dimensional objects from unorganized points set", report , 2000, LIGIM, Claude Bernard university.

[MES 97] MESEURE P., CHAILLOU C., "Deformable Body simulation with Adaptive Subdivision and Cuttings", *5^{th} Int. Conf. in Central Europe on Comp. Graphics and Visualization WSCG'97, Plzen, CZ*, Feb. 1997, p. 361-370, IFIP working group 5.10 on Computer Graphics and Virtual Worlds.

[MUR 91] MURAKI S., "Volumetric Shape Description of Range Data using "Blobby Model"", *Computer Graphics*, vol. 25, num. 4, 1991, p. 227-235.

[PAR 94] PARK J., METAXAS D., YOUNG A., "Deformable Models with Parameter Functions : Application to heart-Wall Modeling", *Conf. on Computer Vision and Pattern Recognition*, June 1994, p. 437-442.

[PEN 86] PENTLAND A., "Perceptual Organization and the Representation of Natural Form", *Artificial Intelligence*, vol. 28, num. 3, 1986, p. 293-331.

[PON 00] PONTIER S., "Reconstruction d'objets déformables à l'aide de fonctions implicites", PhD thesis, University Lyon 1, April 2000.

[REE 83] REEVES W. T., "Particle systems: a technique for modeling a class of fuzzy objects", *Proceedings of SIGGRAPH'83*, Computer Graphics, 1983, p. 359-376.

[SAV 95] SAVCHENKO V. V., PASKO A. A., OKUNEV O. G., KUNII T. L., "Function Representation of Solids Reconstructed from Scattered Surface Points and Contours", *Computer*

Graphics Forum, vol. 14, num. 4, 1995, p. 181-188.

[SHI 95] SHIMADA K., GOSSARD D. C., "Bubble mesh: automated triangular meshing of non-manifold geometry by sphere packing", *Third Symposium on Solid Modeling and Applications, Salt Lake City, Utah*, May 1995, p. 409-419.

[SZE 91] SZELISKI R., TONNESEN D., "Surface modeling with oriented particle systems", report num. CRL 91/14, Dec. 1991, DEC, Cambridge Research Lab.

[TER 88] TERZOPOLOUS D., FLEICHER K., "Modeling inelastic deformation: viscoelasticity, plasticity, fracture", *Proceedings of SIGGRAPH'88*, Computer Graphics, 1988, p. 269-278.

[TER 91] TERZOPOULOS D., METAXAS D., "Dynamic 3D Models with local and Global Deformations: Deformable Superquadrics", *IEEE Transactions on pattern analysis and machine intelligence*, vol. 13, num. 7, 1991, p. 703-714.

[TSI 95] TSINGOS N., BITTAR E., GASCUEL M.-P., "*Implicit Surfaces for Semi-automatic Medical Organ Reconstruction*", Chapter Techniques in Modelling Virtual Environments, p. 3-15, Academic Press, Leeds, UK, juin 1995.

[WAL 91] WALLIN A., "Constructing Isosurfaces from CT Data", *IEEE CG and A*, vol. 11, num. 6, 1991, p. 28-33.