

# Physically-Based Model for Simulating the Human Trunk Respiration Movements

Emmanuel Promayon<sup>1</sup>, Pierre Baconnier<sup>1</sup>, Claude Puech<sup>2</sup>

<sup>1</sup> TIMC - IMAG, Faculté de Médecine, I.A.B., 38706 La Tronche cedex, France

<sup>2</sup> iMAGIS\*\*\* GRAVIR-IMAG INRIA, BP53, 38041 Grenoble cedex 09, France  
e-mail: Emmanuel.Promayon@imag.fr

**Abstract.** We have developed a physically-based model where an object is represented by a set of mass points on its contour. Each object may be defined locally and physically using surface regions. Physical properties (such as elasticity, motor functioning, or rigidity) can be assigned to the regions. Physical constraints, like incompressibility (i.e. constant volume deformation), can be set. Moreover, additional constraints can be used to control the animation and the object behaviour (e.g. nailed point, pre-defined trajectory). Depending on this physical properties and constraints, forces and movements are deduced.

Using this model, we can handle bio-mechanical movements. Our simulation provides an anatomical and functional model of the evolutions of the human trunk structures during respiration.

## 1 Introduction

In this model, the movements are generated by the diaphragm muscle contraction and the incompressible property of the abdomen. After giving a review of the medical background of our project and explaining what are the phenomena to model (section 2), this paper will describe our physically-based model (section 3). The constraint problem is one of the critical point. In section 4 the constraint resolution algorithm is given. Examples in section 5 show preliminary results of the respiration modelling. Section 6 compares our model to classical mass-spring model and then explains how to generalize our algorithm to other models or constraints.

## 2 Simulating respiration

Several methods allowing 3D analysis of respiratory mechanical structures are now available in the clinical environment: MRI and CT scans allow geometric analysis of diaphragm and rib cage structures, video 3D positioning of markers placed on the skin allows continuous external monitoring of trunk movements. The interpretation of such data requires the use of models. Such an approach could be helpful for non-invasive diagnosis of some respiratory diseases encountered in intensive care units (e.g. diaphragmatic fatigue, hemothorax) or surgery (e.g. scoliosis, lung resection consequences).

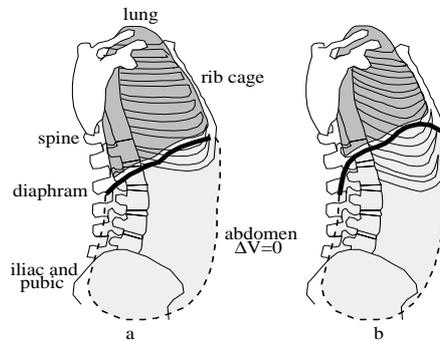
---

\*\*\* iMAGIS is a joint project of CNRS, INRIA, INPG and UJF.

In this article, we describe how our model manages multiple constraints and takes into account our particular needs (such as the use of segmented data and computing time efficiency).

## 2.1 Mechanism to be modelled

At least *four trunk structures are needed* in a mechanical analysis: the rib cage, the lung, the diaphragm and the abdomen. We first consider the quiet normal breathing. There are two phases [10]: inspiration and expiration (see figure 1). During the inspiratory phase, the diaphragm contracts and pushes on the abdomen which acts as a fulcrum and moves up the sternum and the rib cage. As the rib cage volume increases, the pressure inside the lung decreases and generates an inwards air flow. The elastic properties of the system make it return to its rest position during the expiratory phase.



**Fig. 1.** The respiration phases : (a) inspiration: the diaphragm contracts, (b) expiration: the system returns to its rest configuration.

## 2.2 Structures to be modelled

This paper presents a model of diaphragm and abdomen set where a schematic pair of ribs is modelled. This set is the more critical part since the diaphragm and the abdomen are active deformable bodies. The abdomen is an incompressible deformable object which has a rest position (due to its surrounding muscles and to the pelvis and spine skeleton). The diaphragm is made of muscular tissues which can contract. It is attached to the rib-cage. This articulation has to be modelled.

Complex deformable 3D objects with various kinds of properties have to be simulated in particular situations or scenarii (like normal or topological mechanical properties, body position or motion). In a computer model we can express all these needs in *topological parameters and constraints* on the objects. This is presented in section 3.

An interesting approach to simulating respiratory mechanics is described in [6]. Starting from a physiological description of respiratory mechanics (in terms of pressure and volume changes) and from an anatomical model, the authors obtain

quantitative results. A three-dimensional model of the lung is then animated for quiet, normal breathing and for a pneumothorax pathologic case.

### 3 The model

The model is presented in four sections. By way of introduction, some related work including deformable object modelling and constraint resolution methods are presented. Section 3.2 explains how objects are modelled. Sections 3.3 and 3.4 explain how forces could be generated from the objects and focus on local shape memory force and muscular force (see [8] for more details on local shape memory and constraint resolution). The section 3.5 presents the rigid region modelling and how those regions are linked to deformable parts.

#### 3.1 Related work

It has been proven that physically-based models are a good means of modelling and simulating natural motions and realistic-looking flexible and elastic objects ([11]). Physically-based models are described using a small amount of data (scene and object geometries, and relations between objects or parts of objects). From this, an animation motor (using forces, energies, or direct displacements), based on dynamic laws, generates the object movements and deformations. With this kind of approach direct control of the object is difficult. In [7], three kinds of method are explained to constrain physically-based models: reaction constraints, penalty and augmented lagrangian methods. In [5] and [12] articulated rigid bodies are constrained by separating forces generated by constraints from other forces. Our constraint resolution algorithm is an extension of the reaction constraint method. The constraints are separately processed. This allows to *exactly fulfill a global constraint* on a deformable model and, for example, it achieves constant volume deformations without the need of an iteration process. Other works on constant volume deformation ([1] and [9]) use a lagrangian approach. In these cases, the algorithm iteratively minimizes the error on constraints by using the lagrangian multiplier method. [1] and [9] approaches are based on Free-Form-Deformation and are not physically-based. In [9], interactive deformations are obtained.

#### 3.2 Object modelling

An object is represented by *its contour points*.

Each vertex knows its neighbours. As the object topology stays constant, the surface is defined by a triangular mesh. This allows us to directly use segmentation and reconstruction algorithm results as well as 3D point acquisition systems as inputs (see [2] and [4]). Triangular facets simplify our constant volume constraint computing as well as the computer display.

To generate forces and dynamic, a mass is assigned on each contour point.

In the remainder of this paper we use the following notation:  $O$  will denote an object,  $R$  a homogeneous property region of an object,  $P$  a point of an object; the neighbours of a considered point  $P_i$  will be denoted by  $N_i^j$  (where  $j = 1 \dots \eta_i$ ,  $\eta_i$  is the number of neighbours of  $P_i$ ).

### 3.3 Force modelling

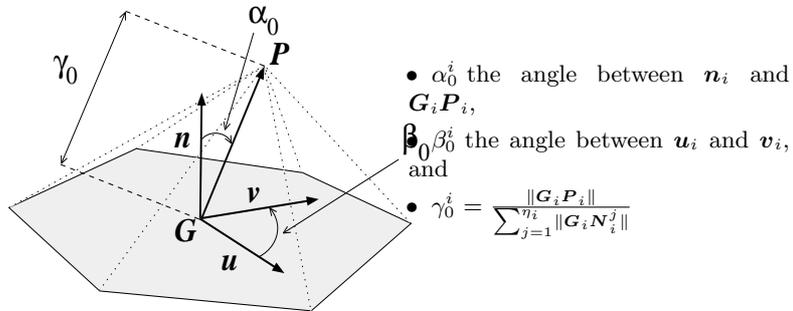
Forces acting on the contour point generate displacements and deformations. Three kind of forces can be used in our model: force fields (e.g. gravitation force), locally applied forces (e.g. user manipulation), and *forces induced by attractive points*. The last-mentioned force is used when an ideal position  $\mathbf{P}_{ideal}$  is known for a given point  $\mathbf{P}$  of the contour. We simply introduce a spring between the actual position  $\mathbf{P}$  and the ideal position  $\mathbf{P}_{ideal}$ . Thus, the point  $\mathbf{P}$  is attracted by the point  $\mathbf{P}_{ideal}$ . The resulting force is:  $\mathbf{F} = k_{attract} \cdot (\mathbf{P} - \mathbf{P}_{ideal})$ . This kind of forces can be seen as a potential force that tends to minimize a distance (e.g. muscle contraction, elasticity).

All forces act on the contour points of the objects. A weight  $k$  is set for each force that parametrizes the model. It can be interesting to vary this weight during the object deformation, as shown in section 3.4.

### 3.4 Regions of homogeneous physical property

We focus on three particular region types which take parts in the biomechanical modelling: elastic, muscular and rigid regions.

**Elastic region** The elastic property of an object region is simply its *ability to come back in its original shape* once deformed. To model this property we define a local shape coordinate system (see figure 2) where a point  $\mathbf{P}_i$  is defined relatively to its neighbours thanks to three parameters:  $\alpha_0^i$ ,  $\beta_0^i$  and  $\gamma_0^i$ . Afterwards we can generate a local shape memory force on each point. The following geometric variables are needed (see figure 2):  $\mathbf{n}_i$  (the normal unit vector of the surface in  $\mathbf{P}_i$ , computed as the mean of the normals of the triangular facet surrounding point  $\mathbf{P}_i$ ),  $\mathbf{v}_i$  (normal to  $\mathbf{n}_i$ ),  $\mathbf{G}_i$  (the isobarycenter of the  $\eta_i$  neighbours  $\mathbf{N}_i^j$  of  $\mathbf{P}_i$ ), and  $\mathbf{u}_i = \mathbf{n}_i \times \mathbf{G}_i \mathbf{P}_i$ .



**Fig. 2.** The local shape coordinates system for a point  $\mathbf{P}$ .

At each iteration, given the neighbour positions and this three parameters, we compute a shape ideal position  $\mathbf{P}_{ideal}^i$ . This defines an attraction point and the local shape memory force  $\mathbf{F}_{shape}$ . When  $\mathbf{P}_i$  is at the shape ideal position, the local deformation is null and  $\mathbf{F}_{shape} = 0$ .

An elastic region  $R_e$  is a region where the local shape memory force is applied to all the points.

**Muscular region.** Our aim is not to develop a complete model of the muscle (as in [3]) but to simulate muscular behaviour linked to other structures. Motor functioning results in tissue contraction. Contraction is simply the diminution of the distance between contour points. In our model, a muscular point  $\mathbf{P}$  is attracted towards some of its neighbours. The contraction directions determine a set of neighbours as ideal points. The force induced by these attractive points contracts the contour.  $\mathbf{P}$  and its neighbours define a muscular region.

During the evolution of the object, the force stiffness weight  $k_{muscle}$  can be increased and decreased according to a contraction force function.

A muscular region is a region where the muscular and the local shape memory forces are applied to all the points.

### 3.5 Rigid region

Rigid region can be linked with two different kinds of structures: with muscular or elastic tissues and with other rigid regions. In the first case, we model the link using a constraint. In the second case we put a spring to link several rigid regions.

The first link is presented here. The muscular or elastic regions have to be constrained in order to remain attached to rigid regions. Let  $R_e$  be an elastic region and  $R_r$  be a rigid region (e.g. the abdomen-diaphragm set  $R_e$  where a pair of ribs  $R_r$  is added, see section 5). The rigid body  $R_r$  is defined using a position  $\mathbf{C}_r$  and an orientation  $\mathbf{q}_r$  (where  $\mathbf{C}_r$  is a 3D vector and  $\mathbf{q}_r$  a quaternion). If no point of  $R_r$  is nailed,  $\mathbf{C}_r$  is the isobarycenter of the  $R_r$  masses, if one point is nailed  $\mathbf{C}_r$  is equal to this point position (and will remain constant), if two points are nailed  $\mathbf{C}_r$  is the barycenter of the two fixed points. Each point  $\mathbf{P}_r^i$  of  $R_r$  is defined in a local coordinate system of the rigid body.

Each point  $\mathbf{P}_r^i \in R_r$  is connected to a point  $\mathbf{P}_e^i \in R_e$ . Hence  $R_e = R_e^l + R_e^f$ , where  $R_e^l$  contains all the points  $\mathbf{P}_e^i$  that are linked to a point  $\mathbf{P}_r^i \in R_r$ .  $R_e^f$  contains all the remaining points of  $R_e$ .

At each time step, the new position of the rigid region  $R_r$  is computed and all the points  $\mathbf{P}_e^i$  of  $R_e^l$  are *projected* on this region new position. To compute the movements of  $R_r$ , we use the following algorithm:

1.  $\forall \mathbf{P}_e^i \in R_e$  compute  $\mathbf{F}_e^i$  (the sum of external and internal elastic forces),
2.  $\forall \mathbf{P}_r^i \in R_r$  compute  $\mathbf{F}_r^i$  (the sum of external forces),
3.  $\forall \mathbf{P}_e^i \in R_e^l$ , do  $\mathbf{F}_r^i = \mathbf{F}_r^i + \mathbf{F}_e^i$  (direct transmission of forces from  $\mathbf{P}_e^i$  to its connected point  $\mathbf{P}_r^i$ ),
4.  $\forall \mathbf{P}_r^i \in R_r$  compute  $\mathbf{F}_r = \sum \frac{\mathbf{F}_r^i}{m_i}$  (the sum of forces applied to  $R_r$ ),
5.  $\forall \mathbf{P}_r^i \in R_r$  compute  $\mathbf{T}_r = \sum \mathbf{C}_r \mathbf{P}_r^i \times \mathbf{F}_r^i$  (the sum of torques applied to  $R_r$ ),
6. compute new position  $\mathbf{C}_r$  and new orientation  $\mathbf{q}_r$  by integrating  $\mathbf{F}_r$  and  $\mathbf{T}_r$  through time (using Euler – or another one – integration scheme),
7.  $\forall \mathbf{P}_r^i \in R_r$  compute the new position and speed.
8. *assign* to all the points  $\mathbf{P}_e^i \in R_e^l$  the position and speed of the corresponding connected point  $\mathbf{P}_r^i$ .

## 4 Constraint resolution

Our model can deal with two kinds of constraints:

- *local constraints* are applied to only one point (for example to nail a point or to keep it in a particular region of the space),
- *global constraints* are applied to a set of points (for example to keep the object volume constant).

Both kind of constraints can be applied in position, velocity or any other parameter space. Constraints are *separately processed* from the other forces. We extend the projection method ([7]) to handle global constraints.

### 4.1 Constraint solving

The idea is that verifying a constraint is equivalent to staying in the corresponding parameter region. Inside this region (see figure 3) the constraint is verified. To explain our algorithm, we take the example of the constant volume constraint (see also [8]). Let  $\mathbf{X}^t$  be a vector which holds the positions of the object points at time  $t$ . We want to deform this object without any change in its volume.  $\mathbf{X}$  is composed by all the  $\mathbf{P}_i \in O$ .

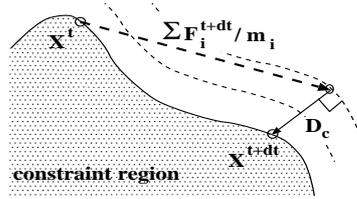


Fig. 3. constraint resolution using projection method.

The algorithm :

1.  $\forall \mathbf{P}_i \in O$  compute the sum of forces,
2. Integrate dynamic equations without considering the constraints,
3. Project  $\mathbf{X}$  on constraint region (take constraint into account),
4. Adjust parameters (like speed).

### 4.2 Finding the projection vector $D_c$

$D_c$  is the projection vector. It allows us to find the nearest point of  $\mathbf{X}_{free}^{t+dt}$  that is in the constraint region. In figure 3 one can see that  $D_c$  is simply the *gradient vector* of the constraint region. Two cases appear:

- the constraint region is a *known point*  $\mathbf{X}_c$ . This is the case for nailed points, or for rigid part constraint (see section 3.5). Then,  $D_c = \mathbf{X}_{free}^{t+dt} - \mathbf{X}_c$ .
- the constraint region is *analytically defined* in function of  $\mathbf{X}$ , we use, the gradient of the region as direction vector :  $D_c = \varphi \cdot \nabla$ . We have to find  $\nabla$  in point  $\mathbf{X}_{free}^{t+dt}$  and the scalar  $\varphi$ .

Finally, the constraint resolution amounts to solving the following system:

$$\begin{cases} \mathbf{X}^{t+dt} &= \mathbf{X}_{free}^{t+dt} + D_c \\ \Phi(\mathbf{X}^{t+dt}) &= 0 \end{cases} \quad (1)$$

$\Phi(X)$  is the constraint function. For the constant volume constraint, for example,  $\Phi(\mathbf{X}^{t+dt}) = V(\mathbf{X}^{t+dt}) - V(\mathbf{X}^0)$  (where  $V(\mathbf{X}^0)$  is the volume at rest shape).

### 4.3 Constant volume

As an example of constraint resolution, we show how to obtain a constant volume deformation. We have to analytically express the volume of our object (or region) depending on the positions of its contour points. The use of triangular facets highly simplifies the discretization of the divergence theorem. We finally found:

$$V(X) = \frac{1}{6} \mathbf{X}^T \left( \sum_{k=1}^{\rho} \Psi_k(\mathbf{X}) \right) \mathbf{X}, \quad (2)$$

where  $\rho$  is the number of facets.  $\Psi_k(\mathbf{X})$  is a function of a facet  $k$  such as  $\mathbf{X}^T \Psi_k(\mathbf{X}) \mathbf{X}$  gives the volume contribution of facet  $k$ . From this, we can deduce the gradient vector of the volume which is:

$$\nabla(X) = \frac{\partial V(\mathbf{X})}{\partial \mathbf{X}} = 3 \cdot \left( \sum_{k=1}^{\rho} \Psi_k(\mathbf{X}) \right) \mathbf{X}. \quad (3)$$

We can now use formulas (2) and (3) in system (1). Optimizing the computing of functions  $\Psi_k(\mathbf{X})$  results in a high simplification of the system since it boils down to solve a third degree equation in function of  $\varphi$ . By this mean, the volume is kept constant in one iteration.

**5 First results**  
 We model the abdomen-diaphragm set by a *incompressible* spherical object separated in two regions (see figure 4). The region on the top is defined as a muscular region (modelling the diaphragm). The other region is an elastic region (modelling the rest of the abdomen and its surrounding muscles). A schematic pair of ribs is attached to this set. The ribs are modelled by a non-closed circular rigid region linked to the muscular region. Some points of the objects are nailed in the space by a local constraint (modelling the spine, the iliac and pubic bones). As the muscular region contracts, and due to the incompressibility of the abdomen, the rigid region moves up symbolizing the rib cage movements.

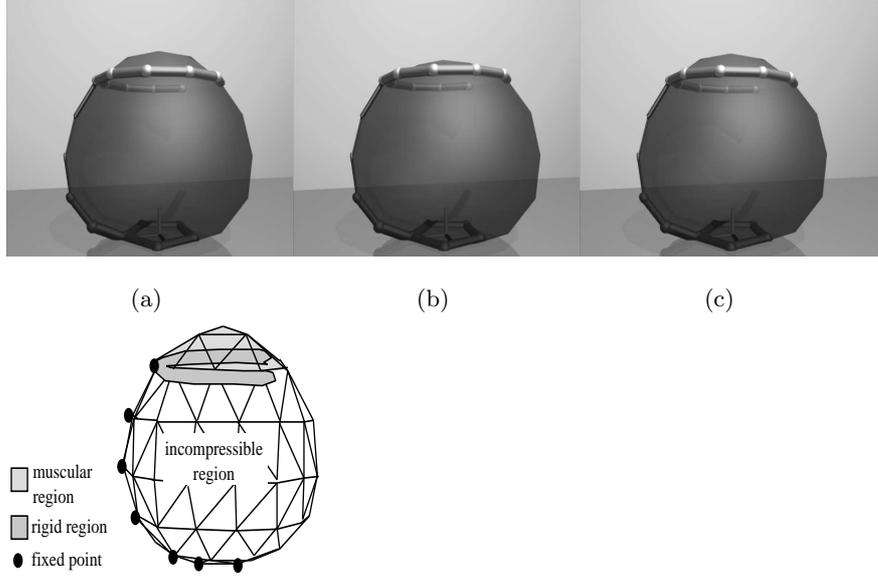
## 6 Model study

This section presents a comparison of our local shape memory force model to mass-spring model and a constant volume constraint generalization.

### 6.1 Comparison

We choose to compare our model to another simple elastic model: the mass-spring model where points on the surface are connected to their neighbours by springs. We applied the classical method of stability analysis to both systems. We first formalize the models as systems of difference equations and we compare their stability for small variations.

We express the Euler integration scheme as a system of difference equations. We only apply an elastic and a viscosity force (proportional to speed with the



**Fig. 4.** *The abdomen-diaphragm set with a schematic pair of ribs fixed on the top. (a) shows the rest shape (b) the muscle contracts, the ribs move up (c) when muscle contraction is released the system returns to its original shape (80 pts, 156 facets, 0.07s/iteration on a DEC alpha 3000/500, 90Mhz; Phong shaded images). The left figure shows the geometry of the model.*

coefficient  $k_v$ ). The difference equations of both methods are expressed as a function of  $h$  (the integration time step), and the elastic force function  $F^e$  (different for the two models),  $k_v$ , and  $m$ , the mass of each point.

$$\begin{cases} \mathbf{X}^{n+1} = \mathbf{X}^n + h \cdot \dot{\mathbf{X}}^n \\ \dot{\mathbf{X}}^{n+1} = \dot{\mathbf{X}}^n + h \cdot \left( \frac{F^e(\mathbf{X}^n)}{m} - k_v \cdot \dot{\mathbf{X}}^n \right) \end{cases} \quad (4)$$

From system (4), one obtain an iterative system, the stability of which may be studied by its *Jacobian*. We have performed a stability analysis in a very simple case. We also simplified the object at the extrema, taking only three points, two of them being fixed (this allows us to work in a two dimensional space  $(x, y)$  for both position and speed), the last one is free to move. In this case, the jacobian matrix for both models is as follows:

$$J = \begin{pmatrix} 1 & 0 & h & 0 \\ 0 & 1 & 0 & h \\ -\frac{\partial F_x^e}{m \partial \dot{x}} & -\frac{\partial F_y^e}{m \partial \dot{x}} & 1 - \frac{h \cdot k_v}{m} & 0 \\ -\frac{\partial F_x^e}{m \partial \dot{y}} & -\frac{\partial F_y^e}{m \partial \dot{y}} & 0 & 1 - \frac{h \cdot k_v}{m} \end{pmatrix} \quad (5)$$

The stability of such a system is found when:

$$\forall \lambda_i, \text{if } \lambda_i \in \mathbb{R} \text{ then } |\lambda_i| < 1, \\ \text{if } \lambda_i \text{ is complex then } |\lambda_i| < 1 \text{ and } \text{Re}(\lambda_i) \neq 0,$$

where  $\lambda_i$  are the eigenvalues of the jacobian matrices. To compare the models we match the weights  $k_{ms}$  (for the mass-spring method) and  $k_{sm}$  (for the local shape memory method) of the elastic forces by taking a similar recall force for a given displacement from resting position. The theoretical limit  $k_{sm} = \frac{3}{2}k_{ms}$  gives an equivalence between the elastic force weights. Then, we compute the eigenvalues of the jacobian matrix of each method in function of  $h$ ,  $k$  (the unified weight),  $k_v$  and  $m$  (see table 1). The comparative results are given in table 2.

mass-spring	local shape memory
$\left\{ 1, 1 - \frac{hk_v}{m}, \frac{2m - hk_v \pm \sqrt{h^2 k_v^2 - \frac{16}{3} m h^2 k}}{2m} \right\}$	$\left\{ \frac{2m - hk_v \pm \sqrt{h^2 k_v^2 - 4m h^2 k}}{2m} \right\}$

**Table 1.** Eigenvalues  $\lambda$  defined in function of  $h$ ,  $k$ ,  $k_v$  and  $m$ .

	mass-spring	local shape memory
maximum $h$ values	$m/k_v$	$2m/k_v$
maximum $k$ values	$3/16m$	$1/4m$

**Table 2.** Analytical comparison.

The mass-spring system is limited as compared to the local shape memory system since the integration time step may be twice for the local shape memory and the elasticity coefficient may be higher.

These theoretical results have been strengthened by simulations on a 3 dimensional object (an icosaedron) pertubated in a non isotropic way. The perturbation amplitude is given in function of  $l_0$  the distance between points at rest shape (see table 3). We compare the number of iteration needed to reach equilibrium and compute the the limit values of  $h$  beyond which the system was not able to reach its steady state.

	mass-spring	local shape memory
small perturbation ( $0.3 \cdot l_0$ )	3.8	4.0
high perturbation ( $0.7 \cdot l_0$ )	1.7	2.6

**Table 3.** Simulation comparison, maximum  $h$  values.

The results given in table 3 indicate that the local shape memory is more robust than the mass-spring in this situation, as predicted by the theory in a simpler situation.

## 6.2 Generalizations

We can demonstrate that the volume gradient direction  $\nabla_i$  at a point  $\mathbf{P}_i$  of the object surface is *colinear to the surface normal* at this point  $\mathbf{n}_i$ . This allows to state that:

- our constraint projection is similar to add forces that simulate the intern pressure (since pressure forces on an object membrane are directly linked to its compressibility property),
- our method can be applied to every *other model* where the volume can be analitically expressed as a function of the object surface (e.g. FFD, mass-spring network, particle system).

Moreover, by taking  $\mathbf{X}^{t+dt}$  that verifies the volume constraint (see figure 3) as an attractive point, we can simulate the compressibility property.

## 7 Conclusion

In this paper, a physically-based approach has been proposed to model and handle constraints. Object region properties can be defined (elasticity, motor functioning, rigidity). The way an object is modelled allows us to define a simple efficient algorithm to take global constraints into account (such as constant volume deformations) and to easily interact with objects. Our constraint resolution method can be generalized to various kind of deformable models. Simulating the trunk respiration movements during respiration is nearly ended.

### Acknowledgements

We wish to thank Arleen C. Kingston and Niamh Lions for carefully rereading this paper.

### References

1. F. Aubert and D. Bechmann. Dformation d'objets volume constant. Technical report, actes de la 3me journe AFIG, 1995.
2. N. Ayache, P. Cinquin, I. Cohen, L. Cohen, F. Leitner, and O. Monga. Segmentation of complex medical objects: a challenge and a requirement for computer assisted surgery planning and performing. In R. Taylor, S. Lavallee, G. Burdea, and R. Mosges, editors, *Computer Integrated Surgery*. MIT Press, Cambridge, MA, 1996.
3. D.T. Chen and D. Zeltzer. Pump it up: computer animation of a biomechanically based model of muscle using the finite element method. In *Computer Graphics (SIGGRAPH'92)*, volume 26(2), pages 89–98, July 1992.
4. F. Ferrigno, P. Carnivalli, A. Aliverti, F. Molteni, G. Beulcke, and A. Pedotti. Three-dimensional optical analysis of chest wall motion. *Journal of Applied Physiology*, 77(3):1224–1231, 1994.
5. M.P Gascuel and J.D. Gascuel. Displacement constraints for interactive modeling and animation of articulated structures. *The Visual Computer*, 10(4):191–204, March 1994.
6. J. Kaye, F. Primiano, and D. Metaxas. Anatomical and physiological simulation for respiratory mechanics. *Journal of Guided Surgery*, 1:164–171, 1995.
7. J.C. Platt and A.H. Barr. Constraint methods for flexible models. In *Computer Graphics (SIGGRAPH'88)*, volume 22(4), pages 279–288, August 1988.
8. E. Promayon, P. Baconnier, and C. Puech. Physically-Based Deformations Constrained in Displacements and Volume. In *Computer Graphics Forum (EuroGraphics'96)*, volume 15(3), pages 155–164, August 1996.
9. A. Rappoport, A. Sheffer, and M. Bercovier. Volume-preserving free-form solids. In *Solid Modeling'95*, pages 361–372, May 1995.

10. A.R. Taylor, K. Rehder, R.E. Hyatt, and J.C. Parker. *Clinical respiratory physiology*. Saunders, 1989.
11. D. Terzopoulos, J. Platt, A. Barr, and K. Fleischer. Elastically deformable models. In *Computer Graphics (SIGGRAPH'87)*, pages 205–214, July 1987.
12. C. Van Overveld. An iterative approach to dynamic simulation of 3-D rigid-body motions for real-time interactive computer animation. *The Visual Computer*, 7:29–38, 1991.