Interactive elastic deformation of 3D images

Francisco J. R. Prados, Alejandro León Salas and Juan Carlos Torres

Grupo de Investigación en Informática Gráfica
Universidad de Granada, Spain

ABSTRACT

The standard workflow of a haptic device takes the position of the effector in space and sends a force vector in return. This paradigm naturally meets the behaviour shown in elastic materials, where the resistant force of a body is proportional to the displacements created. In fact many works have designed methods to simulate interaction with linear-elastic bodies, covering different techniques from spring-mass structures to specific volumetric representations.
However, realistic simulation of volumetric data is still an open issue. Not only is it challenging due to the complex relations that govern the real elastic behaviour, but also because of the huge amount of data that needs to be processed.
In this paper we propose a technique to interactively deform 3D images, such as those acquired by a CT scanner. While producing a physically plausible haptic feedback, deformation and visualization algorithms produce an efficient and natural feeling. Using a free form deformation structure as a wrapper, it is possible to deform complex structures at high frame rates, independently of the size of the volume.

Categories and Subject Descriptors (according to ACM CCS): I.3.5/6 [Computer Graphics]: Physically based modeling—Interaction techniques

1. Introduction

Simulation of rigid body deformation has been a field of great interest among the computer graphics community. Realistic deformation models have immediate and obvious applications in many areas, such as virtual reality, surgery simulation or virtual sculpting. Since 1984, when Sederberg et al. presented the Free Form Deformation method [SP86], a vast collection of works have aimed to create efficient and realistic methods to create the illusion of elasticity or plasticity.

However, and similarly to any other physical simulation, realism and efficiency are confronted. On the one side, physically based simulations require many resources and computation time, which consequently limits the size of the model involved. On the other side, simplified or heuristic deformations are easier to compute, but obviously produce less robust results.

The proposed method tries to find an equilibrium between these two approaches. A natural tactile experience is generated using physical principles, whereas simplifications are made in the deformation model – the huge number of elements required by volumetric models would be hard to handle otherwise.

1.1. Contribution

The main contribution of this work is a deformation technique which allows *interactive deformation of volumetric images*. User experience is furthermore enhanced with the addition of a *realistic, physically plausible* haptic feedback. Deformation, collision detection and force calculation are computed independently at *high frame rates*, ensuring a soft, continuous and natural tactile sensation. Since a wrapping Free Form Deformation structure is used, the deformation

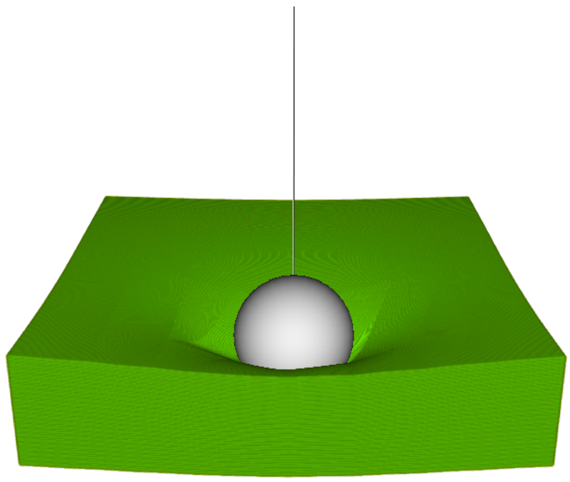

Figure 1: *Deformation of a solid block of elastic material.*

rates are *independent from the size* of the volume being deformed.

1.2. Previous Works

A number of methods have been presented aiming to create both *ad-hoc* and general solutions to the elastic problem. In this section a selection of them will be referenced, providing an interesting overview of the state of the art on deformable models and haptic interaction.

1.2.1. Geometric Deformations

Complex models with a large number of nodes usually require some kind of simplification in the deformation algorithms to achieve interactive response times. This is the case of spring-mass based solutions, which have been widely used to calculate deformations on human tissues or elastic materials [Del98]. Despite of the simplifications done, works such as those of Choi *et al.* [CSH03] and McDonnell [MQW01] have shown satisfactory results when they are combined with haptic interaction. Usually, spring-mass models require the addition of heuristic knowledge to acquire physical reality [NC10].

The *Chain-Mail Algorithm*, developed by Frisken-Gibson [FG99], makes use of a volumetric structure to compute both elastic deformations and haptic responses. For this, voxel of the volume admits a certain displacement within its original position. When the voxel reaches its maximum offset, the displacement is chained to the neighbors. The result is a linked volume which accepts geometrical deformations.

Dewaele and Cani [DC04] presented a method that made possible the global plastic deformation of a volumetric structure. Displacements fields were computed directly on the grid, using two ore more contact points.

1.2.2. Physics-based Simulation

Some researches have confronted the intrinsic problems of using physical models — which usually implies complex integration methods to compute deformations — presenting remarkable works which, with some small simplifications, have overcome the obvious time limitations.

Many works have also used *Finite-Elements Models* (FEMs) to represent deformable solids, such as the one presented by Nesme et al. [NPF05]. Also noteworthy is the work done by Delingette et al. [DA04], who used continuum mechanics to compute physically plausible deformations. Delingette proposed a FEM for data representation, which was created from the initial dataset in a precomputation stage. Haptic interaction was granted with a predefined set of interaction operators, since the complexity of the model would not allow for runtime force calculation. Peterlík [PM07] improved Delingette's method in some aspects, using the Linear Theory of Elasticity in a more efficient way.

Metzger et al. [MTPS08] proposed a method to deform a mesh by constructing a skeleton of quadratic tetrahedra. Continuum mechanics were used to compute deformation, using limited time integration schemes to grant interactivity even in models comprehending 1,000 tetrahedra.

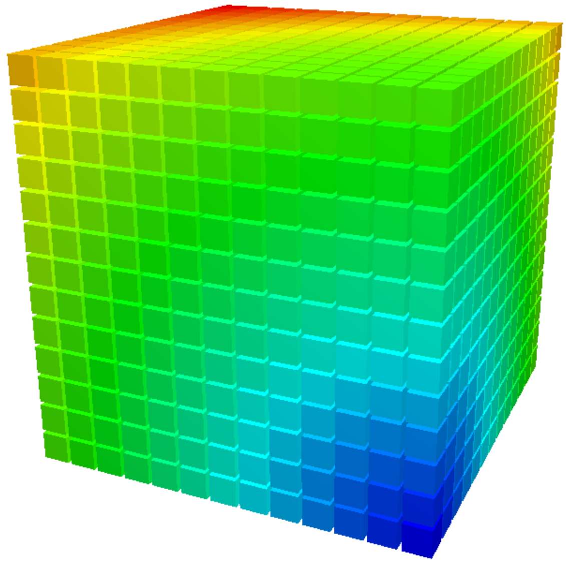

Figure 2: *Conceptual representation of the volume. Each cube represents a voxel with an uniform property value. Color gradient shows different voxels with different property values. A small spacing has been added between voxels for clarity.*

2. Volume Representation

In this section a brief description of the volumetric representation used in this work will be introduced.

2.1. Voxel Domain

The data structure used to represent the material is a finite set of homogenous cubes partitioning the three-dimensional euclidean space. Such distribution creates a grid which can be conveniently indexed with three integer numbers i, j and k. The surjective function $v(\vec{x}) : \mathbb{R}^3 \to \mathbb{Z}^3$ defines the many-to-one relation between any point of the space and its correspondent area v_{ijk}, also called *voxel*. Each voxel v_{ijk} will have an associated positive real value. This number shall be interpreted as the *occupation level*, and referred as *density* from here on. Density function reads consequently as:

$$d(v_{ijk}) : \mathbb{Z}^3 \to \mathbb{R}^+ \cup \{0\} \tag{1}$$

Therefore the volume can be seen as a set of cubes, each of them representing an uniform density area (see Fig. 2). As a natural extension of this abstraction, the space outside the volume will also be partitioned in voxels of the same size. Zero density will be assigned to any point outside the volume data. The whole object, defined as the set of all the voxels with density bigger than 0, will be noted as:

$$V = \{v_{ijk} / d(v_{ijk}) > 0\} \tag{2}$$

2.2. Tool Representation

The tool will be defined with the implicit function:

$$t(\vec{x}) : \mathbb{R}^3 \to \{v_{ijk}\} \tag{3}$$

This function provides, for any point of the space $\vec{x}$, a discrete set of voxels which are included in the tool volume.

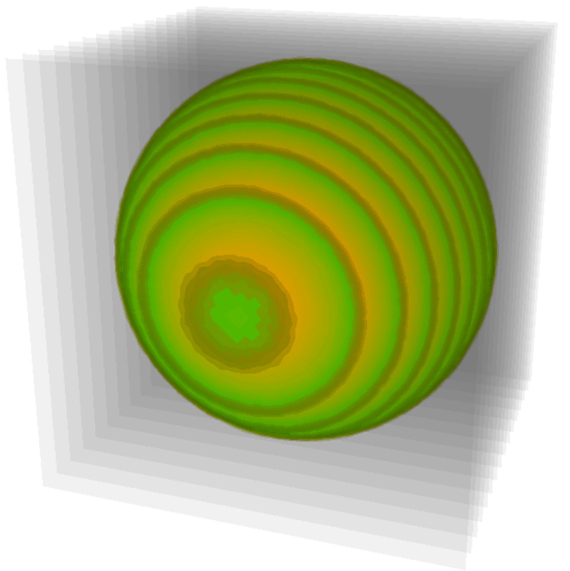

Figure 3: *Schematic representation of the Texture Hardware Volume Rendering technique. Volume is represented by a set of slices, aligned with the camera. In the picture a sphere is composed by a set of sections (circles), each one painted in one of the 16 slices (grey polygons).*

For the sake of simplicity, a spherical tool will be used in the following.

3. Volume Rendering

For later use, the visualization method used in this work — commonly known as direct volume rendering — shall be here described.

The volume rendering is done by displaying a set of polygons, aligned with the camera view. A three-dimensional texture is then assigned to those polygons, showing each of them a *slice* of the volumetric texture. The slices are composed using hardware alpha blending, therefore adding a depth sensation to the scene [MLM03]. This situation is depicted in Fig. 3.

The volume can be examined from different views angles just rotating or translating the 3D texture. The polygonal slices do not, therefore, move from their original position. Additionally, using shader programs — when supported by the underlying graphics hardware — makes possible to colour single-valued voxels. Using a colour transfer function, different RGBA values can be assigned to each of the property values of an unidimensional domain.

As it will be later pointed, this rendering technique turns out to be very convenient for the purposes of this work, precisely due to the polygonal structure in which the volume is enclosed.

4. Force feedback

The tactile sensation generated by this method is supported by the Theory of Linear Elasticity, which is a branch of the Continuum Mechanics which relates the forces applied to a material (modeled as a continua) and the displacements produced in the body. Prior to detailing the force computation algorithm, a brief note on the continuum mechanics principles used will be here introduced.

4.1. Linear elasticity

Let $\mathbf{x}$ be the spatial configuration of a material. For a certain time instant t, the relation between the external forces applied to an elastic, isotropic media reads as follows [OdSB06]:

$$\begin{aligned} \boldsymbol{\sigma}(\mathbf{x},t) &= \mathbb{C} : \boldsymbol{\varepsilon}(\mathbf{x},t) \\ \sigma_{ij} &= \mathbb{C}_{ijkl}\,\varepsilon_{kl} \end{aligned} \tag{4}$$

with $i,j,k,l \in \{1,2,3\}$, and where $\boldsymbol{\sigma}$ is the second order Cauchy stress tensor, $\boldsymbol{\varepsilon}$ is the infinitesimal strain tensor, and $\mathbb{C}$ is the fourth order stiffness tensor, which, by definition:

$$\begin{aligned} \mathbb{C} &= \lambda \mathbf{1} \otimes \mathbf{1} + 2\mu \mathbf{I} \\ \mathbb{C}_{ijkl} &= \lambda \delta_{ij}\delta_{kl} + \mu(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}) \end{aligned} \tag{5}$$

where λ and μ are the Lamé constants. Substituting Eq. 5 in Eq. 4:

$$\begin{aligned} \boldsymbol{\sigma} &= \lambda Tr(\boldsymbol{\varepsilon})\mathbf{1} + 2\mu\boldsymbol{\varepsilon} \\ \sigma_{ij} &= \lambda \delta_{ij}\varepsilon_{ll} + 2\mu\varepsilon_{ij} \end{aligned} \tag{6}$$

the *Constitutive Equation for Linear Elastic Isotropic Materials* is obtained, with tensor indexes $i,j \in \{1,2,3\}$. This equation determines, for a certain known strain tensor, the stress applied in the body. Assuming that only one force is acting on the material at each time and therefore that the body does not rotate, shear components of both the strain and stress tensors can be obviated.

4.2. Haptic force generation

In order to produce a smooth, continuous haptic response, it is necessary to generate a force vector at the common update rate of 1kHz. Therefore, collisions and elastic analysis must be computed within the strict time limit imposed by the device. Furthermore, it is necessary to compute the elastic problem taking the stress tensor $\boldsymbol{\sigma}$ as the main variable, for the haptic device only samples the probe position.

The force computation cycle naturally starts with the collision detection. Since the tool used in this work is defined as an implicit function over the volume domain as seen in Eq. 3, collision can be easily checked with the expression:

$$\sum^{t(\vec{x})} d(v_{ijk}) > 0 \tag{7}$$

for any position $\vec{x}$ of the tool in space. When the condition in Eq. 7 is met, the volumetric tool $t(\vec{x})$ intersects one or more voxels of the volume V. Contact with the material has been therefore achieved, and a force must be generated.

Let $\vec{x}_0$ be the last sampled position of the haptic tool where $\sum^{t(\vec{x})} d(v_{ijk}) = 0$, and $\vec{x}_1 \ldots \vec{x}_n$ the next sampled positions which satisfy the collision condition defined in Eq. 7. The surface of the solid will be deformed following the path of the haptic tool. Hence, the differential displacement vector $\vec{x}_n - \vec{x}_0$ will be the distance covered by the body surface as long as the haptic tool is pressing. Since the strain tensor $\boldsymbol{\varepsilon}$ is known for any point of the surface contacting the tool, and knowing that no other forces are acting on the body, $\boldsymbol{\sigma}$ can be easily computed with Eq. 6.

Once the differential stresses are known for each one of the surface units contacting the tool, and as explained in previous works of the authors [PST09], the resulting force can be computed just surface-integrating over the body surface. In the discrete expression, the equivalent total strain reads:

$$\sigma_t = \sum_S \sigma_i \tag{8}$$

which is, obviating discretization effects, approximately equal to the force $\vec{f}$ felt by the user holding the haptic probe.

5. Elastic Deformation

Resolving the elastic problem for the entire volumetric image would require solving a system with 15 partial differential equations and 15 variables in $\mathbb{R}^3 \times \mathbb{R}$ (space and time) [OdSB06]. Besides, the system should be solved each time a differential strain was found. Even though realistic results have been achieved previously with limited-time integration schemes for a small number of nodes [MTPS08], it is not possible to offer interactive frame rates when considering the huge amount of samples which conform 3D images.

Even small volumes would require unacceptable response times if all the voxels had to be processed. A cubic image with 128 voxels side would require computing more than two million points per interaction frame. This intrinsic complexity, which grows exponentially with larger volumes, naturally leads to interpolation when it comes to process the whole volume. Also, the theoretical solution, which requires a somewhat complex integration scheme, will be approximated by an heuristic geometrical transformation which creates an intuitive and natural deformation.

5.1. Free Form Deformation Enclosure

The approach proposed here uses an intermediate Free Form Deformation to properly deform the 3D image when an interaction occurs.

The Free Form Deformation technique requires a set of points which are implicitly defined in terms of a set of control nodes. The points will change accordingly to the position of the control nodes, and therefore *deforming* the object. The 3D image is rendered as a block by the graphics hardware, but the visualization method explained before does require a set of polygons, associated each one to a certain segment of the volume (see Fig. 2). If each slice is additionally fragmented in a set of rectangular pieces, the result will be a regular structure of aligned polygons which can be easily *transformed*:

$$P(u,v,w) = f(P(x,y,z)) \tag{9}$$

where each point of the polygonal mesh $P(u,v,w)$ is a function of the initial configuration $P(x,y,z)$. Each point $P(x,y,z)$ of the mesh will have associated a certain point of the 3D image $V(x,y,z)$ (assuming a continuous, interpolated 3D texture). Displacing a point $P(x,y,z)$ using the deformation function f will also displace its associated point in the volume domain $V(x,y,z)$. Consequently, the volume slices will be deformed following the pattern given by f. Obviously, the quality and time required to compute the deformation will directly depend on the number of slices used to display the volume, and also in the number of rectangular patches used for each slice.

5.2. Control Points

Since the deformation will be triggered and controlled by the haptic device, a single control node will be dynamically defined when a collision is detected. When a collision occurs in the volume domain (see Eq. 7), a virtual node will be defined in the same position $\vec{x}_0$ used for the force computation. Control node $\vec{x}_0$ will be translated to $\vec{x}_n$ as long as the collision condition is met.

All the points of the polygonal mesh will be transformed the following pattern given by the elastic deformation $\vec{x}_n - \vec{x}_0$:

$$P(u,v,w) = P(x,y,z) + \frac{(\vec{x}_n - \vec{x}_0)}{\|P(x,y,z) - \vec{x}_s\|} \tag{10}$$

where $\vec{x}_s$ is the point of the tool surface in the direction $P(x,y,z) - \vec{x}_n$. If $P(x,y,z) - \vec{x}_n$ does not intersect the tool surface then $P(x,y,z)$ is *inside* the tool and then $\vec{x}_s = P(x,y,z)$. Eq. 10 displaces all the points of the mesh uniformly in the direction of the current elastic strain $\vec{x}_n - \vec{x}_0$, whereas the module of the displacement vector for each point diminishes linearly with the distance to $\vec{x}_n$.

This effect can be clearly seen in Fig. 1, where a block of material has been pressed with the haptic tool. Points of the polygons in the intermediate structure have been displaced, and the 3D image has been coherently interpolated in the reallocated polygons.

6. Implementation and results

The force computation and elastic deformation algorithms were implemented in a prototype running on an Intel Core 2 Quad CPU at 2.93GHz with 3.50GB of RAM computer. Force computation speeds are shown in Table 1. The performance of the force computation strictly depends on the tool size, for collisions need to be checked all over the tool surface voxelization.

Table 1: *Framerates for the force computation thread*

	Tool diameter		
Volume size	16	32	64
128^3	4654Hz	566Hz	74Hz
256^3	4612Hz	567Hz	71Hz
512^3	4567Hz	560Hz	67Hz

The visual feedback of the deformation is computed in the graphical thread, for the polygonal mesh only needs to be updated once per frame. Performance in this case does only depend on the number of points that need to be transformed (i.e.: the resolution of the FFD). Screen captures shown in Figures 1, 4, 5, 6 and 7 were taken with a mesh composed by 500 slices, 20x20 square patches each (a total number of $21^2 \cdot 500 = 220500$ points). Slices are low resolution, but the high density of slices lead to acceptable visual results. A mean framerate of 15fps was obtained for most of the models, with a slight reduction in performance when working with 512^3 volumes — big texture manipulation heavily decreases performance, even in view-only applications.

7. Conclusions and future work

A new method for volumetric elastic simulation was here introduced. The technique described here allows interactive

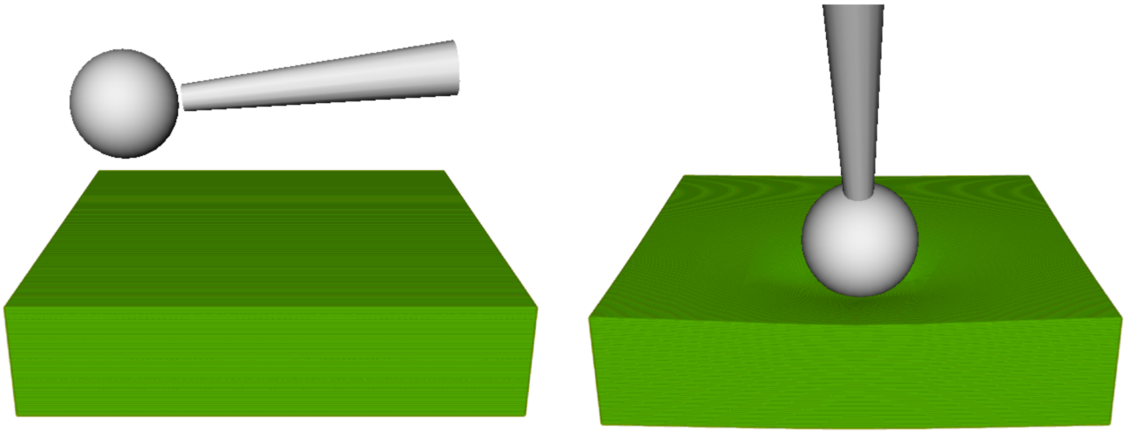

Figure 4: *Elastic deformation of a volume representing a solid block of an homogeneous material. Left: initial status. Right: haptic tool posed on the material (gravity effect). A heavier deformation of this model was shown in Fig. 1.*

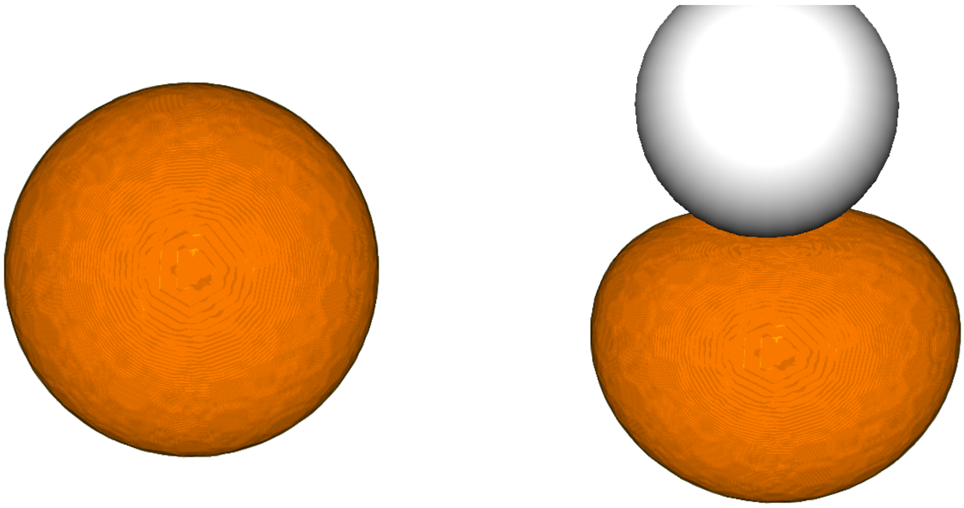

Figure 5: *Elastic deformation of a volumetric sphere.*

deformation of a volume, such as the 3D images obtained with a CT scan. The volume is treated as an elastic, isotropic medium. Force feedback is computed following continuum mechanics principles, which leads to a natural and plausible tactile sensation.

Future improvements of the described technique comprises the addition of permanent (plastic) deformations, topic which has been already developed for volumes in previous works by the authors [PST09]. Also, it would be interesting to expand the current method to work with anisotropic materials, for 3D images can represent different properties (as clearly seen in the CT scan depicted in Fig. 7).

8. Acknowledgments

This work has been partially funded by the Spanish Ministry of Science and Technology and the EU FEDER funds (project TIN2007-67474-C03-02). It has also been funded by the Andalusian Government under project PE-TIC-5276.

References

[CSH03] Choi K.-S., Sun H., Heng P.-A.: Interactive deformation of soft tissues with haptic feedback for medical learning. *Information Technology in Biomedicine, IEEE Transactions on 7*, 4 (Dec. 2003), 358–363. 2

[DA04] Delingette H., Ayache N.: Soft tissue modeling for surgery simulation. In *Computational Models for the Human Body*, Ayache N., (Ed.), Handbook of Numerical Analysis (Ed : Ph. Ciarlet). Elsevier, 2004, pp. 453–550. 2

[DC04] Dewaele G., Cani M.-P.: Virtual clay for direct hand manipulation. In *Eurographics (short papers)* (2004). 2

[Del98] Delingette H.: Towards realistic soft tissue modeling in medical simulation. *Proceedings of the IEEE : Special Issue on Surgery Simulation* (April 1998), 512–523. 2

[FG99] Frisken-Gibson S. F.: Using linked volumes to model object collisions, deformation, cutting, carving, and joining. *IEEE Transactions on Visualization and Computer Graphics 5*, 4 (1999), 333–348. 2

[MLM03] Ma K.-L., Lum E. B., Muraki S.: Recent advances in hardware-accelerated volume rendering. *Computers & Graphics 27*, 5 (2003), 725 – 734. 3

[MQW01] McDonnell K. T., Qin H., Wlodarczyk R. A.:

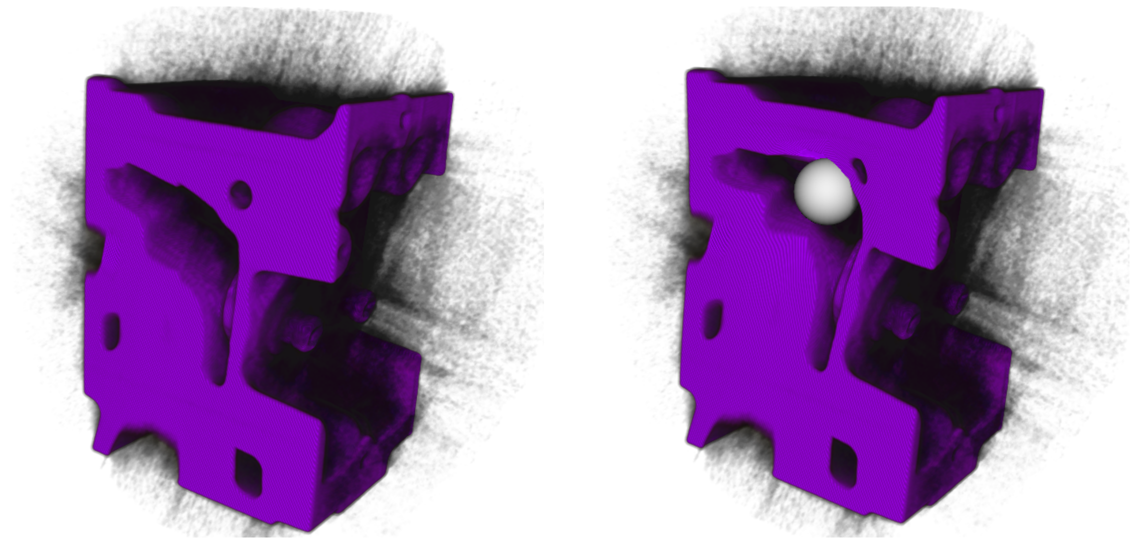

Figure 6: *Elastic deformation of a CT scan of a motor (image courtesy of General Electric USA).*

Figure 7: *Elastic deformation of a CT scan of a foot. The volumetric image is considered as an isotropic elastic material - i.e.: elastic properties do not change all along the body. Left: relaxed state. Right: interaction with the haptic probe. (Image courtesy of Philips Research, Hamburg, Germany).*

Virtual clay: a real-time sculpting system with haptic toolkits. In *I3D '01: Proceedings of the 2001 symposium on Interactive 3D graphics* (New York, NY, USA, 2001), ACM Press, pp. 179–190. 2

[MTPS08] MEZGER J., THOMASZEWSKI B., PABST S., STRASSER W.: Interactive physically-based shape editing. In *Proceedings of the 2008 ACM symposium on Solid and physical modeling* (New York, NY, USA, 2008), SPM '08, ACM, pp. 79–89. 2, 4

[NC10] NATSUPAKPONG S., CENK M.: Determination of elasticity parameters in lumped element (mass-spring) models of deformable objects. *Graphical Models 72*, 6 (2010), 61 – 73. 2

[NPF05] NESME M., PAYAN Y., FAURE F.: Efficient, physically plausible finite elements. In *Eurographics (short papers)* (august 2005), Dingliana J., Ganovelli F., (Eds.). 2

[OdSB06] OLIVELLA X. O., DE SARACÍBAR BOSCH C. A.: *Mecánica de medios continuos para ingenieros*. Universitat Politècnica de Catalunya, 2006. 3, 4

[PM07] PETERLIK I., MATYSKA L.: An algorithm of state-space precomputation allowing non-linear haptic deformation modelling using finite element method. In *WHC '07: Proceedings of the Second Joint EuroHaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems* (Washington, DC, USA, 2007), IEEE Computer Society, pp. 231–236. 2

[PST09] PRADOS F. J. R., SALAS A. L., TORRES J. C.: Elastoplastic material deformation with haptic feedback: a physically correct method. In *SIACG 2009 - IV Symposium Iberoamericano de Computación Gráfica* (6 2009). 4, 5

[SP86] SEDERBERG T. W., PARRY S. R.: Free-form deformation of solid geometric models. *SIGGRAPH Comput. Graph. 20* (August 1986), 151–160. 1