Robust and Scalable Interactive Freeform Modeling of High Definition Medical Images

Noura Faraj[1], Jean-Marc Thiery[1], Isabelle Bloch[1], Nadège Varsier[2], Joe Wiart[2], and Tamy Boubekeur[1]

[1]Télécom ParisTech, CNRS LTCI, [2]Orange Labs and [1,2]Whist Lab

Abstract. Whole-body anatomically correct high-resolution 3D medical images are instrumental for physical simulations. Unfortunately, only a limited number of acquired datasets are available and the scope of possible applications is limited by the patient's posture. In this paper, we propose an extension of the interactive cage-based deformation pipeline VoxMorph [1], for labeled voxel grids allowing to efficiently explore the space of plausible poses while preserving the tissues' internal structure. We propose 3 main contributions to overcome the limitations of this pipeline: (i) we improve its robustness by proposing a deformation diffusion scheme, (ii) we improve its accuracy by proposing a new error-metric for the refinement process of the motion adaptive structure, (iii) we improve its scalability by proposing an out-of-core implementation. Our method is easy to use for novice users, robust and scales up to 3D images that do not fit in memory, while offering limited distortion and mass loss. We evaluate our approach on postured whole-body segmented images and present an electro-magnetic wave exposure study for human-waves interaction simulations.

1 Introduction

Despite the increasing number of medical simulations performed on high resolution whole-body 3D images, only a small number of such models are available and their use is limited by the unavoidable upright acquisition position and unwanted links between body parts (e.g. hand and hip). Our goal is to perform deformations of these datasets – possibly made of hundreds of millions of voxels – interactively while providing a suitable input for physical simulations. In this context, we need the deformations to be detail- and volume-preserving. In this paper, we propose an extension the VoxMorph [1] interactive deformation pipeline. We improve the quality of the deformation, we extend the scope of usable datasets by making the pipeline robust and scalable to datasets that do not fit in memory.

1.1 VoxMorph in a nutshell

In order to preserve the details and the volume, the deformation needs to be quasi-conformal and stretch-minimizing properties when looking at their mathematical expression. VoxMorph [1] copes with these requirements, while preserving the interactivity constraint, by using a 3-scales deformation algorithm with a suitable deformation method at each scale. VoxMorph offers an intuitive way to control the deformation by the means of a coarse bounding polygonal mesh - a *cage*. At coarse scale, a high-quality non-linear variational method allows to control the cage deformation using a few vertex constraints only while solving for all others in an As-Rigid-As-Possible (ARAP) fashion [2]. At mid-scale, the space inside the cage is deformed using a linear quasi-conformal space deformation method, i.e. the Green Coordinates (GC) [3] which locally preserve angles and distances. The combination of an ARAP cage manipulation and GC space deformations offers high quality volume deformations. Deforming the complete voxel grid using GC is still prohibitive; this problem is solved at the third scale through a linearization scheme of the deformation using a tetrahedral structure which is aware of the grid topology and a defined through a refinement strategy adapting the structure resolution to the deformation. A final rasterization step transfers efficiently the interactively defined cage deformation to the high resolution 3D image.

The VoxMorph system has so far three main limitations: first, the entire model has to **strictly** fit in the controlling cage which makes its construction very difficult either in an automatic and supervised context [3]. Second, when transferring the deformation from the coarse scales to the finer one, the adaptive metric plays a critical role in the deformation quality which can be improved. Last, although VoxMorph is able to deform large images, it remains restricted to in-core datasets limiting the scope of usable data since the increasing accuracy of modern 3D acquisition systems generate huge high definition datasets.

1.2 Overview

Consequently, we improve this pipeline, making VoxMorph more robust, accurate and scalable. First, we improve the robustness by proposing a deformation diffusion scheme which allow us to offer a limb separation method and to deform parts of the model that are outside the cage. Second, we improve the accuracy by proposing a new error metric which avoid refining to deeply the transission structure between mid and high resolution data. Third, we propose an out-of-core implementation in order to scale up to datasets that do not fit in memory. As a result, our new system can handle a larger range of datasets. We validate our pipeline by performing digital dosimetry analysis for radio-wave effects studies on high resolution segmented 3D medical images posed with our approach. All the representations in this pipeline are generated with our system.

2 Background

The deformation of volume datasets are often performed using space deformation methods. The main idea is to control interactively the deformation of a low resolution closed 3D object – the *cage* – and transmit it to the embedded space using some form of generalized barycentric coordinates. The deformation of whole-body medical models for *Specific Absorption Rate* (SAR) analysis is

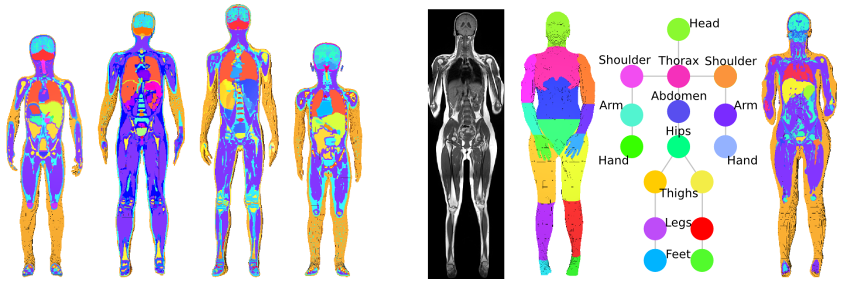

Fig. 1. Typical examples of input whole-body segmented voxel grids.

often done using the original free form deformation definition [4], as done by Nagaoka et al. [5]. Recent work in computer graphics improves the quality of such volume deformation. In particular, GC are computed from the integral of the Laplacian Green function on the boundary of the domain which tends to better preserve details. Volume deformations using skeleton-based methods have been proposed by using a dummy model [6] or the interface between the grid and the background [7] to transmit the deformation to the volume. Both methods offer angular control over the joints but need a tedious per model pre-process. Despite the existence of these techniques, most medical image deformations are still performed by cutting and pasting limbs along with a tedious manual adjustment at the junctions [8].

3 Image data and segmented volumes

The input of our system are medical images and the output are images containing interactively deformed models. These images are segmented and the derived models are posed to perform the SAR analysis. Whole-body *magnetic resonance images* (MRI) of children have been acquired thanks to collaborating hospitals. Depending on the patient, around 32 coronal slices are acquired with a slice thickness of $6mm$. The reconstructed voxel size for all images is $1.3 \times 1.3 \times 7.2\ mm^3$. This strong anisotropy causes the data to exhibit a lot of partial volume effects. Due to the use of multiple coils, the images actually result from the composition of 4 or 5 images (depending on the patient's height), and some artifacts may appear such as missing parts due to field size or lower intensity at the transition between two images. To tackle the problems emerging from the resolution and the position of the patient (e. g. often the patient had his hands leaning on his thigh or the arms stuck to the thorax in the armpit region – Fig. 1 right), a body part segmentation is performed, as described in [9]. Additionally, we use whole-body children models from the Virtual population project [10]. These models, composed of 1mm cubic voxels, are highly detailed with up to 84 represented organs (Fig. 1 left). To demonstrate the scalability of our approach, we also use the Visible Human dataset [11].

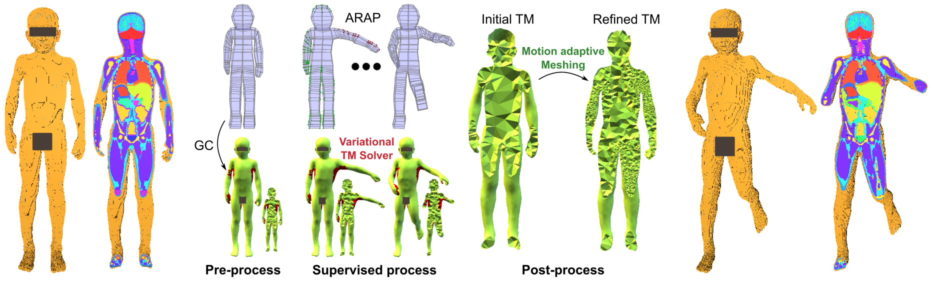

Fig. 2. Pipeline overview The input is a segmented voxel grid. We construct a cage and a tetrahedral mesh (TM) with segmentation-aware connectivity from it. The user manipulates the cage – deforming the TM vertices that are inside it using cage coordinates – and the tetrahedral solver computes the position of the outlier vertices interactively. The deformation is transferred from the TM to the grid in real-time. The adaptive resolution of the TM is driven by the deformation.

Segmentations A segmented medical image is represented by a 3D voxel grid containing the discrete labels assigned to the image. By convention, null values represent the background. In this paper, we note $\mathbb{G}$ the grid representing the organ segmentation and $\mathbb{S}$ the one representing the body part segmentation. If none is available or necessary (i. e. no limb separation needed), we use a binary voxel grid with $\mathbb{S}[v] = 0$ for null voxels and $\mathbb{S}[v] = 1$ otherwise. Note that the user can create a rough segmentation interactively.

4 A Robust and Scalable Deformation Method

The cage can very often cross the model, excluding part of the voxels from the space deformation; we tackle this problem by designing a new variational optimization technique which conveys the inner space deformation to outer voxels. To improve the quality of the deformation, we propose a new error metric to guide the motion adaptive refinement process. Finally, to make the pipeline scalable, we propose an out-of-core deformation process for the high resolution voxel grid. Our general pipeline is illustrated in Fig. 2.

4.1 Robust Cages

To initialize the system, the user supervises a morphological cage creation process acting on $\mathbb{S}$. In the VoxMorph [1] system, a uniform dilation of $\mathbb{S}$ is performed. Then, a high resolution 2D restricted Delaunay triangulation (RDT) is generated, capturing the interface between the resulting grid and the background. To do so, Boissonnat et al. 's refinement process [12] is applied and the result is simplified to a prescribed resolution (typically a few tens or hundreds of vertices) using the *Quadric Error Metric* [13]. This process can create unwanted links between limbs (e.g. armpit, hand and hip) and therefore generate a cage with an incorrect topology. In our system, we propose a limb separation supervision process by offering the possibility to create inset cages: the user selects a region where two limbs need to be separated, then we automatically erode the voxels located at their interface while adaptively dilating elsewhere. Finally, the aforementioned surface creation and simplification process is applied.

4.2 Accurate Motion-Adaptive Structure

GC require the cage to enclose the model to deform, making its creation extremely difficult even for expert users, as well as for automatic methods. Moreover, the full deformation of $\mathbb{G}$ using GC can be prohibitive (e.g., several hours for the models we present in Sec. 5). More precisely, Lipman et al. explained how to extend linearly the coordinates *through* a set of faces of the cage, making the use of partial cages possible. In their setup, the system has to identify the set of coordinates that are *valid*, and the values for the other coordinates are found by inverting a linear system. This approach requires the user to specify manually the faces that need to be extended. More importantly, we need the extension of the deformation to help separating *distinct glued limbs*, and therefore it has to be driven by a *space separation method* similar to the use of a cage. Consequently, the direct use of the extension of the coordinates through facets of the cage [3] is impossible in our context.

In VoxMorph, the scalability issue is solved using a volumetric tetrahedral structure based on $\mathbb{G}$: the *Transfer Mesh* (TM), which is built to enclose the volume to process and has a resolution adapted to the deformation. We propose to apply a deformation diffusion method performed on this structure to solve the robustness problem. We denote $TM = \{V, E, T\}$ with $V = \{v_i\} \subset \mathbb{R}^3$ its vertices, $E = \{e_{ij}\}$ its edges connecting adjacent vertices v_i and v_j,and $T = \{t_k\}$ its tetrahedra.

All TM vertices that are inside the cage are deformed using GC, and all others' positions are recovered by minimizing a *rigidity energy* on TM (Fig. 3). The space deformation is transmitted to the voxel grid using barycentric interpolation, which guarantees *consistent rasterization* of the target voxel grid, and allows real-time deformations of $\mathbb{G}$.

Finally, to cope with the approximations introduced by the linearization scheme, we propose a new error metric for the iterative refinement strategy of the TM, making its structure adaptive to the on-going deformation.

In the following, $T_1(v_i)$ denotes the set of tetrahedra adjacent to a vertex v_i, $T_1(e_{ij})$ the set of tetrahedra adjacent to an edge e_{ij}, and $T_1(t)$ the set of tetrahedra sharing a face with a tetrahedron t. We note B_t the 3×3 transformation matrix of each tetrahedron t, uniquely defined by the transformation of its 4 vertices, $|t|$ its volume before deformation, and $\triangle\ (B_t)$ the local change of the transformation matrices expressed as:

$$\triangle\ (B_t) = B_t - \frac{\sum_{t_n \in T_1(t)} |t_n| \cdot B_{t_n}}{\sum_{t_n \in T_1(t)} |t_n|}. \tag{1}$$

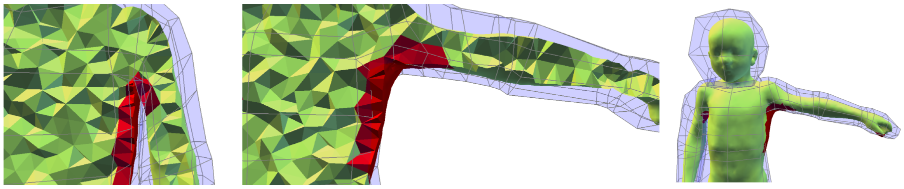

Fig. 3. Deformation of the outlier vertices: the green tetrahedra are deformed using GC, the red tetrahedra's geometry – partially located outside the cage – is recovered by minimizing e_{rigid}.

We model the energy to minimize as:

$$e_{rigid} = \sum_{t} |t| \cdot || \bigtriangleup B_t||^2. \tag{2}$$

using the Frobenius norm.

Set up We construct TM as a restricted adaptive Delaunay multi-material *tetrahedrization* generated from $\mathbb{S}$, the limb segmentation associated with $\mathbb{G}$, and constrained by a *sizing field* [14], which allows to control the spatially varying tetrahedron size explicitly. The sizing field is stored in a grid – noted $\mathbb{F}$ – with the same dimension as $\mathbb{G}$ and initialized to a uniform value (5% of the voxel grid's diagonal in our experiments). Depending on the segmentation, the resulting mesh can be composed of several subdomains, therefore a label l_k is associated with each tetrahedron.

Label Separation We use the limb segmentation to split TM in user-defined regions in order to remove unwanted limb connections. The user draws a selection area and provides a pair of labels to separate. In this area, we duplicate the vertices of TM that belong to a face common to two tetrahedra to separate,i. e. one labeled in the first subdomain and the other in the second. We add the new vertices to the mesh, and re-index the tetrahedra in the second subdomain over them.

Improving Robustness - Tetrahedral Solver The minimization of e_{rigid} is performed in two steps. We note T_c the tetrahedra of TM whose four vertices are inside the cage (green in Fig. 3), and V_c the vertices of TM that are inside the cage.

In the first step, we compute B_{t_c} for all $t_c \in T_c$ (these are deformed using GC) and set them as constraints, that we note $\tilde{B}_{t_c}$. We then set $\bigtriangleup (B_{t_u})$ to be the zero 3×3-matrix for all others (red in Fig. 3). This is done by solving the following linear system in the least squares sense, with B_t its unknowns:

$$\begin{cases} \sqrt{|t_c|} \cdot B_{t_c} = \sqrt{|t_c|} \cdot \tilde{B}_{t_c} & \forall t_c \in T_c \\ \sqrt{|t_u|} \cdot \bigtriangleup (B_{t_u}) = 0_{33} & \forall t_u \in T \setminus T_c \end{cases} \tag{3}$$

The result is a set of transformation matrices B_t for all $t \in T \setminus T_c$.

In the second step, we recover the vertex positions of TM, using the transformation matrices of the tetrahedra. We add the positions of all vertices in V_c as constraints, and set the edges to be the initial ones transformed by B_t. This is done by solving the following linear system in the least squares sense:

$$\begin{cases} \sqrt{\sum_{t \in T_1(v_c)} |t|} \cdot v_c &= \sqrt{\sum_{t \in T_1(v_c)} |t|} \cdot \tilde{v}_c \\ \sqrt{\sum_{t \in T_1(e_{ij})} |t|} \cdot (v_i - v_j) &= \frac{\sum_{t \in T_1(e_{ij})} |t| \cdot B_t \cdot (v_i^{init} - v_j^{init})}{\sqrt{\sum_{t \in T_1(e_{ij})} |t|}} \end{cases} \tag{4}$$

$\forall v_c \in V_c$ and $\forall e_{ij} \in E$. The result is a set of positions for all vertices of TM that were outside the cage during the embedding. The linear systems are factorized at the creation of TM and solved efficiently at each frame using a linear algebra library [15].

Improving the Quality - New Error-Metric To cope with the piecewise linear approximation introduced by TM, we make the sizing field adaptive to the deformation by refining TM iteratively.

At each step, we define an error e^T for each tetrahedron t as

$$e_t^T = || \vartriangle B_t||, \tag{5}$$

and an error e^V for each vertex v as

$$e_v^V = \frac{1}{\sum_{t \in T_1(v)} |t|} \sum_{t \in T_1(v)} |t| e_t^T \tag{6}$$

the mean of the errors of its adjacent tetrahedra. We then obtain a smooth *error metric* e^G on the initial grid by rasterizing each tetrahedron, filling up the grid with the barycentric interpolation of its vertices' errors. The sizing field is updated according to the error metric with the following rule:

$$\mathbb{F}[v] *= \sqrt[3]{\frac{e_{mean}}{\max(e^G(v), e_{mean})}}. \tag{7}$$

with e_{mean} the grid mean error. This process allows to obtain a uniform error on the grid at convergence. This new metric allows to reduce the final resolution by about 40% on average compared to the original VoxMorph system.

4.3 Scalable out-of-core Deformation Upsampling

In the VoxMorph [1] system, the deformation of the grid is performed in-core by computing the barycentric coordinates of the center of all the deformed grid voxels contained in the deformed tetrahedra. Using these coordinates and the initial TM vertex positions, we find their initial positions and project them in the initial grid to get the labels to assign to the final grid. In order to process out-of-core datasets, we propose to perform the entire pipeline on a low resolution version of the grid, which is down-sampled on the fly, and to apply an out-of-core rasterization of the high resolution one. We implemented a streaming process to perform the per-voxel deformation, therefore, both the initial and deformed high resolution voxel grids do not need to be loaded in memory. We use the TM of the low resolution image to assign the label to the high resolution one. To do so, we apply the previously described process to all the voxels of the deformed high resolution grid contained in the deformed tetrahedra and fetch in memory the voxels values where the positions project and directly write the result in a file.

5 Results

Deformation evaluation We implemented our system in C++ with OpenGL for rendering. We used CHOLMOD [15] as a Cholesky solver, GSL [16] for the SVD and the CGAL library [14] for tetrahedral mesh generation, surface meshing and simplification. Performances were measured on an Intel Core2 Duo at 2.4 GHz with 8GB of main memory and an nVidia Quadro FX580 device.

Fig. 4 illustrates the results obtained with our method for 5 high-resolution whole-body segmented images. The deformation of the Visible Human is performed out-of-core. The 7 postures of the Thelonious model have been interactively designed using our system and are further used to perform dosimetry analysis. The highest weight variation is -7% for Posture 5, which is still acceptable. For this posture, which is supposedly the worst case, the blue histograms illustrate the quasi-conformal nature of the resulting deformation. For all other postures, the difference is less than 1%. Table 1 summarizes various performance measures of our system on models ranging from 122 to 260 million voxels in core and one out-of-core deformation resulting in a grid of 862 million voxels. Note that, in all cases, the framerate (FPS) is limited by the rendering capabilities and not by our deformation workload.

Model	Voxels	CV	FPS	TV	Outliers	CT
Thelonious	122M	303	11.	36 157	393	2m31
Eartha	243M	479	9.	33 308	1890	3m10
Louis	260M	512	7.	27 588	2	4m03
Dizzy	141M	479	10.	29 226	532	2m38
LR Visible Human	108M	152	11.	17 036	1317	1m28
HR Visible Human	*374M*	-	-	-	-	*11m33*

Table 1. Performance table: with CV (resp. TV) the cage (resp. transfer mesh) vertex count, FPS the framerate during interaction and CT the final full deformation pre- and post-process time. The last row shows the out-of-core processing time to deform the High-Resolution Visible Human model, using the Low-Resolution version for the user interaction process, into a grid of 862 million voxels.

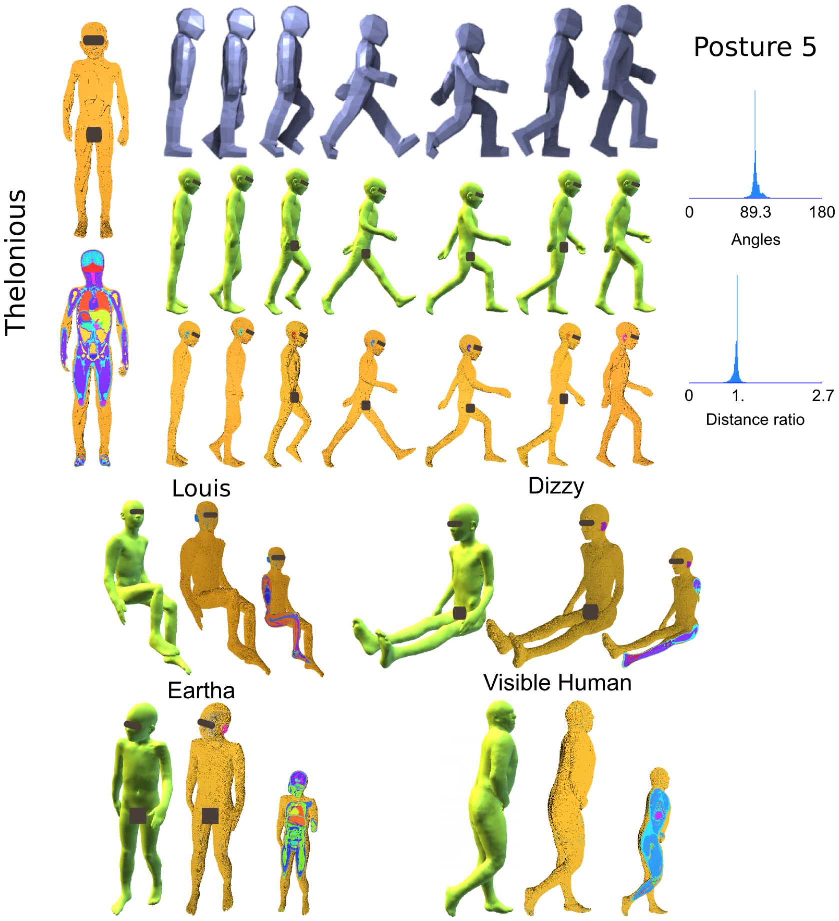

Fig. 4. Whole-body deformations The top part illustrates 7 poses simulating a walk animation on the Thelonious model [10]. The first row shows the cage controlled by the user, the second row the automatic motion adaptive TM and the last row the resulting, full resolution segmented voxel grids after deformation. The deviation from the quasi-conformal deformation is computed for the fifth position which has the highest volume change. The bottom part shows 4 other HD 3D images deformed with our system, the Visible Human deformation was performed out-of-core. Note that all these models contain outliers, and therefore the tetrahedral solver was required to deform them.

5.1 Dosimetry analysis

Materials and numerical simulation conditions We simulated a far-field exposure of the walking Thelonious, consisting of more than 70 different tissues with a 2 mm resolution, using an incident plane wave polarized vertically, with a left side incidence, emitting at the frequency of 2100 MHz. We used the well-known Finite Difference Time-Domain (FDTD) method [17] to evaluate the

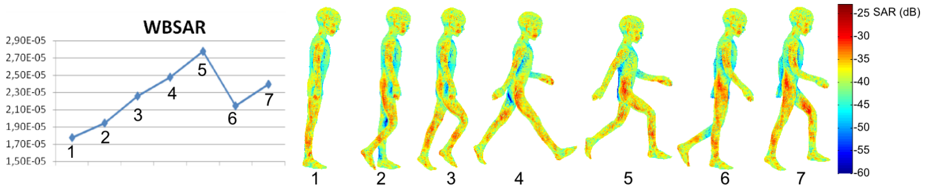

Fig. 5. (left) WBSAR calculated for E = 1 V/m. (right) SAR distribution evolution while Thelonious is walking.

whole-body exposure of Thelonious while walking for 7 different postures (see Fig. 5) and to calculate the SAR. The SAR, expressed in W/kg, quantifies the exposure to EMFs and represents the power absorbed per mass unit of tissue.

Radiofrequence exposure variations of a walking child Fig. 5 shows that the localization of the maximum SAR depends on the posture. We calculated, for each model, the whole-body averaged SAR (WBSAR) in order to analyze the influence of the posture. The whole-body averaged SAR is equal to the whole power absorbed by the numerical model divided by the body weight. This was computed for an incident field E = 1 V/m (which corresponds to a Surface Power Density of 2.65 e-3 W/m2). Fig. 5 plots the variations of the WBSAR with the posture of Thelonious. We can observe, as expected, that the WBSAR is proportional to the cross-section: from Posture 1 to Posture 5 the cross-section increases and so does the WBSAR. Then it decreases from Posture 5 to Posture 6 and increases again at Posture 7.

6 Conclusion and Future Work

We proposed an extension of the cage-based 3D image deformation pipeline VoxMorph [1], allowing to deform high-resolution whole-body anatomically correct models interactively. We made the pipeline robust by relaxing the constraint of enclosing cages and proposed a separation scheme. We improved the quality of the deformation by proposing a new error-metric for the refinement process of the motion adaptive structure. Finally, we solved the scalability issue by proposing an out-of-core deformation scheme. As a result, our system extended the scope of usable datasets, to deform in-core high resolution voxel grids of over 260 million voxels within less than 4 minutes and to perform out-of-core deformations of the 3D images that do not fit in memory, e.g. 862 million voxels. The resulting deformed images are well suited for physical simulations, as well as variability and uncertainty studies. In particular, we used it to for SAR analysis. Future work includes feed the results of our deformation pipeline to a physically-based deformation system.

Acknowledgment. This work was partially funded by ANR KidPocket, ANR FETUS and EU REVERIE IP projects. Medical images are taken from the *Virtual Population* and *Visible Human* projects.

References

1. Faraj, N., Thiery, J.M., Boubekeur, T.: Voxmorph: 3-scale freeform deformation of large voxel grids. Computers & Graphics **36**(5) (2012) 562 – 568 Shape Modeling International (SMI) Conference 2012.
2. Sorkine, O., Alexa, M.: As-rigid-as-possible surface modeling. In: Proceedings of the fifth Eurographics symposium on Geometry processing, Aire-la-Ville, Switzerland, Switzerland, Eurographics Association (2007) 109–116
3. Lipman, Y., Levin, D., Cohen-Or, D.: Green coordinates. ACM Trans. Graph. **27** (August 2008) 78:1–78:10
4. Sederberg, T.W., Parry, S.R.: Free-form deformation of solid geometric models. SIGGRAPH Comput. Graph. **20** (August 1986) 151–160
5. Nagaoka, T., Watanabe, S.: Postured voxel-based human models for electromagnetic dosimetry. Physics in Medicine and Biology **53**(24) (2008) 7047
6. Gao, J., Munteanu, I., Müller, W.F.O., Weiland, T.: Generation of postured voxel-based human models for the study of step voltage excited by lightning current. Advances in Radio Science **9** (2011) 99–105
7. Nagaoka, T., Watanabe, S.: Voxel-based variable posture models of human anatomy. **97**(12) (December 2009) 2015–2025
8. Na, Y.H., Zhang, B., Zhang, J., Caracappa, P.F., Xu, X.G.: Deformable adult human phantoms for radiation protection dosimetry: anthropometric data representing size distributions of adult worker populations and software algorithms. Physics in Medicine and Biology **55**(13) (2010) 3789
9. Fouquier, G., Anquez, J., Bloch, I., Falip, C., Adamsbaum, C.: Subcutaneous adipose tissue segmentation in whole-body MRI of children. In: CIARP 2011. Volume LNCS 7042., Chili (November 2011) 97–104
10. Christ, A., Kainz, W., Hahn, E.G., Honegger, K., Zefferer, M., Neufeld, E., Rascher, W., Janka, R., Bautz, W., Chen, J., Kiefer, B., Schmitt, P., Hollenbach, H.P., Shen, J., Oberle, M., Szczerba, D., Kam, A., Guag, J.W., Kuster, N.: The virtual family-development of surface-based anatomical models of two adults and two children for dosimetric simulations. Physics in Medicine and Biology **55**(2) (2010) N23
11. Spitzer, V., Ackerman, M.J., Scherzinger, A.L., Whitlock, D.: The visible human male: a technical report. J Am Med Inform Assoc **3**(2) (1996) 118–130
12. Boissonnat, J.D., Oudot, S.: Provably good sampling and meshing of surfaces. Graph. Models **67** (September 2005) 405–451
13. Garland, M., Heckbert, P.S.: Simplifying surfaces with color and texture using quadric error metrics. In: IEEE Visualization'98. (1998) 263–269
14. : CGAL, Computational Geometry Algorithms Library http://www.cgal.org.
15. Chen, Y., Davis, T.A., Hager, W.W., Rajamanickam, S.: Algorithm 887: Cholmod, supernodal sparse cholesky factorization and update/downdate. ACM Trans Math Softw **35**(3) (2008) 1–14
16. Contributors, G.P.: GSL - GNU scientific library - GNU project - free software foundation (FSF). http://www.gnu.org/software/gsl/ (2010)
17. Taflove, A., Hagness, S.C.: Computational Electrodynamics: The Finite-Difference Time-Domain Method 2nd edition. Norwood, MA: Artech House (2000)