

# VoxMorph: 3-Scale Freeform Deformation of Large Voxel Grids

Noura Faraj, Jean-Marc Thiery and Tamy Boubekeur  
Telecom ParisTech - CNRS

---

## Abstract

We propose VoxMorph, a new interactive freeform deformation tool for high resolution voxel grids. Our system exploits cages for high-level deformation control. We tackle the scalability issue by introducing a new 3-scale deformation algorithm composed of a high quality as-rigid-as possible deformation at coarse scale, a quasi-conformal space deformation at mid-scale and a new deformation-adaptive local linear technique at fine scale. The two first scales are applied interactively on a visualization envelope, while the complete full resolution deformation is computed as a post-process after the interactive session, resulting in a high resolution voxel grid containing the deformed model. We tested our system on various real world datasets and demonstrate that our approach offers a good balance between performance and quality.

---

## 1. Introduction

The fidelity of 3D digital models, obtained from real objects, is increasing with the volumetric acquisition systems' accuracy. Beyond their use for medical diagnostics, they also allow to perform physical simulations (e.g., digital dosimetry for radiowave effects studies) and are instrumental in body-centric engineering design (e.g., prosthesis design) and virtual reality (e.g., surgery training). These new applications all exploit the full volumetric extend of such data, beyond the classical surface capture setups. In practice, these datasets are generated via magnetic resonance imaging (MRI) or scanners and are represented as large, high resolution voxel grids (also called 3D images). In particular, an increasing number of high resolution full body 3D medical images have been generated recently, such as in the context of the Visible Human and Virtual Population projects. These models may be provided in raw format (i.e., 3D scalar or vector fields) or undergo specific segmentation algorithms, leading to quantized voxel grids indexing  $k$  components (e.g., one per organ) as illustrated in Figure 1.

Unfortunately, high quality deformation methods are usually not linear and require costly iterative solving processes, even on small models. Moreover, they are often defined on surface or volume meshes and cannot be applied directly to voxel grids. In many scenarios, offline deformation computation is not an option – since the user might need to explore the space of possible shapes interactively.

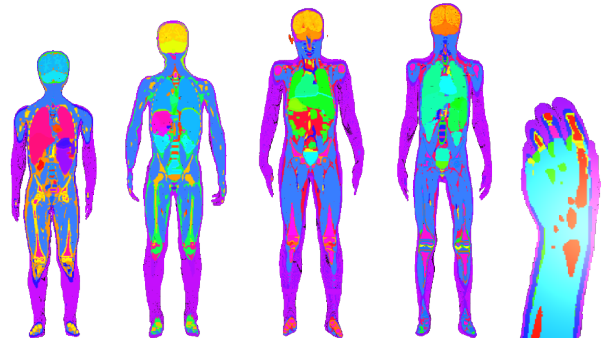


Figure 1: Typical examples of input segmented voxel grids.

In this paper, we propose a new interactive system, called *VoxMorph*, that allows to perform quasi-conformal deformations of high resolution voxel grids with a low computation time. The goal is to provide high quality results suitable for physical simulation such as SAR measures on real datasets (i.e., segmented full human body images) for various poses or as a plausible input for heavier, physics-based deformation frameworks. The quasi-conformality and the scalability of the deformation are critical properties of VoxMorph, especially in the context of medical data. Applying deformations to typical high-resolution datasets (several hundreds of millions of voxels) would take several hours, even on a powerful work station. Therefore, we introduce a deformation adaptive linearization scheme, that reduces the full processing time by several orders of

magnitude, enabling the typical try-and-test loop unavoidable in any modeling session. Furthermore, our system behaves intuitively by preserving local geometric features and global shape structures, while providing a real time feedback to control the deformation.

From the user’s point of view, VoxMorph is a *cage-based* freeform deformation system, exploiting a simple (coarse) cage embedding the voxel model. This cage acts as the unique deformation control interface and is classically represented as a closed polygonal surface mesh. The shape can be edited by moving selected handles on the cage and freezing others.

From a technical point of view, VoxMorph is a **3-scale** deformation algorithm using suitable deformation methods at different scales. At coarse scale, a non-linear variational method allows to control the cage’s shape using a few vertex constraints only while solving for all others in an *as-rigid-as possible* fashion. Such a behavior is known to be the most intuitive in interactive shape deformation [1, 2]. As the cage vertex count is typically low (e.g., a few hundreds), real time performances are preserved. At medium scale, a linear quasi-conformal space deformation method is used to transfer the cage modification to its inner space while locally preserving angles and distances. Here, the deformation can be evaluated smoothly on several tens of thousands of points interactively. At fine scale, all the actual input voxels need to be deformed but computing space deformation coordinates for each single voxel is prohibitive. Therefore we propose a new motion-adaptive linear approximation by the means of a tetrahedral *transfer mesh*, deformed at medium scale, and defining a linear (barycentric) deformation on a per-tetrahedron basis. This adaptive approximation provides a fast rasterization process for the input full resolution model in the transformed space defined by the cage deformation. Although we illustrated this paper using regular voxel grids, our method is cage-based and therefore compatible with any discrete volumetric data (e.g., adaptive, point-sampled and even meshes).

## 2. Background on Interactive Shape Deformation

Interactive freeform deformation (FFD) methods can be classified into two main categories: surface deformation and space deformation.

*Surface Deformation.* Linear variational surface deformation methods offer a flexible handle-based FFD framework. The user manipulates a typically sparse set of positional and/or rotational constraints (called handles) and the surface is updated accordingly.

Efficient for simple deformations, linear methods cannot cope with large shape modifications [3], and introduce many visible artifacts. Therefore non-linear variational approaches have been introduced to offer better detail and volume preservation at the cost of scalability and, often, implementation easiness. This includes, the PriMo system [4], its extension to rigid cells [5] and the As-Rigid-As-Possible (ARAP) surface deformation system [6].

*Space Deformation.* The idea of defining the deformation of an object by the means of its embedding space has been introduced by Sederberg et al. [7]. Without any constraint on the actual model representation (e.g., mesh, point set), such techniques suppose the preliminary creation of a coarse, closed polygonal surface mesh – called *cage* – surrounding the object to deform. At initialization time, all the points of the model are expressed as a weighted combination of cage vertices. These weights are *coordinates* in the cage. At runtime, when a user manually moves the cage vertices, all the points are updated by multiplying their original cage coordinates by the current cage vertices. Good coordinate systems ensure at least smoothness and thus allow users to control the shape of a potentially complex object by only editing its cage. Among the popular cage coordinate systems, we can cite the *Mean Value Coordinates* (MVC) [8] which smoothly interpolate vertex data (e.g., positions) using arbitrary, closed manifold cages but may not preserve the model features; the *Green Coordinates* (GC) [9] which offer shape-preservation – actually similar to surface deformation techniques – by considering both cage vertex positions and cage face normals; and the *Harmonic Coordinates* [10] which are defined as the solution of a Laplace equation with boundary conditions on the cage itself. Note also, that cage coordinates can be combined with a skeleton to control the deformation using a sparse set of constraints [11].

*Combined Deformation.* Recent works have tackled the problems stemming from both categories by combining surface and space deformation. For instance, Borosan et al. [12] proposed to perform surface deformation by using ARAP on a subsampled model before transferring the resulting deformation to the full resolution one using MVC coordinates. Similarly, [13] proposed to combine an ARAP deformation and a skeleton to better preserve volume. Unfortunately, such solutions cannot scale above a few tens of thousands of samples whereas the typical size of voxel grids usually exceeds hundreds of millions of samples.

**Voxel Grid Deformation.** Looking at the particular case of voxel grids, most existing techniques make only use of the original space deformation technique by Sederberg et al. [7] to get plausible poses of 3D segmented medical images in order to perform SAR analysis for instance. Nagaoka et al. [14] represent the segmented voxel grid as an hexahedral mesh that is rasterized after the cage-based deformation. Skeleton-based volume deformation methods have also been used in the context of 3D medical images [15]: a mesh representing the outer boundary of the voxel grid is deformed using a manually defined skeleton and the final 3D image is then computed using a mapping and a volume filling algorithm. An alternative method based on a dummy model deformation was proposed by Gao et al. [16]. In both cases, an interface allowing the user to specify the articulation angles is used. Unfortunately, both methods impose a tedious model-specific manual pre-process to get plausible deformations. In fact, most medical images deformation used in today’s simulations are still performed by cutting and pasting real or synthetic limbs and organs with a tedious manual adjustment at the junctions.

In the following, we focus on two recent techniques which are instrumental in our approach.

**As-Rigid-As Possible FFD.** ARAP systems are defined on a surface through an iterative process. In the first step, rotation matrices are estimated for each vertex according to the geometry of its 1-neighborhood. In the second step, new vertex positions are found according to the rotation matrices defined on the vertices and the vertex positional constraints. This process is iterated until convergence.

**Quasi-conform FFD.** GC express the position of a 3D point  $\eta$  inside a triangular mesh  $P$  (i.e. *cage*) as a linear combination of its vertex positions and its triangle normals:

$$\eta = F(\eta; P) = \sum_{i \in I_V} \varphi_i(\eta) v_i + \sum_{j \in I_T} \psi_j(\eta) n(t_j)$$

with  $\mathbb{V} = \{v_i\}_{i \in I_V} \subset \mathbb{R}^3$ , the cage vertices, and  $\mathbb{T} = \{t_j\}_{j \in I_T}$ , its triangular faces and  $n(t_j)$  the outward face normal of  $t_j$ .  $\varphi_i(\cdot)$  and  $\psi_i(\cdot)$  are the cage coordinates. The deformation induced by the deformed cage  $P'$  – with  $\mathbb{V}' = \{v'_i\}_{i \in I_V}$  its vertices,  $\mathbb{T}' = \{t'_j\}_{j \in I_T'}$  its faces – is defined as:

$$\eta \rightarrow F(\eta; P') = \sum_{i \in I_V} \varphi_i(\eta) v'_i + \sum_{j \in I_T} \psi_j(\eta) s_j n(t'_j)$$

with  $\{s_j\}_{j \in I_T}$  the *stretch factor* of  $t_j$  insures scale invariance.

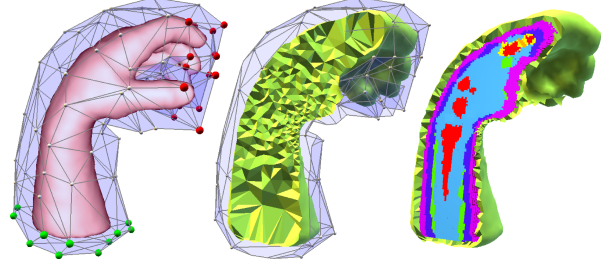


Figure 2: **A 3-scale deformation pipeline:** the input voxel grid deformation process is controlled using a coarse cage with non-linear ARAP deformation (left). When reaching a satisfying shape, the deformation is first transferred to a deformation-adaptive mid-resolution volume mesh using GC (middle right). The full resolution voxel grid is finally rasterized in its deformed space using the so-defined per-simplex linear deformation (right).

### 3. VoxMorph

#### 3.1. Overview

VoxMorph is a freeform deformation system for voxel grids based on a 3-scale deformation procedure:

1. a non-linear variational deformation [6], operated by the user on a coarse cage embedding the model
2. a linear global space deformation transferring the the shape defined at lower scale in a quasi-conformal way [9] to a mid-resolution visualization surface for interactive rendering
3. a new local linear deformation technique transferring the shape modification from a mid-scale tetrahedral deformation-adaptive mesh, undergoing the two previous steps, to the full resolution voxel grid in an efficient 3D rasterization post-process.

The resulting voxel grid contains a deformed model which can be used for simulation, visualization and virtual reality (see Figure 2).



Figure 3: **Visualization envelope:** Starting from the input voxel grid (left) a binary grid captures the foreground/background interface (middle) from which a surface mesh is sampled for interactive visual feedback (right).

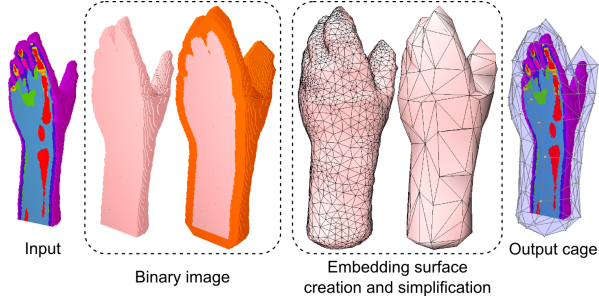


Figure 4: **Deformation Cage**: a dilated version of the foreground/background binary grid (middle left) is meshed at its boundary; the resulting surface is simplified (middle right) and used as the interaction cage (right).

### 3.2. Preprocess

Our system runs on a 3D voxel grid  $\mathbb{G}$  (e.g., medical image), which defines a function  $I : \mathbb{N}^3 \rightarrow E$  with  $E$  being on a continuous (e.g., raw data) or discrete (e.g., post-segmentation) scale. By convention, null values represent the background.

In order to provide the user with a high resolution visual feedback during interactive deformation, we start by extracting the boundary of the foreground volume (e.g., corresponding for instance to the skin of a full body medical dataset). To do so, we generate a binary grid  $\mathbb{B}$  from  $\mathbb{G}$  with  $\mathbb{B}[v] = 0$  for null voxels and  $\mathbb{B}[v] = 1$  otherwise. We extract a smooth mesh approximation of the interface between the two labels using a restricted Delaunay triangulation (RDT) with adaptive refinement as proposed by Boissonnat et al. [17]. We call the resulting visualization surface *envelope* (see Figure 3) and will update it at each frame during interaction by applying the first two scales of our approach. Note that this envelope has typically 3 orders of magnitude less samples than the input high resolution voxel grid.

Although the cage may be designed manually, in our system we propose an optional automatic generation process (see Figure 4). We reuse  $\mathbb{B}$  and apply a supervised dilatation to it to prevent artifacts occurring when the cage is too close to the data. The interface between the resulting domains and the background is meshed again using a RDT before being simplified to a prescribed resolution (typically few tens or hundreds of vertices) using QEM simplification [18].

Last, we compute the GC encoding (see Section 2) of the envelope relative to the cage. At each frame during the modeling session, we transmit the cage deformation to the envelope in a quasi-conformal way using GC deformation. Here, the intermediate envelope resolution

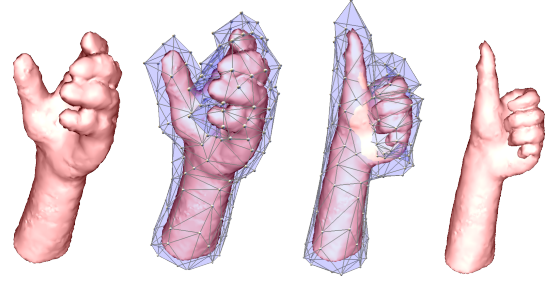


Figure 5: **Interactive FFD**: ARAP deformation performed on the cage, with GC deformation transfer to the visualization envelope.

complies with GC complexity to preserve interactivity at this stage. One interesting property of GC, compared to other coordinate systems (e.g., MVC), is their good space separation property: nearby components can undergo very different deformations as long as cage faces can be located between them (see Figure 6 left).

### 3.3. Interactive deformation

During interaction, the user controls the deformation by manipulating cage vertices only. Indeed, moving each cage vertex independently is a tedious task, even for experienced users, and the result is often unpleasant. Therefore, we provide the user with a higher level mechanism by letting her specifying a few positional constraints on the cage and solve for all the others using ARAP (see Section 2). In particular, the user can specify fixed constraints and new positions to reach (green and red spheres respectively in Figure 2) for an arbitrary

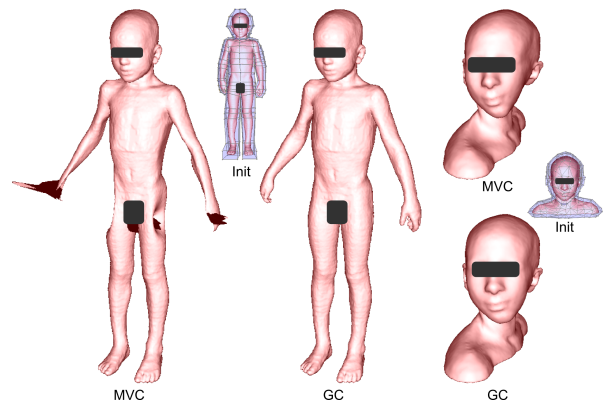


Figure 6: **Comparison**: Mean Value (MVC) versus Green (GC) coordinates in our application context. **Left**: the cage proximity between the arm and the hip leads to a strong distortion when deforming using the MVC whereas GC offer a good separation of the space. **Right**: Interpolating the coarse deformation with MVC exhibits ARAP distortions which are compensated by GC.

subset of the cage vertices and the entire cage will undergo a plausible motion fitting the constraints (example Figure 5). The low number of cage vertices allows to solve for ARAP interactively. The ARAP system can be disabled at any time by the user to allow per-vertex modeling.

Furthermore, the quasi-conformal property of GC compensates for part of the distortions that can still appear with ARAP (see Figure 6 right). The quasi-conformal nature of the deformation is needed in the context of medical image deformation, as the result is as close as possible to what can be obtained with physics-based deformations, using only a linear scheme. GC are the only linear coordinates that ensure this property.

### 3.4. Full Resolution Deformation

While the combination of ARAP and GC can cope with tens of thousands of samples,  $\mathbb{G}$  is several orders of magnitude larger. Not only does this prevent us from using ARAP/GC interactively, but also the computation of GC (with  $O(|P|)$  complexity for a single sample to deform) is prohibitive, even as a postprocess performed once the cage has reached a satisfactory shape. For instance, the full deformation of the complete voxel grids can take up to 5 hours and 20 minutes on the examples we provide in Figure 9.

Therefore, we propose a *third* deformation scale, performed after the interactive FFD session, which is local and linear, and allows to quickly rasterize all input voxels in the deformed space. As ARAP/GC deformations are smooth, they can be approximated, up to a target level of accuracy, using a linear operator. We propose to use an adaptive tetrahedral mesh – the *transfer mesh* (TM) – generated in the inner volume of the cage, and which both captures the ARAP/GC deformations and transfers it linearly, on a per-tetrahedron basis, to its inner space, providing quick deformation queries for any fine voxel.

The TM ensures the correct rasterization of the target grid, without introducing holes in the result. Furthermore, it captures the volume deformation in real-time and allows to track volumetric measures (distortion, stretch, ...) interactively. This feature is essential for the user to control the quality of the on-going deformation.

The key observation here is that linear deformation transfer reproduces rigid motions effectively. Therefore, the TM structure should adapt locally to quasi-rigid deformations, i.e., each tetrahedron should be small enough to bound non-rigidity in its motion. To do so, we start by generating an initial TM to sample a deformation error metric before refining the mesh structure

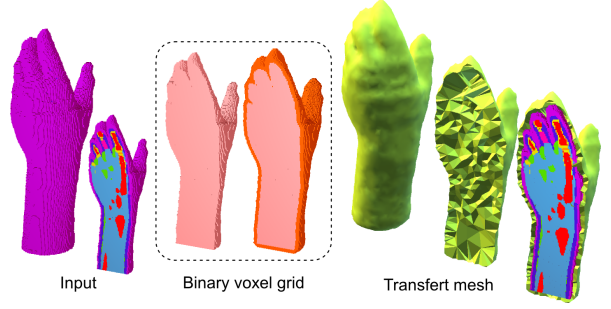


Figure 7: **Transfer Mesh**: an adaptive Delaunay tet-mesh undergoes the ARAP/GC deformation defined interactively and transmits it linearly to the input voxel grid for a limited per-voxel computational cost.

adaptively using a spatially varying sizing field. This leads to a deformation-adaptive TM which is used in a 3D rasterization process to output a deformed voxel grid.

*Initialization.* We generate the initial TM by reusing  $\mathbb{B}$ . This time, we define the TM as an adaptive Delaunay *tetrahedrization* restricted to the binary volume, subject to a *sizing field* grid [19]  $\mathbb{F}$  with the same resolution as  $\mathbb{G}$  so that  $\mathbb{F}[v]$  prescribes a target tetrahedron size in the vicinity of voxel  $v$  (see Figure 7). We initialize  $\mathbb{F}$  to a constant value trading intuitively speed for accuracy (typically 5% of the voxel grid’s diagonal in our experiments). The resulting TM is noted  $M = \{S, T\}$  with  $S = \{s_i\}_{i \in I_S} \subset \mathbb{R}^3$ , its vertices, and  $T = \{t_j\}_{j \in I_T}$ , its tetrahedra indexed over  $S$ . The TM that is deformed using ARAP/GC is noted  $M' = \{S', T\}$ .

*Deformation-Adaptive Meshing.* Each tetrahedron  $t_i$  of  $M$  defines a basis  $B_{t_i}$  which is transformed into  $B'_{t_i}$  by the ARAP/GC deformation. With  $B'_{t_i} = B_{t_i} D_{t_i}$ , the basis change matrix  $D_{t_i}$  is defined as:

$$D_{t_i} = B'_{t_i} B_{t_i}^{-1}.$$

Quick local variation of  $D_{t_i}$  in the volume indicates that a linear interpolation would provide only poor approximation quality. Therefore, we refine the TM iteratively using an adaptive meshing strategy (see Figure 8). First, we compute  $\mathcal{L}_{t_i}$  the Laplacian of the basis change weighted by the tetrahedra volume:

$$\mathcal{L}_{t_i} = \frac{\left\| \sum_{j \in \mathcal{N}(t_i)} (D_{t_j} - D_{t_i}) v_j \right\|}{\sum_{j \in \mathcal{N}(t_i)} v_j}$$

with  $\mathcal{N}(t_i)$  the 1-neighborhood of  $t_i$ ,  $v_j$  the volume of the neighbor  $j$  and  $\|\cdot\|_f$  the Frobenius norm. We measure a per-tetrahedron resizing factor  $\alpha_{t_i}$  from  $\mathcal{L}_{t_i}$  and a



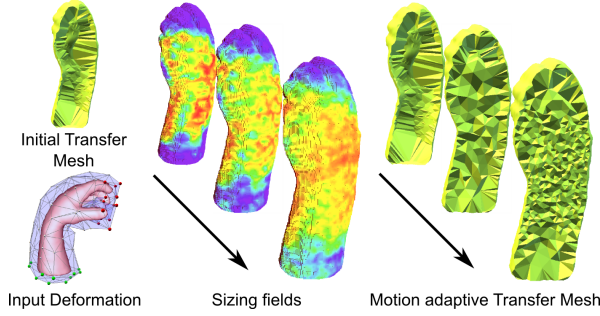


Figure 8: **Adaptive Linear Deformation Transfer:** the Transfer Mesh (right) is generated from a deformation-adaptive sizing field (left).

threshold  $h$  as:

$$\alpha_{t_i} := \begin{cases} 1 & \text{if } \|\mathcal{L}_{t_i}\|_f < h \\ 1/2 & \text{otherwise} \end{cases}$$

Second, we compute a per-vertex smooth value:

$$\tilde{\alpha}_{s_j} = \frac{\sum_{t_i \in \mathcal{N}(s_j)} \alpha_{t_i}}{\text{card}(\mathcal{N}(s_j))}$$

with  $\mathcal{N}(s_j)$  the incident simplices of  $s_j$ . Third, we compute a per-voxel value  $\hat{\alpha}_v$  using a barycentric interpolation:

$$\hat{\alpha}_v = \sum_{s_k \in t_v} \lambda_{s_k} \tilde{\alpha}_k$$

with  $\lambda_{s_k}$  the barycentric coordinates of the center of  $v$  in the tetrahedron  $t_i$  intersecting it. Note that the interpolation of resizing factors is critical to avoid discretization artifacts during remeshing. Last, we update  $\mathbb{F}$ :

$$\mathbb{F}[v] := \mathbb{F}[v] \hat{\alpha}_v$$

and update the TM accordingly. This process is iterated until  $\mathbb{F}$  converges or a prescribed TM resolution is reached.

**Barycentric Rasterization.** The final step of this third-scale deformation consists in computing the actual output grid  $\mathbb{G}'$ . We start by computing the bounding box of the deformed cage and create  $\mathbb{G}'$  with a voxel size similar to  $\mathbb{G}$ . Then, we iterate over the TM tetrahedra, and assign a label to the voxels contained inside it. To do so, we compute the bounding box of each tetrahedra  $t'_i$ , then we iterate over the voxels in it: for each voxel  $v'$  that has not been visited, and is in  $t'_i$ , we compute its barycentric coordinates in  $t'_i$  and multiply them with the coordinates of the corresponding tetrahedron  $t_i$  in  $M$  to obtain a 3D position in the initial

space and a corresponding voxel  $v$  in  $\mathbb{G}$ . Finally, we set  $\mathbb{G}'[v'] = \mathbb{G}[v]$ . Once all the tetrahedra have been processed, the voxels that have not been visited are set to 0, since they are not in TM (i.e., background).

Finally, note that the construction of the TM we presented is generic and is not restricted to a particular underlying deformation scheme.

## 4. Implementation and Results

We implemented our system in C++ with OpenGL for rendering. We used CHOLMOD [20] as a Cholesky solver and the CGAL library [19] 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 a nVidia Quadro FX580 device.

In Figure 9, we illustrate the deformation of four voxel grids. The blue histograms represent the deviation from a locally rigid deformation: the first one shows angle distribution of genuine orthogonal pairs after deformation (no distortion at  $90^\circ$ ) and the second the stretch (no distortion at 1). We can see that the resulting deformations are close to locally rigid ones, even under strong motions, and that the stretch is minimized. Multi-color histograms show the volume change on a per-label basis after deformation. The global volume change is bounded, in the worst case, to 11.6% for all the experiments we conducted. These measures validate our 3-scale approach, in particular the new adaptive fine-scale stage. Table 1 summarizes the performance of our system on the models from Figure 9 (and others). On the largest model, a full resolution deformation can still be obtained in less than 8 minutes, which is more than 50 times faster than a full GC deformation.

Model	Voxels	CV	EV	FPS	TV	CT
Hand	2 985k	263	2 545	50	9 879	28s
Head	4 792k	263	15 628	7.8	31 672	1m47
Arm	8 990k	95	6 264	28	15 314	56s
Thel.	122M	303	61 347	3.	78 789	4m17
Louis	412M	512	44 748	3.	168 712	7m55

Table 1: **Performance table:** with CV (resp. EV and TV) the cage (resp. envelope and transfer mesh) vertex count, FPS the framerate during interaction and CT the final full deformation postprocess time.

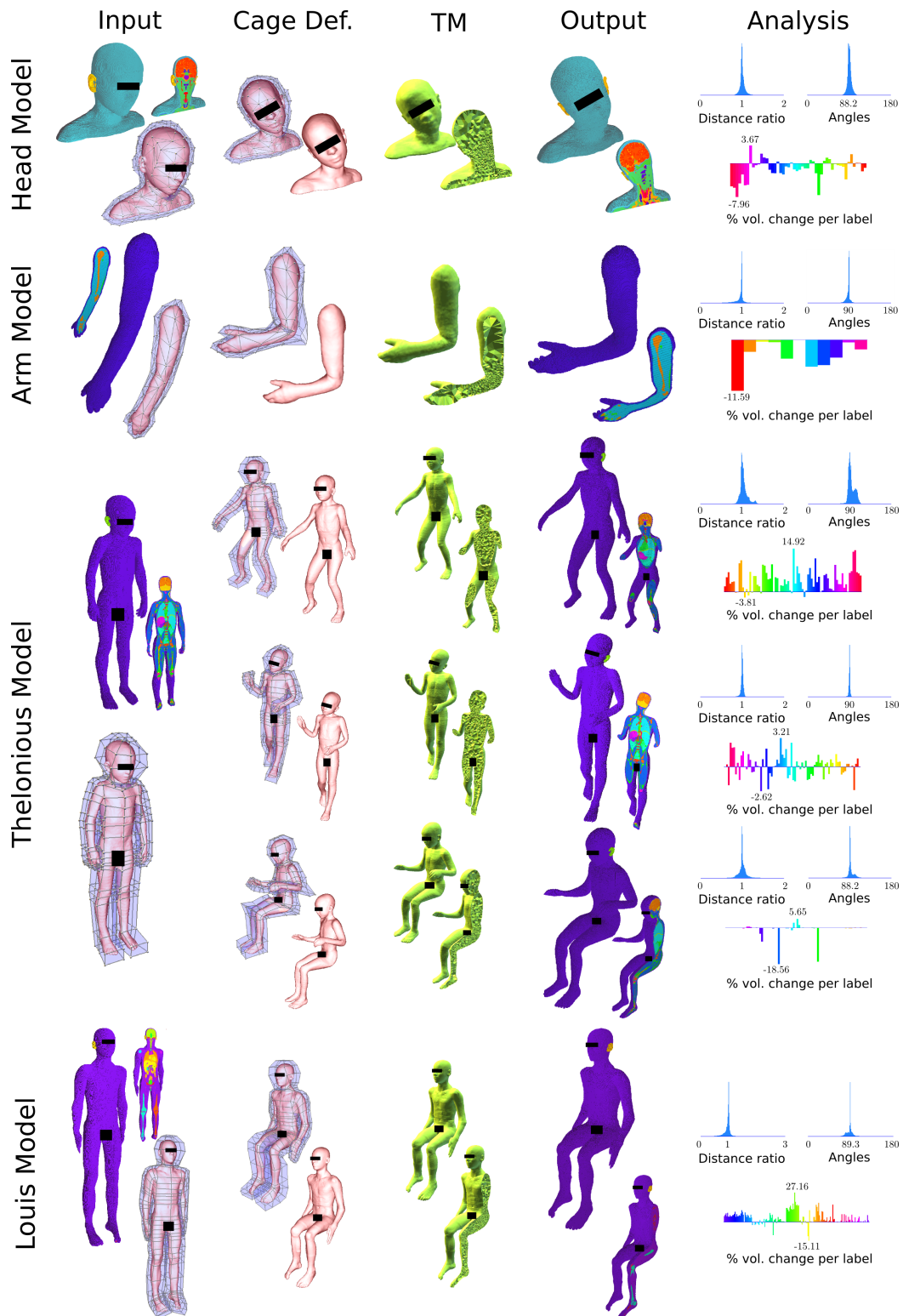


Figure 9: Deformation of voxel grids using VoxMorph. The *distance ratio* and *angles* are calculated on the dual of the voxel grid. The errors histograms indicate that the deformations are mostly conformal, and induce neither stretch nor angle distortions. The labels that endure large volume changes are the ones with an originally very small volume: these variations come from unavoidable rasterization artifacts.

## 5. Limitations & Future Work

Our deformation pipeline successfully generates high quality deformations of high resolution voxel models without resorting to full resolution meshing. As we target general shape modeling scenarios, our system allows for arbitrary deformations. If physically-based deformation is mandatory (e.g., posing applications), an interesting venue for future work would be to combine our 3-scale scalable approach with recent, computationally demanding, material-aware methods [21]. Similarly, while bending an arm is easier with a skeleton, editing the morphology of a face is easier with our system. In fact, a cage-based systems such as ours are typically complementary to a skeleton-based one. Here again, a 3-scale approach could be developed.

The (optional) automatic cage creation process raises numerous interesting questions for future work. For instance, how to handle two distinct components that are too close, the dilatation step may not lead to a cage allowing to manipulate them independently. Also, the QEM simplification does not guarantee to still bound the volume of interest. Fortunately, both cases can be fixed by adjusting interactively the dilation factor, imposing manual editing of the cage structure only in very rare cases. Still, high quality automatic cage creation is an important direction for future work.

Currently, our system is limited to voxel grids fitting in memory. Scaling up could be achieved by using compressed representations or out-of-core/streaming environments. Although existing spatial or surface deformation methods could be used to replace the two first scales of our pipeline, our particular combination of ARAP and GC seems to outperform consistently other choices in the numerous experiments we conducted.

We mainly used our system for medical images but it can also be used to deform other kind of volume data. It is currently intensively used by physicists for editing voxelized body models used in human-waves interactions studies and digital dosimetry simulations, known to be computed better on regular grids. This initial non-expert user feedback indicates that our system can be instrumental in other scenarios.

## 6. Conclusion

We have proposed VoxMorph, a 3-scale interactive freeform deformation tool for voxel grids. By combining a high quality non-linear deformation at coarse scale, a quasi-conformal space deformation at mid-scale and a new adaptive local linear deformation transfer at fine-scale, our system allows to deform easily high

resolution voxel grids. VoxMorph provides a flexible yet predictable way for modifying volumetric data and completes a more general methodological framework that may include skeleton-based and physics-based tools.

**Acknowledgment.** This work was funded by ANR Kid-Pocket, ANR FETUS and EU REVERIE IP projects. Medical images are taken from the *Virtual Population project*.

## References

- [1] M. Alexa, D. Cohen-Or, D. Levin, As-rigid-as-possible shape interpolation, in: SIGGRAPH, 2000, pp. 157–164.
- [2] T. Igarashi, T. Moscovich, J. F. Hughes, As-rigid-as-possible shape manipulation, ACM ToG 24 (2005) 1134–1141.
- [3] M. Botsch, O. Sorkine, On linear variational surface deformation methods, IEEE TVCG 14 (1) (2008) 213–230.
- [4] M. Botsch, M. Pauly, M. Gross, L. Kobbelt, Primo: coupled prisms for intuitive surface modeling, in: EG Symp. On Geo. Proc., 2006, pp. 11–20.
- [5] M. Botsch, M. Pauly, M. Wicke, M. Gross, Adaptive space deformations based on rigid cells, Computer Graphics Forum 26 (3) (2007) 339–347.
- [6] O. Sorkine, M. Alexa, As-rigid-as-possible surface modeling, in: EG Symp. on Geo. Proc., 2007, pp. 109–116.
- [7] T. W. Sederberg, S. R. Parry, Free-form deformation of solid geometric models, in: SIGGRAPH, 1986, pp. 151–160.
- [8] T. Ju, S. Schaefer, J. Warren, Mean value coordinates for closed triangular meshes, ACM ToG 24 (2005) 561–566.
- [9] Y. Lipman, D. Levin, D. Cohen-Or, Green coordinates, ACM ToG 27 (2008) 78:1–78:10.
- [10] P. Joshi, M. Meyer, T. DeRose, B. Green, T. Sanocki, Harmonic coordinates for character articulation, ACM ToG 26 (3) (2007) 71.
- [11] M. Ben-Chen, O. Weber, C. Gotsman, Variational harmonic maps for space deformation, ACM ToG 28 (2009) 34:1–34:11.
- [12] P. Borosán, R. Howard, S. Zhang, A. Neal, Hybrid mesh editing, in: Proc. Eurographics, 2010.
- [13] S. Zhang, A. Nealen, D. Metaxas, Skeleton Based As-Rigid-As-Possible Volume Modeling, pp. 21–24.
- [14] T. Nagaoka, S. Watanabe, Postured voxel-based human models for electromagnetic dosimetry, Physics in Medicine and Biology 53 (24) (2008) 7047.
- [15] T. Nagaoka, S. Watanabe, Voxel-based variable posture models of human anatomy 97 (12) (2009) 2015–2025.
- [16] J. Gao, I. Munteanu, W. F. O. Müller, T. Weiland, 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.
- [17] J.-D. Boissonnat, S. Oudot, Provably good sampling and meshing of surfaces, Graph. Models 67 (2005) 405–451.
- [18] M. Garland, P. S. Heckbert, Simplifying surfaces with color and texture using quadric error metrics., in: IEEE Vis., 1998, pp. 263–269.
- [19] CGAL, Comp. Geom. Alg. Lib., <http://www.cgal.org>.
- [20] Y. Chen, T. A. Davis, W. W. Hager, S. Rajamanickam, Algorithm 887: Cholmod, supernodal sparse cholesky factorization and update/downdate, ACM Tr. Math Soft 35 (3) (2008) 1–14.
- [21] F. Faure, B. Gilles, G. Bousquet, D. K. Pai, Sparse Meshless Models of Complex Deformable Solids, in: SIGGRAPH, 2011, pp. 73:1–73:10.