Volume Warping for Adaptive Isosurface Extraction

Laurent Balmelli* Christopher Morris Gabriel Taubin Fausto Bernardini

IBM Research
T.J. Watson Center
Hawthorne, NY, USA

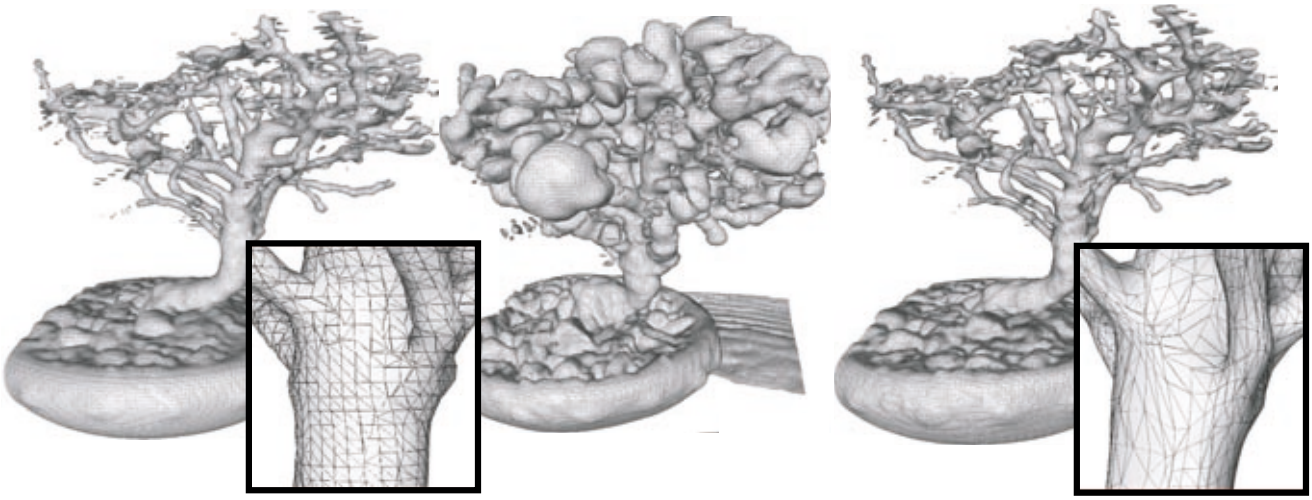


Figure 1: Adaptive isosurface extraction of the Bonsai tree dataset. Left: isosurface extracted from the original 128^3 dataset. In this mesh, the vertices are evenly distributed. Center: isosurface extracted from a warped 128^3 volume. Regions containing containing small details, such as branches, have been inflated to the expense of coarser ones. Right: unwarped version of the surface shown in the center. The resulting mesh is more densely tessellated in regions containing details as a result of the warping process.

Abstract

Polygonal approximations of isosurfaces extracted from uniformly sampled volumes are increasing in size due to the availability of higher resolution imaging techniques. The large number of primitives represented hinders the interactive exploration of the dataset. Though many solutions have been proposed to this problem, many require the creation of isosurfaces at multiple resolutions or the use of additional data structures, often hierarchical, to represent the volume.

We propose a technique for adaptive isosurface extraction that is easy to implement and allows the user to decide the degree of adaptivity as well as the choice of isosurface extraction algorithm. Our method optimizes the extraction of the isosurface by warping the volume. In a warped volume areas of importance (e.g. containing significant details) are inflated while unimportant ones are contracted. Any extraction algorithm can be applied to the warped volume. The extracted mesh is subsequently unwarped such that the warped areas are rescaled to their initial proportions. The resulting isosurface is represented by a mesh that is more densely sampled in regions decided as important.

CR Categories: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—curve, surface, solid, and object representations; I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—hierarchy and geometric transformations;

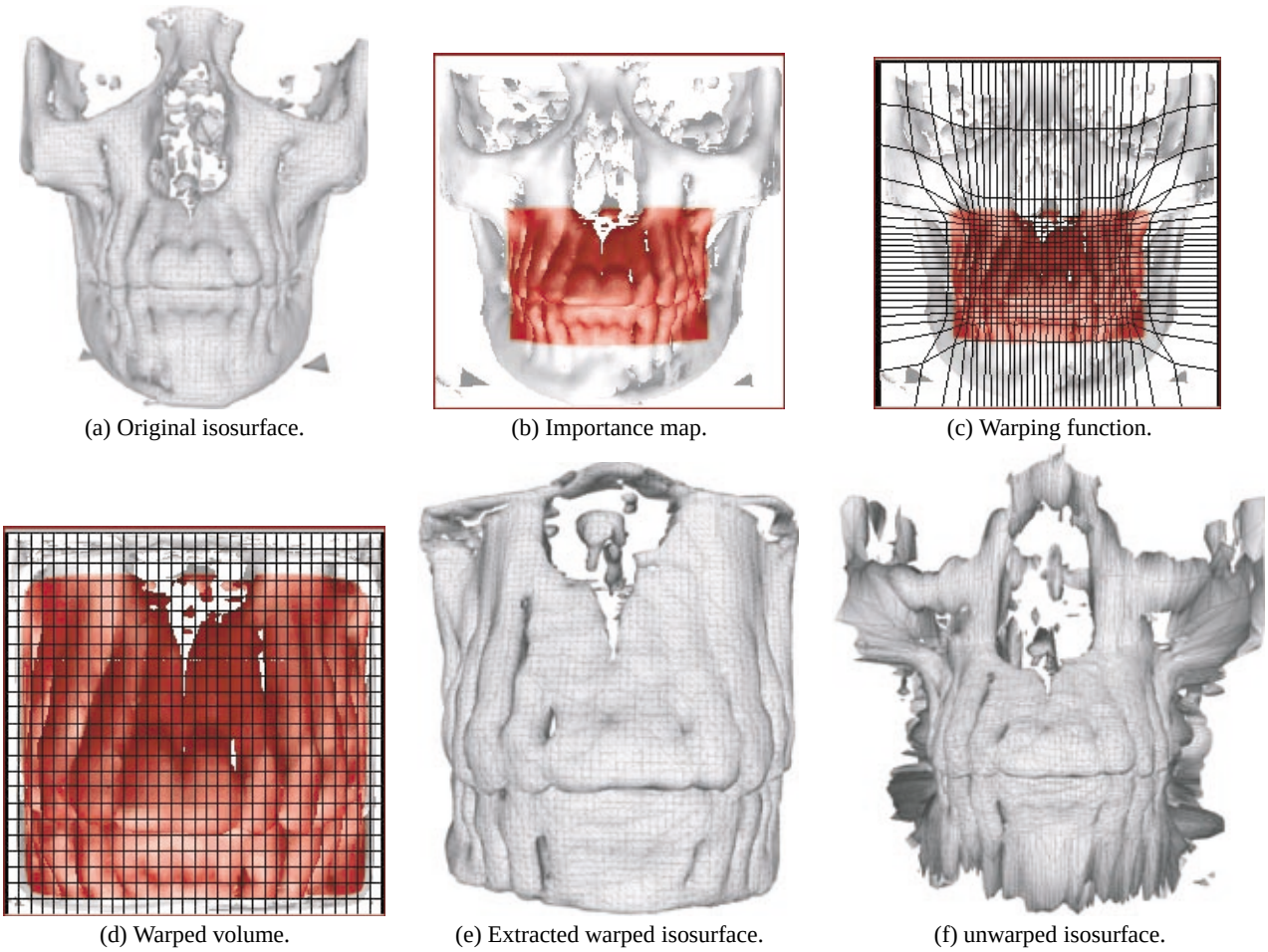
Keywords: isosurfaces, adaptive isosurface extraction, volume warping, adaptive tessellation

*Laurent Balmelli is the contact author and can be reached at balmelli@us.ibm.com

1 Introduction

With the advancement of data acquisition and storage techniques, volume datasets are steadily increasing in both size and resolution. As a result, we need efficient methods to visualize these datasets and cope with bandwidth and storage limitations. Researchers are investigating ways to adapt a common visualization technique, namely *isosurface extraction*, to the focus of the visualization. A typical strategy is to make the extraction adaptive to the local complexity of the isosurface in order to produce meshes that are fine in areas of interest and coarse in the remaining ones. This is opposed to extracting vertices uniformly over the isosurface. In this way the overall density of extracted vertices is significantly reduced while preserving the quality of the isosurface. The immediate benefit is the reduction of the cost in storage, transmission and rendering. We address this problem, i.e. *adaptive isosurface extraction*, by proposing a new methodology based on a *volume warping technique.*

Our technique is an extension of the space-optimized texture map formulation presented in [2] to volume datasets. Given an *importance map* (represented as a volume dataset) we use a relaxation algorithm to warp the input volume. In the warped



(a) Original isosurface. (b) Importance map. (c) Warping function.

(d) Warped volume. (e) Extracted warped isosurface. (f) unwarped isosurface.

Figure 2: Volume warping pipeline. In this example an importance map is given by the user. The map assigns maximum importance to the jaws of the Skull model. Consequently this region is agressively inflated in the warped volume. The resulting isosurface is densely tessellated around the jaws and coarse elsewhere. A close-up of the output mesh is shown in Figure 3.

volume, areas designated as important by the map are inflated to the expense of less important ones. Consider the example of Figure 1: The model on the left handside is extracted from an *unoptimized volume*, i.e. the volume has not been warped. The close view illustrates the uniformity of the tessellation. The central model is extracted from the warped volume. Importance is given to the branches and small fluctuations along the model's surface. Hence these regions are inflated consequently.

Once a mesh is extracted from the warped volume, the vertex locations are unwarped in order to locally rescale the isosurface to its original proportions. The rescaled isosurface looks identical to the one extracted without warping the volume, with the difference that the sampling of the mesh is finer in areas regarded as important. The model on the right handside of Figure 1 is the unwarp version of the central model. The close-up view shows the adaptive tessellation as a result of the warping operation. The extraction algorithm is applied to a smaller (warped) version of the input volume in order to get a total vertex count identical to the one obtained with the extraction from an unoptimized volume. Our method is therefore more economic in computational and storage requirements with the benefit of producing an adaptively sampled isosurface. Obviously our method requires to previously warp the input volume; for this task we use an efficient multigrid approach allowing the operation to be performed efficiently.

Our method has several advantages: For example, our algorithm does not require the use of any complex data structures, such as octrees and is relatively simple to implement. Furthermore, our method allows the user to control the importance map such that the extraction is best suited to the goal of the visualization. The importance map can be computed automatically by analyzing the content of the volume, but can also be fully specified by the user. Our method relies on the fact that a warped volume can be resized (i.e. downsampled) up to a certain extend without loss of details. A rapid analogy in signal processing is the downsampling of a discrete signal without aliasing. This is possible only if no frequency component (in the underlying continuous function) is higher than half the resampling frequency (Shannon's Sampling Theorem). Therefore a warped volume can be downsized without loss in details up to a critical size. Obviously this size depends directly on the characteristics of the dataset, i.e. on the amount of high-frequency details.

As said previously, our method does not require the use of a specific extraction algorithm. Hence our method can be used jointly with the latest extraction techniques. Finally, our algorithm produces isosurfaces that are adapted to the volume structure and the user's specifications and does not suffer from any mesh discontinuities or gap-filling problems that typically require additionnal pro-

cessing [11].

2 Previous work

Many researchers have sought to extend common isosurface extraction methods, such as Marching Cubes [6], to produce adaptively tessellated isosurfaces (see for example [**?**].) A prevalent idea, as is presented by Saupe [10], for the solution to this problem requires the use of algorithms and data structures that are often very specific to certain visualization scenarios and typically require large amounts of additional storage. Commonly, structures such as octrees [14, 12] and interval trees [3] have been applied to address this problem. Furthermore, to facilitate the transition from finely sampled regions to coarser ones, multiresolution hierarchies are used [8, 7]. These techniques provide results of quality however their costs in storage can be a limitation.

Several techniques have been implemented that also provide feature-driven extraction [15, 5] as our method. These techniques do not give to the user the flexibility in determining which areas to refine without the use of a multiresolution hierarchy. One approach to address this problem has been to use compressed volumes [12, 13] or the extracted isosurface itself [9]. However, once again, these techniques require the use of intensive calculations and complex data structures.

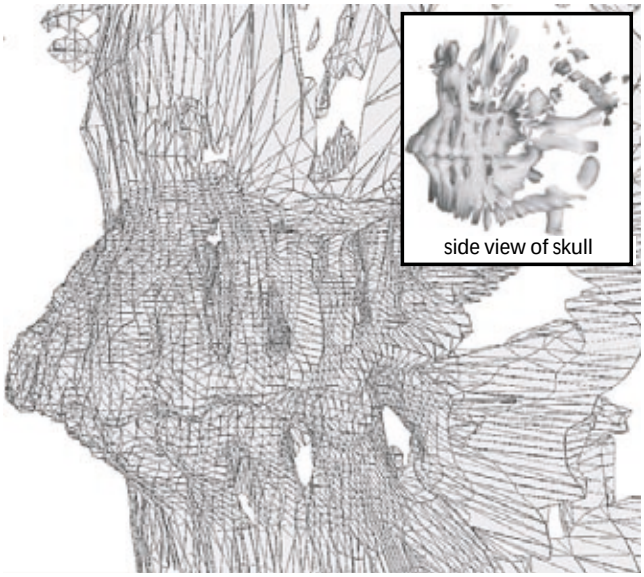


Figure 3: Close-up view of the adaptive tessellation of the jaws (example model in the volume warping pipeline depicted in Figures 2a-f.)

3 Approach

Our method has four stages represented in the example of Figures 2a-f. Given an input volume (the isolevel to be extracted is represented in Figure 2a), an importance map is first specified for the isolevel, as suggested in Figure 2b. The map is given as a volume dataset (not having necessarily the same size as the input dataset). However in the examples of this paper, we give an intuitive representation of a map as a painted isosurface. The darker the shade of isosurface region the more important the area in the volume. The map can be given by the user, as in Figure 2b, or computed automatically (e.g. as in Figure 4, see next section.) In the map of Figure 2b we assign the highest importance to the jaws of the skull model.

A *warping function* is computed using the map and a relaxation algorithm. The function is modeled using a tridimensional regular grid of user-defined resolution, as suggested in Figure 2c. The input volume is then resampled using the warping function, leading to a warped volume (Figure 2d). Finally, a mesh is extracted from the warped volume (Figure 2e) and the vertex locations are unwarped using again the warping function depicted in Figure 2c. In the resulting adaptive isosurface represented in Figure 2f, the mesh is denser in important areas. In Figure 3 we illustrate the dramatic increase in mesh density in the jaws.

3.1 Feature-sensitive importance map

A powerful component of our pipeline is the importance map. The values in the map define a measure of importance for each corresponding voxel (or group of voxel) location. Importance maps for volume visualization have previously been investigated, for example, by Bajaj *etal* in [1]. This map is used to compute the warping function (e.g. as in Figure 2c) which, in turn, is used to resample the volume (e.g. as in Figure 2d.) The importance map can be explicitly defined by the user as it is in Figure 2b. In this example, the jaws has been designated as important. Alternatively, a common goal is create adaptive isosurfaces that preserve areas of significant detail across the entire volume. To do so we propose a method to compute an importance map based on the number of crossings for the given isolevel in an user-defined neighborhood for each voxel. We call the map *neighborhood-crossing map.*

We explain the algorithm to compute the map in more detail below. For each voxel we search for intensity values within a user-defined range of the isolevel. If such a crossing is found, then each voxel in a neighborhood surrounding the voxel is then checked for an isolevel crossing. Each time an isolevel crossing is found, a counter is incremented. The resulting number of crossings is stored for each voxel in the importance map. The isolevel, its range, and the size of the neighborhood kernel are specified by the user.

In Figure 4 we give a representation of the neighborhood-crossing map obtained for the Bonsai tree dataset. The isosurface extracted from the volume warped using the importance map above is shown in Figure 1. More examples of neighborhood-crossing maps can be seen in Figures 8 and 9.

3.2 Volume warping

We describe now the volume warping operation more formally. We start with a scalar function $f : Q \rightarrow \mathbb{R}$ defined on the unit cube Q. Our goal is to generate an adaptive polygon-mesh approximation of the level set $S = \{p : f(p) = L\}$, where L is the isolevel of interest. We first compute a warping function $w : Q \rightarrow Q$ (we actually estimate $\eta = w^{-1}$, see Figure 5). Then, we generate a uniform polygon-mesh approximation M' of the isolevel $S' = \{p' : f'(p') = L\}$, where $f' = f \circ \eta$.

The continous function f is represented as a regularly sampled volume dataset $F = \{F_{\alpha_j} : 0 \leq \alpha_j < N_j, j = 0, 1, 2\}$, and incorporates trilinear interpolation to achieve continuity. The uniform polygon mesh M' is computed as an isosurface extracted from the warped scalar function f', which, for this purpose, must also be represented as a regularly sampled dataset F'. To generate F', we use the 3D equivalent of inverse texture mapping. We evaluate the inverse warping function at the coordinates of the

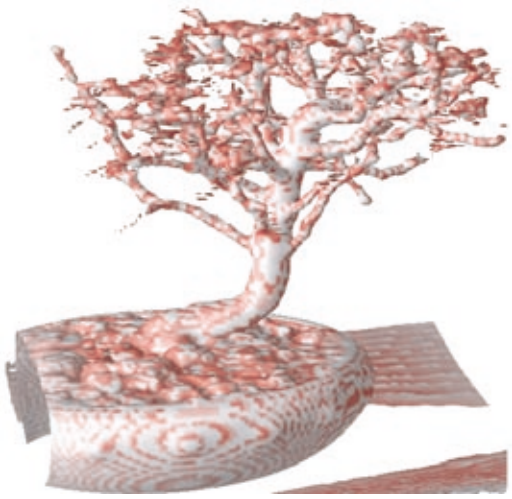


Figure 4: Importance map for the Bonsai tree dataset. The map is represented as a painted isosurface. The darker the shade of isosurface region the more important the area in the volume.

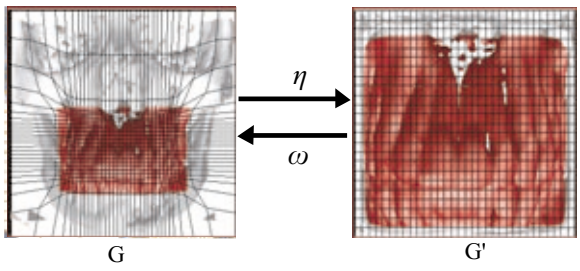


Figure 5: Warping ω and inverse warping η.

centroid p'_α of each voxel in F' to obtain a point p_α in the domain of f, and set the value F'_α of the voxel equal to the result $f(p_\alpha)$ of evaluating the original scalar function f the computed point p_α. Note that this procedure does not require the warped data set F' to be of the same size as the input dataset F. The sampling rate can be changed (increased or decreased) during the inverse mapping step.

The warping function is represented by two associated grids, denoted by G and G' (Figure 5), with the same connectivity, and again, trilinear interpolation for continuity. A more complex interpolation scheme can be used, but we found trilinear interpolation to be good enough for our experiments for both the volume data and the warping function A regular grid G' covers the dataset F', and the (inverse) warped grid G covers the dataset F. To evaluate the inverse warping function at a point $p' \in Q$, we first need to determine which cell C' the point belongs, then compute trilinear coordinates of the point with respect to the eight vertices of the containing cell on F', and finally evaluate the trilinear coordinates at the eight vertices of the corresponding cell C on F. Since the grid covering F' is regular, determining which cell p' belongs to is simply done by quantization. Evaluating the forward warping function is more expensive because, since the grid G is not regular, determining which cell a point $p \in Q$ belongs to requires a spatial search. However, this is irrelevant because our approach does not require to evaluate the forward warping function at all.

3.3 Relaxation

As explained in the previous section, the warping function is representated using two associated grids G and G'. The function is computed using an extension of the relaxation algorithm described in [2] to tridimensional grids. We briefly describe the relaxation process below and refer the reader to [2] for more details.

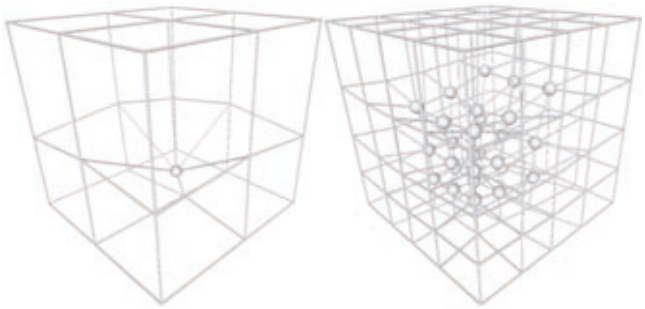


Figure 7: .

Initially, the vertices in the grid are initialized to a uniform cube in the unit volume. At each step of the relaxation, a displacement vector is computed for each vertex. These vectors are evaluated by sampling the intensity of the importance map at each vertex location in the volume. Note that since the importance map does not have necessarily the same size as the original volume, trilinear interpolation is used to retrieve the intensity. A weight is evaluated for each pair of neighbor vertices[1] using their sampled intensity. A weight will be large for a vertex sampling a high activity region. As a result, vertices in low activity regions are pulled by the ones sampling high activity regions (e.g. as suggested in Figure 7.)

To improve convergence accuracy and speed, we use a multigrid approach. Initially, the warping function is modeled with a coarse grid, e.g. a $3 \times 3 \times 3$ grid (see the left handside of Figure 7.) When the relaxation converges for the current grid resolution, the grid is subdivided using linear interpolation (right handside of Figure 7.) Then, the process is reiterated until convergence and again the grid is subdivided, etc. In order to ensure that the warping function spans the whole volume during the relaxation process, constraints are applied to vertices lying on faces and at corners. Corner vertices are not allowed to move, whereas face vertices can only move within their face. In Figure 7 only the internal grid vertices are represented using a sphere for clarity.

3.4 Isosurface extraction

Once the volume is warped, any isosurface extraction algorithm can be used to generate the corresponding mesh. This flexibility is allowed due to the fact that areas of interest have already been expanded within the warped volume in contrast to the remaining areas of lesser interest which have been contracted. Subsequently, any extraction algorithm will define these regions appropriately. Our volume warping and extraction pipeline is implemented in OpenDx[2]. In this environment the *Alligator* extraction algorithm [4] is implemented (a modification of the Marching Cubes algorithm [6]). In Figure 10 we show the result obtained using a standard implementation of the Marching Cubes algorithm. The isosurface is extracted from a 128^3 volume dataset warped using the importance map de-

[1] We consider only the neighbors in the six cardinal directions.

[2] www.opendx.org

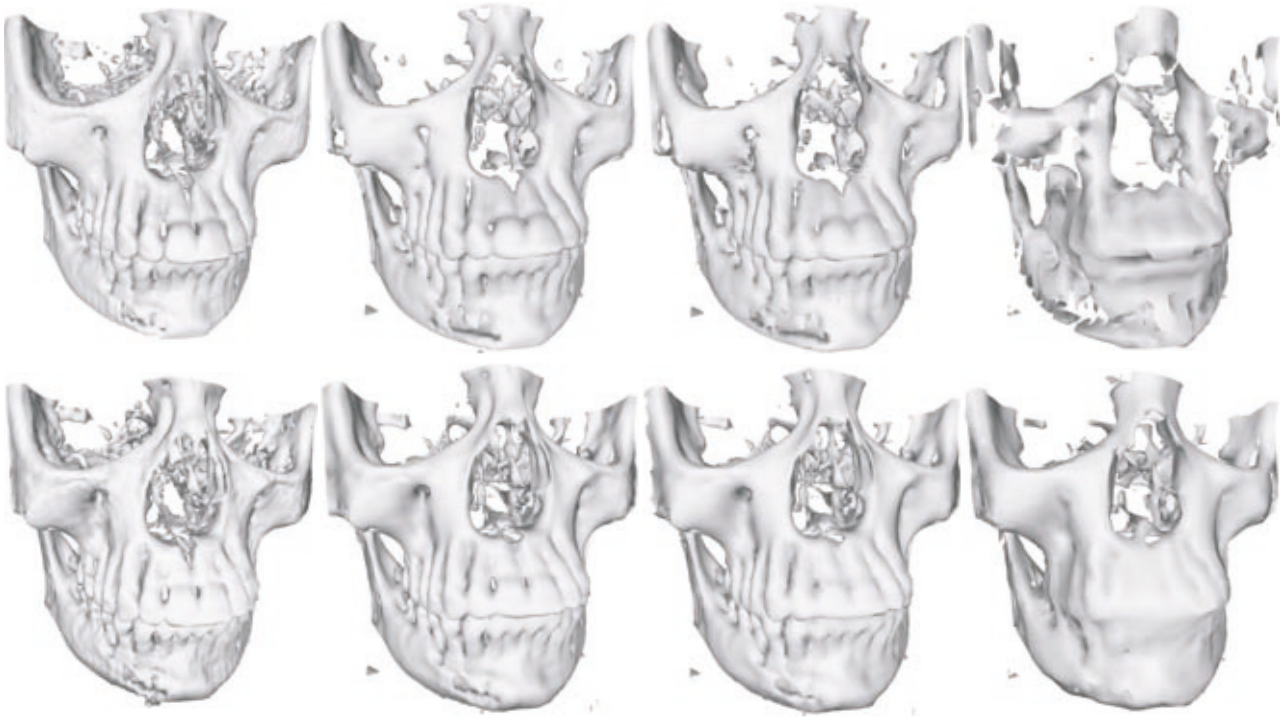


Figure 6: Extraction from downsampled datasets. First row: extraction from optimized volume. Second row: extraction from warped volume. The volumes are warped using the importance map shown in Figure 8. The volume dimensions are, from left to right, 128^3, 96^3, 64^3 and 32^3, respectively.

picted in Figure 8. The same isosurface obtained using the Alligator algorithm [4] is shown in the second row of Figure 8 (left handside.)

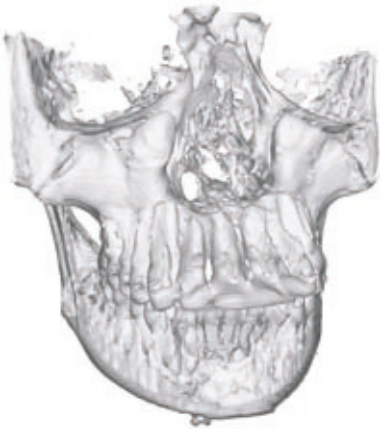


Figure 10: Unwarped isosurface obtained when using a standard implementation of the Marching Cubes algorithms [6]. The isosurface is extracted from a 128^3 volume dataset warped using the importance map depicted in Figure 8. The same isosurface obtained using the Alligator algorithm [4] is shown in the second row of Figure 8 (left handside.)

3.5 Unwarping of the extracted isosurface

The vertices M' of the extracted isosurface are recovered from the warped volume F'. In order to obtain the original proportion of the mesh, i.e. M, the vertex location must be unwarped. Since the vertices of the warped mesh lie on edges of the grid G', the final location of each vertex is computed using the same warping function used to warp the volume, i.e.

$$M = \eta(M'). \tag{1}$$

The output polygon mesh M has the same connectivity as M'.

4 Results

We tested our volume warping with common volume datasets. These datasets have size 128^3 and are freely available on the Internet[3].

In Figures 8 and 9 we show experimental results. In both figures, the importance map used to warp the volume is shown in the central frame. In the first rows, we show the volumes extracted from the unoptimized and warped dataset, respectively. Both datasets (Skull and Foot) have size 128^3. In both figures, the model at the second row, right handside has a number of vertices close to the one obtained using the unoptimized 128^3 volume. For the Skull model, this is obtained with a warped 103^3 volume, whereas for the Foot model this is obtained for a warped 116 volume. For the latter dataset, we observed that the number of vertices obtained from the extraction decreases very quickly when downsampling the volume. We suggest that this behavior is due to an important amount of noise in the dataset for the chosen isolevel. The isosurface extracted from the warped 103^3 volume (second row, right handside in Figure 9) is smooth compared to the one obtained for the unoptimized volume, while conserving most

[3] www.gris.uni-tuebingen.de/areas/scivis/volren/datasets/datasets.html

of the significant details. Finally, we show examples of adaptive tessellations in the third row of both figures.

We give in the table below the number of extracted vertices obtained with our implementation for different volume dimensions. We use isolevels 42, 35, 50 for the Bonsai, Skull and Foot dataset, respectively (the volumes are encoded using two bytes per intensity.)

Model	*Original*	*Warped*				
	128^3	128^3	103^3	96^3	64^3	32^3
Bonsai	69'811	157'337	77'909	67'676	30'373	5'748
Skull	186'009	313'840	162'762	143'661	64'480	12'801
Foot	125'052	175'890	77'340	67'820	30'545	6'193

Table 1: Number of vertices extracted from the unoptimized and warped datasets with our implementation.

We show in Figure 6 that details are longer preserved in a warped volume compared to an unoptimized one when reducing the size of the datasets. In the figure, the isosurfaces in the first row are extracted from the (unoptimized) Skull dataset, where the second row contains the ones extracted from a warped dataset. The dataset is warped using the importance map shown in Figure 8. The number of extracted vertices in each case is given in the table below.

	128^3	96^3	64^3	32^3
Unoptimized	186'009	68'510	29'647	4'669
Warped	313'840	143'661	64'480	12'801

Table 2: Number of vertices extracted from the reduced unoptimized and warped datasets.

5 Conclusion

6 Acknowledgements

We would like to thank Gregory Abram for his assistance with OpenDX. The volume warping pipeline was implemented entirely in OpenDX and was used to generate the isosurfaces shown in this paper.

References

[1] Chandrajit Bajaj, Insung Ihm, and Sanghun Park. Visualization-specific compression of large volume data. In *Proceedings of Pacific Graphics 2001*, pages 212–222, Tokyo, Japan, October 2001.

[2] Laurent Balmelli, Gabriel Taubin, and Fausto Bernardini. Space-optimized texture maps. *To appear in Proceedings of Eurographics*, September 2002. preprint available at http://www.balmelli.net/download/RC22328.pdf.

[3] Paolo Cignoni, Paola Marino, Claudio Montani, Enrico Puppo, and Roberto Scopigno. Speeding up isosurface extraction using interval trees. *Transactions on Visualization and Computer Graphics*, 3(2):158–170, April - June 1997. ISSN 1077-2626.

[4] Alan D. Kalvin. *Segmentation and surface-based modeling of objects in 3-D biomedical images*. Ph.d. thesis, New York University, 1991.

[5] Leif P. Kobbelt, Mario Botsch, Ulrich Schwanecke, and Hans-Peter Seidel. Feature-sensitive surface extraction from volume data. In *SIGGRAPH 2001*, Computer Graphics Proceedings, Annual Conference Series, pages 57–66. ACM Press / ACM SIGGRAPH, August 2001. ISBN 1-58113-292-1.

[6] William Lorenson and Harvey Cline. Marching cubes: A high resolution 3d surface construction algorithm. *Computer Graphics*, 21(4):163–169, July 1987.

[7] Ralf Neubauer, Mario Ohlberger, Martin Rumpf, and Ralph Schwörer. *Visualization in Scientific Computing '97*, chapter Efficient visualization of large-scale data on hierarchical meshes. Springer, Wien, 1997. ISBN 3-211-83049-9.

[8] Mario Ohlberger and Martin Rumpf. Hierarchical and adaptive visualization on nested grids. *Computing*, 59(4):365–385, 1997.

[9] Dietmar Saupe and Jens-Peer Kuska. Compression of isosurfaces for structured volumes. In *Vision, Modelling and Visualization 2001*, Stuttgart, Germany, November 2001. IEEE.

[10] Dietmar Saupe and Jürgen Toelke. Optimal memory constrained isosurface extraction. In *Vision, Modelling and Visualization 2001*, Stuttgart, Germany, November 2001. IEEE.

[11] Raj Shekhar, Elias Fayyad, Roni Yagel, and J. Fredrick Cornhill. Octree-based decimation of marching cubes surfaces. In *Visualization 1996*, pages 335–344. IEEE, October 1996. ISBN 0-89791-864-9.

[12] Francisco Velasco and Juan Carlos Torres. Cells octree: a new data structure for volume modeling and visualization. In *Vision, Modelling and Visualization 2001*, Stuttgart, Germany, November 2001. IEEE.

[13] Ruediger Westermann. A multiresolution framework for volume rendering. In Arie Kaufman and Wolfgang Krueger, editors, *1994 Symposium on Volume Visualization*, pages 51–58, 1994.

[14] Jane Wilhelms and Allen Van Gelder. Octrees for faster isosurface generation. *ACM Transactions on Graphics*, 11(3):201–227, July 1992. ISSN 0730-0301.

[15] Z. J. Wood, M. Desbrun, P. Schröder, and David Breen. Semi-regular mesh extraction from volumes. In *Visualization 2000*, pages 275–282. IEEE, October 2000. ISBN 0-7803-6478-3.

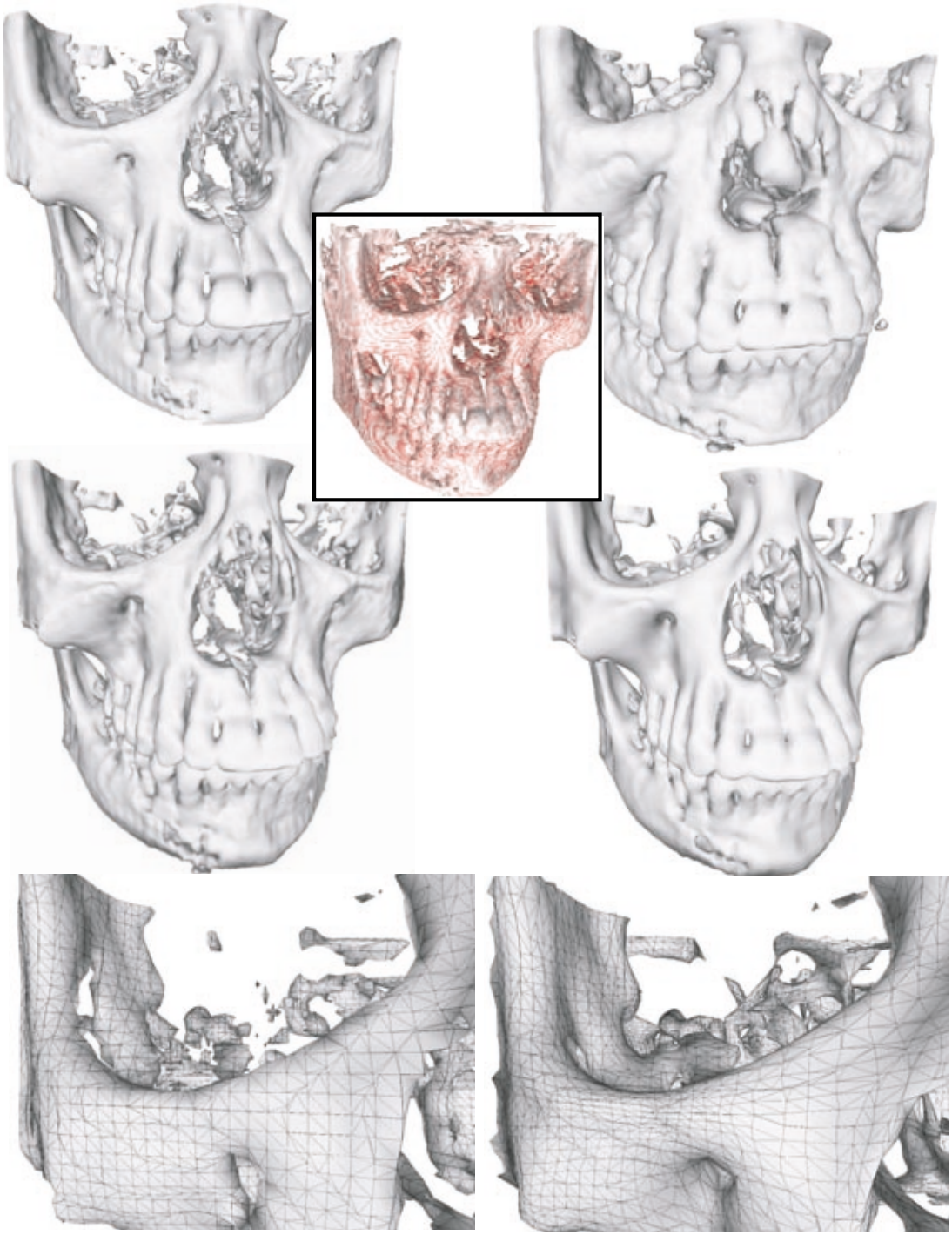


Figure 8: Skull model. First row, left to right: extraction from unoptimized 128^3 volume (186'007 vertices) and warped 128^3 volume (313'840 vertices), respectively. Framed: importance map used for computing the warping function. Second row, left to right: unwarped surface extracted from (warped) 128^3 volume and 103^3 volume (162'762 vertices). Third row, left to right: tessellation of the isosurface extracted from an unoptimized 96^3 volume and optimized 64^3 volume, respectively.

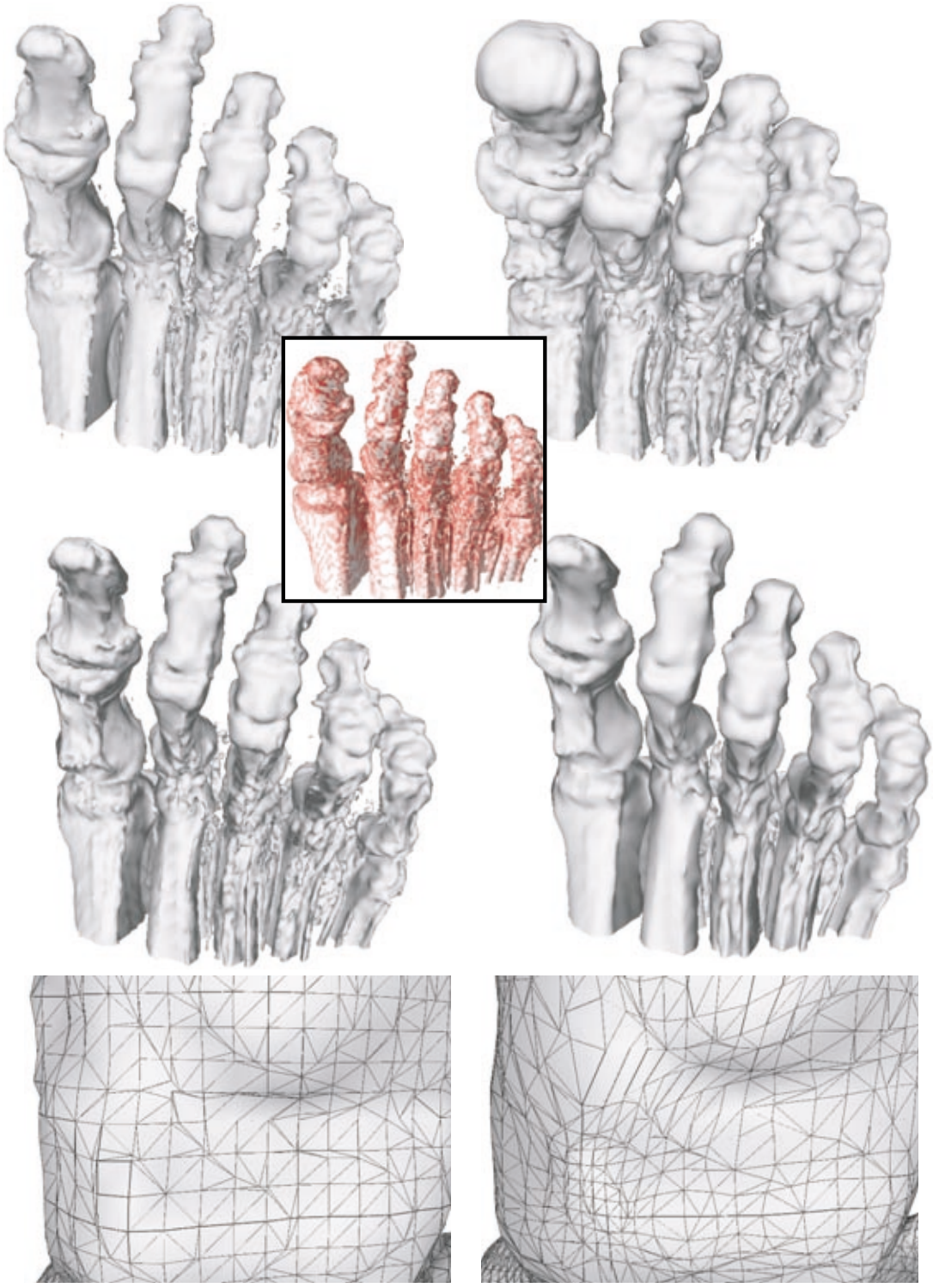


Figure 9: Foot model. First row, left to right: extraction from unoptimized 128^3 volume (125'052 vertices) and warped 128^3 volume (175'890 vertices), respectively. Framed: importance map used for computing the warping function. Second row, left to right: unwarped surface extracted from (warped) 128^3 volume and 116^3 volume (96'935 vertices), respectively. Third row, left to right: tessellation of the isosurface extracted from the unoptimized 128^3 volume and warped 116^3 volume, respectively.