Fast Iso-Surface Extraction using Marching Gradients

Olga Tebeleva
Department of Computer Science, Technion, Haifa, Israel
olgat@csc.cs.technion.ac.il
and
Gershon Elber
Department of Computer Science, Technion, Haifa, Israel
gershon@csa.cs.technion.ac.il

Abstract

Given an explicit trivariate hyper-surface defined over a three dimensional image data set, $\mathcal{D}$, and an iso-surface S_1 of $\mathcal{D}$ at some iso-value v_1, we present an algorithm to extract a new iso-surface S_2 at iso-value v_2, with v_2 sufficiently close to v_1. Off-line continuous reconstruction of $\mathcal{D}$ as a high order Bspline trivariate is employed, yielding a linear time complexity for the extraction of S_2 in the size of the data of iso-surface S_1. The end result allows real-time incremental modification of the iso-value. Hence, users can potentially modify and refine an extracted iso-surface to a precise iso-value, in an interactive manner.

1. Introduction

For the last decade, volumetric data have been widely exploited by medical imaging based applications. Volumetric data have been coined to denote a three dimensional data set, $\mathcal{D}$, that is generated in discrete form, for example with the aid of CT or other medical imaging scanner. Numerous techniques were developed to extract surfaces of constant scalar value out of $\mathcal{D}$, also known as iso-surfaces. The most widespread method, the Marching Cubes (MC) algorithm, was originally presented in [1, 2]. The MC algorithm processes voxels in $\mathcal{D}$ and isolates the ones that interfere with the desired iso-surface level. Then, each isolated voxel is further processed to extract a polygonal approximation of the intersection of the voxel with the iso-surface, using a table driven mechanism. The complete set of polygons that results, approximates the desired iso-surface.

Many variation such as [3, 4] were sought to improve upon the efficiency of the MC algorithm as well as to eliminate the ambiguity problems introduced in the original MC algorithm. Hierarchical data structures, such as octrees [5], multi-resolution approaches [6] and space partitioning methods [7] have been proposed to reduce the complexity of the processing of the enormous amount of information that is involved. Nevertheless, all of these techniques treat $\mathcal{D}$ as a discrete data set. Such an approach is resolution dependent and almost render impossible the exploitation of mathematical methods to extract the surface in a continuous space.

In recent years, some methods have been proposed to represent $\mathcal{D}$ as a continuous hyper-surface in four dimensional Euclidean space, $\mathbb{R}^4$. Early approach describing a trivariate in terms of Bernstein polynomial fitting was made for the purpose of Free-Form Deformation (FFD) [8]. Tuohy et al. suggests the use of Bspline trivariate-based implicit surfaces to partition the domain and detect all the critical points of the hyper-surface yielding a more robust (iso-surface) analysis [9]. Trilinear approximation by Bspline basis functions was developed in [10] to build hierarchical data structures from volumetric data set to speed up MC. Higher order trivariates are also used in [10] to estimate normals for the purpose of Gouraud and/or Phong shading the extracted iso-surface. Herein, we employ similar a approach and exploit higher order trivariate functions fitted to the volumetric data for fast iso-surface *incremental* reconstruction.

In the presented work, complexity considerations served as a major motivation. The naive extraction of an iso-surface, S_1, with a constant scalar iso-value ,v_1, out of the data set, $\mathcal{D}$, requires the processing of the entire data set. If $\mathcal{D}$ is of size $\mathcal{N}$ by $\mathcal{N}$ by $\mathcal{N}$, then $O(\mathcal{N}^3)$ cells are to be processed. Some algorithms were developed to reduce the computation time, using proper preprocessing, such as in [7] down to $O(\sqrt{\mathcal{N}^3} + k)$, where k is the number of voxels intersecting S_1. Let S_2 be another iso-surface with a corresponding iso-value of v_2, where v_2 is sufficiently close to v_1. Even though S_1 and S_2 are similar to each other and both have only local differences in their geometry, one needs the same amount of computation to construct S_2 as was for S_1. It is reasonable to assume that the coherence in the geometry of the iso-surfaces S_1 and S_2 can be exploited to reduce the complexity of computing S_2.

We present an algorithm, called *Marching Gradients (MG)*, that allows the efficient extraction of a new iso-surface, S_2, out of $\mathcal{D}$ and S_1 in time $O(k)$, where k is the number of vertices in S_1. In section 2, we present the Bspline trivariate based approximation of three dimensional data set $\mathcal{D}$. In section 3, we introduce the algorithm itself. In section 4, several results are presented and demonstrated. Finally, in section 5, we conclude.

All the figures are a result of an implementation that is based on the IRIT solid modeling system [11], developed at the Technion.

2. Bspline Trivariate Based Representation

Assume $\mathcal{D}$ is equal in size in all three directions, that is $\mathcal{D}$ is a volumetric data set of size $\mathcal{N}$ by $\mathcal{N}$ by $\mathcal{N}$. For each discrete location $0 \leq i, j, k \leq \mathcal{N} - 1$, on the three dimensional grid of $\mathcal{D}$, a single scalar value is provided as p_{ijk}. Moreover, the original data set, $\mathcal{D}$, can be employed to define an explicit parametric Bspline trivariate function in $\mathbb{R}^4$ over the three dimensional parametric space of $(u, v, w) \subset \mathbb{R}^3$:

$$f(u, v, w) = \sum_{i=0}^{\mathcal{N}-1} \sum_{j=0}^{\mathcal{N}-1} \sum_{k=0}^{\mathcal{N}-1} p_{ijk} B^n_{i,\tau_u}(u) B^m_{j,\tau_v}(v) B^l_{k,\tau_w}(w), \tag{1}$$

where n, m, l are the orders of the trivariate $f(u, v, w)$. Herein, the Bspline basis functions are defined over uniform knot vectors:

$$\tau_u \;\; = \;\; 0,\ 1,\ \ldots,\ \mathcal{N} + n - 1,$$

$$\begin{aligned} \tau_v &= 0,\ 1,\ ...,\ \mathcal{N}+m-1, \\ \tau_w &= 0,\ 1,\ ...,\ \mathcal{N}+l-1, \end{aligned} \tag{2}$$

and $n-1 \leq u < \mathcal{N}$, $m-1 \leq v < \mathcal{N}$, $l-1 \leq w < \mathcal{N}$. Hence after, the knot sequences will be omitted for the purpose of clarity.

Trilinear Bspline approximations are employed in [10]. Trilinear Bsplines are also the only order that interpolate the given data set, $\mathcal{D}$, if $\mathcal{D}$ is employed as the control mesh of the trivariate. Higher order trivariates no longer interpolate $\mathcal{D}$ while they are close to it. One can consider solving an interpolatory problem over the Bspline trivariate, forcing the interpolation of $\mathcal{D}$. While feasible, the result will most likely undulate. Hence, in this work we select to employ the trivariate as an approximation scheme, using $\mathcal{D}$ as the control mesh of the trivariate.

Because $\mathcal{D}$ is, in general, uniformly ordered in three space, uniform knot sequences were selected to all three axes, as seen in Equation (2) . Several possible approaches can handle the boundaries of the defined trivariate over $\mathcal{D}$. One can simply purge the first and last order rows, columns, and planes as less than order basis functions are defined there. Alternatively one can duplicate the boundaries order-1 times, forcing the interpolation of the boundary. A possibly more proper boundary interpolation can be achieved via open end uniform knot sequence, at the lose of uniformity near the boundary. With all these methods, it should be recalled that the boundary is rarely a relevant domain in the medical data and whatever boundary treatment is selected has minor importance in the forecoming analysis.

3. The Marching Gradients Algorithm

In this section, the trivariate function f defined over $\mathcal{D}$ is exploited to solve the following problem: let S_1 be an iso-surface of constant scalar iso-value v_1 extracted out of $\mathcal{D}$; extract a new iso-surface S_2 out of $\mathcal{D}$ with iso-value v_2, such that v_2 is sufficiently close to v_1.

Once the trivariate function f is defined, one can treat f as an explicit continuous function in $\mathbb{R}^4$. A cubic Bspline trivariate with uniform knot vectors guarantee C^2 continuity. Herein, we proposed an approach to the computation of S_2 that exploits the coherence between S_2 and an already computed iso-surface S_1. Given a point $P \in S_1$, we are interesting in finding the position of a nearby point $\tilde{P} \in S_2$. Move from $P \in S_1$ in the direction of the *gradient* of f an amount that is estimated from $v_2 - v_1$ and a first or second order trivariate Taylor expansion around $P \in S_1$.

The gradient vector, ∇f can be easily computed from Equation (1) as follows:

$$\begin{aligned} f_u(u,v,w) &= \frac{\partial f(u,v,w)}{\partial u} = \sum_{i=0}^{\mathcal{N}-1}\sum_{j=0}^{\mathcal{N}-1}\sum_{k=0}^{\mathcal{N}-1} p_{ijk} B_i^{n\prime}(u) B_j^m(v) B_k^l(w), \\ f_v(u,v,w) &= \frac{\partial f(u,v,w)}{\partial v} = \sum_{i=0}^{\mathcal{N}-1}\sum_{j=0}^{\mathcal{N}-1}\sum_{k=0}^{\mathcal{N}-1} p_{ijk} B_i^{n}(u) B_j^{m\prime}(v) B_k^l(w), \\ f_w(u,v,w) &= \frac{\partial f(u,v,w)}{\partial w} = \sum_{i=0}^{\mathcal{N}-1}\sum_{j=0}^{\mathcal{N}-1}\sum_{k=0}^{\mathcal{N}-1} p_{ijk} B_i^{n}(u) B_j^m(v) B_k^{l\prime}(w), \end{aligned} \tag{3}$$

and the gradient $\nabla f = (f_u, f_v, f_w)$.

To find a set of points $\tilde{P} \in S_2$, one needs to solve for $f(u, v, w) = v_2$. Alternatively, by exploiting S_1, one can traverse the entire set of points $P \in S_1$ and *march* in the gradient direction, forming iso-surface S_2. For each point, one calculates a gradient vector (f_u, f_v, f_w) using Equation (3). With the gradient direction, the amount of offset can be obtained from a first order approximation to f, using the modified Newton's formula of,

$$v_1 \approx v_0 + \langle \delta \ \nabla f_N, \nabla f \rangle = v_0 + \delta \|\nabla f\|, \tag{4}$$

where $\|\nabla f\| = \sqrt{f_u^2 + f_v^2 + f_w^2}$, ∇f_N is the normalized gradient: $\nabla f_N = \frac{\nabla f}{\|\nabla f\|}$, $\delta \ \nabla f_N$ is the actual step in $(u, v, w) \subset \mathbb{R}^3$, and $\langle \cdot, \cdot \rangle$ denotes the inner product.

The point $\tilde{P}$ can now be obtained from,

$$\tilde{P} = P + \delta \nabla f_N, \tag{5}$$

or, coordinatewise:

$$\begin{aligned} \tilde{u} &= u + \delta \frac{f_u}{\|\nabla f\|}, \\ \tilde{v} &= v + \delta \frac{f_v}{\|\nabla f\|}, \\ \tilde{w} &= w + \delta \frac{f_w}{\|\nabla f\|}. \end{aligned}$$

In extreme cases when the value of the function changes rapidly along the gradient's direction, the first order approximation might be insufficient. Then, a Taylor's expansion up to the second order can be employed,

$$v_1 \approx v_0 + \langle \delta \ \nabla f_N, \nabla f \rangle + \frac{1}{2} \delta \ \nabla f_N \ H \ \delta \ \nabla f_N, \tag{6}$$

where H is the Hessian matrix of second partial derivatives:

$$H = \begin{bmatrix} f_{uu} & f_{uv} & f_{uw} \\ f_{vu} & f_{vv} & f_{vw} \\ f_{wu} & f_{wv} & f_{ww} \end{bmatrix}.$$

δ is computed using the quadratic Equation (6). The root with the same sign as the offset value from Equation (4) should be selected.

In the proposed algorithm, we ignore the possible topological changes during the transition. That is, f has no points of local minimum/maximum between the v_1 and the v_2 iso-values. While we cannot guarantee the convergence of the MG method at all times, for local regions and for small iso-value steps the proposed method can yield adequate results. Moreover, interactive modification of the iso-value is now a feasible option than ever with the introduced linear time complexity, using the MG algorithm. The MG algorithm is adequate for the task of fine tuning the iso-surface's iso-value when changes occur in very small steps.

The computation of Equations (3), (5), and (6) seems intensive for each and every point $P \in S_1$. Yet, proper preprocessing can alleviate most of the overhead. Given a Bspline trivariate f, its gradient vector can be computed symbolically. That is, the three functions of $\nabla f = (f_u, f_v, f_w)$ can all be precomputed as three Bspline trivariates, a-priori. Moreover, the second order derivatives of f, in H, can similarly be computed before hand. Then, at interaction time, only trivariate evaluation operations are conducted, of f and its first and second order derivative functions. It should be recalled that the derivative of a (piecewise) polynomial function is a (piecewise) polynomial function of lower order. Finally, and because uniform knot sequences are employed, the Bspline basis functions of f as well as the basis functions of f's first and second order derivatives can all be precomputed into tables, further reducing the total evaluation times in actual interaction.

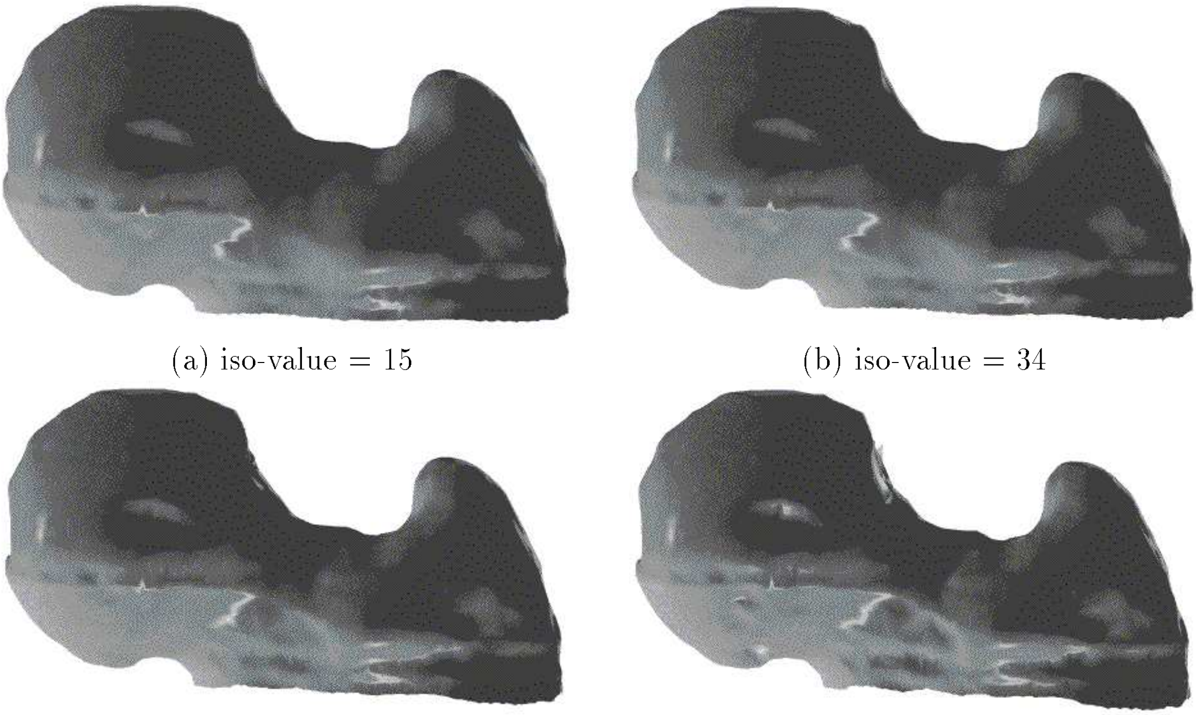
One can compute S_2 by successively applying Equations (4) or (6) to the entire set of points $P \in S_1$. However, increasing the number of iterations, while improving the accuracy of approximation of S_2, affects the interactivity of the algorithm. In practice, a single iteration yields sufficient accuracy in most instances. The dependencies between the computation error, the computation time, and the number of iterations will all be considered in the next section, Section .

4. Applications and Examples

Given an iso-surface S_1, one needs to extract a new iso-surface, S_2, where iso-value v_2 is sufficiently close to v_1, by successively applying the MG algorithm and obtaining the desired iso-surface. Such operations are sometimes necessary for the refinement and fine-tuning of an iso-surface for different, closely related iso-values. Two sequences of the successive transformations for increasing iso-values are presented in the Figures 1 and 2 of a femoral head model resulting from a medical scan and the cuboid implicit surface of $x^4 + y^4 + z^4 - x^2 - y^2 - z^2 = c$, respectively.

The orders of the trivariate, the number of vertices in the iso surface, and the number of iterations computed are all factors that affect the accuracy, quality, and efficiency of the resulting iso-surface. In Table 1, the extraction times for several resolutions of the model of the cuboid data set (See Figure 2) with different number of vertices are compared to the Marching Cubes algorithm. Iso-surface computation using only first order approximation (Equation (4)) of MG, and using second order MG (Equation (6)) are both compared to MC using various number of vertices. This comparison shows a speed up of close to 100% for second order MG approximation and more than 100% if only a first order MG approximation is employed. In Table 2, the convergence rate as more iterations are computed is examined, only to conclude that for practical and interactive use, a single iteration is more than adequate.

A side effect of the use of the trivariate's representation is the alleviation of aliasing in the medical images, following [10]. Most objects extracted out of the volumetric data have severe aliasing problem due to the discrete nature of the input data and



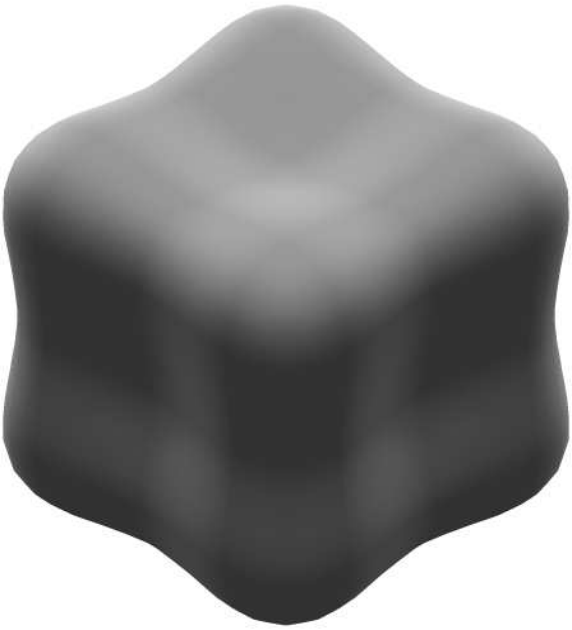
(a) iso-value = 15 (b) iso-value = 34

(c) iso-value = 48 (d) iso-value = 76

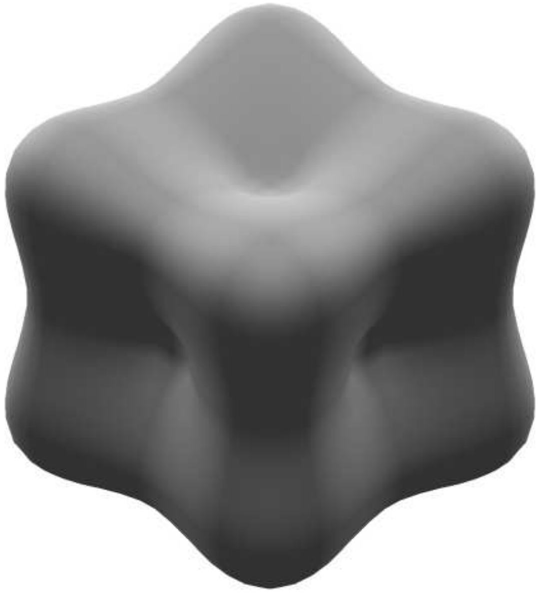
Figure 1: A sequence of consecutive MG transformations of a femoral head. (a) was extracted using MC and (b) to (d) were derived using MG.

Table 1: Iso-surface extraction time for the Cuboid data set (See Figure 2) using a Bspline trivariate of order 4.

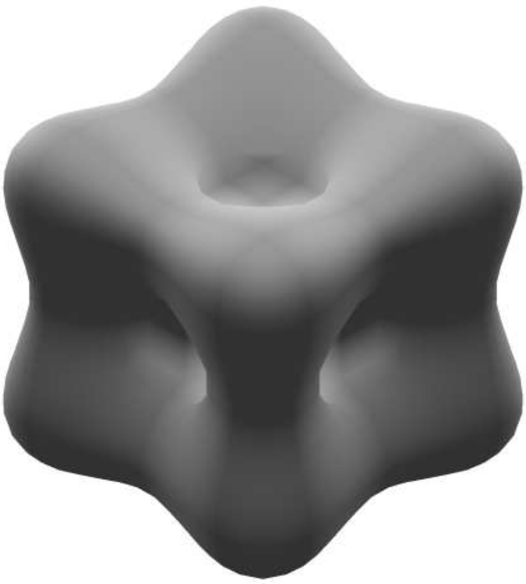
# Vertices	MG (MSec) (First Order)	MG (MSec) (Second Order)	MC (MSec)
216	90	140	120
458	190	270	350
936	360	540	700
1202	460	710	1120
2256	880	1330	2170



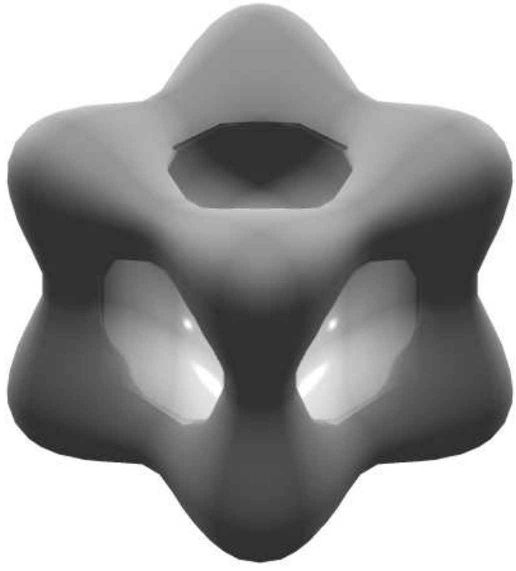
(a) c = 0



(b) c = -0.28



(c) c = -0.35



(d) c = -0.42

Figure 2: A sequence of consecutive MG transformations of $x^4+y^4+z^4-x^2-y^2-z^2=c$. Compare with Figure 4.

Table 2: Relative error (first order) of Cuboid data set (See Figure 2), order = 4, # vertices = 458.

Change of Iso-value	One iteration	Two iterations	Three iterations
-0.00 $\rightarrow$ -0.07	3.610%	0.488%	0.450%
-0.07 $\rightarrow$ -0.14	1.998%	0.206%	0.181%
-0.14 $\rightarrow$ -0.21	1.376%	0.114%	0.111%
-0.21 $\rightarrow$ -0.28	1.300%	0.219%	0.083%
-0.28 $\rightarrow$ -0.35	1.037%	0.058%	0.036%

low quality estimation of *normals*. Figure 3(a) represents a polygonal model of a femur, obtained by applying the Marching Cubes algorithm.

One can significantly improve this image (compare with Figures 3(b) and 3(c)) by estimating normals to each point on the polygonal surface of the femur, exploiting the Bspline trivariate approximation of different order. That is, each such point is considered a point in the implicit trivariate surface. The normal to the point is calculated as a normalized gradient, following Equation (3). Evidently, the continuous Bspline trivariate serves as a low pass filter and the approximation results in an improved normal estimation compared to the discrete differencing approach exploited in MC. The selection of higher degree trivariates that act as stronger low pass filters can produce smoother normal fields, at the expense of less accurate model and more expensive over whole computation. This same approach was exploited in [10] and similar results were obtained.

5. Conclusion

An algorithm for the computation of a new iso-surface from an existing iso-surface has been developed. The presented method allows one to compute each point of the new iso-surface using only local neighborhood of $\mathcal{D}$. The MG algorithm significantly reduces the time complexity of the extraction of the iso-surface. Initial iso-surface computation, applying the traditional Marching Cubes algorithm, can be viewed as a preprocessing stage with time complexity $O(\mathcal{N}^3)$, possibly optimized to a better time such as $O(\sqrt{\mathcal{N}^3} + k)$. Herein, the computation of any new sufficiently close iso-surface has a linear time complexity in terms of the number of points forming the initial iso-surface.

The MG algorithm has a significant drawback. It cannot properly follow the topological changes of the iso-surface. Figure 4 represents the results of applying the MC algorithm to the cuboid's surface with different iso-values. It is clear that the surface in Figure 2(b) differs from the surface at the Figure 4(b), because of the topological changes that took place in the iso-surface. Moreover, as more and more MG transformations take place, severe distortions in the surface's topology can be accumulated (see Figure 2(c), 2(d)). A possible solution to this accumulated error

(a) Original view (b) Order = 4 (c) Order = 8

Figure 3: Aliasing elimination in normal computation using a Bspline trivariate of the specified order. (a) is computed using regular MC while (b) and (c) are MC with normals estimated from the Bspline trivariate of the specified order.

is to interleave the MC and MG algorithms, applying MC when distortions in the iso-surface's topology are no longer acceptable and/or interactive manipulation is not a high priority.

While MG can be combined with MC to reconstruct the correct topology, Marching Cubes is more expensive computationally. Proper analysis of local extreme values of $\mathcal{D}$ can yield the locations where new branches should be formed or old branches should be eliminated. Holes formed within the same connected branch are harder to detect and process, and more effort should be invested into this topological analysis. Both adaptive polygonal refinement when the iso-surface's complexity is increasing and adaptive polygonal decimation when the iso-surface's complexity is declining should be considered, while the Gradient Marching is continuously applied. The sought result should be the ability to exploit only the Marching Gradient algorithm throughout while properly maintaining the topology, yielding the correct iso-surface in all instances.

References

[1] W. E. Lorensen and H. E. Cline. Marching Cubes: A High Resolution 3D Surface Construction Algorithm. *Computer Graphics (SIGGRAPH'87 Proc.)*, Vol 21, No. 4, pp 163-169, July 1987.

[2] G. Wyvill, C. McPheeters and B. Wyvill. Data Structure for Soft Objects. *The Visual Computers*, Vol 2, No. 4, pp 227-234, August 1986.

[3] P. Ning and J. Bloomenthal. An Evaluation of Implicit Surface Tiler. *IEEE Computers Graphics and Applications*, Vol 13, No. 6, pp 33-41, November 1993.

[4] O. Micami and Y. Ohno. A Surface Construction Algorithm Based on Volume Data. *Proceedings of 6th ICECGDG*, pp 375-379, August 1994.

[5] D. J. Meagher, Modeling Using Octree Encoding. *Computer Graphics and Image Processing 19*, 2 (June 1982), 129-147.

[6] P. Gignoni, L. D. Floriani, C. Montani, E. Puppo and R. Scopingo. Multiresolution Modeling and Visualization of Volume Data Based on Simplicial Complex. *Proceedings of the 1994 Symposium on Volume Visualization*, pp 19-26, Washington D.C., October 1994.

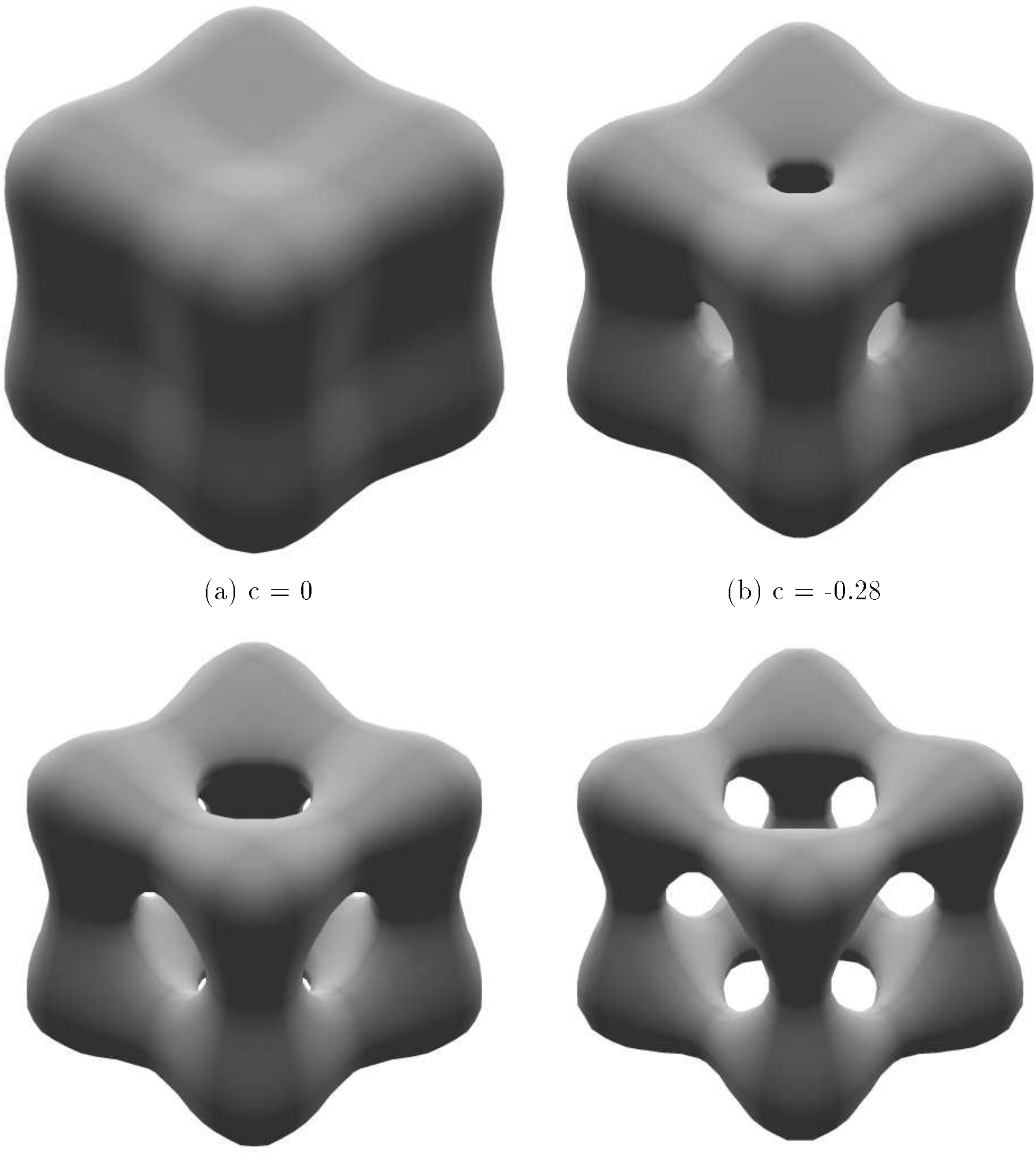
[7] Y. Livnat, H. W. Shen and C. R. Johnson. A near optimal iso-surface extraction algorithm for structured and unstructured grids. *IEEE Trans. Vis. Comp. Graphics* , 1995.

[8] T. W. Sederberg and S. R. Parry. Free-Form Deformation of Solid Geometric Models. *Computer Graphics (SIGGRAPH'86 Proc.)*, Vol. 20, No 4, pp 151-160, August 1986.

[9] S. T. Tuony, J. W. Yoon and N. M. Patrikalakis. Reliable Interrogation of 3D Non-linear Geophysical Databases. Computer Graphics: Developments in Virtual Environment, Academic Press Ltd, pp 327-341, 1995.

[10] G. Elber, T. Malzbender. Fast Iso-Surface Extraction using A-priori Volumetric Data Processing. *HP Lab. Tech. Report*, April, 1995.

[11] IRIT. Irit 5.0 User's Manual, Technion, March 1995.



(a) c = 0 (b) c = -0.28

(c) c = -0.35 (d) c = -0.42

Figure 4: A sequence of consecutive application of MC algorithm for surface $x^4 + y^4 + z^4 - x^2 - y^2 - z^2 = c$. Compare with Figure 2.