Fast Iso-Surface Extraction using A-priori Volumetric Data Processing

Gershon Elber
HP Israel Science Center*

and

Tom Malzbender
Computer Systems Lab†

Keywords: Trivariates, B-spline, Marching Cubes, Octrees, Sub-Linear Complexity.

Abstract

Given an image data set, $\mathcal{D}$, defined as an explicit trivariate hypersurface in four space, we present an algorithm to extract arbitrary iso-surfaces out of $\mathcal{D}$ in sub-linear complexity, in terms of the number of voxels in $\mathcal{D}$. Off line construction of an hierarchical high order trivariate fit to $\mathcal{D}$ with arbitrary precision extends a similar notion of Octree optimization of three dimensional images. During the interaction stage, the hierarchy is employed so arbitrary iso-surfaces can be extracted in a sub-linear time, and with maintained accuracy.

*Address: HP Israel Science Center, Technion City, Haifa 32000, Israel. Email: gershon@hp.technion.ac.il

†Address: Hewlett Packard Laboratories, 1501 Page Mill Road, Bldg 1U, Palo Alto, CA 94304, U.S.A. Email: tom_malzbender@hpl.hp.com

1 Introduction

Let $\mathcal{D}$ be a voxel data set of size $\mathcal{W}$ by $\mathcal{W}$ by $\mathcal{W}$. For each discrete location $P(i,j,k)$, $0 \leq i,j,k < \mathcal{W}$, on the three dimensional grid, $\mathcal{D}$, a single scalar value is provided as p_{ijk}. Throughout this paper, and unless otherwise stated, we assume $\mathcal{D}$ is equal in size in all the three of its axes, for simplicity.

Marching cubes [8, 18], was developed to provide a mechanism to extract surfaces of constant scalar value throughout the prescribed volume, also known as iso-surfaces. The marching cube technique as originally presented in [8, 18], and in many of its derivatives [5, 12, 16], processes voxels in $\mathcal{D}$ and isolates the ones that interfere with the desired iso-surface level. Then, the marching cubes algorithm continues to process the isolated voxels and extracts a polygonal approximation of the intersection of the voxel with the iso-surface, using a table driven mechanism. The complete set of polygons that result, approximates the desired iso-surface.

Recent research [10, 2, 14] suggests the treatment of medical images as hypersurfaces in four dimensional Euclidean space. In [2], the domain is subdivided into tetrahedral simplices and multiresolution is achieved through the decimation of and manipulation of the tetrahedra. Multiresolution takes a major role in recent research [2, 7, 17] due the enormous amount of information that is involved. In [11], trilinears are employed to create a topologically consistent piecewise linear approximation to the isosurface. In [14], the use of trivariates is geared toward robust preprocessing analysis of critical points of the trivariate, alleviating the drawbacks in the traditional marching cube's approach. Herein, we employ similar approaches, but exploit higher order trivariate functions fitted to the volumetric data, in order to further optimize the isosurface extraction as well as achieve better rendering results via more accurate normal computation.

The original data set, $\mathcal{D}$, can be viewed as an explicit trilinear parametric B-spline function in four space defined over the three dimensional parametric space of $u_0 \leq u \leq u_1$, $v_0 \leq v \leq v_1$, $w_0 \leq w \leq w_1$,

$$\mathcal{D}(u,v,w) = \sum_{i=0}^{\mathcal{W}-1} \sum_{j=0}^{\mathcal{W}-1} \sum_{k=0}^{\mathcal{W}-1} p_{ijk} B^2_{i,\tau}(u) B^2_{j,\tau}(v) B^2_{k,\tau}(w), \tag{1}$$

and a uniform open end knot vector $\tau = (0, 0, 1, 2, ..., \mathcal{W}-2, \mathcal{W}-1, \mathcal{W}-1)$. p_{ijk} are the scalar coefficient of the data. Clearly, this representation can be extended to higher orders. $\mathcal{D}$ in (1) is frequently exploited, although $\mathcal{D}$ is not always represented as in Equation (1). The marching cube technique assumes an axis-independent linear interpolation throughout the interior of each voxel.

Let $\mathcal{D}_1 : p_{ijk} = 1,\ 0 \leq i,j,k < \mathcal{W}$. Any iso-surface that is naively computed for $\mathcal{D}_1$, at an iso-level other than one, would require the processing of the entire data set, $O(\mathcal{W}^3)$ voxels over whole, only to return empty handed. Now set p_{000} to zero and call this new data set $\mathcal{D}_2$. Clearly, every iso-surface at a value between zero and one intersects $\mathcal{D}_2$. only a neighborhood of 8 voxels of $\mathcal{D}_2$ contributes to the iso-surface, while $O(\mathcal{W}^3)$ voxels are being processed.

One can alleviate this overhead. Assume, for simplicity and throughout this paper, that $\mathcal{W} = 2^n$, $n \in \mathcal{Z}^+$. Compute the two extremum values of $p_{max} = \underset{i,j,k}{max}\ (p_{ijk})$, $p_{min} = \underset{i,j,k}{min}\ (p_{ijk})$.

Algorithm 1

Input:
$\mathcal{D}$, a volumetric data set of size w by w by w, $w = 2^n$, $n \in \mathcal{Z}^+$.
Output:
$\mathcal{H}$, an hierarchical subdivision of $\mathcal{D}$ of scalar domain bounds.
Algorithm:

```
VolumetricDomainBound( D )

    H.p_min  <=  min_{i,j,k} (p_ijk);

    H.p_max  <=  max_{i,j,k} (p_ijk);
    if ( Width(D) > 1 ) then
    begin
        D(i), 0 <= i <= 7 <= Eight sub-volumes D in an hierarchical
                                                   subdivision;
        H.SubVol(i) <= VolumetricDomainBound( D(i) ), 0 <= i <= 7;
    end;
    return H;
end.
```

Construction of hierarchical min/max values of sub-volumes.

The set of polygons approximating the iso-surface at a value above p_{max} or below p_{min} is an empty set. Subdivide $\mathcal{D}$ into sub-volumes in x, y, and z, eight sub-volumes in all. Recursively apply the extremum test computation to the eight sub-volumes of $\mathcal{D}$ that are formed. This spatial subdivision based computation can be made uniform or non uniform, possibly becoming adaptable to the geometry of the data set [16]. See Algorithm 1.

By exploiting the resulting hierarchy of $\mathcal{H}$ (Algorithm 1), one can trivially reject the constant one volumetric data set, $\mathcal{D}_1$, that is iso-surfaced at a level of a half. Moreover, using $\mathcal{H}$ one can efficiently converge to the single voxel for which $p_{ijk} = 0$, in $\mathcal{D}_2$. $O(log(w))$ steps would be required to step through the entire hierarchy of $\mathcal{H}$ and reach the only non constant voxel in $\mathcal{D}_2$ that intersects the iso-surface at a half.

Nonetheless, while the use of an hierarchy that bounds p_{ijk} alleviates the complexity involved in the extraction of the iso-surface, it inherits the same deficiency of the original data representation by remaining a piecewise triconstant approximation. Triconstant hierarchy is used, for example, in [7]. Quoting from [16], "we are not dealing with an object, or a small set of objects, which occupies a possibly small portion of the volume, but rather a function that is defined throughout the volume". Consider the data set $\mathcal{D}_3 : \ p_{ijk} = ijk, \ 0 \leq i, j, k < w$. $\mathcal{H}$ is going to form into a nontrivial hierarchy, when Algorithm 1 is applied to $\mathcal{D}_3$. Yet, a single trilinear can exactly represent the data set of $\mathcal{D}_3$, for an arbitrary width w.

Assume one is able to form an hierarchy of higher order approximations of trivariate functions to the image data, bounded by a preset tolerance. Furthermore, assume one is able to iso-surface each of the higher order trivariate functions in the hierarchy, again with a bounded tolerance. Then, one might be able to create a compact and dense hierarchical representation, compared to the triconstant hierarchical approach while entertaining faster iso-surface extraction times, compared to both the non hierarchical marching cubes or the triconstant hierarchical methods.

In this work, we consider the construction of an hierarchical representation of higher order trivariate functions. This paper is organized as follows. Section 2 describes the approach employed to form the higher order trivariate hierarchy with a prescribed accuracy. Section 3 considers the problem of extracting a polygonal iso-surface approximation from the hierarchy of trivariates, again with a bounded tolerance. Finally, in Section 4, examples are presented with the appropriate statistics. We conclude in Section 5.

This work took the advantage of the trivariate package of the IRIT [6] solid modeling system, developed at the Technion.

2 An Hierarchical Trivariate Fit

Let $\mathcal{D}_1(u, v, w)$ and $\mathcal{D}_2(r, s, t)$ be two explicit parametric trivariates in the four space of $u_0 \leq u, r \leq u_1,\ \ v_0 \leq v, s \leq v_1,\ \ w_0 \leq w, t \leq w_1$. Then,

Definition 1 *The distance in four space between point $(u_0, v_0, w_0, \mathcal{D}_i(u_0, v_0, w_0))$ and trivariate $(r, s, t, \mathcal{D}_j(r, s, t))$ is*

$$\begin{aligned} & Dist\left((u_0, v_0, w_0, \mathcal{D}_i(u_0, v_0, w_0)), (r, s, t, \mathcal{D}_j(r, s, t))\right) \\ & \quad = \min_{r,s,t} \ ||(u_0, v_0, w_0, \mathcal{D}_i(u_0, v_0, w_0)) - (r, s, t, \mathcal{D}_j(r, s, t))||, \end{aligned} \tag{2}$$

where $\|\cdot\|$ denotes the L_2 norm. Then,

Definition 2 *The distance in four space between the two trivariates of $(u, v, w, \mathcal{D}_i(u, v, w))$ and $(r, s, t, \mathcal{D}_j(r, s, t))$ is*

$$\begin{aligned} & Dist\left((u, v, w, \mathcal{D}_i(u, v, w)), (r, s, t, \mathcal{D}_j(r, s, t))\right) \\ & \quad = \max_{u,v,w} \ \min_{r,s,t} \ ||(u, v, w, \mathcal{D}_i(u, v, w)) - (r, s, t, \mathcal{D}_j(r, s, t))||. \end{aligned} \tag{3}$$

In other words, we define the distance between two (explicit in four space) trivariates as an upper bound over all points in $\mathcal{D}_i$, of the (minimal) distance between the point in $\mathcal{D}_i$ and trivariate $\mathcal{D}_j$.

Algorithm 2

Input:

$\mathcal{D}$, a volumetric data set of size $\mathcal{W}$ by $\mathcal{W}$ by $\mathcal{W}$, $\mathcal{W} = 2^n$, $n \in \mathcal{Z}^+$.

ϵ, tolerance of hierarchical fit.

Output:

$\mathcal{H}$, an hierarchical subdivision of $\mathcal{D}$ of trilinear fits to within ϵ.

Algorithm:

VolumetricTriilinearFit($\mathcal{D}$, ϵ)

$\hat{\mathcal{D}}(r,s,t) \Leftarrow \sum_{i=0,\mathcal{W}-1} \sum_{j=0,\mathcal{W}-1} \sum_{k=0,\mathcal{W}-1} p_{ijk} B^2_{i,\xi}(r) B^2_{j,\xi}(s) B^2_{k,\xi}(t)$, $p_{ijk} \in \mathcal{D}$;

(1) if ($Dist(\mathcal{D}(u,v,w), \hat{\mathcal{D}}(r,s,t)) > \epsilon$) then

$\mathcal{D}(i)$, $0 \le i \le 7 \Leftarrow$ Eight sub-volumes of $\mathcal{D}$ in an hierarchical subdivision;

$\mathcal{H}.SubVol(i) \Leftarrow$ **VolumetricTrilinearFit**($\mathcal{D}(i)$), $0 \le i \le 7$;

else

$\mathcal{H}.\mathcal{D} \Leftarrow \mathcal{D}$

endif;

return $\mathcal{H}$;

end.

Construction of trilinear hierarchical approximation of volume data set.

Let $\mathcal{D}$ be a voxel data set of size $\mathcal{W}$ by $\mathcal{W}$ by $\mathcal{W}$. We consider the simplest yet practical case of a trilinear. Construct a trilinear over the eight corners of $\mathcal{D}$,

$$\hat{\mathcal{D}}(u,v,w) = \sum_{i=0,\mathcal{W}-1} \sum_{j=0,\mathcal{W}-1} \sum_{k=0,\mathcal{W}-1} p_{ijk} B^2_{i,\xi}(u) B^2_{j,\xi}(v) B^2_{k,\xi}(w), \tag{4}$$

and $\xi = (0, 0, \mathcal{W}-1, \mathcal{W}-1)$.

Assume one is able to bound the distance (Definition 2) between $\mathcal{D}(u,v,w)$ (Equation (1)) and $\hat{\mathcal{D}}(u,v,w)$ (Equation (4)) computed using the eight corners of $\mathcal{D}(u,v,w)$. Given the volumetric data set, $\mathcal{D}$, and a tolerance $\epsilon > 0$ one can approximate $\mathcal{D}$ by fitting a trilinear, $\hat{\mathcal{D}}$, to the eight corners of $\mathcal{D}$. If the computed distance between the two trivariates is larger than ϵ, $\mathcal{D}$ is subdivided into eight sub-volumes and trilinears are fitted to their corners, recursively. Otherwise, if the fit is sufficiently tight, the construction terminates. See Algorithm 2.

Hence, we are left with the key question of bounding the distance between a trivariate B-spline volumetric data set, $\mathcal{D}$, and a trilinear that has been constructed from the eight corners of $\mathcal{D}$, $\hat{\mathcal{D}}$.

Assume $\mathcal{D}$ is a trilinear. Let $\overline{\mathcal{D}}$ be $\hat{\mathcal{D}}$ refined [1, 3] in all three axes at all the interior knots $t_i \in \tau$ of $\mathcal{D}$, $1 < i < \mathcal{W}-1$. Then, $\overline{\mathcal{D}}$ and $\mathcal{D}$ share the same function space. That is, they are both trilinear and both possess the same continuity as is prescribed by τ.

Lemma 1 *The distance in four space between*

$$\overline{\mathcal{D}}(u,v,w)=\sum_{i=0}^{\mathcal{W}-1}\sum_{j=0}^{\mathcal{W}-1}\sum_{k=0}^{\mathcal{W}-1}\overline{p}_{ijk}B_{i,\tau}^{2}(u)B_{j,\tau}^{2}(v)B_{k,\tau}^{2}(w),$$

and

$$\mathcal{D}(r,s,t)=\sum_{i=0}^{\mathcal{W}-1}\sum_{j=0}^{\mathcal{W}-1}\sum_{k=0}^{\mathcal{W}-1}p_{ijk}B_{i,\tau}^{2}(r)B_{j,\tau}^{2}(s)B_{k,\tau}^{2}(t),$$

where

$$u_0\le u,\ r\le u_1,\quad v_0\le v,\ s\le v_1,\quad w_0\le w,\ t\le w_1,$$

is bounded by,

$$Dist\left(\overline{\mathcal{D}}(u,v,w),\mathcal{D}(r,s,t)\right)\le\max_{i,j,k}\left(\left|\overline{p}_{ijk}-p_{ijk}\right|\right).$$

Proof:

$$\begin{aligned}
&Dist(\overline{\mathcal{D}}(u,v,w),\mathcal{D}(r,s,t))\\
&=\max_{u,v,w}\min_{r,s,t}\left\|\left(u,v,w,\overline{\mathcal{D}}(u,v,w)\right)-(r,s,t,\mathcal{D}(r,s,t))\right\|\\
&\le\max_{u,v,w}\left\|\left(u,v,w,\overline{\mathcal{D}}(u,v,w)\right)-(u,v,w,\mathcal{D}(u,v,w))\right\|\\
&=\max_{u,v,w}\left|\overline{\mathcal{D}}(u,v,w)-\mathcal{D}(u,v,w)\right|\\
&=\max_{u,v,w}\left|\sum_{i=0}^{\mathcal{W}-1}\sum_{j=0}^{\mathcal{W}-1}\sum_{k=0}^{\mathcal{W}-1}\overline{p}_{ijk}B_{i,\tau}^{2}(u)B_{j,\tau}^{2}(v)B_{k,\tau}^{2}(w)\right.\\
&\qquad\left.-\sum_{i=0}^{\mathcal{W}-1}\sum_{j=0}^{\mathcal{W}-1}\sum_{k=0}^{\mathcal{W}-1}p_{ijk}B_{i,\tau}^{2}(u)B_{j,\tau}^{2}(v)B_{k,\tau}^{2}(w)\right|\\
&=\max_{u,v,w}\left|\sum_{i=0}^{\mathcal{W}-1}\sum_{j=0}^{\mathcal{W}-1}\sum_{k=0}^{\mathcal{W}-1}\left(\overline{p}_{ijk}-p_{ijk}\right)B_{i,\tau}^{2}(u)B_{j,\tau}^{2}(v)B_{k,\tau}^{2}(w)\right|\\
&\le\max_{u,v,w}\left|\sum_{i=0}^{\mathcal{W}-1}\sum_{j=0}^{\mathcal{W}-1}\sum_{k=0}^{\mathcal{W}-1}\max_{i.j.k}\left(\left|\overline{p}_{ijk}-p_{ijk}\right|\right)B_{i,\tau}^{2}(u)B_{j,\tau}^{2}(v)B_{k,\tau}^{2}(w)\right|\\
&=\max_{i,j,k}\left(\left|\overline{p}_{ijk}-p_{ijk}\right|\right)\max_{u,v,w}\left|\sum_{i=0}^{\mathcal{W}-1}\sum_{j=0}^{\mathcal{W}-1}\sum_{k=0}^{\mathcal{W}-1}B_{i,\tau}^{2}(u)B_{j,\tau}^{2}(v)B_{k,\tau}^{2}(w)\right|\\
&=\max_{i,j,k}\left(\left|\overline{p}_{ijk}-p_{ijk}\right|\right).
\end{aligned}\tag{5}$$

■

Therefore, line (1) in Algorithm 2 reduces into first refining the trivariate function of $\hat{\mathcal{D}}$ and elevating it to the function space of $\mathcal{D}$, and then computing the maximum over the difference between the corresponding coefficients p_{ijk}.

If either $\mathcal{D}$ or both $\mathcal{D}$ and $\hat{\mathcal{D}}$ are higher order trivariates, the exact same steps of elevating $\hat{\mathcal{D}}$ to the same function space of $\mathcal{D}$ must be followed. In the trilinear case, it required the refinement of $\hat{\mathcal{D}}$ at the proper knot values. If $\mathcal{D}$ and $\hat{\mathcal{D}}$ do not share the same degree, it also requires the degree raising of $\hat{\mathcal{D}}$ to the proper degree of $\mathcal{D}$.

Corollary 1 *The distance in four space between*

$$\overline{\mathcal{D}}(u, v, w) = \sum_{i=0}^{\mathcal{W}-1} \sum_{j=0}^{\mathcal{W}-1} \sum_{k=0}^{\mathcal{W}-1} \overline{p}_{ijk} B^l_{i,\tau}(u) B^m_{j,\tau}(v) B^n_{k,\tau}(w),$$

and

$$\mathcal{D}(r, s, t) = \sum_{i=0}^{\mathcal{W}-1} \sum_{j=0}^{\mathcal{W}-1} \sum_{k=0}^{\mathcal{W}-1} p_{ijk} B^l_{i,\tau}(r) B^m_{j,\tau}(s) B^n_{k,\tau}(t),$$

where

$$u_0 \leq u,\ r \leq u_1, \quad v_0 \leq v,\ s \leq v_1, \quad w_0 \leq w,\ t \leq w_1,$$

is bounded by,

$$Dist\left(\overline{\mathcal{D}}(u, v, w), \mathcal{D}(r, s, t)\right) \leq \max_{i,j,k} \left(\left|\overline{p}_{ijk} - p_{ijk}\right|\right).$$

A uniform trilinear B-spline interpolates all the data points of the volumetric data set. Let $\mathcal{D}_3$ be a data set with one interior coefficient equal to one while all other coefficients are identically zero. The trilinear $\hat{\mathcal{D}}_3$ is identically zero and hence its coefficients as well as the coefficients of $\overline{\mathcal{D}}_3$ are all zero. Therefore, the bound established in Lemma 1 can be tight. This is not necessarily the case for higher order trivariates because the B-spline basis functions are no longer interpolatory. Better bounds can be established for higher order B-spline trivariates in a similar way, but they will not be as tight, in general.

Once the hierarchy has been computed, for a prescribed tolerance, the extraction of a polygonal approximation of an iso-surface at a given value requires the ability to form a polygonal approximation for the iso-surface set of a trivariate. Section 3 addresses this issue.

3 Polygonal Approximation for a Constant Set of a Trivariate

Let $\mathcal{D}$ be a trivariate that needs to be iso-surfaced at a given value, using a polygonal approximation and a prescribed accuracy. In Section 2, a trilinear $\hat{\mathcal{D}}$ is constructed from the eight corners of $\mathcal{D}$, and using refinement as well as degree raising, $\hat{\mathcal{D}}$ was elevated to the function space that $\mathcal{D}$ is in. Their coefficients were then compared, and a bound on the distance between them was established. Herein, we consider the next question of extracting a polygonal approximation to an iso-surface from the higher order hierarchy.

$S(u,v) = P_{00}(1-u)(1-v) + P_{01}(1-u)v + P_{10}u(1-v) + P_{11}uv, \quad 0 \leq u,v \leq 1$ is a bilinear surface defined over the four points P_{ij}. In [4], it is shown that the distance between $S(u,v), \quad 0 \leq u,v, \quad u+v \leq 1$ and triangle $\overline{P_{00}P_{01}P_{10}}$ is bounded from above by $\frac{1}{4}$ of the distance from P_{11} to the plane containing $\overline{P_{00}P_{01}P_{10}}$. Herein, we extend this result a dimension, for a trilinear in four space. Let,

$$\mathcal{D}(u,v,w) = \sum_{i=0,1} \sum_{j=0,1} \sum_{k=0,1} p_{ijk} B^2_{i,\tau}(u) B^2_{j,\tau}(v) B^2_{k,\tau}(w), \tag{6}$$

be a trilinear defined over a single voxel. Denote by $\mathcal{P}$ the hyperplane in four-space through the four points $P_{000} = (0,0,0,p_{000})$, $P_{001} = (0,0,1,p_{001})$, $P_{010} = (0,1,0,p_{010})$, $P_{100} = (1,0,0,p_{100})$.

Lemma 2 *The distance in four space between $\mathcal{D}(u,v,w), \quad 0 \leq u,v,w, \quad u+v+w \leq 1$ and the hyperplane $\mathcal{P}$ is bounded from above by*

$$\Delta = \frac{d_{011} + d_{101} + d_{110}}{4} + \frac{d_{111}}{27}, \tag{7}$$

where d_{ijk} denotes the distance between point P_{ijk} and plane $\mathcal{P}$, in four space.

Proof: From the construction of $\mathcal{P}$, it is clear that $d_{000} = d_{001} = d_{010} = d_{100} = 0$. Denote the axes of the four space by X, Y, Z, W. Rotate and translate $\mathcal{D}(u,v,w)$ into $\tilde{\mathcal{D}}$ so that $\mathcal{P}$ becomes the $W = 0$ hyperplane. Let $\tilde{p}_{ijk}$ be the coefficients of $\tilde{\mathcal{D}}$. Because form (6) is invariant under rigid motion, the distance between $\mathcal{D}(u,v,w)$ and $\mathcal{P}$ is the same as the distance between $\tilde{\mathcal{D}}$ and the hyperplane $W = 0$. Hence, this distance is equal to the value of the W axis or the value of $\tilde{\mathcal{D}}$,

$$\begin{aligned}
\Delta(u,v,w) &= \tilde{p}_{011}(1-u)vw + \tilde{p}_{101}u(1-v)w + \tilde{p}_{110}uv(1-w) + \tilde{p}_{111}uvw \\
&= \tilde{d}_{011}(1-u)vw + \tilde{d}_{101}u(1-v)w + \tilde{d}_{110}uv(1-w) + \tilde{d}_{111}uvw \\
&= d_{011}(1-u)vw + d_{101}u(1-v)w + d_{110}uv(1-w) + d_{111}uvw \\
&\leq \frac{d_{011} + d_{101} + d_{110}}{4} + \frac{d_{111}}{27},
\end{aligned} \tag{8}$$

because for $0 \leq u,v,w, \quad u+v+w \leq 1$, $(1-u)vw$ and uvw assume maximal values of $\frac{1}{4}$ (at $u = 0, \ v = w = \frac{1}{2}$) and $\frac{1}{27}$ (at $u = v = w = \frac{1}{3}$), respectively.

■

The valid parametric domain of the hyperplane, $\mathcal{P}$, in four space is prescribed by the tetrahedron through the four non coplanar points in three space. Lemma 2 considered the tetrahedron of $\mathcal{D}(u,v,w), \quad 0 \leq u,v,w, \quad u+v+w \leq 1$, with its parametric domain defined over the four points of P_{000}, P_{001}, P_{010}, P_{100}. By subdividing the voxel into five or six such tetrahedrons as is done in [12] (for disambiguation reasons), one can establish upper bounds for the entire trivariate parametric domain which is the entire voxel. If the established bound is insufficiently loose, the voxel must be subdivided, in order to create the proper polygonal approximation, computed by iso-surfacing the resulting tetrahedron.

4 Examples

We compare the iso-surface extraction to an implementation of traditional marching cubes' algorithm. We have employed one dataset called *3dhead* from the models offered by the University of North Carolina. In our first test, the model was filtered down into a volume of 64x64x27 voxels. The model is iso-surfaced at a level of 250, extracting the shape of the skin. Figure 1 shows the computed iso-surfaces of *3dhead* at a level of 250, but with different hierarchical tolerances. Table 1 provides the times that were necessary to extract these iso-surfaces. Table 1 also presents the number of cells visited in the resulting hierarchy. Of special interest is the reduction in iso-surface extraction times. Approximation tolerances smaller than 200 were found to create visually acceptable iso-surfaces. Time-wise, the iso-surface extracted from a hierarchy of tolerance of 200 was computed in about four seconds compared to the eleven that were necessary using marching cubes. All times were measured on an HP9000, 735 machine.

In [16, 18], min/max bounds on the cells in their Octree hierarchy are exploited to minimize traversal during the iso-surface extraction. This simple optimization can clearly be incorporated into our higher order hierarchy scheme for the hierarchy traversal during the iso-surface extraction. These computation costs are also shown as the rightmost column of Table 1. The Octree in [16] is not fully expanded and the notion of branch-on-need is introduced to create partial Octrees, possibly with less than eight sons. Herein, the hierarchy is a binary tree for which each subdivision can occur at either u or v or w according to the largest domain of the three. The selection to employ a binary tree allows us to maintain image data of arbitrary dimension, in an optimal fashion.

The prescribed accuracy guarantees that the constructed polygonal iso-surfaces will not deviate from the real iso-surfaces by more than the defined tolerances. By constructing several hierarchies, at different resolutions, one is able to honor an iso-surface extraction request from the user in several steps. A rough approximation can be presented to the user in a very short time, only to be refined using a higher tolerance hierarchy should the user desire.

One can exploit the trivariate representation to approximate normals for the derived vertices, for the purpose of Gouraud or Phong shading. By raising the degree of the trivariate, the B-spline trivariate representation low pass filters the data set. Therefore, the gradient of the trivariate scalar field can be used to approximate the normals while the selection of higher degrees can produce smoother normal fields.

Figure 2 shows the result of using a tricubic function over the input image data to produce the normal field. Figure 3 shows a specific hierarchical tolerance level of 100 with normal fields that are computed as gradients of different trivariate orders.

The construction of hierarchical data structure representation of a medical image introduces black holes or cracks in the formed polygonal approximation of the iso-surface. Because the subdivision is adaptive, it can very well may be the case that one sub-volume will be subdivided further than a neighboring sub-volume. When the isosurface is extracted a gap, commonly called black hole, may form. A solution that exploits a data structure that maintains the topological adjacency information must be employed to resolve this problem, in a similar way to solving the black holes problem in the adaptive polygonization of bivariate surfaces. A simple solution is to move the interior vertex of the refined sub-volume's isosurface approximation to be on the line of the coarse sub-volume's isosurface approximation.

Figure 1: Eleven levels of iso-surfacing *3dhead* at different tolerance resolutions (flat shading). The top left shows the result of the traditional Marching Cube algorithm. Next to it to the right is the iso-surface extracted with tolerance value of 6 up to the bottom right image which is computed with tolerance 500. See also Table 1.

See Figure 4

5 Conclusions

Recently, posterior algorithms were suggested to decimate large data sets of polygons [13, 15]. Herein, we have proposed an a-priori approach to create a high order trivariate hierarchy that significantly reduces the overhead of processing uniformly sampled images. It offers the obvious advantage that prohibits the need to process the huge intermediate data set for every computed iso-surface.

We have presented an a-priori construction of hierarchical data structure that enables one to iso-surface the volumetric image at a sub-linear processing time at any iso-surface level and with an established upper bound on the accuracy. One can preprocess several such hierarchies, at different tolerances, creating multiresolution data structures for the extraction of arbitrary iso-surfaces at different resolutions, and different response times.

Table 1: Eleven levels of iso-surfacing *3dhead* at different tolerance resolution (iso-surface level is 250), filtered down to 64x64x27 voxels. See also Figure 1. Time was measured on an HP 735.

Tolerance	# of Polygons	# of Cells Visited	Secs. for hierarchy	Secs. for iso-surface	Secs. for iso-surface with min/max
Marching Cubes	24304	110592	-	11	-
500	8209	8136	30	2	1
400	9974	10687	37	2	2
300	12082	14896	46	3	2
200	14361	21845	60	4	3
100	17494	30681	79	6	4
75	18784	33410	84	7	4
50	19992	36453	100	7	6
25	21129	41029	107	8	6
15	21506	45756	119	9	6
10	21628	52275	134	11	6
6	21704	74629	188	16	8

The bound that is established by Lemma 2 is tight if exactly one of the four coefficients of d_{011}, d_{101}, d_{110}, and d_{111} is non zero. One could consider the establishment of a tighter bound that is input dependent. The usefulness of this tighter bound must be examined for the specific type of data that is processed.

In the work presented herein, the implementation employed the trivariate package of the IRIT [6] solid modeling system. This package is implemented using floating point calculus which undoubtly slows down the computation. Moreover, only uniform B-spline basis functions are exploited in this work and these basis function can be easily preevaluated into tables, further reducing the computation overhead. The trivariate package of IRIT supports the general, non uniform, B-spline basis functions with no special treatment of and/or optimization for the uniform case.

The employed trivariate package also imposed large memory overhead of over %300. One can represent a trilinear using eight scalar values, float in our implementation. However, the trivariate package of IRIT supports general trivariate function and allows one to prescribe orders, length, and knot vectors in all three axes as well as supports non scalar control points. An hierarchical structure where each node is subdivided into n sub-nodes each of approximately size $\frac{1}{n}$ (plus some overhead along the common boundary) will remain approximately the same size as the original data. Because the trilinear approximation is usually smaller than the trivariate it approximates, we expect that a careful implementation developed specifically for this application, will gain in memory and reduce the size that is necessary for the representation. The amount of saving will depend on the complexity of the original trivariate data.

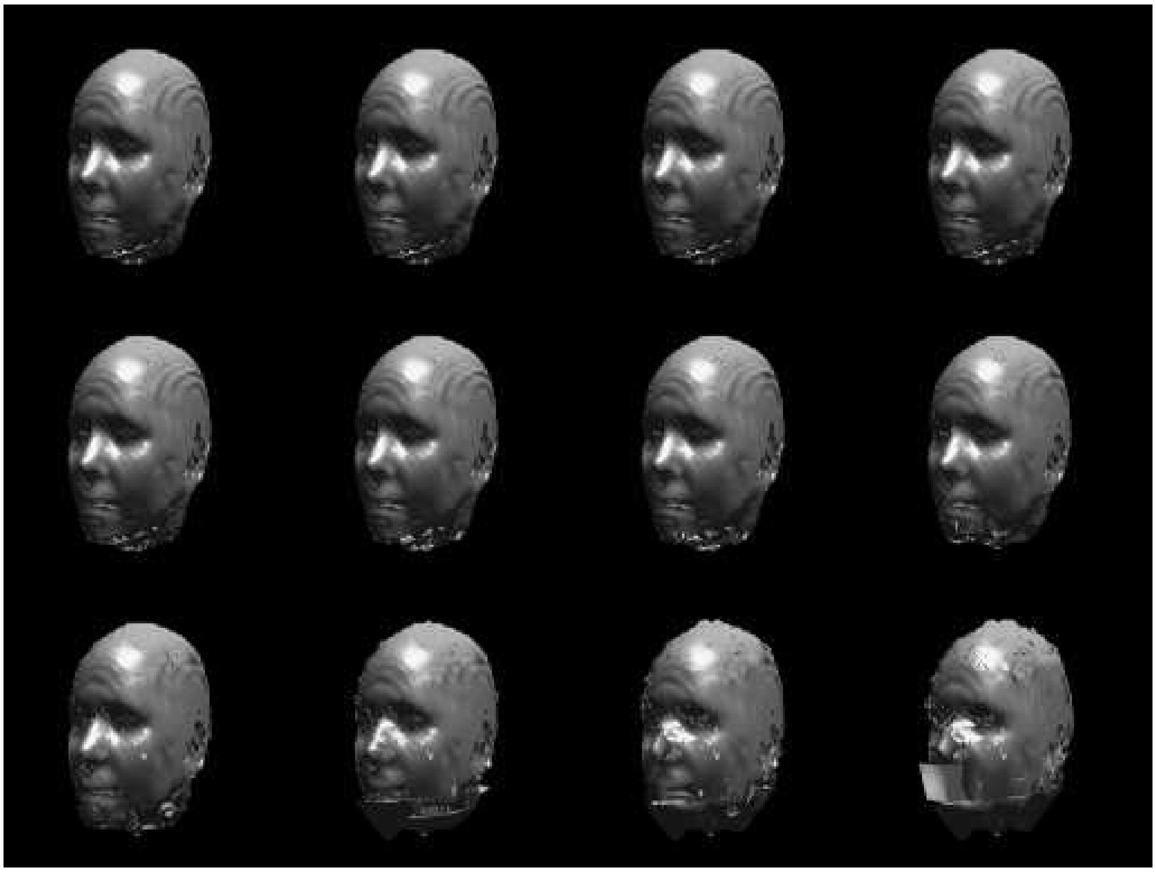

Figure 2: Twelve levels of iso-surfacing *3dhead* at different tolerance resolutions (tolerance 6 at top left to tolerance 600 at the bottom right, same steps as in Table 1) with a gradient of a tricubic used to derive the normals. See also Figure 1.

It is certain that the use of higher order trivariate functions enables one to more faithfully represent the original geometry and as such are more suitable for the task for medical imaging. It is yet to be explored how difficult it is to optimally utilize these higher order functions for the required medical tasks.

6 References

[1] W. Boehm. Inserting New Knots into B-spline Curves. Computer Aided Design, vol. 12, No. 4, pp. 199-201, July 1980.

[2] P. Cignoni, L. D. Floriani, C. Montani, E. Puppo, and R. Scopigno. Multiresolution Modeling and Visualization of Volume Data Based on Simplicial Complexes. Proceedings of the 1994 Symposium on Volume Visualization, pp 19-26, Washingon D.C., October 1994.

[3] E. Cohen, T. Lyche, and R. Riesenfeld. Discrete B-splines and Subdivision Techniques in Computer-Aided Geometric Design and Computer Graphics. Computer Graphics and Image Processing, 14, 87-111 (1980).

Figure 3: The *3dhead* rendered at tolerance level 100 with flat shading (top left), and normals computed from the gradient of a trilinear, a triquadratic, and a tricubic.

[4] G. Elber. Proper Piecewise Linear Approximation of Freeform Surfaces. CIS Report #9413, Computer Science Department, Technion, October 1994.

[5] L. H. Figueiredo, J. M. Gomes, D. Terzopoulos, and L. Velho. Physically-Based Methods for Polygonization of Implicit Surfaces. Graphics Interface 92, pp 250-257, Vancouver, Canada, May 1992.

[6] IRIT solid modeller. IRIT 5.0 User's Manual.

[7] D. Laur and P. Hanrahan. Hierarchical Splatting: A progressive Refinement Algorithm for Volume Rendering. Computer Graphics (SIGGRAPH '91 Proceedings), Vol. 25, No. 4, pp 285-288, July 1991.

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

[9] M. Levoy. Efficient Ray Tracing of Volume Data. ACM Transaction on Graphics, Vol. 9, No. 3, pp 245-261, July 1990.

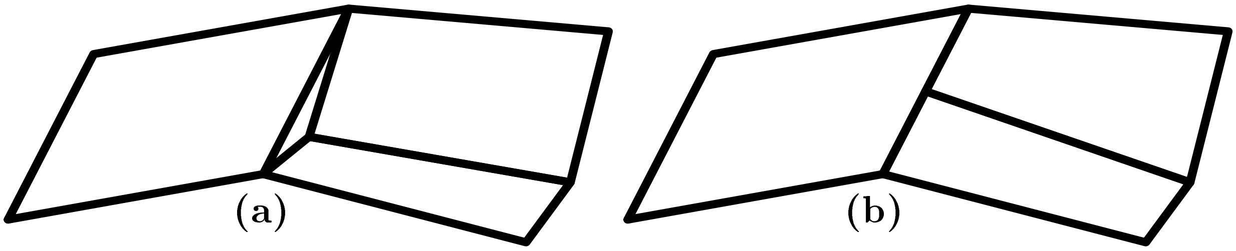

Figure 4: The black holes, or cracks, in (a) are filled by moving the interior vertex of the refined side to be on the line of the coarse side (b).

[10] O. Monga, and S. Benayoun. Using Differential Geometry in R^4 to Extract Typical Features in 3D Images. IEEE Conference on Vision and Pattern Recognition, pp 684-685, June 1993.

[11] B. K. Natarajan. On generating topologically consistent isosurfaces from uniform samples. The Visual Computer, Vol 11, No. 1, pp 52-62, 1995.

[12] P. Ning, and J. Bloomenthal. An Evaluation of Implicit Surface Tilers. IEEE Computer Graphics and Applications, Vol 13, No. 6, pp 33-41, November 1993.

[13] W. J. Schroeder, J. A. Zarge and W. E. Lorensen. Decimation of Triangle Meshes. Computer Graphics (SIGGRAPH '92 Proceedings), Vol. 26, No. 2, pp 65-70, July 1992.

[14] S. T. Tuohy, J. W. Yoon, and N. M. patrikalakis. Reliable Interrogation of 3D Non-linear geophysical Databases. Computer Graphics Developments in Virtual Worlds, R. A. Earnshaw and J. A. Vince (Eds.), Computer Graphics International 1995 (CGI 95), Leeds, England, pp 327-341, June 1995.

[15] G. Turk. Re-tiling Polygonal Surfaces. Computer Graphics (SIGGRAPH '92 Proceedings), Vol. 26, No. 2, pp 55-64, July 1992.

[16] J. Wilhelms, and A. V. Gelder. Octrees for Faster Isosurface Generation. ACM Transaction on Graphics, Vol. 11, No. 3, pp 201-226, July 1992.

[17] J. Wilhelms and A. V. Gelder. Multi-Dimensional Trees for Controlled Volume Rendering and Compression. Proceedings of the 1994 Symposium on Volume Visualization, pp 27-34, Washingon D.C., October 1994.

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