

# Dual Marching Tetrahedra: Contouring in the Tetrahedral Environment

Gregory M. Nielson

Arizona State University

**Abstract.** We discuss the dual marching tetrahedra (DMT) method. The DMT can be viewed as a generalization of the classical cuberille method of Chen et al. to a tetrahedral. The cuberille method produces a rendering of quadrilaterals comprising a surface that separates voxels deemed to be contained in an object of interest from those voxels not in the object. A cuberille is a region of 3D space partitioned into cubes. A tetrahedral is a region of 3D space decomposed into tetrahedra. The DMT method generalizes the cuberille method from cubes to tetrahedra and corrects a fundamental problem of the original cuberille method where separating surfaces are not necessarily manifolds. For binary segmented data, we propose a method for computing the location of vertices this is based upon the use of a minimal discrete norm curvature criterion. For applications where dependent function values are given at grid points, two alternative methods for computing vertex positions are discussed and compared. Examples are drawn from a variety of applications, including the Yes/No/Don't Know data sets resulting from inconclusive segmentation processes and Well-Log data sets.

## 1 Introduction

The original cuberille method [3] is primarily concerned with the rendering of a surface that separates the voxels that are part of an object from the voxels that are not part of the object. Due to the wide spread need for such techniques, there continues to be a fair amount of published literature on this topic. See [2], [4], [6], [9], [13] and [19] for example. Here we discuss the tetrahedral version of this method or, in other words, the dual marching tetrahedra (DMT) method. In addition, here we are not only interested in the rendering, but we are also interested in methods that will produce the geometry consisting of a polygon mesh representation of the separating surface. This geometry can not only be used for rendering, but it also allows for the efficient application of surface parameterizations, curvature texture applications and many other geometry processing tools. The next four paragraphs constitute an annotated outline of the present paper.

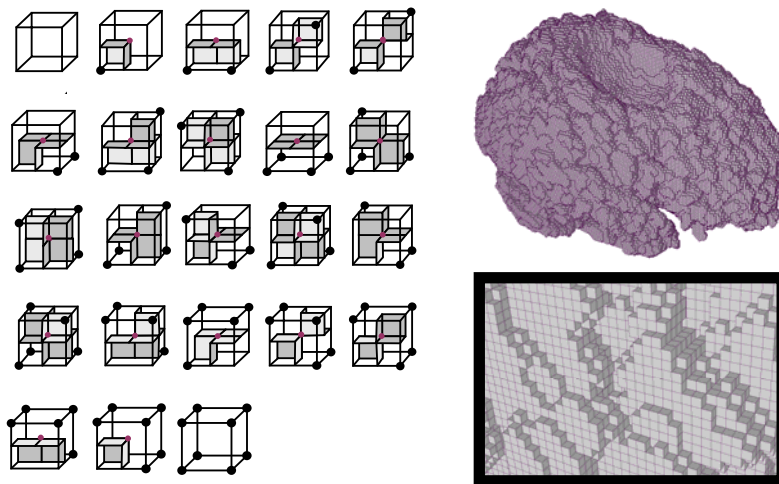
1. As motivation and background, we first discuss the polygon mesh surface which would result from the application of the cuberille method. Among other properties, it is noted that this surface is not always guaranteed to be a manifold. This may not be a problem when only a rendering of the surface is required, but geometric processing of the polygon mesh surfaces such as parameterization, volume inside/outside determination or curvature computations can not be applied to nonmanifold surfaces.

2. Next, we discuss the dual marching tetrahedral method, DMT, which produces a polygon mesh surface with a vertex lying in each tetrahedron that has both marked and unmarked grid points. In Section 2.1, we describe a method for computing the actual vertex positions which is based upon discrete norm curvature. In Section 3, we show examples where the new tetrahedral method is applied to the general class of segmented data sets called “**Yes/No/Don’t\_Know**”. We also illustrate the application of the tetrahedral method to the general class of data often called “**Well-Log**” data.
3. Using methods of decomposing cubes into tetrahedra without using additional grid points, we also show in Section 3 how to apply the DMT method to the more conventional rectilinear lattice data or cuberille data.
4. Next, we note that it is very simple to extend the application of the method from binary classified grid points to the case where there are dependent function values given at every grid point. The generalization to this type of data is accomplished by using thresholding to create a binary classification and so the basic method immediately applies. In addition to the minimum discrete norm curvature method for computing vertex positions, in Section 4. we describe two methods based upon dependent function values. We note the connection to the marching tetrahedra (MT) method through its dual graph and compare results.

## 2 Background and Motivation Based Upon the Cuberille Method

As we previously mentioned, much of the motivation for the DMT method is based upon the original Cuberille Method which is described in [3]. This approach to rendering surface bounded objects from computed tomography (CT) data is classical due to the simplicity of the method. A cuberille is a dissection of space into cubes called voxels. The total collection of all of these voxels is segmented into two groups, namely those voxels that belong to an object of interest and those voxels that do not belong to this object. A rendering of the surface bounding the object is accomplished by rendering all of the directed faces which separate the voxels of the object and from those of the background. The process of determining these distinguished voxels that belong to the object is often referred to as segmentation. See [11], [13], [21] and [23]. While the objective of the original cuberille method was mainly the rendered image of the separating surface, here we are also interested in obtaining a representation of the separating surface as a polygon mesh surface. This geometry not only can be used for rendering of the object, but it can also facilitate many subsequent geometry processing operations involving the object of interest; such as, parameterizing or computing the volume or the area of the separating surface. The collection of centroids of the voxels forms a three dimensional rectilinear lattice we denote by  $P_{ijk}$ ,  $i = 1, \dots, N_x$ ,  $j = 1, \dots, N_y$ ,  $k = 1, \dots, N_z$ . The cubes with corners from this lattice are denoted by  $C_{ijk}$ . The centers of the cubes of the lattice points are the “corners” of the voxels. The separating surface  $S$ , is a polygon surface comprised of quadrilaterals with vertices  $V_a$ ,  $a = 1, \dots, N$  taken as the centroids of the cubes  $C_{ijk}$  where at least one of the voxels intersecting  $C_{ijk}$  is in the object and one is not in the object.

The topology (edge connectivity) of the vertices is determined by the various configurations shown in Figure 1. At the onset, there are a total of  $256 = 2^8$  possible cases to consider, but with the use of rotations, the number of cases is reduced to 23 equivalence classes with representatives of each class shown in Figure 1. See [17]. As can be easily seen from this table, nonmanifold surfaces are produced by configurations C3, C6, C7, C10, C12, C13, C15, C16 and C19 (where C0 is upper left and C22 is lower right). While these configurations are not overly abundant in typical applications, they do occur, on the average, approximately 0.07%, 0.08%, 0.015%, 0.025%, 0.05%, 0.00005%, 0.015%, 0.08%, 0.07% respectively.

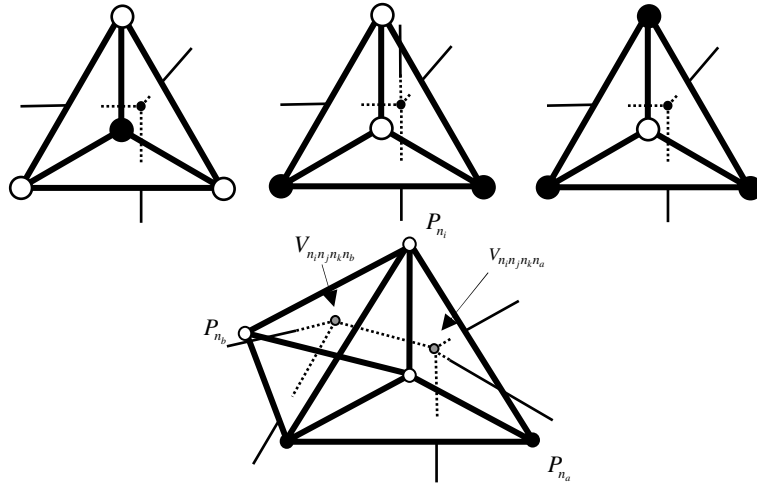


**Fig. 1.** The figure on the left shows the connectivity of the quad patches of the cuberille method for the various configurations. Each configuration is a representer of an equivalence class determined by rigid rotations of the 256 possible cases. See [17] for a table indicating the rotation group element, the representer and the cases. An example of the cuberille method is shown in the top, right image where a tumor has been segmented from the brain. The data is the Harvard brain tumor data (see [11], [23]) available from Surgical Planning Laboratory at Brigham and Women’s Hospital, [www.spl.harvard.edu](http://www.spl.harvard.edu) and consists of an array of size  $124 \times 256 \times 256$ . The bottom image is a zoom-in of the top image so that the voxel features of the image are clearly visible.

**2.1 The Dual Marching Tetrahedra Method**

Let  $P_i = (x_i, y_i, z_i), i = 1, \dots, N$  denote the grid points which are segmented as marked or unmarked. We assume these points are not collectively coplanar. We assume that the grid points have been arranged into a collection of tetrahedra to form a tetrahedralon. A tetrahedralon consists of a list of 4-tuples which we denote by  $I_t$ . Each 4-tuple,  $ijkl \in I_t$  denotes a single tetrahedron with the four vertices  $P_i, P_j, P_k, P_l$  which is denoted as  $T_{ijkl}$ . A valid tetrahedralon requires: i) No tetrahedron  $T_{ijkl}, ijkl \in I_t$  is degenerate, i. e. the points  $P_i, P_j, P_k, P_l$  are not coplanar, ii) The interiors of any two

tetrahedra do not intersect and iii) The boundary of two tetrahedra can intersect only at a common triangular face. See [16] for a survey of methods for computing optimal (Delaunay) tetrahedrals of the convex hull of scattered 3D point sets.



**Fig. 2.** The three active cases of the DMT method. From left to right: one, two and three points classified as being contained in the object of interest. The lower image illustrates the notation of the tetrahedral method.

A tetrahedron is said to be active if among its 4 grid points there are both marked grid points and unmarked grid points. There are three distinct configurations for these active tetrahedra as shown in Figure 2. Similarly, a triangular face of an active tetrahedron is active provided it contains both marked and unmarked grid points. Interior to each active tetrahedron,  $T_{n_i n_j n_k n_l}$ , there is a vertex  $V_{n_i n_j n_k n_l}$ . For each interior active triangular face, there is an edge of  $S$  joining the two vertices of the two tetrahedra sharing this triangular face. If  $F_{n_i n_j n_k}$  denotes the interior active triangular face and  $T_{n_i n_j n_k n_a}$  and  $T_{n_i n_j n_k n_b}$  denote the two active tetrahedra sharing this face then an edge joins  $V_{n_i n_j n_k n_a}$  and  $V_{n_i n_j n_k n_b}$ .

In addition to the surface nets approach [9], we propose a scheme for computing the positions of the vertices that is based upon minimizing discrete norm curvature estimates at each vertex which we now describe. For the tetrahedron defined by the points  $P_i, P_j, P_k, P_l$  we consider the tetrahedra lattice points  $V_{a,b,c,d} = (aP_i + bP_j + cP_k + dP_l)/N$  where the integers  $a, b, c, d$  satisfy the two conditions  $0 < a, b, c, d < N$  and  $a + b + c + d = N$ . For each point  $V_{a,b,c,d}$  in  $T_{i,j,k,l}$  we use the discrete curvature methods of Dyn et al [8] and Kim et al [12] to compute an estimate of the norm curvature  $k_1^2(S) + k_2^2(S) = 4[M(S)]^2 - 2K(S)$  where

$k_1(S)$  and  $k_2(S)$  are the principal curvatures,  $M(S)$  is the mean curvature and  $K(S)$  is Gaussian curvature. These estimates are based upon a triangulation of  $V_{a,b,c,d}$  and the vertices of its 1-ring that maintain the edges of separating surface  $S$  and do not introduce any additional edges containing  $V_{a,b,c,d}$ . The estimates are computed as  $K(S) = 3(2\pi - \sum \alpha_i) / A$ ,  $M(S) = .75 \sum \|e_i\| \beta_i / A$ , where  $e_i$  is an edge joining  $V_{a,b,c,d}$  and a vertex of its 1-ring,  $\beta_i$  is its dihedral angle,  $\alpha_i$  is a subtended angle and  $A$  is the sum of the areas of all the adjacent triangles. We take as our first approximation  $V_{i,j,k,l}^{(1)}$  the point  $V_{a,b,c,d}$  associated with the smallest estimate of norm curvature. These values are computed for all tetrahedra containing vertices of the separating surface  $S$ . We do another pass over all of the tetrahedra containing vertices of  $S$  leading to the approximations  $V_{i,j,k,l}^{(2)}$ . This is continued until the user specified criteria for convergence is satisfied. In practice usually 7 or 8 digits of accuracy are obtained in less than 6 iterations (a complete loop through all active tetrahedra). A resolution of  $N = 5, \dots, 9$  is a typical choice.

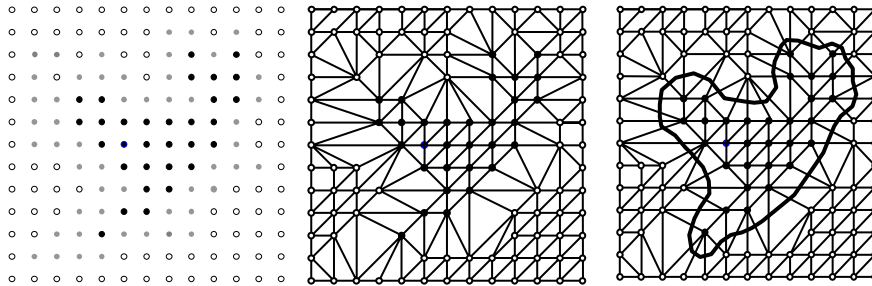
### 3 Application to the General Types of Data Sets: Yes/No/Don't\_Know, Well-Log and Cuberille

In this section we describe three fairly widely observed types of data sets for which the DMT method applies. These are the so-called Yes/No/Don't\_Know, Well-Log and cuberille data sets. The first type is from the application of segmentation results applied to conventional rectilinear data and the second and third types result from some very common types of methods for taking samples.

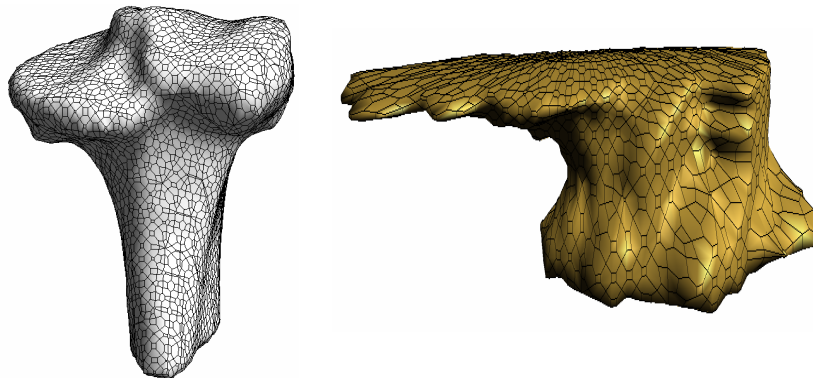
#### 3.1 Yes/No/Don't\_Know Data Sets

MRI and CAT scan data sets can be viewed as samples of a function defined over a rectilinear lattice. Often segmentation algorithms are invoked to determine which of the lattice points belong to a certain specified type and which do not. For example, we may wish to determine which lattice points of a MRI scan are brain matter and which are not. See [11] and [23] for more discussion on this. Quite often, segmentation algorithms require the user to specify the values for certain parameters. Even if domain experts are used to "tune" these algorithms, the results can be, in some applications, "inconclusive". This means that, for some lattice grid points, the algorithm reports that these lattice points definitely belong to the object, for others it reports that these lattice points definitely do not belong to the object. But for some points the algorithm cannot definitely report one way or the other and so these lattice points could or could not be part of the object of interest. The technique, we present, for handling this type of data is illustrated in Figure 3. The left image illustrates with the black circles the lattice points known to be part of the object and the white circles indicate lattice points which are definitely not part of the object. The gray circles indicate the "Don't\_Know" lattice points where the segmentation algorithm is inconclusive. The

gray points are simply removed and the convex hull of the remaining white and black lattice points are collectively triangulated or, in the 3D case, tetrahedralized into a tetrahedral. For a survey on methods of tetrahedral zing general scattered points see [16]. Once we have a tetrahedral with segmented grid points, the methods of Section 2 immediately apply.



**Fig. 3.** In the left image the white circles indicate lattice (grid) points that are definitely not contained in an object of interest, the black points are ones that are definitely belonging to the object and the grey points are undetermined. In the middle image, the grey points have been removed and the convex hull of the remaining white and black points are triangulated. In the right image a separating polygon is obtained by the 2-D version of the DMT method.



**Fig. 4.** The results of the DMT method applied to a **yes/no/don't know** data set resulting from a segmentation algorithm applied to rectilinear data obtained from the scan of a Chimp tibia bone (only the top portion is used). The right image illustrated the application of the DMT to a Well-Log data set yielding a bounded region where copper exists at beyond trace levels.

In Figure 4 we show the results of applying this method to segmented data representing a portion of a bone. Lattice points in the voxel grid have been classified as definitely bone, definitely not bone and inconclusive. The lattice points that could not conclusively be determined as bone or not are removed and the remaining 3D lattice points are used as the bases of a tetrahedral. The tetrahedral method is applied where the actual vertex positions are determined by the minimal discrete curvature method described in Section 2.

### 3.2 Well-Log Type Data

In the geophysical sciences, it is quite common to collect measurements at various depths (or heights) at several locations. The locations  $(x_i, y_i)$  are often positions on the earth and the depth (or height) may possibly vary from location to location. Also the number of measurements per location may vary. This type of data can be represented as  $(x_i, y_i, z_{ik}, M_{ik}), i = 1, \dots, N; k = 1, \dots, n_i$  where  $n_i$  is the number of measurements taken at location  $(x_i, y_i)$  and  $M_{ik} = 0$  or  $1$ . The example in the right image of Figure 4 utilizes data courtesy of D. Kinsel (See [15]) and represents from 9 to 43 samples taken from 13 well locations. The samples are tested for copper concentrations above trace levels and marked 0 or 1 accordingly. The region where copper can be found is bounded by the isosurface.

### 3.3 Cuberille Data Sets

We illustrate the application of the tetrahedral method to cuberille data with the Harvard brain data set mentioned above in the example of Figure 1. In Figure 6 we show the contour surface of the tumour which has been segmented from the brain data using both the alternating 5-split and the CFK split of a cube into tetrahedral.

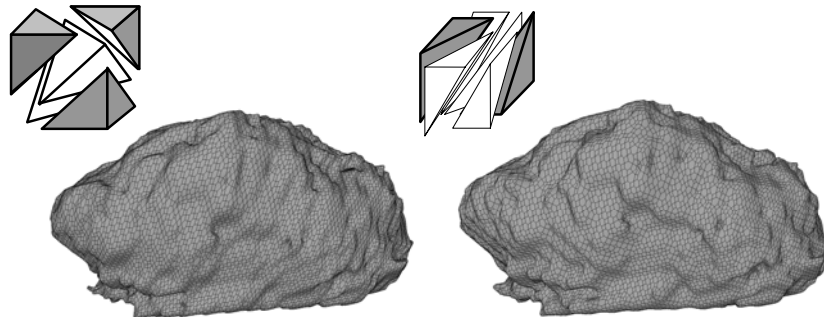


Fig. 5. Two different decomposition of a cube into tetrahedra and the resulting DMT surfaces

## 4 Extension to Function Valued Data Sets

The previous sections have dealt with the case where the grid points,  $P_i, i = 1, \dots, N$  of the tetrahedral grid have the binary property of being marked or unmarked. Many applications result in tetrahedral grids where there is a dependent function  $F_i$  value associated with each vertex  $V_i$ . That is, the grid is viewed as a decomposition of the domain of a trivariate function and the dependent values  $F_i$  are samples of an underlying trivariate function defined over the tetrahedral grid domain volume. Given a threshold value,  $F$ , we can classify whether or not the grid points  $P_i$  are marked or unmarked based upon whether or not  $F_i \geq F$  thus allowing the level sets of the

sampled function to be viewed by the tetrahedral method. Here we have a richer context where the dependent sample values can be used to determine the location of the vertices of the approximating separating surface. A large variety of choices are possible. We describe two possibilities here. The first is based upon the idea that if there is a sign change on an edge then the plane passing through these points is also close to sign changes in the interior of the tetrahedra. This plane will be determined by three or four points. The vertex point is selected as a point on this plane that intersects the line joining the centroid of a face and the opposing grid vertex or the line joining the midpoints of two opposing edges. We call this method: **Intersection between Centroids, IbC**, and it is further illustrated in Figure 6.

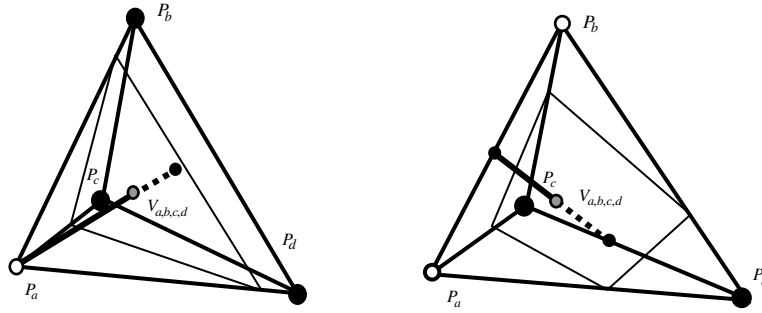


Fig. 6. The computation of the vertex  $V_{a,b,c,d}$  in Case 1 and 2 of the **Intersection between Centroids, IbC** method

The second method is based upon the idea that surface intersection points can be computed by linear interpolation on all edges where function values at the end points encompass the threshold value. The vertex point,  $V_{a,b,c,d}$ , is taken to be the centroid of these points. We refer to this method as the **Centroid of Intersections, CoI**.

Detailed computational formulas for both of these two methods are given below, where we assume that the points have been labelled so that the following holds:  $F_a \leq F_b \leq F_c \leq F_d$ . The values  $V_{ij}$  represent the intersection of linear interpolation along an edge  $e_{ij}$ .

**Method: Intersection between Centroids, IbC**

$$\begin{aligned}
 F_a \leq F < F_b < F_c < F_d & \quad V_{abcd} = \frac{(F_b - F_a)V_{ab} + (F_c - F_a)V_{ac} + (F_d - F_a)V_{cd}}{(F_b + F_c + F_d - 3F_a)} \\
 F_a \leq F_b < F < F_c < F_d & \quad V_{abcd} = \frac{(F_d - F_a)V_{ad} + (F_d - F_b)V_{bd} + (F_c - F_a)V_{ac} + (F_c - F_b)V_{bc}}{2(F_c + F_d - F_a - F_b)} \\
 F_a \leq F_b \leq F_c < F < F_d & \quad V_{abcd} = \frac{(F_d - F_a)V_{ad} + (F_d - F_b)V_{bd} + (F_d - F_c)V_{cd}}{(3F_d - F_a - F_b - F_c)}
 \end{aligned} \tag{1}$$

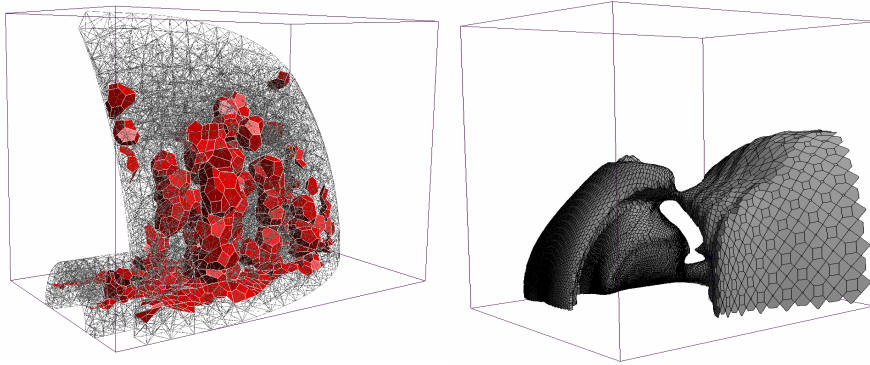
**Method: Centroid of Intersections, CoI**

$$\begin{aligned}
 F_a \leq F < F_b < F_c < F_d & \quad V_{abcd} = \left( \frac{1}{3} \left( \frac{F_b - F}{F_b - F_a} \right) + \frac{1}{3} \left( \frac{F_c - F}{F_c - F_a} \right) + \frac{1}{3} \left( \frac{F_d - F}{F_d - F_a} \right) \right) P_a \\
 & \quad + \frac{1}{3} \left( \left( \frac{F - F_a}{F_b - F_a} \right) P_b + \left( \frac{F - F_a}{F_c - F_a} \right) P_c + \left( \frac{F - F_a}{F_d - F_a} \right) P_d \right)
 \end{aligned}$$



$$\begin{aligned}
 F_a \leq F_b < F < F_c < F_d & \quad V_{abcd} = \frac{1}{4} \left( \left( \frac{F_c - F}{F_c - F_a} \right) + \left( \frac{F_d - F}{F_d - F_a} \right) \right) P_a + \frac{1}{4} \left( \left( \frac{F_d - F}{F_d - F_b} \right) + \left( \frac{F_c - F}{F_c - F_b} \right) \right) P_b \\
 & \quad + \frac{1}{4} \left( \left( \frac{F - F_b}{F_c - F_b} \right) + \left( \frac{F - F_d}{F_c - F_a} \right) \right) P_c + \frac{1}{4} \left( \left( \frac{F - F_a}{F_d - F_a} \right) + \left( \frac{F - F_b}{F_d - F_b} \right) \right) P_d \\
 F_a \leq F_b \leq F_c < F < F_d & \quad V_{abcd} = \left( \frac{1}{3} \left( \frac{F - F_a}{F_d - F_a} \right) + \frac{1}{3} \left( \frac{F - F_b}{F_d - F_b} \right) + \frac{1}{3} \left( \frac{F - F_c}{F_d - F_c} \right) \right) P_d \\
 & \quad + \frac{1}{3} \left( \left( \frac{F_d - F}{F_d - F_a} \right) P_a + \left( \frac{F_d - F}{F_d - F_b} \right) P_b + \left( \frac{F_d - F}{F_d - F_c} \right) P_c \right)
 \end{aligned} \tag{2}$$

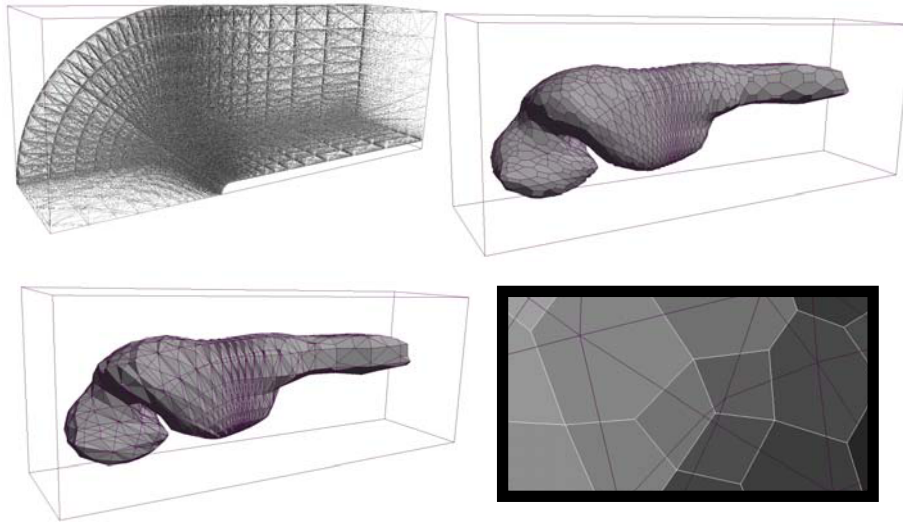
Typical results of the application of the tetrahedral method with dependent functional values are shown in Figure 7. The left example is from a FEM (finite element method) combustion simulation. The tetrahedral has 47025 grid points and 215040 tetrahedra. Here we have used the **Centroid of Intersection, CoI**, method of determining the isosurface vertices.



**Fig. 7.** On the left is shown the results of a Finite Element Method (FEM) combustion simulation. The isosurface is displayed along with the edges of the tetrahedral. The **CoI** method of computing isosurface vertices is used. On the right, the Delta Wing data is displayed at isolevel 0.22977 which yields a surface with some interesting topology. The tetrahedral has 211,680 grid points and 1,005,657 tetrahedra. The **IbC** method of computing vertex locations is used.

An example which utilizes the “Delta Wing at 40° Attack” data set available from NASA Ames is shown in the right image of Figure 7. This data in its original form consists of a curvilinear grid of size 56x54x70. Here, each of the hexahedral cells has been decomposed into 5 tetrahedra using the decomposition of Figure 5.

In Figure 9, we show an example which uses the well known “Blunt Fin” data set. Here, the **IbC** method of computing vertex positions is used. In addition, we include the isosurface produced by the Marching Tetrahedra (MT) method. See [1], [10], [14] and [15]. We use the variation that produces a quadrilateral (rather than two triangles) in the case where function values at two grid vertices are above the threshold and two are below. With this version, the surfaces produced by the MT method and the present tetrahedral method are formal mathematical dual polygon mesh surfaces of each other. Each vertex of one, uniquely associates with a face of the other. This is illustrated in the blow-up shown at the bottom, right of Figure 8 where the (white) edges of the tetrahedral method are displayed along with the (black) edges of the surface produced by the MT method.



**Fig. 8.** The grid of the “Blunt Fin” data set is shown in upper left. The upper right is the DMT using the IbC method of vertex selection and the lower left is the MT. The lower right illustrates the duality of the MT and the DMT.

## 5 Remarks

1. Since a tetrahedron can be formed from an arbitrary scattered point cloud (by computing the Delaunay tetrahedralization of the convex hull, see [16]), the methods described here have very widespread application; including datasets with binary and/or function dependent grid values. The method is very easily implemented particularly in the case of function dependent grid values where the formulas of the Intersection between Centroids (**IbC**) and the Centroid of Intersections (**CoI**) are available (Equations (1) and (2)).

2. While both the **IbC** and **CoI** methods of computing vertex positions are both simple and effective (and these are the main reasons we chose to report only on these methods here), there are many possibilities we have yet to fully test. One particularly intriguing possibility which we hope to report upon in the near future is to combine the marching diamond method of [1] with our **CoI** method.

3. Even for rectilinear grid data applications, the implementation of the DMT method presented here is considerably easier and less tedious than the standard marching cubes (MC) (see [17]) or dual marching cubes (DMC) (see [18]) due to the differences in complexity of the algorithm. There are 256 cases leading to 23 distinct configurations for both the MC and the DMC while the DMT method has only 16 cases and 3 distinct configurations.

4. While the present method is designed for tetrahedral application, it applies immediately to rectilinear data also by means of the decompositions of Figure 5. It has been noted in the past, that when the marching tetrahedra (MT) method is applied to rectilinear data, empirical evidence suggests that typically the number of vertices

and/or triangles only increases by a factor of approximately 2.5. This suggests that the main deterrent to the general adoption of the MT for rectilinear data is not the increase in the complexity of the isosurface, but rather the over all poor triangle quality of the resulting mesh surfaces. The present method eliminates this deterrent as evidenced by the examples presented here. We plan to report on an analysis and empirical study on this issue in the near future as these results become available.

## Acknowledgements

We wish to acknowledge the support of the US Army Research Office under contract W911NF-05-1-0301 and the US National Science Foundation. We wish to thank Narayanan Chatapuram Krishnan, Vineeth Nallure Balasubramanian and Ryan Holmes for their help and contributions to this project.

## References

1. Anderson, J.C., Bennett, J., Joy, K.: Marching Diamonds for Unstructured Meshes. In: Proceedings of IEEE Visualization 2005, pp. 423–429 (2005)
2. Bonneau, G.-P., Hahmann, S.: Polyhedral Modeling. In: Proceedings of Visualization 2000, pp. 381–387 (2000)
3. Chen, L.-S., Herman, G.T., Reynolds, R.A., Udupa, J.K.: Surface shading in the cuberille environment. *Computer Graphics and Applications* 10, 33–43 (1985)
4. Chica, A., Williams, J., Andujar, C., Crosa-Brunet, P., Navazo, I., Rossignac, J., Vinacua, A.: Pressing: Smooth Isosurfaces with flats from binary grids. In: *Computer Graphics Forum 2007*, pp. 1–10 (2007)
5. Co, C.S., Joy, K.: Isosurface generation for large-scale scattered data visualization. In: Proceedings of VMV 2005, pp. 233–240 (2005)
6. Crosa-Brunet, P., Navazo, I.: Solid Representation and Operation using Extended Octrees. *ACM Transactions on Graphics*, 123–456 (1990)
7. Crosa-Brunet, P., Andujar, C., Chica, A., Navazo, I., Rossignac, J.: Optimal Isosurfaces. *Computer Aided Design and Applications* (2004)
8. Dyn, N., Hormann, K., Kim, S.-J., Levin, D.: Optimizing 3D triangulations using discrete curvature analysis. *Mathematical Methods for Curves and Surfaces*, 135–146 (2001)
9. Gibson, S.F.F.: Constrained elastic surface nets: Generating smooth surfaces from binary segmented data. In: Wells, W.M., Colchester, A.C.F., Delp, S.L. (eds.) *MICCAI 1998*. LNCS, vol. 1496, pp. 888–898. Springer, Heidelberg (1998)
10. Gregorski, B.F., Wiley, D.F., Childs, H., Hamann, B., Joy, K.: Adaptive contouring with quadratic tetrahedral. In: *Scientific Visualization: The Visual Extraction of Knowledge from Data*, pp. 3–15 (2006)
11. Kaus, M., Warfiedl, S.K., Nabavi, A., Black, M., Kikinis, R.: Segmentation of MRI of brain tumors. *Radiology* 218, 586–591 (2001)
12. Kim, S.-J., Kim, C.-H., Levin, D.: Surface simplification using a discrete curvature norm. *Computers & Graphics* 26, 657–663 (2002)
13. Lagopoulos, J.: Voxel-based morphometry made simple. *Acta Neoropsychiatrica* 9(3), 213–214
14. Linsen, L., Prautzsch, H.: Fan Clouds – An Alternative to Meshes. In: *Geometry Morphology, and Computational Imaging*, p. 451. Springer, Heidelberg (2003)

15. Nielson, G.M.: Modeling and visualizing volumetric and surface-on-surface data. In: Focus on Scientific Visualization, pp. 219–274. Springer, Heidelberg (1992)
16. Nielson, G.M.: Tools for Triangulations and Tetrahedrizations and Constructing Functions Defined over Them. In: Scientific Visualization: Overviews, Methodologies Techniques, pp. 429–526. IEEE Computer Society Press, Los Alamitos (1997)
17. Nielson, G.M.: On Marching Cubes. *Transactions on Visualization and Computer Graphics* 9(3), 283–297 (2003)
18. Nielson, G.M.: Dual Marching Cubes. In: *Proceedings of Visualization 2004*, pp. 489–496. CS Press (2004)
19. Ning, P., Bloomenthal, P.: An evaluation of implicit surface tilers. *IEEE Computer Graphics and Applications* 13(6), 33–41
20. Sreevalsan-Nair, J., Linsen, L., Hamann, B.: Topologically accurate dual isosurfacing using ray intersection. *Journal of Virtual Reality and Broadcasting* 4(4), 1–12 (2007)
21. Udupa, J., LeBlanc, V., Schmidt, H., Imielinska, C., Saha, P., Grevera, G., Zhuge, Y., Molholt, P., Jin, Y., Currie, L.: A methodology for evaluating image segmentation algorithms. In: *SPIE Conference on Medical Imaging, San Diego*, pp. 266–277 (2002)
22. Vivodtzev, F., Bonneau, G.-P., Linsen, L., Hamann, B.: Hierarchical Isosurface Segmentation based on Discrete Curvature. In: *Proceedings of ACM International Conference Proceeding Series* (2003)
23. Warfield, S., Kaus, M., Jolesz, F., Kinkinis, R.: Adaptive template moderate spatially varying statistical classification. *Med. Image Anal.* 4, 43–55 (2000)