On Accuracy of Marching Isosurfacing Methods

Cuilan Wang, Timothy S. Newman, and Jong Kwan Lee

The University of Alabama in Huntsville, Huntsville, AL, USA

Abstract

Representational accuracy of the isosurface meshes produced by well-known marching isosurfacing methods is considered. Accuracy is determined versus the isosurface given by trilinear interpolation and quantified using a new measure of volumetric divergence. The new measure is an accurate estimate of spatial discrepancy between a produced mesh and the isosurface of trilinear interpolation. Through experimental testing using the measure, suitability of the considered methods to well-model the isosurface of trilinear interpolation's behavior on grid cell interiors is also established.

Keywords: Isosurfaces, Marching Cubes, Trilinear Interpolant, Volumetric Data

1. Introduction

Visualizing isosurfaces is an important means to explore data and discover knowledge in volumetric datasets. Many isosurface extraction methods have been developed for application to datasets comprised of scalar data values at 3D rectilinear grid points and produce a triangular mesh approximation of the true isosurface. Such meshes have an advantage of being rendered quickly, even on older commodity graphics hardware. Many methods have also been developed to improve the performance, topological correctness or accuracy of the popular isosurface extraction methods, with much of the effort focused on marching methods [NY06].

Production of an isosurface mesh from rectilinear grid datasets typically entails use of a data value variation model for placement of mesh vertices. Early isosurfacing works usually modeled variation using a linear model only along grid lines. Some recent work has proceeded under less simple models of variation. That line of work has culminated with schemes that produce isosurface meshes with properties tailored to follow those of the isosurface given by trilinear interpolation.

In this paper, we examine the discrepancies of well-established marching isosurface methods against the isosurface given by trilinear interpolation. The work here seeks to gauge the suitability of well-established methods that arose from simple variation model assumptions and of one of the recent works that uses a less simple variation model. In addition to providing analysis that addresses suitability, this paper introduces and employs a new measure of divergence between isosurfaces (in particular, watertight isosurfaces). The measure can potentially aid designers in the development of new isosurfacing algorithms and help isosurface visualization users evaluate potential isosurfacing solution suitability according to the requirements of their problem, such as the level of quality, computational cost, etc.

The paper is organized as follows. In Section 2, background and related work are introduced. In Section 3, the new measure is described. In Section 4, the analysis results are presented. The paper is concluded in Section 5.

2. Background and Related Work

Marching Cubes (MC) has come to be a prototypical isosurfacing method for scalar rectilinear volumetric datasets. In MC, the dataset is processed one grid cube (cell) at a time in a scan-line (marching) fashion. As it marches, the cell vertices less than the isovalue are labelled as *outside*. The other vertices are labelled as *inside*. If the isosurface encloses some portion of the volume (i.e., if it is watertight and does not intersect the boundaries of the volumetric dataset), then the vertices marked as inside are within a region wrapped by the isosurface and the other vertices are on the isosurface or outside the region it wraps. There are 256 possible sets of cube labelings. The original MC considered rotational and reflective cube symmetry, which reduces the 256 sets to 15 base configurations. It also employed linear interpolation along edges to determine isosurface mesh vertices (on grid edges terminated by oppositely labelled end points). MC links the vertices to form the mesh according to the connection topologies for the cube's base configuration.

2.1. Marching Isosurfacing

The original MC can produce an isosurface mesh that is not watertight due to its face ambiguity problem (i.e., the common face of adjoining cells can have mesh vertices that are linked differently on one side than the other). Many workarounds to this problem have been proposed [NY06]. Exploiting only rotational symmetry to determine the topological base configurations is one. Such an approach yields 23 base cases [Nie03]. The Marching Tetrahedra (MT) [PT90] method, which first decomposes each cube into tetrahedra and then extracts the approximating mesh from the tetrahedra, is another. MT produces a mesh with more facets than MC, however [SM04].

Some recent work ([LB03, Nie03]) has addressed both the face and internal ambiguity problems of the original MC. The internal ambiguity problem for the original MC is that the mesh topology within cubes can lack or add linking "tunnels" between nearby isosurface components. Specifically, the linking topologies that are produced can differ from the isosurface given by the trilinear interpolant on the cube's values [Che95]. The approaches of Lopes and Brodlie [LB03] and Nielson [Nie03] allow creation of an internal ambiguity-free (topologically correct) isosurface that aims at maintaining the trilinear interpolant's component linking topology. Carr [Car07] has recently shown that the configurations given in Nielson's method [Nie03] are complete and correct. The Lopes and Brodlie [LB03] method also seeks to produce an isosurface that closely matches the position of the isosurface given by the trilinear interpolant, mainly by allowing some isosurface mesh vertices to be at locations other than grid edges.

Cignoni et al. [CGMS00] have also described a method that can produce isosurface mesh vertices that are not only on grid edges. The method progressively refines its mesh by subdividing facets and moving facet vertices toward the isosurface given by trilinear interpolation.

Variants on MC that can produce higher resolution meshes or reduce the computational cost of isosurfacing have also been presented. For example, Dividing Cubes (DC) [CLL88] produces a high resolution isosurface by first recursively subdividing each active cube into eight equal size subcubes until certain termination criteria are satisfied and then applying MC in the subcubes. The DC isosurface mesh can have a large number of facets when the degree of subdivision is high. Another variant is Discretized Marching Cubes (DMC) [MSS94]. It can quickly produce a compact mesh representation. Since it uses the midpoints of intersected edges as the mesh vertices, its mesh is lower resolution than the MC mesh, however.

2.2. Trilinear Interpolation

The isosurfacing methods that consider the trilinear interpolant use the trilinear interpolation function G:

$$\begin{aligned} G(x,y,z) = {} & G_{000}(1-x)(1-y)(1-z) \\ & + G_{100}x(1-y)(1-z) + G_{010}(1-x)y(1-z) \\ & + G_{110}xy(1-z) + G_{001}(1-x)(1-y)z \\ & + G_{101}x(1-y)z + G_{011}(1-x)yz + G_{111}xyz, \end{aligned} \tag{1}$$

where x, y, and z represent a location in a cube in local coordinates and $G_{i,j,k}$ denotes the value at a grid cell corner. Since the local coordinates vary from 0 to 1, the i, j, and k values can be 0 or 1. An isosurface defined on such a scalar field is denoted as the *isosurface given by trilinear interpolation* or the *trilinear interpolation isosurface*.

2.2.1. Representation Accuracy Measurements

Due to the complex shape of the trilinear interpolation isosurface, developing good metrics to measure the accuracy of the isosurface mesh produced by an isosurfacing method against the trilinear interpolation isosurface is challenging. The most common way to check it has been to use visual comparison. Often a very high resolution triangular mesh has been used instead as the reference for comparison with the extracted isosurface (e.g., [LB03, HTF97]). In addition, some researchers have used visual comparison to compare a newer isosurfacing method's extracted isosurface mesh against the mesh produced by a classic algorithm (e.g., [LB03,HTF97,CGMS00]). Visual examination can allow assessment of smoothness and of retention of notable shape features (such as protrusions).

When a high resolution mesh is used as the benchmark for a computational comparison of one extracted isosurface against a reference isosurface, often measures related to the closeness of the meshes are used to determine extracted mesh accuracy. Use of distance-based metrics or derived global geometric metrics are two closeness measure strategies.

Examples of global geometric metrics are isosurface surface area and volume inside the isosurface [TPG99]. However, such metrics don't well-measure local deviation of meshes. The nature of this shortcoming is illustrated in Figure 1, which shows cut away views of regions R and R' which contain two surfaces A (shown in red) and B (shown in blue). Surfaces A and B have the same boundary size and they enclose the same amount of space in both subfigures. Yet, the surfaces are more distant in (b) than in (a). The boundary size and enclosed region measures here fail to differentiate that the left case has more similar boundaries than does the right case.

Distance-based metrics, such as quadric error metrics [GH97], Hausdorff distance [KLS96], etc., are commonly used in graphics (e.g., in mesh simplification) for

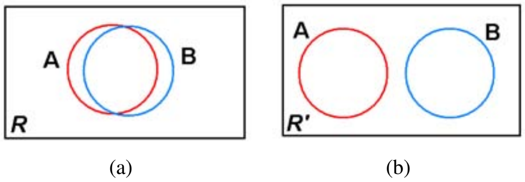


Figure 1: *Global geometric measure shortcomings illustration.*

measuring closeness between meshes. However, such metrics usually consider only some points (e.g., mesh vertices) of the mesh to determine the metric value. Some distance-based metrics attempt to produce the sum of mesh deviations at densely sampled locations of the meshes (e.g., not only deviation between mesh vertices), such as the Metro scheme [CRS98]. Since such an approach is not exhaustive, it can not always well-approximate divergence between meshes. Its measure is also only one-sided (i.e., not symmetric) [CRS98]. Distance-based metrics also often use approximate measures of mesh distance.

3. Isosurfacing Accuracy Evaluation

In this paper, the *Volumetric Divergence Computation (VDC)* is proposed to evaluate the quantitative accuracy of marching isosurfacing methods.

3.1. The New Metric: Volumetric Divergence for a Mesh

We first define the **volumetric divergence** between two closed surfaces. The regions inside (i.e., enclosed by) and outside (i.e., not enclosed by) some surface A are denoted A_{in} and A_{out}, respectively. The volumetric divergence between two closed surfaces A and B uses a region R, defined as:

$$R = (A_{in} \cap B_{out}) \cup (A_{out} \cap B_{in}) \ . \tag{2}$$

The volumetric divergence between surface A and B is defined as the volume of the region R, which is denoted as $V(R)$. $V(R)$ can also be called a symmetric difference. Such differences have been used previously for 2D divergences (e.g., [Vel01]). Figure 2 illustrates the volumetric divergence of two surfaces in the same 3D space in a 2D view. Figure 2 (a) shows a cut-away view of the region enclosed by surface A (the yellow region is inside A). In Figure 2 (b), a cut-away view of the region enclosed by B is shown (the blue region is inside B). The cut-away view of the volumetric divergence between the two surfaces is the red region in Figure 2 (c). Volumetric divergence measures absolute divergence only; it (or a variant on it) must be used judiciously if relative differences are needed.

The new metric we propose to evaluate marching isosurfacing methods is the volumetric divergence between the triangular mesh isosurfaces extracted by a marching isosurfacing method and the trilinear interpolation isosurface. Only isosurfaces that should form a closed enclosure of space within the dataset's extents are considered. Next, the computation of the volumetric divergence is described.

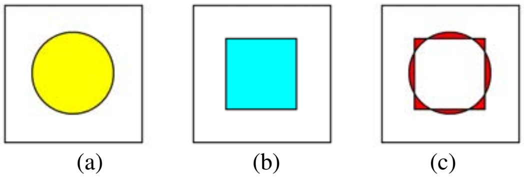


Figure 2: *Illustration of a 2D view of volumetric divergence between two surfaces.*

3.2. Divergence Metric Calculation

Since the shape of a trilinear interpolation isosurface is very complex, it is very hard to directly compute the exact volumetric divergence between it and the extracted isosurfaces. Instead, we utilize a divide and conquer process that nearly produces exact volumetric divergence.

In the computation, first volumetric divergence within each unit cube (i.e., grid cell of the volume) is computed. Then, $V(R)$ is obtained by summing each cube's volumetric divergence (i.e., between the trilinear interpolation isosurface and the extracted isosurface). Specifically, if $V(R_C)$ denotes the volumetric divergence for a unit cube, where $R_C \subseteq R$, then

$$V(R) = \sum_{j=1}^{N} V(R_{C_j}), \tag{3}$$

where R_{C_j} is the volumetric divergence in the j^{th} cube and N is the number of cubes. Finding each cube's volumetric divergence involves first recursively and evenly subdividing the cube into 8 subcubes until a certain subdivision depth is reached. The data value at each corner vertex of a subcube is defined by trilinear interpolation. Then, each subcube is determined to be inside, outside, or on (i.e., intersecting) the trilinear interpolation isosurface and labeled as such. In addition, each subcube is tested to find if it is inside, outside, or on the extracted triangular mesh isosurface and labeled as such. The steps to find the subcube's locational relationships with the trilinear interpolation and extracted isosurfaces are given in Sections 3.3 and 3.4, respectively.

One subcube's locational relationships with the trilinear interpolation and extracted isosurfaces are shown in Figure 3. In this and other figures in the paper, a unit cube vertex marked with a black dot is an inside vertex. In this case, the subcube is inside the trilinear interpolation isosurface and outside the extracted isosurface.

When the subcube does not intersect either the extracted isosurface or the true isosurface, comparing its labels can determine whether it belongs to region R_C: if the subcube is inside or outside both the trilinear interpolation and extracted

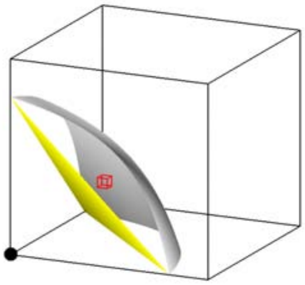


Figure 3: *A scenario showing a subcube's locational relationship with the trilinear interpolation isosurface (shown in grey) and the extracted isosurface (shown in yellow) for one unit cube. The subcube is shown in red.*

With Tri. Interp. Isosurfaces	With Extracted Isosurfaces	Contribution to $V(R_C)$
inside	inside	0
inside	outside	v
inside	on	0.5v
outside	inside	v
outside	outside	0
outside	on	0.5v
on	inside	0.5v
on	outside	0.5v
on	on	0

Table 1: *Illustration of the final cube's contribution to volumetric divergence $V(R_C)$ for all locational relationships with the trilinear interpolation and extracted isosurfaces. The volume of the final cube is v.*

isosurfaces, then it does not belong to region R; otherwise, it belongs to region R_C and its volume should be included in $V(R_C)$.

If a subcube does intersect either the trilinear interpolation or extracted isosurfaces, then it will be further subdivided (unless the subdivision limit has been reached). We term the smallest cube (i.e., when the subdivision limit is reached) the *final cube*. If the final cube still intersects the trilinear interpolation (or extracted) isosurface, then it is considered as half inside and half outside the trilinear interpolation (or extracted) isosurface. For a final cube with volume v, Table 1 shows its contribution to $V(R_C)$ for all possible locational relationships with the trilinear interpolation and extracted isosurfaces.

The core steps of VDC are given in Figure 4, as function *compute divergence (cube, level)*. In the function, the *cube* parameter contains the locations of and data values at the 8 corner vertices of the cube. Also, *level* is the current level of subdivision and the *subdivision depth* is the maximum subdivision level. The volumetric divergence for a unit cube is computed by calling the function with parameters (unit cube, 1). The VDC is performed for each unit cube.

If the final cube is infinitely small, then the result of the VDC method is the exact value of $V(R)$. Since our process is finite, the VDC yields an approximate value for $V(R)$. Later in this paper, we show experiments that indicate when the subdivision depth is large enough, this approximation can be reasonable.

3.3. Subcube vs. Trilinear Interpolation Isosurfaces

Because of the property of linear variation along each axis of the trilinear interpolation, it is not hard to prove that the value determined using the trilinear interpolation on the 8 corner values of the unit cube is the same as the value determined by the the trilinear interpolation function defined by 8 corner values of the subcube (for any point in the subcube). Based on this property, it is obvious that if the values at the 8 corner vertices of the subcube are all larger than or equal to the isovalue, then the data values inside the subcube will be all larger than or equal to the isovalue and hence the entire subcube will be inside the region enclosed by the trilinear interpolation isosurface. Similarly, if the values at the subcube corners are all smaller than the isovalue, then the subcube will be totally outside the region enclosed by the trilinear interpolation isosurface. Thus, by checking the 8 values at the subcube corners, the subcube's locational relationship with the trilinear interpolation isosurface can be determined.

3.4. Subcube vs. Triangular Meshes

Testing the subcube's locational relationship with isosurface triangular meshes is done using the following process. First, determining if the subcube intersects the triangular mesh within the unit cube is done using Voorhies' [Voo92] algorithm to test for cube-triangle intersection. Any subcube that does not intersect any triangle of the unit cube's triangular mesh is either totally inside or outside the region(s) enclosed by the triangular mesh.

To determine if the subcube is actually inside or outside, a strategy that tests one point in the subcube is utilized. The strategy is based on observation of a closed surface mesh that divides 3D space into two regions: one inside (enclosed by) this surface, and the other outside this surface. For such surfaces, if a line segment has one endpoint that is inside and one endpoint that is outside, it must intersect the surface an odd number of times. If the line segment has two endpoints that are both inside the surface, then it must intersect the surface an even number of times. Thus, in our strategy, we choose a corner vertex, U, (of the unit cube) that has a data value larger than the isovalue as one endpoint of a line segment. This point is an inside vertex. We choose the center of the subcube, F, as illustrated in Figure 5, as the other endpoint. By testing whether the number of intersection points between line segment UF and the triangular mesh is even or

```
compute divergence ( cube, level ) {
    V=0;   //V is the volume divergence
    divide cube into 8 subcubes;
    for each subcube do {
        E = compare subcube and extracted isosurface;
        // E = 1, 0, or −1 if subcube is inside, outside, or on the extracted isosurface, respectively.
        T = compare subcube and trilinear interpolation isosurface;
        // T = 1, 0, or −1 if subcube is inside, outside, or on the trilinear interpolation isosurfaces, respectively.
        if ((E = 1) and (T = 0)) or ((E = 0) and (T = 1))   //if subcube is in region R
            V = V + 1 / 8;
        if (E=−1) or (T=−1)//if subcube intersects either the extracted or trilinear interpolation isosurface
            if (level < subdivision depth)
                //if the deepest subdivision level is not reached, recursively call this function
                V = V + compute divergence ( subcube, level+1 ) / 8;
            else //if the deepest subdivision level is reached
                if ((E = −1) and (T = 0)) or ((E = −1) and (T = 1)) or ((E = 0) and (T = −1)) or ((E = 1) and (T = −1))
                    V = V + 0.5 / 8;
    }
    return V;
}
```

Figure 4: *Divergence Calculation Function*

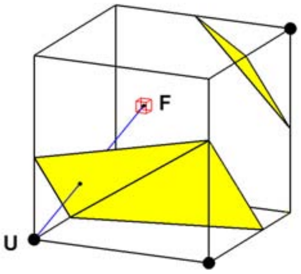


Figure 5: *Illustration of testing the subcube's locational relationship with triangular meshes.*

odd, it can be determined whether F is inside or outside the triangular mesh. Once F's status is known, we know if the whole subcube is inside or outside the triangular mesh.

4. Results

Next, analyses of marching isosurfacing accuracy are reported. Eight datasets, named CT1, CT2, CT3, CT4, MR1, MR2, MR3, and MR4 are used in testing. The first four are CT datasets. The last four are MRI datasets. Table 2 describes characteristics of the datasets, including the number of active cubes in each dataset for isovalue 70.5. Except for MR4, all these datasets are from Roettger's Volume Library. For the experiments here we have also padded each dataset with bounding zeros so that isosurfaces will totally enclose regions of space within the dataset.

Datasets	Name	Dimension	# of Active Cells
MR1	Tomato	256×256×64	4704
MR2	Orange	256×256×64	396364
MR3	Frog	256×256×44	144798
MR4	Human Brain	256×256×72	85908
CT1	Engine	256×256×256	313133
CT2	Lobster	301×324×56	133200
CT3	Monkey Head	256×256×62	194417
CT4	Human Head	128×256×256	453575

Table 2: *Volumetric datasets*

4.1. Subdivision Depth

As mentioned before, the subdivision depth in dividing the unit cube into small final cubes defines the precision of the isosurface volumetric divergence measure $V(R)$. Here, we report tests that consider the effect of subdivision depth on the precision of $V(R)$. The tests involved computing $V(R)$ of a dataset using subdivision depths from 1 to 10, with the 23 case MC applied in each final cube. The results from the Frog (MR3) and Lobster (CT2) data sets are plotted in Figure 6. The results show that the volumetric divergence initially changes greatly but then converges (as subdivision depth increases). The difference between the subdivision depth 7 and 10 results are less than 0.007% in both datasets. If the length of a unit cube side is 1, for the subdivision depth 7, each final cube has a side length $1/2^7 = 0.0078125$.

We also considered the difference in results for the 7 and 10 subdivision depths on a base case-by-base case basis for the 23 case Marching Cubes. 100 instances of each base case were randomly generated for Monte Carlo testing as follows. In these tests, since the datasets were byte-formatted for our study, the outside labeled vertices were randomly generated

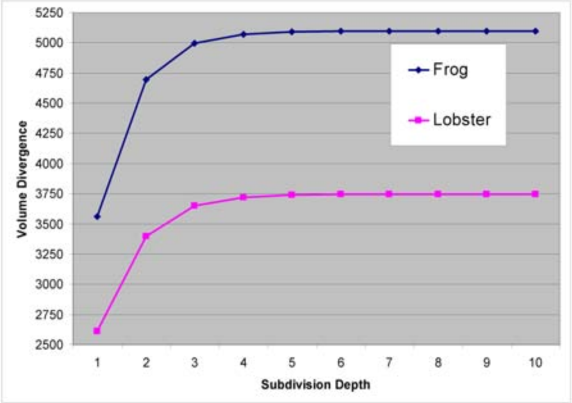


Figure 6: *Volumetric divergence computed by VDC with subdivision depths 1 to 10*

in the range of $[0, isovalue)$; the inside labeled vertices were randomly generated in the range of $(isovalue, 255]$. Several isovalues were tested. The maximum divergence in any case between the level 7 and level 10 results was 0.00027. The average divergence between these levels was 0.00003. Thus, even for the worst base case, the difference between 7 and 10 as the terminal level is small. Hence, we used subdivision depth 7 as the baseline for comparison in the rest of our experiments.

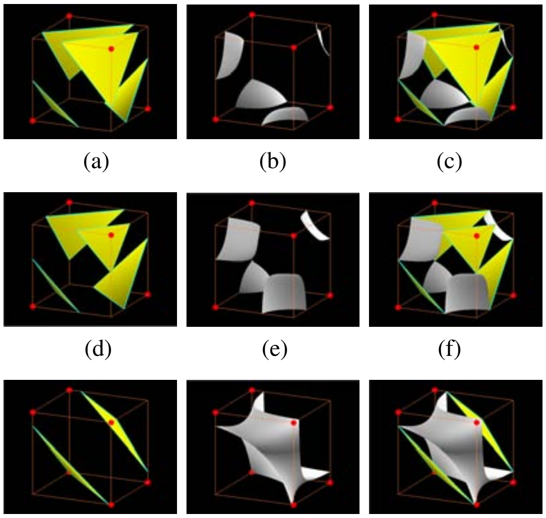
4.2. Study of MC23 Cases

Next, the volumetric divergence from the baseline trilinear interpolation isosurface for each base case of the 23 case Marching Cubes Algorithm is studied. Given an isovalue, for each base case Monte Carlo testing on 100 instances were randomly generated in the way described in Section 4.1. Three isovalues were tested: 50, 128, and 180. For each isovalue, the average, maximum, and standard deviation of the volumetric divergences for each base case are shown in Table 3. Since in Case 0 and Case 22 all vertices have the same labels, the isosurfaces don't intersect the cube. Thus, these cases have a volumetric divergence of 0.

The results show that for a given isovalue, some cases tend to have quite large volumetric divergences, such as Cases 13 and 7 for isovalue 50. However, some cases (e.g., Case 2 for isovalue 50) tend to have quite small volumetric divergences. Which case may have large (or small) volumetric divergences is somewhat dependent on the isovalue. Figure 7 (a)–(c) display renderings of the instance with maximum volumetric divergence for isovalue 50. In this case, it was Case 13. (The figure shows the MC23 result, the trilinear interpolant isosurface, and the two overlaid.) It is obvious that the reason that Case 13 has large volumetric divergence is because face ambiguity, while resolved, was not resolved consistent with the trilinear interpolant. Figure 7 (d)–(f) show the instance with maximum volumetric divergence (i.e., Case 13) for isovalue 128. Figure 7 (g)–(i) show the instance with maximum volumetric divergence for isovalue 180. The base case here was Case 18. For this instance, the internal ambiguity is not resolved correctly.

Case	Isovalue								
	50			128			180		
	Max.	Avg.	Std.	Max.	Avg.	Std.	Max.	Avg.	Std.
0	0	0	0	0	0	0	0	0	0
1	0.3390	0.1365	0.0919	0.0824	0.0205	0.0204	0.0188	0.0033	0.0041
2	0.3388	0.1716	0.0608	0.1502	0.0432	0.0264	0.0721	0.0165	0.0130
3	0.6260	0.3518	0.1138	0.3310	0.0794	0.0660	0.0691	0.0129	0.0145
4	0.6391	0.3077	0.1905	0.1752	0.0385	0.0374	0.0396	0.0064	0.0084
5	0.2693	0.1113	0.0419	0.2400	0.0859	0.0441	0.2441	0.1185	0.0632
6	0.5973	0.3816	0.0921	0.4119	0.1337	0.0893	0.1669	0.0353	0.0359
7	0.6647	0.5120	0.1213	0.5279	0.1815	0.1428	0.2706	0.0394	0.0530
8	0.2134	0.0529	0.0455	0.1608	0.0408	0.0255	0.1732	0.0422	0.0301
9	0.2605	0.1441	0.0407	0.2055	0.0912	0.0419	0.3358	0.1500	0.0599
10	0.5145	0.3489	0.0829	0.4181	0.2304	0.1096	0.3767	0.0672	0.0749
11	0.2588	0.1721	0.0376	0.2404	0.1173	0.0415	0.3289	0.1473	0.0486
12	0.4306	0.2305	0.0783	0.3680	0.1806	0.0771	0.3551	0.1363	0.0810
13	0.7461	0.6538	0.0509	0.7340	0.3744	0.1972	0.5570	0.0798	0.1094
14	0.2979	0.1760	0.0362	0.3056	0.1206	0.0432	0.3040	0.1587	0.0485
15	0.3357	0.2108	0.0636	0.4010	0.2315	0.0864	0.3543	0.1667	0.0694
16	0.4239	0.2249	0.0690	0.4426	0.2892	0.0759	0.4900	0.1715	0.0792
17	0.2360	0.0782	0.0425	0.1719	0.0677	0.0333	0.1481	0.0755	0.0274
18	0.0234	0.0028	0.0040	0.2764	0.0566	0.0572	0.5952	0.2187	0.1561
19	0.1592	0.0527	0.0346	0.2562	0.1046	0.0431	0.2042	0.1047	0.0288
20	0.0657	0.0089	0.0113	0.1149	0.0400	0.0253	0.2336	0.1088	0.0504
21	0.0120	0.0008	0.0016	0.1234	0.0186	0.0214	0.2940	0.0706	0.0611
22	0	0	0	0	0	0	0	0	0

Table 3: *Volumetric Divergence for 23 Cases of MC23*


(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 7: *Max. volumetric divergence case for Isovalue 50 (Case 13): (a)-(c), Isovalue 128 (Case 13):(d)-(f), Isovalue 180 (Case 18): (g)-(i). (a), (d), and (g) show the extracted isosurfaces. (b), (e), and (h) show the trilinear interpolation isosurfaces. (c), (f), and (i) show overlays of the isosurfaces.*

In summary, MC23 isosurfacing produces only marginally accurate results for some base cases. Morever, for some isovalues, some case can exhibit meshes that are quite errant.

4.3. Accuracy of Marching Isosurfacing Algorithms

Next, we report an evaluation of several marching isosurfacing algorithms based on the volumetric divergence between the algorithms' isosurfaces and the true trilinear interpolation isosurfaces. To our knowledge, the work here is the first quantification of the accuracy of classic marching isosurfacing algorithms versus the trilinear interpolant isosurface. Focus here is on isosurfaces that are watertight.

The algorithms tested include 23 case Marching Cubes (MC23), Discretized Marching Cubes (DMC), Dividing Cubes with dividing depths of 1, 2, 3, and 4 (DC1, DC2, DC3, and DC4), and Marching Tetrahedra (MT). DMC was implemented to produce mesh vertices at midpoints of intersected cube edges. The connection topologies used for DMC were based on the 23 base cases that result from use of only rotational symmetry. The Dividing Cubes algorithms were also based on MC23. The MT used subdivision of each cube into 6 tetrahedra.

Several isovalues were tested for the 8 datasets. Due to the spacing limit, we only reported results for isovalue 70.5. The results we report include the average, maximum, and standard deviation of the volumetric divergence per cube. Table 4 summarizes these results. Three primary conclusions can be drawn from the results. First, it is clear that MC23 is more accurate than MT, even though MC23 produces fewer mesh facets than MT. In MT, the intersection points' locations are computed using linear interpolation along the diagonals of the cube faces and diagonals of the cubes. Those locations are not on the true trilinear interpolation isosurfaces. Thus, the mesh vertex locations do not always well-match locations on the trilinear interpolation isosurface. Second, the results show that DMC has even less accuracy than MT. DMC's poor accuracy is due to its use of edge midpoints as mesh vertices. Third, the results also show that Dividing Cubes (DC) has better accuracy than MC23. In DC, the mesh vertex points on each final cube at least are on the true trilinear interpolation isosurface. Since DC has more triangle vertices on the true trilinear interpolation isosurfaces than those from MC23 (due to DC's subdivisions), DC is more accurate. In addition, the deeper the dividing cube depth, the more vertices there are on the true isosurfaces, thus the better the triangular mesh becomes. The results reflect this tendency among the different levels of DC and show a factor of 3 to 4 accuracy improvement for each increase in depth here.

Lopes and Brodlie's marching isosurfacing method [LB03] was also tested on three of the datasets, Tomato (MR1), Engine (CT1), and Lobster (CT2). The average volume divergence per cube for them are 0.0091, 0.0063, and 0.0033, respectively. We note that the Lopes and Brodlie method, of course, achieved good accuracy because of its improved topological correctness in the interior of the cube. It is also interesting that DC even with just one subdivision is already almost as accurate as the Lopes and Brodlie results (e.g., for MR1). Lastly, we note that testing on other data suggests that RMS and Metro schemes [CRS98] sometimes lead to mistaken conclusions about the relative goodness of Lopes and Brodlie results.

Data		DMC23	MT	MC23	Dividing-MC23			
					1	2	3	4
MR1	Max.	0.6788	0.4530	0.4068	0.1201	0.0305	0.0087	0.0041
	Avg.	0.0886	0.0393	0.0310	0.0096	0.0032	0.0012	0.0005
	Std.	0.0964	0.0553	0.0532	0.0135	0.0040	0.0014	0.0006
MR2	Max.	0.9114	0.6403	0.7299	0.2140	0.0532	0.0139	0.0065
	Avg.	0.1141	0.0609	0.0482	0.0136	0.0043	0.0016	0.0006
	Std.	0.1110	0.0743	0.0711	0.0167	0.0047	0.0016	0.0006
MR3	Max.	0.9159	0.6134	0.7162	0.1587	0.0398	0.0116	0.0044
	Avg.	0.1137	0.0451	0.0352	0.0110	0.0038	0.0015	0.0006
	Std.	0.0996	0.0569	0.0553	0.0138	0.0041	0.0015	0.0006
MR4	Max.	0.7422	0.6053	0.5802	0.1587	0.0419	0.0138	0.0042
	Avg.	0.0911	0.0401	0.0362	0.0114	0.0037	0.0013	0.0005
	Std.	0.0944	0.0571	0.0611	0.0158	0.0045	0.0015	0.0006
CT1	Max.	0.4936	0.2863	0.3841	0.1458	0.0458	0.0181	0.0054
	Avg.	0.1256	0.0076	0.0121	0.0054	0.0023	0.0010	0.0005
	Std.	0.0916	0.0101	0.0289	0.0098	0.0033	0.0013	0.0006
CT2	Max.	0.8304	0.5503	0.5774	0.1701	0.0476	0.0122	0.0041
	Avg.	0.1091	0.0306	0.0281	0.0092	0.0033	0.0013	0.0006
	Std.	0.0934	0.0470	0.0499	0.0128	0.0039	0.0015	0.0006
CT3	Max.	0.7439	0.5182	0.4328	0.1330	0.0408	0.0110	0.0060
	Avg.	0.1165	0.0256	0.0199	0.0075	0.0028	0.0012	0.0005
	Std.	0.0907	0.0322	0.0364	0.0109	0.0035	0.0013	0.0006
CT4	Max.	0.7658	0.5688	0.5500	0.1777	0.0508	0.0169	0.0313
	Avg.	0.1342	0.0460	0.0389	0.0125	0.0039	0.0014	0.0007
	Std.	0.1250	0.0558	0.0561	0.0143	0.0042	0.0015	0.0021

Table 4: *Volumetric divergence per cube for several marching isosurfacing methods with isovalue 70.5*

One visual comparison between the marching isosurfacing algorithms is shown in Figure 8. Figure 8 (a) shows one view of a Lobster dataset isosurface from MC23. The region in the yellow box in Figure 8 (a) is enlarged in the other subfigures to show the details of the isosurfaces. In those results, it is evident that DMC produces the worst isosurface shape. (Its isosurface is much more jagged than the others.) MT is also poor as it produces some extraneous, false features due to its topological incorrectness. These features are most visible near the right lower corner where there are wave-like features and some jagged edges to isosurface components. MC23 produces medium quality isosurfaces. The isosurface extracted by the Lopes and Brodlie method is smoother and has more topological accuracy than the first three methods. The DC results are quite similar to the Lopes and Brodlie results. As the subdividing depth of DC increases, the extracted isosurface becomes quite smooth.

5. Conclusion

This paper introduces use of volumetric divergence to assess isosurfacing accuracy. This metric allows evaluation of the meshes produced by common marching isosurfacing approaches versus the isosurface given by trilinear interpolation. Our study shows that DMC has the worst accuracy among all the marching isosurfacing methods we tested. MT and MC23 produce medium quality isosurfaces and don't always maintain good topological correctness. It is also worth

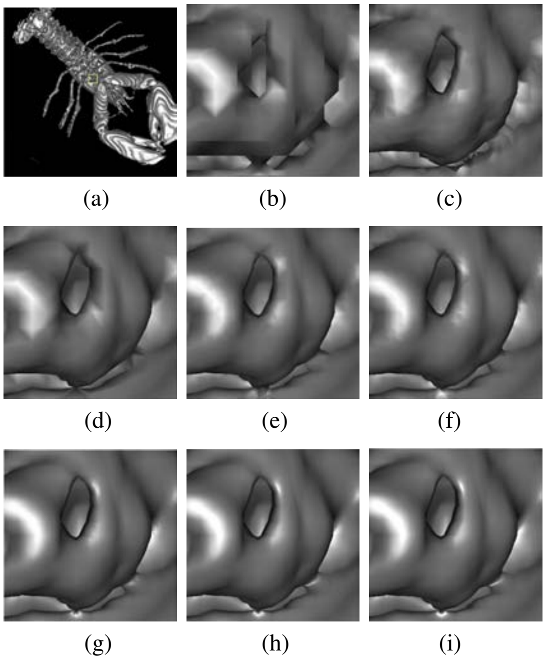


Figure 8: *Zoom-in Rendering Comparisons. A whole view of Lobster dataset rendered by MC23 is shown in (a). Zoomed-in views of isosurfaces extracted by DMC, MT, MC23, Lopes' method, DC1, DC2, DC3, and DC4 for the yellow box region in (a) are shown in (b), (c), (d), (e), (f), (g), (h), and (i), respectively.*

noting that MT tends to produce worse results than MC23, despite its production of a mesh with more facets. The Lopes and Brodlie method appears to have similar accuracy as DC with subdivision level 1. Both methods can maintain good topological correctness. However, as the subdividing depth of DC increases, DC has higher accuracy. Thus, DC, which is straightforwardly implemented, is a good choice for producing accurate isosurface meshes that are also face ambiguity free.

Acknowledgement

We acknowledge and appreciate Dr. Adriano Lopes for providing the access to the code for the Lopes and Brodlie isosurfacing method.

References

[Car07] CARR H.: (no) more marching cubes. In *IEEE/EG Int'l Symp. on Volume Graphics'07* (Prague, Czech Republic, 2007), pp. 81–90.

[CGMS00] CIGNONI P., GANOVELLI F., MONTANI C., SCOPIGNO R.: Reconstruction of topologically correct and adaptive trilinear isosurfaces. *Computers & Graphics 24*, 3 (2000), 399–418.

[Che95] CHERNYAEV E. V.: *Marching Cubes 33: Construction of Topologically Correct Isosurfaces*. Tech. Rep. Technical Report CERN CN 95-17, CERN, 1995.

[CLL88] CLINE H., LORENSEN W., LUDKE S.: Two algorithms for the three-dimensional reconstruction of tomograms. *Medical Physics 15*, 3 (1988), 320–327.

[CRS98] CIGNONI P., ROCCHINI C., SCOPIGNO R.: Metro: Measuring error on simplified surfaces. *Comp. Graphics Forum 17*, 2 (1998), 167–174.

[GH97] GARLAND M., HECKBERT P.: Surface simplification using quadric error metrics. In *Comp. Graphics (Proc., SIGGRAPH 97)* (1997), vol. 31, pp. 209–216.

[HTF97] HAMANN B., TROTTS I., FARIN G.: On approximating contours of the piecewise trilinear interpolant using triangular rational-quadratic bezier patches. *IEEE Trans. on Visualization and Comp. Graphics 3*, 3 (1997), 215–227.

[KLS96] KLEIN R., LIEBICH G., STRASSER W.: Mesh reduction with error control. In *Visualization '96* (San Francisco, 1996), pp. 311–318.

[LB03] LOPES A., BRODLIE K.: Improving the robustness and accuracy of the marching cubes algorithm for isosurfacing. *IEEE Trans. on Visualization and Comp. Graphics 9*, 1 (2003), 16–29.

[MSS94] MONTANI C., SCATENI R., SCOPIGNO R.: Discretized marching cubes. In *Visualization '94* (Washington D.C., 1994), pp. 281–287.

[Nie03] NIELSON G.: On marching cubes. *IEEE Trans. on Visualization and Comp. Graphics 9*, 3 (2003), 283–297.

[NY06] NEWMAN T. S., YI H.: A survey of the marching cubes algorithm. *Computers & Graphics 30*, 5 (2006), 854–879.

[PT90] PAYNE B. A., TOGA A. W.: Surface mapping brain function on 3d models. *IEEE Comp. Graphics and Apps. 10*, 5 (1990), 33–41.

[SM04] SCHROEDER W. J., MARTIN K. M.:. Overview of Visualization, *The Visualization Handbook*, C. Hansen and C. Johnson (eds.), Elsevier, New York, 2004.

[TPG99] TREECE G., PRAGER R., GEE A.: Regularised marching tetrahedra: Improved iso-surface extraction. *Computers & Graphics 23*, 4 (1999), 583–598.

[Vel01] VELTKAMP R. C.: Shape matching: Similarity measures and algorithms. In *Int'l Conf. on Shape Modeling & Apps.* (Genova, Italy, 2001), pp. 188–197.

[Voo92] VOORHIES D.:. Triangle-cube Intersection, *Graphics Gems III*, David Kirk (editor), Academic Press, San Diego, 1992.