

Marching Cubes without Skinny Triangles

Carlos A. Dietrich, Carlos E. Scheidegger, João L. D. Comba, Luciana P. Nedel
and Cláudio T. Silva, *Senior Member, IEEE**†

Abstract

Most computational codes that use irregular grids depend on the triangle quality of the single worst triangle in the grid: skinny triangles can lead to bad performance and numerical instabilities. Marching Cubes is the standard isosurface grid generation algorithm, and while most triangles it generates are good, it almost always generates some bad triangles. Here we show how simple changes to Marching Cubes can lead to a drastically reduced number of degenerate triangles, making it a more practical choice for isosurface grid generation, reducing or eliminating the need and costs of post-processing.

1. Introduction

Marching Cubes [9] is currently the most popular algorithm for isosurface extraction. It is elegant, simple, fast, and robust. While the output mesh Marching Cubes generates is adequate for visualization purposes, it is far from being suitable for use in numerical simulations. This deficiency arises from the degenerate triangles that MC typically generates, and that, for example, a single badly-shaped triangle can lead to ill-conditioning of an entire finite element simulation [12]. The current practice is to solve this problem by post-processing [1, 14], but here we present a simpler alternative. We first elucidate the causes of bad triangles in Marching Cubes, and then mitigate the problem with small specific changes.

Our discussion of Marching Cubes is based on the notion of Edge Groups, recently introduced by Dietrich et al. [3]. Each MC case generates up to 5 triangles, which are directly encoded in a fixed table. More importantly, each triangle is created using vertices placed along the edges of a fixed cube, and so there's only a limited number of ways a triangle is generated. We then identify equivalent triples of

edges under the cube's symmetries, and arrive at 8 different edge groups, illustrated in Figure 1.

Surprisingly, a single edge group is responsible for most degenerate triangles in MC. Some cases in the Marching Cubes table admit different triangulations, which use different edge groups. By systematically analyzing each case in the Marching Cubes table, we generate a table that leads to improved triangle qualities, building on our previous work [3, 4].

In the remainder of the paper, we focus on the practical aspects of improving MC to generate better-shaped triangles. The new improved table is available at <http://XXX>, together with supplemental material showing more extensive comparisons and results.

2. Marching Cubes Tables

Given a node-centric volumetric array of data approximating a scalar field $f(x, y, z) : R^3 \rightarrow R$ and a scalar value $k \in R$, MC produces a triangular surface that approximates the level set $f(x, y, z) = k$ (called the *isosurface*). The implementation of Marching Cubes follows a straightforward pipeline of actions that are executed for each cell in a given volume. It starts by computing the sign of each cell ver-

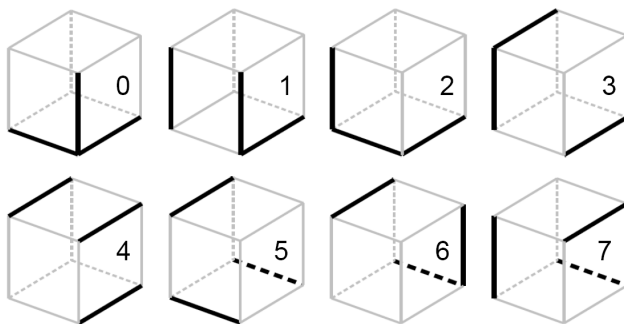


Figure 1. The eight Edge Groups in Marching Cubes. Every triangle in every MC configuration is created by one of these edge combinations.

* Carlos A. Dietrich, João L. D. Comba and Luciana P. Nedel are with the Instituto de Informática, UFRGS, Brazil, E-mail: [cadietrich,comba,nedel]@inf.ufrgs.br.

† Carlos E. Scheidegger and Cláudio T. Silva are with the Scientific Computing and Imaging Institute, University of Utah, USA, E-mail: [cscheid,csilva]@sci.utah.edu.

tex, determined by simply comparing a given vertex’s scalar value with k . The signs of all vertices from a cube define an eight-bit value that identifies a particular *case* in MC. There are two pre-defined tables that are indexed by this value: an *active edge* table, and a *triangulation* table (Figure 2).

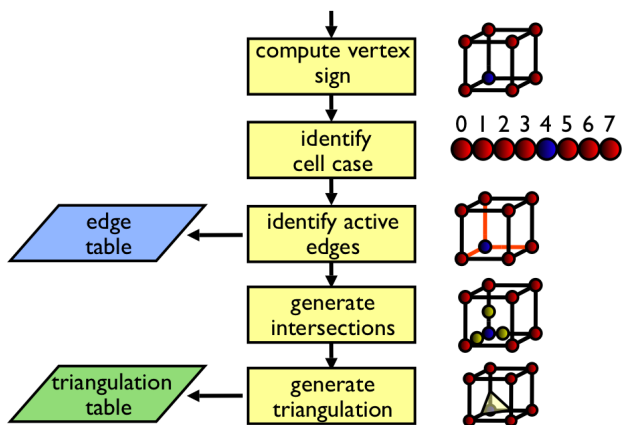


Figure 2. Marching Cubes pipeline. The active edges encoded in the edge table necessarily cross the isosurface, and are illustrated in orange. The triangulation table determines how to connect the vertices that lie on the active edges, creating the triangles for each patch. Creating the entries of the triangulation table carefully improves the triangle quality of MC.

The active edge table identifies, for each case, which edges of the cell are crossed by the isosurface, and therefore which intersections must be computed. The triangulation table correspondingly gives the set of triangles that will be generated from the active edges. A single MC case can generate up to 5 triangles. Most importantly, the encoding of some cases is not unique, as illustrated in Figure 3. Consideration of the triangulation tables is commonly given only up to homeomorphism of the reconstruction. In other words, any triangulation that has the same topology as the continuous level set that it is approximating is seen as equally good. As we will explain in the next section, the notion of Edge Groups allows us to effectively choose triangulations that generate systematically better triangles.

3. Edge Groups of Marching Cubes Cells

Our approach to improve MC is to use the quality information given by the edge groups involved in any particular

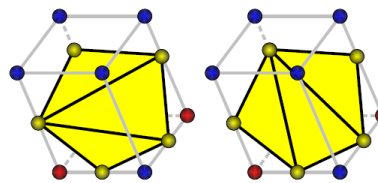


Figure 3. Two possible triangulations for the same MC case. In some situations, different triangulations will cause markedly different triangle qualities.

triangulation, and pick the one that maximizes some criteria. Here, we mainly use the ratio of incircle to circumcircle normalized to lie between zero and one; an equilateral triangle has maximum quality one [15]. However, the same idea directly applies to other measures such as min-angle and max-angle, as we show in Figure 4.

Our first analysis of the impact of different edge groups comes from plotting the probability density function of triangle quality for randomly selected triangles from each of the edge groups. In this initial model, the triangle distribution is given by assuming a uniform distribution of triangle vertices along edges, and assuming that the vertex choices are independent across edges. This gives a PDF for each edge group, illustrated in Figure 4. It is clearly apparent that edge group 2 has a qualitatively different behavior than the others: a substantial fraction of the triangles it creates are degenerate.

To test the robustness of the distribution assumptions for each edge group, Dietrich et al. collected edge group statistics on a collection of 30 volume datasets [3]. First, they collected edge group frequency data over isosurfaces extracted from each of the 30 volumes. These are summarized in Figure 5. As would be expected, edge groups are not equally probable. The second set of statistics presents a much clearer picture. By counting the edge groups of the 1000 worst triangles in each of the 30 extracted isosurfaces (presented in Figure 6), it becomes clear that edge group 2 is responsible for, typically, over 60% of the worst 1000 triangles in any given dataset, and in some cases this number is closer to 95%. Our strategy, then, is to systematically change the Marching Cubes tables to remove occurrences of edge group 2.

4. Improving Marching Cubes

Edge groups motivate a simple criterion for improving the MC table. Dietrich et al [3] propose a re-triangulation in certain table entries to prevent edge group 2 from occurring. Their proposal focuses on only a few MC cases, namely case 5, 12, 11, and the complement of case 6 [8].

Quality	Edge Groups							
<i>min/max</i>	0.0/1.0	0.0/1.0	0.0/0.828	0.0/1.0	0.698/1.0	0.698/1.0	0.0/1.0	0.698/1.0
<i>histogram</i>								

Figure 4. Edge Groups and their corresponding triangle quality.

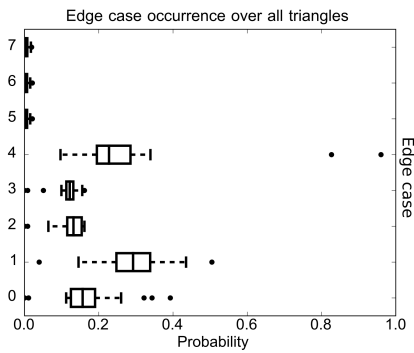


Figure 5. Edge groups occurrence in 30 datasets.

These changes update 96 entries of the MC table (120 entries if the table is constructed with the complement of case 6 [8]), but still leaves 56 entries with occurrences of edge group 2. Figure 7 shows examples where edge case 2 is removed. For some MC cases, however, it is not possible to remove edge group 2 by simply retriangulating the case: *ev-*

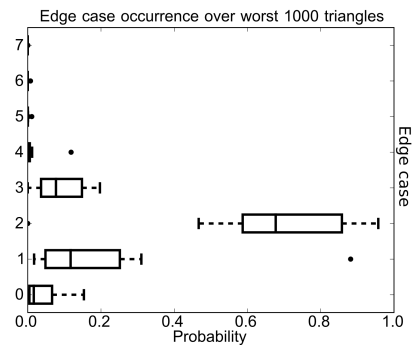


Figure 6. Edge groups occurrence in the 1000 worst triangles of 30 datasets.

ery triangulation of these cases include an instance of edge group 2 (see the left column of Figure 8).

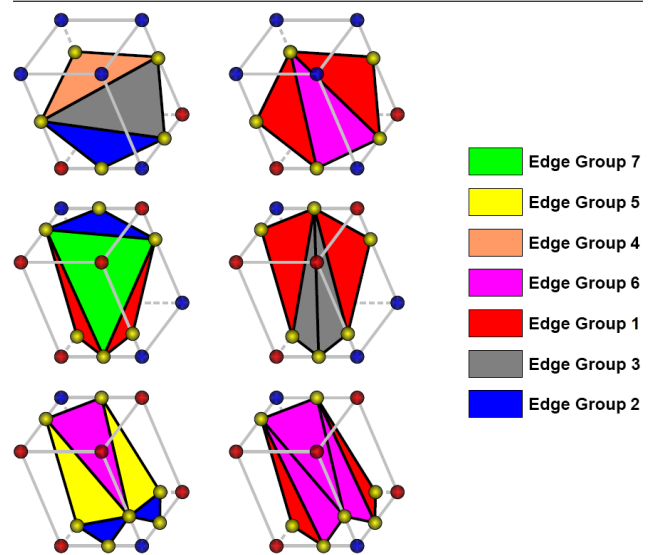


Figure 7. Reconnecting intersection vertices to remove the Edge Group 2. As described by Dietrich et al. [3], we can remove the Edge Group 2 from MC cases 5 (top), 11 (middle), and the complement of case 6 (bottom). Removing edge group 2, in its turn, results in an improvement of the generated mesh quality. The right column shows the retriangulation results.

4.1. Inserting a New Vertex in the Cell

As we have shown, retriangulating the intersection vertices cannot remove instances of edge group 2 for some MC cases. In these situations, we turn to an alternative approach.

By adding an *additional vertex* in the cell’s center and connecting it to the intersection vertices of active edges, we remove edge groups entirely from the Marching Cubes table. The resulting triangulations are illustrated in Figure 8. A similar approach is used in contexts as diverse as dual MC meshes [11] and MC mesh simplification [10], but here we emphasize its impact in connection to MC mesh quality. Additionally, the implementation of this change is quite straightforward and requires only small changes to the MC code.

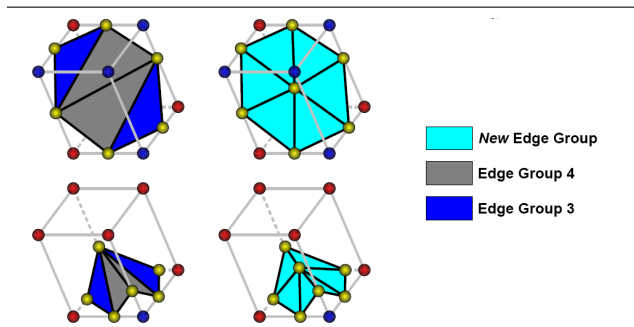


Figure 8. Placing a new vertex in the middle of the cell to remove the Edge Group 2. The retriangulation of the intersection vertices with help of an additional vertex allows the removing of the Edge Group 2 in MC cases 9 (up row) and the complement of the case 3 (bottom row).

To get an idea for the quality improvement created by adding an extra vertex, we note that the new configuration using the cell center can be seen as an additional edge group with only two edges. This single group generates all triangles shown in the right column of Figure 8. More importantly, its quality histogram is comparable to the best edge groups of the cubic cell.

The position of the new vertex in the cell is dependent on the MC case. The center of the cell can be a good choice for MC case 9, illustrated in the first row of Figure 8. In this case, a new triangulation with a vertex in the center of the cell will be close to the original MC triangulation. On the other hand, a new triangulation with a vertex in the center of the cell can result in artifacts in the complement of MC case 3. The artifacts are visible in situations where all intersection vertices are close to the *negative* vertices of the cell (blue vertices in Figure 8), in which the distance of the new vertex to the isosurface is maximum. To alleviate this problem, the new vertex is placed along one of the edges of the original MC triangulation, that is, in the middle of the longest edge of the triangulation. This guarantees that the

new triangulation is close to the original triangulation generated by MC.

These changes in the edge table improve the triangulation quality. However, most of the value comes from the synergy the new table has with the change to MC we will describe in the next section. Together, these two changes are such that the triangles generated by the MC suggested compare favorably to the state of the art.

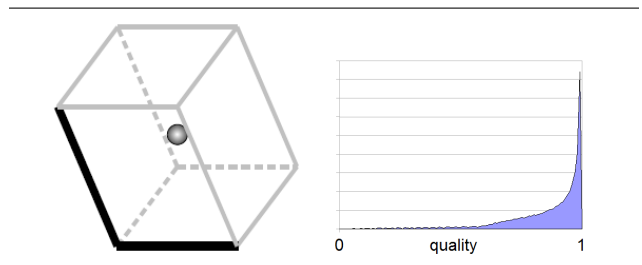


Figure 9. (Left) The Edge Group resultant from the retriangulation of cases 9 and the complement of case 3, which generates all triangles of Figure 8. (Right) The quality histogram of the triangles generated by the new Edge Group.

4.2. Transforming active edges

The second change to MC consists of perturbing the active edges on which intersection vertices are computed. The two edge endpoints are moved (by a small amount) inside the volume, and then the computation of the edge vertex proceeds as normal. Dietrich et al’s Macet (“Marching Cubes with Edge Transformations”) [4] adds two new intermediate steps to the MC pipeline, as described in Figure 10. The edge transformation step alters the positions of each edge extreme along the gradient or tangent directions (Figure 11). The second step, when necessary, displaces the intersection points away from edge extrema. Together, these steps tend to create active edges that are locally perpendicular to the isosurface, which leads to improved triangle quality. In order to enforce valid placement of edge endpoints (i.e. not crossing the isosurface), edge transformations are performed in several steps with smaller displacements along the proposed direction (in our experiments, we use eight steps).

As described, the drawback of the Macet proposal is that they do not have a criteria to choose which edge transformation to use. Instead, they perform both transformations, and do a neighborhood analysis that chooses the transformation that leads to local improved triangle quality. While

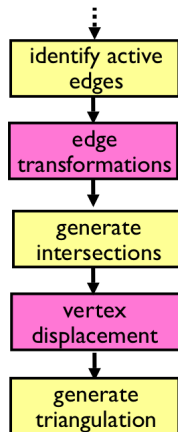


Figure 10. Macet pipeline adds two new stages to the MC pipeline: edge transformations and vertex displacement.

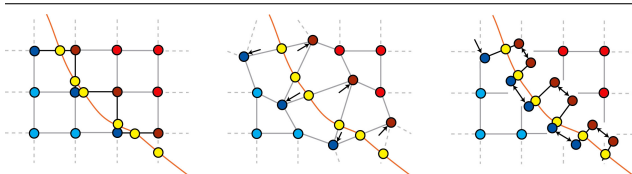


Figure 11. MC original grid, and after gradient and tangential transformations.

the local analysis is fast, the cost of using both transformations still leaves room for improvement.

In [3] they gave a different interpretation for the edge transformations that serves as room for the unification of edge transformations. They formulate the edge transformation as a projection operation of the edge midpoint onto the plane tangent to the isosurface. The same result can be accomplished using a new approach with unified edge transformations.

The idea is as follows. First we identify the edge extrema closest to the isosurface. This one will be subject to interleaved edge transformations using gradient and tangential transformations (8 in total, 4 for each type). The use of alternate transformations in sequence combines the properties of each transformation without requiring a second edge transformation step or subsequent neighborhood analysis. For the other extrema, it is moved to the edge midpoint, which under ideal circumstances is what the projection operation advocates.

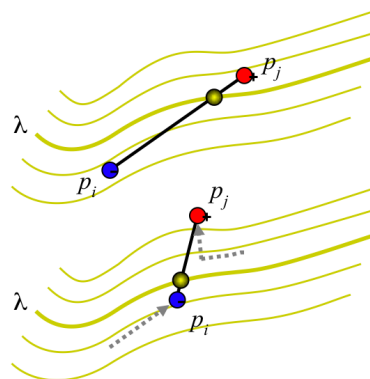


Figure 12. Unified Macet: one endpoint is subject to alternate tangential and gradient transformations, while the other is fixed to the original edge midpoint.

5. Results

The impact of the new MC table and Unified Macet were evaluated with experiments using a collection of 23 datasets. We summarize the results in Table 1¹. We compare results using two methods: the original MC and the Unified Macet with the extended edge table. For each case, we report minimum and maximal angles (θ_0 and θ_∞) and radii ratio (ρ).

Results clearly demonstrate that the Unified Macet approach using the new MC table generates consistently improved triangle quality in all datasets (worst radii ratio is 0.43). An intuition of the impact of the changes of the Unified Macet, we show in Figure fig:retriangulation-snapshot a zoomed version of a part of the Bonsai dataset.

Name	MC with old table			Macet with new table		
	θ_0	θ_∞	ρ	θ_0	θ_∞	ρ
Chest CT	0.08	179.0	0.0	17.9	118.6	0.46
Bonsai	0.38	178.7	0.0	17.6	119	0.45
Shockwave	1.26	175.7	0.0	20.7	110.7	0.52
Silicium	0.66	177.4	0.0	18.7	117.3	0.47

Table 1. Triangle quality for MC and suggested variants. Results are typical of all datasets tested (full set of results with all 30 datasets available online).

Table 1 also shows that the removal of the edge group 2 from MC table results in an improvement of the maximum internal angle (θ_∞) quality measure varying from 25°

¹ Full results are available online at <http://XXX>

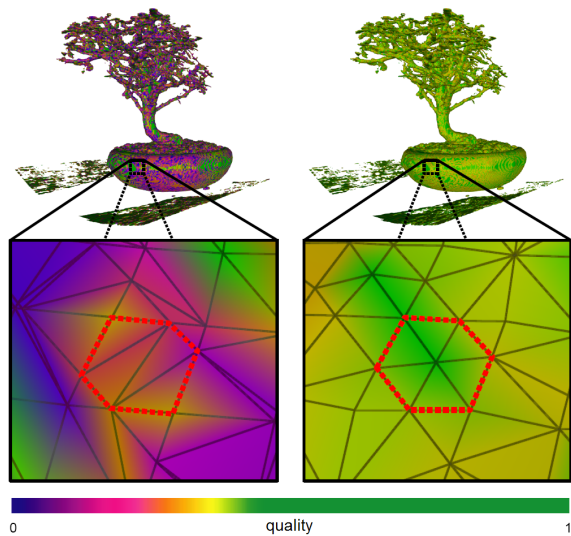


Figure 13. The MC mesh (left) shows many bad-shaped triangles generated from Edge Group 2, as the one highlighted in the zoomed image, while the new MC table using the Unified Macet algorithm results in an optimal mesh.

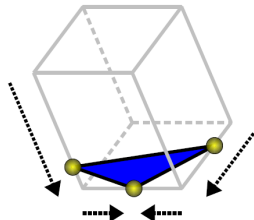


Figure 14. Flattened triangles (triangles with an internal angle close to 180°) which can be only generated by Edge Group 2.

in *Cross* dataset to 47° in *Neghip* dataset, even in the original MC algorithm. Edge group 2 is the *only* group that can generate arbitrarily obtuse triangles, as illustrated in Figure 14. With this case removed, the largest angle in MC is bounded above by 118.6 in all cases we tested.

6. Related Work

Our proposal for improving the quality of the triangulations of MC is simple and effective, and it is one of many proposals in the area. Gibson [5] (with improvements by Bruin et al. [2]) propose a method based on MC that places sampling points at the center of each active cell (a cell

crossed by the isosurface), and connects them to sampling points in adjacent cells. These generate meshes that are in a sense *dual* to the traditional MC triangulation. Nielson specifically proposed the Dual MC algorithm [11]. Our insertion of an extra vertex in MC cases where edge group 2 cannot be completely removed can be seen as an application of these dual techniques.

Our proposal for an improved MC involves directly changing the polygonization process. A similar idea also motivated Tzeng [16], and Labelle and Shewchuk [7] not only improve tetrahedral mesh quality by warping the grid in which the boundary extraction happens, but they also use a BCC lattice instead of the traditional cubic.

Finally, Raman and Wenger [13] propose a slightly different approach: instead of warping the computational grid, they directly perturb the scalar field, and explicitly treat the cases where the isosurface touches vertices of the grid. The modified MC table is much larger (3^8 entries before coalescing symmetric cases, instead of 2^8 in the regular MC algorithm), and the authors recommend a computer-based table construction. In addition, their method tends to change the topology of the resulting mesh, and generates non-manifold surface meshes. Still, the method is conceptually very simple and amenable to parallelization.

Ju [6] discusses ways to modify the triangulation encoded in the MC tables. Instead of a static table, the proposed algorithm uses decision trees to identify the triangulation to choose in a such way that forms convex contours. Such an approach might be usable for choosing the best possible triangulation based on the actual configuration of a particular cell.

Acknowledgments

Extra material An online supplement to this paper contains the full source code for the improved MC algorithm as suggested by this paper together with the improved case table. It is available at <http://XXX>.

References

- [1] P. Alliez, G. Ucelli, C. Gotsman, and M. Attene. Recent advances in remeshing of surfaces. In *State-of-the-art report of the AIM@SHAPE EU network*. Springer, 2005.
- [2] P. W. de Bruin, F. Vos, F. H. Post, S. F. F. Gibson, and A. M. Vossepoel. Improving triangle mesh quality with surfacetets. In *MICCAI 2000*, pages 804–813, 2000.
- [3] C. Dietrich, C. Scheidegger, J. L. D. Comba, L. P. Nedel, and C. T. Silva. Edge groups: An approach to understanding the mesh quality of marching methods. *14(6)*, IEEE Transactions on Visualization and Computer Graphics:1651–1658, 2008.
- [4] C. Dietrich, C. Scheidegger, J. Schreiner, J. L. D. Comba, L. P. Nedel, and C. T. Silva. Edge transformations for im-

- proving mesh quality of marching cubes. *IEEE Transactions On Visualization and Computer Graphics*, 15(1):150–159, 2009.
- [5] S. F. F. Gibson. Constrained elastic surface nets: Generating smooth surfaces from binary segmented data. In *MICCAI '98: Proceedings of the First International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 888–898, London, UK, 1998. Springer-Verlag.
 - [6] T. Ju, S. Schaefer, and J. Warren. Convex contouring of volumetric data. *The Visual Computer*, 19(7-8), 2003.
 - [7] F. Labelle and J. R. Shewchuk. Isosurface stuffing: fast tetrahedral meshes with good dihedral angles. *ACM Transactions on Graphics*, 26(3):57:1–57:10, 2007.
 - [8] T. Lewiner, H. Lopes, A. W. Vieira, and G. Tavares. Efficient implementation of marching cubes' cases with topological guarantees. *Journal of Graphics Tools*, 8(2):1–15, 2003.
 - [9] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *SIGGRAPH '87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pages 163–169, New York, NY, USA, 1987. ACM Press.
 - [10] C. Montani, R. Scateni, and R. Scopigno. Discretized marching cubes. In *VIS '94: Proceedings of the conference on Visualization '94*, pages 281–287, Los Alamitos, CA, USA, 1994. IEEE Computer Society Press.
 - [11] G. M. Nielson. Dual marching cubes. In *VIS '04: Proceedings of the conference on Visualization '04*, pages 489–496, Washington, DC, USA, 2004. IEEE Computer Society.
 - [12] P. P. Pebay and T. J. Baker. A comparison of triangle quality measures. In *10th International Meshing Roundtable*, pages 327–340, 2001.
 - [13] S. Raman and R. Wenger. Quality isosurface mesh generation using an extended marching cubes lookup table. *Computer Graphics Forum*, 27(3):791–798, 2008.
 - [14] J. Schreiner, C. Scheidegger, and C. Silva. High-quality extraction of isosurfaces from regular and irregular grids. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):1205–1212, 2006.
 - [15] J. R. Shewchuk. What is a good linear element? - interpolation, conditioning, and quality measures. In *11th International Meshing Roundtable*, pages 115–126, 2002.
 - [16] L. Tzeng. Warping cubes: Better triangles from marching cubes. In *European Workshop on Computational Geometry*, 2004.