Editors: Cláudio T. Silva, csilva@cs.utah.edu
Joel E. Tohline, tohline@rouge.phys.lsu.edu

MARCHING CUBES WITHOUT SKINNY TRIANGLES

By Carlos A. Dietrich, Carlos E. Scheidegger, João L.D. Comba, Luciana P. Nedel, and Cláudio T. Silva

Most computational codes that use irregular grids depend on the worst triangle's quality. Marching cubes (MC) is the standard isosurface grid generation algorithm, and, whereas most triangles it generates are good, it almost always generates bad triangles. Here, we show how simple changes to MC can lead to a drastically reduced number of degenerate triangles, making it a more practical choice for isosurface grid generation.

Marching cubes[1] (MC) is currently the most popular algorithm for isosurface extraction. It's elegant, simple, fast, and robust. Although the output mesh that MC generates is adequate for visualization purposes, it's far from suitable for use in numerical simulations. This deficiency arises from the degenerate triangles that MC typically generates—a single badly shaped triangle can lead to the ill conditioning of an entire finite element simulation.[2] The current practice is to solve this problem by postprocessing,[3,4] but here we present a simpler alternative. We first elucidate the causes of bad triangles in MC, and then mitigate the problem with small specific changes.

Edge Groups

We base our discussion of MC on the notion of edge groups, recently introduced in another study.[5] Each MC case generates up to five 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, which limits the number of ways a triangle is generated. We then identify equivalent triples of edges under the cube's symmetries, and arrive at eight different edge groups, illustrated in Figure 1.

Surprisingly, a single-edge group produces most degenerate triangles in MC. Some cases in the MC table admit different triangulations, which use different edge groups. By systematically analyzing each case in the MC table, we build on our previous work to generate a table that leads to improved triangle quality.[5,6] Here, we focus on the practical aspects of improving MC to generate better-shaped triangles. The new, improved table is available at www.sci.utah.edu/~cscheid/edge_groups, together with supplemental material showing more extensive comparisons and results.

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). MC's implementation follows a straightforward pipeline of actions executed for each cell in a given volume. It starts by computing each cell vertex's sign, determined by simply comparing a given vertex's scalar value with k. The signs of all vertices from a cube define an 8-bit value that identifies a particular case in MC. This value indexes two predefined tables: an active-edge table and a triangulation table (Figure 2).

The active-edge table identifies, for each case, which of the cell's edges the isosurface crosses and therefore which intersections to compute. The triangulation table correspondingly gives the set of triangles that the active edges will generate. A single MC case can generate up to five triangles. Most importantly, the encoding of some cases isn't unique. Any triangulation that has the same topology as the continuous level set that it's approximating is seen as equally good. As we will explain in the next section, the notion of edge groups lets us effectively choose triangulations that generate systematically better triangles.

Analysis of Marching Cubes' Edge Groups

In our approach to improve MC, we use quality information given by the edge groups involved in any particular triangulation. We then pick the one that maximizes some criteria. Here, we mainly use the radii ratio of incircle to circumcircle normalized to lie between zero and one; with zero representing a degenerate triangle and one an equilateral triangle.[7] However, the same idea directly applies to other measures such as minimal and maximal angles, as we show in Figure 3.

Our first analysis of the impact that different edge groups have comes from plotting the probability density function (PDF) of triangle quality for randomly selected triangles from each of the edge groups. In this ini-

 1521-9615/09/$25.00 © 2009 IEEE

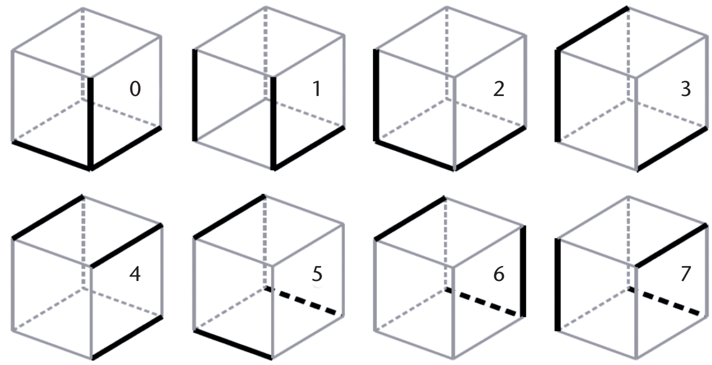

Figure 1. The eight edge groups in marching cubes (MC). Every triangle in every MC configuration is created by one of these edge combinations.

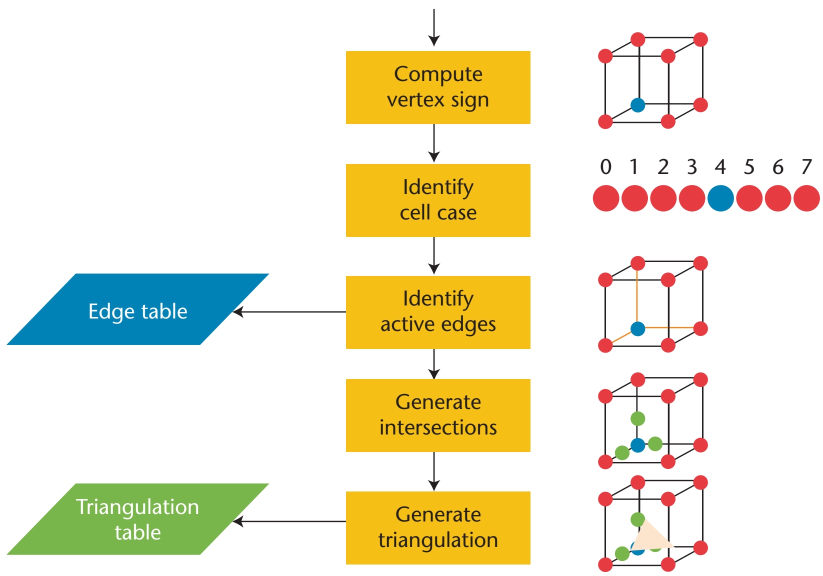

Figure 2. MC pipeline. The active edges encoded in the edge table necessarily cross the isosurface and are illustrated in orange. The triangulation table determines how to connect the vertices that lie on the active edges, which creates the triangles for each patch. Creating the entries of the triangulation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

tial model, the triangle distribution is given by assuming a uniform distribution of triangle vertices along edges and that the vertex choices are independent across edges. This gives a PDF for each edge group (see Figure 3). Clearly, edge group 2 has a qualitatively different behavior than the others: it creates a substantial fraction of degenerate triangles.

Dietrich and his colleagues collected edge group statistics on a collection of 30-volume datasets to test the robustness of the distribution assumptions for each edge group.[5] One experiment shows edge group frequency data over isosurfaces extracted from each of the 30 volumes. The results show, as we would expect, that edge groups are not equally probable. The second set of statistics presents a much clearer picture. By counting the edge groups of the 1,000 worst triangles in each of the 30 extracted isosurfaces, they found that edge group 2 is responsible for, typically, more than 60 percent of the worst 1,000 triangles in any given dataset, and, in some cases, close to 95 percent. Our strategy, then, is to systematically change the MC tables to remove the occurrence of edge group 2.

Improving Marching Cubes

Edge groups motivate a simple criterion for improving the MC table. In another study, Dietrich and his colleagues propose that a retriangulation in certain table entries prevents edge group 2 from occurring.[5] Their proposal focuses on only a few MC cases, namely cases 5, 12, 11, and the complement of case 6.[8]

These changes update 96 entries of the MC table (120 entries if the table is constructed with the complement of case 6) but still leave 56 entries with occurrences of edge group 2.[8] For some MC cases, however, we can't remove edge group 2 by simply retriangulating the case: every triangulation of these cases includes an instance of edge group 2 (see the left column of Figure 4).

Inserting a New Vertex in the Cell

As we've discussed, retriangulating the intersection's vertices can't 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's vertices of active edges, we remove edge groups entirely from the MC table. We illustrate the resulting triangulations in Figure 4. A similar approach was used in contexts as diverse as dual MC meshes[9] and MC mesh simplification,[10] but here we emphasize its impact in connection to MC mesh quality. Additionally, im-

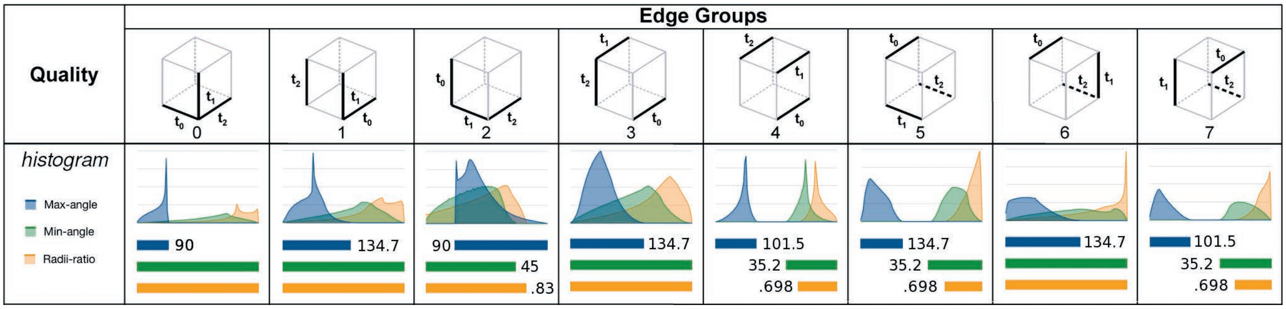

Figure 3. Edge groups and the quality of their corresponding triangles. Histograms for radii ratio, minimum angle, and maximum angle show how triangle quality varies in each edge group. Edge group 2's radii-ratio histogram shows that this group produces more degenerate triangles than any other edge group.

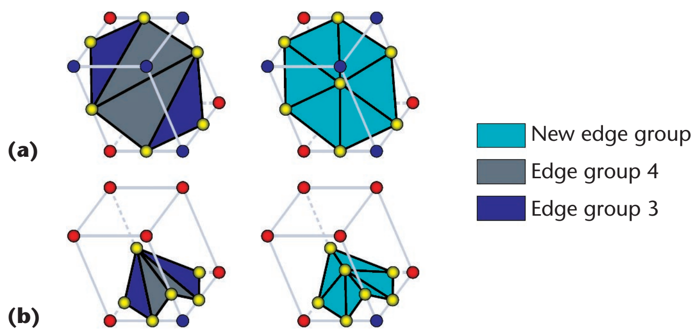

Figure 4. New triangulation. Placing a new vertex in the middle of the cell allows the removal of edge group 2 in MC case 9 (a) and the complement of case 3 (b) by retriangulating the intersection's vertices with the help of an additional vertex.

plementing this change requires only small changes to the MC code.

To understand how adding an extra vertex can improve triangle quality, look at the new configuration using the cell center as an additional edge group with only two edges. This single group generates all triangles shown in the right column of Figure 4. More importantly, its quality histogram is comparable to the best edge groups of the cubic cell.

The new vertex's position in the cell depends on the MC case. The cell's center can be a good choice for MC case 9 (see Figure 4a). 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 cell's negative vertices (blue vertices in Figure 4), in which the new vertex's distance to the isosurface is maximal. To alleviate this problem, the new vertex is placed along one of the original MC triangulation's edges—that is, in the middle of the longest edge of the triangulation. This guarantees that the new triangulation is close to the original triangulation that the MC generates.

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, discussed in the next section. Together, these two changes are such that the triangles that the suggested MC generates compare favorably to the state of the art.

Transforming Active Edges

The second change to MC is based on Macet algorithm[6] and consists of perturbing the active edges on which intersection vertices are computed. In this work, they propose to move (by a small amount) the two edge endpoints inside the volume, and then the computation of the edge vertex proceeds as normal. Macet adds two new intermediate steps to the MC pipeline. The edge transformation step alters the positions of each edge extreme along the gradient or tangent directions. The second step, when necessary, displaces the intersection points away from edge extreme. Together, these steps tend to create active edges that are locally perpendicular to the isosurface, which leads to improved triangle quality. To enforce valid placement of edge endpoints (that is, not crossing the isosurface), Macet performs edge transformations in several steps with smaller displacements along the proposed direction (in our experiments, we used eight steps).

As described, the Macet proposal's drawback is that it doesn't have a criterion for choosing which edge transformation to use. Instead, it performs both transformations and does a neighborhood analysis that chooses the transformation that leads to local improved triangle quality. Whereas the local analysis is fast, the cost of using both transformations still leaves room for improvement.

In another study,[5] we gave a different

Table 1. Triangle quality for MC and suggested variants.*						
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

**Results are typical of all datasets tested—a full set of results with all 30 datasets available online at www.sci.utah.edu/~cscheid/edge_groups.*

interpretation for edge transformation that makes room for edge transformation unification. We formulate the edge transformation as a projection operation of the edge midpoint onto the plane tangent of the isosurface. The same result can be accomplished by using a new approach with unified edge transformations. To accomplish this, we identify the edge extreme closest to the isosurface—this one will be subject to interleaved edge transformations using gradient and tangential transformations (eight in total, four 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. We move the other extreme to the edge's midpoint, which is what the projection operation advocates, under ideal circumstances.

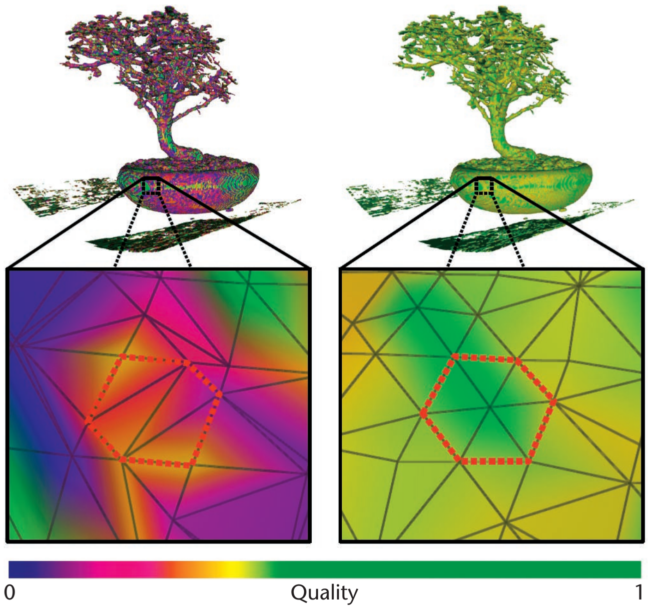

Figure 5. Triangulation results. The MC mesh (left) shows many badly shaped triangles generated from edge group 2, as the one highlighted in the zoomed image, whereas the new MC table using the unified Macet algorithm results in an optimal mesh.

We evaluated the impact of the new MC table and unified Macet with experiments using a collection of 23 datasets. We summarize the results in Table 1 (full results are available online at www.sci.utah.edu/~cscheid/edge_groups). We compare results using two methods: the original MC and the unified Macet with the extended edge table. For each case, we report minimal 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, with the worst radii ratio being 0:43. An intuition of the impact of the unified Macet's changes (see Figure 5) shows a zoomed version of a portion of the Bonsai dataset.

Table 1 also shows that edge group 2's removal from the MC table results in an improvement of the maximum internal angle (θ_∞) quality measure varying from 25 degrees in cross dataset to 47 degrees in *Neghip* dataset, even in the original MC algorithm. Edge group 2 is the only group that can generate arbitrarily obtuse triangles. Removing this case, the largest angle in MC is bound by 118.6 in all cases we tested.

As future work, we intend to continue our analysis of edge groups and use our findings to design more efficient triangulation algorithms. We also want to inspect how the proposed techniques work for adaptive subdivisions (that is, where you have cells of different resolutions).

Acknowledgments

The work of Carlos Dietrich is supported by a CNPq scholarship. Carlos Scheidegger is supported by an IBM PhD Student Fellowship. João Comba is supported by CNPq grant 485853/2007-0. This research has also been funded by the US Department of Energy, the Na-

Related Work

Our proposal for improving the MC triangulation quality is simple and effective, and represents one of many proposals in the area. Gibson[1] (with improvements by other researchers[2]) proposes 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 proposes the dual MC algorithm.[3] Our insertion of an extra vertex in MC cases where we can't completely remove edge group 2 is an application of these dual techniques.

Our proposal for an improved MC involves directly changing the polygonization process. A similar idea also motivated Tzeng,[4] and Labelle and Shewchuk[5] not only to improve tetrahedral mesh quality by warping the grid in which the boundary extraction happens, but also to use a body-centered cubic lattice instead of the traditional cubic one.

Finally, Raman and Wenger propose a slightly different approach:[6] instead of warping the computational grid, they perturb the scalar field directly and explicitly treat the cases where the isosurface touches the grid's vertices. The modified MC table is much larger (38 entries before coalescing symmetric cases, instead of 28 in the regular MC algorithm), and the authors recommend a computer-based table construction. Additionally, their method tends to change the resulting mesh's topology and generates nonmanifold surface meshes. Still, the method is conceptually very simple and amenable to parallelization.

Ju discusses ways to modify the triangulation encoded in the MC tables.[7] Instead of a static table, the proposed algorithm uses decision trees to identify the triangulation to choose in such a way that forms convex contours. This approach can be used to choose the best possible triangulation based on a particular cell's actual configuration.

References

1. S.F.F. Gibson, "Constrained Elastic Surface Nets: Generating Smooth Surfaces from Binary Segmented Data," *Proc. 1st Medical Image Computing and Computer-Assisted Intervention* (MICCAI 98), Springer-Verlag, 1998, pp. 888–898.
2. P.W. de Bruin et al., "Improving Triangle Mesh Quality with Surfacenets," *Proc. 3rd Medical Image Computing and Computer-Assisted Intervention* (MICCAI 00), Springer-Verlag, pp. 804–813.
3. G.M. Nielson, "Dual Marching Cubes," *Proc. Conf. Visualization* (VIS 04), IEEE CS Press, 2004, pp. 489–496.
4. L. Tzeng, "Warping Cubes: Better Triangles from Marching Cubes," *European Workshop on Computational Geometry*, 2004.
5. F. Labelle and J.R. Shewchuk, "Isosurface Stuffing: Fast Tetrahedral Meshes with Good Dihedral Angles," *ACM Trans. Graphics*, ACM, vol. 26, no. 3, 2007, pp. 57:1–57:10.
6. S. Raman and R. Wenger, "Quality Isosurface Mesh Generation Using an Extended Marching Cubes Lookup Table," *Computer Graphics Forum*, vol. 27, no. 3, 2008, pp. 791–798.
7. T. Ju, S. Schaefer, and J. Warren, "Convex Contouring of Volumetric Data," *The Visual Computer*, vol. 19, nos. 7–8, 2003, pp. 513–525.

tional Science Foundation, and IBM Faculty Awards (2005, 2006, and 2007). An online supplement to this article (www.sci.utah.edu/~cscheid/edge_groups) contains the full source code for the improved MC algorithm as suggested by this article together with the improved case table.

References

1. W.E. Lorensen and H.E. Cline, "Marching Cubes: A High Resolution 3D Surface Construction Algorithm, *Proc. 14th Ann. Conf. Computer Graphics and Interactive Techniques* (SIGGRAPH 87), ACM Press, 1987, pp. 163–169.
2. P.P. Pebay and T.J. Baker, "A Comparison of Triangle Quality Measures," *Proc. 10th Int'l Meshing Roundtable*, Sandia National Laboratories, 2001, pp. 327–340.
3. P. Alliez et al., *Recent Advances in Remeshing of Surfaces*, Springer-Verlag, 2008.
4. J. Schreiner, C. Scheidegger, and C. Silva, "High-Quality Extraction of Isosurfaces from Regular and Irregular Grids," *IEEE Trans. Visualization and Computer Graphics*, vol. 12, no. 5, IEEE CS Press, 2006, pp. 1205–1212.
5. C. Dietrich et al., "Edge Groups: An Approach to Understanding the Mesh Quality of Marching Methods," *IEEE CS Trans. Visualization and Computer Graphics*, vol. 14, no. 6, IEEE CS Press, 2008, pp. 1651–1658.
6. C. Dietrich et al., "Edge Transformations for Improving Mesh Quality of Marching Cubes," *IEEE CS Trans. Visualization and Computer Graphics*, IEEE CS Press, vol. 15, no. 1, 2009, pp. 150–159.
7. J.R. Shewchuk, "What is a Good Linear Element? Interpolation, Conditioning, and Quality Measures," *Proc. 11th Int'l Meshing Roundtable*, 2002, pp. 115–126.
8. T. Lewiner et al., "Efficient Implementation of Marching Cubes Cases with Topological Guarantees," *J. Graphics Tools*, vol. 8, no. 2, 2003, pp. 1–15.
9. G.M. Nielson, "Dual Marching Cubes," *Proc. Conf. Visualization* (VIS 04), IEEE CS Press, 2004, pp. 489–496.
10. C. Montani, R. Scateni, and R. Scopigno, "Discretized Marching Cubes," *Proc. Conf. Visualization* (VIS 94), IEEE CS Press, 1994, pages 281–287.

Carlos A. Dietrich is a software developer at Dell. His research interests include image processing, computer graphics, and medical applications. Dietrich has a PhD in computer science from Federal University of Rio Grande do Sul (UFRGS). Contact him at cadietrich@inf.ufrgs.br.

Carlos E. Scheidegger is a research assistant and PhD candidate at the University of Utah. His research interests include scientific visualization, geometry processing, and computer graphics. Scheidegger has a BS in computer science from UFRGS. Contact him at cscheid@sci.utah.edu.

João L.D. Comba is an associate professor at Instituto de Informática, UFRGS. His research interests include scientific visualization, geometric algorithms, and graphics hardware. Comba has a PhD in computer science from Stanford University. Contact him at comba@inf.ufrgs.br.

Luciana P. Nedel is an associate professor at Instituto de Informática, UFRGS. Her research interests include nonconventional interaction, scientific visualization, and animation. Nedel has a PhD in computer science from Ecole Polytechnique Fédérale de Lausanne. Contact her at nedel@inf.ufrgs.br.

Cláudio T. Silva is an associate professor at the University of Utah. His research interests include visualization, geometry processing, graphics, and high-peformance computing. Silva has a PhD in computer science from Stony Brook University. He is a member of IEEE, the ACM, Eurographics, and Sociedade Brasileira de Matematica. Contact him at csilva@cs.utah.edu.