Department: Visual Computing: Origins
Editors: Chris Johnson, crj@sci.utah.edu
Dave Kasik, dave.kasik@gmail.com
Mary C. Whitton, mcwhitton@gmail.com

# History of the Marching Cubes Algorithm

**William E. Lorensen**

*Editor's note:*
The Marching Cubes paper by Bill Lorensen and Harvey Cline, "Marching Cubes: A High Resolution 3D Surface Construction Algorithm," was published at SIGGRAPH 1987.[1] According to Google Scholar, their paper has 15,667 citations (as of January 17, 2020), the most highly cited paper in computer graphics. Sadly, while writing this article Bill Lorensen passed away on December 12, 2019. Origins Department Editor Chris Johnson contributed the text in italics.

## EARLY HISTORY OF MARCHING CUBES

**In 1968, I** graduated from Rensselaer Polytechnic Institute (RPI) with a B.S. in Mathematics. I intended to get a Ph.D. in Math and teach Mathematics. The Vietnam War changed that. I tried to find an employer who could get a job-related draft deferment. Both IBM and GE Corporate Research (CRD) gave me reasonable offers but could not get me a deferment. They did promise to hold my job if I was drafted. I had a low lottery number (108) and was sure to be drafted. The U.S. Army Maggs Research Center could get the deferment, so I took that job as a Scientific Programmer, even though the salary was less than the IBM and GE offers.

The threat of being drafted changed my career and directed me to the nascent field of computer graphics. At my Army job, I had access to an EAI analog plotter. An IBM Reproducing Punch Machine drove the plotter with a specially wired board that interpreted $x$, $y$, pen code coordinates, one coordinate per punched card. I wrote programs to plot curves and 2-D models, punching one card per coordinate. The analog

*Digital Object Identifier 10.1109/MCG.2020.2971284*
*Date of current version 28 February 2020.*

plotter tended to draw curved lines for long straight lines. Soon I was able to purchase a CalComp Incremental Plotter. The CalComp drew lines in small increments in one of eight directions. I was able to draw more complex representations. Finally, I purchased a Lundy Vector Refresh System, and my work moved to three dimensions. I wrote my first 2-D contouring program for the Lundy. The algorithm started with a seed point and tracked that point as it moved through the 2-D scalar field.

My work for the Army involved finite element analysis pre- and postprocessing, developing grid generators and contouring algorithms. Graphics was just a means to an end. The finite element and graphics expertise landed me a job at GE Corporate Research and Development in Niskayuna, NY, USA. GE hired me to work in the computer service group that provided programming to GE scientists. I hit the road running with the Solid Mechanics Group and became an active contributor to their finite element research.

In 1984, Carl Crawford was working in GE's Medical Systems Business Group in Milwaukee. Carl worked in the Applied Sciences Lab and was a recognized expert in Computed Tomography (CT) reconstruction. Carl was also part of a company task force that was looking for applications for an exciting new GE product, the Graphicon. The GE Graphicon was a high-performance rendering engine created at GE's Flight Simulator Group. Each GE division was asked to look for applications of this technology. Carl gave a seminar in Building 37, at the downtown Schenectady GE manufacturing facility. Several of the research labs that did advance engineering work were in Building 37.

The main research campus where I worked was about 4 miles away.

Carl described the capabilities of the (not yet built) Graphicon. He described his vision for using this polygon engine for 3-D medical surface display. The current GE 3-D medical product was based on cuberilles and was optimized for implementation in the limited-memory Data General computers (only 32k memory). GE licensed the cuberille technology from the University of Pennsylvania. Carl challenged the seminar attendees to think about how they might replace the cuberille technology with polygon-based technology.

Harvey Cline and I attended the seminar. Harvey was in the Electronic Materials Lab, and I was in Computer Systems and Services, part of the central computer service group at CRD. Harv and I had been doing some work on reconstructing 3-D surfaces from interferograms. The application was the reconstruction of microscopic surfaces of electronic materials. The final step of our reconstruction algorithm used Movie.BYU's Mosaic program to generate triangles between adjacent contours. We had been looking for medical applications and saw Carl's challenge as a way to get our foot into GE Medical Systems. Carl's talk was in the morning, and Harv and I returned to my office to brainstorm. Rubick's Cube was the rage at the time, and Harv was analyzing the Cube problem using the symmetry of cubical lattices. The accepted way of generating surfaces from volume data was to create 2-D contours on each slice and then try to connect the contours with triangles. This is exactly what Mosaic did. Whenever Harv and I got together, it was an explosion of ideas. It is difficult to say who originated or how we came up with the idea. But somehow, we determined that solving the problem one cube at a time would remove a lot of the complexity. The inside/outside concept of labeling of vertices may have come from the Movie.BYU Mosaic program. We quickly moved to the notion of solving the volume-to-surface problem one cube at a time. Harv and I started to solve the problem for each of the 256 cases. After just a few, we realized the effort was fruitless. But, using cube symmetry operations, we reduced the number of unique cases to 14. Harv and I worked out the triangulations for the basic cases (see Figure 1). I started to write code to permute the base cases into the 256 cases.

Then, I began coding the algorithm. The algorithm was straightforward to implement. I had something running the next day. The first implementation read volumes in as ASCII files. Harv put together a simple volume of a few slices, so I had some data to test the code. I used the Movie.BYU display program to produce hidden line and shaded surface models. Movie.BYU could only render up to 8192 polygons. The

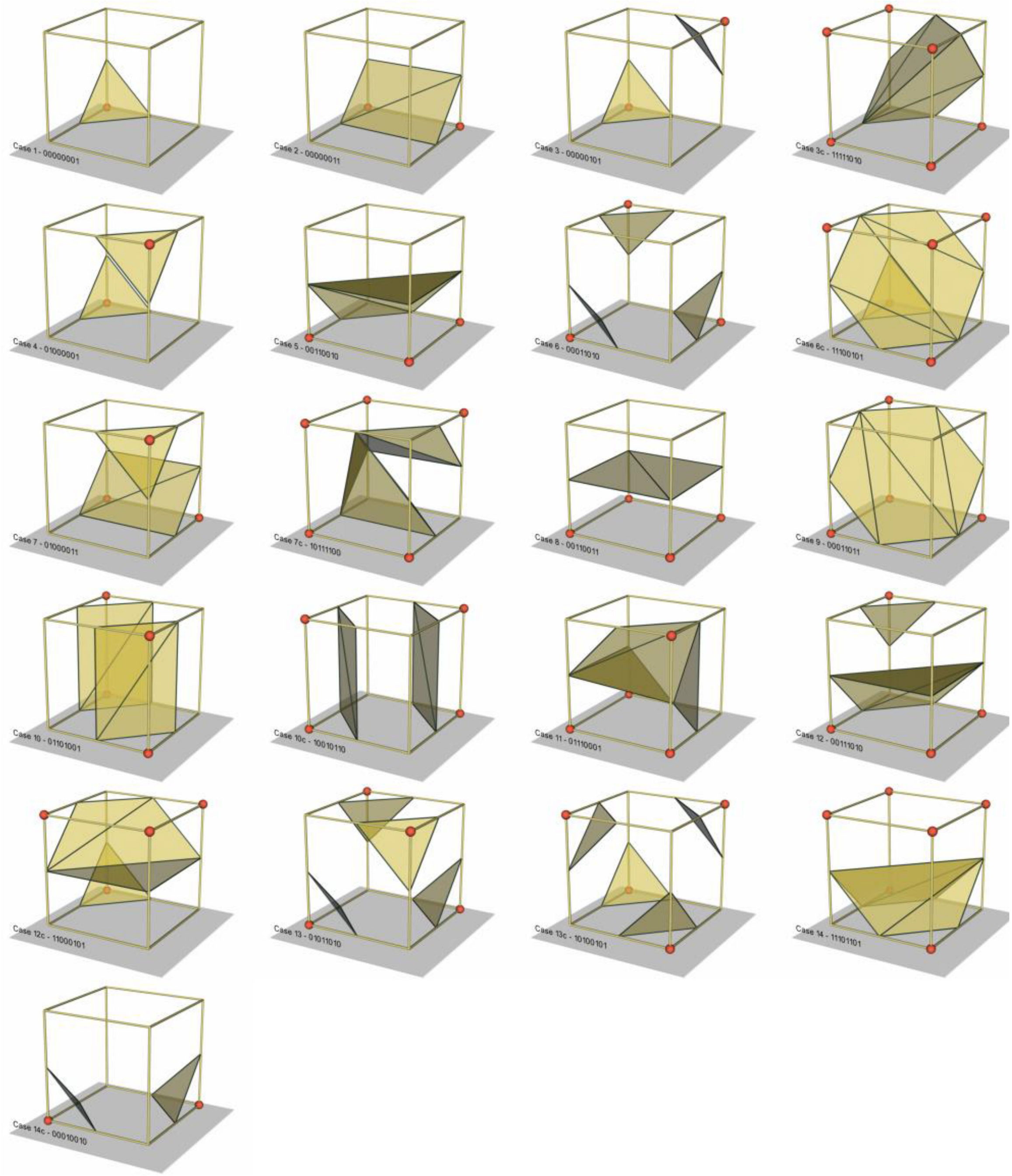


**Figure 1.** Marching Cubes case table.

original Marching Cubes implementation created polygons, although I later switched to generating only triangles. Once the code was debugged, we tried the code on some medical data. We obtained a small CT dataset that included a spine. We extracted a small region of interest that isolated the spine. Since we were restricted to rendering 8192 triangles, the data had to be reasonably small. We were excited about the results.

## POST SIGGRAPH 1987

*Soon after the paper was published, it spawned new research in a number of important areas in computer graphics and visualization* *(e.g., computer-aided geometric design,[6] polygon simplification,[9,27] and volume rendering[7]). Because generation of surfaces and structures from images is important to many medical and biological applications,[5,8] Marching Cubes has been used in a wide array of applications including medical diagnosis, surgical planning, biomolecular graphics, and the creation of biomedical models for functional simulation. Similarly, understanding isosurfaces of three-dimensional scalar fields is essential for computational science and engineering applications such as computational fluid dynamics, weather simulation, computational mechanics, and computational geosciences. Marching Cubes is also used in computer games and virtual reality and most recently in a deep neural network based method, DeepOrganNet, to generate and visualize high-fidelity 3-D / 4-D organ geometric models from a single-view medical image in real time.[20]*

*In 1998, Marching Cubes was recognized as one of SIGGRAPH's seminal papers in Computer Graphics (Figure 3).[3]*

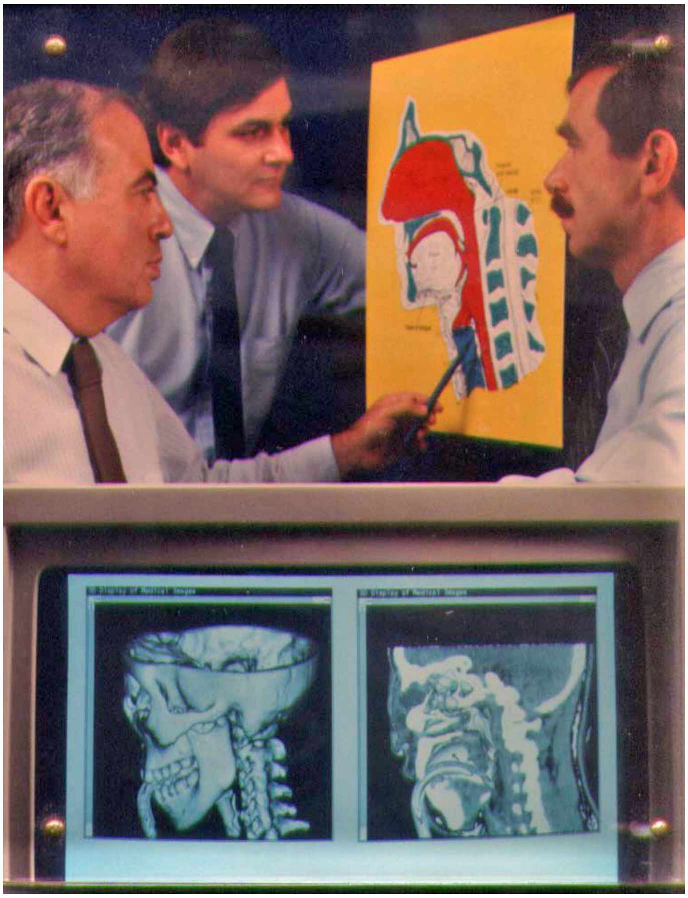

**Figure 2.** Early visualization using Marching Cubes. Harvey Cline (Left), Bill Lorensen (Middle), Sieg Ludke (Right), November 1988.

## DECIMATION OF TRIANGLE MESHES AND MARCHING CUBES

Jon Zarge was a member of the Software Technology Program (STP) at CRD. The STP recruited high potential candidates with Bachelor's degrees and offered them an opportunity to get a Master's degree at a local university (usually Rensselaer Polytechnic Institute). While they worked on their degrees at Company expense, they rotated through two or three projects within CRD. Jon was assigned to our group to work on a new medical imaging modality, biomagnetism. One of the challenging problems of this project was to solve an inverse problem. From measurements obtained on the scalp of the patient, the inverse problem was to find the source of the biomagnetic field. Most researchers used idealized, spherical models of the skull and brain. The Biomag team set out to generate patient specific models. Their first choice for model generation was, of course, Marching Cubes. Marching Cubes posed two problems. First, the models generated by Marching Cubes were huge in polygon count. Second, as we found out, the models had topology problems. Jon's task was to generate models that could be analyzed. As I recall, we did not actually have the code to do the analysis.

Jon was being mentored by Will Schroeder (see Figure 6). Will laid out a high-level approach to solve the problem. The approach eliminated single vertices according to some out of plane criterion. Then, the neighborhood of the removed vertex was retriangulated. Will had a strong background in polygonal model topology. Step one of the process was to determine and encode the topology of the triangular surface. Jon wrote code to build a topological data structure to enable the vertex remove, hole-fill approach. This is when Jon brought the severity of the Marching Cubes topology problem* to my attention. This

*MC can produce a topologically inconsistent isosurface that contains holes caused by facetization ambiguity. Ambiguity analysis and disambiguation methods have been considered extensively.[19]

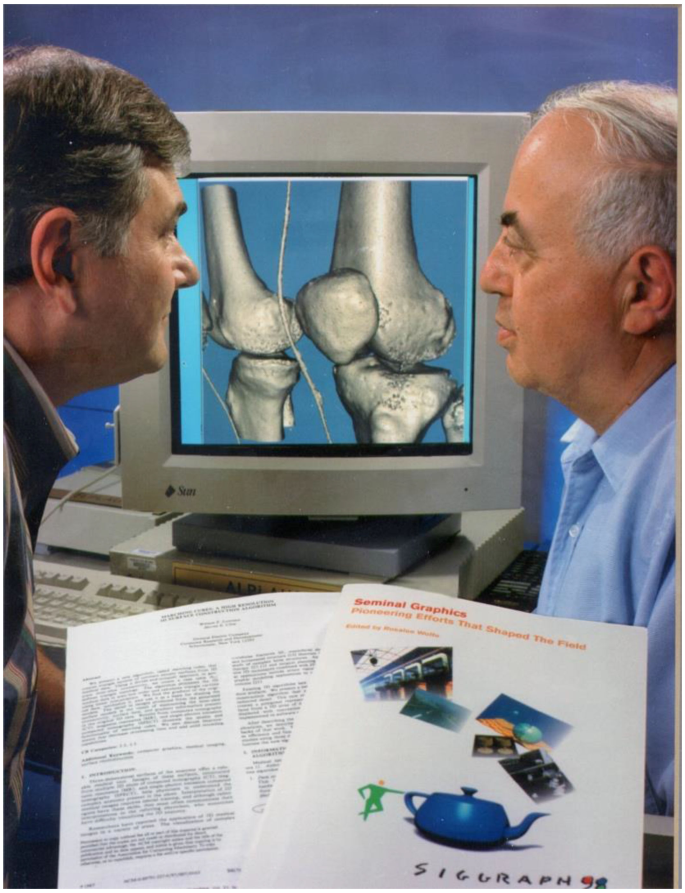


**Figure 3.** Bill Lorensen and Harvey Cline on the occasion of the selection of Marching Cubes as one of SIGGRAPH's seminal papers in Computer Graphics.[3]

had already been pointed out by Martin Durst's letter to the *ACM SIGGRAPH Quarterly*.[4] To be honest, I had tried to verify the extent of Durst's discovery but did not have the right tools. Jon's topology builder was just what I needed. While he worked on refining the initial decimation implementation, I worked on fixing the hole problem in Marching Cubes. The topology problem was easier to solve than I had anticipated. The root of the problem was my early assumption that all of the complementary cases, cases where inside/outside was reversed, were the same. Once I realized that the complementary cases needed different treatment, I was able to resolve the ambiguity. Fortunately, my early code to generate the cases had separate inputs for the complementary cases. I generated a new case table, tested some generated models against Jon's topology checker and felt confident I had solved the problem.

Jon's implementation was in Lisp. I recall that he called the algorithm *intelligent triangle decimation*. At the time, our group was using LYMB as its development environment and delivery platform. Like many large systems, LYMB had a bit of a learning curve. The Biomag group needed a quick implementation of a decimation algorithm and Jon was not a LYMB expert. We decided to let him work out the details of decimation outside of the LYMB system. Will reimplemented Jon's stand-alone algorithm in LYMB. In the process of defining the object-oriented design, Will was able to generalize the decimation algorithm. Generalization of algorithms is a common theme followed by our group over the years.

I was a bit on the sidelines of the decimation algorithm development, concentrating on generating valid surfaces from Marching Cubes. I realized though, that this was a SIGGRAPH quality algorithm. I joined Will and Jon to start writing a description of the algorithm. I searched the literature for surface reduction techniques. I was surprised to find that there was nothing published on this topic. I also took on the task of generating examples to illustrate the power of the algorithm. Successful SIGGRAPH papers have two major components: innovative algorithms and compelling examples. I chose examples from medical imaging, industrial inspection, and terrain modeling.

SIGGRAPH 1992 was held in Chicago. Although we could not find other published work on polygon decimation, there were three papers accepted that year that dealt with reducing polygon count. Hughes Hoppe from the University of Washington and Greg Turk from Georgia Tech had the other two papers in the session. Will gave the talk to a packed room. I recall the talk was not in the main meeting room. SIGGRAPH had parallel sessions.

The decimation algorithm had a large impact on my career. The SIGRAPH paper[4] is highly cited, but more than that, decimation made Marching Cubes a practical algorithm. It became a cornerstone of our model creation pipeline.

*Because isosurface extraction from 3-D volume data is so pervasive and important, there has been significant research in speeding up Marching Cubes and creating new isosurface*

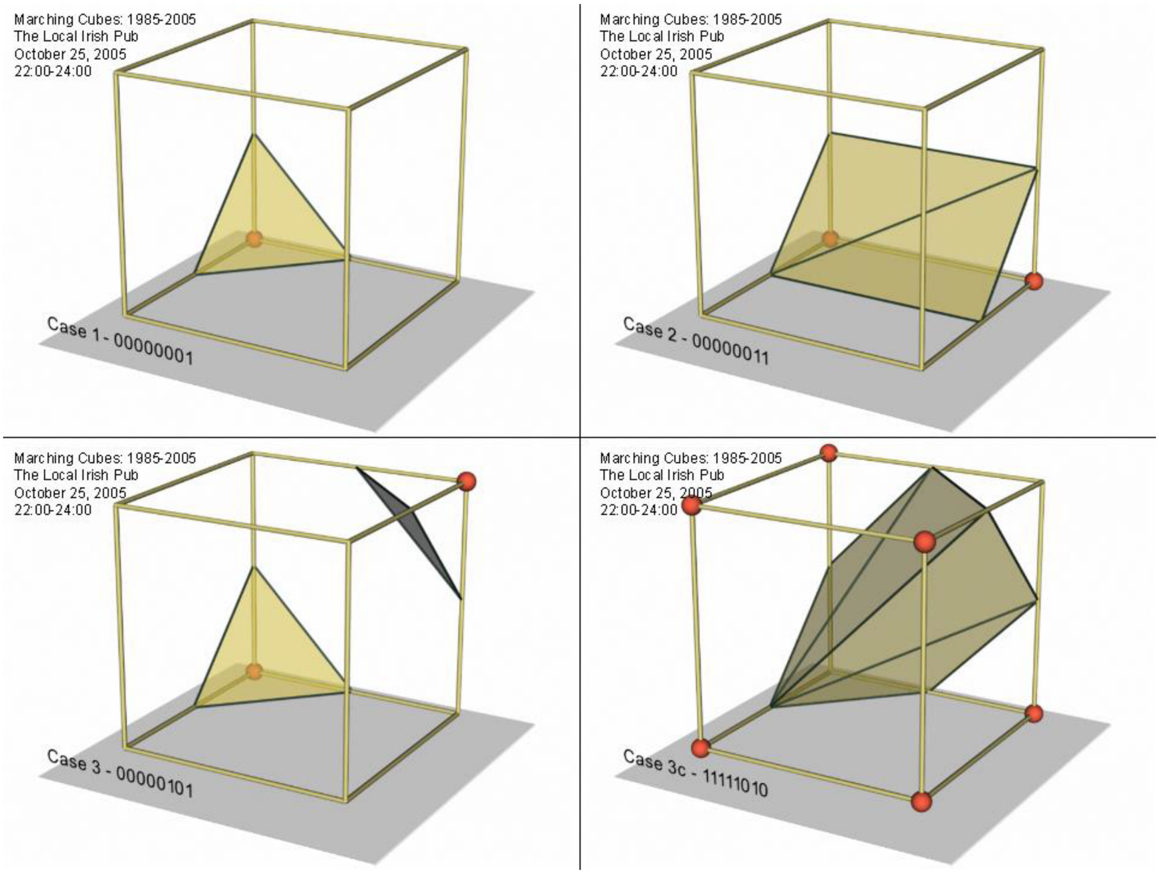


**Figure 4.** Admission tickets for the Marching Cubes Patent Wake held at *The Local Irish Pub*, Minneapolis, MN, USA, on October 25, 2005.

*algorithms.[9–18,27] As a personal aside, in 1994, Han-Wei Shen and I wrote a paper on isosurface extraction for unstructured grids. We spent a significant amount of time trying to come up with a catchy name for the algorithm to mimic Marching Cubes and finally settled on Sweeping Simplices.[12] Needless to say, the name did not catch on. In 2006, Newman and Yi wrote a survey paper on the Marching Cubes algorithm.[20] The paper's aim was to survey the development of the algorithm*

**Figure 5.** Bill Lorensen in 2019.

**Figure 6.** GE Research Visualization Team in 1992. Counterclockwise: Cathy Chalek, Chris Volpe, Will Schroeder, Bill Lorensen, Boris Yamron.

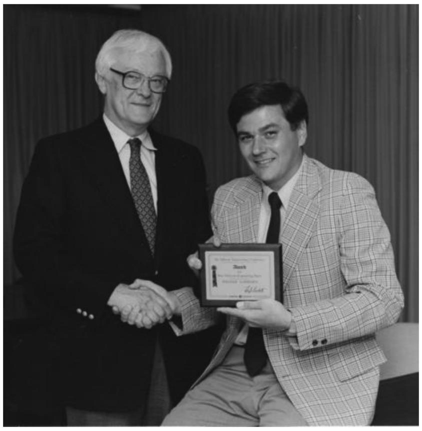

**Figure 7.** Bill Lorensen receiving the Best Software Engineering Paper from the 5th International Conference on Software Engineering in 1984.

*and its computational properties, extensions, and limitations (including the attempts to resolve its limitations). A rich body of publications related to Marching Cubes was included in that paper.*

*One day I received an envelope in the mail from Bill. When preparing to move from New York to California, Bill found the original reviews of the Marching Cubes paper and mailed them to me with a note saying that he thought that I would find the reviews interesting and especially a comment made by reviewer number three who commented that he thought the name Marching Cubes was "misleading and unhelpful." A copy of the original reviews of the Marching Cubes paper is available here.*[2]

*Given Marching Cubes' impact, especially in medical imaging, Bill wondered why GE did not aggressively leverage their patent on the Marching Cubes algorithm, but they did not. In 2005, Bill hosted a party to celebrate the expiration of the GE patent. The tickets to the party were, not surprisingly, from the Marching Cubes case table (see Figure 4).*

*According to Microsoft Academic, between 1987 and 1990, 100 papers cited the Marching Cubes paper. This climbed to 472 citations in the next five years and then to thousands over subsequent five-year intervals. In 2019 alone, there were 337 new citations. Marching Cubes lives on with large-scale parallel implementations via VTK-m*[21] *and Flying Edges,*[18] *and discrete Marching Cubes for segmentation masks. The algorithm is used in a variety of systems for performing visualization, modeling, and meshing such as R,*[22] *Unity,*[23] *edge group analysis,*[24] *and even weather forecasting.*[25] *Chances are many of us have directly or indirectly been impacted by the Marching Cubes algorithm, and it is likely that it will remain a staple of computing for the foreseeable future. Until his death, Bill maintained a Marching Cubes wiki.*[26] *The website is now being maintained by Will Schroeder and includes a tribute section to Bill.*

## ACKNOWLEDGMENTS

The Origins Department Editors would like to thank Will Schroeder and Chuck Hansen for their help with this article.

## REFERENCES

1. W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3D surface construction algorithm," *ACM Comput. Graph.*, vol. 21, no. 4, pp. 163–169, 1987. [Online]. Available: https://dl.acm.org/doi/pdf/10.1145/37402.37422
2. "Reviews of the marching cubes paper," 1987. [Online]. Available: http://www.sci.utah.edu/Marching_Cubes_Reviews_1987.pdf
3. R. J. Wolfe, *Seminal Graphics: Pioneering Efforts That Shaped the Field*. Los Angeles, CA, USA: ACM SIGGRAPH, 1998.
4. M. J. Dürst, "Re: Additional reference to "marching cubes," in *Proc. Int. Conf. Comput. Graph. Interactive Techn.*, vol. 22, no. 5, 1988, Art. no. 243.
5. H. E. Cline, W. E. Lorensen, S. Ludke, C. R. Crawford, and B. C. Teeter, "Two algorithms for the three-dimensional reconstruction of tomograms," *Med. Phys.*, vol. 15, no. 3, pp. 320–27, 1988.
6. J. Bloomenthal, "Polygonization of implicit surfaces," *Comput. Aided Geometric Des.*, vol. 5, no. 4, pp. 341–355, 1988.
7. M. Levoy, "Display of surfaces from volume data," *IEEE Comput. Graph. Appl.*, vol. 8, no. 3, pp. 29–37, May 1988.
8. B. A. Payne and A. W. Toga, "Surface mapping brain function on 3D models," *IEEE Comput. Graph. Appl.*, vol. 10, no. 5, pp. 33–41, Sep. 1990.
9. W. J. Schroeder, J. A. Zarge, and W. E. Lorensen, "Decimation of triangle meshes," in *Proc. 19th Annu. Conf. Comput. Graph. Interactive Techn.*, vol. 26, pp. 65–70, 1992.
10. J. Wilhelms and A. VanGelder, "Octrees for faster isosurface generation," *ACM Trans. Graph.*, vol. 11, no. 3, pp. 201–227, Jul. 1992.

11. T. Itoh and K. Koyyamada, “Isosurface generation by using extrema graphs,” in *Proc. IEEE Vis.*, 1994, pp. 77–83.
12. H. Shen and C.R. Johnson, “Sweeping simplicies: A fast iso-surface extraction algorithm for unstructured grids,” in *Proc. IEEE Vis.*, 1995, pp. 143–150.
13. Y. Livnat, H. W. Shen, and C. R. Johnson, “A near optimal isosurface extraction algorithm using the span space,” *IEEE Trans. Vis. Comput. Graphics*, vol. 2, no. 1, pp. 73–84. Mar 1996.
14. P. Cignoni, C. Montani, E. Puppo, and R. Scopigno, “Optimal isosurface extraction from irregular volume data,” in *Proc. IEEE Symp. Vol. Vis.*, 1996, pp. 31–38.
15. C. L. Bajaj, V. Pascucci, and D. R. Schikore, “The contour spectrum,” in *Proc. IEEE Vis.*, 1997, pp. 167–173.
16. S. G. Parker, P. Shirley, Y. Livnat, C. D. Hansen, and P.-P. Sloan, “Interactive ray tracing for isosurface extraction,” in *Proc. IEEE Vis.*, Oct. 1998, pp. 233–238.
17. Y. Livnat and X. Tricoche, “Interactive point based isosurface extraction,” in *Proc. IEEE Vis.*, 2004, pp. 457–464.
18. W. Schroeder, R. Maynard, and B. Geveci, “Flying Edges: A high-performance scalable isocontouring algorithm,” in *IEEE Symp. Large Data Anal. Vis.*, 2015, pp. 33–40.
19. T. S. Newman and Hong Yi, “A survey of the marching cubes algorithm,” *Comput. Graph.*, vol. 30, no. 5, pp. 854–879, 2006.
20. Y. Wang *et al.*, “DeepOrganNet: On-the-Fly reconstruction and visualization of 3D / 4D lung models from single-view projections by deep deformation network,” *IEEE Trans. Vis. Comput. Graphics*, vol. 26, no. 1, pp. 960–970, Jan. 2020.
21. K. Moreland *et al.*, “VTK-m: Accelerating the visualization toolkit for massively threaded architectures,” *IEEE Comput. Graph. Appl.*, vol. 36, no. 3, pp. 48–58, May/Jun. 2016.
22. D. Feng and L. Tierney, “Computing and displaying isosurfaces in R,” *J. Statist. Softw.*, vol. 28, 2008, pp. 1–24, doi: 10.18637/jss.v028.i01.
23. Unity, “High speed CPU-based marching cubes,” 2020. [Online]. Available: https://assetstore.unity.com/packages/tools/modeling/high-speed-cpu-based-marching-cubes-117483
24. C. A. Dietrich, C. A. C. Scheidegger, J. L. D. Comba, L. Nedel, and C. T. Silva, “Edge groups: An approach to understanding the mesh quality of marching methods,” *IEEE Trans. Vis. Comput. Graphics*, vol. 14, no. 6, pp. 1651–66, Nov./Dec. 2008.
25. H. Skotnes, G. Hartvigsen, and D. Johansen, “3D visualization of weather forecasts and topography,” [Online]. Available: https://munin.uit.no/handle/10037/401, 1994.
26. B. Lorensen, “Marching cubes Wiki,” 2020. [Online]. Available: marchingcubes.org
27. J. Wilhelms and A. Van Gelder, “Octrees for faster isosurface generation,” *Comput. Graph.*, vol. 24, no. 5, pp. 57–62, Nov. 1990.


Contact department editors Chris Johnson at crj@sci.utah.edu, Dave Kasik at dave.kasik@gmail.com, or Mary C. Whitton at mcwhitton@gmail.com.