Marching cubes

From Wikipedia, the free encyclopedia

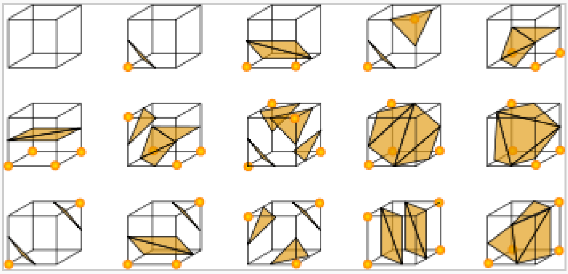
Head and cerebral structures (hidden) extracted from 150 MRI slices using marching-cubes (about 150,000 triangles)

Marching cubes is a computer graphics algorithm, published in the 1987 SIGGRAPH proceedings by Lorensen and Cline,[1] for extracting a polygonal mesh of an isosurface from a three-dimensional discrete scalar field (sometimes called voxels). This paper is one of the most cited papers in the computer graphics field.[citation needed] The applications of this algorithm are mainly concerned with medical visualizations such as CT and MRI scan data images, and special effects or 3-D modelling with what is usually called metaballs or other metasurfaces. An analogous two-dimensional method is called the marching squares algorithm.

Contents [hide]

History [edit]

The algorithm was developed by William E. Lorensen and Harvey E. Cline as a result of their research for General Electric. At General Electric they worked on a way to efficiently visualize data from CT and MRI devices.[citation needed]



The originally published 15 cube configurations

Their first published version exploited rotational and reflective symmetry and also sign changes to build the table with 15 unique cases. However, in meshing the faces there are possibly ambiguous cases.[2] These ambiguous cases can lead to meshings with holes. Topologically correct isosurfaces can still be constructed with extra effort.[3]

The problem was that for cases with "rippling" signs, there are at least two correct choices for where the correct contour should pass. The actual choice does not matter, but it has to be topologically consistent. The original cases made consistent choices, but the sign change could lead to mistakes. The extended table in [3] shows 33 configurations.

The ambiguities were improved upon in later algorithms such as the 1991 asymptotic decider of Nielson and Hamann[4] which corrected these mistakes.[5][6] Several other analyses of ambiguities and related improvements have been proposed since then; see the 2005 survey of Lopes and Bordlie for instance.[6]

Algorithm [edit]

The algorithm proceeds through the scalar field, taking eight neighbor locations at a time (thus forming an imaginary cube), then determining the polygon(s) needed to represent the part of the isosurface that passes through this cube. The individual polygons are then fused into the desired surface.

This is done by creating an index to a precalculated array of 256 possible polygon configurations (2^8=256) within the cube, by treating each of the 8 scalar values as a bit in an 8-bit integer. If the scalar's value is higher than the iso-value (i.e., it is inside the surface) then the appropriate bit is set to one, while if it is lower (outside), it is set to zero. The final value, after all eight scalars are checked, is the actual index to the polygon indices array.

Finally each vertex of the generated polygons is placed on the appropriate position along the cube's edge by linearly interpolating the two scalar values that are connected by that edge.

The gradient of the scalar field at each grid point is also the normal vector of a hypothetical isosurface passing from that point. Therefore, we may interpolate these normals along the edges of each cube to find the normals of the generated vertices which are essential for shading the resulting mesh with some illumination model.

Patent issues [edit]

An implementation of the marching cubes algorithm was patented as United States Patent 4,710,876.[7] Another similar algorithm was developed, called marching tetrahedra, in order to circumvent the patent as well as solve a minor ambiguity problem of marching cubes with some cube configurations. The patent expired in 2005, and it is now legal for the graphics community to use it without royalties since more than 17 years have passed from its issue date (December 1, 1987[7]).

Sources [edit]

1. ^ William E. Lorensen, Harvey E. Cline: *Marching Cubes: A high resolution 3D surface construction algorithm.* In: *Computer Graphics*, Vol. 21, Nr. 4, July 1987
2. ^ *The Marching Cubes*.
3. ^ *a* *b* *Marching Cubes 33: Construction of Topologically Correct Isosurfaces*.
4. ^ Nielson, Gregory M.; Hamann, B. (1991). "The asymptotic decider: resolving the ambiguity in marching cubes". *Proceeding VIS '91 Proceedings of the 2nd conference on Visualization '91*.
5. ^ Charles D. Hansen; Chris R. Johnson (2004). *Visualization Handbook*. Academic Press. p. 9. ISBN 978-0-12-387582-2.
6. ^ *a* *b* A. Lopes; K. Bordlie (2005). "Interactive approaches to contouring and isosurfaces for geovisualization". In Jason Dykes; Alan M. MacEachren; M. J. Kraak. *Exploring Geovisualization*. Elsevier. pp. 352–353. ISBN 978-0-08-044531-1.
7. ^ *a* *b* Marching Cubes, US Patent Office entry

See also [edit]

- Image-based meshing
- Marching tetrahedrons
- Asymptotic decider
- Marching squares

External links [edit]

Wikimedia Commons has media related to ***Marching cubes***.

- Lorensen, W. E.; Cline, Harvey E. (1987). "Marching cubes: A high resolution 3d surface construction algorithm". *ACM Computer Graphics* **21** (4): 163–169. doi:10.1145/37402.37422.
- Nielson, G. M.; Hamann, Bernd (1991). "The asymptotic decider: resolving the ambiguity in marching cubes". *Proc. 2nd conference on Visualization (VIS' 91)*: 83–91.
- Montani, Claudio; Scateni, Riccardo; Scopigno, Roberto (1994). "A modified look-up table for implicit disambiguation of Marching cubes". *The Visual Computer* **10** (6): 353–355. doi:10.1007/BF01900830.
- Nielson, G. M.; Sung, Junwon (1997). "Interval volume tetrahedrization". *8th IEEE Visualization (VIS'97)*. doi:10.1109/VISUAL.1997.663886.
- Paul Bourke. "Overview and source code".
- Matthew Ward. "GameDev overview".
- "Introductory description with additional graphics".
- "Marching Cubes".. Some of the early history of Marching Cubes.
- Newman, Timothy S.; Yi, Hong (2006). "A survey of the marching cubes algorithm". *Computers and Graphics* **30** (5): 854–879. doi:10.1016/j.cag.2006.07.021.
- Stephan Diehl. "Specializing visualization algorithms" (PDF).

Categories: Computer graphics algorithms | 3D computer graphics | Mesh generation