A Literature Review of Medical Imaging 3D Mesh Generation

Abstract

This literature review shall provide a chronological analysis of the techniques and algorithms used to generate organic 3D meshes, but more specifically, how 3D meshes can be generated from medical scan images. The objective of this literature review is to identify the trending methodologies and algorithms shared between academics, how these algorithms have evolved over time, and what is missing from the current research pool. A meaningful conclusion and implementation plan can then be derived and used to aid myself in developing the techniques and algorithms for myself.

Introduction

Medical Imaging 3D Mesh Generation refers to the process of converting the 2D image outputs of a medical scan, such as Magnetic Resonance (MR) and Computerised Tomography (CT), into a digital 3D model. This process can not only be used by medical professionals to aid diagnosis and treatment, but also be used for medical training purposes and to help patients, with an untrained eye, better visualise their scans (Beveridge et al. 2013). IBM (2020) also support these findings and take the notion one step further, by 3D printing the digital models. IBM (2020) specifically comment on improvements to the communication between radiologists and planning surgical procedures. The above evidence clearly denotes the use and importance of this technology, however, does not highlight the technicalities of how this 3D visualisation is achieved. This literature review will critically explore the methodologies behind the implementation of this technology and how they have changed over time.

Background Research

A Generalization of Algebraic Surface Drawing, proposed by Blinn (1982), is one of the first papers to tackle realistic mesh generation of organic molecular structures. Blinn's (1982) methodology involved creating a mesh by drawing quadric and bivariate curved surfaces "directly from their mathematical definitions" (p. 235), as opposed to dividing the mesh into many polygons. The motivation of the paper was to replace the current ball-and-stick/space-filling molecular models, with a representation that could see individual molecules realistically stretch and break (Figure 1).

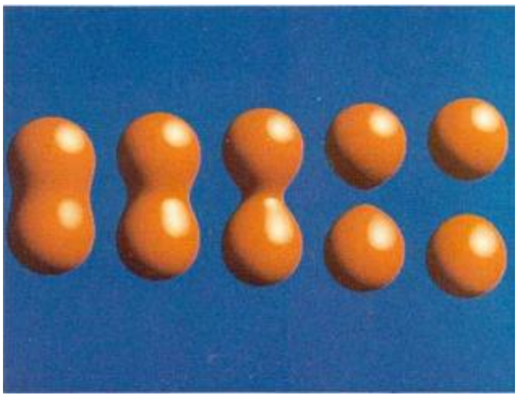

Figure 1: Molecule bonds stretching and breaking (Blinn 1982, p. 237).

Despite the algorithm's successful simulation, the class of shapes that could be drawn had their limitations (Blinn 1982). Due to the curved and symmetrical nature of mathematical functions, the algorithm may not have been best suited to drawing the irregularities of humanoid geometry, despite Blinn's (1982) basic demonstrative efforts (Figure 2).

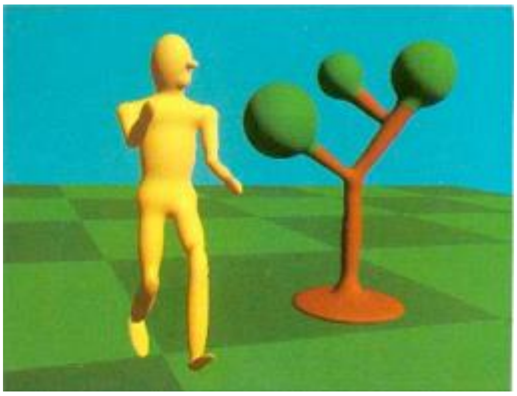

Figure 2: Humanoid mesh using general quadric exponents (Blinn 1982, p. 255).

Some years later, Lorensen and Cline (1987) established an algorithm known as Marching Cubes, that could use image point data from medical scan images, to generate a 3D mesh. Their algorithm required the following four-stage approach (Figure 3): Data Acquisition, Image Processing, Surface Construction and Display.

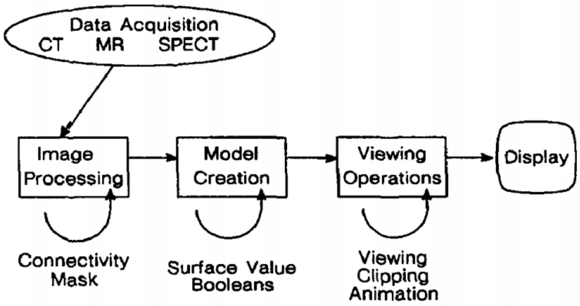

Figure 3: Beginning to end, workflow diagram (Lorensen and Cline 1987, p. 163).

Stage 1: Data Acquisition
Initially, an MR, CT or other medical scan needed to be conducted to create and obtain a dataset. The resultant data from the scan would consist of many 2D images, captured at incremental points throughout the scan, to create an image stack.

Stage 2: Image Processing
Each image from the stack would then be sampled at equally determined intervals to create a grid plane, where the corner of each square would be associated with a pixel from the original image. Two parallel grid planes could then be stacked to outline the vertices of cubes (Figure 4).

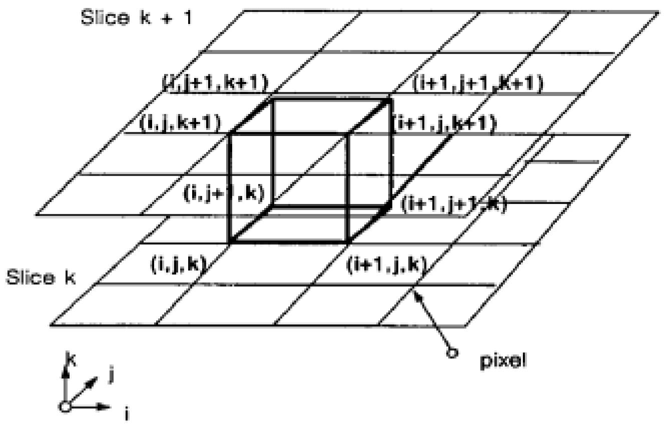

Figure 4: Stacked parallel grid planes to create a cube (Lorensen and Cline 1987, p. 164).

Stage 3: Surface Construction

The Marching Cubes algorithm could then be executed, by first querying the colour value of each of the sampled points that makes up each cube and determining whether they lie within the shape of interest (isosurface), or not. For example, if the isosurface is white, sampled points with a white colour value will lie within or on the boundary of said surface, whereas all other colour values will lie outside. Lorensen and Cline (1987) concluded a total of 256 (2^8) possible cube-point configurations, which could be condensed down into 15 unique configurations (Figure 5), when disregarding symmetries. Each individual cube would then have its required number of polygons drawn to match its configuration, where all cubes will contribute towards the final resulting mesh.

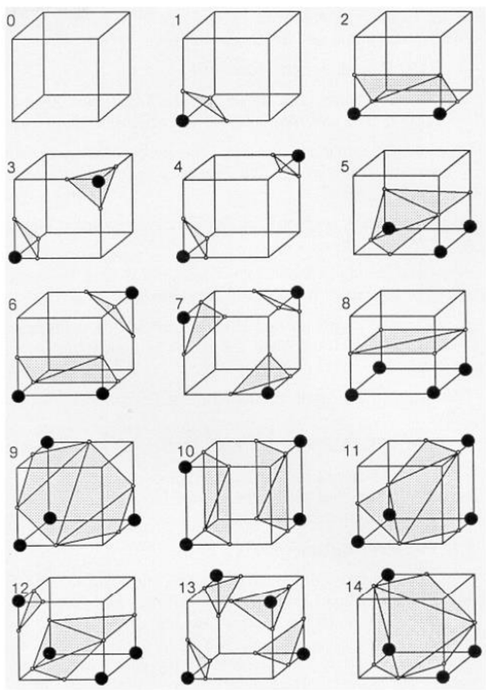

Figure 5: Triangulation Table (Lorensen and Cline 1987, p. 165).

Stage 4: Display
Finally, the resulting mesh would be displayed to the screen using ray casting, depth shading and colour shading (Lorensen and Cline 1987).

Criticisms and Development

Whilst the original Marching Cubes methodology has not changed much over time (Custodio et al. 2019), Lorensen and Cline (1987) have received a significant amount of criticism from other academics about the incompletion of their triangulation table. Their triangulation table consisted of many ambiguities where actually more than 15 unique patterns existed (Custodio et al. 2019). Tcherniaev (1995) went as far as extending the triangulation table to contain a total of 33 possible cube-point configurations (Figure 6), also later proven by Nielson (2003).

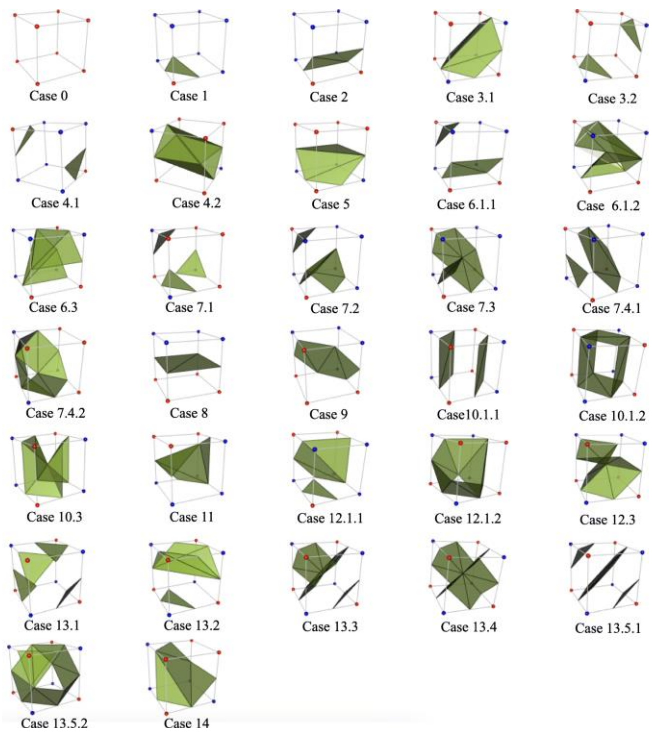

Figure 6: Tcherniaev’s proposed Triangulation Table (Custodio et al. 2019).

Furthermore, despite the simplicity, robustness and cheap computational cost of the Marching Cubes algorithm, its primary weakness lies in the quality of the resulting mesh (Custodio et al. 2019). The algorithm often produces a mesh containing many degenerate and skinny triangles; however, this was partially tweaked and resolved with an extended lookup table by Raman and Wenger (2008). The lookup table took to labelling each grid vertex as “+”, “-“ or “=”, as opposed to just “+=“ or “-” in relation to its position within, outside or on the isosurface. However, Raman and Wenger’s (2008) methodology only extended as far as satisfying 23 of the 33 extended configurations. As of recent, Custodio et al. (2019) have produced a method that guarantees a correct interpretation of the generation of all 33 Marching Cubes configurations. Custodio et al. (2019) conclude their paper by highlighting what is currently missing from the literature which is to implement a generalisation of the extended triangulation to higher dimensions.

Conclusion and Implementation Plan

In conclusion to my research surrounding the literature of the Marching Cubes algorithm, I plan to implement the algorithm for myself as a part of my outstanding Animation Software Engineering and Computer-Generated Imagery Techniques assignments. I intend to create a visualisation tool that will read medical scan images, apply the Marching Cubes algorithm, and draw the resulting mesh to the screen, using the NCCA graphics library and OpenGL. The visualisation tool shall be required to allow the user to pan and zoom around the generated mesh, and as an extended feature be able to toggle on/off layers of the geometry. For example, the user may choose to inspect the skin, bones, or organs of a mesh. OpenNEURO (2020) will act as my primary resource for acquiring data, by providing a free and open platform for the distribution of many types of medical scan images. The initial implementation of the algorithm will aim to satisfy Lorensen and Cline’s (1987) 15 unique configurations, using a similar methodology to Bourke (1994). As an extension, I can then begin to work towards Tcherniaev (1995) and Nielson’s (2003) 33 configurations and tackle the unwanted consistency of generating, degenerate and skinny triangles.

References

Beveridge, E., Ma, M., Rea, P., Bale, K. and Anderson, P., 2013. 2013 International Conference on Computer Medical Applications (ICCMA). *3D Visualisation for Education, Diagnosis and Treatment of Iliotibial Band Syndrome* [online], 1-6.

Blinn, J. F., 1982. A generalization of algebraic surface drawing. *ACM Transactions on Graphics*, 1 (3), 235-256.

Bourke, P., 1994. *Polygonising a scalar field* [online]. Available from: http://paulbourke.net/geometry/polygonise/ [Accessed 12 November 2020].

Custodio, L., Pesco, S. and Silva, C., 2019. An extended triangulation to the Marching Cubes 33 algorithm. *Journal of the Brazilian Computer Society* [online], 25 (6).

IBM, 2020. *How 3D surgical model printing can help improve insights from medical imaging data* [online]. Available from: https://www.ibm.com/blogs/watson-health/model-improve-medical-data/ [Accessed 27 November 2020].

Lorensen, W. E. and Cline, H. E., 1987. Marching Cubes: A High Resolution 3D Surface Construction Algorithm. *ACM SIGGRAPH Computer Graphics*, 21 (4), 163-169.

Nielson, G. M., 2003. On marching cubes. *IEE Transactions on Visualization and Computer Graphics* [online]. 9 (3), 283-297.

OpenNEURO., 2020. *OpenNEURO A free and open platform for sharing MRI, MEG, EEG, iEEG, and ECoG data.* Available from: https://openneuro.org/ [Accessed 28 November 2020].

Raman, S. and Wenger, R., 2008. Quality Isosurface Mesh Generation Using an Extended Marching Cubes Lookup Table. *Computer Graphics Forum* [online]. 27 (3), 791-798.

Tcherniaev, E. V., 1996. Marching Cubes 33: Construction of Topologically Correct Isosurfaces. *European Organization For Nuclear Research (CERN)* [online].