Advances in Engineering Software 102 (2016) 83–96

Contents lists available at ScienceDirect

Advances in Engineering Software

journal homepage: www.elsevier.com/locate/advengsoft

Research paper

2-manifold surface meshing using dual contouring with tetrahedral decomposition

Tanweer Rashid, Sharmin Sultana, Michel A. Audette*

Department of Modeling, Simulation and Visualization Engineering, Old Dominion University, 4700 Elkhorn Ave, Norfolk, VA 23529, USA

ARTICLE INFO

Article history:
Received 24 February 2016
Revised 28 August 2016
Accepted 25 September 2016

Keywords:
Surface meshing
2-manifold
Watertight
Dual contouring
Mesh generation
Tetrahedral decomposition

ABSTRACT

The Dual Contouring algorithm (DC) is a grid-based process used to generate surface meshes from volumetric data. The advantage of DC is that it can reproduce sharp features by inserting vertices anywhere inside the grid cube, as opposed to the Marching Cubes (MC) algorithm that can insert vertices only on the grid edges. However, DC is unable to guarantee 2-manifold and watertight meshes due to the fact that it produces only one vertex for each grid cube. We present a modified Dual Contouring algorithm that is capable of overcoming this limitation. Our method decomposes an ambiguous grid cube into a maximum of twelve tetrahedral cells; we introduce novel polygon generation rules that produce 2-manifold and watertight surface meshes. We have applied our proposed method on realistic data, and a comparison of the results of our proposed method with results from traditional DC shows the effectiveness of our method.

1. Introduction

Surface meshing is an invaluable tool and one of the most commonly used methods in scientific research for visualizing volumetric data. A surface mesh of a real-world object can be generated in one of two ways: (1) by using a scanning device such as the NextEngine 3D Laser Scanner or Microsoft's Kinect, or (2) by isosurface extraction from volumetric data such as MRI or CT using contouring algorithms such as Marching Cubes (MC) [1], Dual Contouring (DC) [2] or Dynamic Particle Systems [3]. In both cases, the resulting polyhedral mesh may contain geometric errors such as non-manifold edges and/or vertices, holes and intersecting polygons, especially if the surface being meshed is complex. The survey of Ju in [4] discusses the wide range of techniques that has been developed for repairing polygonal models.

Non-manifold geometry is problematic for a variety of situations, such as rendering of refractive surfaces, computation of surface normals and curvatures, bounding tetrahedral meshes suitable for finite element analysis and fluid simulations, as well as CAD-based manufacturing and 3D printing. The repairing of geometric errors in meshes is an active research area and there is no one-fits-all algorithm that can fix all the different types of geometric errors. Of course, this is not to say that topologically and geometrically correct surface mesh generation is a poorly researched field. [5] presents an extensive review of the many variants of the MC algorithm that have been developed over the years. *Tight cocone* [6] is another meshing algorithm that guarantees watertight meshes. *Marching Tetrahedra* [7] is another method similar to MC that can produce topologically correct meshes.

This work focuses primarily on Dual Contouring. DC offers the advantage of producing meshes with sharp features [2]. In MC, the newly created vertices are constrained to the edges of the grid while in DC, the vertices can be anywhere inside the grid cube. However, the traditional DC algorithm produces non-manifold edges and vertices in certain situations. In this work, we present a modified Dual Contouring algorithm that is capable of generating 2-manifold meshes and thereby avoid non-manifold geometric errors in the first place.

The remainder of this paper is divided into the following sections: Section 2 discusses in general how the traditional DC algorithm works and what the current state of the art is. Section 3 describes our proposed solution in detail. Section 4 shows some of the results of the proposed method and Section 5 concludes with a discussion of some of the limitations of the proposed method.

2. Dual contouring

2.1. An overview of dual contouring

Dual Contouring (DC) is a method used for extracting the surface boundary of an implicit volume. The method is dual in the sense that vertices generated by DC are topologically dual to faces

* Corresponding author. Fax: 17576836940.
E-mail addresses: trash001@odu.edu (T. Rashid), ssult003@odu.edu (S. Sultana), maudette@odu.edu (M.A. Audette).

http://dx.doi.org/10.1016/j.advengsoft.2016.09.008
0965-9978/

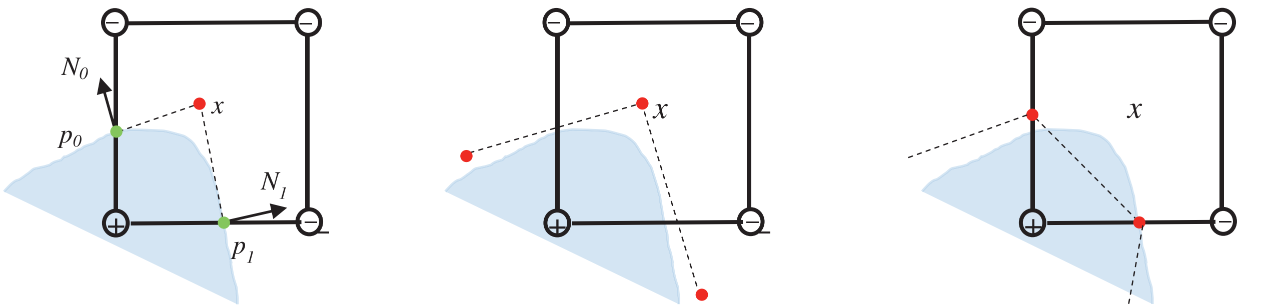

Fig. 1. (Left) Formulation of quadratic error functions. The blue region represents the surface/volume. (Middle) Edges as well as a sharp feature generated with DC, (Right) Edges generated with MC. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

in the Marching Cubes (MC) algorithm. In DC, a uniform grid is superimposed on the implicit volume. The grid cubes are represented as nodes in an octree data structure. For each grid cube intersecting the volume, the eight corners of the cube are assigned inside/outside labels, and a quadratic error function (QEF) is defined as $E[x] = \Sigma(N_i \cdot (x - p_i))^2$ where x is the computed dual vertex or *minimizer*, and p_i and N_i represent the intersections and unit normal, respectively, of the volume boundary with the edges of the cube.

Fig. 1 (left) illustrates the basic concept of QEFs in 2D. The bounding surface of the volume shown in light blue color intersects the lower left corner of a cube. The lower left corner of the cube is marked with a "+" sign indicating that it lies inside the volume while the remaining corners of the cube are marked with a "−" sign indicating that they lie outside the volume. Furthermore, the surface intersects the left and bottom edges of the cube at points p_0 and p_1 (green points), respectively. If a tangent were drawn from points p_0 and p_1 and extended inside the cube, they would intersect each other somewhere inside the cube at x (red point). This point would be a vertex of the isosurface. Typically, one minimizer is computed for each grid cube containing a sign change. The minimizer can be anywhere inside the grid cube, rather than being restricted to the edges of the cube as in MC. This feature allows DC to produce meshes with sharp features, as shown in Fig. 1 (middle), whereas MC cannot, as shown in Fig. 1 (right).

The objective function $E[x]$ can be expressed as the inner product $(Ax - b)^T(Ax - b)$ where A is a matrix whose rows are the normals N_i and b is a vector whose entries are $(N_i \cdot p)$. The function $E[x]$ can then be expanded as

$$E[x] = x^T A^T A x - 2x^T A^T b + b^T b \tag{1}$$

where $A^T A$ is a symmetric 3×3 matrix, $A^T b$ is a column vector of length three and $b^T b$ is a scalar. This representation of a QEF can be solved using the QR decomposition [8], and it should be noted that Singular Value Decomposition (SVD) [9,10] can also be employed for solving this system.

In traditional DC, a recursive method using the three recursive functions *cellProc(), faceProc()* and *edgeProc()* is used to traverse through the octree during the polygon generation phase. For each minimal edge exhibiting a sign change, a quadrangle or two triangles are generated by connecting the minimizers of the cubes containing the minimal edge.

2.2. Background and literature review

One of the main disadvantages of DC is that it does not guarantee 2-manifold and intersection-free surfaces. A polygonal mesh is considered as being 2-manifold if each edge of the mesh is shared by only two faces, and if the neighborhood of each vertex of the mesh is the topological equivalent of a disk. Ju and Udeshi address the issue of intersecting triangles in [11] by proposing a hybrid method where dual vertices (inside grid cubes) as well as face vertices and edge vertices (inserted on the cube's face and edges, respectively) are used to create polygons according to new polygon generation rules. Zhang et al. in [12] present a topology-preserving algorithm for surface simplification using vertex clustering and an enhanced cell representation, but this method is unable to avoid non-manifold edges and vertices. Varadhan et al. [13] suggest an approach that combines edge intersection testing, adaptive subdivision, and dual contouring to reconstruct thin features. Schaefer et al. use a vertex clustering method in [14], where they present an additional topology criterion that must be satisfied to ensure manifoldness.

Zhang and Qian in [15] take a different approach by first generating a base mesh using standard DC, and then analyzing and categorizing the octree leaf cells into 31 topology groups. For ambiguous cubes, multiple minimizers, as many as three in some instances, are inserted whereby a new topologically correct mesh is created by reconnecting the vertices of the mesh with the newly inserted minimizers. In [16], Zhang and Qian decompose ambiguous cubes into twelve tetrahedral cells, each having one minimizer, and construct a series of polygons and polyhedrons to create tetrahedral meshes. This method can avoid topological ambiguities in tetrahedral meshes but does not produce surface meshes.

Our proposed method uses an approach similar to that in [16] by decomposing an ambiguous cube into several tetrahedral cells. In this work, we introduce novel polygon generation rules that result in 2-manifold and watertight triangular surface meshes.

3. 2-Manifold dual contouring

Our proposed method begins the same way as in classical Dual Contouring (DC) by superimposing a uniform virtual grid onto the implicit volume. Depending on the isovalue chosen, the corners of each cube of the grid can have 2^8 or 256 possible configurations. By taking rotation and symmetry into account, these configurations can be reduced into 14 fundamental cases, as shown in Fig. 2. Cases 0, 1, 2, 5, 8, 9 and 11 are simple unambiguous cases, meaning there is only one possible surface intersecting the grid cube (no surface for Case 0). Cases 3, 4, 6, 7, 10, 12 and 13 are ambiguous, meaning that there is more than one possible surface that intersects the cube. It is the presence of these ambiguous cubes, as well as the fact that standard dual contouring produces only one minimizer for each cube, that causes non-manifold surfaces to arise. Additionally, in our experiments we have observed that the complement of Case 4 (that is, a situation where the two diagonally opposite corners of the cube are in background and the rest are in the foreground) is also responsible for the generation of non-manifold vertices, as shown in Fig. 3. These particular non-manifold vertices occur inside the surface mesh. In [17], Sohn shows that a cubic cell can be decomposed into a set of