Computer-Aided Design 37 (2005) 899–907

COMPUTER-AIDED DESIGN

www.elsevier.com/locate/cad

Block Cartesian abstraction of a geometric model and its application in hexahedral mesh generation

Y. Su[a,*], A. Senthil Kumar[b]

[a]Institute of High Performance Computing, 1 Science Park Road 01-01, The Capricorn, Singapore Science Park II, Singapore 117528
[b]Department of Mechanical Engineering, National University of Singapore, 10 Kent Ridge Crescent, Singapore 119260

Accepted 20 September 2004

Abstract

In this work, a fuzzy logic approach is proposed to transform a geometric model of arbitrary shape to its block Cartesian abstraction. This abstraction is topologically similar to the original model and it contains geometric sub-entities which are all aligned in the Cartesian directions. This is achieved by calculating the modifications made to the face normal vectors as a result of the influences of the adjacent faces. A fuzzy logic inference engine is developed by combining heuristics to emulate the local changes in face normal vectors with respect to the changes in the global space. A three-dimensional field morphing algorithm is used to position the features of this block Cartesian abstraction so that a congruent geometric model can be reconstructed. Such a model is useful for the generation of structured quadrilateral boundary element meshes or structured hexahedral meshes based on grid-based meshing method, mesh mapping or sweeping. This approach is also able to overcome the traditional problem of having poorly shaped elements at the boundary using the grid-based method of mesh generation. As the topology of the block Cartesian abstraction is congruent to the original model, the mesh can be mapped back to the original model by employing an inverse operation of the transformation.

Keywords: Block Cartesian abstraction; Fuzzy logic; Field morphing; Laplacian-Isoparametric transformation; Hexahedral mesh generation

1. Introduction

In the preparation of a simulation model for numerical analysis, it is often required to pre-process the geometric model so that it can be meshed effectively. Processes like feature recognition and suppression, and domain decomposition are commonly employed with mesh generation algorithms to automatically create the finite element mesh. While automatic tetrahedral mesh generation techniques have matured, robust automatic hexahedral mesh generation remains a challenge. Given the rigid nature of the hexahedral grid [1], it becomes even more difficult if a structured mesh is required.

The research in automatic hexahedral meshing algorithms can be classified under three main categories: the block decomposition method, the superposition method and the advancing front method. The block decomposition approach involves subdividing the domain into meshable sub-entities and then using appropriate algorithms to discretize these sub-parts. Examples of such algorithms are the swept volume decomposition and recomposition method [2], the medial axis transformation [3–5] and the midpoint subdivision and integer programming method [6], and the basic logical bulk shape (BLOBS) method [7–9]. The advancing front approach generates the mesh by starting at the boundary of the model and progressively building elements into the interior of the model. Some examples of algorithms employing this approach are the whisker weaving method [10,11] and the plastering method [12,13]. In the superposition approach, a sufficiently large mesh is superimposed on the model and it is then adapted to the boundary of the model. An example of such a class of algorithms is the modified grid-based method which either employ the isomorphic transformation approach [14] or the projective approach [15]. Other variants involve using

* Corresponding author. Tel.: +65 6419 1557; fax: +65 6419 1230.
E-mail address: suyi@ihpc.a-star.edu.sg (Y. Su).

0010-4485//$ - see front matter © 2004 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cad.2004.09.017

the octree scheme [16], or a sculpting algorithm [17] to generate the initial mesh.

The advantages of the different meshing techniques are balanced between two important issues: the quality of the boundary mesh and the quality of the core mesh. Achieving one usually compromises the other. A comparison of the different mesh generation algorithms was made by Su et al. [15] on their strengths and weaknesses.

In this paper, a fuzzy logic inference engine is proposed to map the geometric domain to a block Cartesian space. By meshing this block Cartesian abstraction, both the boundary and core mesh can be of high quality. Moreover, the mesh is essentially structured in nature. Apart from the immediate application in automatic hexahedral mesh generation, the block Cartesian abstraction is also useful in the area of feature recognition and domain decomposition. The paper is organized as follows: Section 2 describes the methodology of the algorithm, Section 3 presents the usefulness of this method with respect to the new grid-based hexahedral mesh generation algorithm, and Section 4 concludes the paper.

2. Methodology

The objective of this paper is to develop a robust algorithm to obtain a block Cartesian abstraction of a solid model with arbitrary shape. The task is to modify the original geometric model such that its sub-entities conform to the Cartesian directions, that is, its faces lie along the *xy*-, the *yz*- or the *zx*-plane, and its edges are parallel to the *x*, *y* or *z*-direction. Chiba et al. [18] has proposed a method to generate such a recognition model. However, the algorithm faces stability problems and it fails to converge when certain features are encountered, like a 45° chamfer. In this paper, a new fuzzy logic engine is proposed. The major differences are as follows:

(i) The use of surface normal vectors to calculate the new orientations of the sub-entities of the model rather than using edge directions.
(ii) The application of a different fuzzy logic inference engine for the computation of the new orientations of geometric entities.
(iii) The employment of a feature placement algorithm for positioning the features of the model.

2.1. Creation of a tessellated model

To generate a Cartesian abstraction, it is first required to obtain a tessellated model so that every curved edge is approximated by straight line segments and every curved surface is approximated by triangular facets while planar faces are approximated by polygonal boundaries. The degree of tessellation must be such that the number of line segments and facets is minimal yet adequately represents the original model. An estimated length of the arc segment l used in the tessellation is given by

$$l \approx \frac{\pi}{4K_{\max}} \quad (1)$$

where $K_{\max}$ is the maximum curvature of the edge. If the curve is a straight line ($K=0$), it is not tessellated. Next, a set of triangular facets are used to approximate all non-planar faces using the tessellated edges as contraints to the triangulation, which is achieved by standard Delaunay's algorithm, as illustrated in Fig. 1.

2.2. Face normal reassignment

In order to create a Cartesian abstraction, all the face normals of the tessellated model must be reoriented in the *x*, *y*, or *z*-direction. There is, however, no unique way of determining the directions of the face normals and the problem is made much more complicated since changes made in local regions have an impact in the global sense. To solve this problem, a fuzzy logic system with three inputs (antecedent) and one output (consequent) is implemented. Consider two adjacent faces *A* and *B* as shown in Fig. 2, the probabilities ($P_{\eta,A}$ and $P_{\eta,B}$) that their face normals are assigned to the η-Cartesian directions are determined based on relation shown in Fig. 3(a), where θ_η is the angle between the face normal and the η-direction. The choice of the function is such that

$$P_x + P_y + P_z = \cos^2\theta_x + \cos^2\theta_y + \cos^2\theta_z = 1 \quad (2)$$

The probability P_α that these two faces are assigned to the same direction is also determined based on relation shown in Fig. 3(b), where θ is the angle between the two adjacent faces *A* and *B*.

Given these values, the interest is to find the effect which one surface has on the other in terms of the assignment modification $\Delta P_{\eta,A}$ in each of the η-Cartesian direction. The logic of this system is described as follows:

'If surfaces *A* and *B* are close to planar (the dihedral angle is small), then the assignment modification tends to change the normal vector of surface *A* in the direction of that

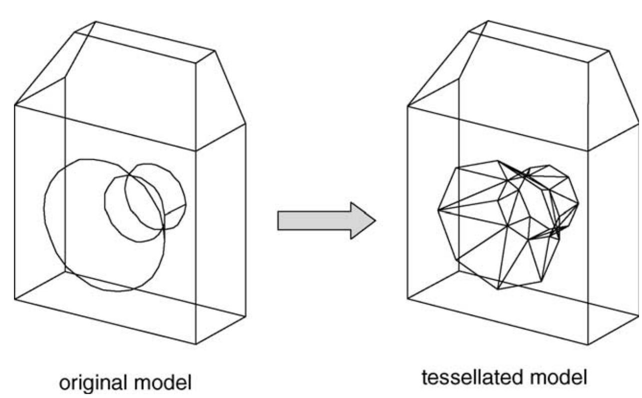

Fig. 1. Tessellation of a geometric model.