Journal of Theoretical and Applied Information Technology
15th December 2012. Vol. 46 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

AUTOMATIC RECOGNITION OF FEATURES FROM CAD MODELS FOR MESH GENERATION

[1]XIANGWEI ZHANG, [2]LILI HUANG

[1]School of Management, Shandong University, Jinan 250100, Shandong, China;

[2]Institute of Engineering Mechanics, Shandong Jianzhu University, Jinan 250101, Shandong, China.

ABSTRACT

Automatic recognition of features from CAD models is one of the key technologies for mesh generation. This paper studies the algorithm for automatic recognition of features from CAD models in STL data format. Firstly, a new STL data file with topological connection is constructed for the calculation. Secondly, the coplanar relationships among all the triangle facets are constructed initially. The third step is to determine whether all the boundary edges on the same face can make up an end to end closed circle. And the boundary edges are filled out by combining the edge-based method and the face-based method. Fourthly, the solid model face, feature lines and feature point are recognized. Finally, two examples are given to prove the feasibility and validity of the algorithm.

Keywords: *Feature Recognition, CAD Models, Boundary Edge, Mesh Generation*

1. INTRODUCTION

With the development of computer technology and the improvement of numerical method, numerical simulation methods such as finite element method, finite volume method, and finite difference method play more and more important roles in the fields of the science researches and engineering applications. The common ground of these numerical methods is to generate a desired mesh or grid system of the analyzed model. The main steps of numerical analysis or numerical simulation include geometric molding, mesh generation, computation and post-process, etc. According to the research and statistics, the step of mesh generation takes eighty percent of all the times and expenses [1]. Thus the mesh generation for complicated geometries has become a major concern.

Automatic mesh generation has seen dramatic improvements over the last twenty years [2-6]. One of the most important and often overlooked aspects to mesh generation is accessing CAD geometry. Accessing CAD geometry for mesh generation is one of the major technical issues related to moving simulation forward as an essential part of the design process. Many researchers have made much effort on developing various algorithms for identifying regular shaped features from CAD models. Lim et al. [7] presented an edge-based identification of depression and protrusion features on freeform solids represented as B-Rep model with trimmed NURBS patches. Chappuis et al. [8] proposed a method for identifying geometrical primitives from a finite element mesh or set of points of mechanical parts for the purpose of improving the quality of surface regeneration in CAE applications. Sunil et al. [9] presented the design and implementation of a system for automatic identification of features from freeform surface CAD models of sheet metal parts.

At present, typically there are four techniques for CAD geometry access [10,11]: (1) Translation and Healing, (2) Discrete Representations, (3) Direct Geometry Access, (4) Unified Topology Accessing Geometry Directly. Translation and healing is commonly used techniques among of them and it sends messages through industry standard file formats, such as IGES, STL, STEP and Parasolid. The shortcoming of this method is that the solid model is often distorted or polluted in the conversion and the restoration process due to the different precision set of different geometric modeling software.

The research work reported in this paper is motivated by the above observation. The work mainly focuses on developing algorithm for automatic recognition of features from CAD models based on STL files. The rest of the paper is organized as follows: Section 2 presents the solid model construction technique. Section 3 describes the geometrical characters recognition of the solid model. Section 4 gives out two application examples. Conclusions from this research work are summarized in Section 5.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

2. SOLID MODEL CONSTRUCTION

The first required input for mesh generation is a solid model, which can define the geometric characters of objects. In this paper, triangulated boundary representations generated by CAD systems are used in STL (STereo Lithography or Standard Triangulation Language) format. The content of STL files is the data information of a series of triangle patches that approach the surfaces of three dimensional solid models. Each triangle facet is represented in terms of its unit normal pointing toward the exterior of the model and three vertices (x , y , z coordinates) in counterclockwise direction. And each triangle facet of the geometry definition is associated with the local surface curvature, which satisfies the requirement of mesh generation algorithm. Figure 1 shows the triangulation of a supporting pedestal solid model generated by three-dimensional modeling software of UG, which its STL file takes 188 KB.

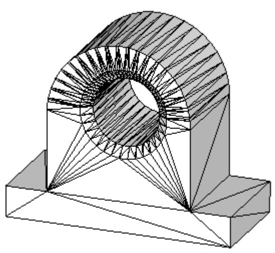


Figure 1: Supporting Pedestal–Triangulated Geometry

Because the data of the triangle facets are irregularly arrayed in STL files, and with no particular ordering, so the data are lack of topology between the facets. In addition, the type of data storage in the STL file results in a large number of redundant data and makes a trouble for the data treatments in the follow-up steps. In order to solve the above problem, this paper reconstructed the index of the STL file according to the data information of the triangle facets in the STL file and established the topology information between the triangle facets and all the vertices of the solid model. The triangle facets and the vertices of the solid model are stored respectively. The storage structure is showed as follows.

```
struct SFace        // Triangle facet structure
{
    double l, m, n;  // The normal vector of the facet
    int V1, V2, V3;  // The vertex index in the facet
};
struct SVertex      // Vertex structure
{
    double x, y, z;  // The coordinates of the vertex
    int  PointID;    // The vertex index
};
```

Where, the triangle facets of the solid model are stored by dynamic arrays of SFace structure, which saves the normal vectors and the vertex number of the facet. The vertices of the solid model are stored by dynamic arrays of SVertex structure, which saves the coordinates and the vertex index. In this way, the vertex coordinates can be read from the vertex structure by the vertex index, thereby avoiding repeatedly storage of the vertex coordinates. By using the dynamic array structures, the data of the STL files can be saved effectively and the computation efficiency of the algorithm can be improved evidently.

3. GEOMETRICAL CHARACTERS RECOGNITION

Geometrical characters recognition of spatial CAE model is one of the prerequisites to generate good quality meshes. There are two main methods to identify the geometric characters: the edge-based method and the face-based method.

The edge-based method is tried to find the points on boundary edges and then join these points to edges. A closed region formed by a series of interconnected edges is called a surface. The points on boundary edges are obtained by the principal curvature of each vertex on the triangle facets. But, the problem with this method is that some non-feature points with discontinuous normal vector or curvature sometimes may be mistakenly identified as feature points. And that will cause the inaccuracy of the geometrical characters recognition.

The face-based method is to identify the surface firstly and then identify the feature edges and the feature points. The surface of the solid model is identified according to the coplanar relationship of each triangle facets, which usually judged through calculating the angle between the normal vector of the triangle facet and those of the three other edge-adjacent triangle facets. The existing question of this method is that the angle threshold is usually given by experience. If the threshold set too small, some edges which is not on the feature boundaries will be misjudged as feature edges and that will cause confusion for mesh generation. If the threshold set too small, some feature edges will be omitted and that will affect the mesh quality likewise.

Journal of Theoretical and Applied Information Technology
15th December 2012. Vol. 46 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

In this paper the geometric characters of the solid models is correctly identified by combining the edge-based method and face-based method. To facilitate the description, some concepts of the triangle facets are defined firstly. Figure 2 shows the neighborhood relationship among triangle facets. Figure 2(a) gives the neighborhood relationship of arbitrary point V_i . The points adjacent to the point V_i is referred to as the neighboring point set V_{ij} . The triangle sides sharing the point V_i is referred to as the neighboring edge set E_{ij} and the triangle facets sharing the point V_i is referred to as the neighboring facet set F_{ij} , where $j = 1,\cdots,m$. In figure 2(b), the neighborhood information of the two triangle facets sharing the edge e which composed by the points V_i and V_j are recorded as (F_i , F_{i+1}). In figure 2(c), an arbitrary facet F is composed by the point V_i , V_j and V_k . The triangle facets sharing edge with the facet F is recorded in the collection of (F_i , F_{i+1} , F_{i+2}).

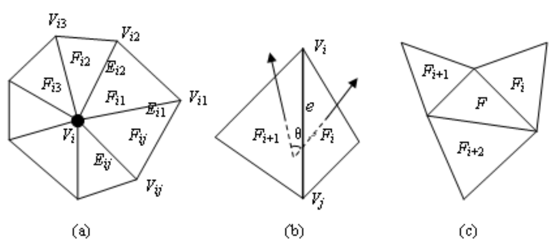


Figure 2: The Neighborhood Relationship Among Triangle Facets

In this paper, the dihedral angle corresponded to an edge is defined as the angle between the outer-normal direction of the two triangle facets which sharing this edge. As shows in figure 2(c), the dihedral angle corresponded to the edge e is the angle θ between the outer-normal direction of the two facets F_i and F_{i+1} which sharing the edge e . A boundary edge is defined as an edge having the dihedral angle greater than a threshold value (usually taken as 20~45 degrees) and record as $CE(i) = \{e_1, e_2, \cdots, e_i\}$. If the triangle side is not a boundary edge, but the perimeter ratio of the two adjacent triangles is greater than a threshold value (here the perimeter ratio of the big triangle and the small triangle is taken as greater than 10), then this triangle side is defined as the candidate boundary edge. The two vertex of the candidate boundary edge is defined as the candidate boundary point. As shown in figure 3, the edge e is not a boundary edge because the corresponding dihedral angle is 15 degrees and less than the threshold value. But the perimeter ratio of the two adjacent triangles is greater than 10, so the edge e is a candidate boundary edge.

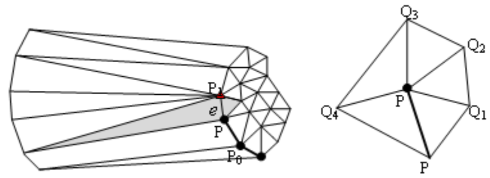


Figure 3: Fill Out The Boundary Edges

The main steps of the geometrical characters recognition are as follows:

(1) Construct the coplanar relationship among all the triangle facets initially. The initial coplanar relationship can be obtained by traversing all the triangle facets. The relationship of every triangle facet with their edge-adjacent triangle facets is recorded in a corresponded coplanar array. The order of the elements in the array is this triangle facet, edge-adjacent triangle facet 1, edge-adjacent triangle facet 2, edge-adjacent triangle facet 3. A triangle facet is coplanar with the edge-adjacent triangle facet if the common edge of them is not a boundary edge. And then the edge-adjacent triangle is saved in the coplanar array of this triangle facet. Otherwise, the array element corresponded to the edge-adjacent triangle facet is set as 0.

Most of the boundary edges can be extracted in this step. But for solid model with smoothing regions, some boundary edges sometimes can be omitted, as shown in figure 3. Therefore, it is necessary to check up whether there are some omissions after the step of boundary edge extraction.

(2) Each solid model in STL format is combined by one or server closed space surfaces. The boundary edge on each surface must make up a closed circle according to geometry continuity of the solid model. Hence, it is required to determine whether all the boundary edges on the same face can make up an end to end closed circle. The boundary edges which can make up end to end closed circle is written to the CLine array. If the point on boundary edges is shared by two boundary edges, it is called boundary point. If it is shared by only on boundary edge, it is called half boundary point. If there exist half boundary point, it means that there exists boundary edges which can not joined end to end, that is, there are some boundary edges incomplete and need to fill out.

Journal of Theoretical and Applied Information Technology
15th December 2012. Vol. 46 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

(3) Complete boundary edge. Taking figure 3 as examples, the algorithm of filling out boundary edges is introduced below.

a) Take a point from the half boundary point and set as the initial point P. The edge P_0P is the edge sharing half boundary point P. Successive point is sought among neighboring points of point P. Candidate successive point is wrote in array if the neighboring points have boundary point, half boundary point or candidate boundary point and recorded as $NQ(i)=\{Q_1,Q_2,\cdots,Q_i\}$. If there are more than one candidate boundary points, then the point which has the maximal angle between the edge P_0P_1 and PQ_i is selected as the successive point P_1. If there is no candidate boundary points, the dihedral angle corresponded to the neighboring edge sharing node P is calculated and the neighboring point on the neighboring edge with maximal dihedral angle is selected as the successive point P_1.

b) The edge PP_1 is the new adding boundary edge. The two triangle facets sharing the edge PP_1 are judged as coplanar in the initial coplanar relationship construction. Because the common edge PP_1 becomes new adding boundary edge, then the two triangle facets sharing the edge PP_1 revise to non-coplanar.

c) Judge the attribute of the point P_1. If the point P_1 is boundary point or half boundary point, then take the next half boundary point as the starting point and repeat the above process. If the point P_1.is just an ordinary surface point, then take P_1 as the starting point and repeat the above process until all the boundary edges joined end to end, namely, the half boundary point no longer exist.

(4) Identify the solid model face, feature lines and feature point. According to the coplanar relationship among the triangle facet and those of the three other edge-adjacent triangle facets, the solid model face (*SF*) can be obtained by layers of searching all the triangle facets. The feature lines (*FL*) are the boundary lines in CLine array. The feature vertices (*FV*) are the point which is shared by three or more feature lines.

The geometrical characters can be identified by using above method. Figure 4 shows the flow chart of geometrical characters recognition of solid model.

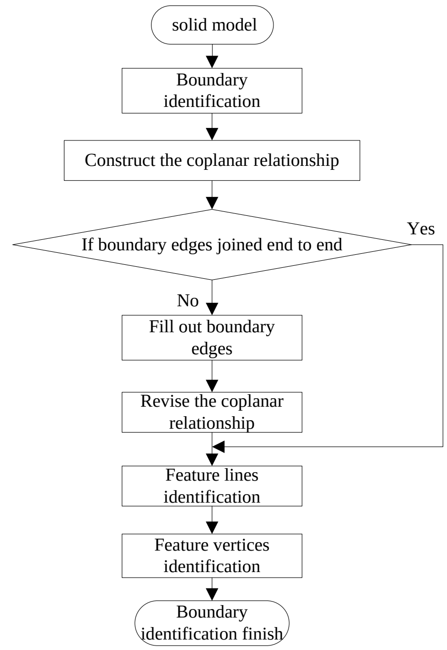


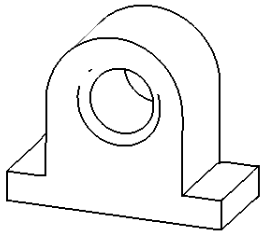
Figure 4: Flow Chart Of Geometric Characters Recognition

4. APPLICATIONS

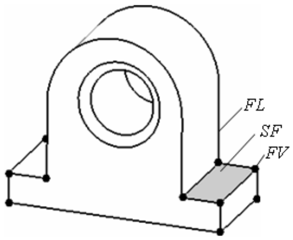
As shown in figure 5, the geometrical characters of the solid model in figure 1 are identified. Figure 5(a) shows the boundary edges identified according to the coplanar relationship among all the triangle facets. Due to the complexity of the surface characteristics, some boundary edges are omitted. If the boundary edges are uncompleted, it will affect the precise matching of the element meshes to the boundaries of the solid model and thus influence the mesh quality. Figure 5(b) shows the geometrical characters of the solid model after the boundary edges completed. The solid circle nodes represent the feature vertices (*FV*), the thick real lines are the feature lines (*FL*), and the shadowed face represents a sub-surface of the solid model (*SF*). Figure 5(c) is the jagged core meshes generated by hexahedral mesh generation algorithms. The number of the jagged core mesh elements is 24787. Figure 5(d) shows the finally generated hexahedral element mesh which is well matched to the boundaries of the solid model and it includes 32781 elements.

Journal of Theoretical and Applied Information Technology
15th December 2012. Vol. 46 No.1

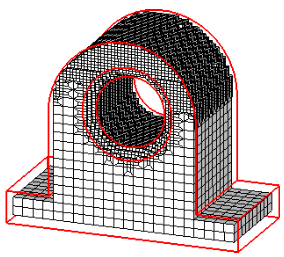
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195



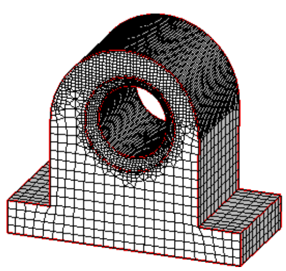
(a)



(b)



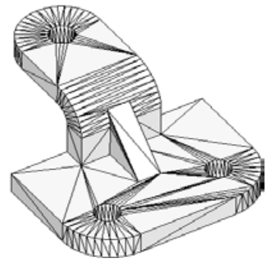
(c)



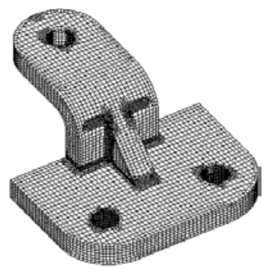
(d)

Figure 5: Boundary Recognition Of Solid Model And Hexahedral Mesh Generation

Figure 6 shows the hexahedral mesh of a linkage model. Figure 6(a) is the triangulation of the linkage solid model. The boundary edges are identified according to the coplanar relationship among all the triangle facets. Figure 6(b) shows the hexahedral mesh. Its refinement field is mainly determined by its surface curvatures and local thickness.



(a)



(b)

Figure 6: Hexahedral Mesh Of A Linkage Model

5. CONCLUSIONS

The automatic recognition of features from CAD models based on STL files is studied in this paper. The STL file generated with the well-known CAD software UG is used to transfer the surface geometrical data information. A new STL data file with topological connection is constructed for the calculation in the next. By using the dynamic array structures, the data of the STL files can be saved effectively and the computation efficiency of the algorithm improved evidently. Geometrical characters recognition of spatial CAE model is one of the prerequisites to generate good quality meshes. There are two main methods to identify the geometric characters: the edge-based method and face-based method. In this paper, the geometric characters of the solid models are correctly identified by combining the edge-based method and face-based method. This study will establish a foundation for precise matching of the element meshes to the boundaries of the solid model and thus improve the mesh quality.

REFERENCES:

[1] B. Raphael, C. S. Krishnamoorthy, "Automating finite element development using object oriented techniques", *Engineering Computations*, Vol. 10, No. 3, 1993, pp. 267-278,.

[2] C. D. Antonopoulos, F. Blagojevic, A. N. Chernikov, et al, "Algorithm, software, and

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

hardware optimizations for Delaunay mesh generation on simultaneous multithreaded architectures", *Journal of Parallel and Distributed Computing*, Vol. 69, No. 7, 2009, pp. 601-612.

[3] Y. Ito, A. M. Shih, B. K. Soni, "Octree-based reasonable-quality hexahedral mesh generation using a new set of refinement templates", *International Journal for Numerical Methods in Engineering*, Vol. 77, No. 13, 2009, pp. 1809-1833.

[4] H. Si, "Constrained Delaunay tetrahedral mesh generation and refinement", *Finite Elements in Analysis and Design*, Vol. 46, No. 1-2, 2010, pp. 33-46.

[5] G. Q. Zhao, H. M. Zhang, L. J. Cheng, "Geometry-adaptive generation algorithm and boundary match method for initial hexahedral element mesh", *Engineering with Computers*, Vol. 24, No. 4, 2008, pp. 321-339.

[6] L. L. Huang, G. Q. Zhao, Z. L. Wang, "Shape quality improvement of three-dimensional hexahedral meshes", *Journal of Information and Computational Science*, Vol. 8, No. 16, 2011, pp. 4007-4014.

[7] T. Lim, H. Medellin, C. Torres-Sanchez, et al, "Edge-based identification of DP-features on free-form solids", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 27, No. 6, 2005, pp. 851–860.

[8] C. Chappuis, A. Rassineux, P. Breitkopf, et al, "Improving surface meshing from discrete data by feature recognition", *Engineering with Computers*, Vol. 20, No. 3, 2004, pp. 202-209.

[9] V. B. Sunil, S. S. Pande, "Automatic recognition of features from freeform surface CAD models", *Computer-Aided Design*, Vol. 40, No. 4, 2008, pp. 502-517.

[10] L. M. Galantucci, G. Percoco, U. Dal Maso, "A volumetric approach for STL generation from 3D scanned products", *Journal of Materials Processing Technology*, Vol. 204, No. 1-3, 2008, pp. 403-411.

[11] D. Wang, O. Hassan, K. Morgan, et al, "Enhanced remeshing from STL files with applications to surface grid generation", *Communications in Numerical Methods in Engineering*, Vol. 23, No. 3, 2007, pp. 227-239.