International Journal of Grid and Distributed Computing
Vol. 9, No. 3 (2016), pp.21-26
http://dx.doi.org/10.14257/ijgdc.2016.9.3.03

Name and Maintain Topological Faces in Rotating and Scanning Features

Gao Xue-Yao[1], Li Jia-Qi[1], Guo Hao[1], Gao Yun-Feng[1], Liu Yu-Hong[1]

[1]*School of Computer Science and Technology, Harbin University of Science and Technology, Harbin 150080, China*
xueyao_gao@163.com

Abstract

Features created in rotating and scanning operations are very complex. Naming and identifying their topological faces is an important problem in CAD fields. In this paper, a new method of coding topological faces in rotating and scanning features is proposed. Firstly, contour segments are numbered. Secondly, an angle between contour segment and rotating axis is computed. Thirdly, all topological faces are named based on contour segments' numbers, rotating axis and other information. When a face splits and several subfaces merge, a method of processing their codes is given. The proposed method is applied to HUST-CAID feature modeling system. Experimental results show that it can name and identify topological faces effectively in operations.

Keywords: *topological faces; contour segments; rotating axis; feature modeling system*

1. Introduction

Hepworth gives a method of naming, caching and normalizing topological entities to identify faces and edges efficiently in models, which can save the modeling time[1]. Liu applies local topological information and geometrical information to name entities in which directed isoparametric lines and face-face relationships are considered[2]. Li uses state vectors to store causal relation among model operations and checks concurrent deletion operations in order to decide whether current operation is masked in collaborative CAD systems[3]. Li analyzes topological entities' matching relations and gives a method to match topological entities in integration of heterogeneous CAD systems. The method consists of information retrieval, information combination and the matching mechanism[4]. Zhang analyzes the design intent and faces' topological evolution process. He uses a dynamic naming method based on topological evolution and variable length string to record entities[5]. Li presents a classification-based approach to match one dimension entities in collaborative designs among heterogeneous CAD systems, and gives conditions of combining entities[6]. Liu applies faces to name all referenced entities, in which local topological information, geometrical characteristics and face-face relationships are used[7]. Jing proposes a new approach to match topological entities among collaborative CAD systems in order to guarantee the correctness and consistency[8]. Jing analyzes four fundamental conditions of solving topological naming problems and their internal relationships[9]. Huang gives a method to match entities in collaborative CAD systems, in which undoing operation, doing operation and redoing operation are adopted to restore contexts of CAD commands[10]. Wang applies graphs to represent the propagation of topological entities in which rules and algorithms are used to identify genetic entities in a model[11]. Wu analyzes parts' design history and gives a face-based mechanism for naming, recording and retrieving topological entities. At the same time, he uses parametric space information to solve ambiguities in modeling process[12]. Zheng proposes a face-based and history-based naming mechanism to record topological entities. He uses face table, edge table and vertex table to manage all entities

ISSN: 2005-4262 IJGDC

in models[13]. Cheng implements synchronization between locking operations and unlocking operations in order to maintain spatial entities' consistency[14]. Sang gives a partial entity structure to represent model boundaries in order to reduce storage size, from which topological adjacent relationships can be derived without the loss of efficiency[15].

In this paper, rotating and scanning features are analyzed. A new method of naming topological faces is proposed in which contour edge and an angle between contour segment and rotating axis are used. At the same time, a method of coding and maintaining topological faces is given.

2. Naming Topological Faces

Features have some specific semantics in models. Semantics of features is usually unchanged in modeling process. In a model, features are composed of multiple elements including topological faces, topological edges and topological vertexes. For faces and edges, they may split and merge in operations. It is very difficult to deal with this problem. When a contour edge rotates at an angle and scans around an axis, a rotating and scanning feature will be gotten. Here, contour edge is a closed ring. The contour edge can be a linear segment or a curving segment. Because the contour edge contains multiple segments, there are multiple topological faces in rotating and scanning features. Each topological face is determined uniquely by a contour segment and an axis. For rotating and scanning features, it is more complicated to name and maintain their topological faces. In this paper, topological faces in rotating and scanning features are coded as RFace_ID=(*fid*, *ty*, *sid*, *ang*, *sp*, *addi*). The code format is shown in Figure 1.

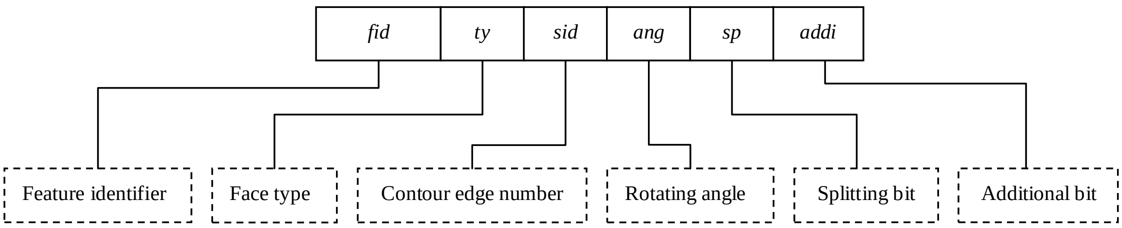

Figure 1. Code Format of Topological Faces

There are six fields in RFace_ID. Here, field *fid* is feature identifier. Field *ty* is the type of a face. When *ty* is 0, it is a flat face. When *ty* is 1, it is a curving face. Field *sid* is contour segment's number and is used to record a contour segment's number. Segments in a contour edge are numbered orderly. Field *ang* is applied to record the angle between a contour segment and rotating axis. Field *sid* and *ang* are used to determine topological faces in rotating and scanning features. Field *sp* is splitting mark and determines whether a topological face splits or not. When *sp* is 1, it means that the face splits. When *sp* is 0, it means that the face does not split. Field *addi* is additional mark, it decides whether a face is geometric boundary in a model or not. When *addi* is 1, the face is geometric boundary. When *addi* is 0, the face is not geometric boundary and it is called as a virtual face. In process of rebuilding a model, a virtual face will be hidden.

A contour edge *con* has n segments including C_1, C_2…, C_n and *ax* is a rotating axis. When contour edge *con* rotates around *ax*, rotating and scanning feature RF is gotten. The algorithm of naming topological faces in RF is shown as follows:

(1) Allocate unique number *fid* for RF and *fid* is inserted into a list of feature numbers.
(2) for(i=1; i<=n; i++)
A[i]=-1.

(3) for(i=1; i<=n; i++)
Calculate the distance from each vertex of C_i to *ax*. Insert minimum value into A[i].
(4) Travel array A and select minimum value A[j]. C_j is used as a reference segment.
(5) Initialize *sid*=1.
(6) for(i=j; i<=n-1; i++)
① Calculate angle *ang* between *ax* and C_i.
② If *ang*=90° then *ty*=0. Otherwises, *ty*=1.
③ *sp*=0, *addi*=1.
④ (*fid*, *ty*, *sid*, *ang*, *sp*, *addi*) is code of a topological face corresponded with C_i.
⑤ *sid*++;
(7) for(i=n; i>=j-1; i--)
① Calculate angle *ang* between *ax* and C_i.
② If *ang*=90° then *ty*=0. Otherwises, *ty*=1.
③ *sp*=0, *addi*=1.
④ (*fid*, *ty*, *sid*, *ang*, *sp*, *addi*) is code of a topological face corresponded with C_i.
⑤ *sid*++;

In a contour edge, a segment is selected whose vertex is nearest to the rotating axis. It is viewed as a reference segment in order to name topological faces. When C_i is a curving segment, its two vertexes are found. For each vertex, an angle between its tangent and *ax* is computed. Then two angles are gotten. A half of the sum of these two angles is set to *ang*. The algorithm can generate a code automatically for each topological face. In order to name topological faces orderly, a reference segment is selected. A segment whose vertex is nearest to *ax* is named firstly. For all segments, vertexes are only considered. Here, field *sid* is applied to number contour segments.

3. Maintain the Consistency of Topological Faces' Codes

In process of modeling operations, topological faces usually split and merge. Here, a virtual face is applied to solve the problem that faces split and merge. RF is a rotating and scanning feature. When a union operation is implemented on features RF and F, a topological face may split into multiple subfaces. When a topological face splits, its code should be converted into a virtual one. It means that this face changes into a virtual one. In process of rebuilding this model, virtual faces do not display. When RF and F fuse, their intersection line is L including L_1, L_2, …, L_n. Calculate intersection point between L_i and boundary line on topological faces in RF. Based on L, intersection point, boundary line, it is decided whether one face splits or not. If this face splits, it is divided into many subfaces according to L, intersection point and boundary line. Code every subface in RF. It is the same to solve the problem that topological faces in feature F split.

A virtual face is named according to code format shown in Figure 1. Its *addi* feild is set to 0. When RF and F fuse, a topological face splits into multiple subfaces and it changes into a virtual face. At the same time, this face's code turns into a virtual one. If the redoing operation is implemented, these subfaces merge into the original face again. In original face's virtual code, *addi* is set to 1. In order to solve the problem that a face splits and merges, a father-son relationship between a face and its subfaces is marked. After a face splits in a rotating and scanning feature, this model is reconstructed. In order to store the relationship between a face and its subfaces, a father-son relationship table is introduced. Its format is shown in Figure 2.

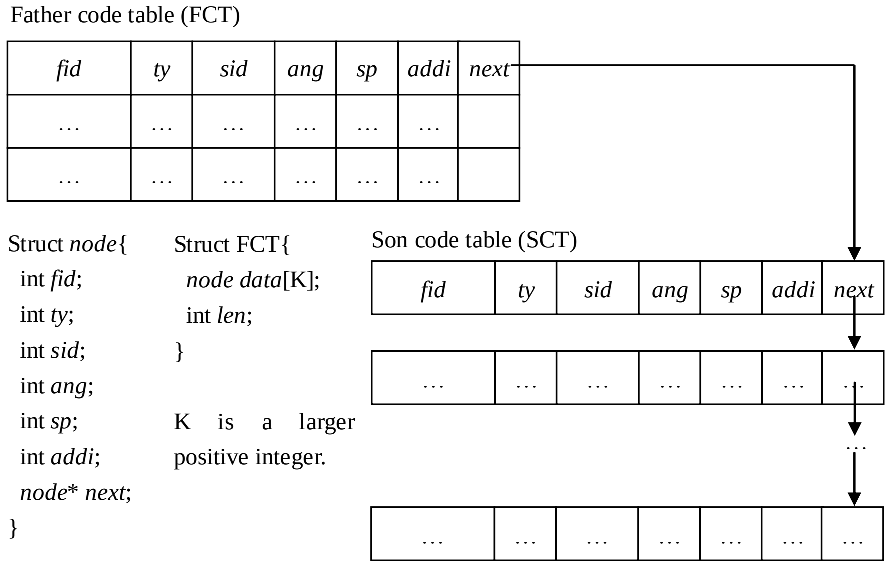

Figure 2. Father-son Relationship Table

All split faces' codes are stored in father code table FCT. The length of FCT is *len*. Here, *node* is a structure and contains field *next* pointing to son code table SCT in which codes of a face's subfaces are stored. When a union operation is implemented on two features, travel all boundary faces in this model and find faces which have split. At the same time, their corresponding codes are gotten. Field *sp* is set to 1 and field *addi* is set to 0. Create PCode whose type is *node* to store this topological face's code, and insert PCode into FCT. Find n subfaces produced by this original face and code them. Create nodes $CCode_1$, $CCode_2$, …, $CCode_n$ to store these n subfaces' codes respectively. Field *next* in $CCode_i$ points to $CCode_{i+1}$, i=1, 2, …, n-1. Field *next* in $CCode_n$ is set to NULL. SCT is built. At the same time, field *next* in PCode points to $CCode_1$. The constraint solving module is called to reconstruct this model and display it.

When a union operation is revoked, n subfaces which will merge are determined. At the same time, travel SCT to find nodes $CCode_1$, $CCode_2$, …, $CCode_n$ which store these n subfaces' codes. After the union operation is revoked, these n subfaces disappear and their father face displays. Their father face's code is stored in node PCode. Delete SCT which PCode.*next* points to and release nodes $CCode_1$, $CCode_2$, …, $CCode_n$. Field *sp* in PCode is set to 0 and field *addi* is set to 1. Apply node PCode to update this topological face's code. Then node PCode is released. Meanwhile, the constraint solving module is called to reconstruct this model and display it.

In father-son relationship table, all split faces' codes and their corresponding subfaces' codes are only stored. Do not store codes of topological faces that do not split. When a face splits, this face's code need to be retained. Its *sp* field and *addi* field should be dealt with. When multiple topological subfaces merge together, their codes will not be retained.

4. Experiments

The method of naming topological faces proposed in this paper is applied to a feature modeling system HUST-CAID. In module that rotating and scanning features generate, the proposed method is applied to code topological faces in features. In modeling process, the method is used to manage and maintain topological faces' codes. A rotating and scanning feature is built in HUST-CAID system. In 2d modeling interface, a contour ring is drawn. The ring is surrounded by five contour segments including a bezier curving line, an arc curving line and three linear lines. In 3d modeling interface, the ring rotates around an axis. Then, a rotating and scanning feature is gotten. It is shown in Figure 3. In Figure

3, this rotating and scanning feature has five topological faces including one inside face and four outside faces. In HUST-CAID system, the proposed algorithm is applied to code each topological face.

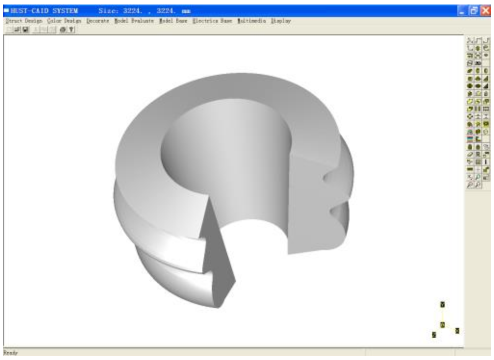

Figure 3. Rotating and Scanning Feature

In 3d modeling interface, two elliptic cylinder features are drawn. Code 3 topological faces in each elliptic cylinder respectively. Finally, a union operation is used to fuse the rotating and scanning feature with each elliptic cylinder together. Use father-son relationship table to store a split face's code and its corresponding subfaces' codes. The reconstructed model is shown in Figure 4.

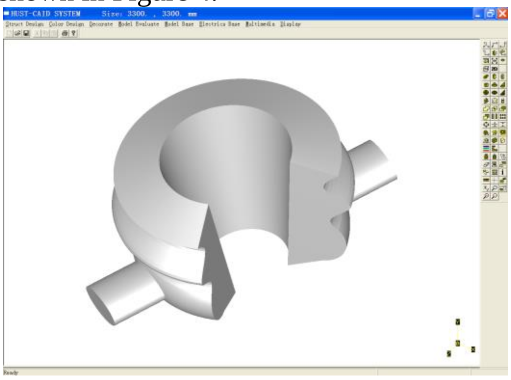

Figure 4. The Reconstructed Model

5. Conclusion

Characteristics of rotating and scanning features are analyzed in this paper. A method of naming and identifying topological faces is proposed, which refers to contour segment's number and an angle between contour segment and rotating axis. A unified code format of topological faces in rotating and scanning features is defined. In union and revocation operations, a new method of maintaining topological faces' codes is given. Field *addi* is used to guide a model's reconstruction. Experimental results show that this method can maintain topological faces' codes effectively that have generated in modeling process.

Acknowledgements

This work is supported by National Natural Science Foundation of China (61502124), Natural Science Foundation of Heilongjiang Province of China (F201420), and Programs of Innovation and Entrepreneurship for Undergraduates of Harbin University of Science and Technology.

References

[1] A. Hepworth, D. Staves, L. Hill and K. Tew, "Enhancements for improved topological entity identification performance in multi-user CAD", Computer-Aided Design and Applications, vol. 12, no. 5, **(2015)**, pp. 627-638.
[2] J. Liu, Z. M. Chen, S. M. Gao and Z. J. Wang, "Design intent maintenance and topological entities naming in design process", Journal of Computer-Aided Design and Computer Graphics, vol. 21, no. 8, **(2009)**, pp. 1106-1117.
[3] X. X. Li, F. Z. He, X. T. Cai and D. J. Zhang, "Consistency maintenance based on the matching of topological entity", Proceedings of the 2012 IEEE 16th International Conference on Computer Supported Cooperative Work in Design, **(2012)**, pp. 464-469.
[4] X. X. Li, F. Z. He and X. T. Cai, "A method for topological entity matching in the integration of heterogeneous CAD systems", Integrated Computer-Aided Engineering, vol. 1, no. 20, **(2013)**, pp. 15-30.
[5] Y. Z. Zhang, X. F. Luo and H. Q. Wang, "A face-based approach to persistently naming and identifying topological entities", Journal of Dalian University of Technology, vol. 4, no. 50, **(2010)**, pp. 508-514.
[6] X. X. Li, F. Z. He and X. T. Cai, "An approach for one-dimensional topological entity matching in collaborative design among heterogeneous CAD systems", Journal of Southeast University, vol. 2, no. 41, **(2011)**, pp. 270-273.
[7] J. Liu and Z. M. Chen, "A new approach for naming and identifying topological entities", Proceedings of the 2nd International Conference on Intelligent Computing Technology and Automation, **(2009)**, pp. 159-163.
[8] S. X. Jing, F. Z. He, S. H. Han, X. T. Cai and H. J. Liu, "A method for topological entity correspondence in a replicated collaborative CAD system", Computers in Industry, vol. 60, no. 7, **(2009)**, pp. 467-475.
[9] S. X. Jing, F. Z. He and H. J. Liu, "Survey of persistent naming problem for topological entities", Journal of Computer-Aided Design and Computer Graphics, vol. 19, no. 5, **(2007)**, pp. 545-552.
[10] Z. Y. Huang, F. Z. He, X. X. Li and Y. Cheng, "To correspond topological entities in non-quiescent context of replicated collaborative modeling system", Proceedings of the 11th IEEE International Conference on Computer-Aided Design and Computer Graphics, **(2009)**, pp. 394-399.
[11] Y. W. Wang, J. J. Wu, L. P. Chen and Z. D. Huang, "Identity propagation method for tracing alterations of a topological entity in a history-based solid modeling system", International Journal of Advanced Manufacturing Technology, vol. 27, no. 3, **(2005)**, pp. 305-312.
[12] J. J. Wu, T. B. Zhang, X. F. Zhang and J. Zhou, "A face based mechanism for naming, recording and retrieving topological entities", CAD Computer Aided Design, vol. 33, no. 10, **(2001)**, pp. 687-698.
[13] J. J. Zheng, M. Fan, R. F. Tong and J. X. Dong, "A mechanism for persistently naming topological entities", Proceedings of the 8th International Conference on Computer Supported Cooperative Work in Design, **(2004)**, pp. 84-89.
[14] C. X. Cheng, P. W. Shen and F. Lu, "Extended locking method for topologically spatial entities", Journal of Computer-Aided Design and Computer Graphics, vol. 17, no. 2, **(2005)**, pp. 209-214.
[15] H. L. Sang and K. Lee, "Partial entity structure: a compact non-manifold boundary representation based on partial topological entities", Proceedings of the 6th ACM Symposium on Solid Modeling and Applications, **(2001)**, pp. 159-170.

Authors

Gao Xue-Yao, She is a Ph.D. She is also an associate professor in Harbin University of Science and Technology. Her research interests are CAD and model retrieval. She has authored and coauthored more than 30 journal and conference papers in these areas.

