http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 6 (6): 914-918, 2007
ISSN 1812-5638

Extracting 2D Projection Contour from 3D Model Using Ring-Relationship-Based Method

[1,2]Zheng Qin, [1]Ji Jia, [1]Tian-Tian Li and [1]Jiang Lu
[1]Department of Computer Science and Technology, Xi'an Jiaotong University, Xi'an, China
[2]School of Information Science and Technology, Tsinghua University, Beijing, China

Abstract: In this study, we propose a novel method to extract 2D projection contour from 3D models. 2D projection contour is a significant feature and data of a 3D shape. It can be used in the fields of 3D model retrieval, analysis of shape, free distortion and joint of curved surface. Triangular mesh is generally adopted as the data representation form of 3D models. Therefore, as a case study, we consider 2D projection contour extraction from triangular mesh of 3D models. In our approach, we perform a discriminant of the visualization to every 3D model curved surface and get the boundary vertexes of them, extract contour vertexes after projection, construct rings and perform a ring-relationship-based judge mechanism to get the 2D projection contour. Our method is unique in that we can extract contours from both single-valued and arbitrary complex space curved surface by ring-relationship-based strategy. Theoretical analysis and experimental results show that we can significantly improve the accuracy and efficiency in contour extraction by using ring-relationship-based method.

Key words: 3D model, triangular mesh, contour vertex, 2D projection contour, ring-relationship-based

INTRODUCTION

Surface mesh data is a discrete data type which is produced by numerical value sampling on mesh vertexes of 3D model. The mesh data distributed on the surfaces of model are composed of numerical data and geometric data (Zhang *et al*., 2003). The geometric data determine the shape of data distribution and data sample type, as well as the numerical data determine the distribution characteristics of attribute values on the curved surface (Tan and Wang, 2000). As a very important data format, surface mesh data is widely used in 3D model retrieval, visualization in scientific computing, CAD, 3D modeling and remodeling, finite element analysis and other fields (Moorhead and Zhu, 1995).

3D model characteristic extraction is a noticeable research subject (Ma and Interrante, 1997; Luebke and Erikson, 1997). For example, when extracting 3D model characteristics for 3D objects shape matching, we are probably concerned with the 3D model curved surfaces or its 2D projection outer contour, the surface mesh data in this instance may not contain numerical data (Zhang *et al*., 2003; Funkhouser *et al*., 2003).

As 3D model surface mesh data are distributed on space curved surface and sampling data present irregular distribution in the spatial coordinates, it is nearly impossible to use the majority of current characteristic analysis methods of image processing field to handle this data type (Zhang *et al*., 2003).

A general approach for extracting contour of 3D polygon mesh model is to apply a judgment to each edge while iterate over all edges of model. The judgment is defined by the dot product of two vectors. One is the normal vector of adjacent patch of an edge and the other is a vector from a point on the edge to the view point. As the formula described in (Zhang and Hoff, 1997), if and only if $(n_1 \bullet (v-e)) (n_2 \bullet (v-e))) = 0$, then the edge is a contour line. Where, v denotes a point on an edge, e is the view point and $n_i (i = 1,2)$ are the normal vectors of the two patches sharing the edge. As contour is viewpoint-dependent, all edges need rejudging when the position of the object or viewpoint is changed (Wu and Liu, 2001). Though the idea is simple, the costs of both computing time and system resource are large, because there is an amount of floating point multiplications and the real-time judging for each edge of the complex 3D model is too time consuming, particularly in the case of parallel and perspective projection. To solve this problem, an improved algorithm for random contour detection is proposed, which is better in real-time response. The key idea is to detect visible edges which are quite likely belonged to the contours by an interactive way (Markosian *et al*., 1997). The acceleration of the extracting process brought by the algorithm depends on the facts that the contour line is composed of only a small proportion of the edges and the edges which constitute new contour are adjacent to the edges of the former contour when the model changes position

Corresponding Author: Ji Jia, No. 28 West Xianning Road, P.O. Box 1841, Xi'an City, Shaanxi Province, 710049, People's Republic of China Tel: +86 13891828753

(Kettner and Welzl, 1997). That is to say, we only need to detect contour by choosing a small quantity of edges randomly, instead of iterate over all edges of a model. The main advantage of this method is that it accelerates the speed of extraction to some extent. However, it still unable to guarantee detecting all edges of contour line rapidly and entirely (Wu and Liu, 2001; Xia and Huang, 2003).

Triangular mesh curved surface is subdivision result of the discrete sampling of surface. Triangulation is one of the most common sampling types in practice. Therefore triangular mesh is generally adopted as the data representation form of sampling surface (Zhang *et al*., 2003). When dealing with triangular mesh, there are some techniques for boundary detection and contour extraction, for instance, extracting boundary vertex by judging whether its adjacent vertexes can form a closed curve, meanwhile, obtaining the boundary curves by sorting the boundary vertexes (Zhang *et al*., 2003). Another approach exploits the property of local maximum value and connectivity of contour to get part of silhouettes and then extracts external silhouettes of 3D models by simple comparisons (Wu and Liu, 2001). The advantage of the approaches suggested by Wu and Liu (2001) and Zhang *et al*. (2003) is high efficiency in extracting contour of triangular mesh. However, since those methods are associated with strict sense of single-valued curved surface, they are inadequate as contour extraction approaches for general curved surface because they are sensitive to overlapping when projecting the curved surface onto plane (Qiu *et al*., 2004).

In this research we introduce a ring-relationship-based method to extract 2D projection contour from 3D object which only contains geometric data. As a case study, we consider extraction from triangular mesh models. We also demonstrate how this technique is applicable to not only single-valued curved surface but also non-single-valued curved surface. Meanwhile theoretical analysis shows high efficiency of our technique.

Methodology of contour extraction: Here, the process of extracting 2D projection contour of triangular mesh 3D model is described.

Definition 1: A patch is visible if the angle between its normal vector and the projection axes is less than 90° Otherwise the patch is invisible.

Definition 2: A curved surface is visible if it is constructed by connecting all the visible patches. Otherwise, the curved surface is invisible.

Definition 3: The vertex on the boundaries that are constructed by connecting the edges between visible and invisible surfaces is called boundary vertex. By projecting those boundary vertexes onto projection plane, we can get contour vertexes. 2D projection contour of 3D model is a closed curve that is created by connecting all those contour vertexes.

Definition 4: A vertex is adjacent to another vertex, if the line connecting the two vertexes is an edge of a patch in the model.

EXTRACTION OF VISIBLE CURVED SURFACE

Extraction of visible curved surface is a process of eliminating all the invisible patches in the 3D mesh model. We first define the following formal description: TriPatch = $\{Tri_1, Tri_2,\ldots,Tri_n\}$ is a set of triangular patches and $Normal_i$ is normal vector corresponding to the triangular patch Tri_i. The algorithm can be summarized four steps as follows:

Step 1: Calculate normal vector $Normal_i$ of triangular patch Tri_i.

Step 2: Calculate the cosine of θ_x, where θ_x is the angle between $Normal_i$ and projection axis X. The function specification is given as (1):

$$\cos\theta_x = \frac{(Normal_j, x)}{\| Normal_i \| \| X \|} \quad (1)$$

where, $(Normal_i, X)$ represents the inner product of $Normal_i$ and projection axis X.

Step 3: Detect the visible triangular patch by the judgment described in (2).

$$visible = \begin{cases} 1 & \cos\theta_x \geq 0 \\ 0 & \cos\theta_x < 0 \end{cases} \quad (2)$$

Step 4: Get the next triangular patch Tri_{i+1}, then turn to step 1, until all the elements in TriPatch are visited.

According to the method, a certain number of single-valued visible 3D model surfaces are obtained. Note that, this method is not only available for the circs of no concave existence, but also concave existence. After projection direction is oriented and judgment operation has been done, the concave part of the visible surface may be false-visible, that means it is actually covered by other visible surfaces. The false-visible surface is handled in subsequent steps.

Finding 2D projection contour vertexes: Here, we describe how to extract 2D projection contour vertexes after obtaining the visible surfaces.

Firstly, we extract the boundary vertexes of every single-valued space triangular mesh curved surface (Zhang *et al*., 2003). Secondly, we project the vertexes and vertex-edge-patch information that is related to those vertexes onto the given 2D plane. Through above steps, we can obtain a set of contour vertexes and a new data structure of vertex-edge-patch.

Constructing ring: After projecting triangular mesh curved surface onto 2D plane, the sampling shows irregular distribution. With the vertex-edge-patch information, we can obtain the closed rings by joining all contour vertexes.

Definition 5: If along the projection edge, the projection vertexes of the mesh surface can be connected to generate a closed curve, then the closed curve is called a ring. Ring is a set of rings.

Definition 6: A vertex is multi-forked vertex if three or more edges connect to it. Otherwise, it is a general projection contour vertex.

Definition 7: If one edge connecting to a multi-forked vertex has been visited, it is marked as visited. If all the edges connecting to a multi-forked vertex have been visited, then the vertex is marked as whole-visited.

After obtaining all 2D projection contour vertexes, we need to connect all of them to generate rings. The notation ProBdyVer is used to denote the set of 2D projection contour vertexes of a 3D model. The following steps show how to connect contour vertexes:

Step 1: Initialization. Set all contour vertexes as not-yet-visited and label multi-forked vertexes.

Step 2: If there are any not-yet-visited contour vertexes in the contour vertexes set, select any one $ProBdyVer_i$ of them and create a new ring $Ring_i$, of which the start vertex is $ProBdyVer_i$. Otherwise, perform step 4.

Step 3: Take the last element in $Ring_i$ as current vertex, find the next 2D projection contour vertex along one edge of the current vertex and add to the end of $Ring_i$, then judge as following: If the vertex is both a general projection contour vertex and the first element in $Ring_i$, label it as visited and return to step 2. Otherwise, repeat step 3. If the vertex is a multi-forked one and the first element in $Ring_i$ is a general projection contour vertex, then label the edge connecting this vertex and current vertex as visited and invert $Ring_i$, repeat step 3. If both this vertex and the first element in $Ring_i$ are multi-forked, then label the edge connecting this vertex with current vertex as visited and return to step 2.

Step 4: If there are any not-yet-whole-visited multi-forked vertexes in the contour vertexes set, selected one of them, take it as the first vertex of a new ring $Ring_i$, find the next multi-forked vertex along the not-yet-visited edge of this vertex and label the edge as visited, repeat step 4 until visited the first vertex again. Otherwise, stop.

Getting 2D projection contour by ring-relationship-based treatment: After connecting all the contour vertexes to closed rings we get the 2D projection contour, which is an important procedure in the contour extraction algorithm. The set of any two rings in Ring is described as $ComRing = \{(Ring_i, Ring_j)\}$.

Definition 8: If every vertex in $Ring_A$ is also in $Ring_B$, then we say $Ring_A$ is embedded in $Ring_B$.

Definition 9: $Ring_A$ and $Ring_B$ are intersecting if one edge of $Ring_A$ intersects one edge of $Ring_B$ at a point. We put the point into $InterVertex_{AB}$, which is defined as an intersection vertexes set of $Ring_A$ and $Ring_B$.

Definition 10: $ProBdyVer_i$ is the inner-vertex of $Ring_A$ if $ProBdyVer_i$ is inside of $Ring_A$. The inner-vertex set is $VertexInRing_A$.

The detailed algorithm of obtaining 2D projection contour shows as:

Step 1: Iterate over Ring, judge the relationship of $Ring_m$ and $Ring_n$. Delete $Ring_m$ if it is embedded in $Ring_n$.

Step 2: Choose any element ($Ring_p$, $Ring_q$) in set ComRing, perform step 3 if $Ring_p$ and $Ring_q$ are intersecting. Otherwise, repeat this step until iterate over the whole elements in the set ComRing.

Step 3: Get the intersection vertexes set $InterVertex_{pq}$, insert all elements of $InterVertex_{pq}$ into $Ring_p$ and $Ring_q$ in the corresponding position.

Step 4: Delete all vertexes and their adjacent edges, if the vertexes are both in $Ring_p$ and $VertexInRing_q$. Similarly, delete all vertexes and their adjacent edges when the vertexes are both in $Ring_q$ and $VertexInRing_p$.

Step 5: Repeat to perform the above steps until no intersection rings exist and we can get 2D projection contour.

According to the method described earlier, we can extract the 2D projection contour of a triangular mesh 3D model. Through judging the relationship of the rings and doing relevant treatment, we can elicit that, if concave surface exists in a 3D model, then false-visible surface may occur when extracting the visible surface along the given projection direction. But there are still embedment and intersection relationship between the ring of the false-visible surface and the ring of the covering surface. Therefore, the problem of false-visible surface has been solved.

RESULTS AND DISCUSSION

In order to test the contour extraction method, experiments were conducted to test the accuracy of this new solution. In the procedure of obtaining closed rings and contour, the ring-relationship-based mechanism is applicable to solve the overlapping problem of the projection boundaries. Compared to other approaches in Wu and Liu (2001) and Zhang *et al*. (2003), our method of extract 2D projection contour can be applied not only to single-valued space curved surface, but also to arbitrary complex curved surface of 3D model. Figure 1 presents the contour extraction results for some objects. The color pictures in the Fig. 1 are thumbnail images of 3D models, while the others are 2D projection contours viewed from front, top and side.

The time to determine the contour is dominated by the extraction of both visible curved surface and projection vertexes as well as handling closed rings. In the experiment, n and m are respectively the number of all patches and contour lines along the fixed projection, where (Wu and Liu, 2001). By simply theoretically analysis, $m = O(\sqrt{n})$ the maximum number of edge is 3 n. The computation cost of our method is O(n). Comparing this result with those reported in Wu and Liu (2001) of others work, it should be noted that our method outperforms all the other methods on time complexity. We have presented the time complexity results in Table 1.

Compared to previous contour extraction techniques, present method is more adaptable for it is applicable to single-valued and non-single-valued curved surface. Also, it is proved to be less time consuming. The following two reasons are introduced to explain. The first reason is that a wealth of points, edges and faces are eliminated by determine the invisibility of each curved surface. This procedure results in reducing the number of data that need processing in the successional steps.

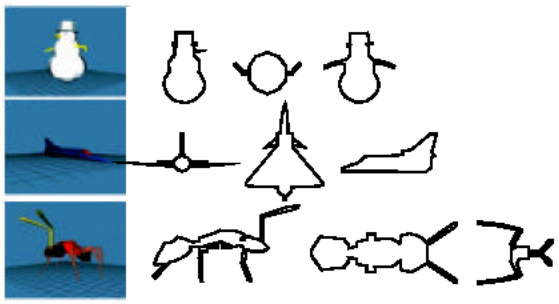

Fig. 1: The contour detection results of snowman, plance and ant

Table 1: Analysis of time complexity

Method	Time complexity
(Zhang and Hoff, 1997)	3n×11 = 33n times multiplication operations
(Wu and Liu, 2001)	3n×2 = 6n times division operations, O(m) times compare operations
Our method	3n times multiplication operations, less than n times simple judgment operations

The second reason is that, after extracting boundary vertexes of 3D model and projecting those vertexes to 2D plane, the data managed are 2-dimensional. Whereas other methods operate 3-dimensional data in the whole procedure, which first detect the boundary in 3D space, then project this boundary to 2D space and at last perform a judgment again to get the 2D contour line.

CONCLUSIONS AND FUTURE WORK

In this study we have presented a new approach based on ring relationship for the accurate, efficient and automatic extraction of 2D projection contour of triangular mesh 3D model. Our experiments indicate that the solution proposed in this STUDY provides a fast and efficient extraction of the 2D projection contour.

Currently, this method has been applied to 3D model retrieval system of the project of National Grand Fundamental Research 973 Program of China. Possible future directions involve sub-parts contour detecting, free distortion and joint of curved surface.

ACKNOWLEDGMENTS

We would like to thank the members in 973 Intelligent Design team. Supplemental support from the members of EC institute of Xi'an Jiao Tong University is acknowledged. We also thank the anonymous reviewers of this study for their making many valuable comments. We wish to thank the National Grand Fundamental Research 973 Program of China under Grant No. 2004CB719401.

REFERENCES

Funkhouser, T., P. Min, M. Kazhdan, J. Chen, A. Halderman, D. Dobkin and D. Jacobs, 2003. A search engine for 3D models. ACM Trans. Graph., 22: 83-105.

Kettner, L. and E. Welzl, 1997. Contour edge analysis for polyhedron projections. Geometric Modeling: Theory Practice, pp: 379-394.

Luebke, D. and C. Erikson, 1997. View-Dependent Simplification of Arbitrary Polygonal Environments. In: Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques. Annual Conference Series, ACM. Siggraph, W.T. (Ed.), Los Angeles: ACM Press/Addison-Wesley Publishing Co. New York, NY, USA., pp: 199-208.

Ma, K.L. and V. Interrante, 1997. Extracting feature lines from 3D unstructured grids. In: Proceedings of the 8th IEEE Visualization '97 Conference, pp: 285-292.

Markosian, L., M.A. Kowalski, D. Goldstein, S.J. Trychin, J.F. Hughes and L.D. Bourdev, 1997. Real-Time Nonphotorealistic Rendering. In: Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques. Annual Conference Series, ACM. Siggraph, W.T. (Ed.), Los Angeles: ACM Press/Addison-Wesley Publishing Co., New York, NY, USA., pp: 415-420.

Moorhead, R.J. and Z.F. Zhu, 1995. Signal processing aspects of scientific visualization. IEEE Signal Processing Magazine, 12: 20-41.

Qiu, Z.Y., X.Y. Song, S.S. Zhang, D.H. Zhang and H.C. Yang, 2004. A new method for the extraction of boundary points from scattered data points. Mechanical Sci. Technol., 23: 1037-1039.

Tan, Z. and L. Wang, 2000. Detection of edge of surface mesh data. J. Comput. Aided Design Comput. Graph., 12: 580-584.

Wu, Y.D. and Y.S. Liu, 2001. Extracting silhouettes from 3D models. J. Image Graph. A, 6: 191-194.

Xia, W.S. and X.Y. Huang, 2003. Research and realization of silhouette detecting technique for 3D modeling. J. North China Univ. Technol., 15: 11-14.

Zhang, H. and K. Hoff, 1997. Fast Backface Culling Using Normal Masks. In: Proceedings of the 1997 Symposium on Interactive 3D Graphics. Annual Conference Series, ACM. Siggraph (Ed.), Providence, Rhode Island: ACM Press, pp: 103-106.

Zhang, X., M. Zhou and G. Geng, 2003. A Method of Detecting the Edge of Triangular Mesh Surface. J. Image Graph. A, 8: 1223-1226.