Vol.16 No.6 J. Comput. Sci. & Technol. Nov. 2001

# A Sufficient Condition for a Wire-Frame Representing a Solid Modeling Uniquely

WANG Jiaye (汪嘉业)[1], CHEN Hui (陈 辉)[1] and WANG Wenping (王文平)[2]

[1]*Department of Computer Science, Shandong University, Jinan 250100, P.R. China*

[2]*Department of Computer Science, The University of Hong Kong, P.R. China*

E-mail: jywangz@jn-public.sd.cninfo.net; wenping@ns.cs.hku.hk

Received June 9, 2000; revised July 14, 2001.

**Abstract** Generally speaking, it is impossible for a wire-frame to define a 3D object uniquely. But wire-frame as a graphics medium is still applied in some industrial areas. A sufficient condition is presented in this paper. If this condition is satisfied by a wire-frame, then the wire-frame can represent a 3D object uniquely. The result is applied to manufacturing of progressive stripe.

**Keywords** wire-frame, solid modeling, CSG tree (constructive solid geometry tree), computer aided manufacturing

## 1 Introduction

There are tremendous papers on the problem of reconstructing a solid modeling from data obtained from different ways[1−7]. Although wire-frame has been applied to representing a 3D object for long time, extracting an object from a wire-frame has not been studied fully. Fig.1 is an example to prove that a wire-frame is impossible to define a 3D object uniquely. Although this assertion and the example are well known, wire-frame as a graphics medium is still applied in some industrial areas nowadays. For example manufacturing companies often receive files of wire-frame representing design of progressive stripe without any geometric and topologic description. From the point of view of manufacturing it is necessary to convert the wire-frame to a representation of solid modeling automatically. Actually the important question that must be answered is how to make the conversion feasible and unique.

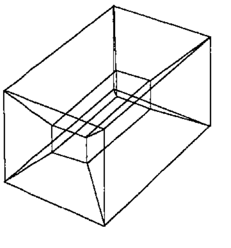


Fig.1. An example.

A sufficient condition of defining a 3D object by a wire-frame uniquely is proved in this paper. To check that if a wire-frame satisfies the condition is always realizable. We prove that this condition is satisfied by the wire-frame of progressive stripe design. In this paper we also present an algorithm to create a CSG tree from a wire-frame.

## 2 Basic Theory

There is no strict definition of a wire-frame for an object. There are only some common understandings in product design field. Summarizing the common understandings, we specify the following definition.

**Definition 1.** *A wire-frame represents an object in a system, if the following conditions are satisfied.*

(1) *The line segments of the wire-frame must be totally on the boundary of the object.*

(2) *From the wire-frame a sequence of polyhedrons can be recognized, and they do not have common inside points. The object can be the union of some of the polyhedrons.*

This work is supported by the National Natural Science Foundation of China (No.6612551).

(3) *The polyhedrons are bounded by polygons called boundary polygons of the polyhedrons. The sides of the polygons can be curves that the system declared, but they are "almost straight line seg- ments". It means that replacement of the curves by straight-line segments linking two end vertices of the curves will not effect any changes of the topological structure of the polygons and relative polyhedrons. Every polygon must be a simple polygon, and has at least three sides.*

(4) *The polyhedron has the following property: If straight-line segments replace the sides of the boundary polygons and the polygons are triangulated in case the vertices of the polygons are not on a plane, it will not make any topological change of the polyhedrons and the object. Every polyhedron has at least four boundary polygons.*

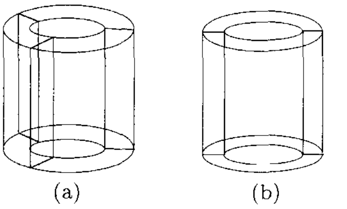


Fig.2. (a) Acceptable wire-frame. (b) Not acceptable wire-frame.

Fig.2(a) is an acceptable wire-frame of an object. But Fig.2(b) is not acceptable.

From the above definition of wire-frame to recognize an object, the primary step is to find all polyhedrons in the wire-frame. Firstly the polygons in the wire-frame are recognized. According to the relationship of two adjacent polygons, a polyhedron can be assembled polygon by polygon gradually. Two polygons are adjacent, if they share one edge. In case the wire-frame can define only one object, and we can find the combination of polyhedrons to represent the wire-frame, to create the object is a simple matter of gluing all the adjacent polyhedrons in the combination one by one. Thus, a more important problem is the uniqueness of the object recognized. It is possible that there are different combinations of the polyhedrons recognized from the wire-frame, and the union of each combination can produce the original wire-frame, but the unions of the polyhedrons of the combinations represent different objects.

Although a wire-frame can have different understandings. For example, the wire-frame in Fig.1 can have three understandings to present three different 3D objects. It is assumed that for each understanding the wire-frame only represents one 3D object. Here, in Fig.1 every understanding of the wire-frame is a pierced rectangular box.

If there are two different objects that both of them can be represented by the same wire-frame, there is at least one polyhedron $P$ of the polyhedrons found from the wire-frame. $P$ belongs to the space occupied by one object, but it does not belong to the space occupied by another object. The polyhedron $P$ is called non-unique caused polyhedron (NUCP) in this paper.

A polyhedron $P_1$ is inserted in a polyhedron $P_2$, if all the edges of $P_1$ are the edges of $P_2$. In the following description we always assume that the polyhedrons are recognized from a wire-frame. In Figs.3(a) and 3(b), the polyhedrons whose vertices are circled are inserted in the big polygons.

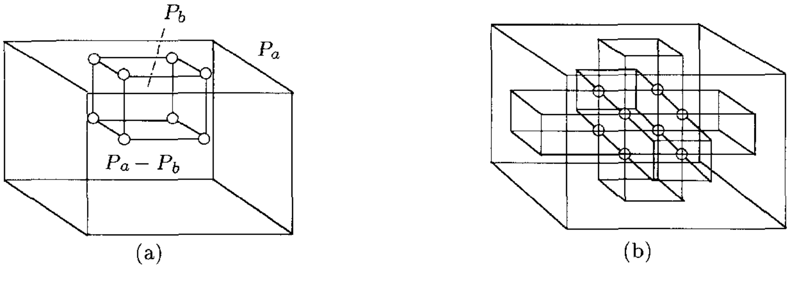



Fig.3. (a) There is no NUCP. (b) There is an NUCP.

**Theorem 1.** *If $P_1$ is inserted in a polyhedron $P_2$, if and only if the edges of $P_1$ is also the edges of the union of $P_1$ and $P_2$, $P_1$ is an NUCP.*

The proof is trivial.

Theorem 1 is useful to check if a polyhedron inserted in another polygon is an NUCP. In Fig.3(b) two polyhedrons can be found. One is a big box pierced by three thin rectangular boxes, the other is a small box being the intersection of the three thin rectangular boxes. The vertices of the small box are circled. We can find that the union of the small box and the big box pierced by the three rectangular boxes still have the edges whose ends are circled. From Theorem 1 the small box is an NUCP. Actually the wire-frame in Fig.3(b) can be understood to present two different 3D objects. One is the big box pierced by the three rectangular boxes. The other is the big box pierced by the three rectangular boxes and the small box inside. Let $P_a$ denote the larger polyhedron and $P_b$ the polyhedron whose vertices are circled shown in Fig.3(a). The polyhedrons recognized are $P_b$ and $P_a - P_b$. Since the union of $P_b$ and $P_a - P_b$ is $P_a$, the sides where the end vertices are circled will disappear in the union of $P_b$ and $P_a - P_b$. It is therefore that $P_b$ is not an NUCP.

**Definition 2.** *Two polyhedrons are adjacent, if they have a common boundary polygon.*

Now we create a graph called structure graph for a wire-frame. Without losing generality, we can assume that every polyhedron recognized is a convex polyhedron, otherwise it can be divided into several convex polyhedrons by some virtual polygons. Define every polyhedron recognized as a graph vertex. Link the centers (or any point in the polyhedron) of every two adjacent polyhedrons and the polyhedrons share one edge. The linking line segments are defined as sides of the graph (Fig.4(b)).

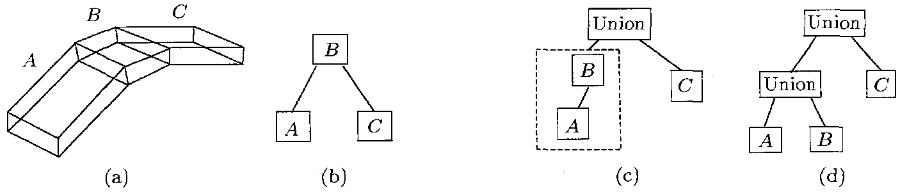
**Theorem 2.** *The sufficient condition that a wire-frame can be recognized uniquely is: There are no rings in the structure graph of the wire-frame, i.e., the construction graph is a tree.*

*Proof.* It is only necessary to prove that there is no NUCP, i.e., if we remove any polyhedron, the union of the rest polyhedrons cannot be a single object or the object cannot be represented by the original wire-frame. Since all sub-trees with the same parent node in the construction graph cannot be adjacent, if we remove any non-leaf node polyhedron from the construction graph, the object represented by the wire-frame will be separated. The wire-frame will represent more than one object. It contradicts the fact that the wire-frame only represents one object.

Since a leaf node polyhedron can only have a common polygon or a common side with its parent polyhedron, it must have some sides not common with other polyhedron. If a leaf node polyhedron is removed, those sides being not common with other polyhedron will disappear from the union of the rest polyhedrons. It means the union of the rest polyhedrons cannot be represented by the original wire-frame. □

If the condition of Theorem 2 is satisfied, the object created by gluing all polyhedrons recognized is the unique object represented by the wire-frame.

Once we have the structure graph of a wire-frame of an object, a CSG tree of the object can be created. If there are rings in the graph, delete one of the edges of every ring to make the graph to be a tree. Then the sides of the structure graph are changed one by one into the nodes with operation "Union" of a CSG tree. For example, Fig.4(b) is the structure graph of the wire-frame of Fig.4(a). Firstly we change the side $BC$ into a node of the CSG tree. The two trees linked by the side $BC$ become two subtrees of the new CSG node (Fig.4(c)), and two pointers of the node point to the two subtrees. If the subtree has only one vertex, the CSG node representing the subtree is a leaf node with the polyhedron in the vertex of the subtree as the CSG node primitive (node $C$ in Fig.4(c)). If the subtree has more than one vertex, recursively apply the above procedure (Fig.4(d)), and the eventual CSG tree can be created. The transformation matrixes in the CSG node are always a unit matrix.

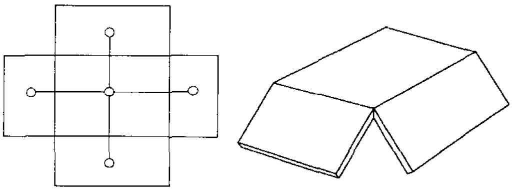



(a) (b) (c) (d)

Fig.4. (a) A wire-frame. (b) Structure graph. (c) Change side BC. (d) CSG tree.

## 3 An Application

Although a progressive stripe is produced by cutting and bending at the same time step by step, the procedure is equivalent to cutting first and bending second (Fig.5). Since after bending (Fig.5(b)) the topologic structure of the object is not changed, the structure graph of the cut plate (Fig.5(a)) is the same as the bent plate (Fig.5(b)). It is obvious that the structure graph of a cut plate (Fig.5(a)) is a tree, so is the bent plate. From Theorem 2 the wire-frame of a progressive stripe always can be recognized uniquely.



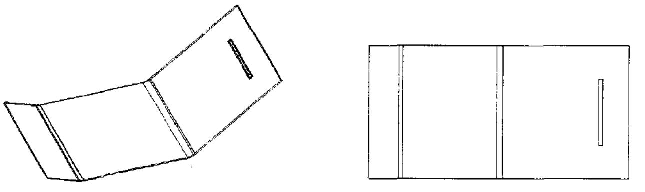
(a) (b)

Fig.5. (a) Flatten plate. (b) Bent plate.

Although there are some forms and slots on a progressive stripe, the forms and slots are isolated and attached to some planes of the bent plate. They are not difficult to be recognized separately.

## 4 Example

We developed a prototype system of converting the wire-frame of progressive stripe to a representation of solid modeling automatically. Fig.6(a) is the original wire-frame of a progressive stripe. After recognizing, a solid modeling is created. Fig.6(b) shows the flatten result of the modeling.



(a) (b)

Fig.6. (a) The original wire-frame. (b) The flatten progressive stripe.

## 5 Conclusions

To study the conditions of defining a 3D object by a wire-frame uniquely is not only an interesting theoretical problem, it also has important application background. In this paper a sufficient condition of uniquely defining an object from a wire-frame is presented. It proves that wire-frame can be applied to transferring the design and manufacturing information of progressive stripes. To seek more practical sufficient conditions and eventually to find necessary and sufficient conditions are our further interest.

## References

[1] Hamid A, Kailath T. Subspace-based line detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 1994, 16(11): 1057–1073.

[2] Tamas Varady, Ralph R Martin, Jordan Cox. Reverse engineering of geometric models — An introduction. *Computer-Aided Design,* 1997, 29(4): 255–268.

[3] Zhang Kuo-liang. Computing horizontal/vertical convex shape's moments on reconfigurable meshes. *Pattern Recognition,* 1996, 29(10): 1713–1717.

[4] Mantyla M. An Introduction to Solid Modeling. Computer Science Press, 1988.

[5] Sugihara K. Machine Interpretation of Line Drawings. MIT Press, 1986.

[6] Chivate P N, Jablokow A G. Solid-modelling generation from measured point data. *Computer-Aided Design,* 1993, 25(9): 587–600.

[7] Hoover A, Goldgof D, Bowyer K W. Extracting a valid boundary representation from a segmented range image. *IEEE Trans. Pattern Analysis and Machine Intelligence,* 1995, 17(9): 920–924.