A Convex Hull Algorithm and Its Application to Shape Comparison of 3-D Objects

Susumu FURUKAWA, Shinji MUKAI** and Mitsuru KURODA+*
** Department of Mechanical Systems Engineering, Yamanashi University, Takeda 4-3-11, Kofu 400, Japan, Tel. (+81) 552 20 8576, Fax (+81) 552 20 8771, email : furukawa %sun3.esb.yamanashi.ac.jp@yu-gate.yamanashi.ac.jp*
*** Maebashi City College of Technology, Kamisadori-machi 460, Maebashi 371, Japan, Tel. (+81) 272 65 0111, Fax (+81) 272 65 3837*
+ Toyota Technological Institute, Hisakata 2-12, Tempaku, Nagoya 468, Japan, Tel. (+81) 52 802 1111, Fax (+81) 52 802 6069, email : M.Kuroda@toyota-ti.ac.jp

Abstract

A new method for constructing the convex hull of a concave polyhedron represented by B-Reps is presented as a basic tool for feature processing of 3-D objects. The algorithm is based on the idea of taking out faces which are used for constructing the convex hull and making new faces to cover gaps between the faces taken out from the polyhedron. It is shown that constructing convex hulls of polyhedra can be applicable to deciding similarities of 3-D objects numerically. Various examples of concave polyhedra for determining similarities have been tested. The results show that the theoretical considerations are reasonable and matches with our feeling.

Keywords

Convex Hull Algorithm, Convex Decomposition, Feature Recognition, Shape Comparison, Solid Model, CAD/CAM

1 INTRODUCTION

At the final stage of manufacturing processes, checking whether products are made within specified accuracy is required to make high quality products. It is then necessary to detect the difference between the design specification and the manufactured products[Gra 92, Furu 94]. Specifically, deciding similar and dissimilar parts between a solid model and its product is required. Thus, developing methods for dealing with features, tolerances and similarities of solid models is necessary. Treating such geometrical properties are also required in process planning, parts assemblies in CAD/CAM processes. Therefore, there are many research works in this field.

Approaches to specifying the features of objects explicitly during the design process have been proposed[Cut 88, Sha 88]. The main advantage of this approach is that the designer can immediately build models which include the necessary information in design,

Q. Sun et al. (eds.), *Computer Applications in Production Engineering*

manufacturing and checking. However, the problem with this approach is that users have to define every features prior to each process.

Another way of treating geometrical properties of objects is by feature recognition[Fal 89, Saku 90, Kim 92]. In this case, the designer need not take into account form features at his design work. The main problem with this approach is that only features which are implicitly stored in the data base can be derived.

Combining these two approaches and the system architecture have been proposed[Dix 90, Mar 93]. However, better methods for recognizing features of objects is required in order to reduce the user's load.

Algorithms for determining congruencies of polyhedra[Furu 90] and deciding simlarities of convex polyhedra[Muka 94-1] have been developed and an idea for determining similarities of concave polyhedra[Muka 94-2] has been proposed by the authors. The idea of computing similarities of concave polyhedra proceeds as follows. First, the concave polyhedra to be compared are decomposed into convex components. Then each convex component of one polyhedron is compared with the corresponding component of another polyhedron, and their similarities are computed numerically.

In this paper, a new method for constructing the convex hull of a concave polyhedron represented by B-Reps is presented first as a basic tool for decomposing a polyhedron into convex components. The algorithm utilizes the fundamental properties of a convex polyhedron.

Next, with this convex decomposition, a data structure called two layer data structure of convex components with relationships is discussed briefly. As an example of its applications, similarity tests have been achieved for polyhedra. The results show that the theoretical considerations on this problem are reasonable and matches with our feeling.

2 PREVIOUS RESEARCH WORK

The convex hull algorithm may be applied not only to feature extraction but also in many other fields such as computer graphics, pattern recognition and operations research.

Various methods for computing the convex hull have been developed for 2-dimensional space[Gra 72, Jar 73, Sei 86]. Preparata[Pre 77] and Furukawa[Furu 86] have discussed the methods for given points in 3-D space.

Preparata presented a method which uses the "divided and conquer" technique, recursively merging two nonintersecting convex hulls for points in 3-D space. The convex hulls can be constructed in $O(n\log n)$, where n represents the number of points.

Furukawa developed the method which is based on the idea of determining the outline loop of a convex polyhedron viewing from a point and connecting the point with each vertex of the outline loop. A convex hull polyhedron can be constructed with $O(n)$ operations, if the points are distributed uniformly.

However, methods for constructing convex hulls from solid models already described by a data structure such as B-Reps are not well approached. One method has been developed by Appel[App 77] for a polyhedron for which the data structure is already constructed. This is based on the idea of testing whether a vertex is inside the specified polygon or not, on a projection plane. The estimation of time requirement, however, was not discussed.

3 CONVEX HULL ALGORITHM FOR POLYHEDRA

As the basis for constructing a data structure called "two layer structure" of a polyhedron which can be used for extracting form features, a method for constructing the convex hull of a concave polyhedron represented specified data structure such as B-Reps is required. At the first process for constructing the two layer structure, the convex hull of the concave polyhedron is constructed and faces of the polyhedron are classified into the following three types, i.e., the used and unused faces, and the faces of which parts are used.

3.1 Fundamental properties of convex polyhedra

Before discussing the convex hull algorithm, fundamental properties of convex polyhedra are discussed.

Consider the convex polyhedron shown in Fig.1. Let G_c be a point inside polyhedron P, and R_i be a open region like to a bottomless pyramid surrounded by faces S_j constructed by point G_c and edge line L_j of face F_i for j=1,2,...,m_i, where m_i represents the number of edges of face F_i.

[Property 1]

If polyhedron P is convex, regions R_i and R_k have no intersection for all i and k (i, k=1,2,...,n_f ; $i \neq k$), where n_f represents the number of faces of the polyhedron.

[Property 2]

Every interior angles between two faces sharing an edge is less than π.

3.2 Visibility test of faces

Consider a fundamental principle for constructing the convex hull of a polyhedron P in this section.

We discuss visibilities of faces of the polyhedron viewing from a point p on a face F_i of

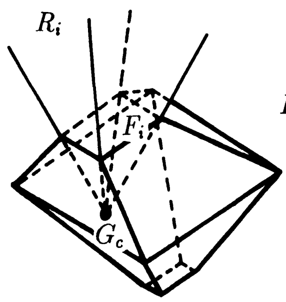


Figure 1: A convex polyhedron and open region.

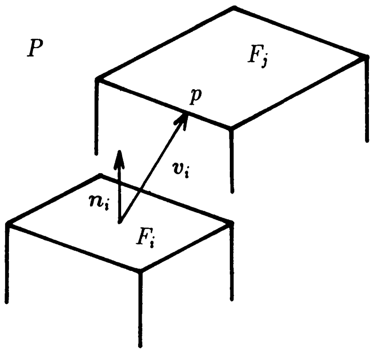


Figure 2: Visibility test of a face.

P. Let $\boldsymbol{n}_i$ and $\boldsymbol{v}_i$ be an outward normal vector of a face F_i and a vector starting with a point on F_i to point p, respectively. The visibility of F_i can be easily determined as follows. If $\boldsymbol{n}_i \cdot \boldsymbol{v}_i > 0$, then F_i is visible from point p. Otherwise, F_i is invisible.

If F_i is visible from a point on a face F_j of P, then face F_i is not used to construct the convex hull of P, because F_i should be inside the resulting convex hull of P[Furu 86](Fig.2).

3.3 Intersection check of regions

We designate the resulting convex hull polyhedron $P^{(c)}$.

Consider the case where two regions R_i and R_j sharing an edge, which are constructed by a face F_i and point G_c, and F_j and G_c respectively, intersect each other. We can classify the relationship between faces F_i and F_j into the following two cases.

Case (1) The interior angle θ of two faces F_i and F_j is less than π.

Case (2) Angle θ is greater than π.

Let $\boldsymbol{g}_i$ be a vector starting with a point on face F_i to point G_c.

In case (1), one of the inequalities

$$\boldsymbol{n}_i \cdot \boldsymbol{g}_i > 0, \quad \boldsymbol{n}_j \cdot \boldsymbol{g}_j > 0 \qquad (1)$$

is valid(Fig.3), where $\boldsymbol{n}_i$ represents an outward normal vector of F_i defined in the previous discussions. Assuming $\boldsymbol{n}_i \cdot \boldsymbol{g}_i > 0$, then F_i is not used for constructing the convex hull $P^{(c)}$. Hereafter, such a face is called unused face. In this case, we need not further test of intersections between R_i and the other regions.

In case (2), two faces F_i and F_j are both unused faces.

The edge sharing by F_i and F_j is called a folded edge in this case.

Consider the case where R_i and R_j sharing an edge have no intersection. There are two cases :

Case (3) $\boldsymbol{n}_i \cdot \boldsymbol{g}_i < 0$ and $\boldsymbol{n}_j \cdot \boldsymbol{g}_j < 0$, (2)

Case (4) $\boldsymbol{n}_i \cdot \boldsymbol{g}_i \geq 0$ and $\boldsymbol{n}_j \cdot \boldsymbol{g}_j \geq 0$. (3)

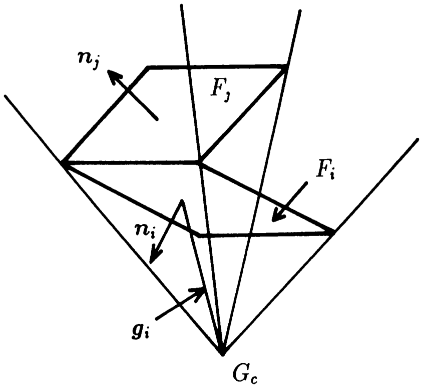


Figure 3: The case where two succesive regions intersect and the interior angle of the faces is less than π.

In case (3), there are possibilities that both faces F_i anf F_j are used to construct the convex hull.

In case (4), two faces F_i and F_j are both unused faces.

Hereafter, this intersection test of regions is called "region check".

3.4 Algorithm for constructing convex hulls of polyhedra

We consider a new algorithm for constructing the convex hull from a polyhedron P.

The faces of the resulting convex polyhedron are classified into the following three types.

(a) Real face : the same face also exists in polyhedron P.
(b) Partial real face : a face including a face of P.
(c) Imaginary face : a face not existing in P.

The procedure for constructing the convex hull $P^{(c)}$ of P is as follows.

(1) Outward normal vectors of all faces are calculated.

(2) A point inside the polyhedron G_c is determined.

(3) Find the most distant vertex v_0 of P from point G_c. This vertex becomes to a vertex of $P^{(c)}$.

(4) Construct a face $F_0^{(c)}$ including vertex v_0 by using so called "gift wrapping" method[Cha 70], where $F_0^{(c)}$ represents a face of $P^{(c)}$. At this stage, whether $F_0^{(c)}$ is a real face or not is detected.

(5) Make the region of "bottomless pyramid" R_0 defined in section 3.1, by using G_c and $F_0^{(c)}$.

(6) Select a face F_i sharing an edge of $F_0^{(c)}$ and make region R_i from G_c and F_i. If the edge is a folded edge, the relation between F_i and $F_0^{(c)}$ corresponds to case (1) in

section 3.3. Then, face F_i is unused face. Otherwise, there is possibility that face F_i is used for constructing $P^{(c)}$.

This procedure is performed for all edges of face $F_0^{(c)}$. Then faces $F_j(j=1,2,...,k)$ which have possibilities to become faces of $P^{(c)}$ and corresponding region R_j are obtained, where k represents the number of such faces.

Those regions are stored in a list, which is called "region table", for intersection tests in the following step.

(7) The following procedures are performed for all faces sharing edges with faces $F_0^{(c)}$, $F_1,...,F_k$. Let F_i be one of the faces F_1, $F_2,...,F_k$ and F_j be a face sharing an edge with F_i. If the relation between F_i and F_j corresponds to case (1), then F_j is an unused face. If it corresponds to case (2), both faces F_i and F_j are unused faces. In this case, region R_i corresponding to face F_i is removed from the region table. At the same time, region checks for region R_j and the regions written in the region list have to be achieved.

The procedure for removing regions from the region table is continued till no intersection is detected.

If the relation corresponds to case (3), then region R_j is added in the region table. At this time, the "angle test" whether the interior angle of faces F_i and F_j is less than π or not is performed. If the angle is greater than π, faces F_i and F_j are unused faces and region R_i is removed from the region table.

(8) Consider whether faces sharing edges with unused faces may be used to construct the convex hull or not.

Let F_i be an unused face and F_j be a face to be tested.

Consider the case where case (1) is valid for F_i and F_j. If F_j is nearer than F_i to point G_c, then F_j is an unused face. Otherwise, region R_j is added into the region table.

In case (2), face F_j is an unused face.

If the relation between F_i and F_j corresponds to case (3) or (4), further checking of intersections between region R_j and regions written in the region table is necessary.

First, we consider the case where region R_j has intersection with region R_l corresponding to face F_l. In both case (3) and (4), F_j is an unused face. In case (4), if F_j is nearer than F_l to point G_c, region R_l is removed from the table. The same discussion is valid for the faces of which regions intersect with R_j.

Next, consider the case where the regions have no intersection. If case (3) is valid for F_i and F_j, region R_j is stored in the region table. Otherwise, face F_j is an unused face.

For faces sharing edges with the faces of which regions are stored in the region table, the same procedure as (7) is applicable.

This procedure (8) is continued until all faces of polyhedron P are tested.

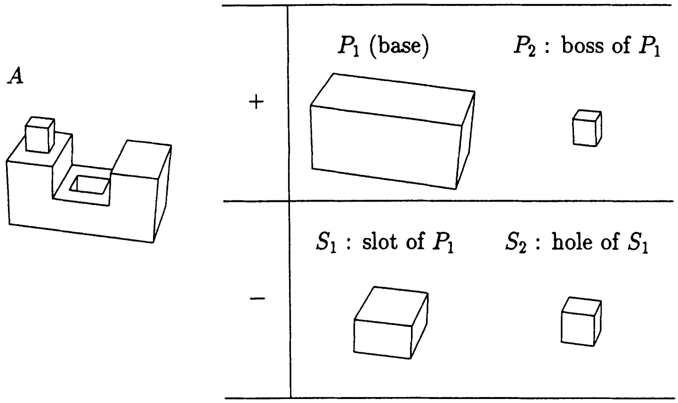


Figure 4: An example of a two layer structure of object A.

(9) At the last step, visibilities are tested for each couple of faces corresponding to neighbouring regions stored in the region table. If both faces are invisible each other, then an imaginary face is construct to connect these two faces. On the contrary, in the case where the faces are visible or one of them is visible, the faces become unused faces and an imaginary face is also constructed.

4 SHAPE COMPARISON OF TWO OBJECTS

Before discussing similarities of two polyhedra, a concave polyhedron is decomposed into convex components[Furu 85, Furu 86], using convex hull algorithm described in the previous section, and a data structure called "two layer structure of convex components with relationships" is constructed for each object.

An example of such a data structure is shown in Fig.4, wherein convex polyhedra belonging to the first layer represent the parts to be added, while those in the second layer represent the parts to be subtracted. Thus object A is given by $P_1 + P_2 - (S_1 + S_2)$.

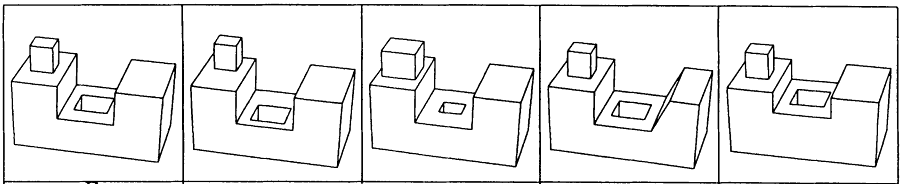
Although this data structure seems to be same to the hierarchical tree structure presented in [Furu 85], it is more flexible and easier to understand shapes of objects.

The data structure includes the relationships between those convex components too. For example, object P_2 belongs to P_1 as an added part and object S_1 is subtracted from P_1.

By using this type of data structure, we can discuss how to determine similarities of two concave polyhedra. The basic algorithm of shape comparison is as follows.

Suppose that the two layer data structures of given concave objects A and A' are both represented as shown in Fig.4 and only shapes of components to be compared are slightly different from each other.

At first, the bases P_1 and P_1' of objects A, A' are compared and the similarities are calculated according to the procedure described in [Muka 94-1].



P_1	$\mu = 0.000$ $\sigma = 0.000$	$\mu = 0.000$ $\sigma = 0.000$	$\mu = 0.000$ $\sigma = 0.000$	$\mu = 0.000$ $\sigma = 0.000$
P_2	$\mu = 0.000$ $\sigma = 0.000$ $\delta_G = 0.179$	$\mu = 0.390$ $\sigma = 0.097$ $\delta_G = 0.000$	$\mu = 0.000$ $\sigma = 0.000$ $\delta_G = 0.000$	$\mu = 0.000$ $\sigma = 0.000$ $\delta_G = 0.000$
S_1	$\mu = 0.000$ $\sigma = 0.000$ $\delta_G = 0.000$	$\mu = 0.000$ $\sigma = 0.000$ $\delta_G = 0.000$	$\mu = 0.134$ $\sigma = 0.063$ $\delta_G = 0.045$	$\mu = 0.189$ $\sigma = 0.099$ $\delta_G = 0.089$
S_2	$\mu = 0.000$ $\sigma = 0.000$ $\delta_G = 0.119$	$\mu = 0.283$ $\sigma = 0.087$ $\delta_G = 0.000$	$\mu = 0.000$ $\sigma = 0.000$ $\delta_G = 0.000$	$\mu = 0.294$ $\sigma = 0.074$ $\delta_G = 0.089$

Figure 5: Example of similarity tests.

Next, shapes of P_2, P_2', S_1, S_1' and S_2, S_2' are compared respectively. Then, for example, we may see that two bases P_1 and P_1' are equal, P_2 and P_2' are equal too, but the slots S_1, S_1' and the holes S_2, S_2' are slightly different.

In the case where the data structures of two concave objects are different, we need not compare each pair of convex components, because it is clearly understandable that they are quite different shapes.

An example of this similarity test is shown in Fig.5. The parameters written in the diagram represent numerals of similarities, i.e., if μ and σ equal zero, then corresponding components are congruent, and if δ_G=0, the position of the components are coincident. Therefore, objects tend to exhibit higher similarities, those parameters decrease.

5 CONCLUSIONS

A new algorithm for constracting the convex hull of a polyhedron represented by B-Reps is presented.

Using this algorithm, a data structure called two layer structure which consists of only convex components is constructed and it is applied to deciding similarities of concave polyhedra.

The main features of the paper are as follows.

1. The convex hull algorithm uses fundamental properties of convex polyhedra. Therefore, it is very simple and easy to understand.

2. After constructing the convex hull polyhedron, we need not to check whether faces of the convex hull exist also in the original polyhedron, because necessary informations on those faces are stored while the convex hull is constructed.

3. The results of similarity tests are reasonable and match with our feeling.

REFERENCES

[Cha 70] Chand D.R. and Kapur S.S. : *An Algorithm for Convex Polytopes*, J. ACM, vol.17, no.1, pp.78-86 (1970)

[Cut 88] Cutkosky M.R., Tenenbaum J.M. and Muller D. : *Features in Process-Based Design*, Proc. Comp. Eng., ASME, pp.557-562 (1988)

[Dix 90] Dixon J.R., Libardi E.C. and Nielsen E.H. : *Unresolved Research Issues in Development of Design-with-Features System*, In Geometric Modeling for Product Engineering, North-Holland, pp.183-196 (1990)

[Fal 89] Falcidieno B. and Giannini F. : *Automatic Recognition and Representation of Shape-Based Features in a Geometric Modeling System*, Computer Vision, Graphics, and Image Processing, vol.48, pp.93-123 (1989)

[Furu 86] Furukawa S. : *An Efficient Convex Hull Algorithm for Finite Points Set in 3-D space*, Technical Report of M41 Group in Yamanashi University, no.1, pp.1-7 (1986)

[Furu 85] Furukawa S. : *An Algorithm for Constructing Hierarchical Tree Structures of Polyhedra*, Journal of JSPE., vol.51, no.11, pp.2071-2076 (1985) (in Japanese)

[Furu 90] Furukawa S., Kimura F., Sata T. and Obi M. : *An Algorithm for Deciding Congruence of Polyhedra*, Bulletin of the JSME., vol.56, no.528, pp.2311-2317 (1990) (in Japanese)

[Furu 94] Furukawa S. : *Computer Aided Checking of Products*, Proceedings of Intellectual Facilitation of Creative Activities, SCF., pp.52-57 (1994)

[Gra 72] Graham R.L. : *An Efficient Algorithm for Determining the Convex Hull of a Finite Planar Set*, Information Proc. Letter, pp.132-133 (1972)

[Gra 92] Grabowski H., Ander R., Geiger K. and Schnitt M. : *Vision Based On-Line Inspection of Manufactured Parts*, Human Aspects in Computer Integrated Manufacturing, ed. by G.J.Olling and F.Kimura, Elsevier Science Pub., pp.593-607 (1992)

[Jar 73] Jarvis R.A. : *On the Identification of the Convex Hull of a Finite Set of Points in the Plane*, Information Processing Letter, no.2, pp.18-21 (1973)

[Kim 92] Kim Y.S. : *Recognition of Form Features Using Convex Decomposition*, Computer-Aided Design, vol.24, no.9, pp.461-476 (1992)

[Mar 93] Martino T., Falcidieno B., Giannini F., Haßinger S. and Ovtcharova J. : *Integration of Design by Features and Feature Recognition Approaches Through a Unified Model*, Modeling in Computer Graphics, Springer-Verlag, pp.423-437 (1993)

[Muka 94-1] Mukai S., Furukawa S., Kimura F. and Obi M. : *An Algorithm for Determining Similarities of Convex Polyhedra*, Computers & Graphics, vol.18, no.2, pp.171-176 (1994)

[Muka 94-2] Mukai S., Furukawa S. and Kuroda M. : *A Method for Determining Similarities of Polyhedra using Form Feature Decomposition*, The 49th Annual Conference of Japan Society for Information Processing, pp.2·393-2·394 (1994) (in Japanese)

[Pre 77] Preparata F.P. : *Convex Hulls of Finite Sets of Points in Two and Three Dimensions*, CACM., vol.20, no.2, pp.87-93 (1977)

[Pre 85] Preparata F.P. and Shamos M.I. : *Computational Geometry*, An Introduction, Springer-Verlag, (1985)

[Saku 90] Sakurai H. and Gossard D.C. : *Recognizing Shape Features in Solid Models*, IEEE Computer Graphics & Applications, vol.10, no.5, pp.22-32 (1990)

[Sei 86] Seidel R. : *Constructing Higher-Dimensional Convex Hull at Logarithmic Cost per Face*, ACM, pp.404-413 (1986)

[Sha 88] Shah J.J. and Rogers M.T. : *Functional Requirements and Conceptual Design of the Feature-Based Modeling System*, Computer-Aided Engineering Journal, pp.9-15 (1988)