Recognition of pass features for automatic parting surface generation in injection moulds

K Chung[1], **K Lee**[2]* and **T Kim**[3]
[1]R&D Team, INUS Technology, Korea
[2]School of Mechanical and Aerospace Engineering, Seoul National University, Korea
[3]Department of Digital Contents, Sejong University, Korea

Abstract: This paper proposes a topology-based algorithm for recognizing passage features using the concept of a *multiface hole loop*. The multiface hole loop is a conceptual hole loop that is formed over several connected faces and serves as an entrance or an exit of a passage. A passage feature is thus recognized by identifying two multiface hole loops corresponding to its entrance and exit. To generate the core and the cavity of an injection mould for a part with passage features, either the entrance or the exit of each passage must be covered by a surface, and this surface constitutes the parting surfaces. The algorithm proposed in this paper checks the connectivity of the two multiface hole loops to recognize passage features. The total number of passage features in a part is calculated from the Euler equation, and the identification procedure continues until the number of identified passage features equals this number. To find all of the multiface hole loops in a part, the proposed approach considers all of the combinations of connected faces. The edge convexity is used to judge the validity of multiface hole loops. By using the algorithm proposed in this paper, the passage features could be recognized effectively. The approach proposed is illustrated with several example cases.

Keywords: passage feature, feature recognition, multiface hole loop, injection mould

1 INTRODUCTION

Injection moulding is the most prevalent process for the production of thermoplastic polymer parts. In this process, a thermoplastic polymer is injected into the cavity formed by the core and the cavity. This mould cavity determines the shape of the plastic part. The general configuration of an injection mould is shown in Fig. 1.

Figure 1 represents a simplified configuration of an injection mould and illustrates the plates, the core and the cavity. The actual injection mould has many standard parts, ejectors, a slide system and a cooling system, in addition to those shown. Historically, mould designers have generated the two-dimensional drawings of a mould using a two-dimensional computer aided design (CAD) system. Recently, however, attempts to design injection moulds using three-dimensional CAD systems have been made as three-dimensional CAD systems running on personal computers have become widely available.

By using a three-dimensional CAD system for mould design, many design tasks can be automated or facilitated. In particular, the parting surfaces can be generated automatically or at least much more conveniently [**1**–**4**]. Parting surfaces are used to cut the external block enclosing the part to be moulded into two pieces, i.e. the core and the cavity. When the part has passages, either their entrances or exits have to be covered by the parting surfaces. Even though the parting surfaces are generated fairly easily by extending the parting lines towards the outside of the part, identifying all the passage features and covering them with proper surfaces are error-prone tasks when the shape of the part is complicated. In this paper, an algorithm for finding such passage features passing through multiple faces, as shown in Fig. 2, is proposed.

2 BACKGROUND AND MOTIVATION

In injection mould design systems that are based on three-dimensional CAD systems, the design process

The MS was received on 21 May 2001 and was accepted after revision for publication on 21 December 2001.
**Corresponding author: Department of Mechanical and Aerospace Engineering, Seoul National University, San 56-1, Shillim-Dong, Kwanak-Gu, Seoul 151-742, Korea.*

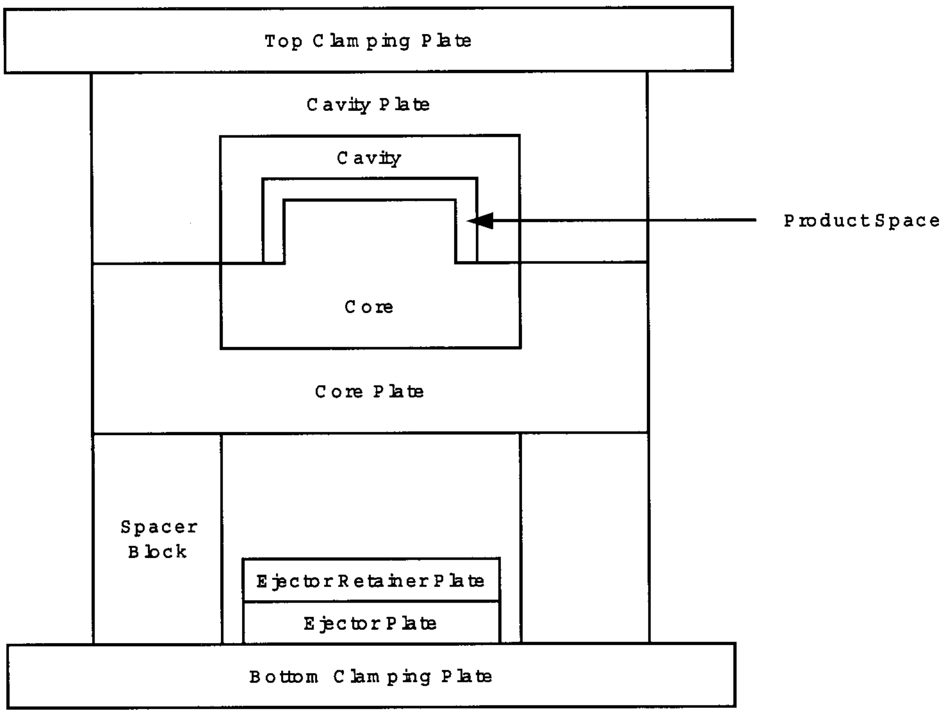


Fig. 1 General configuration of an injection mould

starts from the three-dimensional CAD model of a part. Then, the core and the cavity are created from the part model. The procedure is shown in Fig. 3. The core and the cavity of the part model shown in Fig. 4 are generated as follows.

Firstly, a designer creates a cubic block enclosing the part model, as illustrated in Fig. 5. The rectangular block is called the core block, which will eventually become the core and the cavity. Next, the parting surfaces, which split the core block into the core and the cavity, are generated along the outer boundaries of the part model as shown in Fig. 6a. The parting surfaces, if required, are also generated on one end of the passage feature as shown in Fig. 6b.

The core block is split into the core and the cavity by the parting surfaces and the faces of the part model. The core is the lower portion of the core block, and the cavity is its upper portion. The created core and cavity are illustrated in Fig. 7.

As illustrated in Fig. 6b, the parting surfaces should be generated on one end of each passage feature to split the core block completely into two pieces, i.e. the core and the cavity. In the case of a simple passage feature, as shown in the example, the parting surface of the passage feature is easily generated. However, in the case of complicated passage features formed over several faces as illustrated in Fig. 8, the recognition of the loops bounding the ends of the passage feature and the

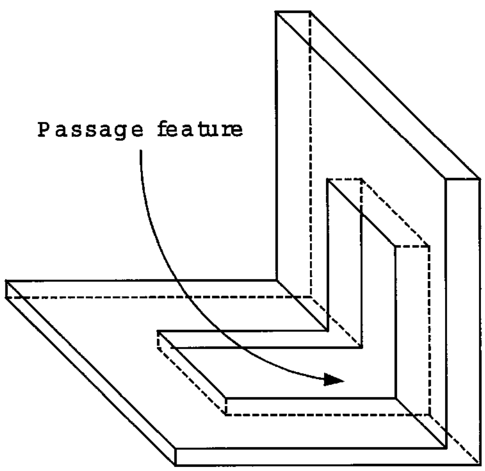


Fig. 2 Passage feature passing through multiple faces

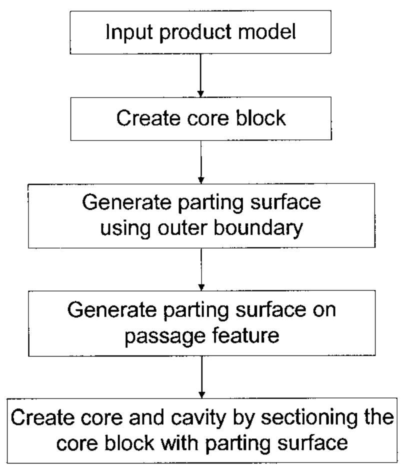
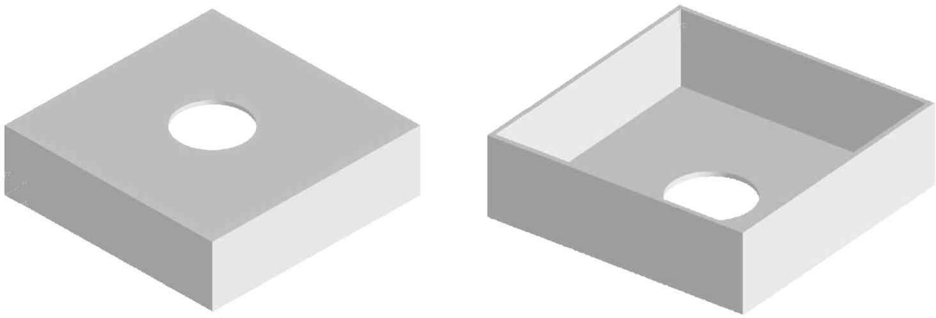


Fig. 3 Procedure for generating the core and the cavity



(a) (b)

Fig. 4 Top and bottom views of an example part with one passage feature

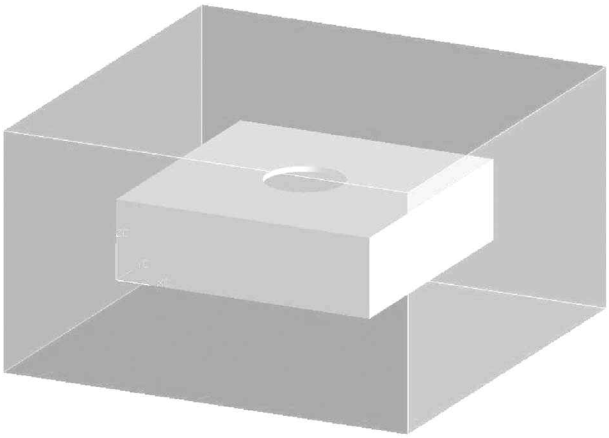
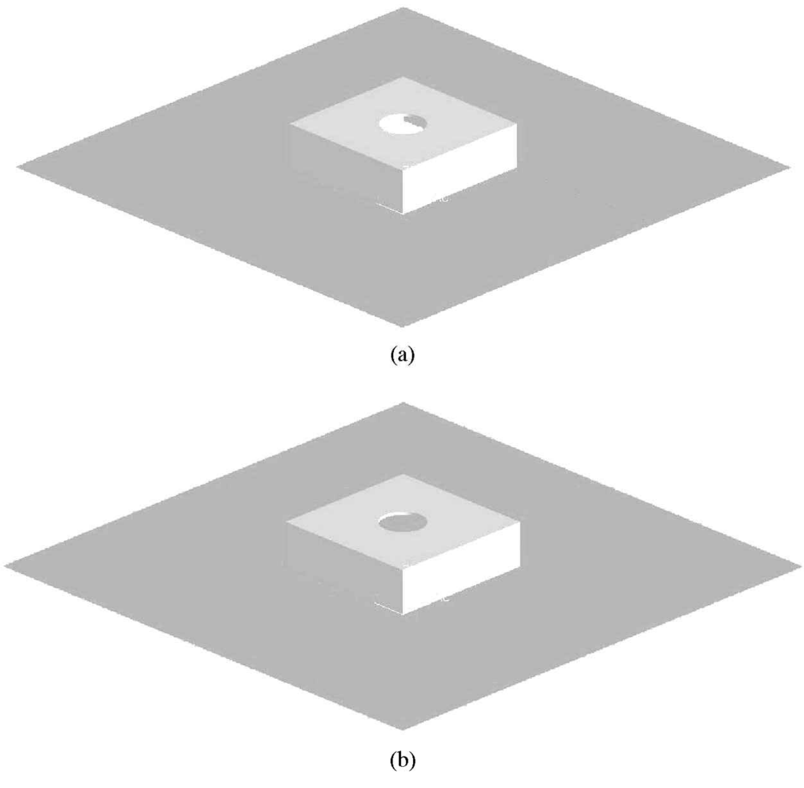


Fig. 5 Core block

generation of the parting surfaces on them are very difficult and time consuming tasks.

Figure 9 shows a part model with the parting surfaces with the corresponding passage features, and Fig. 10 shows the surfaces that will form the parting surface by covering the passage features. As shown in Fig. 10, these surfaces for separating the passage region consist of many complicated faces.

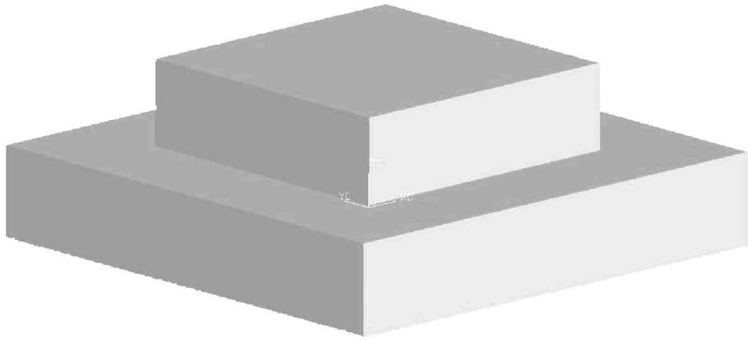
Most plastic parts fabricated by injection moulding have passage features passing through multiple faces. This makes it difficult to recognize passage features and to generate the parting surfaces automatically. In this paper, an algorithm to detect passage features is proposed as the basis for automatic parting surface generation.



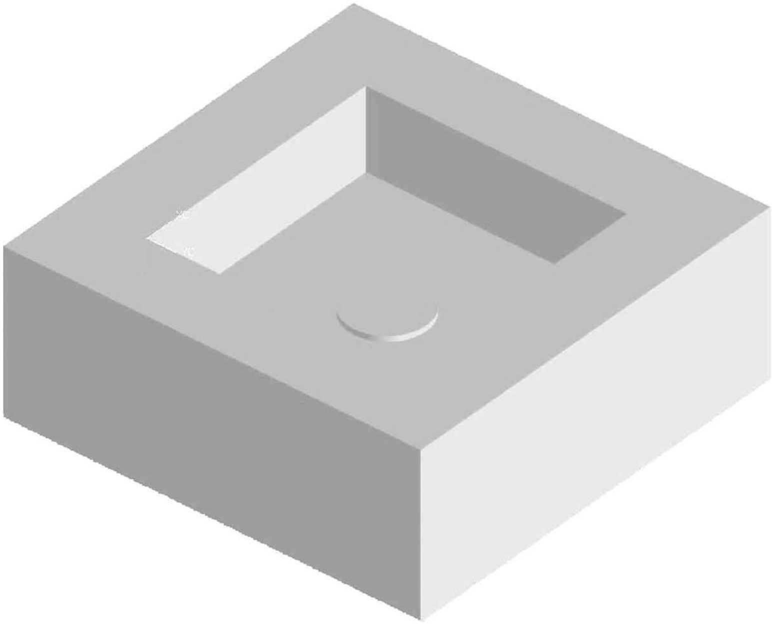
(a)

(b)

Fig. 6 Generation of parting surfaces



(a)



(b)

Fig. 7 Generation of the core and the cavity

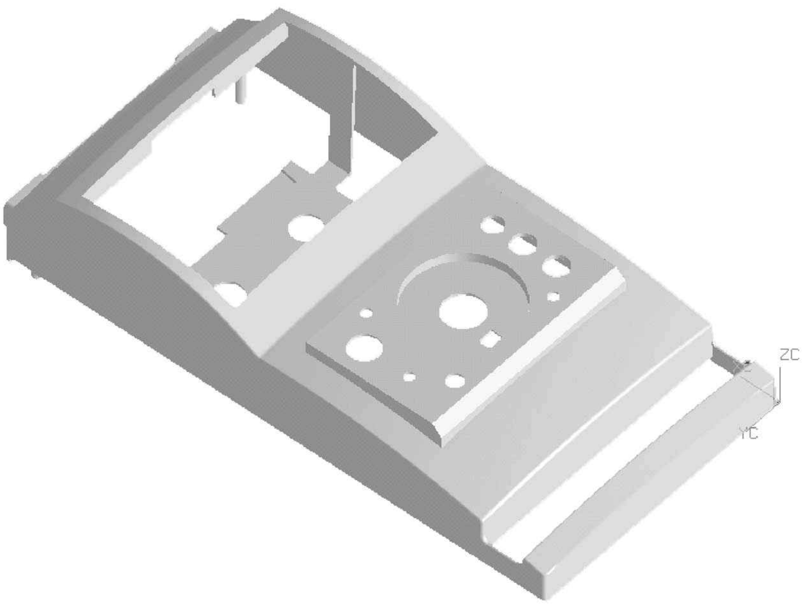


Fig. 8 Simplified front cover of an audio system

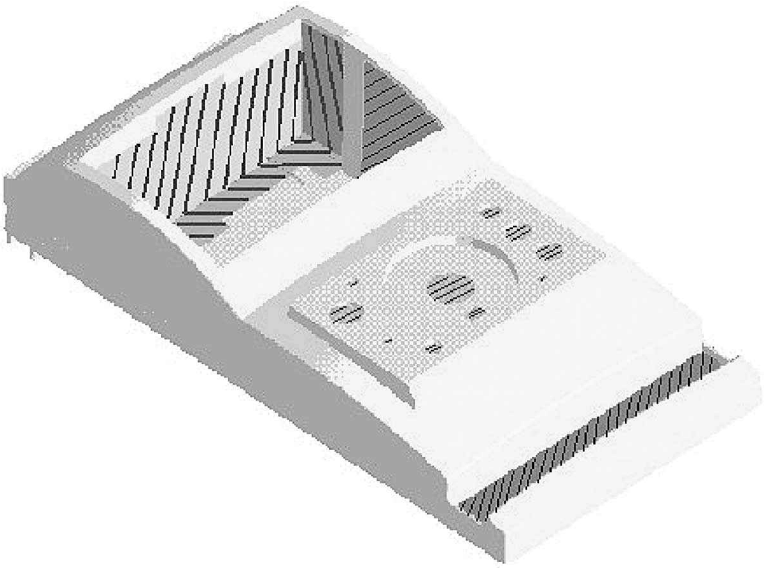


Fig. 9 Surfaces generated from passage features that constitute parting surfaces

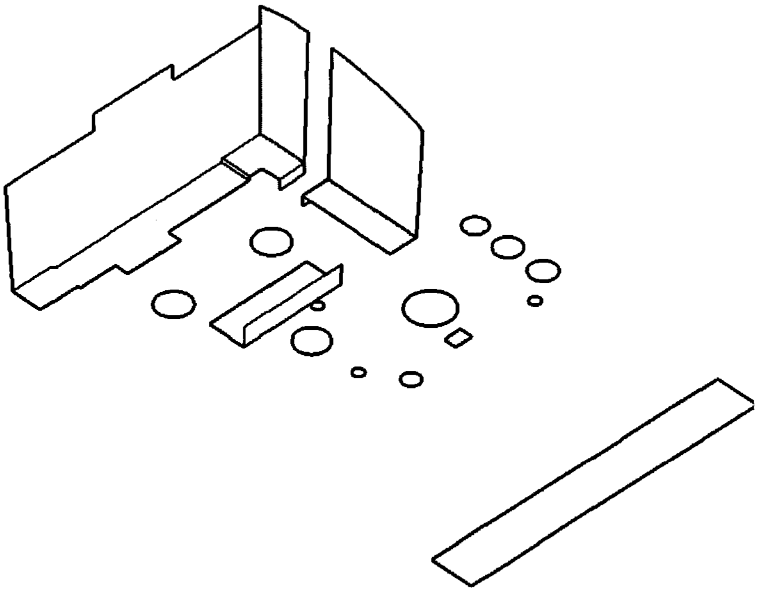


Fig. 10 Surfaces covering passage features

2.1 Related work

Since Kyprianou pointed out the necessity for shape classification in CAD systems in his PhD thesis [**5**], feature recognition in solid models has been a popular research topic. Until now, the extraction of machining features from CAD data has been the main subject of research, and feature recognition techniques have been derived to extract machining features, such as holes, slots and pocket features [**6**]. There are four distinct approaches to feature recognition:

(a) the graph pattern matching approach [**7**],
(b) the convex hull decomposition approach [**8**, **9**],
(c) the cell-based decomposition approach [**10**],
(d) the hint-based reasoning approach [**11**].

The graph pattern matching approach was introduced by Joshi and Chang, and has proven to be the most popular method in the feature recognition field [**12**]. In this approach, the B-Rep data structure of a part is mapped onto a graph whose nodes represent faces and whose branches represent edges. This graph is called the face adjacency graph (FAG). Then, a subgraph isomorphism is used to search subgraphs that match the feature templates. However, in the case of intersecting features, the subgraphs in the part are deformed, which makes feature recognition impossible.

The convex hull decomposition approach was proposed by Woo [**9**]. This approach recognizes the features by Boolean operations between a part model and a convex hull surrounding the part model. The Boolean operation between the part model and convex hull is applied recursively. When the output of the Boolean operation is empty, the operation is terminated. Unfortunately, the decomposition process may not necessarily converge. To overcome this problem, Kim proposed the alternating sum of volumes with partitioning (ASVP) decomposition [**9**], which recognizes features by comparing the part model faces with its convex hull faces.

The cell-based decomposition approach was proposed by Sakurai and Chin [**10**]. This method recognizes features by decomposing the delta volume into minimal cells and then combining these cells. The delta volume is the difference between a stock and a part model. Using this approach, the number of cells derived from the decomposed delta volume is large. Consequently, a large number of combinations are needed to recognize the features by combining the cells.

The hint-based reasoning approach was proposed by Vandenbrande [**11**]. This approach was proposed in order to recognize intersecting features, which are the main problem of the graph pattern matching approach. The technique recognizes features from the traditional traces of features. For example, a slot hint is generated when a pair of parallel opposing planar faces is encountered, which correspond to slot walls. With such hints, the algorithm recognizes the slot feature by searching the slot floor between the slot walls.

As is suggested by the above, research on machining feature recognition is being actively pursued. Some research has also been performed on feature recognition specifically related to moulds, for example upon undercut features [**13**]. However, as far as the present authors are aware, no research has been undertaken to date on the recognition of passage features for parting surface generation.

3 OVERVIEW

A passage feature is composed of a pair of hole loops and the faces connecting the hole loops, and thus passage feature recognition is a process of finding paired hole loops. Because a passage feature can pass through several connected faces, the hole loops lying over several connected faces should be identified at first.

When a hole loop is found, its pair hole loop is searched. If the pair of hole loops satisfies a certain condition to become a passage, it is registered as a passage feature.

3.1 Definition

Since the passage features passing through many connected faces are the target features to be recognized, a special hole loop, called a *multiface hole loop*, formed over several faces, is defined in this paper as follows:

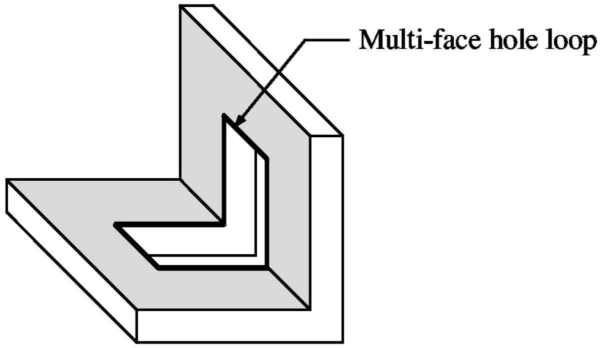


Fig. 11 Example of a multiface hole loop

> For any pair of connected faces, all the edges shared by two faces in the pair are eliminated and an expanded face is generated. If internal loops of edges inside the expanded face exist, these internal loops are defined as multiface hole loops.

An example of a multiface hole loop is illustrated in Fig. 11.

3.2 Basic concepts

A passage feature is composed of two hole loops including multiface hole loops and the faces coupling these hole loops. These faces are called side faces. Thus, the procedure for finding passage features can be conceptually simplified to find two connected hole loops. The detailed procedure will be explained in Section 4.

4 ALGORITHM

In this section the proposed algorithm will be explained in detail. The detailed algorithm for recognizing the simple passage feature will be explained first, and then the approach used to recognize the hole loops on multiply connected faces will be described. A passage feature is defined as simple if its entrance and exit lie over a single face. Finally, the conditions necessary for multiface hole loops to compose a passage, and the method for finding all the combinations of connected faces, will be explained.

4.1 Algorithm for passage feature recognition

As explained earlier, a passage feature is composed of two hole loops connected to each other by side faces. The detailed procedure for passage feature recognition is described using the following example.

The sample part illustrated in Fig. 12 has one passage feature and each symbol is defined as follows:

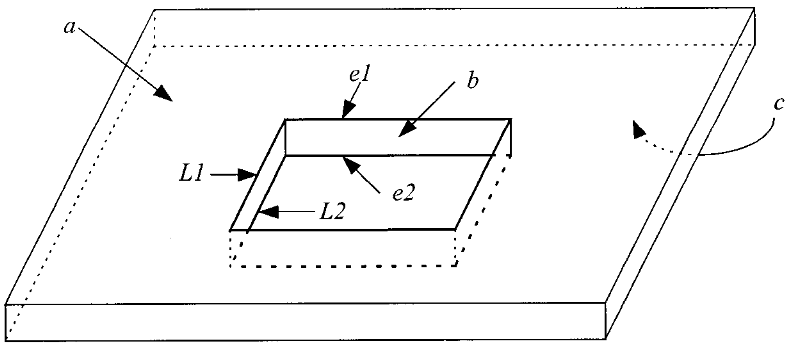


Fig. 12 Example using a simple passage feature

a = top face of the sample part
b = face connecting the top and bottom surfaces
c = bottom face of the sample part
$e1$ = edge shared by face a and face b
$e2$ = edge shared by face c and face b
$L1$ = hole loop in face a
$L2$ = hole loop in face c

In the example part, the loops composing the passage feature are $L1$ and $L2$, and the faces, such as face b, connect these loops. Generally, the face (face b) composing a passage feature has edges shared by the neighbouring faces (face a and face c), and these edges form hole loops in the neighbouring faces (face a and face c). That is, face b is a face composing the passage feature and is named the *side face*. Faces a and c are named the *entrance face* and the *exit face* respectively. The procedure for finding the passage features is as follows:

1. Search for hole loops in a face (entrance face of a passage).
2. Search for edges composing the hole loop searched for in step 1.
3. Search for faces (side faces) sharing the edges searched for in step 2.
4. If the edges of the face (side face) searched for in step 3 form a hole loop inside another face (exit face), this hole loop and the hole loop searched for in step 1 form a passage feature.

The algorithm explained above is for recognizing passage features composed of hole loops residing on a single face. To recognize the passage features composed of hole loops lying over multiple faces, the concept of the multiface hole loop must be applied.

4.2 Recognition of hole loops on multiply connected faces

As described earlier, a multiface hole loop is defined as the internal loop formed over an expanded face formed by removing edges shared by connected faces. Three steps are needed in order to recognize the hole loops on multiply connected faces:

(a) remove the edges shared by the connected faces,
(b) construct loops with the remaining edges,
(c) identify the internal loop.

The procedure for recognizing a multiface hole loop can be described as follows. A sample part for explaining the multiface hole loop is shown in Fig. 13, where the grey faces are the connected faces being investigated. The grey faces share edges $e1$ and $e2$. If the shared edges in

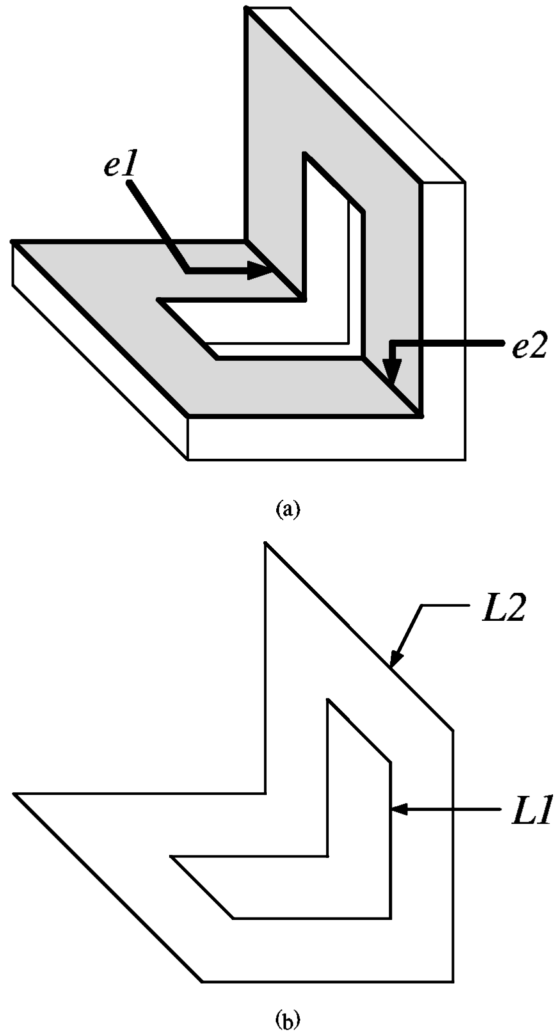


Fig. 13 Example for recognizing a multiface hole loop

the two faces are removed and the two faces are considered as one face, as shown in Fig. 13b, two loops are formed, i.e. *L1* and *L2*. Loop *L1* corresponds to the internal loop and loop *L2* corresponds to the peripheral loop. Consequently, the multiface hole loop is loop *L1* in this case. Likewise, the internal loops formed over *n* arbitrary connected faces are defined as multiface hole loops if internal loops exist when the shared edges of connected faces are removed. Once the multiface hole loops are identified, the procedure for finding a simple passage described earlier can be applied in order to recognize the passage features in the multiple faces.

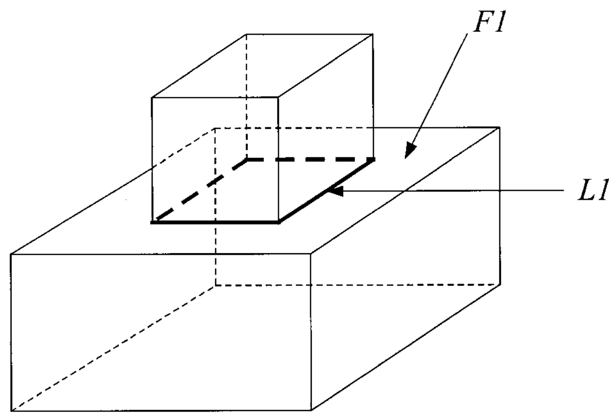


Fig. 14 Hole loop that does not constitute a passage feature

4.3 Conditions necessary for a multiface hole loop to form a passage

For the multiface hole loops to compose passage features, two conditions are required. First of all, all the edges composing the multiface hole loop must be convex. For example, for the part illustrated in Fig. 14, loop *L1* is a hole loop in face *F1*, but the edges composing the hole loop *L1* do not compose a passage feature because they are concave. This may be a natural conclusion considering that the edges of the multiface hole loops become either the entrance or exit of the passage and only convex edges may form the boundary of the entrance or the exit. Therefore, a procedure for judging the convexity of edges is required. The classification of edges by Kyprianou is illustrated in Fig. 15 [**14**].

Edges are classified by the angle between the two faces that share the corresponding edge; i.e. if the angle measured inside a part is smaller than 180°, the edge is convex. Likewise, if it is larger than 180°, the edge is concave. If the two faces are connected smoothly as shown in Figs 15c or d, the edges are classified by the local curvature [**14**]. By judging the convexity of the edges, the multiface hole loops composed of concave or smooth concave edges are excluded during the procedure for recognizing passage features.

Even if all the edges satisfy the first condition, there are cases where passage features cannot be composed because of the relationship between the multiface hole loops and the faces composing the model. Here, the second condition comes into play. The multiface hole loops can be considered as a collection of edges that will bound the parting surfaces later for the corresponding passage

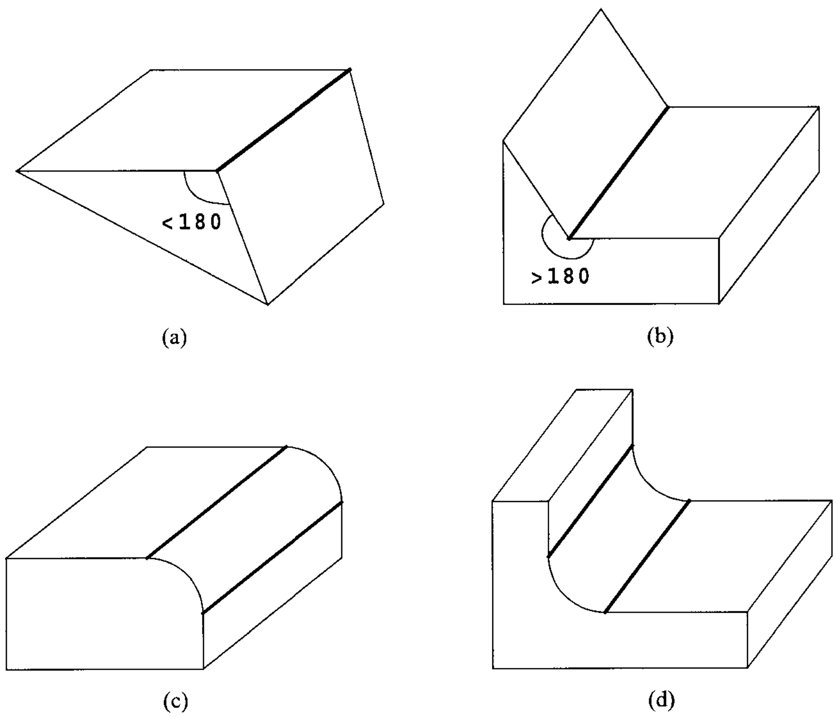


Fig. 15 Kyprianou's edge classification [**14**]

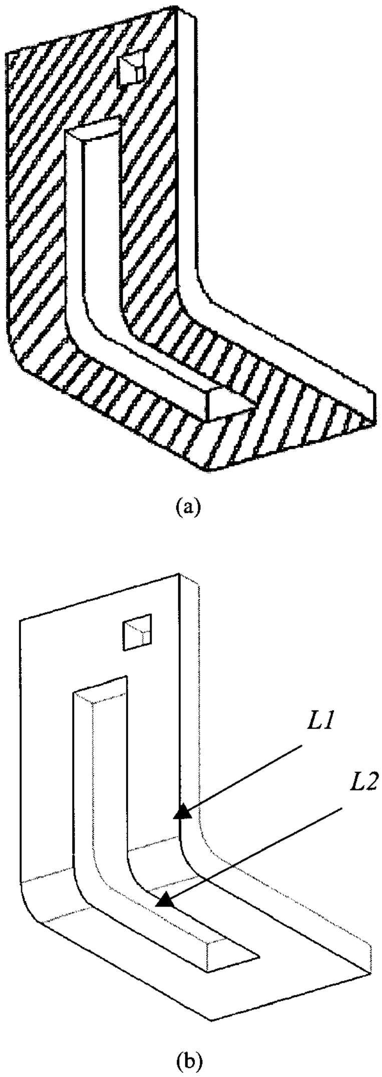


Fig. 16 Example of a valid multiface hole loop

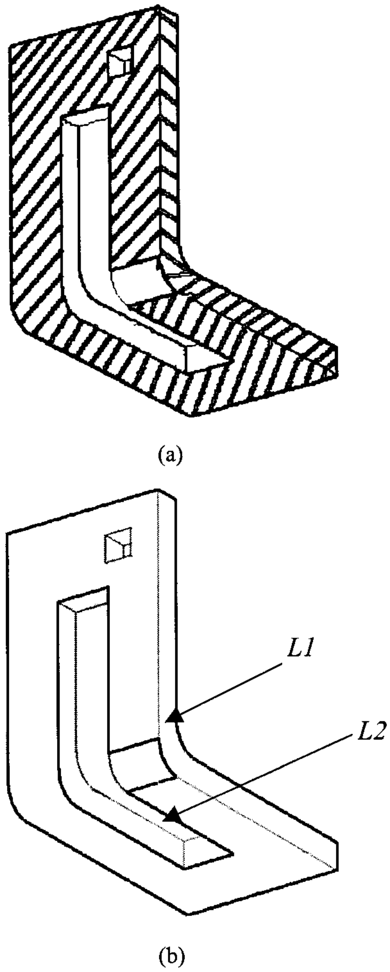


Fig. 17 Example of an invalid multiface hole loop

features. Since the generated parting surfaces must not overlap the faces of a part, multiface hole loops whose inside region overlaps the part surfaces must be excluded from consideration.

A multiface hole loop that satisfies such a condition is shown in Fig. 16, and Fig. 17 shows a multiface hole loop that violates the condition. In other words, the parting surface to be generated using multiface hole loop *L2* shown in Fig. 16b, which is identified from the merged faces hatched in Fig. 16a, does not overlap the faces composing the part. Merged faces are a set of connected faces within which a multiface hole loop is searched for. However, the parting surface to be generated using multiface hole loop *L2* in Fig. 17b, which is searched for among the merged faces hatched in Fig. 17a, overlaps the part faces. Therefore, the case in Fig. 17 must be excluded from consideration in finding the passage features. The detailed condition for invalid multiface hole loops is as follows. The readers may wonder why an odd situation like this is considered. All possible cases of multiface hole loops have to be considered because they are generated automatically by merging connected faces.

The faces that share the edges composing a multiface hole loop and do not participate in the merging process are side faces. If both the starting point and the end-point of an edge shared by two side faces are on the multiface hole loop, this edge can compose a loop with several other edges that belong to the multiface hole loop. A parting surface generated using this loop should be the same face as one of the side faces, which makes the parting surface overlap the part face.

Figure 18b represents a two-dimensional layout of the part in Fig. 18a. Faces *F1*, *F2*, *F3* and *F4* correspond to the merged faces, and faces *S1*, *S2*, *S3*, *S4* and *S5* correspond to the side faces. The boundary edges of the multiface hole loop are indicated by thick lines. Edge *a* shown in Fig. 18b is an edge shared by the side faces, and both the starting point and the end-point of edge *a* are on the multiface hole loop. Thus, edge *a* forms a loop together with other edges of *S5* that also belong to the multiface hole loop. The parting surface generated from this loop becomes the existing face *S5*, and thus the parting surface overlaps *S5*. In this way,

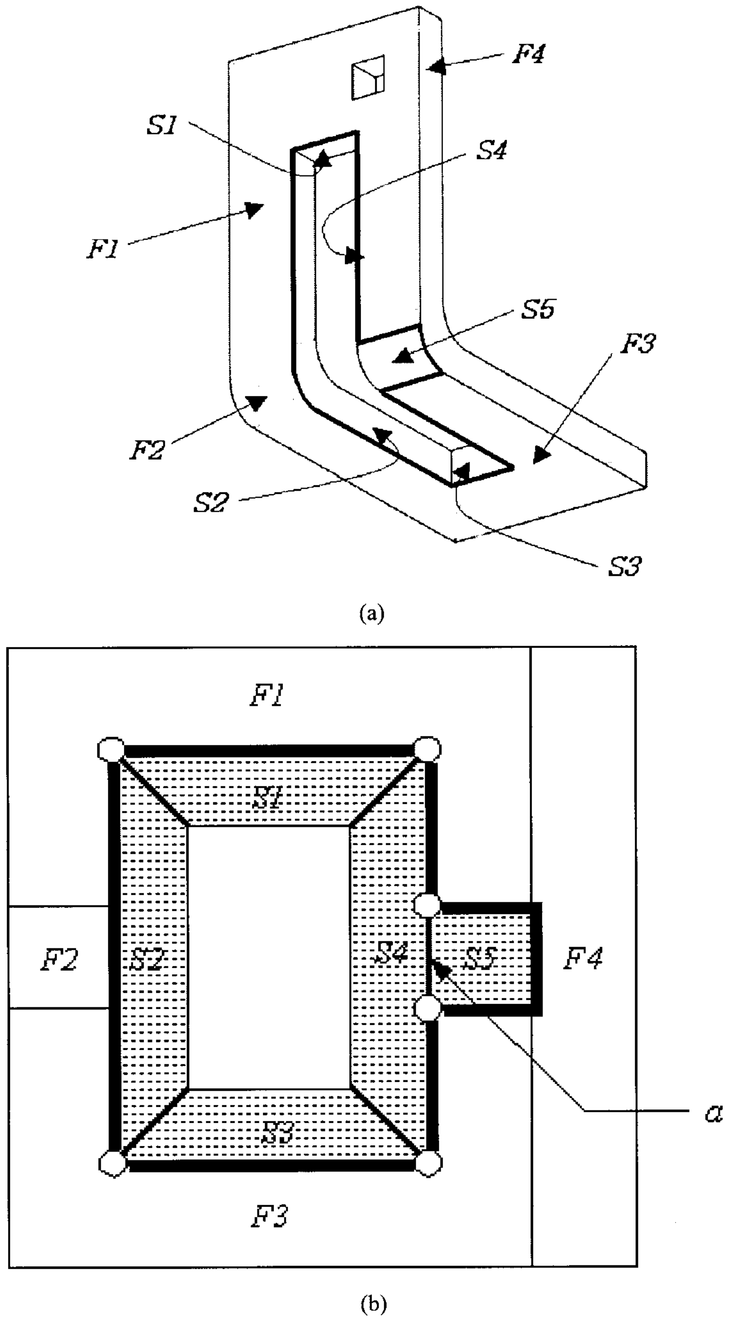


Fig. 18 Planar display of participating faces

the recognition of multiface hole loops yielding invalid parting surfaces can be avoided by investigating the relationship between the multiface hole loops and the end points of edges shared by the side faces.

4.4 Procedure for finding all sets of connected faces

To find the multiface hole loops in a part, a process that identifies and merges the connected faces is needed. Two approaches can be applied to generate the sets of connected faces. The first approach involves selecting the sets of faces to be merged by judging the connectivity between the member faces out of face sets generated arbitrarily. However, in this approach, the number of sets generated arbitrarily increases approximately by an order of n^r when the total number of faces is large:

$$\binom{n}{r} = \frac{n!}{r!(n-r)!} = \frac{n(n-1)(n-2)\cdots(n-r+1)}{r(r-1)(r-2)\cdots 1}$$

$$X \approx O(n^r)$$

where

$X =$ number of combinations
$n =$ total number of faces

r = number of faces to be merged

For example, if the total number of faces is 200, the number of candidate face sets is 64 684 950 when four faces are to be merged. Thus, this approach is not efficient in generating the sets of connected faces to be merged.

The second approach involves generating face sets by identifying faces connected to the faces that have been already identified as the elements of a feasible face set. In this approach, the member faces of a registered face set are stored in a stack. Then, it searches faces connected to the last registered face among the registered faces. If there are no more faces connected to the last registered face, the last registered face is removed from the stack and the searching process continues with the registered face just below the removed face. This approach can efficiently shorten the time for generating face sets by searching for connected faces only, whereas the first approach searches all possible sets and checks the connectivity. The generated sets of faces to be merged are used in the module used for finding multiface hole loops.

5 IMPLEMENTATION

5.1 Program architecture

The basic concept of the proposed algorithm is as follows. Up to the moment when the number of searched passage features equals the total number of passage features in a model, the number of faces to be merged is increased starting from 2. The total number of passage features is derived using the Euler equation [**15**]. In each step, the multiface hole loops are found and passage features are recognized by searching for the connected multiface hole loops. Thus, the implemented program is composed of four subroutines, i.e. routines for finding the total number of passage features, finding all combinations of connected faces, recognizing multiface hole loops and recognizing passage features. The program was developed with Unigraphics V15.0 API on Windows NT.

5.2 Analysis of running time

The algorithm proposed in this paper searches for passage features with an increasing number of faces to be merged. Thus, the running time for recognizing all passage features in a model can be determined by the passage feature involving the maximum number of connected faces. Therefore, the running time can be represented by the maximum number of faces to be merged.

As mentioned earlier, the procedure for generating sets of faces to be merged involves searching for the faces connected to a face and searching again for the faces connected to the searched faces until all members of the connected face sets of the given number of faces are collected. If the total number of faces in a part is n, the average number of faces connected to each face is m and the number of faces to be merged is r, the number of feasible face sets is of the order of $n \cdot m^{r-1}$:

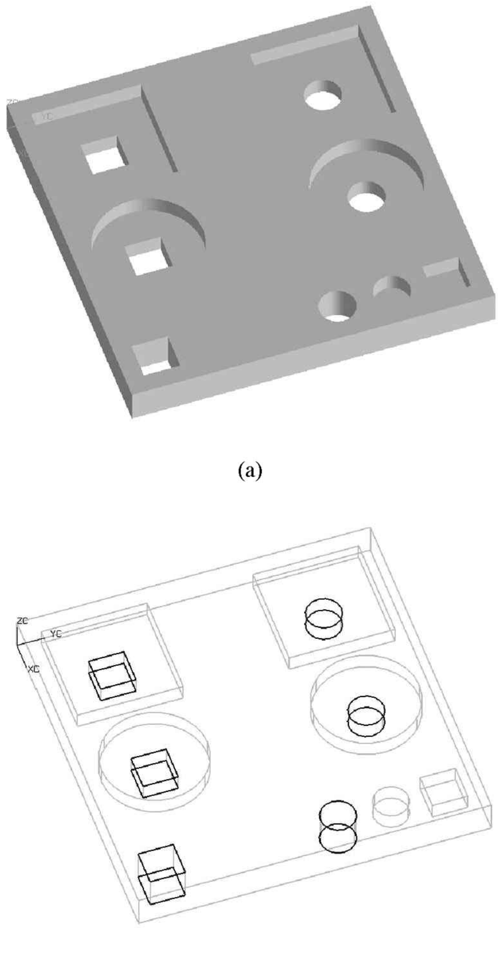
$$X \approx O(n \cdot m^{r-1})$$

Generally, the average number of faces connected to each face is much smaller than the total number of faces in a part.

For the part illustrated in Fig. 8, the total number of faces is 286 and the average number of faces connected to each face is 5. In this case, the expected number of

Table 1 Order of combinations for the part in Fig. 8

Number of faces to be merged	Expected number of combinations
2	1430
3	7150
4	35 750



(a)

(b)

Fig. 19 Example part with floorless pockets and through holes combinations according to the number of faces to be merged is as shown in Table 1.

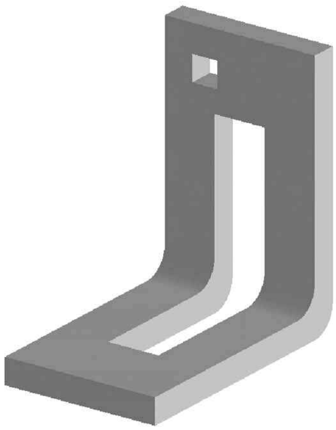
6 CASE STUDY

6.1 Example part

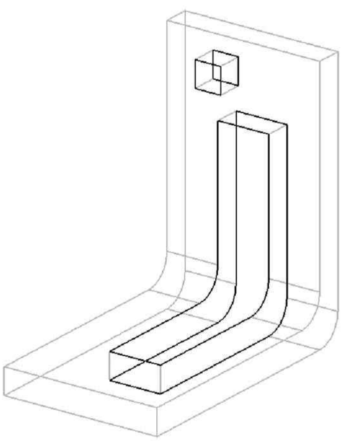
The example part shown in Fig. 19a has several pockets, holes and passage features bounded by the hole loops formed over one face. Figure 19b shows the six passage features recognized.

6.2 L-shaped part

The sample part shown in Fig. 20a is L-shaped and has a passage feature composed of hole loops formed over multiple faces in addition to a simple passage feature. In this case, the number of faces forming the multiface hole loops is 4. The recognized passage features are illustrated in Fig. 20b.



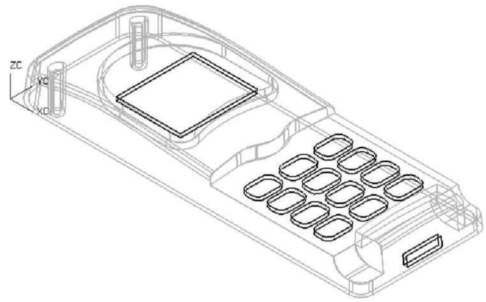
(a)



(b)

Fig. 20 L-shaped example part

(a)



(b)

Fig. 21 Example part—cover of a cellular phone

6.3 Casing of a cellular phone

The sample part shown in Fig. 21a is the front cover of a cellular phone and has 14 passage features. The recognized passage features are illustrated in Fig. 21b.

6.4 Minicomponent front cover

The sample part shown in Fig. 22a is the front cover of a minicomponent and has passage features residing over multiple faces. This part has 15 passage features. The recognized passage features are shown in Fig. 22b.

The running time for the four cases illustrated above is shown in Table 2.

7 CONCLUSIONS

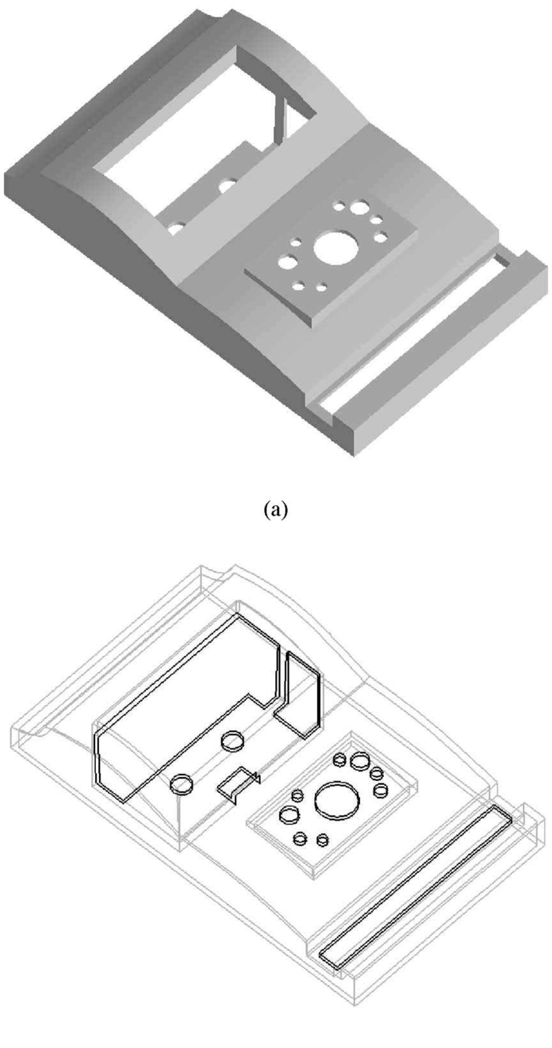
In this paper, an algorithm is proposed for automatic recognition of passage features using the concept of the multiface hole loop. The generation of the parting surfaces is one of the most natural target tasks to be automated in mould design using a three-dimensional CAD system. The parting surfaces are easily generated along the boundaries of the part to be made. However, if the part to be made has passage features, their entrance

Table 2 Data sheet of example parts

	Simple part	L-shaped part	Housing of cellular phone	Minicomponent front cover
Total number of passage features	6	2	14	15
Running environment	Pentium II 400MHz CPU, 256 Mbyte RAM			
Total number of combinations searched	48	1335	189	5631
Running time (s)	1.687	20.625	11.250	163.906

and exit areas should also be covered by the parting surfaces. Sometimes, it is a tedious and error-prone task to identify all the passage features when the part has a complicated shape. The algorithm proposed in this paper could shorten the time for designing injection moulds by automatically recognizing passage features passing through multiple faces.

The algorithm proposed in this paper recognizes passage features with increase in the number of faces to be merged. Thus, the time for recognizing passage features is influenced by the maximum number of faces forming the hole loops that compose such passage features. As the number of faces to be merged increases, the time for selecting the faces to be merged also increases because the number of face sets increases. Thus, the algorithm for generating sets of faces to be merged needs to be improved by excluding unrealistic cases in advance by exploiting the relationship between faces in a part. Furthermore, an algorithm for automatic generation of parting surfaces based on the recognized passage features needs to be developed. Generating a parting surface would be even more difficult when the parting surface is located between the entrance and the exit of a passage. This situation occurs when the side faces of the passage have a convex curvature.



(a)

(b)

Fig. 22 Example part—front cover of an audio system

REFERENCES

1 **Weinstein, M.** and **Manoochehri, S.** Geometric influence of a molded part on the draw direction range and parting line locations. *Trans. ASME*, *J. Mech. Des.*, March 1996, **118**, 29–39.

2 **Tan, S. T., Yuen, M. F., Sze, W. S.** and **Kwng, K. W.** Parting lines and parting surfaces of injection molded parts. *Proc. Instn Mech. Engrs*, *Part B*, *Journal of Engineering Manufacture*, 1990, **204**, 211–221.

3 **Nee, A. Y. C., Fu, M. W., Fuh, J. Y. H., Lee, K. S.** and **Zhang, Y. F.** Automatic determination of 3-D parting lines and surfaces in plastic injection mould design. *Ann. CIRP*, 1998, **47**(1), 95–98.

4 **Wong, T., Tan, S. T.** and **Sze, W. S.** Parting line formation by slicing a 3D CAD model. *Engng with Computers*, 1998, **14**, 330–342.

5 **Kyprianou, L. K.** Shape classification in computer aided design. PhD dissertation, Kings College, University of Cambridge, 1980.

6 **Han, J. H.** Survey of feature research. Technical Report IRIS-96-346, Institute for Robotics and Intelligent Systems, USC, USA, 1996.

7 **Little, G., Tuttle, J. R., Clark, D. E. R.** and **Corney, J. R.** The Heriot-Watt feature finder: a graph-based approach to recognition. In Proceedings of 1997 ASME Design Engineering Technical Conferences and Computers in Engineering Conference, 14–17 September 1997.

8 **Woo, T.** Feature extraction by volume decomposition. In Proceedings of Conference on *CAD/CAM Technology in Mechanical Engineering*, Cambridge, Massachusetts, March 1982.

9 **Kim, Y.** Recognition of form features using convex decomposition. *Computer Aided Des.*, 1992, **24**(9), 461–476.

10 **Sakurai, H.** and **Chin, C.** Defining and recognizing cavity and protrusion by volumes. In Proceedings of ASME Conference on *Computers in Engineering*, September 1993, pp. 59–65.

11 **Vandenbrande, J. H.** Automatic recognition of machinable features in solid models. PhD dissertion, Electrical Engineering Department, University of Rochester, 1990.

12 **Joshi, S.** and **Chang, T. C.** Graph based heuristics for recognition of machined features from a 3-D solid model. *Computer Aided Des*., 1988, **20**, 58–66.

13 **Fu, M. W., Fuh, J. Y. H.** and **Nee, A. Y. C.** Undercut feature recognition in an injection mould design system. *Computer Aided Des*., 1999, **31**, 777–790.

14 **Shah, J. J.** and **Mantyla, M.** Feature recognition technique. In *Parametric and Feature-based CAD/CAM*, pp. 110 (John Wiley, New York).

15 **Lee, K.** *Principles of CAD/CAM/CAE Systems* (Addison-Wesley).

