Advanced Design Software for the Injection Moulds Design Based on Solid Modelling Techniques

Ir. T. S. Huang
Division of Production engineering, Machine design and Automation
Faculty of Engineering, Department of Mechanical Engineering
Katholieke Universiteit Leuven
Celestijnenlaan 300 B
B-3001 Heverlee Belgium

Keywords: Solid modelling, Three-dimensional convex hull, Injection moulds , Undercuts, Parting lines and parting surfaces.

Abstract

This paper describes a CAD software that focuses on computer-aided design for injection moulds using solid modelling techniques. The ultimate goal of this CAD software is to enable the designer to do various mould design works. It consists of two main modules, geometric modelling and mould design. The former consists of a set of functions whcih can be used to create solid primitives and excute Boolean operations and the latter is able to support the design of standard mould components and some further applications such as the automatic determination of undercut, parting lines and parting surfaces, core and cavity plates, etc. This software is now implemented on a Unigraphics CAD/CAM system and operated in a dialogue mode to communicate with the user.

1. Introduction

Solid modelling techniques began to develop in the late 1960s [1,9,18,19]. It has been adopted in many engineering application due to the information complete, valid, and unambiguous representations of objects [3,4,11]. During the last two decades, the use of solid modelling in design and manufacturing is increasing rapidly because of the reduced computing cost, fast computing hardware, and software improvements. A CAD/CAM system not only focuses on improving the productivity of draftsmen but focuses on modelling engineering objects. Therefore, a basic requirement is that a geometric model should be an unambiguous representation of its corresponding object. That is to say, the model should be unique and complete to all engineering functions from drafting to engineering analysis to manufacturing [2,15,20]. Solid modelling offers these properties for further applications such as interference analysis, mass property calculation, and computer aided process planning (CAPP), etc [5,6].

The ultimate goal of this software, IMD- Injection Moulds Design, is to enable the designer to do injection mould design works using solid modelling techniques. It consists of two main parts, geometric modelling and mould design. In the geometric modelling part, it involves a neutral CAD programming interface, IMD-drawing routines, IMD-make function, creation of primitives, and solid Boolean operations. A designer can create several primitives such as blocks, cones, spheres, extruded solids, etc. A complex solid model can also be created by performing Boolean operations on two or more primitives. A final model created in this geometric modelling part will be used in the second part, mould design, for further application such as undercuts extracting, and parting lines and parting surface creating. The second part involves creation of standard mould components, detecting undercut, core and cavity plates, and creating parting lines and parting surfaces that are going to develop.

For the automatic determination of undercut application, this paper discusses an algorithm to determine the three-dimensional convex hull of a moulding and its application to detect potential undercut problems in mould design [7,14,16,17]. The convex hull algorithm is devised to determine the smallest convex polyhedron containing the concave polyhedral product provided as a solid CAD model. A different algorithm included in the software automatically generates real undercuts for improving the disadvantages using three-dimensional convex hull algorithm [10,13]. The new algorithm generates the minimal contour box and extensional solids from the final solid model. Undercuts are obtained through a Boolean difference operation on the minimal contour box by subtracting the original solid model and the extensional solids. The final purposes of developing this software not only try to help designer to execute mould design works but also improve user's understanding of using solid modelling techniques.

2. What is IMD?

IMD (Injection Moulds Design) is a software, which allows designer to execute mould design work by using solid modelling characteristics. The software architecture shown in figure 1 not only includes solid modelling geometrical functions such as creation of primitives and boolean operations but also adopts some special techniques for different applications. A detailed description of the software is in the following sections.

3. Geometrical Modelling Part

3-1. A Neutral CAD Programming Interface

In order to ensure portability to various existing CAD/CAM systems a CAD programming interface, neutral CAD programming interface, has been developed and implemented. The user application program accesses CAD system functions through a number of the interface functions which must be linked on a CAD system. By implementing this limited amount of functions the user application program becomes portable to another CAD-system. At this moment the interface has been integrated with the Unigraphics CAD/CAM system. The relationship among user application program, neutral programming interface, and Unigraphics CAD/CAM system as shown in figure 2.

The set of interface functions allow to:

-create solid primitives;

-inquire geometrical and topological information from the CAD-database;

-execute Boolean operations;
-transform and select solid objects.

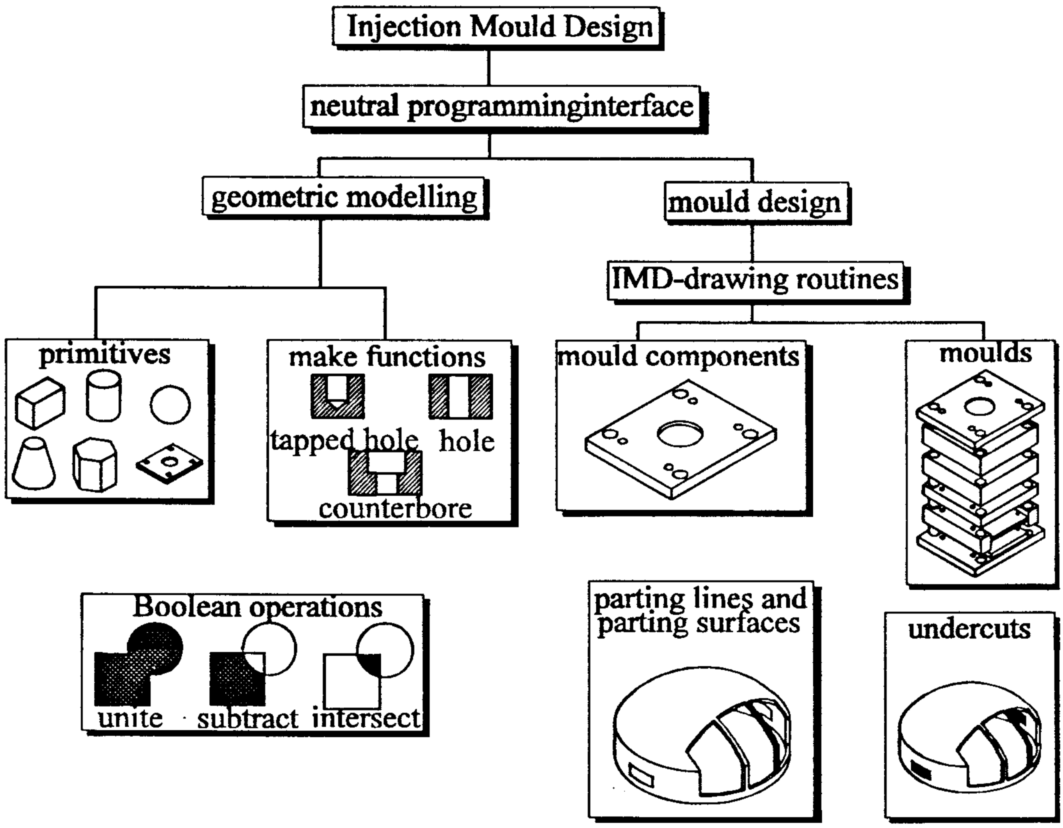


FIGURE 1. Structure of Injection Moulds Design (IMD).

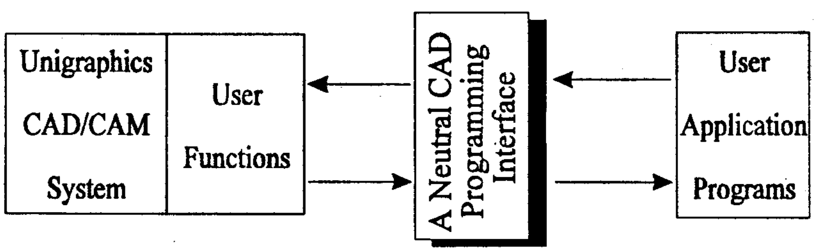


FIGURE 2. The relationship among user program, IMD interface, and Unigraphics system

3-2. IMD Creation of primitives

Most of the solid modelling modelers provide users with a certain set of primitives. Using Unigraphics CAD/CAM system as an example, it can create different types of primitives, such as blocks, cylinders, cones, spheres, and prisms etc.

In the IMD software a primitive requires a set of geometric data, location data, and orientation data to define it completely. Geometrical data defines the types of solid primitive and will be inputted by user. Location data requires an input point for the origin of the primitive. Orientation data is used to orient primitives properly relative to the world co-ordinate system. Following are descriptions of the common primitives that are used in Unigraphics CAD/CAM system (refer to Fig. 3):

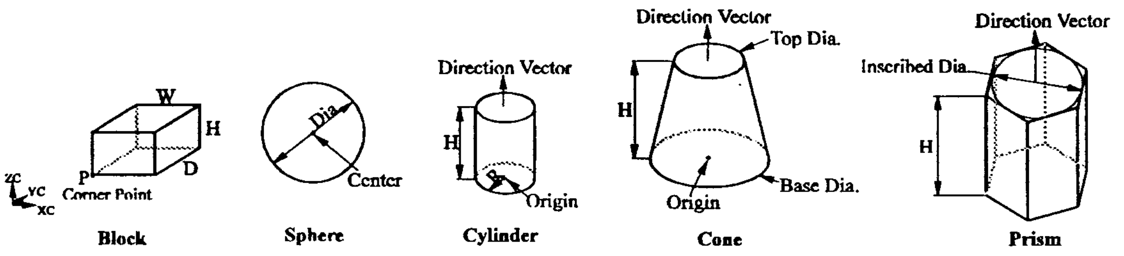
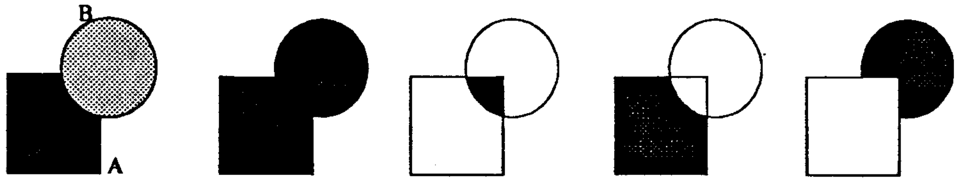


FIGURE 3. Common primitives used in Unigraphics CAD/CAM system.

(1). Block: Creates a block by defining the length of each edge and a corner point. The length of each edge is geometric data and the corner point is location data. The edges of the block align with the axes of the world coordinate system.
(2). Sphere: Sphere is created by specifying diameter and is centered about the center of its local coordinate system.
(3). Cylinder: This is a circular cylinder which be created by specifying an orientation, radius (or diameter), height, location. Geometric data is defined by its radius (or diameter) and height.
(4). Cone: This is a circular cone or frustum circular cone whose geometry is defined by its base diameter, top diameter, and height values. The direction determines the cone orientation.
(5). Prism: Prism is created by entering the number of sides, inscribed diameter, prism height, and the prism axis. The sign of the height value determines the creation direction. A positive height value cause the system to create the prism in the same direction as the prism axis vector. A negative height value cause the system to create the prism in the opposite direction of the specified vector.

3-3. IMD Boolean operations

Primitives are simple basic shapes that can be combined by a mathematical set of boolean operations to create the solid. The user usually translates and/or rotates primitives to position and orients them properly before applying boolean operations to construct the final solid. A complex solid body can be created by performing Boolean operations on two or more primitives. Almost all contemporary solid modelers provide some form of support for Boolean operations on solid objects. In IMD software the available Boolean operators are union, intersection, and subtraction. The union operator ($\cup$) is used to combine several primitives or objects. The resultant object occupies all the space of these initial primitives. The intersection ($\cap$) operator creates a resultant object that occupies only that space common to both initial objects. The subtraction operator (–) is used to subtract solids from the other. Figure 4 shows Boolean operations of one cube and one sphere.



(a). Primitives (b). A $\cup$ B (c). A $\cap$ B (d). A - B (e). B - A

FIGURE 4. Boolean operations of cube and sphere.

4. Mould Design-Special Features of IMD

4-1. IMD Drawing routines

A set of drawing routines allow to support the creation of standard mould components such as clamping plate, leader pin, and bushing etc. This part is quite important in injection mould design, because a final result of the whole mould should be display on screen. The designer can look and check correction of the final result from the display. Thus, we have to use these drawing routines to create and draw mould components.

4-2. IMD Make functions

This is very special part in IMD software. In standard mould components most of the plates have holes, tappeds, or counterbores on it. We can quickly create a hole, tapped, or counterbore on a plate component by using these make functions. The information, which was used to create these features, is quite useful in CAM. For example, a tapped hole can be created by specifying screw diameter, deepness, and center position, and this information can be used in manufacturing directly. This function allows to create hole, tapped, and counterbore. (Figure 5)

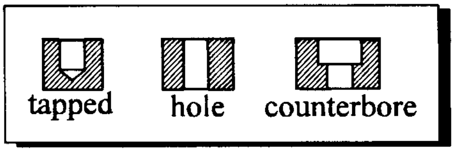


FIGURE 5. IMD Make functions.

5. Industrial Application of IMD

5-1. Creation of standard mould components

Each component is created by specifying a group parameters, which will be inputted into the drawing routine [8]. A group parameters consist of the length, height, and width of each plate and diameter & position of pin, bush, etc. Figure 6 shows a explanation of moving clamping plate according to the HASCO catalogue. A complete mould design shown in Figure 7 is composed of various components such as plates, guide pillars, guide bushes, and locating rings, etc. It will cause confusion and complication to designer because of a lot of lines displaying on the screen by using conventional wireframe modelling (Figure 7(a)).

Figure 7(b) shows the application of hidden-line removal. Showing hidden lines as dashed is also possible in the Unigraphics CAD system. Meanwhile both step section function and exploded function implemented in this software help the designer to expose internal details of a final mould design (Figure 7(c.d)). A mould can be represented with these auxiliary visualized functions so that the user can see the mould more clearly, and these functions can be executed more convenience than wireframe or surface modelling.

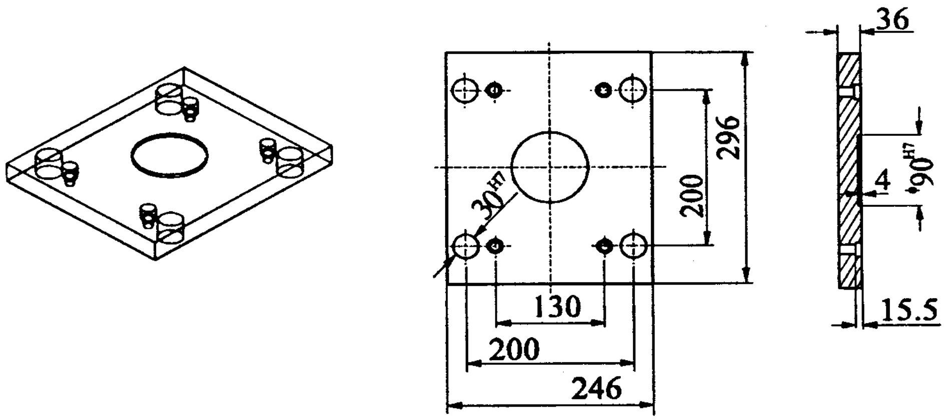
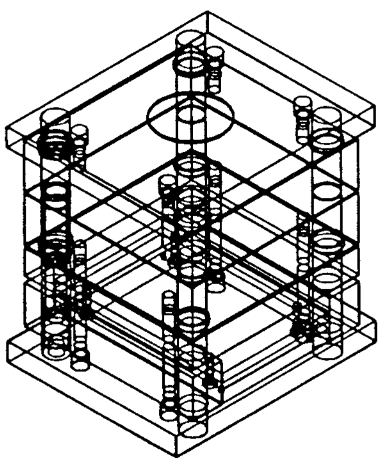
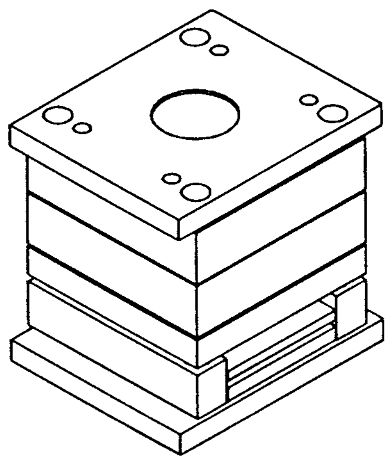


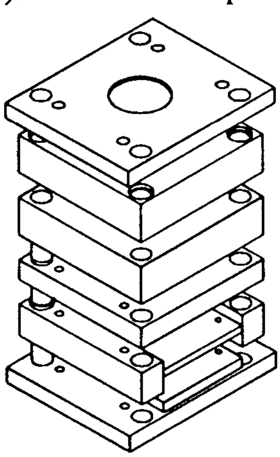
FIGURE 6. A explanation of plate parameters (HASCO K10).



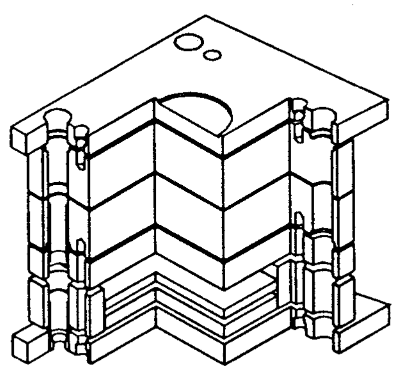
(a). Mould base components with visible view.



(b). Hidden-line removal



(c). Exploded view.



(d). Step section view

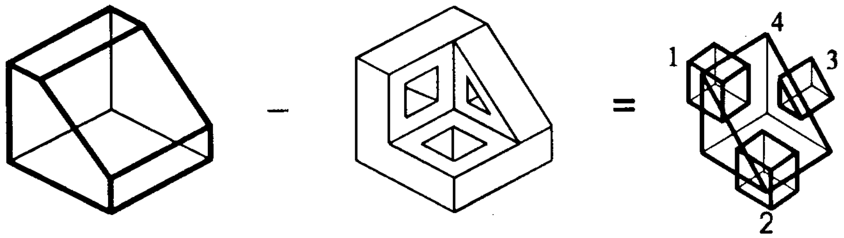
FIGURE 7. A completed mould with different display methods.

5-2. Automatic determination of undercut

Designers of plastic injection moulds are frequently confronted with demoulding problems resulting from undercuts. An undercut can be defined as any interference occurring between the mould and the moulded product when the product is ejected from the mould. The design of a suited mould for this type of components is inescapably more complex than for in-line-of-draw components. Mould designers spend lots of time in detecting all possible undercuts from the engineering drawing of the components to be moulded and designing the appropriate lateral sliding mechanisms intended to release the undercut before product ejection. This has led to a strong desire for realizing an automatic undercut detection during mould design.

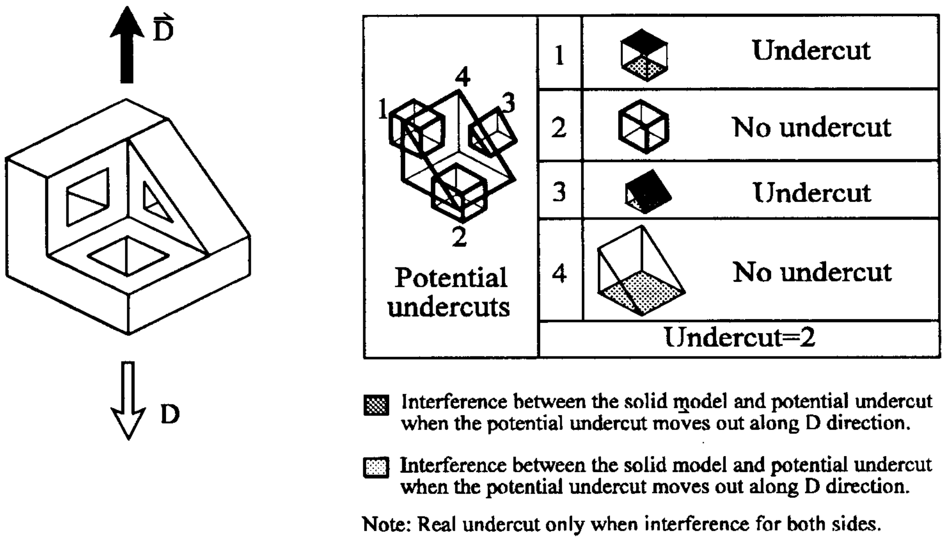
A method for extracting potential undercut from moulding has been developed [16,17]. It comprises two processes: 1) generating convex hull of the moulding 2) handling interference between such hull and moulding. The first process uses a algorithm, called three-dimensional convex hull, which features the use of the Euler operators to generate the convex hull. This hull must be represented by a solid model having the same data structure as the solid model of the given concave moulding. The second step determines "potential undercuts" by executing a subtraction set operation between the convex hull and the concave moulding.

Using the moulding in Fig. 8b as an example, figure 9 shows a procedure for extracting potential undercuts. According to the withdrawal direction $\vec{D}, D$ as shown in Fig. 9a., potential undercuts No.1 and No.3 become real undercuts: real undercuts interfere with the given part when they move out along withdrawal direction (Fig. 9b).



(a). Three-dimensional convex hull (b). Concave polyhedron (c). Potential undercuts

FIGURE 8. A procedure for extracting potential-undercuts.



(a). Withdrawal direction (b). Real undercuts

FIGURE 9. Real undercuts.

According to the proposed algorithm described above, it has a disadvantage when it determines real undercuts by designating a moulding product withdrawal direction. The potential undercuts shown in Fig. 8c are actually a solid part and users must perform a split operation to cut it [22]. An attempt was made in this example to split the potential undercut to four separately potential undercuts as shown in Fig. 9b. So, the user has to develop a split algorithm by using Euler operators to split the potential undercut.

This paper presents a new algorithm for the automatic extraction of real undercuts, which has been developed and realized in the IMD software. The hierarchy of the algorithm is show in Figure 10 [10,13].

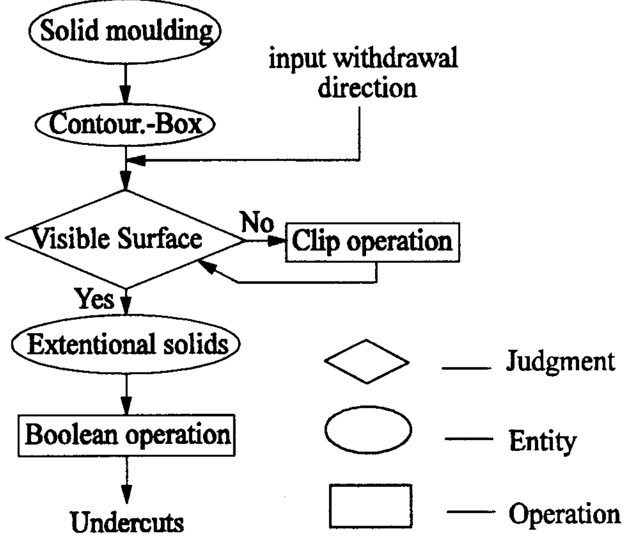
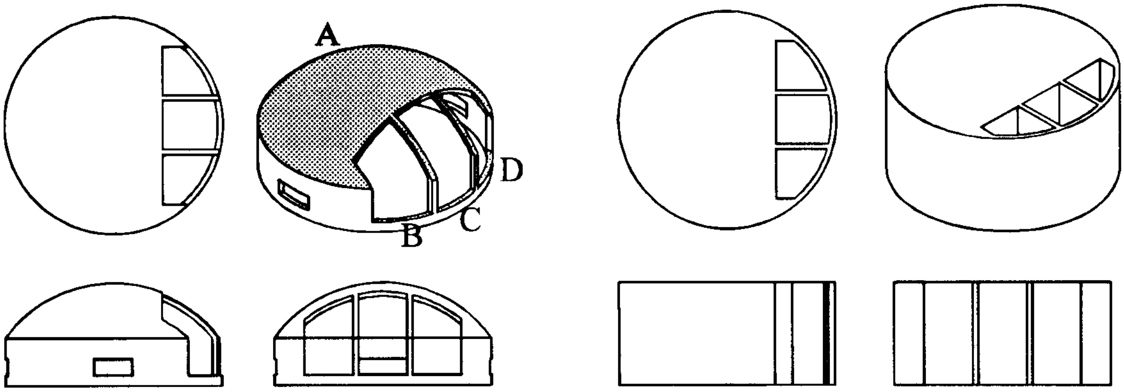
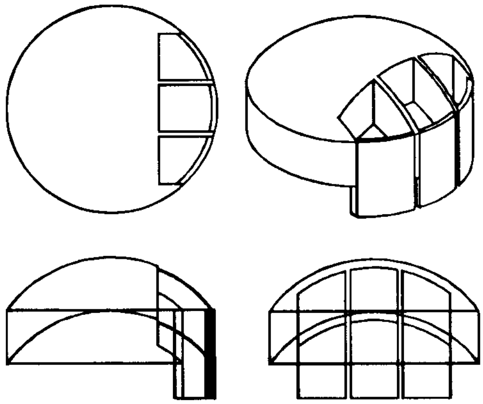


FIGURE 10. Hierarchy of extracting undercuts.

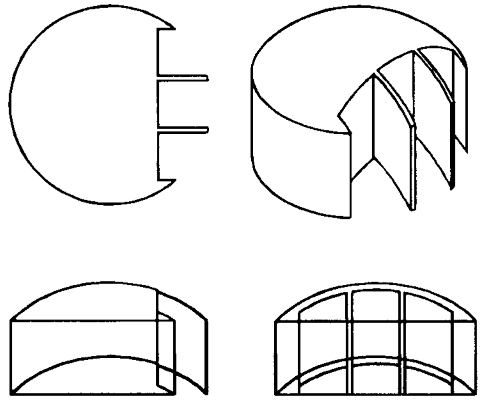
The first step creates a minimal contour solid which contains the specified moulded plastic part. Second step creates extensional solids by performing solid extrude operation along the withdrawal direction. Real undercuts can be determined by performing a difference set operation to the minimum contour box, extensional solids, and the given moulding. This procedure that extracts real undercuts for a mouse solid model is shown in Figure 11. For more detailed describtion about this algorithm, please refer to the reference [10,13].



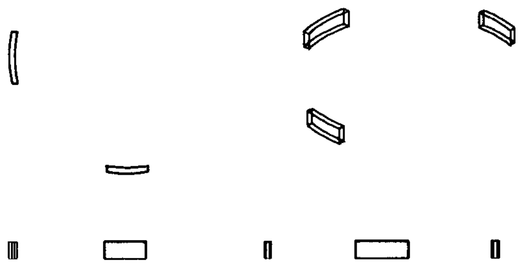
(a). A mouse solid model. (b). Contour solid



(c). Extentional solids obtained from A, B, C, D surfaces along $\vec{D}$ direction



(d). Extention solid created from inside sphere surface along ***D*** direction.



(e). Real undercuts

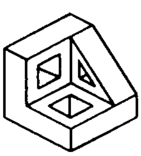
FIGURE 11. Real undercuts can be obtained by subtracting solid model , extentional solid (c), and (d) on contour solid.

5-3. Parting lines and parting surfaces

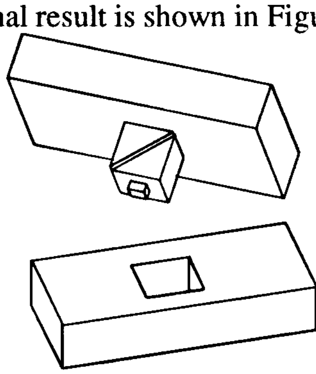
The parting line of a moulding in relation to the withdrawal direction separates the faces of the moulding into two portions. If *D* is the vector along the withdrawal direction and *N* is the surface normal at the neighbourhood of a point on the moulding's surface, the parting line would pass through all the points such that some of their localities satisfy $N \bullet D > 0$ while others satisfy $N \bullet D < 0$. For a detailed exposition on this topic, please refer to reference [12,21].

5-4. Core and cavity plates

After finishing the parting lines and parting surfaces design, a core and cavity plate are created. The core plate can be obtained; the program defines a minimal surrounding box, which involve the full size of the moulding, to difference the moulding from the box using boolean operation and then the final pattern will union with core plat component. Finally, the cavity plate can be created by subtracting the box from cavity plate component. The final result is shown in Figure 12.



(a). Moulding



(b). Core and cavity plate

FIGURE 12. A moulding and the core/cavity plates.

6. Conclusion

The design of injection moulds is a complex process that comprises several sub tasks. According to the characteristics of solid modelling techniques it can provide various methods for visualizing and modifying a complex geometry. It can also be used at various applications such as interference checking between mould components.

This paper briefly described the basic features of solid modelling to support designer to execute mould design works. Besides the auxiliary visualized functions and drawing of the standard mould components, it provides the automatic determination of undercuts for the further design of slides.

Reference

[1]. Allen G., "AN INTRODUCTION TO SOLID MODELLING", Comput. & Graphics Vol. 8, No. 4, pp. 439-447, 1984.

[2]. Baer A., C.Eastman and M. Henrion, "Geometric modelling: a Survey", Institute of Physical Planning, Carnegie-Mellon University, Pittsburgh, USA, Feb. 1980.

[3]. BOULTER T. W., "Why solid modeling?", AEROSPACE AMERICA, JANUARY 1985, PP. 94-96.

[4]. Brandli N., Mittelstaedt M., "Exchange of solid models: current state and future trends", computer-aided design, Vol. 21, No. 2, March 1989, pp. 87-96.

[5]. Chiyokura H., Kimura F., "A Method of Representing the Solid Design Process", IEEE CG&A, April 1985, pp. 32-41.

[6]. Foley J. D., Computer graphics: principles and practice, Addison-Wesley Publishing Company, Inc., 1990.

[7] Furukawa S., "An Efficient Algorithm for Constructing Convex Hulls of 3-D Points", JSPE, Japan, Vol. 50, No. 11, 1984.

[8]. Huang T. S., "Using solid drawing routines to create standard mould components of injection moulds", Internal report KUL-PMA, October 1991.

[9]. Huang T. S., "Computer Aided Design of Injection Moulds Using Solid Modelling Techniques: Part 1- An Introduction to Solid Modelling.", Internal report KUL-PMA, November 30, 1993.

[10]. Huang T. S., "Computer Aided Design of Injection Moulds Using Solid Modelling Techniques: Part 4- Extracting Undercuts.", Internal report KUL-PMA, January 21, 1994.

[11]. Jablokow A. G., J.J. Uicker, and D.A. Turcic, "TOPOLOGICAL AND GEOMETRIC CONSISTENCY IN BOUNDARY REPRESENTATIONS OF SOLID MODELS", The ASME design technical conference-16th design automation conference, advances in design automation, DE-Vol. 23-1, Sep. 16-19, 1990, pp. 59-66.

[12]. kim T., Jee H., Chung J., "Development of Modular CAD System for Injection Molding", Report of CAD/CAM Lab., KIST, South Korea, 1988.

[13]. Kruth J. P. , Huang T. S., "Locating Demoulding Undercuts on a Solid Geometrical Computer Model", Proc. International Symposium of Young Investigators on Information, Computer and Control, Beijing, Feb. 2-4, 1994.

[14]. Mantyla M., Sulonen R., "GWB: A Solid Modeler with Euler Operators", IEEE CG&A, September 1982, pp. 17-31.

[15]. Meguid S. A., Integrated computer-aided design of mechanical systems, ELSEVIER APPLIED SCIENCE PUBLISHERS LTD., 1987.

[16]. Mochizuki T., Yuhara N., "Methods of Extracting Potential Undercut and Determining Optimum Withdrawal Direction for Mold Designing", Int. J. Japan Soc. Prec. Eng., Vol. 26, No. 1 (Mar. 1992).

[17]. Mochizuki T., Yuhara N., "Determining Algorithm of Three-Dimensional Convex Hull Used for Set Operations of Solids and Its Application to Interference Problem", ISCIE , Japan, Vol. 3, No. 11, 1990.

[18]. Myers W., "An Industrial Perspective on Solid Modeling", IEEE CG&A, March 1982, pp. 86-97.

[19]. Requicha A. A. G., H.B Voelcker, "Solid Modeling: A Historical Summary and Contemporary Assessment", IEEE CG&A, March 1982, pp. 9-23.

[20]. Requicha A. A. G., "Representations for Rigid Solids: Theory, Methods, and Systems", Computing Surveys, Vol. 12, No. 4, December 1980, pp. 437-464.

[21]. Tan T. S., Yuen M. F., Sze W. S., Kwong K. W., "Parting lines and parting surfaces of injection moulded parts", Proc. Instn. Mech. Engrs Vol. 204, pp. 211-221, 1990.

[22]. Zeid I., CAD/CAM theory and practice, McGraw-Hill, Inc., 1991.