Founded 1905

AUTOMATED PARTING METHODOLOGIES FOR INJECTION MOULDS

ZHAO ZHIQIANG

(B. Eng., M. Eng.)

A THESIS SUBMITTED

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF MECHANICAL ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2009

ACKNOWLEDGMENTS

There are many people supporting and helping me during my graduate study. I would like to express my sincere gratitude to all of them.

First my heartfelt gratitude goes to my supervisors, Professor Andrew Y. C. Nee and Professor Jerry Y. H. Fuh, for their guidance and advice throughout the duration of my graduate study in National University of Singapore. Their support, help and quick response to my questions, reports and papers made my research progress well. They guided me not just in academic research, and also gave me the confidence and support when I needed them. Their enthusiasm and encouragement gave me great motivation and confidence in many aspects. They will always be gratefully remembered.

I would like to thank my colleagues, Wang Ying and Goon Tuck Choy from Manusoft Technologies Pte Ltd, for their support and help during my graduate study. They provided me with a lot of valuable industrial input for my research in addition to the support on my daily work.

I wish to thank my parents and in-laws, who have waited for this thesis for many years, for their moral support and patience.

Finally, my sincerely thanks go to my wife, Yuan Meizhen, and my two daughters, Zhao Siting and Zhao Siyu, for their understanding and support during my graduate study. This thesis is especially dedicated to them.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

CHAPTER 2

LITERATURE REVIEW

CHAPTER 3
AUTOMATED PARTING METHODOLOGY BASED ON FACE TOPOLOGY AND MOULDABILITY REASONING

CHAPTER 4
AUTOMATIC GENERATION OF PARTING SURFACES

CHAPTER 5

AUTOMATIC GENEARTION OF SHUT-OFF SURFACES

CHAPTER 6

AUTOMATIC DESIGN OF CAVITY/CORE INSERTS AND LOCAL TOOLS

SUMMARY

Injection moulds play an important role in the industry since plastic moulded parts are significantly being used in engineering and consumer products. The high demand for automated design, high precision and short lead time has remained as bottlenecks in the mould industry. Software applications are able to provide automated and intelligent tools and functions to achieve such demand effectively. Consequently, the development of a Computer-Aided Injection Mould Design System (CAIMDS) and intelligent methodologies for CAIMDS has been the research focus in the industry as well as the academia in the last few decades.

In CAIMDS, the parting system of a mould is one of the most difficult and important tasks because it deals with the complex geometry of moulded products and generates the moulding inserts which form the product. The current parting systems cannot fully satisfy the parting requirement in terms of speed, quality and functionality for complex moulded products since most of them are incapable of dealing with complex geometries and especially geometric imperfections of industrial products. They also do not implement an error correction and feedback mechanism to improve their compatibility and capability for the various industrial applications. In addition, the generated parting and shut-off surfaces do not always satisfy the moulding requirements in terms of mouldability and manufacturability of injection moulds. Since multi-injection moulds are being used widely to satisfy special functionalities, a parting approach for multi-injection moulds is deemed necessary. Solving the problems mentioned above successfully is crucial to the realization of an efficient and powerful parting system for injection mould design applications.

The objective of this research is to develop a robust parting system, which provides more feasible, powerful and compatible parting methodologies for moulded products. An automated parting approach based on Face Topology and Mouldability Reasoning (**FTMR**) was developed to automatically identify cavity/core faces, inner/outer parting lines and undercut features. Case studies show that the **FTMR** parting approach can provide satisfactory results for the moulded products with free-form surfaces, complex geometry and geometric imperfections. An Error Correction and Feedback System (**ECFS**) was developed and incorporated within the **FTMR** parting approach to visibly locate and correct possible errors during the parting process. Automated and novel approaches were developed for creating parting and shut-off surfaces from parting line loops. The generated surfaces are compliant with mould applications because the algorithms consider the manufacturing and mouldability criteria as well as geometrical requirements. Case studies show that the approaches are efficient in creating parting and shut-off surfaces from the complex parting lines of moulded parts. Automatic approaches and procedures were developed for the design of cavity/core inserts and associated local tools. Case studies have demonstrated that the approaches are effective for generating all the moulding inserts and their local tools in a single process. In addition, a parting approach was presented to generate the sets of cavity/core inserts and their local tools corresponding to each moulding injection stage (represented by a set of homogeneous moulding objects) for multi-injection moulds. The approach has been implemented and industrial case studies were used to validate the results of the approach.

NOMENCLATURE

CAD	Computer-Aided Design
CAIMDS	Computer-Aided Injection Mould Design System
CAM	Computer-Aided Manufacturing
CAPP	Computer-Aided Process Planning
B-Rep	Boundary Representation
CSG	Constructive Solid Geometry
2D	Two-Dimension
3D	Three-Dimension
D	Direction
$\mathbf{P_D}$	Parting Direction along which a moulding opens
$\mathbf{P_{D+}}$	Parting Direction along which the cavity insert opens
$\mathbf{P_{D-}}$	Parting Direction along which the core insert opens
PL	Parting Lines
OPL	Outer Parting Lines
IPL	Inner Parting Lines
UF	Undercut Features
PS	Parting Surfaces
SO	Shut-off Surfaces
S	3D Solid Body
E	Edge
F	Face of a solid body
V	Vertex (End points of an edge)
L	Loop (a closed edge list)

B	Boundary (formed by a connected edge list)
E⇔F	Edge⇔Face Relationship
F⇔F	Face⇔Face Relationship
L⇔F	Loop⇔Face Relationship
B⇔E	Boundary⇔Edge Relationship
V⇔E	Vertex⇔Edge Relationship
FTMR	Face Topology and Mouldability Reasoning
ECFS	Error Correction and Feedback System
G-Map	Graph Map
V-Map	Visibility Map
N	Normal Vector
T	Triangle Net
AAM	Attributed Adjacency Matrix
A	Area
$\mathbf{F_a}$	An Adjacent Face of a Face **F**
R	Ray (defined by a point and a direction)
Rg	Region (formed by closed boundaries)
Rg_p	2D Region projected from 3D boundaries
$F_{Adjacent}$	All Adjacent Faces of a Face **F**
$F_{CavitySeed}$	Cavity Seed Face
$F_{CoreSeed}$	Core Seed Face
$F_{\text{int}\,er\,\text{sec}\,t}(S, R, P_{D+})$	Intersected Faces of ray R towards direction P_{D+} with solid body S
$\mathbf{G_k}$	Face Category/Group (k=1, 2, 3, 4)
L_{3D}	Length of 3D parting lines

L_{2D}	Length of 2D projected parting lines
PSF	Pseudo-Straddle Faces
AOA	Area Accuracy required for a moulding
AOD	Draft Angle Accuracy required for a moulding
AOL	Length Tolerance required for a moulding
FMT	Feature Manager Tree
API	Application Programming Interface
$\mathrm{C}_{\mathrm{INTERSECT}}$	An Intersected Curve of two surfaces
NURBS	Non-Uniform Rational B-Splines
$\mathrm{C}_{\mathrm{TRIM}}$	Trimmed Curves
$C(u)$	NURBS Curve
$R_{i,p}(u)$	Rational B-Spline Curve Basis Function
$R_{i,p,j,q}(u,v)$	Rational B-Spline Surface Basis Function
$S(u,v)$	NURBS Surface
U	Knot Values of NURBS
$\mathbf{D}_{\mathbf{UF}}$	Release Direction of an Undercut Feature
EC	End Condition associated with a guide direction in boundary constraint
$\mathbf{PS}_{\mathbf{R}}$	Ruled Parting Surfaces
$\mathbf{PS}_{\mathbf{C}}$	Corner Parting Surfaces
$\mathbf{PS}_{\mathbf{A}}$	Skinned Parting Surfaces
$\mathbf{V}_{\mathbf{NE}}$	Vertex in parting line loop in the North-Eastern direction
$\mathbf{V}_{\mathbf{NW}}$	Vertex in parting line loop in the North-Western direction
$\mathbf{V}_{\mathbf{SE}}$	Vertex in parting line loop in the South-Eastern direction
$\mathbf{V}_{\mathbf{SW}}$	Vertex in parting line loop in the South-Western direction

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Background of injection mould design

Injection moulds play an important role in the industry since plastic moulded parts are significantly being used in engineering and consumer products. The high demand for rapid design, short lead time and high precision has always been the bottleneck in the mould industry. For mould-making companies wishing to maintain the leading edge in local and international markets, they should attempt to shorten the manufacturing lead time and enhance the design quality by using advanced manufacturing equipments and automated software applications. Interestingly, the injection mould industry has shown several characteristics and trends recently. Firstly, more plastic components are being used instead of metals or alloys in automobiles, airplanes as well as traditional consumer products, and more are becoming the key elements of products in major industries. Secondly, the geometry and structure of plastic components are becoming more complex for satisfying both aesthetic and functional requirements. More free-form surfaces and humanoid styles are being designed in toys, medical components, etc. Thirdly, multi-injection moulds are being used widely in order to manufacture more complex components and satisfy special functionalities. Fourthly, Computer-Aided Injection Mould Design System (CAIMDS) is now commonly used in the design of injection moulds and 2D manual drafting mould design is out-dated. In adopting these new challenges, CAIMDS is encountering higher requirements such as efficiency, functionality and standardization for mould design. Consequently, the development of intelligent methodologies for CAIMDS has been the research focus in the industry as well as the academia in the last few decades.

CAIMDS strives to provide automated and intelligent tools and functions to assist the design of injection moulds. An injection mould is an assembly of components (as shown in Fig.1.1), including impression, moldbase, ejector, slider, lifter, cooling system, feed system, etc. Among all these components, the impression sub-assembly is the key component since it forms the part geometry. An impression is composed of the core, cavity and associated inserts (see Fig.1.2 (b)), and the part is finally ejected after the core and cavity inserts are opened. All the other components and sub-assemblies serve the function of the impression either directly or indirectly.

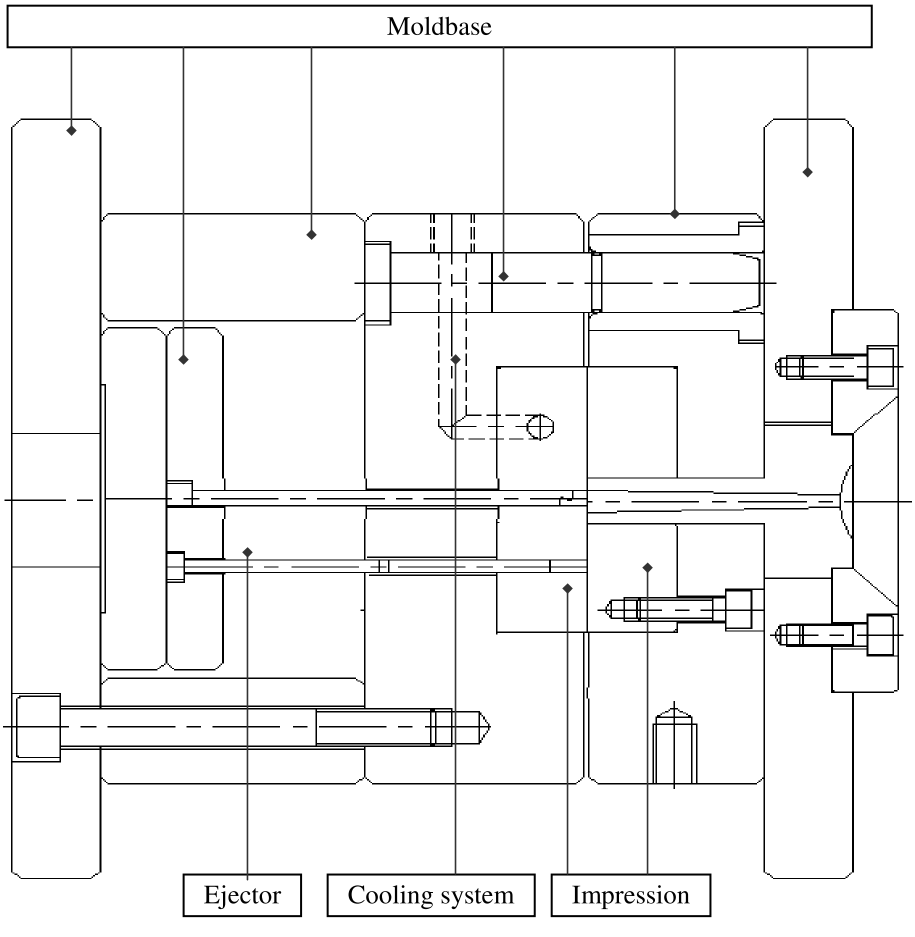


Fig.1.1. Typical structure of an injection mould

In order to design different components of an injection mould, CAIMDS would need to comprise of a parting system to split the moulded parts, tools to design slider and lifter mechanisms, approaches to design cooling and feed system, and libraries for moldbase, etc. Among these portions, the parting system is the core of CAIMDS since it aims to analyze the part's mouldability, deals with the various structures and complex geometry of moulded products and finally generates the impression assembly. The fundamental concepts and denominations of a parting system are introduced in the following section.

1.2 Overview of the parting system in CAIMDS

A plastic moulding is cooled and formed in an impression, which is composed of cavity and core inserts, and their local tools (e.g. side-cores and side-cavities) in case of the presence of any undercut features. The parting system attempts to identify the cavity and core faces, inner and outer parting lines, and to recognize undercut features based on the geometry and mouldability of a moulded product. It further creates parting surfaces from the outer parting line loop and patches the inner parting line loops using shut-off surfaces. Finally, it generates the core, cavity inserts and the associated local tools.

1.2.1 Dominations in parting system

Fig.1.2 (a) and (b) illustrate the key entities in a parting system. For a given moulding, the moulded product is formed between the core and cavity inserts, and ejected after the core and cavity inserts are opened. The pull direction along which the core and cavity inserts are opened is called the parting direction ($\mathbf{P_D}$). $\mathbf{P_{D+}}$ is the moving direction of the cavity insert, while $\mathbf{P_{D-}}$ represents the moving direction of the core insert. All the faces moulded by the cavity insert are designated as cavity faces, and those surfaces moulded by the core insert, as core faces. Undercut features are defined as the convex and concave portions of a moulding, which are not able to be moulded by the core and cavity inserts. They would require the incorporation of local tools and the slider or lifter mechanism to withdraw from the mould structure. The faces of the undercut features are called undercut faces. Parting lines are then defined as the intersection boundaries among the core faces, the cavity faces and undercut features. In principle, there are two types of parting lines, *i.e.* inner parting lines (**IPL**) and outer parting lines (**OPL**). **OPL** is composed of the largest parting line loop, while **IPL** is composed of the other parting line loops located inside the body of the part model. Parting surfaces **(PS)** are defined as the mating surfaces between the core and cavity inserts, which are extended from the **OPL** loop. Shut-off surfaces **(SO)** are the surfaces, which cover all **IPL** loops among the cavity insert, the core insert and undercut features.

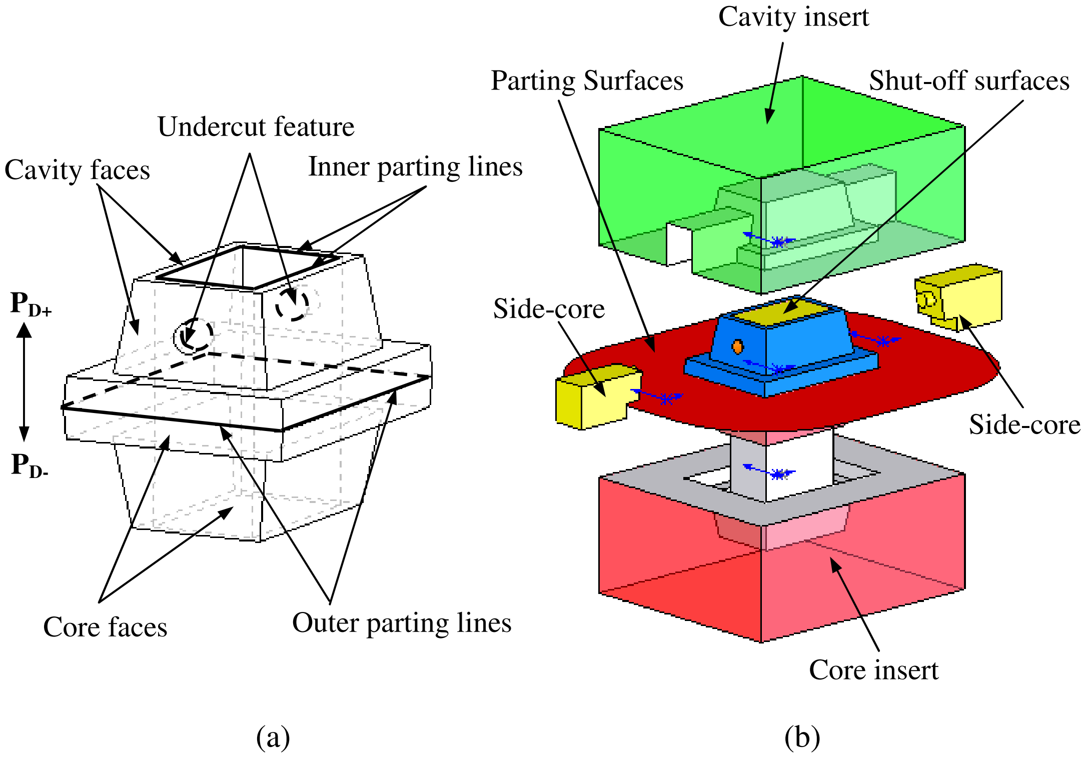


Fig.1.2. Dominations in a parting system

In a moulding, the convex and concave portions are considered as undercut features **(UF)**. If the core, cavity and their inserts cannot mould the undercut features, they would require the incorporation of so called local tools such as side-cores and side-cavities in the mould structure as shown in Fig.1.2 (b). These local tools must be withdrawn by a mechanism prior to the ejection of the moulding. Side-cores and side-cavities are normally removed by slider and lifter mechanisms.

1.2.2 Boundary representation (B-Rep)

In this research, boundary representation (B-Rep) solid models are used as research objects since B-Rep has been widely used in CAD, CAM and CAPP systems. Therefore, the boundary representation (B-Rep) scheme is briefly introduced in this chapter in order to assist the understanding of the algorithms and methodologies presented later in this thesis.

Three types of CAD model representations are commonly used, namely, decomposition, constructive and boundary representations. The decomposition model and constructive model view 3D solids as point sets and seek representations for the point set either by decomposing it or by constructing it from simpler points sets [Mäntylä1988]. The decomposition method uses a regular subdivision of the occupied space of a 3D object. It typically consumes a large amount of memory and has poor accuracy. Constructive solid representation uses the combination of different 3D primitives. It is different from the decomposition representation models primarily in the nature of the method of their combination. In constructive solid representation modeling, solids are described through a combination of some basic primitive elements “glued” together. Boolean operations are used to combine primitives. The other difference between them is in the types of primitives used. In decomposition representation modeling, cubes and rectangular prisms are used, while in constructive methods, any primitive that can be directly represented as a point set can be used. In contrast, the constructive models use much more powerful combination operations. The most common constructive representation method is called constructive solid geometry or CSG.

In contrast to decomposition and constructive models, boundary representation (B-Rep) does not attempt to model a 3D solid as a combination of primitives; it models a solid indirectly by presenting the bounding faces of the solid. The boundaries of a solid are assumed to be partitioned into a finite number of bounded subsets called faces, where each face is, in turn, represented by its bounding edges, and each edge represented by its vertices. In B-Rep, boundary elements (faces, edges and vertices) are combined using Euler operations.

Boundary representation (B-Rep) describes an object by means of faces which enclose it. A boundary representation of an object is a combined geometric and topological description of its boundary, which is partitioned into a finite number of geometric entities, namely, faces, edges, and vertices. Boundary models have wide applicability. They are complete and also unique [Requicha1980]. Furthermore, they are able to present finer object characteristics and are sensitive to local modifications. B-Rep is the closest representation to a geometric model that can be directly used for CAM due to the fact that most manufacturing processes deal with surfaces.

The advantages of the boundary representation (B-Rep) have been well understood above, and the most popular CAD applications use the boundary representation (B-Rep) as the geometry modeling methodology, for instance Parasolids, ACIS, etc. Therefore, the boundary representation (B-Rep) solid models are chosen as the research object in this research. Consequently, all algorithms presented in this thesis are only executable for B-Rep CAD models. However, the methodologies and theories of all the represented algorithms can be applied in models with other types of geometric representation.

The main entities in B-Rep and their relationships are summarized below. These entities and definitions are applied in all the algorithms and methodologies later on.

i. A 3D solid body (**S**) is enclosed by a set of faces (**F**), which are composed of edges (**E**) and vertices (**V**), thus can be expressed as **S** = {**F**, **E**, **V**}, where **S**, **F**, **E**, **V** denote the solid model, set of its faces, edges and vertices respectively.

ii. A face (**F**) is enclosed by a set of edges and contains an external loop and one or more internal loops.

iii. An edge (**E**) can be closed or open. An opened edge has two vertices, so called start point and end point, while a closed edge does not have vertices.

iv. A loop (**L**) is a closed chain of edges bounding it. These edges can be from a single face or multiple faces.

v. A boundary (**B**) is a closed loop of edges. These edges normally represent a hole or gap of a solid model.

In the above definitions, vertices, edges and faces are the primary entities; loops (edge-loop) and boundary are secondary entities. Faces, edges and vertices have their corresponding geometric entities: an edge refers to a curve and a vertex to a point on the object boundary. A face refers to a surface which contains the equations and parameters of the face. In B-Rep, the information required to describe a 3D model is called topological information, which is concerned with the connectivity relationship between pairs of individual entities; and the geometrical information which defines the shape, location and orientation of each primitive entity in the 3D space. Such a data structure contains all the topological and geometrical information related to a solid body. Fig.1.3 illustrates the relationship among different geometric entities applied in the thesis.

i. Vertex⇔Edge relationship (**V⇔E**): An edge has two vertices and every vertex is shared by its corresponding edges. The information of which two vertices belong to the given edge and which two edges sharing the given vertex is important for the identification of parting lines, the generation of parting surfaces and shut-off surfaces, and the design of core and cavity inserts.

ii. Edge⇔Face relationship (**E⇔F**): A face is closed by its bounded edges, and an edge is always shared by two faces of a body. The information of how many edges belonging to the given face and which two faces sharing the given edge is needed in the design activities. This relationship is important in the identification of parting entities, the recognition of undercut features, and the generation of shut-off surfaces.

iii. Face⇔Face relationship (**F⇔F**): The information of the target faces and their adjacent faces is crucial in the automated parting methodology, the recognition of undercut features and the generation of moulding inserts.

iv. Loop⇔Face relationship (**L⇔F**): Every face is bounded by certain edge-loops and each edge-loop belongs to its associated face. The edge-loops and face relationships are crucial for identifying inner and outer parting line loops.

v. Boundary⇔Edge relationship (**B⇔E**): Each boundary is bounded by certain connected edges. The boundary and edge relationships are applied for the identification of parting lines and the generation of parting surfaces and shut-off surfaces.

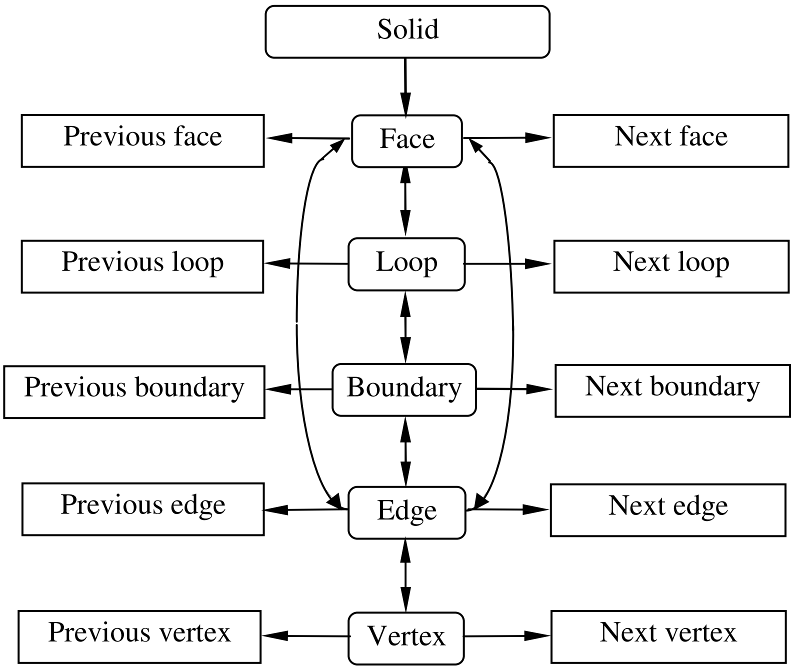


Fig.1.3. Typical winged-edge data structure [modified from Clark1990]

1.3 Bottlenecks of parting systems in CAIMDS and the research objectives

Various automatic parting methodologies for injection moulds have been developed in recent years. Regarding the parting line identification, three approaches were introduced, *i.e.* in-order tree structure [Weinstein1997], graph-based feature recognition [Nee1998] [Ye2001] and the approaches based on face visibility and mouldability [Fu2002]. In the context of parting surface generation, radiating surfaces by offsetting parting lines [Tan1990] [Ravi Kumar2003], and sweeping surfaces along parting lines [Fu2001] are the two approaches reported for creating parting surfaces. Boolean operation and sweeping operation are the two common approaches for the design of the core, cavity inserts and the associated local tools. A more detailed literature review of these methodologies will be provided in the next chapter. These pioneering works have provided good references for this research although there are some limitations from the viewpoint of practical industrial applications. Firstly, the previous parting systems cannot satisfy the parting requirements in terms of speed, quality, functionalities and standardization requirement for complex moulded products since most of the current parting methodologies are incapable of dealing with free-form surfaces, complex geometries and especially geometry imperfections of industry products. In addition, the generated parting surfaces and shut-off surfaces do not always satisfy the moulding requirements in terms of mouldability and manufacturability of injection moulds. Moreover, these parting methodologies do not implement an error correction and feedback mechanism to improve their compatibility and capability for industrial applications since it is impractical for a single parting system to split all products automatically and perfectly. The current parting methodologies also cannot satisfy the design and application of multi-injection moulds due to their complexity in the molding process and their interactive effects. Consequently, injection mould design of complex products becomes challenging and time consuming.

The overall objective of this research is to develop a robust parting system for overcoming the bottlenecks of the previous parting methodologies, and to make the parting methodologies more feasible, powerful and compatible for the practical industry application of injection moulds. More specifically, this research aims to achieve the following features and functionalities:

1) **Automatic identification of parting entities, *i.e.* inner and outer parting lines, cavity and core faces, and undercut features**

   An automated parting methodology based on Face Topology and Mouldability

Reasoning (**FTMR**) has been developed for the automated identification of **IPL** and **OPL**, cavity and core faces, and undercut features for moulded products. The approach is able to deal with the complex geometry and geometric imperfections of part models.

2) **An error correction and feedback system (ECFS)**

   An Error Correction and Feedback System (**ECFS)** has been developed and incorporated into the developed parting methodology for checking and correcting the possible errors during the parting process. The **ECFS** can also enhance the compatibility and capability of the parting system for various practical industrial applications.

3) **Automatic generation of parting surfaces (PS) and shut-off surfaces (SO)**

   Effective algorithms and approaches for generating parting surfaces from outer parting lines and shut-off surfaces from inner parting lines using trimmed NURBS surface have been developed. The generated surfaces are compliant with mould applications because the algorithms consider the manufacturing and mouldability criteria as well as geometrical requirements.

4) **Automatic design the core/cavity inserts and associated local tools**

   Automated approaches and procedures have been developed for the design of the cavity/core inserts and their associated local tools (side-cores and side-cavities) in a single process. Practical industrial requirements were taken into account in the approaches.

5) **Parting approach for multi-injection moulds**

   By applying parting algorithms and approaches previously developed for single injection moulds, a parting approach for multi-injection moulds has been developed. As a result, the sets of cavity, core inserts and associated local tools can be generated corresponding to each moulding injection stage (represented by a set of homogeneous moulding objects) for multi-injection moulds.

Achieving the features and functionalities mentioned above successfully is crucial to the realization of an efficient and powerful parting system for injection mould design applications. Moreover, the error correction and feedback system developed in this research would provide a good reference for visually managing and revising parting entities and features for the various industrial products.

This research is focused on the automated generation of cavity, core inserts and local tools for moulded products. The design of other components of injection moulds is beyond the scope of this research. Moreover, all algorithms developed in this research are restricted to boundary representation (B-Rep) geometric models. In addition, all algorithms for the generation of parting surface and shut-off surface are described using NURBS format since NURBS can represent more complex surfaces (*i.e.* trimmed surfaces) and is compatible with common CAD platforms.

1.4 Layout of the thesis

In the following chapters, the related literature will be first reviewed in Chapter 2. Then, the parting methodology for the determination of parting entities (*i.e.* inner and outer parting lines, cavity and core faces, and undercut features) based on Face Topology and Mouldability Reasoning (**FTMR**) will be presented in Chapter 3. In addition, an Error Correction and Feedback System (**ECFS**) incorporated within the **FTMR** parting approach will be also presented in this chapter. Chapter 4 will introduce the approach to automatically generate parting surfaces from the outer parting line loop using ruled and loft NURBS surfaces. In Chapter 5, a novel and automated approach will be introduced for patching all the inner parting line loops of moulded products using shut-off surfaces. Using previously defined parting entities, generated parting surfaces and shut-off surfaces, the methodologies and procedures to design cavity/core inserts and their local tools (side-cores and side-cavities) will be presented in Chapter 6. Since multi-injection moulding is playing an increasingly more important role in the injection moulding industry, Chapter 7 will introduce a parting approach to generate the sets of cavity/core inserts and their local tools corresponding to each moulding injection stage for multi-injection moulds.

CHAPTER 2

LITERATURE REVIEW

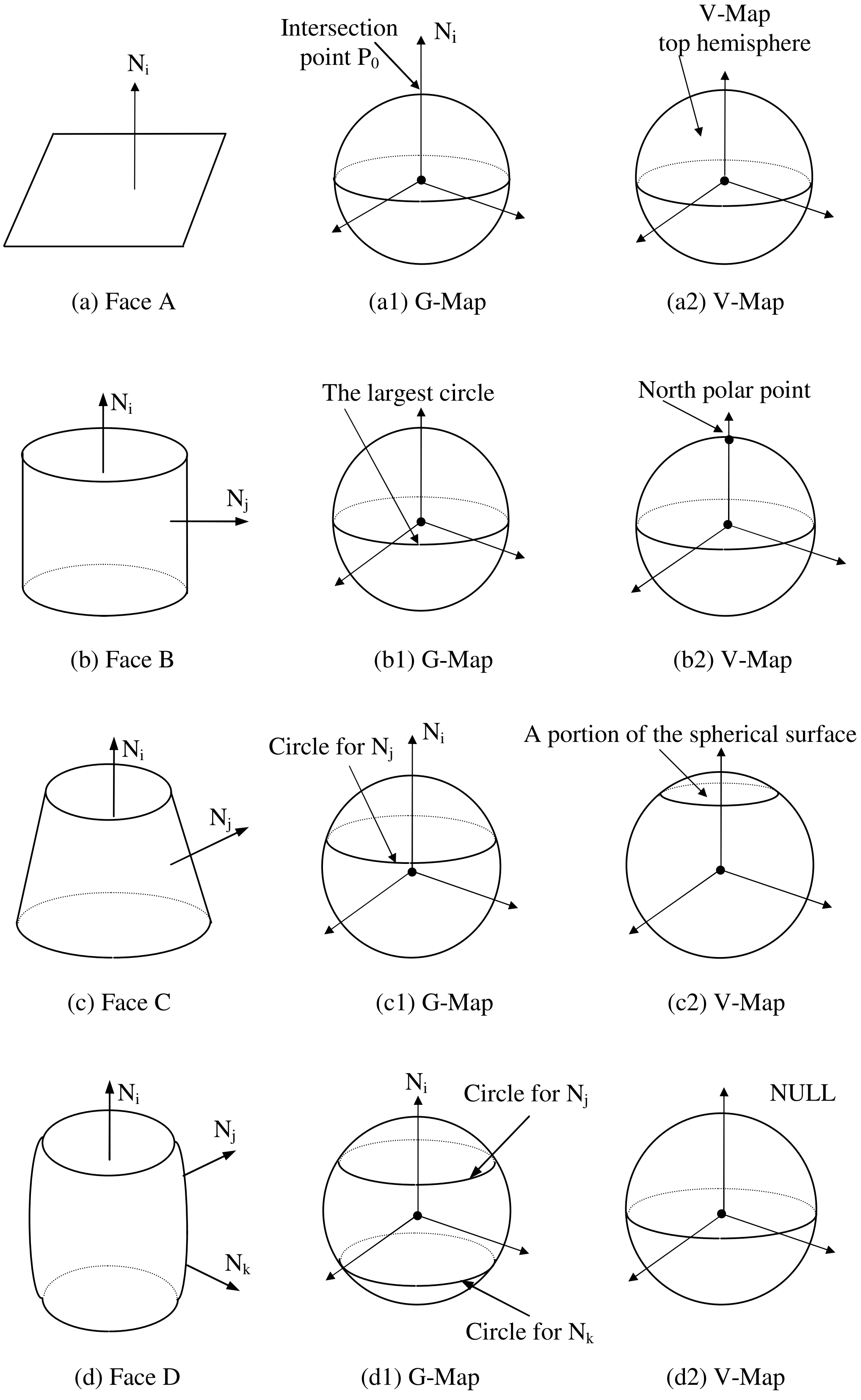
In this chapter, the literature survey is reported. The merits and demerits of the previous work are summarized. The survey of the previous work shows that the problems arising from CAD of injection moulds have generated a great deal of interest and some pioneering work in solving these problems has been conducted. In recent years, much literature on the automatic determination of parting entities, the recognition and extraction of undercut features, and the generation of parting surfaces for injection moulded parts has been published.

2.1 Visibility map (V-Map) and Gauss map (G-Map)

From a basic view point, the parting of a moulding is a process to find those faces which can be drawn from a particular parting direction. As the elemental geometric approach, V-Map and G-Map concepts have been widely applied in determining parting direction, undercut feature direction, and have provided the criteria of mouldability of faces in an injection mould. Gauss introduced the concept of mapping the face normal onto the face of a unit sphere to define the local curvature of a given point [Hilbert1983]. The G-Map is a representation of the face normal. To generate a G-Map, the face normal of any point on a given face F is first transferred to the unit sphere such that the direction is the same as the original normal vector. The transferred vector passes through the centre of the unit sphere and the intersection point of the transferred normal vector with the face of the sphere. When all the intersection points on the unit sphere are produced, they form the G-map of face F.

The V-Map of a face is formed by the points on a unit sphere where the face is completely visible from infinity. Since every point in the V-Map differs from its corresponding point in the G-Map by at most 90 degrees, therefore, the V-Map of a face can be constructed by computing the intersection of hemispheres, each having its pole as a point on the G-Map [Gan1994].

Fig.2.1 shows the generation of G-Map and V-Map for a few common faces. For a planar face A in Fig.2.1 (a), its face normal is N_i. The first step is to transfer N_i to the unit sphere shown in Fig.2.1 (a1) and then determine the intersection point P_0. The intersection point P_0 in Fig.2.1 (a1) is therefore the G-Map of the face A. Since every point in the G-Map has a hemisphere V-Map, the V-Map of a face is the intersection of hemispheres with its pole as a point on the G-Map. The V-Map of face A is the hemisphere as shown in Fig.2.1 (a2). Using the similar approach, Fig.2.1 also shows the results of G-Map and V-Map for cylindrical, conic and drum-shape faces B, C and D respectively. The G-Map for a cylindrical face B is the largest circle of the unit sphere as shown in Fig.2.1 (b1). The V-Map of the face B is therefore represented by the North polar point of the unit sphere. The G-Map and V-Map of a face C (conic face) are the partial sphere and a portion of the spherical surface respectively (as shown in Fig.2.1 (c1) and (c2)) since its face normal N_j has a angle with its central axis. Only the V-Map of a face D in Fig.2.1 (d) is NULL since the G-Map of the face D (in Fig.2.1 (d1)) contains two circles computed from its normal N_k and N_j, which are located onto two halves of the unit sphere respectively. Face D is called straddle face in the thesis.



(a) Face A (a1) G-Map (a2) V-Map

(b) Face B (b1) G-Map (b2) V-Map

(c) Face C (c1) G-Map (c2) V-Map

(d) Face D (d1) G-Map (d2) V-Map

Fig.2.1. G-Map and V-Map of faces

As for an arbitrary free-form face as shown in Fig.2.2 (a), the V-Map and G-Map are determined using the integration of approximation planar normal N_i of all the finest triangulation T_i by tessellation. For each T_i, there is an associated G-Map G_i and corresponding V-Map V_i. The final V-Map V_f of the face is calculated from:

$$V_f = \cap V_i \qquad \text{for } i = 1,2,\ldots\ldots n \qquad (2\text{-}1)$$

If V_f of a face is NULL, the face is a straddle face and not able to be released in a particular parting direction.

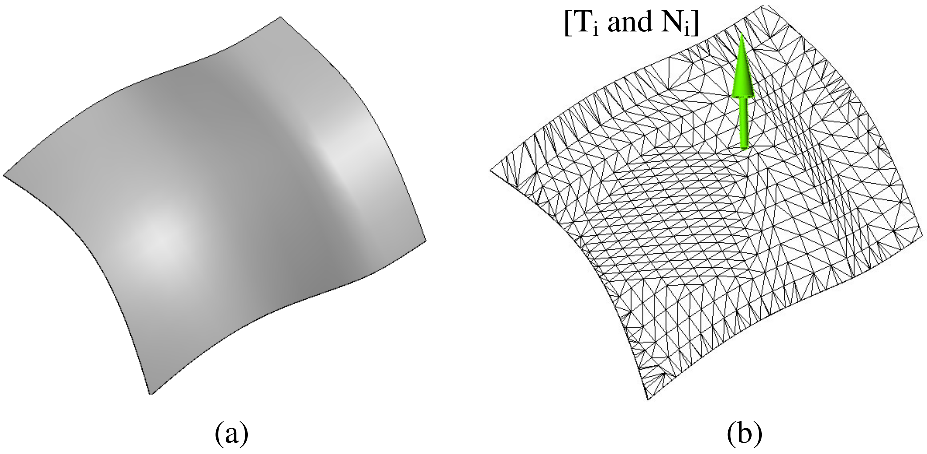


Fig.2.2. G-Map and V-Map of a free-form surface

The G-map and V-map have been used to determine the optimal parting direction in Chen, Chou and Woo's pioneering work [Chen1993] [Chen1995]. Fu *et al.* [Fu2002] also used it to determine the pull directions for undercut features.

2.2 Automatic identification of parting entities

In order to split a moulded product and generate the cavity, core inserts and the associated local tools, all faces of the part model must be fully identified as cavity, core or undercut faces. In addition, inner and outer parting lines must also be determined for the generation of shut-off surfaces and parting surfaces respectively.

All these entities could be determined based on the geometric characteristics and mouldability of a moulded part. Much literature has been published for the automated identification of these parting entities for injection moulded products in recent years.

One of the simple automated approaches to determine the parting lines of a moulding is called in-order tree approach [Weinstein1997]. In this approach, the parting line sets are described in an in-order tree structure which represents the faces formed by the two halves of the mould. The parting line follows the external edges of a set of faces in a given moulding half. Based on the tree branches, the faces are classified into different groups and the edges of each group represent one parting line loop. The optimum parting lines can be determined based on multi-objective criteria, including draw depth, flatness, machining complexity, etc. This approach introduces a simple way to determine parting lines. However, it is not robust in dealing with practical products which models contain free-form surfaces, combined features or faces with geometry imperfections.

As an advanced method for determining parting entities, the graph-based feature recognition approach has been successfully applied in recognizing undercut features and identifying parting lines for injection moulds. In such an approach, an object model is organized into a graph structure using its faces, edges and vertices. In the graph, the geometric entities are expressed as nodes and the connectivity between any of the two entities as arcs. The graph is then split into sub-graphs using graph manipulation algorithms based on their connectivity attributes. These sub-graphs are further mapped with those pre-defined graph patterns derived from known machining and geometric features. Several studies [Chang1990] [Gavankar1990] [Mochizuki1992] and [Henderson1994] have indicated that a successful mapping result represents an undercut feature. With the similar concept, Fu *et al.* [Fu1999] developed a graph-based feature recognition methodology to detect possible undercut features using the rule-based approach based on the definition, classification and criteria of the most common types of machining features of the moulded products. By combining the topological information of the faces into graph-based theory, Ye *et al.* [Ye2000] developed an Attributed Adjacency Graph (AAG) approach to recognize possible undercut features of a moulded part. Each arc of the AAG is assigned a corresponding attribute according to their edge convexity or concavity between the two geometric entities. An attributed adjacency matrix is used to describe the topological relationships of any two faces. Based on these conditions, the algorithm decomposes the AAG into sub-graphs by deleting the nodes, which are only connected by convex edges, and these sub-graphs are then further analyzed to identify the pre-defined feature types. After careful investigation, it was found that the graph-based feature recognition methods are not robust in examining parting entities in two aspects. Firstly, it can only recognize pre-defined features, thus cannot recognize other unknown features. In addition, the approach could fail in the case of geometry imperfections and combined features since the sub-graphs derived from these models are not perfect and therefore their geometric graph cannot be successfully matched with the pre-defined sub-graphs.

Recent research has focused on the potential of employing face visibility and mouldability of moulded parts to automatically determine parting entities. Based on V-Map and G-Map concepts, Tan *et al.* [Tan1990] classified all the part faces into visible and invisible faces based on the face normal and the given parting direction. If the face contains positive vector components, it is visible. On the other hand, it is invisible if the face contains negative vector components in the parting direction.

When an edge is shared by a visible face and an invisible face, it is considered as a parting edge. A series of these tentative parting edges, when properly connected, form the required parting lines. One obvious shortcoming of the approach is that it only considers the visibility of faces and does not consider their mouldability reasoning. Recently, Dhaliwal *et al.* [Dhaliwal2003] described a global accessibility analysis approach for determining the mouldability of a polyhedral CAD model. By computing and examining the exact semi-infinite inaccessibility region (V-Map) of each face represented by triangular facets, a set of possible moulding directions (named as global accessibility cones) for each face was then obtained. The optimum parting direction can be obtained correspondingly. One of the merits of this approach is the effectiveness of the developed algorithms for large-size models. More recently, Rubio *et al.* [Rubio2006] proposed a systematic approach for the automated analysis of the mouldability for a moulded part based on visualization techniques. In this approach, visibility algorithms including slicing by a set of parallel planes, scan line segment and Z-buffer methods were developed to determine V-Map of faces and further identify parting lines. However, the identification of undercut features is not discussed in this approach. Different from the above approaches which examine polyhedral models, Elber *et al.* [Elber2005] presented an aspect graph computation technique to solve mouldability problems for moulded products represented by NURBS surfaces. In their approach, a set of algorithms was developed for computing partitioned viewing sphere, corresponding silhouettes and aspect graph cell decomposition on the sphere of viewing directions. Finally, accurate parting lines were represented using vision curves (parabolic curves, flecnodal curves, and bi-tangency curves). This approach is an extension of the V-Map concept from discrete polyhedral models to curved NURBS models. However, the approach is still arguable in dealing with industrial products due to the slow speed for large size models. Moreover, splitting the associated faces of the original model using generated curved parting lines has not been addressed yet.

In the previous research, Fu *et al.* [Fu2002] developed an approach to determine the parting lines based on the face visibility and associated mouldability. The moulding faces are first classified into three main groups according to their visibility with respect to the given parting direction. Then, an algorithm is developed to generate the edge-loop in different face groups based on their geometric topological relationships and mouldability. The largest edge loop is finally defined as the outer parting line loop.

The research reports reviewed have generated some good results for automated determination of parting entities for mid-complex moulded products. However, the above approaches cannot fully satisfy the parting requirement of functionality and compatibility for industrial products. Firstly, they are not intelligent and robust enough in dealing with industrial products with complex geometry and combined features. Secondly, none of the previous studies have considered model geometric imperfections. Therefore, the methodologies are not robust for the products with geometry imperfections which can commonly appear in industrial products. In addition, inner parting lines were not considered and identified effectively in all the previous approaches. Finally, all the parting methodologies have not addressed an error correction and feedback mechanism to improve the parting results and enhance their compatibility for the various industrial applications.

It is clear that the capability and functionalities of parting systems need to be improved to satisfy the mould design requirements for complex industrial products that are emerging. Better parting results could be obtained if the capability of visibility and mouldability is improved to deal with complex geometry and geometric imperfections. Moreover, the results could also be enhanced through an incorporation of an error correction and feedback mechanism within the parting approach based on face visibility and mouldability reasoning.

2.3 Automatic generation of parting surfaces (PS)

The parting surfaces are the mating surfaces between the core and cavity inserts of a mould. For a given parting direction and defined outer parting lines, parting surfaces can be generated based on the geometrical characteristics of the parting lines and the mouldability of a moulded part. Fu [Fu1998-1] developed an approach to generate the sweeping parting surfaces. In their research, parting lines are classified into three types, *i.e.* flat, step and complex parting lines. If all the parting lines are in the same plane, it is considered as flat parting lines. If the parting lines are not in the same plane, but all lines are linear, it is treated as step parting lines. If the parting lines are not in the same plane and linear, this category represents complex parting lines. With respect to the first two types of parting lines, the parting surfaces are created using extruded surfaces towards the boundaries of a moulded part. In the case of complex parting lines, Fu created the parting surfaces by sweeping a line along the parting lines in three steps. The parting lines are first projected onto a plane perpendicular to the given parting direction. Then, a convex hull of the parting lines is generated. Each edge of the convex hull is projected onto any two adjacent vertical side faces of the mould block in the direction perpendicular to the parting direction but parallel to its side face normal. The direction with the longer projection length is chosen for sweeping the parting edges outwards until they meet the side faces of the mould block. The swept surfaces are the parting surfaces for a moulding.

The obvious disadvantage of sweeping parting surfaces is that the generated parting surfaces are not always suitable for machining of injection moulds since the surfaces could be twisted due to the sweeping algorithm for a complex outer parting line loop. Ruled and loft parting surfaces are able to give better machining property. This will be discussed in a subsequent chapter.

In addition, none of the previous studies have discussed the creation of shut-off surfaces for patching all the inner parting line loops. The shut-off surfaces are necessary for automated generation of cavity, core inserts and undercut features of a moulding.

2.4 Automatic design of core and cavity inserts

Core and cavity inserts are the main components of impression, which form the geometry of a moulded product. Hui and Tan [Hui1992] presented a method to design the core and cavity inserts of a mould with sweeping operations. This method is intelligent and efficient compared to the manual process of determining the geometry of the core and cavity, which is tedious, time-consuming and error-prone. It may sometimes produce incorrect geometry involving interlocking regions between the two halves of the moulds. The procedures to generate the core and cavity of a mould are outlined as follows:

i. Generate a solid by sweeping the moulded part in the parting direction of the mould and determine the core and cavity sides of the swept solid.

ii. Construct a cavity mould block with the required parting surfaces and subtract it with the swept solid at the parting line location.

iii. Generate the second mould block and subtract it with the swept solid from the core side at the parting line location.

iv. Subtract the result of step 2 from that of step 3 with the moulded plates in the closed position to obtain the core block.

Fu *et al.* [Fu2001] introduced a methodology to generate the core and cavity inserts using Boolean difference operation based on the parting direction, parting lines and parting surfaces. The procedure comprises three steps as below.

i. Generation of a containing block, which encloses the moulded part with suitable dimensions.

ii. The Regularized Boolean Difference Operation is carried out between the containing block and the moulded part. After the Boolean operation, the containing block would have an empty space inside.

iii. The hollow block is split into two mould halves using the parting surfaces generated previously. As a result, one half is the core insert, and the other is the cavity insert.

Kwon and Lee [Kwon1991] also presented the algorithms to generate the core and cavity inserts automatically from a B-Rep model. Different from the Boolean and sweeping operation used by Hui, Tan and Fu, Euler operations are used to generate the core and cavity inserts based on the model of a moulded product in B-Rep. The main procedures consist of the following three steps:

i. The faces of a moulded part are first separated into two groups according to the pre-defined parting lines.

ii. Parting surfaces are attached to each face group by applying Euler operations.

iii. The initial mould inserts are generated using two groups of surfaces.

However, the above methodologies did not consider the inner parting lines and the presence of undercut features. Inner parting lines have to be patched so as to fully split a moulding, and the geometry of the undercut features should be retrieved for the side-cores and side-cavities as well.

2.5 Automatic design of local tools

Local tools (e.g., side-cores and side-cavities) are needed to release undercut features in a moulding. Shin and Lee [Shin1993] designed the side-cores and side-cavities based on the interference results between the mould and the part model. In their methodology, the faces of the mould that prevent the part from being withdrawn are identified and these faces are used for generating the side-cores and side-cavities. The primary and the secondary interference faces are detected. Then, the mating surfaces, which include the interference faces, are also selected. The external boundary edges of the mating surfaces are picked to generate the side-cores and side-cavities using Euler operations.

Zhang *et al.* [Zhang1997] presented an algorithm to design the local tools (i.e. side-cores and side-cavities). All the edges of a part, which form the undercut features, are extracted first. Then, the faces of the undercut features are derived from the identified edges and grouped to form individual undercut features. For the depression undercut feature, Boolean operations are used to create the local tool, while a sweeping operation is used to create the local tool for the protrusion undercut features.

Although there are some limitations in the above methodologies, such as too many manual manipulations are needed in the process and the procedures do not consider the generation of cavity/core inserts and incorporated local tools in a single process, etc., these methodologies are good exploratory work in the research of injection mould design.

2.6 Moulding strategy and parting approach for multi-injection moulds

Multi-injection moulding processes allow products made up of heterogeneous materials or components to be moulded, thus can help improve the functionalities and properties. Moreover, a multi-material object is produced as an integral piece, thus it eliminates the assembly process and reduces the lead time. There are two subjects in the parting system for multi-injection moulds, *i.e.,* moulding strategy and parting methodology. Moulding strategy helps to determine the moulding sequence in multi-injection processes, while parting methodology attempts to determine the parting entities, to generate parting and shut-off surfaces, and to design the set of cavity/core inserts and their local tools corresponding to each moulding sequence. Some research on developing moulding strategy has been reported in recent years. However, little work has been conducted on parting methodology of multi-injection moulds.

Kumar and Gupta [Kumar2002] have developed a moulding strategy algorithm for a multi-injection mould using the decomposition approach. In order to find a feasible moulding stage sequence, the multi-material object is decomposed into a number of homogeneous components to find a feasible sequence of homogeneous components that can be added in a sequence to produce the desired multi-material object. The algorithm starts with the final object assembly and considers removing components either completely or partially from the object one-at-a-time such that it results in the previous state of the object assembly. If a component can be removed from the target object leaving the previous state of the object assembly a connected solid, then they consider such decomposition a valid step in the stage sequence. This step is recursively repeated on new states of the object assembly until the product assembly reaches a state where it only consists of one component. When an object-decomposition that leads to a feasible stage sequence has been found, the gross mould for each stage is computed and decomposed into two or more pieces to facilitate the moulding operation.

Li and Gupta [Li2003] have also developed a moulding strategy algorithm for automated design of rotary-platen type of multi-material injection moulds. The approach first classifies the given multi-material object into several basic types based on the relationships among different components of the moulded product. As for each basic type, the moulding strategies are found according to the resulting precedence constraints due to accessibility and disassembly requirements. Then, starting from the last moulding stage, the moulded pieces are identified for every moulding stages recursively.

2.7 Summary

In summary, the research works reviewed have generated some promising results for the automated determination of parting entities for mid-complex moulded products. However, the above approaches cannot fully satisfy the functional and compatible parting requirements for industrial products. Firstly, none of the previous studies have considered the geometric imperfections of a model. Therefore, the approaches are not robust for parts with geometric imperfections which may commonly appear in industrial products. Secondly, the approaches cannot effectively determine all the parting entities (*i.e.* cavity/core surfaces, inner/outer parting lines and undercut features) in a single process with good performance. Thirdly, none of the parting methodologies has provided an error correction and feedback mechanism to improve the parting results and enhance their compatibility for various industrial applications. Little study has been reported on the automated generation of shut-off surfaces from the inner parting line loops of injection moulds. Moreover, the generated parting and shut-off surfaces are not always satisfied with the moulding requirements in terms of mouldability and manufacturability of injection moulds. Finally, the parting methodology for multi-injection moulds, which attempts to design the set of cavity/core inserts and their local tools corresponding to the given moulding sequence, has not been studied yet.

CHAPTER 3

AUTOMATED PARTING METHODOLOGY BASED ON FACE TOPOLOGY AND MOULDABILITY REASONING

Automatic identification of parting entities is one of the most difficult and important tasks of the parting system in CAIMDS. In this chapter, the methodology to automatically identify outer parting lines and inner parting lines, cavity and core faces, and undercut faces based on Face Topology and Mouldability Reasoning (**FTMR**) analysis along the given parting direction is presented. The approach classifies all the faces of a moulded part into different groups based on their geometry visibility and mouldability, and further determines their moulding attributes using the iterative face growth algorithm. The approach has improved the algorithms for applying face visibility and mouldability and can effectively manipulate zero draft faces and Pseudo-Straddle Faces (**PSF**) which may commonly appear in industrial products. In addition, an Error Correction and Feedback System (**ECFS**) has been developed and incorporated into the **FTMR** parting approach. The **ECFS** provides the capability of locating and correcting possible errors during the parting process. Consequently, it enhances the compatibility and capability of the parting approach for complex and varied industrial applications. The approach has been implemented and tested based on the SolidWorks platform. Case studies show that the approach can yield satisfactory results for part models with complex geometry structure and undercut features. At the same time, the presented algorithms are also effective for large-size part models.

3.1. Findings and criteria of parting in moulded products

After studying industrial products and practices carefully, it is found that the parting of an injection moulded product should satisfy a few criteria based on the mouldability, graphic visibility and face geometry topology for the given parting directions $\mathbf{P_{D+}}$ and $\mathbf{P_{D-}}$ respectively. Most of them will be applied in the parting approach introduced in this research.

1) A cavity face can be drawn away along $\mathbf{P_{D+}}$. Therefore, it is visible and not blocked by obstacle faces alone $\mathbf{P_{D+}}$. Similarly, a core face should be visible and there are no obstacle faces along $\mathbf{P_{D-}}$. A potential undercut face is the one which fails to be identified as either a cavity face or a core face. It could be either a straddle face or a face which is not able to be drawn along either $\mathbf{P_{D+}}$ or $\mathbf{P_{D-}}$ [Nee1998].

2) All cavity faces should be connected with each other and form a single group named the cavity face group, while all the core faces form a connective group named the core face group. This is restricted by the structure of a single injection mould, or else the profiles of the cavity side or core side cannot be sewn together successfully.

3) Parting lines are the intersection boundaries among core faces, cavity faces and undercut faces. The outer parting line loop has the maximal projection area at the plane perpendicular to $\mathbf{P_D}$, while inner parting line loops are those which projection areas are smaller [Fu2002].

4) There is only one outer parting line loop and maybe a few inner parting line loops in a moulding. If there are more than one potential outer parting line loops or alternative branches in an outer parting line loop, further optimization is needed to

determine the unique outer parting line loop based on their geometry and machining properties.

5) Undercut features are isolated by inner parting line loops and the outer parting line loop. The geometry characteristics of the parting line loops, which isolate the undercut feature, can imply the local tool mechanism and assist to determine the pull direction of the undercut feature.

6) An entire undercut feature should be determined based on not only the geometry visibility and mouldability of their faces for a given parting direction but also its machining properties and design criteria.

7) A convex undercut feature could be moulded by the cavity and core inserts by means of splitting and redefining the associated undercut faces.

3.2. Face classification based on geometry visibility and mouldability

Essentially, the parting analysis is related to the face visibility and mouldability information along the given parting direction. In this research, the faces of a part model are classified into one of the four categories ($\mathbf{G_+}$, $\mathbf{G_-}$, $\mathbf{G_0}$, and $\mathbf{G_s}$ respectively) according to their geometric characteristics and mouldability. Tab.3.1 describes the definition of each category and the relevant algorithms to determine which category a face should be classified into.

Face categories	**Description of algorithms and criteria**
G+	$N_i \bullet P_{D+} \geq 0$ for i=1, 2, 3, ….. n (N_i is the normal vector of the ith triangular facet of face F; n represents the total number of triangular facets of the face F)
	$F_{\mathrm{int}\,er\sec t}(S, R_j, P_{D+}) \equiv NULL$ for j=1, 2, 3, ….. m ($F_{\mathrm{int}\,er\sec t}$ represents the intersected faces among rays R_j and the body **S** along **P**$_{\mathbf{D+}}$, R_j is a ray at jth position at 2D projection region, and m is the total number of positions of the projected region of face F)
G$_0$	$N_i \bullet P_D \equiv 0$ for i=1, 2, 3, ….. n
G-	$N_i \bullet P_{D-} \geq 0$ for i=1, 2, 3, ….. n
	$F_{\mathrm{int}\,er\sec t}(S, R_j, P_{D-}) \equiv NULL$ for j=1, 2, 3, … m ($F_{\mathrm{int}\,er\sec t}$ represents the intersected faces among rays R_j and the body **S** along the parting direction **P**$_{\mathbf{D-}}$)
G$_s$	$N_i \bullet P_{D+} > 0$ for i=1,2,3,…k (for normal vectors of some triangular facets of face F) $N_j \bullet P_{D-} > 0$ for j=1,2,3,….1 (for normal vectors of other triangular facets of face F)

Tab.3.1. Face classification based on geometry visibility and mouldability (modified from [Fu2002])

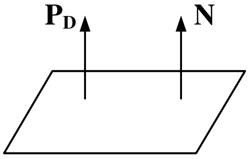
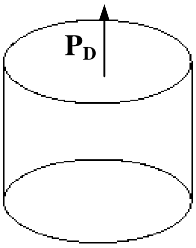
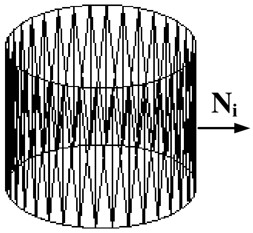
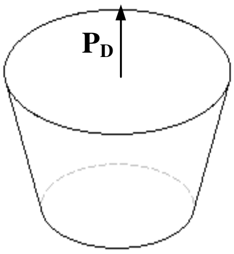
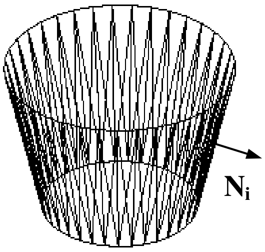
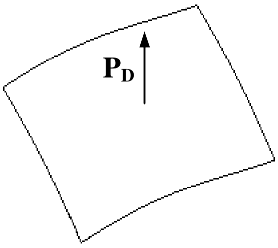
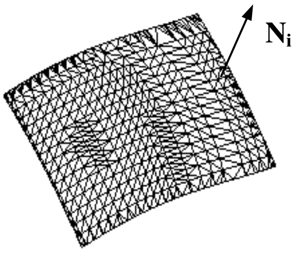
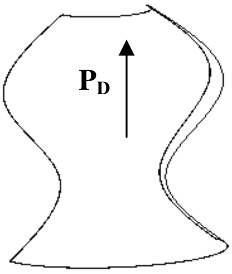
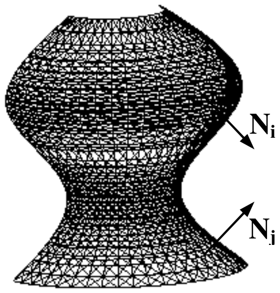
Based on the definitions of different classifications in Tab.3.1, a face in the group of **G+** satisfies two conditions, *i.e.* (a) all the normal vectors of triangular facets contained by the face F have positive vector products towards **P**$_{\mathbf{D+}}$, represented by $N_i \bullet P_{D+} \geq 0$, and (b) no other faces exist inside the 2D projected region of the face F towards **P**$_{\mathbf{D+}}$, represented by $F_{\mathrm{int}\,er\sec t}(S, R_j, P_{D+}) \equiv NULL$. It implies that this face is not only geometrically visible but also not blocked by any other faces towards **P**$_{\mathbf{D+}}$; therefore, it is obviously a potential cavity face. Similarly, a face in the group of **G-** is geometrically visible and will not be blocked by any other faces towards the direction **P_{D-}**; it is, therefore, a potential core face. A face in the group of $\mathbf{G_0}$ is a zero draft face. All the normal vectors of triangular facets contained by this face have a zero vector product with $\mathbf{P_D}$, represented by $N_i \bullet P_D \equiv 0$. It could be further classified as a cavity face, a core face or an undercut face based on the proposed approach and criteria. A face in the group of $\mathbf{G_s}$ is invisible along both $\mathbf{P_{D+}}$ and $\mathbf{P_{D-}}$ geometrically because the normal vectors of the triangular facets contained by this face give positive vector products for both $\mathbf{P_{D+}}$ and $\mathbf{P_{D-}}$ partially, represented by $N_i \bullet P_{D+} > 0$ and $N_j \bullet P_{D-} > 0$. It is obviously a potential undercut face. However, it could still be classified as a cavity face or a core face as long as it satisfies the definition of a pseudo-straddle face or it can be split using an interactive function to be presented later in this thesis.

Determination of the parameter ‘n’ and normal vector ‘N’

In Tab.3.1, the parameter ‘n’ is the total number of triangular facets which are used to determine the geometrical visibility of a target face F along the parting direction $\mathbf{P_D}$. **N** represents normal vectors of the face F. There exist two cases as follows:

1) For a planar face, n equals to 1 and **N** is the normal direction of the plane as shown in Tab.3.2. Here, the triangulation process is not necessary.

2) All other faces are then considered as free-form faces in this thesis. In this case, n is the total number of triangular facets after the triangulation process of a face F in B-Rep. n is given by the triangulation process based on a given accuracy (0.01mm as default) from Parasolid library. Tab.3.2 illustrates the triangulation results of typical geometry faces including cylindrical, conic, free-form and straddle faces. $\mathbf{N_i}$ represents the normal vector of the ith triangular facet.

Face types	Face geometry	Triangulation result	Description of Normal
Planar face		Nil	**N** = plane normal n = 1
Cylindrical face			N_i represents normal vector of ith triangular facet. n = the total number of triangles
Conic face			N_i represents normal vector of ith triangular facet. n = the total number of triangles
Free-form face			N_i represents normal vector of ith triangular facet. n = the total number of triangles
Straddle face			N_i and N_j represent normal vector of ith and jth triangular facet. n = the total number of triangles

Tab.3.2. Determination of normal vectors by the triangulation process for various faces

Determination of parameter 'm' and the array of ray 'R'

Different from 'n', the parameter 'm' is the total number of locations where the array of rays R are generated in order to find the intersected faces with the target face F towards the given parting direction $\mathbf{P_D}$. It determines whether there exist other faces inside the 2D projected region of face F blocking the moving of F towards $\mathbf{P_D}$. An algorithm is developed to determine m and R. Fig.3.1 illustrates the three steps of the algorithm.

Step 1:

The face F is first projected onto a plane perpendicular to the parting direction $\mathbf{P_D}$ as shown in Fig.3.1 (a). As a result, a 2D region is drawn.

$$Rg_P = \{B_i\} \qquad \text{for i=1,2,...numBoundry} \qquad (3\text{-}1)$$

Where Rg_P represents the projected 2D region of face F at a plane perpendicular to $\mathbf{P_D}$, and B_i is the ith closed boundary of the region.

Step 2:

A regular triangle net is created based on the boundary of region from step 1 with respect to the given accuracy (0.01mm as default) as shown in Fig.3.1 (b).

$$N_r = \{T[i][j]\} \qquad \text{for i=0,1,2,... } numX \text{ -1; j=0, 1,2,... } numY \text{ -1} \qquad (3\text{-}2)$$

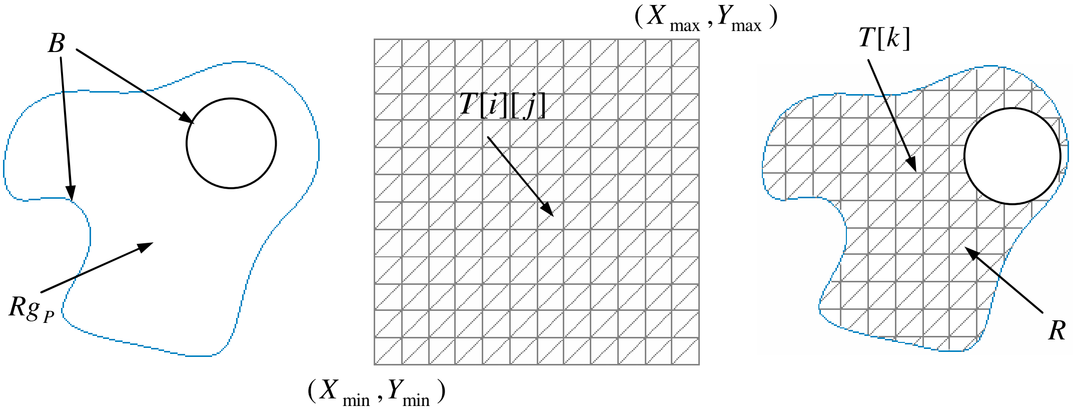
Where $T[0][0] = \{X_{\min}, Y_{\min}\}$ and $T[numX-1][numY-1] = \{X_{\max}, Y_{\max}\}$. $X_{\min}, Y_{\min}$, $X_{\max}$ and $Y_{\max}$ represent the containing boundary of the 2D region Rg_P. *numX* and *numY* are the number of locations at X and Y directions respectively.

Step 3:

All the positions for rays R are constructed by mapping the 2D region Rg_P with N_r shown in Fig.3.1 (c).

$$R = \{T \mid T[k] \in Rg_P, P_{D+} \mid P_{D-}\} \qquad \text{for k=1,2,...m} \qquad (3\text{-}3)$$

Where m is the total number of the mapping points inside the 2D region and the array of rays R is constructed by the position $T[k]$ and the given parting direction.

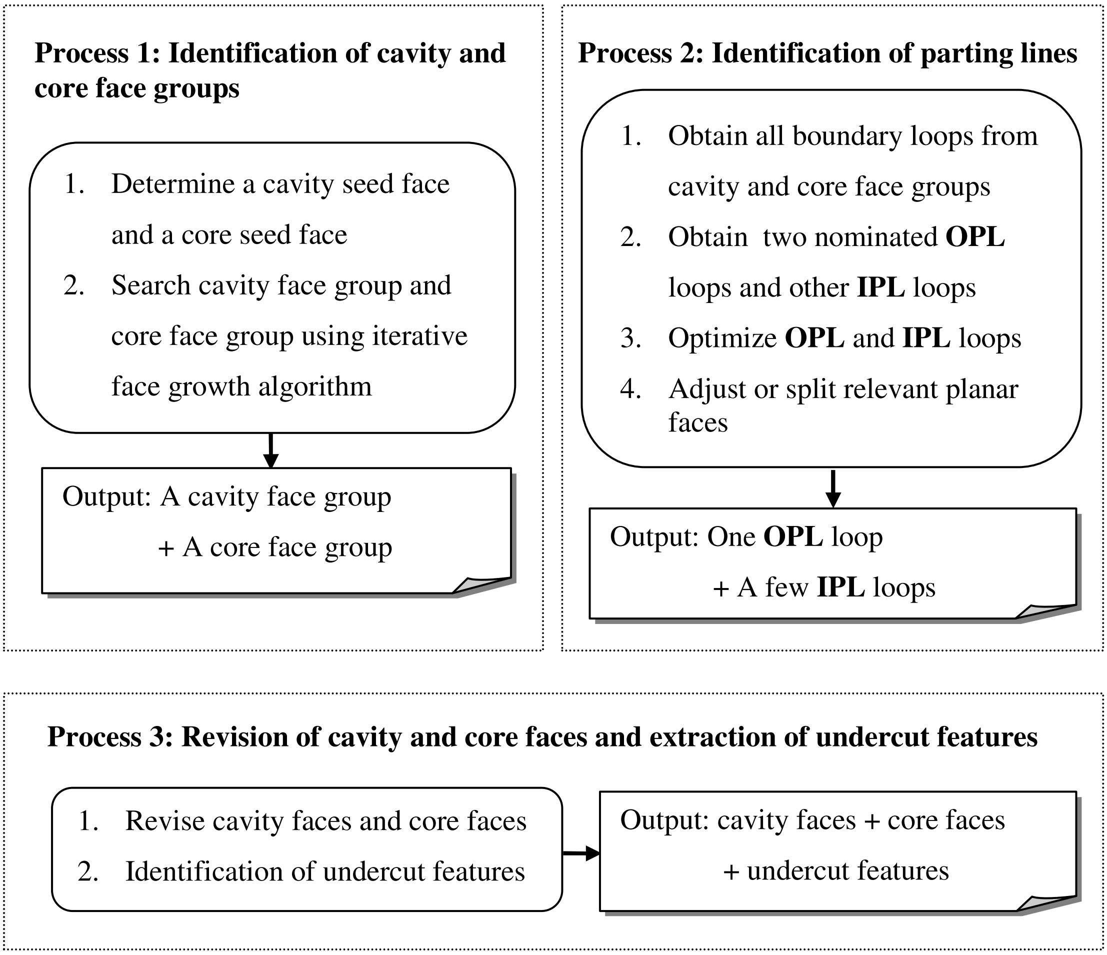


(a) 2D projected region Rg_P (b) Regular triangle net N_r (c) Positions

Fig.3.1. Determination of the parameter 'm' and rays 'R'

3.3. Flow chart of the FTMR parting approach

The **FTMR** parting approach can be divided into three major processes as illustrated in Fig.3.2, *i.e.* (1) identification of cavity and core face groups, (2) identification of parting lines and (3) revision of cavity and core faces and extraction of undercut features. In the first process, a cavity face group and a core face group are identified. Other faces will remain as undefined. In the process of parting line identification, a unique outer parting line (**OPL**) loop and a few inner parting line (**IPL**) loops are identified and optimized based on the proposed algorithms and criteria. In the last process, all cavity faces and core faces are revised based on the result of the parting line loops. All the undercut faces are then extracted and regrouped into different undercut features based on their connectivity and mouldability. As a result, all the faces of a part model are fully defined using a cavity face group, a core face group, several undercut features. In addition, a unique **OPL** loop and a few **IPL** loops are also determined.



Process 1: Identification of cavity and core face groups

1. Determine a cavity seed face and a core seed face
2. Search cavity face group and core face group using iterative face growth algorithm

Output: A cavity face group + A core face group

Process 2: Identification of parting lines

1. Obtain all boundary loops from cavity and core face groups
2. Obtain two nominated **OPL** loops and other **IPL** loops
3. Optimize **OPL** and **IPL** loops
4. Adjust or split relevant planar faces

Output: One **OPL** loop + A few **IPL** loops

Process 3: Revision of cavity and core faces and extraction of undercut features

1. Revise cavity faces and core faces
2. Identification of undercut features

Output: cavity faces + core faces + undercut features

Fig.3.2. Flow chart of the **FTMR** parting approach

As can be seen from Fig.3.2, there are two steps in the process of face group identification. A cavity seed face and a core seed face are first pre-determined. Then, a cavity face group and a core face group are searched using the presented iterative face growth algorithm starting from the cavity seed face and the core seed face respectively. During the process, zero draft faces and pseudo-straddle faces are also manipulated using the proposed algorithms so as to improve the parting results and deal with geometric imperfections. After that, a nominated cavity face group and a nominated core face group are identified as well as those remaining undefined faces in the part model.

Fig.3.2 also describes the four steps of parting lines identification. Firstly, all possible parting line loops from the cavity and core face groups are extracted and classified as cavity parting line loops and core parting line loops. In the step 2, two potential **OPL** loops from the cavity side and the core side are found based on the largest projection area criterion. The remaining parting line loops are then set as **IPL** loops. Subsequently, the nominated **OPL** loops and **IPL** loops are optimized and refined based on the proposed criteria and algorithms presented later in this section to obtain the unique **OPL** loop and other **IPL** loops. Finally, the relevant planar faces are manipulated to obtain the optimal **OPL** loop [explained in 3.6] and other **IPL** loops by adjusting their sides (*i.e.* cavity or core sides) or splitting them into two halves.

3.4. Determination of the cavity seed face and the core seed face

A pre-defined cavity seed face ($F_{CavitySeed}$) and a core seed face ($F_{CoreSeed}$) are used as the seed faces to search for other cavity faces or core faces respectively. Therefore, they must be confirmed cavity face and core face and are determined using the below conditions:

$$F_{CavitySeed} = \{F \mid F \in G_{+} \cap F_{Adjacent} \in G_{+} \parallel G_{0}\} \qquad \text{(Con. 3-1)}$$

$$F_{CoreSeed} = \{F \mid F \in G_{-} \cap F_{Adjacent} \in G_{-} \parallel G_{0}\} \qquad \text{(Con. 3-2)}$$

Where $F_{Adjacent}$ represents all the adjacent faces of a face **F**. If the cavity seed face or the core seed face cannot be obtained automatically, an interactive determination is needed.

3.5. Search cavity and core face groups using the iterative face growth algorithm

Fig.3.3 introduces the iterative face growth algorithm used to search for the cavity face group. The algorithm starts from the cavity seed face **F** ($F_{CavitySeed}$). All adjacent faces $\mathbf{F_a}$ of **F** are then passed through by cycling its boundary loops **L**[iLoop] and associated edges **E**[iEdge] in each loop. If an adjacent face $\mathbf{F_a}$ is verified as a new cavity face, it will be added to the cavity face list pCavityFace and assigned as a new seed face **F** for another round of search iteratively. Subsequently, the cavity face group will propagate gradually from the cavity seed face and all searched faces are connected with each other.

The algorithm to verify whether an adjacent face $\mathbf{F_a}$ is a valid new cavity face is described in Fig.3.4. Obviously, a face cannot be a new cavity face if it is an existing cavity face ($\mathbf{F_a}\in$pCavityFace). Then, $\mathbf{F_a}$ is classified into four cases based on its face categories described in Tab.1, *i.e.* $\mathbf{G_+}$, $\mathbf{G_-}$, $\mathbf{G_0}$, and $\mathbf{G_s}$ respectively. If it is a $\mathbf{G_+}$ face, then $\mathbf{F_a}$ will be identified as a new cavity face. If it is a $\mathbf{G_-}$ face, it is definitely not a new cavity face. If this is a $\mathbf{G_0}$ face, an algorithm to manipulate zero draft faces to be presented later is used to determine whether it can be identified as a new cavity face. If this is a $\mathbf{G_s}$ face, another algorithm to manipulate pseudo straddle faces will be used to determine whether it can be identified as a new cavity face.

Using the algorithms described above, all cavity faces have been identified. Using a similar process, all core faces can also be searched starting from the core seed face. It is noted that all faces of cavity face group and core face group are able to be knitted together respectively.

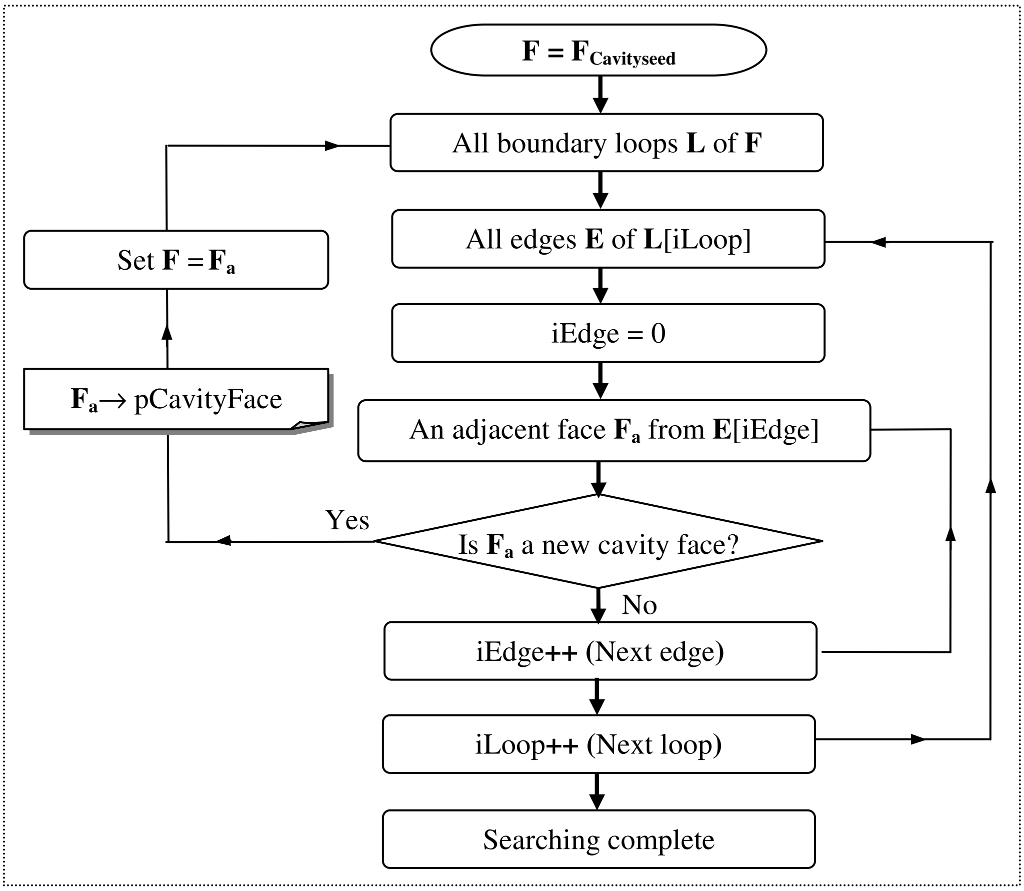

F = $F_{Cavityseed}$
All boundary loops **L** of **F**
All edges **E** of **L**[iLoop]
Set **F** = F_a
iEdge = 0
F_a→ pCavityFace
An adjacent face F_a from **E**[iEdge]
Yes
Is F_a a new cavity face?
No
iEdge++ (Next edge)
iLoop++ (Next loop)
Searching complete

Fig.3.3. The iterative face growth algorithm for searching the cavity face group

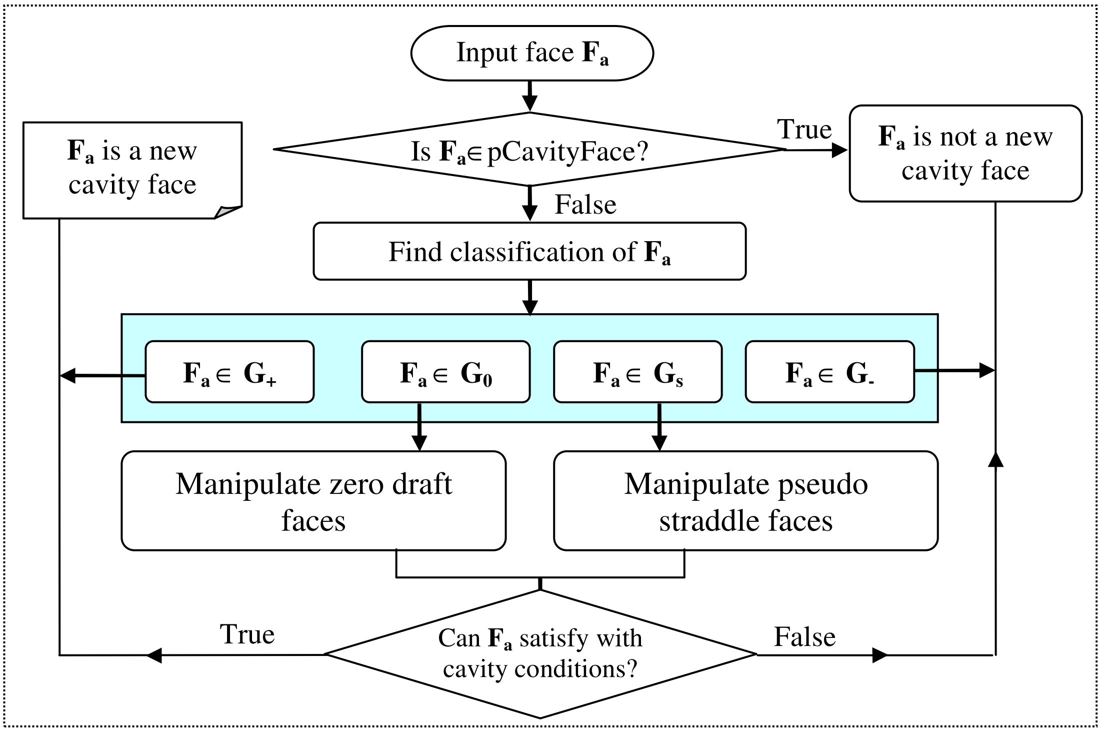
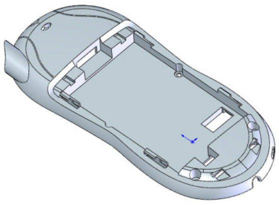
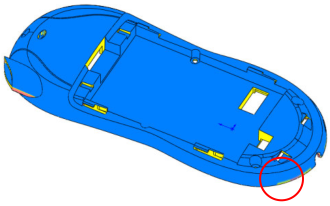
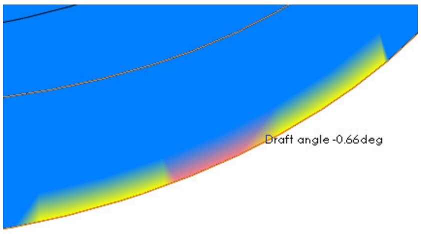
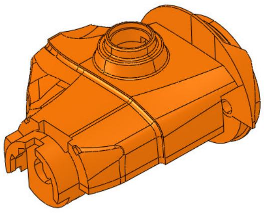
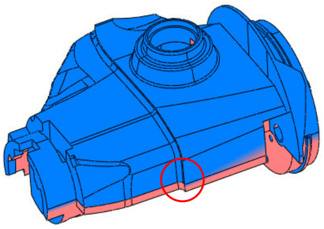
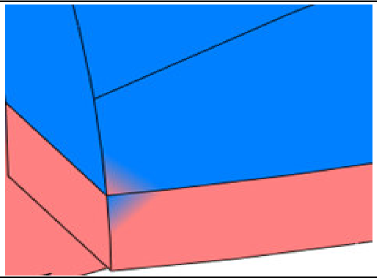
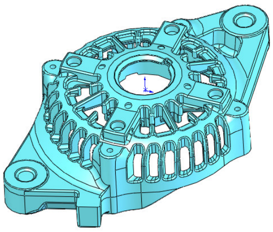
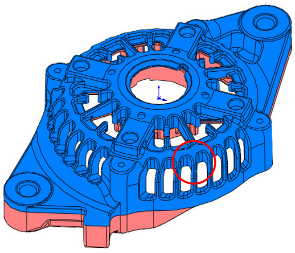
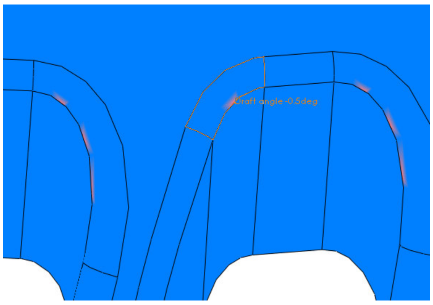

Input face F_a
F_a is a new cavity face
Is $F_a \in$ pCavityFace?
True
F_a is not a new cavity face
False
Find classification of F_a
$F_a \in G_+$
$F_a \in G_0$
$F_a \in G_s$
$F_a \in G_-$
Manipulate zero draft faces
Manipulate pseudo straddle faces
True
Can F_a satisfy with cavity conditions?
False

Fig.3.4. Algorithm to verify the validity of a new cavity face

Manipulating pseudo-straddle faces (PSF)

The first column of Tab.3.3 shows three mid-complex moulded products. The second column gives the draft analysis results of all the faces of the products. Blue color means positive draft; pink means negative draft, and yellow zero draft. From the zoomed views shown the third column, it can be seen that the imperfect draft of some faces of these products. Mathematically, these highlighted faces are undercuts. However, they should be moulded with the cavity or core insert practically. These draft or geometric imperfections are common in complex moulded products caused by the design of free-form surfaces, blending surfaces and data transfer among different CAD applications.

Moulded products	Draft analysis results	Faces with imperfect draft
		
		
		

Tab.3.3. Draft imperfections of moulded products (Models from Manusoft Technologies Pte Ltd)

Therefore, a concept called pseudo-straddle faces (**PSF**) is introduced to improve the parting result for industrial products with geometric imperfections. Based on the definition of **PSF** in this thesis, a **G_S** face could be determined as a **PSF** and further identified as a cavity face or a core face by only adjusting the given moulding accuracy and geometric tolerance at **P_{D+}** or **P_{D-}** direction respectively.

To define a **PSF**, four key parameters are first defined below:

Parameter 1: $A_+ = \sum A_i$ for i=1,2,…k. Where A_i is the area of ith triangular facet, which normal vector is N_i and satisfies $N_i \bullet P_{D+} > 0$. Therefore, A_+ is the total area of the face where its normal components are towards **P_{D+}** and thus can be drawn at **P_{D+}**.

Parameter 2: $A_- = \sum A_j$ for j=1,2,…l. Where A_j is the area of jth triangular facet, which normal vector is N_j and satisfies $N_j \bullet P_{D-} > 0$. Thus, A_- is total area of the face which normal components are towards **P_{D-}** and thus can be drawn at **P_{D-}**.

Parameter 3: $\alpha_+ = \max\{\alpha_i\}$ for i=1,2,…k. Where $\alpha_i = N_i \bullet P_{D+}$. α_i is the area of ith vector product between **P_{D+}** and face normal vector of the ith triangular facet, which normal vector is N_i and satisfies $N_i \bullet P_{D+} > 0$. Therefore, α_+ represents the closest face normal to **P_{D+}**.

Parameter 4: $\alpha_- = \max\{\alpha_j\}$ for j=1,2,…l. Where $\alpha_j = N_j \bullet P_{D-}$. α_j is the area of jth dot product between **P_{D-}** and face normal of the jth triangular facet, which normal vector is N_j and satisfies $N_j \bullet P_{D-} > 0$. Thus, α_- represents the closest face normal to **P_{D-}**.

If a straddle face satisfies one of the two conditions below, then it is a **PSF**.

$$A_{-} < AOA \cap \alpha_{-} < COS(AOD) \cap F_{\text{int}\,er\,\text{sec}\,t}(S, R_j, P_{D+}) \equiv NULL \quad \text{(Con. 3-3)}$$

$$A_{+} < AOA \cap \alpha_{+} < COS(AOD) \cap F_{\text{int}\,er\,\text{sec}\,t}(S, R_j, P_{D-}) \equiv NULL \quad \text{(Con. 3-4)}$$

Where AOA (2 mm^2 by default) is the moulding area accuracy and AOD (0.5deg by default) is the draft angle accuracy required by the moulded product.

If a **PSF** satisfies Con. 3-3 and is adjacent to any defined cavity face, it can be identified as a cavity face. While, if a **PSF** satisfies Con. 3-4 and is adjacent to any defined core face, it could be identified as a core face. If the face does not satisfy either Con. 3-3 or Con. 3-4, it will be considered as a real undercut face in the automated process. To attempt removing undercut features, an interactive function is provided to examine their mouldability and further attempt to split them into the cavity and core sides (presented in the later section 3.7).

Manipulating zero draft faces

If a zero draft face **F** can move away towards **P_{D+}**, it is a potential cavity face, while it is a potential core face if it can move towards **P_{D-}**. If a zero draft face is partially blocked by other faces, it will be split into two faces based on the projection lines. Then, they can be identified as cavity and core faces correspondingly. The algorithm is described below.

If **F** satisfies $F \in G_0 \cap F_{\text{int}\,er\,\text{sec}\,t}(S, R_j, P_{D+}) \equiv NULL$, then it can be set as a new cavity face (*i.e.* **F**∈pCavityFace) since no faces blocking **F** move away from **P_{D+}**. While if **F** satisfies $F \in G_0 \cap F_{\text{int}\,er\,\text{sec}\,t}(S, R_j, P_{D-}) \equiv NULL$, then it will be a new core face (*i.e.* **F**∈pCoreFace) since no faces are blocking **F** to move away from **P_{D-}**.

If the 2D projection of $F_{\mathrm{int}\,er\sec t}$ is smaller than the 2D projection of **F**, then **F** is split using the projection lines of $F_{\mathrm{int}\,er\sec t}$ onto **F** and two faces **F₁** and **F₂** are generated correspondingly. Subsequently, **F₁** and **F₂** will be further identified as cavity or core faces based on previous conditions, *i.e.* $F_{\mathrm{int}\,er\sec t}(S, R_j, P_{D+}) \equiv NULL$ or $F_{\mathrm{int}\,er\sec t}(S, R_j, P_{D-}) \equiv NULL$ respectively.

The approach to find $F_{\mathrm{int}\,er\sec t}$ is similar to the method described in section 3.2. The only difference is the way to generate the array of rays R because the projected region Rg_P from a **G₀** face onto a plane perpendicular to the parting direction is now a 2D curve rather than a 2D region. In this case, a region Rg_P is generated by offsetting the 2D projected curve of **F** by a length AOL along the face normal direction. AOL (1 mm by default) is a pre-defined length of gap for a moulding. Subsequently, the rays R can be generated from this offset region using the previous approach as illustrated in Fig.3.1.

3.6. Identification of parting lines

After the cavity face group and the core face group have been identified in the previous process, all parting line loops can be extracted accordingly. As shown in Fig.3.3, the identification of parting lines includes four steps:

<u>Step 1</u>:

Since all the faces of the cavity face group and the core face group are connected with each other, all boundary loops on the cavity and core side can be easily extracted and stored into two lists, *i.e.* plCavityLoop and plCoreLoop respectively.

Step 2:

Two potential **OPL** loops are then found from plCavityLoop and plCoreLoop by comparing the projection area A of each loop onto a plane perpendicular to $\mathbf{P_D}$. The loop with the maximum projection area is the preferred **OPL** loop. The equation to compute the projection area A of a loop is

$$A = \sum_{k=1}^{n}\sum_{i=0}^{m-1} (x_i y_{i+1} - x_{i+1} y_i) \tag{3-4}$$

Where A is the projected area of the loop. n represents the number of edges in the loop. (x_i, y_i) is the coordinate of ith points in the kth edge. m is the number of represented points of the kth edge. As for a linear edge, m equals to 2, and for other kinds of edges, m is determined using the tessellation process on the edge. It is noted that a coordinate transformation from $\mathbf{P_D}$ to Z axis must be done using Eq.3-4 if $\mathbf{P_D}$ is different from the Z axis in a moulding. In this thesis, $\mathbf{P_D}$ is assumed the same as Z axis if there is not special indication.

As a result, a cavity **OPL** loop oplCavityLoop and a core **OPL** loop oplCoreLoop are found. The other parting line loops are therefore known as **IPL** loops.

Step 3:

It is known that there is only one **OPL** loop in a moulding. Thus, oplCavityLoop and oplCoreLoop have to be compared and optimized further to obtain the unique **OPL** loop. If they are the same, oplCavityLoop will be set as the final **OPL** loop; or else, the favorite **OPL** loop is determined according to the flatness factor C_f [Ravi1990] expressed using the equation below.

$$C_f = L_{2D} / L_{3D} \tag{3-5}$$

$$L_{3D} = \sum L_i \qquad \text{for i=0,1,2...nEdge} \qquad (3\text{-}5\text{-}a)$$

$$L_{2D} = \sum L_i^{\Pr ojection} \qquad \text{for i=0,1,2...nEdge} \qquad (3\text{-}5\text{-}b)$$

Where L_{3D} represents the total length of the parting lines in the 3D space, while L_{2D} represents the total length of the parting lines at the 2D projection onto the plane perpendicular to $\mathbf{P_D}$. L_i is the length of ith edge in the loop and nEdge is the total number of edges of the loop. The loop which has the larger C_f will be the preferred **OPL** loop.

As for the other nominated **IPL** loops, there are three cases, and each case will be processed respectively.

Case 1: If a cavity **IPL** loop or a core **IPL** loop is not connected and does not intersect with any other core **IPL** loops or a cavity **IPL** loop, it is set as the final **IPL** loop.

Case 2: If a cavity **IPL** loop or a core **IPL** loop is exactly same as another core **IPL** loop or a cavity **IPL** loop, then the core **IPL** loop will be removed and the cavity **IPL** loop is maintained as the final **IPL** loop.

Case 3: If a cavity **IPL** loop or a core **IPL** loop is intersected with another core **IPL** loop or a cavity **IPL** loop, then C_f is used to determine the optimal **IPL** loop.

Step 4:

All the adjacent zero draft faces of **OPL** or **IPL** are reviewed in this step. An algorithm is executed to obtain optimal (with large C_f) parting lines by adjusting these faces' attributes (*i.e.* cavity side or core side) or by splitting them into two halves using the lines generated from the associated vertices of the parting lines, which isolate these faces from the other parting lines. C_f is always used to determine the optimal parting lines.

As a result, the unique **OPL** loop and a few **IPL** loops are finally identified. More effectively, all the parting line loops also maintain the information of their preferred mould side, *i.e.* cavity side or core side, and the information will be able to assist the determination of the characteristics of undercut features in later design of local tools.

3.7 Error correction and feedback system (ECFS)

It is impractical to expect a single parting approach to split any moulded products with perfect results. For instance, an undercut feature could be moulded by the cavity and core inserts after the further splitting. The boundary shape of an undercut feature could also need to be adjusted. In this thesis, an error correction and feedback system (**ECFS**) is therefore introduced to improve the capability and compatibility of the presented **FTMR** approach. If a moulded product cannot be completely split by the **FTMR** approach for any reason, the **ECFS** can assist in locating the places where the parting process is not well performed, correcting the errors correspondingly, modifying and highlighting parting entities for different purposes. The system is composed of a visible feature manager tree (**FMT**) and some built-in functionalities.

3.7.1 Feature manager tree (FMT)

As shown in Fig.3.5, the feature manager tree (**FMT**) is designed to manage all the parting entities and features of a part model during the parting processes. All these parting entities are organized into two major categories, *i.e.* body faces and parting lines. The body faces comprise of cavity faces, core faces, undercut faces and undefined faces. Furthermore, undercut faces are grouped into different isolated undercut features, which faces are connected with each other and not connected with the faces of other undercut features. The parting lines are classified as inner and outer parting lines. All the parting lines are further organized using edge loops which edges are connected with each other. There is normally one loop under the outer parting line node and a few loops under inner parting lines node.

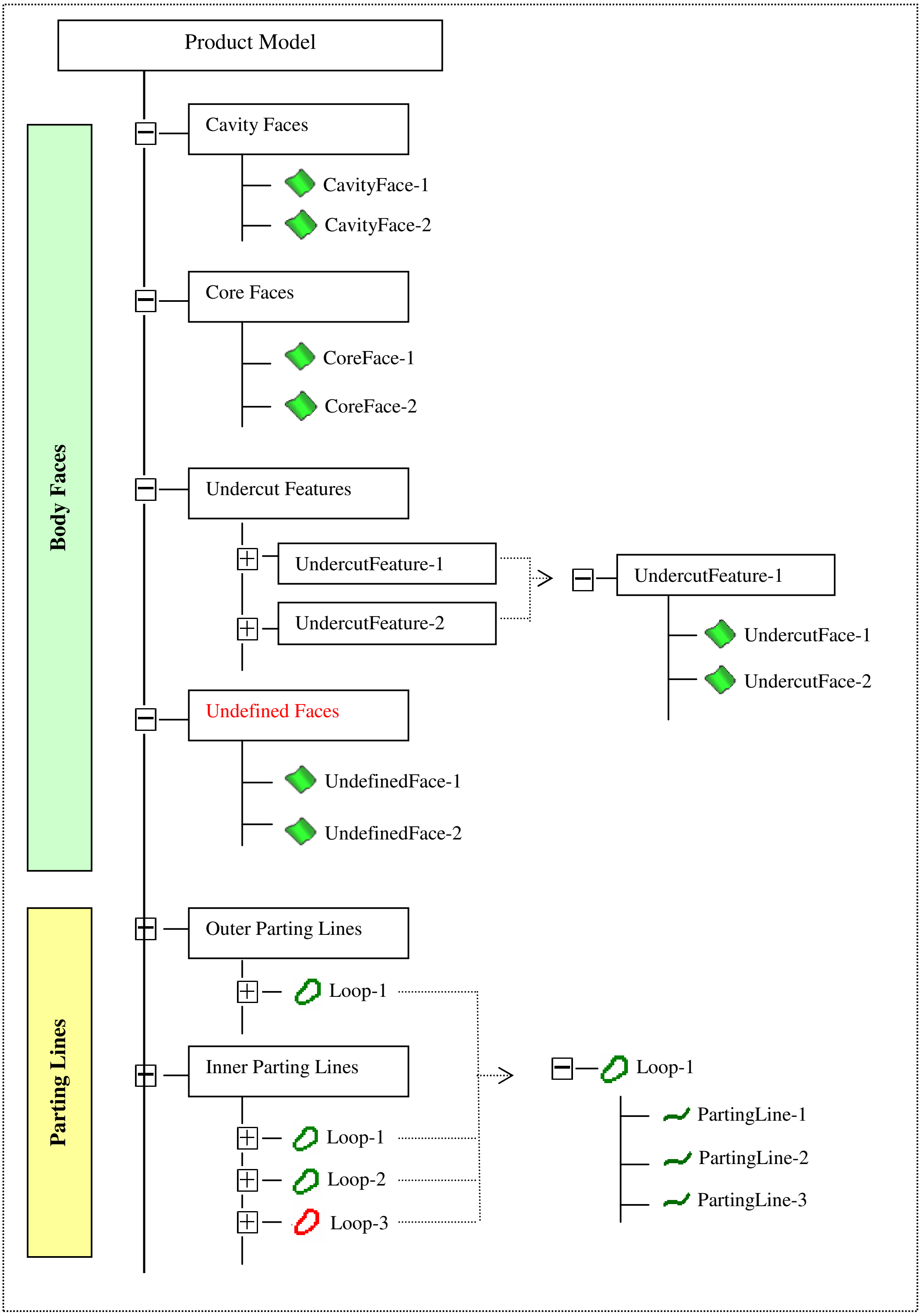


Fig.3.5. The structure of feature manager tree (**FMT**)

3.7.2 Built-in functionalities for the ECFS

Besides the typical functions available in B-Rep, several built-in functionalities have been developed for the **ECFS**. Fig.3.6 illustrates the relationships among all parting entities and how built-in functionalities work with them. These functionalities are classified into two categories, *i.e.* real-time functions and interactive functions.

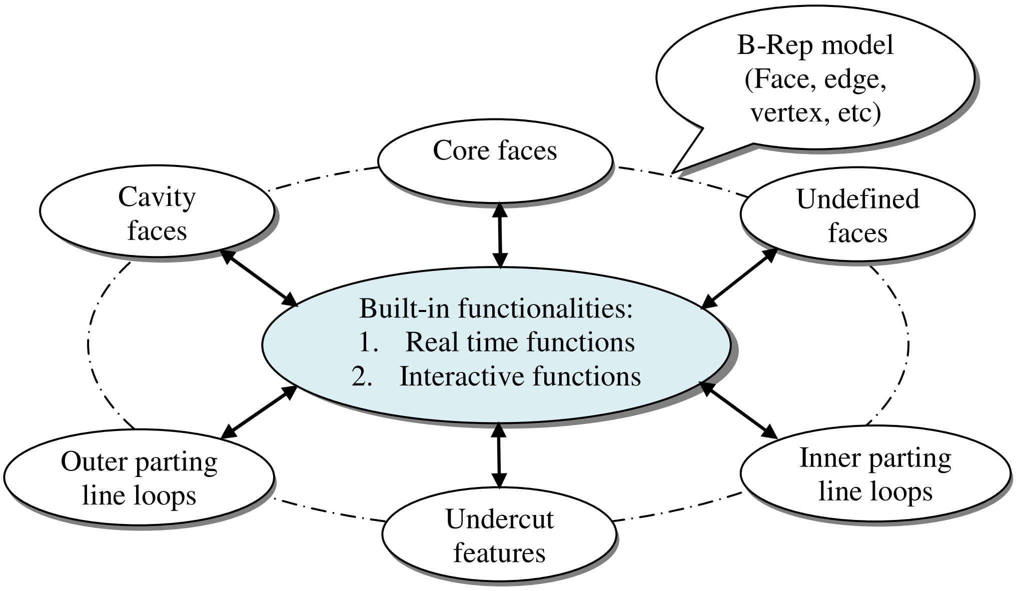


Fig.3.6. The relationships among parting entities and built-in functions

Real-time functions are developed for updating the feature tree, checking and highlighting the errors correspondingly. All the entities of **FMT** will be automatically updated during each step of the parting process using real-time functions. At the beginning of the parting process, there are only undefined faces in **FMT** and other nodes are empty. After the cavity and core faces have been identified, they can be seen from **FMT**. Similarly, outer parting lines and inner parting lines will be added to **FMT** and organized using loops after the parting lines loops have been identified and optimized. Finally, undercut features and their associated faces will appear in the tree after they are extracted and regrouped using the **FTMR** approach. If there still exist any items under undefined faces node, which implies the incompletion of the parting process, the node (Undefined Faces) will be highlighted (as shown in Fig.3.5) and designers may consider checking and redefining these undefined faces. If there are more than one outer parting line loops in the feature tree, designers would probably need to check them and make further decisions because a moulding does not allow two or more outer parting line loops. In the content of inspection of errors, all parting line loops under outer parting line node and inner parting line node will be checked whether they are closed. The parting line loops will be highlighted in the feature tree using red nodes (as shown in Fig.3.5) if they have not been closed yet.

Since undercut features are always the major cost factor and the arguable points for mould designs, interactive functions can assist to improve or modify the results of undercut features. It has been found that a convex undercut feature could be moulded using the cavity and core inserts by splitting and redefining the associated faces of the undercut feature. Therefore, a function for splitting undercut features (as described in Fig.3.7) is developed to examine the demouldability of an undercut feature and further attempt to split associated faces into cavity and core sides. In the function, the silhouette algorithm is applied to split a straddle face based on its zero draft curve and this is provided in the Parasolid library.

The **ECFS** also provides interactive information functions for the part design and modification from the view of mould design. The functions are developed for inspecting and highlighting improper geometry entities in terms of draft angle, geometric imperfection, improper design, etc. Designers can change the definition of the associated entities or make part modifications correspondingly.

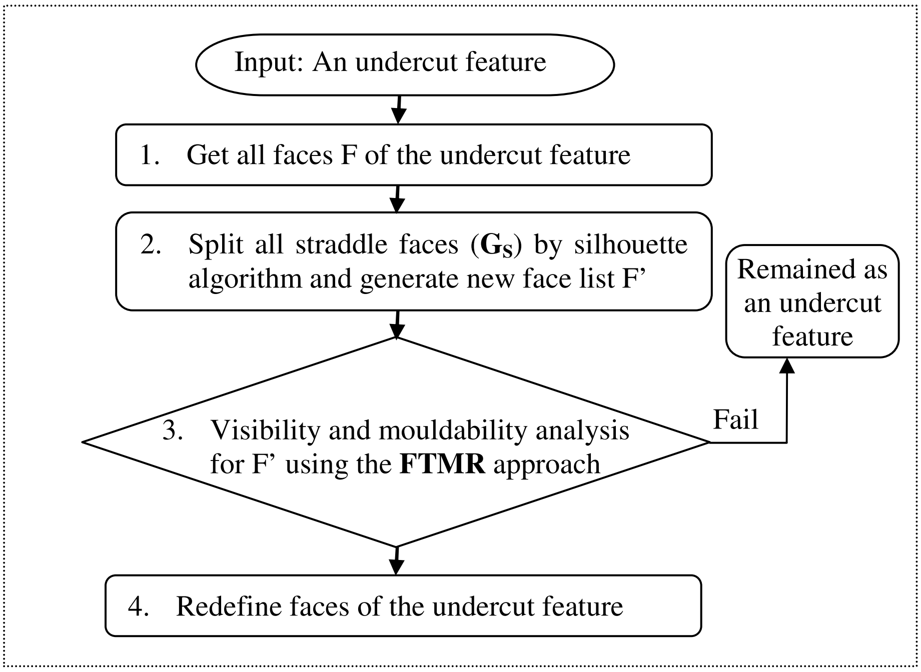

Input: An undercut feature
1. Get all faces F of the undercut feature
2. Split all straddle faces (G_S) by silhouette algorithm and generate new face list F'
3. Visibility and mouldability analysis for F' using the FTMR approach
Fail
Remained as an undercut feature
4. Redefine faces of the undercut feature

Fig.3.7. Algorithm for splitting an undercut feature

3.8. Implementation and case studies

The **FTMR** approach introduced in the thesis has been implemented based on the SolidWorks platform. Here, two case studies are given. One part has no geometric imperfections while the other has some geometric imperfections.

3.8.1. Case study 1

Fig.3.8 (a) shows a plastic moulded part with a few inner parting line loops and an undercut feature. It has perfect draft angle and no geometric imperfections. Based on the **FTMR** approach, a cavity seed face and a core seed face are first identified as shown in Fig.3.8 (b) and (c). Next, the cavity face group and the core face group are searched in turn using the iterative face growth algorithm. The results are shown in Fig.3.8 (d) and (e). In the process of parting line identification, all parting line loops are pre-determined from the cavity face group and the core face group. As shown in Fig.3.8 (f), two branches of **OPL** loops exist. Therefore, the optimization process is executed to remove one of the branches based on the flatness criterion (with large C_f). Fig.3.8 (g) shows the result of **OPL** loop and **IPL** loops. The final result after the extraction of the undercut features is shown in Fig.3.8 (h).

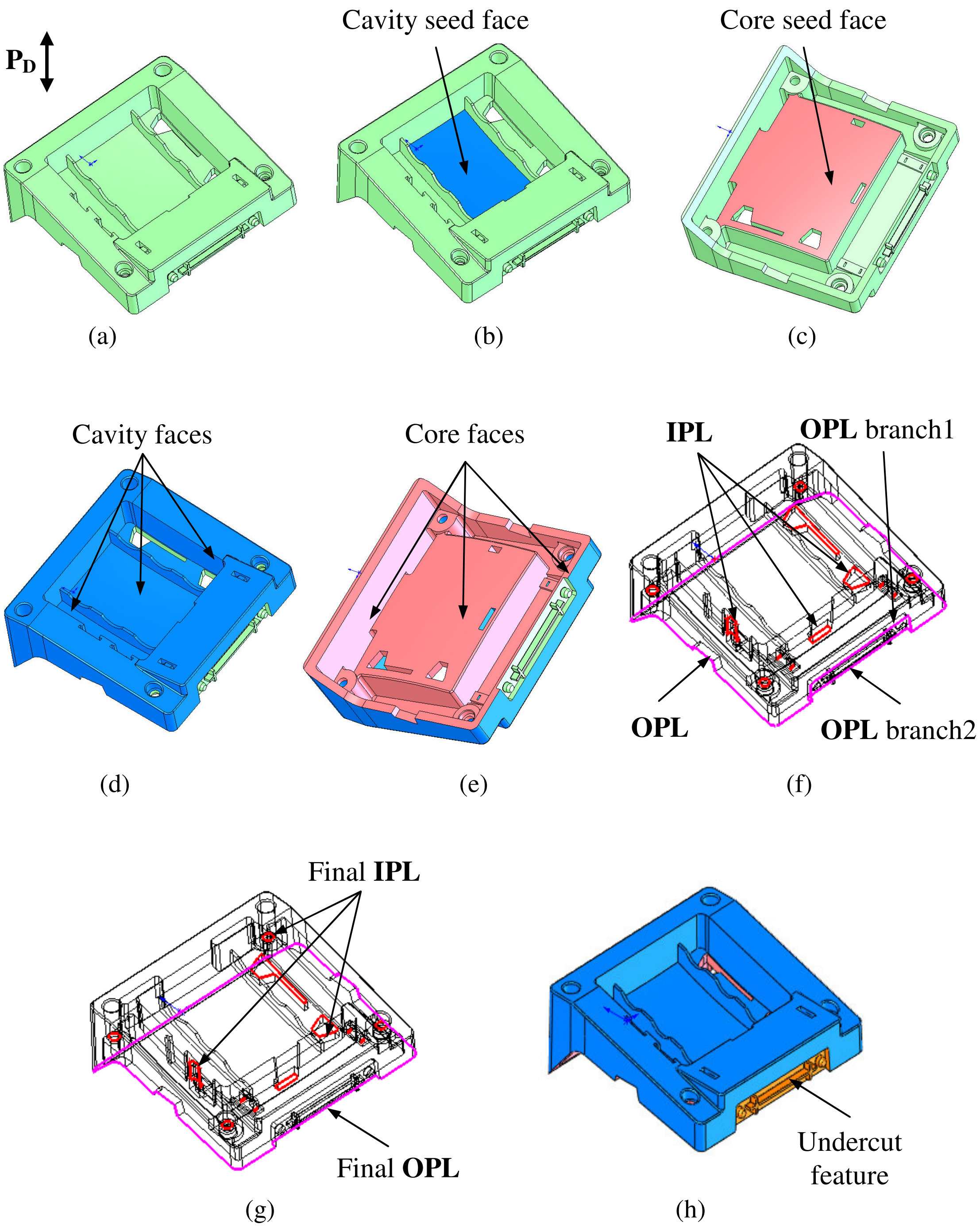


Fig.3.8. Case study 1 for the **FTMR** parting approach

3.8.2. Case study 2

Fig.3.9 (a) shows another plastic moulded part with geometric imperfections. Some zero draft faces are located at the vertical sides of the slots. It also has a few imperfect draft faces as shown in Fig.3.9 (b). As shown in Fig.3.9 (b1), these faces are not perfect **G.** faces since a small region of the faces does not satisfy $N_i \bullet P_{D-} > 0$, but $N_i \bullet P_{D+} > 0$. However, they satisfy the definition of a pseudo-straddle face proposed in this thesis. Fig.3.9 (a1) shows the real-time result of **FMT**, in which there are only undefined faces.

Based on the **FTMR** approach, cavity and core seed faces are first obtained as shown in Fig.3.9 (c) and (d) respectively. Then, the cavity and the core face groups are generated. Those zero draft faces are identified as cavity faces in terms of the proposed criteria. Fig.3.9 (e) shows the cavity face group. Furthermore, the pseudo-straddle faces are reasonably identified as core faces based on the proposed algorithm for manipulating pseudo-straddle faces as shown in Fig.3.9 (f). The result of **FMT** is shown in Fig.3.9 (f1), in which there are a number of cavity faces in the cavity face node and core faces in the core face node, as well as the remaining undefined faces. The existence of undefined faces implies that the parting process is still incomplete. Subsequently, the **OPL** loop and **IPL** loops are identified from the results of cavity and core face groups. There are an **OPL** loop and a few **IPL** loops. The result of parting lines is highlighted in Fig.3.9 (g), and the result of **FMT** is shown in Fig.3.9 (g1) correspondingly. Finally, all the remaining undefined faces are identified as a single undercut feature since they are isolated by two **IPL** loops and are connected with each other. Fig.3.9 (h) and (h1) show the final results of the parting model and **FMT** respectively. The part has been well split since there are no undefined faces and open parting line loops.

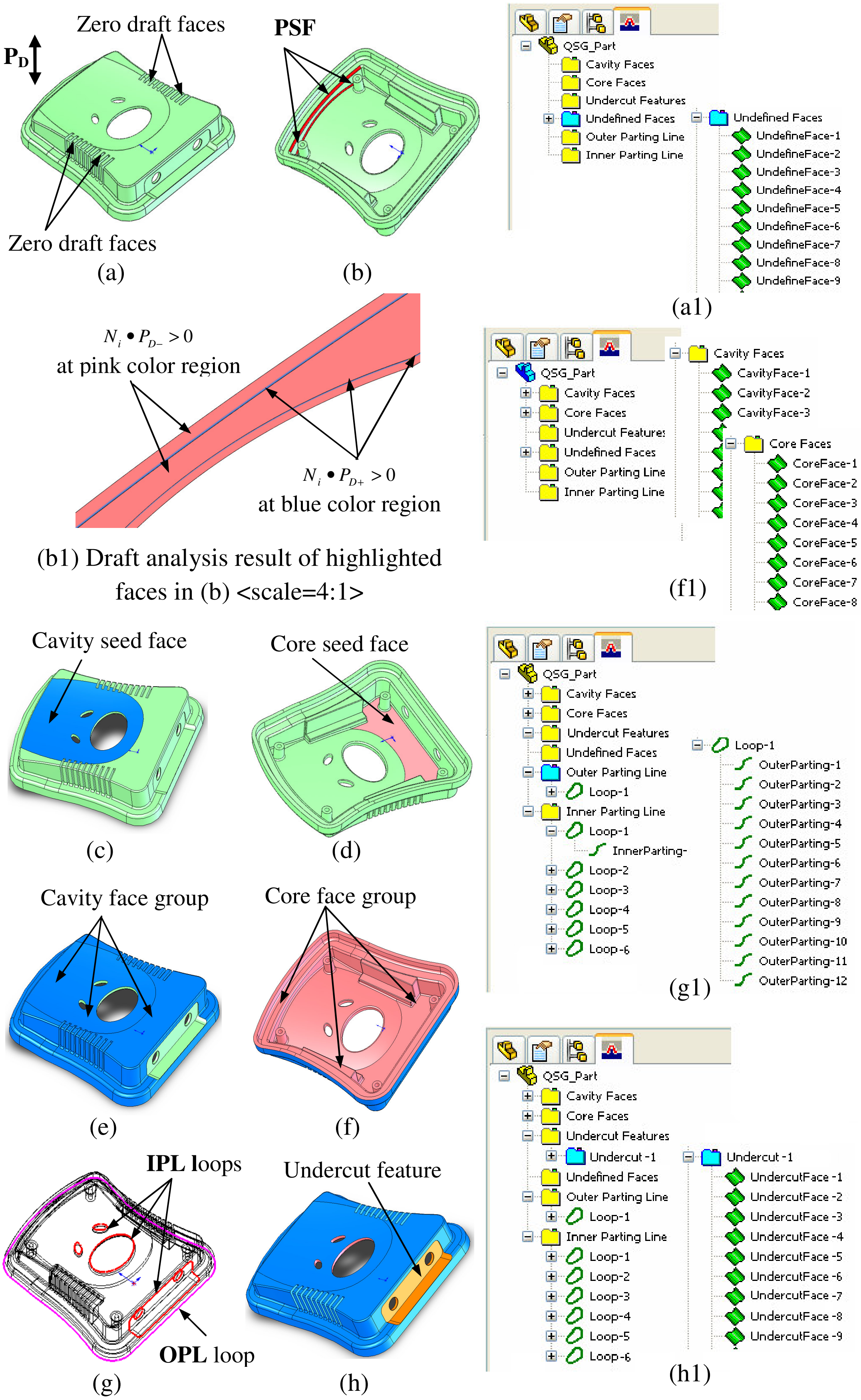
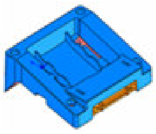
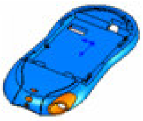
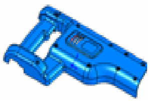


Fig.3.9. Case study 2 for the **FTMR** parting approach and the **ECFS**

3.9. Performance results

Tab.3.4 shows the execution time data for the above two cases and a few additional parts tested. As for part models which have fewer than 500 faces, the whole parting process took less than 11 seconds. As for a complex part model with 2533 faces, the approach took 48 seconds to identify all the parting entities. Regarding a part with complex geometry structure (10 undercut features); the parting process can complete within 18 seconds. The results indicate that the **FTMR** approach is quite effective for handling simple and complex parts. However, the data shown in Tab.3.4 does not include the execution time for any interactive function of the **ECFS**. Compared to the running time taken by the entire parting process, each individual interactive function can run even faster.

Parts						
Number of Faces	125	374	437	591	872	2533
Number of Parting lines	29	79	29	188	113	438
Number of undercut	1	1	1	10	0	1
Execution Time (seconds)	2	11	8	18	10	48

Tab.3.4. Execution time of the **FTMR** parting approach (Intel(R) Pentium (R) M, 1600 MHz, 768 MX of RAM, Windows XP; SolidWorks 2008; Models from Manusoft Technologies Pte Ltd and SolidWorks Inc.)

3.10. Summary

This chapter presented an automated parting approach based on face topology and mouldability reasoning (**FTMR**) to automatically identify cavity and core faces, inner and outer parting line loops, and extract undercut features for injection moulded products. An Error Correction and Feedback System (**ECFS**) has also been developed and incorporated within the **FTMR** approach. The **ECFS** provides the capability of visibly locating and correcting possible errors during the parting processes. Compared to the other parting methodologies, the presented approach has successfully overcome a few crucial bottlenecks for injection mould design applications as stated below:

1) This thesis is the first to put forward the concept and criterion of pseudo-straddle faces (**PSF**) in order to deal with geometric imperfections which commonly appear in industrial products.

2) The **FTMR** approach is robust for free-form faces and complex geometries because the approach enhances the capability of parting method based on the geometric topology and mouldability, and the algorithms are independent of the complexity of geometric feature and structure.

3) The **FTMR** approach is effective because all the parting entities, *i.e.* cavity/core faces, undercut features, inner and outer parting line loops, are identified in a single process and the presented iterative face growth algorithm does not need repetitive querying/cycling model entities (*e.g.* face, edge, vertex etc).

4) This thesis has presented a novel Error Correction and Feedback System (**ECFS**) and has successfully incorporated it into the **FTMR** approach. The **ECFS** can enhance the compatibility and capability of the parting approach for complex and various industrial applications.

It is noted that the functionalities of the **ECFS** are still in the initial stage of development and need to be improved so that the **ECFS** can be more powerful for more complex situations. Future work will explore ways to implement a knowledge-based environment for the **ECFS** to fulfill various design purposes and applications. In addition, as a pre-condition of the developed methodologies, this study assumes that all moulded products cannot be modified during parting processes. It did not discuss how to obtain a better parting solution by revising the design of original products. It should be helpful for an intelligent mould design system if parting methodologies can detect some unreasonable design and give corresponding suggestions for possible modification from the view of mould design. A knowledge-based engine can be implemented within the **ECFS** to achieve this idea.

CHAPTER 4

AUTOMATIC GENERATION OF PARTING SURFACES

Parting surfaces (**PS**), which are the mating surfaces of the core and cavity outside the moulded part in a moulding, are necessary to fully split a mould into the cavity and core sides. In the previous chapter, the outer parting line (**OPL**) loop has been identified. For a given parting direction, **PS** can be generated from the **OPL** loop based on its geometric characteristics and the required moulding size. Two factors must be taken into consideration while generating **PS** from a complex **OPL** loop, *i.e.* surface geometry and practical machining property of injection moulds. Radiating surfaces by offsetting parting lines [Tan1990] [Ravi Kumar2003] and sweeping surfaces [Fu2001] along parting lines are the two typical approaches to generate **PS**. They have been investigated and reported in some literature. In this thesis, an approach is developed to automatically generate parting surfaces using ruled and loft NURBS surfaces from an **OPL** loop.

4.1 Procedure of generating parting surfaces

Fig.4.1 shows the procedure of generating parting surfaces from an **OPL** loop. The input is all the edges of the **OPL** loop identified in the previous chapter. Firstly, the four corner vertices of the **OPL** loop are identified so that all edges will be grouped into four groups according to their positions with respect to the defined corners. Subsequently, the extruding directions for each group are determined based on their relevant positions, *i.e.* X_-, X_+, Y_- and Y_+ respectively. In the third step, a set of ruled NURBS surfaces $\mathbf{PS_R}$ is generated for all the valid edges with their corresponding extruding directions. The invalid edges are those which form a concave portion or overlap towards their assigned extruding directions and therefore are invalid for generating ruled surfaces $\mathbf{PS_R}$. In the last step, the four corner surfaces $\mathbf{PS_C}$ are created using loft surfaces and skinned surfaces $\mathbf{PS_A}$ are created for those invalid side edges for $\mathbf{PS_R}$ remained from the third step. As a result, the overall parting surfaces are generated from all edges of the **OPL** loop and comprise of all $\mathbf{PS_R}$, $\mathbf{PS_C}$ and $\mathbf{PS_A}$.

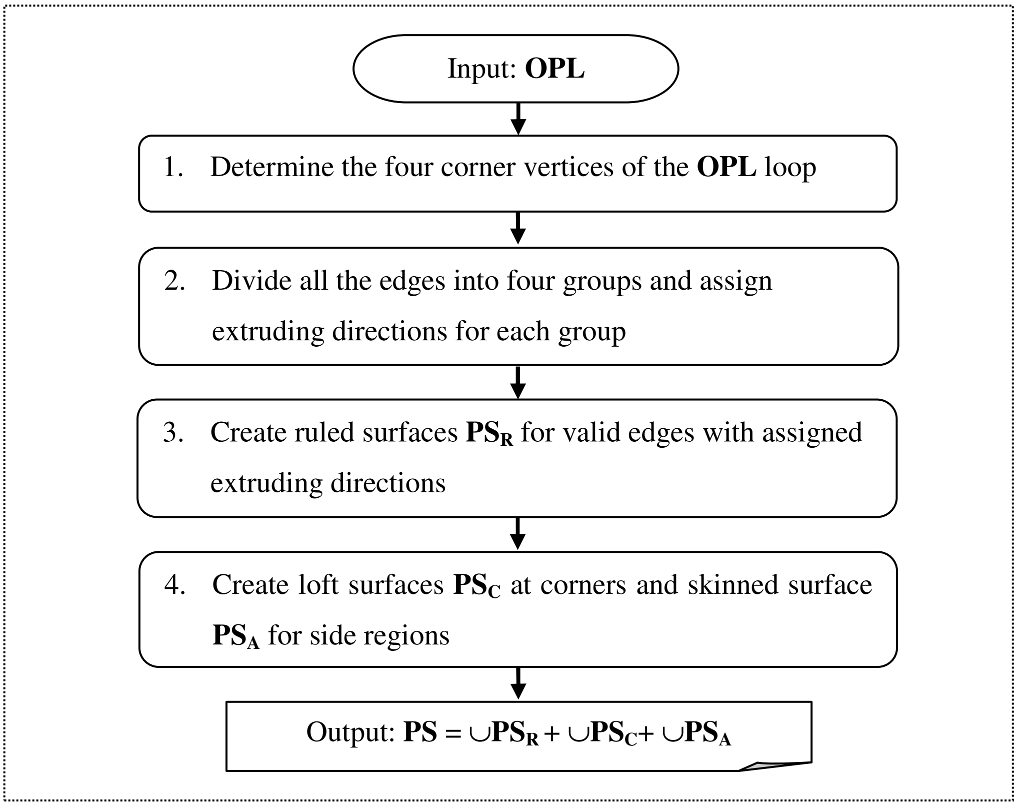

Input: **OPL**
1. Determine the four corner vertices of the **OPL** loop
2. Divide all the edges into four groups and assign extruding directions for each group
3. Create ruled surfaces $\mathbf{PS_R}$ for valid edges with assigned extruding directions
4. Create loft surfaces $\mathbf{PS_C}$ at corners and skinned surface $\mathbf{PS_A}$ for side regions
Output: $\mathbf{PS} = \cup\mathbf{PS_R} + \cup\mathbf{PS_C} + \cup\mathbf{PS_A}$

Fig.4.1. Procedure of generating parting surfaces

4.2 Generation of parting surfaces

4.2.1 Determination of the four corners of the OPL loop

In the thesis, $\mathbf{V_{NW}}$, $\mathbf{V_{NE}}$, $\mathbf{V_{SW}}$ and $\mathbf{V_{SE}}$ represent the northwestern, northeastern, southwestern and southeastern corners of an **OPL** loop respectively. To determine the four corner vertices, the **OPL** loop is first projected onto a 2D plane perpendicular to the given parting direction $\mathbf{P_D}$ as illustrated in Fig.4.2 (b). Then, the four corner vertices are determined by Con. 4-1, 4-2, 4-3 and 4-4 respectively [Fu1998-1]. If a vertex meets Con. 4-1, the vertex will be located at the extreme north-eastern region in the projected contour. It is the extreme north-eastern vertex and is denoted by $\mathbf{V_{NE}}$. If the vertex meets Con. 4-2, it is the extreme south-western vertex and can be denoted by $\mathbf{V_{SW}}$. Similarly, $\mathbf{V_{SE}}$ is the vertex which meets Con. 4-3 and $\mathbf{V_{NW}}$ is the vertex which meets Con. 4-4.

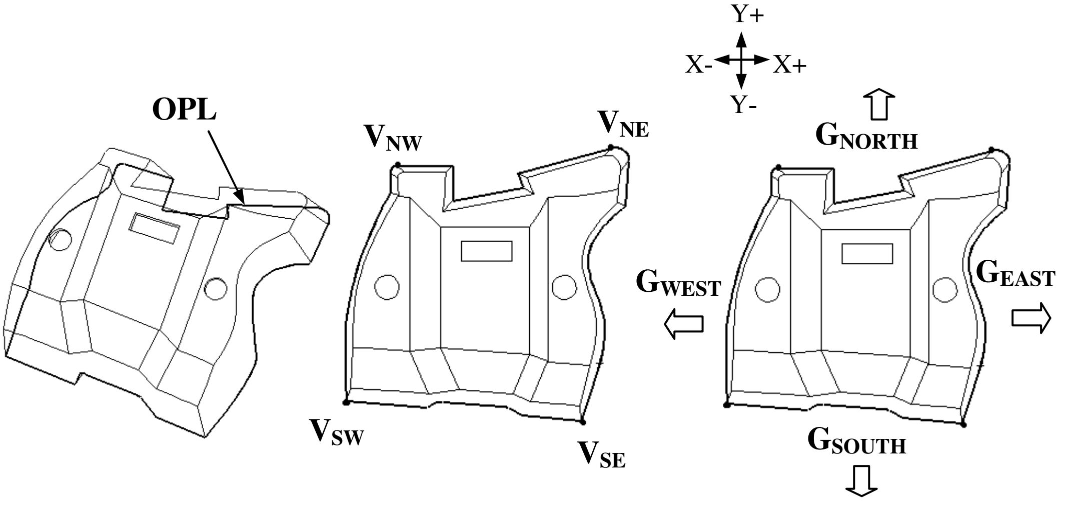
$$\mathrm{Max}(X_i+Y_i) \quad (i=1, 2, \ldots\ldots \text{numVertices}) \qquad \text{(Con. 4-1)}$$

$$\mathrm{Min}(X_i+Y_i) \quad (i=1, 2, \ldots\ldots \text{numVertices}) \qquad \text{(Con. 4-2)}$$

$$\mathrm{Max}(X_i-Y_i) \quad (i=1, 2, \ldots\ldots \text{numVertices}) \qquad \text{(Con. 4-3)}$$

$$\mathrm{Min}(-X_i+Y_i) \quad (i=1, 2, \ldots\ldots \text{numVertices}) \qquad \text{(Con. 4-4)}$$

Where numVertices represents the total number of vertices of the **OPL** loop.



(a) The **OPL** loop (b) Four corner vertices (c) Four edge groups and assigned extruding directions

Fig.4.2. Determination of the four corner vertices and extruding directions for the corresponding edge groups

4.2.2 Divide all edges into four groups and assign extruding directions for each group

Based on the four corner vertices identified in 4.2.1, all edges of the **OPL** loop are therefore divided into the four groups, named as $\mathbf{G_{EAST}}$, $\mathbf{G_{SOUTH}}$, $\mathbf{G_{WEST}}$ and $\mathbf{G_{NORTH}}$ as shown in Fig.4.2 (c). The extruding directions assigned for each group are listed in Tab.4.1, which are perpendicular to the given parting direction $\mathbf{P_D}$.

Edge groups	Extruding directions
$\mathbf{G_{EAST}}$	X+ direction
$\mathbf{G_{SOUTH}}$	Y- direction
$\mathbf{G_{WEST}}$	X- direction
$\mathbf{G_{NORTH}}$	Y+ direction

Tab.4.1 Extruding directions assigned for each edge group

4.2.3 Create ruled surfaces PS_R for edges with assigned extruding directions

Based on the results of the edge groups and assigned extruding directions, the ruled parting surfaces $\mathbf{PS_R}$ for all the edges are generated in the steps below:

Step 1:

Compute the boundary box of the part body so as to set up four reference planes RP_{X-}, RP_{X+}, RP_{Y-}, and RP_{Y+} corresponding to X-, X+, Y- and Y+ directions respectively in terms of the given boundary gap as shown in Fig.4.3 (a). The four reference planes are used to determine the end of ruled surfaces. The given boundary gap is determined by the moulding inserts.

Step 2:

Find the edges which are not able to generate valid ruled surfaces towards the assigned extruding directions using the algorithm below. As illustrated in Fig.4.3

(b), valid ruled surfaces cannot be generated for edges E_A and E_B along their extruding directions because the generated surfaces will self-intersect. The ruled surface for E_C along its extruding direction is a point, and therefore E_C will not be considered for the generation of parting surfaces. In addition, the ruled surfaces for E_D and E_E along their extruding directions are invalid because the generated surfaces will intersect with the mould part body and cannot be used to split the mould.

```
Algorithm to check the validation of an edge for the generation a ruled surface
{
    Input: an edge E, a reference plane RP with the extruding direction D.
    i.   Collect an ordered points list P[n_Pnt] from E given in the Parasolid library. n_Pnt represents the total number of points.
    ii.  Project P[0] and P[n_Pnt-1] onto RP and obtain the two new points $P_{Ref}[0]$ and $P_{Ref}[n\_Pnt-1]$. Then, $D_{Ref}$ represents the vector direction of the two points, i.e. $D_{Ref} = P_{Ref}[n\_Point-1] - P_{Ref}[0]$.
    iii. Compute the projected point $P_{Ref}[i]$ of P[i] onto RP, and check whether its vector direction ($P_{Ref}[i] - P_{Ref}[i-1]$) is identical with $D_{Ref}$. If it is not identical, there will be self-intersection or overlapping on the generated ruled surface. As a result, the edge E is invalid for creating a ruled surface.
}
```

Step 3:

The ruled surfaces for all the valid edges with the assigned extruding directions and the corresponding end reference plane are generated using the algorithm described below:

Algorithm to generate a ruled surface for an edge E with end reference plane RP (Illustrated in Fig.4.4 (a))

```
{
  i.   The edge E is first represented by a NURBS curve [Les1995]:
           $C(u) = \sum_{i=0}^{m-1} R_{i,p}(u) P_{i,0}$
       Where $P_{i,0}$ represents the control points of $C(u)$ which represents edge E. m
       represents the number of control points given in the Parasolid library.
  ii.  $P_{i,1}$ = Projection of $P_{i,0}$ onto plane RP (for i=0, 1, 2…m-1)
       Where $P_{i,0}$ represents the control points of $C(u)$
  iii. The ruled surface is represented as [Les1995]:
           $S(u,v) = \sum_{i=0}^{m-1} \sum_{j=0}^{1} R_{i,p,j,1}(u,v) P_{i,j}$
       Where p represents the degree of the surface.
}
```

Fig.4.3 (c) shows the result of ruled parting surfaces **PS$_R$**.

4.2.4 Create loft surfaces PS$_C$ at the corners and skinned surfaces PS$_A$ for the side regions

From 4.2.3, there are two types of regions which are still not completely covered, *i.e.* the four corners and the other concave side regions as seen in Fig.4.3 (c). As for those side regions, skinned surfaces **PS$_A$** are generated for each closed contour using SolidWorks API InsertFilledSurface [SolidWorks2008]. A closed contour comprises of the edges of concave side regions, two lines from adjacent ruled surfaces and a line connecting the two open ends of the adjacent ruled surfaces as shown in Fig.4.3 (d). With regard to the four corners, each corner region can be represented by either a point or an edge. If a corner is a point, then a planar surface can be generated defined by the two side lines. If the corner is an edge, a loft surface is generated using the edge and its two adjacent guide lines using the algorithm below.

<u>Algorithm</u> for generating a loft surface from an edge E and two guide lines L_1 and L_2 (Illustrated in Fig.4.4 (b))

{

i. First, the edge E is represented by the NURBS curve:

$$C(u) = \sum_{i=0}^{m-1} R_{i,p}(u) P_i$$

ii. Then, the control points $P_{i,1}$ are computed using below equation:

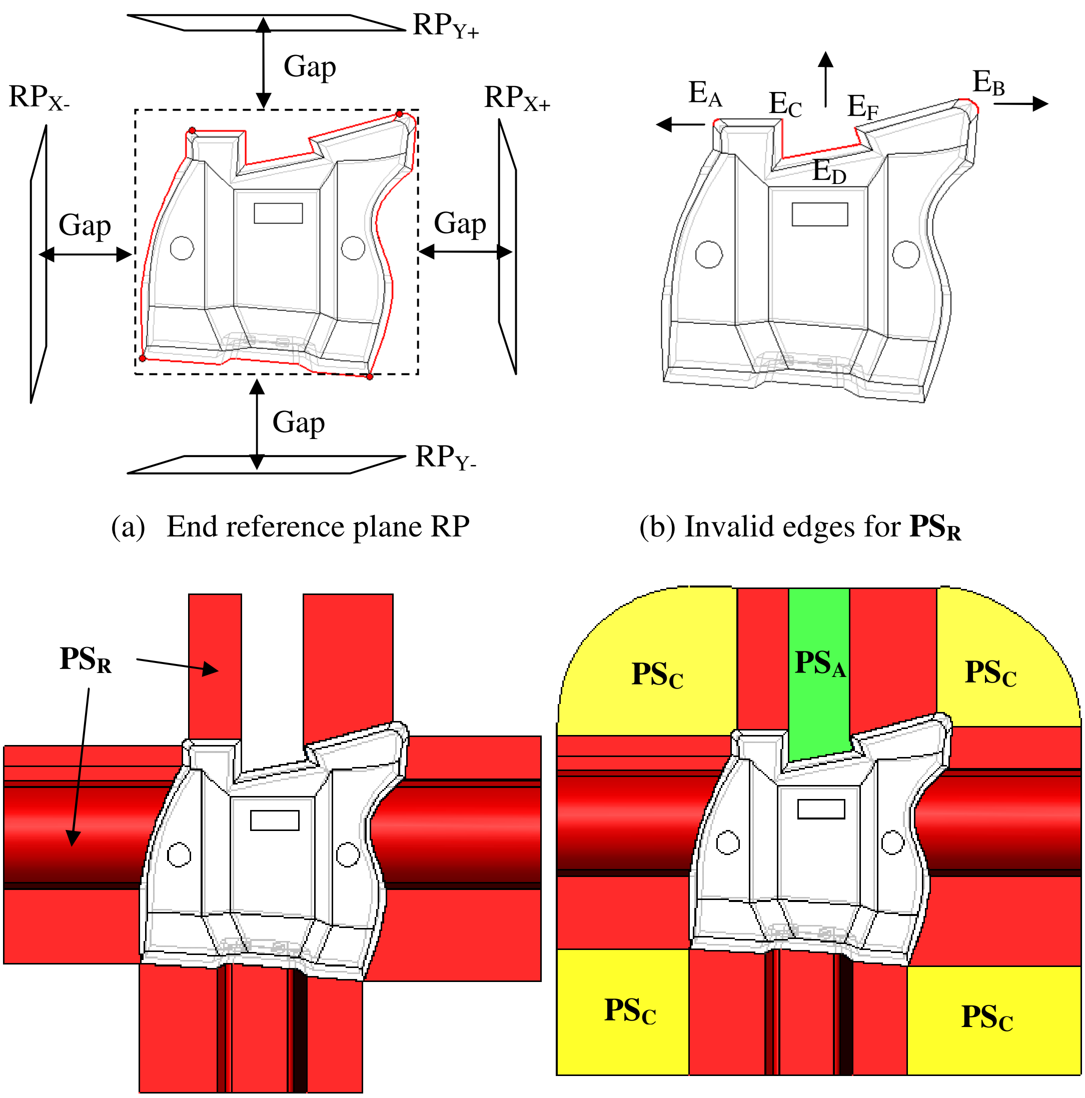
$$P_{i,1} = P_{i,0} + \overline{P_{0,0}P_{0,1}} + (\overline{P_{m-1,0}P_{m-1,1}} - \overline{P_{0,0}P_{0,1}}) \times (U_k - U_0) / (U_{m-1} - U_0) \text{ (i=0,1…m-1).}$$

Where $P_{i,0}$ represents the control points of $C(u)$. U represents the knot values of $C(u)$. The term $(U_k - U_0)/(U_{m-1} - U)$ gives the chord distribution based on the knot values. k is equal to (i+p-1) since the number of knots is equal to (m+p-1).

iii. As a result, the loft surface is represented as:

$$S(u,v) = \sum_{i=0}^{m-1} \sum_{j=0}^{1} R_{i,p,j,1}(u,v) P_{i,j}$$

}



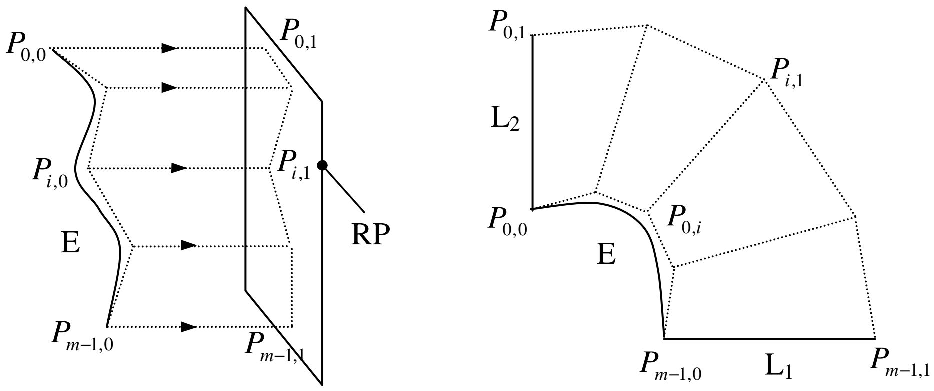
(a) End reference plane RP

(b) Invalid edges for **PS$_R$**

(c) Ruled surfaces **PS$_R$**

(d) Corner surfaces **PS$_C$** and **PS$_A$**

Fig.4.3. Illustration of the approach for generating parting surfaces



(a) Ruled surface

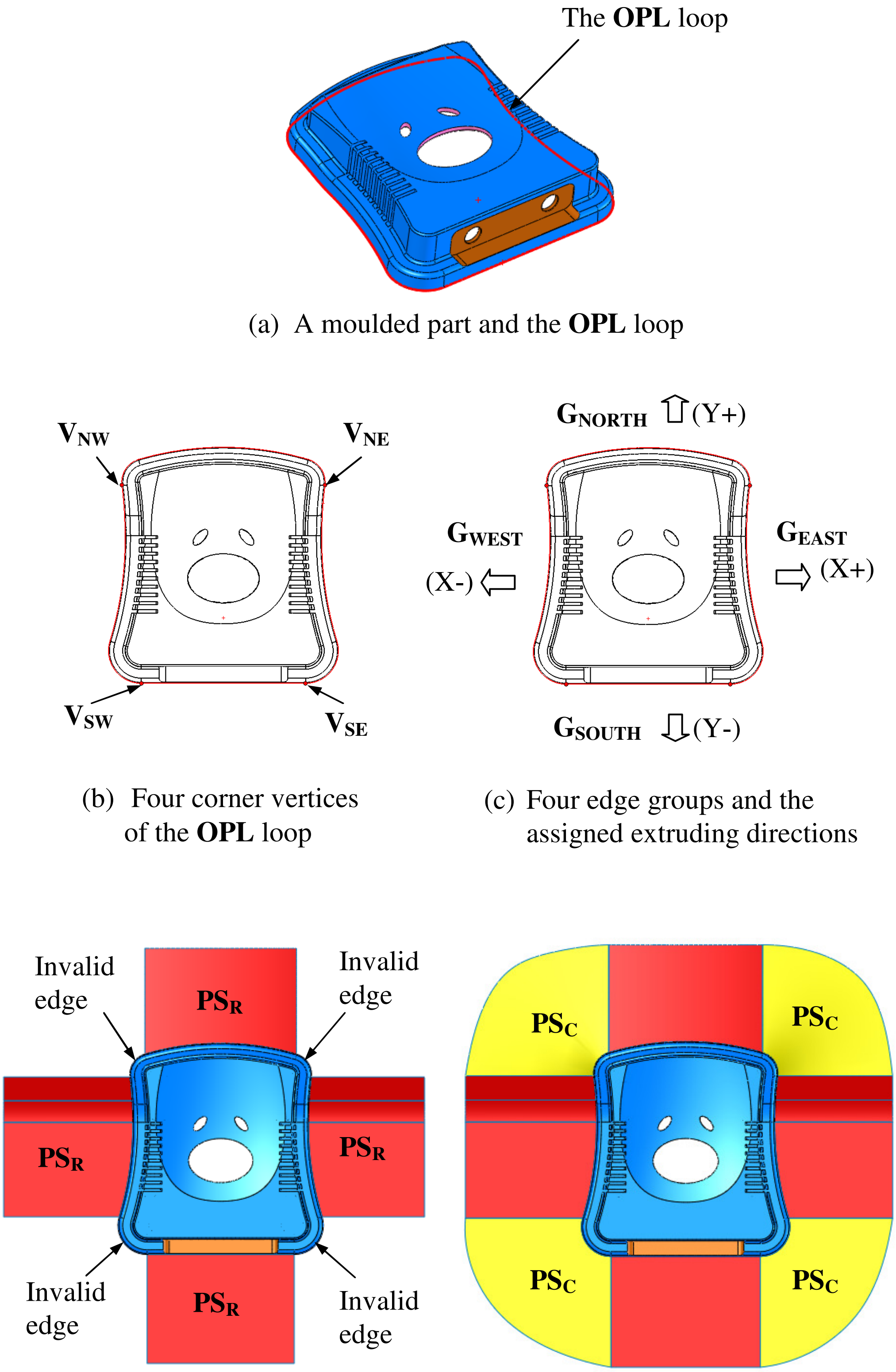
(b) Loft Surface

Fig.4.4. Illustration of the algorithms for generating parting surfaces

4.3 Case studies

4.3.1 Case study1

Fig.4.5 (a) shows a moulded part in which the **OPL** loop is highlighted and used to generate parting surfaces using the presented approach. The four corner vertices (*i.e.* $\mathbf{V}_{\mathbf{NW}}$, $\mathbf{V}_{\mathbf{NE}}$, $\mathbf{V}_{\mathbf{SW}}$ and $\mathbf{V}_{\mathbf{SE}}$) are first identified using the presented algorithm. The result can be seen in Fig.4.5 (b). Furthermore, all the edges of the **OPL** loop are classified into four groups (*i.e.* $\mathbf{G}_{\mathbf{EAST}}$, $\mathbf{G}_{\mathbf{SOUTH}}$, $\mathbf{G}_{\mathbf{WEST}}$ and $\mathbf{G}_{\mathbf{NORTH}}$) and extruding directions (*i.e.* X+, X-, Y+ and Y-) are assigned for each edge groups respectively as shown in Fig.4.5 (c). After that, ruled surfaces $\mathbf{PS}_{\mathbf{R}}$ are created from those valid edges along their extruding directions using the presented algorithm in Fig.4.5 (d). In this example, the ruled surfaces cannot be generated successfully in the four corner arcs since the ruled surfaces computed from them towards their extruding directions will self-intersect as indicated in Fig.4.5 (d). In the last step, the four corner arcs are covered with the loft NURBS surfaces $\mathbf{PS}_{\mathbf{C}}$ using the algorithms presented in 4.2.4. The final results of all parting surfaces are shown in Fig.4.5 (e).



(a) A moulded part and the **OPL** loop

(b) Four corner vertices of the **OPL** loop

(c) Four edge groups and the assigned extruding directions

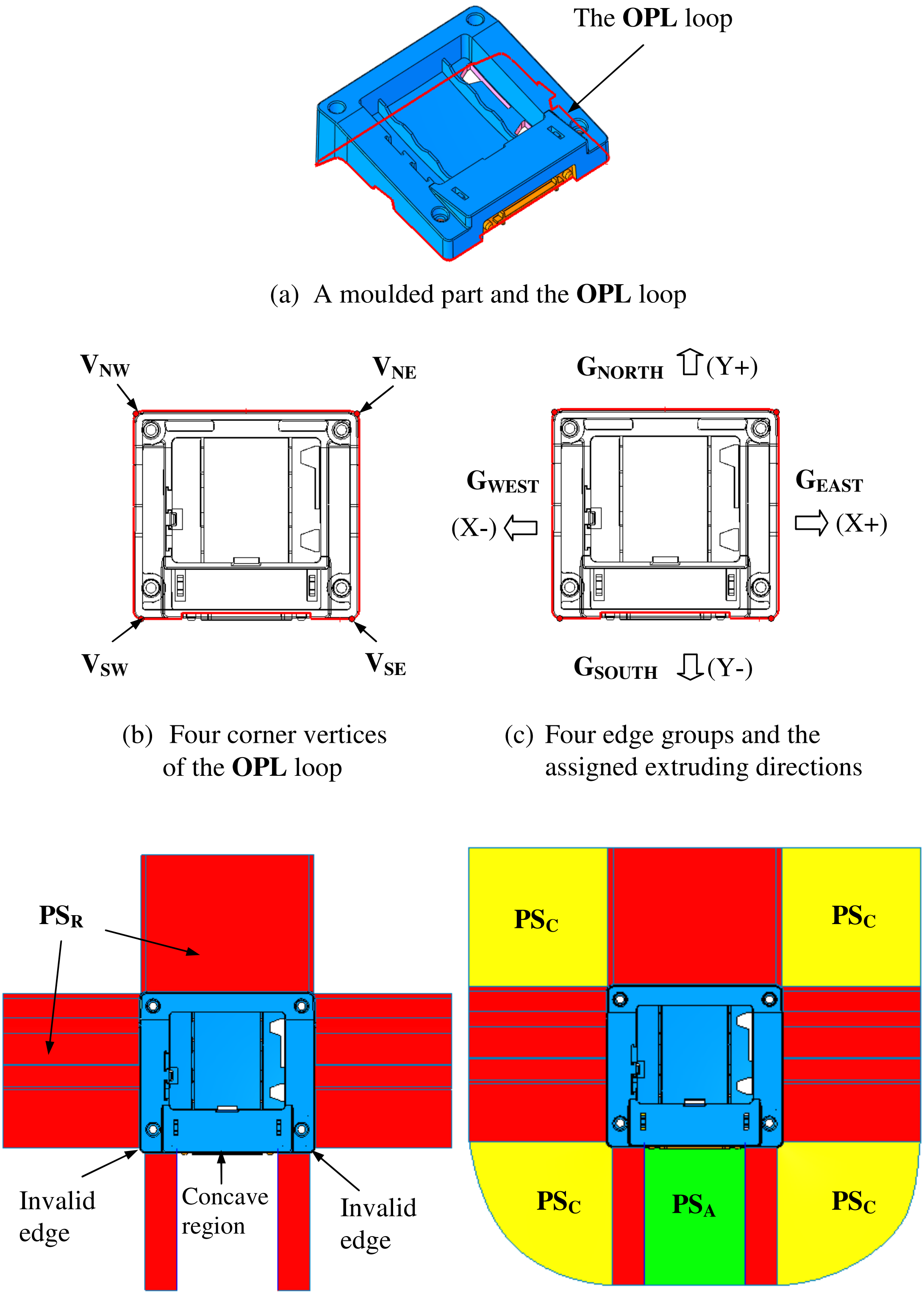
(d) Ruled surfaces **PS**$_{\mathbf{R}}$ and invalid edges for **PS**$_{\mathbf{R}}$

(e) Loft surfaces **PS**$_{\mathbf{C}}$ at four corner arcs

Fig.4.5. Case study 1 for creating parting surfaces

4.3.2 Case study 2

Fig.4.6 (a) shows another example part in which the highlighted **OPL** loop is used to generate parting surfaces using the presented approach. Firstly, the four corner vertices (*i.e.* $\mathbf{V_{NW}}$, $\mathbf{V_{NE}}$, $\mathbf{V_{SW}}$ and $\mathbf{V_{SE}}$) are identified as shown in Fig.4.6 (b). Subsequently, all the edges of the **OPL** loop are classified into four groups and assigned with extruding directions correspondingly in Fig.4.6 (c). Then, all valid ruled surfaces $\mathbf{PS_R}$ are generated from the edges along their extruding directions as shown in Fig.4.6 (d). In this example, there exists one concave side region as well as two corner edges where ruled surfaces cannot be generated successfully as indicated in Fig.4.6 (d). In the last step, one skinned surface $\mathbf{PS_A}$ is generated to patch the remained concave side region using SolidWorks API InsertFilledSurface. Two planar surfaces $\mathbf{PS_C}$ are then generated using two adjacent guide lines of the two upper corners which are represented by two points respectively. In addition, two loft NURBS surfaces are generated using the algorithm illustrated in Fig. 4.4 (b) for the other two corners which are represented by the two edges respectively. The final parting surfaces are therefore composed of all ruled surfaces $\mathbf{PS_R}$, one skinned surface $\mathbf{PS_A}$, two corner planner surfaces and two corner loft surfaces $\mathbf{PS_C}$ altogether as shown in Fig.4.6 (e).



(a) A moulded part and the **OPL** loop

(b) Four corner vertices of the **OPL** loop

(c) Four edge groups and the assigned extruding directions

(d) Ruled surfaces $\mathbf{PS_R}$ and invalid regions for $\mathbf{PS_R}$

(e) The skinned surface $\mathbf{PS_A}$ at side region and the four surfaces $\mathbf{PS_C}$

Fig.4.6 Case study 2 for creating parting surfaces

4.4 Summary

This chapter presented an approach for generating parting surfaces using ruled and loft NURBS surfaces from an **OPL** loop of a moulded product. Case studies show that the approach is efficient in automatic generation of parting surfaces from the complex parting line loops of moulded products. The advantages of the approach are listed below:

1) All the generated surfaces can be easily knitted together since all the adjacent surfaces share a linear edge defined by the same mathematical equation and parameters. This will avoid the failure of generating solid bodies for moulding inserts caused by the failure of knitting surfaces due to invalid tolerance.

2) The generated surfaces can be easily elongated or shortened since they are controlled by a single parameter towards their extruding directions. This makes the automated modification of the parting surfaces much more stable and simple while the sizes of moulding inserts have to be changed.

3) The generated surfaces give square shape boundaries rather than round ones of radiating surfaces, which are well suited for common square inserts.

4) Compared to radiating surfaces, ruled surfaces are easy to machine since the ruled surfaces are of linear property towards their extruding directions, *i.e.* X+, X-, Y+ and Y-. It implies that the tool paths for such a surface are parallel in one direction. Therefore, the tool path is accurate and easier for machining.

The presented approach in this chapter is well suited for the generation of the parting surfaces for the complex **OPL** loops of moulded products. As for the simple geometry of an **OPL** loop, the approach does not always generate the simplest parting surfaces. For instance, a single planar surface can be generated if all the edges of an **OPL** loop are located in a single plane. A radiating surface may give simpler results if an **OPL** loop is composed of a full circle or a combination of several arcs. These two situations can be treated as special cases and not covered in this thesis.

CHAPTER 5

AUTOMATIC GENERATION OF SHUT-OFF SURFACES

In order to fully split a moulded product with the core, cavity inserts and their local tools, all the inner parting line (**IPL**) loops have to be patched with shut-off surfaces (**SO**). A novel methodology of creating shut-off surfaces is introduced in this chapter. The methodology first obtains all the inner parting line loops using previous definitions. Then, the target **IPL** loops are identified by omitting those **IPL** loops which should not be patched due to the presence of undercut features. Finally, all target **IPL** loops are patched with shut-off surfaces (**SO**) using the presented approach. The approach classifies all **IPL** loops into four categories based on their geometric characteristics. Furthermore, the algorithms are developed for each category to generate the corresponding shut-off surfaces.

5.1 Search for the target IPL loops

In the previous **FTMR** parting methodology, all the **IPL** loops have been identified. However, not all the **IPL** loops should be patched in order to split a mould due to the presence of undercut features. First, those **IPL** loops enclosing blind undercut features should not be patched because the undercut features will be released along with them. Secondly, a through-hole undercut feature is enclosed by multiple **IPL** loops. Some of these **IPL** loops are connected with the cavity side, while the others are connected with the core side. Only one side of the **IPL** loops should be patched since the other **IPL** loops must be opened to release the side-core or side-cavity of the undercut feature correspondingly. Therefore, the target **IPL** loops for shut-off surfaces should satisfy the criteria below:

1) A target **IPL** loop should not be connected with a blind undercut feature. A blind undercut feature is connected with only one side of faces, either cavity side faces or core side faces.

2) As for those **IPL** loops which are connected with a through-hole undercut feature, the ones which are located towards the negative undercut's release direction are the target **IPL** loops to be patched since the other **IPL** loops stop the release of the undercut feature.

According to the criteria described above, the algorithm to identify the target **IPL** loops is described below.

Algorithm to identify the target **IPL** loops for shut-off surfaces:

```
{
  i.   Input: all the IPL loops;
  ii.  Search blind undercut features and remove associated IPL loops;
  iii. Search all through-hole undercut features and determine their release directions $D_{UF}$ [Mochizuki1992] [Fu1997] [Ye2001];
  iv.  Identify those IPL loops, which are connected with the through-hole undercut features;
  v.   Find those IPL loops $L_{OMIT}$ which are located towards the associated undercut's release directions;
  vi.  Remove $L_{OMIT}$ from all the IPL loops;
  vii. Output: the target IPL loops.
}
```

5.2 Methodology for creating shut-off surfaces

Fig.5.1 describes the methodology for creating shut-off surfaces (**SO**) from all the target **IPL** loops obtained in section 5.1. Based on their geometric characteristics, each **IPL** loop is classified into one of the four categories (*i.e.* Category 1, Category 2, Category 3 and Category 4), and the associated algorithms (*i.e.* NS_TY1, NS_TY2, NS_TY3 and NS_TY4) for the four categories are applied for generating the corresponding shut-off surfaces.

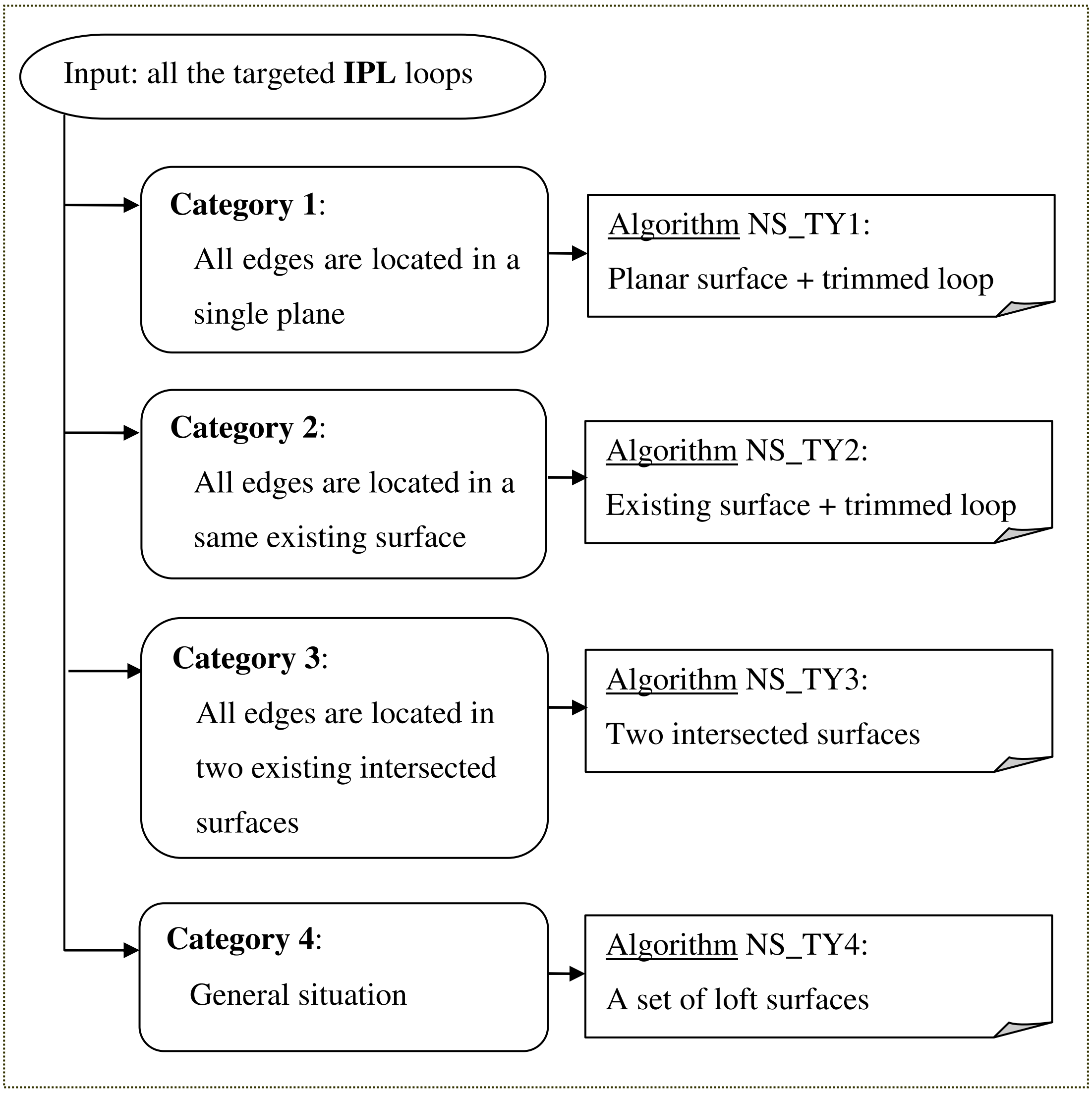

Input: all the targeted **IPL** loops
Category 1:
All edges are located in a single plane
Algorithm NS_TY1:
Planar surface + trimmed loop
Category 2:
All edges are located in a same existing surface
Algorithm NS_TY2:
Existing surface + trimmed loop
Category 3:
All edges are located in two existing intersected surfaces
Algorithm NS_TY3:
Two intersected surfaces
Category 4:
General situation
Algorithm NS_TY4:
A set of loft surfaces

Fig.5.1. Methodology for creating shut-off surfaces for the target **IPL** loops

5.2.1 Category 1 (NS_TY1)

When all the edges of an **IPL** loop are located on the same plane, the **IPL** loop is identified as Category 1. Correspondingly, an algorithm NS_TY1 is applied for generating shut-off surfaces for the **IPL** loop.

Algorithm NS_TY1: (see Fig.5.2)

{

NS_TY1 = $\mathrm{NS}_{\mathrm{PLANE}}$ (a boundary planar surface) + $\mathrm{C}_{\mathrm{TRIM}}$ (a trimmed curve loop) (see Appendix A for the definition)

$$\mathrm{NS}_{\mathrm{PLANE}} = \sum_{i=0}^{1}\sum_{j=0}^{1} R_{i,1,j,1}(u,v)P_{i,j}$$

Where $\mathrm{NS}_{\mathrm{PLANE}}$ is determined by four points $\mathbf{P}_{0,0}$ $\mathbf{P}_{0,1}$ $\mathbf{P}_{1,0}$ $\mathbf{P}_{1,1}$, which represent the boundary of the **IPL** loop at the planar surface. $\mathbf{P}_{0,0}$ $\mathbf{P}_{0,1}$ $\mathbf{P}_{1,0}$ $\mathbf{P}_{1,1}$ are the four control points of NURBS surface.

$\mathrm{C}_{\mathrm{TRIM}}$ = C[k]$\in$the **IPL** loop for k=1,2,…nEdge

$$\mathrm{C[k]} = \sum_{i=0}^{m-1} R_{i,p}(u)P_i$$

Where nEdge is the number of edges in the **IPL** loop. C[k] represents the kth trimmed curve.

}

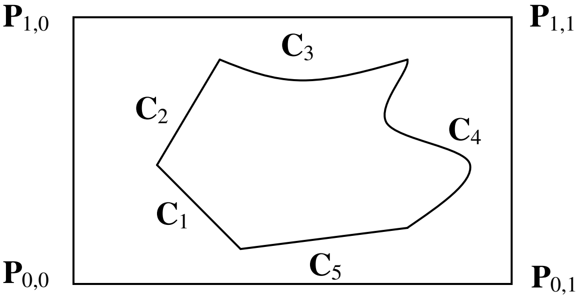


Fig.5.2. Description of control points and trimmed curves of NS_TY1

5.2.2 Category 2 (NS_TY2)

When all the edges of an **IPL** loop are located in an existing surface, the **IPL** loop is identified as Category 2. Correspondingly, an algorithm NS_TY2 is applied for generating shut-off surfaces for the **IPL** loop.

Algorithm NS_TY2:

{

i. First, a trimmed surface NS_TY2 is generated:

NS_TY2 = $\mathrm{NS_{EXIST}}$ (an existing surface) + $\mathrm{C_{TRIM}}$ (a trimmed curves loop)

Where $\mathrm{NS_{EXIST}}$ is the existing surface of the model that all edges belong to. $\mathrm{C_{TRIM}}$ is the same meaning as described in the algorithm NS_TY1.

ii. If all the edges of the **IPL** loop are shared by two existing surfaces $\mathrm{NS_{EXIST}}1$ and $\mathrm{NS_{EXIST}}2$, the criteria to determine the chosen $\mathrm{NS_{EXIST}}$ are:

- Priority of the single result of the generated NS_TY2. It implies that a $\mathrm{NS_{EXIST}}$ is chosen if the surface result NS_TY2 based on it does not generate any new boundaries.
- Minimum area priority. A $\mathrm{NS_{EXIST}}$ is chosen if the surface result NS_TY2 based on it gives the smaller surface area computed using Eq. 5-1.

$$Area = \sum_{i=0}^{m-1}\sum_{j=0}^{n-1} A_{Square}(u_i, v_j, u_{i+1}, v_{j+1}) \qquad (5\text{-}1)$$

Where A_{Square} expresses the small area of a square defined by two corner points of the NS_TY2 surface at (u_i, v_j) and (u_{i+1}, v_{j+1}). m and n are given in terms of the boundary size of the NS_TY2 surface.

Fig.5.3 (a) gives a sample of this case, in which the **IPL** loop is shared by two existing surfaces $NS_{EXIST}1$ and $NS_{EXIST}2$. According to the above criteria, $NS_{EXIST}1$ is chosen for the generation of NS_TY2 as shown in Fig.5.3 (b).

iii. Check whether the new NS_TY2 generates new open boundaries. If it does generate new edge boundaries, then these new edges are used to generate a skinned surface.

Fig.5.4 (a) shows a sample of this case. The NS_TY2 surface generates a new open boundary as shown in Fig.5.4 (b). Therefore, a skinned surface is needed to patch it in Fig.5.4 (c).

}

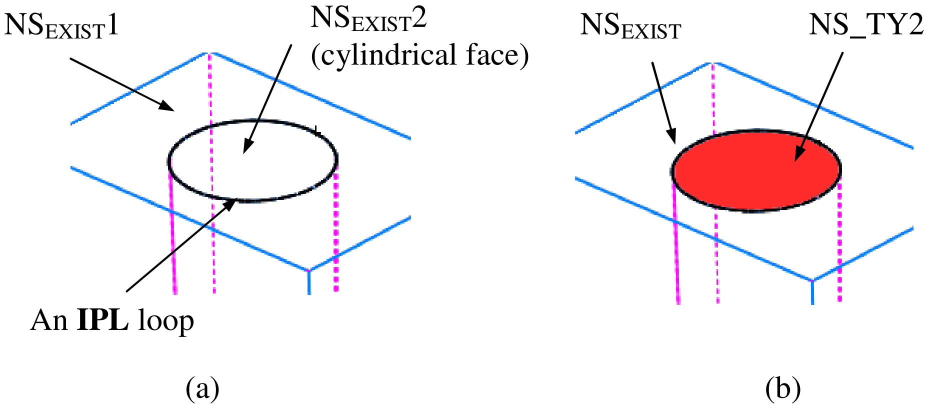


Fig.5.3. A sample in which an **IPL** loop is shared with two existing surfaces

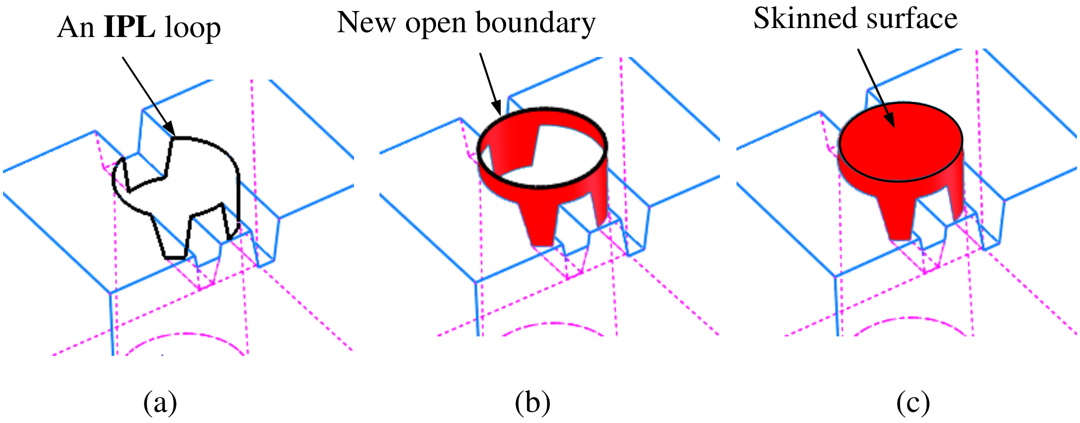


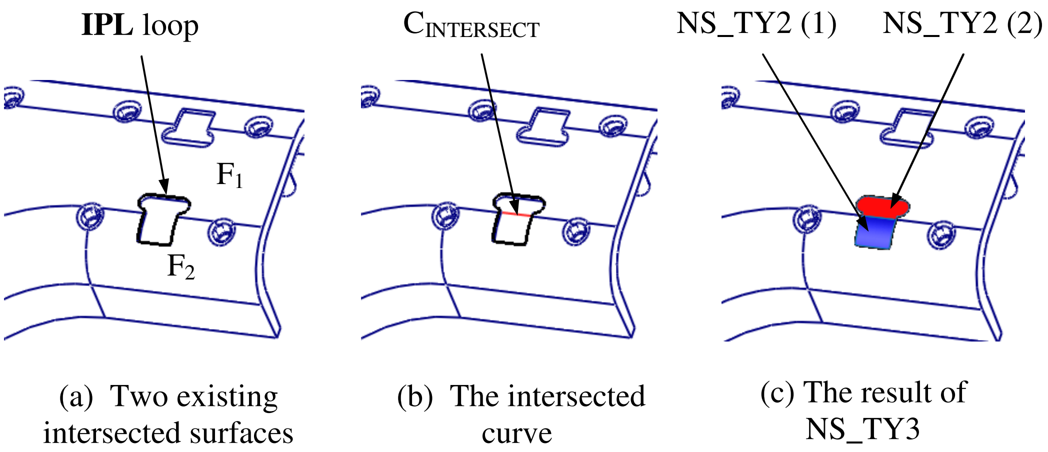
Fig.5.4. A sample in which NS_TY2 generates a new open boundary

5.2.3 Category 3 (NS_TY3)

If all the edges of an **IPL** loop are located in two existing intersected surfaces, the **IPL** loop is identified as Category 3. Correspondingly, an algorithm NS_TY3 is applied for patching the **IPL** loop.

```
Algorithm NS_TY3:
{
    NS_TY3 = $C_{INTERSECT}$ (the intersected curve) + NS_TY2(1) + NS_TY2(2)
        $C_{INTERSECT}$ = IntersectCurve ($F_{INTERSECT}1$, $F_{INTERSECT}2$)
    Where $F_{INTERSECT}1$ and $F_{INTERSECT}2$ represent the two existing surfaces which
    the IPL loop belongs to. $C_{INTERSECT}$ is the intersected curve between the two
    existing surfaces $F_{INTERSECT}1$ and $F_{INTERSECT}2$ (computed using SolidWorks
    API IntersectCurve) [SolidWorks2008].
        NS_TY2(1) = $F_{INTERSECT}1$+ $C_{TRIM}1$
        NS_TY2(2) = $F_{INTERSECT}2$+ $C_{TRIM}2$
    Two NS_TY2 surfaces are generated using $F_{INTERSECT}1$ and $F_{INTERSECT}2$ with
    their associated trimmed loops $C_{TRIM}1$ and $C_{TRIM}2$ respectively. $C_{TRIM}1$ and
    $C_{TRIM}2$ represent the two edge loops separated from the IPL loop by the
    intersected curve $C_{INTERSECT}$.
}
```

Fig.5.5 (a) illustrates a sample of Category 3. The **IPL** loop is located in two existing intersected surfaces F_1 and F_2. Fig.5.5 (b) shows the intersected curve $C_{INTERSECT}$ of F_1 and F_2. As a result, two NS_TY2 surfaces are generated to patch the entire **IPL** loop as shown in Fig.5.5 (c).



(a) Two existing intersected surfaces

(b) The intersected curve

(c) The result of NS_TY3

Fig.5.5. Illustration of the algorithm NS_TY3

5.2.4 Category 4 (NS_TY4)

If all the edges of an **IPL** loop cannot be classified into the above three categories, it will be considered as a general complex case (Category 4). A set of loft shut-off surfaces is generated using algorithm NS_TY4 which comprises of two steps, *i.e.* determining boundary constraints and generating loft shut-off surfaces based on defined boundary constraints.

Determining boundary constraints

From an **IPL** loop comprising of a closed ordered edge list (E_1, E_2 , … E_n) as shown in Fig.5.6, two sets of connected boundary faces can be obtained from the part model based on moulding necessity. One is on the cavity side, named pListCavityFace (F_1, F_2 ….F_n), and the other on the core side, named pListCoreFace (F_1’, F_2’ ….F_n’). In a part model, each edge is shared by two faces, one of which belongs to pListCavityFace and the other pListCoreFace. Each edge has two vertices and each vertex is also shared by two edges of an **IPL** loop.

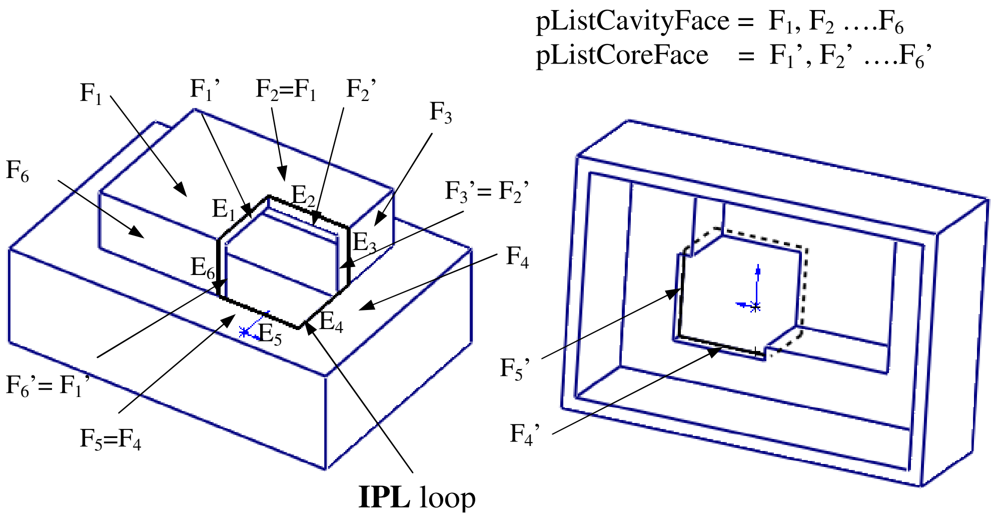


Fig.5.6. Cavity boundary faces and core boundary faces of an **IPL** loop

To create a loft surface for an edge E_1 shown in Fig.5.7, two guide paths have to be defined for its two end vertices V_1 and V_2 in two ways: (1) an adjacent edge as the guide path; (2) a guide direction D and corresponding end condition EC (end location along direction D). As for any vertex V_i of an edge E_i, the guide path could be defined in three ways: (1) an adjacent edge E_{i+1} or E_{i-1} as the guide edge; (2) A guide direction tangent to F_i and the corresponding EC; (3) A guide direction tangent to F_i' and the corresponding EC. Consequently, there exist three cases of boundary constraints for an edge E_1, *i.e.* (1) the guide paths at both V_1 and V_2 are determined by guide directions (D_1 and D_2) and the corresponding end conditions as shown in Fig.5.7 (a); (2) the guide path at one of the vertex V_1 is determined by the guide direction and the end condition, while the other one at V_2 is determined by the adjacent guide edge E_2 as shown in Fig.5.7 (b); (3) the guide paths at both V_1 and V_2 are determined by their adjacent guide edges E_0 and E_2 as shown in Fig.5.7 (c).

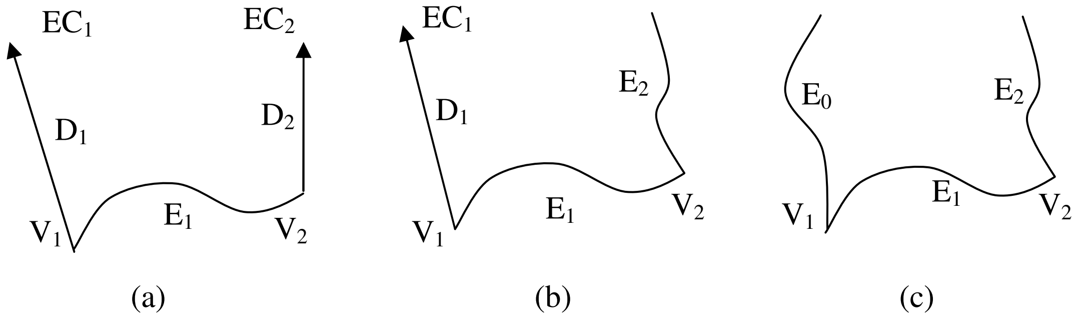


Fig.5.7. Three cases of guide path for a loft shut-off surface

Boundary constraints for shut-off surfaces should be determined based on mouldability reasoning and geometric topology of a moulded part. Based on the moulding requirements, the generation of shut-off surfaces should satisfy the below criteria and priorities in the following order:

i. Mouldability validity: The guide path or direction chosen for a shut-off surface should satisfy that the loft surface generated along it can be released towards the given parting direction. Fig.5.8 (a) and (b) show two invalid guide directions since the loft surfaces generated along them block the opening of the cavity insert. Fig.5.8 (c) illustrates another invalid guide direction in which the generated loft surface intersects with a core surface F_k'. In addition, a generated loft surface cannot be a straddle surface which is obviously invalid for the moulding requirement as shown in Fig.5.8 (d).

ii. Guide edge priority: If an adjacent edge can be a valid guide path, the edge is chosen as the guide path for generating the loft surface rather than other tangent directions. This is to ensure no gap among shut-off surfaces and adjacent faces of the moulded part and to minimize the number of loft shut-off surfaces for an **IPL** loop.

iii. Same guide direction priority: In two possible guide directions, the one which is the same as the previous one is chosen. This is to create simple geometry and to maintain good machining property of the loft surfaces.

iv. Same tangent boundary face priority: If a boundary face F_i corresponding to V_i is the same as the previously chosen boundary face F_{i-1}, F_i should be chosen to determine its guide direction prior to other adjacent surfaces. This is to give simple shut-off surfaces.

v. Same side of tangent boundary face priority: If two guide directions tangent to both F_i (from pListCavityFace) and F_i' (from pListCoreFace) are valid for a shut-off surface, the one which is on the same side as the previously chosen boundary face (pListCavityFace or pListCoreFace), should be chosen. This is to maintain good machining property for moulding inserts.

vi. Angle validity for guide directions: The included angle between the two guide directions assigned for the two vertices of an edge must be smaller than 60 degree in this thesis. This is to avoid twisted loft surface and maintain good manufacturing property. A bad sample surface is shown in Fig.5.8 (e) where the included angle between D_i and D_{i+1} is 90 degree.

vii. Same guide directions for the same vertex necessity: With regards to the same vertex V_i, which is shared by two edges E_i and E_{i+1}, the two given guide directions must be the same if the adjacent edge is not chosen as the guide path at V_i. Otherwise, the generated adjacent two shut-off surfaces will have a gap as illustrated in Fig.5.8 (f). In this research, the mean value of the two given guide directions will be applied as the guide direction at the specific vertex if it satisfies with other criteria.

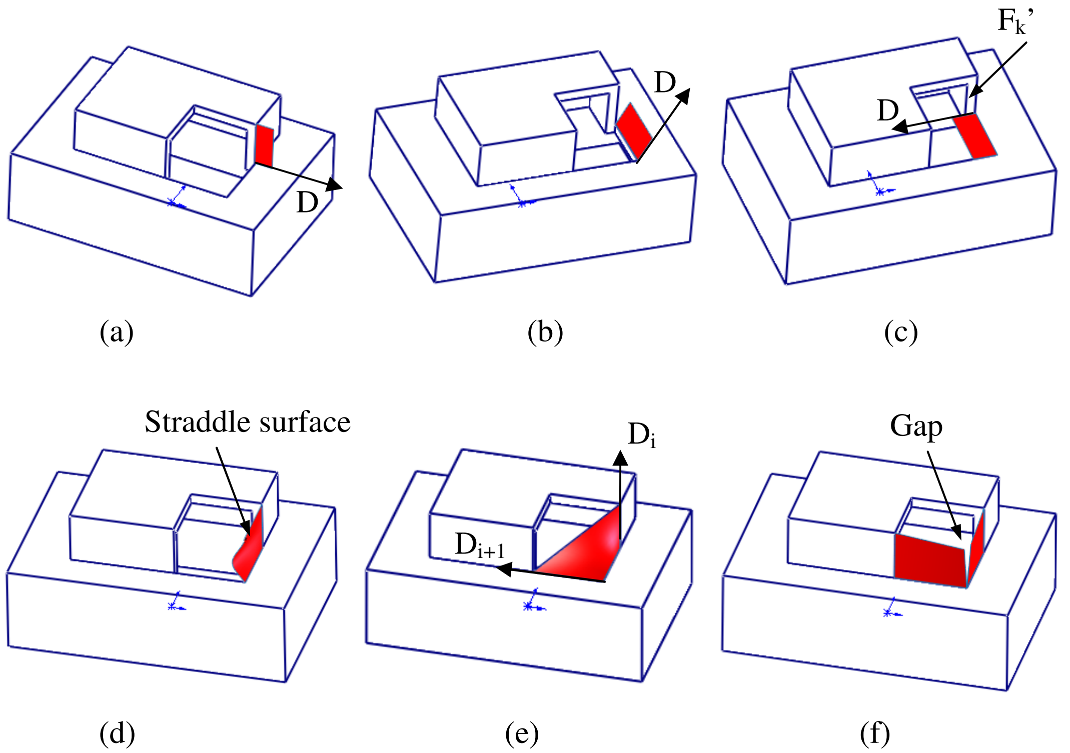


Fig.5.8. Invalid guide directions for shut-off surfaces based on mouldability reasoning and geometric characteristics

According to the previous investigation, there exist two possible guide directions, which are tangent to two adjacent boundary faces, and one adjacent guide edge to define the guide path for each vertex of an edge for the generation of a loft surface. However, it is found that some of guide directions or guide edges are not valid because the loft surfaces generated along these directions or path will block the opening of the cavity and core inserts, or are not suitable for the moulding process. There is a unique choice of the guide path for some vertices based on their mouldability reasoning and geometric characteristics. For instance in Fig.5.9 (a), the adjacent edge E_{i+1} is the only choice of the guide path at V_i of E_i since the guide direction tangent along F_i' is invalid and the other possible guide direction is the same as the one given by E_i (see criteria ii). In Fig.5.9 (b), the only chosen guide direction for V_i of E_i and E_{i+1} is the guide direction D since their guide edges and other guide directions are all invalid based on the moulding requirements. Consequently, all these special vertices of the **IPL** loop have to be found and their guide paths must be identified in advance in order to define the entire boundary constraints for the generation of shut-off surfaces. The algorithm is described below:

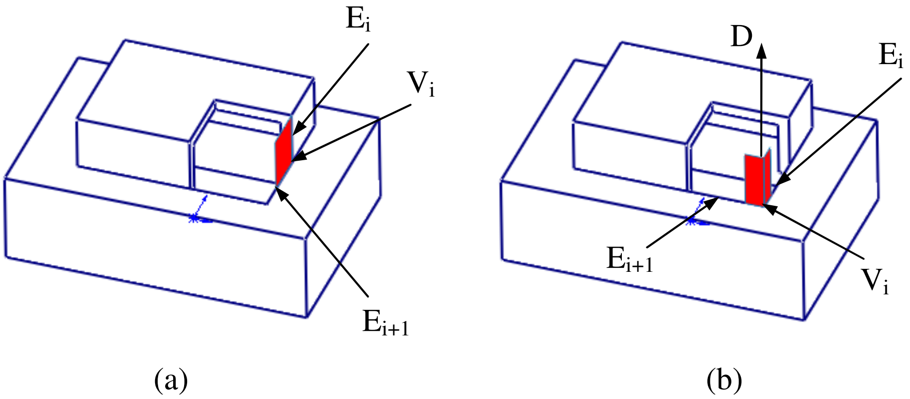


Fig.5.9. Samples in which the only guide path or direction should be chosen at the vertices based on mouldability requirements

Algorithm for searching special vertices and determining their boundary constraints:

{

i. Input: Edge list (E_1, E_2, …E_n) of an **IPL** loop, vertex list (V_1, V_2, …V_n), pListCavityFace (F_1, F_2 ….F_n) and pListCoreFace (F_1’, F_2’ ….F_n’); For a vertex V_i: $V_i \in E_i \cap E_{i+1}$; $E_i \in F_i \cap F_i$’; $E_{i+1} \in F_{i+1} \cap F_{i+1}$’; F_i is connected with or the same as F_{i+1}; F_i’is connected with or same as F_{i+1}’.

ii. Compute tangent directions D_{ii} and D_{ii}’for each vertex V_i of each edge E_i.

$$D_{ii} = \text{Tangent}\{V_i, F_i, E_i\} \text{ and } D_{ii}' = \text{Tangent}\{V_i, F_i', E_i\}$$

Where the function Tangent{V, F, E} = Normal[F, V]×Tangent[E, V].

iii. Evaluate these possible guide directions and guide paths (the adjacent edges) based on the previous moulding requirements and criteria.

iv. Identify all special vertices and assign their associated boundary constraints.

}

Fig.5.10 illustrates the algorithm to check the moulding validity of a guide direction D for a vertex V of an edge E. A ray R is first formed by a point P_t and the given parting direction $\mathbf{P_D}$:

$$R = \{P_{t,} \mathbf{P_D}\} \qquad (5\text{-}2)$$

$$P_t = V + d*D \qquad (5\text{-}3)$$

Where P_t represents a point extended from V along its guide direction D by a given small distance d (0.5mm by default).

As a result, a guide direction is invalid if there are mapping points of R with the part body S as illustrated in Fig.5.10 (b). The existence of mapping points implies that the vertex V cannot be extended along the guide direction D, and therefore the guide direction D is invalid for the generation of a shut-off surface.

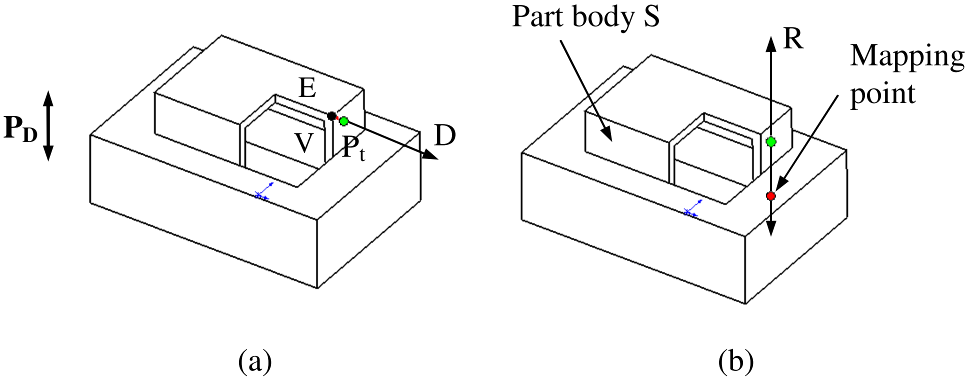


Fig.5.10. Illustration of checking the validity of a guide direction

The algorithm to verify whether a surface generated is a straddle surface has been described previously in section 3.2. The algorithm to check the mouldability of a loft surface is described below.

Algorithm to check the mouldability of a loft surface:

```
{
    i.   Input: the IPL loop and a loft surface F;
    ii.  Construct a 2D region $Rg_P$ by projection of the IPL loop onto an plane
         perpendicular to the given parting direction $\mathbf{P_D}$;
    iii. A regular triangle net T is created from the loft surface F (refer to Fig.3.1);
    iv.  Verify whether all projection points of the regular triangle net are located
         inside the 2D region;
    v.   If there are projection points outside the 2D region, the surface is invalid for
         the mouldability requirement.
}
```

All the entities of the **IPL** loop can be illustrated as an entity loop (see Fig.5.11). E_1 represents the first edge. F_1 and F_1' are the two faces shared by E_1. D_{11} indicates the guide direction of vertex V_1 of E_1 tangent along F_1, while D_{11}' indicates the guide direction of vertex V_1 of E_1 tangent along F_1'. D_{12} indicates the guide direction of vertex V_1 of E_1 tangent along F_2, while D_{12}' indicates the guide direction of vertex V_1 of E_1 tangent along F_2'. The rest may be deduced by analogy. Based on the criteria and priorities of loft shut-off surfaces and those pre-defined special boundary constraints, the overall boundary constraints are consequently determined by cycling all edges and their corresponding vertices of the **IPL** loop orderly. The start vertex will be the one which boundary constraints have been defined before or the first vertex based on index number (*i.e.* V_1 of E_1) if there is no any special vertex. As a result, all the vertices of the edges are assigned with their guide directions or guide edges.

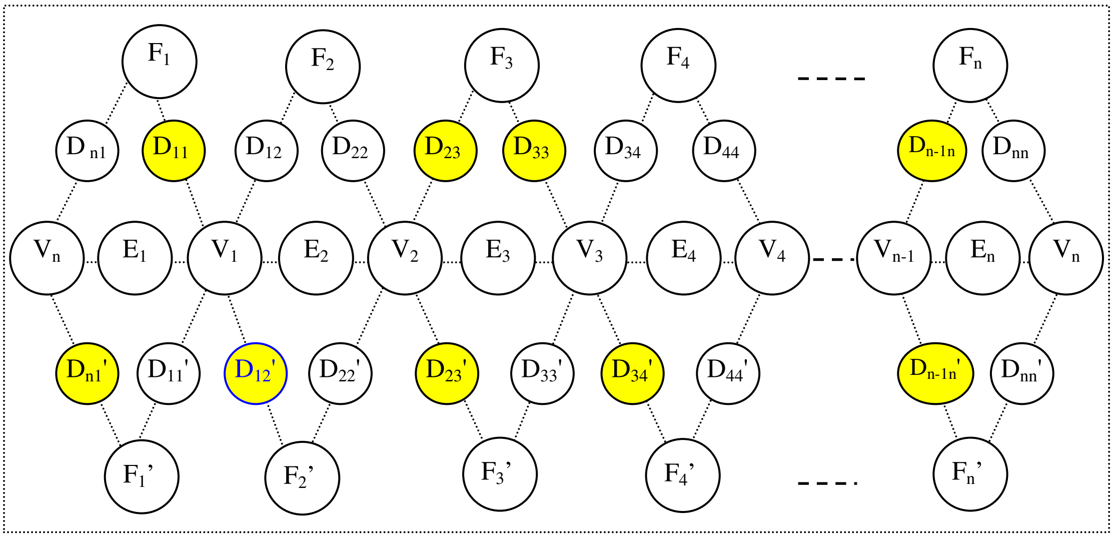


Fig.5.11. Determination of the overall guide paths for all the vertices

As indicated previously, an end condition EC has to be determined correspondingly if the boundary constraint of a vertex is assigned with a guide direction rather than a guide edge. There are two steps to determine the end condition corresponding to a guide direction as described below.

Step 1:

A container box of the **IPL** loop is computed in terms of the given parting direction. The container box is defined by six parameters X_{MIN}, Y_{MIN}, Z_{MIN}, X_{MAX}, Y_{MAX} and Z_{MAX}, each of which can be considered as a reference boundary plane. As shown in Fig.5.12 (a), X_{MIN}, Y_{MIN}, X_{MAX} and Y_{MAX} are tight boundary of the **IPL** loop onto a plane perpendicular to the parting direction $\mathbf{P_D}$. Z_{MIN} and Z_{MAX} are defined with a gap along the given parting direction. This implies that the generated shut-off surfaces can be extended along $\mathbf{P_D}$, but cannot be exceeded the boundary in other directions based on the mouldability requirement.

Step 2:

The end condition EC corresponding to a guide direction at a vertex V_i is then determined by the projected point of V_i onto the closest boundary plane along the guide direction. As illustrated in Fig.5.12 (b), EC is the intersected point of the guide direction and the Z_{MAX} plane.

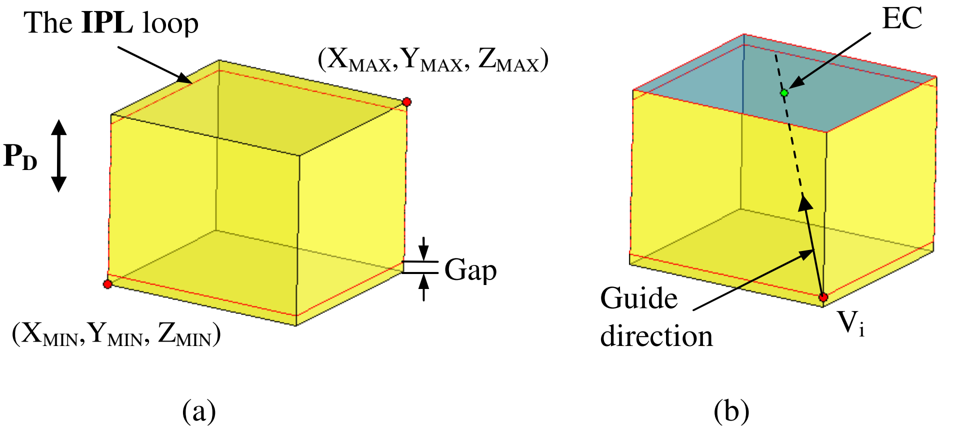


Fig.5.12. Determination of the end condition EC corresponding to a guide direction at a vertex

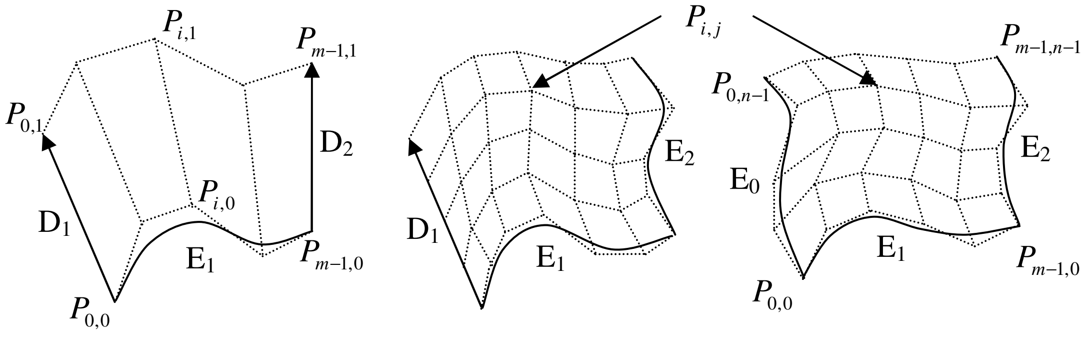
Generating loft shut-off surfaces based on boundary constraints

Based on the boundary constraints defined previously, there are three cases for a single loft shut-off surface:

1) A loft surface generated from an edge with two guide directions and their end condition in Fig.5.13 (a);

2) A loft surface generated from an edge with a guide direction, end condition and a guide edge in Fig.5.13 (b);

3) A loft surface generated from an edge with two guide edges in Fig.5.13 (c).



(a) (b) (c)

Fig.5.13. Three cases of boundary constraints for loft shut-off surfaces

<u>Algorithm</u> for generating a loft surface in Fig.5.13 (a):

{

i. First, E1 is represented by the NURBS curve [Les1995]:

$$C(u) = \sum_{i=0}^{m-1} R_{i,p}(u) P_i$$

ii. Then, the control points $P_{i,1}$ are computed using equation:

$$P_{i,1} = P_{i,0} + \overline{P_{0,0}P_{0,1}} + (\overline{P_{m-1,0}P_{m-1,1}} - \overline{P_{0,0}P_{0,1}}) \times (U_k - U_0)/(U_{m-1} - U_0) \text{ (i=0,1...m-1).}$$

Where m represents the number of control points of $C(u)$. $P_{i,0}$(i=0,1...m-1) represents the control points of $C(u)$. U represents the knot values of $C(u)$. k is equal to (i+p-1) since the number of knots is equal to (m+p-1). The term $(U_k - U_0)/(U_{m-1} - U)$ gives the chord distribution based on knot values.

iii. As a result, the loft shut-off surface is represented as:

$$S(u,v) = \sum_{i=0}^{m-1} \sum_{j=0}^{1} R_{i,p,j,1}(u,v) P_{i,j}$$

}

<u>Algorithm</u> for generating a loft surface in Fig.5.13 (b) and (c):

{

i. E1 is represented by the NURBS curve $C_1(u)$: $C_1(u) = \sum_{i=0}^{m-1} R_{i,pk}(u) P_i^k$

E0 is represented by the NURBS curve $C_0(u)$: $C_0(u) = \sum_{i=0}^{l-1} R_{i,pk}(u) P_i^k$

E2 is represented by the NURBS curve $C_2(u)$: $C_2(u) = \sum_{i=0}^{l-1} R_{i,pk}(u) P_i^k$

ii. The curve degree and knot values of $C_0(u)$ and $C_2(u)$ are uniformed [Les1995]. The new $C_0'(u)$ and $C_2'(u)$ are obtained. Then all control points $P_{i,j}$ for the new loft surface are computed using equation:

$$P_{i,j} = P_{i,j-1} + \overline{P_{i,j-1}P_{i,j}} + (\overline{P_{m-1,j-1}P_{m-1,j}} - \overline{P_{i,j-1}P_{i,j}}) \times (U_k - U_0)/(U_{m-1} - U_0)$$

(i=0,1…n-1; j=0,1…m-1)

Where m represents the number of control points of $C_1(u)$. n represents the number of control points of $C_0'(u)$ and $C_2'(u)$. U represents knot values of $C_1(u)$. $P_{0,j}$ and $P_{m-1,j}$ (j=0,1, … n-1) represent the control points of $C_0'(u)$ and $C_2'(u)$ respectively. k is equal to (i+p-1) since the number of knots is equal to (m+p-1). The term $(U_k - U_0)/(U_{m-1} - U)$ gives the chord distribution based on knot values of $C_1(u)$.

iii. As a result, the loft shut-off surface is represented by equation:

$$S(u,v) = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} R_{i,p,j,q}(u,v) P_{i,j}$$

}

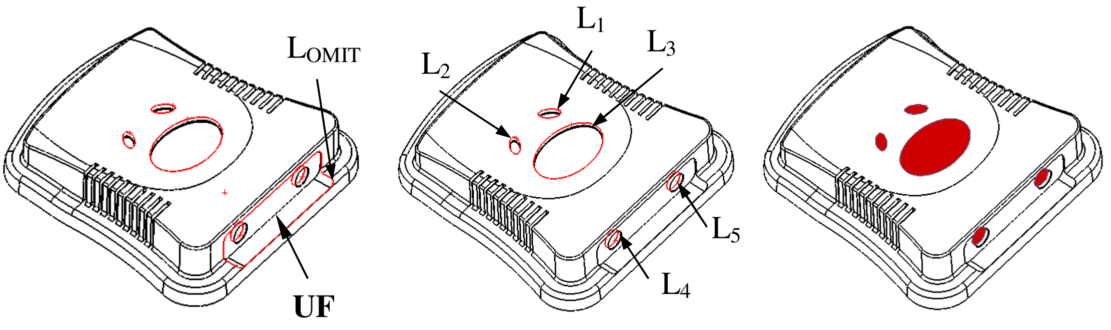
After all the single loft surfaces have been generated; the final shut-off surfaces are subsequently obtained by trimming all the intersected surfaces with each another. If a surface is trimmed into a few pieces, the one, which is connected with the **IPL** loop, should remain. In this study, SolidWorks APIs (GetIntersectSurafceCount and IIntersectSurface) are used to trim shut-off surfaces [SolidWorks2008].

5.3 Case studies

5.3.1 Case study 1

Fig.5.14 shows an example of generating shut-off surfaces using the presented algorithms. All the **IPL** loops are first obtained from the previous definitions. Based on the approach, the target **IPL** loops are then identified by removing L_{OMIT} since it stops the release of undercut feature **UF** illustrated in Fig.5.14 (a). After that, all the target **IPL** loops are patched based on their geometric categories and the corresponding algorithms. As shown in Fig.5.14 (b) and (c), L_1 and L_2 are classified into Category 2 based on their geometry and the algorithm NS_TY2 is used to generate their shut-off surfaces. While, L_3, L_4 and L_5 are classified into Category 1 and the algorithm NS_TY1 is used to generate their shut-off surfaces.

Loop	L_1, L_2	L_3, L_4, L_5
Category	NS_TY2	NS_TY1



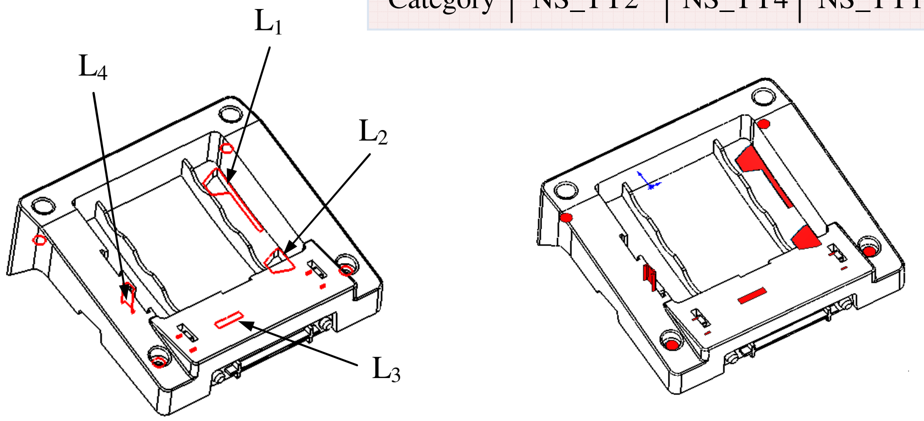
(a) all **IPL** loops (b) Targeted **IPL** loops (c) Shut-off surfaces

Fig.5.14. Case study 1 for creating shut-off surfaces

5.3.2 Case study 2

Fig.5.15 gives another example of generating shut-off surfaces. All the **IPL** loops are highlighted in Fig.5.15 (a). In this case, there is no L_{OMIT} since all the **IPL** loops do not block the release of undercut features. Based on the presented approach, L_1, L_2 and L_3 are classified into Category 2 based on their geometric characteristics and the algorithm NS_TY2 is used to generate shut-off surfaces. L_4 is classified into Category 4 and the algorithm NS_TY4 is therefore applied for generating a set of loft surfaces for the loop. The other **IPL** loops are classified into Category 1 and the algorithm NS_TY1 is used to generate their shut-off surfaces. Fig.5.15 (b) shows the overall shut-offs for the moulded part.

Loop	L_1, L_2, L_3	L_4	Others
Category	NS_TY2	NS_TY4	NS_TY1



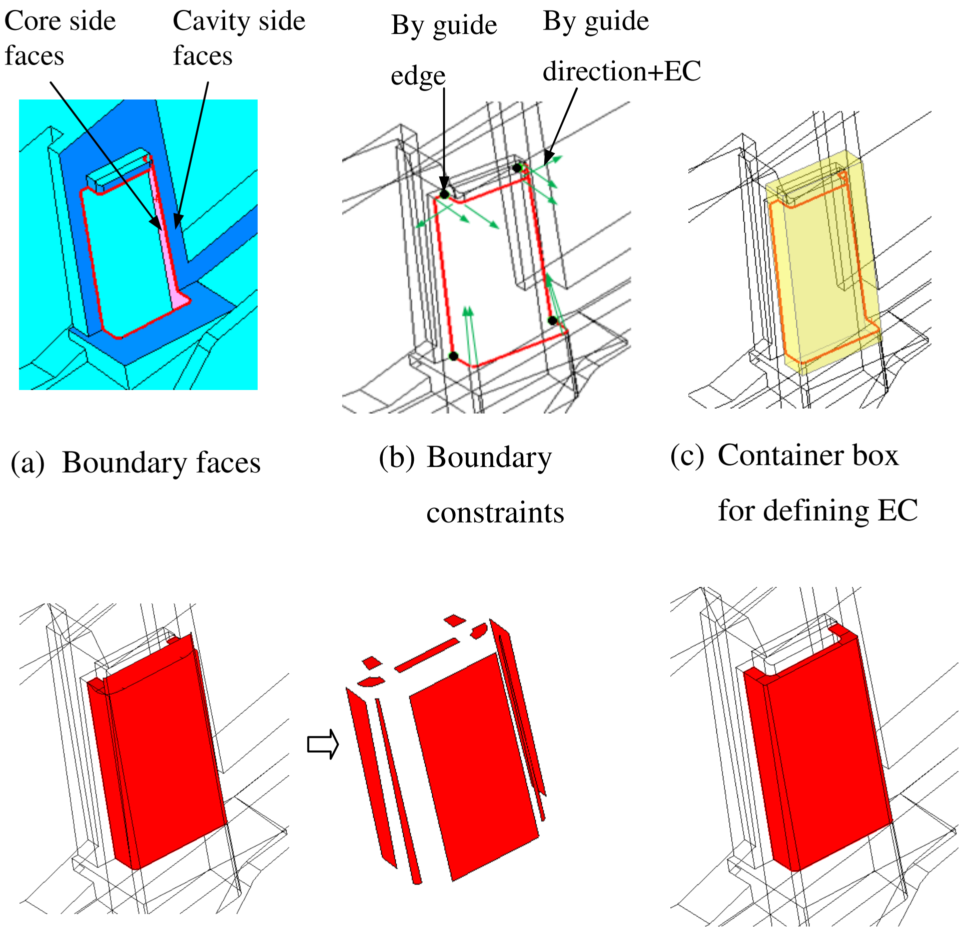
(a) All the **IPL** loops

(b) Shut-off surfaces

Fig.5.15. Case study 2 for creating shut-off surfaces

With regards to the **IPL** loop L_4 which is classified into Category 4, the intermediate results of NS_TY4 surfaces are illustrated in Fig.5.16. The cavity and core boundary faces are first searched as shown in Fig.5.16 (a). Fig.5.16 (b) shows the results of boundary constraints corresponding to their vertices. The boundary constraints at the four vertices (highlighted in solid circle) are defined using their adjacent guide edges.

The boundary constraints for the other vertices are defined by the guide directions and the end conditions EC. The container box to define EC is shown in Fig.5.16 (c). Fig.5.16 (d) and (d1) shows all the individual loft surfaces generated using the presented algorithms. After trimming all the intersected surfaces with one another using SolidWorks APIs, the final shut-off surfaces for the **IPL** loop are generated as shown in Fig.5.16 (e).



(a) Boundary faces

(b) Boundary constraints

(c) Container box for defining EC

(d) Loft surfaces

(d1) Exploded view of (d)

(e) Final shut-off surfaces

Fig.5.16. Illustration of creating NS_TY4 shut-off surfaces

5.4 Summary

This chapter presented an automated approach for generating shut-off surfaces from all the inner parting line loops defined using the **FTMR** approach previously. The approach classifies all the inner parting line loops into four categories based on their geometric characteristics and the associated algorithms were developed for each of the categories to generate shut-off surfaces. The algorithms consider mouldability criteria as well as the geometrical requirements for moulded products. Compared to a single patch surface from a closed edge boundary [Kato1992] [Pla-Garcia2006], the generated shut-off surfaces presented in this thesis are more robust for mould applications.

CHAPTER 6

AUTOMATIC DESIGN OF CAVITY/CORE INSERTS AND LOCAL TOOLS

In the previous chapters, all the faces of a moulded part have been identified as cavity faces, core faces and undercut faces. All the inner and outer parting lines have also been defined. In addition, the parting surfaces (**PS**) have been generated from the outer parting line (**OPL)** loop and all the inner parting line (**IPL)** loops have also been patched using the shut-off surfaces (**SO)**. In this chapter, the approaches and procedures to automatically design the cavity/core inserts and associated local tools (*i.e.* side-cores and side-cavities) are presented.

6.1 Procedure to design cavity/core inserts and incorporated local tools

In this thesis, the procedure to design cavity/core inserts and incorporated local tools comprises of three steps as below:

Step 1:

The preliminary cavity and core inserts are first created using the parting entities defined and parting and shut-off surfaces generated previously regardless of undercut features. As a result, the moulded product are completely split into the cavity and core inserts, which are not the final cavity and core inserts in the case of any presence of undercut features since the resultant cavity and core inserts still contain the portions of undercut features which must be released by local tools in advance.

Step 2:

Local tools are created for all the undercut features **(UF)** in terms of their geometry and release directions.

Step 3:

The final cavity and core inserts are consequently generated by Boolean subtraction of the preliminary inserts from the associated local tools generated in the step 2.

6.2 Design of the preliminary cavity and core inserts

Fig.6.1 describes the procedure to design the preliminary cavity and core inserts. Firstly, an impression assembly is created [SolidWorks2008]. The assembly is composed of a void cavity part and a void core part as well as the moulded product itself.

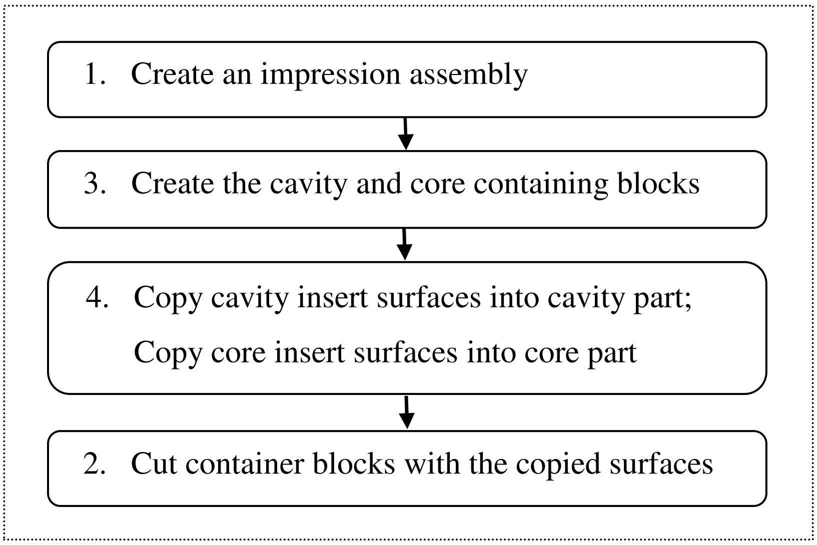

1. Create an impression assembly
3. Create the cavity and core containing blocks
4. Copy cavity insert surfaces into cavity part; Copy core insert surfaces into core part
2. Cut container blocks with the copied surfaces

Fig.6.1. Procedure to design the preliminary core and cavity inserts

Secondly, two containing blocks (rectangular shape or circular shape) enclosing the moulded part are created. The sizes of the containing blocks are computed based on the boundary box of the moulded part and the given gap values, which imply the allowable spaces for the screws and process parameters of the moulding.

Thirdly, all the cavity insert surfaces (**pCavityInsertSurface**) and the core insert surfaces (**pCoreInsertSurface**) are obtained from the moulded part and copied into the cavity part and the core part respectively. **pCavityInsertSurface** comprises of three groups of surfaces, *i.e.* cavity side faces (**pCavitySideFace**) from part body, all the shut-off surfaces and all the parting surfaces (as expressed in Eq. 6-1). Similarly, **pCoreInsertSurface** comprises of core side faces (**pCoreSideFace**) from part body, all the shut-off surfaces and all the parting surfaces (as expressed in Eq. 6-2).

$$\mathbf{pCavityInsertSurface = pCavitySideFace + SO + PS} \qquad (6\text{-}1)$$

$$\mathbf{pCoreInsertSurface = pCoreSideFace + SO + PS} \qquad (6\text{-}2)$$

It is noted that **pCavitySideFace** and **pCoreSideFace** in (6-1) and (6-2) are not exactly equal to the cavity faces and the core faces identified in the **FTMR** parting approach in the case of any presence of undercut features**.** Cavity side faces (**pCavitySideFace)** and core side faces (**pCoreSideFace**) are fully determined by the **OPL** loop and the targeted **IPL** loops where shut-off surfaces have been generated**.** Consequently, all faces of the part body are separated into **pCavitySideFace** and **pCoreSideFace** based on the parting lines. Undercut features are included within **pCavitySideFace** and **pCoreSideFace**.

Finally, **pCavityInsertSurface** and **pCoreInsertSurface** are knitted together and used to cut the container blocks of the cavity and core parts respectively. As a result, the preliminary cavity insert and the core insert are created. Local tools for undercut features have not been considered for the inserts yet.

The final cavity and core inserts will be generated using Boolean subtraction of the preliminary inserts from their incorporated local tools designed in next section.

6.3 Design of local tools

In a moulding process, incorporated local tools must be removed away from the moulding prior to its ejection. The depression features are moulded by the side-cores and the protrusion features by the side-cavities.

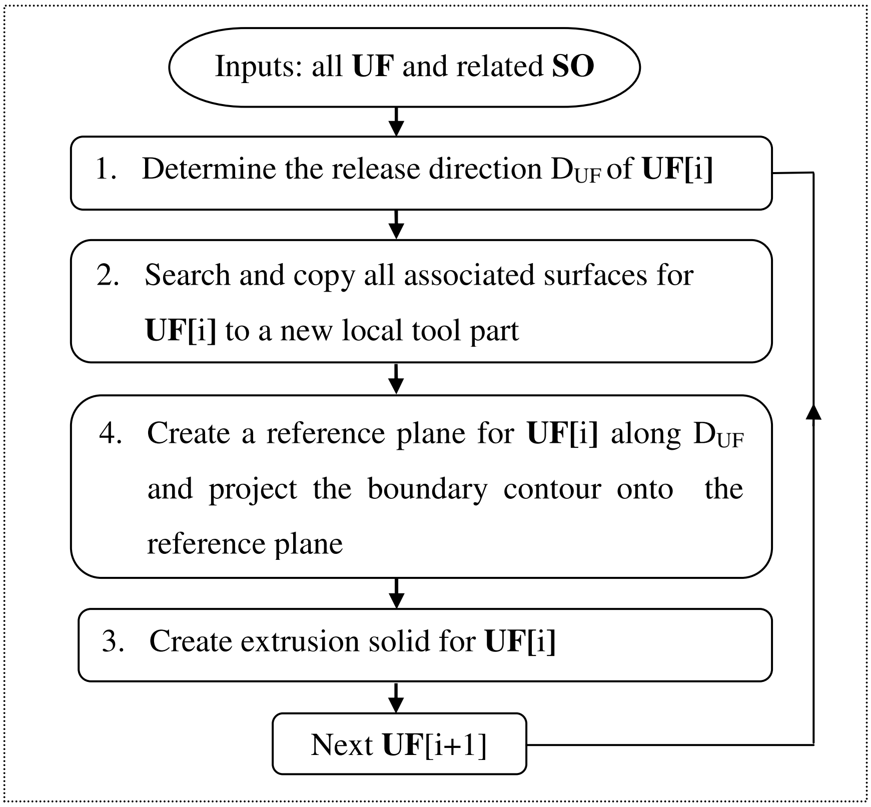

Inputs: all **UF** and related **SO**
1. Determine the release direction D_{UF} of **UF[i]**
2. Search and copy all associated surfaces for **UF[i]** to a new local tool part
4. Create a reference plane for **UF[i]** along D_{UF} and project the boundary contour onto the reference plane
3. Create extrusion solid for **UF[i]**
Next **UF**[i+1]

Fig.6.2. Approach for creating local tools

Fig.6.2 introduces an approach to creating local tools for a mould. Inputs are all undercut features defined and associated shut-off surface generated in the previous chapters. As for an undercut feature **UF**[i], its release direction D_{UF} is first determined [Fu1997] [Ye2001] [Mochizuki1992]. Then all the faces of **UF**[i] and associated shut-off surfaces are searched and copied into a new local tool (side-core or side-cavity) part. Subsequently, the boundary contour of **UF**[i] is projected onto a reference plane. As for an external undercut feature [Ye2001], the reference plane is referred to one side of the main inserts, which release direction D_{UF} intersects with as illustrated in Fig.6.3. In the case of an internal undercut feature which needs lifter mechanism to mould, the reference plane is defined by its release direction D_{UF}, drawn depth [Ye2001] and the given gap as illustrated in Fig.6.4. Finally, the local tool of the undercut feature **UF** is created by means of an extrusion solid [SolidWorks2008] towards the **UF**'s reverse release direction ($-D_{UF}$) as shown in Fig.6.5.

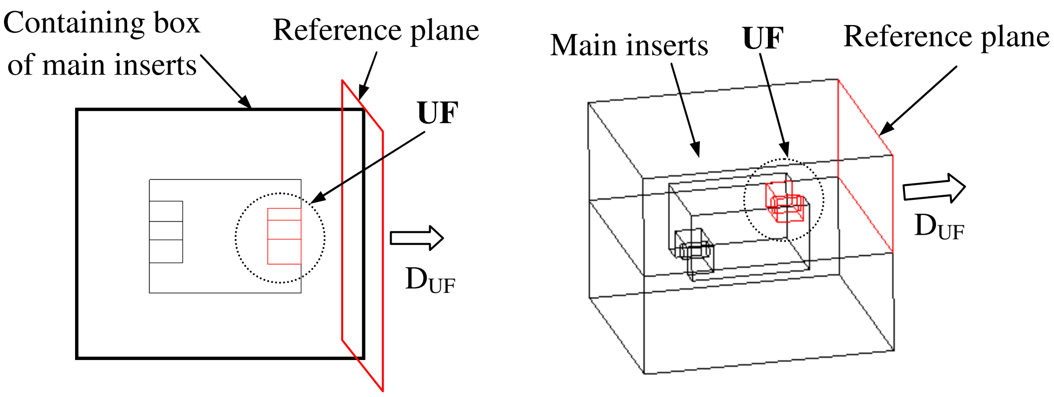


Fig.6.3. Illustration of the reference plane for an external undercut feature

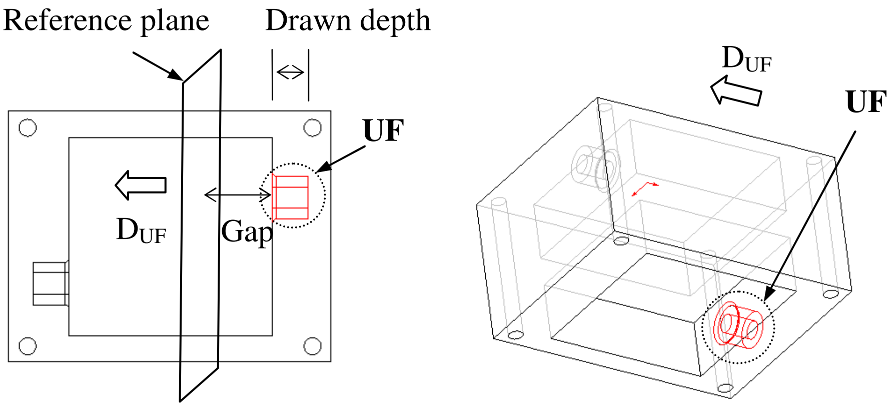
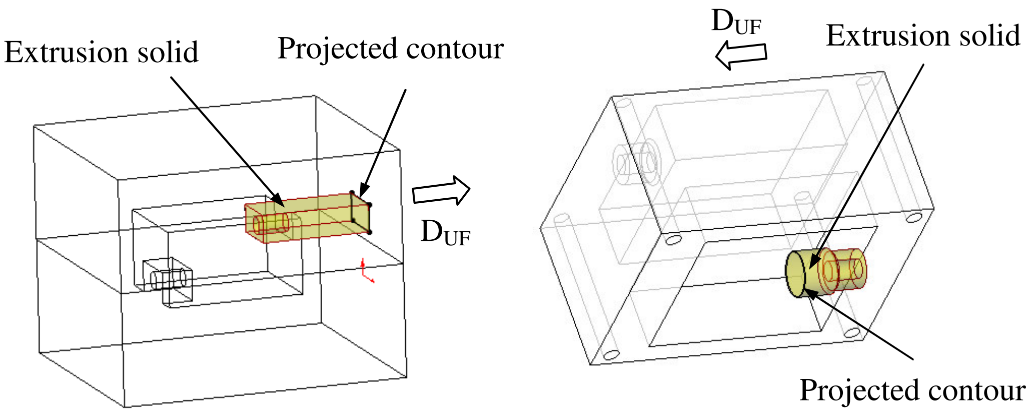


Fig.6.4. Illustration of the reference plane for an internal undercut feature



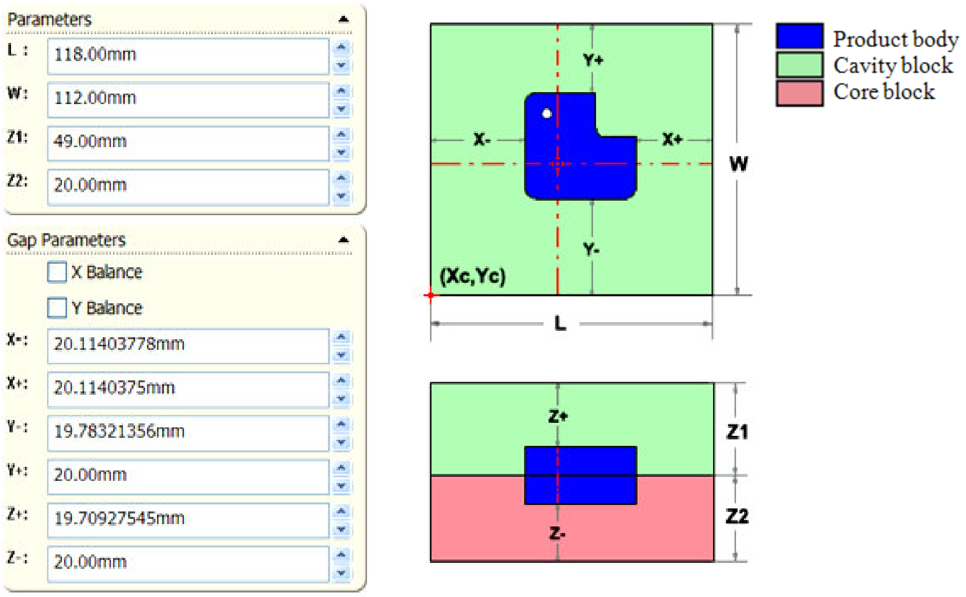
(a) External undercut feature

(b) Internal undercut feature

Fig.6.5. Create the extrusion body for an undercut feature

6.4 Implementation and case studies

The approaches have been implemented based on the SolidWorks platform. Fig.6.6 shows the user interface for defining the size of the containing blocks for the cavity and core inserts.


Parameters

L : 118.00mm

W: 112.00mm

Z1: 49.00mm

Z2: 20.00mm

Gap Parameters

□ X Balance

□ Y Balance

X-: 20.11403778mm

X+: 20.1140375mm

Y-: 19.78321356mm

Y+: 20.00mm

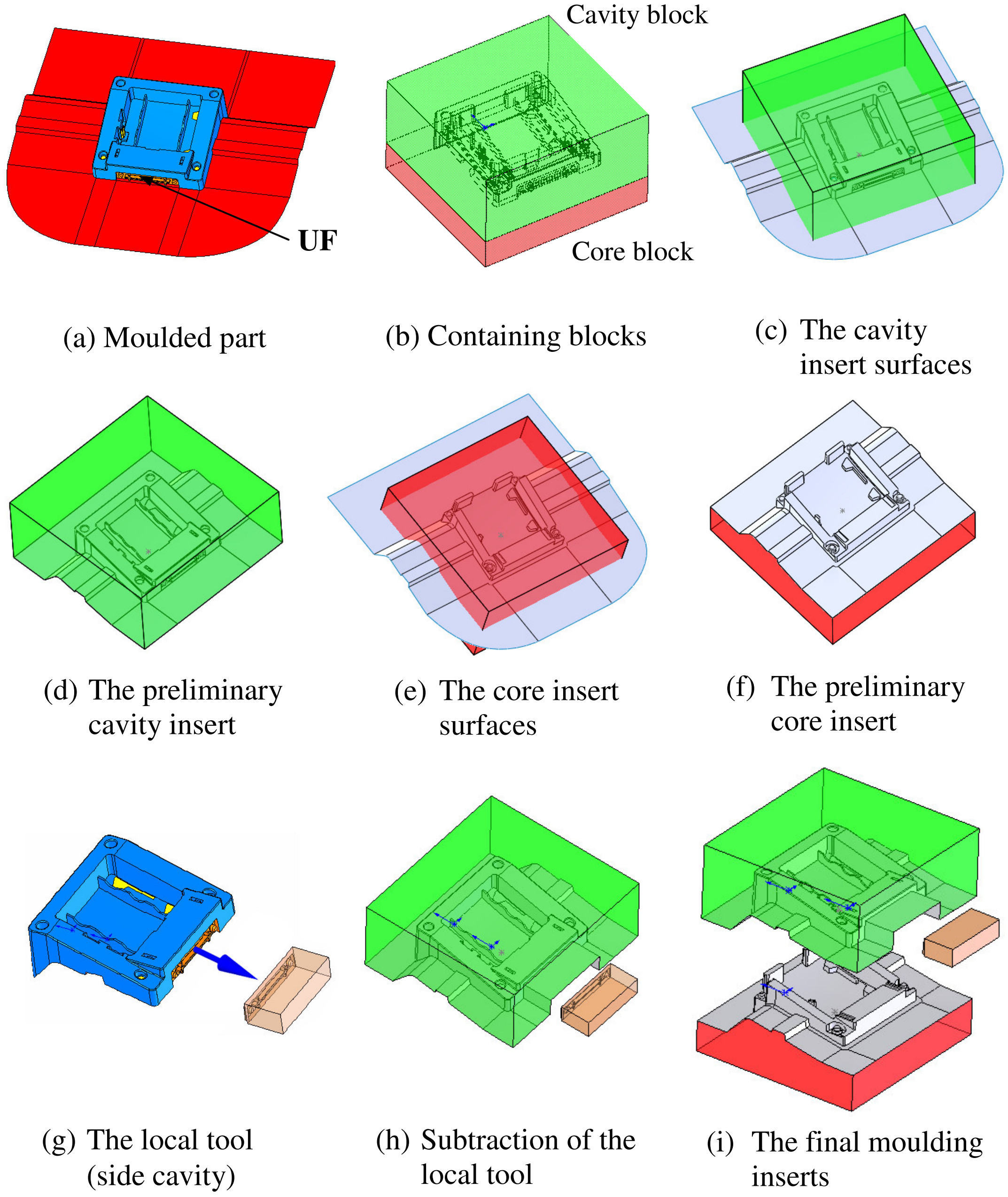
Z+: 19.70927545mm

Z-: 20.00mm

Fig.6.6. User interface for defining the size of container blocks

6.4.1 Case study 1

Fig.6.7 (a) shows a plastic moulded part with one undercut feature **UF**. All parting entities have been defined previously, and all the shut-off and parting surfaces have been generated as well. Based on the presented approaches for generating cavity/core inserts and local tools, two containing blocks are first generated for the cavity and the core inserts respectively in Fig.6.7 (b). Then, all the cavity insert surfaces are copied into the cavity insert in Fig.6.7 (c) and used to cut the cavity containing block so as to generate the preliminary cavity insert body as shown in Fig. 6.7 (d). Similarly, the preliminary core insert is created by copying all the core insert surfaces and trimming with the core containing block as shown in Fig.6.7 (e) and (f) in turn. Now, the preliminary cavity and core inserts are created. To create the local tool of the undercut feature **UF**, all the faces of the undercut feature are extracted and an extrusion body is created for using the approach presented in 6.3. The result of the local tool is shown in Fig.6.7 (g). The final cavity insert is consequently generated by Boolean subtraction with this local tool in Fig.6.7 (h). The final core insert is the same as the preliminary core insert since the core side of the moulding does not contain any undercut feature. As a result, Fig.6.7 (i) shows the exploded view of the final cavity/core inserts and the only local tool for the moulded part.



(a) Moulded part

(b) Containing blocks

(c) The cavity insert surfaces

(d) The preliminary cavity insert

(e) The core insert surfaces

(f) The preliminary core insert

(g) The local tool (side cavity)

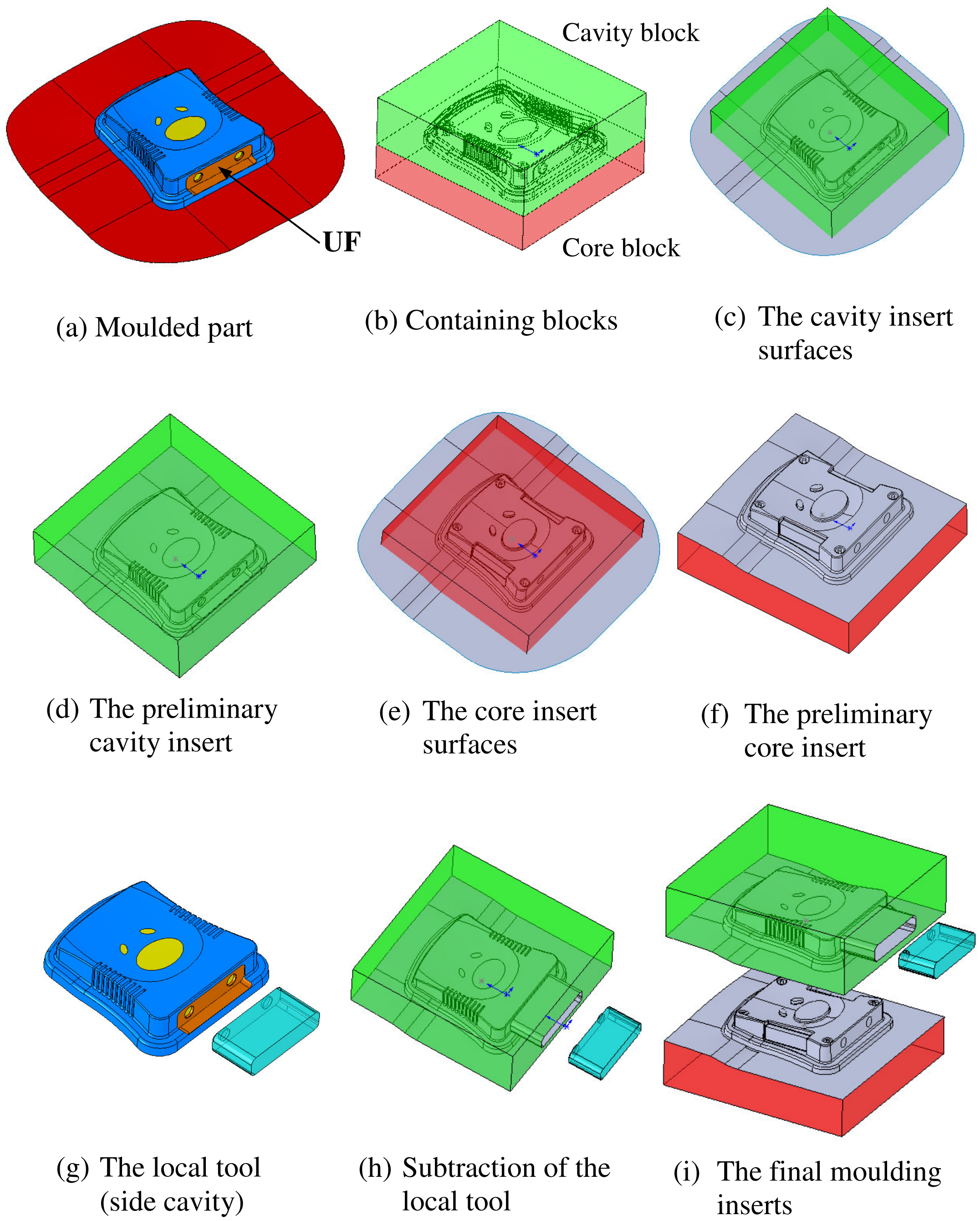
(h) Subtraction of the local tool

(i) The final moulding inserts

Fig.6.7. Case study 1 for the design of cavity/core inserts and local tools

6.4.2 Case study 2

Fig.6.8 (a) shows another plastic moulded part. First of all, two containing blocks are generated for the cavity insert and the core insert respectively in Fig.6.8 (b). Fig.6.8 (c) and (e) show the copied cavity insert surfaces and core insert surfaces respectively. After trimming these surfaces with the corresponding containing blocks, the preliminary cavity and core inserts are generated and shown in Fig.6.8 (d) and (f). In the following step, the local tool for the undercut feature is created using the presented method in as shown in Fig.6.8 (g). Fig.6.8 (h) shows the final cavity insert after the subtraction of the preliminary cavity insert from the local tool. Fig.6.8 (i) shows the final cavity/core inserts and the local tool for the moulded part.



(a) Moulded part

(b) Containing blocks

(c) The cavity insert surfaces

(d) The preliminary cavity insert

(e) The core insert surfaces

(f) The preliminary core insert

(g) The local tool (side cavity)

(h) Subtraction of the local tool

(i) The final moulding inserts

Fig.6.8. Case study 2 for the design of cavity/core inserts and local tools

6.5 Summary

This chapter presented the approaches and procedures to automatically design the cavity/core inserts and the incorporated local tools for a moulded product. The preliminary cavity and core inserts are first created using pre-defined parting entities, parting surfaces and shut-off surfaces. Then, the local tools are created using extrusion method. The final cavity and core inserts are generated using Boolean subtraction of the preliminary cavity and core inserts from their associated local tools. However, the combination of multiple local tools into one piece is not implemented in the presented approach. In addition, the simplification of a moulding by modifying the shapes of local tools is not discussed within the scope of this thesis. These two issues should be considered in practical applications.

CHAPTER 7

PARTING APPROACH FOR MULTI-INJECTION MOULDS

The design for multiple injection moulds is more complex than the one for single injection moulds. Different from a single injection mould, the procedure to design multi-injection moulds can be separated into two major processes, *i.e.* moulding strategy and parting approach. In the part of moulding strategy, moulding injection sequences are determined based on the decomposition and disassembly possibility of all the homogeneous portion of moulded products. Kumar and Gupta [Kumar2002] have developed a moulding strategy algorithm to generate a feasible moulding sequence for multiple injection moulds based on the decomposition analysis of multi-material objects of the moulded products. Li and Gupta [Li2003] also presented a moulding strategy algorithm for several typical rotary-platen types of multi-injection moulds according to the precedence constraints resulting from the accessibility and disassembly requirements of the moulded product.

Regarding the parting approach, the generation of the moulding inserts has not been discussed in the previous studies. The objective of parting approach aims to identify parting entities for the moulded products and furthermore to generate the sets of cavity inserts, core inserts and incorporated local tools corresponding to each moulding injection sequence.

This research studies the parting approach of multi-injection moulds rather than developing the moulding strategy. The parting approaches developed previously were applied only for single injection mould applications. One possible way to extend the capability of the parting approaches for single injection moulds to multiple injection moulds is to describe a multiple injection moulded product using a part model with multiple bodies, and then to split one of the part bodies, which is moulded in the first injection stage, exactly as a single injection mould. Subsequently, the associated inserts and local tools for the other homogeneous bodies can be generated using Boolean operations with necessary manufacturing and moulding considerations.

In this chapter, a parting approach for multi-injection moulds has been developed using the parting approaches and algorithms for single injection moulds presented in previous chapters and Boolean operations. As a result, sets of cavity inserts, core inserts and incorporated local tools are generated for each injection sequence.

7.1 Parting approach for multi-injection moulds

Fig.7.1 introduces the parting approach for multi-injection moulds. In this research, a multi-injection moulded part is represented with a CAD part model which comprises of multiple bodies S[nBody] expressed in Eq.7-1. Each body S[i] represents a homogeneous object of the product. As a pre-condition, the moulding sequence for the multi-injection product is given in advance. According to the given moulding sequence, all bodies S[nBody] of the moulded part are then classified into a few body groups $S_{INJECTION}$[nInjection] in terms of injection sequences as expressed in Eq. 7-2 and 7-3.

$$\text{Moulded Product} = \{S[i]\} \qquad \text{for } i= 0,1,2, \ldots \text{ nBody} \qquad (7\text{-}1)$$

$$\text{Moulding Sequence} = \{S_{INJECTION}[j]\} \quad \text{for } j= 0,1,2, \ldots \text{ nInjection} \qquad (7\text{-}2)$$

$$S_{INJECTION}[j] = \{S[k]\} \qquad \text{for } k = a, b, c, \ldots \qquad (7\text{-}3)$$

Where nBody is the total number of bodies of a moulded part, the index j indicates the order of moulding sequences. nInjection is the total number of injection stages for the moulded product; S[k] represents those homogeneous moulding bodies which are moulded during the jth injection stage.

According to above definition, $S_{INJECTION}$[0] is therefore the first moulding object and must be a single body as considered in this thesis. Other $S_{INJECTION}$ are the moulding objects corresponding to other injection sequences. It may contain more solid bodies for a single injection stage.

As described in Fig.7.1, the parting approach for an multi-injection mould is composed of two major processes, *i.e.* (1) Create inserts for the first injection object $S_{INJECTION}$[0], (2) Create inserts for the other injection stages $S_{INJECTION}$.

In the first process, the cavity insert, core inserts and their local tools are generated for the moulding object $S_{INJECTION}$[0], which will be moulded in the first injection stage. Since $S_{INJECTION}$[0] is a single body, the approaches and algorithms introduced in the previous chapters can be fully applied to split the moulding. Firstly, all parting entities (*i.e.* cavity faces, core faces, undercut features, **OPL** and **IPL**) of $S_{INJECTION}$[0] are identified using the **FTMR** parting approach. Then parting surfaces **PS** and shut-off surfaces **SO** are generated using the approaches and algorithms developed in Chapter 4 and Chapter 5. Lastly, the cavity inert, the core insert and incorporated local tools are designed for $S_{INJECTION}$[0] using the approaches introduced in Chapter 6. One thing should be noted here is that the containing block sizes of the cavity/core inserts for a multi-injection mould are determined based on all the bodies of the moulded product, not the single body $S_{INJECTION}$[0] only because the cavity/core inserts must be large enough to enclose all the bodies during different injection stages. Consequently, the output of the first process is the cavity insert CavityInsert[0], the core insert

CoreInsert[0] and their local tools for the first injection stage. CavityInsert[0] and CoreInsert[0] will be used to design the other inserts in the second process.

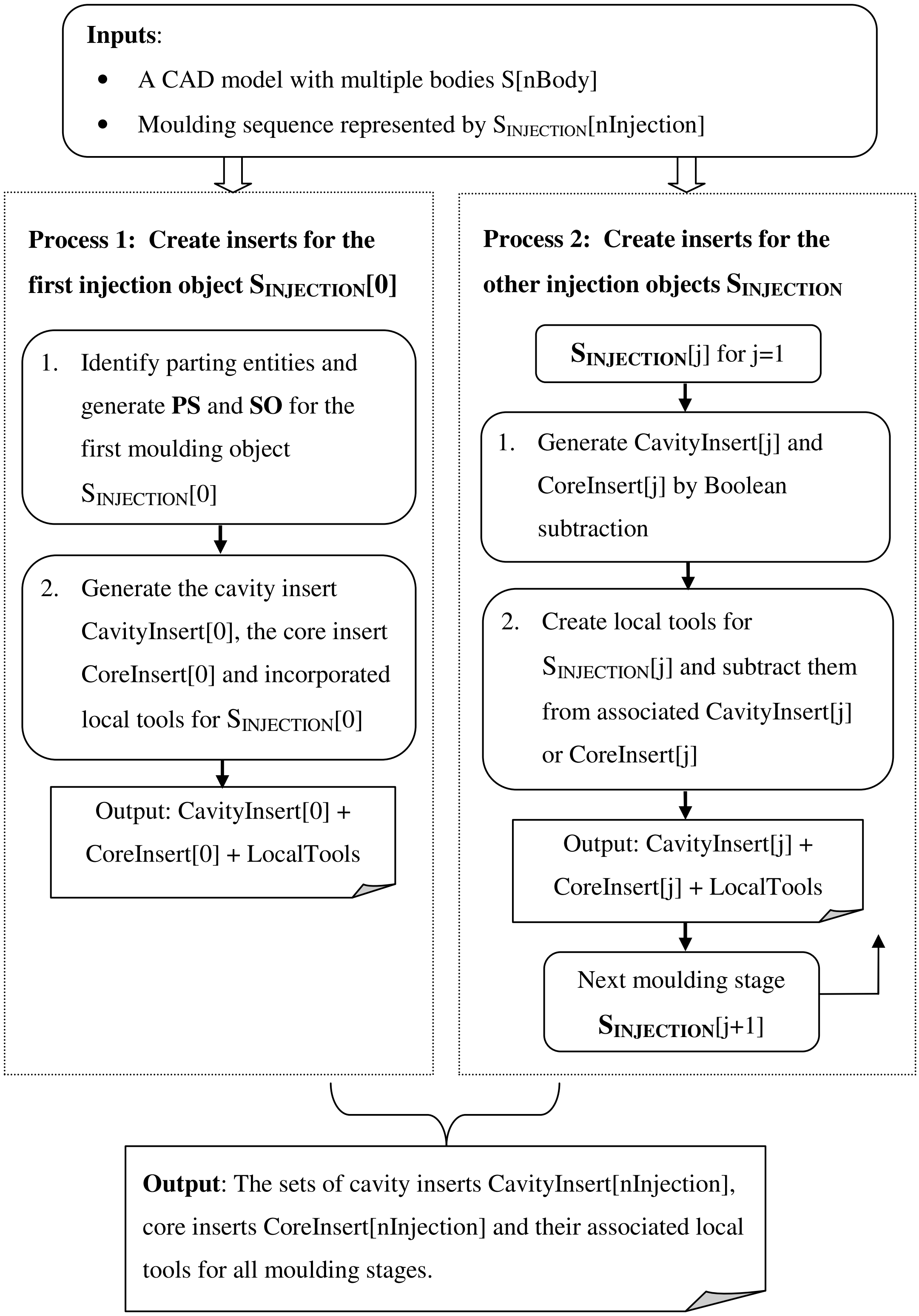

Inputs:
A CAD model with multiple bodies S[nBody]
Moulding sequence represented by S_INJECTION[nInjection]
Process 1: Create inserts for the first injection object S_INJECTION[0]
1. Identify parting entities and generate PS and SO for the first moulding object S_INJECTION[0]
2. Generate the cavity insert CavityInsert[0], the core insert CoreInsert[0] and incorporated local tools for S_INJECTION[0]
Output: CavityInsert[0] + CoreInsert[0] + LocalTools
Process 2: Create inserts for the other injection objects S_INJECTION
S_INJECTION[j] for j=1
1. Generate CavityInsert[j] and CoreInsert[j] by Boolean subtraction
2. Create local tools for S_INJECTION[j] and subtract them from associated CavityInsert[j] or CoreInsert[j]
Output: CavityInsert[j] + CoreInsert[j] + LocalTools
Next moulding stage S_INJECTION[j+1]
Output: The sets of cavity inserts CavityInsert[nInjection], core inserts CoreInsert[nInjection] and their associated local tools for all moulding stages.

Fig.7.1. Parting approach for multiple injection moulds

The second process is to design other sets of cavity/core inserts and the associated local tools corresponding to their injection sequences using Boolean operations (in Fig.7.1). As for the jth moulding object represented by $S_{INJECTION}$[j], its cavity insert CavityInsert[j] and core insert CoreInsert[j] are generated using Boolean subtraction of $S_{INJECTION}$[j] from the last inserts CavityInsert[j-1] and CoreInsert[j] respectively. Then, the associated local tools for each body of $S_{INJECTION}$[j] are generated using the approaches presented in the previous chapter 6.3 since each body of $S_{INJECTION}$[j] can certainly be treated as a single part. Furthermore, the final CavityInsert[j] and CoreInsert[j] for the jth moulding stage should be subtracted from all the local tools of $S_{INJECTION}$[j] using Boolean operations.

After completing the entire processes 1 and 2, the sets of cavity inserts CavityInsert[nInjection], core inserts CoreInsert[nInjection] and incorporated local tools are generated corresponding to the overall moulding stages.

7.2 Case studies

7.2.1 Case study1

Fig.7.2 shows a multi-injection moulded product (handle). The product comprises of four solid bodies represented by S[0], S[1], S[2] and S[3] respectively. As described in Tab.7.1, S[0] is the plastic fame. S[1] is the handle cover which material is rubber. S[2] and S[3] are the two indicating lamps which are made of transparent plastic. Tab.7.1 also gives the moulding sequences for the product. The moulding needs three injection stages in terms of the corresponding homogeneous objects represented by $S_{INJECTION}$[0], $S_{INJECTION}$[1] and $S_{INJECTION}$[2] in turn. The first injection object $S_{INJECTION}$[0] contains the single body S[0]. The second injection object $S_{INJECTION}$[1]

contains the two homogeneous bodies S[2] and S[3]. While, The third injection object $S_{INJECTION}[2]$ contains the rubber body S[1].

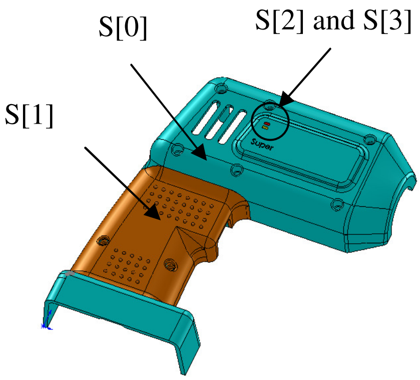
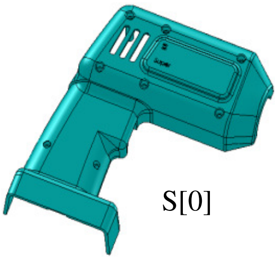
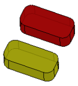
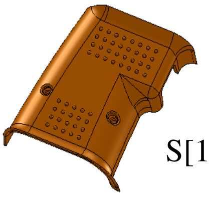


Fig.7.2. A multi-injection moulded product (handle)
(Model from Manusoft Technologies Pte Ltd)

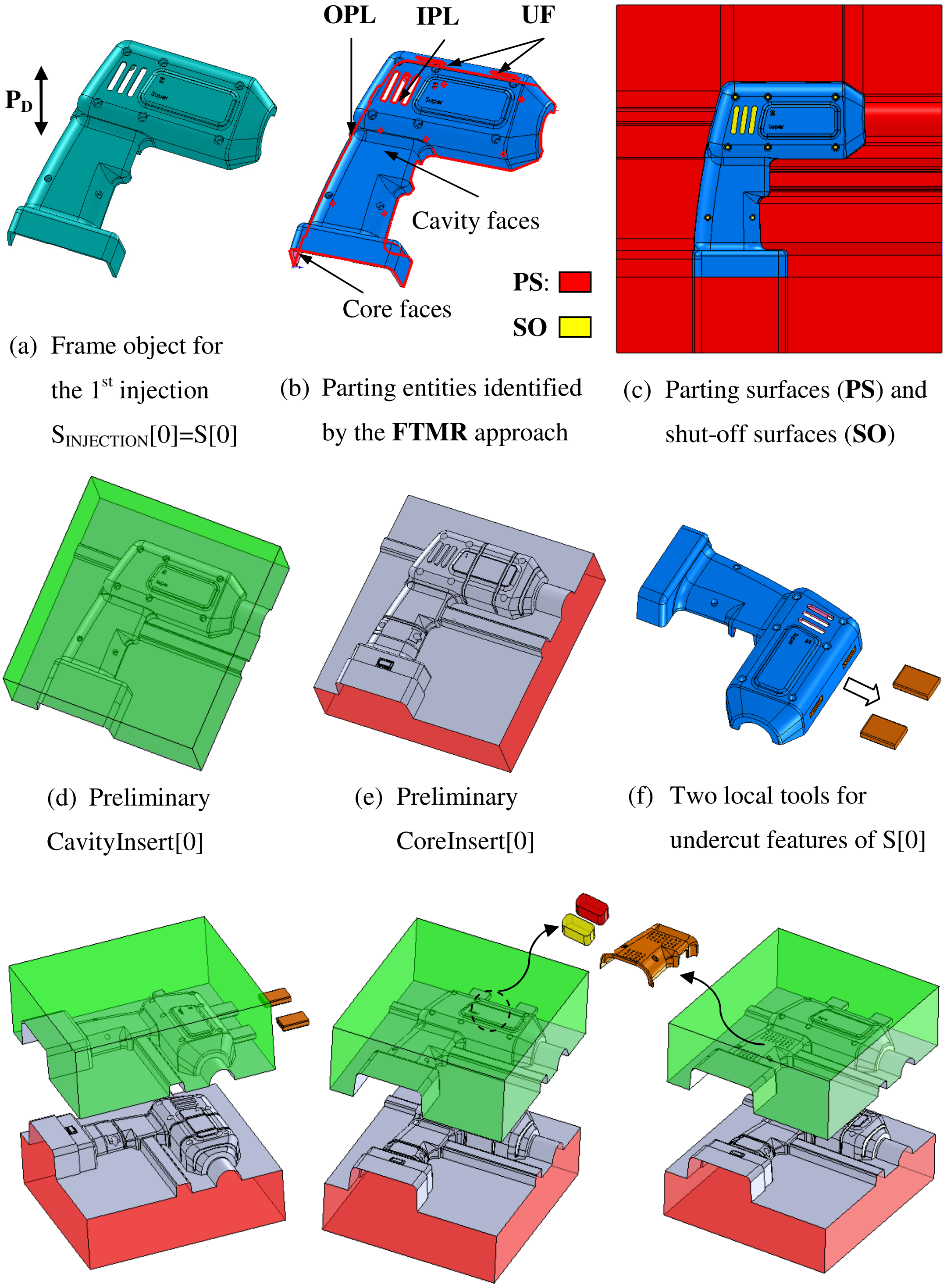
Moulding objects	 S[0]	 S[2] and S[3]	 S[1]
Function	Frame	Indicating lamps	Handle cover
Material	Plastic	Transparent plastic	Rubber
Moulding sequence	1st injection $S_{INJECTION}[0] = S[0]$	2nd injection $S_{INJECTION}[1] = \{S[2], S[3]\}$	3rd Injection $S_{INJECTION}[2] = S[1]$

Tab.7.1 Description of moulding objects and moulding sequences of a multi-injection product (handle)

Fig.7.3 illustrates the parting processes and the intermediate results of the three moulding stages using the presented parting approach. Fig.7.3 (a) shows the body S[0] and the given parting direction $\mathbf{P_D}$ for the first injection $S_{INJECTION}[0]$. First, all parting entities of S[0] (*i.e.* cavity faces, core faces, undercut features, **OPL** and **IPL**) are indentified using the **FTMR** parting approach presented in Chapter 3 as shown in Fig.4.3 (b). There are two undercut features in this product. Then, parting surfaces (**PS**) and shut-off surfaces (**SO**) are generated using the approaches presented in Chapter 4 and Chapter 5 respectively. The results are shown in Fig.7.3 (c). Subsequently, the preliminary cavity and core inserts are created using the defined parting entities and related surfaces (**PS** and **SO**) as shown in Fig.7.3 (d) and (e) respectively. Fig.7.3 (f) shows the two local tools (side-cavities) generated for the two undercut features of the moulded object S[0] using the extrusion approach presented in 6.3. The final cavity insert CavityInsert[0] is obtained using Boolean subtraction of the two local tools from the preliminary cavity insert. The final core insert CoreInsert[0] is the same since there is no side-core in the moulding object S[0]. Consequently, the set of inserts for the first injection stage are generated successfully and shown in Fig.7.3 (g).

The sets of inserts for the other two injection stages are generated based on CavityInsert[0] and CoreInsert[0]. As for the second injection stage $S_{INJECTION}$[1] which comprises of two bodies S[2] and S[3], the corresponding cavity insert CavityInsert[1] is generated by the subtraction of S[2] and S[3] from CavityInsert[0]. The core insert CoreInsert[1] is the same as CoreInsert[0] since there are no related objects to be moulded in this stage. Fig.7.3 (h) gives the results. Similarly, the cavity insert CavityInsert[2] is generated using Boolean subtraction of the body S[1] from the last cavity insert CavityInsert[1]. CoreInsert[2] is still same as CoreInsert[1] because there is no any side core in $S_{INJECTION}$[2]. Fig.7.3 (i) gives the final inserts for the third moulding stage $S_{INJECTION}$[2].

As a result, the moulded product (handle) can be moulded by the three sets of inserts corresponding to three moulding stages.



(a) Frame object for the 1st injection $S_{INJECTION}[0]=S[0]$

(b) Parting entities identified by the **FTMR** approach

(c) Parting surfaces (**PS**) and shut-off surfaces (**SO**)

(d) Preliminary CavityInsert[0]

(e) Preliminary CoreInsert[0]

(f) Two local tools for undercut features of S[0]

(g) Final CavityInsert[0], CoreInsert[0] and local tools for 1st injection

(h) CavityInsert[1] and CoreInsert[1] by Boolean subtraction from $S_{INJECTION}[1]$

(i) CavityInsert[2] and CoreInsert[2] by Boolean subtraction from $S_{INJECTION}[2]$

Fig.7.3. Case study 1 for the parting approach for multi-injection moulds

7.2.2 Case study2

Fig.7.4 shows another multi-injection moulded product (toothbrush). The product comprises of two homogeneous solid objects represented by S[0] and S[1] respectively. As described in Tab.7.2, S[0] is the plastic frame moulded in the first injection, while S[1] is the rubber handle moulded in the second injection stage.

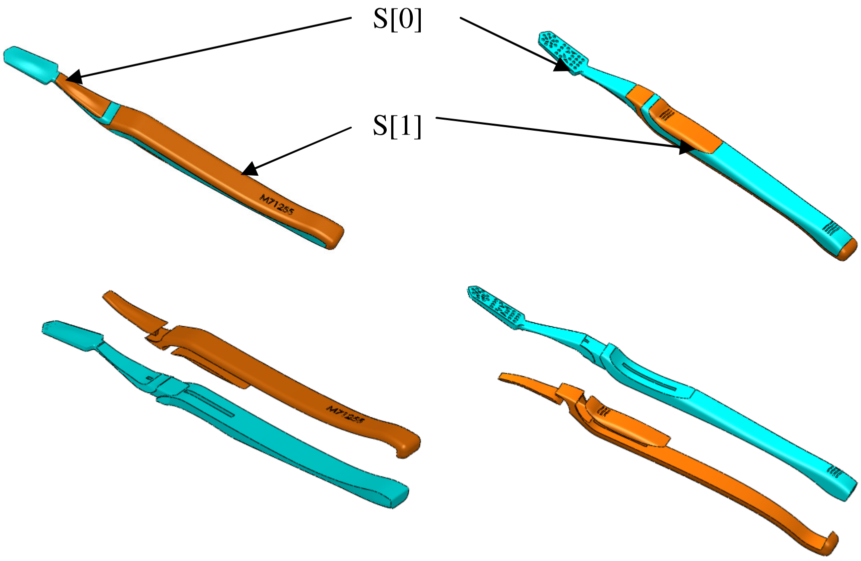
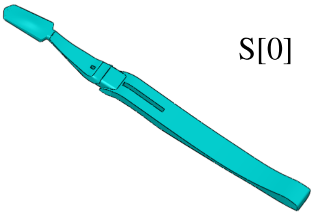
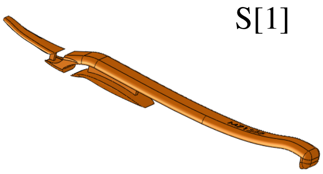


Fig.7.4. A multi-injection moulded product (toothbrush)
(Model from Manusoft Technologies Pte Ltd)

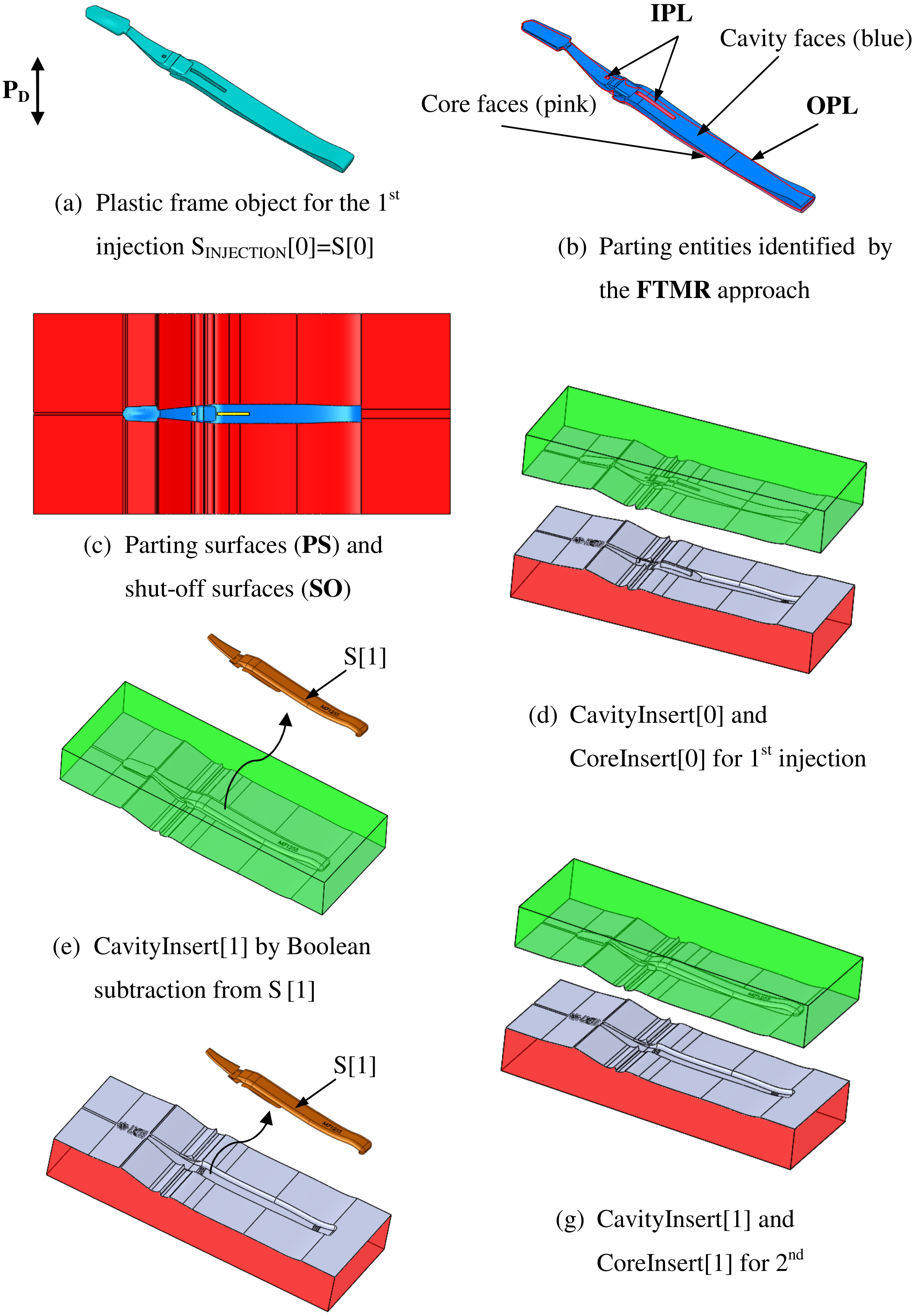
Moulding objects		
Function	Plastic frame	Rubber handle
Material	Plastic	Rubber
Moulding sequence	1st injection $S_{INJECTION}[0] = S[0]$	2nd Injection $S_{INJECTION}[1] = S[1]$

Tab.7.2 Description of moulding objects and moulding sequences of a multi-injection product (toothbrush)

Fig.7.5 illustrates the parting processes and the results for the toothbrush which needs two moulding stages. Fig.7.5 (a) shows the body S[0] and the given parting direction $\mathbf{P_D}$ for the first injection $S_{INJECTION}$[0]. First, the body S[0] is split as a single injection mould. All cavity and core faces, **OPL** and **IPL** are indentified using the **FTMR** parting approach as shown in Fig.7.5 (b). Furthermore, parting surfaces (**PS**) and shut-off surfaces (**SO**) are also generated as shown in Fig.7.5 (c). Subsequently, the cavity insert CavityInsert[0] and the core insert CoreInsert[0] for the first injection $S_{INJECTION}$[0] are created based on the defined parting entities and related surfaces (**PS** and **SO**) using the approach presented in Chapter 6 as shown in Fig.7.5 (d).

The set of cavity and core inserts for the second injection stage $S_{INJECTION}$[1], represented by S[1], is then generated using Boolean operations from CavityInsert[0] and CoreInsert[0]. CavityInsert[1] is generated by the subtraction of S[1] from CavityInsert[0] as illustrated in Fig.7.5 (e). Similarly, CoreInsert[1] is generated by the subtraction of S[1] from CoreInsert[0] as shown in Fig.7.5 (f). Fig.7.5 (g) gives the final results for the second injection.

As a result, the moulded product (toothbrush) can be moulded by the two sets of inserts (in Fig.7.5 (d) and (g)) corresponding to the required two moulding stages respectively.



(a) Plastic frame object for the 1st injection $S_{INJECTION}[0]=S[0]$

(b) Parting entities identified by the **FTMR** approach

(c) Parting surfaces (**PS**) and shut-off surfaces (**SO**)

(d) CavityInsert[0] and CoreInsert[0] for 1st injection

(e) CavityInsert[1] by Boolean subtraction from S [1]

(f) CoreInsert[1] by Boolean subtraction from S [1]

(g) CavityInsert[1] and CoreInsert[1] for 2nd

Fig.7.5. Case study 2 for the parting approach for multi-injection moulds

7.3 Summary

In this chapter, a parting approach for generating the sets of cavity inserts, core inserts and their local tools corresponding to each moulding injection stage is presented. The approach applies the parting approaches and algorithms introduced in the previous chapters and Boolean subtraction operations. The approach focuses on the automated generation of solid inserts for each moulding sequence. However, the approach does not discuss the design of the feeding system and the associated mechanisms for different moulding sequences, which is also important in multi-injection mould design. In addition, the optimization of moulding design for a multi-injection mould is not addressed in this thesis. For instance, a few local tools could be combined into a single local tool in practical design. Moreover, moulding processes could be simplified by adjusting the size of local tools. These issues would need to be studied in future.

CHAPTER 8

CONCLUSIONS AND RECOMMENDATIONS

8.1 Conclusions

Automated parting methodologies for the identification of parting entities, the generation of parting surfaces and shut-off surfaces, and the design of core/cavity inserts and associated local tools for injection moulded products have been developed in this research. The presented parting methodologies have been implemented based on the SolidWorks platform using Visual C++ programming language. Case studies show that the developed parting methodologies are able to automatically identify parting entities, create shut-off surfaces and parting surfaces, and finally generate the cavity, core inserts and their local tools for complex injection moulded products.

An automated parting approach based on Face Topology and Mouldability Reasoning (**FTMR**) was developed to automatically identify cavity and core faces, inner and outer parting line loops, and undercut features for injection moulded products. The case studies show that the **FTMR** approach is robust for moulded products with free-form surfaces, complex geometry and geometry imperfections. An Error Correction and Feedback System (**ECFS**) was developed and incorporated within the **FTMR** parting approach**.** The **ECFS** provides the capability of visibly locating and correcting possible errors during the parting process. If a moulded product cannot be split completely by the **FTMR** parting approach for some reasons, the **ECFS** can assist in locating the places where the parting process is not well performed and correcting the errors correspondingly. An automated approach was developed for creating parting surfaces from the outer parting line loop for injection moulds. The generated NURBS ruled parting surfaces are easy to modify and machine since all the surfaces are expressed using linear equations along one of the four major extrusion directions, *i.e.* X+, X-, Y+ and Y-. In addition, an automated and novel approach for generating shut-off surfaces from the target inner parting line loops was also developed in the thesis. The approach classifies all inner parting line loops into four categories based on their geometric characteristics and the related algorithms were developed for each of the categories to generate ruled and loft shut-off surfaces. The generated shut-off surfaces are satisfied with mould applications because the algorithms consider the mouldability criteria as well as the geometrical requirements. Subsequently, an automated method was developed for the design of the cavity, core inserts and associated local tools. Case studies have demonstrated that the method can determine the size of inserts and generate the solid bodies for the cavity, core inserts and local tools using all the identified parting entities and generated surfaces. The approach is effective because all main inserts and local tools are able to be generated in a single process. Moreover, a parting approach was presented to generate the sets of cavity/core inserts and associated local tools corresponding to different moulding sequences for multi-injection moulds in a single process by means of the developed approaches and algorithms for single injection moulds and corresponding Boolean operations.

This research has successfully overcome a few crucial bottlenecks of parting systems for injection mould design applications. Some significant contributions are as follows:

1) The **FTMR** parting approach is effective because all parting entities, *i.e.* cavity/core faces, undercut features, inner and outer parting line loops, are identified in a single process. The **FTMR** parting approach is also robust for free-form surfaces and complex geometries because the approach analyzes moulded

products mainly based on the geometry visibility and mouldability, and therefore it is independent of the complexity of geometry feature and structure.

2) This study is the first to put forward the concept and criterion of Pseudo-Straddle Face (**PSF**) in order to deal with practical products with imperfect draft angles or geometric imperfections commonly resulting from poor design or data transfer among different CAD applications.

3) This research has presented an Error Correction and Feedback System (**ECFS**) and has successfully incorporated it into the parting approach. The **ECFS** can enhance the compatibility and capability of the parting approach for complex and various industrial applications.

4) Compared to radiating and sweeping surfaces from edge boundaries [Tan1990] [Ravi Kumar2003] [Fu2001], the generated ruled parting surfaces in this research can yield better results for moulding applications because the algorithms consider several key parameters of mould design, including knitting of geometry boundaries, machining property, ease of modification, etc.

5) This research is the first to develop a novel approach for patching inner parting loops using shut-off surfaces for injection moulds. The generated shut-off surfaces are robust for moulding applications because the algorithms take mouldability reasoning into account as well as boundary geometry constraints. In contrast, a single patch surface from edge boundary cannot always satisfy the moulding requirements [Kato1992] [Pla-Garcia2006].

6) This research has presented an automated parting approach to design the sets of the cavity, core inserts and associated local tools corresponding to different moulding sequences for multi-injection moulds in a single process.

8.2 Recommendations

The limitations of this research are stated and the recommendations for future study are discussed as below.

1) The functionalities of the Error Correction and Feedback System (**ECFS**) are still in the initial stage of development and thus need to be improved so that the **ECFS** can be more powerful and flexible for more complex situations. Future work will explore ways to implement a knowledge-based environment for the **ECFS** to fulfill various design purposes and applications.

2) As a pre-condition of the developed methodologies, this research assumes that all moulded products cannot be modified during parting processes. It did not discuss how to obtain a better parting solution by revising the design of the original products. It should be helpful for an intelligent mould design system if parting methodologies can detect some poor designs and provide corresponding suggestions for possible modification from the view of mould design. A knowledge-based engine could help the implementation of this idea.

3) This research has introduced an approach to design the sets of cavity/core inserts and their local tools corresponding to each moulding sequence. The optimization of the design of local tools and the design of feeding system with associated mechanism has not been addressed yet. Future work will explore ways to optimize the mould design and the moulding processes by combining multiple local tools into one and adjusting the size of local tools. In addition, the design of the feeding system and the associated mechanism for different moulding sequences is also critical for the automated design of multi-injection moulds and need to be studied in future.

8.3 Potential applications

Besides the applications on an intelligent plastic injection mould system, the presented methodologies and algorithms can also be used in automatic design and tools for the following four areas.

1) The presented work can be extended to the design of casting dies because the structure of die-casting moulds is similar to the one of plastic injection moulds. Both die-casting moulds and plastic injection moulds form the product's profiles between the cavity and core inserts, and the products are ejected after the opening of the core and cavity. Although they use different materials (*i.e.* metal and plastic respectively), both need to determine cavity/core face, identify parting lines, generate parting surfaces, and create cavity/core inserts.

2) The presented parting approaches can also be extended to the design of forging dies because the mould of a forging die also contains two halves similar to the core and the cavity. Die forging uses the two halves as tools to directly deform solid metal to the desired shape. Therefore, there is also the same need to determine the parting direction, parting lines, generate parting surfaces, and create cores and cavities.

3) The presented approaches and algorithms can be used for Computer-Aided Process Planning (CAPP) for mould industry. The information obtained and the results generated by means of the approaches and algorithms presented in this thesis are able to assist the manufacturing plan and the standardization process for mould products.

4) The presented approaches and algorithms in this thesis can be also applied for the optimization of plastic product design. Using the parting results generated by the

presented approaches, product designers are able to detect and correct those improper geometric structure designs in the early stage of product process in terms of their mouldability result. In addition, the parting result can also assist product managers to predict the manufacturing cost of moulded products easily in terms of the number of local tools and their geometric characteristics.

REFERENCES

[Bruzzone1991]

Bruzzone, B. and Floriani, L.D., “Extracting adjacency relationships from a modular boundary model”, Computer-Aided Design, Vol.23, pp.344-356, 1991.

[Chand1970]

Chand D.R. and Kapur, S.S., “An algorithm for convex polyhedron”, Journal of Association for Computerizing Machinery, Vol.17, pp.78-86, 1970.

[Chang1990]

Chang, S.H. and Henderson, M.R., “Three-dimensional shape pattern recognition using vertex classification and vertex-edge graphs” Computer-Aided Design, Vol.22, pp.377-387, 1990.

[Chen1993]

Chen, L.L., Chou, S.Y. and Woo, T.C., “Parting direction for mould and die design” Computer-Aided Design, Vol.25, pp. 762-767, 1993.

[Chen1995]

Chen, L.L., Chou, S.Y. and Woo, T.C., “Partial visibility for selecting a parting direction in mould and die design” Journal of Manufacturing Systems, Vol.14, pp.319-330, 1995.

[Clark1990]

Clark, D.E.R. and Corney, J.R., “Extending FAG based feature recognition” in Applied Surface Modeling, Ed. by Creasy C. F. U. and Craggs C., pp.46-56, Ellis Horwood Limited, 1990.

[DeFloriani1989]

DeFloriani, L., "Feature extraction from boundary model of 3D objects", IEEE Pattern Analysis and Machined Intelligence, Vol.11, pp.785-798, 1989.

[Dhaliwal2001]

Dhaliwal, S., Gupta, S.K., Huang, J. and Kumar, M., "A feature-based approach to automated design of multi-piece sacrificial molds" ASME Journal of Computing and Information Science in Engineering, Vol.1, no.3, pp. 225-234, 2001.

[Dhaliwal2003]

Dhaliwal, S., Gupta, S.K., Huang, J., and Priyadarshi, A., "Algorithms for computing global accessibility cones" Journal of Computing and Information Science in Engineering, Vol. 3(3), pp.200-208, 2003.

[Elber2005]

Elber, G., Chen, X. and Cohen, E., "Mold accessibility via Gauss map analysis" Journal of Computing and Information Science in Engineering, Vol. 5(2), 2005.

[Emad2006]

Emad S. Abouel Nasr; Ali K. Kamrani, "A new methodology for extracting manufacturing features from CAD system", Computers & Industrial Engineering, 51(3), pp,389-415, 2006.

[Fu1997]

Fu, M.W., Nee, A.Y.C. and Fuh, J.Y.H., "Extraction of undercut features and determination of the optimal parting direction in intelligent mould design" Journal of the Institution of Engineers, Vol. 37(4), pp.45-50, 1997.

[Fu1998-1]

Fu, M.W., “Determination of optimal 3-D parting in plastic injection mould design”, PhD Thesis, National University of Singapore, 1998.

[Fu1998-2]

Fu, M.W., Nee, A.Y.C. and Fuh, J.Y.H., “Automatic determination of 3-D parting lines and surfaces in plastic injection mould design”, Annals of the CIRP, Vol.47/1, pp.95-98, 1998.

[Fu1999]

Fu, M.W., Nee, A.Y.C. and Fuh, J.Y.H., “Undercut feature recognition in an injection mould design system” Computer-Aided Design, Vol.31, pp.777-790, 1999.

[Fu2001]

Fu, M.W., Nee, A.Y.C. and Fuh, J.Y.H., “A core and cavity generation method in injection mold design” International Journal of Production Research, Vol. 39, pp. 121-38, 2001.

[Fu2002]

Fu, M.W., Fuh, J.Y.H. and Nee, A.Y.C., “The application of surface visibility and mouldability to parting line generation” Computer-Aided Design, Vol.34, pp469-480, 2002.

[Gan1994]

Gan, J.G., Woo, T.C. and Tang, K., “Spherical Maps: their construction, properties, and approximation” ASME Journal of Mechanical Design, Vol. 116:3573, 1994.

[Gavankar1990]

Gavankar, P. and Henderson, M.R., “Graph-based extraction of protrusions and depressions from boundary representations” Computer-Aided Design; Vol.22, pp.442-450, 1990.

[Gu1995]

Gu, Z., Zhang, Y.F. and Nee, A.Y.C., “Generic form feature recognition and operation selection using connectionist modeling” Journal of Intelligent Manufacturing, Vol. 6, pp.263-273, 1995.

[Han1996]

Han, J.H. and Requicha, A.A.G., “Integration of feature based design and feature recognition” Computer-Aided Design, Vol. 29, pp.393-403, 1996.

[Helen2005]

Helen L. Locket and Marin D. Guenov, “Graph-based feature recognition for injection moulding based on a mid-surface approach” Computer-Aided Design, Vol.37, pp. 251-262, 2005.

[Henderson1994]

Henderson, M.R., Srinath, G., Stage, R., Walker, K. and Regli, W., “Boundary representation-based feature identification” in Advances in Feature Based Manufacturing, Ed. by J.J. Shah, M. Mäntylä and D.S. Nau, Elsevier Science, 1994.

[Hilbert1983]

Hilbert, D. and Cohn-Vossen, S., “Geometry and the imagination”, Translated by P Nementi, Chelsea, NY, 1983.

[Hoffman1989]

Hoffmann, C.M., “Geometric and solid modeling - an introduction”, Morgan Kaufmann Publishers, Inc. 1989.

[Hui1992]

Hui, K.C. and Tan, S.T., “Mould design with sweep operations - a heuristic search approach”, Computer-Aided Design, Vol.24, pp.81-91, 1992.

[Hui1996]

Hui, K.C., “Geometric aspects of the mouldability of parts”, Computer-Aided Design, Vol. 29, pp.197-208, 1996.

[Joshi1988]

Joshi, S and Chang, T.C., “Graph-based heuristics for recognition of machined features from a 3D solid model”, Computer-Aided Design, Vol.20, pp.58-66, 1988.

[Kato1992]

Kato K, “Generation of N-sided surface patches with holes” Computer-Aided Design, Vol.23, pp.676-683, 1992.

[Khardekar2006]

Khardekar, R., Burton, G. and McMains, S., “Finding feasible mold parting directions using graphics hardware” Computer-Aided Design, Vol. 38(4), pp.327–341, 2006.

[Kumar2002]

Kumar, M. and Gupta, S.K., “Automated design of multi-stage molds for manufacturing multi-material objects” Journal of Mechanical Design, Vol.124, no.3, pp. 399-407, 2002.

[Kumar2007]

Kumar, N., Ranjan R. and Tiwari M.K., "Recognition of undercut features and parting surface of moulded parts using polyhedron face adjacency graph" International Journal of Advanced Manufacturing Technology, Vol.34, pp. 47-55, 2007.

[Kwon1991]

Kwon, B.O. and Lee, K., "Automatic generation of core and cavity for injection mould design" Trans. of the Korean Society of Mechanical Engineers, Vol.15, pp.1225-1232, 1991.

[Laakko1993]

Laakko, T. and Mäntylä, M., "Feature modelling by incremental feature recognition", Computer-Aided Design, Vol.25, pp.479-492, 1993.

[Lee1997]

Lee, K.S., Li, Z., Fuh, J.Y.H., Zhang, Y.F. and Nee, A.Y.C., "Knowledge-based injection mould design system", The Proceedings of CIRP International Conference and Exhibition on Design and Production of Dies and Moulds, June, Turkey, pp.45-50, 1997.

[Les1995]

Les Piegl and Wayne Tiller, "The NURBS Book" ISBN 3-540-61545-8, 2nd Edition, Springer-Verlag Berlin Heidelberg New York, 1995.

[Li2003]

Li, X.J. and Satyandra K. Gupta, "Geometric algorithms for automated design of rotary-platen multi-shot molds" Department of Mechanical Engineering, Institute for Systems Research, University of Maryland, College Park, MD 20742, USA.

[Li2005]

Li C.L., Yu K.M. and Li C.G. "A new approach to parting surface design for plastic injection moulds using the subdivision method" International Journal of Production Research, Vol. 43, pp. 537-561, 2005.

[Lim1994]

Lim, J. T. ,Rho, H. M. and Cho, K. K., "A knowledge-based process planning for injection mold", Computers and Industrial Engineering, Vol. 27, Nos 1-4, pp. 95-98, 1994.

[Mäntylä1988]

Mäntylä, M., "An introduction to solid modeling", Computer Science Press, Rockville, Maryland, 1988.

[Marafat1990]

Marafat, M. and Kashyap, R.L., "Geometric reasoning for recognition of three-dimensional object features", IEEE Trans. Pattern Analysis & Machine Intell., Vol.12, pp.945-965, 1990.

[Mochizuki1992]

Mochizuki, T. and Yuhara, N., "Methods of extracting potential undercut and determining optimum withdrawal direction for mould designing" Int. J. Japan Soc. Prec. Eng., 26, pp.68-73, 1992.

[Nee1998]

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" Annals of CIRP; Vol.47, pp. 95-98, 1998.

[Pal2005]

Pralay Pal, Tigga, A.M. and Kumar, A., "Feature extraction from large CAD databases using genetic algorithm" Computer-Aided Design, Vol. 37(5), pp.545-558, 2005.

[Pla-Garcia2006]

Pla-Garcia, N., Vigo-Anglada, M. and Cotrina-Navau, J., "N-sided patches with B-Spline boundaries" Computers & Graphics, 30, pp.959–970, 2006.

[Preparata1977]

Preparata, F.P. and Hong, S.J., "Convex hulls of finite sets of points in two and three dimensions", Commun. ACM, 2, pp.87-93, 1977.

[Ravi1990]

Ravi, B. and Srinivasan, M.N., "Decision criteria for computer-aided parting surface design" Computer-Aided Design, Vol.22, pp. 11-18, 1990.

[Ravi Kumar2002]

Ravi Kumar, G.V.V., Shastry, K.G. and Prakash, B.G., "Computing non-self-intersecting offsets of NUBRS surfaces" Computer-Aided Design, Vol. 34, pp. 209-228, 2002.

[Ravi Kumar2003]

Ravi Kumar, G.V.V., Shastry, K.G. and Prakash, B.G., "Computing offsets of trimmed NUBRS surfaces" Computer-Aided Design, Vol. 35, pp. 411-420, 2003.

[Requicha1980]

Requicha, A.A.G., “Representation of rigid solid: theory, method and systems”, ACM. Comput. Surveys, Vol.12, pp.437-464, 1980.

[Roson1992]

Roson, D.W., “A feature-based representation to support the design process and the manufacturability evaluation of mechanical components”, PhD Thesis, University of Massachusetts, 1992.

[Rosen1994]

Rosen, D.W., “Towards automatic construction of moulds and dies” Computers in Engineering, ASME, Vol. 1, pp.317-326, 1994.

[Rubio2006]

Rubio Paramioa, M.A., Perez Garciab, J.M., Rios Chuecoc, J., Vizan Idoipeb, A. and Marquez Sevillano, J.J., “A procedure for plastic parts demoldability analysis” Robotics and Computer-Integrated Manufacturing, Vol. 22(1), pp.81-92, 2006.

[Sawai1994]

Sawai, S. and Kakazu, Y., “A study on automatic generation of the mould cavity and core geometry” Proceedings Japan-USA Symposium of Flexible Automation-A Pacific Rim Conference, Japan, pp.729-732, 1994.

[Shah1991]

Shah, J.J., “Assessment of features technology”, Computer-Aided Design, 23, pp.331-343, 1991.

[Shin1993]

Shin, K.H. and Lee, K., “Design of side cores of injection moulds from automatic detection of interference faces” Journal of Design and Manufacturing, Vol. 3, pp.225-236, 1993.

[SolidWorks2008]

SolidWorks 2008 and API help, Dassault Systèmes SolidWorks Corp., 2008.

[Tan1988]

Tan, S.T., Yuen, M.F., Sze, W.S. and Kwong, W.K., “A method for generation of parting surface for injection moulds”, Proc. of 4th Int. Conf. on Computer Aided Production Engineering, Edinburgh, UK, pp.401-408, 1988.

[Tan1990]

Tan, S.T., Yuen, M.F., Sze, W.S. and Kwong, W.K., “Parting lines and parting surfaces of injection moulded parts” Proc. Instn Mech. Engrs, 204, pp.201-219, 1990.

[Weinstein1997]

Weinstein, M. and Manoochehri, S., “Optimum parting line design of moulded and cast parts for manufacturability” Journal of Manufacturing Systems, Vol.16, pp.1-12, 1997.

[Wu1996]

Wu, M.C. and Liu, C.R., “Analysis on machined feature recognition techniques based on B-rep” Computer-Aided Design, Vol.28, pp.603-616, 1996.

[Ye2000]

Ye, X.G., “Feature and Associativity-Based Computer-Aided Design for Plastic Injection Moulds” PhD Thesis, National University of Singapore, 2000.

[Ye2001]

Ye, X.G., Fuh, J.Y.H. and Lee, K.S., "A hybrid method for recognition of undercut features from moulded parts" Computer-Aided Design, Vol.33, pp.1023-1034, 2001.

[Yin2004]

Yin, Z.P., Ding, H., Li, H.X. and Xiong, Y.L., "Geometric mouldability analysis by geometric reasoning and fuzzy decision making" Computer-Aided Design, Vol.36, pp.37-50, 2004.

[Yin2006]

Yin, Z.P., Ding, H., Li, H.X. and Xiong, Y.L., "Geometric reasoning on molding planning for multishot mold design" Journal of Computing and Information Science in Engineering, Vol.6, Issue 3, pp.241-252, 2006.

[Zhang1997]

Zhang, Y.F., Lee, K.S., Wang, Y., Fuh, J.Y.H. and Nee, A.Y.C., "Automatic side core creation for designing slider/lifter of injection moulds" The Proceedings of CIRP International Conference and Exhibition on Design and Production of Dies and Moulds, June, Turkey, pp.33-38, 1997.

The class definition of NURBS curve, surface and trimmed curve loop in Visual C++ in this thesis:

```
Class CNURBSCurve
{
      long   order,          //order of a curve
             period,         //period of a curve
             dim,            //dimension of a curve
             nb_ctrpts,      //number of control points
             nb_knots,       //number of knots
             deg;            // degree of curve

      double *KnotValue;   //knots of a curve
      double *CtrlPoint;     //control point of a curve
}

Class CNURBSSurface
{
      long   dim,            //dimension of a surface
             u_order,        //order in U direction of a surface
             v_order,        //order in V direction of a surface
             u_nb_ctrl,      //number of control points in U
             v_nb_ctrl,      //number of control points in V
             u_period,       //period in U direction
             v_period,       //period in V direction
             u_nb_knots,     //number of knots in U direction
             v_nb_knots;     //number of knots in V direction

      double  *knotValuesU;          //knot values at U direction
      double  *knotValuesV;          //knot values at V direction
      double  *ctrlPts;              //control points of a surface

      int            num_TrimLoop  //number of trimmed loop of a surface
      CTrimLoop      *pTrimLoops      //all trimmed loops
}

Class CTrimLoop
{
      int                   nbCurves            //number of curves in a trimmed loop
      CNURBSCurve     *pNURBSCurve  //all trimmed NURBS curves
}
```

PUBLICATIONS ARISING FROM THE RESEARCH

Z. Zhao, J.Y.H. Fuh and A.Y.C. Nee, “A hybrid parting method based on iterative surface growth algorithm and geometric mouldability”, Computer-Aided Design and Application, Vol. 4, No.6, pp.783-794, 2007.