

國立臺灣大學工學院機械工程學研究所

碩士論文

Graduate Institute of Mechanical Engineering

College of Engineering

National Taiwan University

Master Thesis

三角網格特徵線萃取與分割於汽車鈹件之應用
Application of Feature Line Extraction and Triangular
Mesh Segmentation for Automobile Sheet Metals

李康豪

Kang-Hao Lee

指導教授：尤春風 博士

Advisor: Chun-Fong You, Ph.D.

中華民國 99 年 7 月

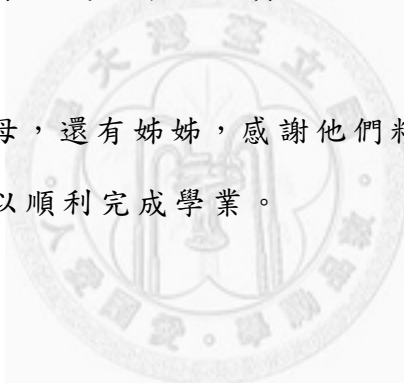
July, 2010

誌謝

感謝指導教授尤春風博士於碩士班的兩年期間耐心地指導我研究的方法與方向，論文寫作期間提供許多的寶貴建議，使本論文能夠順利的完成，並感謝口試委員莊勝雄與陳俊銘對本論文提供寶貴的意見與指正。

另外感謝兩年來實體模型實驗室的學長姊甦農、易龍、亦蓮、振揚、正富、孜佑在電腦相關及程式方面的支援；感謝友瑜、育銘、昆育同學在課業與研究上的互相切磋；也要感謝學弟妹又璇、孟真、建成、大堃、順源、淑婷使我快樂度過實驗室生活。

最後感謝我的父母，還有姊姊，感謝他們精神上的支持，還有生活上的照顧，使我可以順利完成學業。



三角網格特徵線萃取與分割於汽車鈹件之應用

摘要

逆向工程是從現有的實物中或樣品模型中，透過三次元量測儀器取得點資料，再由點資料將曲面重建。在逆向曲面重建裡，曲面重建無法直接處理龐大的網格資料，必須將網格資料以區域來處理。本研究針對大量三角網格資料提出一套處理網格分割與分群之程序，不僅可快速有效的縮短曲面重建時程，也提升曲面與原始網格的精確度。

網格分割與分群程序中所用的演算法包含三個部分：特徵線萃取、網格面分割以及網格分群。首先，特徵線萃取是以曲率變化及透過門檻值的過濾產生特徵點，再將所產生的點連接成為特徵線，並進行特徵線延伸取得封閉網格塊，同時利用網格拓撲關聯性進行三角網格面分割，最後將封閉網格塊進行網格分群，取得完整區塊。

關鍵字：逆向工程、三角網格、特徵線萃取、網格面分割、網格分區域。

Application of Feature Extraction and Triangular Mesh Segmentation for Automobile Sheet Metal

Kang-Hao Lee

Abstract

Reverse engineering is a process of reconstructing CAD models based on an existing object. With the help of a coordinate measuring machine (CMM), we can model the data points into triangular meshes, then reconstruct the surface models from the triangular meshes. In surface reconstruction, it is impossible to construct a surface with a huge number of data points; therefore, we must segment the meshes into smaller regions. In this paper, a triangular mesh segmentation procedure is proposed. The procedure not only efficiently shortens the surface reconstruction, but also improves the accuracy of the reconstructed model to the existing object.

The basic approach consists of three steps: feature line extraction, mesh segmentation, and finally mesh separation. Firstly, feature line extraction is based on the changes of principal curvatures and filter through a threshold value to generate feature vertices. Next, join the feature vertices to form feature lines and extend them to form closed loops. In the meantime, using the topological information of triangular mesh, we can start mesh segmentation. Lastly, with the help of the closed loops, we can decompose the mesh models into smaller regions.

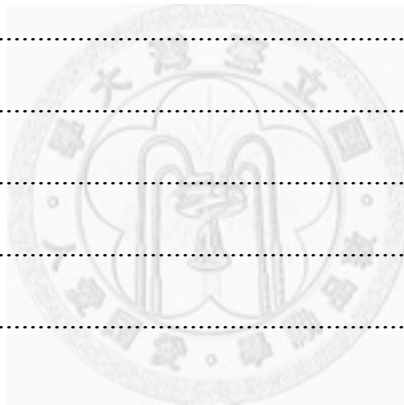
Keywords: reverse engineering, triangular mesh, feature line extraction, mesh segmentation, mesh separation.

目錄

誌謝.....	i
摘要.....	ii
Abstract.....	iii
目錄.....	iv
圖目錄.....	vii
表目錄.....	x
第一章 緒論.....	1
1.1 前言.....	1
1.2 文獻回顧.....	2
1.2.1 特徵線萃取.....	3
1.2.2 三角網格分割與分群.....	4
1.3 研究內容與方法.....	5
1.4 系統開發環境.....	7
1.5 論文整體架構.....	7
第二章 特徵線萃取與處理.....	9
2.1 前言.....	9
2.2 SOD 與 ESOD 演算法.....	9
2.2.1 SOD 演算法.....	10
2.2.2 ESOD 演算法.....	11
2.2.3 SOD 與 ESOD 的比較.....	11
2.3 Geomagic Studio 逆向工程軟體.....	12
2.3.1 Geomagic Studio 簡介.....	13
2.3.2 Geomagic Studio 之特徵萃取功能.....	13
2.4 RidgeAndRavine 演算法.....	15

2.4.1 網格平順化與法向量重新取值.....	16
2.4.2 網格主軸曲率、主軸方向以及一次微分.....	16
2.4.3 特徵線建構.....	19
2.4.4 特徵線後處理.....	23
2.5 特徵萃取演算法及逆向工程軟體的成果比較.....	26
第三章 三角網格分割與網格面分群.....	28
3.1 前言.....	28
3.2 拓樸資料的建立.....	29
3.2.1 點資料的建立.....	29
3.2.2 邊資料的建立.....	30
3.2.3 面資料的建立.....	30
3.3 特徵線延伸.....	31
3.3.1 特徵線延伸流程說明.....	32
3.3.2 曲線上取點.....	34
3.4 三角網格分割.....	35
3.4.1 尋找分割三角網格中的封閉迴路.....	36
3.4.2 處理與分割特徵面之封閉迴路.....	37
3.5 三角網格面之錯誤法向量.....	42
3.6 網格區域分群.....	44
3.6.1 暫存網格面容器.....	44
3.6.2 十字相交法.....	45
3.6.3 網格分群流程.....	46
3.6.4 建立完整區塊.....	47
第四章 實例驗證.....	50
4.1 前言.....	50
4.2 檔案讀取.....	51

4.3 特徵線萃取與特徵線延伸.....	52
4.3.1 設定門檻值與特徵線萃取.....	52
4.3.2 點選特徵線及網格邊界點之特徵線延伸.....	53
4.4 整體網格邊界建立.....	55
4.5 網格面分割與網格面分區域.....	57
4.6 區域成長與特徵線萃取分群比較.....	58
4.7 網格點資料重建 B-spline 曲面.....	60
4.8 雜線的串聯.....	63
4.9 門檻值的判斷.....	65
第五章 結論與未來展望.....	67
5.1 前言.....	67
5.2 研究結論.....	67
5.3 未來展望.....	68
參考文獻.....	70
作者簡歷.....	72



圖目錄

圖 1.1 逆向工程流程圖.....	2
圖 1.2 特徵辨識分割流程圖.....	6
圖 2.1 SOD 演算法(引用[3]).....	10
圖 2.2 ESOD 演算法(引用[3]).....	11
圖 2.3 SOD 與 ESOD 特徵萃取結果比較示意圖.....	12
圖 2.4 Geomagic Studio 中 Detect Contours 的功能與設定參數之介紹.....	14
圖 2.5 引擎蓋外鈑使用 Geomagic Studio 中的 detect contours 之測試結果.....	14
圖 2.6 特徵線演算法流程圖.....	15
圖 2.7 極限係數示意圖.....	19
圖 2.8 特徵線建構之流程圖.....	20
圖 2.9 特徵曲率敏感值比較示意圖.....	21
圖 2.10 連接特徵點示意圖(引用[11]).....	23
圖 2.11 特徵線後處理流程圖.....	24
圖 2.12 線段後處理之示意圖.....	25
圖 2.13 原始特徵線與近似曲線 (經過最小平方法處理之特徵線).....	25
圖 3.1 三角網格分割與網格面分群流程圖.....	28
圖 3.2 半邊結構示意圖.....	29
圖 3.3 特徵線延伸之流程圖.....	32
圖 3.4 特徵線延伸.....	33
圖 3.5 曲線上取點之範例.....	34
圖 3.6 曲線上取點之示意圖.....	35
圖 3.7 尋找分割三角網格中的封閉迴路之流程圖.....	37
圖 3.8 三種特徵線分割情況.....	38
圖 3.9 特徵網格分例探討.....	39

圖 3.10 分割十邊之結果.....	40
圖 3.11 分割七邊之結果.....	40
圖 3.12 分割六邊之結果.....	41
圖 3.13 分割五邊之結果.....	41
圖 3.14 分割四邊之結果.....	42
圖 3.15 法向量不一致.....	43
圖 3.16 法向量不一致處理.....	44
圖 3.17 暫存網格面資料容器.....	45
圖 3.18 十字相交之圖.....	46
圖 3.19 網格分群示意圖.....	47
圖 3.20 設定網格區塊邊界編號之流程圖.....	48
圖 3.21 建立完整區塊之圖.....	49
圖 4.1 完整介面.....	50
圖 4.2 讀檔介面.....	51
圖 4.3 測試檔案網格資料.....	52
圖 4.4 (預設)引擎蓋外鈑及葉子鈑特徵線萃取.....	53
圖 4.5 (調整後)引擎蓋外鈑及葉子鈑特徵線萃取.....	53
圖 4.6 特徵線延伸之介面.....	54
圖 4.7 特徵線延伸.....	54
圖 4.8 曲線上取點之介面.....	55
圖 4.9 整體的網格邊界.....	56
圖 4.10 手動內部邊界之介面.....	56
圖 4.11 手動點選邊界邊之介面.....	56
圖 4.12 手動網格塊邊界之介面.....	57
圖 4.13 儲存後以 CATIA 開檔實例.....	58
圖 4.14 區域成長分群之結果.....	59

圖 4.15 二次分割前之引擎蓋外鈹的網格與裁切曲面誤差比對.....	61
圖 4.16 二次分割後之引擎蓋外鈹的網格與裁切曲面誤差比對.....	61
圖 4.17 二次分割前之葉子鈹的網格與裁切曲面誤差比對.....	62
圖 4.18 二次分割後之葉子鈹的網格與裁切曲面誤差比對.....	63
圖 4.19 雜線的串聯示意圖.....	65



表目錄

表 1.1 開發平台列表.....	7
表 2.1 特徵萃取演算法與逆向工程軟體的成果比較.....	27
表 3.1 邊結構對應表.....	30
表 4.1 引擎蓋外鈹與葉子鈹的各網格面的邊數量.....	57



第一章 緒論

1.1 前言

隨著科技不斷地創新，電腦軟硬體的持續發展，電腦輔助設計(CAD)及電腦輔助製造(CAM)被廣泛的應用在產品的設計製程上。由於業界三次元掃描系統的開發與進步，帶動逆向工程(RE)以及快速原型(RP)的發展，此一系的技術不但縮短產品開發與製造時間，同時也降低產品生產成本。

一般產品開發過程可分為兩類:正向工程、逆向工程。正向工程又稱為傳統工程，是指在設計產品，憑著產品需求的構思、功能與規格，經由設計、製造、反覆的檢驗等開發流程生產產品;逆向工程是以專案方式進行成品或樣品的仿製。一般逆向工程的處理方式分為兩類:傳統逆向工程與數位化逆向工程。傳統逆向工程是藉由立體雕刻機、仿削機之類的器材進行類比仿製，雖然能依照樣品原始外型仿製出成品，但是無法建立工件尺寸圖檔、無法保存資料、無法進行樣品外型或細部的修改，種種缺失會造成設計者在處理相關產品時很大的限制;而數位化逆向工程可以針對樣品，利用 3D 掃描量測儀器，如非接觸式光學掃描量測系統(ATOS, advanced topometric sensor)，把樣品外型掃描並以數位化方式儲存，再藉由此數位化檔案進行實體或曲面的繪製、編輯與修改，或是藉由此檔案重複修改直到符合需求。最後再將此檔案送到快速成型機製作硬體模型，如圖 1.1 顯示逆向工程作業流程。

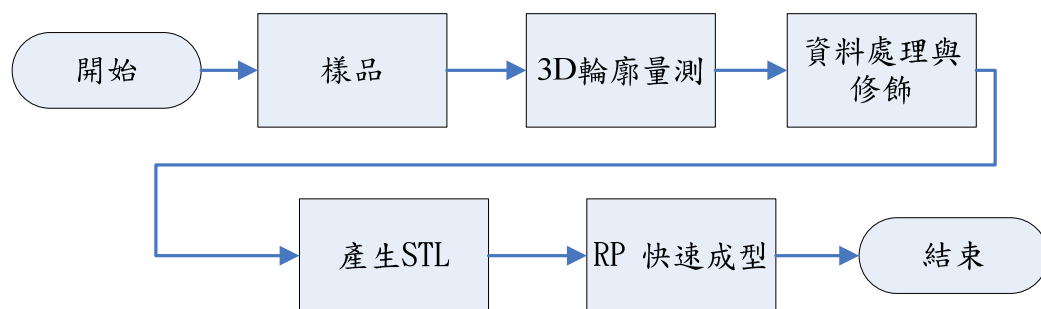


圖 1.1 逆向工程流程圖

數位化逆向工程所儲存的數位化資料都以三角網格作為儲存媒介，一般常見的三角網格有 STL(stereo lithography)、TRI(triangle)、OBJ(object)等不同的檔案格式。由於本研究採用 STL 作為網格輸入格式，所以本文只針對 STL 檔案格式略作說明。STL 是一個由三角網格面組合而成的檔案，每一個三角面是以一個單位法向量(normal vector)及三個頂點(vertices)座標來表示。STL 檔案格式是一個簡單明瞭的資料格式，產生檔案或讀取數據相當容易，但是因為包含的資訊較少，所以資料的重複性相當高，相形之下會增加檔案容量。所以根據 Szilvasi-Nagy[8]等人所提出的 STL 檔案設定滿足以下條件：

- (1) 每個邊最多只能被兩個三角網格面使用。
- (2) 每個三角網格面至少有一個點與其他三角網格面相連。
- (3) 不能有自交三角網格面出現。
- (4) 網格面法向量方向必須滿足右手定則。

1.2 文獻回顧

本研究主要目的在於尋找特徵線、網格面分割與區域劃分，因此參考文獻將分為兩大類：第一類為特徵線萃取、第二類為網格分割與分區

域。以下將針對這兩大類做相關研究介紹。

1.2.1 特徵線萃取

Yoshizawa[11]等人將網格特徵性質以網格點的形式呈現，計算每個網格點的最大與最小主軸曲率 (principal curvature)、最大與最小主方向 (principal direction)。再利用 Maxima/Minima Test 在原始網格中尋找並建構特徵線，最後再進行特徵線過濾，刪除較小線段或合併線段來進行特徵線萃取。

Hubeli[3]等人將網格邊作為特徵線萃取對象，首先對每條網格邊進行權重計算，透過上下門檻值的過濾，判斷該邊是否為特徵。接著將特徵邊歸類成不同特徵區域，並對每塊特徵區域進行細化，成為一條特徵線，細化是利用 skeletonizing 演算法則達成。

Rossel [7]等人所提出的演算法跟 Hubeli[3]等人的演算法很類似，唯一的相異處在於 Rossel 以網格點作為特徵點萃取對象，利用 discrete curvature analysis 來計算權重並設定門檻值進行特徵線萃取。

劉勝藍[14]等人對三角網格模型進行兩次特徵點辨識，第一次為簡單計算網格面的曲率變化，第二次則是計算各頂點的法向量。然後計算曲面頂點的主軸曲率與主軸方向，在主軸方向上主軸曲率為極值的點即特徵點，將找到的點進行特徵點連接成為特徵線。

由以上的文獻可以了解，特徵線萃取通常是以曲率變化作為判斷標準，再透

過門檻值的過濾產生特徵點，再將所產生的點連成為特徵線，以利後續應用。不同的判斷的方法會影響特徵萃取，如細線化。對每條網格邊進行權重計算，透過雙門檻值過濾，將特徵邊歸類成不同特徵區域，並對每塊特徵區域進行細線化，成為特徵線。

1.2.2 三角網格分割與分群

Razdan[6]等人將特徵邊兩端點定義為特徵點(feature vertex)，並根據每個三角網格內的特徵點與特徵邊的數目，新增點於邊上產生新的三角網格，由新增後的資料以幾何資訊分離原始三角網格。

Kim[4]等人將網格資料分為規則點資料(regular data)與雜訊點資料(noisy data)，利用每一個網格與相鄰網格的夾角作為規則點區域分離的依據，利用網格邊的曲率計算作為雜訊點區域分離依據。

Gelfand[1]等人提出以每點周圍 N 個點作為初始區域，分別計算每塊區域的 Slippable Motion(6x6 矩陣)，利用此矩陣的特徵值與特徵向量，評估相鄰的區域是否合併為一塊，若能合併，需要重新計算矩陣，直到無相鄰區域能合併為止。

曾凱威[14]利用網格點的最大主軸曲率，進行網格面區域成長。區域成長的主要目的在於對特徵面與平緩面進行相鄰面同性質分群動作，概念主要是不斷的運算，使得網格面能與其它相鄰面進行特徵屬性的比較，並透過外擴搜尋，將所有一樣的特徵屬性分群為網格塊，並且達到初步網格分群效果。

在三角網格分割方面，各種分割的形式都有其基本定義，利用網格間的關聯性分割原始網格；在網格分群方面，大致上先達到初步的網格分群效果，依據不同的需求進行修補，如網格面區域成長，先取得初步網格塊，再透過一些特定方法取得完整網格塊，達到網格分群的目的。

1.3 研究內容與方法

本論文主要的研究目的，在於利用特徵線萃取方法在現有網格資料中，透過特定的演算法則，將整筆網格資料依照多條封閉的特徵線包圍成小區域，最後進行曲面重建。圖 1.2 為特徵萃取分割流程圖，而整體的研究方法，可分為下列三大部分：

(1) 特徵線萃取

文獻中已有許多特徵線萃取之方法，本研究套用 Yoshizawa 方式自動萃取特徵線。此方式用意是萃取網格資料上明顯的稜線，作為網格面分割的依據。擷取特徵線可以預先定義各區域之間的邊界資訊，待所有的邊界(特徵線)確立後，即可建立邊界曲線，並利用此邊界進行分割，最後將各個分割的網格面區域重建成曲面。

(2) 特徵線延伸與網格面分割

由於以上特徵線萃取的特徵線並不完整，無法進行網格區域劃分，所以必需將特徵線延伸至整體網格的邊緣，才能使網格有分割的依據。當特徵線段都確定後，必須檢查是否所有網格皆

為三個邊，若不為三邊，則必須進行網格面分割。

(3) 網格分區域

在完成特徵線延伸與網格分割後，可得到封閉區域以及網格位於封閉區域內部或外部的判斷；多條封閉特徵線所包圍的區域可以進行後續曲面重建。

本論文研究範圍限制的部分，所讀取的 STL 檔案必須符合 STL 檔案建立的基本規則；在特徵線萃取僅著重於一般汽車鈑件，如引擎蓋外鈑、葉子鈑，無法使用於 3D 動物模型或肖像模型等封閉模型。

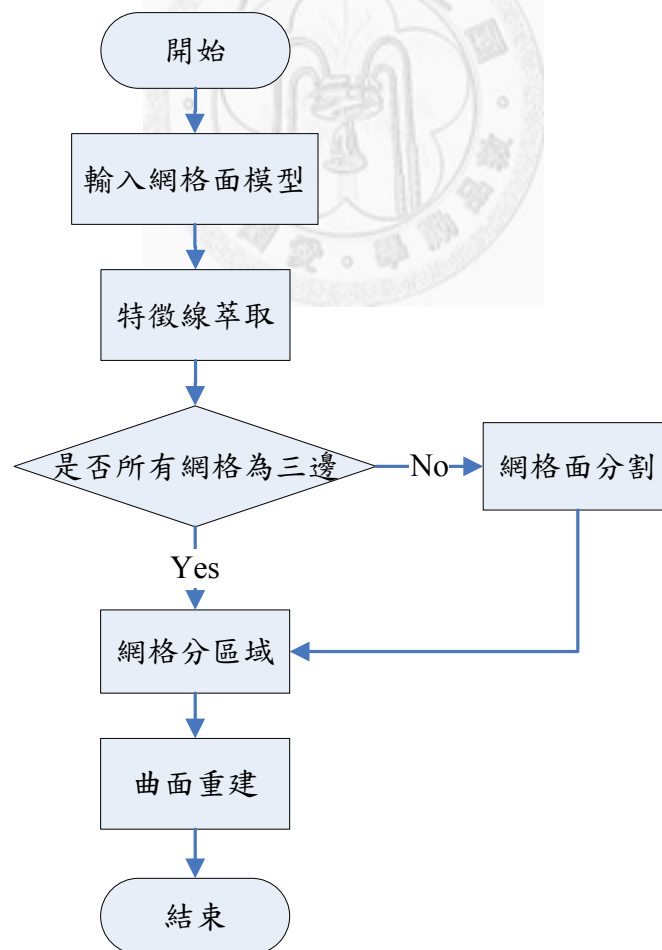


圖 1.2 特徵辨識分割流程圖

1.4 系統開發環境

本研究所用到 CAD 系統運算核心是由尤春風[12]的 Spring Solid System 所提供，Spring Solid System 是一個以純 Java 語言撰寫，可跨平台運作的 CAD 核心。表 1.1 為本研究開發平台列表。

表 1.1 開發平台列表

硬體規格	Intel® Core(TM) 2 Quad CPU Q9550 @ 2.83GHz, 8GB of RAM
作業系統	Microsoft Windows XP Professional x64 Edition Version 2003 SP2
開發環境	Java 2 Platform Standard Edition 1.5.0.11 64bit Spring Solid CAD System
開發平台	Eclipse SDK 64bit Version 3.4.0

1.5 論文整體架構

本文共分五個章節，內容架構如下：

第一章：為緒論。介紹本研究相關背景、文獻之概略介紹、研究內容與方法。

第二章：為特徵線萃取與處理。針對本論文所選用的特徵線萃取演算作討論，並與其他市面上逆向工程軟體或演算法做成果比較。

第三章：為三角網格分割與網格面分群。主要解釋特徵線延伸與

網格分割的演算法，最後建立出多條封閉的特徵線所包圍的區域，進行曲面重建。

第四章：為實例驗證。透過實際測試鈹件廠商所提供的汽車鈹件網格資料，驗證本研究網格分割與區域劃分結果。

第五章：為結論與未來展望。歸納研究結果並提出未來特徵線萃取及網格面分割的看法，以利後續研究者之參考。



第二章 特徵線萃取與處理

2.1 前言

在大多數的特徵線處理方法裡，特徵線之建構通常是以主軸曲率的變化產生特徵點為依據，再連接所有特徵點即成為特徵線。特徵萃取的全自動作業至今仍是一件困難的事，由於每個網格資料差異性太大，自動判斷特徵區域乃具有挑戰性，因此以半自動特徵萃取方法，由使用者手動輸入門檻值，作為本研究的主要方向。

在逆向曲面重建過程中，特徵線的萃取為重要步驟之一，因為曲面重建無法直接處理龐大的網格資料，必須將網格資料以區域性來處理，而特徵線萃取可以使網格分區域有所依據，根據特徵線的性質可將龐大網格資料簡化為數個小網格區域，在後續處理上將會比較可行及迅速。

特徵線萃取與處理主要應用在曲面重建，讓網格的後續處理能夠迅速與方便。特徵辨識主要是利用半自動化的方式，如手動設定門檻值，利用特徵權重超過門檻值與否判別特徵點。以下小節，將分別針對三種特徵萃取演算法及一個逆向工程軟體進行說明與比較。

2.2 SOD 與 ESOD 演算法

於眾多特徵萃取演算法中，本研究的第一種特徵萃取演算法為 Second Order Difference(SOD)；第二種為 Extended Second Order Difference (ESOD)，因為所有的特徵萃取演算法中，這兩種方法較為

簡潔且容易操作。在整個演算過程中，只需用一個公式以及使用者所定的雙門檻值，就能萃取出特徵線段。

設定雙門檻值的好處不僅可以模擬單門檻值的情況，只要設定上下限為同一個值，就可以達到單門檻值的效果，而且還能夠加強特徵連續性以及有較好的特徵群聚效果。

2.2.1 SOD 演算法

SOD 的演算法是將每個邊所接觸的雙網格面法向量作角度計算，並設定為此邊的權重值，如式(2.1)為權重值計算式。

$$w(e) = \cos^{-1} \left(\frac{n_i \cdot n_j}{\|n_i\| \|n_j\|} \right) \times \frac{180}{\pi} \quad (2.1)$$

其中 $w(e)$ 為此邊的權重值其單位為徑度， e 為網格邊， n 為網格面之法向量。當雙網格面的角度越大，權重值也越大，這代表此邊成為特徵的可能性也越大。圖 2.1 為 SOD 演算法之示意圖。

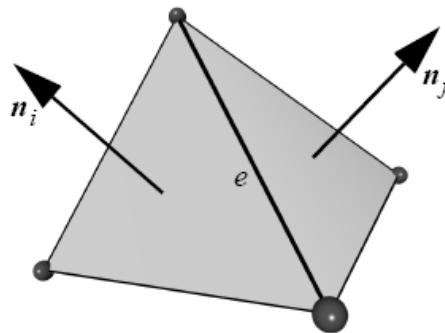


圖 2.1 SOD 演算法(引用[3])

2.2.2 ESOD 演算法

ESOD 為 SOD 的延伸，其運算方式與 SOD 類似，也是利用式 (2.1) 來計算每個邊的權重值，但唯一不同之處在於法向量的計算方法。法向量的計算主要是透過統計網格點(P)周圍網格法向量的平均達成，如圖 2.2，若網格有疏密不均的情況，可以利用面積做為權重，計算式如下：

$$N_p = \frac{\sum_{i=1}^n W_i N_i}{\left\| \sum_{i=1}^n W_i N_i \right\|}, \quad W_i = \frac{A_i}{\sum_{i=1}^n A_i} \quad (2.2)$$

其中 N 為點 P 周圍的網格數目， W_i 為第 i 個網格面積 A_i 與網格點 P 周圍一圈的面積的比值， N_i 為第 i 個網格法向量。

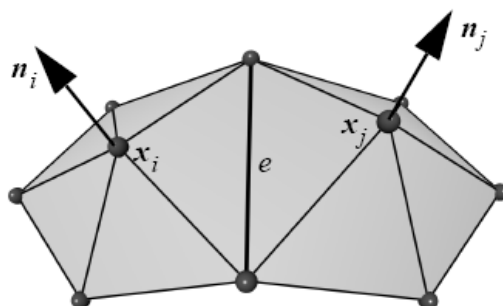


圖 2.2 ESOD 演算法(引用[3])

2.2.3 SOD 與 ESOD 的比較

在同樣的門檻值下，ESOD 所萃取的特徵比 SOD 所萃取的特徵多了許多，這代表 ESOD 對特徵邊的容許範圍較為敏感，對於不同的網格資料，SOD 與 ESOD 有著不同的優缺點。對於網格比較不精細的模型，使用 ESOD 是一個錯誤的選擇，因為 ESOD 對特徵門檻值過於敏

感，會造成一整塊區域成為特徵，因此選擇使用 SOD 較為恰當。對精細模型的網格資料，使用 ESOD 是比較好的方法，因為它能展現對網格的敏感，而產生較完整的特徵。圖 2.3 分別利用 SOD 與 ESOD 對引擎蓋外鈑萃取特徵。藉由不同的門檻值，獲取最佳狀況的結果，再來比較兩種演算法。圖 2.3(a)是使用 SOD，門檻值上限為 2.0 與門檻值下限為 1.0 的情況；圖 2.3(b)是使用 ESOD，門檻值上限為 6.0 與門檻值下限為 5.0 的情況。各演算法最好的結果中，可看出 ESOD 所產生的特徵比 SOD 多了許多，這證明 ESOD 演算法對於特徵辨識較為敏感，特徵發生處較為完整。

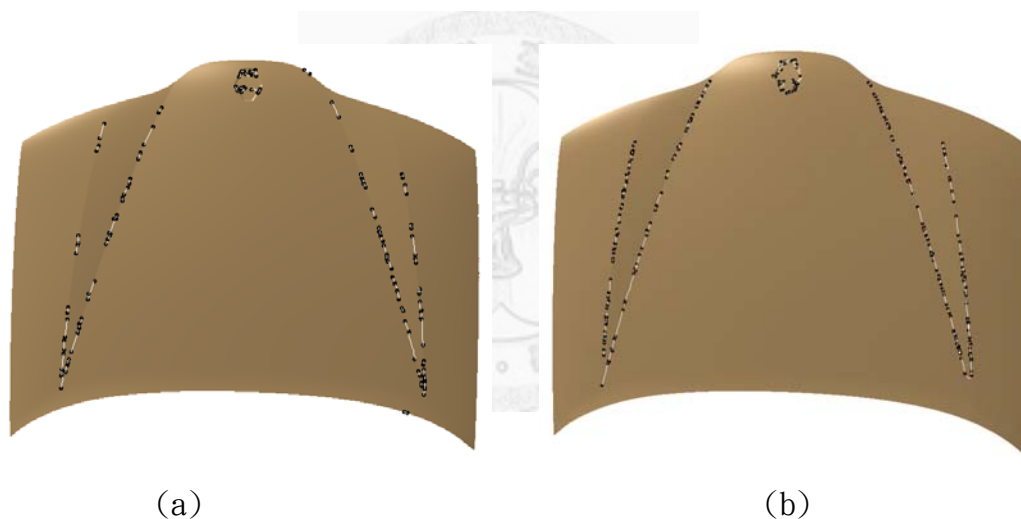


圖 2.3 SOD 與 ESOD 特徵萃取結果比較示意圖

2.3 Geomagic Studio 逆向工程軟體

在逆向工程的領域中，有不少的逆向軟體已被開發出來，如 Imageware、PolyWorks、Geomagic Studio、RevCAD、CopyCAD 等，本研究使用 Geomagic Studio 逆向軟體其中的特徵萃取功能(detect contours)，以下做進一步說明。

2.3.1 Geomagic Studio 簡介

Geomagic Studio 是由美國 RainDrop 出品的逆向工程和 3D 檢測軟體。此軟體可輕易地從掃瞄得到的雲點資料建立完美的多邊形模型和網格，並可自動轉換為 NURBS 曲面。Geomagic Studio 主要包括 Qualify、Shape、Wrap、Decimate、Capture 等五個模塊。主要特色包括：

- (1) 自動將雲點資料轉換為多邊形(polygons)
- (2) 快速減少多邊形數目(decimate)
- (3) 把多邊形轉換為 NURBS 曲面
- (4) 曲面分析(公差分析等)
- (5) 輸出檔案格式(IGS、STL、DXF 等)

2.3.2 Geomagic Studio 之特徵萃取功能

如圖 2.4 所示為本文所使用的 Geomagic Studio 軟體 12.0.0 版的試用版。Geomagic Studio 的特徵萃取是屬於 Shape 模組裡的一個功能，利用半自動化的方式進行辨識，藉由使用者自行輸入三個參數(曲率值、區分線段、最小面積)進行特徵萃取。此特徵萃取是利用區域成長演算法搭配網格塊細線化來取得特徵線段。當特徵都萃取完畢時，若有特徵必須進行修改，此功能提供一套完善的編輯工具來修改特徵，此外亦可設定特徵線的長度，若長度不符合所設定的值，則進行特徵線的後處理，合併較短線段。圖 2.5 為 Geomagic Studio 對引擎蓋外鈹

的特徵線辨識，參數設定為：曲率值為 70.0、區分線段值為 60.0、最小面積為 827.1mm²、最短長度為 11.368mm。

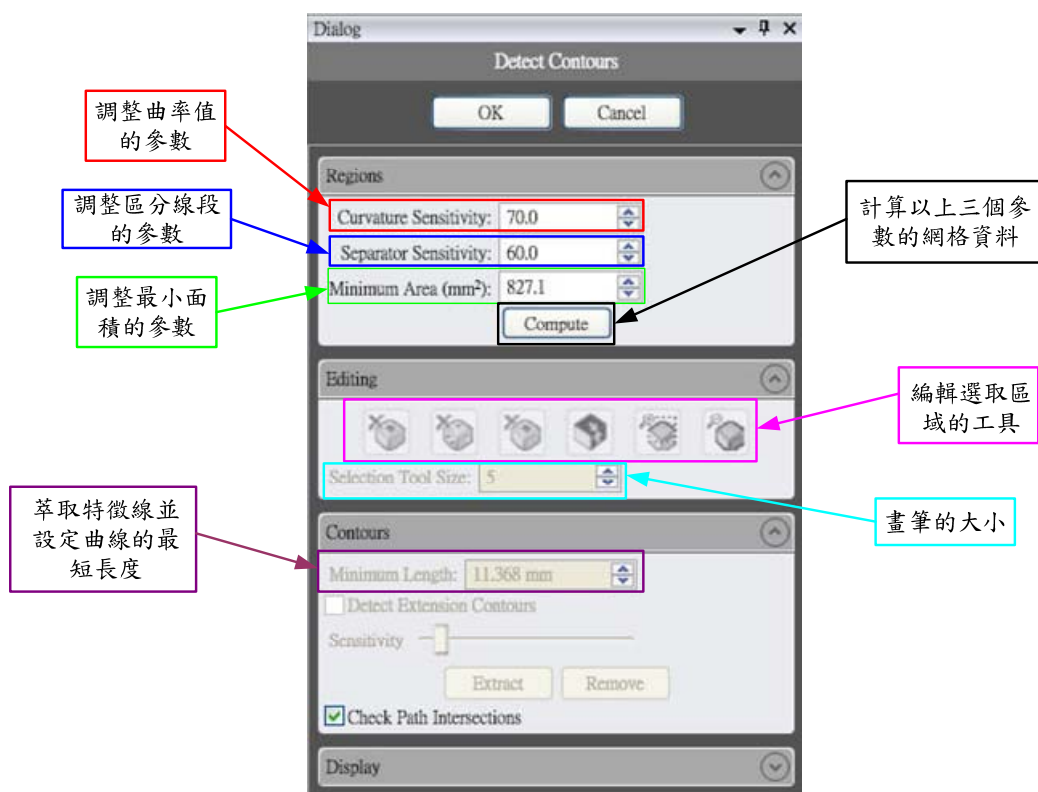


圖 2.4 Geomagic Studio 中 Detect Contours 的功能與設定參數之介紹

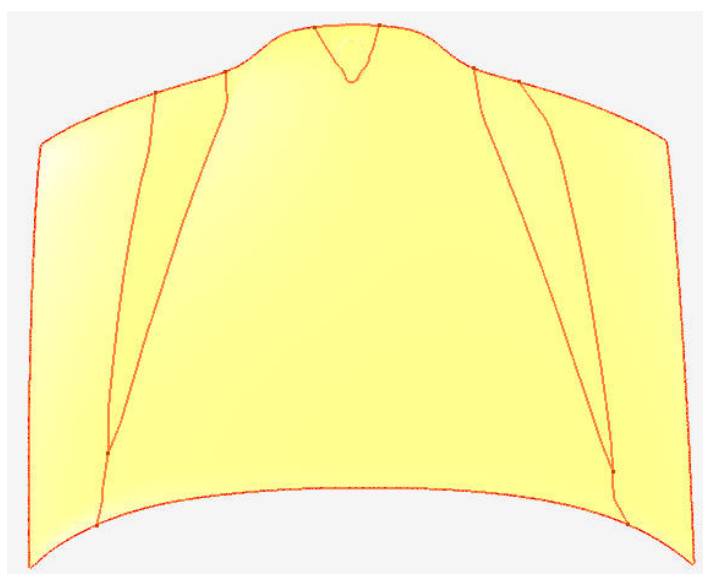


圖 2.5 引擎蓋外鈑使用 Geomagic Studio 中的 detect contours 之測試結果

2.4 RidgeAndRavine 演算法

本章節將依據 Yoshizawa[11]等人所提出的特徵線萃取理論，並修正特徵線後處理，定出一套適用於汽車鈹件的演算法。圖 2.6 為此特徵線演算法流程圖，此流程中包含四個主要部份：

- (1) 針對每個網格點做平順化以及計算法向量。
- (2) 計算每個網格點的主軸曲率(κ_{max} , κ_{min}) 跟主軸方向(t_{max} , t_{min}) 以及第一次微分。
- (3) 將特徵線分為凹(ravine)與凸(ridge)，並在原始網格中分別尋找並建構凸特徵線及凹特徵線。
- (4) 特徵線後處理。

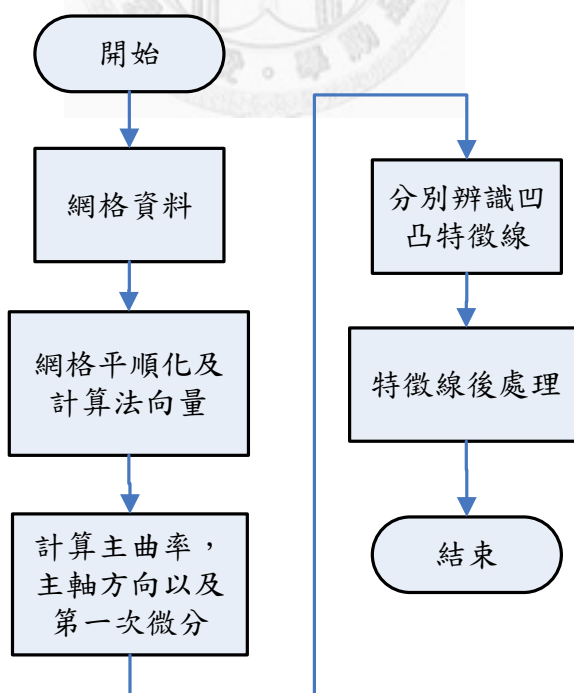


圖 2.6 特徵線演算法流程圖

2.4.1 網格平順化與法向量重新取值

由於在特徵線的方法中會利用網格點的微分，因此為了使特徵線更容易辨識，在輸入網格資料後，會進行局部平順化的處理，以確保每個網格點的微分可以被求得，並提高特徵線辨識的成功率。對於網格面中的每一個網格點 P ，找出與 P 所有相鄰的三角網格，並利用這些三角網格的質心位置計算新的網格點 P' ，以 P' 來取代 P 的位置。當所有的 P' 都計算完成，再利用 P' 點建立平順化的網格模型。

建立好新網格模型後，必須重新計算網格點的法向量。此計算可參考 2.2.2 小節的 ESOD 的法向量計算。

2.4.2 網格主軸曲率、主軸方向以及一次微分

在 3D 空間中，曲面 S 上的每一點 P ，可算出一個單位法向量(unit normal vector)。在點 P 的一個法平面包含該法向量以及與曲面相切的一個方向平面，在曲面上割出一條平面曲線，而點 P 在不同的法平面上有不同曲率，這些曲率值中極大和極小值被稱作主軸曲率值，其對應的曲率方向也被稱作主軸方向，主軸方向與法向量互相垂直，可以當做空間上的一組基底。主軸曲率及主軸方向的計算，可藉由 Goldfeather[2]所提出的 Weingarten 曲率矩陣及 Adjacent Normal Cubic Approximation 法則來計算，其定義如下：

$$W = \begin{bmatrix} L & M \\ M & N \end{bmatrix}^{-1} \begin{bmatrix} e & f \\ f & g \end{bmatrix} = \begin{bmatrix} \frac{eN - fM}{LN - M^2} & \frac{fN - gM}{LN - M^2} \\ \frac{fL - eM}{LN - M^2} & \frac{gL - fM}{LN - M^2} \end{bmatrix} = \frac{1}{LN - M^2} \begin{bmatrix} eN - fM & fN - gM \\ fL - eM & gL - fM \end{bmatrix}$$

其中 $e = N_p \cdot \frac{\partial^2 S}{\partial x^2}$ 、 $f = N_p \cdot \frac{\partial^2 S}{\partial x \partial y}$ 、 $g = N_p \cdot \frac{\partial^2 S}{\partial y^2}$ 、 $L = \frac{\partial S}{\partial x} \cdot \frac{\partial S}{\partial x}$ 、 $M = \frac{\partial S}{\partial x} \cdot \frac{\partial S}{\partial y}$ 、 $N = \frac{\partial S}{\partial y} \cdot \frac{\partial S}{\partial y}$ 、 N_p 為網格點的單位法向量

$$N_p = \left(\frac{\partial S}{\partial x}, \frac{\partial S}{\partial y}, -1 \right) \quad (2.3)$$

在 Weingarten 矩陣中， L 、 M 、 N 為第一基本形式的參數，代表曲面的度量性質； e 、 f 、 g 為第二基本形式，代表曲面的彎曲性質，詳細說明於 [10]。

當 $\frac{\partial S}{\partial x}$ 與 $\frac{\partial S}{\partial y}$ 是正交單位向量時，Weingarten 曲率矩陣可以簡化為

$$W = \begin{bmatrix} e & f \\ f & g \end{bmatrix} \quad (2.4)$$

在利用曲面方程式呈現相鄰網格點資料，其曲面方程式為 $S = f(x, y) = \frac{A}{2}x^2 + Bxy + \frac{C}{2}y^2 + Dx^3 + Ex^2y + Fxy^2 + Gy^3$ 。然後將曲面方程式代入式 (2.4)，可得到：

$$W = \begin{bmatrix} A & B \\ B & C \end{bmatrix} \quad (2.5)$$

該法則只取用 A 、 B 、 C 三參數。接著將曲面方程式代入式 (2.3)，可得到：

$$N_p = (Ax + By + 3Dx^2 + 2Exy + Fy^2, Bx + Cy + Ex^2 + 2Fxy + 3Gy^2, -1) \quad (2.6)$$

假設 $(a_i, b_i, c_i) = \left(-\frac{a_i}{c_i}, -\frac{b_i}{c_i}, -1\right)$ 為點 (x_i, y_i, z_i) 的法向量，而曲面上每個點

可整理為 $Ux=d$ 的形式， U 、 x 、 d 的關係如下：

$$U = \begin{bmatrix} \frac{1}{2}x_i^2 & x_i y_i & \frac{1}{2}y_i^2 & x_i^3 & x_i^2 y_i & x_i y_i^2 & y_i^3 \\ x_i & y_i & 0 & 3x_i^2 & 2x_i y_i & y_i^2 & 0 \\ 0 & x_i & y_i & 0 & x_i^2 & 2x_i y_i & 3y_i^2 \end{bmatrix} \quad (2.7)$$

$$x = [A \ B \ C \ D \ E \ F \ G]^T \quad (2.8)$$

$$d = \begin{bmatrix} z_i \\ -\frac{a_i}{c_i} \\ -\frac{b_i}{c_i} \end{bmatrix} \quad (2.9)$$

其中 U 為 $3n \times 7$ 的矩陣、 d 為 $3n$ 向量、 n 為此網格點的相鄰點的數量。再將所得到的 A 、 B 、 C 代入(2.5)，而計算矩陣 W 的兩個特徵值 (eigenvalue) 為主軸曲率，兩個的特徵向量 (eigenvector) 則為主軸方向。

接著再利用主軸曲率以及主方向計算一次微分如下：

$$e = \frac{\partial k}{\partial t} = \begin{bmatrix} t_1^2 \\ t_2^2 \end{bmatrix}^T \begin{bmatrix} e & f \\ f & g \end{bmatrix} \begin{bmatrix} t_1 \\ t_2 \end{bmatrix} \quad (2.10)$$

其中 k 為主軸曲率、 $t = [t_1 \ t_2]^T$ 為主軸方向、 e 為網格點之極限係數 (extremality coefficient)。如圖 2.7 所示， e 為主軸曲率對主方向的微分，曲線上的不同點有不同的極限係數。而此係數為點於曲線的斜率，當係數等於 0，代表此點有極值，若再對此係數作微分，可歸類此點為

凹或凸特徵線，詳細說明請參考[9]。

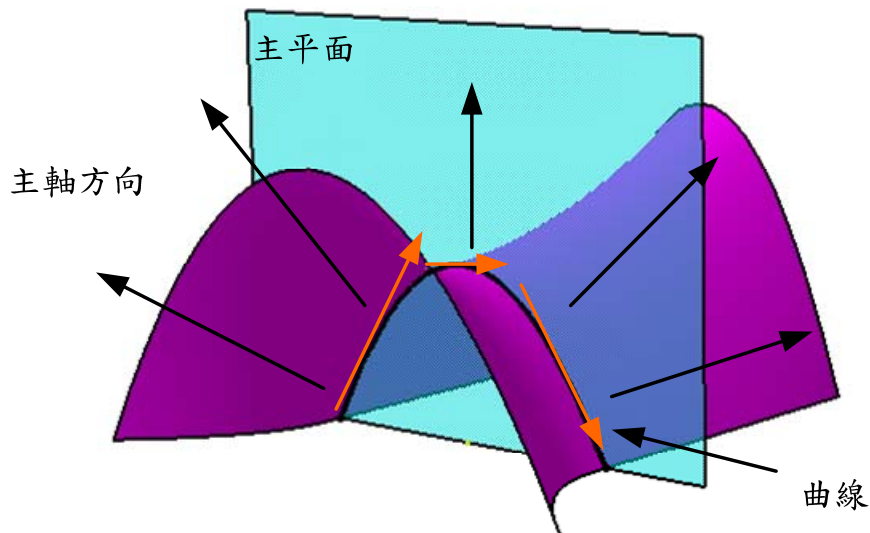


圖 2.7 極限係數示意圖

2.4.3 特徵線建構

特徵線建構可分為兩種：凸特徵線及凹特徵線，而建構可分為兩部分，如圖 2.8 所示。以下利用建構凸特徵線作為說明。

- (1) 第一部份為尋找特徵面。首先，計算網格模型中每個網格面的三頂點主軸曲率，並找出模型中的最大主軸曲率值。然後，針對每個網格面的三頂點主軸曲率平均，經過 *curveSen* 門檻值判斷是否成立，若成立，則進行特徵線建構的第二部份。若不成立，則進行下個網格面的三頂點主軸曲率計算。*curveSen* 門檻值可稱為曲率敏感值 (curvature sensitivity)，用意為確保重要特徵都能偵測到。此曲率敏感值為使用者手動設定，範圍為

0~1, 0 為最大敏感值, 若曲率有小變動就會被偵測, 如圖 2.9(a) 所示, 曲率敏感值為 0, 1 為最低敏感值, 需要很大的曲率變化才會被偵測, 如圖 2.9(b) 所示, 曲率敏感值為 0.8。

計算單一網格面的三頂點主軸曲率平均計算為下式：

$$verAvg = \frac{average}{maxStrength} \quad (2.11)$$

其中 *average* 為單一網格面的最大主軸曲率平均值、*maxStrength* 為在整體網格模型中最大主軸曲率平均網格面值。

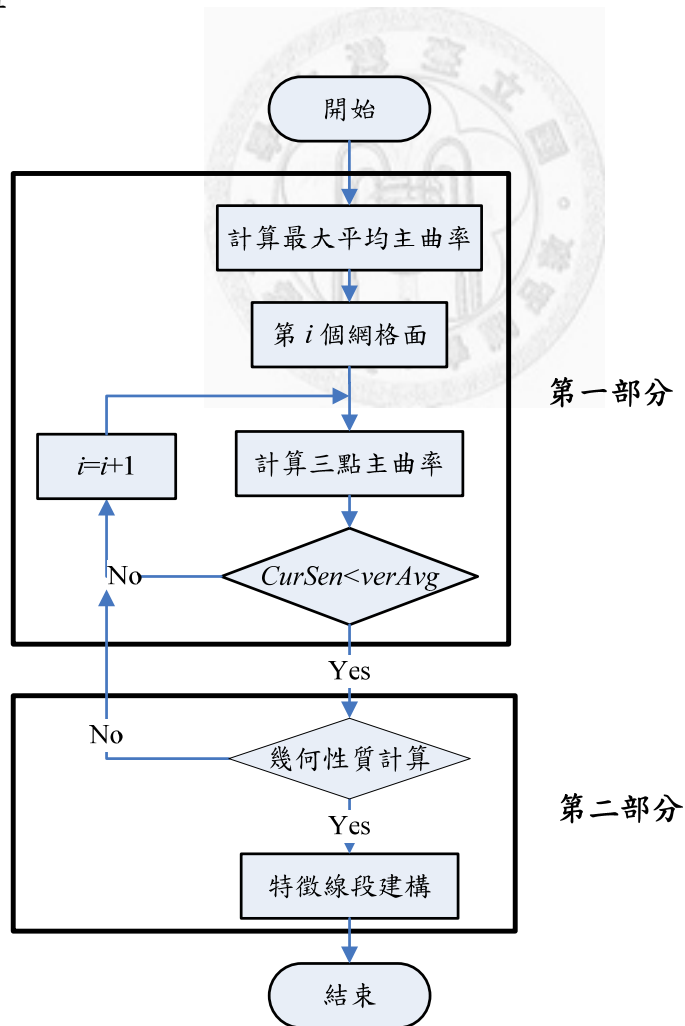


圖 2.8 特徵線建構之流程圖

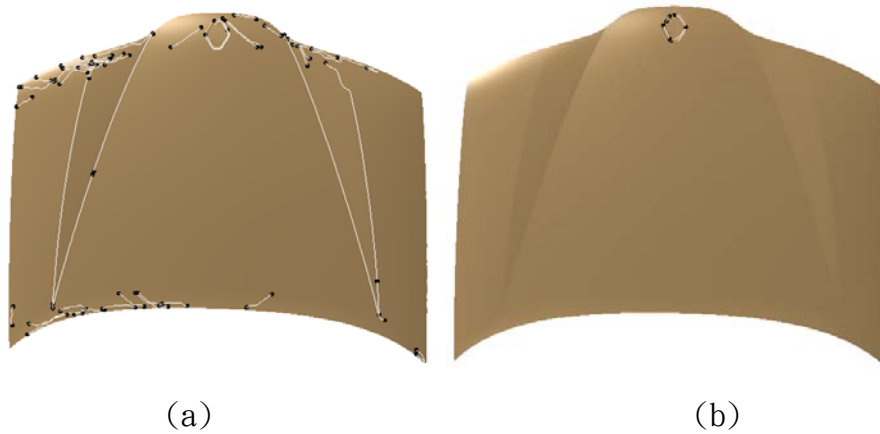


圖 2.9 特徵曲率敏感值比較示意圖

- (2) 第二部份為根據幾何性質計算取得特徵線段，若不符合條件，則回到第一部份，計算下個網格面的三頂點主軸曲率平均等程序。凸特徵線的定義如式(2.12)、凹特徵線的定義如式(2.13)。

$$e_{\max} = 0, \quad \frac{\partial e_{\max}}{\partial t_{\max}} < 0, \quad k_{\max} > |k_{\min}| \quad (2.12)$$

$$e_{\min} = 0, \quad \frac{\partial e_{\min}}{\partial t_{\min}} > 0, \quad -|k_{\max}| > k_{\min} \quad (2.13)$$

建構特徵線之前必須依序在每個網格邊上尋找特徵點的位置，以下為搜尋凸特徵點方法。

在網格邊 (v_1, v_2) 中，先檢查 $t_{\max}(v_1)$ 與 $t_{\max}(v_2)$ 之間的夾角，若夾角大於90度，則將 $t_{\max}(v_2)$ 的方向反轉： $t_{\max}(v_2) = -t_{\max}(v_2)$ ， $e_{\max}(v_1) = -e_{\max}(v_1)$ 。

接著判斷是否滿足以下條件：

$$k_{\max}(v_i) > |k_{\min}(v_i)| \quad \text{for } i = 1, 2 \quad (2.14)$$

$$e_{\max}(v_i)e_{\max}(v_{i+1}) < 0 \quad \text{for } i = 1 \quad (2.15)$$

$$e_{\max}(v_i)[(v_{3-i} - v_i) \cdot t_{\max}(v_i)] > 0 \quad \text{for } i = 1, 2 \quad (2.16)$$

若上述條件都滿足，則利用內差法求得 V_1 與 V_2 之間的特徵點。當所有的網格邊都搜尋完後，連接所有特徵點即可成為凸特徵線。

凹特徵點的搜尋與凸特徵點的方法雷同：

在網格邊 (v_1, v_2) 中，先檢查 $t_{\min}(v_1)$ 與 $t_{\min}(v_2)$ 之間的夾角，若夾角小於 90 度，則將 $t_{\min}(v_2)$ 的方向反轉： $t_{\min}(v_2) = -t_{\min}(v_2)$ ， $e_{\min}(v_1) = -e_{\min}(v_1)$ 。

接著判斷是否滿足以下條件：

$$-|k_{\max}(v_i)| > k_{\min}(v_i) \quad \text{for } i = 1, 2 \quad (2.17)$$

$$e_{\min}(v_i)e_{\min}(v_{i+1}) < 0 \quad \text{for } i = 1 \quad (2.18)$$

$$e_{\min}(v_i)[(v_{3-i} - v_i) \cdot t_{\min}(v_i)] < 0 \quad \text{for } i = 1, 2 \quad (2.19)$$

若上述條件都滿足，則利用內差法求得 v_1 與 v_2 之間的特徵點。當所有的網格邊都搜尋完後，連接所有特徵點即可成為凹特徵線。

連接特徵線是以單一三角網格為單位來判斷，如圖 2.10 所示，若網格面中沒有特徵點或只有一個特徵點，則網格面保持不動；若網格面中兩個邊上有特徵點，則連接兩個特徵點為一條特徵線；若網格面上三個邊都有特徵點，則三個特徵點分別與網格面的質心連接，形成三個特徵線段。

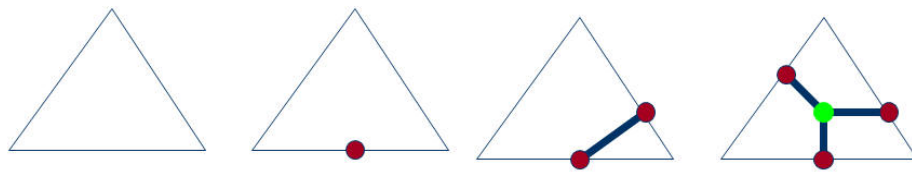


圖 2.10 連接特徵點示意圖(引用[11])

2.4.4 特徵線後處理

擷取特徵線演算法的最後部份為線段後處理，此用意為減少線段數量，此線段後處理流程圖，如圖 2.11 所示。首先將上一步所算出的特徵線設定頭尾兩點座標，再進行線段連接。線段連接的處理分為兩種：*JoinConnectCurves* 與 *JoinNonConnectCurves*。*JoinConnectCurves* 作用是把相同點做連接；而 *JoinNonConnectCurves* 作用是把不連續點(有間隙)做連接。圖 2.12 為線段後處理之範例，(a)與(c)為原始線段，(b)與(d)為經過後處理之線段。(a)與(b)為經過 *JoinConnectCurves* 處理；(c)與(d)為經過 *JoinNonConnectCurves* 處理。

當所有都線段連接完成，再尋找所有線段裡最長的線段並且設定每條線段之 *ratio* 值，若此值大於程式所設定的門檻值，則進行線段處理，若小於所設定的門檻值則刪除該線段。*Ratio* 值計算公式為：

$$ratio = \frac{length}{maxLength} \quad (2.20)$$

其中 *length* 為單一線段的長度、*maxLength* 為所有線段裡最長的線段。如圖 2.12(b)與圖 2.12(d)所示，有很明顯的細小線段被刪除以及細小特徵線被合併為較長線段。

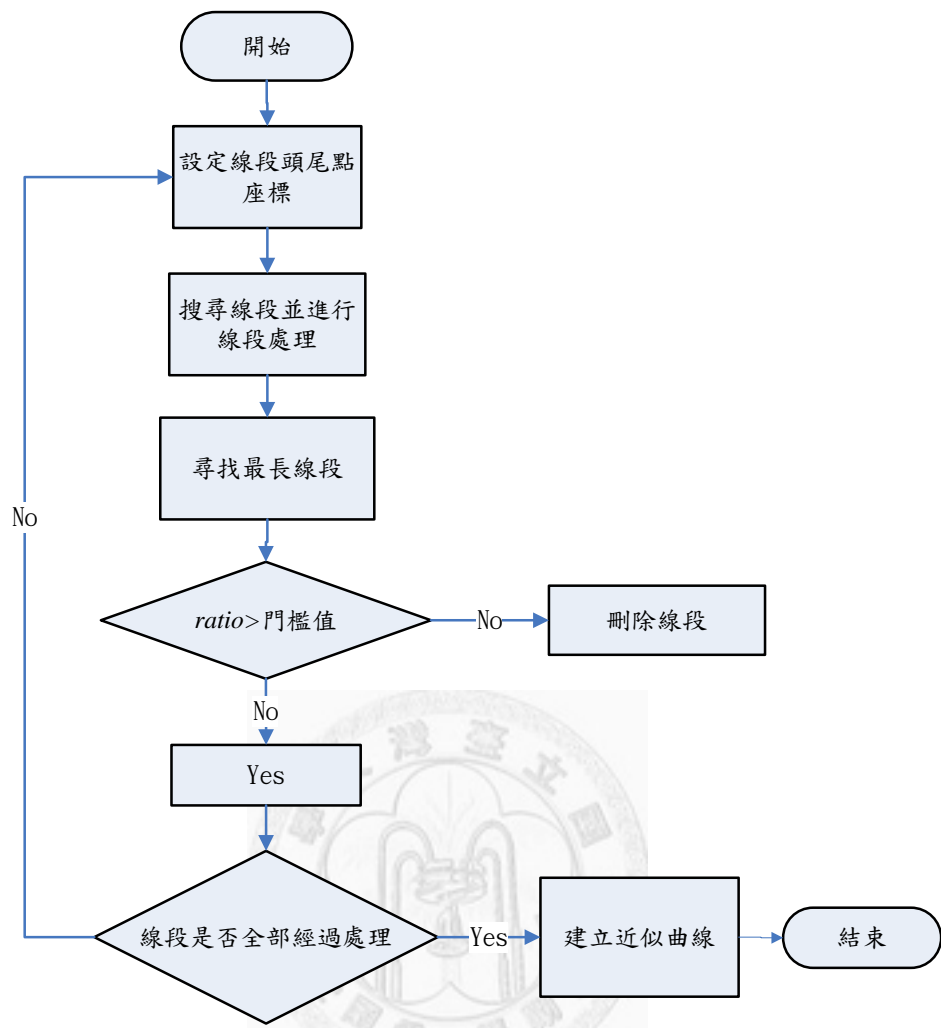
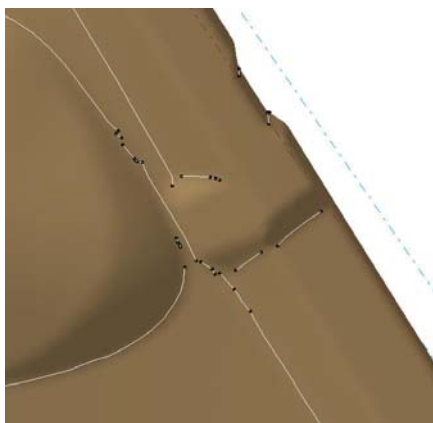
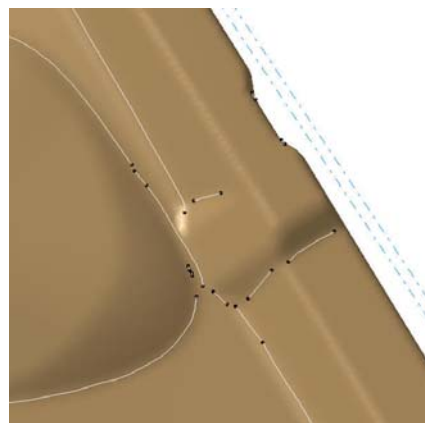


圖 2.11 特徵線後處理流程圖



(a)



(b)

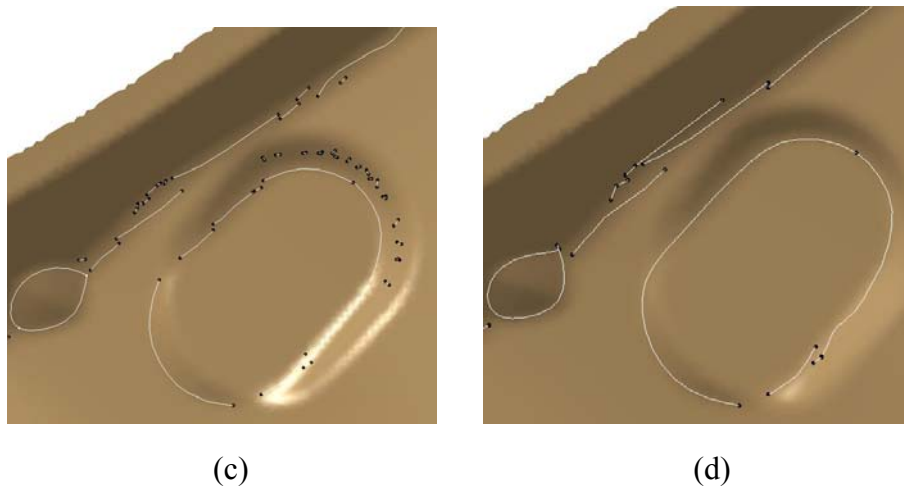


圖 2.12 線段後處理之示意圖

當所有線段都經過處理後，再利用 Pieg1[5]等人所提出的最小平方
法 (least squares method) 建立一條近似曲線 (approximate curve)。圖 2.13
為經過最小平方方法之特徵線結果，其中藍色線段為原始特徵線、紅色
線段為近似曲線。

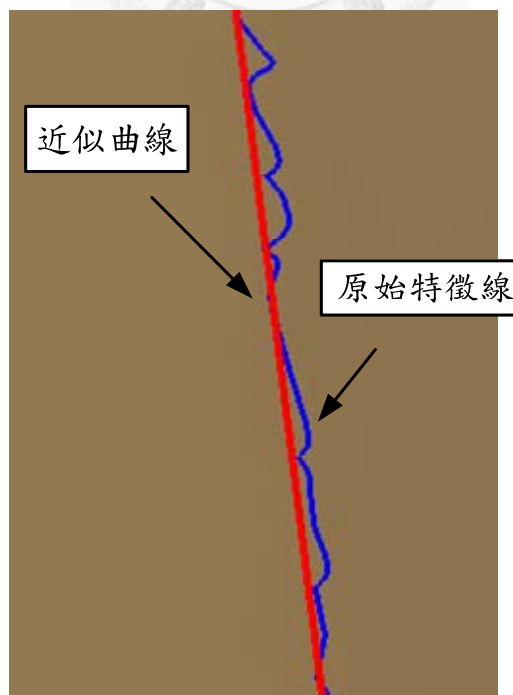


圖 2.13 原始特徵線與近似曲線 (經過最小平方方法處理之特徵線)

2.5 特徵萃取演算法及逆向工程軟體的成果比較

在介紹過三種演算法以及一個逆向工程軟體之後，本小節將整理這四種方法作一比較與說明：

- (1) SOD 是最簡單的演算法，只用到一個公式和雙門檻值，就能顯示萃取的特徵線段，由於它的簡單明確，利用程式建構此演算法較為容易。此演算法適合用於網格優化前處理之前或較粗糙的網格資料，但因為該演算法只針對局部區塊，在高精細或複雜的網格資料中結果不佳[3]。
- (2) ESOD 為 SOD 的延伸演算法，因其顧及周邊網格資訊，所萃取的特徵線段雜線減少許多，而特徵發生處較為完整，但在較粗糙的網格資料，所萃取的特徵表現不佳[3]。
- (3) RidgeAndRavine 演算法雖然概念有點複雜且計算時間較久，但是在特徵萃取可得到不錯的結果。此演算法將特徵分為凹凸，分別利用最小及最大主軸曲率來計算，所以網格資料裡所有的特徵都可被正確的萃取出來，而且最後搭配特徵線段後處理，可將特徵線完美的顯示，並不會有過多較短的線段。
- (4) Geomagic Studio 是利用區域成長演算法搭配細線化來取得特徵線。雖然萃取特徵功能完善，但是所萃取的特徵無法進行下階段動作：網格分割及網格分群，所以此軟體只能用來與以上所提的演算法做比較或大概確定特徵所在位置。

表 2.1 將上述四種特徵線萃取方法以運算時間、參數值、參數調整與辨識率分析探討。在這四種方法的辨識結果比較中，運算時間方面 Geomagic Studio 最快，而 RidgeAndRavine 最久；參數調整方面 RidgeAndRavine 只需要輸入一個參數值，而 Geomagic Studio 必須輸入三個參數值來萃取最完整的特徵；辨識率方面則以 RidgeAndRavine 表現最佳，所有的特徵區域皆有辨識出來，因此本研究將利用此演算法進行特徵線萃取動作與後續曲面重建。

表 2.1 特徵萃取演算法與逆向工程軟體的成果比較

特徵線萃取 方法 比較項目	SOD	ESOD	RidgeAndRavine	Geomagic Studio
計算時間	約 30 秒	約 45 秒	約 85 秒	約 20 秒
參數值	上限 2.0	上限 6.0	曲率值 0.05	曲率值 50.0
	下限 1.0	下限 4.0		曲分線段值 10.0
				最小面積 150.0mm ²
參數調整	2	2	1	3
辨識率	65%	80%	95%	90%
	線段 過於零碎	線段 過於零碎	特徵區域 皆已辨識完成	主要特徵區已辨識 完成，但特徵線結 果與特徵稜線未重 合
	有區域未辨識	有誤判區域		

第三章 三角網格分割與網格面分群

3.1 前言

在逆向工程中，曲面的二次分割會將大網格區域分成許多小網格區域，而本研究利用特徵線(feature line)萃取方法來進行網格分群。不論利用何種特徵線萃取演算法則，都無法直接進行網格分群動作，因為萃取的特徵線難以全然完整，因此必須對所搜尋的特徵線做延伸。

本章節將提出三角網格分割與網格面分群法則，此法則主要可分為三個階段，如圖 3.1 所示。首先將依照第二章所萃取的特徵線分佈延伸至整體網格邊緣並找出特徵線所經過三角網格的特徵點(feature point)，再針對特徵線所通過的三角網格作網格分割(mesh segmentation)，最後將特徵線與邊界所圍出來的封閉區域進行分群，而此區域將成為許多不同的網格資料區域。

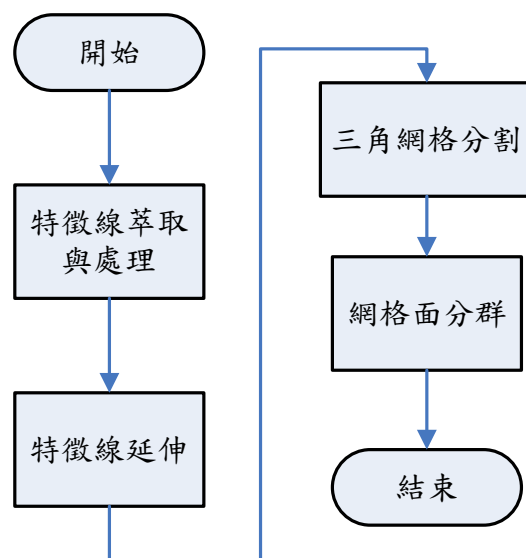


圖 3.1 三角網格分割與網格面分群流程圖

在討論特徵線延伸法則、網格分割流程以及網格面分群等之前，必須將網格拓樸架構做進一步的說明。本研究沿用曾凱威[14]建立的半邊資料結構(half-edge data structure)進行後續應用與功能開發，因此先簡單介紹此資料結構，再針對本章節所提出的三角網格分割與網格面分群法則進行細部說明。

3.2 拓樸資料的建立

建立正確網格間的拓樸資料是一件非常重要的工作。以下小節將分別對點(vertex)、邊(edge)、面(facet)資料的建立進行說明。

3.2.1 點資料的建立

每筆點資料都會記錄其在空間中的 X 、 Y 、 Z 座標，同時依照讀取的順序指定編號(*vertexID*)。另外每筆點資料必須紀錄該點為起始點的邊編號(*starEdgeIDArray*)以及邊數量(*starEdgeNumber*)。如圖 3.2 所示，以 V_2 為範例，此點所記錄的邊編號為 E_2 及 E_5 ，邊數量為 2。

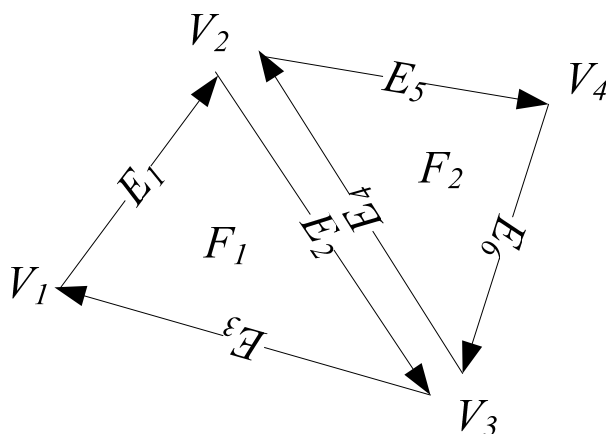


圖 3.2 半邊結構示意圖

3.2.2 邊資料的建立

邊資料為半邊結構的主角，紀錄大部分的網格相關資料，如起點編號(*startID*)、終點編號(*endID*)、所在面編號(*facetID*)、次邊編號(*nextEdgeID*)、前邊編號(*prevEdgeID*)、反向邊編號(*adjEdgeID*)及邊編號(*edgeID*)，其中反向邊編號為邊資料中重要的拓樸資訊。例如，若要檢查此邊是否屬於孔洞或是邊界，只需檢查該邊的反向邊編號，若反向邊編號小於零，則此邊屬於孔洞或是邊界。以圖 3.2 的邊為例，表 3.1 為此邊結構對應表。

表 3.1 邊結構對應表

邊資料	資料點		所在面	關聯邊資訊		
	起點	終點		ID	次編	前邊
E_1	V_1	V_2	F_1	E_2	E_3	
E_2	V_2	V_3	F_1	E_2	E_1	E_4
E_3	V_3	V_1	F_1	E_1	E_2	
E_4	V_3	V_2	F_2	E_5	E_6	E_2
E_5	V_2	V_4	F_2	E_6	E_4	
E_6	V_4	V_3	F_2	E_4	E_5	

3.2.3 面資料的建立

每片面資料是由三個點資料座標建構而成，而這些點資料的排序必須滿足此面的法向量方向的右手定則。面資料除了儲存三個點點資料(*vertexIDArray*)及法向量(*facetNormal*)之外，還儲存此面編號

(*facetID*)、網格區塊編號 (*patchID*) 及三個邊資料構成此面 (*edgeArrayID*)。以圖 3.2 中的 F_1 為例，此面的三點點資料為 V_1 、 V_2 、 V_3 ；三邊邊資料為 E_1 、 E_2 、 E_3 。

3.3 特徵線延伸

楊偉倫[15]所提出的延伸法則主要是尋找特徵線端點所有相關的邊，並使每個延伸邊的方向都投影到平面上，再進行角度比較，取最小角度發生的方向作為延伸的邊，直到接觸到網格邊緣線並結束。而本研究由其概念延伸如圖 3.3 所示，以直線相交分割三角網格及不斷的投影進行搜尋，不會受到曲率變化的影響，與偏離預期路徑而造成延伸線成為鋸齒狀，以下將逐步說明特徵線延伸的流程。



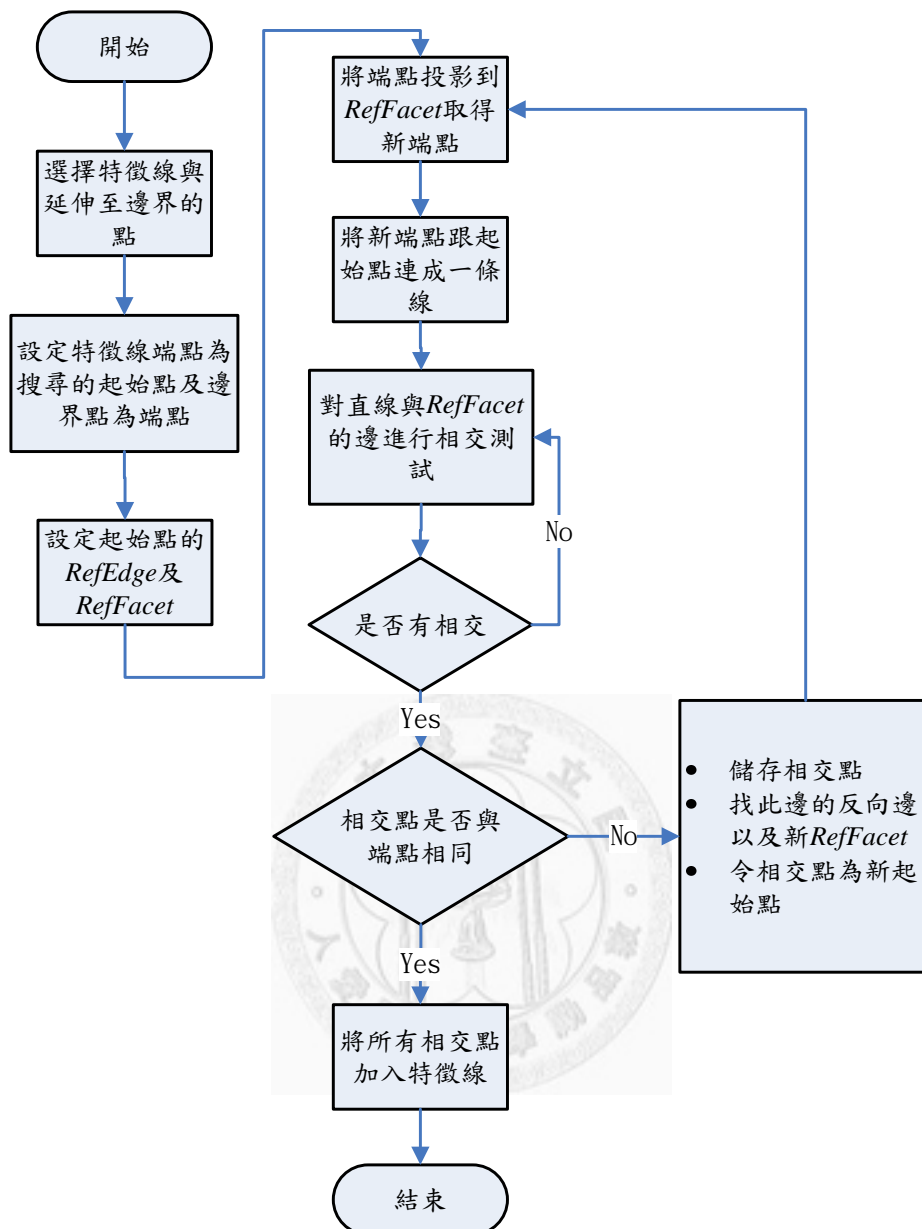


圖 3.3 特徵線延伸之流程圖

3.3.1 特徵線延伸流程說明

首先先選定一條特徵線，再選取一個邊界點為延伸至整體網格邊緣之終點。對其線段的兩端點與邊界點做距離比對，距離較小的設定為搜尋的起始點 (*startPoint*)，而延伸至網格邊緣的點為終點 (*endPoint*)。接著設定起始點所屬於的面 (*refFacet*) 跟邊 (*refEdge*)，將終

點投影到 *refFacet* 取得新終點。將新終點與起始點連成一條線並開始對 *refFacet* 的三個邊進行相交測試，若有相交點，定義此點為特徵點。然後檢查是否此特徵點與終點相同，若為否，則將儲存相交點、尋找相交點的反向邊以及對應的面資料、最後設定相交點為新起始點，重複以上步驟直到起始點與端點相同便結束此特徵線的延伸。此法則是以起始點不斷的更新而端點保持不變。圖 3.4 為特徵線延伸之新特徵線，其中紅色線段為原始特徵線、藍色點為特徵點、綠色線段為延伸線段。

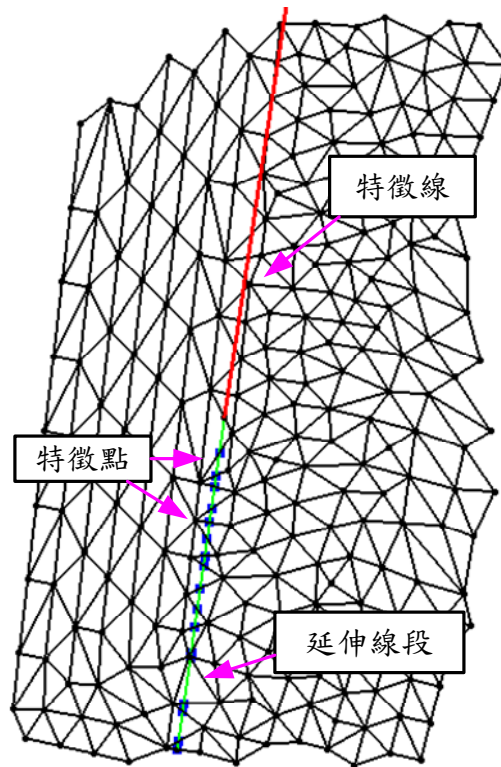


圖 3.4 特徵線延伸

3.3.2 曲線上取點

在特徵線延伸裡，不一定每條特徵線段都會延伸至整體網格的邊緣，有些線段必須延伸至別條特徵線段，如圖 3.5，由線段 2 延伸至線段 1 上，但線段 1 並沒有任何可點選的點，所以必須利用曲線上取點之功能新增點，便於應用 3.3.1 小節所提出的特徵線延伸功能，因此發展曲線上取點功能。

曲線上取點的計算可分為兩類。第一類為利用點在於曲線上的長度比例，第二類則利用曲線上的端點與新點的距離，這兩種計算方式都能將新點的座標算出並正確的將點顯示在曲線上。本研究利用第一類將點新增至特徵線段上，以作為特徵線延伸的輸入條件之一。以下將簡單介紹曲線上取點之流程。

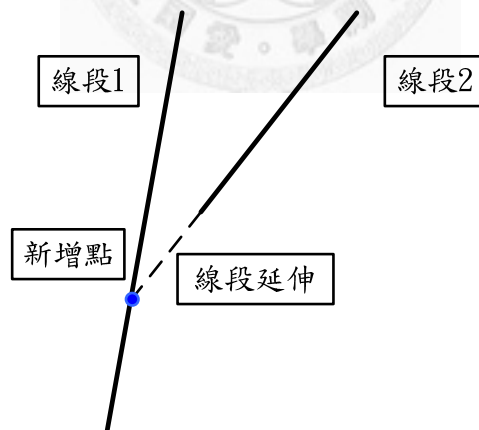


圖 3.5 曲線上取點之範例

首先由人工點選欲新增點的特徵線段，設定新點在曲線上的長度比例。然後將此點座標與點選的特徵線裡的特徵點做比對，將距離最近的特徵點座標取出並儲存此座標為新點座標。當新點座標已定，就可將點選特徵線一分為二。如圖 3.6 所示，此特徵線段以比例為 0.5 將

一條特徵線分割成兩條新線段，而且方向與原本特徵線段的方向相同，並且在新點上產生線段 1 的終點與線段 2 的起點。

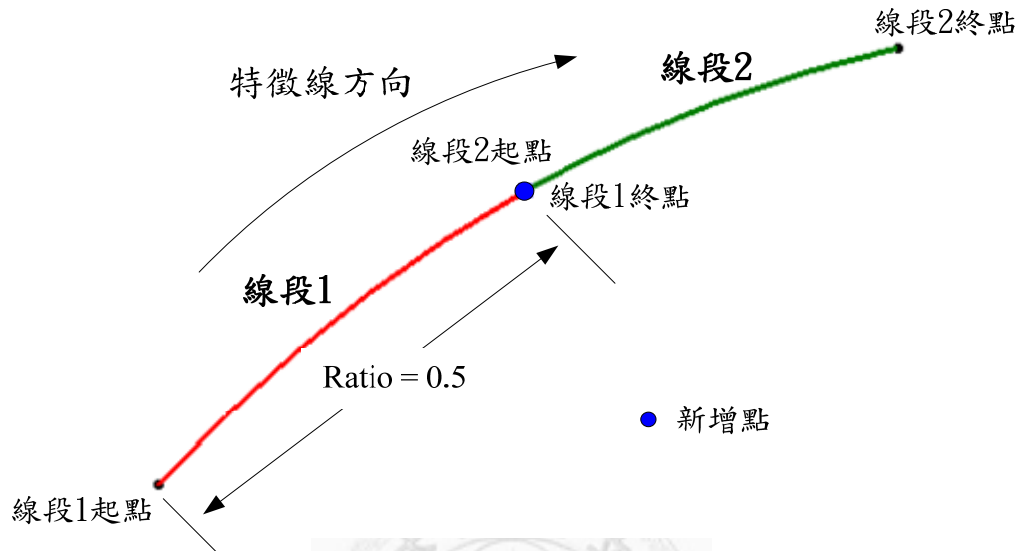


圖 3.6 曲線上取點之示意圖

3.4 三角網格分割

經由 3.2 節的特徵線延伸，可串連對應特徵點以建立新網格邊，依照所屬的網格面加入網格邊，可將所有的特徵網格分割，並判斷網格位於特徵線的左邊或右邊，利於後續網格分群。不過以單條網格邊無法建構完整的三角網格，所以必須透過三角網格分割法則來達成。

三角網格分割法則主要是以邊資料作為依據，因此在進行三角網格分割之前，必須先檢查所有特徵網格面的邊資料是否正確，其中包括正向邊與反向邊、邊方向、起始點、終點、網格面法向量計算。本分割法則可分為兩個步驟，第一步是尋找分割三角網格中的封閉迴路；第二步驟則是處理與分割該封閉迴路，並建立與更改拓樸資料。以下

小節將分別加以介紹。

3.4.1 尋找分割三角網格中的封閉迴路

在特徵線延伸完成後，每個特徵面都儲存許多的邊，而這些邊必須進行如圖 3.7 所示的流程，將網格裡的邊建立為許多的封閉迴路以利後續網格分割。首先，依序讀取特徵面中的每筆邊資料，判斷各筆面資料的邊數量，若邊數量為 0 就結束此流程並輸出該流程所找出的封閉迴路；若邊數量為大於 0 則進行邊的先後順序判斷，以第一個邊為起始邊進行循環式的搜尋。搜尋方法為檢查是否有邊的起點等於第一個邊的終點，若有，則將此邊設定為次邊，而第一邊設定為此邊的前邊，重複上述的動作，當連接關係重新回到第一個邊的起點，整個連接關係即達到封閉的狀態，間接表示該邊界迴圈已建立完成，並儲存該封閉迴圈以及重新設定邊的數量，並回到進行以上邊數量的判斷進行尋找下個封閉迴路或結束。當特徵面完成封閉迴路的搜尋，則進行三角網格分割法則的第二步驟：處理與分割特徵面。

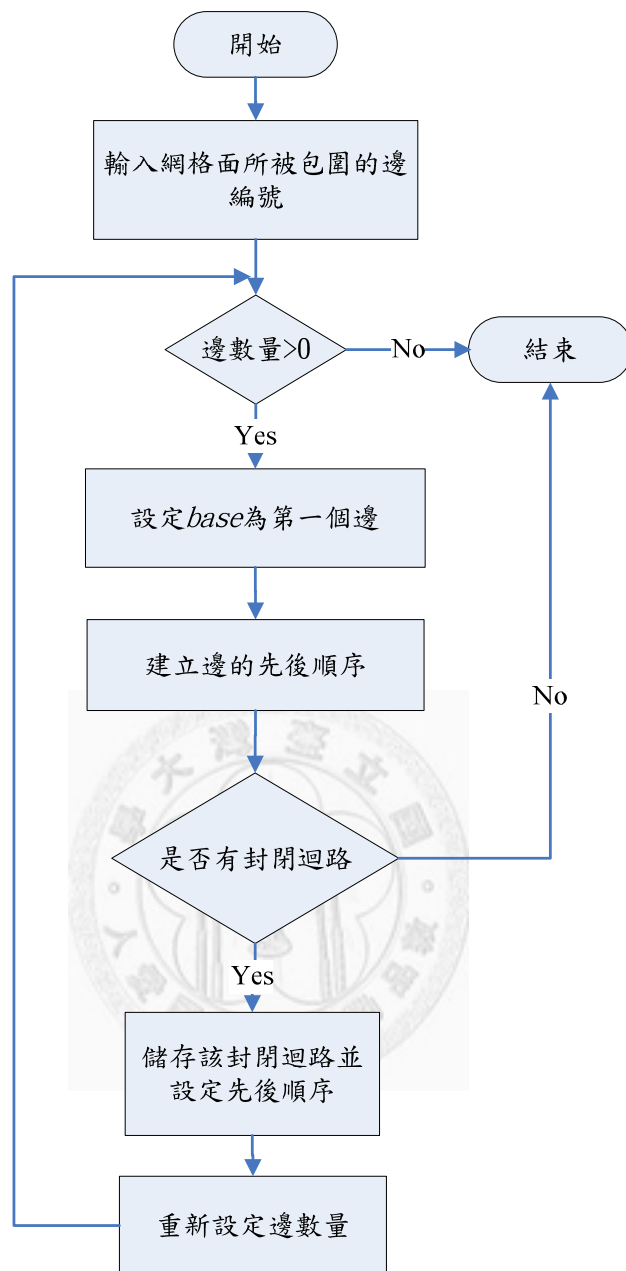


圖 3.7 尋找分割三角網格中的封閉迴路之流程圖

3.4.2 處理與分割特徵面之封閉迴路

當特徵面之封閉迴路都建立完成，則開始分割特徵面成數個小三角網格，並重新建立拓樸資料。首先，將 3.4.1 所搜尋的封閉迴路進行迴圈邊數量判斷，若邊數為 3，則直接建立新網格面的基本資料，其資訊包含網格面編號、網格塊編號、頂點資料容器、邊資料容器、以及

網格面法向量。接續建立網格面的三邊資料連接關係，設定邊資料容器的第一邊為 *rootEdge*，依序設定次邊以及前邊，並且設定此三邊的網格面編號為此面編號，便於後續網格分區域。若邊數量為 4，則直接將第一跟第三頂點連成一條線以及第二跟第四頂點連成第二條線，兩條線相比距離，距離最小者作為分割邊依據。當分割邊已建構成形，就可以開始分割四邊封閉迴路，成為兩個小三角網格，並且依照上述邊數量為三的步驟建立新網格面的基本資料。當網格面裡的新三角網格都建立完成後，將所有小新網格面加入整體網格面的資料容器裡，並把原始網格面設為 *null*，防止後續網格分區域時用到未分割網格面。

由 3.3.1 小節所搜尋的封閉迴路，網格邊數量皆不一至，並產生三種特徵線分割情況，分別為兩條、一條、零條之特徵線，如圖 3.8 所示。

- (1) 兩條為雙特徵線切割原始網格，如圖 3.9(a)。
- (2) 一條為單一特徵線切割原始網格，如圖 3.9(b)。
- (3) 零條為特徵線切割剛好於網格邊重疊或相交點與網格面頂點距離差距很小的情況，如圖 3.9(c)。

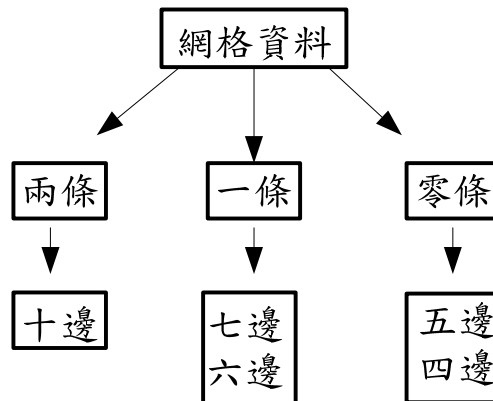


圖 3.8 三種特徵線分割情況

在三種情況之下，顯示五種不同的案例，如圖 3.9，其中綠色線段為特徵線、紅色框選區域為特徵面。

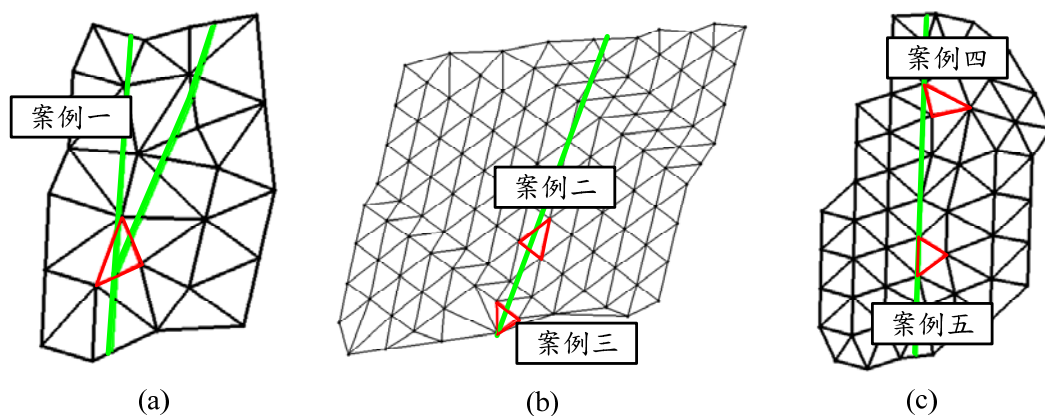


圖 3.9 特徵網格分例探討

以下針對各案例處理方式進行討論。以下五個案例，其中(a)為原始網格、(b)為經過特徵萃取或特徵線段延伸所分割之網格、(c)為經過三角網格分割法則所分割之網格面。

(1) 案例一：十邊分割成四個三角網格

由圖 3.10(b)之十個箭頭為十邊，而此邊經過上述 3.3.2 小節的處理可將一個原始網格分割為四個新三角網格，即為十邊狀況，並重新建立網格的拓撲關聯，便於後續搜尋與處理，如圖 3.10(c)所示為分割處理後結果。

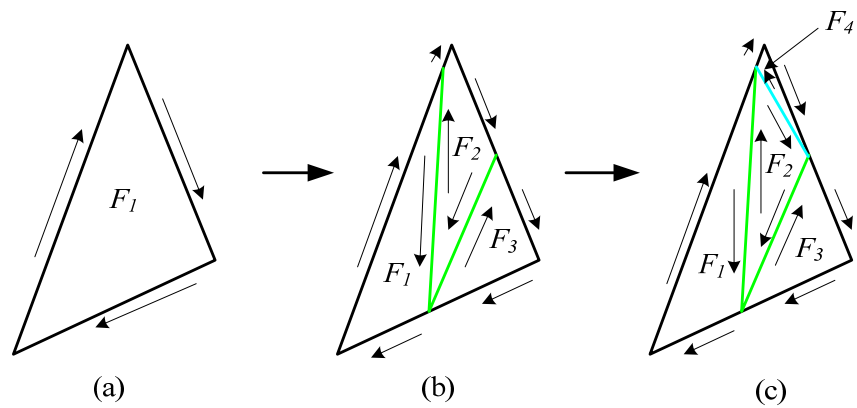


圖 3.10 分割十邊之結果

(2) 案例二：七邊分割成三個三角網格

此案例跟案例一雷同，由圖 3.11(b)之七個箭頭為七邊，而此邊經過 3.3.2 小節的處理可將一個原始網格分割為三個新三角網格，並重新建立網格的拓樸關聯，便於後續搜尋與處理，如圖 3.11(c)所示。

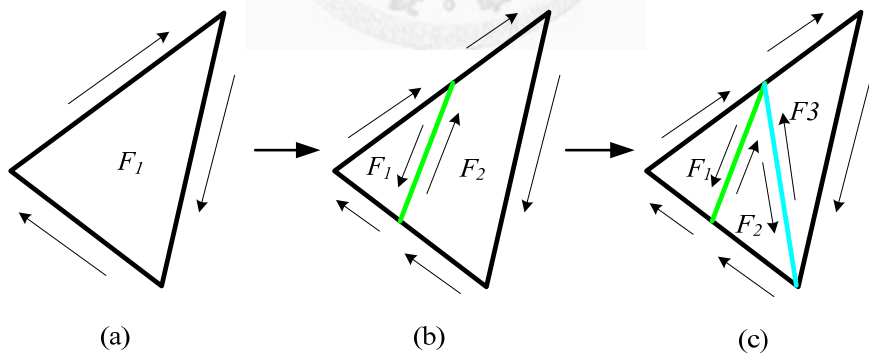


圖 3.11 分割七邊之結果

(3) 案例三：六邊分割成兩個三角網格

由圖 3.12(b)之六個箭頭為六邊，此案例特別處，為特徵線段的端點剛好跟原始網格的頂點相同，不需要額外新增邊來分割網格，因此特徵線段成為分割此網格的依據將一個原始網格分

割為兩個新三角網格，並重新建立網格的拓樸關聯，便於後續搜尋與處理。

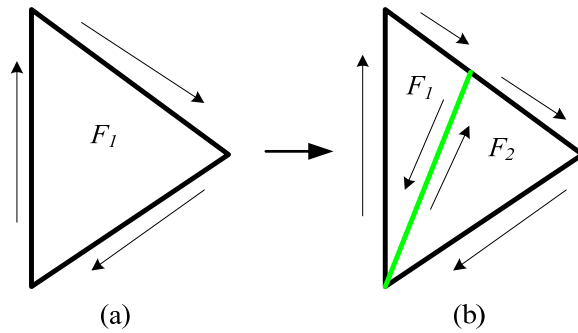


圖 3.12 分割六邊之結果

(4) 案例四：五邊分割成一個三角網格

圖 3.13(b)之五個箭頭為五邊，此案例不進行網格分割並保持原始網格的拓樸關聯，因為特徵線段的兩端點剛好跟原始網格的頂點相同，只需要將特徵線段所產生的正向邊與反向邊刪除即可，因此五邊網格將被縮減為三邊網格。

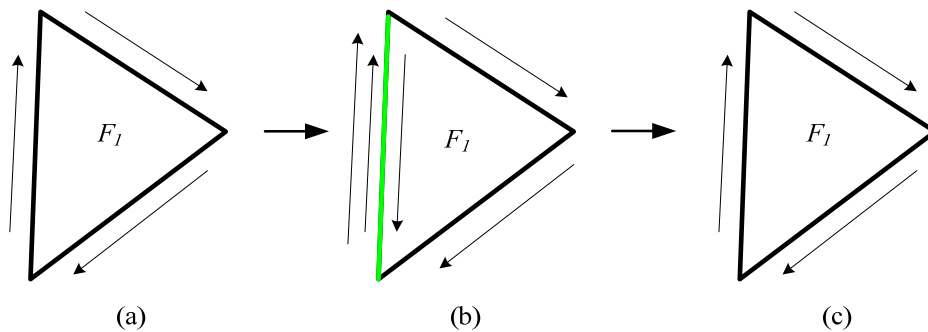


圖 3.13 分割五邊之結果

(5) 案例五：四邊分割成一個三角網格

根據本研究的分割特徵邊，若特徵點的距離與原始網格頂點小

於 0.005mm，則此特徵點將被合併至最近的原始頂點。由圖 3.14(b)，特徵線其中一個端點的距離與原始網格的頂點距離小於 0.005mm，因此只有四個箭頭為四邊。此案例不進行網格分割並保持原始網格的拓樸關聯，只需要將已分為兩個邊的邊還原為原始網格邊，如圖 3.14(c)所示。

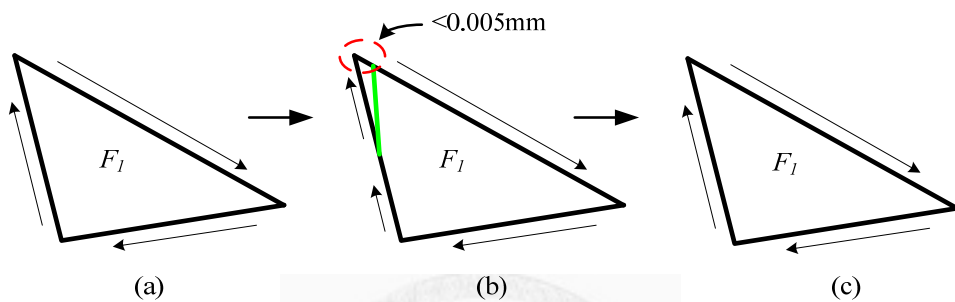


圖 3.14 分割四邊之結果

3.5 三角網格面之錯誤法向量

在 3.3 節裡，將特徵面分割完後，都必須新增網格邊以及網格面，所以必須重整拓樸資料，每個網格面都必須要有五個基本資料，分別為網格面編號、網格塊編號、頂點資料容器、邊資料容器、以及網格面法向量。若其中一項有錯或是空的，將嚴重影響拓樸關聯性，導致整個三角網格分割流程因此出現錯誤而無法有正確的結果，其中網格面之法向量錯誤最為普遍，以下提出解決方法。

法向量方向的判斷主要是以三角網格的三個頂點的順序來決定，通常都是以右手定則來決定法向量方向以及大小。法向量主要發生錯誤分為兩類。第一類為法向量的值與三角網格面頂點所計算出的值完

全不一樣。第二類為法向量不一致的情況，如圖 3.15 所示，一個網格頂點順序為順時針方向，然而它相鄰的網格頂點順序為逆時針方向，兩個網格則會產生法向量不一致狀況。

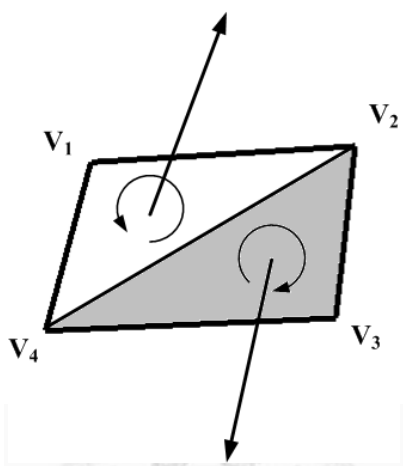


圖 3.15 法向量不一致

在進行特徵網格分割時，因為法向量的不一致導致在輸出網格塊過程中，出現許多不正常的網格面，如圖 3.16(a)。因此在修正法向量後，不正常邊都會完全的消失，如圖 3.16(b)，其中綠色虛線段為網格邊界、紅色區域為不正常法向量網格面、黑色線段為普通三角網格邊。

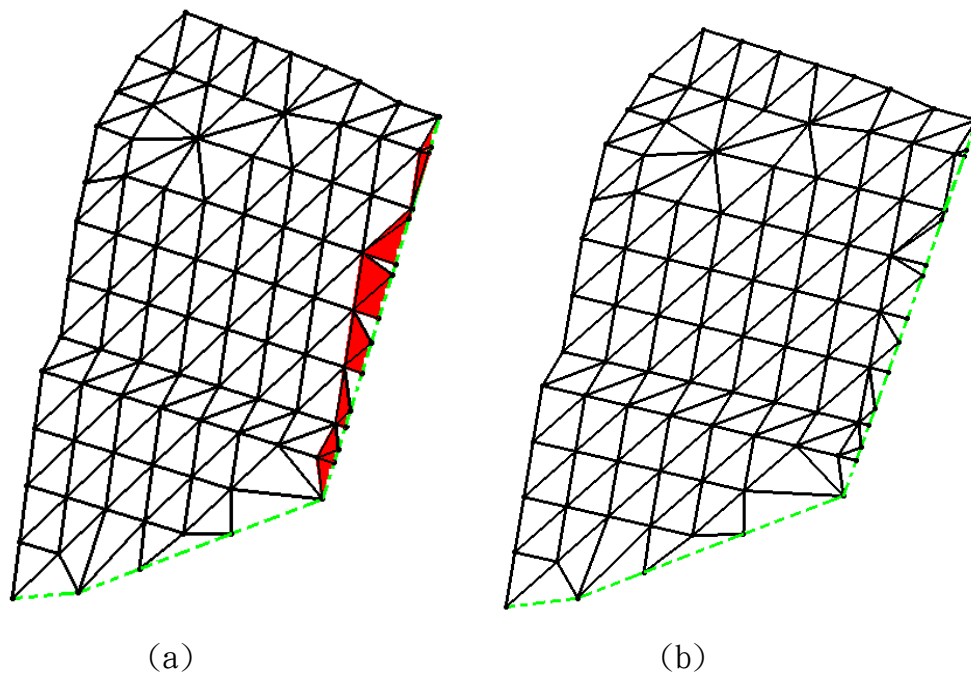


圖 3.16 法向量不一致處理

3.6 網格區域分群

網格區域的數量關連到與原始鈹件的精確度，網格區域分的越多，表示曲面與原始鈹件的精確度越高，但是當區域越多，處理的時間越久，所以盡量以大區塊為主。當曲率較大時，大區塊將會被分割成較小區塊。以下為區域劃分法則與流程說明。

3.6.1 暫存網格面容器

由以上小節得到延伸並且完成網格分割的封閉區域後，接續進行網格區域的分類，並建立兩個暫存網格面資料容器，分別為 *FeatureFacetArray* 以及 *NonFeatureFacetArray*。*FeatureFacetArray* 要用來暫存有特徵點與特徵邊的網格面；*NonFeatureFacetArray* 則暫存沒有任何與特徵有關係的網格面，如圖 3.17 所示，(a)為

FeatureFacetArray 所暫存的特徵網格，(b)為 *NonFeatureFacetArray* 所暫存的非特徵網格。



圖 3.17 暫存網格面資料容器

3.6.2 十字相交法

十字相交法(cross intersection)是用來決定網格區塊邊界，以及網格面屬於哪個網格區塊的一個方法。此方法概念很簡單，首先必須將所有的邊界利用延伸擠壓(extrusion)的方法建立曲面，再從網格面的質心建立四條射線，方向為 $+X$ 、 $-X$ 、 $+Y$ 、 $-Y$ 。然後將所有的曲面與四條射線做相交，紀錄各方向的相交邊界編號，若某方向有多個相交點，則紀錄最靠近質心的邊界編號。如圖 3.18 所示，此網格面的形心所射出的四條延伸線所相交到的邊界為[7,6,1,5]。

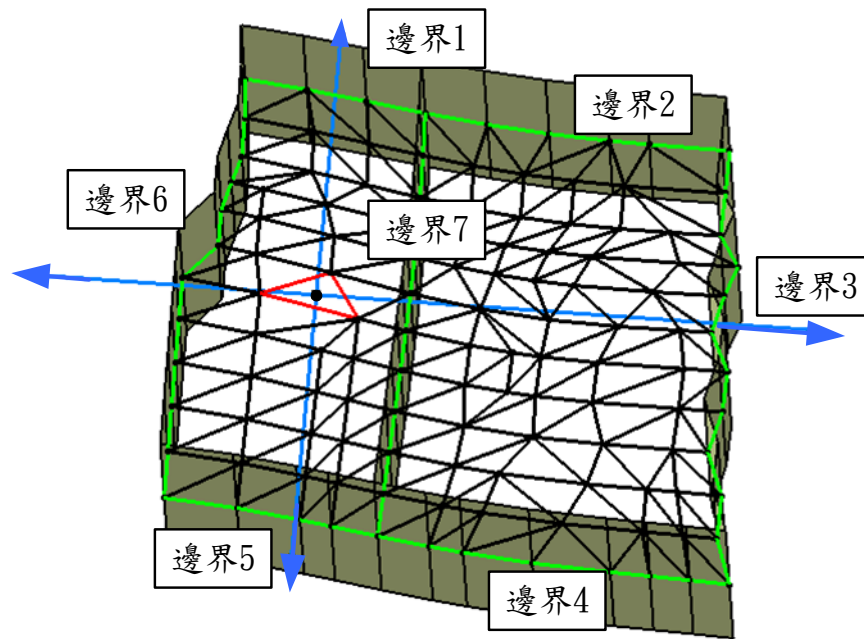


圖 3.18 十字相交之圖

3.6.3 網格分群流程

此網格分群是參考曾凱威[14]所提出的區域成長，並作局部修正，擬定一個適用於本小節的網格分群。網格分群利用整體網格的 *FacetArray* 容器進行擴散式的搜尋，在搜尋之前必須將每個三角網格面所包含的網格塊編號初始化，確保沒有任何三角網格面已經屬於某網格塊，避免後續的麻煩。處理的方式，以圖 3.19 為例，是由第一網格面 F 為中心，開始判斷此網格面的三個邊是否有反向邊，若為有就找出該邊的反向邊以及網格面編號，並將該面加入該網格塊之中，避免網格面重複加入網格塊，在網格面加入網格塊之中就必須設定該網格塊編號在該網格面裡。當 F 的所有相鄰面都確認完畢後，進行下一圈的網格面搜尋，但在下一圈網格相鄰面或反向邊為 *null* 或是碰到網格塊的邊界，則停止該網格方向搜尋。當所有面資料都完成以上的程

序時，表示該圈擴散式的搜尋已結束。若 *FacetArray* 容器裡的網格面尚未有網格塊編號，則必須執行以上的程序。若 *FacetArray* 容器裡的所有網格面都有網格塊編號，則表示整體網格塊已結束分群。

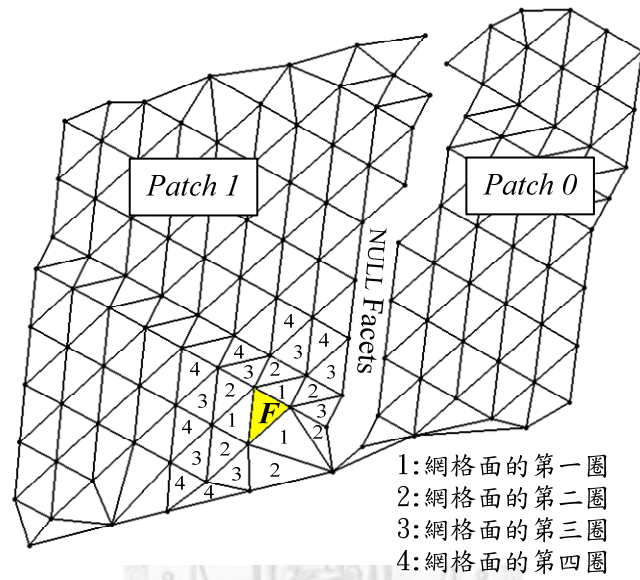


圖 3.19 網格分群示意圖

3.6.4 建立完整區塊

本研究定義完整區塊為網格面分割後，建立封閉迴路，取得之獨立網格區塊。在建立完整區塊時，利用 3.5.1 小節所提的暫存網格面容器存取網格資料。再透過 3.2 節所延伸的特徵線，網格塊與網格塊才能使網格有分割的依據，也才能使 *FeatureFacetArray* 裡的特徵面加入由 3.5.3 小節所建立的網格塊裡，以完整的區塊呈現。在執行特徵面加入未完整網格塊之前，必須先透過如圖 3.20 流程設定網格區塊邊界編號。

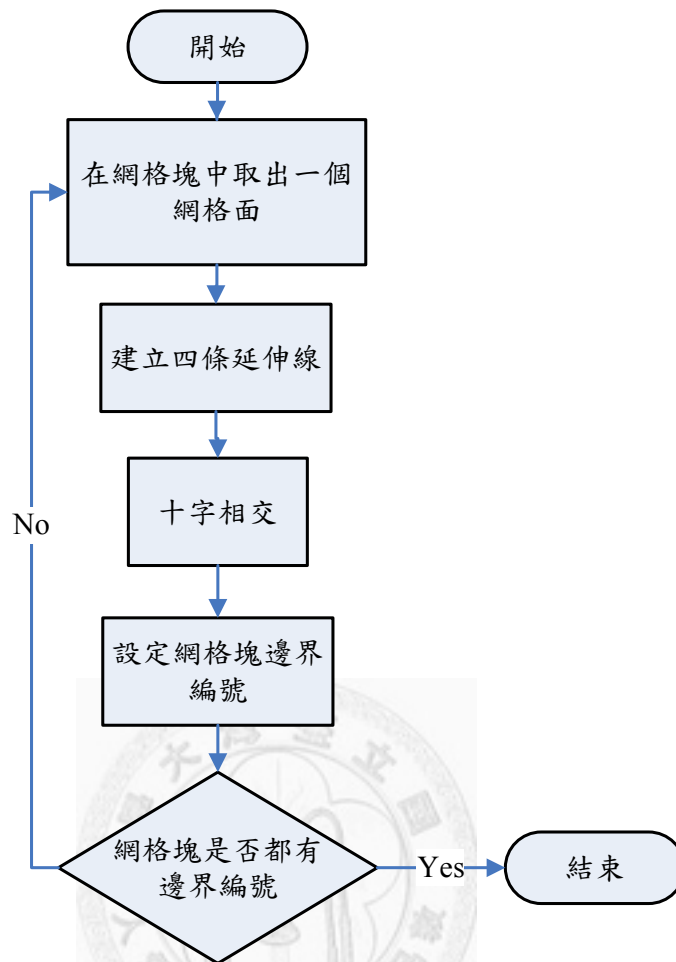


圖 3.20 設定網格區塊邊界編號之流程圖

在設定網格塊的邊界編號之前，必須確保所有特徵線都已延伸並與邊界圍成的區域構成封閉迴路，若為否，則必須回到 3.2 節完成未延伸的特徵線。若確定所有特徵線都已延伸，就可分別從每個網格塊中取出一個網格面，如圖 3.21(a)所示，以網格面的形心建立四條延伸線，然後做十字相交，並紀錄交點邊編號為此網格塊的邊界編號，如圖 3.21(b)。

由於本研究是透過網格面的區域成長來進行網格塊的資料建立，而其中網格面的特徵屬性又是依據三頂點的特徵屬性來判斷，所以會

產生零散小網格塊，為求解決此問題，將小網格塊中的網格面分別作十字相交，若相交出來的邊界編號與大塊網格塊的邊界編號相同，則合併至大塊網格塊裡。

當所有網格塊都有邊界編號時，就可以依據各網格區塊的邊界編號對 *FeatureFacetArray* 裡的每個網格面做十字相交，如圖 3.21(c)，並得到一組邊界編號，再利用得到的邊界編號跟所有網格塊邊界編號做比對，視哪組的網格塊邊界編號跟網格面的邊界編號最多相同的，即設定網格面為該網格塊。當 *FeatureFacetArray* 裡的所有網格面都完成測試，就可加入所屬於的網格塊裡，建構成完整的區塊，如圖 3.21(d) 及圖 3.21(e)。

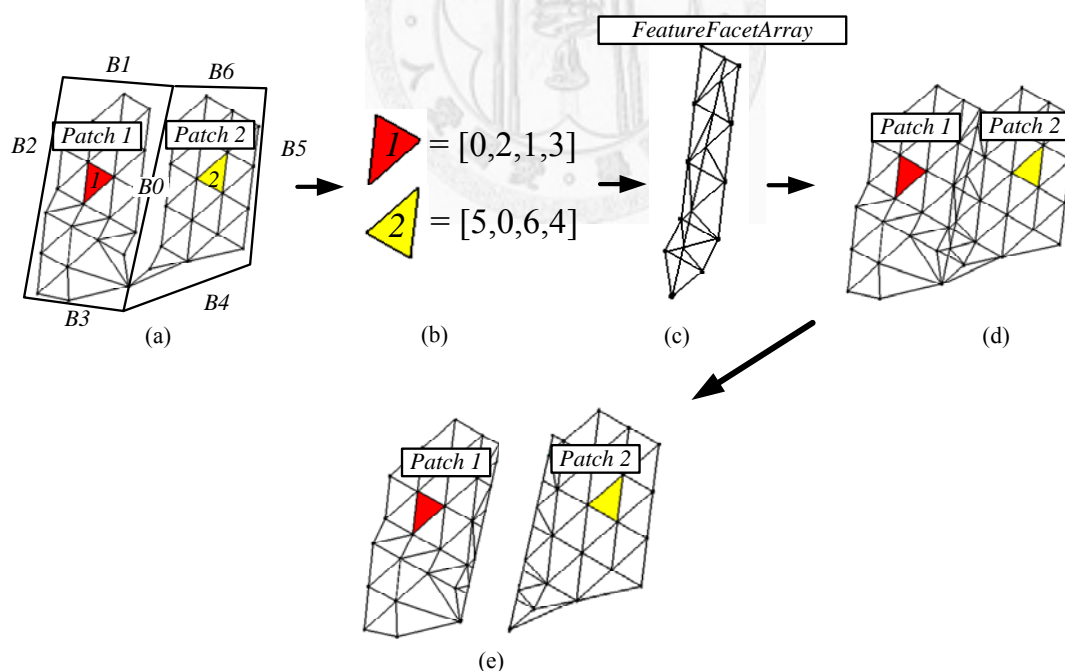


圖 3.21 建立完整區塊之圖

第四章 實例驗證

4.1 前言

為了驗證本研究的處理方式，本章節將實際測試 AM 鈹件廠商所提供的汽車鈹件網格資料，採用的範例檔案是引擎蓋鈹金件以及葉子鈹件。首先，讀取鈹件的 STL 網格檔案作為輸入，接著計算網格點的微分幾何性質，依使用者需求進行特徵線(凹特徵線、凸特徵線)萃取，透過特徵線分割與網格面分區域處理，最後對所有已分出的網格區域鋪面，並與其他市面上逆向工程軟體作比較。以下將針對本研究的網格分割以及網格分區域流程做成果說明。圖 4.1 為本研究的完整介面，包含 Menubar、ToolBar、Tree、Viewer 等。

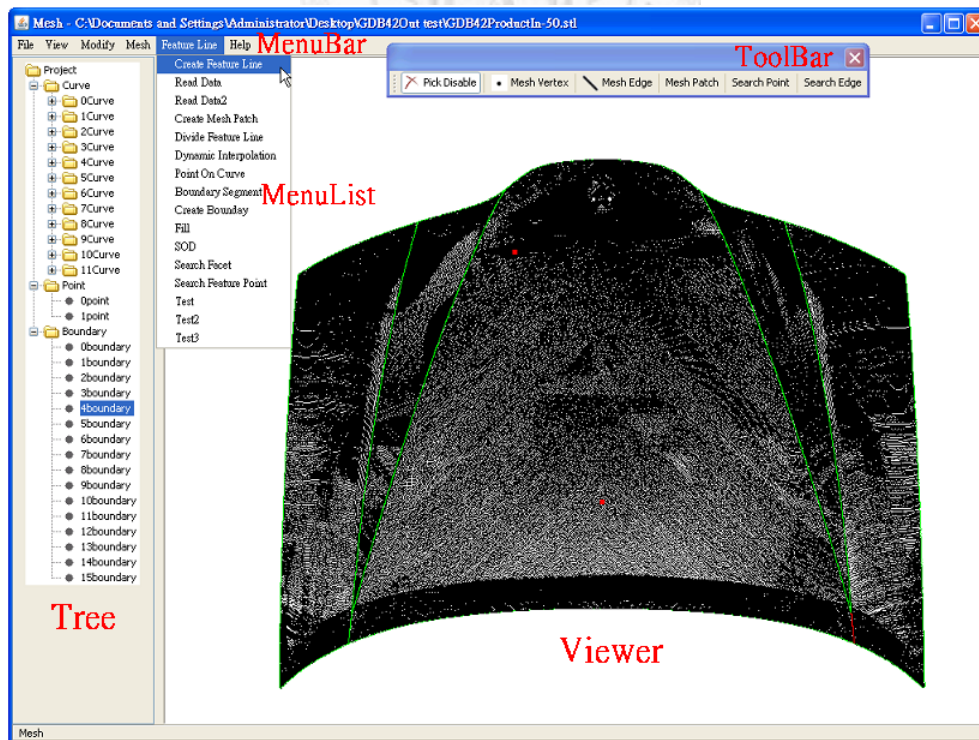


圖 4.1 完整介面

4.2 檔案讀取

本研究檔案輸入格式為 STL 網格資料，圖 4.2 為讀取檔案之介面。在檔案讀取的過程中，點、邊、面的拓樸關聯隨著半邊結構而建立。

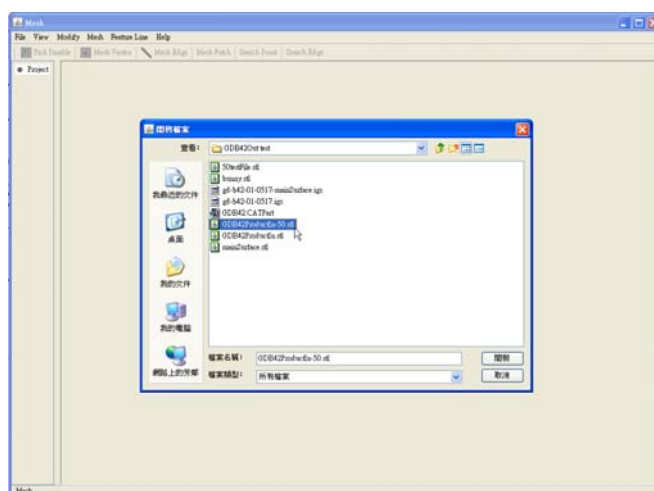
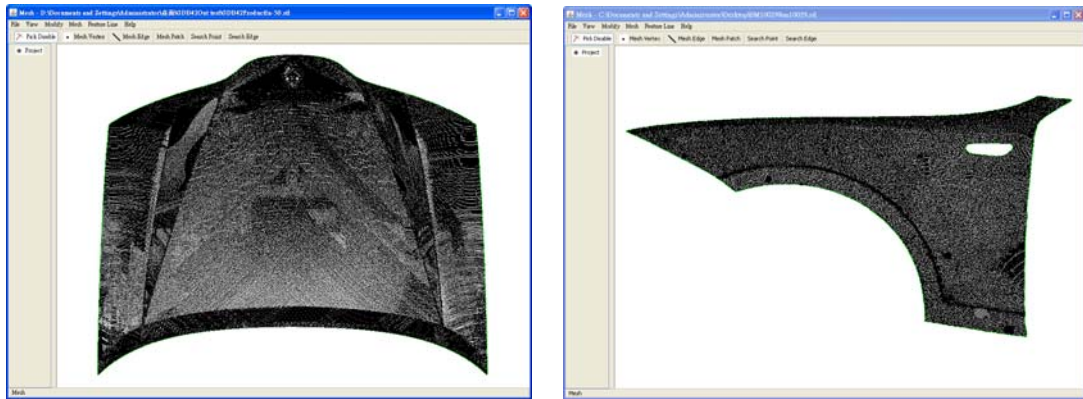


圖 4.2 讀檔介面

圖 4.3(a)為本研究實際測試的汽車引擎蓋外鈹網格資料，網格面數量為 654,666 面、網格頂點數量為 328,278 點、開檔時間為 9.86 秒。圖 4.3(b)為實際測試的葉子鈹網格資料，網格面數量為 326,261 面、網格頂點數量為 164,575、開檔時間為 5.02 秒。



(a)

(b)

圖 4.3 測試檔案網格資料

4.3 特徵線萃取與特徵線延伸

完成網格資料讀取之後，便可進行特徵線萃取，依照不完整的特徵線進行延伸流程，分別為：萃取特徵線與設定門檻值、特徵線延伸與點選特徵線以及整體網格邊的邊界點。

4.3.1 設定門檻值與特徵線萃取

在進行特徵線萃取之前，必須由使用者輸入一個自定的門檻值，而此門檻值主要是藉由網格點的最大曲率與最小曲率進行比對判斷。

首先，先利用預設門檻值 0.005，分別對引擎蓋外鈑及葉子鈑進行萃取特徵線，可得到如圖 4.4 所示的結果，觀察此圖，可發現預設之門檻值太過於敏感，在引擎蓋外鈑的上下兩邊都被誤判為特徵，而整個葉子鈑幾乎都被誤判為特徵。為求更完整的特徵線，在此提高門檻值的方式，將誤判的特徵線段過濾。如圖 4.5 所示，引擎蓋外鈑門檻值提

升至 0.05、葉子鈑門檻值提升至 0.07，即得到不錯的結果。

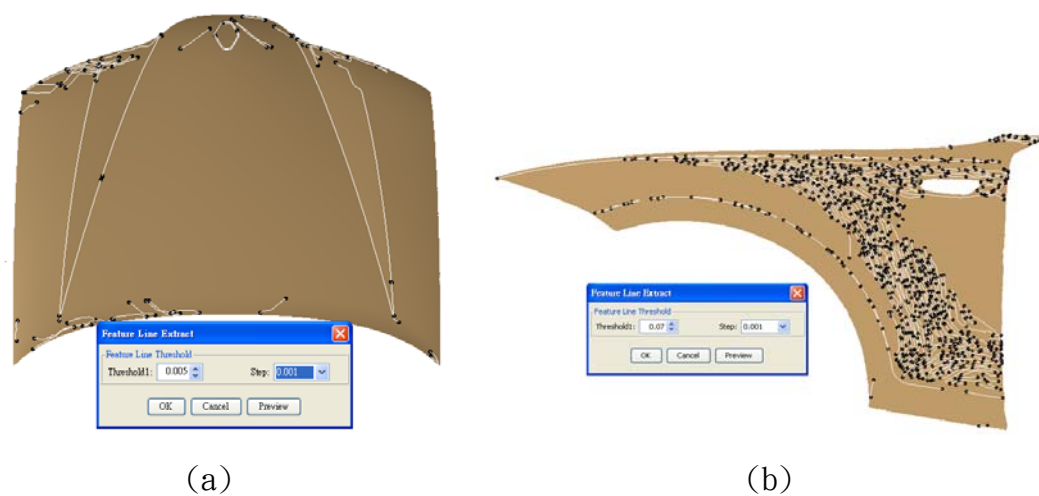


圖 4.4 (預設)引擎蓋外鈑及葉子鈑特徵線萃取

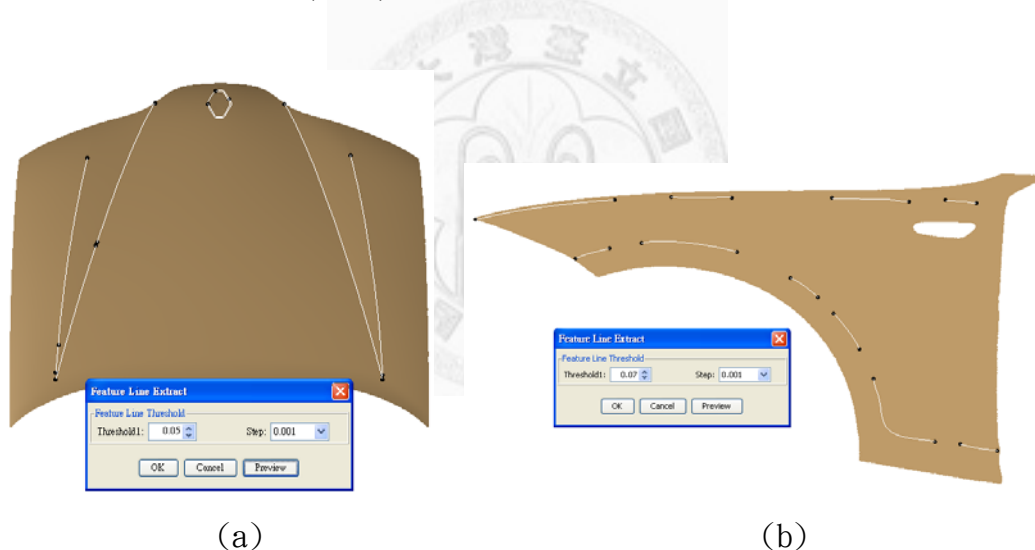
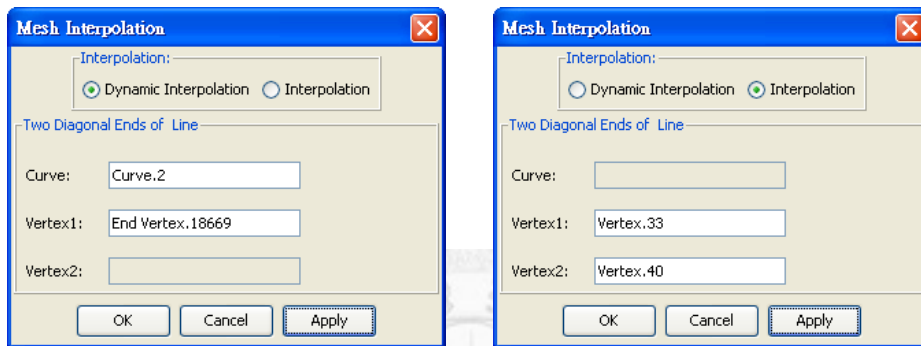


圖 4.5 (調整後)引擎蓋外鈑及葉子鈑特徵線萃取

4.3.2 點選特徵線及網格邊界點之特徵線延伸

當找到適合的門檻值後，接續進行特徵線的延伸處理。此功能主要是點選必須延伸的曲線以及延伸至網格邊的邊界點，如圖 4.6(a)，而圖 4.6(b)為相接兩條曲線並產生的線段。當這兩樣條件都建完畢，就可進行延伸如圖 4.7 所示。圖 4.7(a)延伸八次：四次為鈑件上方、四次

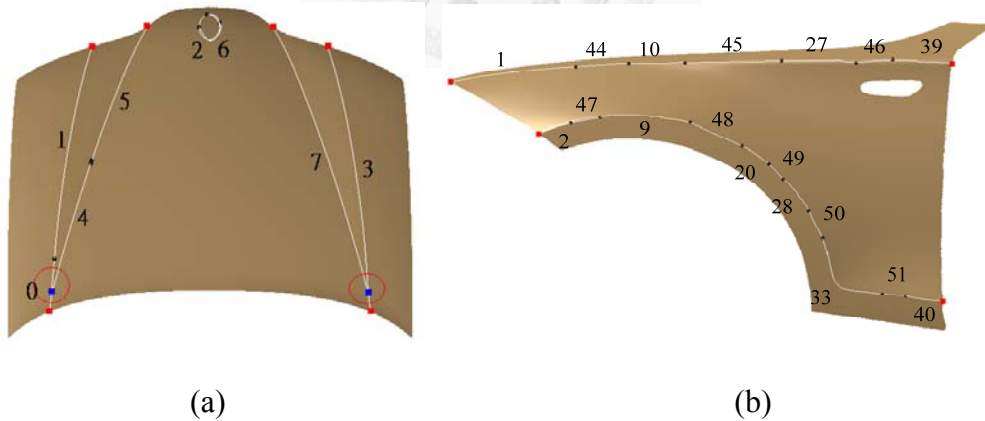
為鈹件的下方，而圖中的兩個紅色圈的延伸方法比較特別，因為該兩條特徵線並不是要延伸至網格邊界而是要延伸至別條特徵線，所以必須先在 0 與 3 特徵線先進行曲線上取點之操作，再進行特徵線 4 與 7 延伸至特徵線 0 與 3。圖 4.7(b)延伸至邊界四次、相接八次兩條曲線並產生八條新線段。



(a)

(b)

圖 4.6 特徵線延伸之介面



(a)

(b)

圖 4.7 特徵線延伸

圖 4.8 為曲線上取點之功能介面，此介面所需要的輸入資料為特徵線段，以及線段上的位置比例；介面上的 Add 是用來儲存此點資料且顯示在整體界面左邊的樹狀圖上；介面上的 Split 是用來將特徵線段依

標定點分割為兩條曲線，方便以後邊界設定。

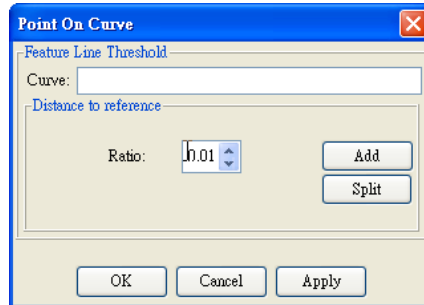


圖 4.8 曲線上取點之介面

4.4 整體網格邊界建立

本研究建立網格塊邊界資料，主要用意在於獲取網格塊之間的相鄰關係，便於後續建立網格區域的邊界以及零散網格塊合併。邊界可分為兩種：第一為內部網格邊界、第二為整體網格邊界，如圖 4.9 所示，藍色標號代表內部網格邊界編號，而紅色標號代表整體網格邊界編號，黃色點為邊界點。在建立內部網格邊界、整體網格邊界、網格區域邊界操作上，本研究利用三個不同介面來手動挑選邊界，如圖 4.10 至圖 4.12，三個介面都是手動挑選邊界或邊界點，挑選的順序不重要但需確定每次作業完成後，按下 Apply 遞送資料，確保儲存此邊界資料且顯示在介面左邊的樹狀圖上。

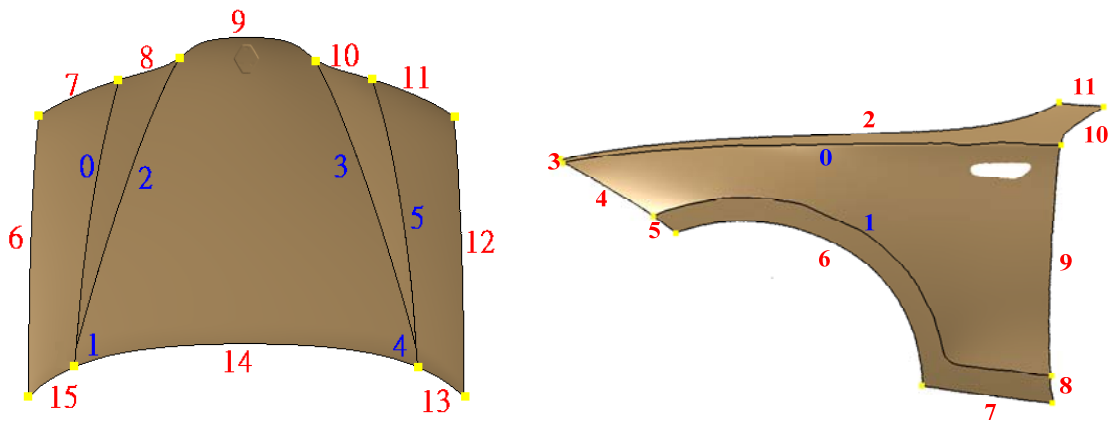


圖 4.9 整體的網格邊界

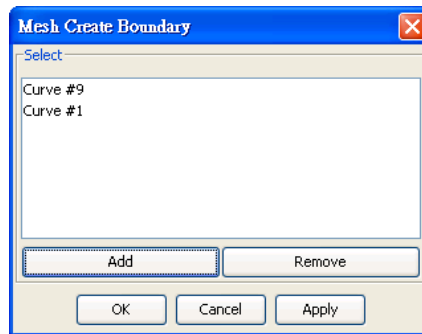


圖 4.10 手動內部邊界之介面

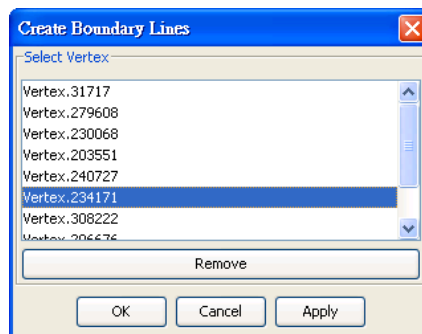


圖 4.11 手動點選邊界邊之介面

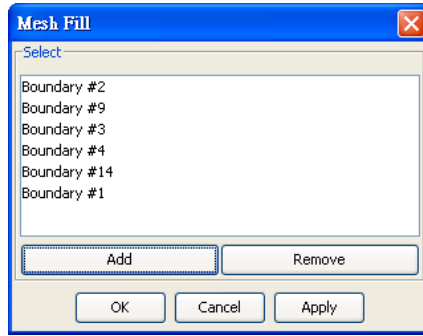


圖 4.12 手動網格塊邊界之介面

4.5 網格面分割與網格面分區域

完成以上功能後，即可自動對整體網格做分割與分區域動作，表 4.1 為網格面分割後的邊數量；圖 4.13 (a)為引擎蓋外鈑分群為五個小區域；圖 4.13 (b)葉子鈑分群為三個小區域網格，最後儲存為 STL 檔。

表 4.1 引擎蓋外鈑與葉子鈑的各網格面的邊數量

範例 邊數量	引擎蓋外鈑	葉子鈑
10	2	2
7	4377	3354
6	19	20
5	7	5
4	11	23

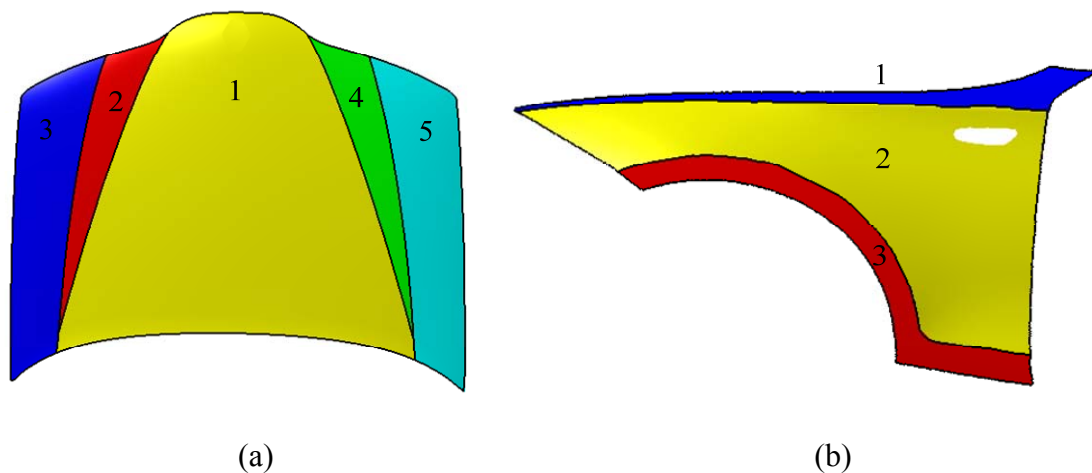


圖 4.13 儲存後以 CATIA 開檔實例

4.6 區域成長與特徵線萃取分群比較

網格分割與分群可藉由區域成長與特徵線萃取，兩種不同的特徵區域萃取法達成。區域成長在於曾[14]之研究已作相關呈現，而本研究著力於特徵線萃取法則實作。以下分別針對兩種不同法則進行概略說明並評比其優劣。

區域成長是一種藉由網格面擴展的方式，將相同屬性的相鄰網格面做結合的搜尋法則。在進行區域成長之前，每個網格點必須經過微分幾何性質的計算，其計算可參考[14]，獲得網格點資料的最大主軸曲率。對於特徵點的判斷主要藉由使用者所訂的門檻值，將網格點所屬於的區域分為特徵區域及平緩區域。若網格點的主軸曲率大於所訂的門檻值，則此點位於特徵區域，同時定義為特徵點，若網格點的主軸曲率小於所訂的門檻值，則此點位於平緩區域。完成網格點的特徵屬性判定以後，就可以進行特徵面選定，最後利用區域成長進行歸類，將特徵面與平緩面進行相鄰面同性質分群動作，取得獨立的網格塊。

以引擎蓋外鈹及葉子鈹為例，圖 4.14(a)為引擎蓋外鈹、圖 4.14(b)為葉子鈹，為利用區域成長分群之結果。

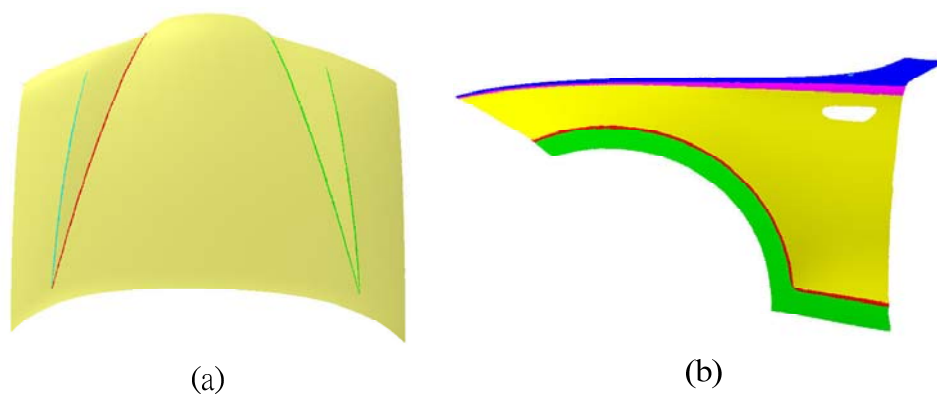


圖 4.14 區域成長分群之結果

區域成長分群速度快，但是此法則分群不佳，如圖 4.14(a)，此法則將引擎蓋外鈹分為四大網格區域：一個 v 型區塊、兩條細線區塊以及其餘鈹面一大區塊。於圖面觀之，細線區並不適以鋪面作業為後處理；不規則之大鈹面區塊亦不適直接鋪成單一曲面。另外此法則將葉子鈹分為五大網格區域，如圖 4.14(b)，其中兩個網格區域為兩條細長區塊。

特徵線萃取法是另外一種特徵區域萃取法，而本研究採用此方法來達成網格分割與分群。在進行特徵線萃取之前，必須計算每個點的微分幾何性質，其計算可參考 2.4.2 小節，獲得網格點資料的最大與最小主軸曲率。藉由使用者所訂的門檻值與主軸曲率的比較，定義凹與凸特徵點並串聯相同屬性的特徵點形成特徵線，其計算可參考 2.4.3 小節，對於稜邊特徵即可識別，並定義為網格群邊界。後續添增特徵線延伸及特徵線切分功能，取得封閉迴路之網格塊，得以適當的將網格

區域切分為完整區塊，如圖 4.13。於圖 4.13(a)與圖 4.14(a)比較引擎蓋外鈹實作成果。本研究實作成果，將此鈹金切分為五個大區塊，相較於曾[14]的區域成長具備較為完整區域特性，利於後續作業進行。於圖 4.13(b)與圖 4.14(b)比較葉子鈹實作成果。本研究實作成果，將此鈹金切分為三個大區塊，相較於曾[14]的區域成長切分為三個大區塊及兩個細長區塊。

4.7 網格點資料重建 B-spline 曲面

以下將利用林友瑜[13]的 B-spline 曲面重建法則及二次分割法則，針對引擎蓋外鈹的五筆網格區域進行曲面重建，利用 CATIA 對引擎蓋外鈹進行裁切曲面與原始網格資料誤差比對。圖 4.15 顯示五筆網格區域的裁切曲面與原始網格資料誤差。此圖顯示在靠近網格區域邊界處產生較大的誤差，但整體而言，仍然有 93%以上的點資料所對應的曲面誤差控制在 $\pm 0.30\text{mm}$ 的精度範圍之內，而 $\pm 0.30\text{mm}$ 則是汽車鈹件對於誤差精度的要求。若要整體網格資料都在 $\pm 0.30\text{mm}$ 的精度範圍之內就必須對所有曲面進行二次分割，在曲率變化較大的地方將一塊網格區塊分為兩塊網格區塊，如圖 4.16 將引擎蓋外鈹分割為 11 塊區域。整體引擎蓋外鈹之最大誤差為 0.277 mm、平均誤差 0.025 mm、誤差最大處發生於 *I* 與 *J* 的共邊。

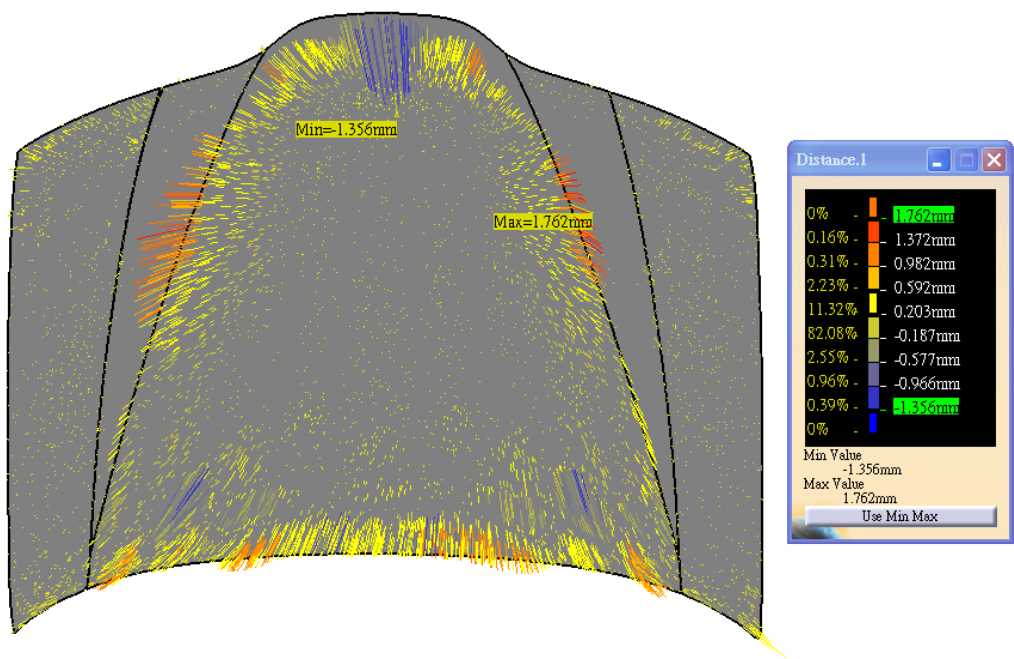


圖 4.15 二次分割前之引擎蓋外鈹的網格與裁切曲面誤差比對

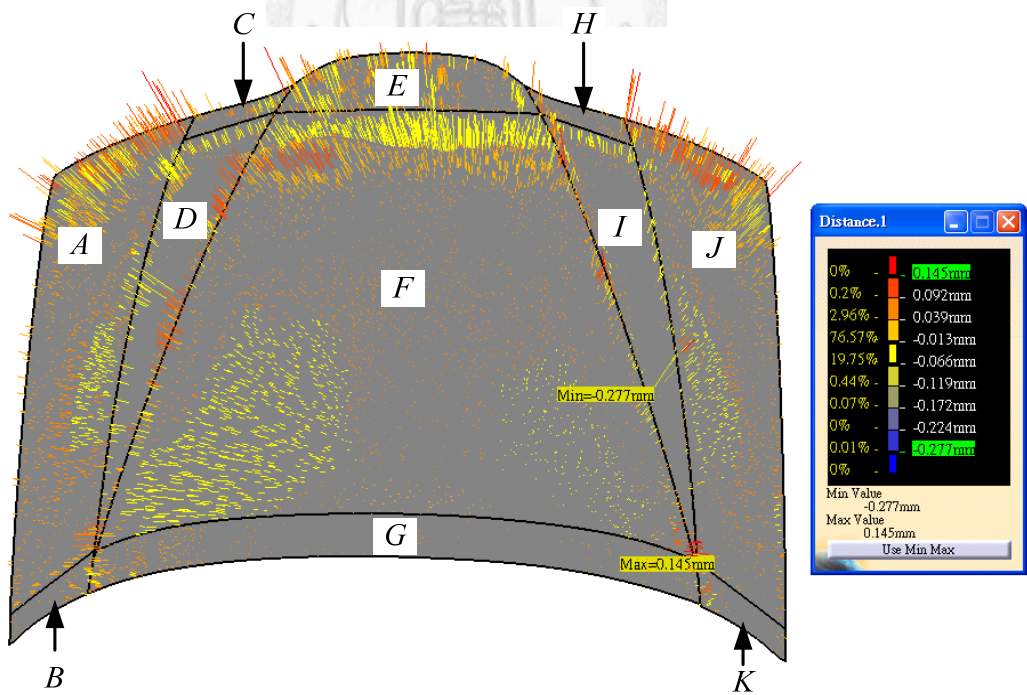


圖 4.16 二次分割後之引擎蓋外鈹的網格與裁切曲面誤差比對

以下將針對葉子鈹三筆網格區域進行曲面重建，並以 CATIA 對葉子鈹進行裁切曲面與原始網格資料誤差比對。圖 4.17 顯示三筆網格區域的裁切曲面與原始網格資料誤差。此圖一樣顯示在靠近網格區域邊界處產生較大的誤差，但整體而言，仍然有 89% 以上的點資料符合工業界對於汽車鈹件的精度要求。若要提升裁切曲面與原始網格資料誤差比例，就必須對此區域進行二次分割，結果如圖 4.18 分割為 13 塊區域。此圖顯示在靠近網格區域邊界處產生較大的誤差，但整體而言，有 98% 以上的點資料所對應的曲面誤差控制在 $\pm 0.30\text{mm}$ 的精度範圍之內。整體引擎蓋外鈹之最大誤差為 1.255 mm、平均誤差 0.229 mm、誤差最大處發生於 B 與 G 的共邊。

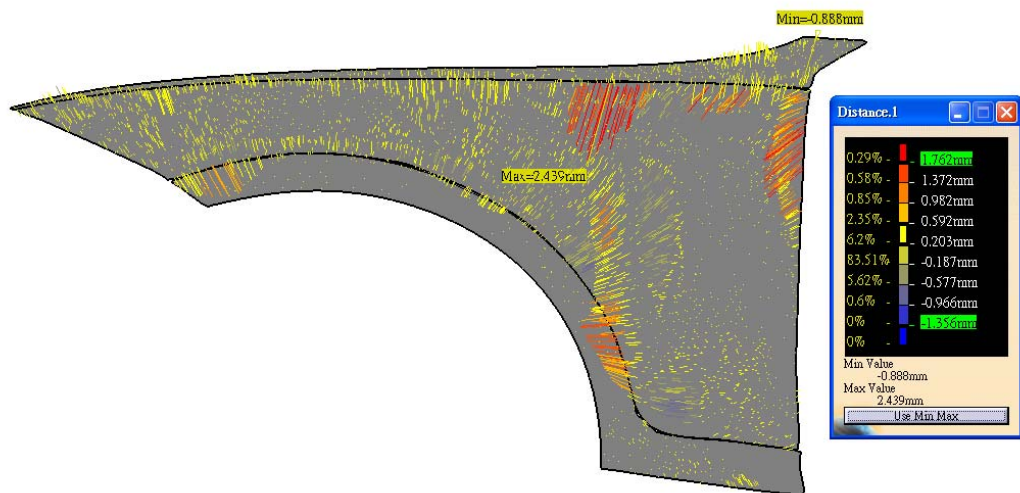


圖 4.17 二次分割前之葉子鈹的網格與裁切曲面誤差比對

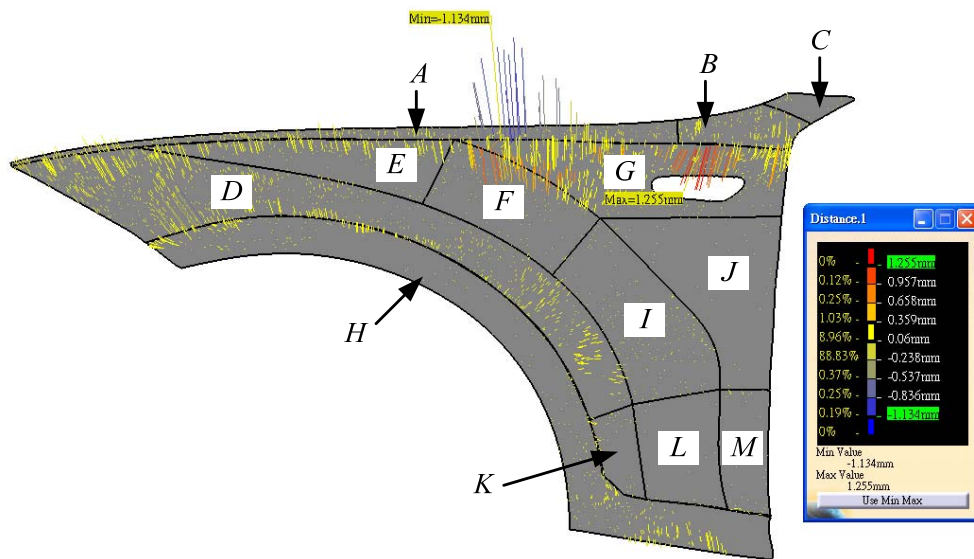


圖 4.18 二次分割後之葉子板的網格與裁切曲面誤差比對

4.8 雜線的串聯

特徵線是以曲率變化產生特徵點，再將點連接即成為特徵線，但是當網格資料有特徵區域時，產生的特徵線段顯得複雜，因此必須進行線段過濾或串聯。以葉子板為範例，由圖 4.19 所示，線段的串聯分為三部分。

(1) 雜線段分群

雜線段分群是以特徵區域為依據，如圖 4.19(a)所示。此圖有兩個特徵區域，分別為葉子板的上側與下側。將上側所有線段及下側所有線段分別儲存，並可進行下部分動作。

(2) 點選轉折點與線段分群

當線段群分好後，必須在曲率變化較大之處進行人工點選轉折點。轉折點的用意為在建立趨勢曲線時，確保線段的結果不會

偏離原始鈹件的鄰線太多。點選轉折點之後，則進行分點群。首先，先將每個雜線段群的頭尾兩點以及轉折點建立垂直鈹件的平面，然後以每點的 X 座標做為依據進行由負到正的點排序，如圖 4.19(b)所示。葉子鈹上側因為沒較大曲率變化，只有頭尾兩個平面，並產生一群的點群。葉子鈹下側有兩個轉折點，在進行點分群，則分出三群點群，分別由平面三至平面四為第一點群、平面四至平面五為第二點群、最後平面五至平面六為第三點群。

(3) 建立趨勢曲線

當所有點群完成排序後，利用最小平方法建立近似曲線，將每個點群建立平順的線段，如圖 4.19(c)所示。



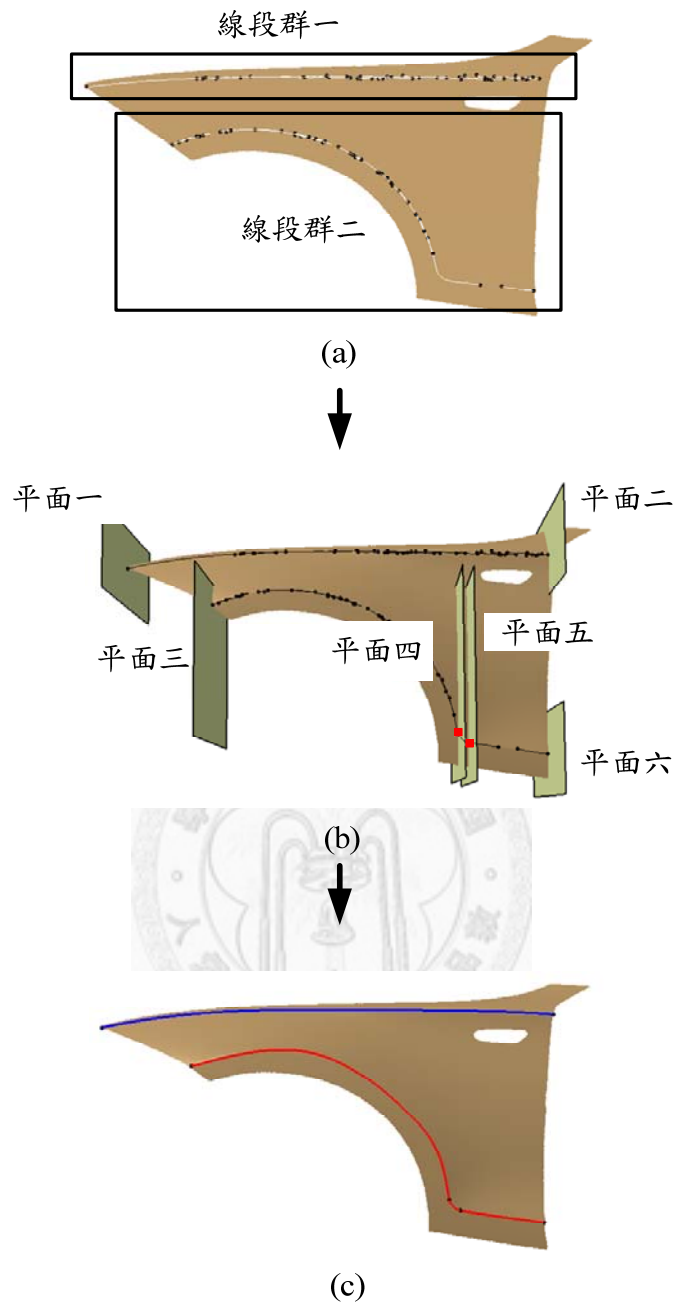


圖 4.19 雜線的串聯示意圖

4.9 門檻值的判斷

三角網格分區塊的成功與否的關鍵在於是否可以準確的設定特徵線萃取的門檻值，選定適合的門檻值，不僅方便執行網格分割區域的程序，還可以減少一些不必要的處理與時間，如過濾雜線。每個

鈹件都必須經過多次的測試才能取得適當的門檻值，本論文針對引擎蓋外鈹與葉子鈹進行不斷的測試，最後找出適合的門檻值並進行特徵線萃取。

引擎蓋外鈹的門檻值設定為 0.05，如圖 4.5(a)所示，可完整的顯示該鈹件的所有特徵，並分出五個網格區域。對葉子鈹而言，雖然適合門檻值為 0.07，如圖 4.5(b)所示，經過線段的過濾，過濾掉一些過短或是錯誤線段，剩下一些零碎線段，必須透過一些處理才能分出三個網格區域。



第五章 結論與未來展望

5.1 前言

本研究以大量三角網格資料為輸入端，在龐大的三角網格中，依據 Yoshizawa[11]等人所提出的特徵線萃取演算法，並以局部的修正，對網格資料設定門檻值，以進行特徵線的萃取，後續將特徵線作延伸並且取得封閉特徵線，以作為網格分割的依據，加速逆向曲面重建的過程。以下將針對整體研究結果進行總結，並提出未來的展望，以供後續研究參考。

5.2 研究結論

綜合以上各章節，結論分成三大部分進行討論，第一部分為分別針對文獻中的特徵線萃取演算法進行結果比較並選定本論文的主要特徵萃取演算法。第二部份主要針對不完整的特徵線進行延伸以及對每個特徵網格面進行分割，確保所有網格為三邊之總結。第三部份以多條封閉特徵線所包圍的區域透過區域成長的擴展進行網格分群。整體研究成果分述如下：

- (1) 以汽車鈑件之引擎蓋外鈑與葉子鈑為範例，本文採用特徵線萃取演算法，並修正特徵線後處理，使得線段數量減少，並合併小線段為較長線段。經過特徵萃取演算法的比較，可以發現 RidgeAndRavine 之演算法是所有效果最佳者。

- (2) 對於線段延伸，本文提出一些工具搭配網格面分割，使得特徵線段不會受到較大曲率變化的影響，結果偏離預期路徑，造成鋸齒狀線段。延伸線段是透過不斷的投影及直線相交取得特徵點直到接觸到網格邊緣。然後套用本文所提出的三角網格分割法則，此法則是以邊資料作為依據來分割所有特徵面，將不為三邊的網格分割為三邊並建立新網格以及重整拓樸資料。
- (3) 採用網格面區域成長，並作局部修正，取得初步的網格塊，再透過十字相交取得每個網格塊的邊編號，然後將 *FeatureFacetArray* 容器裡的所有網格分別進行十字相交，將所得到的邊編號與每塊區域的邊編號進行比對，再將容器裡的網格分別加入所屬的網格塊裡，並取得數個完整區塊。

透過實例驗證，本文已成功提出一套利用特徵線辨識處理網格分割與分群程序，並應用於汽車外觀鈹件上，所有線段透過特徵網格面分割及延伸法則，線段得以平順呈現，最後取得完整區塊。

5.3 未來展望

本節對於可探討及發展之工作說明如下：

- (1) 依主軸曲率的特徵分離可以得到較佳的分離結果，但是有時會出現些雜線或是誤判及所計算時間通常較為長，如何改善且維持一定的精度是未來的目標。

- (2) 為解決特徵門檻值需要手動調整的麻煩，故需將網格模型加以歸類，以便後續應用，例如輸入檔案為引擎蓋外鈑，判斷是否可利用之前所測試的鈑件資料主動判斷適合的門檻值，使特徵線萃取能成為自動化。
- (3) 進行特徵線延伸之前，先對整體網格做前處理，這可解決曲線及曲線相交的誤差問題，使取得特徵點更加容易。
- (4) 為使網格特徵面分割可實際對所有鈑件作分割，尚需增加多特徵線段分割原始網格之案例，如三條、四條等特徵線段分割原始網格產生更多不同的案例。



參考文獻

- [1] Gelfand N, LJ Gubas. 2004. Shape segmentation using local slippage analysis. *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, 214-223.
- [2] Goldfeather J, V Interrante. 2004. A novel cubic-order algorithm for approximating principal direction vectors. *ACM Transactions on Graphics* 23(1):45-63.
- [3] Hubeli A, Gross M. 2001. Multiresolution feature extraction for unstructured meshes. *Proceedings of the Conference on Visualization*, 287-294.
- [4] Kim HC, SM Hur, SH Lee. 2002. Segmentation of the measured point data in reverse engineering. *International Journal of Advanced Manufacturing Technology* 20(8):571-580.
- [5] Piegl LA, Tiller W. 1997. *The NURBS book*, 2nd edition. Springer.
- [6] Razdan A, MS Bae. 2003. A hybrid approach to feature segmentation of triangle meshes. *Computer-Aided Design* 35(9):783-789.
- [7] Rossel C, Kobbelt L, Seidel HP. 2000. Extraction of feature lines on triangulated surfaces using morphological operators. *In Smart Graphics, Proceedings of the 2000 AAAI Symposium*, 71-75.
- [8] Szilvasi-Nagy M, Matyasi G. 2003. Analysis of STL files. *Mathematical and Computer Modeling* 38(7): 945-960.
- [9] Thirion JP. 1996. The extremal mesh and the understanding of 3D surfaces. *International Journal of Computer Vision* 19(2):115-128.

- [10]Thorpe J.A. 1979. *Elementary Topics in Differential Geometry*, Springer-Verlag, 53-92.
- [11]Yoshizawa S, Belyaev A, Seidel HP. 2005. Fast and robust detection of crest lines on meshes. *Proceedings of ACM Symposium on Solid and Physical Modeling*, 227-232.
- [12]尤春風，Spring Solid System 實體模型系統，國立臺灣大學機械工程研究所實體模型實驗室。
- [13]林友瑜，2010，汽車鈹件之三角網格處理與 B-spline 曲面重建，國立台灣大學機械工程學研究所碩士論文。
- [14]曾凱威，2008，三角網格分割與曲面重建於汽車鈹件之應用，國立台灣大學機械工程學研究所碩士論文。
- [15]楊偉倫，2007，三角網格錯誤偵測與特徵線萃取，國立台灣大學機械工程學研究所碩士論文。
- [16]劉勝蘭、周儒榮、張麗豔，2003，三角網格模型特徵線提取，*計算機輔助設計與圖形學學報* 15(4): 444-448。

作者簡歷

姓名：李康豪

生日：中華民國 74 年 7 月 19 日

學歷：碩士 國立臺灣大學機械工程研究所 2008~2010

學士 國立臺灣大學機械工程學系 2003~2008

高中 West Point Grey Academy 高級中學 2000~2003

國中 West Point Grey Academy 中學 1997~2000

國小 Edith Cavell 小學 1991~1997

住址：台北縣蘆洲市永安南路一段一巷一弄十一號七樓

電話：(02)2983-8808

電子郵件：R97522617@ntu.edu.tw

