

滑動葉片旋轉式壓縮機之研究與發展
(Ⅲ)—滑動葉片旋轉式壓縮機之電腦
輔助設計系統研究
Research about the Computer-Aided
Design System of the Sliding Blade
Rotary Compressor

計畫編號：NSC 88-2212-E-002-039
執行期間：87年8月1日至88年7
月31日
主持人:陳漢明 副教授

一、前言：

本子計畫『滑動葉片旋轉式壓縮機之電腦輔助系統研究』的目的是要將電腦輔助設計的原理實際應用在產品的開發上，減少傳統設計方式中設計生產力低、前置時間長、修改設計麻煩、雛形機測試等的缺點，同時增加設計的準確度，使設計更具標準化，提供較佳的產品。本研究主要的內容是滑動葉片旋轉式壓縮機的電腦繪圖動態模擬展示，其中包含了在電腦繪圖中關於隱面消除及多邊形切割法則等問題之研究。

本計劃延續第一年及第二年之計劃，為了能使整個動態模擬過程能更迅速的顯現出來，本計劃今年將重點放在任意多邊形三角化方法之研究，以達到節省電腦運算時間的目的，這將對整個壓縮機動態模擬有相當大的助益。

二、研究目的及方法：

在電腦繪圖學中，多邊形的切割幾乎是無法避免的事，傳統的多邊形切割法則在切割凹多邊形後沒有將切割出的多邊形獨立分開而發生錯誤，而對於此種情形的解決的方法有兩種，一為改進傳統的多邊形切割法則；另一為將需切割的多邊形變成凸多邊形，三角形必為凸多邊形，所以將多邊形三角形化就可以解決這個問題。

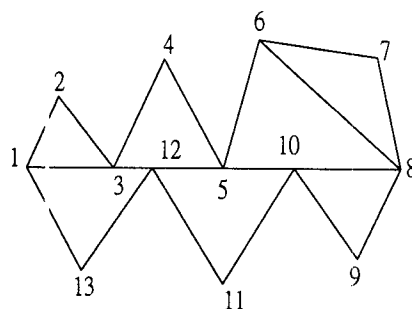
本文將先介紹幾個多邊形三角

形化的法則，針對要使用的多邊形基本性質做說明，並對在多邊形三角形化過程中會遇到的問題提出解決方法，然後介紹我們發展的簡單多邊形三角形化法則，它保證三角形化的三角形數為最少。

最後，我們探討非簡單多邊形的三角形化，並將選取三種形態的非簡單多邊形為例子，將之變成多個簡單多邊形的組合，再利用前面的簡單多邊形三角形化的法則，將之三角形化。

2.1 簡單多邊形的三角形化：

假設 $v_0, v_1, v_2, \dots, v_{n-1}$ 是在一個平面上的 n 個點，而這 n 個點依序循環排列，而 v_0 是排在 v_{n-1} 之後。現在讓 $e_0 = v_0v_1, e_1 = v_1v_2, e_2 = v_2v_3, \dots, e_i = v_iv_{i+1}, \dots, e_{n-1} = v_{n-1}v_0$ ，是連接這些點的 n 個線段。若每一對相鄰線段的交集只有二者共用的頂點，且不相鄰的線段之間並沒有任何交集，若滿足以上兩個條件則為一個簡單多邊形。在一個簡單多邊形中，若不相鄰的兩點所連的線段 $e_k = v_iv_j$ ，(1)與多邊形邊界的交點除了 v_i 與 v_j 之外沒有其它交點。(2)線段 e_k 位於多邊形的內部。則我們稱 e_k 為多邊形之一個對角線。而在我們的研究中，對對角線的定義增加一項不可相交，以方便我們三角化的步驟。將一個具有 n 個頂點的多邊形三角形化，在一般情形下，最少需要 $n-3$ 條對角線，而三角形的數目最少是 $n-2$ 個。而是否一個具有 n 個頂點的多邊形三角化之後最少的三角形數目一定是 $(n-2)$? 答案是否定的，在出現頂點共線的



在平面上1、3、12、5、10、8為
共線的頂點

情形下，我們可以用更少的三角形數目分割原多邊形。以下為我們計算含有頂點共線多邊形的三角化法則：

$$n + 2 * (m - 1) - \sum_{i=1}^m k_i$$

```

graph TD
    Start([i=1, 多邊形數目m=1]) --> CheckI[檢查 i <= m]
    CheckI -- No --> End([結束])
    CheckI -- Yes --> FindConvex[尋找多邊形 i 一組共線頂點，  
將找到的點存入點序列中]
    FindConvex -- Yes --> SortX[依 X 座標將頂點由左  
至右排序 V1, V2, ..., Vk]
    FindConvex -- No --> IncI[i=i+1]
    SortX --> K2[k=2]
    K2 --> CheckCollinear[檢查共線頂點  
X 座標是否相等]
    CheckCollinear -- No --> SortY[依 Y 座標將頂點由下  
至上排序 V1, V2, ..., Vk]
    CheckCollinear -- Yes --> SortY
    SortY --> J1[j=1]
    J1 --> JK[k-1]
    JK -- Yes --> CalcD[線段 d = vjvj+1]
    CalcD --> CheckD[檢查 d 是否為對角線]
    CheckD -- No --> IncJ[j=j+1]
    CheckD -- Yes --> Split[將多邊形分割成兩個子  
多邊形，m=m+1]
    Split --> IncI
    IncJ --> JK
  
```

個步驟，就可以將這個角變成凸角。
綜合以上的法則，所以當我們在處理
簡單多邊形時，步驟如下：

- ```

graph TD
 Start([開始]) --> ReadN[將多邊形的四角都找出來，
並存入串列V中，假設有n個]
 ReadN --> SetN[n=0]
 SetN --> SetI[i=1]
 SetI --> LoopCond{i <= n}
 LoopCond -- 否 --> End([結束])
 LoopCond -- 是 --> CheckBack[檢查與後第二點
可否為對角線]
 CheckBack -- 否 --> SetI
 CheckBack -- 是 --> CheckFront[檢查與前第二點
可否為對角線]
 CheckFront -- 否 --> SetI
 CheckFront -- 是 --> RemoveFront[將點i前第一點移除
並記錄三角形]
 RemoveFront --> CheckAngle[檢查∠i是否仍
為四角]
 CheckAngle -- 否 --> SetI
 CheckAngle -- 是 --> RemoveBack[將點i後第一點移
除並記錄三角形]
 RemoveBack --> SetI

```

2

- 4-5，P3:5-6-7 及 P4:7-8-9-10。
- (2)檢查子多邊形是否存在共線頂點，已經無共線頂點，進行下一步驟。
- (3)P1 為凹多邊形，故要去除凹角。
- 1.尋找多邊形的凹角， $\angle 12$ 、 $\angle 13$ 、 $\angle 17$  為凹角，將這 3 個凹角存入串列 V 中。
  - 2.先處理 $\angle 12$ ，段線 12-10 可為對角線，得三角形 T1，將點 11 自 P1 點串列中移除。
  3. $\angle 12$  仍為凹角但無法繼續處理，故先處理 $\angle 13$ ，線段 13-15 可為對角線，得三角形 T2， $\angle 13$  成凸角，將 $\angle 13$  自串列 V 中移除。
  - 4.處理下一個凹角 $\angle 17$ ，線段 17-2 為對角線，得三角形 T5，將點 1 自 P1 點串列中移除， $\angle 17$  仍為凹角，線段 17-10 為對角線，得三角形 T6，將點 2 自 P1 點串列中移除， $\angle 17$  成為凸角，將 $\angle 17$  自串列 V 中移除。
  5. $\angle 12$  為最後一個凹角，串列 V 是仍有凹角 $\angle 12$  存在，線段 12-15 可為對角線，得三角形 T3，將點 13 自 P1 點串列中移除， $\angle 12$  仍為凹角，線段 12-16 為對角線，得三角形 T4，將點 15 自 P1 點串列中移

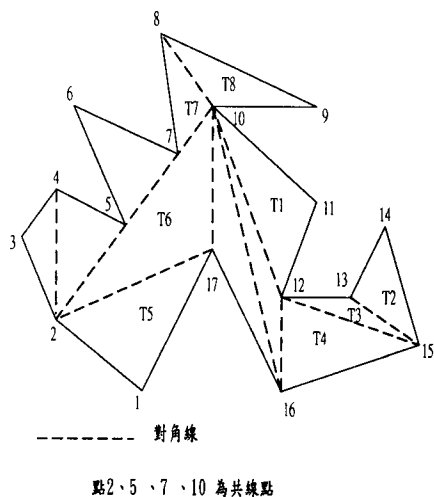


圖 2.2

除， $\angle 12$  成為凸角，將 $\angle 12$  自串列 V 中移除，串列 V 已無任何凹角。

- (5)多邊形 P2 及 P3 為凸多邊形，P4 有凹角存在， $\angle 10$  為凹角，線段 10-8 為對角線，得兩三角形 T7 和 T8。
- (6)再將凸多邊形三角化。

## 2.2 非簡單多邊形的三角形化：

不是簡單多邊形的多邊形就是非簡單多邊形，其實大部分的非簡單多邊形都可簡化成由簡單多邊形所組成。非簡單多邊形的種類非常的多，有四個以上的邊共頂點的情形，有邊相交的情形，有在多邊形內出現內孔和邊重疊的情形等等，因非簡單多邊形的種類太多，所以我們只討論以上的三種類型。

### 1. 四個以上的邊共頂點：

尋找相同的頂點，並以相同的頂點將原多邊形分割成兩個以上的子多邊形。如果無四個以上的邊共頂點，檢查下一個子多邊形，直到所有的子多邊形都無四個以上的邊共頂點，使其成為簡單多邊形。

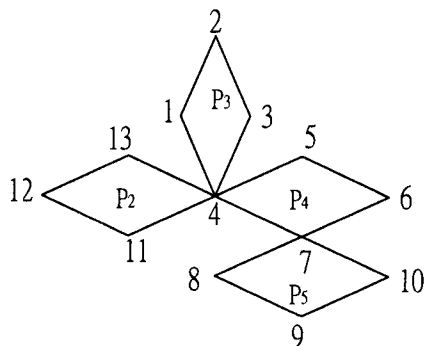


圖 2.3：四個以上的邊共頂點

### 2. 邊相交：

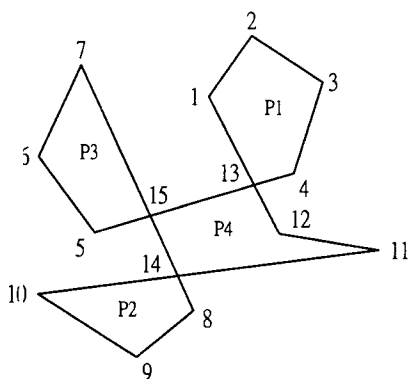


圖 2.4 邊相交：

只要將邊相交的頂點求出並加入多邊形的點串列中，就可將之視為四個以上的邊共頂點的非簡單多邊形。再用處理四個以上的邊共頂點的方式，使其成為簡單多邊形。

### 3. 內孔及邊重疊：

有內孔之多邊形的邊會發生重疊的情形，如重疊線段(走出多邊形)的前一頂點與後任一個頂點形成一對角線，則這條對角線與重疊邊上的兩頂點就可形成一新多邊形，而將這個內孔去除。

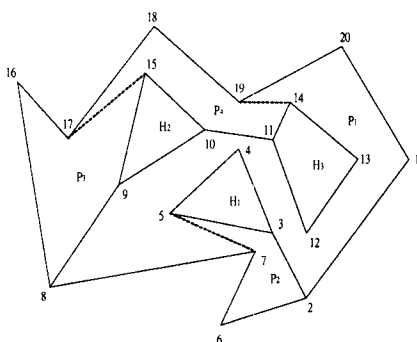
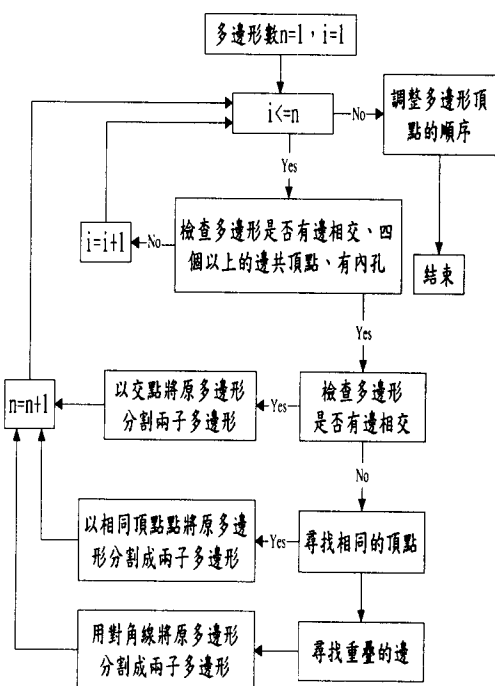


圖 2.5 有內孔之多邊形

將這些非簡單多邊形變成簡單多邊形，即可以用簡單多邊形的法則將之三角化。以得到最少的三角形。以下為非簡單多邊形三角化的流程圖：



### 三、討論：

就最佳化的觀點來看多邊形三角化，多邊形三角化的改進空間還是很大的，我們應該避免三角形的內角的角度過大或過小，使切割出來的三角形能盡量的均勻。

對於非簡單多邊形的切割方面，本文僅討論了三種類型的非簡單多邊形三角化，可能尚存在其他類型的非簡單多邊形，所以三角化的方法在非簡單多邊形方面還有很多的討論空間。

### 四、參考文獻：

- [Berg97], Mark de Berg, Marc vans Kreveld and Mark Overbears. *Computational Geometry Algorithms and Applications*, Springer-Verlag, Berlin Heidelberg, 1997. (Chapter 3: Polygon Triangulation.)
- [Bois93], Jean-Daniel Boissonnat, *Algorithmic Geometry*, Cambridge University Press, 1998 (Chapter 12 : Triangulation in Dimension 2)
- [Clar89], Kenneth L. Clarkson, Robert E Tarjan, and Christopher J. Van Wyk. "A Fast Las Vegas Algorithm for Triangulating a Simple polygon". *Discrete Computer Geometry*, Vol. 4, 1989, pp.423-432.
- 劉致宏，物件排序法則及應用，國立臺灣大學機械工程學研究所，碩士論文，中華民國 86 年 6 月。
- 張志強，多邊形之最佳三角化，國立臺灣大學機械工程學研究所，碩士論文，中華民國 88 年 6 月。