

國立臺北大學電機工程學系
104 學年度下學期學生專題製作報告

Convex Hull

組員：

學號：410287003 姓名：李昆翰

學號：410287015 姓名：林佑宣

指導老師：詹景裕 老師

中 華 民 國 1 0 5 年 6 月 8 日

一、摘要:

在多維空間中散佈著許多點，Convex Hull 為包覆所有點的外殼當中，外殼為凸的且表面積為最小的，因此又譯為凸包。

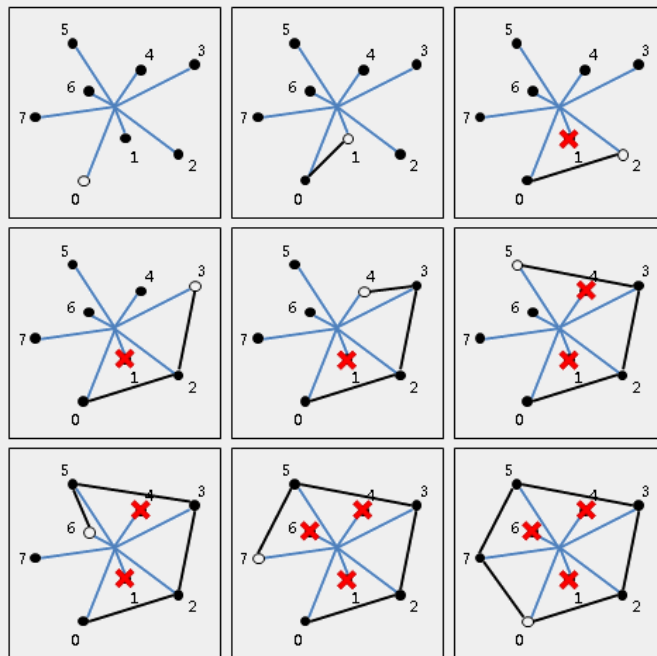
二、簡介:

Convex Hull 常被應用在圖像處理、模式識別及地理信息系統等方面，因為 Convex Hull 的”凸”，使我們可以輕易判斷一個點是否包含在其內部，而一般的多邊形判斷則須先作一射線，並繞多邊形每條邊來求出交點個數，來判斷是否位於多邊形內部，此種方法時間複雜度是較大的。

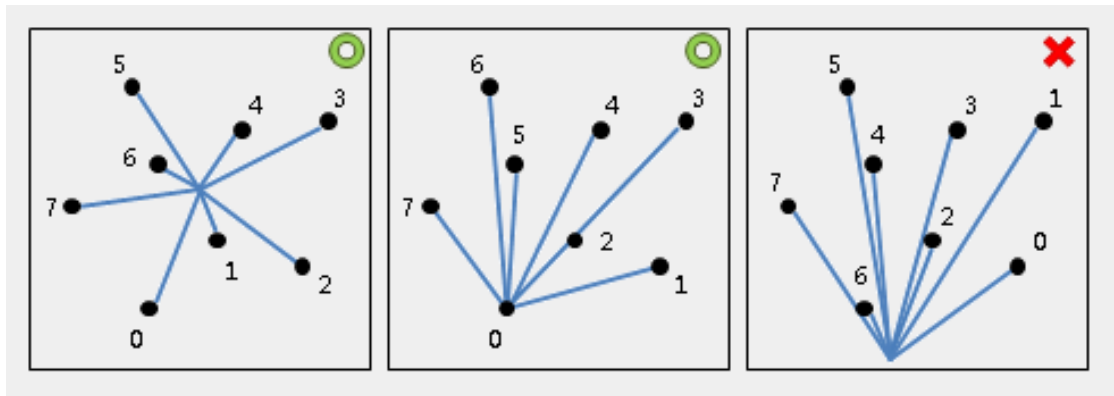
三、理論:

凸包常見演算法：Graham's Scan

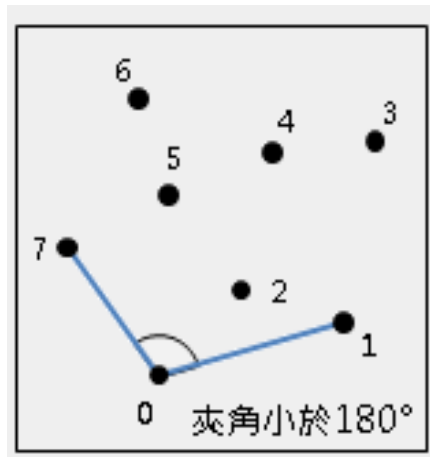
- 嘗試先將所有點依照時針順序排好，再來做繞行一圈的動作，繞行途中順便掃除凸包內部的點，如此就不必以窮舉所有點的方式來尋找最外圍的點。



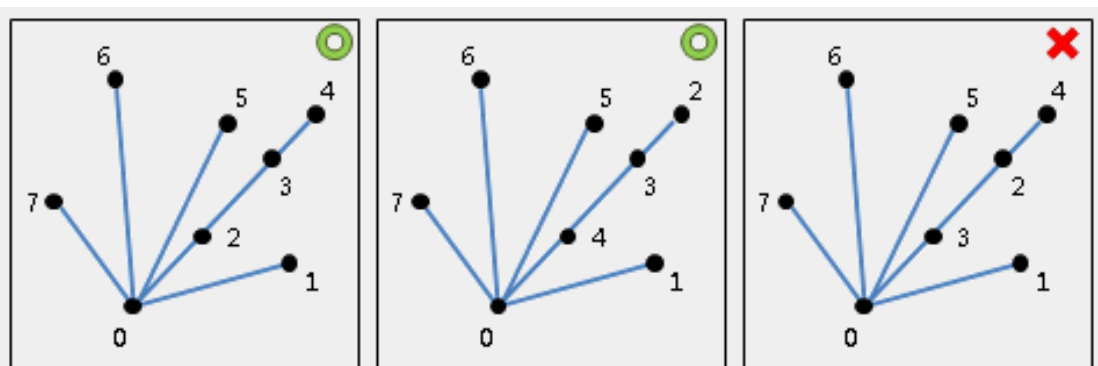
- 要讓所有點依照時針順序排好，只要將中心點設定在凸包內部或者凸包上，然後讓各點依照角度排序即可。如果把中心點設定在凸包外部，結果就不見得是時針順序了。包覆時必須按照時針順序，才能確保結果正確。



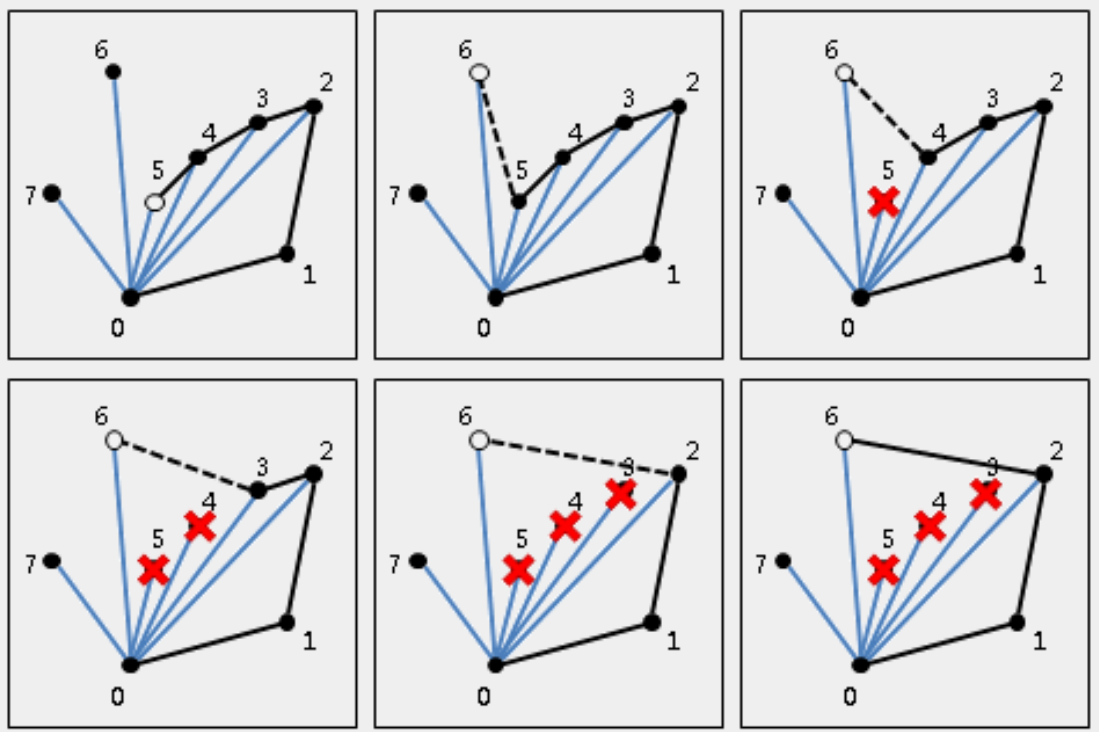
- 一般來說，選擇凸包上的頂點當作中心點是比較好的，因為角度排序時的夾角皆小於 180° ，可以使用叉積運算來排序（大於 180° 叉積得負值、等於 180° 叉積等於零，導致排序錯誤）。另一個好處是，中心點也可以作為包覆的起點。



- 角度排序時，遇到角度相同的情況，要小心排序。通常是讓距離中心點較近的点排前面。也可以排後面，但是不能亂排。



- 掃除的過程當中，經常株連許多點。使用 stack 資料結構來儲存凸包，逐一判斷 stack 頂端的點，逐一彈出凹陷的點。凹陷的點必定不是凸包上的頂點。



四、編寫步驟:

專題的演算法-1

1. 分別找出最上、最下、最左及最右點。
2. 先處理 S1，以 p1p4 連線為底，分別找出最上、最左及最右點形成一矩形，左右邊線上若多點共線選擇較靠外側點，上邊多點全部選擇。
3. 以相同方式依次處理 S2、S3、S4
4. 檢查是否有凹處
5. 檢查剩餘點數，若剩餘點數小於等於 $n / \log n$ ，或處理次數達到 $\log n$ 次，則使用 Graham's Scan 減少時間複雜度。

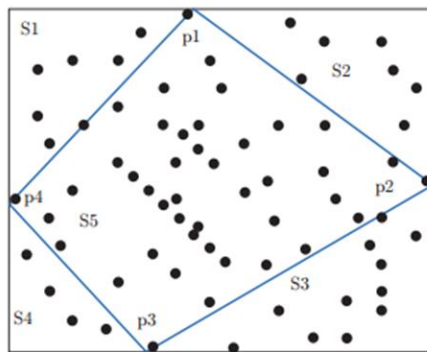
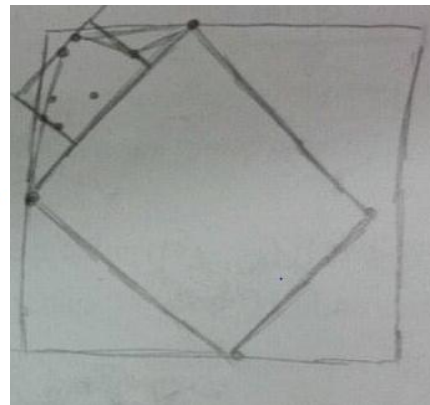


Fig. 1: Divided point set

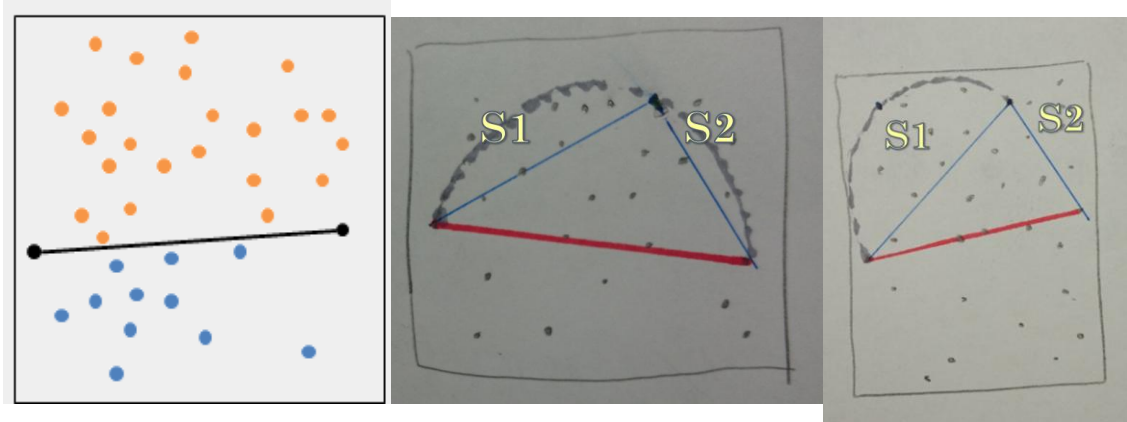


專題的演算法-2

1. 先找出最左及最右點連成線段，若有多於 1 個，則取線段最長者
2. 找到一個點形成一個最大的半橢圓區域，然後將此點與

線段兩端點相連

3. 以新線段為底重複步驟 2
4. 檢查剩餘點數，若剩餘點數小於等於 $n / \log n$ ，或處理次數達到 $\log n$ 次，則使用 Graham's Scan 減少時間複雜度。



五、結論:

此兩種方法，雖然在時間複雜度上不如 Granham's Scan、Jarvis' March 優秀，但可以一次排除掉大部分位於 Convex Hull 中的點，結合 Granham's Scan、Jarvis' March 等方法可使得操作時間大幅度縮短。

六、參考資料:

演算法筆記:<http://www.csie.ntnu.edu.tw/~u91029/ConvexHull.html>

維基百科:<https://zh.wikipedia.org/wiki/%E5%87%B8%E5%8C%85>