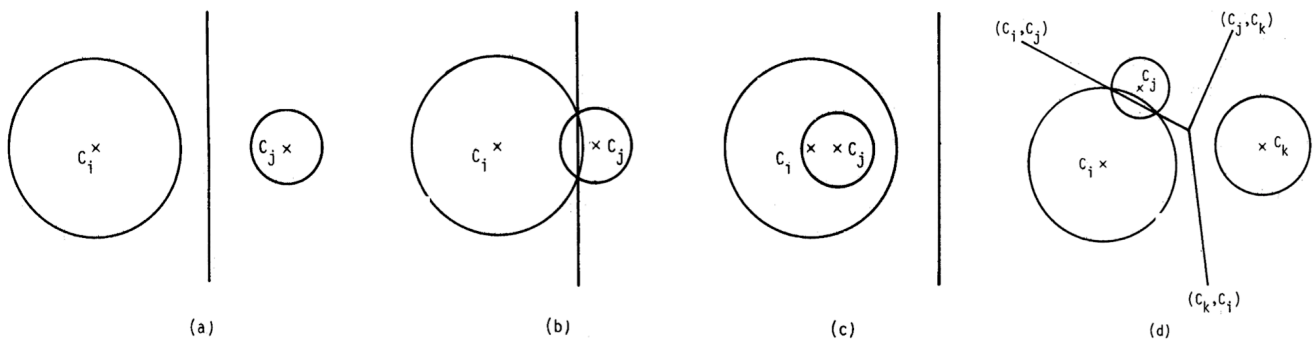


Voronoi Diagram in the Laguerre Geometry

問題定義

Voronoi diagram 為根據幾何圖形的距離將空間分割的方法。本篇所探討的二維 **Laguerre geometry** 敘述如下。首先，在平面上給定一圓 $C_i = C_i(Q_i; r_i)$ ，其圓心為 $Q_i = (x_i, y_i)$ ，半徑為 r_i ，以及一點 $P = (x, y)$ ，定義圓跟點的距離 $d_L^2(C_i, P) = (x - x_i)^2 + (y - y_i)^2 - r_i^2$ ，當 P 在圓內、圓上、圓外， $d_L^2(C_i, P)$ 分別是負數、零、正數。當 P 在圓外， $d_L(C_i, P)$ 正是 P 到 C 之切線的長度。接著，在平面上給定兩圓 C_i 和 C_j ，我們可以在平面上找到一直線，其上所有的點與兩圓等距，此直線稱作 **radical axis**。再者，在平面上給定三圓 C_i 、 C_j 和 C_k ，其圓心不共線，則所產生的三條 radical axis 會交在一點，稱作 **radical center**。Radical axis 和 radical center 的例子如下圖。



在平面上給定 n 個圓，定義 **Voronoi polygon** 為 $V(C_i) = \cap_j \{P \in \mathbb{R}^2 | d_L^2(C_i, P) \leq d_L^2(C_j, P)\}$ ，在空間中所構成的區域為一 convex set，而 n 個圓各自的 Voronoi polygon 之聯集會形成平面的分割 (partition)，這樣的圖形稱作 **Voronoi diagram**。Voronoi polygon 的頂點和邊分別稱作 **Voronoi point** 和 **Voronoi edge**。一圓之 Voronoi polygon 若非 (是) 空集合，則分別稱作 **substantial (trivial) circle**。一圓若與其 Voronoi polygon 有 (不) 相交，則分別稱作 **proper (improper) circle**。以上名詞透過下圖來舉例：

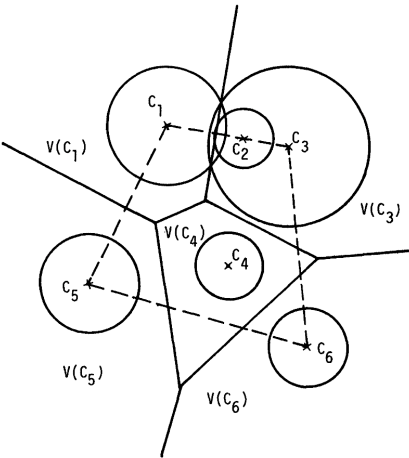
名詞	圖 2
<p>Substantial: 1, 2, 4, 5, 6, 7, 8</p> <p>Trivial: 3</p> <p>Proper: 1, 4, 5, 6, 7, 8</p> <p>Improper: 2, 3</p>	

解法敘述

首先，針對 n 個圓之 Voronoi diagram，我們有以下的輔助定理：

1. Trivial circle 必定是 improper。Improper circle 會被包含在 proper circles 的聯集。
2. 給定 n 個圓，Voronoi edges 的個數為 $O(n)$ 。
3. 考慮 Q_1, Q_2, \dots, Q_n 所形成之 convex hull：
 - 若 Q_i 落在 convex hull 之頂點，則 $V(C_i)$ 非空集合且無界。
 - 若 Q_i 落在 convex hull 之不含頂點之邊界，則 $V(C_i)$ 是空集合或無界。
 - 若 Q_i 不落在 convex hull 之頂點或邊界，則 $V(C_i)$ 是空集合或有界。

下圖是輔助定理 3 之三情形圖示：

名詞	圖 3
<p>落在頂點：C_1, C_3, C_5, C_6 無界。</p> <p>落在不含頂點之邊界：C_2 是空集合。</p> <p>不落在頂點或邊界：C_4 有界。</p>	

本篇借用了針對 n 個點之 Voronoi diagram 的演算法繼續發展。給定二維平面上的 n 個點之集合 $S = \{P_1, P_2, \dots, P_n\}$ ，目標是求出其 Voronoi diagram $V(S)$ 。我們首先對每個點的座標排序，按照排序重新編號，再利用 divide and conquer 來解決 Voronoi diagram：

- Divide：將所有的點分成左右兩半， $L = \{P_1, P_2, \dots, P_{[n/2]}\}$ 和 $R = \{P_{[n/2]+1}, \dots, P_n\}$ 。
- Conquer：利用遞迴分別解決 $V(L)$ 和 $V(R)$ 。
- Merge：找到分割 L 和 R 之分割線 (dividing line)，將所有的線段合併形成 $V(S)$ 。

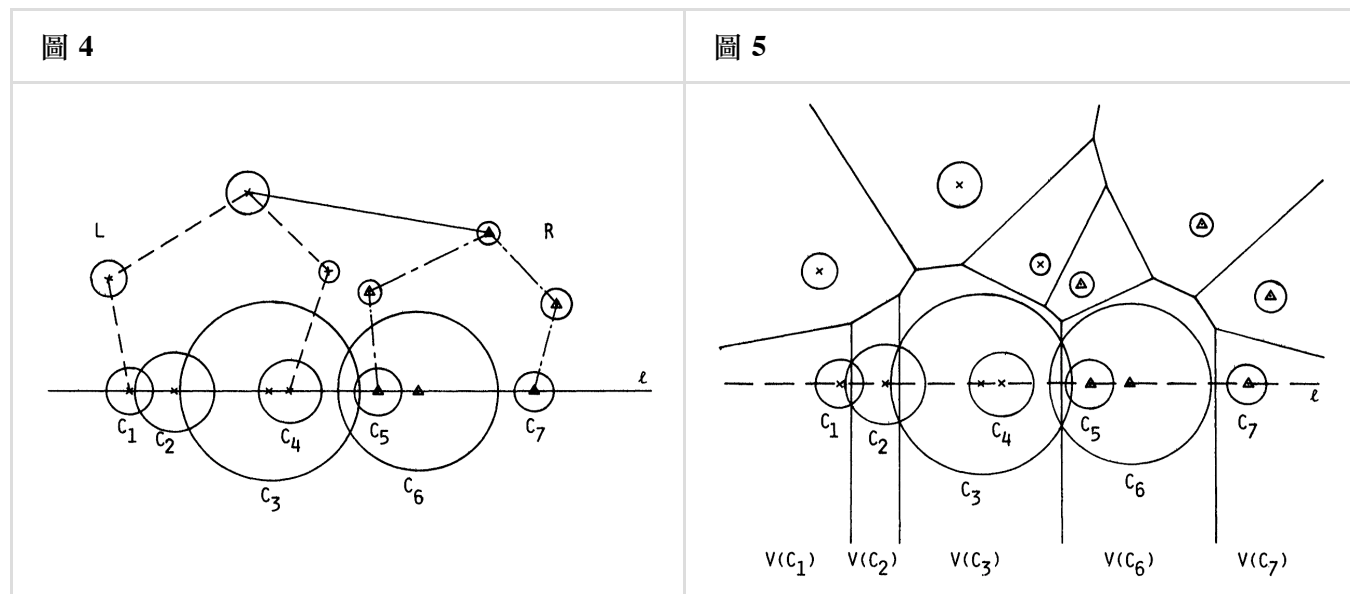
接著，針對 n 個點之 Voronoi diagram，我們有以下的輔助定理：

4. 分割線是由兩條無限延伸的射線與有限的線段組成。每條射線和線段都是由兩個 Voronoi polygon： $V(P_i) \in V(L)$ 和 $V(P_j) \in V(R)$ 之交集所構成。
5. 每條射線都是 S 所形成之 convex hull 上連續兩點的中垂線，此兩點必須滿足一點落在 L ，另一點落在 R 。

針對 n 個圓之 Voronoi diagram，我們可以設計出類似的演算法。首先，我們將所有的圓之圓心作排序，並分成左右兩半， L 和 R 。接著，利用遞迴分別解決 $V(L)$ 和 $V(R)$ 。最後，找到分割 L 和 R 之分割線，將所有的線段合併形成 $V(S)$ 。以下是關於分割線的一個輔助定理：

6. 分割線是由兩條無限延伸的射線與有限的線段組成。分割線左（右）邊的任何點到左（右）邊的某個圓比到右（左）邊的任何圓都還近。註：此性質不適用於 n 個點之 Voronoi diagram。

以下詳述如何找到分割 L 和 R 之分割線。首先，我們必須找到所有圓心所形成之 convex hull。接著，假設圓心不共線（non-degenerate case），我們可以直接利用定理 5 找到分割線的兩端射線。最後，從一端射線開始，逐一作順時針和逆時針搜尋下一條線段，直到抵達另一端射線。假設圓心共線（degenerate case），則情況會變得比較複雜，無法直接利用定理 5 找到兩端射線。以下圖 4 和 5 為例， C_4 和 C_5 之圓心是橫跨左右兩半距離最近的兩圓心（圖 4），理論上， C_4 和 C_5 的 radical axis 會是一端射線，但實際上， C_4 和 C_5 之 radical axis 並沒有出現在 Voronoi diagram 上（圖 5）。原因是 C_4 和 C_5 之圓心與 C_1, C_2, C_7 之圓心在 convex hull 的邊界上共線（圖中的 l ），故定理 5 不適用。



為了解決多個圓心在 convex hull 邊界上共線的這個問題，有了下面這個定理：

7. 令 L_l 和 R_l 是圓心落在 l （convex hull 上的一個邊界），且位於左右兩半之圓的集合，並令 $C_i \in L_l \subseteq L$ 和 $C_j \in R_l \subseteq R$ 為兩個圓，使得其 radical axis 為 Voronoi diagram 上的 Voronoi edge e^* ，則 e^* 就是分割左右兩半之分割線的一端射線。

搜尋 e^* 的步驟如下：

- 找到 convex hull 上有多個圓心的邊界 l 。
- 找到 $L_l \subseteq L$ 和 $R_l \subseteq R$ 。
- 分別建構 $V(L_l)$ 和 $V(R_l)$ ：以 $V(L_l)$ 為例。首先，從 $V(L)$ 取出 L_l 任兩圓所形成之 Voronoi edge（如果存在的話），所形成的集合就是 $V(L_l)$ 。 $V(R_l)$ 可以透過相同方式建構。
- 將 $V(L_l)$ 和 $V(R_l)$ 合併成 $V(L_l \cup R_l)$ ：已知 $V(L_l)$ 和 $V(R_l)$ 所有的 Voronoi edges 會互相平行（因為都垂直於 l ）。相鄰兩 Voronoi edges 所圍成的區域都是兩個 Voronoi regions 的交集，分別是 $V(C_i)$ 和 $V(C_j)$ ，其中 $C_i \in L_l \subseteq L$ 且 $C_j \in R_l \subseteq R$ 。
- 檢查 $V(L_l \cup R_l)$ 的每個區域，假設在這個區域中，包含 C_i 和 C_j 所構成的 radical axis e^* ，則 e^* 就是分割線的一端射線。

搜尋 e^* 的例子如下：

- Convex hull 上有多個圓心的邊界 l 。
- $L_l = \{C_1, C_2, C_3, C_4\}$ 。 $R_l = \{C_5, C_6, C_7\}$ 。
- 觀察與 l 垂直的五條虛線，正是 $V(L_l \cup R_l)$ 所包含的其中五條 Voronoi edges，彼此互相平行，且相鄰兩 Voronoi edges 所圍成的區域都是兩個 Voronoi regions 的交集，以圖 6 的斜線區域為例，是 $V(C_3)$ 和 $V(C_6)$ 的交集，其中 $C_3 \in L_l \subseteq L$ 且 $C_6 \in R_l \subseteq R$ 。
- 考慮同樣的斜線區域，在這個區域中，包含 C_3 和 C_6 所構成的 radical axis e^* （圖 7 中的粗實線），故 e^* 就是分割線的一端射線。

圖 6

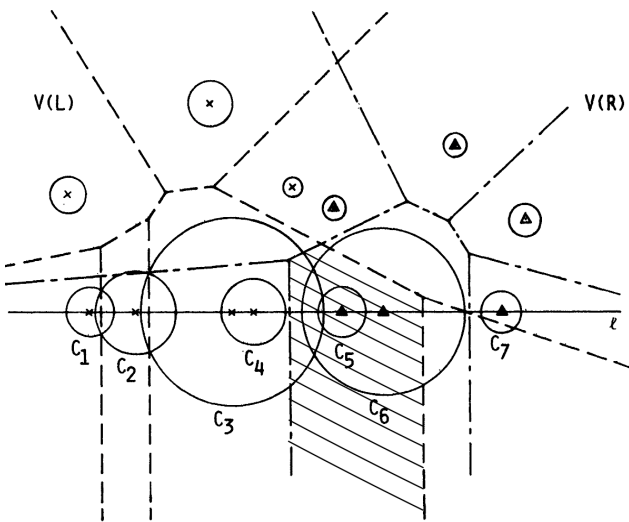
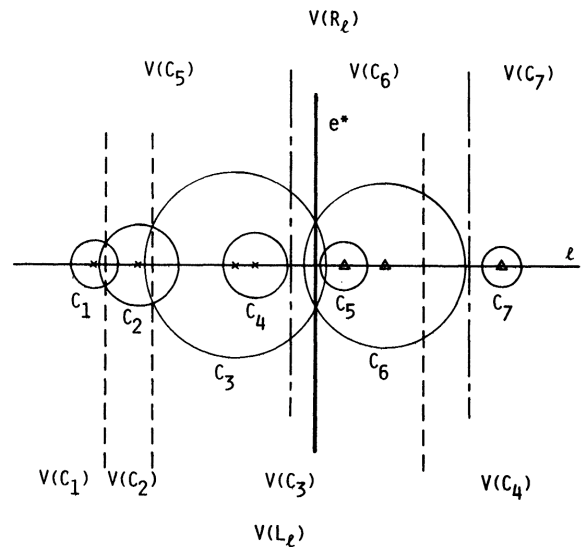


圖 7



總結本篇的演算法：

1. 將所有的圓心之 x 座標中點。 $O(n)$
2. 將所有的圓根據 x 座標中點分成左右兩半， L 和 R 。 $O(1)$
3. 遞迴分別解決 $V(L)$ 和 $V(R)$ 。 $2T(n/2)$
4. 從 L 和 R 各自的 convex hull 建構出所有圓心之 convex hull。 $O(n)$
5. 若圓心不共線（non-degenerate case）： $O(n)$
 1. 找到 convex hull 上橫跨左右的相鄰兩點 C_i 和 C_j 。
 2. C_i 和 C_j 之中垂線是分割線之一端射線。
6. 否則就是圓心共線（degenerate case）： $O(n)$
 1. 找到 convex hull 上有多個圓心的邊界 l 。
 2. 找到 $L_l \subseteq L$ 和 $R_l \subseteq R$ 。
 3. 分別建構 $V(L_l)$ 和 $V(R_l)$ 。
 4. 將 $V(L_l)$ 和 $V(R_l)$ 合併成 $V(L_l \cup R_l)$ 。
 5. 檢查 $V(L_l \cup R_l)$ 的每個區域，假設在這個區域中，包含 C_i 和 C_j 所構成的 radical axis e^* ，則 e^* 就是分割線的一端射線。
7. 從一端射線開始，逐一作順時針和逆時針搜尋下一條線段，直到抵達另一端射線。 $O(n)$

根據上述的演算法，時間複雜度 $T(n)$ 有以下的遞迴關係： $T(n) = 2T(n/2) + O(n)$ ，故 $T(n)$ 為 $O(n \log n)$ 。

讀後心得

本篇所提出的演算法中，運用到多種演算法設計的概念，目的都是為了使 merge 的步驟在 $O(n)$ 內完成。例如，步驟 1 尋找 x 座標的中點，運用到 prune and search 的技巧，步驟 4 為從左右兩半各自的 convex hull 建構出所有圓心之 convex hull，運用到了 divide and conquer。另外，整個演算法也借用了許多解決 n 個點之 Voronoi diagram 的概念，故我認為這篇論文完整地體現 divide and conquer 的強大之處，而最難的地方莫過於將 Voronoi Diagram in the Laguerre Geometry 化簡成 divide and conquer 可以解決的形式，一個問題一旦可以透過 divide and conquer 解決，時間複雜度必然可以獲得大幅改善。

連同上課所提到的例子，我認為在用 divide and conquer 解決問題，最難的步驟正是 merge，而「merge」這個單字反而沒有出現在 divide and conquer，某種程度會讓人忽略最後最困難的一步其實是如何將所有的解合併起來。聽完課堂的例子以及本篇論文，對於各式各樣 merge 的方法感到肅然起敬，都不是輕而易舉可以想到的步驟。

解決 Voronoi Diagram in the Laguerre Geometry 的演算法可以用來解決其他類似的問題，比如說（1）平面上給定 n 個圓，決定一點 P 是否落在所有圓的聯集之內。（2）將平面上的 n 個圓分割出 connected components。（3）找到平面上 n 個圓之聯集所形成的輪廓。以上三個例子在本篇的應用皆有提及，而 $O(n \log n)$ 的演算法對這類的問題已經是最佳的，因為解決這類問題的 lower bound 已被證明是 $\Omega(n \log n)$ 。故我認同本篇論文對於計算幾何學、電腦視覺的貢獻程度，尤其是這些領域的問題往往十分抽象，無法容易地將圖形問題轉變成數學邏輯式，需要透過大量的視覺輔助、圖形等等來幫助理解，而有學者能設計出最佳的演算法，更屬難能可貴。