

二、研究計畫內容：

壹、摘要

隨著科技爆炸性的進步，交通越來越便利，人們開始注重旅遊、拍照記錄生活，上傳到社群網站與朋友、家人分享且記錄自己的生活點點滴滴。據統計，2013 年 Google+ 的照片上傳量將近八千萬張，2014 年元旦 Facebook 當日的照片上傳量高達六億張照片，由此可知，拍照已經是人們不可或缺的生活方式。

如上所述，本計畫想利用這些數量龐大的照片，去提供給人們更加優質的旅遊生活，好比我們常常在前往一個不熟悉的景點之前，並不僅僅想知道路名，我們更想要看到的是道路實際上的真實畫面。

本計畫利用照片的地理資訊進行索引與模擬照片所涵蓋的範圍 (Field of View)，達到快速篩選使用者提供的照片以即時合成全景圖的效果，本計畫將開發一個可以讓使用者上傳照片或規畫路線的平台，利用上傳的照片，分析位置以提供周遭景點全景圖或是合成該照片的全景圖，而規劃完成後的路線，將提供直接將路口映入眼簾的全景圖給使用者預覽，也讓使用者在計畫旅行的時候有比平板的地圖更實際的資訊。

貳、研究動機與研究問題

旅行能帶來生命的驚嘆號，一個美好的旅行常能讓人放鬆心情甚至豐富生命的涵養，而在我們展開旅行前，常會利用線上的地圖服務，如 Google Map、OpenStreetMap，來進行路徑查詢以得到豐富的在地資訊；而 Google Map 在提供搜尋結果的同時也提供了完善的街景服務，讓我們得以對沿途風景有更清楚的認識與了解。然而，當我們想知道某位置的全貌時，街景圖在使用上卻有著種種的不便，必須經過多次點選、移動和讀取才能掌握其全貌，也因此往往需要花費許多精力及時間；且 Google Map 提供的常是較久之前的街景，與最新的狀況並不一定會完全一致。假若現在我們只需要在路徑上輕輕點擊一個位置，就可以得到該位置的全景圖，並且透過存取由大家提供的圖片 (Crowd-Sourcing) [6] 來提供更即時、變化度更高、愈加豐富有趣的全景圖，聽起來是不是很有意思？本著這樣的熱情，我們決定研究一個嶄新的系統，希望帶來更便捷的旅遊方式、更好的使用者體驗。

利用 FOV (Field of View) [1] 可以模擬出照片的涵蓋範圍、拍攝的位置、角度、時間…等資訊，讓系統得以有效率地存取及運算；而如何透過適當的資料結構來加速系統反應時間，也是個重要的議題，目前我們構想以 R-tree 讓系統可以快速地篩選出適當的圖片，以進行一連串的運算處理。另外，Eye-tracking 是一種追蹤人眼焦點的技術，我們預計利用智慧型手機的前置鏡頭來擷取拍照時的注視區域，並與其他資訊一併儲存，拍照時的注視區域可以讓系統更精確模擬出照片的 FOV，以求最佳的處理結果。

最終，我們希望提供一個平台，使用者能夠在地圖上點選任意位置，系統會

自動從圖片資料庫挑選出合適的照片組成全景圖；而這個應用，也可以導入路徑規劃中，一般來說，若只是單純在地圖上標示行經路線，對於方向感比較不好的使用者來說，其實並不是一個非常易懂的方式，也因此市面上的導航機會在路口顯示一個 3D 實景繪製圖；而我們想要做的是將顯示有方向的全景圖依序標示在每個路口，除了讓使用者能輕易明瞭實際動向外，比起 3D 實景繪製圖將更具真實感，也免去了 3D 繪製的成本。當使用者在遊歷時也可以朝著景點拍一張含有地理資訊的照片並上傳，藉此分析使用者位置，周遭有哪些熱門景點，使用者也可在地圖上獲得景點的全景圖。

當使用者在景點遊玩時，更可以把自己在景點中拍到照片上傳，使用者將會得到一個以自己所在位置為主題的旅遊全景圖。

以下為本計畫的研究問題：

1. 如何利用 FOV 進行一連串的处理與運算，去篩選出適當的照片，以得到最佳的全景圖。
2. 如何處理大量的來自不同使用者、不同拍攝器材的質量不一的圖片。
3. 如何利用 Eye-tracking 讓 FOV 的模擬更加精確。

參、文獻回顧與探討

全球定位系統(Global Positioning System, GPS)的普及，Sakire 等作者[1]提出了(Field of View, FOV)和角度的概念能夠改善現有照片蒐尋不便的問題，照片有了 FOV 和角度就能夠將範圍縮減，而他們也使用了 R-Tree 的資料結構用來加速搜尋效率[3]，R-Tree 是近年來在多維資料搜尋最有效率的方法，其中最小矩形範圍(Minimum Bounding Rectangle, MBR)的概念可以篩掉不可能正確的資料，因此可以加速搜尋的速度；[6]、[7]及[8]中利用了照片夾帶的地理資訊及一些參數來進行像是 Keyframe 的挑選或全景應用等，而我們將運用他們提到的方法，來進行我們的計畫—將 Crowd-Sourcing 的圖象依據地點做全景應用。

肆、研究方法及步驟

1. Field of view

本研究以模擬照片所涵蓋的視野(Field of view, 簡稱 FOV)[1]為概念，藉此有效率的挑選出適當的照片合成全景圖。

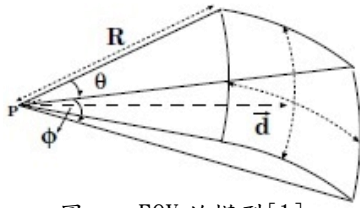
第一個步驟是照片資料的蒐集，我們將以大眾提供的照片(Crowd-Sourcing)作為圖片來源，建立與管理一個時空資料 (Spatio-temporal database)，利用這個方式來獲得更豐富多元的照片。

下個步驟是建構出 FOV 的模型，我們使用這些有透過 GPS、3D 感測器所記錄的地理資訊的照片，透過一連串運算（如公式一），去計算出其他我們所需要的地理資訊，由這些地理資訊我們可以建構出 FOV 的模型(如圖一)，並描述成

FOVScene($P, \vec{d}, \theta, \varphi, U, R, t$)的形式，再把地理資訊的 metadata 存入資料庫，

地理資訊內容如表格一。

我們也預計加入 Eye-tracking，以智慧型手機的前鏡頭獲得人眼注視的區域的資訊，以獲得最佳的搜尋結果。



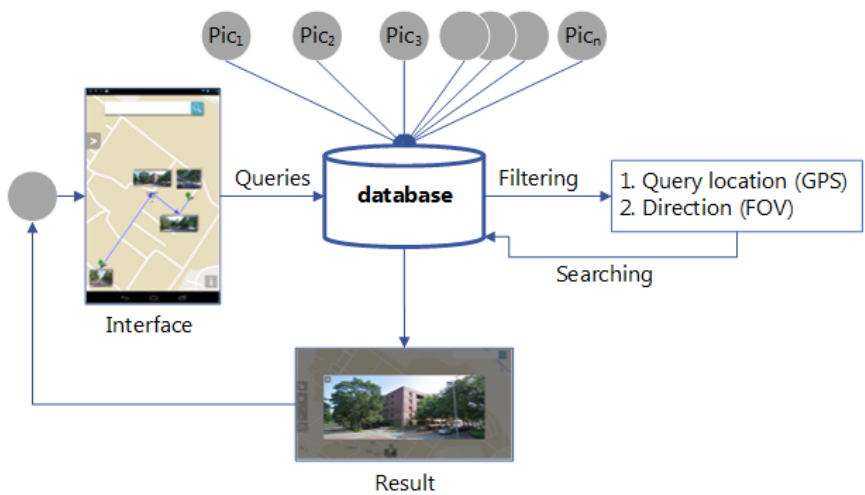
圖一、FOV 的模型[1]

$$\theta = 2 \tan^{-1} \frac{y}{2f} \quad (\text{公式一、}\Theta \text{ 算式})$$

符號	意義
P	相機位置，包含(經度, 緯度, 高度)，從 GPS 取得
\vec{d}	相機拍攝的方向，
R	可視的距離
Θ	相機水平視角大小，會根據鏡頭收縮放而改變
Φ	相機垂直視角大小，會根據鏡頭收縮放而改變
U	相機拍攝的俯仰角
y	影像感測器轉換的規格大小
f	鏡頭焦距
h	目標物的實際高度
t	拍攝時間

表格一、此表格整理出 FOV 的模型所使用的地理資訊

以 FOVScene(P, \vec{d} , θ , ϕ , U, R, t)方式描述 FOV 的扇形模型時，可以不需要對於全部的照片進行查詢，能夠以較洽當的作法去避免運算的負擔，也就是以一個直覺的結構 R-tree 來進行多維度的索引[2]，這部份在 R-tree 會詳細的說明。



圖二、系統架構

在我們預計完成的網頁平台上，當使用者要進行全景圖的合成時，系統則會到資料庫裡去搜尋與查詢的範圍重疊的照片，再從中找出較適合合成全景圖的照片。以下為我們所使用 FOV 的篩選演算法[1]。

Algorithm 1 *SceneIntersect*($Q, FOVScene(t, P, \vec{d}, \theta, R)$)

```

 $Q \leftarrow$  Given convex polygon shaped query region
 $Q = \langle V, E \rangle$  ( $V \leftarrow$  Set of vertices in  $Q$ ;  $E \leftarrow$  Set of edges in  $Q$ )
 $FOVScene(t, P, \vec{d}, \theta, R) \leftarrow$  FOV region for frame  $t$ 
 $pointPolygonIntersect(P, Q) \leftarrow$  returns true if point  $P$  is in  $Q$ 

if  $pointPolygonIntersect(P, Q)$  is true then
    return true;
end if
for  $\forall v \in V$  do
    if  $pointFOVIntersect(v, FOV(t, P, \vec{d}, \theta, R)) == true$  then
        return true;
    end if
end for
for  $\forall e \in E$  do
    if  $edgeFOVIntersect(e, FOV(t, P, \vec{d}, \theta, R)) == true$  then
        return true;
    end if
end for
return false;
end for

```

圖三、篩選演算法一

Algorithm 2 *pointFOVIntersect*($q, FOVScene(t, P, \vec{d}, \theta, R)$)

```

 $q \leftarrow$  Given query point
 $FOVScene(t, P, \vec{d}, \theta, R) \leftarrow$  FOV region for frame  $t$ 
 $EarthDistance(q, P) \leftarrow$  returns the earth distance between points  $q$  and  $P$ 

if  $EarthDistance(q, P) \leq R$  then
     $\alpha =$  angle of  $\vec{d}$  with respect to north
     $\delta =$  angle of vector  $\vec{P}q$  with respect to north
    if  $\alpha - (\frac{\theta}{2}) \leq \delta \leq \alpha + (\frac{\theta}{2})$  then
        return true;
    end if
end if
return false;

```

圖四、篩選演算法二

Algorithm 3 *edgeFOVIntersect*($q, FOVScene(t, P, \vec{d}, \theta, R)$)

```

 $e \leftarrow$  Given edge
 $FOVScene(t, P, \vec{d}, \theta, R) \leftarrow$  FOV region for frame  $t$ 
 $LinePointDistance(P, e) \leftarrow$  returns the shortest earth distance between point  $P$  and closest point on line  $e$ 

 $\langle p_1, p_2 \rangle = lineCircleIntersect(e, P, R)$ ; {return the two points that line  $e$  intersects with the circle centered at  $P$  with radius  $R$ . Note that  $p_1 = p_2$  if  $e$  is tangent to circle}
if  $LinePointDistance(P, e) \leq R$  then
     $\alpha =$  angle of  $\vec{d}$  with respect to north
     $\delta_1 =$  angle of vector  $\vec{P}p_1$  with respect to north
     $\delta_2 =$  angle of vector  $\vec{P}p_2$  with respect to north
    if  $[\alpha - (\frac{\theta}{2}), \alpha + (\frac{\theta}{2})]$  overlaps with  $[\delta_1, \delta_2]$  then
        return true;
    end if
end if
return false;

```

圖五、篩選演算法三

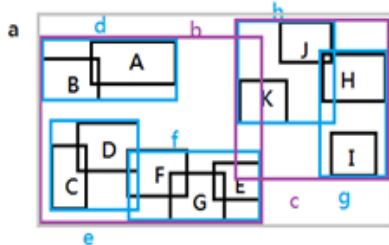
圖三中，Q 為使用者要查詢的範圍，當要判斷一張照片涵蓋的範圍是否與 Q 重疊時，首先判斷已建好 FOV 模型的照片的位置 P 是否在 Q 的範圍內，若 P 在 Q 的範圍內則回傳 true，若不在，則依序用圖四、圖五演算法，判斷這張照片中的 FOV 是否和 Q 的點與邊有所相交，若有的話回傳 true，代表這張照片涵蓋所查詢的範圍，是我們所要挑選的照片。

2. R-tree

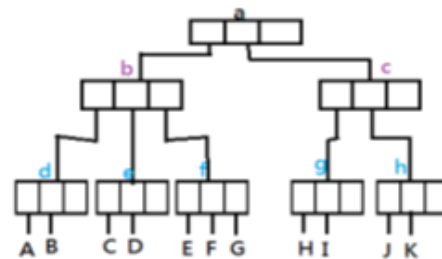
R-tree 是階層式多維度的索引技術[3]，用於處理多維的資料。R-tree 會依照物體的座標相似性將物體分群，以最小矩形(Minimum Bounding Rectangle，簡稱 MBR)包覆並存放在葉節點(leaf-node)中，之後相近的 MBR 繼續分群，接著用更大的 MBR 包覆，依此類推直到所有資料被包含在同一 MBR 裡。

在 R-tree 的索引資料結構中，R-tree 的葉節點和內部節點(internal-node)都有著數個分支，內部節點存放的是 MBR，會包覆其底下所有的子節點，而葉節點則會指向每個物件的資訊，因此根節點(root)存放的 MBR 能夠涵蓋其底下所有子結點的 MBR 且會將所有的資料包覆。

圖六表示所有物件在空間資料散佈的情形，而圖七表示對應的 R-tree 結構，d、e、f、g 和 h 代表葉節點，透過葉節點的 entry 存取 A、B、...、K 物件。a、b、c 代表內部節點，其中根節點 a 包含 b 和 c 兩個節點，b 包含 d、e 和 f 三個葉節點，c 包含 g 和 h 兩個葉節點。從圖六可得知，根節點完全覆蓋子節點的 MBR，並且允許 MBR 之間可有重疊情況發生。



圖六、散佈在資料空間的物件



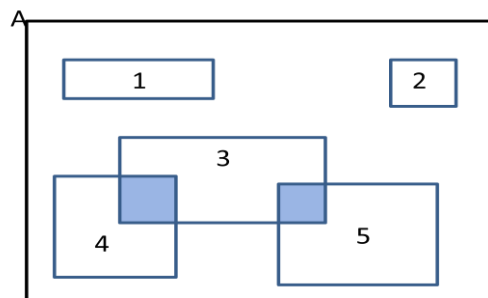
圖七、對應的 R-tree 結構

本次研究會以 R-tree 來做資料庫的索引，以地點為主軸，初步去搜尋和該位置相近的周邊圖片，之後再從這些被挑選出來的周邊照片再去做進一步的篩選和處理，進而合成全景圖，除此之外為了增加 R-tree 的附加效益以及提供更多人性化的功能，R-tree 會融入 Statistical Information Grid (STING) 的方法。STING 是階層式架構，會將資料空間以眾多網格的方式呈現，高階層彙整低階層的資訊[4]。本次研究因為空間資訊涵蓋著面積資訊，

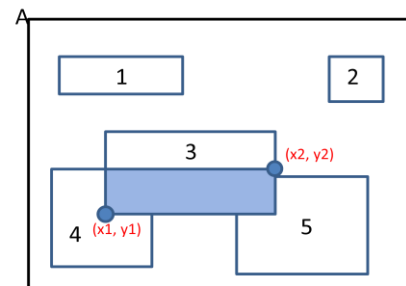
所以我們以 STING 為概念將其應用於 R-tree 的索引[5]，針對物件的面積提出下列的資料探勘項目，分別為 1. 涵蓋所有重疊區域的最小矩形 (Minimum Rectangle of Cover Total Overlap Region，簡稱 MRCTOR)，因為 R-tree 的 MBR 具有重疊的狀況，可以將這些重疊的面積視為"常被旅遊的範圍"，而 MRCTOR 提供涵蓋"所有的常被旅遊的範圍"的最小矩形，2. 面積最大的重疊區域(The

Largest Area of Overlap Region, 簡稱 LAOR), LAOR 提供涵蓋"最大的常被旅遊的範圍"的最小矩形, 3. 同時被最多物件重疊的區域 (Overlap Region with the Most Overlap Objects, 簡稱 ORM00), ORM00 提供涵蓋"最常被旅遊的範圍"的最小矩形, 以下將以圖示說明 MRCTOR、LAOR、ORM00[5]。

(1)MRCTOR: 涵蓋特定範圍內所有重疊區域的最小矩形, 圖八(a)表示(物件3和物件4)、(物件3和物件5)的MBR彼此有重疊圖八(b)表示涵蓋所有重疊區域的最小矩形((x1, y1), (x2, y2))用來表示 MRCTOR, 儲存於物件 A 的父節點, (x1, y1)為 MRCTOR 左下角座標值, (x2, y2)為 MRCTOR 右上角座標值。

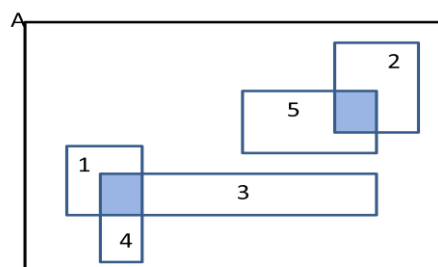


圖八(a)、所有重疊的區域

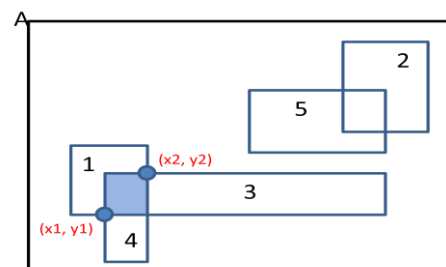


圖八(b)、涵蓋所有重疊區域的最小矩形

(2)LAOR: 特定範圍裡面積最大的重疊區域, 圖九(a)表示(物件1和物件4)、(物件2和物件5)的MBR彼此有重疊圖九(b)表示重疊區域裡最大的面積, 因為兩個斜線區域面積相等, 選擇涵蓋較多 MBR 的斜線區域((x1, y1), (x2, y2), NRectLAOR)用來表示 LAOR, 儲存於物件 A 的父節點, (x1, y1)為 LAOR 左下角座標值, (x2, y2)為 LAOR 右上角座標值, NRectLAOR 紀錄重疊物件的個數。

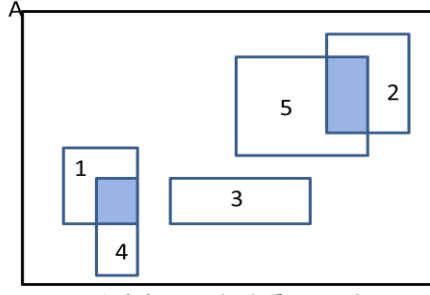


圖九(a)、所有重疊的區域

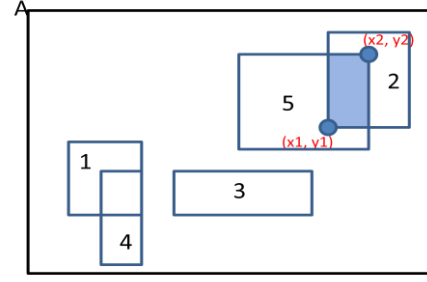


圖九(b)、面積最大的重疊區域

(3)ORM00: 特定範圍裡同時被最多物件重疊的區域, 圖十(a)表示(物件1和物件4)、(物件2和物件5)的MBR彼此有重疊圖十(b)表示被最多物件重疊的區域, 因為兩個斜線區域都是被2個物件重疊, 選擇面積較大的斜線區域((x1, y1), (x2, y2), NRectORM00)用來表示 ORM00, 儲存於物件 A 的父節點, (x1, y1)為 ORM00 左下角座標值, (x2, y2)為 ORM00 右上角座標值, NRectORM00 紀錄重疊物件的個數。



圖十(a)、所有重疊的區域



圖十(b)、同時被最多物件重疊的區域

因為要將 MRCTOR、LAOR、ORMOO 融入 R-tree 內，需要更動 R-tree 結構，子節點和葉節點原本所涵蓋的內容再加上 MRCTOR、LAOR、ORMOO[5]。

3. 全景圖

Baseline Algorithm[6]，簡稱BA-P，是依據與地點q的距離來選出適當的圖像；先選出拍攝位置離地點q在預先訂定好的限制距離內的圖像（例如10公尺，標準GPS的誤差值上限），BA-P會略過那些距離地點q過遠的圖像。但是只考慮圖像拍攝位置還是不夠的，因此我們要會需要的Direction-based Algorithm來協助篩選圖片。

Direction-based Algorithm[6]，簡稱DA-P，我們將利用DA-P來幫我們從零散的圖像中找出最適合的拿來合成全景圖。DA-P利用了 filter-refine paradigm；在filter的階段，利用BA-P（刪除掉那些距離地點 q 過遠的視頻）來過濾掉相片位置超出預先訂定的距離（半徑為r）範圍外的FOV，而剩下所得到的FOV就為CF集合的成員。CF集合是那些被BA-P所篩選出來的視頻設的集合，CF可區分為「重疊」與「覆蓋」兩部分，如圖十一與圖十二。圖十一的公式如下：

- (1) 定義一(重疊)：給予CF集合裡面任意兩個FOV f_1 , f_2 ，然後由 $Overlap(f_1, f_2)$ 來表示 f_1 , f_2 兩者覆蓋的視角角度。

$$Overlap(f_1, f_2) = (f_2 \cdot \frac{\alpha}{2} + f_1 \cdot \frac{\alpha}{2}) - (f_2 \cdot \theta - f_1 \cdot \theta) \quad (\text{公式二})$$

- (2) 定義二(覆蓋)：給予任一子集合 $F = \{f_1, \dots, f_n\}$ ， $F \in CF$ ， f_1 到 f_n 是以視角方向角度以小到大的方式依序排序。然後由 $Cover(F)$ 來表示F的角度集合。

$$Cover(F) = \sum_{i=1}^n f_i \cdot \alpha - \sum_{j=1}^{n-1} Overlap(f_j \cdot \theta, f_{j+1} \cdot \theta) \quad (\text{公式三})$$

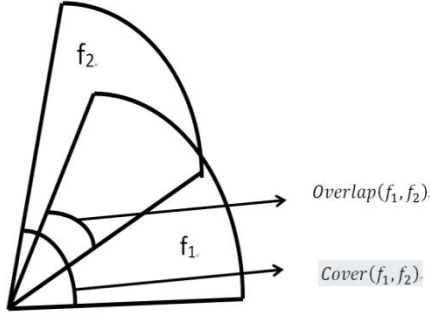
圖十二的參數說明:A(0, 0)與B(X, Y)是相片的位置， γ 為以A為準的角度差、R為可視距離。利用這個簡單的模型就可以計算出 $Cover(f_1, f_2)$ ，即圖與圖之間覆蓋程度。當 $l < R$ 時，如圖十二的(a)，覆蓋比率為

$$\text{sim}(f_A, f_B) = \frac{R \tan \beta - (R-l) \tan(\beta - \gamma)}{2R \tan \beta} \quad (\text{公式四})$$

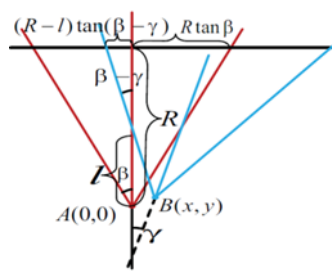
當 $l > R$ 時，如圖十二的(b)，覆蓋比率為 $\text{sim}(f_A, f_B) = \frac{R \tan \beta - (l-R) \tan(\beta - \gamma)}{2R \tan \beta}$ 。以

上類似的公式都是用於 $(\beta \leq -\gamma \leq 2 \times \beta)$ 和 $(-\beta \leq \gamma \leq 0)$ 的範圍。而當 $(\gamma \geq 2 \times \beta)$ 或是 $(\gamma \leq -2 \times \beta)$ 時，覆蓋率一律不存在，因此覆蓋率為 0。

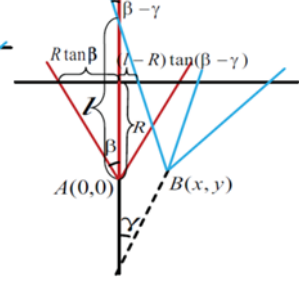
在refine階段，系統將先對CF集合裡的圖像依據FOV的可視角度做從小到大的排序，然後將可視角度最小的圖像做為第一個比較用的圖像，之後在每一個先被選中的圖像 f_{pre} 中，去挑選使其重疊比率適中的 FOV。已知的缺點是，目前這些方法都是針對非Crowd-Sourcing資料來源[6]，故若單以以上這些條件來做篩選，可能會造成圖與圖之間品質不一、特性不一等因素的「不一致」而造成視覺上的瑕疵，在接下來的計畫中，我們將持續探討並設法找出解決之道。



圖十一



圖十二(a)[7]



圖十二(b)[7]

我們可以利用照片的GPS地理位置來幫我們再把CF裡面集合的圖片去做過濾的動作，確保這些圖片之間的瑕疵能夠達到最小化。圖十三是greedy-based Internal Route Generation演算法[8]。此策略是要先去一一地把每張圖片當成起始點(line 2)，然後去找下一張最能夠與此圖接合的照片，把不一致達到最小化為基礎(line 6-7)，直到所有圖片都跑過為止(line 5-10)。因此，當有N個照片的時候，只會產生最多N種順序。最後，最能夠與此圖接合的照片會被傳送回來(line 13)，進而形成圖片集合以組成全景圖，如圖十四。

Algorithm 3 Greedy-based Internal Route Generation (S')

Require: S' : coverage subshots

- 1: $V_{S'} \leftarrow$ cluster S' by their video id
 - 2: **for** $v_i \in V_{S'}$ **do**
 - 3: $V' = V_{S'}, V' - = v_i$
 - 4: attach v_i to $OrderList_i$
 - 5: **while** $V' \neq \emptyset$ **do**
 - 6: $v_j = \arg \min_{v_j \in V'} \{cost(v_i.end, v_j.start)\}$
 - 7: $cost_{ij} = I(v_i.end, v_j.start)$
 - 8: attach v_j to the end of $OrderList_i$
 - 9: $V' - = v_j, Score_i + = cost_{ij}, v_i \leftarrow v_j$
 - 10: **end while**
 - 11: $OrderList \leftarrow OrderList_i$
 - 12: **end for**
 - 13: Select $OrderList_i = \arg \min_{OrderList_i \in OrderList} \{Score_i\}$
-

圖十三

伍、 預期結果

本計畫執行期間將以國立中正大學為實驗進行場所。使用者可以透過點選地圖的任意位置，來觀看該處之全景圖，如圖一所示。

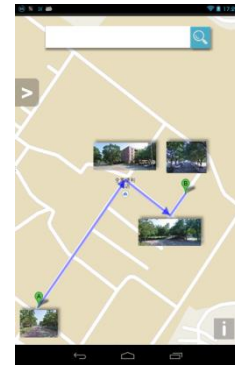


圖十五

基於現有之地圖服務提供的路徑規劃功能，可以在起點、終點、路口加上全景圖，甚至在轉彎處加上方向指引（如圖十五的橘色箭頭），讓使用者可以更好認路、方便旅遊。並且計劃將開發網頁版及行動 App 版，讓使用者不管是在家裡電腦前，或是在外使用行動裝置都可以方便使用；圖十六、圖十七分別是網頁版及行動 App 版之示意圖。



圖十六



圖十七

關於此系統，預期結果是：

1. 能正確地讓使用者點選位置以觀看全景圖，並且達到快速搜尋，系統能快速回應的結果。
2. 能正確地使用來自不同拍攝器材的影像，將全景圖呈現給使用者。
3. 基於現行地圖服務之路徑規劃功能，將全景功能導入其中，讓路徑規劃不再是單純的方向指引，將更添便利性。

陸、 參考文獻

- [1] Sakire Arslan Ay, Roger Zimmermann, Seon Ho Kim. In Viewable Scene Modeling for Geospatial Video Search, pages 311-314.
- [2] Yannis Theodoridis, Michael Vazirgiannis, and Timos Sellis. Spatio-Temporal Indexing for Large Multimedia Applications. In IEEE Intl. Conf. on Multimedia Systems, 1996.
- [3] A. Guttman, "R-Tree: A Dynamic Index Structure for Spatial Searching", *Proc. ACM SIGMODE*, p.47-57, 1984.
- [4] W. Wang, J. Yang, and R. Muntz, "STING: A Statistical Information Grid Approach to Spatial Data Mining", *In Proc. 23rd Intl. Conf. on VLBD*, pp.186-195, 1997.
- [5] 陳靖國、簡汶婷, "附加資料探勘資訊的 R-tree 資料存取", 2008 年資訊科技國際研討會, 朝陽科技大學, 台中霧峰, 台灣, 四月 25~26 日, 2008
- [6] Seon Ho Kim, Ying Lu, Junyuan Shi, Abdullah Alfarrarjeh, Cyrus Shahabi, Guanfeng Wang, Roger Zimmermann. In *Geo-Immersion from Crowd-Sourced, Sensor-Rich Videos*, pages 3-4.
- [7] Jia Hao, Guanfeng Wang, Beomjoo Seo, Roger Zimmermann. In *Keyframe Presentation for Browsing of User-generated Videos on Map Interfaces*, page 4.
- [8] Ying Zhang, Guanfeng Wang, Beomjoo Seo, Roger Zimmermann. In *Multi-Video Summary and Skim Generation of Sensor-rich Videos in Geo-Space*, pages 6-8.

柒、 需要指導教授指導內容

此計畫需指導教授的內容包括：

- Java/c++ programming languages 的程式設計指導。
- 時空資料庫的架構與設計。
- R-Tree 的概念與系統架構。
- 整體系統設計、開發、驗證的指導。
- 實作完成的系統效能與實用性的評估。