

國立高雄應用科技大學



100-101 年度獎勵科技大學及技術學院

典範科大計畫

產學及研究成果轉專題製作教材

無人飛行監控器之定位技術探討

無人飛行監控器之定位技術探討

(科系) (指導教授) (學生)

電機系 黃敬群教授 學生 黃偉立

1. 動機及目的

在此計畫中，我們著重於無人飛行監控器之影像式定位技術問題上的討論。隨著無線通訊的發展、影像取像技術的長足進步、以及無人飛行器的持續改良，利用無人飛行監控器當成移動式遠端監控平台已經成為許多具機動性的監控應用中不可或缺的角色。對於無人飛行監控器的監控任務而言，其中一個重要的工作在於提供定位的服務。然而，要提供穩定且準確的定位服務在技術並非容易的事。在實際的應用情況中，要發展這項服務，在技術需要面臨許多挑戰，這包含(1)無線傳輸的不確定性可能造成資料傳輸上的錯誤或是封包遺失、(2)數據資料傳輸與影像資料傳輸的同步問題、(3)用於定位服務的感測器本身的量測誤差、以及(4)全球定位系統在量測上的精準度限制。

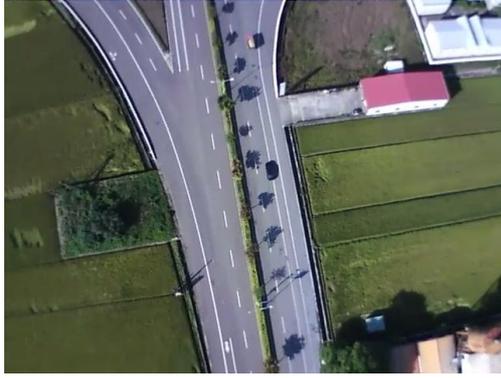
在本計畫中，我們探討利用飛行器所拍攝之監視影像協助提升定位精準度的可能性。搭配目前已經廣為使用的公用全球衛星圖資，如 Google Earth，我們系統的目標在於將監視影像中的每個像素點與衛星圖資建立全球對應關係(Geo-registration)。當此種對應關係被建立，系統提供的定位服務將不再只是飛機的經緯度位置，與拍攝到的監控影像；相反的，這種對應關係讓監控管理者可以清楚知道監控影像中任何感興趣區域在全球

的地理座標。傳統系統提供的定位服務主要包含了飛機經由全球定位系統所提供的飛機目前的經緯度位置，與飛機目前拍攝到的監控影像；在這樣的定位服務中，倘若我們對目前監控影像中的某一區域感興趣，我們僅能知道該區域在全球定位上的大致方位，這樣的資訊對於許多即時性或緊急性的監控應用，例如搜救任務，基本上是粗略且不足夠的。為了延伸這類的定位服務，飛行監控器之影像式定位技術逐漸開始被討論，仰賴近年來影像比對，與多影像之間特徵點對應與疊合技術的快速發展，影像式定位的可行性也逐步提升。在此計畫中，我們著重於無人飛行監控器之影像式定位技術問題上的討論。

2. 研究方法

最主要的概念就在於利用 Google earth 的圖資資料，幫助我們定位。因為 Google earth 的圖資資料裡面包含了我們所需要的資訊，像是地表上所有地面的資訊、所有位置的經緯度，圖片等等。所以我們從無人飛機所提供的參數資訊與測試影像序列當中，大約可以知道無人飛機大約是在地球的哪個位置，再藉由 Google Earth 所提供所在位置的圖資資料庫影像，比對測試影像與圖資資料庫並建立其對應關係，以準確的定位出無人飛機真正的位置。

了解以上的幾個重要的座標系之後，再來就是經過座標系之間的轉換，使得三維空間的座標可以投影到無人飛機所拍攝到的監控影像上。假設我們已知某點在世界上的 GPS 座標，



圖一



圖二



圖三

如圖所示，圖一是無人飛機所截取出來的測試影像序列，圖二是根據無人飛機所提供的參數資訊對應到的 Google Earth 截取下來的空照圖，圖三則是利用影像處理找出圖一與圖二的對應關係以後，將圖一對應投射到圖二的結果。所以當我們知道這個圖片對應到 Google Earth 的位置時，我們同時也會知道現在飛機的位置在世界座標的正確定位。

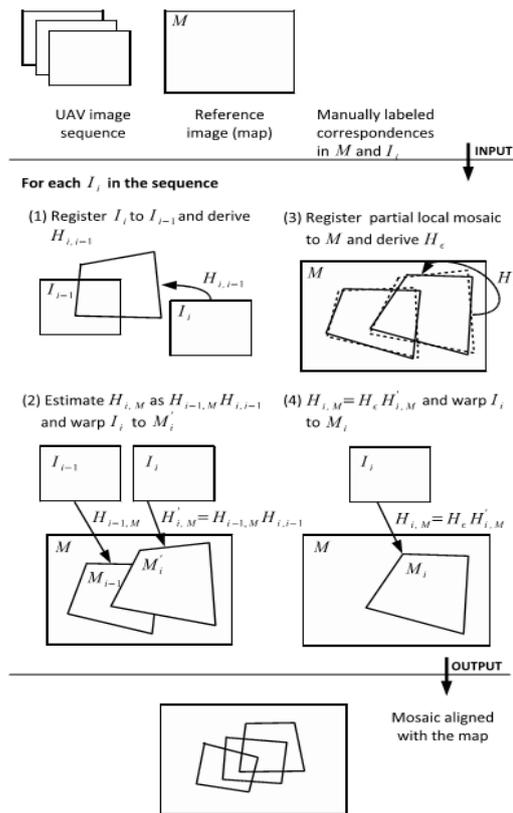
我們主要是根據 Yuping Lin[1]

所提供的方法，將我們的概念實現出來。我們先預備連續的無人飛機影像 UAV image 數張和一張 Google Map 的影像當作 reference map 來作為我們系統的輸入資料。接著我們需要先手動初始化我們第一張 UAV image 與 Google Map 的對應關係，總共有四的步驟，以遞迴的方式執行直到每一張 UAV image 分析結束。最後，我們會將每一張的 UAV image 對應到 Google Map 上面，以得知現在無人飛機的所在位置以及其飛行所經過的路徑。

整個實驗的流程圖如圖四所示，這裡我們先定義每個參數的意義，UAV image 我們使用 $I_1, I_2, I_3, \dots, I_i$ 來表示；當作 reference map 的 Google Map 影像我們稱為 M ； $H_{i,j}$ 表示將 I_i 對應到 I_j 上面的 Homography Matrix； $H_{i,M}$ 表示將 I_i 對應到 M 上面的 Homography Matrix，得到的圖片稱為 M_i 。所以這邊有兩個方程式：

$$H_{i,j} * I_i = I_j \quad (1)$$

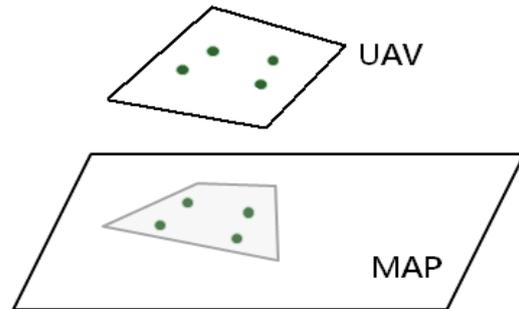
$$H_{i,M} * I_i = M_i \quad (2)$$



圖四、演算法的流程圖[1]

實作的第一步，就是需要先手動初始化我們第一張 UAV image 與 Google Map 的對應關係。正如圖五為例，我們先在 UAV image 與 Google Map 當中點選對應點，以確定這張 UAV image 位於 Google Map 的相對位置。下圖中的綠點代表在 UAV image 與 Google Map 當中對應的點，所謂的對應點，就是影像當中的特徵點，比如說綠點代表一棟明顯的建築物的四個角落，在 UAV image 與 Google Map 中所看見的這個同一棟建築物理論上應該是一樣的，這棟建築試從 UAV image 看是這四個點而從 Google Map 看到地是另外四個點，所以我們在 UAV image 與 Google Map 點選這些對應點，就可以算出這兩張圖的對應關係，也就可以知道這兩張圖片其他所有點的對應關係。

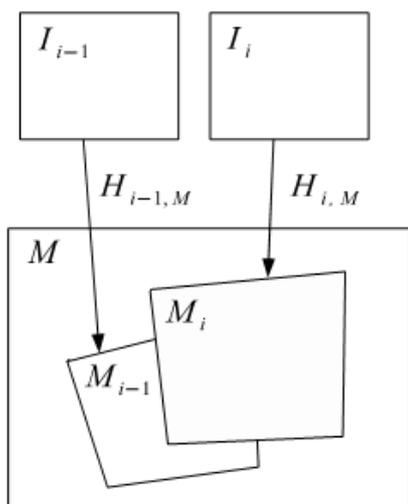
我們將我們所圈選出來的對應點，利用 singular value decomposition (SVD) 算出這兩張圖的 Homography Matrix。這個矩陣在下一個步驟中會繼續扮演著重要的角色。



圖五、手動找出對應點(綠點)，進而找出對應關係[1]

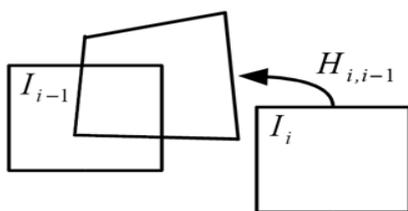
找到第一張圖片與 Google Map 的對應關係以後，我們馬上就可以開始著手進行接下來的實驗步驟。我們這個步驟要作的事情就是讓程式自動的根據上一張圖片的資訊，分析這一張圖片，然後目標求得這一張圖片與 Google Map 的對應關係。

我們可以用下面的圖六來說明，假設我們下一張要分析的圖片為 I_i ，那我們剛剛上一張分析的就是 I_{i-1} ，而我們上一個步驟找到的 Homography Matrix 我們稱為 $H_{i-1,M}$ ，現在的目標就是要找到 $H_{i,M}$ 。這裡主要有四個步驟，第一是先找到 I_i 與 I_{i-1} 彼此之間的關係；第二是利用前一步驟找到的關係粗略地估計出 $H'_{i,M}$ 。第三是根據這個粗略估計的 $H'_{i,M}$ 再進一步加強，找到一個修正的矩陣；最後是將前面三項的結果總合再一起，得到最終的 $H_{i,M}$ 。



圖六、利用前一張的對應關係 $H_{i-1,M}$ ，找出下一張的對應關係 $H_{i,M}$ [1]

接下來找出前後 UAV image 的對應關係，第一步，我們需要先找到 I_i 與 I_{i-1} 彼此之間的關係，也就是 $H_{i,i-1}$ 。這裡我們主要使用的方法是 Scale-invariant feature transform (SIFT)。因為 I_i 與 I_{i-1} 是由同一個攝影機所拍攝的，而且他們之間也是一段連續的圖片，彼此之間的相关性非常的高，所以用 SIFT 可以很正確的找出它們彼此對應的點。當我們用 SIFT 找到兩張圖片大量的對應點的時候，我們可以再使用 RANSAC 來將不正確的對應點刪除，留下正確的對應點，最後再使用 SVD 計算出他們之間的對應關係。如下面的圖七所示。



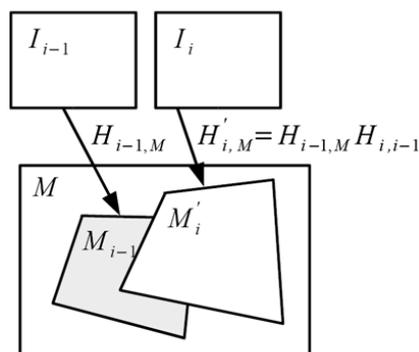
圖七、利用 I_{i-1} 和 I_i 找出 $H_{i,i-1}$ [1]

利用上述找到的 $H_{i,i-1}$ ，我們可以將這張圖片 I_i 對應回到 I_{i-1} ，然後在 I_{i-1} 的空間裡面，我們就可以使用早在第一部就找到的 $H_{i-1,M}$ 將 I_i 對應回到 M 上面。下面圖八為示意圖，我們利用下面這兩個方程式：

$$H'_{i,M} = H_{i-1,M} * H_{i,i-1} \quad (3)$$

$$M'_i = H'_{i,M} * I_i \quad (4)$$

原本的 I_i 經過 $H'_{i,M}$ 對應到 Google map(M) 上面得到 M' ，此時我們投影回去的 M' 是藉著之前我們的 $H_{i-1,M}$ 對應回去的，原本的 $H_{i-1,M}$ 是用來對應 I_{i-1} 的，所以這樣的投影多少會產生一些誤差。但因為這個對應只是一個大概的對應，我們所需要的就是一個大概的對應，接下來我們還要進一步的最影像的分析比對，來修正這些投影產生的誤差，因此我們還需要下一個步驟，幫我們找到一個修正的矩陣。

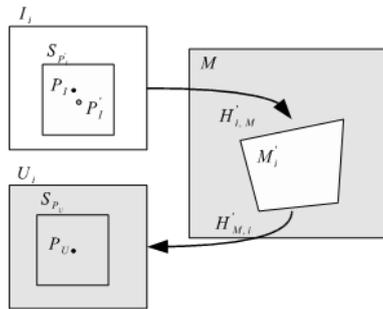


圖八、利用 $H_{i-1,M}$ 和 $H_{i,i-1}$ 求得大約的 M'_i [1]

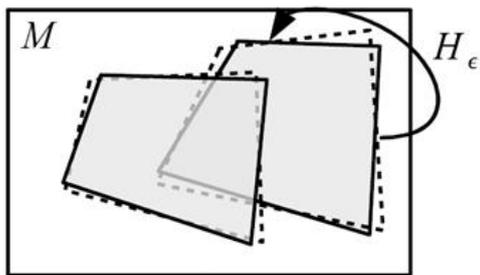
得到上述圖八中的 M' 以後，我們大概知道 I_i 對應到 M 的位置，此時的對應可能會有一些的誤差，我們需要來進一步地修正一下。如圖九所示，我們將 M' 的區域反方向投影到 I_i 的空間，得到 U_i ，然後我們進一步的比對 I_i 與 U_i ，試圖在這兩張影像當中尋找

對應點。

在這兩張影像當中尋找對應點不像上述一樣的單純，因為兩張圖片的差異性太大，像是取像的時間不同，造成兩張圖片的日夜光線變化不同、四季景物不同以及人工建築的差異；還有相機的Sensor不同，導致相機內部參數不同和相機對光線與色彩的感應能力不同。所以我們無法使用像是SIFT來幫助我們尋找對應點。因此，這邊我們主要利用mutual information來強化UAV image與Google Map的對應關係。



圖九、修正比對的示意圖



圖十、得到修正矩陣 H_ϵ 以提升對應正確率[1]

mutual information可以用來量化兩張圖片的相似性，假設我們計算兩張影像的mutual information，就是在測量他們彼此的相似性，換句話說，量化他們彼此有多像，若是mutual

information所計算出來的數值很高，表示這兩張影像非常相像，反之，若是mutual information所計算出來的數值很低，代表這兩張影像非常不相像。

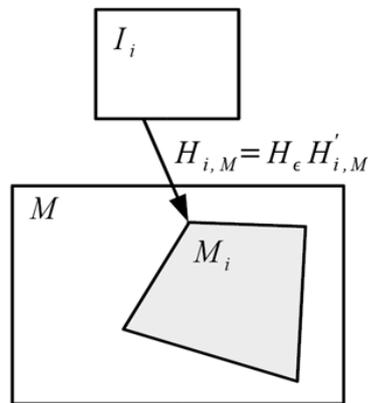
我們在這邊當然不是直接去計算 I_i 與 U_i 的mutual information，我們乃是在 U_i 上面均勻的灑上對應點，然後將這些在 I_i 上面去尋找相對的對應點。我們用圖Figure 3-7(a)來說明， P_U 和 P_I 分別代表在影像 I_i 與 U_i 中的同一個點。若是我們的 $H_{i-1, M}$ 非常的準確，理論上這兩點應該會同一個位置才對，但是 $H'_{i, M}$ 其實沒有那麼的準確，所以我們會再同一個位置找到 P'_I 。但是因為 $H'_{i, M}$ 跟 $H_{i-1, M}$ 很接近，對應到的位置也大約事一樣的，所以 P_I 與 P'_I 應該不會離的很遠，我們的目標就是要在 P'_I 附近找到真正的 P_I 。我們使用的方式是將原本的影像切成較小的區塊 S_{P_I} 和 S_{P_U} 來作分析。 S_{P_I} 和 S_{P_U} 分別是以 P_I 和 P_U 為中心的影像區塊，我們計算這些區塊的mutual information並且作比對，用 $MI(S_{P_I}, S_{P_U})$ 來表示。在 P'_I 附近我們在大量搜尋其他的點， P_J 表示 P'_I 附近的點，計算找出 $MI(S_{P_J}, S_{P_U})$ 最大的值，也就是與 S_{P_U} 最相似的區塊， S_{P_J} 這個區塊的中心點 P_J 就是我們想要找的 P_I 。我們可以用以下的關係式來說明：

$$P_I = \arg \max_{P_J} MI(S_{P_U}, S_{P_J}) \quad (5)$$

因為我們在 U_i 上面均勻的灑上對應點，然後每一個在 U_i 上面灑的點在 I_i 上面也會有一個與它互相對應的點，也就是算出mutual information值最高的點。這樣我們就會有很多組

的對應點，一樣我們可以再次使用 RANSAC 來將不正確的對應點刪除，留下正確的對應點，最後再使用 SVD 計算出他們之間的對應關係。這樣算出來的對應關係就是我們所要的修正矩陣 H_{ϵ} 。

最後，我們將上述步驟中所有算出來的矩陣全部乘在一起，就可以得到最終的 $H_{i-1,M}$ 。如下圖示。



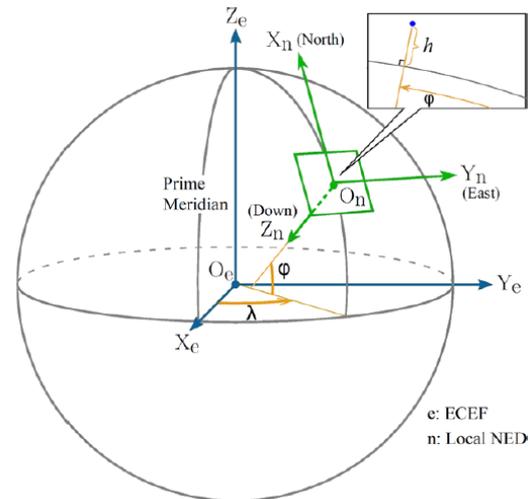
圖十一、最後得出對應較準確的關係 $H_{i,M}$ [1]

接下來，我們可以進行下一張 UAV image 的分析，重複上述的步驟，最終所有 UAV image 分析完了以後，再將這些 UAV image 圖片 warp 回 Google Map。就可以得到飛機全部的路徑。

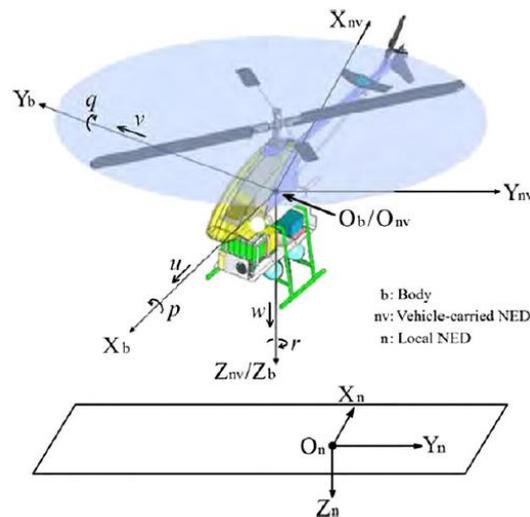
上述的方法是透過 Homography 找出二維影像之間的對應關係，而無人飛機的攝影機必須為下視，才能比較容易達到以上效果，但實際攝影機不一定為下視，也許與無人飛機有個夾角，因此要建立影像與 GPS 的對應關係，就必須利用幾何的關係來建立。

要建立監控影像與全球衛星圖資的幾何關係，首先我們必須要了解幾

個重要的座標系，再利用座標系之間的轉換關係來建立監控影像與全球衛星圖資的對應關係，如下



圖十二



圖十三

A. 大地座標系

大地座標系用於 GPS 導航系統，它記錄了經度、緯度以及高度，如圖十二，分別用 λ 、 ϕ 和 h 來表示。經度 λ 是地球上一個地點離本初子午線以東或以西的度數，本初子午線的經度是 0 度，地球上其它地點的經度是向東到 180 度或向西到 180 度。緯度 ϕ 是指某點與地球球心的連線和地球赤道面所成的線面角，其數值在 0 至

90 度之間。位於赤道以北的點的緯度叫北緯，記為 N，位於赤道以南的點的緯度稱南緯，記為 S。而這裡的高度 h 並不是指海拔高度，是指某點與地球所參考的橢圓球體之間的高度差。

橢圓球體的模型是參考 WGS84 (World geodetic system 84)，此模型的參數有， a 長半徑、 b 短半徑、 f 扁率以及 e 離心率，定義如下

WGS84 Parameters

$$\begin{aligned} a &= 6378137 \\ b &= a(1-f) \\ &= 6356752.31424518 \\ f &= \frac{1}{298.257223563} \\ e &= \sqrt{\frac{a^2 - b^2}{a^2}} \end{aligned}$$

B. 地球中心座標系

地球中心座標系是以地球中心為原點的座標系，如圖十二，此座標系的原點(記作 O_c)位於地球中心， Z 軸(記作 Z_e)指向地球的北極， X 軸(記作 X_e)指向地球經度 0 度和緯度 0 度， Y 軸(記作 Y_e)垂直於 Z 軸與 X 軸(右手定則)。

C. 本地的指北針座標系

本地的指北針座標系是固定在地球橢圓球體表面的座標系，如圖十二，此座標系的原點(記作 O_n)隨機定義在地球表面，但我們通常把無人飛行載具的起飛位置定義為指北針座標系的原點， X 軸(記作 X_n)指向地球的北邊， Y 軸(記作 Y_n)指向地球的東邊， Z 軸(記作 Z_n)指向下垂直於橢圓球面。

D. 載具上的指北針座標系

載具的指北針座標系是指在無人飛行載具上的指北針座標系，如圖十三，此座標系的原點(記作 O_{nv})位於無人飛行載具的重心， X 軸(記作 X_{nv})指

向地球的北邊， Y 軸(記作 Y_{nv})指向地球的東邊， Z 軸(記作 Z_{nv})指向下垂直於橢圓球面。

E. 載具座標系

載具座標系是指無人飛行載具本身的座標系，如圖十三，此座標系的原點(記作 O_b)也是位於無人飛行載具的重心， X 軸(記作 X_b)指向無人飛行載具飛行的方向， Y 軸(記作 Y_b)指向無人飛行載具的右翼， Z 軸(記作 Z_b)指向無人飛行載具的下方。

F. 攝影機座標系

攝影機座標系的原點位於攝影機鏡頭的中心， Z 軸(記作 z_c)就是攝影機光軸的方向，指像無人飛行載具的下方， Y 軸(記作 y_c)指向無人飛行載具的飛行方向， X 軸(記作 x_c)指向無人飛行載具的左翼。

瞭解了以上所有相關的座標系之後，接著就是座標系彼此的轉換關係，如下

A. 大地座標系與與攝影機座標系

座標系之間的轉換需要一個轉換矩陣，轉換矩陣是一個 3×3 的旋轉矩陣和一個 3×1 的位移矩陣組合而成[4]，式(6)表示將大地座標 (X, Y, Z) 經過一個轉換矩陣 T_{World}^{camera} 轉換至攝影機座標系 (x_c, y_c, z_c) 。

$$\begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} = T_{World}^{camera} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (6)$$

B. 攝影機座標系與影像座標系

影像座標系在 yaw 初始有個 180 度的位移，而攝影機座標 (x_c, y_c, z_c) 轉換到影像上的座標 (r, c) ，公式如下

$$r = -\frac{f}{s_x} \frac{x_c}{z_c} + o_r \quad (7)$$

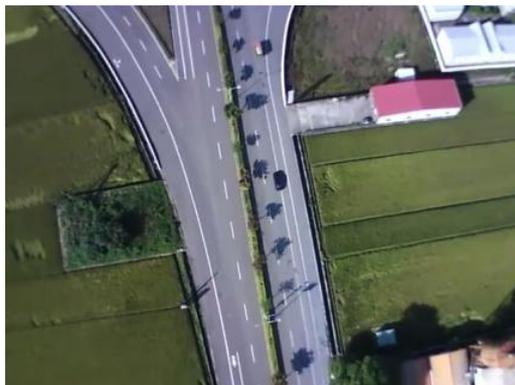
$$c = -\frac{f}{s_y} \frac{y_c}{z_c} + o_c, \quad (8)$$

其中 s_x 和 s_y 是水平和垂直的尺寸係數(公尺/像素)， (o_r, o_c) 是影像座標系的中心點。

經過以上座標系的轉換後就可以建立GPS與影像的對應關係，將GPS座標轉換到無人飛行監控器截取到的影像上。

3. 實驗結果

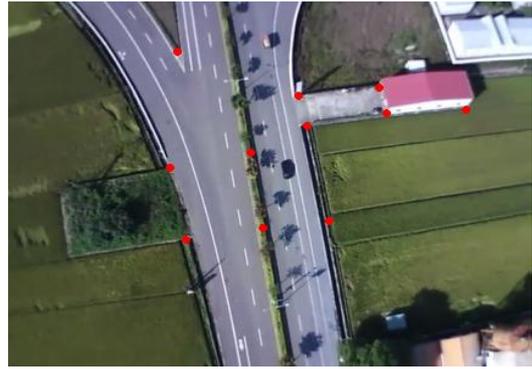
首先是點選對應點的過程，如下圖十四所示。



(a)



(b)



(c)



(d)



(e)

圖十四、(a)測試影像(b)測試影像在 Google map 的相對位置(c), (d)兩張影像點選對應點(e)將圖(a)投影到圖(b)的結果

接下來我們來分析第二張圖片，如下圖十五所示。我們先用SIFT找出第一張圖片與第二張圖片對應的關係，然後疊合成圖(c)。最後將

第二張影像投影到Google map上面得到圖(d)。

圖(d)其實沒有加上我們的修正項，也就是 $H\varepsilon$ 。最後，我們在第二張影像和Google map上面找尋對應點，將上mutual information的資訊，得到最後的結果，如下圖十五所示，圖(c)與圖(d)可以互相作一個比較。



(a)



(b)



(c)

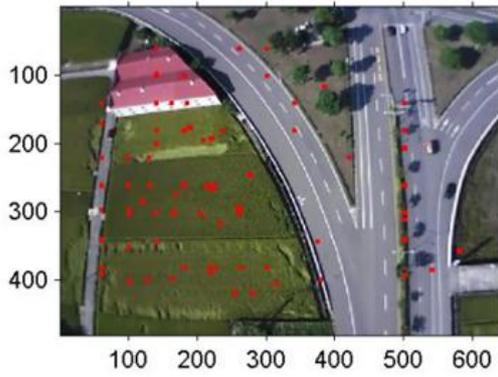


(d)

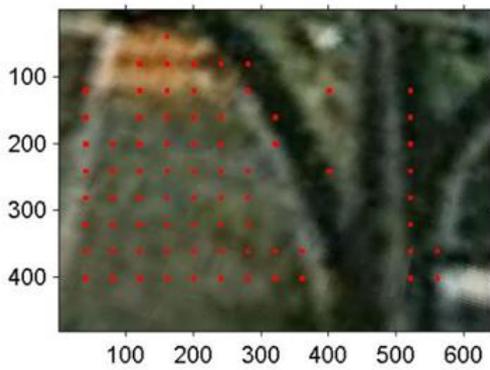
圖十五、(a)第一張測試影像(b)下一張測試影像(c)兩張測試影像用 SIFT 疊合(d)第二張影像投影到 Google map 上面

值得注意的是，在圖十六的圖(c)與圖(d)看起來似乎沒有太大的差異性，這是因為一開始的對應很準，我們初始化的資訊是從第一張開始，在分析第二張圖片的時候，其實沒有太多的誤差累積，所以感覺不出差異。若是我們沒有加入這個修正的 $H\varepsilon$ 進去，則每一次圖片往下一張資料執行時，誤差就會開始慢慢累積，UAV image圖像在用SIFT進行比對時會產生誤差，UAV image與Google Map對的時候也會產生誤差，這樣的誤差持續的累加下來，最後所產生的誤差會非常的大，UAV image和Google Map的相對應位置可能會完全跑掉。

因此，這個修正項就是在確保每一步驟的時候UAV image和Google Map就可以直接的進行一次的比對，因為上一張圖片大概可以告訴我們此時UAV image在Google Map的哪一個位置，我們只要在這個可能的區域作搜尋比對就可以了，所以這樣的演算法的確增加了程式的穩定性。



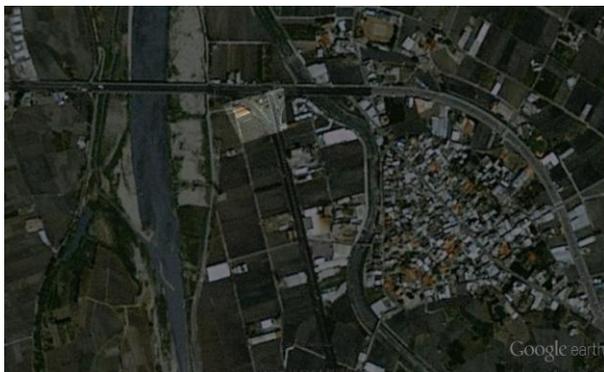
(a)



(b)



(c)



(d)

圖十六、(a)第二張影像的對應點(b)第二張影像在 Google map 上面的對應點(c)第二張影像投影到 Google map 上面(沒有 $H \epsilon$)(d)第二張影像投影到 Google map 上面(加上 $H \epsilon$)

4. 參考文獻

- [1] Yuping Lin, Qian Yu, and Gerard Medioni. "Map-enhanced uav image sequence registration." *In IEEE Workshop on Applications of Computer Vision*, 2007
- [2] Y. Sheikh, S. Khan, M. Shah, and R.W. Cannata. "Geodetic alignment of aerial video frames." *In Video Registration*, 2003.
- [3] SIFT Keypoint Detector
<<http://www.cs.ubc.ca/~lowe/keypoints/>>
- [4] J. J. Craig, "Introduction to Robotics" (Addison-Wesley Publishing Company, Reading, MA, 1989).