


利用紅外線影像之人臉追蹤

研究生：江柏頡

指導教授：莊仁輝 博士

國立交通大學資訊科學與工程研究所

摘 要



在無照明的環境下，紅外線影像可以捕捉物體本身主動散發的輻射熱，其有別於一般可見光影像的特性，能夠發展出不同於可見光影像分析的應用。例如使用紅外線熱影像在黑暗環境中做安全監控時，其影像分析功能之一是進行人臉追蹤的即時處理，在本論文中，是所要探討的課題。我們首先實作影像子空間模型的方法，而不同於一般追蹤所使用的模型，如利用機率或類神經的模型來做人臉追蹤。我們再將一些可能改進的方法加入到影像子空間模型中。我們所加入方法包含運動速度的預測、目標物形狀估算的改進，以及設定臨界值來判定新的影像是否該用於系統更新。實驗結果顯示，這些方法的加入可以使得人臉追蹤達到更精確的效果。

Human Tracking Using Infrared Images

Student: Bo-Jia Jiang

Advisor: Dr. Jen-Hui Chuang

Institute of Computer Science and Engineering
College of Computer Science
National Chiao Tung University

Abstract

In the dark environment, infrared imaging offers a promising alternative to visible light image in various applications owing to its capability of sensing object heat emissions. For example, we can use infrared image to track human face for security surveillances in the dark in real time. In this thesis, we first implement the presented method in [17], that uses an image subspace model. It is different from general tracking models, such as probabilistic or neural network model. In this paper, some adjustments are incorporated into the proposed method, including target motion prediction, bounding box refinement, and outlier detection. Experimental results show that the above approach can indeed improve tracking accuracy.

誌 謝

本篇論文能夠順利完成，首先要感謝我的指導教授莊仁輝老師以及博士班的林泓宏學長。若是沒有老師和學長的熱心指導、建議以及協助，恐怕無法如期完成論文並順利畢業。同時，也要感謝智慧型系統實驗室的各位學長、同學、學弟妹，為我的研究所生涯，除了學習成長之外，更添增了許多歡樂、美好的回憶。

我要感謝我的家人，在這 2 年的碩士生涯，回到家時總是給我支持與鼓勵，也讓我能夠免除經濟上的壓力，專心於課業之上。最後，僅以此篇碩士論文獻給所有關心我的人，謝謝你們的關懷與鼓勵。



目錄

摘要.....	i
1.1 研究動機與目的.....	1
1.2 文獻探討.....	1
1.3 本論文所採用的方法.....	3
第二章 影像子空間的建立.....	4
2.1 系統流程架構圖.....	4
2.2 人臉影像之空間模型.....	5
2.3 子空間之建立— <i>k</i> -means	8
2.3.1 正交軸的建立.....	9
2.4 經由子空間距離計算之追蹤.....	10
2.4.1 樣本的個數.....	11
2.4.2 樣本的位置.....	11
2.4.3 樣本影像的大小.....	13
2.4.4 樣本到影像子空間的最小距離.....	15
2.4.5 子空間的更新.....	15
第三章 人臉影像特徵擷取.....	18
3.1 系統架構圖.....	18
3.2 頭部中心運動方向.....	18
3.3 前景與背景臨界值設定.....	20
3.4 精確的目標物影像切割.....	22
3.4.1 目標物影像的校正.....	22
3.4.2 目標物形狀比例.....	23
3.4.3 目標物影像切割.....	25
第四章 實驗結果.....	27
4.1 頭部追蹤結果比較.....	27
第五章 結論.....	40
參考文獻.....	41

圖目錄

圖 2.1	系統流程架構圖	5
圖 2.2	2 人圍繞梯子活動的紅外線場景配置	6
圖 2.3	對於圖 2.2 右邊人員臉部區域之標示	6
圖 2.4	自圖 2.3 裁剪出的臉部影像	7
圖 2.5	裁剪出來的三十張頭部影像	7
圖 2.6	k -means 演算法產生的 5 個群集中心	8
圖 2.7	Gram-Schmidt 演算法後產生彼此為正交影像軸	10
圖 2.8	以第 30 張影像所產生的藍點為基準	12
圖 2.9	產生出 40 組樣本點	12
圖 2.10	裁剪出 40 張的矩形區域	14
圖 2.11	40 張人臉影像樣本	14
圖 2-12	把目標物影像更新到影像集中	16
圖 3.1	系統流程架構圖	19
圖 3.2	物體中心運動視意圖	19
圖 3.3	當背景居多時的示意圖	21
圖 3.4	框選到的影像做 2 值化處理	21
圖 3.5	第 31 張所框選到的影像	23
圖 3.6	經由放大後的影像。其中頭部影像的左右兩側仍有此許背景存在。	23
圖 3.7	五條水平方向軸的掃描	25
圖 3.8	垂直方向軸的掃描	25
圖 3.9	紅框調整後，變為更精準的黃框，右圖為完整的黃框	25
圖 3.10	用黃框追蹤的結果	26
圖 4.1	2 人手部交錯下紅外線場景	28
圖 4.2	精確頭部切割法，修正後的結果	30
圖 4.3	追蹤頭部持續往攝影機走的情況	31
圖 4.4	3 人隨意走動	33
圖 4.5	因為沒加臨界值判定，所以遇障礙物時追蹤失效	34
圖 4.6	因加上臨界值判定，故遇障礙物時仍可做追蹤	36
圖 4.7	物體在處於較高速下，若無中心運動速度的預測，較易追丟。	37
圖 4.8	若物體加上中心運動速度預測後，目標物體可被正確的追蹤。	39

第一章 序論

1.1 研究動機與目的

目前有越來越多人投入紅外線感測方面的研究，這是因為紅外線影像幾乎不受亮度的影響，可以在明亮或者黑暗的環境中工作。這項特性，使得紅外線感測能夠在軍事、監控、保全、醫療、交通安全和工業製程檢測等方面，提供更多以及更加完善的應用。

紅外線的應用主要分為分光應用與熱應用。光應用的目的為測定、解析物質的紅外線光譜，以瞭解物質的性質。而熱應用的目的又分為兩種：一是利用紅外線進行加熱、加工等；二是利用紅外線感測器，捕捉物體所散發出來的輻射熱，進而測定溫度、監視、追蹤等。在本篇論文中所探討的人物追蹤技術，都是屬於紅外線熱應用的感測部分。例如在夜視上的應用，一台未受監督的攝影機(Unsupervised Camera)可以隨意追蹤一個或多個在移動中的人，當人接近到任何我們所想要保護的區域時，此時監控系統就可以發報出警報聲，代表有突發狀況發生。

1.2 文獻探討

在視訊追蹤上的問題，目前有相當多的演算法都在討論如何做追蹤，一般上，在做人臉視訊追蹤往往不容易達到一個完美的結果。尤其是當碰

到光源(Illumination)、障礙物(Occlusion)、外表(Appearance)改變的問題。而對於這些問題，也分別有相關的論文在討論如何解決之道。

一般在做人臉追蹤時，常被使用的方法是由作者 Isard 與 Blake 所提出的 Condensation 的機率方法[1]，此外也有如動態狀態的估計方法，例如使用 Kalman Filter 的方法[2][3]，Probabilistic Grid-based Method[4]，與使用 Monte Carlo Sampling Methods[5][6]。至於在 3D 物體的追蹤方面，也可以使用基因演算法來做物體追蹤[7][8]。另外使用 Kernel-based Bayesian Filter 來做追蹤也是不錯的方法[9][10]。另外利用子空間來做人臉追蹤也是常見的方法。傳統上都是藉由 PCA 的方法[11]來提昇人臉追蹤的效率。

當在做視訊追蹤時，一般最棘手的問題就是碰到障礙物，一般碰到複雜的障礙物情況下，物體就容易追丟。在 2005 年由 Huang 等人所提出的方法[12]，可以用來對多人物體做追蹤，例如遇到多個物體同時遇到被障礙物遮擋時，仍是可追蹤。其方法是給予編號，再利用基因演算法，來尋找物體在最佳區域的軌跡。隨著時間經過後，當目標物離開障礙物時，也可以清楚的區分出明確的物體，而不會產生物體混淆的結果。而 Han 等人在 2004 年也提出一篇多人追蹤的方法[13]，但它所使用的資料結構是利用樹狀結構，其中每一個節點代表物體在追蹤的狀態，中間節點代表物體有可能會合併(Merge)、分割(Split)、消失(Disappear)、出現(Appear)、誤判(False Detection)等。

此外，偵測(Detection)與追蹤(Tracking)是一體兩面的問題，一般我們都會在影像中先去偵測物體有可能所在的位置，進而在去對這些物體做追蹤。其中 Davis 等人在 2005 年利 Two-stage Template 的方法[14]，來對行動中的人去做偵測。首先使用一般化的樣板[Generalized Template]，來比對影像中有可能出現人的位置，緊接著利用 AdaBoost[15]的方法，自動地去對所有人的可能所在位置逐一去分類。而在此篇方法中，因為是利用紅外線影像，所以物體溫度的高低，也是另一個可參考的依據，來判定人有

可能出現的所在位置。另外[16]在做物體追蹤，由於要預測某物體在三度空間的位置，它會先由物體過去的一些歷史資料，來觀察並預測在此時刻有可能出現的位置，而位置的產生為了要符合自然現象，一般都會搭配高斯分佈的模型去做位置的預測。

1.3 本論文所採用的方法

在本篇論文中，我們主要是採用 Ho 等人的建立影像子空間方法[17]，進行人臉追蹤。我們會在眾多追蹤演算法中，選擇採用子空間的方法，主要是因為傳統的追蹤演算法都較複雜。而此子空間模型，比一般的追蹤演算法更易於了解，實作上也較容易，所以決定採用子空間模型去做人臉追蹤。另外，也希望在完成此模型後，再進一步對此子空間的模型加以作改良，以達到更穩建與及時的人臉追蹤。



第二章 影像子空間的建立

本章中，我們首先將描述 Ho 等人在論文[17]中所提出的人臉追蹤方法，此方法是基於影像子空間模型的建立，而影像空間形成的基底，則為我們所選取的人臉影像所決定。在此章中，我們亦將對 Ho 等人所提的方法做些微的改變。在 Ho 等人所提的方法中，只是框選一張頭部影像，後續影像就可以自動化做頭部追蹤。但在我們的方法中，需先手動框選 30 張人頭影像，才開始執行 k -means 演算法來產生核心影像，進而建立影像子空間，之後影像序列才會逐步的作系統更新，以自動化的方式進行後續的影像追蹤，茲就細部作法的介紹於後。

2.1 系統流程架構圖



首先我們將介紹系統整體運作的方式，而每一個系統區塊的細部內容，會在底下的各小節有細部的介紹。圖 2-1 是介紹整個系統運作的粗略流程。首先我們先手動框選所要追蹤的人頭 30 張，接著我們將 30 張的影像向量去做 k -means 演算法，產生出 5 個具代表性的基底影像向量，進而去建立影像子空間。當子空間建立完成後，為了要偵測下一個時刻頭部的位址，就需產生適量的影像樣本，在此我們先產生 40 個影像樣本出來。一旦影像樣本產生完後，便對產生出來的影像樣本，逐一去對最近建立的影像子空間去算距離，找出與影像子空間距離最小的影像樣本出來，再從影像集之中去刪除最舊的影像，並把新近追蹤到的影像樣本更新到影像集中。

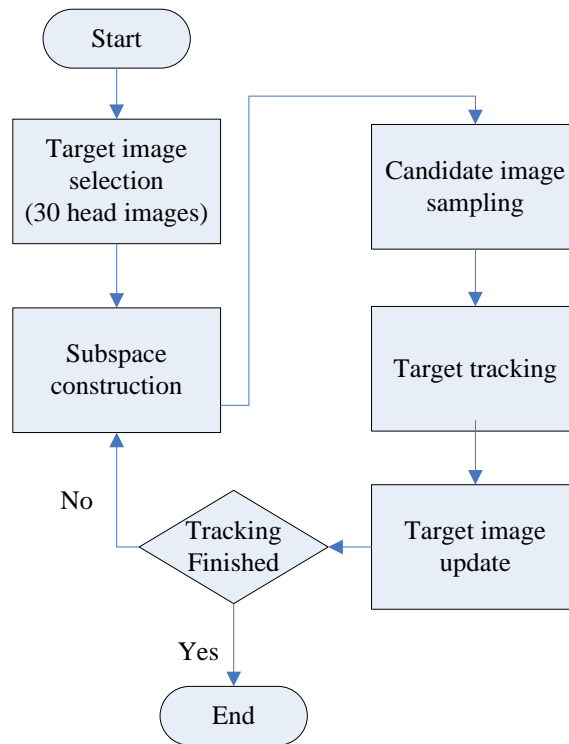


圖 2.1 系統流程架構圖

2.2 人臉影像之空間模型

在人臉追蹤(face tracking)的方法中，有不少可用的模型，包括機率、類神經、子空間模型等。其中使用子空間模型是較方便且省時的方法，所以我們決定採用之。在此我們利用紅外線攝影機，在不同處所做紅外線影像的拍攝，在此我們先以圖 2.2 中，2 個人在梯子附近所拍攝的紅外線影像序列做人臉追蹤。

2.2.1 手動產生 30 張影像

假設我們想要追蹤在圖 2.2 中右方帶眼鏡的人，需先手動做人臉影像的裁剪，如圖 2.3 所示。，如此以一個矩形，裁剪出的臉部影像如圖 2.4 所示。接著依此方法對於連續 30 張影像進行類似處理，直到裁剪出 30 張影像人頭。在此因每張裁剪出來的人頭大小會不一，因此我把人臉影像的尺

寸(size)重新都調整為 19×19 的 2 維影像, 裁剪出的臉部影像如圖 2.5 所示。而影像大小重新調整的目的, 也是為了方便接下來執行 *k*-means 演算法來建立臉部影像子空間的工作。



圖 2.2 2 人圍繞梯子活動的紅外線場景配置



圖 2.3 對於圖 2.2 右邊人員臉部區域之標示



圖 2.4 自圖 2.3 裁剪出的臉部影像



圖 2.5 裁剪出來的三十張頭部影像

2.3 子空間之建立— k -means

在裁剪完 30 張頭部影像和做過影像大小調整後，到目前為止，我們會有 30 張 19×19 的影像集，而為了要方便算執行 k -means 演算法，我們會先把 19×19 的 2 維影像轉換為 361×1 的 1 維的向量。在此實驗中，我們把 k 的值設定為 5，代表我們把 30 張的人頭影像，彙整出 5 張具代表性的核心影像來表現全部的影像集，而此 5 張具代表性的群集影像所展現出來如圖 2.6。



圖. 2.6 k -means 演算法產生的 5 個群集中心

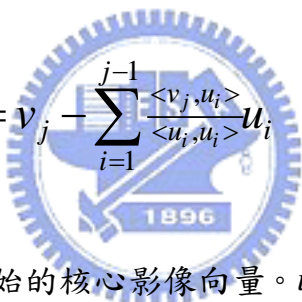
本論文中會利用 k -means 演算法，主要目的是降低影像子空間的維度，原本有 30 張的影像，我們可以把它看成是 30 維 361×1 的影像空間，我們以 k -means 演算法來產生 5 個最具代表性的軸。在此我們把它定義為 V_i : $i=1, \dots, 5$ 。它們代表 5 個軸的方向向量。也把影像空間由 30 維轉換成 5 維的影像子空間，這麼做的目的可節省計算量的時間，而在 k 值的取決也因人而異，我們認為 k 取 5 為所要建立影像子空間的折衷方法。

在此註明， k 值設定因人而異，若給定較大的 k 值，將會產生出更多的核心影像，相對所建出來的影像子空間也變得更大，而在預測的階段上，可以提升追蹤的精確度，但在時間上的耗費，也相對的提高。反之如果 k 取得太小，核心影像空間太小，也導致人臉追蹤的精準度降低。

2.3.1 正交軸的建立

經由 k -means 演算法產生出的 5 張核心影像，若把它轉換到向量空間上，此 5 張核心影像分別代表影像空間中，5 個方向的座標軸，但此 5 個座標軸未必彼此為正交，之後要拿影像樣本(Sample)來計算到影像子空間的最短距離時，計算複雜度較高，所以在此我們需把 k -means 所產生出的 5 個核新影像軸，重新再做校正，另外產生出新的五張正交影像。

產生 5 個正交影像軸的計算過程中，我們主要是使用 Gram-Schmidt 演算，使用公式如下(2.1)式:


$$u_j = v_j - \sum_{i=1}^{j-1} \frac{\langle v_j, u_i \rangle}{\langle u_i, u_i \rangle} u_i \quad (2.1)$$

其中 $v_j, j=1, \dots, 5$, 代表原始的核心影像向量。 u_j 代表計算所得正交的影像向量。在(2-1)式子中，主要是把 v_j 於各軸向的投影量給扣除，使得 u_j 彼此成為正交的影像向量(Orthogonal Vector)。圖 2.7 是經由 Gram-Schmidt 演算法後所產出的正交影像。

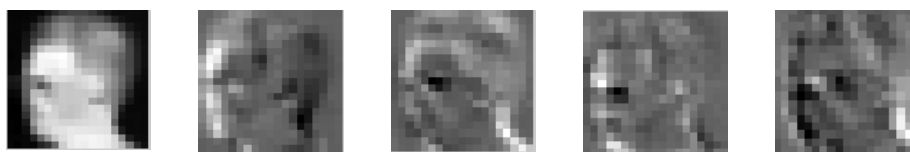


圖 2.7: Gram-Schmidt 演算法後產生彼此為正交影像軸

在獲得影像正交軸後，我們再把這些正交軸轉變成單位正交軸 (Orthonormal Axis)。產生單位正交軸的目的，在於方便計算影像樣本到影像子空間的距離。另外在 Gram-Schmidt 計算過程中，仍是以 361×1 一維的影像向量做計算，一般只有在顯示的過程中，我們才會把影像轉換成 19×19 的 2 維影像作顯示。

2.4 經由子空間距離計算之追蹤

子空間的產生，主要是基於希望能以少數的影像向量所張出的子空間，來代表整體影像集的概念。我們所產生的影像樣本未必都恰好落於此一影像子空間上，當樣本落於子空間外時，可以找出與所建立的影像子空間，相似度最高的樣本出來，也相當於在子空間中二者距離最小的樣本，做為目前所追蹤到的人臉影像，進而重新更新影像子空間。如此不斷地更新子空間，便可以產生出符合所要追蹤的物體子空間。

2.4.1 樣本的個數

首先我們手動剪裁完前 30 張影像後，接下來是對第 31 張之後的人臉影像去做自動追蹤，我們在預測第 31 張人臉影像位置時，是在之前第 30 張影像所裁剪到的位置的附近，產生適當數量的影像樣本。而影像樣本數產生的多寡，也會影響人臉追蹤的速度。為了保有追蹤的即時性，在此我們產生影像樣本數是 40 個。

2.4.2 樣本的位置

有關影像樣本的產生，我們是以第 30 張所框選到的影像為基準，以此張人臉影像左上角位置為基準點，如圖 2.8 標示的藍點，而後我們是產生 40 組 x 與 y 座標的高斯分佈(Gaussian Distribution)。在高斯分佈中，我們把平均值設定為 0，標準差設定為 5。如此設定的理由是可以對 x 位置的左右範圍與 y 位置的上下範圍一併考量進來。若離基準點距離愈遠，所出現的機率就應該愈低，而高斯分佈的模型較符合此情形，所以我們決定採用此模型。圖 2.9 是我們產生 40 組樣本點的位置。



圖 2.8 以第 30 張影像所產生的藍點為基準



圖 2.9 產生出 40 組樣本點

在此我們對樣本位置有做進一步的臨界值判定，如果樣本點 x 座標與 y 座標位置離基準點的距離大於 3 倍的標準差，則捨棄此取樣點，再重新產生新的取樣點。此一作法，是為了因應在連續影像中，作頭部追蹤時，人臉移動的位置不可能瞬間有太大改變，所以加上此判定，可以確保我們所產生的影像樣本，都落在可靠的範圍內，以增加追蹤的穩健性。

另外當我們在產生可能的人臉位置樣本時，因為我們在統計模型中使用

的分佈是常態分佈，所以在平均值與標準差的設定就顯得格外重要了。而對於不同移動速度的物體，有些是以高速移動，例如火車、飛機等，其目前時刻的位置與下一時刻的位置，變化會非常的大，所以設定的標準差就需相當大。但目前一般我們所追蹤的物體，多半是在適當距離範圍內走動中的人，我們所設定的平均值為 0，標準差為 5。對於不同的情境下，我們會因為追蹤物體速度的不同，而去設定不同的標準差。

另外要強調的是，為何我們統計中的模型不使用均勻分佈(Uniform Distribution)主要原因是均勻分佈在選擇範圍內，對於不同遠近所預測的位置，機率都是相同。這樣的假設較不符合常態性。一般的情況是，愈靠近物體的位置，所出現的機率應該愈高，而在範圍內愈遠離物體的位置，所出現的機率應該愈低。所以目前我們認為最合適的模型就屬於常態分佈，但我們只取在常態分配中，所定的標準差在 3 倍以內的值，若產生的值，範圍超過 3 倍標準差，則將之捨去，再重新產生新的預測樣本。

2.4.3 樣本影像的大小

當 40 組樣本產生後，我們就以樣本點所產生的位置，去裁剪各式各樣的矩形區域。在矩形長與寬的考量，是對長與寬做 0.95~1.05 倍的影像縮放，而長與寬的縮放比，所採用的分佈是使用均勻分佈。圖 2.10 是經由均勻分佈，裁剪出來 40 組可能的矩形區域。當 40 組可能的矩形區域裁剪完後，我們可以得到下一個時刻頭部可能出現的樣本位置，如圖 2.11 紅框所標示的頭部影像。



圖 2.10 裁剪出 40 張的矩形區域

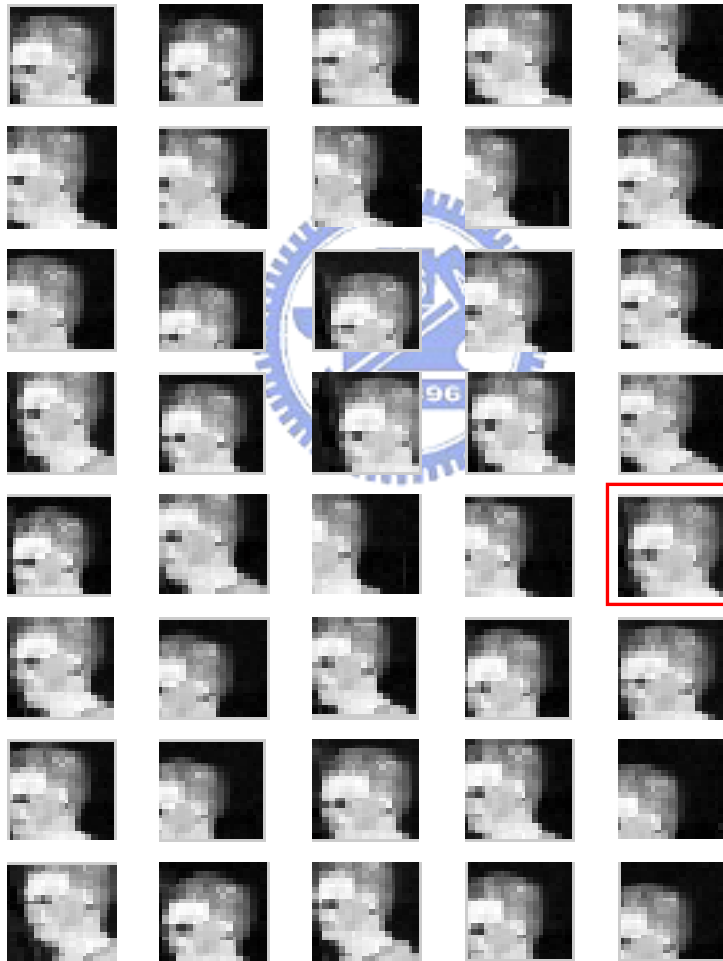


圖 2.11 40 張人臉影像樣本

2.4.4 樣本到影像子空間的最小距離

在建立 40 張影像樣本之後，接下來就是把每張影像樣本，與先前所建立的核心影像子空間，去做距離的比對。因為 k -means 演算法中，當初 k 值設定為 5，所以我們所建立子空間是 5 維，而計算影像樣本到子空間的最小距離如下(2-2)式：

$$\arg \min_i \left| \vec{x}_i - \sum_{j=1}^k \langle \vec{u}_j, \vec{x}_i \rangle \vec{u}_j \right| \quad (2-2):$$

其中 \vec{x}_i ， $i=1, 2, \dots, 40$ 為第 i 個的樣本影像向量。 \vec{u}_j 為單位正交軸向量，而 $\langle \vec{x}_i, \vec{u}_j \rangle$ 為第 i 個樣本影像向量與第 j 個單位正交軸的內積。由此公式，我們把 \vec{x}_i 影像向量扣除它落於子空間 5 個軸上的投影量，剩下的分量長度即為 \vec{x}_i 到子空間的距離。我們可從此 40 個影像樣本中，找出某一個影像樣本到子空間的距離最小的，作為目標物追蹤的結果。從圖 2-11 中，我們可以發現，用紅框標示的第 25 張頭部樣本影像，即是我們所要追蹤的目標影像，同時我們也將其更新到 30 張頭部影像集中。

2.4.5 子空間的更新

當我們找到與子空間距離最小的影像樣本後，接著就把先前建構的 30 張影像集做更新。亦即當我們得出第 31 張所追蹤到的目標影像時，就把它取代到原始影像集中最舊的一張，此一影像更新的動作如圖 2-12 所示。



圖 2-12 把目標物影像更新到影像集中

當影像集更新完畢後，接下來再重新在執行 k -means，產生新的的群集中心(也就是核心影像)，之後重做 Gram-Schmidt, 得出新的正交軸，形成新的子空間，而後再重復以產生新的影像樣本方式進行目標物追蹤。實驗結果發現，如果是在單純的背景做追蹤，其效果都算相當不錯。



第三章 人臉影像特徵擷取

在本章中，我們對於 Ho 等人的方法加入一些改進，包括追蹤目標物運動的預測、目標物影像的精確切割，和在較複雜的背景情況之下，判別目標物影像是否適用於子空間更新。這些方法的使用，主要都是希望在做人臉追蹤時，能使頭部追蹤更為精確，同時也希望能夠達到即時(Real-time)追蹤的效果。

3.1 系統架構圖

在圖 3.1 系統架構圖中，除了先前實作論文[17]的系統架構外，我們還額外加入改進的方法，這些方法如圖 3.1 中用紅框所標示，其中，對於目標物運動預測，我們是把物體的運動方向依慣性作簡單估計，會使得物體追蹤時，更容易鎖定其移動的位置。而關於前景背景臨界值的設定，則是用來判定所追蹤到的目標物影像，是否有受到背景干擾遮蔽，以決定該目標影像應否更新到子空間中。最後系統右下角區塊的目標物影像切割的改良，底下分別對這些改進方法，做細部的介紹。

3.2 頭部中心運動方向

論文[17]的方法在一般情況下，可以得到不錯的追蹤結果，但倘若目標物作快速移動，由於產生的高斯樣本標準差設定為定值，所以容易發生追丟情況，因此我們把物體中心運動(center sport)的預測也加進來，當追蹤的低速物體時，影響並不大，但對於高速物體追蹤，便可以追蹤獲得改善，而不會造成物體追蹤失效。圖 3.2 是物體中心運動的示意圖。其中的 3 塊矩形目標物內，紅點由左至右分別表示 $t-1$, t , $t+1$ 時刻,物體的中心點位置。

而在 $t+1$ 時刻的紅點則是我們所要預測的位置，其預測的公式如(3.1)

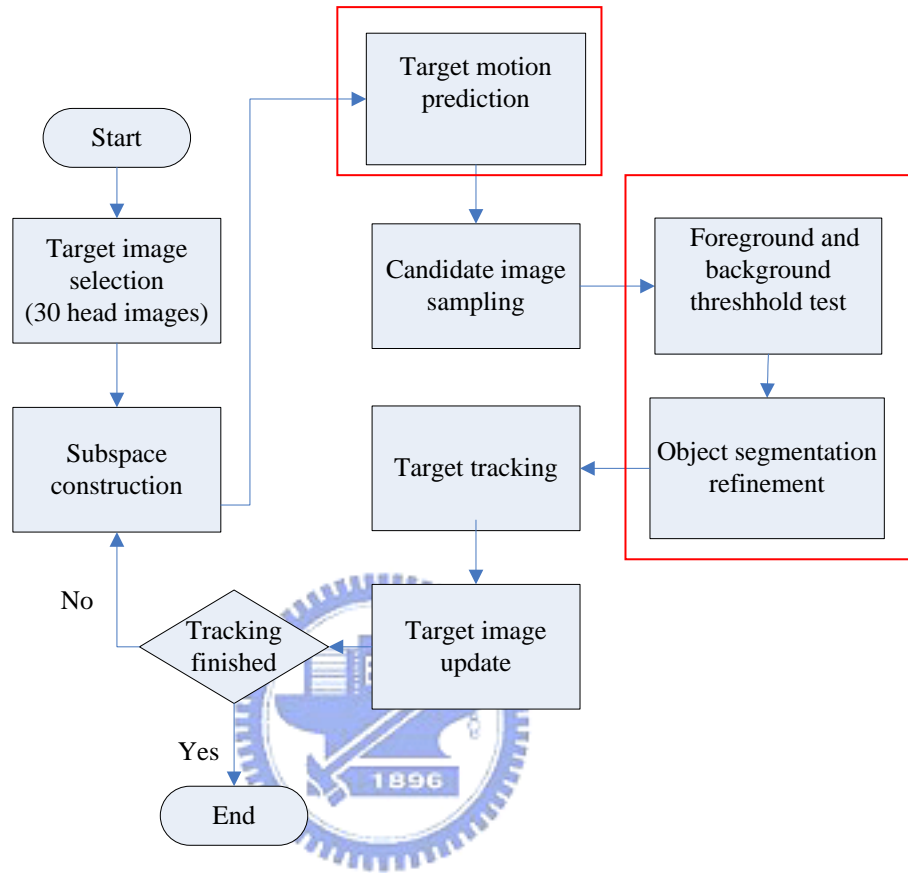


圖 3.1 系統流程架構圖

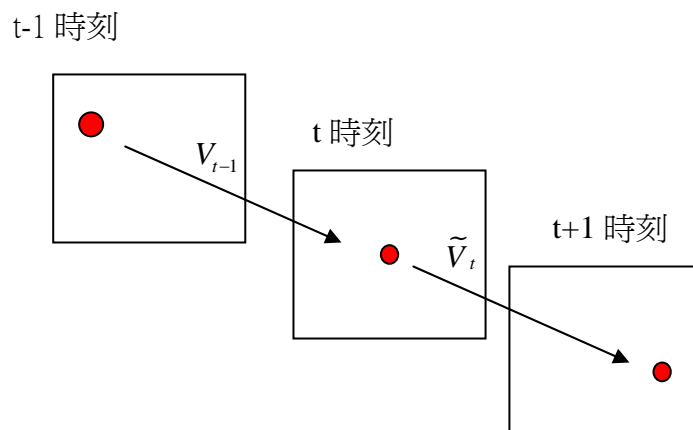


圖 3.2 物體中心運動視意圖

$$\tilde{V}_t = V_{t-1} \quad (3.1)$$

$$V_{t-1} = X_t - X_{t-1} \quad (3.2)$$

$$\tilde{X}_{t+1} = X_t + \tilde{V}_t \quad (3.3)$$

其中 X_t 表是物體在 t 時刻影像上所在的位置， \tilde{X}_{t+1} 表示物體在 $t+1$ 時刻所預估的位置， \tilde{V}_t 為 t 時刻所估計的速度。此速度的估計為(3.2)式，是由先前物體移動的位置計算出來(依慣性原理)。在此我們想要預估目標物在 $t+1$ 時刻的位置，此位置就會等於 t 時刻的位置 X_t 加上 t 時刻所預估的速度方向 \tilde{V}_t 。

此一運動預測的方式，有別於如 Kalman filter 作運動估測的方法，此外子空間追蹤模型與 Kalman filter 可看成是對等的追蹤模型，並不適合混在一起使用，而我們僅以簡單的物體運動慣性估計來預測下一時刻人臉的位置。

3.3 前景與背景臨界值設定

在背景複雜或發生目標物遮掩的情況下，容易發生所追蹤到的目標物影像不精確的情況，如圖 3.3 所示，背景影像佔整張影像比例過高，而後在更新核心影像時，便會將背景比例居高的影像更新進來，使得之後的人臉預測失準，為了不讓背景居多的影像更新進來，我們設定影像中背景與前景比例的臨界值，以判斷是否將追蹤目標物影像更新到子空間中。而對於

影像前景與背景的判別，我們有做 2 值化的處理，如圖 3.4 所示，黑色的部份為我們所定義的背景，白色的部份則為前景。此一二值化的處理是以所框選到的矩形部份，將其中的影像亮度值加總除以像素點個數作為判斷前景與背景的依據。如果落於框內的像素亮度值大於此平均值，則定義此像素為前景(白色)，反之視為背景。



圖 3.3 當背景居多時的示意圖



圖 3.4 框選到的影像做 2 值化處理

而在判定目標物影像是否更新到影像集時，採用的公式如(3.4)式

$$\frac{C_b}{C_f} > T \quad (3.4)$$

其中 C_f 代表前景的像素個數， C_b 代表背景的像素個數，而 T 是我們所定義的臨界值，此臨界值設定為 2.2。當目標物影像計算出的結果小於臨界值，表示影像處於背景居多的情況下，此時就不把影像更新到原先的影像集裡，以免 k -means 產生的核心影像會有所偏差。反之若大於臨界值，則代表狀況良好，可將目標物影像更新影像到影像集中。

3.4 精確的目標物影像切割

在論文[17]的方法中，由隨機樣本中所找出的目標物追蹤影像，其影像框選範圍由於是隨機產生，所以若仔細觀察之，目標物影像仍是有部份的背景存在，為了使得切割出來的影像能夠更精確，底下小節我們提供了目標物影像校正的方法。

3.4.1 目標物影像的校正

當經由比對隨機影像樣本與子空間距離，找出最有可能的追蹤目標物影像後，其初步目標物影像切割如圖 3.5 所示，從裡頭所框選到頭部影像來看，對於頭部的左側與右側，仍有少量的背景存在，放我們加入精準目標物影像切割的方法，可使切割出來的影像更為緊密。同時也避免直接將圖 3.6 所框選到影像，直接更新到影像集裡，而會使得往後的頭部影像集，變得愈來愈不精確。



圖 3.5 第 31 張所框選到的影像

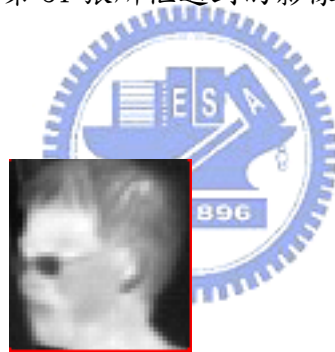


圖 3.6 經由放大後的影像中，頭部影像的左右兩側仍有此許背景存在

3.4.2 目標物形狀比例

為了讓影像能做到更精緻的切割，底下我們提供一個做法。首先把所框選到的目標物矩形，如圖 3.7 所示，由矩形高度的中點，以 5 個像素為一個高度單位，尋找所要做水平方向掃描的最高的位置，並由上往下對水平方向去做掃描，接著就可產生如圖 3.7 (b)(c)(d)(e)的水平掃描線。在圖 3.7(a)

中，左右 2 端標示到的第一個綠點，就視為影像前景部份。經過 5 條水平方向的掃描之後，總共會產生 10 個綠點，由這 10 個綠點的影像座標，我們可以找出其中相距最寬的 2 個綠點，視為要校正為精確切割的影像寬度。至於新的影像裁剪高度，我們設定為寬的 1.2 倍，圖 3.8 則是同理經由中心垂直方向所掃描出綠點的結果，此垂直方向的掃描的範圍，是由原本紅框的頂線，向上延伸 10% 的距離開始往下掃描，直到偵測到第一個綠點就停住，接著向下延伸出寬度的 1.2 倍而得出切割之高度，即可找出新的影像切割長寬範圍。

關於綠點的偵測，在此我們設定臨界值為判定標準，其臨界值計算如(3.5)式，其中 T_L 代表每條掃描線綠點搜尋的臨界值， I_{\min} 代表每條掃描線的亮度最小值，而 I_{\max} 代表每條掃描線的亮度最大值。如圖3.8中有5條水平掃描線，我們從左至右進行像素掃描，若像素的亮度大於 T_L ，則我們就將此像素上標示為綠點，且停止做掃描。同理若從右向左掃描也會產生另一個綠點，在此方法中，水平掃描線數目，可以動態做調整，假設產生 N 條水平掃描線，則有 $2N$ 個綠點，最後在矩形寬度中點上還會有一條垂直掃描線，所以共可有 $2N+1$ 個綠點。

$$T_L = I_{\min} + \frac{I_{\max} - I_{\min}}{5} \quad (3.5)$$

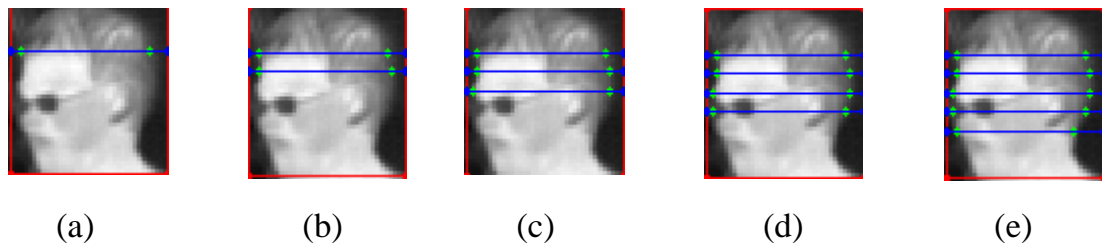


圖 3.7 五條水平方向軸的掃描

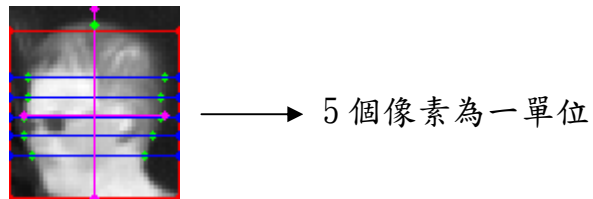


圖 3.8 垂直方向軸的掃描

3.4.3 目標物影像切割

當我們重新調整過目標物影像的比例後，接下來就是對原本的紅框做調整，經由上小節的校正方法，就可以產生出新的黃框出來，如圖 3.9 所示。往後的影像集更新，就以黃框所標示到的範圍，做為更新內容。圖 3.10 是用黃框追蹤的情況下，實際場景的追蹤結果。



圖 3.9 紅框調整後，變為更精準的黃框，右圖為完整的黃框



圖 3.10 用黃框追蹤的結果

經由我們的方法校正後，影像經過一段時間的追蹤，我們可以發現，人臉追蹤變得更為緊密與穩健，使得我們在做頭部追蹤過程，較不容易丟丟。



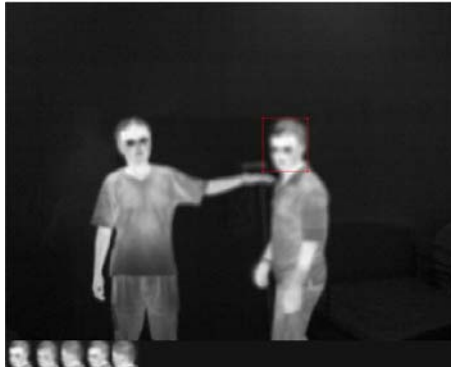
第四章 實驗結果

在實驗結果中，首先我們使用 Matlab 7.0 來模擬其結果，來驗證論文 [17]中所提及的方法，當實作完後，由於在 Matlab 平台上並不容易達到人臉追蹤的即時性，所以我們把論文方法移植到 Microsoft Visual C++ 6.0 以 MFC 做開發。本實驗將測試 Ho 等人所提出的方法與我們自己所加的改良，由實驗結果觀察，改良後可得出更精確的人臉追蹤結果。

4.1 頭部追蹤結果比較

例 1 2 人手部交錯下紅外線影像

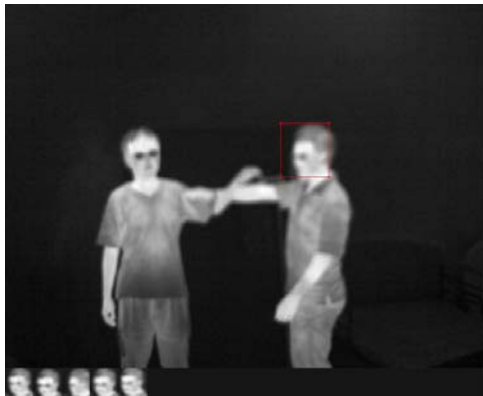
圖 4.1 是在 2 人手部交錯下的紅外線影像追蹤，我們所要追蹤的是位於圖中右方的人，當執行完 Ho 等人的演算法後，可以從圖 4.1(a)中看到，*k*-means 演算法產生的 5 張核心影像在圖中左下角的位置。接著從圖 4.1(b)觀察，我們會發現圖中所框選到的頭部仍不夠緊密，框到範圍左右 2 方仍有部份的背景存在，而從 *k*-means 中的結果發現，核心影像的頭部切割仍是不夠精密。到了圖 4.1(c)時(第 26 張)，因為之前追蹤到的頭部影像不夠精確，在後續更新至影像集時，造成 *k*-means 找出 5 張核心影像的頭部有些不完整，我們可以從圖 4.1(c)中看到 *k*-means 影像集中的第 3 張頭部影像，右邊有些微頭部不完整。到圖 4.1 圖(d)時，在 *k*-means 中的第 2 與第 3 張，頭部都顯得不夠完整，如此下去將會使得核心影像集變得愈來愈不精確。



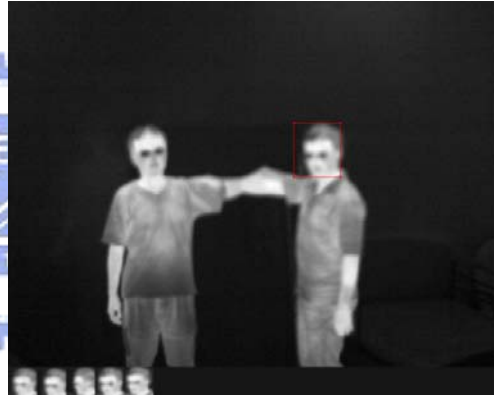
(a) 第 1 張



(b) 第 5 張



(c) 第 26 張



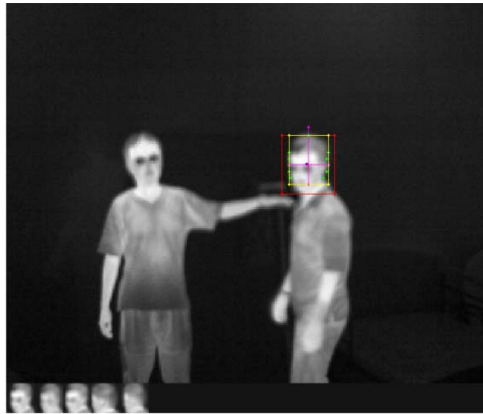
(d) 第 44 張

圖 4.1 2 人手部交錯下紅外線場景

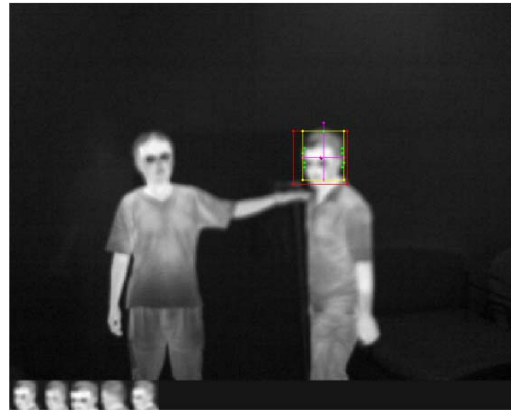
經由頭部精確切割方法的加入後，我們就可以從圖 4.2(a)中，看到原本所追蹤到的紅框，變為更緊密的黃框，在更新人臉影像到影像集時，會以黃框框選到的範圍來做更新，之後經由 k -means 產生的影像子空間，將會變得更精密，使得往後所要追蹤的結果變得更精準。我們從 4.2 圖(c)(d)來觀察 k -means 的結果，我們可以發現，之前不完整的頭部切割被解決了，此時在 k -means 中的核心影像，都變得更加精準。

例 2：單純背景下，2 人隨意走動下之紅外線影像

此一實驗中，我們所要追蹤的是位於圖 4.3(a)裡頭中間偏左的人，同樣地當加入精密頭部切割的方法後，圖 4.3(a)(b)黃框所選框到的範圍，都比紅框來得好，其中在圖 4.3 (b)中，因為人是逐步往攝影機的方向走，所以頭部會變得愈來愈大，而由紅框初步的框選，對於頭部的頂端仍有部份被裁掉，但加入我們方法後，黃框所改善切割的結果，頭的頂部會變得更加完整。當經過一段時間後，如圖 4.3 (c)(d)，人一直往攝影機鏡頭走， k -means 所產生的核心影像都相當不錯。



(a) 第 1 張



(b) 第 5 張

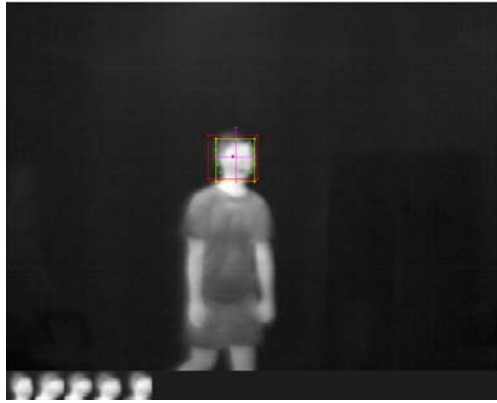


(c) 第 26 張

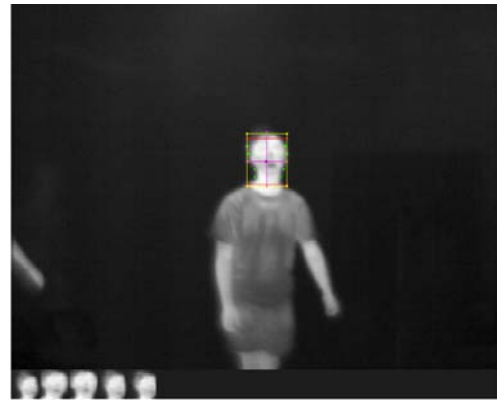


(d) 第 44 張

圖 4.2 精確頭部切割法，修正後的結果



(a) 第 1 張



(b) 第 24 張



(c) 第 56 張



(d) 第 108 張



圖 4.3 追蹤頭部持續往攝影機走的情況

例 3：3 人隨意亂走動之紅外線影像

在此場景下，我們所要追蹤的人位於圖 4.4(a)中間偏右的位置，而此人的動作是把雙手舉起，接著做轉身。我們可從圖 4.4(a)(b)(c)(d)看出，此人是依順時針方向做旋轉，而在這個情況下，臉部目標會有很大的改變，再從圖 4.4(e)(f)來看，原本位於圖中，另一位非追蹤目標的人物，逐漸往追蹤目標人物的方向走近，使得我們所要追蹤的目標人物被覆蓋在後方，如此一來，我們的追蹤演算法會失效，此時人臉追蹤結果變成去追蹤位於前方的人。

例 4：2 人走動，圍繞梯子之紅外線影像

底下實驗我們以 2 人圍繞梯子為例，在圖 4.5 中間有一張梯子，而我們想要追蹤位於中間的人，採用的是 Ho 等人的方法，第一張的追蹤影像如圖 4.5(a)，因為此人在行進中的速度並不會太快，且此人一開始都位於梯子的前面，所以初步追蹤到的結果都相當不錯，如圖 4.5(b)，我們也可以從 *k*-means 的結果看出，開始有頭部背面的影像逐步更新進來，直到圖 4.5(c)，此人開始往梯子後方走時，此時主要的前景變為梯子，而人臉的部份就變少了許多，此時追蹤開始失效，如圖 4.5(d)，而後續追蹤到的綠框也將失效。當人離開梯子沒被遮掩後，因之前的 *k*-means 已產生錯誤的核心影像(包含梯子的影像)，此時所追蹤到的物體反而變為梯子，而不是人臉，如圖 4.5(e)。



(a) 第 1 張



(b) 第 9 張



(c) 第 12 張



(d) 第 16 張

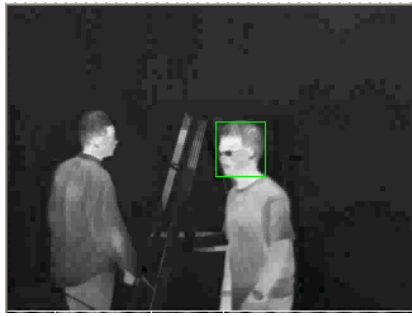


(e) 第 20 張

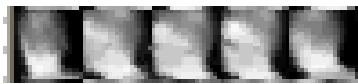


(f) 第 32 張

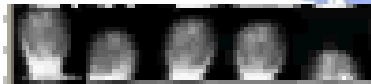
圖 4.4 3 人隨意走動



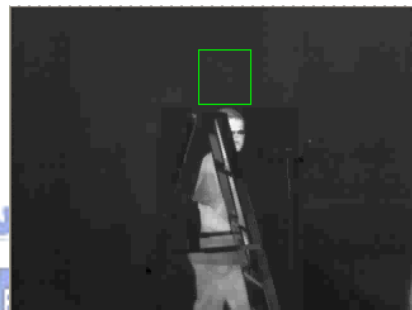
(a) 第 1 張



(b) 第 34 張



(c) 第 62 張



(d) 第 68 張



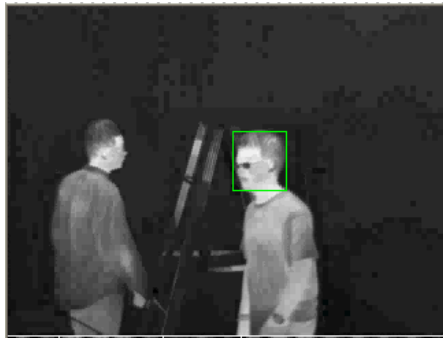
(e) 第 75 張

圖 4.5 因為沒加臨界值判定，所以遇障礙物時追蹤失效

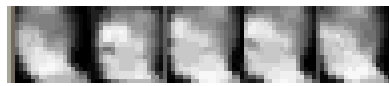
為了克服上述的問題，我們將背景與前景臨界值比例加進來，如前章所述，如果背景佔整張影像比例過多時，此時就不把背景居多的追蹤結果更新到影像集，以免執行 k -means 演算法後，產生不正確的核心影像，進而影響追蹤結果。在圖 4.6(a)(b)，一開始追蹤效果都跟前圖 4.5(a)(b)類似，但當到了圖 4.6(c)時，因我們有加入臨界值的判定，此時所追蹤到的目標物影像並不會更新到影像集，持續到圖 4.6(d)時，這中間過程所框選到的綠框，都是背景居多的情況，所以都不做子空間更新，藉此以保有完整的頭部。當到圖 4.6(e)(f)，追蹤結果的綠框又逐漸框住人臉，此時在綠框框選到的範圍，經由計算就會小於臨界值，而把目前所追蹤到的目標物影像，更新到影像集，如此一來便可繼續追蹤下去。

例 5: 手中拿著紅外線 LED 在室內移動

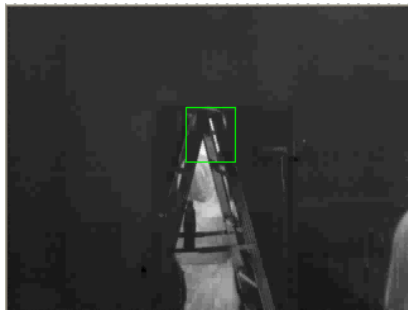
此一實驗，我們在晚上以手中拿著紅外線 LED 燈為例作追蹤，此處我們所設定的高斯隨機取樣的標準差為 5(都與前面例子相同)。在此我們所要追蹤的是拿著 LED 的手，如圖 4.7(a)(b)所示，所採用的是 Ho 等人的方法，一開始追蹤的結果都算穩定的，但當到了圖 4.7(c)(d)時，因物體在移動上的速度瞬間變得較快些，此時物體追蹤的框就變得有些偏離，而最後到圖 4.7(e)時，因前面累積了追蹤誤差的緣故，使得物體追蹤失效。



(a) 第 1 張



(b) 第 27 張



(c) 第 66 張



(d) 第 78 張



(e) 第 90 張

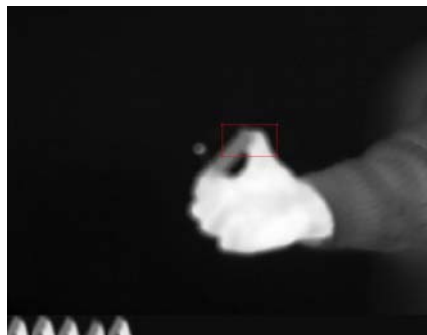


(f) 第 100 張

圖 4.6 因加上臨界值判定，故遇障礙物時仍可追蹤



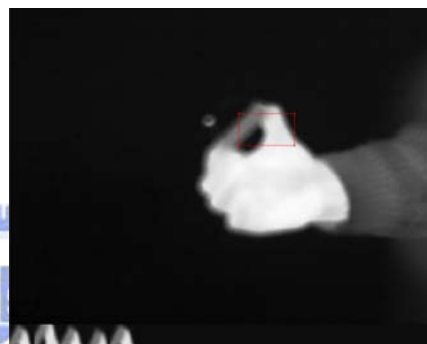
(a) 第 1 張



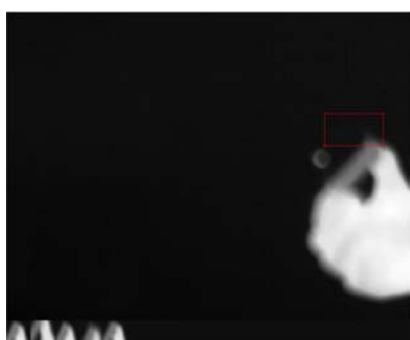
(b) 第 6 張



(c) 第 7 張



(d) 第 8 張

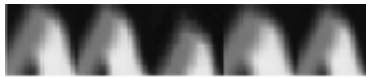
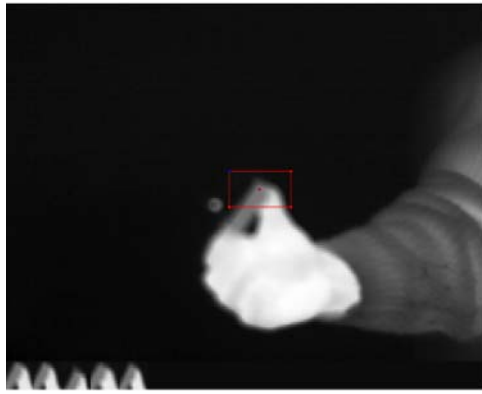


(e) 第 16 張

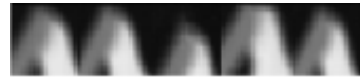
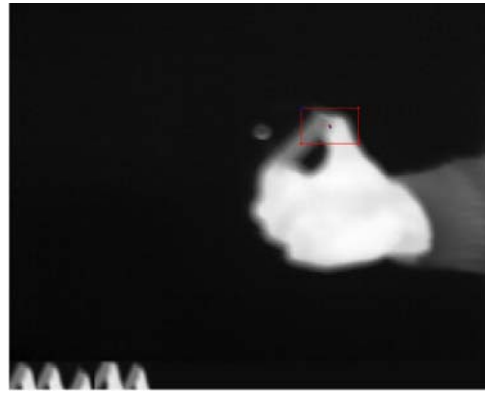
圖 4.7 物體在處於較高速下，若無中心運動速度的預測，較易追丟。

同樣為了克服上述的問題，在此我們把物體中心運動的預測也考量進來，在圖 4.7(b)中，當在追蹤到第 8 張時，所追蹤到物體就有些微的偏離，而後續的追蹤結果偏離愈來愈大。但我們從 4.8 圖(b)(c)(d)觀察，當加入物體中心運動預測的方法，依舊可以追蹤得很穩健。在此註明，如果對於追蹤高速移動中的物體，僅管可以放大取樣時標準差的設定，而解決追蹤失效的問題，但若加上運動預測，則我們可以在不變動標準差設定的情況下，即可追蹤較快速移動的物體。

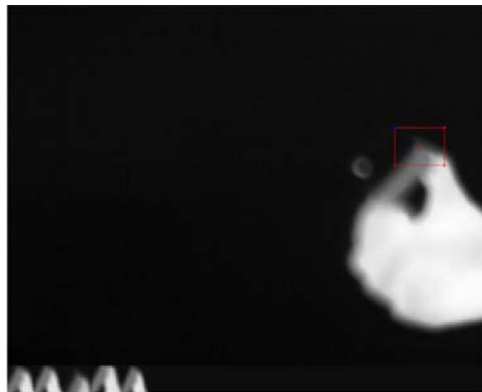




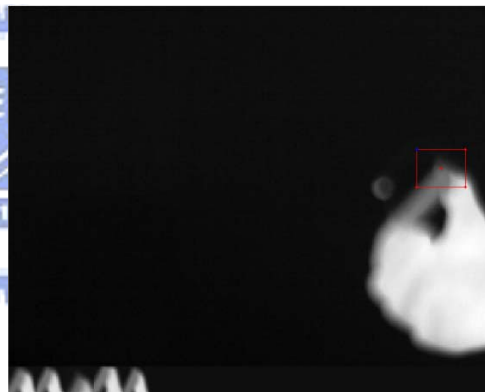
(a) 第 1 張



(b) 第 10 張



(c) 第 15 張



(d) 第 17 張

圖 4.8 若物體加上中心運動速度預測後，目標物體可被正確的追蹤。

第五章 結論

在本論文中，我們介紹了 Ho 等人的方法，並把它實作出來，讓我們瞭解到如何利用影像子空間模型去做人臉追蹤。其中最主要的核心架構在於 k -means 演算法，產生主要的核心影像與子空間，而後對所要預估的影像樣本做子空間的距離計算，找出最小距離的影像樣本，達到追蹤的目的。

而後，我們提出了三個方法，來改進目標物追蹤的效果，分別為物體中心運動的預測、背景前景臨界值比例的判定與精確的物體切割，雖然這些方法都相當簡單，但是從實驗結果發現，加上這些方法後，可使得 k -means 演算法所產生的核心影像與子空間，變得更為完整，且當碰到適度的障礙物時，我們也能克服障礙物，持續做人臉追蹤。

由目前實驗的結果發現，我們仍無法對複雜的障礙物與多人做追蹤，在後續的研究中，這是一個具有挑戰性的研究方向，此外在計算樣本到子空間距離時，是否可應用一些統計量的估測來找出較佳的追蹤目標影像亦是一個可能的研究方向。

參考文獻

- [1] M. Isard and A. Blake, "Condensation-conditional density propagation for visual tracking," *Proc. IEEE International Journal of Computer Vision*, vol. 20, no. 1, pp. 5-28, 1998.
- [2] A. J. Davison, Y. Gonzalez-Cid, and N. Kita, "Real-time 3D SLAM with wide-angle vision," *Intelligent Autonomous Vehicles Symposium*, July 2004.
- [3] D. Margaritis and S. Thrun, "Learning to locate an object in 3D space from a sequence of images," *Proc. International Conference on Machine Learning*, pp. 332-340, 1998.
- [4] L. Taycher, J. W. Fisher, and T. Darrell, "Combining object and feature dynamics in probabilistic tracking," *Proc. International Conference on Computer Vision and Pattern Recognition*, vol. 2, pp 106-113, 2005.
- [5] D. Fox, W. Burgard, F. Dellaert, and S. Thrun, "Monte Carlo localization: efficient position estimation for mobile robots," *Proc. National Conference on Artificial Intelligence*, pp. 343-349, July 1999.
- [6] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for on-line non-linear/non-gaussian bayesian tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174-188, 2002.
- [7] A.M. Boumaza and J. Louchet, "Mobile robot sensor fusion using flies," *Proc. European Conference on Genetic Programming*, pp. 357-367, 2003.
- [8] J. Louchet, "Using an individual evolution strategy for stereovision," *Genetic Programming and Evolvable Machines*, vol. 2, pp. 101-109, 2001
- [9] T. Cham and J. Rehg, "A multiple hypothesis approach to figure tracking," *Proc.*

International Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 239–219, 1999.

[10] B. Han, D. Comaniciu, Y. Zhu, and L. Davis, “Incremental density approximation and kernel-based bayesian filtering for object tracking,” *Proc. International Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 638–644, 2004.

[11] H. Murase and S. K. Nayar, “Visual learning and recognition of 3-D objects from appearance,” *International Journal on Computer Vision*, vol. 14, pp. 5–24, 1995.

[12] Y. Huang and I. Essa, “Tracking multiple objects through occlusions,” *Proc. International Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 1051-1058, 2005

[13] H. Mei, A. Sethi, and Y. Gong, “A detection-based multiple object tracking method,” *Proc. International Conference on Image Processing*, vol. 5, pp. 3065-3068, 2004.



[14] J. W. Davis and M. A. Keck, “A two-stage template approach to person detection in thermal Imagery,” *Proc. IEEE International Workshop on Applications Computer Vision*, vol.1, pp. 364-369, January 2005.

[15] O. M. Mozos, C. Stachniss, and W. Burgard, “Supervised learning of places from range data using AdaBoost,” *Proc. International Conference on Robotics and Automation*, pp. 1742-1747, 2005.

[16] P. Barrera, J. M. Canas, and V. Matellan, “Visual object tracking in 3D with color- based. particle filter,” *Journal of Advanced Technology on Imaging and Graphics*. Volume 1, Number 1, pp 13-15, 2006.

- [17] J. Ho, K. C. Lee, M. H. Yang, and D. Kriegman, “Visual tracking using learned subspaces,” *Proc. International Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 782-789, June, 2004.

