

國立交通大學
電機與控制工程研究所

碩士論文

加入線上模板資料庫建構之
三維物體追蹤平均位移演算法

3D Object Mean-shift Tracking Algorithm with
Online Template Database Construction

研究生： 陳 建 廷

指導教授： 胡 竹 生 博士

中華民國一〇一年九月

加入線上模板資料庫建構之
三維物體追蹤平均位移演算法

3D Object Mean-shift Tracking Algorithm with
Online Template Database Construction

研究生：陳建廷

Student : Chien-Ting Chen

指導教授：胡竹生 博士

Advisor : Prof. Jwu-Sheng Hu



Submitted to Institute of Electrical Control Engineering
College of Electrical and Computer Engineering
National Chiao-Tung University
in Partial Fulfillment of the Requirements
for the Degree of Master
In

Electrical and Control Engineering

September 2012

Hsinchu, Taiwan, Republic of China

中華民國一〇一年九月

加入線上模板資料庫建構之 三維物體追蹤平均位移演算法

研究生：陳 建 廷

指導教授：胡 竹 生 博士

國立交通大學

電機與控制工程研究所碩士班



本論文提出了一套建構在三維物體追蹤演算法下，可用於複雜背景與移動場景的目標模型更新的方法。影像追蹤演算法採用三維平均位移演算法來計算目標物於空間中的位移，並利用主成份分析法來估測物體於三維空間中的大小與旋轉，由於加入了三維空間資訊以及新的目標模型更新方式，因此在複雜背景下以及目標物外觀與顏色產生較大變化時，會比使用其他的目標模型更新方法具有更穩健的追蹤效果。另外提出加入線上模板資料庫建構的方法，資料庫建構的準則以及資料庫內的資料選取將是本論文的另一個重點，這會使得追蹤目標物的過程中，對於目標模型的外觀及顏色變化有更迅速及適應性更高的效果，當目標物因旋轉而導致外觀顏色變化較為劇烈時，比起一般其他直接對目標顏色機率模型做更新的方法，準確性與穩定性來得更高。

3D Object Mean-shift Tracking Algorithm with Online Template Database Construction

Student : Chien-Ting Chen

Advisor : Prof. Jwu-Sheng Hu

Institute of Electrical and Control Engineering
National Chiao-Tung University



In this thesis, a target model updating method based on 3-D mean-shift tracking algorithm is proposed. The algorithm is aimed at reducing the influence of complex and dynamic background. It uses principal component analysis to estimate the size and rotation of the target in 3-Dimensions. With the 3-D spatial information and new target model updating method, this algorithm outperforms other target model updating methods such as traditional Mean-Shift and 3-D Mean-Shift. The method consists of a set of rules to construct online template database. These results in a better and faster tracking process. In particular, when target rotates drastically, it gives a more accurate and stable result compared to other methods that only update the target model.

誌 謝

兩年的碩士生活轉眼間就過去了，感謝指導教授—胡竹生老師的教導，使我不僅學習到電機方面的專業知識，也對於做研究的態度有更上一層樓的進步，老師不斷地強調「創新」與「貢獻」的理念，著實是讓我受益良多。

也要謝謝 905 實驗室的學長姐、同學與學弟，兩年的研究所生涯有了他們才有許多美好的回憶。在專業研究方面，感謝阿吉學長的熱心指導，才能使得實驗研究順利，論文內容完整。另外是碩一帶我的永融學長，也是學長耐心的教導，才奠定我實驗時的品質堅持與研究態度要求。感謝大師兄與 Judo，在他們身上看到了對於研究的態度與堅持，以及德洋學長與震華學長的專業素養。興哥與鏗元學姐對於其專業領域的知析程度也著實令我佩服不已。Alphar 學長的蛋糕香引人入迷，嘟嘟學長則為實驗室帶來歡樂，唐哥不僅提點了我許多關於研究與口試的重點，也是打球的伙伴，更是閒聊的好友，昌言學長在我剛進碩班時幫我解決了不少程式上的麻煩。感謝上一屆的昀軒、湘筑與學文，幽默風趣，讓實驗室充滿歡樂，育成與新文也讓我看到其認真與充滿幹勁的一面，macaca 則提點了我不少研究的想法，建安在我碩一時也幫我解決不少關於影像方面的疑惑。感謝同屆的 Daniel，同為影像組的幫我解決許多程式上的問題，同為實驗室管理員的鳴哥，讓我看到了責任感與貼心的一面，也互相幫彼此打氣，常常一起打球的翰哥，在最後一個月不僅互相加油，也常是實驗室一起關門的好伙伴。逕博的耕維，為人熱心且幽默，在他身上看見了對學術研究的熱情與堅持，一樣逕博的男哥，也是打球的好伙伴，更是閒聊開玩笑的好朋友。學弟中的鳴遠同為影像組，但已然實力深厚，哲宇與罐頭坐我座位附近，都是聊天的好對象，阿文的重訓與羽球教學，讓我看見他對事情的堅持，小山東與大夢的風趣，幫我打了口試時的強心劑。

感謝兩位口試委員—宋開泰老師與周志成老師，有他們的提點，本論文的内容才能更加完善。感謝在新竹工作的欣瑤，別的實驗室的國展、逸棋與曹門、阿康和阿吉，共同分享在新竹生活的點點滴滴，玉琪的關心與陪伴，讓我在研究之餘發現生活中處處有樂趣也滿滿是回憶。最後要感謝我的家人—父母及妹妹，有了父母的努力，讓我衣食無缺而無後顧之憂，有他們的鼓勵、關心與陪伴，讓我一心向上無所懼。在此僅以此篇幅向所有幫助過我的貴人們獻上最誠摯的謝意。

目 錄

摘 要	i
ABSTRACT	ii
誌 謝	iii
目 錄	iv
表 列	vi
圖 列	vii
第一章 緒論	1
1.1 研究動機	1
1.2 文獻回顧	2
1.3 本論文的貢獻與創新	3
1.4 論文架構	4
第二章 影像演算法分析	5
2.1 傳統平均位移演算法	5
2.1.1 前言	5
2.1.2 區域模型表示	5
2.1.3 基於 Bhattacharyya Coefficient 的相似度量測	7
2.1.4 演算法方程式推導	8
2.1.5 平均位移演算法流程圖	11
2.2 三維平均位移演算法	12
2.2.1 三維平均位移演算法模型定義	12
2.2.2 演算法方程式推導	14
2.2.3 目標的深度範圍判定	16
2.2.4 主成份分析法 (Principal Component Analysis)	18
2.2.5 色彩空間表示法	19
2.2.6 三維平均位移演算法流程圖	21
第三章 目標模型的更新	22
3.1 整體目標模型更新方式	22
3.2 選擇性子模型更新方式	23
3.3 結合顏色相似度與中心移動距離的目標模型更新方式	25
3.3.1 新的目標模型更新方法推導	25
3.3.2 加入 CCDMU 的三維平均位移演算法流程圖	31

3.4 加入線上模板資料庫建構的三維物體追蹤演算法.....	32
3.4.1 線上模板資料庫建構的準則.....	32
3.4.2 加入 OTDMU 的三維平均位移演算法流程圖.....	36
第四章 系統設計與實驗測試比較.....	37
4.1 實驗環境.....	37
4.2 實驗測試.....	38
4.2.1 評估追蹤效果係數和實驗參數.....	38
4.2.2 基本物體追蹤.....	38
4.2.3 人體追蹤.....	59
4.3 實驗結果與分析.....	75
第五章 結論.....	76
5.1 結論與未來展望.....	76
參考文獻.....	77



表 列

表 2.1-1: [11]兩種 kernel functions	6
表 4.1-1: [11]Kinect 規格表.....	37
表 4.1-2: [11]演算法規格表	37
表 4.2-1: 基本物體追蹤一數值結果(2D)	43
表 4.2-2: 基本物體追蹤一數值結果(3D)	44
表 4.2-3: 基本物體追蹤二數值結果(2D)	48
表 4.2-4: 基本物體追蹤二數值結果(3D)	49
表 4.2-5: 基本物體追蹤三數值結果(2D)	53
表 4.2-6: 基本物體追蹤三數值結果(3D)	54
表 4.2-7: 基本物體追蹤四數值結果(2D).....	59
表 4.2-8: 基本物體追蹤四數值結果(3D).....	59
表 4.2-9: 人體追蹤一數值結果(2D).....	64
表 4.2-10: 人體追蹤一數值結果(3D).....	65
表 4.2-11: 人體追蹤二數值結果(2D).....	69
表 4.2-12: 人體追蹤二數值結果(3D).....	70
表 4.2-13: 人體追蹤三數值結果(2D).....	74
表 4.2-14: 人體追蹤三數值結果(3D).....	75

圖 列

圖 2.1-1: [11]兩個不同機率分佈的相似度	8
圖 2.1-2: 平均位移演算法流程圖	11
圖 2.2-1: [11]uv 座標與 kienct 三維座標	12
圖 2.2-2: [11]深度量化統計圖 / 深度範圍示意圖	17
圖 2.2-3: 主成份分析示意圖	19
圖 2.2-4: 三維平均位移演算法流程圖	21
圖 3.3-1: α 與 $e^{-\alpha(1-\rho)}$ 關係圖	29
圖 3.3-2: 加入 CCDMU 的三維平均位移演算法流程圖	31
圖 3.4-1: 加入 OTDMU 的三維平均位移演算法流程圖	36
圖 4.2-1: 基本物體追蹤一影像輸出結果(2D)	39
圖 4.2-2: 基本物體追蹤一中心距離誤差(2D)	40
圖 4.2-3: 基本物體追蹤一影像輸出結果(3D)	41
圖 4.2-4: 基本物體追蹤一中心距離誤差(3D)	42
圖 4.2-5: 基本物體追蹤一目標顏色相似度	42
圖 4.2-6: 基本物體追蹤一目標空間分佈相似度	43
圖 4.2-7: 基本物體追蹤二影像輸出結果(2D)	44
圖 4.2-8: 基本物體追蹤二中心距離誤差(2D)	45
圖 4.2-9: 基本物體追蹤二影像輸出結果(3D)	46
圖 4.2-10: 基本物體追蹤二中心距離誤差(3D)	47
圖 4.2-11: 基本物體追蹤二目標顏色相似度	47
圖 4.2-12: 基本物體追蹤二目標空間分佈相似度	48
圖 4.2-13: 基本物體追蹤三影像輸出結果(2D)	49
圖 4.2-14: 基本物體追蹤三中心距離誤差(2D)	50
圖 4.2-15: 基本物體追蹤三影像輸出結果(3D)	51
圖 4.2-16: 基本物體追蹤三中心距離誤差(3D)	52
圖 4.2-17: 基本物體追蹤三目標顏色相似度	52

圖 4.2-18: 基本物體追蹤三目標空間分佈相似度.....	53
圖 4.2-19: 基本物體追蹤四影像輸出結果(2D).....	54
圖 4.2-20: 基本物體追蹤四中心距離誤差(2D).....	55
圖 4.2-21: OTDMU-資料庫內編號 10~21 的資料.....	56
圖 4.2-22: 基本物體追蹤四影像輸出結果(3D).....	57
圖 4.2-23: 基本物體追蹤四中心距離誤差(3D).....	57
圖 4.2-24: 基本物體追蹤四目標顏色相似度.....	58
圖 4.2-25: 基本物體追蹤四目標空間分佈相似度.....	58
圖 4.2-26: 人體追蹤一影像輸出結果(2D).....	60
圖 4.2-27: 人體追蹤一中心距離誤差(2D).....	61
圖 4.2-28: 人體追蹤一影像輸出結果(3D).....	62
圖 4.2-29: 人體追蹤一中心距離誤差(3D).....	63
圖 4.2-30: 人體追蹤一目標顏色相似度.....	63
圖 4.2-31: 人體追蹤一目標空間分佈相似度.....	64
圖 4.2-32: 人體追蹤二影像輸出結果(2D).....	65
圖 4.2-33: 人體追蹤二中心距離誤差(2D).....	66
圖 4.2-34: 人體追蹤二影像輸出結果(3D).....	67
圖 4.2-35: 人體追蹤二中心距離誤差(3D).....	68
圖 4.2-36: 人體追蹤二目標顏色相似度.....	68
圖 4.2-37: 人體追蹤二目標空間分佈相似度.....	69
圖 4.2-38: 人體追蹤三影像輸出結果(2D).....	70
圖 4.2-39: 人體追蹤三中心距離誤差(2D).....	71
圖 4.2-40: 人體追蹤三影像輸出結果(3D).....	72
圖 4.2-41: 人體追蹤三中心距離誤差(3D).....	73
圖 4.2-42: 人體追蹤三目標顏色相似度.....	73
圖 4.2-43: 人體追蹤三目標空間分佈相似度.....	74

第一章 緒論

1.1 研究動機

影像追蹤在影像處理這領域中可說是一個很重要的課題，最終目的就是希望能夠找到一個準確性高、處理速度快且穩定性佳的追蹤方式，而這也是在這領域中眾人所一直想要追求的目標。在追蹤的效能上，會影響的因素很多，一般以影像處理來說，例如背景顏色與目標物顏色過於相似而產生混淆、目標物被障礙物擋住而影響其顏色分佈、因為外在光源的改變或是目標物本身的旋轉，使得目標物的顏色改變等等，上述情形皆會影響到追蹤的效能。於是人們開始使用一些新的輔助儀器來幫助影像的追蹤，如雷射測距儀或是超音波測距儀等等，但是仍然有其各自的缺點，雷射測距儀雖然精準度高，但價格昂貴，不符經濟效益，超音波測距儀雖然價格較為低廉，但其精準度相對來說卻是比較差的。然而在 2010 年時，微軟(Microsoft)公司發表了新一代的遊戲控制器，利用以色列的 PrimeSense 公司所開發的新的技術[1]，Light coding 技術，來做出物體在空間的分佈偵測，其技術理論是利用連續光(近紅外線)對測量空間進行編碼，經感應器讀取編碼的光線，交由晶片運算進行解碼後，產生成一張具有深度的圖像。此技術與舊有的距離偵測器最大的不同點是，它能同時掃描一整個面，而且它獲得資料的頻率極高，可以利用快速的方法得到空間資訊。

本實驗室已藉由新的空間資訊，提出了同時可以應用色彩與空間資訊的三維物體追蹤平均位移演算法，在此基礎上，本論文就是希望藉由考慮目標物的外觀色彩與空間資訊的變化，來將我們所使用的目標顏色機率模型做適當的更新，另外藉由提出線上模板資料庫建構的準則與流程，將變化較為明顯的模型資料儲存進資料庫中，以利之後追蹤的使用，做為進一步加強三維物體追蹤演算法的效能及穩定性。

1.2 文獻回顧

本論文所用到的物體追蹤演算法，其基本理論是由論文[2]所出發，他所提出的 Kernel-based object tracking 是以平均位移演算法(mean-shift)為基礎的追蹤方法，有著運算量低與穩健度高的優點，之後有大量的人由這個方向做更進一步的研究，他們接著加入其他的方法來增進與補強追蹤的穩健度。

論文[2]所提出的演算法雖然有利用空間資訊當作權重，來建立顏色模型，但仍有些地方可以加強，論文[3][4]提出了更進一步的空間資訊使用，利用了不同顏色在空間中的不同分佈情形，來建立新的顏色模型與新的相似方程式，如此對目標模型的描述，不只有顏色，還有空間分佈資訊，可使得追蹤更穩定。

由於在追蹤的時候，目標物不只是單純平移，還會有旋轉跟變形的情況，在此情形上，論文[5][6]提出解決方法，而論文[7]使用了論文[8]中所提供的方法，Image Moments，對於每一次出現的目標模型，都去做目標的大小跟旋轉的預測，使得下次抓取目標資料時，可以得到合適的追蹤範圍，在 Open Source Computer Vision (OpenCV)[9]中的追蹤方法就是依照此論文實作出來。而論文[10]是利用主成分分析法(PCA)，找出目標物在 uv 平面中的主成分軸向與大小，利用這兩項資料去估測出目標物在畫面中的大小與角度，以上方法在追蹤上都有不錯的效果。論文[11]則提出了利用 Kinect sensor 的資訊，結合景深資訊與影像顏色特徵的三維平均位移演算法，其能藉由深度資訊有效分離出目標前景與背景，在追蹤的效果上可說是幫助極大，也有相當不錯的效果。

在色彩空間方面，RGB 是一般會使用的色彩空間，但是遇到環境光源發生變化時，RGB 會不足以對付此情況，論文[7]提出使用 HSV 色彩空間，認為 V 值對光源變化較為敏感，所以做進一步的處理。論文[10]使用 normalized RGB 讓亮度對目標追蹤的影響減至最低。

一般的平均位移演算法都是用單一的目標色彩模型來當追蹤時的依據，也就是說，基於目標模型在追蹤過程中不發生顏色改變的這個假設，會一直持續採用

初始的目標模型，所以當物體因為旋轉而使得顏色改變時，就會造成演算法穩定度有問題產生。論文[12]提出利用目標物不同角度的圖片建立起多個顏色模型，再利用這些模型作非單一色彩模型的追蹤方法，並且適時的選擇最相近的模型當基準，但缺點是事前必須知道每一角度的顏色模型。論文[13]則是在追蹤過程中，採用對目標模型與候選模型進行簡單線性加權的方式來產生出新的目標模型。論文[14]針對追蹤過程中的特徵進行研究，藉由選擇對當前環境可分辨性較高的特徵模型進行目標的追蹤及模型的更新。論文[15][16][17]則將 Kalman filter 用於追蹤過程中的目標模型更新。上述的目標模型更新方式都是將目標模型視為一個整體，同時對模型中的每個分量進行更新，然而在實際的目標移動當中，隨著環境和目標外觀的變化，目標模型內有些分量的變化是緩慢的，而有些則是迅速的，因此整體模型更新的方式顯得過於粗糙。論文[18]則是將特徵模型中的每個分量視為單一子模型，基於每個子模型的匹配貢獻度，分別選擇當前需要更新的子模型及其更新權重。

1.3 本論文的貢獻與創新

本論文為利用 Kinect sensor 的資訊，藉由以傳統的平均位移演算法推廣到三維的平均位移演算法為基礎，針對原本三維平均位移演算法無法有效處理目標物旋轉或受到遮蔽而導致外觀顏色改變的情形，我們推導出一個有效的目標模型的顏色特徵更新方式來解決此種情況，並結合線上模板資料庫建構的方法，同步追蹤並有效建構出不同面向的目標模型，利於追蹤使用，有助於增進準確度與穩定性，實驗發現效能與結果都有不錯的呈現。

本論文的創新：

1. 提出結合顏色機率相似度與目標物中心移動距離的目標模型更新方法。
2. 提出加入線上模板資料庫建構的三維物體追蹤演算法。

1.4 論文架構

本論文由平均位移演算法開始介紹，接著介紹應用 Kinect 所提供的空間資訊拓展到三維的平均位移演算法以及加入目標模型更新部分對整體追蹤效能的提升：

第二章、平均位移演算法：介紹平均位移演算法，從如何建立目標模型、候選模型，接著利用 Bhattacharyya coefficient 來計算相似方程式，最後推導出平均位移向量。

三維平均位移演算法：介紹如何將二維的平均演算法推導至三維空間，以及如何建立三維的目標與候選模型，三維空間的相似方程式，最後得到三維的平均位移向量。

接著介紹初始化的方法，與如何應用主成份分析法來建立目標模型，最後推導出我們所要的色彩空間表示式。

第三章、目標模型更新：首先介紹兩種整體目標模型更新方式和選擇性子模型更新方式。接著介紹如何結合顏色相似度與目標物中心移動距離來做目標模型更新的方法，以及加入線上模板資料庫建構，來建立可信度夠高的目標模型資料庫，進一步加強三維物體追蹤演算法的穩定性與準確性。

第四章、系統設計與實驗測試比較：介紹本論文的實驗環境，並將上述演算法實現出來，同時測試在不同環境底下的各個演算法的穩定性與準確性。

第五章、結論：心得總結以及對本演算法的未來展望。

第二章 影像演算法分析

2.1 傳統平均位移演算法

2.1.1 前言

平均位移演算法是一個以模板(template)為基礎的影像追蹤方式。主要的概念是根據目標模型(target model)，利用疊代法來找出最相近的候選模型(candidate model)，藉由比對目標模型與候選模型的各種顏色機率分佈值，再利用統計上 Bhattacharyya coefficient 的方法來計算出平均位移的向量，之後重複以上步驟直到收斂到最相似的候選區域。

2.1.2 區域模型表示

由於平均位移演算法是一個以樣本為基礎的影像追蹤方式，所以必須找出一個特徵模型來代表整個目標區域，一般來說，都是選擇顏色的機率分佈來代表目標物的模型。首先，定義目標區域的中心位置為 0，候選區域的中心位置為 \mathbf{y} 。接著我們定義目標模型表示式為 \mathbf{q} ，候選模型表示式為 $\mathbf{p}(\mathbf{y})$ ，為了降低運算量，在實作上我們會把顏色資訊量化為 m 種 bin 值，上述定義的目標模型如下：

$$\mathbf{q} = \{q_u\}_{u=1, \dots, m} \quad \sum_{u=1}^m q_u = 1 \quad (2-1)$$

候選區域模型為：

$$\mathbf{p}(\mathbf{y}) = \{p_u(\mathbf{y})\}_{u=1, \dots, m} \quad \sum_{u=1}^m p_u = 1 \quad (2-2)$$

平均位移演算法的第一步是需要從目標中找到一個由顏色特徵所組成的機率分佈函數來代表目標物的顏色特徵。首先假設目標區域內的像素數目為 n ， $\{\mathbf{x}_i^*\}_{i=1, \dots, n}$ 代表該像素所在的位置且目標區域的中央位置定義為 0。再另外定義一

個方程式 $b: R^2 \rightarrow \{1, \dots, m\}$, $b(\mathbf{x}_i^*)$ 代表該位置像素的顏色量化至 $1, \dots, m$ 個 bin, 總共有 m 種顏色 bin, 目標模型的 $u = 1, \dots, m$ 的顏色機率表示式如下所示:

$$q_u = C \sum_{i=1}^n k(\|\mathbf{x}_i^*\|^2) \delta[b(\mathbf{x}_i^*) - u] \quad (2-3)$$

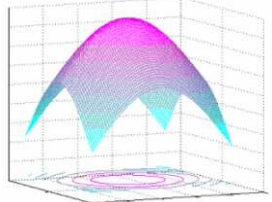
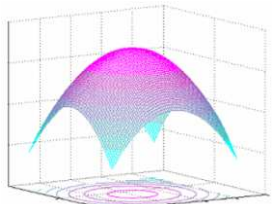
δ 表示 Kronecher delta 方程式, 用來決定此位置的像素的顏色是否屬於此 bin

值, 參數 C 的功能則是把此機率值正規化, 滿足 $\sum_{u=1}^m q_u = 1$ 。

$$C = \frac{1}{\sum_{i=1}^n k(\|\mathbf{x}_i^*\|^2)} \quad (2-4)$$

在式(2-3)與(2-4)中, $k(\|\mathbf{x}_i^*\|^2)$ 為一個 convex and monotonic decreasing kernel 方程式, 如表 2.1-1, 最高的數值在最中央, 離中央點越遠的則其值會越低, 一般而言, 我們認為越接近的中央的像素會越為重要, 也就是可信賴度較高, 而越遠離中央的像素則認為相對的不重要, 也就是可信賴度較低。在某些情況下, 邊緣的像素可能會包含到遮蔽物的像素, 若是給定中央像素權重較高的話, 則可以避免誤判遮蔽物的像素為目標物的像素的情形發生。

表 2.1-1: [11]兩種 kernel functions

函數名稱	定義	Sketch with $d = 2$
Normal Function (Gaussian Function)	$K_N(\mathbf{x}) = \begin{cases} \frac{1}{(2\pi)^{d/2}} \exp(-\frac{1}{2}\ \mathbf{x}\ ^2), & \text{if } \ \mathbf{x}\ < 1 \\ 0, & \text{otherwise.} \end{cases}$	
Epanechnikov Function	$K_E(\mathbf{x}) = \begin{cases} \frac{1}{2C_d} (d+2)(1-\ \mathbf{x}\ ^2), & \text{if } \ \mathbf{x}\ < 1 \\ 0, & \text{otherwise.} \end{cases}$	
d : 空間維度 (2D 空間的狀況, $d = 2$) C_d : 維度為 d 的球體體積 (2D 空間的狀況, $C_d = \pi$)		

接著定義平均位移演算法中候選區域的機率分佈函數。 $\{y_i\}_{i=1,\dots,n_h}$ 代表在候選模型中每個像素的位置，而候選區域的中央位置定義為 y ， $b(y_i)$ 代表 y_i 位置的像素的顏色量化至 $1, \dots, m$ 個 bin，共有 m 種 bin。候選模型的 $u = 1, \dots, m$ 的顏色機率表示式如下所示：

$$p_u(y) = C_h \sum_{i=1}^{n_h} k\left(\left\|\frac{y_i - y}{h}\right\|^2\right) \delta[b(y_i) - u] \quad (2-5)$$

$k\left(\left\|\frac{y_i - y}{h}\right\|^2\right)$ 是與目標模型相同的 kernel 函數， h 是頻寬，由候選區域的大小

來決定， C_h 為滿足 $\sum_{u=1}^m p_u = 1$ 的正規化常數：

$$C_h = \frac{1}{\sum_{i=1}^{n_h} k\left(\left\|\frac{y_i - y}{h}\right\|^2\right)} \quad (2-6)$$



2.1.3 基於 Bhattacharyya Coefficient 的相似度量測

相似度量測函數(similarity measure function)，是用來計算目標區域與候選區域的相似程度。現今有許多相似函數被用來代表不同目標的相似度。一個可微分的 kernel 函數皆可推導出可微分的相似度函數，以及基於梯度概念的有效的最佳化程序，而這個程序可以找到最可能的相似區域，也就是我們所想要得到的跟目標物最相近的區域。

在傳統的平均位移演算法中，Bhattacharyya coefficient 常被用來當作相似度量測的函數。首先相似函數定義為目標區域與候選區域的距離，如下所示：

$$d(\mathbf{y}) = \sqrt{1 - \rho[\mathbf{p}(\mathbf{y}), \mathbf{q}]} \quad (2-7)$$

ρ 為目標區域 $\hat{\mathbf{q}}$ 與候選區域 $\hat{\mathbf{p}}$ 的 Bhattacharyya coefficient。

$$\rho(\mathbf{y}) \equiv \rho[\mathbf{p}(\mathbf{y}), \mathbf{q}] = \sum_{u=1}^m \sqrt{p_u(\mathbf{y})q_u} \quad (2-8)$$

Bhattacharyya coefficient 是根據在 m 維的單位向量 $(\sqrt{\hat{p}_1}, \dots, \sqrt{\hat{p}_m})^T$ 與 $(\sqrt{\hat{q}_1}, \dots, \sqrt{\hat{q}_m})^T$ 的餘弦函數，此概念是一個有效率的統計學上的量測法，由圖 2.1-1 可以看出兩個不同機率分佈的相似程度，其值就是兩個機率分佈相交的淺色區域，值越大表示兩者機率分佈越相近，也就是圖形越相似。

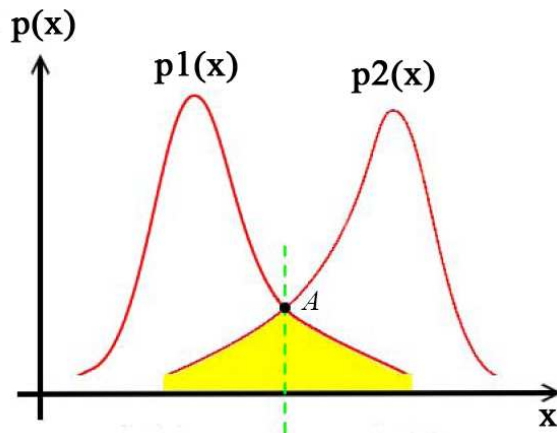


圖 2.1-1: [11]兩個不同機率分佈的相似度

2.1.4 演算法方程式推導

要找出式(2-7)的最小值，相當於要找到式(2-8)中的 Bhattacharyya coefficient 的最大值。因此我們使用泰勒展開式，將 $\rho[\mathbf{p}(\mathbf{y}), \mathbf{q}]$ 對 \mathbf{y} 展開於 $\mathbf{y} = \mathbf{y}_{old}$ 。在疊代前，位置 \mathbf{y}_{old} 代表上一張 frame 的目標位置 \mathbf{x} ，疊代後， \mathbf{y}_{old} 會隨著疊代而更新，其線性估測推導如下：

$$\rho[\mathbf{p}(\mathbf{y}), \mathbf{q}] = \sum_{u=1}^m \sqrt{p_u(\mathbf{y})q_u} = \sum_{n=0}^{\infty} \frac{\partial^n \rho[\mathbf{p}(\mathbf{y}), \mathbf{q}]}{\partial \mathbf{p}(\mathbf{y})^n} \Big|_{\mathbf{y}=\mathbf{y}_{old}} [\mathbf{p}(\mathbf{y}) - \mathbf{p}(\mathbf{y}_{old})]^n$$

$$\begin{aligned}
&= \rho[\mathbf{p}(\mathbf{y}), \mathbf{q}] \Big|_{\mathbf{y}=\mathbf{y}_{old}} + \frac{\partial \rho[\mathbf{p}(\mathbf{y}), \mathbf{q}]}{\partial \mathbf{p}(\mathbf{y})} \Big|_{\mathbf{y}=\mathbf{y}_{old}} [\mathbf{p}(\mathbf{y}) - \mathbf{p}(\mathbf{y}_{old})] \\
&+ \frac{\partial^2 \rho[\mathbf{p}(\mathbf{y}), \mathbf{q}]}{\partial \mathbf{p}(\mathbf{y})^2} \Big|_{\mathbf{y}=\mathbf{y}_{old}} [\mathbf{p}(\mathbf{y}) - \mathbf{p}(\mathbf{y}_{old})]^2 + \dots \\
&\approx \rho[\mathbf{p}(\mathbf{y}), \mathbf{q}] \Big|_{\mathbf{y}=\mathbf{y}_{old}} + \frac{\partial \rho[\mathbf{p}(\mathbf{y}), \mathbf{q}]}{\partial \mathbf{p}(\mathbf{y})} \Big|_{\mathbf{y}=\mathbf{y}_{old}} [\mathbf{p}(\mathbf{y}) - \mathbf{p}(\mathbf{y}_{old})] \\
&= \sum_{u=1}^m \sqrt{p_u(\mathbf{y}) q_u} \Big|_{\mathbf{y}=\mathbf{y}_{old}} + \sum_{u=1}^m \frac{q_u}{2\sqrt{p_u(\mathbf{y}) q_u}} \Big|_{\mathbf{y}=\mathbf{y}_{old}} [p_u(\mathbf{y}) - p_u(\mathbf{y}_{old})] \\
&= \frac{1}{2} \sum_{u=1}^m \sqrt{p_u(\mathbf{y}_{old}) q_u} + \frac{1}{2} \sum_{u=1}^m p_u(\mathbf{y}) \sqrt{\frac{q_u}{p_u(\mathbf{y}_{old})}} \tag{2-9}
\end{aligned}$$

這個近似法是假設在目標物沒有大幅度地從位置 \mathbf{y}_{old} 移動至位置 \mathbf{y} 下，而且圖像是在連續 frame 的情況下，此近似法才可正確地追到物體。

接著我們把式(2-5)代入到式(2-9)，可得：

$$\begin{aligned}
\rho(\mathbf{y}) \equiv \rho[\mathbf{p}(\mathbf{y}), \mathbf{q}] &\approx \frac{1}{2} \sum_{u=1}^m \sqrt{p_u(\mathbf{y}_{old}) q_u} + \frac{1}{2} \sum_{u=1}^m p_u(\mathbf{y}) \sqrt{\frac{q_u}{p_u(\mathbf{y}_{old})}} \\
&= \frac{1}{2} \sum_{u=1}^m \sqrt{p_u(\mathbf{y}_{old}) q_u} + \frac{1}{2} \sum_{u=1}^m C_h \sum_{i=1}^{n_h} k \left(\left\| \frac{\mathbf{y}_i - \mathbf{y}}{h} \right\|^2 \right) \delta[b(\mathbf{y}_i) - u] \sqrt{\frac{q_u}{p_u(\mathbf{y}_{old})}} \\
&= \frac{1}{2} \sum_{u=1}^m \sqrt{p_u(\mathbf{y}_{old}) q_u} + \frac{1}{2} C_h \sum_{i=1}^{n_h} w_i k \left(\left\| \frac{\mathbf{y}_i - \mathbf{y}}{h} \right\|^2 \right) \tag{2-10}
\end{aligned}$$

其中

$$w_i = \sum_{u=1}^m \sqrt{\frac{q_u}{p_u(\mathbf{y}_{old})}} \delta[b(\mathbf{y}_i) - u] \tag{2-11}$$

目標是找到最大的 Bhattacharyya coefficient $\rho(\mathbf{y})$ 。因為 \mathbf{y}_{old} 與 $\rho(\mathbf{y})$ 相互獨立，所以在式(2-10)的前項， $\frac{1}{2} \sum_{u=1}^m \sqrt{p_u(\mathbf{y}_{old}) q_u}$ ，並不會影響 $\rho(\mathbf{y})$ 的值，會影響的

只有後項部分，因此令後項部分為 $f(\mathbf{y})$ ：

$$f(\mathbf{y}) = \frac{1}{2} C_h \sum_{i=1}^{n_h} w_i k \left(\left\| \frac{\mathbf{y}_i - \mathbf{y}}{h} \right\|^2 \right) \quad (2-12)$$

將式(2-12)對 \mathbf{y} 取梯度微分以找出峰值：

$$\nabla f(\mathbf{y}) = \frac{C_h}{2h^2} \sum_{i=1}^{n_h} (\mathbf{y}_i - \mathbf{y}) w_i k' \left(\left\| \frac{\mathbf{y}_i - \mathbf{y}}{h} \right\|^2 \right) \quad (2-13)$$

令 $g(x) = -k'(x)$ ，可得：

$$\begin{aligned} \nabla f(\mathbf{y}) &= \frac{C_h}{2h^2} \sum_{i=1}^{n_h} (\mathbf{y} - \mathbf{y}_i) w_i g \left(\left\| \frac{\mathbf{y}_i - \mathbf{y}}{h} \right\|^2 \right) \\ &= \frac{C_h}{2h^2} \left[\sum_{i=1}^{n_h} \mathbf{y} w_i g \left(\left\| \frac{\mathbf{y}_i - \mathbf{y}}{h} \right\|^2 \right) - \sum_{i=1}^{n_h} \mathbf{y}_i w_i g \left(\left\| \frac{\mathbf{y}_i - \mathbf{y}}{h} \right\|^2 \right) \right] \\ &= \frac{C_h}{2h^2} \left[\mathbf{y} \sum_{i=1}^{n_h} w_i g \left(\left\| \frac{\mathbf{y}_i - \mathbf{y}}{h} \right\|^2 \right) - \sum_{i=1}^{n_h} \mathbf{y}_i w_i g \left(\left\| \frac{\mathbf{y}_i - \mathbf{y}}{h} \right\|^2 \right) \right] \\ &= \frac{C_h}{2h^2} \left[\sum_{i=1}^{n_h} w_i g \left(\left\| \frac{\mathbf{y}_i - \mathbf{y}}{h} \right\|^2 \right) \right] \times \left[\mathbf{y} - \frac{\sum_{i=1}^{n_h} \mathbf{y}_i w_i g \left(\left\| \frac{\mathbf{y}_i - \mathbf{y}}{h} \right\|^2 \right)}{\sum_{i=1}^{n_h} w_i g \left(\left\| \frac{\mathbf{y}_i - \mathbf{y}}{h} \right\|^2 \right)} \right] \end{aligned} \quad (2-14)$$

我們可把式(2-14)分為兩項，前項 $\sum_{i=1}^{n_h} w_i g \left(\left\| \frac{\mathbf{y}_i - \mathbf{y}}{h} \right\|^2 \right)$ 部分是在 \mathbf{y} 的機率分佈值，其值為正，欲找出峰值，我們再令式(2-14)其值為 0。

$$\nabla f(\mathbf{y}) = 0$$

可由後項部分為 0 得到平均位移向量：

$$\mathbf{y}_{new} = \frac{\sum_{i=1}^{n_h} \mathbf{y}_i w_i g \left(\left\| \frac{\mathbf{y}_i - \mathbf{y}_{old}}{h} \right\|^2 \right)}{\sum_{i=1}^{n_h} w_i g \left(\left\| \frac{\mathbf{y}_i - \mathbf{y}_{old}}{h} \right\|^2 \right)} \quad (2-15)$$

平均位移向量會指向目標模型與候選模型相似度最大的方向，把 \mathbf{y}_{old} 位置替換成 \mathbf{y}_{new} ，一直重複以上步驟，由此即可藉由疊代的方法去逼近到最接近目標模型的位置。

2.1.5 平均位移演算法流程圖

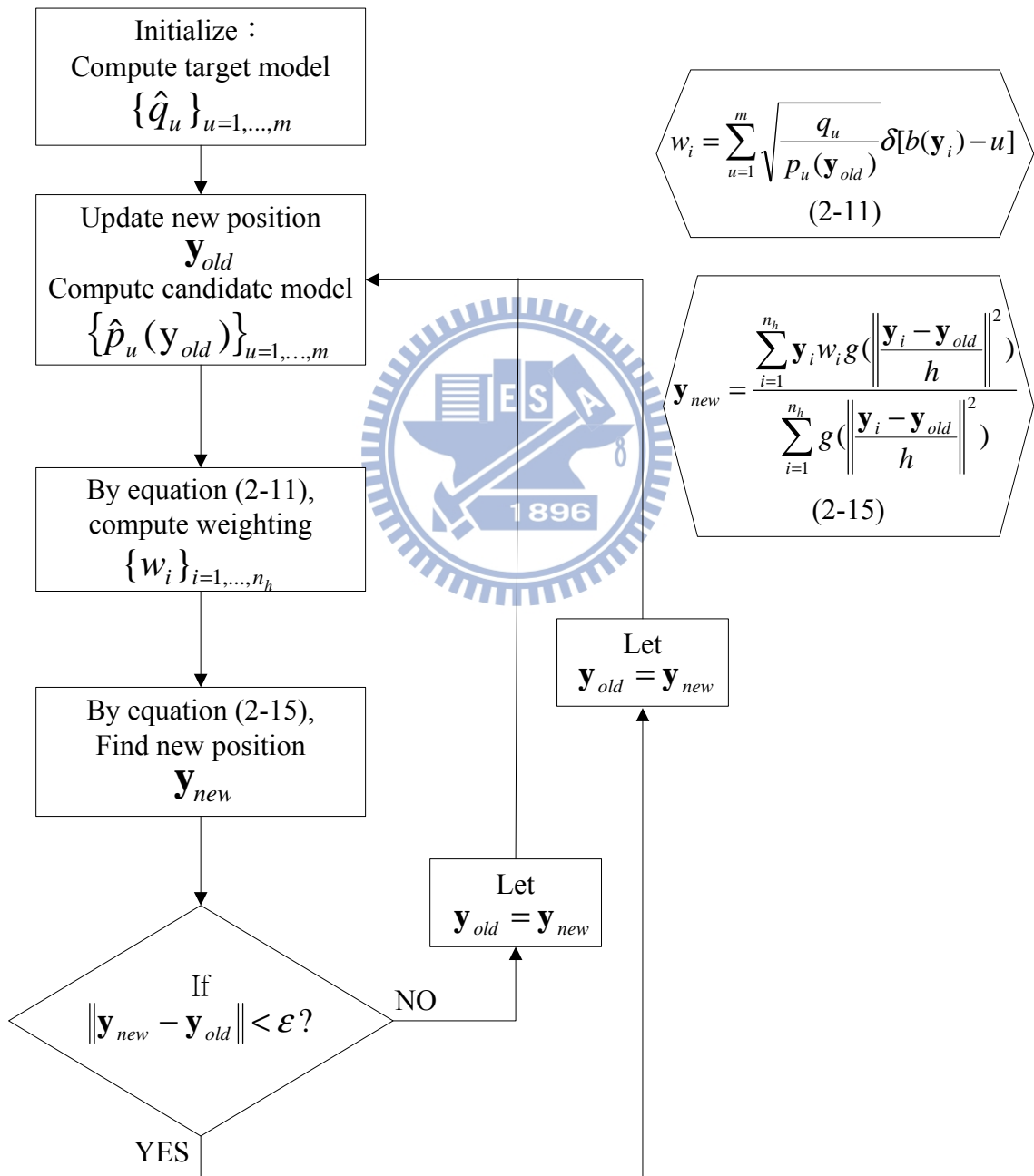


圖 2.1-2: 平均位移演算法流程圖

2.2 三維平均位移演算法

傳統平均位移演算法提供了一個快速且有效的追蹤方法。但因為是利用顏色作為樣本模型的因素，若是遇到畫面中有相同顏色的物體或背景時，此演算法的穩健度會大幅下降。然而微軟公司在 2010 年發表了新一代的感測器，Kinect 感測器，它是一個外形類似網路攝影機的裝置。Kinect 有兩個鏡頭，中間的鏡頭為 RGB 彩色攝影機，左右兩邊則分別為紅外線發射器和紅外線 CMOS 攝影鏡頭。利用紅外線發射器和紅外線 CMOS 鏡頭，可以及時得到物體的空間資訊，利用這新的空間資訊，區分出相同顏色不同物體的狀況，進一步增進演算法的穩定度。下面開始介紹三維平均位移演算法的演算過程。

2.2.1 三維平均位移演算法模型定義

以傳統二維的平均位移演算法的概念為出發點，不同的地方在於原本的方法是在 uv 平面上的二維追蹤，而三維平均位移演算法是轉換到在三維方向上的追蹤。假設在 uv 平面上一點 \mathbf{x} ， Z_x 為在 uv 平面中位置 \mathbf{x} 由 Kinect 所得到的該點在三維空間中的深度值，當有了 \mathbf{x} 與 Z_x 後，就能藉由 Kinect 內部所提供的 API 轉換式 $T: R^2 \rightarrow R^3$ ，得到該像素在空間中的三維位置 \mathbf{x}_{3D} ，轉換的座標如圖 2.2-1 所示。

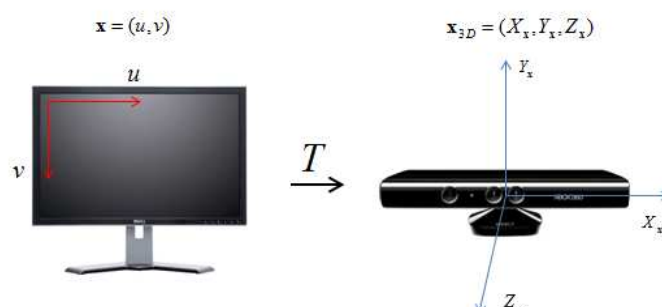


圖 2.2-1: [11]uv 座標與 kinect 三維座標

接著定義候選區域的中心位置為 \mathbf{y}_{3D} 。目標模型的顏色資訊為 \mathbf{q} ，候選模型的則定義為 $\mathbf{p}(\mathbf{y}_{3D})$ ：

$$\mathbf{q} = \{q_u\}_{u=1,\dots,m} \quad \sum_{u=1}^m q_u = 1 \quad (2-16)$$

候選區域模型為：

$$\mathbf{p}(\mathbf{y}_{3D}) = \{p_u(\mathbf{y}_{3D})\}_{u=1,\dots,m} \quad \sum_{u=1}^m p_u = 1 \quad (2-17)$$

接下來定義三維的模型表示。首先假設目標模型內的像素數目為 n ， $\{\mathbf{x}_i^*\}_{i=1,\dots,n}$ 代表原本該像素所在 uv 座標軸的位置，而 $\{\mathbf{x}_{3D,i}^*\}_{i=1,\dots,n}$ 代表轉換至三維空間中的位置，*表示目標區域的中央位置定義為 0。函數 $b: R^2 \rightarrow \{1,\dots,m\}$ ，且 $b(\mathbf{x}_i^*)$ 代表屬於 uv 平面上該位置的像素的顏色量化至 $1,\dots,m$ 個 bin，總共有 m 種顏色的 bin 值。目標模型的 $u=1,\dots,m$ 的顏色機率表示式如下所示：

$$q_u = C_{3D} \sum_{i=1}^n k_{3D}(\mathbf{x}_{3D,i}^*, \Sigma) \delta[b(\mathbf{x}_i^*) - u] \quad (2-18)$$

$$C_{3D} = \frac{1}{\sum_{i=1}^n k_{3D}(\mathbf{x}_{3D,i}^*, \Sigma)} \quad (2-19)$$

在式(2-18)與式(2-19)中， $k_{3D}(\mathbf{x}_{3D,i}^*, \Sigma)$ 為一個三維的 Gaussian kernel function， Σ 為共變異數矩陣，其代表空間的機率分佈情況，最高的數值為物體在三維空間中的中央，而離中央點越遠的話則其值會越低。與先前演算法不同的是，之前的 kernel 方程式是為 uv 畫面中的二維座標，而新的 kernel 方程式則轉為空間中的三維座標，再藉由三維的 kernel function 算出機率值。

因此候選模型的顏色機率表示式如下：

$$p_u(\mathbf{y}_{3D}) = C_{3D,h} \sum_{i=1}^{n_h} k_{3D}(\mathbf{y}_{3D,i} - \mathbf{y}_{3D}, \Sigma_{3D,p}) \delta[b(\mathbf{x}_i^*) - u] \quad (2-20)$$

$$C_{3D,h} = \frac{1}{\sum_{i=1}^{n_h} k_{3D}(\mathbf{y}_{3D,i} - \mathbf{y}_{3D}, \Sigma_{3D,p})} \quad (2-21)$$

其中 $\Sigma_{3D,p}$ 為目標模型在空間中分布的共變異數矩陣，而 Gaussian kernel function 如下所示：

$$k_{3D}(\mathbf{y}_{3D,i} - \mathbf{y}_{3D}, \Sigma_{3D,p}) = \frac{\exp(-\frac{1}{2}(\mathbf{y}_{3D,i} - \mathbf{y}_{3D})^T (\Sigma_{3D,p}^{-1})(\mathbf{y}_{3D,i} - \mathbf{y}_{3D}))}{(2\pi)^{\frac{3}{2}} |\Sigma_{3D,p}|^{\frac{1}{2}}} \quad (2-22)$$

利用目標區域在空間中的分佈情況為基礎，計算出新的機率分佈值。

2.2.2 演算法方程式推導

與傳統的二維平均位移演算法相同，三維的演算法同樣也是用 Bhattacharyya coefficient 來當作相似度量測的函數。目標區域與候選區域彼此間的距離定義為：

$$d(\mathbf{y}_{3D}) = \sqrt{1 - \rho[\mathbf{p}(\mathbf{y}_{3D}), \mathbf{q}]} \quad (2-23)$$

ρ 為目標區域 \mathbf{q} 與候選區域 \mathbf{p} 的 Bhattacharyya coefficient。

$$\rho(\mathbf{y}_{3D}) \equiv \rho[\mathbf{p}(\mathbf{y}_{3D}), \mathbf{q}] = \sum_{u=1}^m \sqrt{p_u(\mathbf{y}_{3D}) q_u} \quad (2-24)$$

將 $\rho[\mathbf{p}(\mathbf{y}_{3D}), \mathbf{q}]$ 利用泰勒展開式於 $\mathbf{y}_{3D} = \mathbf{y}_{3D,old}$ 展開，在疊代開始前，位置 $\mathbf{y}_{3D,old}$ 代表上一張 frame 的目標中心位置 \mathbf{x}_{3D} ，疊代後， $\mathbf{y}_{3D,old}$ 會隨著疊代出來的 $\mathbf{y}_{3D,new}$ 做更新。

$$\begin{aligned} \rho[\mathbf{p}(\mathbf{y}_{3D}), \mathbf{q}] &= \sum_{u=1}^m \sqrt{p_u(\mathbf{y}_{3D}) q_u} = \sum_{n=0}^{\infty} \frac{\partial^n \rho[\mathbf{p}(\mathbf{y}_{3D}), \mathbf{q}]}{\partial \mathbf{p}(\mathbf{y}_{3D})^n} \Big|_{\mathbf{y}_{3D}=\mathbf{y}_{3D,old}} [\mathbf{p}(\mathbf{y}_{3D}) - \mathbf{p}(\mathbf{y}_{3D,old})]^n \\ &\approx \rho[\mathbf{p}(\mathbf{y}_{3D}), \mathbf{q}] \Big|_{\mathbf{y}_{3D}=\mathbf{y}_{3D,old}} + \frac{\partial \rho[\mathbf{p}(\mathbf{y}_{3D}), \mathbf{q}]}{\partial \mathbf{p}(\mathbf{y}_{3D})} \Big|_{\mathbf{y}_{3D}=\mathbf{y}_{3D,old}} [\mathbf{p}(\mathbf{y}_{3D}) - \mathbf{p}(\mathbf{y}_{3D,old})] \end{aligned}$$

$$\begin{aligned}
&= \sum_{u=1}^m \sqrt{p_u(\mathbf{y}_{3D})q_u} \Big|_{\mathbf{y}_{3D}=\mathbf{y}_{3D,old}} + \sum_{u=1}^m \frac{q_u}{2\sqrt{p_u(\mathbf{y}_{3D})q_u}} \Big|_{\mathbf{y}_{3D}=\mathbf{y}_{3D,old}} [p_u(\mathbf{y}_{3D}) - p_u(\mathbf{y}_{3D,old})] \\
&= \frac{1}{2} \sum_{u=1}^m \sqrt{p_u(\mathbf{y}_{3D,old})q_u} + \frac{1}{2} \sum_{u=1}^m p_u(\mathbf{y}_{3D}) \sqrt{\frac{q_u}{p_u(\mathbf{y}_{3D,old})}} \quad (2-25)
\end{aligned}$$

將式(2-20)代入到式(2-25)，可得：

$$\begin{aligned}
\rho(\mathbf{y}_{3D}) &\equiv \rho[\mathbf{p}(\mathbf{y}_{3D}), \mathbf{q}] \approx \frac{1}{2} \sum_{u=1}^m \sqrt{p_u(\mathbf{y}_{3D,old})q_u} + \frac{1}{2} \sum_{u=1}^m p_u(\mathbf{y}_{3D}) \sqrt{\frac{q_u}{p_u(\mathbf{y}_{3D,old})}} \\
&= \frac{1}{2} \sum_{u=1}^m \sqrt{p_u(\mathbf{y}_{3D,old})q_u} \\
&\quad + \frac{1}{2} \sum_{u=1}^m C_{3D,h} \sum_{i=1}^{n_h} k_{3D}(\mathbf{y}_{3D,i} - \mathbf{y}_{3D}, \Sigma_{3D,p}) \delta[b(\mathbf{y}_i) - u] \sqrt{\frac{q_u}{p_u(\mathbf{y}_{3D,old})}} \\
&= \frac{1}{2} \sum_{u=1}^m \sqrt{p_u(\mathbf{y}_{3D,old})q_u} + \frac{1}{2} C_{3D,h} \sum_{i=1}^{n_h} w_i k_{3D}(\mathbf{y}_{3D,i} - \mathbf{y}_{3D}, \Sigma_{3D,p}) \quad (2-26)
\end{aligned}$$

$$\text{其中 } w_i = \sum_{u=1}^m \sqrt{\frac{q_u}{p_u(\mathbf{y}_{3D,old})}} \delta[b(\mathbf{y}_i) - u]$$

在式(2-26)中，前項與 \mathbf{y}_{3D} 獨立，對 \mathbf{y}_{3D} 而言為常數，會影響 $\rho(\mathbf{y}_{3D})$ 的只有

後項部分，所以定義後項為：

$$f_{3D}(\mathbf{y}_{3D}) = \frac{1}{2} C_{3D,h} \sum_{i=1}^{n_h} w_i k_{3D}(\mathbf{y}_{3D,i} - \mathbf{y}_{3D}, \Sigma_{3D,p}) \quad (2-27)$$

方程式對 \mathbf{y}_{3D} 取梯度微分以取得最大值：

$$\begin{aligned}
\nabla f_{3D}(\mathbf{y}_{3D}) &= \frac{1}{2} C_{3D,h} \sum_{i=1}^{n_h} w_i \left(-\frac{1}{2}\right) \Sigma_{3D,p}^{-1} (\mathbf{y}_{3D,i} - \mathbf{y}_{3D}) k_{3D}(\mathbf{y}_{3D,i} - \mathbf{y}_{3D}, \Sigma_{3D,p}) \\
&= -\frac{C_{3D,h}}{4} \sum_{3D,p}^{-1} \sum_{i=1}^{n_h} w_i (\mathbf{y}_{3D,i} - \mathbf{y}_{3D}) k_{3D}(\mathbf{y}_{3D,i} - \mathbf{y}_{3D}, \Sigma_{3D,p})
\end{aligned}$$

$$\begin{aligned}
&= \frac{C_{3D,h}}{4} \sum_{3D,p}^{-1} \left[\sum_{i=1}^{n_h} w_i k_{3D}(\mathbf{y}_{3D,i} - \mathbf{y}_{3D}, \Sigma_{3D,p}) \right] \\
&\quad \times \left[\mathbf{y}_{3D} - \frac{\sum_{i=1}^{n_h} \mathbf{y}_{3D,i} w_i k_{3D}(\mathbf{y}_{3D,i} - \mathbf{y}_{3D}, \Sigma_{3D,p})}{\sum_{i=1}^{n_h} w_i k_{3D}(\mathbf{y}_{3D,i} - \mathbf{y}_{3D}, \Sigma_{3D,p})} \right]
\end{aligned} \tag{2-28}$$

可將式(2-28)分為兩項，前項 $\sum_{i=1}^{n_h} w_i k_{3D}(\mathbf{y}_{3D,i} - \mathbf{y}_{3D}, \Sigma_{3D,p})$ 部分是在 \mathbf{y}_{3D} 的機率分佈值，其值為正，為了取得峰值，我們再令式(2-28)值為 0。

$$\nabla f_{3D}(\mathbf{y}_{3D}) = 0$$

可由後項部分得到平均位移向量：

$$\mathbf{y}_{3D,new} = \frac{\sum_{i=1}^{n_h} \mathbf{y}_{3D,i} w_i k_{3D}(\mathbf{y}_{3D,i} - \mathbf{y}_{3D,old}, \Sigma_{3D,p})}{\sum_{i=1}^{n_h} w_i k_{3D}(\mathbf{y}_{3D,i} - \mathbf{y}_{3D,old}, \Sigma_{3D,p})} \tag{2-29}$$

當經過數次的疊代之後，最後將會取得收斂至位置 $\mathbf{y}_{3D,new}$ 的候選區域，也就是演算法最後估測出來的目標位置。在運算下一張的 frame 之前，這張 frame 跟上一張 frame 中的目標可能會有旋轉或是變形的情況發生，為了改善此情形，我們加入使用主成份分析法(Principal Component Analysis)，估測出此 frame 中目標在空間中新的分佈狀況。我們使用一個三維的橢圓球分佈來近似為目標的分佈，以此作為計算下一張 frame 中目標在空間中的分佈機率。在計算主成份分析之前必須先知道 frame 中的哪些像素是屬於目標的，因此我們會利用深度的不同來分離出屬於目標前景的像素與屬於環境背景的像素。

2.2.3 目標的深度範圍判定

在一般的情況下，當背景與前景有一定的距離時，利用深度資訊將可以有效地分離出前景與背景。首先，將所選取的目標搜尋範圍中的像素深度做出一個量

化的統計圖，如圖 2.2-2 所示，左邊為量化後的統計圖，橫軸為將深度 0~10 公尺的範圍分成 1024 個刻度，縱軸為屬於此深度的像素累加數目。決定目標深度範圍的方法為，從上一個目標位置 y_{3D} 的深度開始，分別往淺和深的方向做搜尋，如果發現此深度刻度下的累計像素數目為 0 時，表示到達目標深度範圍的邊界，邊界由圖中的紅色線跟綠色線表示。右邊為在畫面中的深度表示圖，中央框住人體的黑色方框為所選取的目標搜尋範圍，黑色方框中的亮點表示屬於此目標深度範圍的像素，亮度的深淺表示顏色的機率分佈，所以只要是有亮度的地方，即表示此像素的三維空間深度落在目標深度範圍內，意指屬於目標前景。

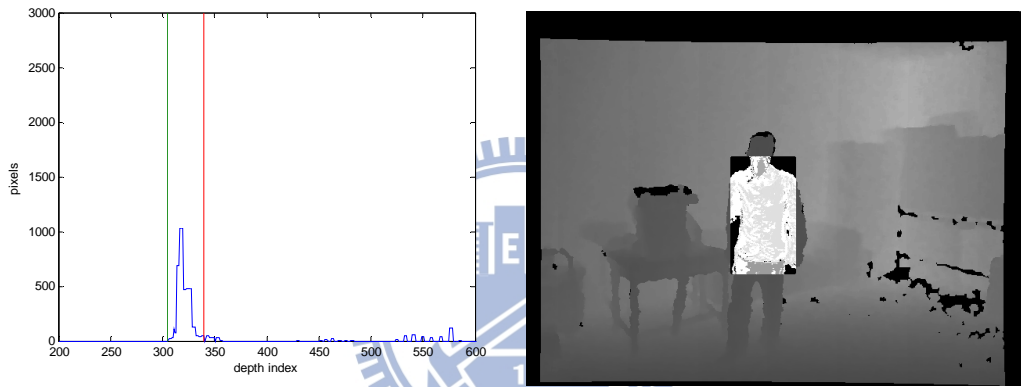


圖 2.2-2: [11]深度量化統計圖 / 深度範圍示意圖

若是一開始要初始化的情況，則沒有之前平均位移所收斂而得的 y_{3D} ，因此我們使用另一個方法求出目標位置的初始深度。

$$Z_m = \frac{1}{\sum_{x \in O} 1} \sum_{x \in O} Z_x \quad (2-30)$$

其中 $O: \{x | x \in \text{目標範圍中央} \frac{1}{9}, Z_x > 0\}$

取 uv 平面中 x 屬於初始化目標範圍中央的 $\frac{1}{9}$ 大小的矩形，假設在初始化過程中，越靠近中央的像素表示可信賴程度越高，另外要濾除掉 Kinect 得到的錯誤資訊的點，因為 Kinect 的深度資訊是利用紅外線 pattern 來做為深度偵測，但若是遇到反光嚴重，或者是深度距離太近時，紅外線偵測會有錯誤的情況發生，此時我們所量測到的深度值便會是 0，為了準確地估測出目標深度，必須濾除掉

錯誤點，如此一來，就能在沒有平均位移所收斂出來的 \mathbf{y}_{3D} 狀況下，找出目標的初始深度。

藉由上述方法就可以找出屬於目標深度的像素，接著即可利用主成份分析法來分析目標在空間中的分佈狀況。

2.2.4 主成份分析法 (Principal Component Analysis)

主成份分析法是一個常用的數學統計分析法，其結果是以新的互相不相關的變數來取代原有相關的變數，而此新的變數為原有變數的線性組合。所以說使用此法後可以得到一個線性變換，數據會經由這線性變換後轉換到新的彼此正交的座標軸，而所有數值投影的第一大變異數會在第一個座標軸上，第二大變異數會在第二個座標軸上，依此類推，這方法可以找出全部資料的分佈情形。其目的是希望用較少的變數去解釋原始資料，並獲取足夠的資訊。論文[10]中也提出使用這方法找出目標在 uv 平面上的分佈，由此推算出物體在平面上的旋轉與平移的情形。如今將此方法由二維的概念推廣至三維，同樣利用此法找出物體在三維空間中的分佈情況。

首先假設標準化矩陣(standardized matrix) \mathbf{S} ($\mathbf{S} \in R^3$) 的共變異數矩陣 Σ 為：

$$\Sigma = \frac{1}{n_h - 1} \mathbf{S}^T \mathbf{S} \quad (2-31)$$

其共變異數矩陣 Σ 的特徵方程式為：

$$\Sigma \mathbf{v} = \lambda \mathbf{v} \quad (2-32)$$

解出特徵方程式(2-32)後可以得到特徵值 $\{\lambda_i\}_{i=1,2,3}$ 與特徵向量 $\{\mathbf{v}_i\}_{i=1,2,3}$ ，其中最大特徵值所對應到的特徵向量代表最大的主成份方向，變異數經過投影後會為最大值，稱為第一主成份，而第二大特徵所對應到的特徵向量代表第二大的主成份方向，稱為第二主成份，同理最小的特徵值所對應到的特徵向量代表最小的主成份的方向，稱為第三主成份。如圖 2.2-3 所示。

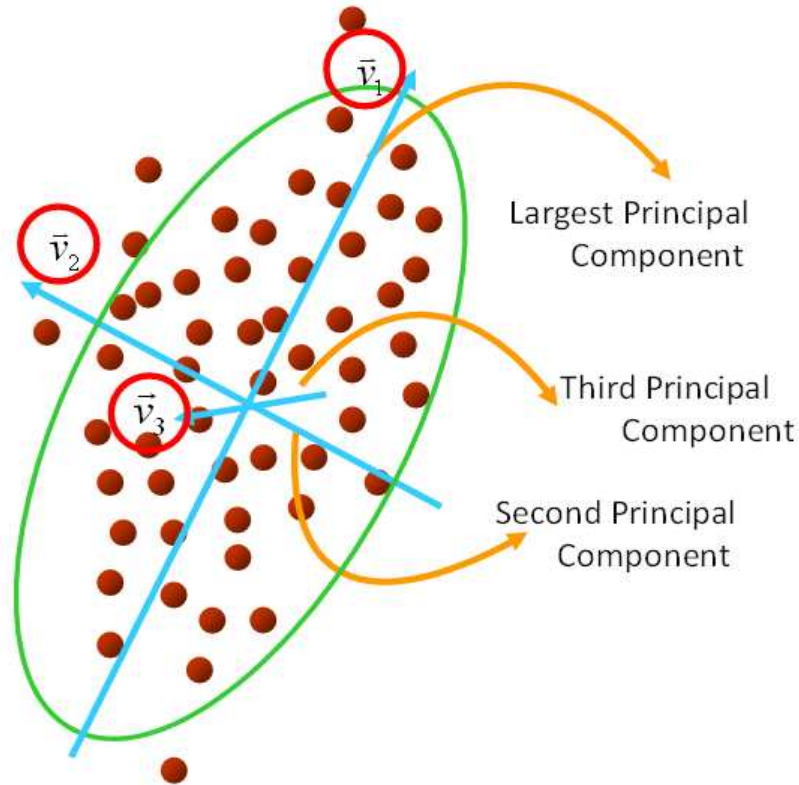


圖 2.2-3: 主成份分析示意圖

將目標物在空間中所有的分佈點代入主成份分析，在過程中會先得到共變異數矩陣 Σ 與中心 μ_{PCA} ，這兩者即是為了計算下一張畫面時的 $\Sigma_{3D,p}$ 與 $\mathbf{y}_{3D,old}$ ，利用上述兩者而得到的空間分佈狀況就是式(2-20)中的 kernel 方程式，

$$k_{3D}(\mathbf{y}_{3D,i} - \mathbf{y}_{3D,old}, \Sigma_{3D,p})。$$

2.2.5 色彩空間表示法

RGB、HSV、YCbCr 都是一般常見的色彩空間表示法，但在追蹤目標物時，往往會因為照度(illumination)的變化而改變其色彩向量的數值，導致我們的追蹤演算法會有偏移情形產生，論文[10]提出使用 normalized RGB 的色彩空間表示式，定義如下：

$$r = \frac{R}{R+G+B}, \quad g = \frac{G}{R+G+B}, \quad b = \frac{B}{R+G+B} \quad (2-33)$$

其中 R 、 G 、 B 分別代表在 RGB 色彩空間中所對應的值，論文[10]中的實驗顯示經由歸一化後色彩空間比原本用 RGB 色彩空間追蹤，有更高的穩健性。

在三維平均位移演算法中，想要從 YCbCr 的方向著手，Y 為顏色的流明 (luminance) 成份，Cb 和 Cr 則為藍色和紅色的濃度偏移量成份，一般而言，若一個物體的亮度改變時，應該只會有 Y 值產生變化，所以在建立色彩模型時，就特別對 Y 做處理，希望能有更高抵抗亮度變化的能力。

定義模型 \hat{q}_u^* 的顏色機率表示式如下：

$$\hat{q}_u = C \sum_{i=1}^n k_{3D} (\|x_i^*\|^2) \delta[b(x_i^*) - u] \quad (2-34)$$

$$C = \frac{1}{\sum_{i=1}^n k_{3D} (\|x_i^*\|^2)} \quad (2-35)$$

$$\hat{q}_u^*(Y, Cb, Cr) = \frac{1}{1 + 6r^*} \left\{ \begin{array}{l} r^* \hat{q}_u(Y + 2, Cb, Cr) + 2r^* \hat{q}_u(Y + 1, Cb, Cr) \\ + \hat{q}_u \\ + 2r^* \hat{q}_u(Y - 1, Cb, Cr) + r^* \hat{q}_u(Y - 2, Cb, Cr) \end{array} \right\} \quad (2-36)$$

其中 r^* 是擴增比例，概念就是在每一個顏色中，把相同 Cb 和 Cr 且在 Y 附近的顏色機率一併增加，由此法就能在追蹤過程中，對於亮度的改變有一定容忍度的提升。

2.2.6 三維平均位移演算法流程圖

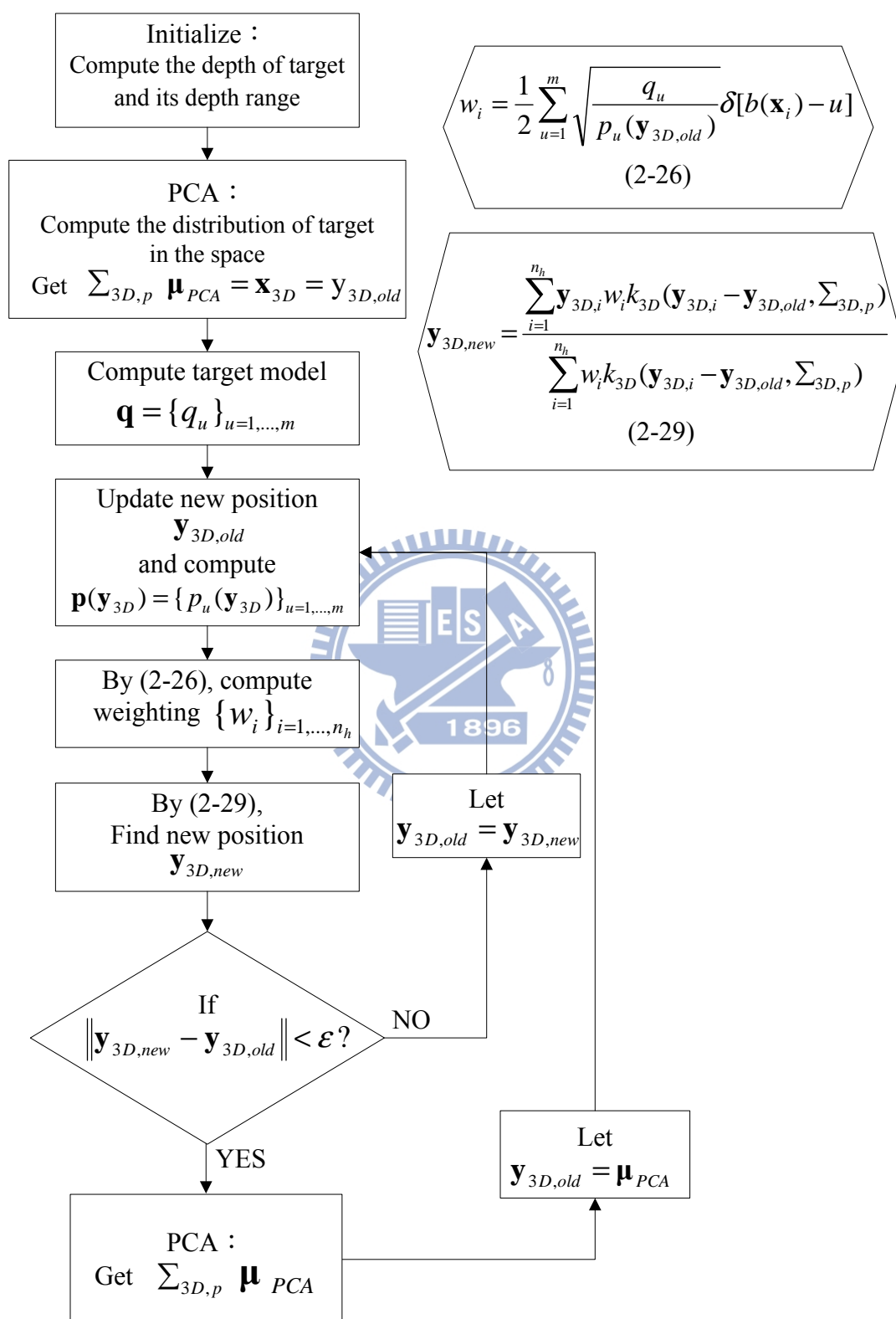


圖 2.2-4: 三維平均位移演算法流程圖

第三章 目標模型的更新

舊有的三維平均位移演算法的流程圖中，並無考慮追蹤的目標物發生旋轉而產生外觀顏色改變的情況(NMU)，因此我們需要一個能夠有效、迅速、準確的目標模型更新方式，來使得我們的目標模型能夠隨著追蹤的目標物，因為外觀的顏色產生改變來進行調適，進一步達成穩定追蹤的目標。

3.1 整體目標模型更新方式

傳統的目標模型更新方法皆是建構在傳統的平均演算法上，也就是單純利用顏色機率模型來做線性加權進行調變。

在論文[13]中，所使用的目標模型更新方式如下：

$$q_{t+1} = (1 - \alpha)q_t + \alpha p_t \quad (3-1)$$

α 為模型更新速度因子。式(3-1)表示說我們當下這個 frame 的目標顏色機率模型，是由上一個 frame 的目標顏色機率模型和候選顏色機率模型進行簡單的線性加權後，所組合而來。由於是同時對整個目標模型進行更新，意指將其視為一個整體，此方法通稱為線性整體模型更新(LTMU)，通常 α 取 0.1 即可。

由於這樣的更新方式表示說目標模型更新的速度至始至終皆為相同，因此在論文[17]中則採取了另外一種目標模型更新方式：

$$q_{t+1} = N_C [q_t + e^{-\alpha(1-\rho)} p_t] \quad (3-2)$$

N_C 為正規化的係數，目的使得 $\sum_{u=1}^m q_{u,t+1} = 1$ 。 ρ 為前面提過的 Bhattacharyya coefficient，表示目標顏色機率模型與候選顏色機率模型的相似度。此方法稱之為非線性整體模型更新(NLTMU)。由式(3-2)可以看出，在選定模型更新速度因子 α 的值之後，當相似度 ρ 越低時，候選顏色機率模型所佔的比例就會越小，意指在這樣的情況下，我們比較不相信候選顏色機率模型的值，比較偏向於保留原

本的目標顏色機率模型的值，也就是說，當相似度較低時，目標模型更新的速度就會變慢，相似度高時，目標模型更新的速度就會變快，最快的情況是目標模型與候選模型完全相同，此時目標模型與候選模型各佔下一個 frame 的目標模型的成份各半。

3.2 選擇性子模型更新方式

由於 LTMU 與 NLTMU 皆把目標模型當成一個整體去做更新，並沒有考慮到當在追蹤的過程中，目標顏色機率模型中每個顏色 bin 所代表的機率值其改變的速度並不全都是相同的，因此若將目標模型視為一個整體而去做整體性的目標模型更新，這樣的方式顯得過於粗糙，而且得到的追蹤結果可能逐漸偏離目標，若我們用這種逐漸不準確的結果去整體性地更新目標模型，往往會導致追蹤結果的不準確，因此論文[18]提出了應將特徵模型中的每個分量視為單獨的個體，也就是把各個分量皆視為獨立的子模型(sub-model)，然後再根據每個子模型的情況來進行更新，此方法稱之為選擇性子模型更新(SSMU)，方法如下所示。

在基於 Bhattacharyya coefficient 的模型相似度函數中，由於特徵模型中每個分量是獨立的，因此模型相似度即為各個子模型分量的相似度的線性總和，所以在此我們先定義一個函數，匹配貢獻度函數(Matching Contribution Degree)：

$$MCD_u = \frac{\sqrt{p_u q_u}}{\rho} = \frac{\sqrt{p_u q_u}}{\sum_{u=1}^m \sqrt{p_u q_u}} \quad (3-3)$$

其所代表的意思為每個子模型的相似度佔全部模型相似度的比重。

接著說明該如何選取需要進行更新的子模型。首先利用 Bhattacharyya coefficient 來決定追蹤結果的準確性，如果相似度 ρ 小於一定程度，表示此次追蹤結果受到劇烈的干擾導致準確性不佳，意指會包含過多的錯誤資訊，所以不應更新目標模型，仍舊保留前一個 frame 的目標模型；若相似度 ρ 滿足一定要求，則根據我們所算出來的每個子模型的匹配貢獻度大小排序，前 k 個貢獻度較高的

子模型表示當前模型與目標模型是較為吻合，為了避免過度更新而造成目標模型偏移，因此這 k 個子模型予以保留，不進行更新，選擇剩下的 $m - k$ 個貢獻度較小的子模型，表示當前模型中與目標模型吻合較差的子模型來進行更新：

$$\begin{cases} k = \arg \min_b (\sum_{u=1}^b MCD_u \geq T_b \rho), & \rho \geq \gamma \\ k = m, & \rho < \gamma \end{cases} \quad (3-4)$$

其中 γ 為模型相似度可靠性門檻， $T_b \in [0, 1]$ 為子模型更新比例，當 $T_b = 0$ 時，可得 $k = 0$ ，表示全部 m 個子模型都要更新，此時等同於整體目標模型更新方式；當 $T_b = 1$ 時，可得 $k = m$ ，表示全部 m 個子模型都不需要更新，此時等同於不更新目標模型方式。

在決定好選取哪些子模型要更新後，接著就是如何更新的方式了。更新的情況可分成五點討論：

- (1) $q_u = p_u = 0$ ：表示候選模型和目標模型中均未出現該子模型，因此不需更新。
- (2) $q_u > 0, p_u = 0$ ：表示目標模型有的子模型在候選模型裡不再出現，這表示追蹤目標外觀發生變化的情況，此時需更新子模型，為避免模型對變化過於敏感，採用一定比例進行加權。
- (3) $q_u = 0, p_u > 0$ ：表示目標模型沒有的子模型在候選模型裡出現，這也表示追蹤目標外觀發生變化的情況，此時需更新子模型，並同樣採用一定的比例進行加權。
- (4) $q_u > 0, p_u > 0, u \geq k$ ：表示該子模型在目標模型與候選模型皆有出現，且其屬於貢獻度較高的 k 個子模型其中之一，所以不需更新子模型。
- (5) $q_u > 0, p_u > 0, u < k$ ：表示該子模型屬於貢獻度較低的 $m - k$ 個子模型其中之一，此時需根據其匹配貢獻度進行加權更新。

綜合以上五點，令 $\beta_u = MCD_u$ ，可得具體的更新方式為：

$$\begin{cases} q_{u_1,t+1} = N_C [(1-\alpha)q_{u_1,t} + \alpha p_{u_1,t}], & \beta_{u_1} = 0 \\ q_{u_2,t+1} = N_C q_{u_2,t}, & \beta_{u_2} > 0, u_2 \leq k \\ q_{u_3,t+1} = N_C [(1-\beta_{u_3})q_{u_3,t} + \beta_{u_3} p_{u_3,t}], & \beta_{u_3} > 0, u_3 > k \end{cases} \quad (3-5)$$

其中 α 為模型更新速度因子， N_C 為歸一化係數(Normalized coefficient)。可發現基於選擇性子模型更新策略，可對目標模型中的每個子模型進行有選擇性地更新，而不再是必須同時更新或不更新整個目標模型。

3.3 結合顏色相似度與中心移動距離的目標模型更新方式

由於我們的目標模型更新方式是適用於三維的位移平均演算法上，表示我們多了三維空間中的深度資訊可用，以下將闡述我們新的目標模型更新方式的演算法由來。

3.3.1 新的目標模型更新方法推導

首先我們先定義兩個 frame 間，所計算出來的目標物的深度差距為：

$$|depth_{t+1} - depth_t| = d \quad (3-6)$$

接著我們定義一個新的函數，顏色 bin 相似度(Similar Degree of every Color Bin)：

$$SDCB_u = \frac{\sqrt{q_u p_u}}{\max(q_u, p_u)}, \quad u = 1, \dots, m \quad (3-7)$$

其代表的意思是，在某個 bin 值下，目標模型與候選模型的顏色機率值的相似程度。接下來是演算法的內容。

首先藉由 Bhattacharyya coefficient 來算出前一個 frame 的目標模型與候選模型的相似度，也就是 ρ ：

$$\rho(\mathbf{y}_{3D}) \equiv \rho[\mathbf{p}(\mathbf{y}_{3D}), \mathbf{q}] = \sum_{u=1}^m \sqrt{p_{u,t}(\mathbf{y}_{3D}) q_{u,t}} \quad (3-8)$$

為了避免下一張 frame 目標模型過度更新而產生追蹤誤差，我們設了一個門檻值來確保說，此時的目標模型是值得信賴的，所以不需更新：

$$\rho \geq \gamma_1: \quad \mathbf{q}_{t+1} = \mathbf{q}_t \quad (3-9)$$

當 ρ 小於門檻值 γ_1 時，此時根據相似度 ρ 與目標中心移動距離會有不同的目標模型更新方式。當目標物中心移動距離小於一定門檻值時，也就是 $\|\mathbf{y}_{3D,new} - \mathbf{y}_{3D,old}\| < \varepsilon_1$ ，對於目標模型更新方式我們主要是將每個顏色 bin 所代表的機率值單獨去做判斷，就像是把每個分量當成是各自獨立的子模型的概念。我們可分為兩種情況討論：

$$1. \text{SDCB}_u \geq \rho: \quad q_{u,t+1} = N_c q_{u,t} \quad (3-10)$$

$$2. \text{SDCB}_u < \rho: \quad q_{u,t+1} = N_c p_{u,t} \quad (3-11)$$

N_c 為歸一化係數。在式(3-10)中，表示在某個 bin 值下，算出來的顏色 bin 相似度比目標模型和候選模型的相似度來得大時，我們認為此 bin 值的顏色機率並無大變化，仍是可信賴，為了避免過度更新，因此繼續採用上一個 frame 的目標模型的顏色機率值，反之，當算出來的顏色機率相似度比模型相似度小時，我們認為此 bin 值的顏色機率發生改變，導致其可信賴的程度降低，因此需要更新，故我們將其更新成上一個 frame 的候選模型的機率值。

當目標物中心移動距離大於一定門檻值時，也就是 $\|\mathbf{y}_{3D,new} - \mathbf{y}_{3D,old}\| \geq \varepsilon_1$ ，我們根據模型相似度的大小而有不一樣的更新方式。

$$1. \gamma_2 \leq \rho < \gamma_1:$$

$$\begin{aligned} \mathbf{q}_{t+1} = & (1 - \min(\frac{\|\mathbf{y}_{3D,new} - \mathbf{y}_{3D,old}\|}{\varepsilon_1} \alpha_1, 0.3)) \mathbf{q}_t \\ & + \min(\frac{\|\mathbf{y}_{3D,new} - \mathbf{y}_{3D,old}\|}{\varepsilon_1} \alpha_1, 0.3) \mathbf{p}_t \end{aligned} \quad (3-12)$$

式(3-12)表示說此時目標模型與候選模型之間已經有了變化，但是彼此間不

同的程度變化不大，所以在更新時，我們利用 ϵ_1 和 $\|y_{3D,new} - y_{3D,old}\|$ 的比值來對目標模型更新的速度進行一定的加權，能夠較為迅速地更新目標模型，使其與真實目標的顏色機率較為相近，當目標中心移動的距離越大時，候選模型所佔的比例就會變大，但為了避免模型更新速度過快，我們令候選模型所佔的比例最大不超過 0.3，最小則為 0.1，此時等同於前面所提過的基本的整體模型更新方式。

2. $\rho < \gamma_2$:

此時表示候選模型與目標模型有很明顯的變化，我們需要加快目標模型更新的速度好提升追蹤的穩定性與準確性，但若是一味地加快更新速度，則會造成引入過多的錯誤資訊，反而使得演算法追蹤的效能下降，因此我們需要去判斷說此時目標物在三維空間中其移動的情況為何，有利於我們來調整目標模型的更新速度。

我們可將目標物在空間中移動的情況分為三種，分別是近似於深度方向的移動、近似於 uv 平面的移動、以及位於上述兩者間的情況。目標模型更新方式茲討論如下。

$$(1) d^2 \geq \epsilon_2 \|y_{3D,new} - y_{3D,old}\|^2 :$$

$$q_{t+1} = N_C [\epsilon_2 \|y_{3D,new} - y_{3D,old}\|^2 e^{-\alpha_2(1-\rho)} q_t + d^2 p_t] \quad (3-13)$$

式(3-13)表示目標物在三維空間中的移動近似於只有深度方向的移動。由於認為目標物在畫面中所在的位置變化可近似為單純深度方向的改變，這代表我們的搜索框是可以準確地框到我們的目標物，也就是候選目標可信賴的程度較高，因此我們藉由 exponential 函數來降低目標模型的比重，並根據模型相似度的大小來調整目標模型與候選模型彼此間的比重，當相似度 ρ 越大時，表示目標物變化較為緩慢，所以 exponential 函數的值會較大，代表目標模型的比重接近候選模型的比重，當相似度 ρ 越小時，表示目標物變化較為劇烈，此時需調快目標模型更新的速度，所以 exponential 函數的值會較小，代表目標模型所佔的比重變得更小。另外我們再藉由目標物的中心移動距離來進一步強化目標模型的更新速

度，也就是進一步加重候選模型的比重。

$$(2) \quad d^2 \leq \varepsilon_3 \left\| y_{3D,new} - y_{3D,old} \right\|^2 :$$

$$q_{t+1} = N_C [d^2 q_t + \varepsilon_3 \left\| y_{3D,new} - y_{3D,old} \right\|^2 e^{-\alpha_3(1-\rho)} p_t] \quad (3-14)$$

式(3-14)代表目標物在三維空間中的移動近似於 uv 平面上的移動，也就是可以不考慮深度方向的移動距離，這代表說目標物在空間中的移動方式可近似為平移，此時搜索框在目標物移動的過程中較為可能會去框到非目標物的錯誤資訊，因此我們就不能讓候選模型的比重過大，以免因為錯誤的顏色資訊而導致目標模型的更新產生誤差。同樣地藉由 exponential 函數來降低候選模型的比重，一樣根據相似度 ρ 來調整候選模型所佔的比例，當 ρ 越大，目標模型更新速度就越快，當 ρ 越小，更新速度就越慢，避免因為引入錯誤資訊而造成更新後的目標模型錯誤。同樣我們藉由目標物的中心移動距離來強化目標模型的更新速度，也就是加重候選模型的比重，這可避免因為目標模型更新速度過慢的問題。

然而在式(3-14)中，我們發現有一個問題存在，就是當 $d^2 \rightarrow 0$ 時，會使得候選模型佔的比重變得極大，這會造成目標模型過度更新的問題，也就是說我們不能將候選模型直接當成新的目標模型來使用，這會造成追蹤結果的穩定性與準確性下降。為了解決這個問題，我們將式(3-14)改寫成另一種形式。

$$d_{xy}^2 \geq (1 - \varepsilon_3) \left\| y_{3D,new} - y_{3D,old} \right\|^2 :$$

$$q_{t+1} = N_C [(1 - \varepsilon_3) \left\| y_{3D,new} - y_{3D,old} \right\|^2 q_t + d_{xy}^2 e^{-\alpha_3(1-\rho)} p_t] \quad (3-15)$$

d_{xy} 表示是扣除深度方向，只剩下在空間中 xy 平面上的移動距離。這樣一來，就能有效地解決當深度變化 d 值過小時的問題。

$$(3) \quad \varepsilon_3 \left\| y_{3D,new} - y_{3D,old} \right\|^2 < d^2 < \varepsilon_2 \left\| y_{3D,new} - y_{3D,old} \right\|^2 :$$

$$\begin{aligned}
\mathbf{q}_{t+1} = N_C [& e^{-\alpha_2(1-\rho)\left(\frac{d^2 - \varepsilon_3 \|y_{3D,new} - y_{3D,old}\|^2}{\varepsilon_2 \|y_{3D,new} - y_{3D,old}\|^2 - \varepsilon_3 \|y_{3D,new} - y_{3D,old}\|^2}\right)} \mathbf{q}_t \\
& + e^{-\alpha_3(1-\rho)\left(\frac{\varepsilon_2 \|y_{3D,new} - y_{3D,old}\|^2 - d^2}{\varepsilon_2 \|y_{3D,new} - y_{3D,old}\|^2 - \varepsilon_3 \|y_{3D,new} - y_{3D,old}\|^2}\right)} \mathbf{p}_t]
\end{aligned} \tag{3-16}$$

當目標物移動介於兩者之間時，我們則利用 d^2 與 $\varepsilon_2 \|y_{3D,new} - y_{3D,old}\|^2$ 和 $\varepsilon_3 \|y_{3D,new} - y_{3D,old}\|^2$ 的比例關係來改變目標模型與候選模型的比例。由式(3-16)，當 $d^2 = \varepsilon_2 \|y_{3D,new} - y_{3D,old}\|^2$ ，此時式(3-16)等同於式(3-13)的結果，當 $d^2 = \varepsilon_3 \|y_{3D,new} - y_{3D,old}\|^2$ ，此時式(3-16)等同於式(3-15)的結果。

圖 3.3-1 為模型更新速度因子 α 與函數 $e^{-\alpha(1-\rho)}$ 在固定相似度 ρ 下的關係圖。我們可由關係圖看出，若函數 $e^{-\alpha(1-\rho)}$ 為目標模型的係數，則為了不讓模型更新速度過快， α 建議選擇 0~5 之間，若函數 $e^{-\alpha(1-\rho)}$ 為候選模型的係數，為了不讓候選模型佔的比例過大， α 建議選擇 5~10 之間。

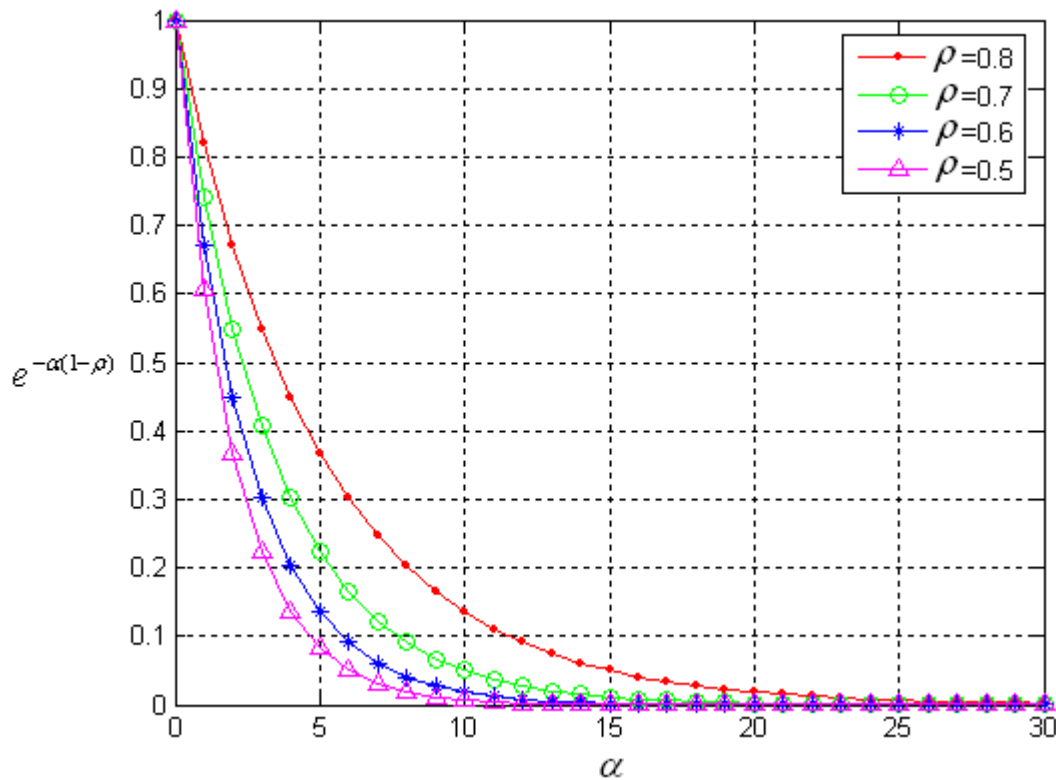


圖 3.3-1: α 與 $e^{-\alpha(1-\rho)}$ 關係圖

藉由上述式子，我們可將此新的目標模型更新方法整理為：

1. $\rho \geq \gamma_1$: $\mathbf{q}_{t+1} = \mathbf{q}_t$
2. $\|y_{3D,new} - y_{3D,old}\| < \varepsilon_1$, $\rho < \gamma_1$:
 - (1) $SDCB_u \geq \rho$: $q_{u,t+1} = N_C q_{u,t}$
 - (2) $SDCB_u < \rho$: $q_{u,t+1} = N_C p_{u,t}$
3. $\|y_{3D,new} - y_{3D,old}\| \geq \varepsilon_1$:
 - (1) $\gamma_2 \leq \rho < \gamma_1$:

$$\mathbf{q}_{t+1} = (1 - \min(\frac{\|y_{3D,new} - y_{3D,old}\|}{\varepsilon_1} \alpha_1, 0.3)) \mathbf{q}_t + \min(\frac{\|y_{3D,new} - y_{3D,old}\|}{\varepsilon_1} \alpha_1, 0.3) \mathbf{p}_t$$

(2) $\rho < \gamma_2$:

- (a) $d^2 \geq \varepsilon_2 \|y_{3D,new} - y_{3D,old}\|^2$:

$$\mathbf{q}_{t+1} = N_C [\varepsilon_2 \|y_{3D,new} - y_{3D,old}\|^2 e^{-\alpha_2(1-\rho)} \mathbf{q}_t + d^2 \mathbf{p}_t]$$
- (b) $d^2 \leq \varepsilon_3 \|y_{3D,new} - y_{3D,old}\|^2$:

$$\mathbf{q}_{t+1} = N_C [(1 - \varepsilon_3) \|y_{3D,new} - y_{3D,old}\|^2 \mathbf{q}_t + d_{xy}^2 e^{-\alpha_3(1-\rho)} \mathbf{p}_t]$$
- (c) $\varepsilon_3 \|y_{3D,new} - y_{3D,old}\|^2 < d^2 < \varepsilon_2 \|y_{3D,new} - y_{3D,old}\|^2$:

$$\mathbf{q}_{t+1} = N_C [e^{-\alpha_2(1-\rho) (\frac{d^2 - \varepsilon_3 \|y_{3D,new} - y_{3D,old}\|^2}{\varepsilon_2 \|y_{3D,new} - y_{3D,old}\|^2 - \varepsilon_3 \|y_{3D,new} - y_{3D,old}\|^2})} \mathbf{q}_t + e^{-\alpha_3(1-\rho) (\frac{\varepsilon_2 \|y_{3D,new} - y_{3D,old}\|^2 - d^2}{\varepsilon_2 \|y_{3D,new} - y_{3D,old}\|^2 - \varepsilon_3 \|y_{3D,new} - y_{3D,old}\|^2})} \mathbf{p}_t]$$

稱此目標模型更新方法為 CCDMU (Combine similar degree of Color with moving Distance of target center Model Updating)。

3.3.2 加入 CCDMU 的三維平均位移演算法流程圖

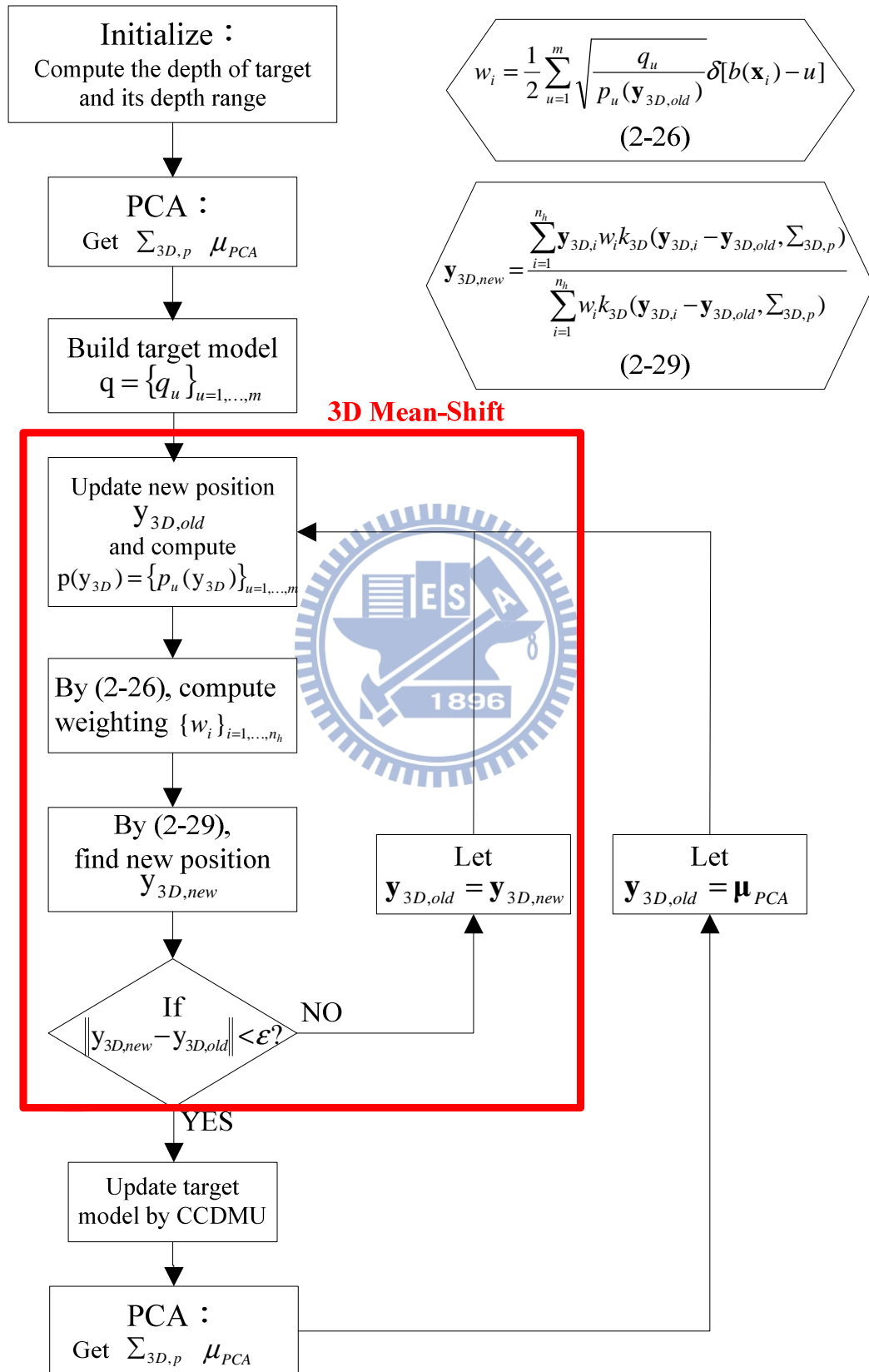


圖 3.3-2: 加入 CCDMU 的三維平均位移演算法流程圖

3.4 加入線上模板資料庫建構的三維物體追蹤演算法

由於所有的目標模型更新方式皆是在既有的目標模型下，根據計算出來的候選模型來進行加權而建構成新的目標模型以利於下一個 frame 使用，這會造成隨著時間的流逝，新的目標模型所包含的顏色資訊會越來越雜，這與實際的情況並不符合，也由於新的目標模型所包含的顏色資訊會是之前好幾個 frame 的累加，只是權重不同，這會造成當目標物在空間中快速旋轉時，因為目標模型更新顏色機率值的速度不夠快，且又包含過多的顏色資訊，而降低了追蹤結果穩定性與準確性。

藉由論文[12]提出利用目標物不同角度的圖片建立出多個顏色模型，我們也想利用同樣的方式來建立好各種適當的目標模型資料庫以利追蹤過程使用，然而論文[12]的缺點是必須在執行追蹤程式前，先行建立好所要追蹤的目標物的各個面向的顏色模型，顯得非常不方便且適應性極低。為了解決上述缺點並同時進一步地增進追蹤的效能，我們提出使用線上模板資料庫建構的追蹤演算法。

3.4.1 線上模板資料庫建構的準則

欲建立目標模型資料庫，必須去判斷計算出來的候選模型與資料庫裡的所有目標模型是否有相似，若有相似，則不儲存，若不相似，則視為新的面向的目標模型儲存進資料庫中。而判斷是否相似的準則我們採用兩個部份：顏色機率模型與三維空間分佈。

1. 顏色機率模型：

利用 Bhattacharyya coefficient 來判斷計算出來的候選模型與資料庫內的所有模型是否有相似：

$$\rho[p'(\mu_{PCA}^{new}), p_{index}(y_{3D,index})] = \sum_{u=1}^m \sqrt{p'_u(\mu_{PCA}^{new}) p_{u,index}(y_{3D,index})} \quad (3-17)$$

前項的 $p'(\mu_{PCA}^{new})$ 代表做完主成份分析之後得到的新的中央位置 μ_{PCA}^{new} ，再代入

求出候選模型的顏色機率值。後項的 $p_{index}(y_{3D,index})$ 代表資料庫裡所有索引的模型的顏色機率值。

2. 三維的空間分佈：

在此我們選用藉由 PCA 所算出的共變異數矩陣 $\Sigma_{3D,P}^{new}$ ，再求解出其三個特徵值來使用，而這三個特徵值即代表 PCA 所算出的三維橢圓球的三軸大小，我們可用此三維橢圓球的三軸來代表候選模型的三維空間分佈情況。由 2.2.4 節的 Eigen equation， $\Sigma v = \lambda v$ ，可以得到三個由大到小的特徵值： λ_{max} ， λ_{med} ， λ_{min} 。同樣地，我們利用 Bhattacharyya coefficient 的概念來判斷候選模型的三維空間分佈與資料庫內的所有模型的三維空間分佈是否有相似。因此我們先將三個特徵值做歸一化(Normalize)的動作：

$$\begin{aligned}\lambda_{max} &= \frac{\lambda_{max}}{\lambda_{max} + \lambda_{med} + \lambda_{min}} \\ \lambda_{med} &= \frac{\lambda_{med}}{\lambda_{max} + \lambda_{med} + \lambda_{min}} \\ \lambda_{min} &= \frac{\lambda_{min}}{\lambda_{max} + \lambda_{med} + \lambda_{min}}\end{aligned}\quad (3-18)$$


然後利用 Bhattacharyya coefficient 的概念來計算三維空間分佈相似度：

$$\rho[\lambda_{candidate}, \lambda_{index}] = \sqrt{\lambda_{max} \lambda_{max,index}} + \sqrt{\lambda_{med} \lambda_{med,index}} + \sqrt{\lambda_{min} \lambda_{min,index}} \quad (3-19)$$

$index$ 代表資料庫內的資料索引。至此，式(3-16)與式(3-18)就可用來判斷資料庫的內容在追蹤的過程中是否要繼續增加。

接下來是判斷資料庫是否新增資料的準則。我們同時針對顏色機率模型相似度與三維空間分佈相似度各自設定門檻值，只要有一個相似度小於門檻值，資料庫就得新增模型。方法如下：

$$1. \rho[p'(\mu_{PCA}^{new}), p_{index}(y_{3D,index})] < T_1 \quad \text{or} \quad \rho[\lambda_{candidate}, \lambda_{index}] < T_2 :$$

資料庫需新增模型，將 $(p'(\mu_{PCA}^{new}), \Sigma_{3D,P}^{new})$ 存入資料庫中。為了給下一次的三

維平均位移演算法使用，我們令

$$\begin{cases} \mathbf{q} = \mathbf{p}'(\boldsymbol{\mu}_{PCA}^{new}) \\ \sum_{3D,p} = \sum_{3D,p}^{new} \end{cases} \quad (3-20)$$

表示說我們直接把新加入的模型拿來做下一個 frame 的追蹤用。

$$2. \rho[\mathbf{p}'(\boldsymbol{\mu}_{PCA}^{new}), \mathbf{p}_{index}(y_{3D,index})] \geq T_1 \quad \text{and} \quad \rho[\lambda_{candidate}, \lambda_{index}] \geq T_2$$

$$\text{and maximize}(\sqrt{\rho[\mathbf{p}'(\boldsymbol{\mu}_{PCA}^{new}), \mathbf{p}_{index}(y_{3D,index})]\rho[\lambda_{candidate}, \lambda_{index}]}):$$

在兩個相似度皆大於門檻值的情形下，表示資料庫內有相似的模型存在，所以不需新增模型。求出資料庫內與候選模型最相似的模型索引，也就是使得兩個相似度的幾何平均數最大的模型索引，藉由前後兩個 frame 所引用的模型索引的不同，可再發展出兩種情形：

$$(1) \text{index}_t = \text{index}_{t-1} :$$

$$\begin{cases} \mathbf{q} = \mathbf{q}_{updating} \\ \sum_{3D,p} = \sum_{3D,p}^{new} \end{cases} \quad (3-21)$$

表示這一個 frame 與上一個 frame 所拿來使用的模型索引是相同的，因此要給下一次三維平均位移演算法使用的目標模型 \mathbf{q} 與共變異數矩陣 $\sum_{3D,p}$ 就不從資料庫中直接拿來使用，而是選用我們所更新出來的目標模型，因為我們認為目標物在每個 frame 皆會有細微的變化，若在此仍繼續選用相同索引的模型給下一次的三維平均位移演算法用，雖然相似度仍然是資料庫中最高的，但這樣就無法讓新的目標模型來適應目標物新的變化情況。

$$(2) \text{index}_t \neq \text{index}_{t-1} :$$

$$\begin{cases} \mathbf{q} = \mathbf{q}_{index_t} \\ \sum_{3D,p} = \sum_{3D,p,index_t} \end{cases} \quad (3-22)$$

表示這一個 frame 與上一個 frame 所拿來使用的模型索引是不同的，因此要給下一次三維平均位移演算法使用的目標模型 \mathbf{q} 與共變異數矩陣 $\sum_{3D,p}$ 就直接取資料庫內屬於該索引的模型來用，代表這個面向的候選模型已經存在資料庫內，

為了追蹤的穩定性與準確性，直接取資料庫內的模型使用，這也能增加追蹤的速度，避免因目標模型的更新速度過慢而導致誤差產生。

綜合式(3-20)、式(3-21)和式(3-22)，可得建立目標模型資料庫的準則為：

$$1. \rho[p'(\mu_{PCA}^{new}), p_{index}(y_{3D,index})] < T_1 \quad \text{or} \quad \rho[\lambda_{candidate}, \lambda_{index}] < T_2 :$$

Save $(p'(\mu_{PCA}^{new}), \Sigma_{3D,p}^{new})$ to database ,

$$\begin{cases} q = p'(\mu_{PCA}^{new}) \\ \Sigma_{3D,p} = \Sigma_{3D,p}^{new} \end{cases}$$

$$2. \rho[p'(\mu_{PCA}^{new}), p_{index}(y_{3D,index})] \geq T_1 \quad \text{and} \quad \rho[\lambda_{candidate}, \lambda_{index}] \geq T_2$$

and maximize $(\sqrt{\rho[p'(\mu_{PCA}^{new}), p_{index}(y_{3D,index})] \rho[\lambda_{candidate}, \lambda_{index}]}) :$

$$(1) \quad index_t = index_{t-1} :$$

$$\begin{cases} q = q_{updating} \\ \Sigma_{3D,p} = \Sigma_{3D,p}^{new} \end{cases}$$

$$(2) \quad index_t \neq index_{t-1} :$$

$$\begin{cases} q = q_{index_t} \\ \Sigma_{3D,p} = \Sigma_{3D,p,index_t} \end{cases}$$



稱此方法為 OTDMU(Online Template Database construction Model Updating)。

3.4.2 加入 OTDMU 的三維平均位移演算法流程圖

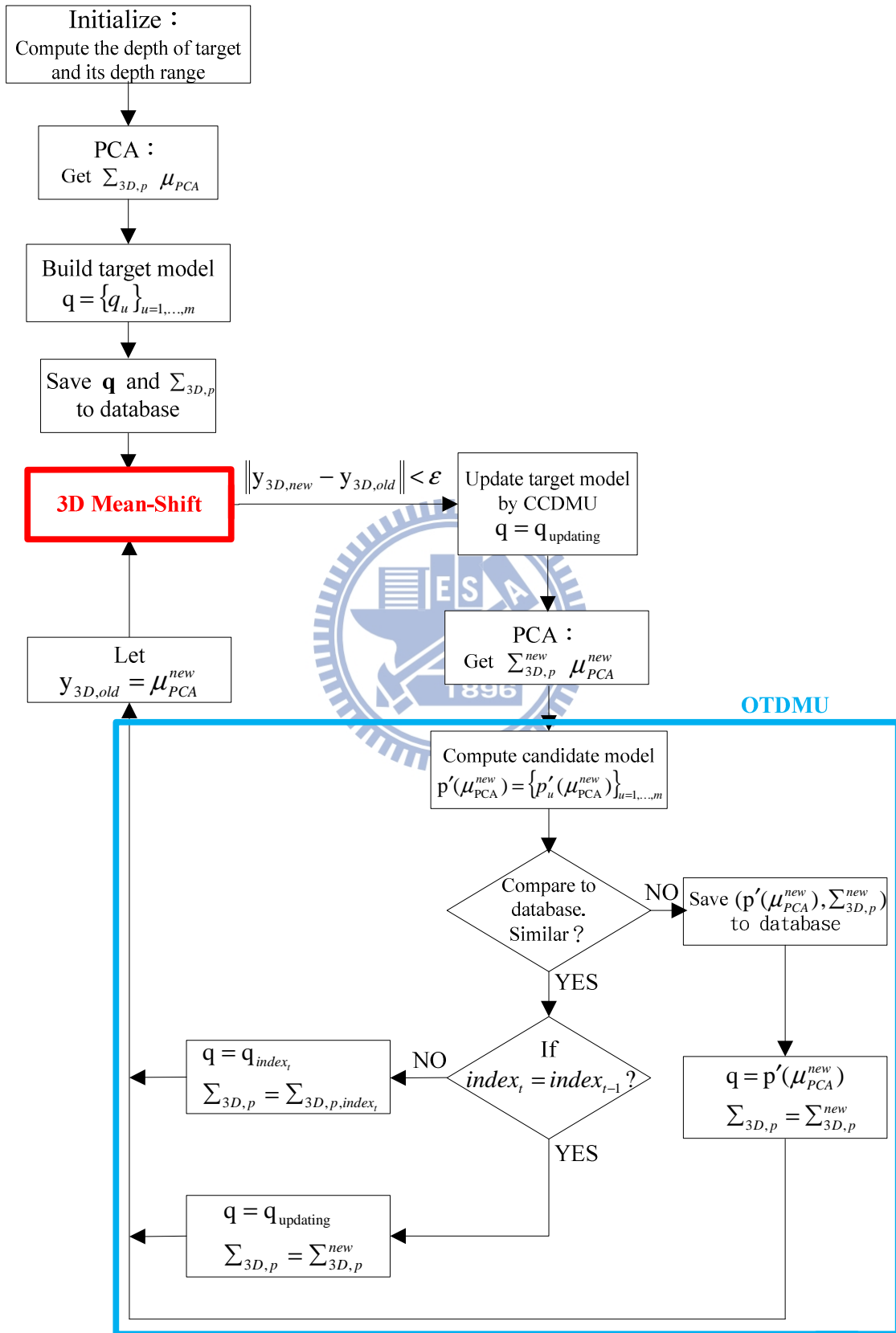


圖 3.4-1: 加入 OTDMU 的三維平均位移演算法流程圖

第四章 系統設計與實驗測試比較

4.1 實驗環境

本實驗測試所使用的硬體平台為桌上型電腦，Intel Core2 Duo CPU E7500 與 2.0GB 的記憶體，使用 Kinect sensor 為本實驗的偵測裝置，軟體環境為 Windows XP sp3，並於 visual studio 2010 上使用 C++ 語言撰寫程式，並透過 openNI 的 API 來抓取 Kinect 資料，由 openCV 的 function 來顯示於螢幕上。

Kinect sensor 的規格如下表：

表 4.1-1: [11]Kinect 規格表

水平視角	57 度
垂直視角	43 度
物理傾斜範圍	±27 度
深度感應器範圍	0.7–6 m
深度感應器	320×240 16-bit 30 frames/sec
彩色攝影機	640×480 32-bit 30 frames/sec

所使用的演算法表如下：

表 4.1-2: [11]演算法規格表

演算法	Color space	Bin number	最大疊代次數	最小疊代距離
平均位移演算	nRGB	8*8*8	10	1 pixel
三維平均位移演算	YCbCr	8*8*8	10	10mm

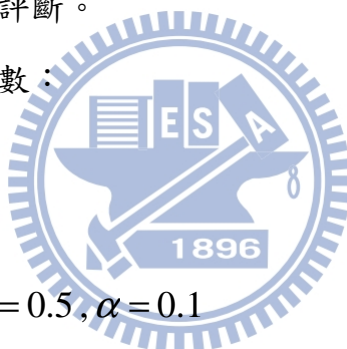
4.2 實驗測試

4.2.1 評估追蹤效果係數和實驗參數

在辨別演算法追蹤的效果，我們總共使用了三種方式。第一個為目標中心與演算法算出來的中心距離差，距離越近表示追蹤效果越好。第二個為目標顏色機率模型與演算法算出來的顏色機率模型的相似度，當相似度越高表示目標框內的顏色分佈越相似。第三個為目標的三維橢圓球的三軸長與演算法算出來的橢圓球的三軸長的相似度，同樣地，當相似度越高時，表示目標框內的三維空間分佈情形越相近。二維的平均位移演算法我們只用第一個方式做評斷，三維的平均位移演算法則使用三個方式來做評斷。

以下為實驗所用各項參數：

1. LTMU : $\alpha = 0.1$
2. NLTMU : $\alpha = 10$
3. SSMU : $\gamma = 0.7, T_b = 0.5, \alpha = 0.1$
4. CCDMU : $\gamma_1 = 0.95, \gamma_2 = 0.8, \varepsilon_1 = 30, \varepsilon_2 = 0.6, \varepsilon_3 = 0.05$
 $\alpha_1 = 0.1, \alpha_2 = 3, \alpha_3 = 7$
5. OTDMU : $T_1 = 0.9, T_2 = 0.999$



4.2.2 基本物體追蹤

實驗圖示的橢圓顏色及其代表的演算法說明：

二維平均位移演算法：NMU(藍)、LTMU(黃)、NLTMU(天藍)、SSMU(紫)。

三維平均位移演算法：NMU(藍)、LTMU(黃)、NLTMU(天藍)、SSMU(紫)、CCDMU(綠)、OTDMU(紅)。另外圖中下方紅色字的 quantity 指的是資料庫中目前已儲存的資料數量，index 則表示當前我們所取出來用的資料編號。

實驗一是追蹤一個長方體盒子。盒子上的顏色分佈主要可分成紅、黃、白三色，運動方式為不規則移動與旋轉。

(1) 使用二維平均位移演算法來進行追蹤，但目標模型的更新方式不同：



圖 4.2-1: 基本物體追蹤一影像輸出結果(2D)

由左至右由上至下分別為 frame 7,27,55,83

由圖 4.2-1 frame 27 可以發現當物體開始旋轉時，因為更新目標模型，將錯誤資訊引進，使得目標追蹤產生漂移，而終致目標丟失。至於沒做目標更新的方式則因白色面在畫面中消失而導致目標追丟。由上述可知，傳統的平均位移演算法即使有做目標模型更新，在複雜背景下仍然無法有效處理，在 frame 83 的時候目標物已經完全追丟。

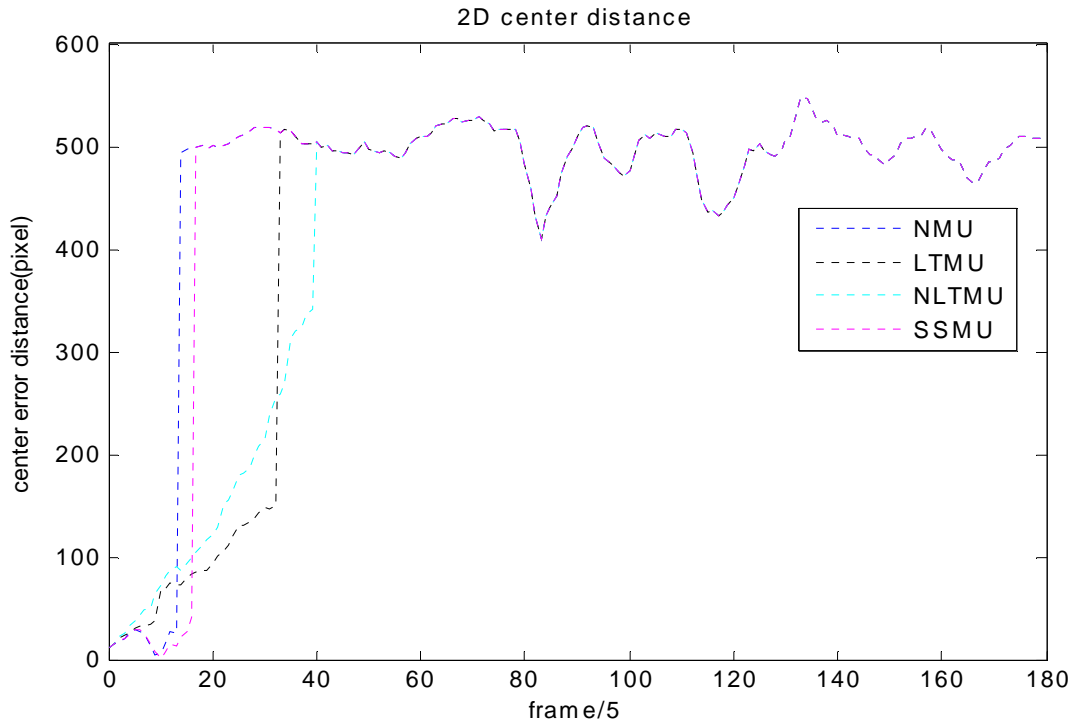


圖 4.2-2: 基本物體追蹤—中心距離誤差(2D)

(2) 使用三維的平均位移演算法來進行追蹤，但目標模型的更新方式不同：





圖 4.2-3: 基本物體追蹤一影像輸出結果(3D)

由左至右由上至下分別為 frame 9,78,119,336,397,448

由 frame 5 可以看出三維平均位移演算法比傳統的二維平均演算法穩健性較高，在 frame 78 時，因白色面在畫面中小到一定程度，NMU 就會因為目標過小而追蹤失敗。而在 frame 119、336 與 397 則分別出現 LTMU、SSMU、NLTMU 產生追蹤出錯，原因為一但將過多的錯誤資訊拿來做更新，則會導致追蹤到錯誤的結果，如此一來就會繼續地把錯誤的資訊選取進來做更新，導致最後追蹤失敗。而 CCDMU 與 OTDMU 這兩種方式則一直很穩定與準確地追蹤在正確物體上，然而在 frame 448 我們可以發現，此時綠色橢圓與紅色橢圓並無準確地將目標物框住，經過分析發現由於日光燈的影響，導致目標物上下兩部分的光影變化不同，導致抓到的目標物中心點也不同，再加上目標物在空間中的分佈情形為上部分較遠，下部分較近，使得在判斷目標物的深度範圍時會因為不一樣的位置而框住不一樣的範圍，導致主成份分析在計算目標物於空間中的分佈狀況時出現不準確的問題，使得橢圓產生縮小且無法準確框住目標物的情形。

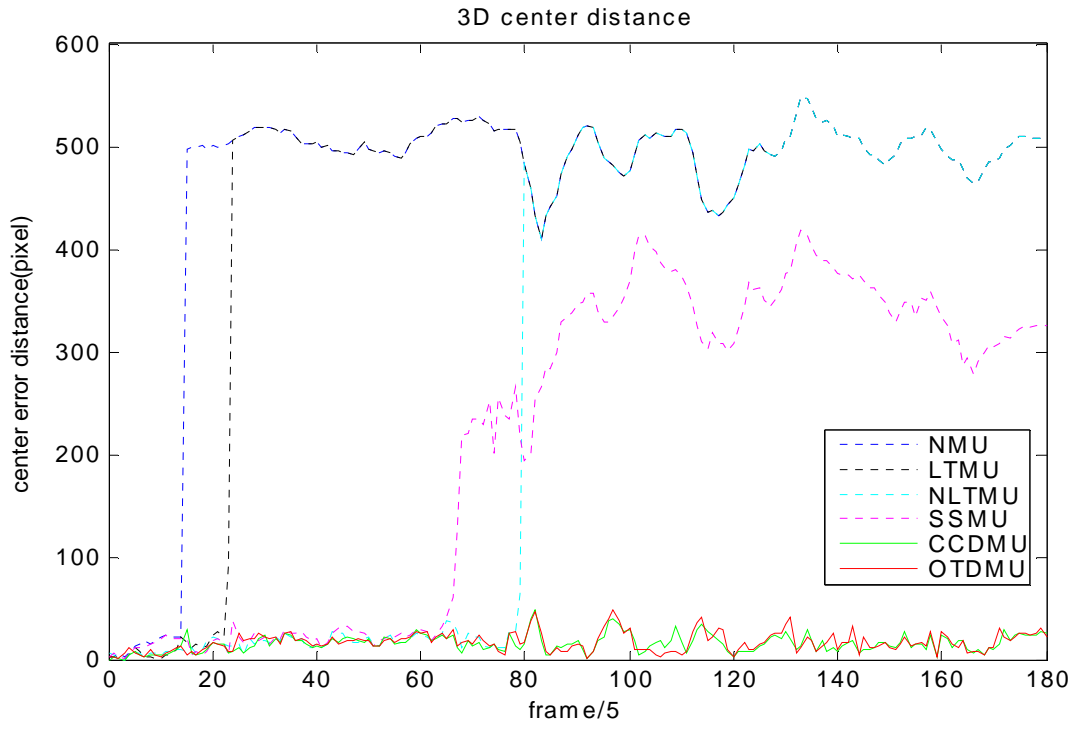


圖 4.2-4: 基本物體追蹤—中心距離誤差(3D)

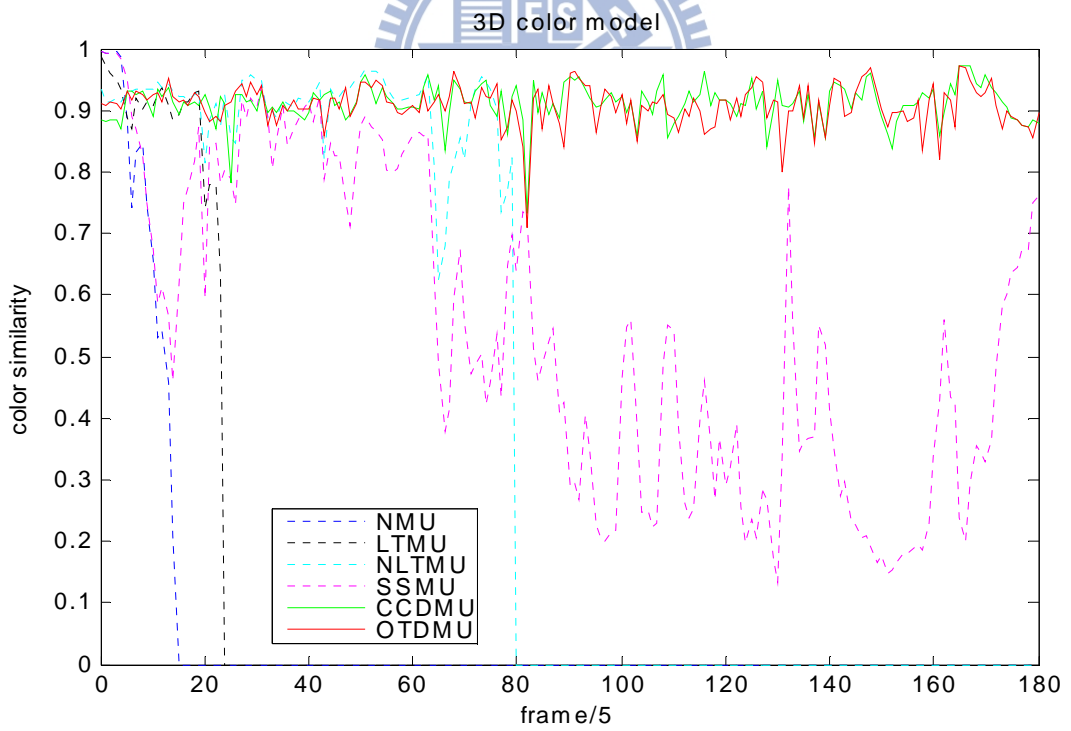


圖 4.2-5: 基本物體追蹤—目標顏色相似度

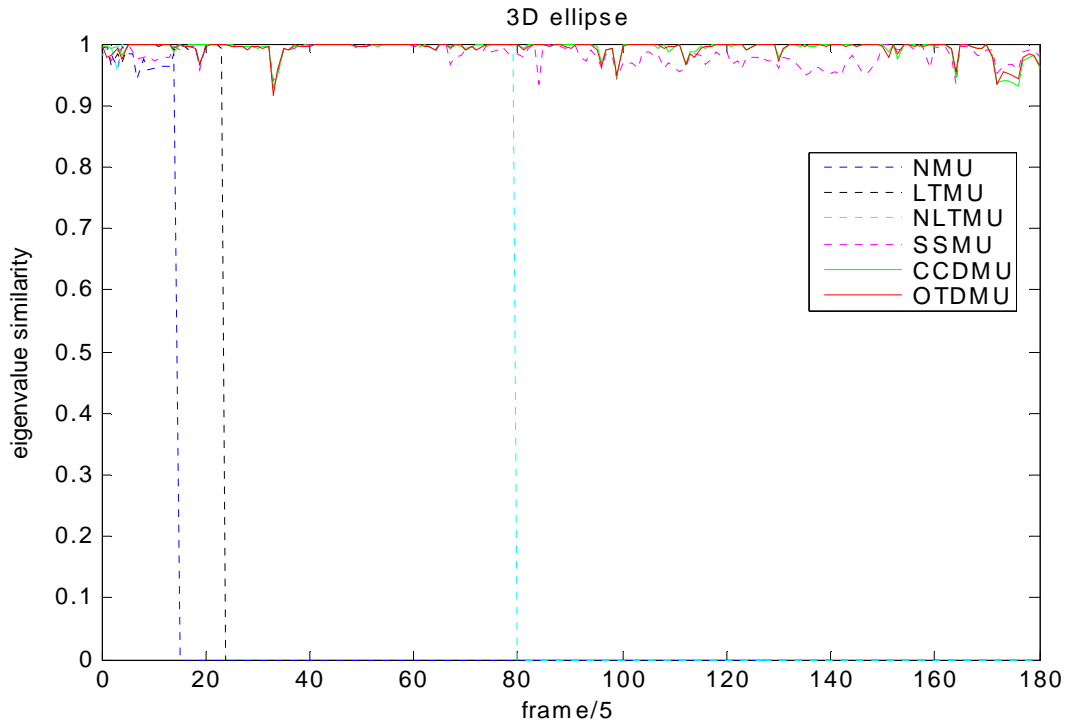


圖 4.2-6: 基本物體追蹤一目標空間分佈相似度

表 4.2-1: 基本物體追蹤一數值結果(2D)

2D Mean-Shift	平均處理時間(sec)	平均中央誤差(pixel)
NMU	0.00068941	461.56
LTMU	0.00079173	421.80
NLTMU	0.0014204	419.52
SSMU	0.0012404	453.63

表 4.2-2: 基本物體追蹤一數值結果(3D)

3D Mean-Shift	平均處理時間 (sec)	平均中央誤差 (pixel)	平均顏色相似度	平均空間分佈 相似度
NMU	0.0053073	458.54	0.063043	0.080728
LTMU	0.0075695	434.28	0.11812	0.13158
NLTMU	0.0084007	282.28	0.39961	0.43956
SSMU	0.010482	214.11	0.54419	0.98254
CCDMU	0.0081236	15.637	0.9117	0.99281
OTDMU	0.011121	16.721	0.90848	0.99267

實驗二為追蹤一個基本的人型娃娃。娃娃主要可分成兩色，分別是正面白色，背面藍色。此實驗在測試目標物非平滑的平面對追蹤演算法的影響。

(1) 二維平均位移演算法：

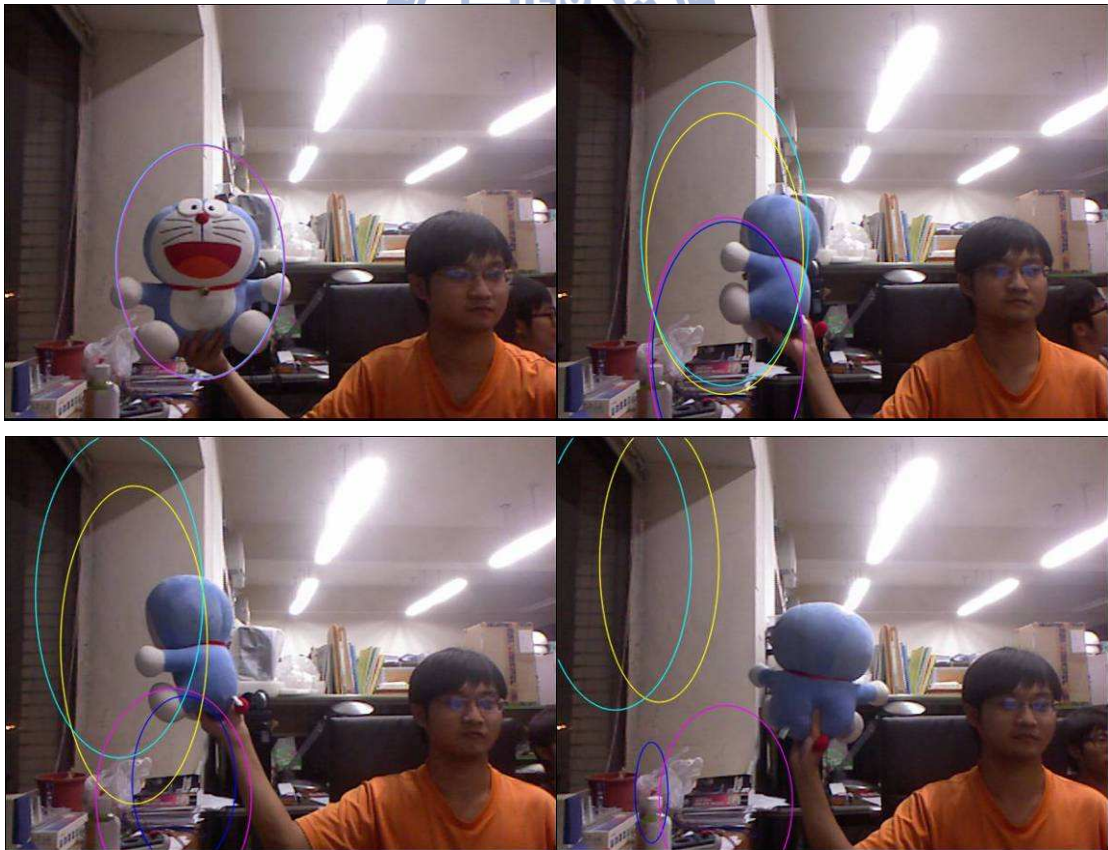


圖 4.2-7: 基本物體追蹤二影像輸出結果(2D)

由左至右由上至下分別為 frame 2,47,94,179

一開始物體為正面，所以白色成份居多，一開始旋轉時，仍可有效地抓住目標，然而 NMU 與 SSMU 各自的橢圓於 frame 47 的時候而下移，原因為 NMU 並無做更新，所以會以手腳的白色為主要目標而下移，而 SSMU 則是做選擇性地更新，其更新以手腳及其附近的顏色資訊為主要目標，最後被下方的背景顏色拉走。LTMU 與 NLTMU 則在更新過程中，因手腳顏色與牆壁相同，將牆壁的顏色資訊引入做更新，最後導致引入過多錯誤資訊而於 frame 94 追蹤失敗。

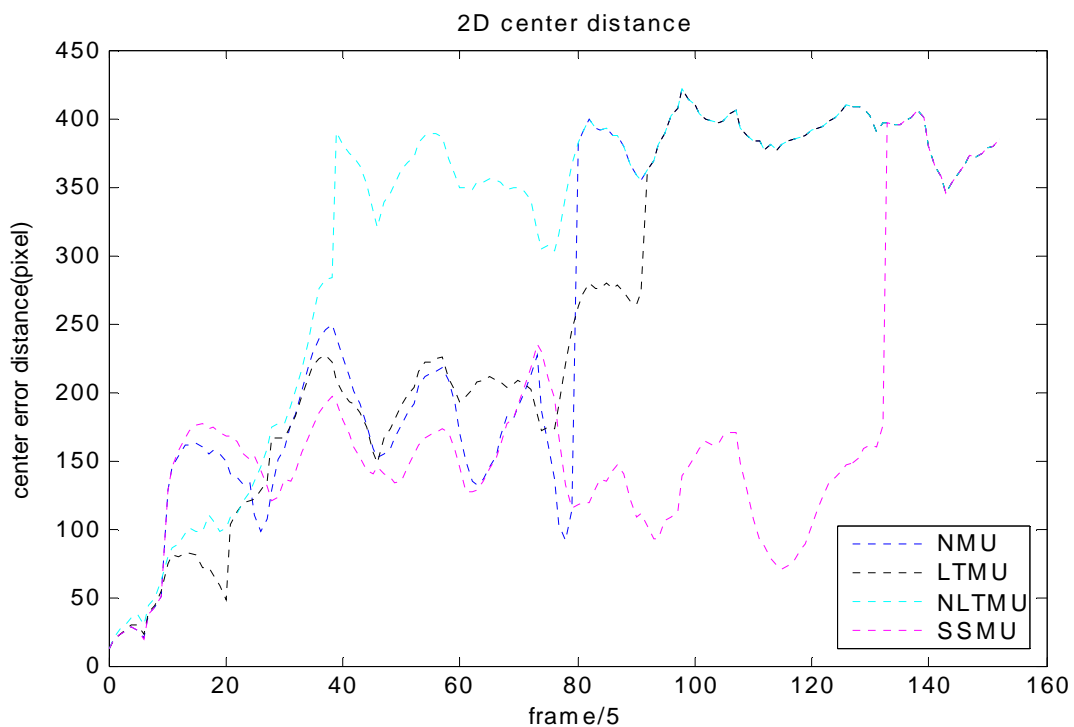


圖 4.2-8: 基本物體追蹤二中心距離誤差(2D)

(2) 三維平均位移演算法：





圖 4.2-9: 基本物體追蹤二影像輸出結果(3D)

由左至右由上至下分別為 frame 8,109,145,253,386,449

NMU 會因正反兩面的顏色成份不同而於 frame 109 追丟。其他更新方式與 CCDMU 比較起來並無顯著不同，一樣都能追到目標物。值得一提的是 OTDMU，一旦因為目標物旋轉造成的顏色分佈改變導致偏移，就可以藉由資料庫內建過去的資料模型的幫助，重新迅速準確地框住整個目標物，而不像其他的更新方法，框住的目標範圍通常比較大，表示在物體旋轉的過程中，一直有引入些許非目標物的顏色資訊來做更新，因此也包括較多非目標物的資訊，如 frame 109、145、253 與 449 所示。

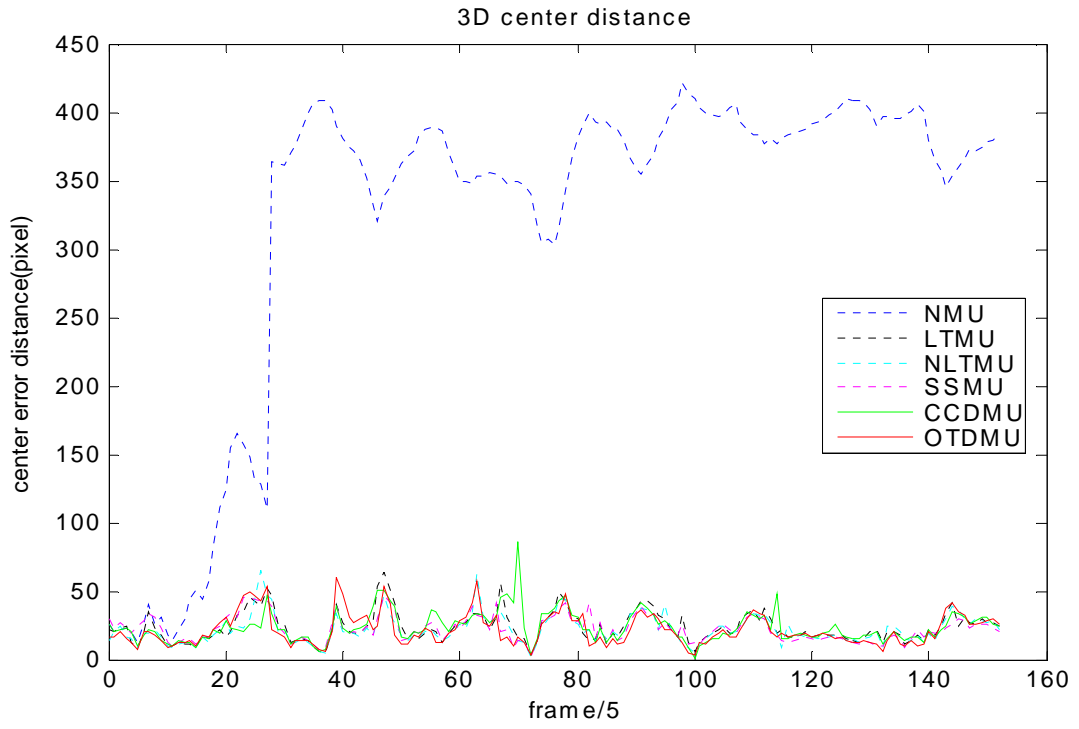


圖 4.2-10: 基本物體追蹤二中心距離誤差(3D)

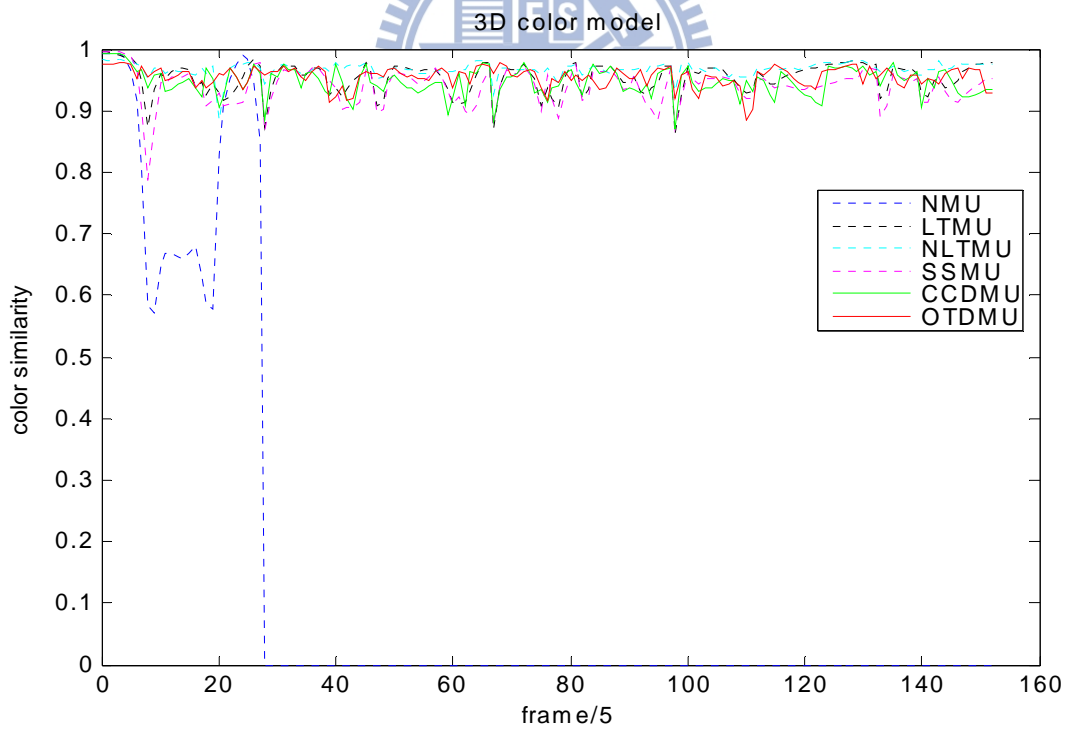


圖 4.2-11: 基本物體追蹤二目標顏色相似度

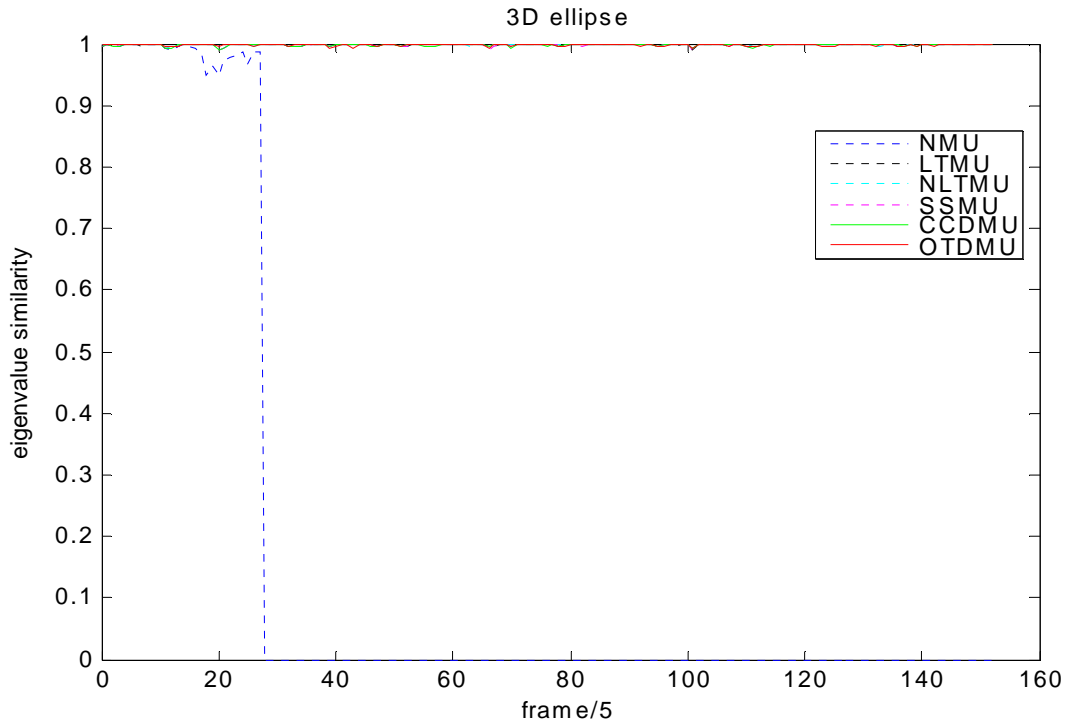


圖 4.2-12: 基本物體追蹤二目標空間分佈相似度

表 4.2-3: 基本物體追蹤二數值結果(2D)

2D Mean-Shift	平均處理時間(sec)	平均中央誤差(pixel)
NMU	0.0015843	265.15
LTMU	0.0039118	256.4
NLTMU	0.0068614	311.22
SSMU	0.0025745	168.85

表 4.2-4: 基本物體追蹤二數值結果(3D)

3D Mean-Shift	平均處理時間 (sec)	平均中央誤差 (pixel)	平均顏色相似度	平均空間分佈 相似度
NMU	0.0087249	319.72	0.14824	0.18068
LTMU	0.011266	23.74	0.95382	0.9984
NLTMU	0.010994	21.924	0.96747	0.99843
SSMU	0.011213	22.478	0.93967	0.99818
CCDMU	0.010911	23.728	0.94707	0.99824
OTDMU	0.013829	21.731	0.95518	0.99818

實驗三為追蹤一個三色的保特瓶，顏色分佈為紅、白、黃，目的是測試當目標物在空間中快速移動時，各演算法的效果如何。

(1) 二維平均位移演算法：

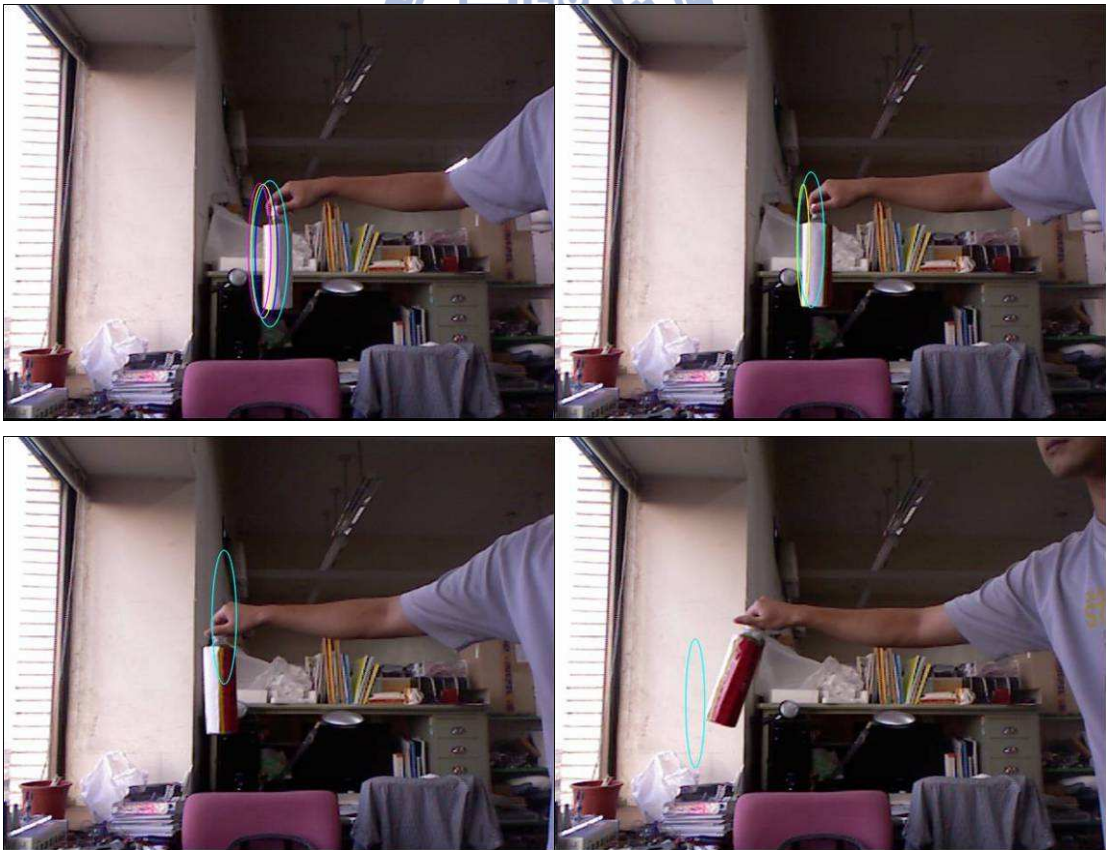


圖 4.2-13: 基本物體追蹤三影像輸出結果(2D)

由左至右由上至下分別為 frame 16,28,36,62

在 frame 16 的時候，被同樣為白色的背景顏色所影響，使得四個方法皆開始往左偏移，並且因為持續地用錯誤的顏色資訊做更新，因此於 frame 28 和 36，SSMU 和 LTMU 追蹤失敗，而 NLTMU 也因為同為白色的牆壁顏色影響，而追錯目標。

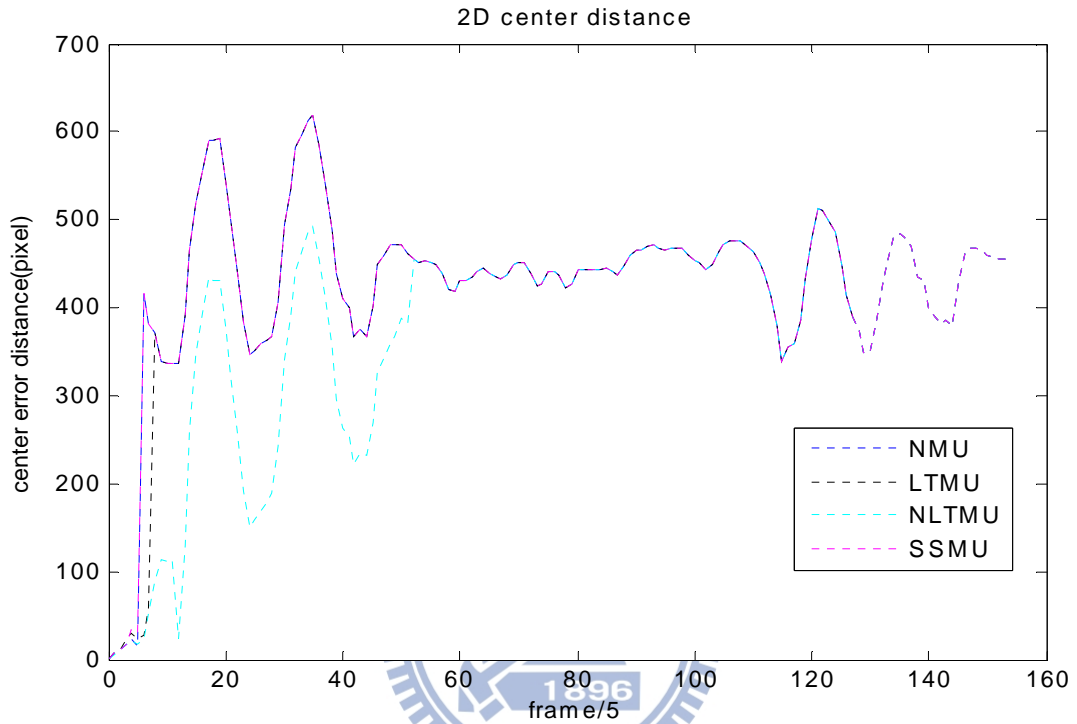


圖 4.2-14: 基本物體追蹤三中心距離誤差(2D)

(2) 三維平均位移演算法：



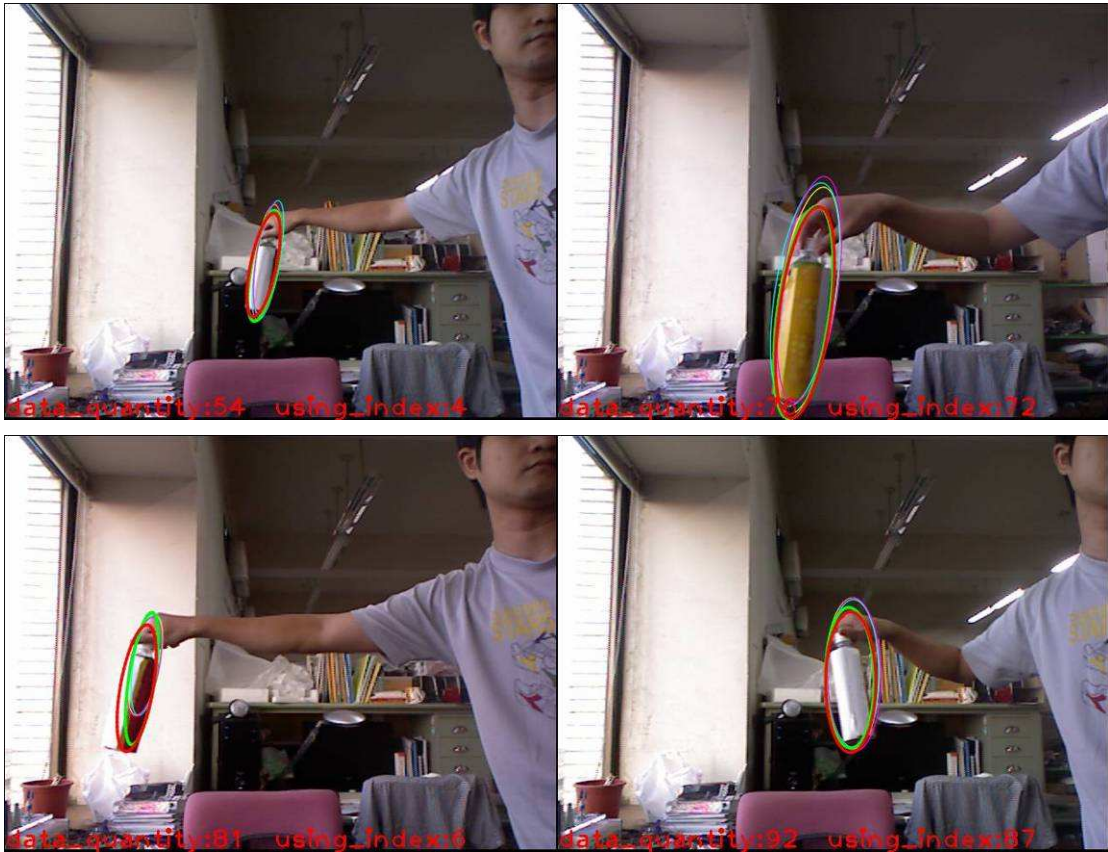


圖 4.2-15: 基本物體追蹤三影像輸出結果(3D)

由左至右由上至下分別為 frame 64,83,330,468,579,666

NMU 因目標物顏色改變而於 frame 83 時追丟。其餘方法至始至終皆能抓住目標物，其中 CCDMU 於目標物為近似於 uv 平面上的移動時，我們有將其更新速度做降速的動作，因此相較起來，其框住目標物的橢圓會有偏移的情況，如 frame 64 所示，但對於目標物為深度方向的移動方式時，我們對其更新速度有做進一步加速的方式，因此在 frame 330 和 468 有表現出其因更新速度較快而準確性較高的優勢，如圖 4.2-17 所示，CCDMU 於圖的中央部分為深度方向的移動時，其顏色相似度與 ground true 較相似。OTDMU 則因為有資料庫內過去的資料模型幫助，因此對於快速移動時的追蹤偏移能夠有較大的效果幫忙其導正，如 frame 330 和 579 所示。

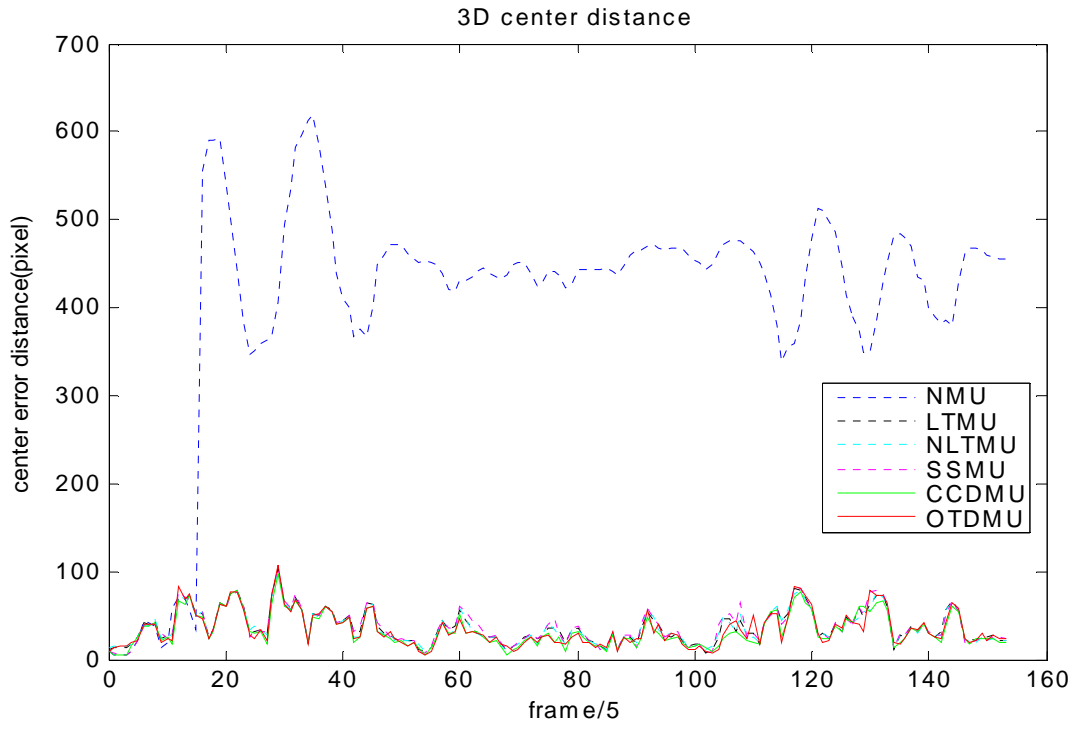


圖 4.2-16: 基本物體追蹤三中心距離誤差(3D)

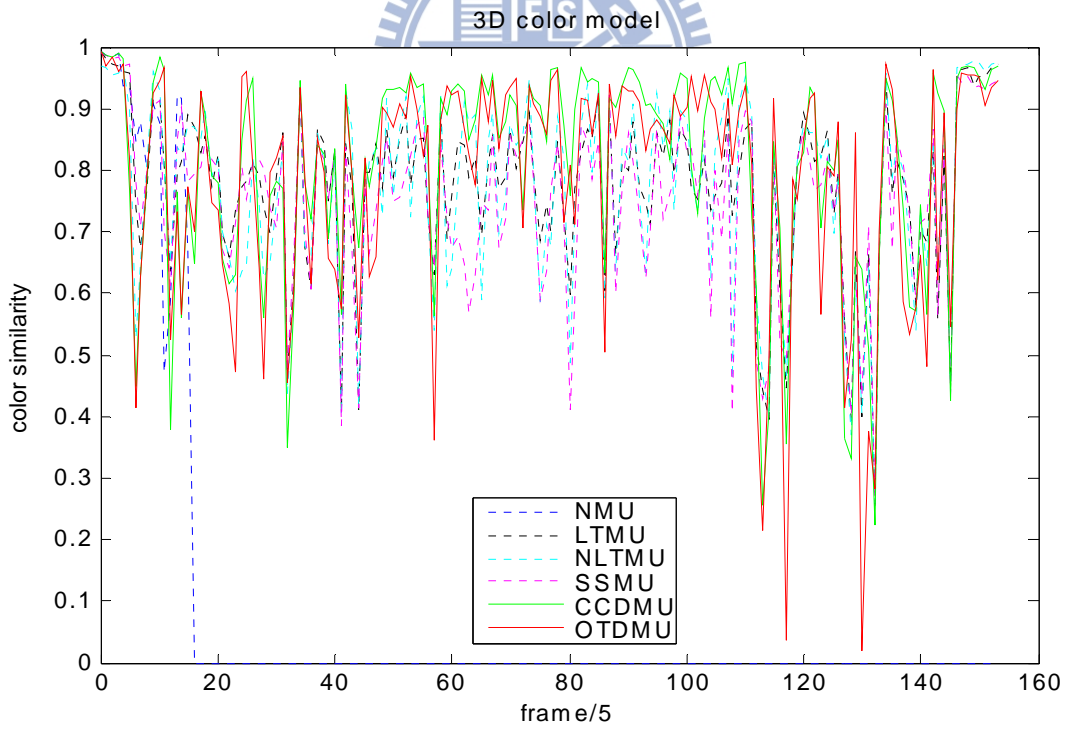


圖 4.2-17: 基本物體追蹤三目標顏色相似度

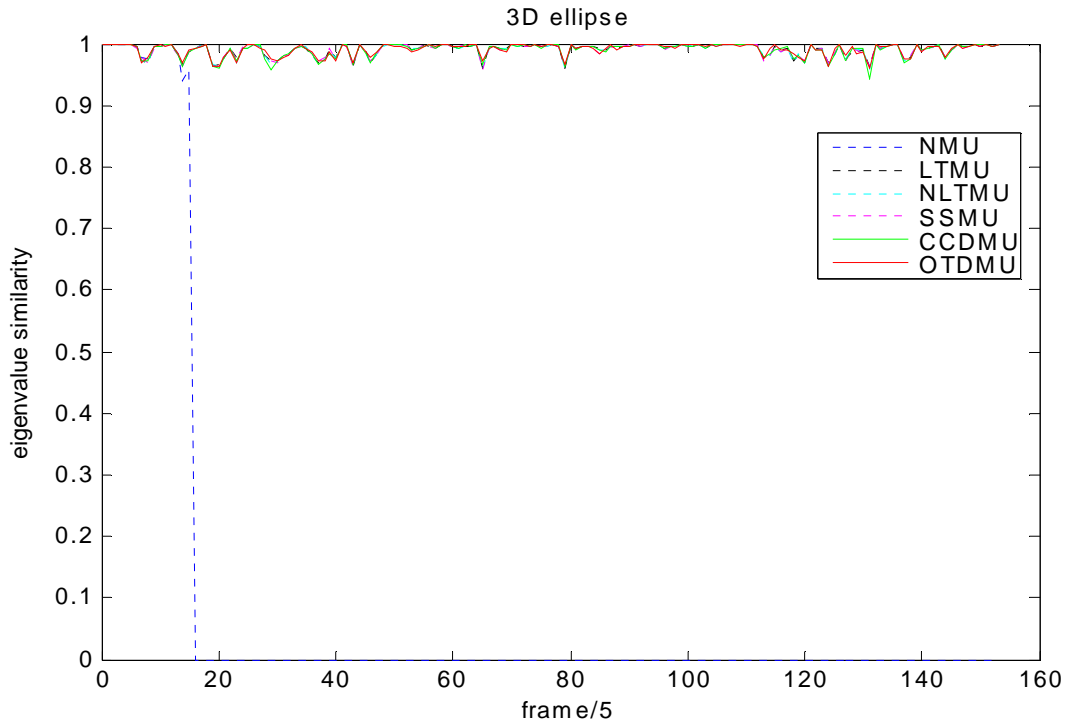


圖 4.2-18: 基本物體追蹤三目標空間分佈相似度

表 4.2-5: 基本物體追蹤三數值結果(2D)

2D Mean-Shift	平均處理時間(sec)	平均中央誤差(pixel)
NMU	0.00054946	428.17
LTMU	0.0010555	423.66
NLTMU	0.00069649	376.68
SSMU	0.0014047	428.33

表 4.2-6: 基本物體追蹤三數值結果(3D)

3D Mean-Shift	平均處理時間 (sec)	平均中央誤差 (pixel)	平均顏色相似度	平均空間分佈 相似度
NMU	0.0067702	405.59	0.089047	0.10272
LTMU	0.0077024	36.932	0.77401	0.99088
NLTMU	0.0077892	36.458	0.77129	0.9908
SSMU	0.0078471	37.721	0.75303	0.99074
CCDMU	0.0080841	33.053	0.81342	0.99052
OTDMU	0.010751	34.505	0.78779	0.99131

實驗四為追蹤一個六色的六面體。顏色分別為白、藍、黃、黑、綠、紅。此實驗目的為測試線上模板資料庫的建構對追蹤效能的影響，我們先行旋轉目標物，使其各個面向能讓 OTDMU 建立線上資料庫供之後快速旋轉使用。

(1) 二維平均位移演算法：

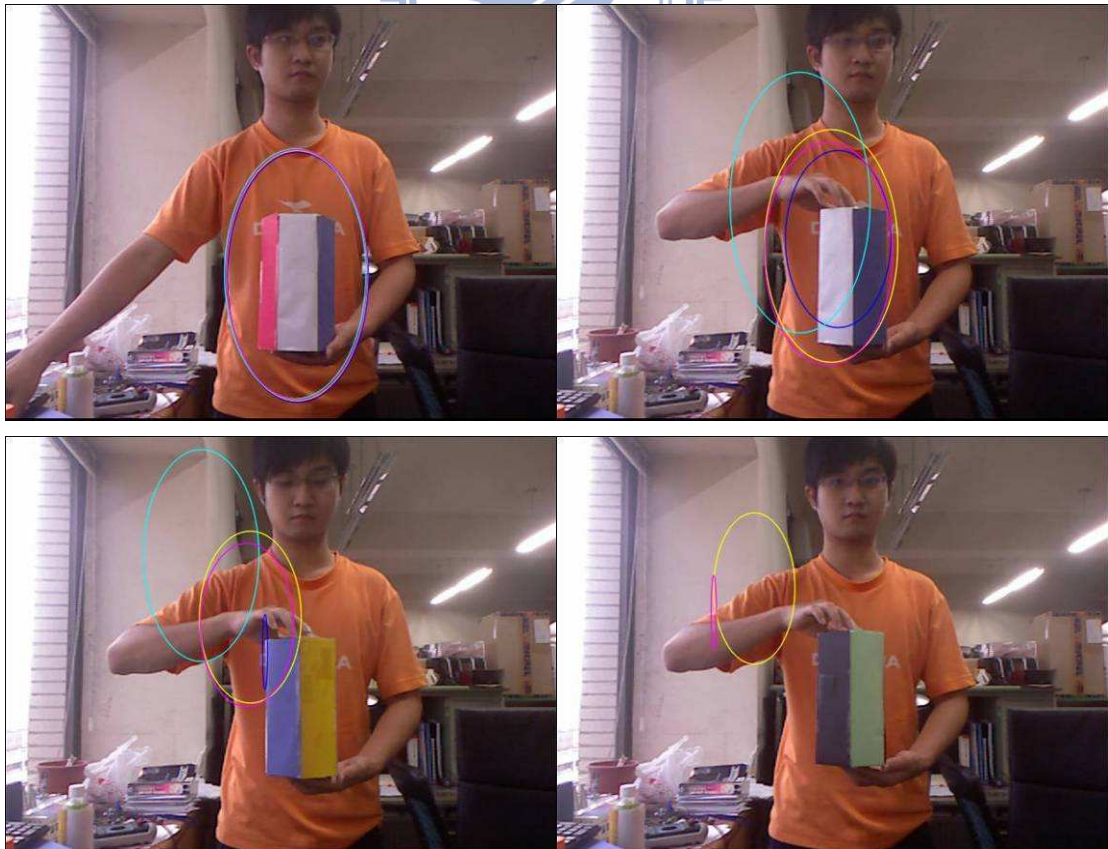


圖 4.2-19: 基本物體追蹤四影像輸出結果(2D)

由左至右由上至下分別為 frame 10,88,212,439

目標物一開始有被準確追到，但在 frame 88 時，NLTMU 將手的顏色資訊拿來更新，因此目標向手部偏移。而 frame 212 時，因白色面於畫面中消失，所以 NMU 追丟，另兩種方法也因將手部顏色資訊拿來做目標更新，因此向手部偏移。

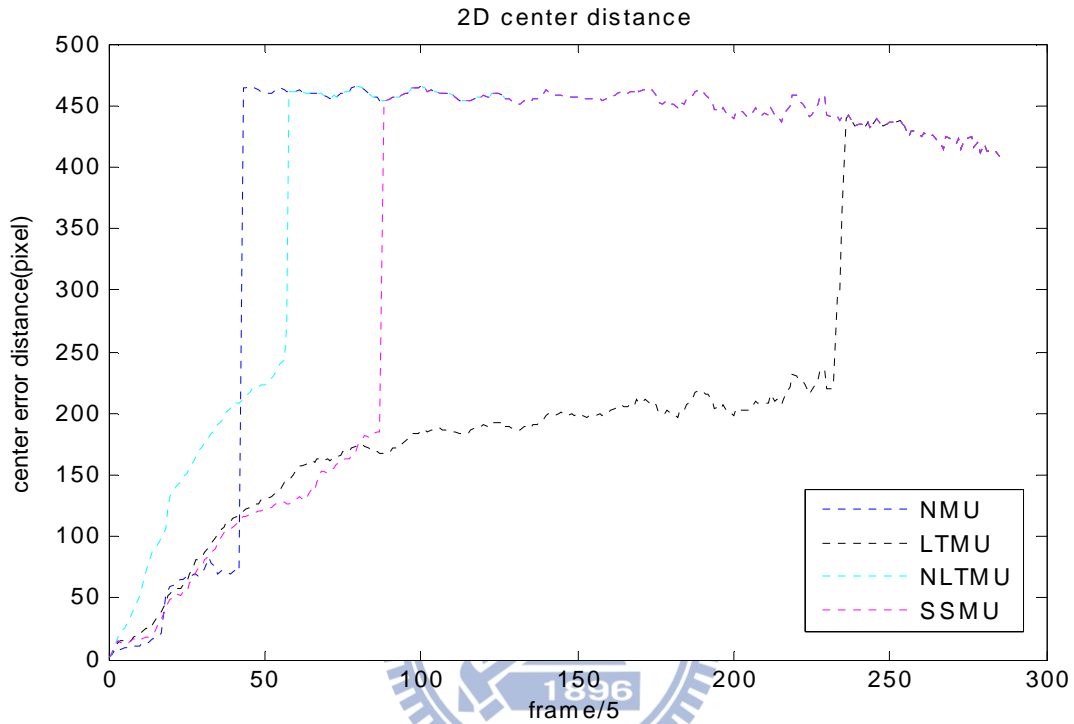


圖 4.2-20: 基本物體追蹤四中心距離誤差(2D)

(2) 三維平均位移演算法：

以下我們先放上這次實驗中所建立出來的線上模型資料庫中的數個。圖中左下角的 color 指的是顏色相似度，space 指的是空間分佈相似度，上述兩個數值是用來決定是否新增資料進資料庫，判斷標準是當顏色相似度小於等於 0.9 或空間分佈相似度小於等於 0.999。quantity 指的是資料庫中資料的數量，index 則表示當前我們所取出來用的資料編號。以下為資料庫內編號 10~21 的資料模型。



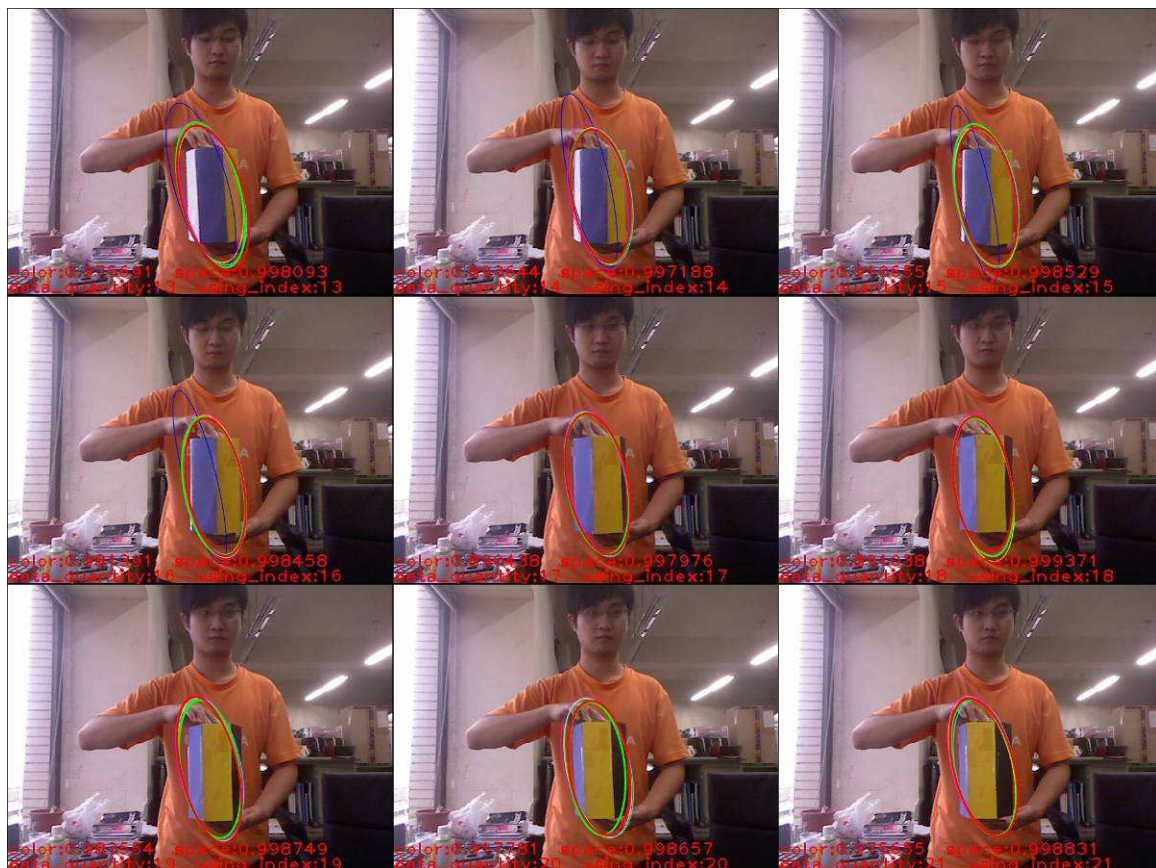


圖 4.2-21: OTDMU-資料庫內編號 10~21 的資料

由左至右由上至下分別為編號 10~21，frame 123,137,157,169,188,200,204,246,258,269,290,297





圖 4.2-22: 基本物體追蹤四影像輸出結果(3D)

由左至右由上至下分別為 frame 208,620,895,1001,1149,1420

一開始緩慢旋轉時，只有 NMU 於 frame 208 因白色消失而追丟，其餘方法皆很穩定。轉完一圈後，之後開始快速旋轉，SSMU 很快地於 frame 895 因更新速度不及而導致追錯目標，NLTMU 也於 frame 1001 時因物體快速旋轉導致其更新速度不夠快而追錯目標。剩餘三種方法皆有追到正確目標，但 CCDMU 比起 LTMU 更新的速度較快，因此對於顏色分佈的快速變化有更好的適應性，如 frame 1149 和 1420 所示。OTDMU 則藉由資料庫的幫助，其表現最佳。

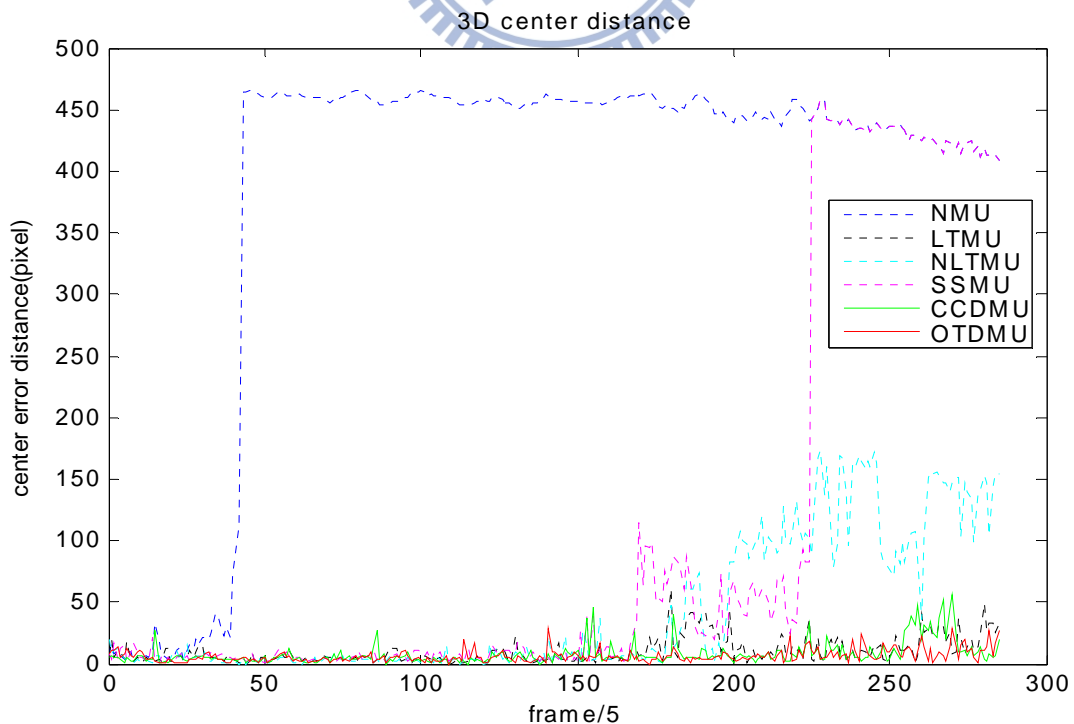


圖 4.2-23: 基本物體追蹤四中心距離誤差(3D)

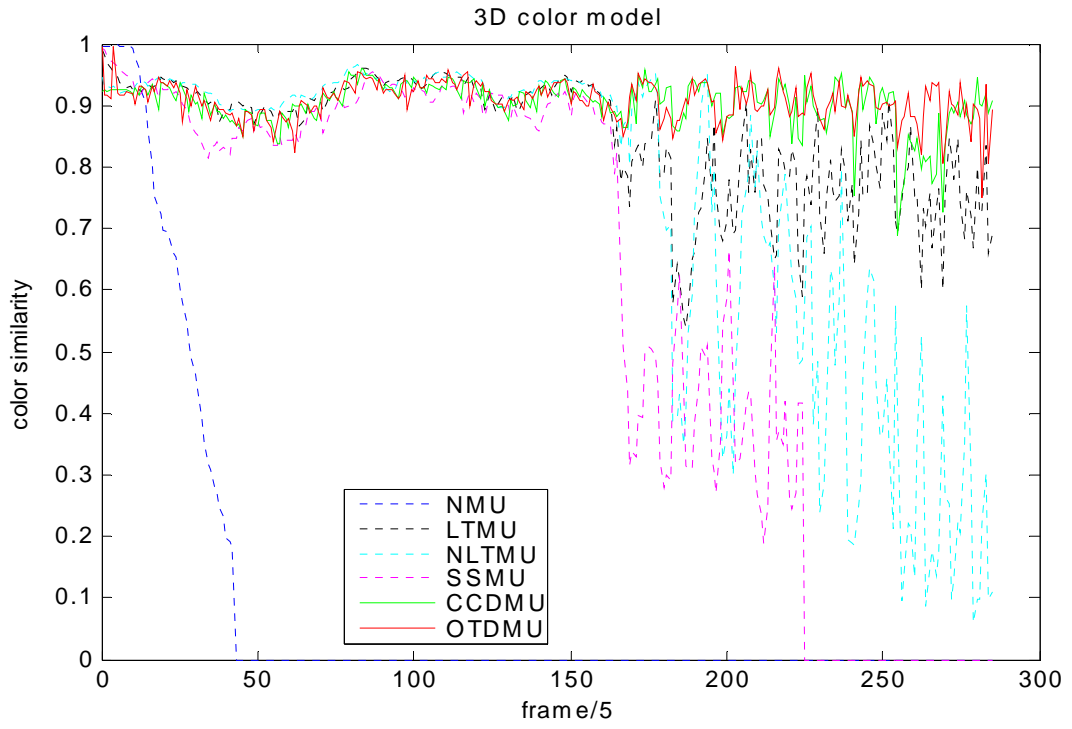


圖 4.2-24: 基本物體追蹤四目標顏色相似度

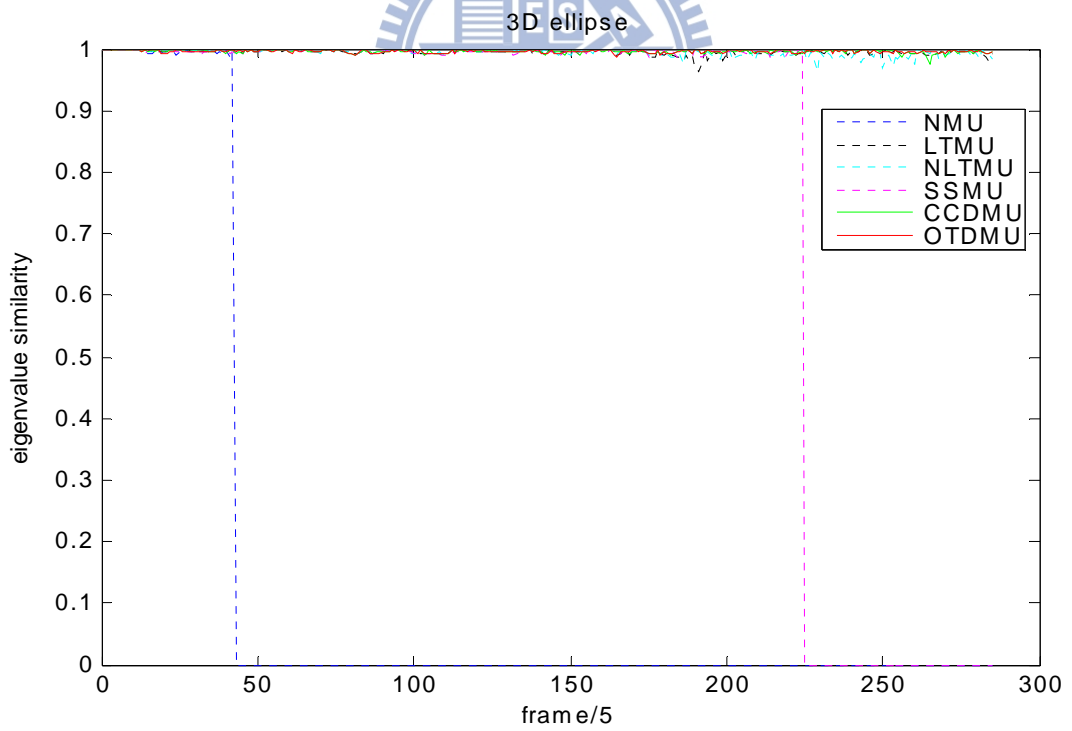


圖 4.2-25: 基本物體追蹤四目標空間分佈相似度

表 4.2-7: 基本物體追蹤四數值結果(2D)

2D Mean-Shift	平均處理時間(sec)	平均中央誤差(pixel)
NMU	0.0019572	388.81
LTMU	0.002175	212.03
NLTMU	0.0041851	387.74
SSMU	0.0026059	340.19

表 4.2-8: 基本物體追蹤四數值結果(3D)

3D Mean-Shift	平均處理時間 (sec)	平均中央誤差 (pixel)	平均顏色相似度	平均空間分佈 相似度
NMU	0.009969	385.08	0.099978	0.14978
LTMU	0.010476	10.107	0.85441	0.99496
NLTMU	0.010811	41.933	0.75325	0.99406
SSMU	0.012427	106.89	0.60354	0.78333
CCDMU	0.010405	7.8922	0.90378	0.99607
OTDMU	0.012912	6.2606	0.90838	0.99606

4.2.3 人體追蹤

實驗圖示的橢圓顏色及其代表的演算法說明：

二維平均位移演算法：NMU(藍)、LTMU(黃)、NLTMU(天藍)、SSMU(紫)。

三維平均位移演算法：NMU(藍)、LTMU(黃)、NLTMU(天藍)、SSMU(紫)、CCDMU(綠)、OTDMU(紅)。另外圖中下方紅色字的 quantity 指的是資料庫中目前已儲存的資料數量，index 則表示當前我們所取出來用的資料編號。

實驗一利用追蹤演算法來追蹤人體。人的移動方式為平移旋轉皆有。

(1) 二維平均位移演算法：



圖 4.2-26: 人體追蹤一影像輸出結果(2D)

由左至右由上至下分別為 frame 1,9,56,78

由於灰色衣服與背景顏色過於相似，再加上左側的光亮影響，因此橢圓皆往左側光亮處移動，很快地，四種方法皆追丟目標。

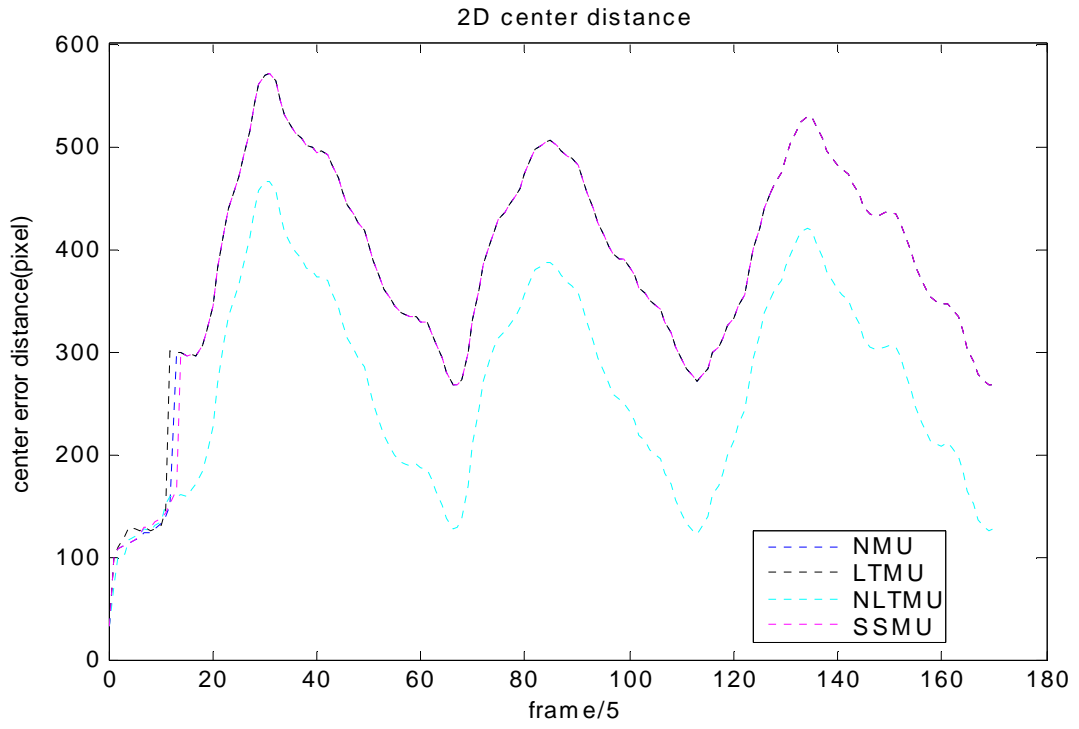


圖 4.2-27: 人體追蹤—中心距離誤差(2D)

(2) 三維平均位移演算法：





圖 4.2-28: 人體追蹤一影像輸出結果(3D)

由左至右由上至下分別為 frame 94,189,368,432,602,677

由於三維的平均位移演算法能夠有效地分離前景與背景，因此各種方法都能追得到目標物。由上述各 frame 可以看出，當追蹤人體的時候，由於衣服的颜色通常為單一色系，在此情況下，NMU 的表現會是最好的，然而人體在旋轉及移動的時候，會有包括手部與褲子的颜色資訊皆被選取做目標模型更新，這會使得除了 NMU 之外的橢圓皆有放大及些許偏移的情況。不過在各種有進行目標模型更新的方法中，可發現 OTDMU 的線上資料庫有助於解決些許目標模型偏移的問題，這是因為利用資料庫，總能夠找到最相近的目標模型來繼續追蹤，而不是像其他方式一樣，會一直將所得到的資訊拿來做目標更新，而不考慮是否是目標本身的颜色。

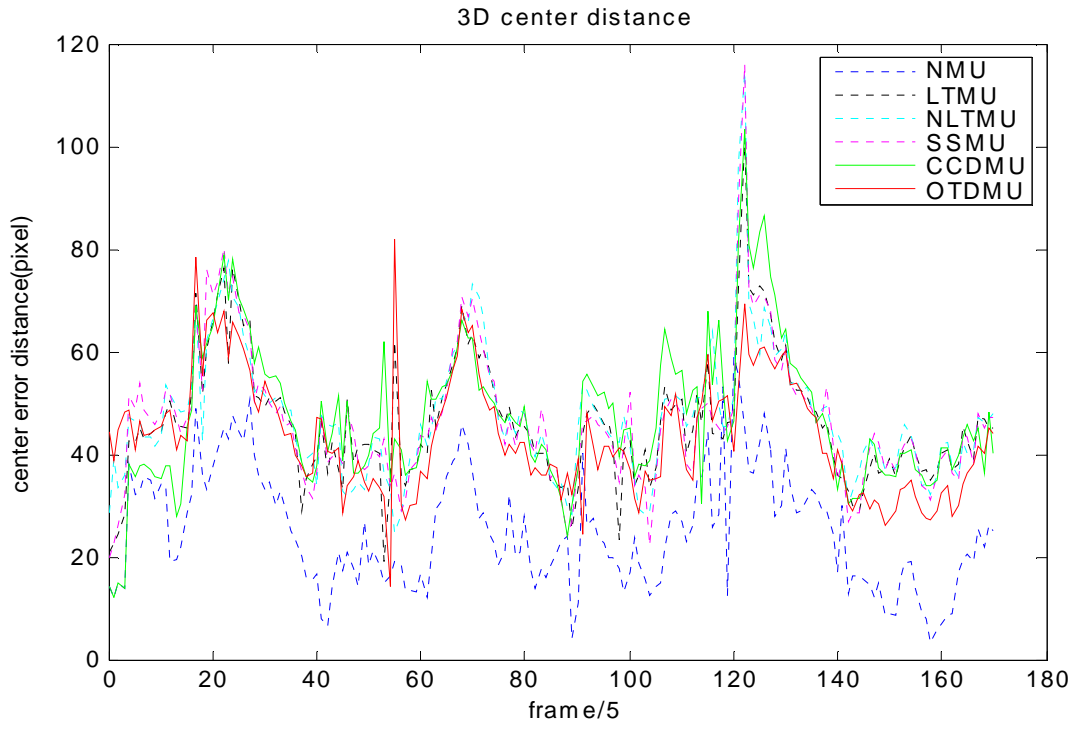


圖 4.2-29: 人體追蹤—中心距離誤差(3D)

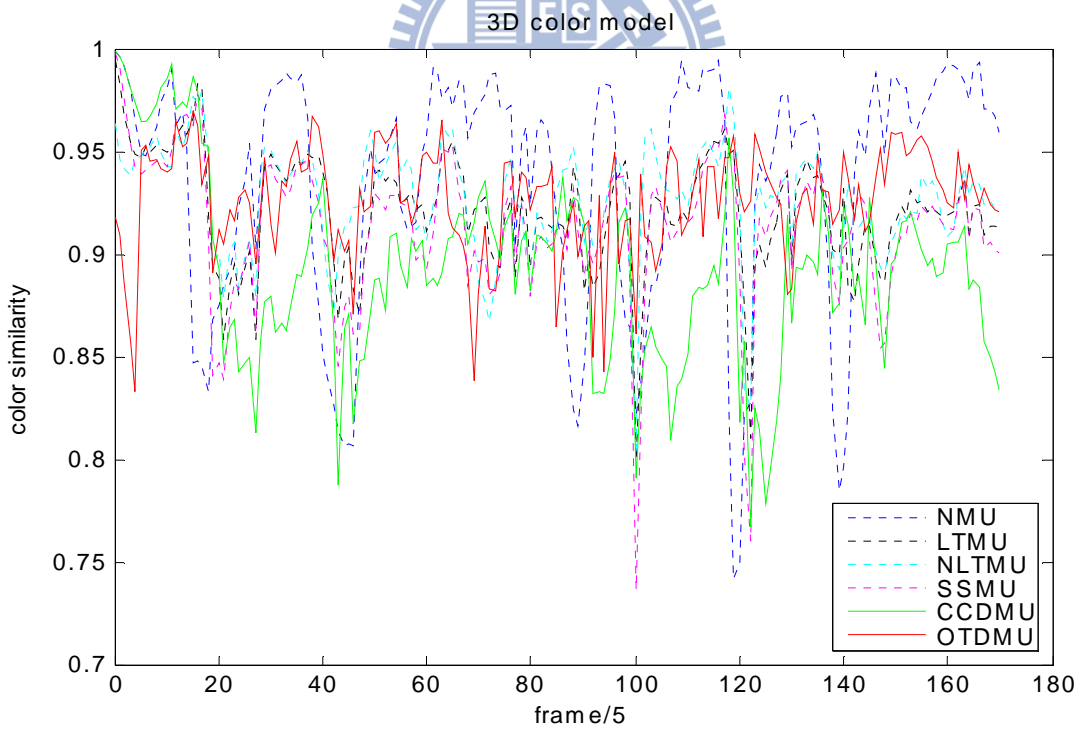


圖 4.2-30: 人體追蹤—目標顏色相似度

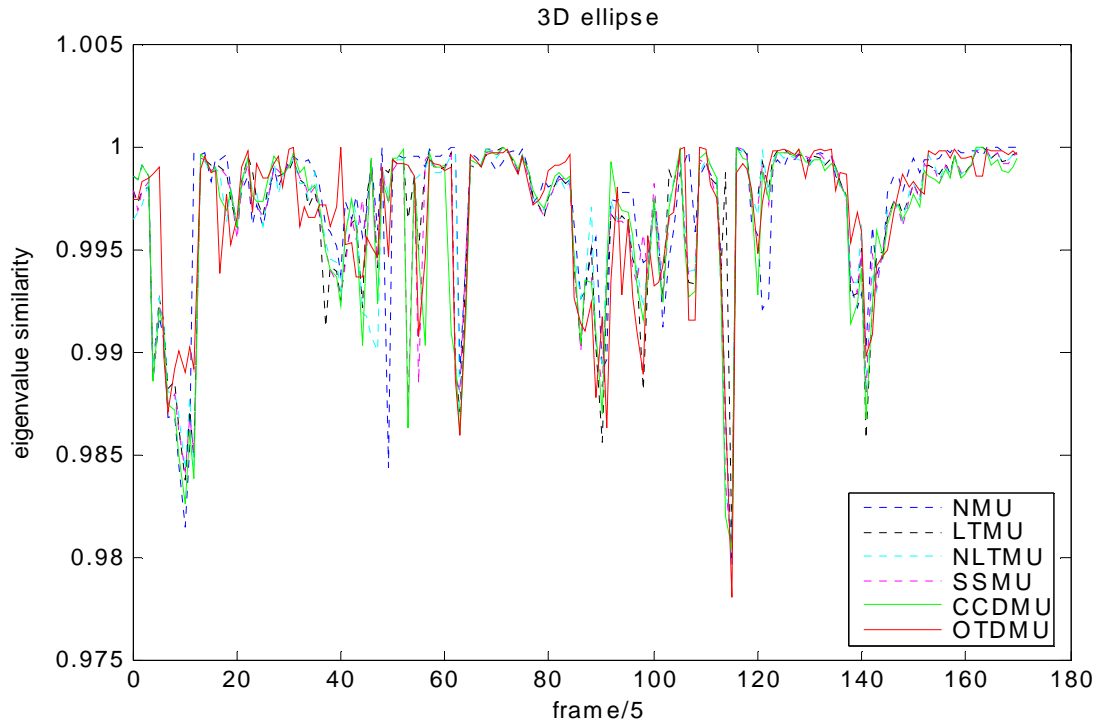


圖 4.2-31: 人體追蹤一目標空間分佈相似度

表 4.2-9: 人體追蹤一數值結果(2D)

2D Mean-Shift	平均處理時間(sec)	平均中央誤差(pixel)
NMU	0.012949	382.26
LTMU	0.012411	383.46
NLTMU	0.0046156	264.7
SSMU	0.013705	381.67

表 4.2-10: 人體追蹤一數值結果(3D)

3D Mean-Shift	平均處理時間 (sec)	平均中央誤差 (pixel)	平均顏色相似度	平均空間分佈 相似度
NMU	0.013723	24.915	0.93498	0.99707
LTMU	0.015813	46.155	0.92139	0.99669
NLTMU	0.015841	47.369	0.92678	0.9967
SSMU	0.016141	46.609	0.91375	0.99654
CCDMU	0.015959	47.384	0.89238	0.99635
OTDMU	0.017862	43.401	0.92634	0.9967

實驗二同樣利用追蹤演算法來追蹤人體，但在畫面中有擺放椅子當障礙物。

(1) 二維平均位移演算法：

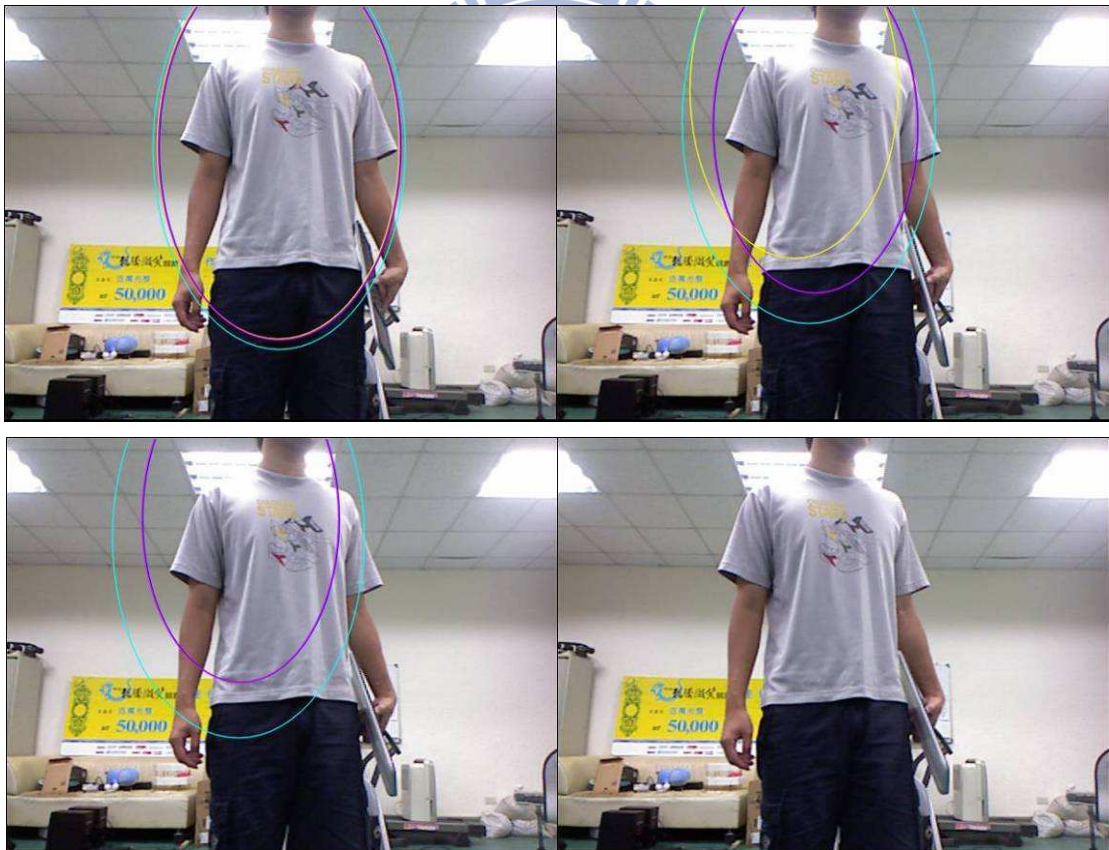


圖 4.2-32: 人體追蹤二影像輸出結果(2D)

由左至右由上至下分別為 frame 3,14,17,19

由於灰色衣服與背景顏色過於相似，且左上角受亮光影響，因此四種方法皆

往左上天花板的亮光處移動，很快地，四種方法皆發散。

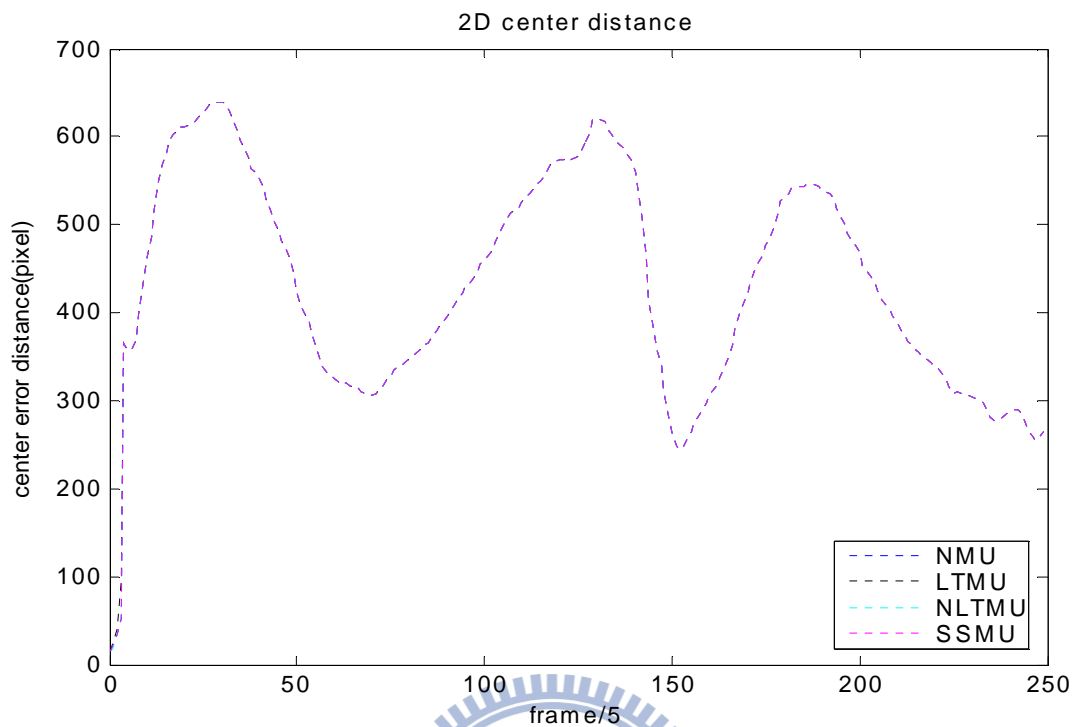


圖 4.2-33: 人體追蹤二中心距離誤差(2D)

(2) 三維平均位移演算法：

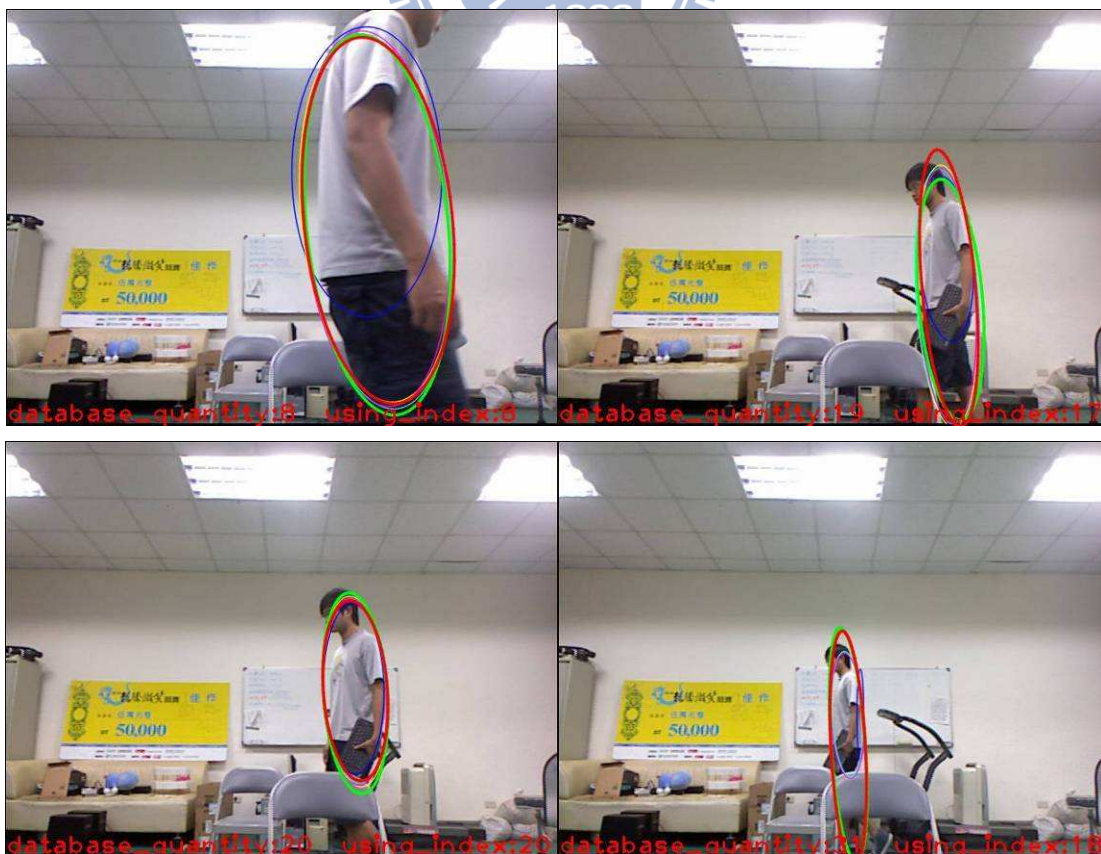




圖 4.2-34: 人體追蹤二影像輸出結果(3D)

由左至右由上至下分別為 frame 52,210,224,1007,1024,1057

一開始我們所選定的目標範圍只有衣服的灰色部份，不包括手臂及褲子的顏色，在 frame 52 的時候，除了 NMU 之外的方法皆有做目標更新，所以把褲子的顏色也更新進去，因此所圈的橢圓明顯比 NMU 的範圍來得大。從 frame 210 到 frame 224，可以發現藉由深度資訊的不同，可以區分出屬於目標前景與背景的資訊。在 frame 1007 時，NMU 受到手臂顏色的阻撓，無法將整個目標範圍框住，使得其判斷目標的中心點移到背部，導致目標範圍過小而消失，在 frame 1024 也發生一樣的事，CCDMU 也因圈選的目標中心移到側身的手臂與身體交界處，無法完整框住目標，最終導致範圍過小而消失。另外在 frame 1007 我們也可發現 CCDMU 和 OTDMU 在框住目標物時，連不同深度的椅背部分也圈選進去，這是因為這兩個方法仍然有將腳的顏色資訊拿進來做更新，由於腳的深度和身體的深度是一樣的，因此在決定目標深度時身體和腳都會包括在裡面，因此位於兩者之間的椅背部分就同時被圈選起來了。在 frame 1057 則是另三種更新方法藉由主成份分析所計算出來的三維橢圓球的三軸長在投影到 uv 平面之後，導致二維橢圓的寬度為 0 而追蹤失敗。最後只剩 OTDMU 藉由線上模板資料庫的幫助，可以準確地抓住目標物，其表現為所有方法中最佳。

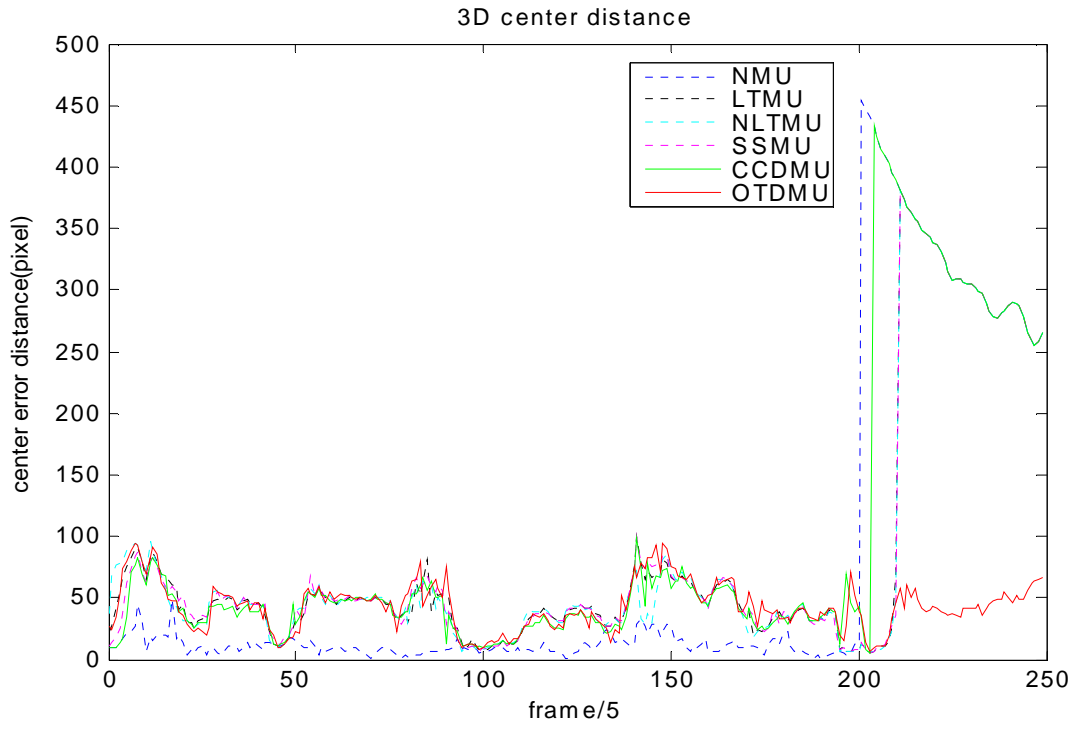


圖 4.2-35: 人體追蹤二中心距離誤差(3D)

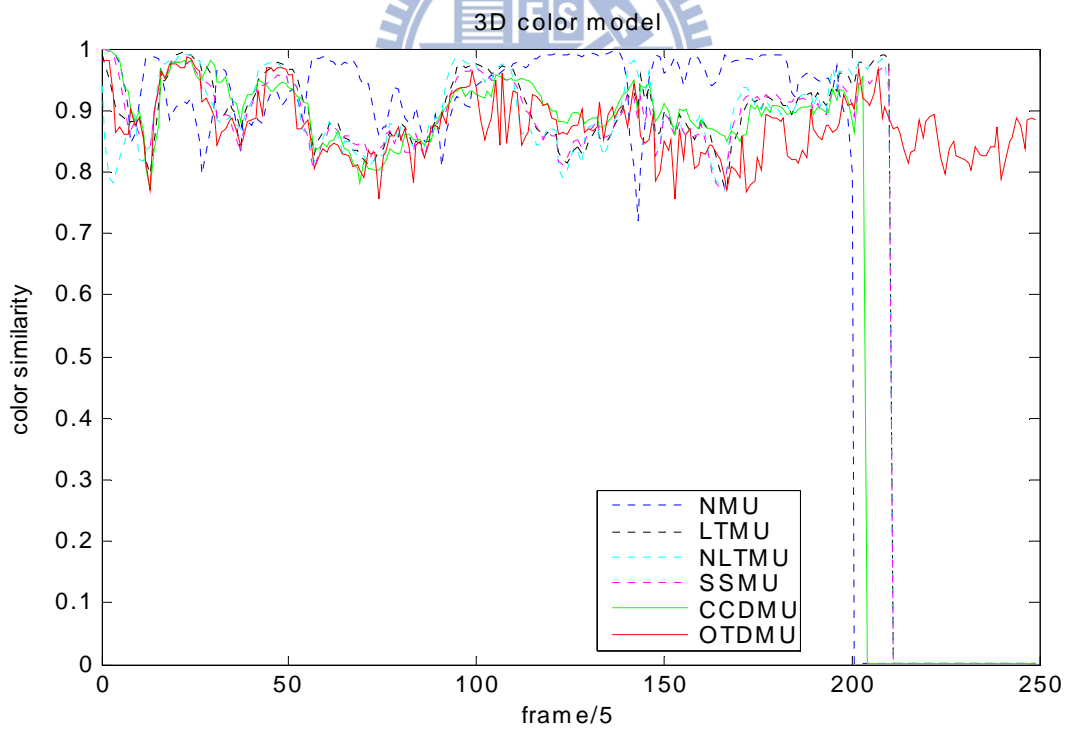


圖 4.2-36: 人體追蹤二目標顏色相似度

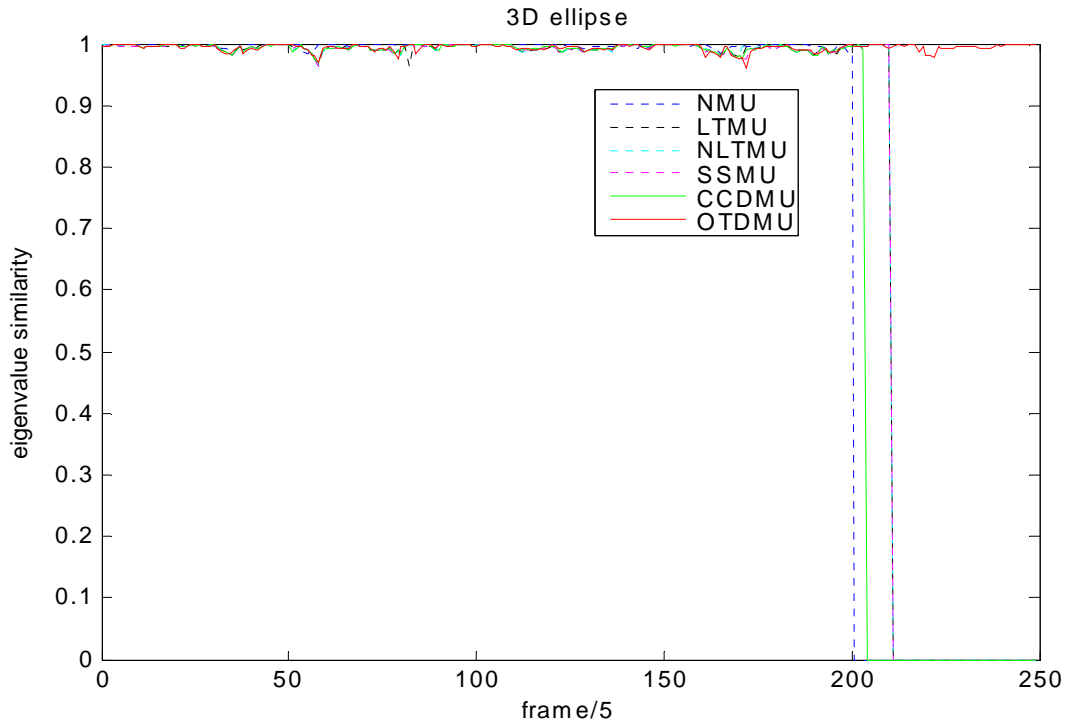


圖 4.2-37: 人體追蹤二目標空間分佈相似度

表 4.2-11: 人體追蹤二數值結果(2D)

2D Mean-Shift	平均處理時間(sec)	平均中央誤差(pixel)
NMU	0.019393	431.53
LTMU	0.020844	431.74
NLTMU	0.024598	431.53
SSMU	0.020343	431.53

表 4.2-12: 人體追蹤二數值結果(3D)

3D Mean-Shift	平均處理時間 (sec)	平均中央誤差 (pixel)	平均顏色相似度	平均空間分佈 相似度
NMU	0.012718	73.929	0.7563	0.80165
LTMU	0.014067	82.901	0.76277	0.83863
NLTMU	0.014364	82.408	0.75838	0.83887
SSMU	0.014523	82.611	0.75745	0.83851
CCDMU	0.014833	92.249	0.73602	0.81081
OTDMU	0.018027	43.592	0.87229	0.99387

實驗三為拿著 Kinect 跟蹤於人體後面，同時測試目標人物與非目標人物交錯而過時，各種演算法的穩定度與準確度。

(1) 二維平均位移演算法：



圖 4.2-38: 人體追蹤三影像輸出結果(2D)

由左至右由上至下分別為 frame 67,81,97,534

一開始四種方法皆有抓到目標物，在 frame 67 時，LTMU 引入過多非目標的資訊而發散，緊接著 frame 81 時，NMU 也因圈選的範圍往上移而追丟目標。在 frame 97，SSMU 也跟著發散，最終只剩 NLTMU 有抓住目標人物。

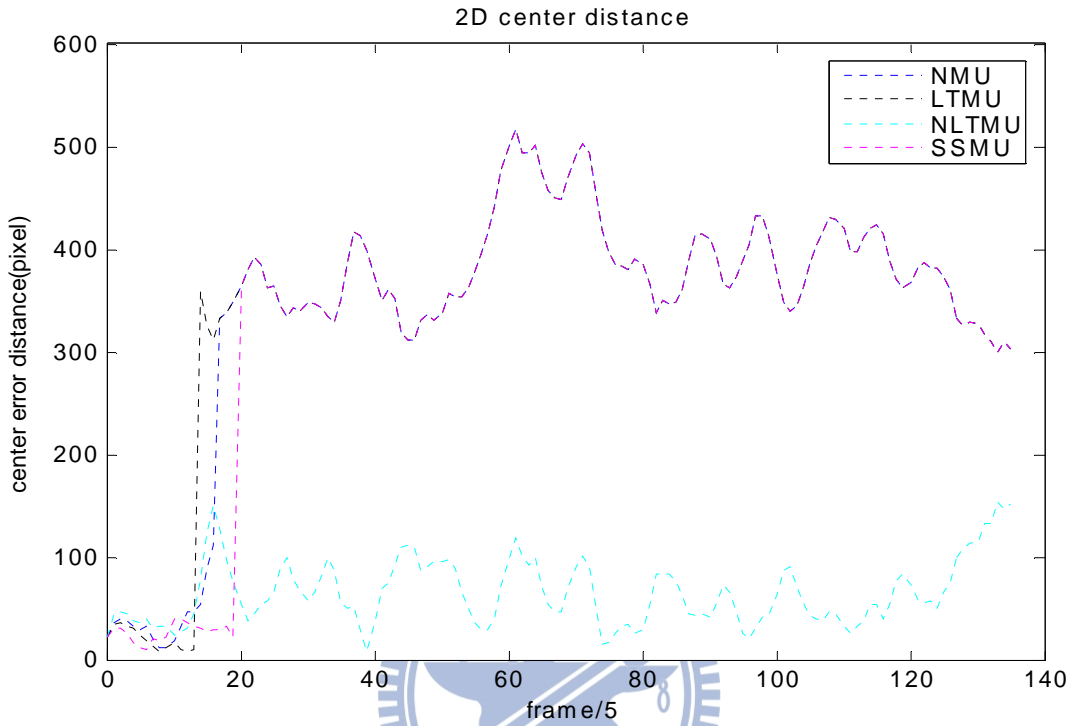


圖 4.2-39: 人體追蹤三中心距離誤差(2D)

(2) 三維平均位移演算法：



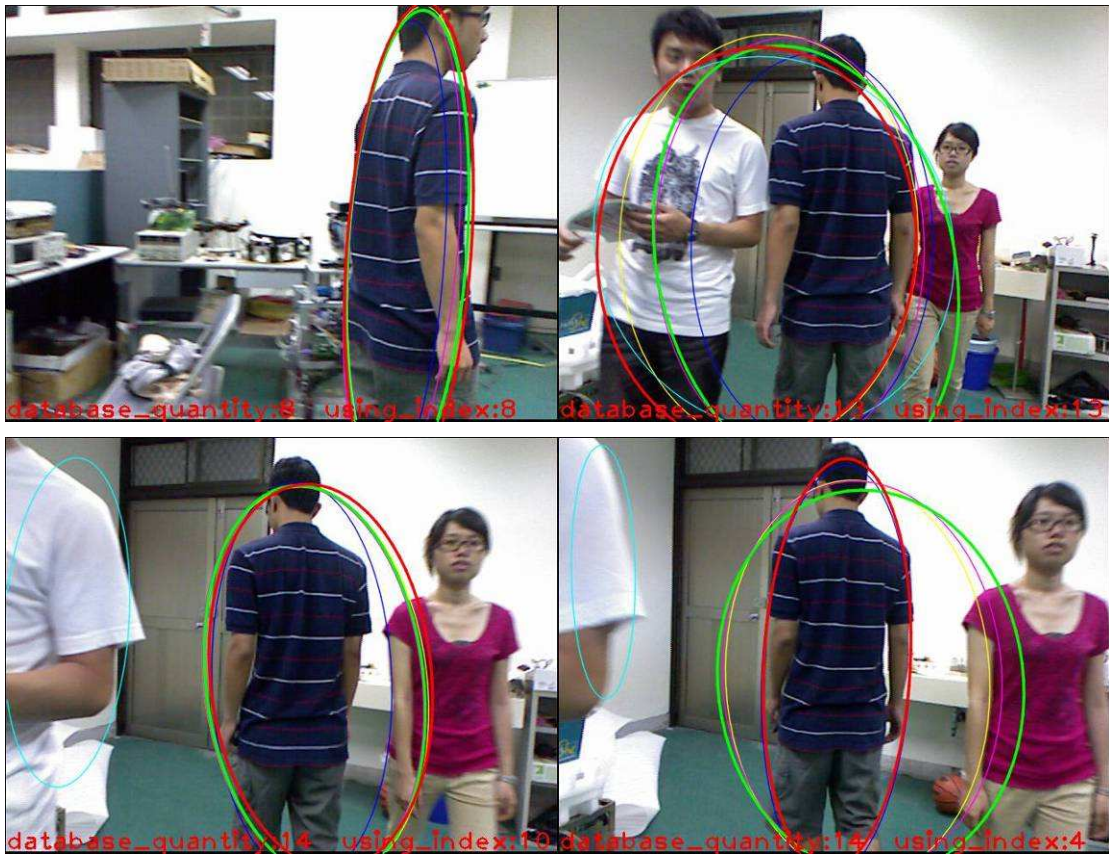


圖 4.2-40: 人體追蹤三影像輸出結果(3D)

由左至右由上至下分別為 frame 77,200,306,535,551,555

在 frame 200 時，因非目標人物與目標物的深度範圍近乎相同，因此與目標物顏色較為相近的手部範圍也被圈選進來。在 frame 535，同樣因深度範圍相同而連同非目標人物一起圈選，其中 NLTMU 在 frame 551 因更新過度而將目標追丟。在 frame 551，同樣因深度範圍相同，另四種更新方法皆因更新而將顏色相近的手部範圍圈選進來，在 frame 555 時，OTDMU 藉由資料庫的幫助，能夠較其他方法更快地重新準確地圈住目標。

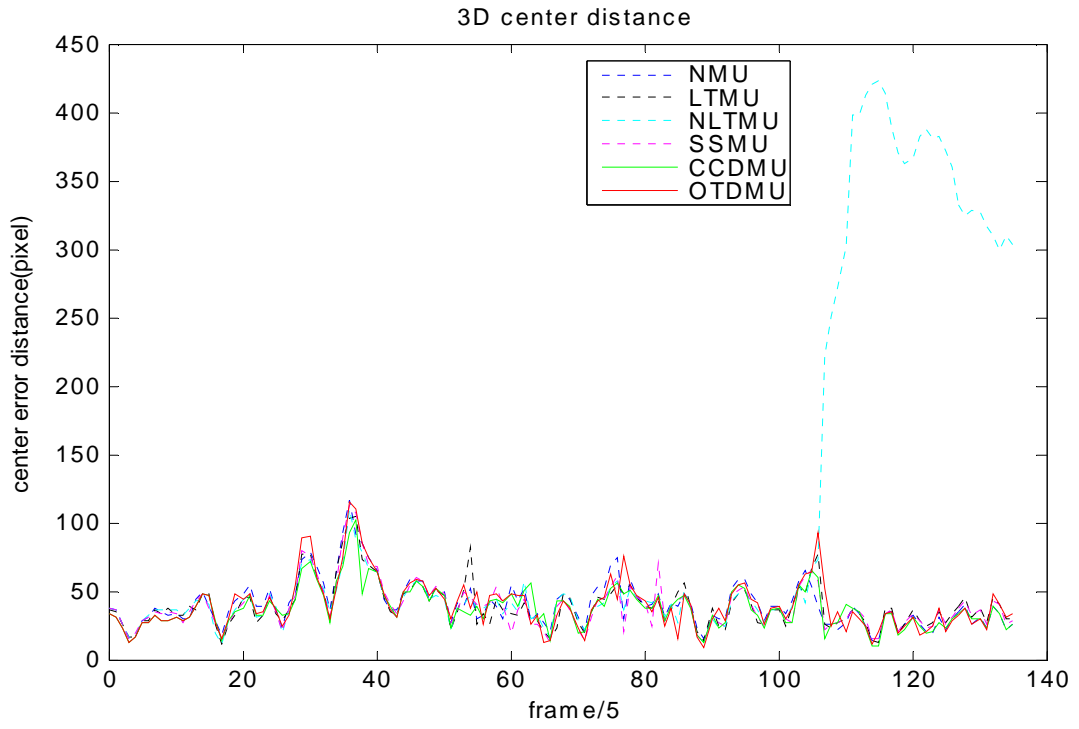


圖 4.2-41: 人體追蹤三中心距離誤差(3D)

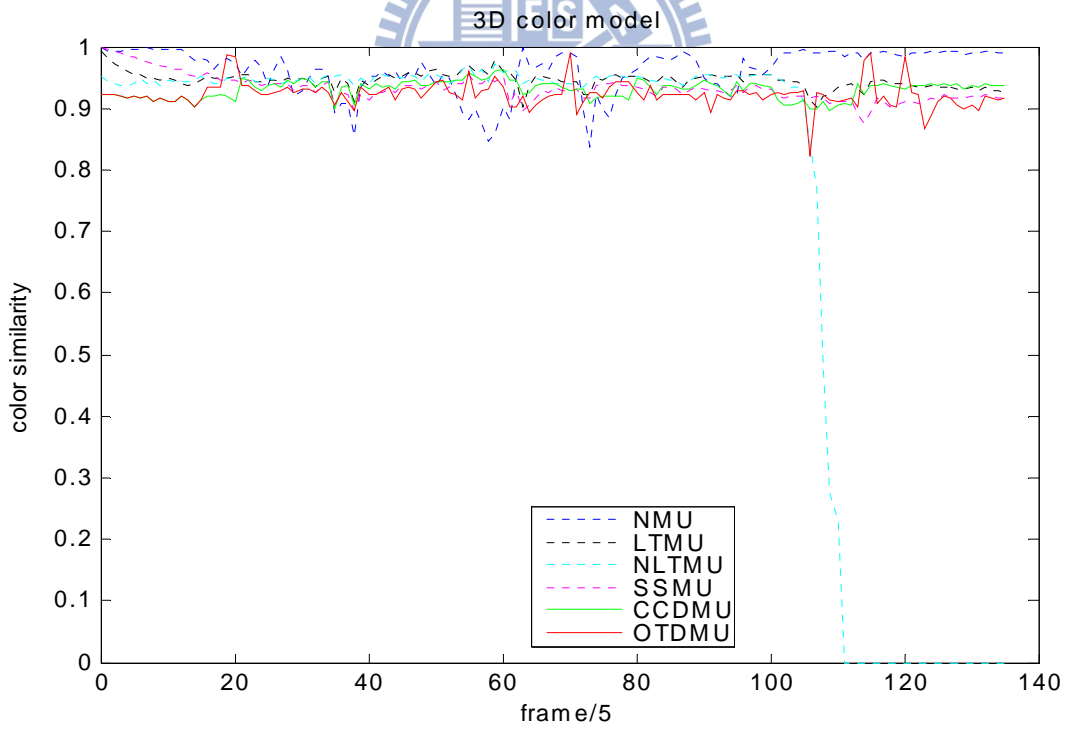


圖 4.2-42: 人體追蹤三目標顏色相似度

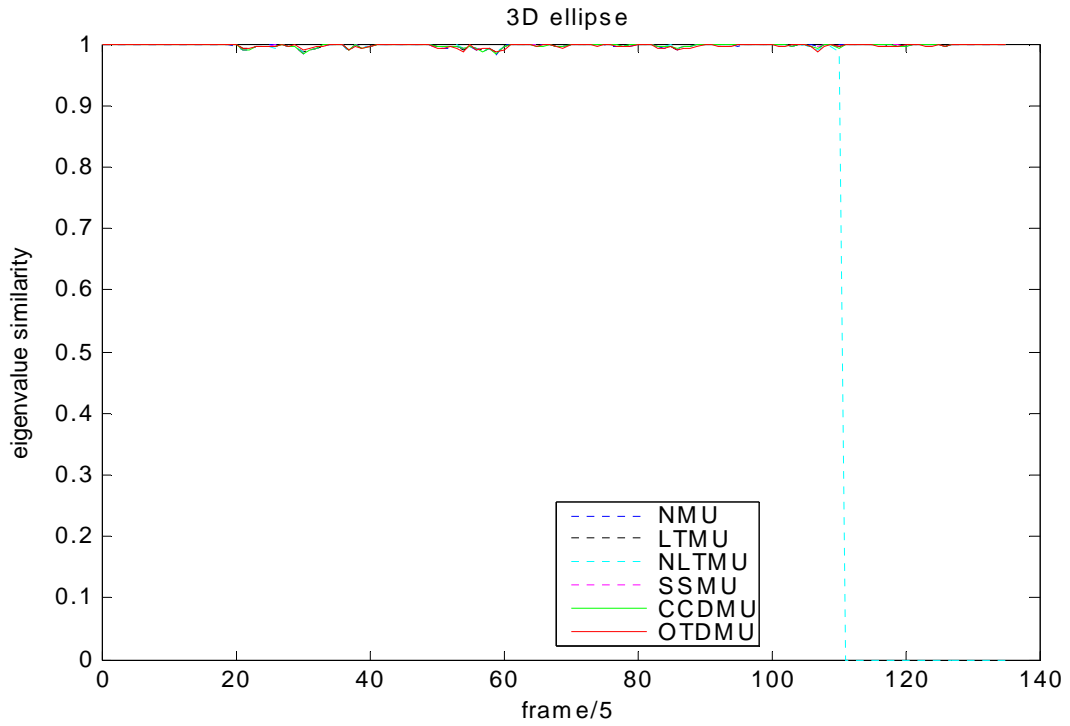


圖 4.2-43: 人體追蹤三目標空間分佈相似度

表 4.2-13: 人體追蹤三數值結果(2D)

2D Mean-Shift	平均處理時間(sec)	平均中央誤差(pixel)
NMU	0.017976	340.56
LTMU	0.024938	344.95
NLTMU	0.010255	64.906
SSMU	0.030017	331.98

表 4.2-14: 人體追蹤三數值結果(3D)

3D Mean-Shift	平均處理時間 (sec)	平均中央誤差 (pixel)	平均顏色相似度	平均空間分佈 相似度
NMU	0.014854	40.611	0.96254	0.99708
LTMU	0.015152	39.128	0.94557	0.99723
NLTMU	0.015347	106.93	0.7576	0.81359
SSMU	0.014984	39.163	0.93216	0.99719
CCDMU	0.015005	37.462	0.9306	0.99718
OTDMU	0.018034	39.859	0.92244	0.99687

4.3 實驗結果與分析

經由上述實驗，依不同演算法分析如下：

二維平均位移演算法：此方法在背景複雜或有相同顏色的情況下穩定度不佳，就算加上目標模型更新的部份對穩定度仍無顯著提升，但其具有高效能，所以在單純背景或目標物顏色與背景有明顯差異時，此方法會較有效率。

三維平均位移演算法：一旦物體顏色跟原始不同，NMU 會很快地失去目標，而在追蹤人體時，因衣服顏色前後不變，故會有最佳的效果。而 CCDMU 能在具有複雜背景的情況下，對於目標模型的顏色變化有更快的適應效果，比起 LTMU、NLTMU 與 SSMU 則會有更佳的效果。而追蹤人體時，因為顏色無變化，若是持續地做目標模型更新，則會不斷地引入皮膚及褲子的顏色，導致模型產生了些許的偏移，然而 OTDMU 對於解決模型偏移的問題是有幫助的，因能藉由資料庫內的相似模型再度取出使用，而能迅速地重新追回目標，OTDMU 比起其他方式，相對來說處理時間稍長，但效果有明顯提升。

第五章 結論

5.1 結論與未來展望

本論文提出了兩個以三維平均位移演算法為基礎的目標模型更新方法。第一個藉由我們所定義的顏色 bin 相似度 *SDCB* 為出發點，結合顏色機率模型相似度與目標物中心移動距離的目標模型更新方式(CCDMU)。第二個為加入線上模板資料庫建構的概念(OTDMU)，藉由顏色機率模型相似度與三維空間分佈相似度做為衡量是否新增資料的標準，就能有效地進行目標追蹤與線上建立資料庫，這會使得追蹤的結果更為穩定。

經過實驗證實 CCDMU 能夠有效地處理目標物在複雜背景中外觀顏色劇烈變化的情況，而 OTDMU 則有助於偏移的目標模型重新追蹤到正確的目標。

大致而言，藉由三維平均位移演算法與 CCDMU 和 OTDMU，我們已可解決大部份三維空間中的追蹤問題，舉凡平移、旋轉、局部遮蔽等等，但若目標物因光影變化產生顏色不同以及空間分佈產生歪斜，就會造成演算法在判斷目標深度範圍時產生誤差，以及當物體移動過於快速時，也就是兩個 frame 之間的物體移動距離過大，使得兩個 frame 的橢圓沒有產生交集，此時會因為平均位移演算法本身的限制，無法找到下一個 frame 的目標物中心點，造成目標追丟，若能加以考慮或是尋找解決方法，對於追蹤的穩定度會有更進一步的提升。

參考文獻

- [1] <http://en.wikipedia.org/wiki/PrimeSense>
- [2] D. Comaniciu, V. Ramesh, and P. Meer, “Kernel-based object tracking,” *IEEE Trans. On Pattern Analysis and Machine Intelligence*, **25**(5):564-577, May 2003.
- [3] Q. Zhao and H. Tao, “Object tracking using color correlogram,” *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2005.
- [4] S. Birchfield and S. Rangarajan, “Spatiograms versus histograms for region-based tracking,” *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2005.
- [5] H. Zhang, Z. Huang, W. Huang, and L. Li, “Kernel-based method for tracking objects with rotation and translation,” *Proc. International Conf. on Pattern Recognition*, 2004.
- [6] Z. Zivkovic and B. Krose, “An em-like algorithm for color-histogram-based object tracking,” *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2004.
- [7] Gary R. Bradski, Santa Clara” Computer Vision Face Tracking For Use in a Perceptual User Interface” Microcomputer Research Lab., CA, Intel Corporation,1998
- [8] W. T. Freeman, K. Tanaka fj, J. Ohta fj and K. Kyuma, Computer vision for computer games.
- [9] <http://opencv.willowgarage.com/wiki/>
- [10] Jwu-Sheng Hu, Chung-Wei Juan, A New Spatial-Color Mean-Shift Object Tracking Algorithm with Scale and Orientation Estimation, National Chiao-Tung University,2006

- [11] Jwu-Sheng Hu, Wei-Ting Weng, A New Meanshift Object Tracking Algorithm Combining Depth Information and Image Color Feature, National Chiao-Tung University, 2011
- [12] Ido Leichter , Michael Lindenbaum, Ehud Rivlin, "Mean Shift tracking with multiple reference color histograms" Computer Science Department, Technion – Israel Institute of Technology, Haifa 32000, Israel, 2010
- [13] Nummiaro K, Meier E K, Gool L J V. An adaptive color-based particle filter. *Image and Vision Computing*, 2003, **21**(1): 99-110
- [14] Wang Yong-Zhong, Liang Yan, Zhao Chun-Hui, Pan Quan. Kernel-based tracking based on adaptive fusion of multiple cues. *Acta Automatica Sinica*, 2008, **34**(4): 393-399
- [15] Nguyen H T, Worring M, van den Boomgaard R. Occlusion robust adaptive template tracking. In: Proceedings of the 8th International Conference on Computer Vision. Vancouver, Canada: IEEE, 2001. 678-683
- [16] Peng N S, Yang J, Liu Z. Mean shift blob tracking with kernel histogram filtering and hypothesis testing. *Pattern Recognition Letters*, 2005, **26**(5): 605-614
- [17] Babu R V, Perez P, Bouthemy P. Robust tracking with motion estimation and local kernel-based color modeling. *Image and Vision Computing*, 2007, **25**(8): 1205-1216
- [18] Shen Zhi-Xi, Yang Xin, Huang Xi-Yue. Study on target model update method in mean shift algorithm. *Acta Automatica Sinica*, 2009, **35**(5): 478-483