基於立體視覺環境下即時人機互動介面之實作

學生：許 志 高　　指導教授：莊 仁 輝　博士

國立交通大學

資訊工程學系

摘　要

在電腦視覺的領域中，利用二維影像資訊重建三維特定座標點的指向系統之研究已行之多年。隨著電腦計算能力的進步，我們已能夠在此種基於電腦視覺的環境下實踐出即時人機互動的應用介面。本論文利用指向物的已知外觀特性，計算出指向物在二維影像中的方向向量，並且在已知的運作空間下，利用投影轉換將方向向量轉換到真實世界座標系中，並且利用攝影機中心與此向量上任兩點共面的特性在三維空間中重建一平面。在系統環境中，我們利用兩重建平面與投影平面三面共點的特性，便可重建出指向點在真實世界中的位置。我們並且發展出一組基於有限狀態機與類神經網路的控制元件，接受並且辨識使用者輸入的軌跡，並且賦予每一個軌跡一個對應的互動行為，根據使用者輸入的軌跡樣式，系統會將不同的行為反饋在介面上，進而達成人機互動的目標。本論文亦提出一些可供實作的系統雛型，以探討本系統在現實生活中的應用性。

An Implementation of a Real-time Human Computer Interaction Application Under Stereo Vision-based Environment

Student：Ji-Gao Hsu　　Advisor：Dr. Jen-Hui Chuang

Institute of Computer Science

National Chiao Tung University

ABSTRACT

The research of vision-based pointing systems through reconstruction of 3-D features using 2-D image information has been studied for decades. With the improvements of computing power, in recent years, one can now implement a real-time human computer interaction application under such a vision-based environment. In this paper, we develop a pointing system which tracks the pointer in two images and calculates its 3-D directional vector. With prior knowledge of some 3-D reference planes, the pointed point can be found without knowing camera's intrinsic parameters and orientations. We develop a control unit to accept and recognize the user inputs in forms of trajectories of the pointed point based on finite state machine and neural network. The goal of developing a human computer interaction application is achieved by responding to the user inputs with proper actions through the user interface. We also implement some useful application prototypes to show the applicability of our system.

目錄

圖目錄

表目錄

Chapter 1　緒論

在傳統上，一個以視覺為基礎的空間主要被賦予監視，事件紀錄，以及遠端視訊溝通的功能，然而隨著電腦計算能力的演進、影像擷取裝置製程的改良以及相關電子裝置價格的下降，我們逐漸可以在這種以視覺為基礎的空間中發展出更具有智慧的應用，在這個領域中，常見的主題有機器人自動導航，場景重建，以及智慧型人機互動介面等。而在場景重建的研究主題中，使用二維資訊來建構三維空間中的室內場景及應用，一直是一項重要的課題，而隨著應用角度的不同，重建的對象及精確度亦有所不同。例如在電腦圖學及虛擬實境的領域中，我們可以利用多個角度的二維影像資訊來重建出整個場景及其中物件的精細三維資訊，在此種應用上，準確度及擬真性是最為重要的考量，他不但影響到整個系統最終是否能夠完整地重現整個場景的資訊，並且讓使用者有身置其中的感覺。而在指向系統的研究中，特定座標點的重建是整個主題的重心，而我們對於重建準確度的考量亦隨著應用上對準確度的要求而有所調整。例如在一個三維的即時射擊瞄準遊戲中，瞄準點重建的準確與否攸關著遊戲使用者的勝負，因此在這種應用中，我們必須發展一套準確且穩定的重建方法，以符合使用者的需求。又例如在一個大型會議簡報系統中，使用者關注的是重建點(通常是螢幕上的滑鼠游標)是否流暢地跟隨著使用者手中指向物而移動，甚至是是否能利用此游標來與電腦系統進行特殊指令的互動，在此時，系統的即時性以及發展一套完善的互動介面機制就成了整體系統的發展重心。

在這篇論文中，我們欲發展一套基於立體視覺空間的人機互動應用程式，利用指向物的形體特徵以及系統運行環境的已知空間資訊，即時地重建出使用者手中指向物在指向平面中的投影軌跡點，並且設計一套辨識軌跡的流程，使得使用者可以有效地與系統溝通，達到人機互動的效果。接下來我們便概略介紹本論文

的動機、流程、以及一些相關的論文回顧。

1.2 研究動機

在一個基於立體視覺的即時人機互動系統中，我們關注的焦點有下列幾項：（1）系統介面的友善度、（2）是否能有效偵測使用者行為、（3）系統的可擴充性、以及（4）系統的穩定性。這些重點在本論文中延伸出下列的問題：（1）是否能發展出一套易於親近的互動機制、（2）是否能有效並快速地在影像中偵測到指向物的動作，並將其轉換成三維空間中的資訊、（3）是否能夠發展出一個易於擴充的系統、以及（4）是否可以發展出一套穩定的系統架構。這些問題是本論文發展的初始動機，我們將從這些問題出發，逐步發展出本論文中各章節的解決方法。

1.3 相關回顧

在這一節中，我們將回顧一些基於電腦視覺的指向及人機互動系統，其中包含利用二維平面或三維空間資訊來實踐的系統，目的都在於利用重建出來的指向點來達到指向功能或者實踐人機互動的應用介面。

1.3.1 基於二維之人機互動

在一般的環境中，基於設置與經費的考量，人機互動藉由單攝影機來偵測軌跡點來實踐。Carsten Kirstein, Heinrich Müller[1]與 Rahul Sukthankar 等人[2]提出的系統即為基於二維平面的雷射筆簡報系統，在此系統中，相機必須先經過校正以取得相機平面、投影平面以及顯示平面之間的對應關係，然後透過偵測位於投影平面上的雷射筆光點，來取得指向的位置，而使用者輸入（例如上一頁、下一頁、關閉等）可以透過將光點停留在特殊位置來實踐，圖 1.1 為此系統的示意圖。

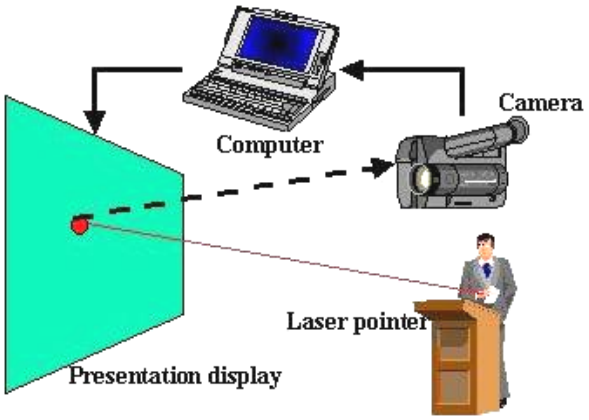


圖 1.1　雷射簡報系統示意圖

在這種系統中，最大的缺憾在於無法偵測多使用者的輸入，James Davis 等人[3]透過利用多台攝影機分別偵測不同的雷射輸入點來達成多使用者輸入的目的(如圖 1.2)，然而系統在建置上耗費較[1][2]為昂貴，且其投影屏與攝影機套裝地建置在一固定場所，故移動性也較低。在基於二維平面的互動系統中，如果指向點座落的範圍落在投影平面之外時，就無法判斷出指向的位置，整體系統的實際應用空間較為狹隘，也因此限制了系統的可能性。此外，偵測雷射筆光點的系統較容易受到環境光源對投影平面的影響而降低其辨識度，這些都是尚有改善空間的地方。

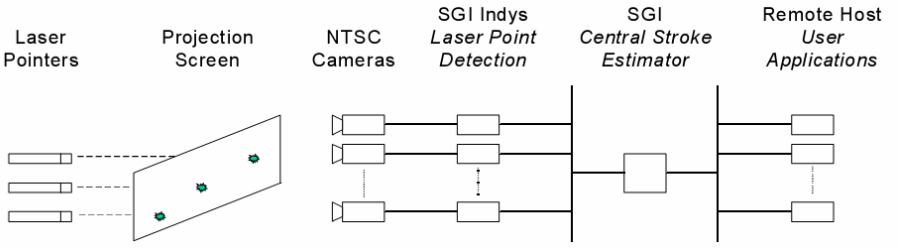


圖 1.2　LumiPoint 系統

1.3.2 基於三維之人機互動

基於三維的指向及人機互動系統中，許多系統利用指向物延伸方向與指向平面的交點來重建軌跡點的位置，如圖 1.3 所示。由於不同的指向物在追蹤上有不同的複雜度，加上指向物的形體通常會限制住三維重建的精確性，因此隨著對系

統的準確度要求不同，在指向物的選擇上亦有所取捨。

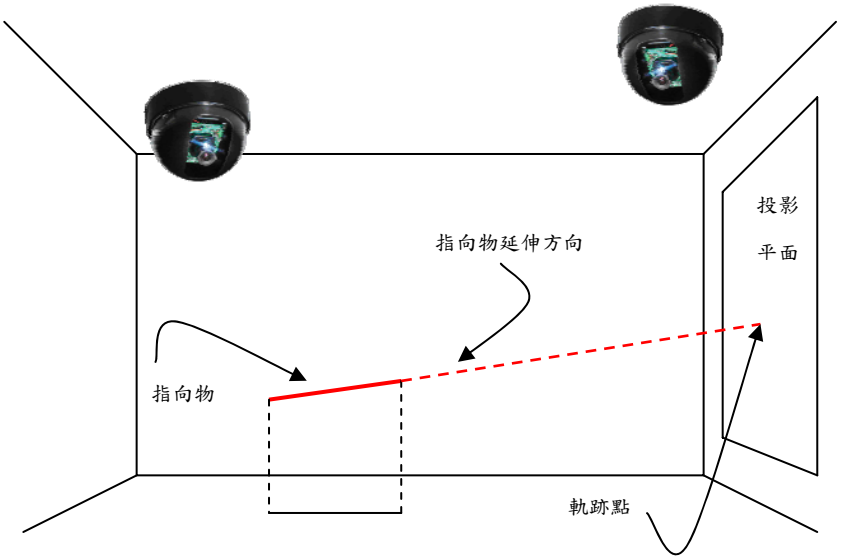


圖 1.3　三維空間中利用指向物延伸方向重建軌跡點

在 Yi-Ping Hung 等人[4]的 Free-Hand Pointer 系統中，使用人眼到指尖之間的連線做為指向方向，如圖 1.4 所示，這個系統使用三台攝影機來重建出此連線（兩台偵測指尖，一台偵測眼珠中心），從系統配置的角度看來，此種系統的空間限制較為嚴格，因為其影像必須包含眼球及指尖，也因此限制了使用者活動的範圍。

圖 1.4　Free-Hand Pointer 系統

在 Christian Leubner 等人[5]的論文中，則提出了一套利用人類手臂作為指向

物的系統，在這個系統中，兩台攝影機會分別找出手臂的中線作為直線向量，作者利用事先知道的空間資訊（包括攝影機位置，空間配置等），將攝影機中心與手臂直線向量之間建立一平面，利用兩台攝影機分別建立的平面與投影平面三面共點的性質重建軌跡點，重建的軌跡點被當作滑鼠游標使用，如圖 1.5 所示，圖 1.6 則表示了圖 1.5 中兩台攝影機所攝得的視角。

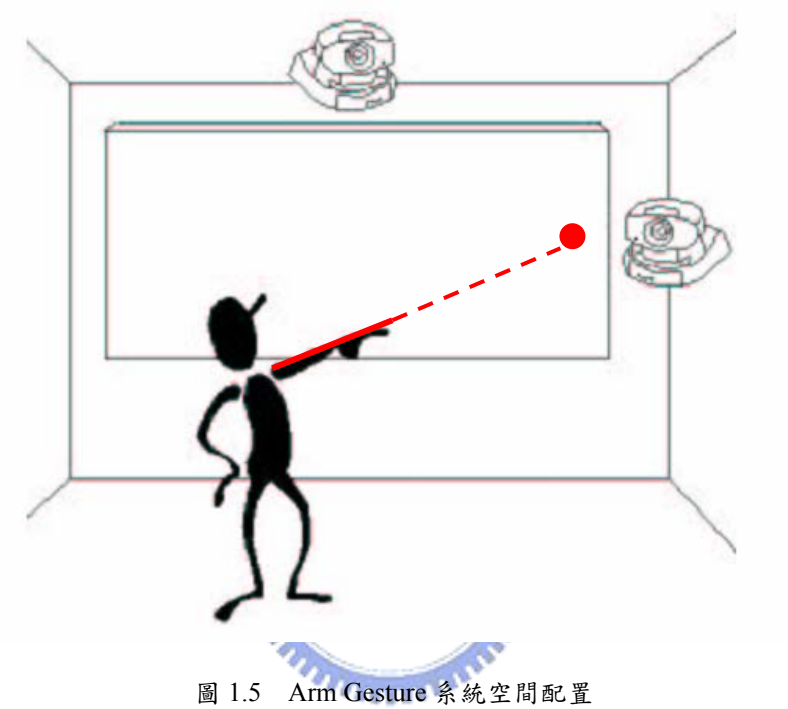


圖 1.5　Arm Gesture 系統空間配置

圖 1.6　Arm Gesture 系統兩視角示意圖

1.4 系統流程

本論文中的系統基於固定立體視覺的基礎下，利用攝影機輸入的連續二維影

像資訊完成追蹤特定指向物的功能，並且利用已知的空間資訊重建出指向軌跡點在三維空間中的座標，接著利用類神經網路辨識在一特定平面區域內的軌跡樣式，將其樣式回傳給控制元件之後，便會產生動作以控制系統，達成人機互動的目標，概略的流程如圖 1.7 所示。

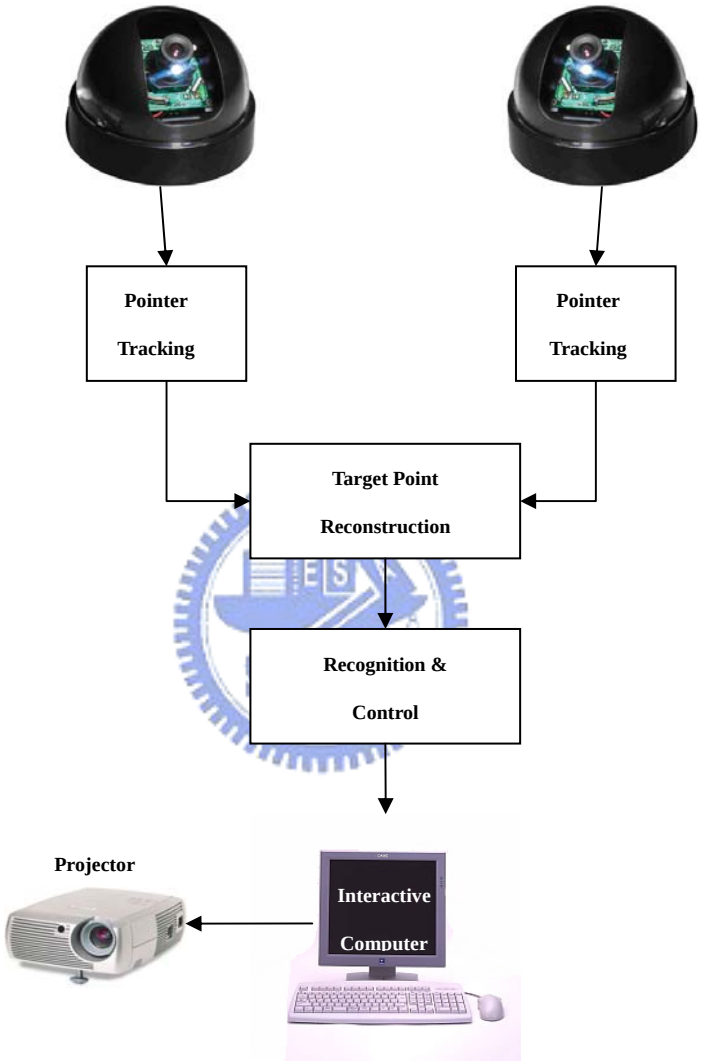


圖 1.7 簡略系統流程

1.5 各章簡介

依照系統流程的安排，本論文共分為六章，除了本章為緒論之外，第二章介紹如何從二維影像串流中偵測到指向物，並且擷取其方向資訊，第三章介紹如何利用已知的空間資訊將二維的空間向量轉換到三維空間，並藉以重建指向軌跡

點，第四章中為了解決電腦與使用者溝通的問題，我們導入類神經網路的概念來克服系統輸入的不確定性，辨識使用者輸入的軌跡樣式，第五章整合整個系統流程，並且介紹我們如何設計整個系統的控制元件部份，使其應用性符合可擴充的目標，第六章中我們會總結這本論文，並且提出未來可能的研究方向。

Chapter 2　指向物追蹤辨識

在本系統場景中，指向物可被用以傳達使用者與系統之間互動所需的資訊，例如方向，軌跡等，因此如何即時且正確地追蹤指向物位置並將其化為有意義的指向資訊，便成為本章的重點。下面我們提出一應用於即時系統上的追蹤流程，如圖 2.1 所示，首先從輸入影像中分析指向物的形體及位置，並將其轉換成三維重建所需要的二維空間方向向量。上述流程可以細分為三個階段，依步驟分別是（1）資料過濾（包含色彩分析與動態物體偵測）、（2）連接物體標記、（3）指向物辨識，在辨識出指向物之後，我們使用主成分分析法來分析出指向物的空間資訊。我們將在接下來的章節中逐一解釋其實際原理。

2.1 色彩過濾

從系統的攝影機中擷取出來的影像為具有 RGB 三色域的彩色影像。在實際的應用中，使用環境的環境光在不同的時間，角度及取景地點會有十分明顯的起伏變化，因而對 RGB 影像的穩定性造成十分嚴重的干擾，亦會降低整體辨識的正確性。再者，同一個場景中也有可能因為光源不均勻分布及照射而使得資訊擷取不足或失誤。為了降低環境中複雜光源變化對影像品質的影響，我們將 RGB 色域影像轉換成 HSL 色域的影像，並賦予其亮度（Luminance）較為彈性的範圍，以降低亮度在整體影像擷取中的影響程度。基於此，我們先將 RGB24 的影像轉換成 HSL 色域的影像，再萃取一特定的色彩範圍，以達到初步過濾影像資訊的效果。

在圖 2.1 的流程中，為了節省計算複雜度而達到即時的系統效能，我們將色彩萃取的步驟設置在流程的前段，以期能在最初的階段就過濾出最接近指向物色彩的資訊，進而在接下來的步驟中免除對於非必要資訊的判斷。本系統中，我們

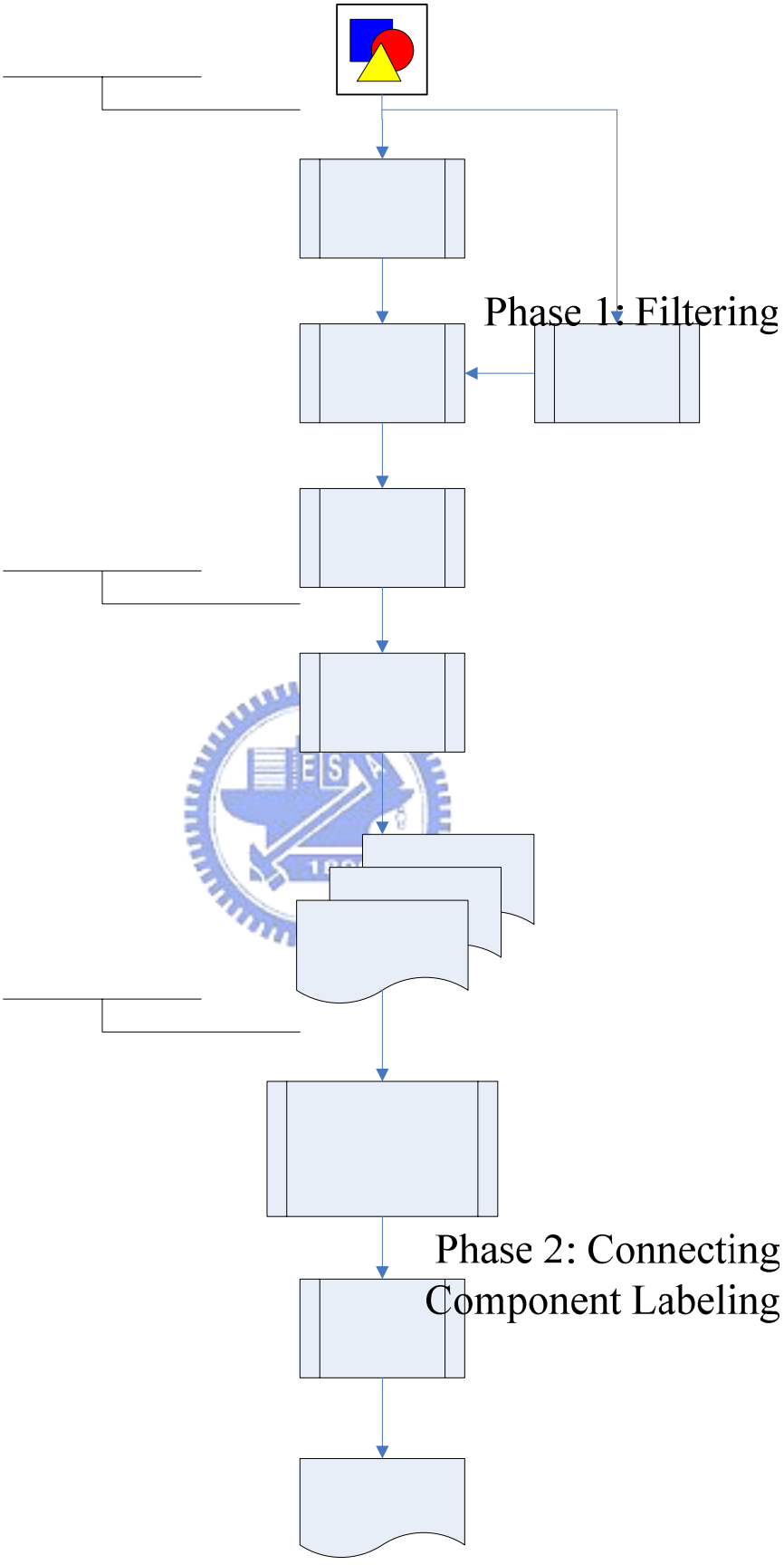


圖 2.1　指向物分析流程圖

使用一色彩單一的棒狀物作為指向裝置，如圖 2.2 所示。

圖 2.2　指向物示意圖

由於指向物的色彩特性為既知事實，我們可事先固定 HSL 色彩萃取器的 Hue 及 Saturation 值（數據範圍分別為 Hue：340°~20°及 Saturation：0.15~1.0，其中 Hue 值所涵蓋的色彩區域如圖 2.4 右所示），而過濾掉與此區間差異太大的環境變因，當我們選取寬鬆的 Saturation 區間時，可以保留較多物體資訊，但是物體會出現較嚴重的渲染情形，而且在接下來的辨識步驟會較耗時；反之，若選取較為嚴苛的 Saturation 區間，則僅得到較少的物體資訊，但是可以進一步解決渲染的問題，如圖 2.3 所示，圖 2.3 的中間是原擷取圖，圖左為取 Saturation 區間為 0.1 至 1.0 所得，圖右為取 0.25 至 1.0 所得，可以看出圖左有較多點資訊，但是指向物周圍出現嚴重的渲染情形，不利於之後重建，圖右則較忠實保留原指向物資訊。

圖 2.3　Saturation 值對指向物資訊的影響

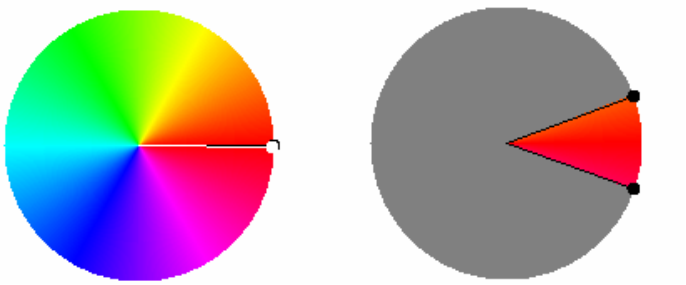


圖 2.4　Hue 值色域示意圖（左）
及本實驗中 HSL 色彩萃取器所使用的 Hue 值範圍（右）

圖 2.5 是一 HSL 影像在經過色彩萃取前後的結果，圖左是在未萃取色彩之前的原圖，影像中可以看到使用者、指向物、以及環境背景等影像資訊，而圖右則是經過 HSL 色彩萃取器（係數為 Hue：340°~20°、Saturation：0.15~1.0、Luminance：0.1~1.0）擷取出特定色域之後的結果，可以看出已大致過濾掉許多明顯不相關的資訊。值得注意的是，由於使用色彩做為過濾資訊的要件，因此當場景中出現大量屬於此色彩區間內的移動資訊時，若其中恰有許多筆直棒狀物體（如紅色的襯衫、手臂等），便容易出現誤判的情形，此時便會造成追蹤的錯誤，因此挑選場景中鮮少出現的顏色為指向物的色彩，便成為在此階段的重要考量。

圖 2.5　色彩過濾前後之比較，
係數為 Hue：340°~20°、Saturation：0.15~1.0、Luminance：0.1~1.0

2.2 動態物體偵測

在經過色彩萃取後，我們開始分析場景中處於運動狀態的物體，在本系統中，我們使用前一張圖像為動態偵測的緩衝影像，流程如下：

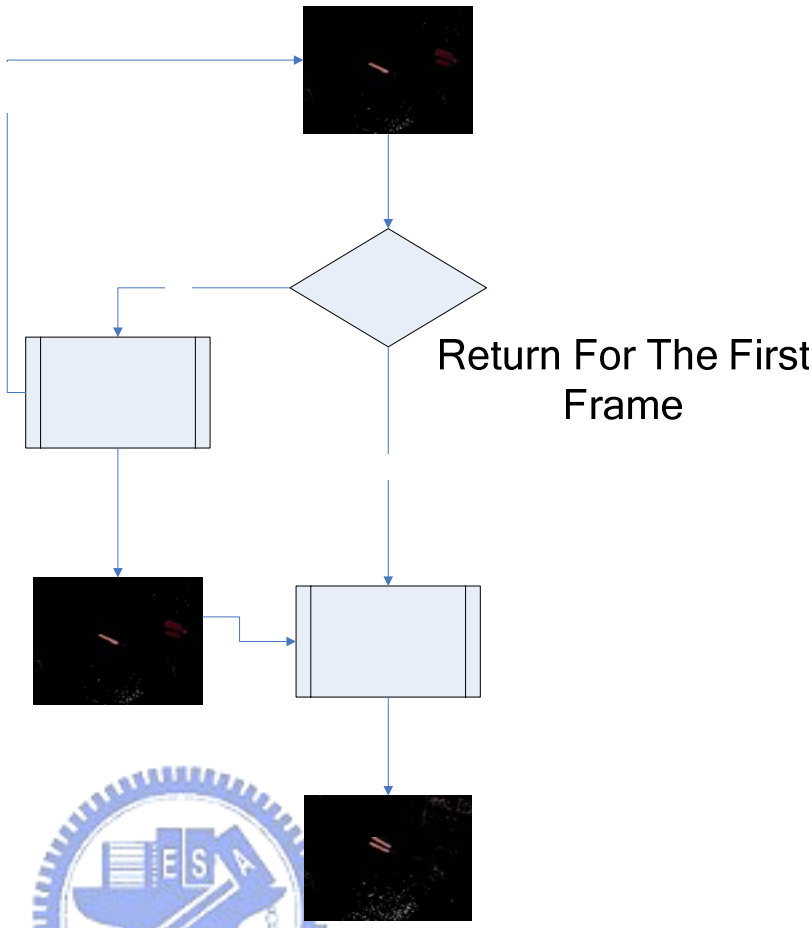


圖 2.6　動態偵測流程圖

此流程需要暫存一張時序上差距為一的圖作為緩衝，將現有影像與緩衝影像相減，所得到的結果即代表整體環境在兩個時序之間的變化量，此方法可用來區分出固定攝影機時整體環境中靜態物體（通常是背景）與動態物體（在本例中是指向物）的不同，圖 2.7 顯示兩張在時序上連續的影像，而圖 2.8 則是這兩張影像經過 Difference Filter 之後的結果，可以看出圖 2.7 中的椅子及牆壁等背景部份已經被移除，並且顯示出指向物在兩張影像中的位置。為了避免影像對雜訊太過敏感，因而產生過多誤判區域，我們在得到最終結果之前必須再經過 Threshold Filter 將影像二質化（Binarize，圖 2.9 為取門檻值為灰階值 50 之結果），當差距大過於一個門檻值時才將其列入後續考量。

圖 2.7　兩張經過濾色之後的連續影像

圖 2.8　連續影像經過 Difference Filter 之後的結果

圖 2.9　圖 2.8 經過門檻限制之後的二質化結果（Threshold = 50）

2.3 連接物體標記

在經過上述步驟之後，我們可以得到經過濾色的原圖中所有可能變化的像素，這些像素混雜著雜訊、運動的背景、以及任何可能的指向物候選者。為了將像素化為有意義的個別區域，我們必須將個別連接的物體一一標記出來。為了定義連接物體，我們假設個別的連接物體在二質化的影像上是連接的像素群集，而兩個像素連接的定義為此兩像素為 4-Neighbors（假設兩點其中一點的座標為（i, j），則另一點必在（$i+1, j$）、（$i-1, j$）、（$i, j+1$）、（$i, j-1$）四個座標之中）或 8-Neighbors（假設兩點其中一點的座標為（i, j），則另一點必在（$i+1, j+1$）、（$i+1, j$）、（$i+1, j-1$）、（$i, j+1$）、（$i, j-1$）、（$i-1, j+1$）、（$i-1, j$）、（$i, j-1$）八個座標之中），如圖 2.9 所示。

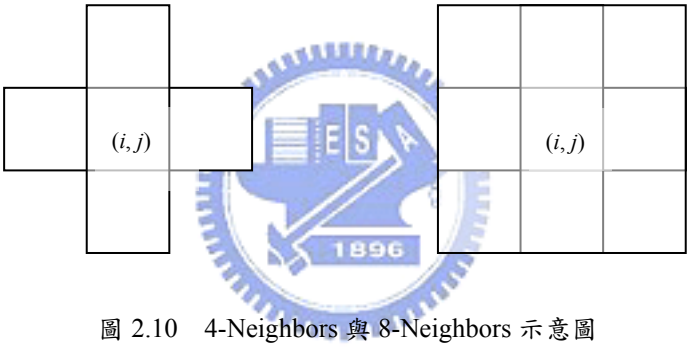


圖 2.10　4-Neighbors 與 8-Neighbors 示意圖

我們從圖上任一點開始標記連接物體，對所有相互成 8-Neighbors 連接的像素標以一獨特數字標記，代表其相互連接，每當發現一未標記的像素時，便以一新數字作標記，依此反覆做到整張二質化影像的所有點都被標記為止，如圖 2.11 所示。為了進一步分析這些小區域的形狀特性，我們使用連接物體標記的方法來將所有個別連接的物體切分開來之後，再計算個別連接物體的區域資訊以記錄物件在影像中的位置及水平／垂直方向的範圍，用來在最近的一張影像中找尋可能的指向物候選者，標記結果如圖 2.12 所示。有了座標資訊的資訊後，我們回到現有影像（即最近一張擷取進來的影像），利用圖 2.13 所細分出的各個區域切割影像，過濾包含過少像素的區域，我們得出圖 2.14 所示的一些可能候選者：

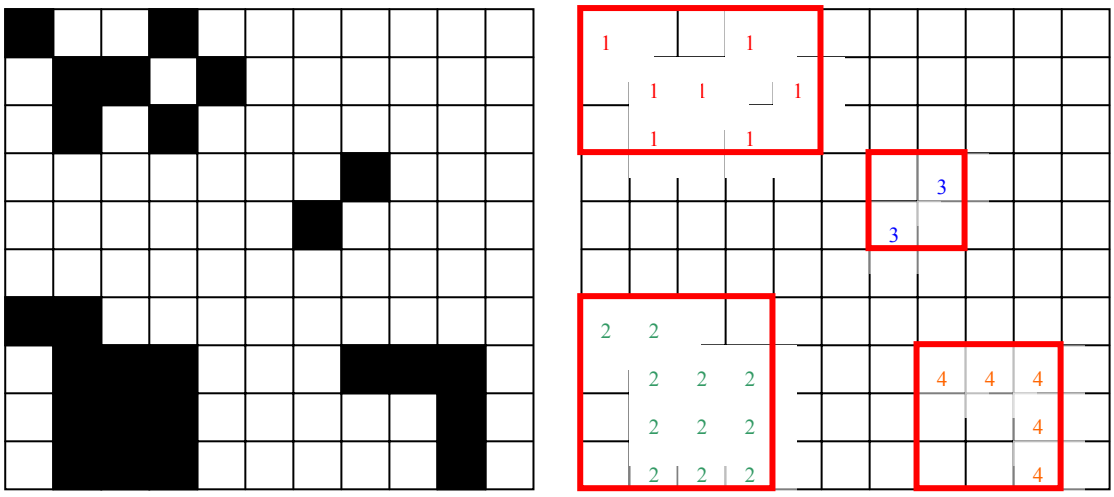


圖 2.11　連接物體標示，
左圖為標示前的二質化影像，右圖為標記之後的區分圖

圖 2.12 經過連接物體標記區分之後的各個物體（左圖，以顏色區分）及其座標範圍資訊（右圖，僅列出長或寬大於 4 像素以上的結果）

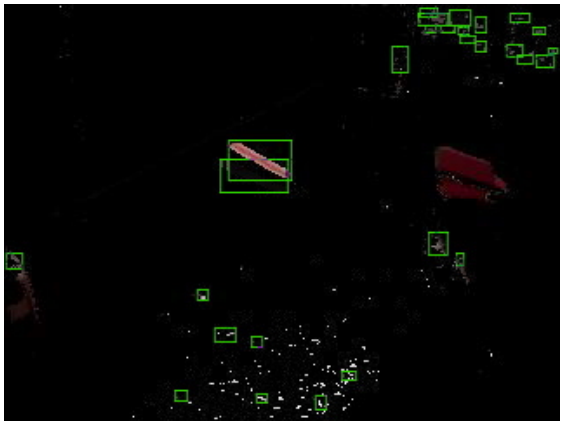


圖 2.13 現有圖像在兩個時序間變化的區域
（僅列出長或寬大於 4 的區域）

圖 2.14、利用區塊分類器所得出的區域資訊在現有圖像中切割出來的各種物件（經二質化處理）

2.4 指向物辨識

在上一節中，我們已得出現有影像中所有可能的候選人圖樣，如何從這些後選人中找出指向物，便有賴於對候選人圖像中形狀資訊的分析。首先我們定義指向物的形狀為：(1) 筆直狀非彎曲物體、(2) 長寬度比例在一特定區間內，使其整體樣貌呈現細長型、(3) 在符合上述條件下，體積最大者。在此階段第一個步驟中，我們首先依物體區域資訊的水平／垂直範圍比例來決定垂直掃描或者水平掃描目標區域，如圖 2.15 所示。

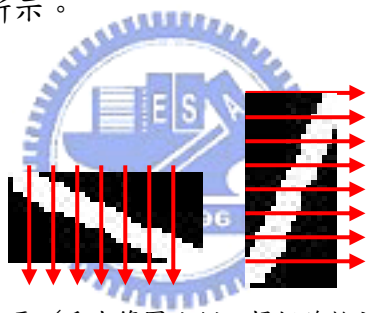


圖 2.15 依區域水平／垂直範圍比例以掃描線檢視物體筆直程度

當一物體是屬於直線非彎曲狀時，每一行（列）的連續白色像素區域必與下一行（列）的連續白色像素區域高度差距在一固定間隔區間內，若y_n代表著第n行的白色像素區域上方y軸高度座標，則行與行之間的高度差距可以以下式表示：

$$\begin{cases} -1 < y_n - y_{n-1} < 0 \\ 0 < y_n - y_{n-1} < 1 \end{cases} \tag{2.1}$$

以圖 2.16 為例，此間隔區間為 0 至+1，而在圖 2.17 中，則表示了另一個彎曲物體，高度差距區間為-2 至 1 的例子。

物體行與行之間上方的高度差。

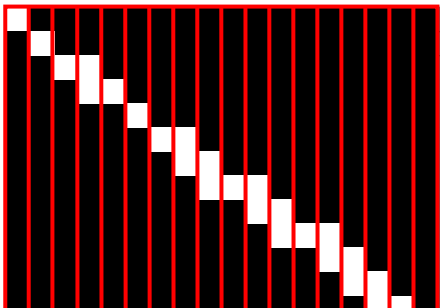


圖 2.16 等寬直線物，兩行之間高度差為 0 至+1

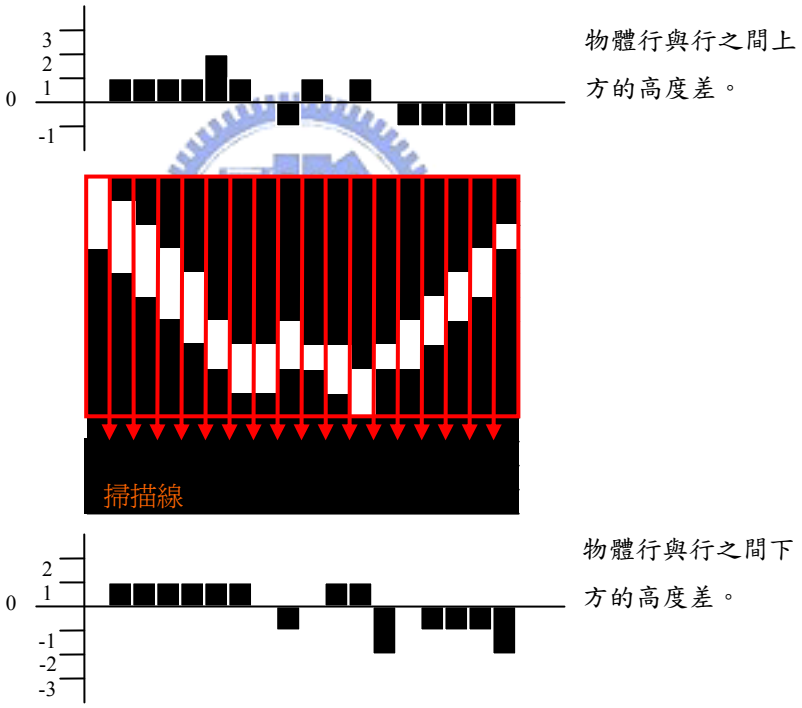


圖 2.17 彎曲物體，兩行之間高度差不規則，介於-2 至 1 間

找出符合上述條件的物體之後，為避免誤認不符合細長外形的物體，須再經過長寬比的檢驗。由於細長外狀的物體有下列特性：物體的寬度會遠小於物體的長度，因此我們以物體平均寬度除以物體區域的對角線長（如圖 2.18 所示），大於一定門檻值的才列入最終考量。最後，剩餘的候選者以體積最大（總白色像素最多者）的為最終的勝出者。

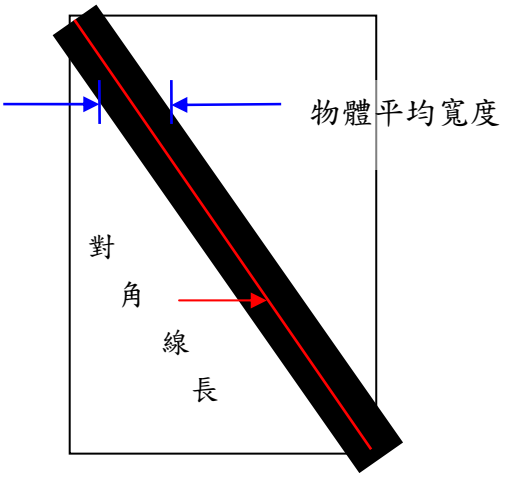


圖 2.18 計算細長物體外觀比例示意圖

2.5 主成分分析法（Principal Components Analysis）

得出上述結果後，我們必須將像素群的資訊轉化為三維重建所使用的直線向量。在傳統的方法上，有最小平方差調適法（Least Square Fitting）與主成分分析法（Principal Components Analysis）等方式可供計算出可靠的空間向量，經過實驗，主成分分析法在指向物趨於水平時仍能有效重建指向物的方向向量，因此在本實作中，我們使用 PCA 這種統計方法來得出三維重建所需要的直線向量。

主成分分析法是一種將資訊從高維度降低至低維度的統計方法，此技巧利用線性轉換將資料轉換到一新座標系，在此座標系內所有原座標系內資訊（即點座標）的最大變異量會沿著第一象限軸方向（稱為第一主成分, First Principal Component），而次大變異量會沿著第二象限軸方向（稱為第二主成分），依此類推。在將資訊從高維度降到低維度的過程中，PCA 會依變異量保留原座標系資訊的特性，因此其優點在於以線性組合保有完整的原座標系變數訊息。我們利用此方法來將指向物的影像像素轉換為主成分向量，藉此表示指向物所欲表達的方向訊息。以下便是 PCA 的計算流程。

已知有 n 個二維空間的點，(x_i, y_i)，$1 \le i \le n$，欲使用 PCA 找出最能表示這些點朝向的線。我們先計算出這些點的重心：

$$(\bar{x},\bar{y}) = \left(\frac{\sum_{i=1}^{n} x_i}{n}, \frac{\sum_{i=1}^{n} y_i}{n}\right) \tag{2.2}$$

再將每個資料點的座標減去重心座標，做一個將座標系原點平移的動作：

$$(x_i', y_i') = (x_i - \bar{x}, y_i - \bar{y}) \text{ ，} 1 \le i \le n \tag{2.3}$$

得到了這些平移後的點座標，我們便可以利用這些點座標計算共變異數矩陣（Covariance Matrix），此矩陣的計算方式如下：

$$C = \begin{pmatrix} \text{cov}(X,X) & \text{cov}(X,Y) \\ \text{cov}(Y,X) & \text{cov}(Y,Y) \end{pmatrix} \tag{2.4}$$

式（2.4）中的 covariance 算法為：

$$\text{cov}(X,Y) = \frac{\sum_{i=1}^{n} (X_i - \overline{X},\ Y_i - \overline{Y})}{(n-1)} \tag{2.5}$$

至此，我們已經由原始資料點算出代表其間相互變異關係的共變異數矩陣，下一步就是對 C 做 eigenvector decomposition，可求得兩個 eigenvalues 及其對應的 eigenvectors。值越大的 eigenvalue 所對應的 eigenvector 表示變異量越大的特徵，因此我們選取值較大的 eigenvalue 所對應的 eigenvector 作為這些資料點的朝向線，此即指向物直線線段於影像中之位置。

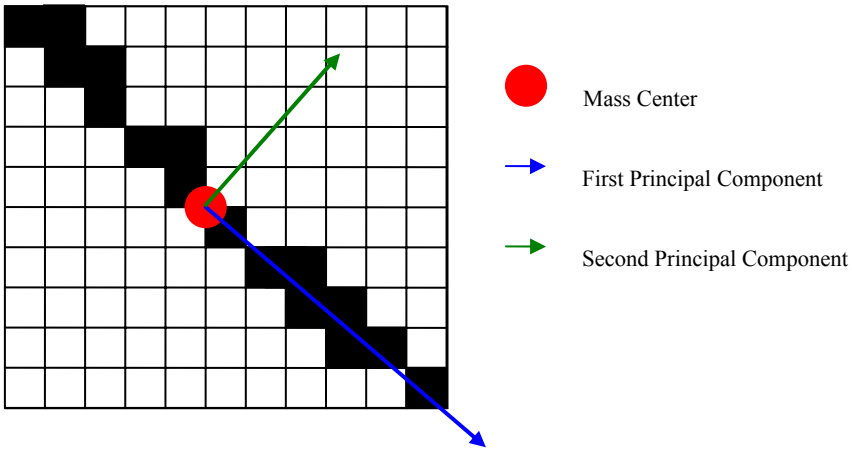


圖 2.19　利用 PCA 得出 Principal Components 之示意圖

Chapter 3　投影幾何轉換

本論文中所發展的系統為基於雙視角環境下的軌跡點重建系統，其主要概念為利用兩張視角重疊的影像計算出影像中特定物體在實際三維空間中的位置，在本論文中使用這種空間轉換的特性來得出指向物以及其與投射平面的交點。在實作上，我們首先將前章節所得出的直線向量透過 Homography Matrix 在擷取影像平面（Image Plane）與實際空間平面（World Plane）之間做投影轉換，此直線向量從擷取影像平面轉換到實際空間平面之後，便可以利用攝影機鏡心與轉換後的直線向量在實際空間中建立一平面，最後計算投射平面與兩台攝影機分別建立的平面相交所得出的交點，此交點即為一軌跡點，如圖 3.1 所示，本章節介紹與上述實作相關的數學背景。

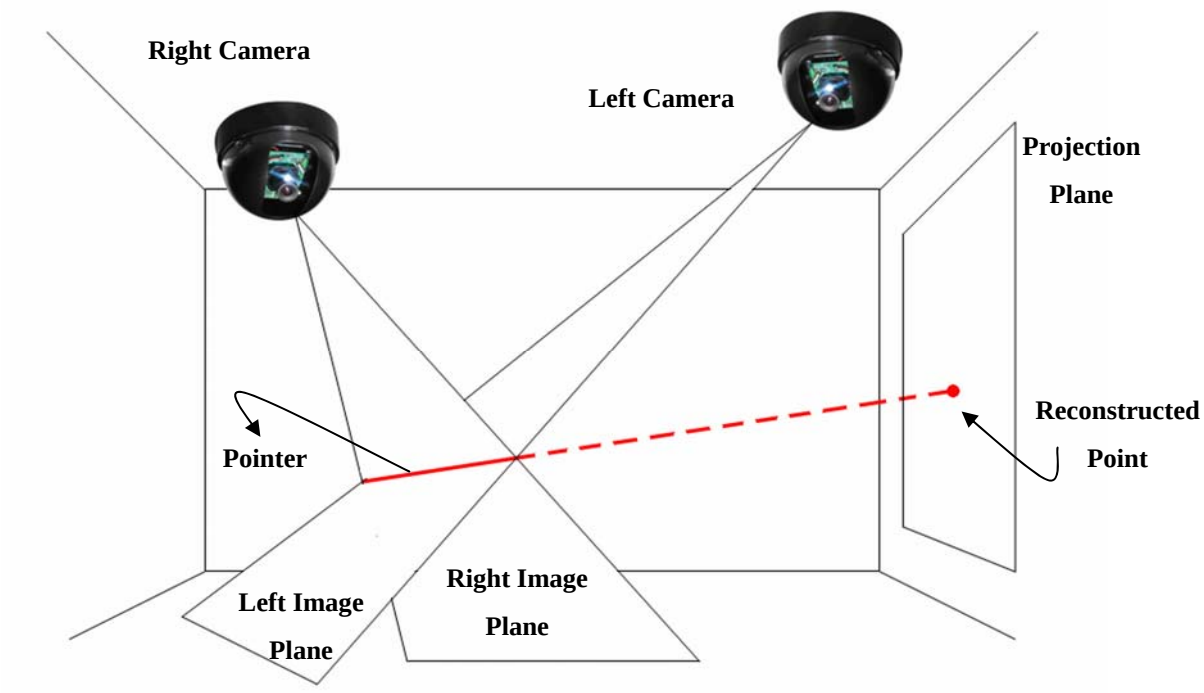


圖 3.1　軌跡點重建示意圖

3.1 Homography Matrix

3.1.1 平面投影轉換

在電腦視覺的領域，我們使用平面投影轉換（Homography Mapping）來將一群位於同一平面上的點集合 x_i 對應到一群另一平面上的點集合 x_i'，其轉換的方式為使用一投影轉換矩陣 H，將 x_i 中的每個點轉換到 x_i' 中相對應的點。若使用齊次座標系來表示 x_i 與 x_i'，則 $x_i=(u_i,v_i,w_i)^T$ 而 $x_i'=(u_i',v_i',w_i')^T$，此時我們可以得到下列的表示式：

$$Hx_i = x_i' \qquad (3.1)$$

其中 H 是一個 3×3 的可逆矩陣。此外，若是允許 Hx_i 不完全等於 x_i'，而是具有相同方向的向量，則我們可以將（3.1）改寫成下面的型態：

$$Hx_i = \lambda x_i' \qquad (3.2)$$

其中 λ 是一個不為零的比例係數。

從系統的角度來看，我們可將 x_i 視為從攝影機擷取進來得影像點座標集合，而將 x_i' 視為實體世界中一特定平面的點座標集合，在 x_i 上的點座標經（3.2）轉換之後可以得到一在 x_i' 上的對應點，如圖 3.2 所示。

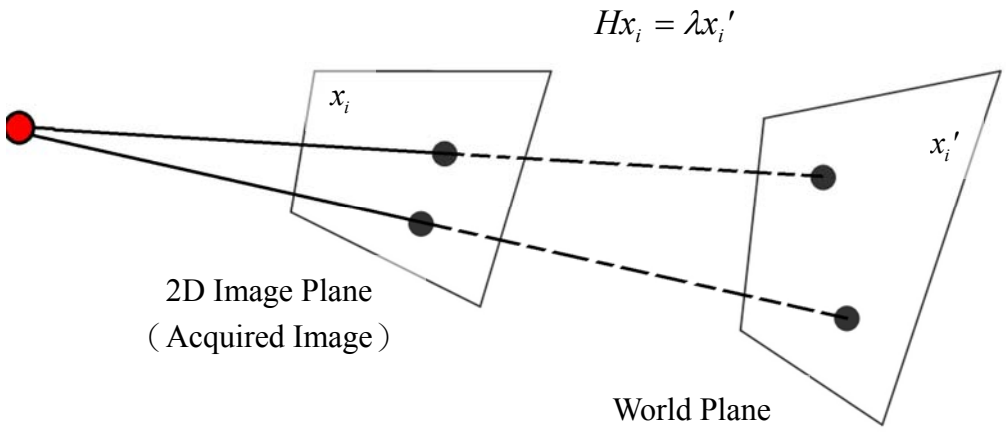


圖 3.2　投影轉換示意圖

3.1.2 投影轉換矩陣

在有了投影轉換的觀念之後，我們必須求出投影轉換矩陣 H，以在接下來的步驟中可以利用 H 在兩個平面之間作點對應點的轉換。從（3.2）的定義可知，H 為一個 3×3 的可逆矩陣，因此有九個變數，然而由於已定義了比例常數 λ，因此可固定 H 中的縮放係數（The Scale of H），使得其自由度為 8。在（3.2）中，藉由展開矩陣，我們知道一組在 x_i 與 x_i' 中兩個點的對應關係可以表示成兩條線性方程式，分別是代表 x 座標的對應關係和 y 座標的對應關係，因此 n 組點與點之間的對應關係可以表示出 $2n$ 條有八個變數（因為 H 中的自由度為 8）的方程式。由於有八個變數，故當 $n = 4$ 時我們可以得出所有變數的解。換句話說，當我們得知此兩平面之間的四組對應關係時，便可以開始著手計算 H 的值，在傳統上，解 H 值的方法有下列三種：

（1）非齊次線性解（Non-homogeneous Linear Solution）：固定九個矩陣元素中的其中一個（通常把固定的值設為 1），對應到其餘八個元素的線性方程式則使用 Pseudo-inverse 的方式求解。這是傳統上最常使用的方法，然而當前述被固定的矩陣元素其實際值為 0 時，此種方法會使得結果產生極大的誤差，因此在本論文中不採用此種方法。

（2） 非線性幾何解（Non-linear Geometric Solution）：此種方法從幾何的角度切入，藉由將預測點與實際映射點之間的誤差最小化來求解，此種方法無封閉解（Closed Form Solution），因此在不保證有解的情形下，並不適合用於我們系統的實作。

（3） 齊次線性解（Homogeneous Solution）：此方法使用 SVD（Singular Value Decomposition）求解，此方法在實作上可去除非齊次線性解在特定情形下產生預測誤差的情形，因此我們使用此種方法來求解，下面我們便詳細解說此種解法的步驟。

3.1.3 齊次線性解

在理想的狀況下，給定一組兩平面上點與點間的對應集合 $x_i \leftrightarrow x_i'$，我們期望能求出一個轉換矩陣 H 以滿足（3.2），若我們以外積的方式表示（3.2），則可以將其改寫如下：

$$x_i' \times Hx_i = 0, \text{ with } \quad H = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix} \tag{3.3}$$

再進一步將 H 的第 j 列表示成 h^{jT}，則我們可以進一步將（3.3）中的Hx_i表示成：

$$Hx_i = \begin{bmatrix} h^{1T} x_i \\ h^{2T} x_i \\ h^{3T} x_i \end{bmatrix} \tag{3.4}$$

如同第 3.1.1 節所述，我們可以利用齊次座標系將 x_i' 表示成 $x_i' = (u_i', v_i', w_i')^T$，則（3.3）中外積的結果為：

$$x_i' \times Hx_i = \begin{bmatrix} v_i' h^{3T} x_i - w_i' h^{2T} x_i \\ w_i' h^{1T} x_i - u_i' h^{3T} x_i \\ u_i' h^{2T} x_i - v_i' h^{1T} x_i \end{bmatrix} \tag{3.5}$$

若已知a與b為矩陣，則有$aP^Tb = b^Ta$的特性，因此 $h^{jT} x_i = x_i^T h^j$，$j = \{1,2,3\}$，我們可以再將（3.3）與（3.5）化簡後，可以得到如下的形式：

$$\begin{bmatrix} 0^T & -w_i' x_i^T & v_i' x_i^T \\ w_i' x_i^T & 0^T & -u_i' x_i^T \\ -v_i' x_i^T & u_i' x_i^T & 0^T \end{bmatrix} \begin{bmatrix} h^1 \\ h^2 \\ h^3 \end{bmatrix} = 0 \tag{3.6}$$

最後，由於在（3.6）中所運算出的三條方程式只有前兩條是線性獨立的（第三條方程式由前兩條方程式所求得），所以（3.6）在只使用前兩條方程式的情形下，可以再進一步化簡成：

$$\begin{bmatrix} 0^T & -w_i' x_i^T & v_i' x_i^T \\ w_i' x_i^T & 0^T & -u_i' x_i^T \end{bmatrix} \begin{bmatrix} h^1 \\ h^2 \\ h^3 \end{bmatrix} = 0 \qquad (3.7)$$

從3.1.2節可知，選取兩個對應點集合 x_i 、x_i' 中的四組對應點套入式（3.7）中便可以產生八個方程式，進而求解出 H ，而求解 H 的演算法詳述如下：

（1） 對於 $x_i \leftrightarrow x_i'$ 中每組對應的點，可從式(3.7)得到一個 2×9 的矩陣 A_i。

（2） 有 n 組對應點，則可得到 n 個 2×9 的矩陣 A_i，將這些 A_i 結合成一個 $2n\times 9$ 的矩陣 A 。

（3） 將 A 做SVD分解，得到 $A = UDV^T$ ，其中 D 為 A 的singular value所構成的對角矩陣（diagonal matrix）、V 為singular vector所構成的正交矩陣（orthogonal matrix）。而我們所要求的 H 就是最小的singular value所對應的singular vector，一般而言，SVD分解後的 D 會將singular value由大到小排列，也就是說 V 中第9行的九個元素即為構成 H 的元素。

（4） V 的第9行為：$\begin{bmatrix} h_1 & h_2 & h_3 & h_4 & h_5 & h_6 & h_7 & h_8 & h_9 \end{bmatrix}^T$，透過式（3.4），可以從 V 的第9行得到 H 。至此，我們已得到H，在接下來的運算中，我們便可以利用（3.2）來將影像平面上的座標轉換到實際空間平面上的座標。

3.2 相關三維幾何運算

在本論文中，我們的系統利用三個平面相交求出交點，用來表示指向物所指的軌跡點，這三個平面分別是：指向物所指的指向平面，以及由攝影機中心與兩個真實世界平面上的座標點所形成的平面（圖 3.3），由於在系統中有兩台攝影機，因此此種平面有兩個，這三個平面就是我們接下來所要探討的三維幾何運算的重點。

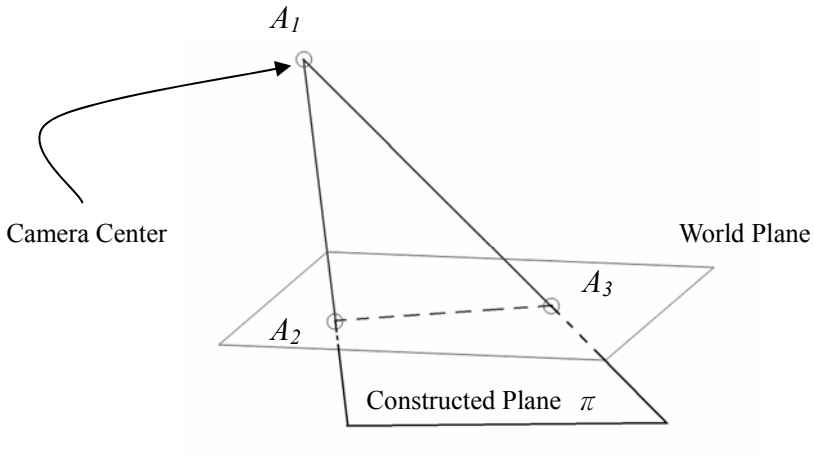


圖 3.3　鏡心與真實世界平面上兩點所形成的平面

3.2.1　三維空間中三點共面方程式

在本系統中，由指向物直線向量在真實世界平面的投影與攝影機中心所共面的平面是軌跡點三維座標重建的關鍵。假設攝影機中心為A_1，而在指向物向量投影上任取兩點A_2與A_3，則可以如圖 3.3 般形成一平面π。三維空間中任一平面可以以$ax+by+cz=d$來表示，其法向量為$\vec{N}=(a,b,c)$。因$\vec{N}$與平面上任一向量垂直，故可以用平面上兩向量$\overrightarrow{A_1A_2}$、$\overrightarrow{A_1A_3}$的外積求得$\vec{N}$：

$$\vec{N}=(a,b,c)=\alpha(\overrightarrow{A_1A_2}\times\overrightarrow{A_1A_3}) \qquad (3.8)$$

其中α為一比例常數。接著再帶入平面上任一點的座標值即可求得d，例如代入$A_1=(x_1,y_1,z_1)$：

$$d=ax_1+by_1+cz_1 \qquad (3.9)$$

依此方式分別求得兩台攝影機與其對應的指向物投影向量所共面的平面之後，再加上指向物所指的指向平面，我們就可以依照接下來的步驟重建出軌跡點的座標。

3.2.2　三維座標中三面共點

在三維空間中，三個平面之間的相互幾何對應關係有以下幾種：

（1）三個平面互相平行

（2）兩平面平行，第三平面與此兩平面各交出一條直線

（3）三平面交於一條直線

（4）三平面兩兩交於一條直線，而這三條直線互相平行

（5）三平面交於一點

如前述，在本篇論文中，我們關注的為第五種情形，也就是三平面交於一點的狀態（圖 3.4），此三平面分別為指向物所指的指向平面π_1、攝影機A與其對應的指向物投影向量所共面的平面π_2、攝影機B與其對應的指向物投影向量所共面的平面π_3，而其交點即為軌跡點。

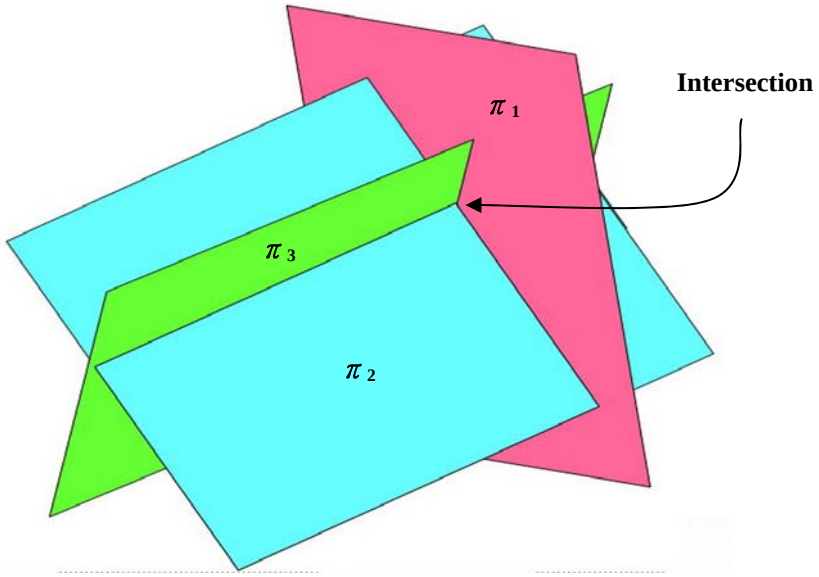


圖 3.4　三平面相交一點的情形

在本論文中，我們使用Cramer’s Rule求解三平面交一點的情形，經過此步驟之後，整個軌跡點重建的步驟便已經完成。

Chapter 4　軌跡辨識

在本論文中，我們希望發展一個即時且可擴充性高的軌跡辨識系統，在傳統研究上，有圖樣匹配 （Pattern Matching）、統計方法分類器（Statistic Classifier）以及類神經網路（Neural Network）等方法用以解決圖樣辨識的問題，其中類神經網路的優點在於當待辨識物體不具明顯統計特性時，仍可經由訓練的方式得出合適的決策函數，因此被廣泛地應用於樣式分類（Pattern Classification）上。本論文中所探討的重建軌跡便具有此種特性，因此在本章中，我們將探討類神經網路應用於指向軌跡之辨識的原理及設計。

4.1 類神經網路

4.1.1　神經元（Neuron）

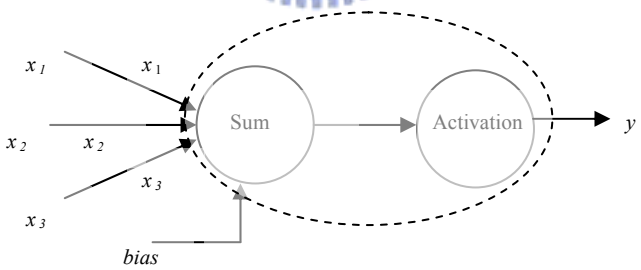


圖 4.1 類神經元示意圖

類神經網路系統是一種可以擷取、儲存並且應用已知經驗知識的系統，應用於實作上的類神經網路系統包含了多個相互連接的類神經元（Neuron）。類神經元是類神經網路的基本元件，如圖 4.1 所示，每個類神經元由三個部份所組成：偏差修正（Bias），輸入值（Input，圖 4.1 中之x_i），以及輸出值（Output，圖 4.1 中的y），其中每個輸入被配以一個權重（Weight，圖 4.1 中的w_i），用以代表此輸入值對於此神經元的重要性，而輸出值取決於所有輸入值乘上權重後與偏差值的

加總（此步驟圖 4.1 中的Sum函式所執行），當此加總值大於一門檻時，輸出值被觸發（Fired），否則則保持靜態（Quiet）。

4.1.2 類神經網路架構

當類神經元定義之後，便可將上述的類神經元分層組織成類神經網路。在類神經網路系統中，最重要的議題為依系統所需選擇合適的架構，常見的架構有倒傳導網路（Back-propagation Network）、霍普菲爾網路（Hopfield Network）、輻射型基底函數網路（Radial Basis Function Network）等，在本論文中，我們選擇應用最為廣泛，架構於多層前授架構的倒傳導網路（Back-propagation Network）為我們的系統架構。

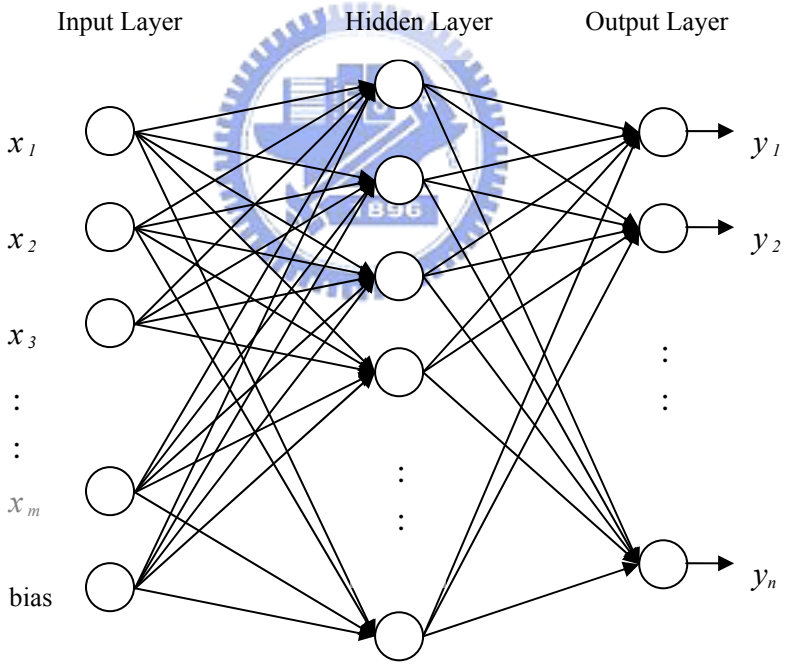


圖 4.2 多層前授網路示意圖，層級為三

多層前授網路一般由三到四層的類神經元所組成。圖 4.2 表示了一個層級為三的多層前授全連通網路，分別由輸入層（Input Layer，並未導入 Sigmoid 函式做為觸發函式，僅單純地輸出其原始值到隱藏層），隱藏層（Hidden Layer）以及輸出層（Output Layer）所組成。由於其在各層之間全連通的特性，故其中每一

層分別會接受其前一層全部的類神經元所傳遞過來的資訊。

4.1.3 訓練學習

在未經過訓練學習的步驟前，類神經網路的輸出值為混亂的雜訊，故其輸出層沒有實質意義。類神經網路藉由調整每一層中類神經元的權重來使輸出層中的每個輸出盡可能地接近所要求的目標值，隨著學習次數增加，輸出誤差會趨向收斂，而漸漸接近目標值。在多層前授網路的架構下，倒傳導演算法（Back-propagation Algorithm）是最常見的訓練方法，倒傳導演算法改進了感知元 Delta 規則（Perceptron Delta Rule）無法應用於多層網路解非線性解的情形，其中在單層網路時，權重誤差函式可以表示如下：

$$\Delta w_i = x_i \delta \quad \text{其中 } \delta = (\textit{desired output}) - (\textit{actual output}) \qquad (4.1)$$

其中 w 是權重陣列，而 x 是輸入陣列，當最後 Δw 的值趨於收斂時，便完成單層網路的訓練。

然而當此法用於多層網路時，我們並無法知道單一層的收斂對其他層會有什麼影響。在倒傳導演算法中，利用對一線性函式微分可以得其斜率的特性，我們將權重總和導入 Sigmoid 函式（4.2），使得觸發函式成為一個非線性、連續並且可微分的函數，如圖 4.3 所示。

$$o = \sigma(\vec{w}\vec{x}) = \frac{1}{1+e^{-\vec{w}\vec{x}}} \qquad (4.2)$$

從微積分的觀點來看，我們藉由針對此函式中 w 的部份偏微分，可以得到誤差的趨勢，此觀點可以分成兩部分來討論：

（1） 若此導數為正，代表當我們增加 w 時，誤差的改變率增加，若將此導數乘上一負值加到權重值，使 w 減小，則可以預期誤差將減小，進一步到達局部最小值；

（2） 反過來看，若此導數為負，則代表當我們增加 w 時，誤差的改變率減小，則我們希望將此導數乘上一負數（使其成為一正值），加到權重值，

進一步觀察誤差的改變率在我們增加 w 時是否持續減小。

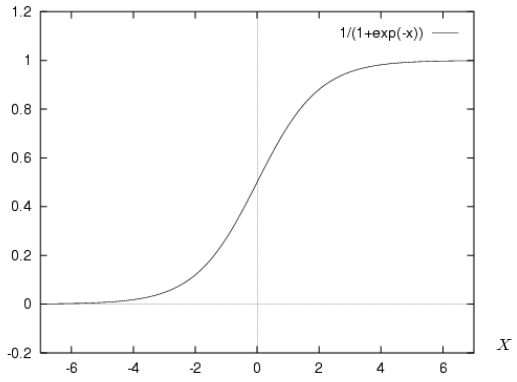


圖 4.3 Sigmoid 函數

導入了Sigmoid函數之後，我們開始計算整體網路平方誤差，首先我們輸入系統予一個至多個的訓練樣本（Training Samples），將輸出層的輸出與目標值比較，如下所示，其中t_k為目標值，o_k為此次輸出值：

$$E(\vec{w}) = \frac{1}{2}\sum_{k\in output}(t_k - o_k)^2 \qquad (4.3)$$

接著我們計算輸出層類神經元的誤差部份，如下所示：

$$\delta_k = o_k(1-o_k)(t_k - o_k) \qquad (4.4)$$

其中δ_k即為輸出層各元件的誤差，接著我們計算隱藏層的誤差值，如下所示：

$$\delta_h = o_h(1-o_h)\sum_{k\in outputs} w_{kh}\delta_k \qquad (4.5)$$

從（4.5）可以看出，隱藏層的誤差（δ_h）與輸出層的誤差（δ_k）息息相關，此即為倒傳導的重要特性。最後我們回過頭來修正各層權重，首先我們必須先得出權重誤差，如（4.6）所示，其中p為層數，由 1 到 3 分別代表輸入層、隱藏層、輸出層，而η為學習率。較低的學習率可以穩定地到達收斂，而較高的學習率可以加快收斂的速度：

$$\Delta w_i = \frac{\eta}{3}\sum_{p=1}^{3} x_{ip}\delta_p \qquad (4.6)$$

為了使式（4.6）更適合於使用程式迭代運算，我們將（4.6）修改如下：

$$\Delta w_{pi} = \eta x_{pi}\delta_p \qquad (4.7)$$

最後就可以將上述的誤差用來修正各層的權重。至此，倒傳導的修正步驟便建立

完成，當將此更新過的權重值作用在類神經網路上後，若可以使（4.3）的誤差小於一門檻值，便完成訓練的步驟：

$$w_{pi} \leftarrow w_{pi} + \Delta w_{pi} \qquad (4.8)$$

4.2 軌跡辨識之應用

在本論文中，我們需要發展一套具有可擴充性的指向軌跡辨識系統，在傳統的多層前授網路中，輸出層的類神經元個數是固定的，也就是說，一旦系統訓練完成，便只能接受一定種類數目的軌跡樣式，當使用者希望加入新的軌跡樣式時，必須針對所有的軌跡樣式重新訓練，因此我們採用串列多層前授網路（Parallel Multi-layer Feed-forward Network）作為我們系統的架構，此種架構中，每一個在串列多層前授網路中的類神經網路單元僅產生一個輸出，經由輸出比較器決定最後的判定結果，由於串列具有可以擴充增加的特性，我們可以在不重新訓練已知網路單元的情形下增加新的類神經網路單元。在此我們將探討串列多層前授網路應用於系統實作時所帶來的優點。此外，針對類神經網路需要固定個數輸入的特性，我們發展軌跡補償的功能，使得系統在軌跡點個數不論在大於輸入值個數或小於輸入值個數時都能得到一固定數目的特徵點。

4.2.1 串列多層前授網路

如圖 4.2 所示，傳統多層前授網路包含一個以上的輸出，在系統設計之初就已經固定了輸出層類神經元的個數，若要增加（或減少）輸出層類神經元的個數，我們必須針對整個類神經網路重新訓練，以得到合適的決策權重值。此種架構的缺點在於我們在增加一個未知的待辨識類別時，必須連同已知的類別一起重新訓練，不利於系統的擴充性，並且在整體系統的維護上也較僵化。

在串列多層前授網路中，每一個個別的網路元件都僅有一個輸出層類神經元，因此每一個個別網路元件都僅單獨負責辨識一種軌跡，當使用者欲增加一種

軌跡樣式時，僅需新增一個類神經網路單元，加以訓練，便可以在不變更類神經網路單元初始架構的情形下辨識新的軌跡輸入。輸入會被分別傳入每種軌跡樣式的類神經網路辨識單元，利用各個類神經網路辨識單元辨識輸入軌跡與軌跡樣式的符合程度，當得到每個輸出單元的結果之後，我們比較各輸出單元的值，取其輸出值最大者為我們軌跡辨識的結果(此動作由圖 4.4 中 Comparator 這個動作元件所執行)。

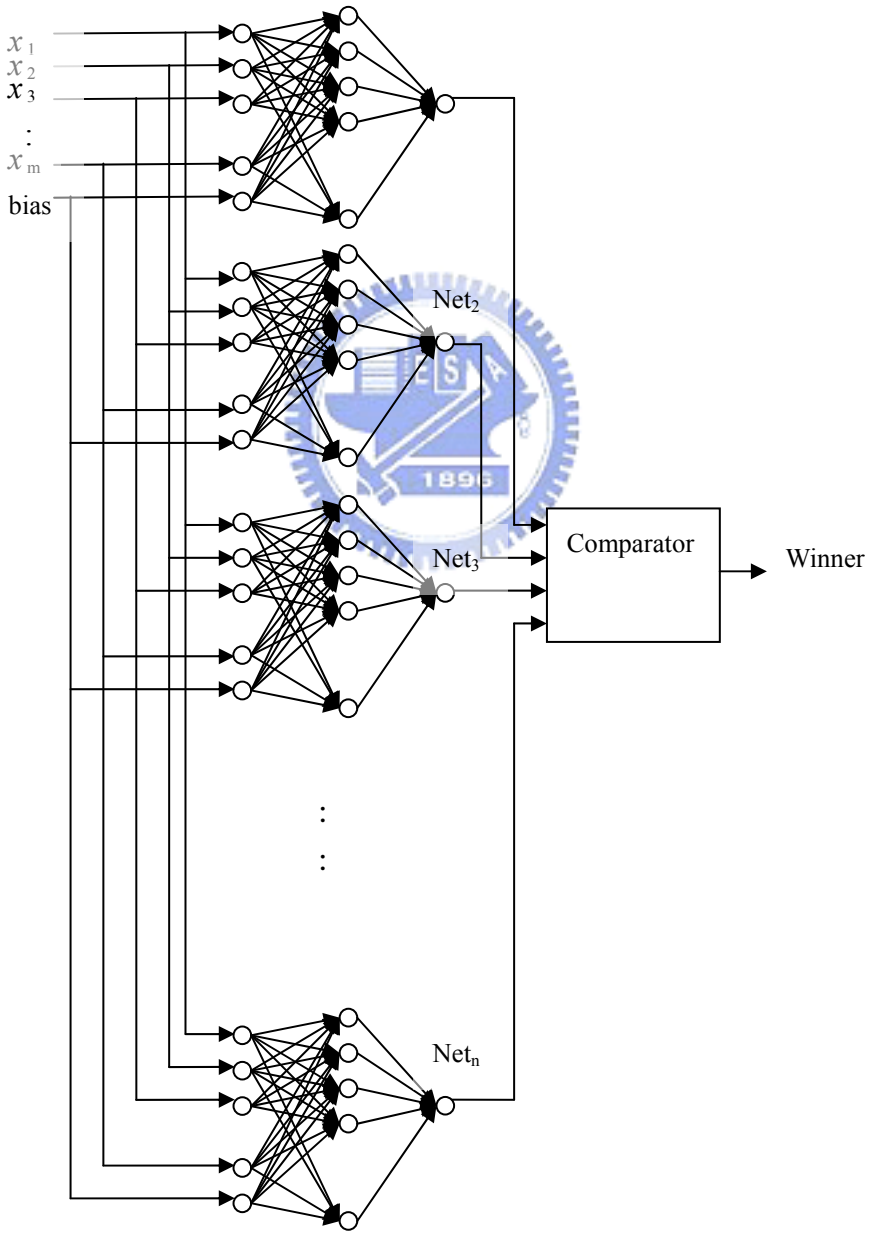


圖 4.4 串列多層前授網路架構圖

4.2.2 軌跡補償

在我們的指向系統中，軌跡並非均勻分布，如下圖 4.5 所示的 V 字圖樣，在剛進入重建（左上角）與轉折處（V 字底轉折處）由於使用者的停頓，造成重建點的密集分佈，然而其他地方則呈稀疏分布狀，此種因輸入而造成的特徵分布不均會使得輸入點無法有效代表整體軌跡的特徵。此外，類神經網路有輸入層類神經元個數固定的限制，當輸入的特徵點個數多於或少於輸入層類神經元個數時都會因為無法吻合輸入層類神經元個數而無法辨識（圖 4.6）。由於以上兩種原因都會造成類神經網路無法有效辨識輸入的軌跡，因此為了使整體輸入具有軌跡的代表性，並且將軌跡特徵點正規化（Normalize）到整條軌跡上，我們從整條軌跡中均勻截取用來輸入給類神經網路的特徵點，以使得類神經網路能夠有效地辨識出使用者輸入的軌跡。

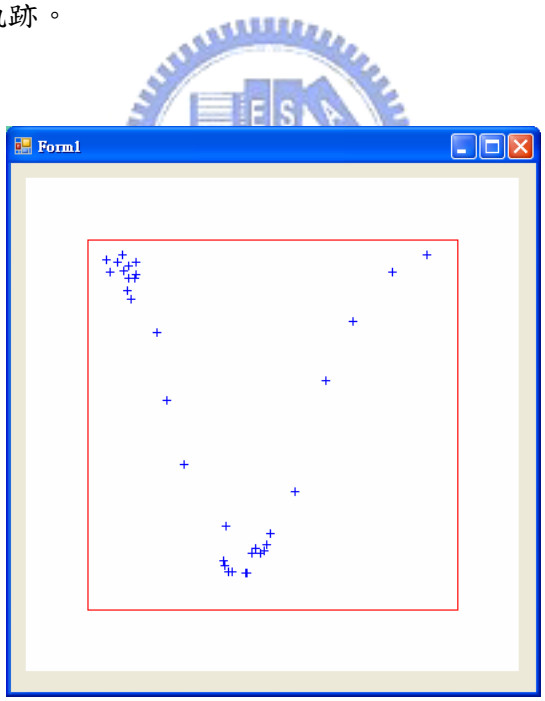


圖 4.5 一分佈不均的軌跡圖，呈 V 字型

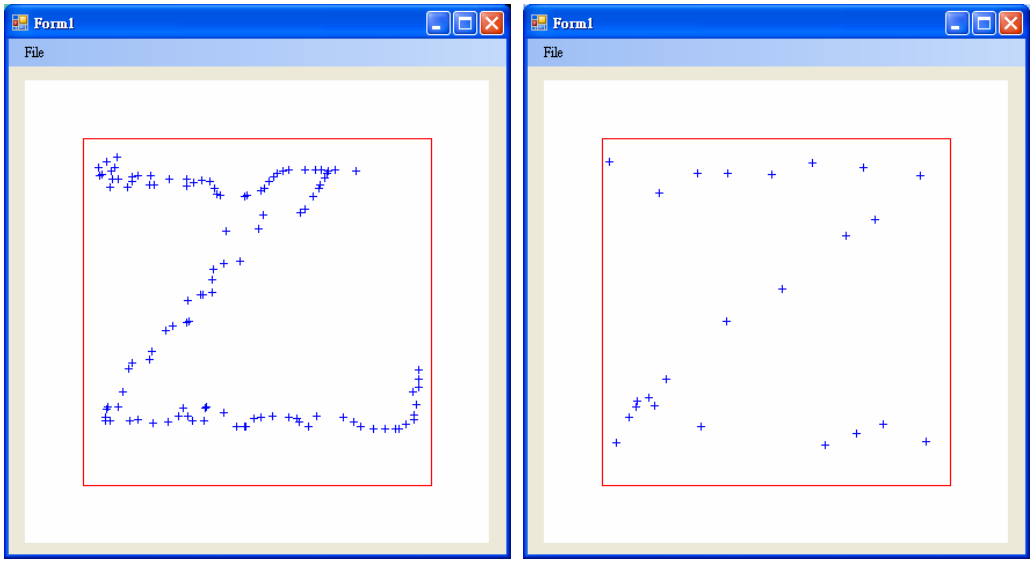


圖 4.6　兩種輸入點個數不符合輸入層類神經元個數的例子，在本系統中輸入層類神經元個數為 48 點，然而左圖軌跡為 76 點(過多)，右圖為 19 點(不足)

軌跡均化補償的步驟首先必須先依序算出點與點之間所有的折線內插點，首先先求得兩個在時序上連續的點(x_{p-1}, y_{p-1})及(x_p, y_p)之間的直線方程式$y = mx+e$：

$$m = (y_p - y_{p-1})/(x_p - x_{p-1})$$
$$e = y_p - x_p \times m \qquad (4.9, 4.10)$$

接著取出此線段上介於兩端點之間的所有內插點，將其更新到所有內插點的集合P：

$$P \leftarrow \left\{ x_{pi}, y_{pi}; where\ x_{pi} \in \left[x_p, x_{p-1} \right], y_{pi} = m \times x_{pi} + e \right\} \qquad (4.11)$$

在經過此步驟之後，所有的內插點都已經求得，我們在集合P裡面均勻地取M點，將其更新到集合P_c中，P_c即為整條軌跡經過均化補償之後的結果：

$$P_c \leftarrow \left\{ x_k, y_k;\ where\ x_k,\ y_k \in P\ and\ 1 \le k \le M \right\} \qquad (4.12)$$

均化補償的結果如圖 4.7，圖左在經過補償前特徵較為分散不均，在 Z 的底部有大量空白的區域，圖右在經過補償之後，軌跡的特徵可均勻地由特徵點看出，此段軌跡便可以用來進一步地擷取資訊，並輸入到類神經網路中。由於此步驟會在整條軌跡中均勻地取 M 點，因此不論在軌跡點大於 M 點或小於 M 點的情形下，均能完整重建出一 M 點的軌跡（圖 4.8）。

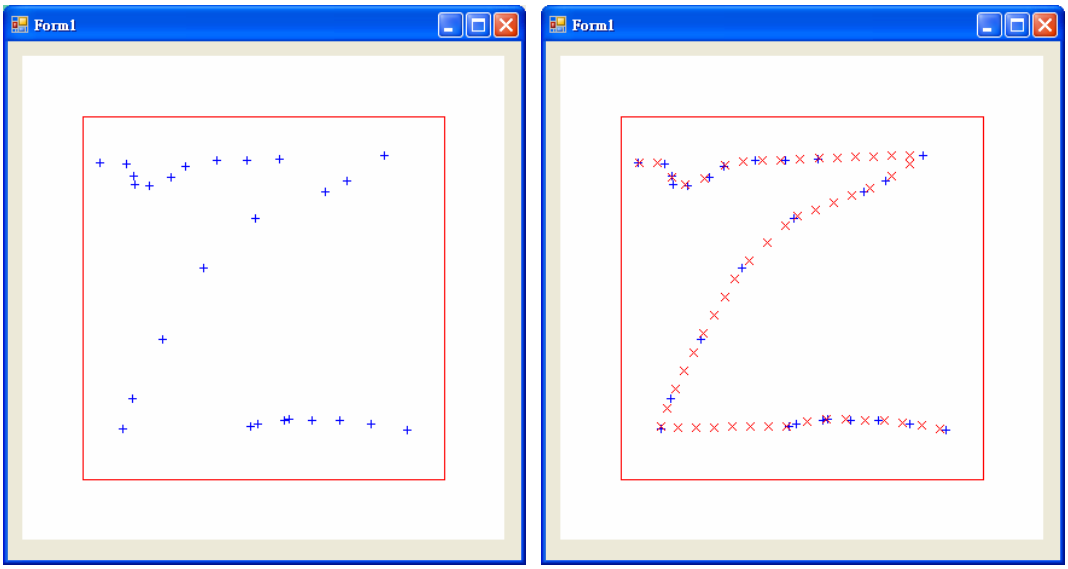


圖 4.7　軌跡補償結果，圖左為補償前，圖右為經過均化補償之後的結果，特徵點為 48 點，與輸入層類神經元個數相同

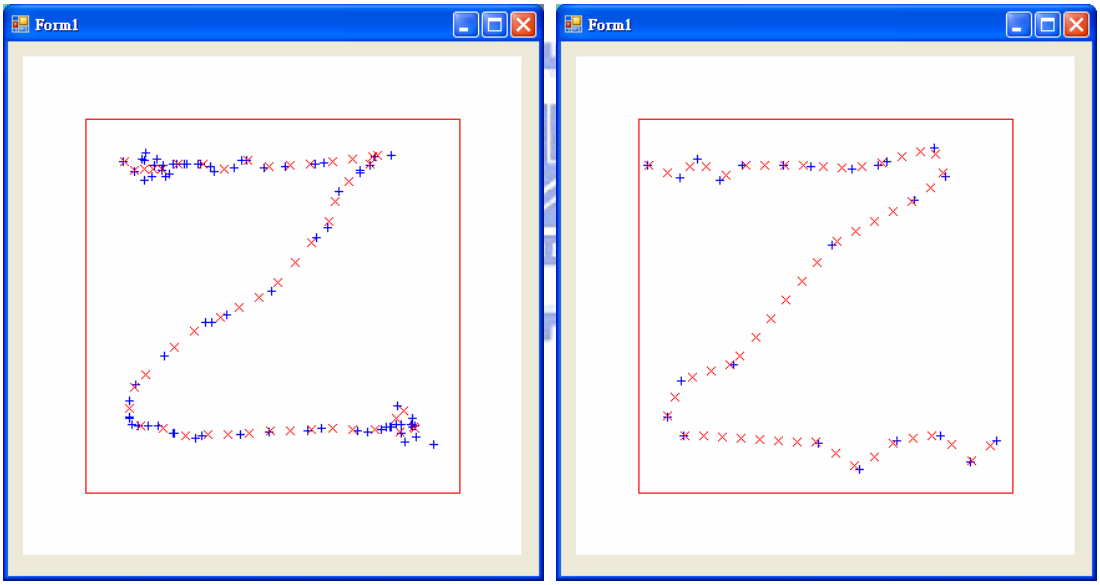


圖 4.8　兩個軌跡點（藍色的點）不符合輸入層類神經元個數的例子，左圖為 68 點，右圖為 25 點，可以看到經過補償之後（紅色的點）均為 48，與輸入層類神經元個數相同

4.2.3　特徵值

由於上述的代表點在二維座標系上會有尺度（Scale）與位偏的問題，因此我們必須將其轉換為基於角度的特徵值，在此我們取兩點之間的角度差作為我們的特徵，如圖 4.9 所示。

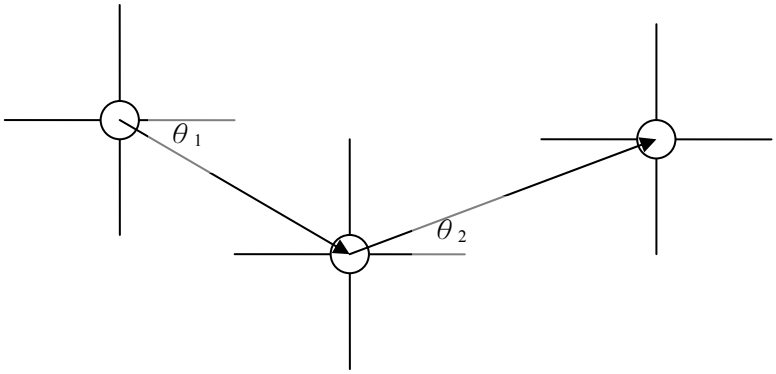


圖 4.9　時序上相連續的點與點之間的角度差

我們首先求得點與點之間的角度，如（4.13）所示：

$$\theta_p = radius\left(y_p - y_{p-1},\ x_p - x_{p-1}\right) \qquad (4.13)$$

然後將此角度的 sine 與 cosine 做為我們用來輸入給串列多層前授網路的輸入值 x，其中 M 為前述的輸入特徵點數量：

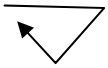
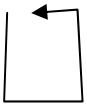
$$\begin{cases} x_{2m} = \sin\theta_m \\ x_{2m+1} = \cos\theta_m \end{cases} \quad 0 \le m \le M \qquad (4.14)$$

4.2.4　辨識率

當我們決定一種類神經網路架構之後，最重要的考量即為在此種架構下是否能夠達成高辨識率的目標。為了說明本系統的軌跡在串列多層前授網路架構下的辨識率，我們舉出一個在七種軌跡樣式下的範例，並針對每種軌跡樣式連續輸入一百個輸入，分析經由類神經網路辨識出來的軌跡樣式種類是否與輸入相符，以計算其辨識率，結果如表 4.1 所示。從表中可以看出，手勢的有效辨識率介於 82% 至 99%之間（多數在 90%以上）。

表 4.1　使用串列多層前授網路架構下七種手勢的辨識率

Input / Output							
	94%		5%	1%	4%		

		99%			9%	1%	
			93%	9%			3%
				90%	1%	1%	
	5%	1%	2%		82%		7%
						98%	2%
	1%				4%		88%

Chapter 5　系統實作

本論文在基於雙視角的立體視覺環境下發展了一套即時且具靈活擴充性的人機互動應用介面，其目的在定義一套使用者與電腦都能了解的軌跡樣式，使得每當使用者比劃出一個軌跡時，電腦都能判定軌跡的類別，進而將此類別所對應的動作反饋到使用者介面上。此外，我們將使每一個使用者都可以定義自己的軌跡樣式，增加系統使用上的友善性。在實作上，我們將發展一組基於類神經網路、有限狀態機、以及可擴充自訂指令模組的反應機制，使得本系統能夠實踐在包含簡報系統，網頁瀏覽系統在內的多項應用。本章節總結論文中關於實作的部份，並且提出幾種在本系統環境下的應用可能性。

5.1 硬體環境

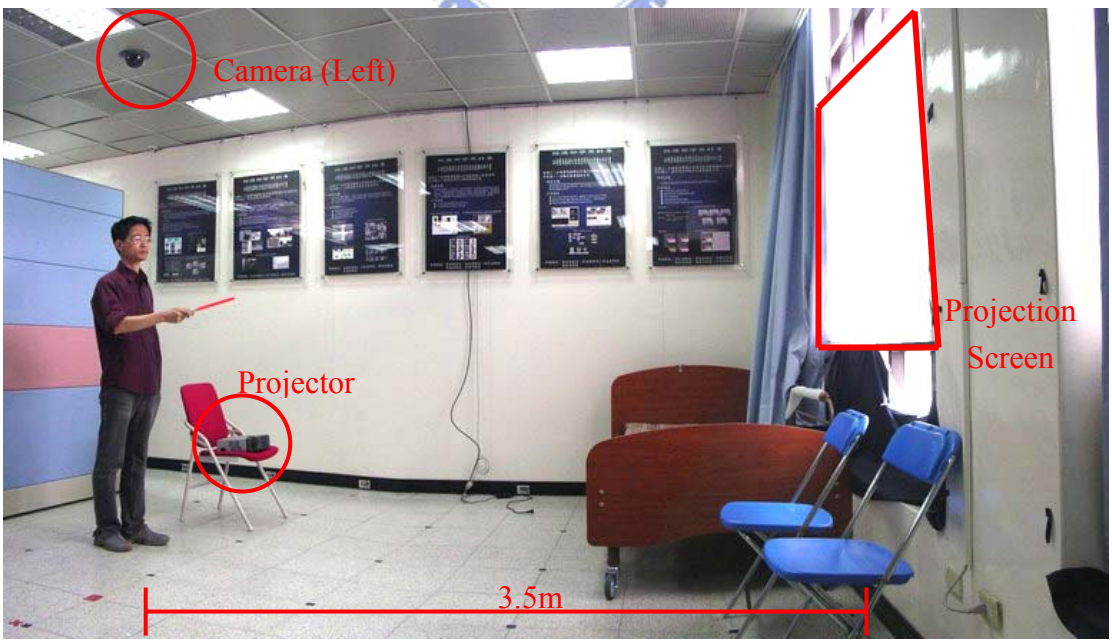


圖 5.1 系統運作空間實景

5.1.1 系統運作空間

本系統的運作場景為一個含有投影布幕、投影機、以及兩台攝影機的空間，如圖 5.1 所示。使用者站在兩台攝影機的取像範圍內（視角大小可參考圖 5.2），利用第二章所描述的指向物對著投影布幕比劃軌跡（此投影布幕上是被投以作業系統的運作影像），此軌跡點的平面位置會被重建並以滑鼠游標的方式移動，當使用者畫出已定義的軌跡時，整體系統便會對此軌跡作反應，並即時反映在使用者程式介面上。

圖 5.2　不同攝影機視角所拍得畫面，左圖為使用者左方之攝影機，右圖為使用者右方之攝影機

5.1.2 硬體設備規格

在本論文的系統實作中，我們利用表 5.1 中的各項硬體來實踐攝影，影像擷取輸入，以及運算處理。

表 5.1　實作系統所使用之硬體規格表

項目	規格	附註
PTZ 攝影機	**BXB 7720 高速智慧球型攝影機** P/T/Z：360°/90°/16X Electronic Shutter：1/60~1/30k sec Minimum Illumination：1 lux	共兩臺，分別設置在使用者左右方
影像擷取卡	**ADLINK Angelo RTV 24 影像擷取卡** Format：NTSC Resolution：CIF（320×240） Framerate：30fps	此系列擷取卡支援四個輸入

主機	**PC** CPU：Pentium4 3.2GHz RAM：2.0GB OS：Windows XP SP2	

5.2 系統運作

本論文所呈現的為一個從影像擷取到產生反應動作的完整系統，主要可以分成軌跡點重建（見第二章與第三章）、軌跡辨識（見第四章）以及目標系統控制三個主要的部份，系統的完整流程圖呈現如圖 5.3。在人機互動系統的應用上，控制部份的設計良劣與否影響使用者觀感最甚，一個設計簡潔且穩定的控制元件會使得使用者在使用上更感親切且直覺易懂。此外，為了使得本系統在應用上有擴充的可能性空間，我們再將控制的部份細分出控制層（由有限狀態機組成，負責與類神經網路溝通，獲得軌跡樣式後往指令層送出訊息，等待指令層傳回的指令）與指令層（負責在不同的應用中對不同的應用程式下各式的指令）的部份，本節描述在實作上關於系統控制的各個細節。

5.2.1 軌跡樣式

在實作上，定義一套直覺且簡單的軌跡樣式有助於使用者快速上手此系統，圖 5.4 中我們列出幾種可為我們系統中類神經網路所辨識的軌跡樣式範例，如第四章所述，軌跡點的集合在被送往類神經網路判斷之前會先經過補償的動作，也因此無論此次在投影區域內的軌跡點是多少，我們皆可以確保有一固定數量的軌跡點被送往類神經網路做正確的判斷。圖 5.5 表示一個真實情形下的 Z 字型軌跡，在系統中，此種軌跡的樣式為 Select，因此我們希望在輸入軌跡點之後，整條軌跡能夠正常地被判定為 Select 樣式。經過 4.2.4 節的說明，我們可以了解此種軌跡的辨識率約有九成，在圖 5.5 的範例可以正常地被判定為一個 Select 樣式，由圖所見，即使經過補償，整條軌跡仍具有許多不規則的特徵誤差（此類誤

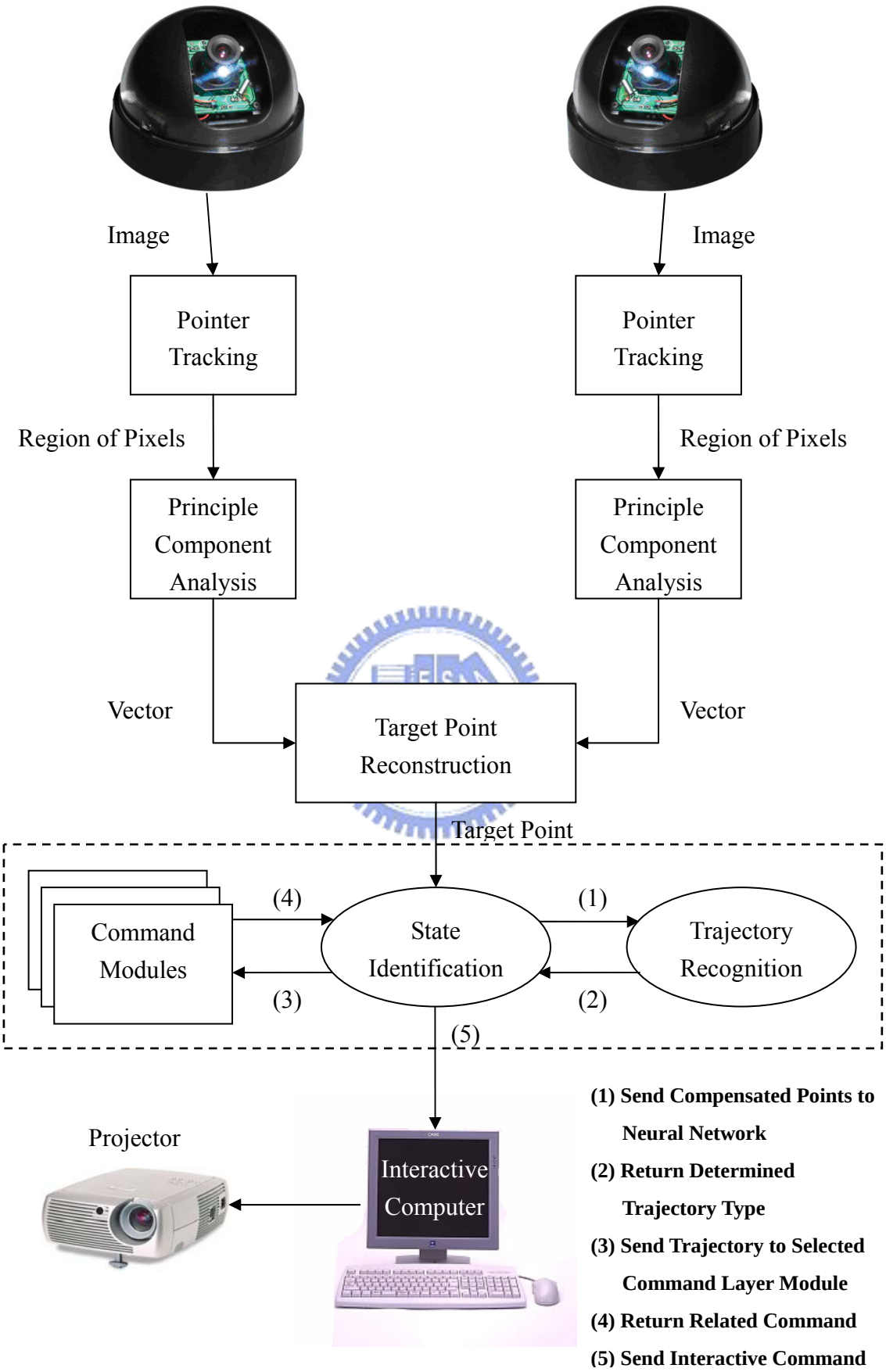


Image
Image
Pointer Tracking
Pointer Tracking
Region of Pixels
Region of Pixels
Principle Component Analysis
Principle Component Analysis
Vector
Vector
Target Point Reconstruction
Target Point
Command Modules
(4)
State Identification
(1)
Trajectory Recognition
(3)
(2)
(5)
Projector
Interactive Computer
(1) Send Compensated Points to Neural Network
(2) Return Determined Trajectory Type
(3) Send Trajectory to Selected Command Layer Module
(4) Return Related Command
(5) Send Interactive Command

圖 5.3　系統流程圖

差來自於重建或追蹤），然而仍然能夠被正常地有效辨別，由此可見使用類神經網路可以有效地解決在系統前端來自追蹤以及重建這兩個部份的誤差，進一步辨識軌跡。此外，由於在我們的系統中採取了串列多層前授網路架構，因此軌跡樣式的數目是可自由擴充的，當使用者在發展新應用時，如果發現想使用的動作沒有合適的手勢來配合時，可以在不影響原先架構的情形下增加軌跡樣式，另外，在測試階段時也可以自由地刪減不合適的軌跡樣式，以符合系統維護的簡潔性。

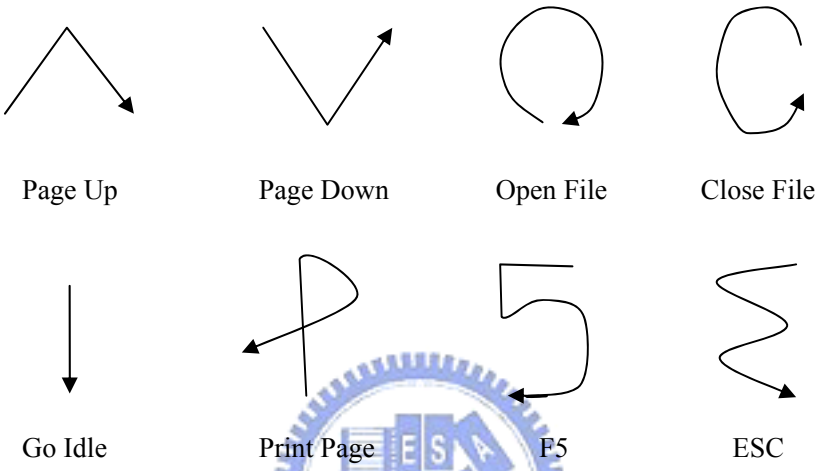


圖 5.4　節選系統中的既定軌跡樣式

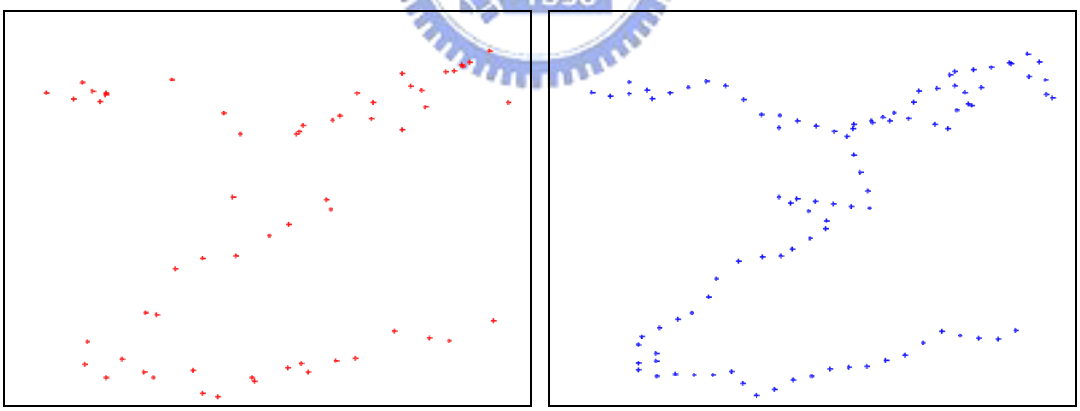


圖 5.5　一個真實系統中的軌跡範例，圖左為未經補償的狀態，

圖右為經補償動作之後的結果，此軌跡範例能夠被正常地辨識

5.2.2　指令層

在系統的實作上，我們將指令依應用分類，這麼做的主要目的是避免同一種應用有太多種軌跡樣式，影響類神經網路判斷的正確性，因此決定一組合適的指

令組合便成為實踐各種應用系統時最主要的考量。舉例來說，當一位使用者在使用簡報系統時，他可能不會考慮使用如 Print 這樣的功能，而當另一位使用者在使用網頁瀏覽系統時，他可能沒有需要用到如 ESC 鍵（PowerPoint 中從全螢幕模式跳回視窗介面模式）這樣的功能，因此我們在不同的系統中使用不同的樣式組合，以增加系統的正確性以及可維護性。

在系統中，透過對各種軌跡樣式定義獨特的名稱，我們可以有效地模組化「軌跡樣式←→動作」之間的一對一關係，表 5.2 展示出三組實作上的對應，當定義完一組指令集之後，整個人機互動介面的程序便告完成，使用者便可以透過系統依其需求進行與電腦之間的溝通。

表 5.2　三組「軌跡樣式←→動作」之間的一對一關係

軌跡樣式名稱	相對應動作
Open File	1. 開啟選擇視窗 2. 跳到選擇狀態，等待使用者輸入欲開啟的文件編號
Go Idle	1. 將系統狀態改到休息狀態，使得有限狀態機單純接收點輸入而不將其送往類神經網路
ESC	1. 對前景(Foreground)文件送出 ESC 鍵

5.2.3　有限狀態機

在我們的系統中，我們使用有限狀態機做為我們控制模組的核心，在此有限狀態機中，最重要的功能就是負責在類神經網路以及動作層間之溝通，從類神經網路得到軌跡樣式後，再將此樣式送往動作層產生動作，最後將動作作用在互動的電腦上，便完成一輪的作業。當使用者對著投影布幕做指向動作的時候，最重要的動作區分的就是指向物軌跡點落在投影布幕內跟落在投影布幕外兩種情形，因此我們依此劃分出有限狀態機的狀態為區域內狀態（Inside State，圖 5.6 左上）與區域外狀態（Outside State，圖 5.6 左下），此外，由於當軌跡點落在投影布幕內時我們可能是想做下面三種動作：

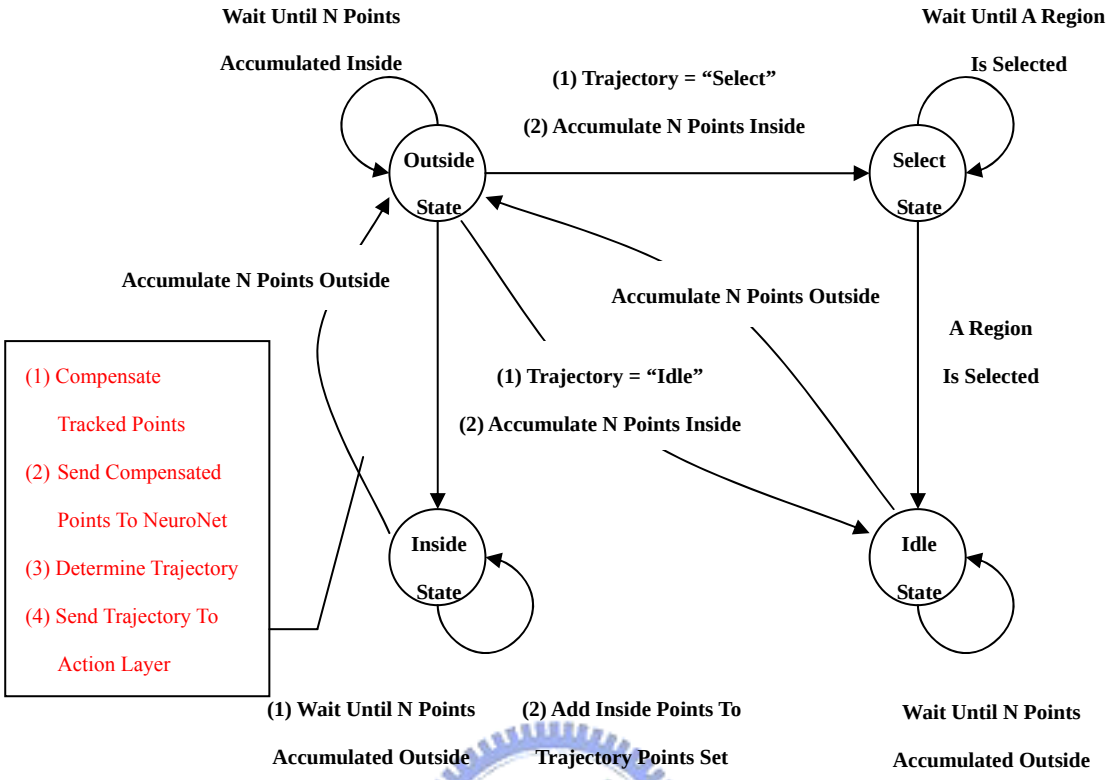


圖 5.6　本系統中有限狀態機示意圖

（1）利用軌跡下指令：當此種情形時，我們會希望我們所有在投影區域內的點都可以被收集，當軌跡畫出區域外時，便可以將收集到的軌跡點送往類神經網路作判斷，並且得到一軌跡樣式回來。此動作的起始條件為軌跡點連續停留在投影區域內 N 個點，且沒有（2）與（3）的動作在等待時，而結束條件則為軌跡點連續停留在投影區域外 N 個點，此時我們知道使用者已經結束軌跡的輸入，便可以開始判斷軌跡。

（2）選擇某個數字：在我們的系統中，數字的選擇由投影區域所劃分，我們將投影區域切割成 2×2 或 3×3 的區域範圍，分別可以代表四個或九個數字，這種動作通常接在得到一個需要選擇的指令之後，例如開啟某個檔案（接下來便利用此種狀態選擇「到底要開啟哪個檔案」），關閉某個檔案，或者列印檔案的某一頁等等。在圖 5.6 中由右上的狀態（Select State）所表示。此種狀態會持續到使用者的軌跡點持續停留在某一個區域內 M 點為止，得到輸入值之後便跳入休息狀態，不再接受輸入或判斷軌跡，直到回到 Outside

State 的事件被觸發。

（3）休息狀態：當有限狀態機停留在此狀態（Idle State）時，被重建的軌跡點除了本身的位置資訊外不代表任何其他資訊(例如上兩例中的數字或者軌跡樣式)，通常當使用者僅是想使用軌跡點介紹畫面中資訊時使用（此時可以把軌跡點當作是滑鼠游標，僅在畫面中游移而不輸入任何訊息)，當此軌跡點移到畫面外並持續停留 N 點之後，我們認為使用者已經介紹完其想表達的資訊，因此整體狀態又回到區域外狀態。

至此，我們以介紹完畢有限狀態機所有的行為模式，我們接下來將介紹兩種可應用於實際系統中的實作雛形。

5.3 實作範例

本系統依不同的需求可以架上不同的指令層模組，在人機互動系統的領域中，傳統上最主要的應用有簡報系統、網頁瀏覽系統以及成果展示系統等，下面我們舉出兩種可能應用系統雛形的指令層概要。

5.3.1 簡報系統

在簡報系統中，使用者關注的是系統是否能讓他流暢地在上下頁之間轉換，甚至是在開啟多檔案的情形下在檔案與檔案之間轉換，因此我們將描述一組可以讓使用者有效地控制簡報檔案的指令組，表 5.3 列出一個簡報系統所需要的指令，圖 5.7 則展示一個展示簡報的範例。

表 5.3　簡報系統所使用的指令集

軌跡樣式名稱	相對應動作	軌跡樣式範例圖示
Open File	1. 開啟選擇視窗 2. 跳到選擇狀態，等待使用者輸入欲開啟的文件編號	
Close File	1. 開啟選擇視窗 2. 跳到選擇狀態，等待使用者輸入欲關閉的文件編號	

Select File (此指令在已開啟多於一個檔案時使用)	1. 開啟選擇視窗 2. 跳到選擇狀態，等待使用者輸入切換的檔案編號	
Go Idle	1. 將系統狀態改到休息狀態，使得有限狀態機單純接收點輸入而不將其送往類神經網路 2. 此指令可將軌跡點單純當做滑鼠使用	
Page Up	1. 對前景(Foreground)文件送出 PGUP 鍵	
Page Down	1. 對前景(Foreground)文件送出 PGDN 鍵	
F5	1. 對前景(Foreground)文件送出 F5 鍵，使其切換到全螢幕狀態	
ESC	1. 對前景(Foreground)文件送出 ESC 鍵，使其脫離全螢幕狀態	

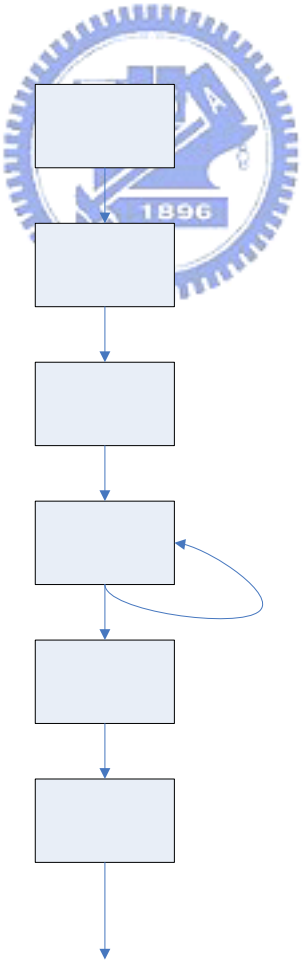


圖 5.7 展示簡報的範例

Open File

5.3.2 特定網頁系統

當人們在展示場空間中使用人機介面瀏覽網頁時，通常在意的不是如何使用鍵盤輸入字母，而是是否能夠藉由人機介面對網頁瀏覽器下簡單的瀏覽指令（如向左、向右、向下等）以及處理網頁（如列印網頁），因此我們將重心擺在這些跟瀏覽器息息相關的輸入指令，下表 5.4 列出一個特定網頁瀏覽系統所會需要的指令，圖 5.8 則展示一個瀏覽網頁並列印的範例。

表 5.4 特定網頁瀏覽系統所使用的指令集

軌跡樣式名稱	相對應動作	軌跡圖樣範例
Open File	1. 開啟瀏覽器選擇視窗 2. 跳到選擇狀態，等待使用者輸入欲開啟的網頁編號	
Close File	1. 開啟瀏覽器選擇視窗 2. 跳到選擇狀態，等待使用者輸入欲關閉的網頁編號	
Select File (此指令在已開啟多於一個檔案時使用)	1. 開啟選擇視窗 2. 跳到選擇狀態，等待使用者輸入切換的網頁編號	
Go Idle	1. 將系統狀態改到休息狀態，使得有限狀態機單純接收點輸入而不將其送往類神經網路 2. 此指令可將軌跡點單純當做滑鼠使用	
Up	1. 對前景(Foreground)網頁送出 UP 鍵	
Down	1. 對前景(Foreground) 網頁送出 Down 鍵	
Print	1. 列印網頁資訊	

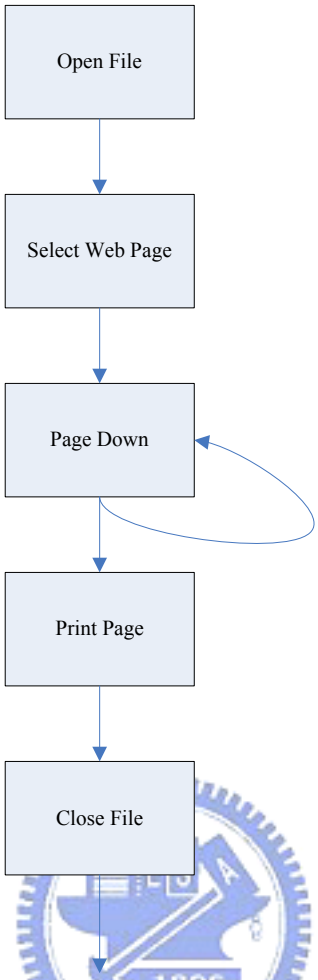


圖 5.8 列印特定網頁的範例

Chapter 6　結論及未來展望

6.1 結論

在本論文中，我們發展並實踐了一套基於立體視覺空間中的即時人機互動系統，並且將研究的重心放在（1）系統介面的友善度、（2）是否能有效偵測使用者行為、（3）系統的可擴充性、以及（4）系統的穩定性這四個方面。在追蹤指向物方面，我們利用指向物的色彩、形體以及運動等特性來從影像中分析指向物，並且利用主成分分析法將像素資訊轉換成方向向量，在本篇論文中，我們基於已知的空間資訊，利用投影轉將指向物方向向量從影像平面轉換到世界座標平面中，並將其與攝影機中心建立一平面，當此平面建立後，我們可以利用兩台攝影機的重建平面與投影平面三面共點的特性，重建出指向軌跡點在真實世界中的位置。接著我們利用類神經網路分析軌跡樣式，並且建立一套控制模組，以使得使用者可以藉由輸入特定軌跡來對電腦下特定指令來達成人機互動的目的。

在研究重心的發展上，我們藉由導入類神經網路的概念來使得使用者可以輕易地與電腦溝通，達成人機介面程式中友善性的要求，並且發展出基於指向物形體特性的追蹤與重建方式有效地偵測了使用者的行為，接著我們設計了一套基於可更換動作模組的控制元件來達成系統的高擴充性，最後，我們利用有限狀態機受規範的行為模式來達成系統的穩定性。

6.2 未來展望

一個理想中的人機互動應用是自然且沒有使用者人數限制的，在未來的發展中，我們希望能夠發展出可以接受更多種指向物種類的系統，並且利用各指向物獨特的外觀特性（例如不同的形體，不同的色彩）來達成基於立體視覺下的多人

互動系統。此外，一個優秀豐富的人機互動介面有賴於更多種應用程式的支援，在可見的未來，必定會有更多相關的研究著眼於人機互動介面的應用性，這也是我們未來研究的重點之一。

參考文獻

[1] C. Kirstein and H. Müller, “Interaction with a Projection Screen Using a Camera-Tracked Laser Pointer,” *Proc. of International Conference on Multimedia Modeling*, IEEE Computer Society Press, pp. 191-192, 1998.

[2] R. Sukthankar, R. Stockton, and M. Mullin, “Smarter Presentations: Exploiting Homography in Camera-Projector Systems,” *Proc. of International Conference on Computer Vision*, pp. 247-253, 2001.

[3] X. Chen and J. Davis, “LumiPoint: Multi-user Laser-Based Interaction on Large Tiled Displays,” *Stanford CS Technical Report TR-2000-04*, 2000.

[4] Y.-P. Hung, Y.-S. Yang, Y.-S. Chen, I.-B. Hsieh, and C.-S. Fuh, “Free-Hand Pointer by Use of an Active Stereo Vision System,” *Proceedings of 14th International Conference on Pattern Recognition*, pp. 1244-1246, 1998.

[5] C. Leubner, C. Brockmann, and H. Müller, “Computer-vision-based Human-Computer Interaction with a Back Projection Wall Using Arm Gestures,” *Proc.s of 27th Euromicro Conference, Warsaw*, IEEE Press, pp. 308-314, 2001.

[6] G.V. Paul, G.J. Beach, and C.J. Cohen, “A Realtime Object Tracking System Using a Color Camera,” *Applied Imagery Pattern Recognition Workshop*, pp. 137-142, 2001.

[7] Y. Wu and T.-S. Huang, “Nonstationary Color Tracking for Vision-based Human-Computer Interaction,” *IEEE Trans. Neural Networks*, Vol. 13, Issue 4, pp. 948 - 960, July 2002.

[8] H. Mei, A. Sethi, H. Wei, and G. Yihong, “A Detection-based Multiple Object Tracking Method,” *International Conference on Image Processing*, pp. 864-871, 2004

[9] S.K. Pal and S. Mitra, “Multilayer Perceptron, Fuzzy Sets, and Classification,” *IEEE Trans. Neural Networks*, Vol 3, Issue 5, pp. 683-697, September 1992.

[10] C. Hummels and P.J. Stappers, “Meaningful Gestures for Human Computer Interaction: Beyond Hand Postures,” *Proc. of 3rd IEEE International*

Conference on Automatic Face and Gesture Recognition, pp.591-596, 1998

[11] R.B. Stone, "Designing Screen-based Interfaces for Advanced Multimedia Functionality," *Proc. of 6th International Conference on Information Visualisation*, pp. 611-616, 2002.

[12] J. Segen and S. Kumar, "Human-Computer Interaction Using Gesture Recognition and 3D Hand Tracking," *Proc. of International Conference on Image Processing*, 1998

[13] C. Stary, T. Riesenecker-Caba, and J. Flecker, "User Interface Evaluation: a Comparison of 18 Techniques When Implementing the EU-Directive on Human-Computer Interaction," *Proc. of 6th Australian Conference on Computer-Human Interaction*, pp 184-193, 1996

[14] H.C. Longuet-Higgins, "A Computer Algorithm for Reconstructing a Scene from Two Projections," *Nature*, vol. 293, pp. 133-135, 1981.

[15] R. Hartley and A. Zisserman, *Multiview Geometry In Computer Vision*. Cambridfe University Press, 2001.

[16] O. Faugeras and Q.-T. Luong, *The Geometry of Multiple Images*. The MIT Press, 2001.