國立臺灣師範大學應用電子科技學系

碩士論文

指導教授：許陳鑑 博士

以軟硬體協同設計之目標物移動方向模糊辨識系統

Hardware/Software Co-Design of a Fuzzy Moving Direction Identification System

研究生：朱書漢 撰

中 華 民 國 一百零一年 七月

以軟硬體協同設計之目標物移動方向模糊辨識系統

學生：朱書漢　　指導教授：許陳鑑　博士

國立臺灣師範大學應用電子科技學系碩士班

摘　　要

本論文利用 Altera DE2-70 實驗板以及模糊邏輯理論，提出一種以軟硬體協同設計之模糊目標物移動方向辨識系統，在 SOPC (System on Program Chip, SOPC)之系統架構下，利用 FPGA (Field Programming Gate Array)的硬體電路優勢，以軟硬體協同設計 (HW/SW Co-design)之方式實現硬體加速之功能。作法上係利用硬體電路獲得目標物影像之歷史軌跡，並用 Nios II 計算上、下、左、右方向的計數，再將此計數傳送到模糊邏輯推論的硬體電路，即可得到目標物體的移動方向。實驗結果證實本方法可以利用機器視覺來辨識目標物的移動方向，實現互動式之人機介面，達到控制機器的運作。論文整體架構大致上可分為： (1)以軟體實現此系統的演算法、 (2)將此系統實現於 Altera DE2-70 開發版之晶片設計，希望利用硬體平行處理的優勢，加速其演算法的速度。

關鍵字：移動估測、模糊理論、FPGA、光流、電腦視覺

Hardware/Software Co-Design of a Fuzzy Moving Direction Identification System

Student：Shu-Han Chu Advisor：Dr. Chen-Chien Hsu

Institute of Applied Electronics Technology

National Taiwan Normal University

ABSTRACT

In this thesis, a hardware and software co-design approach is proposed to develop a fuzzy moving direction identification system using the Altera DE2-70 board and fuzzy logic theory. Under the system architecture of System on the Program Chip (SOPC), we take advantages of the framework of hardware and software co-design, where hardware circuits by Field Programming Gate Array (FPGA) are designed to accelerate the system performance of the historical trajectories of the target image, while direction counts are calculated by the Nios II CPU. A hardware circuit is also design to identify the moving direction of the target object. Experimental results show that the proposed method is able to identify the movement direction of the target direction, providing an interactive man-machine interface to control the operation of the machine. The contents of this thesis can be divided into: (1) proposed algorithm and its implementation by software (2) hardware and software co-design of the proposed algorithm on the Altera DE2-70 development board to accelerate the execution of the proposed algorithm.

Keywords: Motion Estimation, Fuzzy Theory, FPGA, Optical Flow, Computer Vision.

誌　　謝

首先，學生要誠摯的感謝我的指導教授許陳鑑老師對我的耐心指導與教誨，本篇論文才能夠順利的完成，老師對於研究嚴謹的工作態度與做人做事的哲理以及在生涯規劃方面老師更是花了諸多苦心。在此致上最真誠的敬意。同時要感謝口試委員翁慶昌教授、李宜勳教授、王銀添教授與李世安教授，對於論文提出許多寶貴的意見與指正，使得本篇論文能夠更加完善。

在此特別要感謝學長家瑋、慶京，在研究過程中提供寶貴的經驗；同學韋辰、宗翰、士勛、和謙、玟玲、古梵、乃宏、冠佑在相處的日子裡相互討論學習、砥礪與關懷，以及學弟國棠、誠愷、元君、永青、華恩、如瑜、家瑞、亦君、重穎的協助與幫忙。感謝實驗室的所有成員在這兩年來的照顧與愛護，讓我在研究的日子裡增加了許多的歡笑。同時，也要感謝師大能提供學生一個良好的學習環境，在此真摯的感謝學校老師與同學們曾經給予的幫助，這段研究的日子將成為我畢生難忘的回憶。

最後要感謝我最愛的家人，父親朱志龍先生、母親林秀梅女士、哥哥朱健豪先生，一路以來對我的支持，我才能專心的完成學業。另外，還有很多關心我、幫助過我的朋友，因為有大家的幫助，我才能有今日的成果，感謝大家，在此致上最誠摯的謝意。在此，願將這份喜悅與榮耀與親愛的家人朋友們共分享，並恭祝他們身體健康、平安快樂。

朱書漢　謹誌

中華民國一百零一年六月二十四日

目錄

表目錄

圖目錄

第1章 緒論

1.1 研究背景與動機

近年來科技不斷地在提升，人們的生活品質也隨著科技更生活化，許多利用機器視覺為基礎的應用也正蓬勃發展中,例如編碼辨識、物體辨識、組織辨識、移動方向辨識...等等的應用。機器視覺系統就是利用機器代替人眼來作各種測量和判斷，它綜合了光學、機械、電子、計算機軟硬件等方面的技術，涉及到圖像處理、模式識別、人工智能、信號處理、光機電一體化等多個領域。 圖像處理和模式識別等技術的快速發展，也大大地推動了機器視覺的發展。

在生產線上，人來做此類測量和判斷會因疲勞、個人之間的差異等產生誤差和錯誤，但是機器卻會不知疲倦地、穩定地進行下去。 一般來說，機器視覺系統包括了照明系統、鏡頭、攝像系統和圖像處理系統。 對於每一個應用，我們都需要考慮系統的運行速度和圖像的處理速度、使用彩色還是黑白攝像機、檢測目標的尺寸還是檢測目標有無缺陷、視場需要多大、分辨率需要多高、對比度需要多大等。 從功能上來看，典型的機器視覺系統可以分為：影像採集部分、影像處理部分和運動控制部分。

近年來，機器視覺的領用正在蓬勃發展中，利用人類眼睛的特性來設計機器視覺的模型[1]，利用機器視覺的自動檢測系統[2], [3]，利用機器視覺的道路交通監控系統[4]，以及利用機器視覺的服務型機器人[5]，為了讓機器視覺有更好的效能，可以利用類神經網路來校正由於鏡頭產生的失真以及圖像處理所造成的錯誤[6]。

使用機器視覺來達成多媒體影音上的互動也是熱門的研究之一[7], [8],

[9], [10]，而本論文是利用機器視覺來評估物體的移動方向，進而達到人可以不用使用感測器來達到控制機器的目的，例如電視機上裝一個機器視覺系統，使用者就可以不用使用遙控器就可以轉台。也可以使用在監視系統中，估測有移動的物體 (例如人)的移動方向，則可以控制監視器跟著物體移動方向移動。

由於使用機器視覺擷取影像來分析物體的移動方向需要很大的運算量，如果使用軟體端來實現則的執行速度會比較慢，沒有辦法達到即時的功能。因此，本論文將採用多主從 SOPC [11], [12], [13], [14], [15]系統架構，以軟硬體協同設計[16], [17]實現單目標移動物體方向之演算法電路，提升系統之執行效能。

模糊理論起源於 1965 年美國加州柏克萊大學的扎德 (L.A. Zadeh) 教授所發表的論文[18]，他在 1968 年提出模糊演算法概念[19]，1970 年提出模糊決策[20]，在 1971 年提出模糊順序[21], [22]，以及在 1973 年提出的複雜系統與決策程序分析的新方法概要[23]，進而其建立模糊控制的基礎。1974 年，英國的 E.H.Mamdani 首次用模糊邏輯和模糊推理實現了世界上第一個實驗性的蒸汽機控制，並取得了比傳統的控制演算法更好的效果。1980 年丹麥的 L.P. Holmblad 和 Ostergard 在水泥窯爐採用模糊控制並取得了成功，這是第一個商業化的有實際意義的模糊控制器。1996 年，結合模糊理論與神經網路來處理從一個機器視覺系統的訓練數據，包括不確定性的情況，具有不確定性的數據轉化成模糊集，此系統可應用於解決排序問題中的大型商用洗碗操作實施的有效性[24]。2006 年，模糊理論應用於提高機器視覺的零件檢測系統，可以減少檢測的時間[25]。近年來將模糊理論應用於 FPGA 晶片[26], [27], [28]上的研究也越來越多，好處是可以降地成本、提高執行效率。本文使用模糊理論來得到目標物體的移動方向，因為模糊理論具有解決不確定性、能夠表達不同評估者不同信心程度及樂觀程度的特性，他與古典數學理論的差異在於模糊理論是以多值函數的觀點

來描述，而不使用傳統的非零即一的方式來區分，允許不確定性的存在，所以能更顯現出優於古典數學之明確的處理方式。

由於硬體電路的影像擷取、顯示速度、及具有平行處理的功能，所以比純用軟體方式設計的系統還要增快許多。本論文提出的方法係以硬體電路實現歷史軌跡，並用 Nios II 計算上、下、左、右方向的計數，再將此計數傳送到模糊邏輯推論的硬體電路，即可得到單目標物體的移動方向。

1.2 論文架構

本論文架構分為七章。第二章將介紹模糊邏輯理論；第三章將介紹本文使用硬體平台以及軟硬體協同設計之方法；第四章將介紹本實驗的系統架構及軟體模擬；第五章將介紹本實驗會用到的硬體電路，包含色彩空間轉換電路、即時影像相減電路、灰階區間二值化電路、歷史軌跡電路以及模糊邏輯電路；第六章將會敘述本文的實驗結果；第七章為結論與未來研究方向。

第2章 模糊邏輯理論

2.1 前言

所謂模糊概念是指這個概念的外延具有不確定性，或者說它的語意是不清晰的，是模糊的。例如速度的快慢、雨勢的大小、光線的強弱、雜訊的大小這些都沒有辦法明確的定義出實際狀況。例如有人說「天氣很熱」、「那個人很年輕」、「那台車的速度很快」，但是溫度多高才算熱？年紀多小才算年輕？速度要多少才叫快？每個人的定義都不相同，也就是說這些描述的事物，會因為每個人的認知而有所不同，但大部分都不會差太多。電腦強於人類的地方在於計算能力，對於無法以有效的計算法則解決的問題，如思考、識別、推理、概念等，表現就不如人類了。因此模糊理論就是針對人腦對於模糊的訊息或不完全的資料，其不需經過精密繁雜的計算過程，仍能做出正確判斷的特色而發展出來。

模糊理論講究的是近似推理，不以精確計算為手段，較符合歷史悠久的國家對事物的看法,如東方哲理等。以往中國人被譏笑為缺乏科學精神，凡是模稜兩可，只要差不多就好，如今差不多精神卻成為了模糊理論解決問題的利器，但這個差不多指的是根據不清晰的資訊，透過推論過程而得到精確的結果。

2.2 模糊理論基本概念

模糊理論的基礎為模糊集合，須瞭解模糊集合的運算、模糊集合的特質以及歸屬函數 (Membership function)，才能對模糊理論有所理解。其基本精神是接受模糊性現象存在的事實，而以處理概念模糊不確定的事物為其研究目標，並積極的將其嚴密的量化成電腦可以處理的訊息，不主張用

繁雜的數學分析即模型來解決模型。

在古典集合中，論域 (universe of discourse)可以被定義為列出他的所有成員 (列舉法)或指定必須滿足性質的成員集合 (規則法)。列舉法只能用於有限的集合，因此有使用限制，規則法比較通用。在規則法中，集合 A 可表示為

$$A=\{x\in U \mid x\text{滿足一些條件}\} \tag{2-1}$$

另外還可以用歸屬法 (membership method)來表示集合 A，對 A 採用 0 或 1 的歸屬函數，以 $\mu_A(x)$ 表示。

$$\mu_A(x)=\begin{cases}1 & \text{若 } x\in A\\ 0 & \text{若 } x\notin A\end{cases} \tag{2-2}$$

2.3 模糊集合基本運算

本節將要介紹模糊集合間的基本運算，如補集、聯集、交集。

1. 補集 (Complement)

我們以符號 $\overline{A}$ 來表示模糊集合 A 的補集，其定義如式 (2-3)所示。另外還有其他的模糊補集，如式 (2-4)的 Yager 模糊補集，其中參數 $w\in(0,\infty)$、式 (2-5)的 Sugeno 模糊補集，其中參數 $\lambda\in(-1,\infty)$

$$c[\mu_A(x)]=\mu_{\overline{A}}(x)=1-\mu_A(x) \tag{2-3}$$

$$c_w(x)=(1-a^w)^{1/w} \tag{2-4}$$

$$c_\lambda(a)=\frac{1-a}{1+\lambda a} \tag{2-5}$$

2. 交集 (Intersection)

模糊交集 (或稱 t-norms) 是一個具有兩個參數的函數，定義如式 2-6：

$$t:[0,1]\times[0,1]\to[0,1] \text{ 使得 } \mu_{A\cap B}=t[\mu_A(x),\mu_B(x)] \tag{2-6}$$

要使得函數t滿足交集定義，必須滿足下列的條件：

A. $t(0,0)=0$以及$t(a,1)=t(1,a)=a$ (邊界條件)

B. $t(a,b)=t(b,a)$ (交換性)

C. 如果$a\le a'$與$b\le b'$，則$t(a,b)\le t(a',b')$ (非遞減)

D. $t(t(a,b),c)=t(a,t(b,c))$ (結合性)

常用的模糊交集有 Dombi，如式 (2-6)、Dubois-Prade，如式 (2-7)、Yager，如式 (2-8)、徹底乘積，如式 (2-9)、愛因斯坦乘積，如式 (2-10)、代數乘積，如式 (2-11)、最小值，如式 (2-12)。

$$t_\lambda(a,b)=\frac{1}{1+[(\frac{1}{a}-1)^\lambda+(\frac{1}{b}-1)^\lambda]^{1/\lambda}},\lambda\in(0,\infty) \tag{2-6}$$

$$t_\lambda(a,b)=\frac{ab}{\max(a,b,\lambda)},\lambda\in[0,1] \tag{2-7}$$

$$t_w(a,b)=1-\min[1,((1-a)^w+(1-b)^w)^{1/w}],w\in(0,\infty) \tag{2-8}$$

$$t(a,b)=\begin{cases}a, \text{若} b=1\\ b, \text{若} a=1\\ 0, \text{其他}\end{cases} \tag{2-9}$$

$$t(a,b)=\frac{ab}{2-(a+b-ab)} \tag{2-10}$$

$$t(a,b)=ab \tag{2-11}$$

$$t(a,b)=\min(a,b) \tag{2-12}$$

3. 聯集 (Union)

模糊交集 (或稱 t-conorms) 是一個具有兩個參數的函數，定義如式 2-13：

$$s:[0,1]\times[0,1]\to[0,1] \text{ 使得 } \mu_{A\cup B}=s[\mu_A(x),\mu_B(x)] \tag{2-13}$$

其中，模糊聯集函數 s (.,.) 必須符合以下四個條件：

A. $s(1,1)=1$以及$s(a,0)=s(0,a)=a$ (邊界條件)

B. $s(a,b)=s(b,a)$ (交換性)

C. 如果$a\le a'$與$b\le b'$，則$s(a,b)\le s(a',b')$ (非遞減)

D. $s(s(a,b),c)=s(a,s(b,c))$ (結合性)

常用的模糊交集有 Dombi，如式 (2-14)、Dubois-Prade，如式 (2-15)、Yager，如式 (2-16)、徹底乘積，如式 (2-17)、愛因斯坦乘積，如式 (2-18)、代數乘積，如式 (2-19)、最小值，如式 (2-20)。

$$s_\lambda(a,b)=\frac{1}{1+[(\frac{1}{a}-1)^\lambda+(\frac{1}{b}-1)^{-\lambda}]^{-1/\lambda}},\lambda\in(0,\infty) \tag{2-14}$$

$$s_\alpha(a,b)=\frac{a+b-ab-\min(a,b,1-\alpha)}{\max(a,b,1-\alpha)},\alpha\in[0,1] \tag{2-15}$$

$$s_w(a,b) = \min[1,(a^w + b^w)^{1/w}], w \in (0,\infty) \tag{2-16}$$

$$s_{ds}(a,b) = \begin{cases} a, 若 b = 1 \\ b, 若 a = 1 \\ 1, 其他 \end{cases} \tag{2-17}$$

$$s_{es}(a,b) = \frac{a+b}{1+ab} \tag{2-18}$$

$$s_{as}(a,b) = a + b - ab \tag{2-19}$$

$$s(a,b) = \max(a,b) \tag{2-20}$$

2.4 模糊控制系統

模糊系統可以被應用在不同的領域中，例如控制、訊號處理、積體電路製造等等，圖 2-1 為模糊系統的主要架構圖，主要可以分成模糊化、模糊推論引擎、模糊規則庫、解模糊化四個步驟。

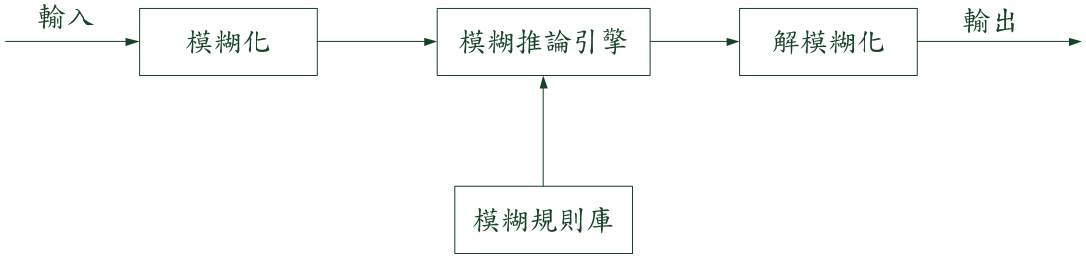


圖 2-1　　模糊系統架構

接下來就將每一個區塊做簡單的說明：

1. 模糊化

模糊化的功能是將明確的外界輸入資料，如輸出變化量、輸出誤差，轉換成適當的語意式模糊資訊；也就是將明確資料模糊化成模糊資訊。以下我們將介紹幾種常見的歸屬函數：

A. 梯形

其數學式如式 2-21，圖 2-2 為函數圖形：

$$\mu_A(x)=\begin{cases}\dfrac{x-a}{m-a}, 若 x\in[a,m]\\ 1, 若 x\in[m,n]\\ \dfrac{b-x}{b-n}, 若 x\in[n,b]\\ 0, 其他\end{cases} \tag{2-21}$$

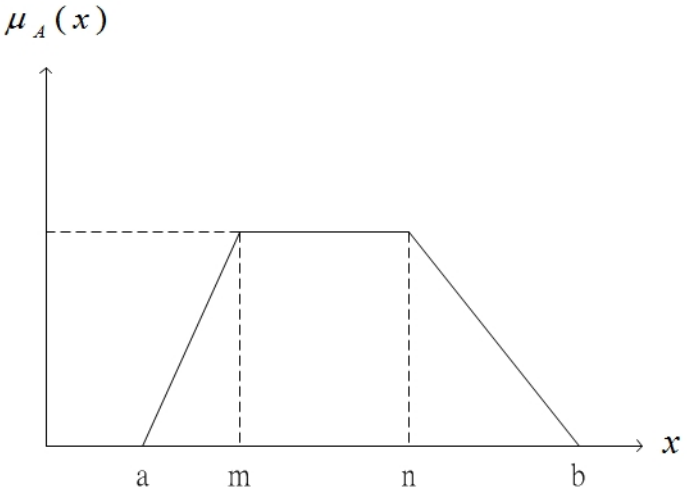


圖 2-2　梯形歸屬函數

其數學式如式 2-22，圖 2-3 為函數圖形：

$$\mu_A(x)=\begin{cases}\dfrac{x-a}{m-a}, 若 x\in[a,m]\\ \dfrac{b-x}{b-m}, 若 x\in[b,m]\\ 0, 其他\end{cases} \tag{2-22}$$

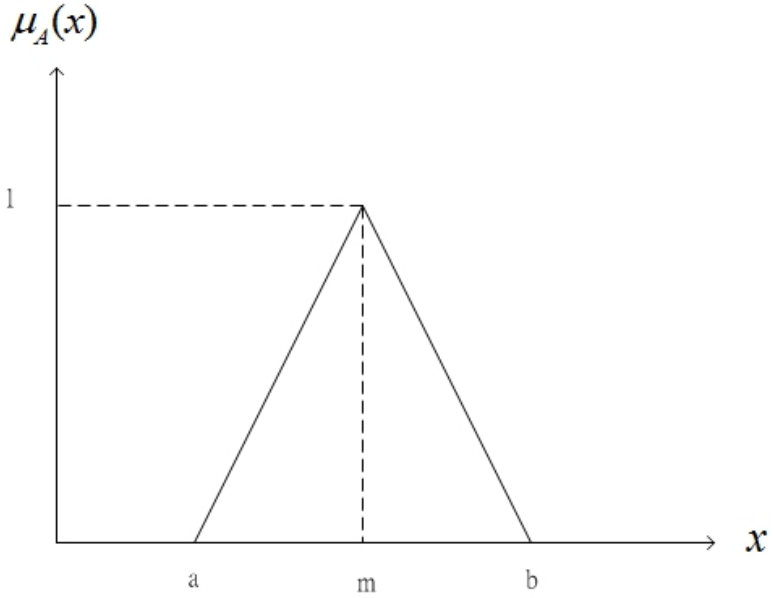


圖 2-3　　三角形歸屬函數

B.　高斯鐘形

其數學式如式 (2-23)，圖 2-4 為函數圖形，當 $x = a$ 時，其歸屬函數值為 1；當 x 越來越遠離 a 時，歸屬函數值則遞減如下式，其中 b 為控制歸屬函數遞減的速率，此種模糊化方式計算方式較複雜，但是假若外界的輸入易被雜訊干擾時，採用高斯模糊化方式則較能有效地消除因為雜訊引起的錯誤。

$$\mu_A(x)=e^{-\frac{(x-a)^2}{b^2}} \tag{2-23}$$

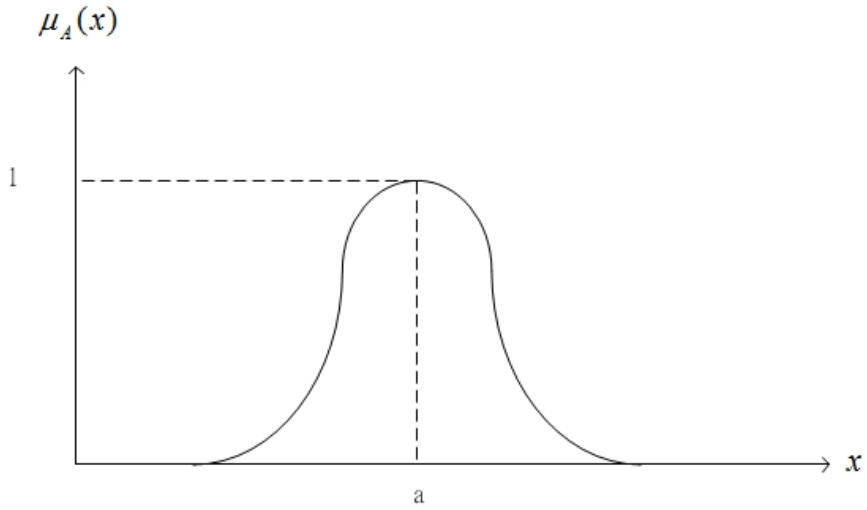


圖 2-4　　高斯歸屬函數

2.　模糊推論引擎

模糊推論引擎是模糊系統的核心，模仿人類的思考決策模式，藉由近似推論或模糊推論，以達到解決問題的目的。推論引擎的運算，可以表示如下：

前提 1: x is A'
前提 2: x is A, then y is B
結論 : y is B'
所以，我們可以得到$B' = A' \circ R = A' \circ (A \to B)$

因為模糊規則庫的構成不只一個規則，所以推論引擎的目的在於如何以規則的集合來推論，以下將介紹兩個以規則的集合來推論[29]：

A.　根據推論的合成 (Composition based inference)

所謂推論的合成就是把模糊規則庫中所有的規則合併成一個單一模糊關係，它可視為一個單一模糊若-則規則。我們首先必須瞭解規則的集合之直覺含意，然後我們再使用合適的邏輯運算子去合併。

對規則的集合意義有以下兩個，第一個是視每一條規則為獨立條件，

則合成規則是使用聯集運算子，如 Mamdani 合成。第二個是視每一條規則為耦合的條件，使得所有的規則條件都能被滿足，則合成規則是使用交集運算子，如 Godel 合成。根據推論的合成之步驟如下：

步驟一：

對於式 (2-24)的形式之 M 條模糊規則來說，根據式 (2-25)來決定歸屬函數 $\mu_{A_1^l\times\cdots\times A_n^l}(x_1,\cdots,x_n), l=1,2,\cdots,M$ 。

$$R^l: \text{若} x_1 \text{是} A_1^l \text{與} \cdots \text{與} x_n \text{是} A_n^l, \text{則} y \text{是} B_l \tag{2-24}$$

$$\mu_{A_1^l\times\cdots\times A_n^l}(x_1,\cdots,x_n)=\mu_{A_1^l}(x_1)\odot\cdots\odot\mu_{A_n^l}(x_n) \tag{2-25}$$

其中⊙代表任意 t-基準運算子。

步驟二：

在蘊含式 (2-26)~式 (2-31)中視 $A_1^l\times\cdots\times A_n^l$ 為 FP_1 ，以及 B^l 為 FP_2 ，並且根據這些蘊含中的任意一個來決定 $\mu_{Ru^{(l)}}(x_1,\cdots,x_n,y)=\mu_{A_1^l\times\cdots\times A_n^l\to B^l}(x_1,\cdots,x_n,y), l=1,2,\cdots,M$ 。其中式 (2-26)為 Dienes-Rescher 蘊含、式 (2-27)為 Lukasiewecz 蘊含、式 (2-28)為 Zadeh 蘊含、式 (2-29)為 Godel 蘊含、式 (2-30)和式 (2-31)為 Mamdani 蘊含。

$$\mu_{Q_D}(x,y)=\max[1-\mu_{FP_1}(x),\mu_{FP_2}(y)] \tag{2-26}$$

$$\mu_{Q_L}(x,y)=\min[1,1-\mu_{FP_1}(x)+\mu_{FP_2}(y)] \tag{2-27}$$

$$\mu_{Q_z}(x,y)=\max[\min(\mu_{FP_1}(x),\mu_{FP_2}(y)),1-\mu_{FP_1}(x)] \tag{2-28}$$

$$\mu_{Q_G}(x,y)=\begin{cases}1, 若\mu_{FP_1}(x)\le\mu_{FP_2}(y)\\ \mu_{FP_2}(y), 其他\end{cases} \tag{2-29}$$

$$\mu_{Q_{MM}}(x,y)=\min[\mu_{FP_1}(x),\mu_{FP_2}(y)] \tag{2-30}$$

$$\mu_{Q_{MP}}(x,y)=\mu_{FP_1}(x)\mu_{FP_2}(y) \tag{2-31}$$

其中 FP_1 與 FP_2 為模糊命題，我們假設 FP_1 是一個定義在 $U=U_1\times\cdots\times U_n$ 的模糊關係，FP_2 是定義在 $V=V_1\times\cdots\times V_m$ 的模糊關係，而且 x 與 y 分別式在 U 與 V 的語意變數。

步驟三：

根據式 (2-32)或式 (2-33)決定 $\mu_{Q_M}(\mathbf{x},y)$ 或 $\mu_{Q_G}(\mathbf{x},y)$。

$$\mu_{Q_M}(\mathbf{x},y)=\mu_{Ru^{(1)}}(\mathbf{x},y)\dot{+}\cdots\dot{+}\mu_{Ru^{(M)}}(\mathbf{x},y) \tag{2-32}$$

$$\mu_{Q_G}(\mathbf{x},y)=\mu_{Ru^{(1)}}(\mathbf{x},y)\odot\cdots\odot\mu_{Ru^{(M)}}(\mathbf{x},y) \tag{2-33}$$

其中 $\dot{+}$ 與 $\odot$ 分別代表 s-norm 與 t-norm 運算子。

步驟四：

已知輸入 A'，模糊推論引擎根據式 (2-34)或式 (2-35)提供輸出。

$$\mu_{B'}(y)=\sup_{\mathbf{x}\in U}t[\mu_{A'}(\mathbf{x}),\mu_{Q_M}(\mathbf{x},y)] \tag{2-34}$$

$$\mu_{B'}(y)=\sup_{\mathbf{x}\in U}t[\mu_{A'}(\mathbf{x}),\mu_{Q_G}(\mathbf{x},y)] \tag{2-35}$$

B. 根據推論的個別規則 (Individual-Rule Based Inference)

在根據推論的個別規則中，模糊規則庫中的每一個規則決定一個模糊輸出，並且整個模糊推論引擎的輸出是 M 個個別模糊輸出的合成，而合成運算為聯集運算或交集運算。根據推論的個別規則之步驟如下：

步驟一：

對於式 (2-24)的形式之 M 條模糊規則，選擇式 (2-25)來決定歸屬函數 $\mu_{A_1^l\times\cdots\times A_n^l}(x_1,\cdots,x_n), l=1,2,\cdots,M$ 。

步驟二：

在蘊含式 (2-26)~式 (2-31)中視 $A_1^l\times\cdots\times A_n^l$ 為 FP_1 ，以及 B^l 為 FP_2 ，並且根據這些蘊含中的任意一個來決定 $\mu_{Ru^{(l)}}(x_1,\cdots,x_n,y)=\mu_{A_1^l\times\cdots\times A_n^l\to B^l}(x_1,\cdots,x_n,y), l=1,2,\cdots,M$ 。

步驟三：

在 U 上給予輸入模糊集合 A' 根據廣義肯定前提式，如式 (2-36)，對每一個個別規則 $Ru^{(l)}$ 計算在 V 上輸出模糊集合 B_l' ，如式 (2-37)。

$$\mu_{B'}(y)=\sup_{x\in U} t[\mu_{A'}(x),\mu_{A\to B}(x,y)] \tag{2-36}$$

$$\mu_{B_l'}(y)=\sup_{\mathbf{x}\in U} t[\mu_{A'}(\mathbf{x}),\mu_{Ru^{(l)}}(\mathbf{x},y)] \tag{2-37}$$

步驟四：

模糊推論引擎的輸出是 M 個模糊集合$\{B_1',\cdots,B_M'\}$的合成，可以使用聯集，如式 (2-38)或交集，如式 (2-39)來合成輸出。

$$\mu_{B'}(y)=\mu_{B_1'}(y)\dot{+}\cdots\dot{+}\mu_{B_M'}(y) \tag{2-38}$$

$$\mu_{B'}(y)=\mu_{B_1'}(y)\odot\cdots\odot\mu_{B_M'}(y) \tag{2-39}$$

其中$\dot{+}$與$\odot$分別代表 s-norm 與 t-norm 運算子。

3. 模糊規則庫

模糊規則庫是由 If-Then 的模糊規則所組成，用以描述系統的輸入輸出關係。模糊規則庫代表此控制系統的思考法則，其中的規則是需要專家的經驗和知識來訂定，所以才說模糊控制其實是一套專家系統。而模糊規則庫的設計會影響系統的好壞，為了能獲得系統的最佳效果，則需要多次使用試誤法來得到好的結果。

以下為三種常見的模糊規則形式：

A. 語意式模糊規則

語意式模糊規則又可稱為 Mamdani 模糊規則，語意式模糊規則表示如式 (2-40)所示。

$$R^{(l)}:\text{If } x_1 \text{ is } A_1^l \text{ and}\cdots\text{and } x_n \text{ is } A_n^l \text{ , Then } y \text{ is } B^l \tag{2-40}$$

其中， $x_1\cdots x_n$ 與 y 為語意變數，代表模糊系統的狀態變數與控制變數，$A_1^l\cdots A_n^l$ 與 B^l 為語意值，代表模糊集合。

B. 函數式模糊規則

函數式模糊規則為語意式模糊規則的另外一種表示法，是將後件部改成系統狀態變數的函數，如式 (2-41)所示。

$$R^{(l)}: \text{If } x_1 \text{ is } A_1^l \text{ and} \cdots \text{and } x_n \text{ is } A_n^l \quad , \quad \text{Then } y = f(x_1, x_2, \cdots, x_n) \tag{2-41}$$

C. Tsukamoto 式模糊規則

此模糊規則的後件部B^l採用的是具有單歸屬函數的模糊集合，因此，每一個模糊規則透過推論後，得到的是一個明確值。

4. 解模糊化

所謂的解模糊化就是經過模糊推論所得到的結果，再經由合理的適當計算，將模糊集合轉換至明確的輸出。目前常見之解模糊化方式有：

A. 重心法 (Center of gravity defuzzifier)：

數學式定義如式 (2-42)，其中$\int_V$是古典積分。

$$y^* = \frac{\int_V y\mu_{B'}(y)dy}{\int_V \mu_{B'}(y)dy} \tag{2-42}$$

B. 最大值平均法 (Mean of maxima defuzzifier)：

數學式定義如式 (2-43)，其中B'是推論引擎輸出的模糊集合，$\int_{hgt(B')}$是對$hgt(B')$的連續部分用積分，對不連續部分使用合計的方法。

$$y^* = \frac{\int_{hgt(B')} ydy}{\int_{hgt(B')} dy} \tag{2-43}$$

C. 權重平均法 (Weight average defuzzifier)：

其數學式定義如式 (2-44)，其中σ_i為權重值，$f_i(x_i, y_i)$為第i個模糊規則經過推論後所得到的結果。

$$y^* = \frac{\sigma_i f_i(x_i, y_i)}{\sum \sigma_i} \tag{2-44}$$

D. 中心平均法 (Center average defuzzifier)：

其數學式定義如式 (2-45)，其中$\bar{y}^l$為第l個模糊集合的中心且w_l為此模糊集合的高度。

$$y^* = \frac{\sum_{l=1}^{M} \bar{y}^l w_l}{\sum_{l=1}^{M} w_l} \tag{2-45}$$

第3章 軟硬體協同設計平台

本章將介紹軟體套件與硬體開發平台，軟體套件主要是 Quartus II 的相關介紹，硬體方面主要是由 altera 公司開發的 FPGA 實體介紹。在軟體套件上可分成 Quaturs II、SOPC Builder、Avalon bus 來探討。硬體方面可分成 DE2-70 開發版、D5M 影像擷取模組、以及 LTM 顯示模組來做說明。

3.1 硬體平台探討

3.1.1 DE2-70 多媒體開發版

DE2-70 是友晶科技利用 altera 的 FPGA 晶片[30]所研發的多媒體開發版，其核心元件為 Cyclone II 2C70 晶片。Altera DE2-70 最新多媒體開發平台配備了數量高達 70,000 個邏輯單元的 Altera Cyclone® II 2C70 和更大容量的記憶體元件，並完全承襲了 Altera DE2 多媒體平台豐富的多媒體、儲存及網路等應用介面的優點。此開發平台提供多樣的產品功能，不僅適用於大專院校實驗室的教學研究與專題製作，也適合工業界做為開發複雜數位系統的工具。

DE2-70 開發版的元件如表 3-1 所示：

表 3-1　　DE2-70 開發板周邊相關元件

FPGA 晶片	➢ Cyclone II 2C70 FPGA
記憶體	➢ 2 個 32-Mbyte SDRAM ➢ 1 個 2-Mbyte SRAM ➢ 8-Mbyte Flash 記憶體 ➢ 支援 SPI 以及 SD 1-bit 兩種 SD Card 讀取模式
介面	➢ USB Blaster 電路 ➢ Altera 序列配置器 ➢ 4 個按鈕 ➢ 18 個滑動開關 ➢ 9 個綠色 LEDs ➢ 18 個紅色 LEDs ➢ 8 個七段顯示器 ➢ 16 × 2 LCD 模組 ➢ 10/100MHz 乙太網路控制器和接頭 ➢ IrDA 收發模組 ➢ 1 個 SMA 接頭 ➢ USB 主/從控制器 ➢ Type A 和 Type B 的接頭 ➢ 50MHz 振盪器 ➢ 28.63MHz 振盪器 ➢ 2 個 TV 訊號解碼器 ➢ 2 個 TV 輸入訊號連接頭 ➢ 一組 RS-232 訊號接頭 ➢ 一組 PS/2 訊號接頭

3.1.2 D5M 擷取模組介紹

DE2-70 開發版上之攝影模組為 D5M 擷取模組，此模組為一個具有 500 萬畫素的 CMOS 擷取模組。其原理是利用感光二極體 (Photo Diode)進行光電轉換，再將圖像的類比訊號轉成數位訊號，其中每一個像素都會有一個放大器及一個數位/類比轉換電路。

CMOS Camera THDB-D5M 可支援一般 VGA 格式，也可以支援 SXGA、XGA、UXGA、QXGA 等，其最高解析度可達 2592 × 1944，使用 Bayer Pattern 色彩濾波陣列，內建 12 bits 之 DAC。詳細規格請參考 THDB-D5M 使用手冊[31][32]。 接下來的小節，將會分別介紹 THDB-D5M 模組之特性及內部腳位、擷取影像格式及影像擷取模組之訊號時序。

3.1.2.1 D5M 模組特性

D5M 擷取模組有以下的特性：

1. 高幀頻以及卓越的低光性能。
2. 卓越的低光性能。
3. 利用列和行跳躍模式，以減少圖像的大小。
4. 支援動態捕抓模式。
5. 可依據環境光源的強烈，來調整 D5M 影像擷取模組的曝光時間，也可以讓使用者作調整。

D5M 擷取模組之規格參數整理如

表 3-2 所示。

表 3-2　　D5M 擷取模組之規格參數

參數	數值
有效像素	2,592H × 1,944V
像素大小	2.2μm x 2.2μm

彩色濾波器陣列		RGB Bayer pattern
最大傳輸速率/主時脈		96 Mp/s at 96 MHz
ADC 分辨率		12-bit
響應率		1.4 V/lux-sec (550nm)
像素動態範圍		70.1dB
最大信噪比		38.1dB
顯示速度	全螢幕	最多 70 fps (frame per second)
	VGA (640×480)	最多 15 fps (frame per second)
提供電壓	power	3.3V
	I/O	1.7V ~ 3.1V

3.1.2.2 D5M 擷取模組腳位說明

本節將介紹D5M的腳位連接,如圖 3-1所示,其中D0~D11為訊號線，LVAL和FVAL分別為D5M擷取模組的Line valid和Frame valid訊號線，PIXCLK 為 D5M 擷取模組提供給外部電路的同步時脈訊號，SDATA 與 SCLK 接腳為透過雙線式串列傳輸介面 (Inter-IC, I^2C)的傳輸方式來設定D5M擷取模組裡面的暫存器數值。

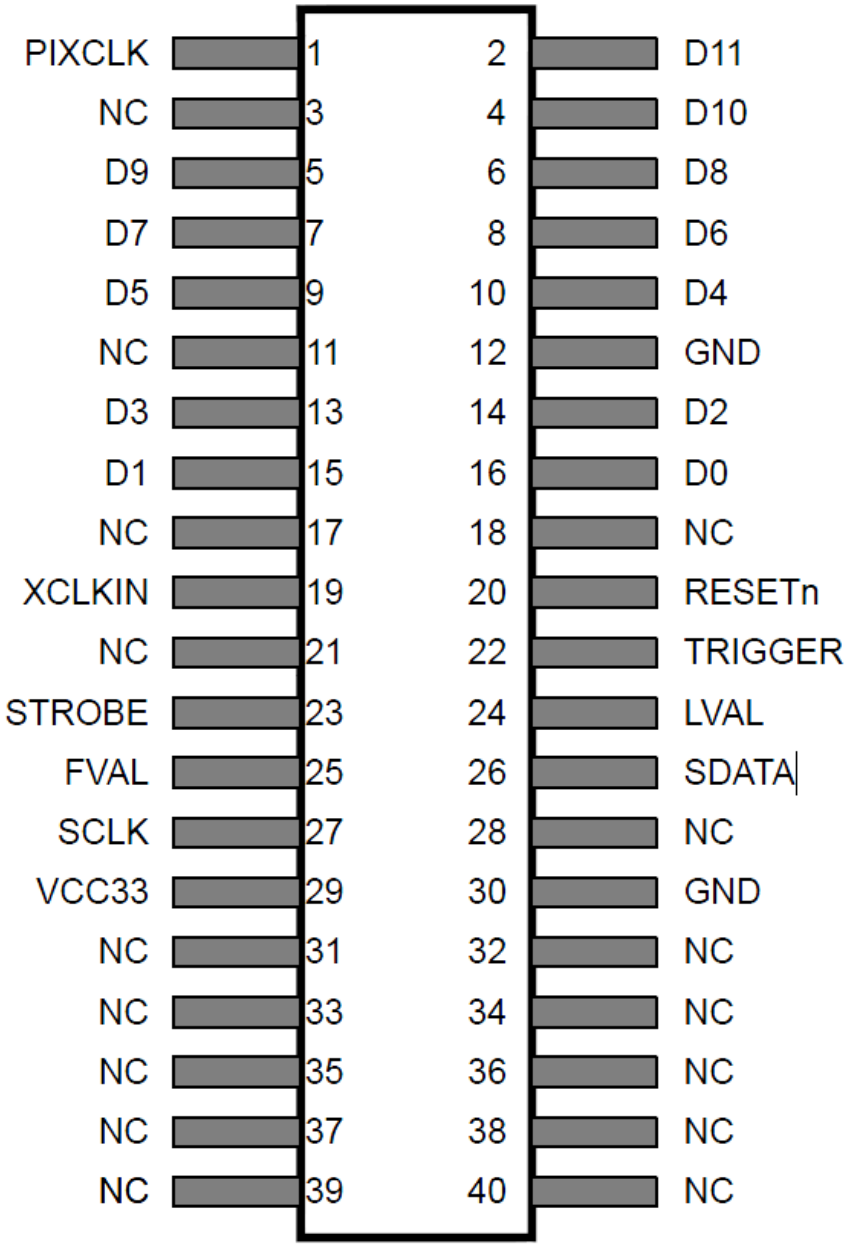


圖 3-1　　D5M 擷取模組之腳位連接圖[31]

3.1.2.3 D5M 影像擷取格式

此節將介紹 D5M 之影像格式，其像素陣列描述如圖 3-2 所示，一張圖片共有 2752 × 2004 個像素點，其中最外圍為黑色邊界，最裡面為有效區域，介於這兩者中間為有效邊界。在有效影像中有 2592 × 1944 個像素點，在有效邊界中，上下各為 4 列，而左 10 行及右 6 行的邊界像素此邊界為色彩修正邊界，為了防止圖像與像素邊界產生模糊效應，使得像素邊

界的值影響到原本圖像的色彩所加上的區隔。

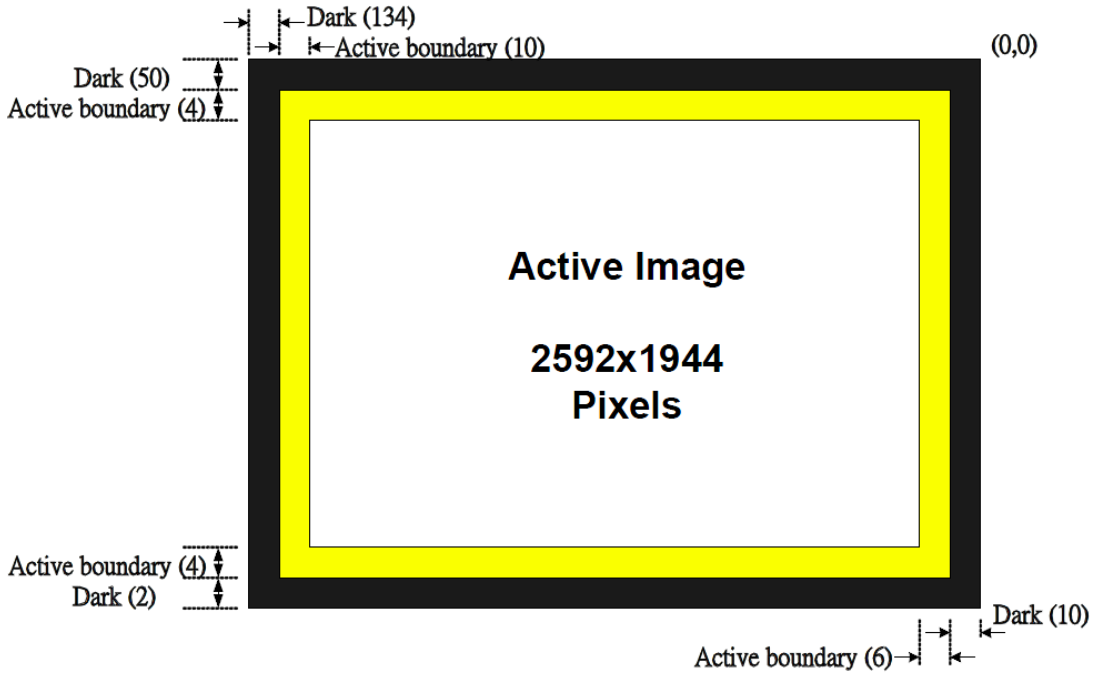


圖 3-2　　D5M 之像素陣列描述圖[32]

接下來要介紹的是 D5M 影像擷取模組之訊號格式，其訊號格式如圖 3-3 之 Bayer Pattern 所示，讀取方向是從右到左，上到下的方式讀取，其奇數列是以 BGBG…之藍綠方式排列，而偶數列是以 GRGR…之綠紅的方式排列。若要將 Bayer Pattern 轉成 RGB 的模式，可以參照下列的步驟即可得到每一個像素點的 RGB 值。

首先將所有的顏色區塊都做編號，數位訊號在一般的情況下是每一格資料位址都有三種顏色的資料，所以類比訊號轉數位訊號就是將類比訊號所對應的資料位址去取周圍相同顏色的顏色平均值，此平均值就是數位訊號所對應位址的三種顏色平均值。

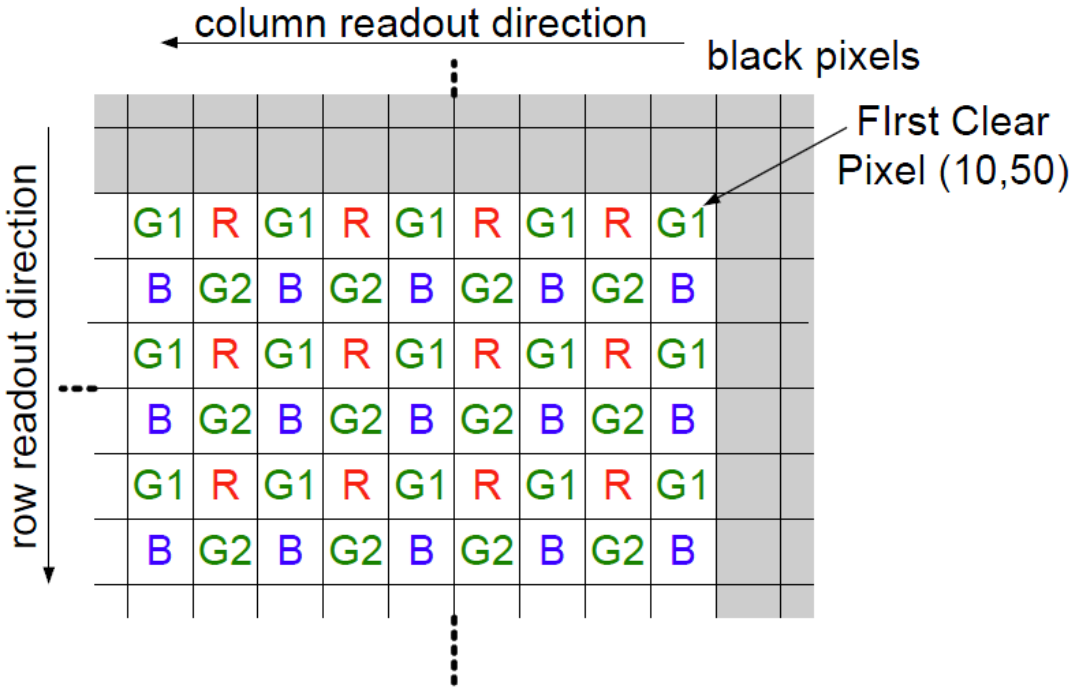


圖 3-3　　D5M 擷取模組之影像訊號[32]

Bayer Pattern 像素圖如圖 3-4 所示，如要求得 G10 位址中的 RGB 值時，則將鄰近 G10 位址的 RGB 值個別取平均值，如下列公式所示，即可獲得 G10 位置的 RGB 顏色值，其分別表示為紅色 R10、綠色 G10、藍色 B10，如式 (3-1)到式 (3-3)所示。

$$R10 = \frac{R9 + R11}{2} \tag{3-1}$$

$$G10 = \frac{G5 + G7 + G10 + G13 + G15}{5} \tag{3-2}$$

$$B10 = \frac{B6 + B14}{2} \tag{3-3}$$

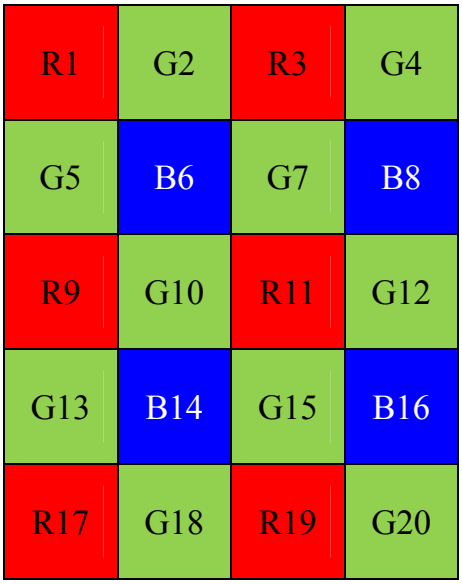


圖 3-4　　Bayer Pattern 像素圖

3.1.2.4 D5M 擷取模組輸出數據之格式

D5M 的影像資料是用逐行掃描的方式讀取資料的，在圖 3-5 中，有效影像的資料是被包圍在水平空白區域與垂直空白區域中。

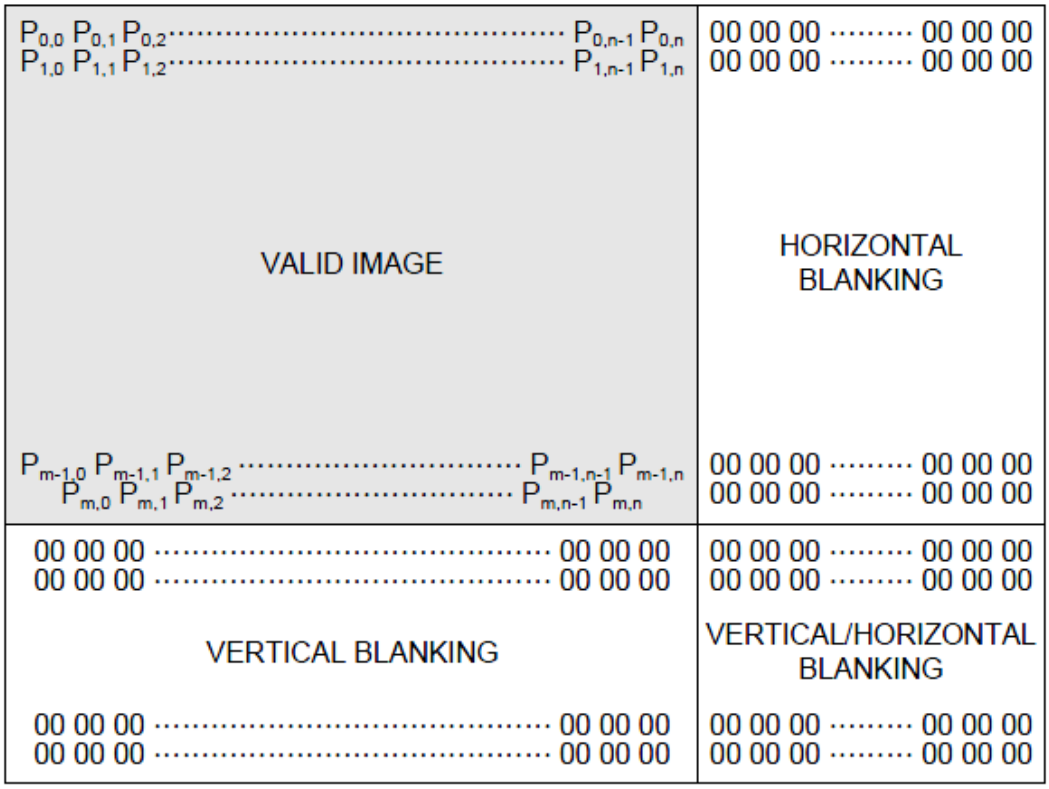


圖 3-5　　D5M 之影像讀取之空間示意圖[32]

D5M 擷取模組在一般預設值下傳送的圖片大小約 1600 × 960，而這 1600 × 960 圖片訊號是依據 D5M 擷取模組所給的 Pixel 時脈訊號而產生對應的掃描線 LINE VALID 資料，如圖 3-6 所示。LINE VALID 表示螢幕的水平掃描線是有效的，舉例來說，一張 1600 × 960 的畫面共有 960 條水平掃描線，代表一個 FRAME VALID 訊號裡會有 960 條掃描線，而每條水平掃描線有 1600 個像素點，表示共有 960 個 LINE VALID 為高準位的訊號。LINE VALID 必須在 FRAME VALID 為高準位的時候，所傳輸的資料才是正確的影像資料訊號，因此，當 LINE VALID 為高準位時，表示螢幕的水平掃描線資料是有效的，而之後的 Pixel 時脈訊號的正緣觸發所讀到的 D[11:0]匯流排資料則為正確的影像像素資料，所以一條 LINE VALID 會有 1600 個 Pixel 時脈訊號。

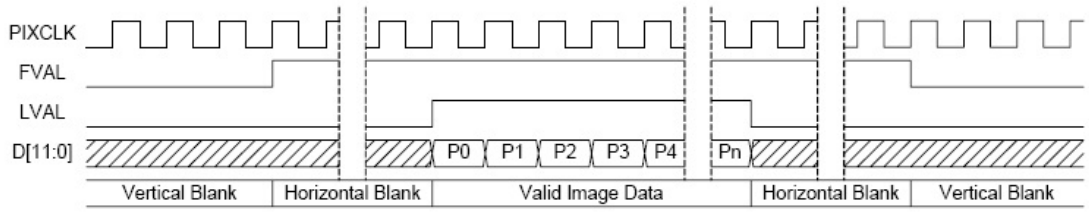


圖 3-6　　D5M 擷取模組時序圖

3.1.3 LTM 顯示模組介紹

4.3" LCD 觸控面板套件又稱為 LTM 顯示模組，具有 800 × 480 高解析度液晶面板，支援 24 位元的平行輸入 RGB 色彩數位訊號，並可透過串列式週邊介面 (Serial Parallel Interface, SPI)的傳輸方式來設定功能與內部暫存器數值，具有內建色彩調整電路、內部對比、亮度及 γ 值調整，提供絕佳畫質的全彩觸控體驗[33]。

LTM 顯示模組如圖 3-7 所示，此模組規格如表 3-3 所示：

表 3-3　　LTM 之規格參數表

規格參數	數值
螢幕尺寸	4.3 英吋
尺寸比例	15 : 9
有效面積 (H × V)	93.6 × 56.16 (mm)
資料點數量 (H × V)	800 × RGB × 480
每點間隔 (H × V)	0.039 × 0.117 (mm)

圖 3-7　　LTM 顯示模組實體圖

LTM 顯示模組腳位如圖 3-8 所示，訊號腳位說明如表 3-4 所示。其中 LCD 綠色資料匯流排 (LCD green data bus)為 G0 ~ G7 腳位，LCD 紅色資料匯流排 (LCD red data bus)為 R0 ~ R7 腳位，LCD 藍色資料匯流排 (LCD blue data bus)為 B0 ~ B7 腳位，而一個資料匯流排一次傳送 8 位元的資料。NCLK 接腳為 LCD 時脈訊號線，GREST 為重置訊號線，HD 與 VD 則為 LTM 顯示模組送給外部控制裝置的同步訊號線，以確保每一筆 RGB 像素資料訊號都是完整的，而 SDA 和 SCEN 接腳則是 SPI 串列傳輸協定的傳輸腳位，主要的功用是設定 LTM 內的暫存器數值。

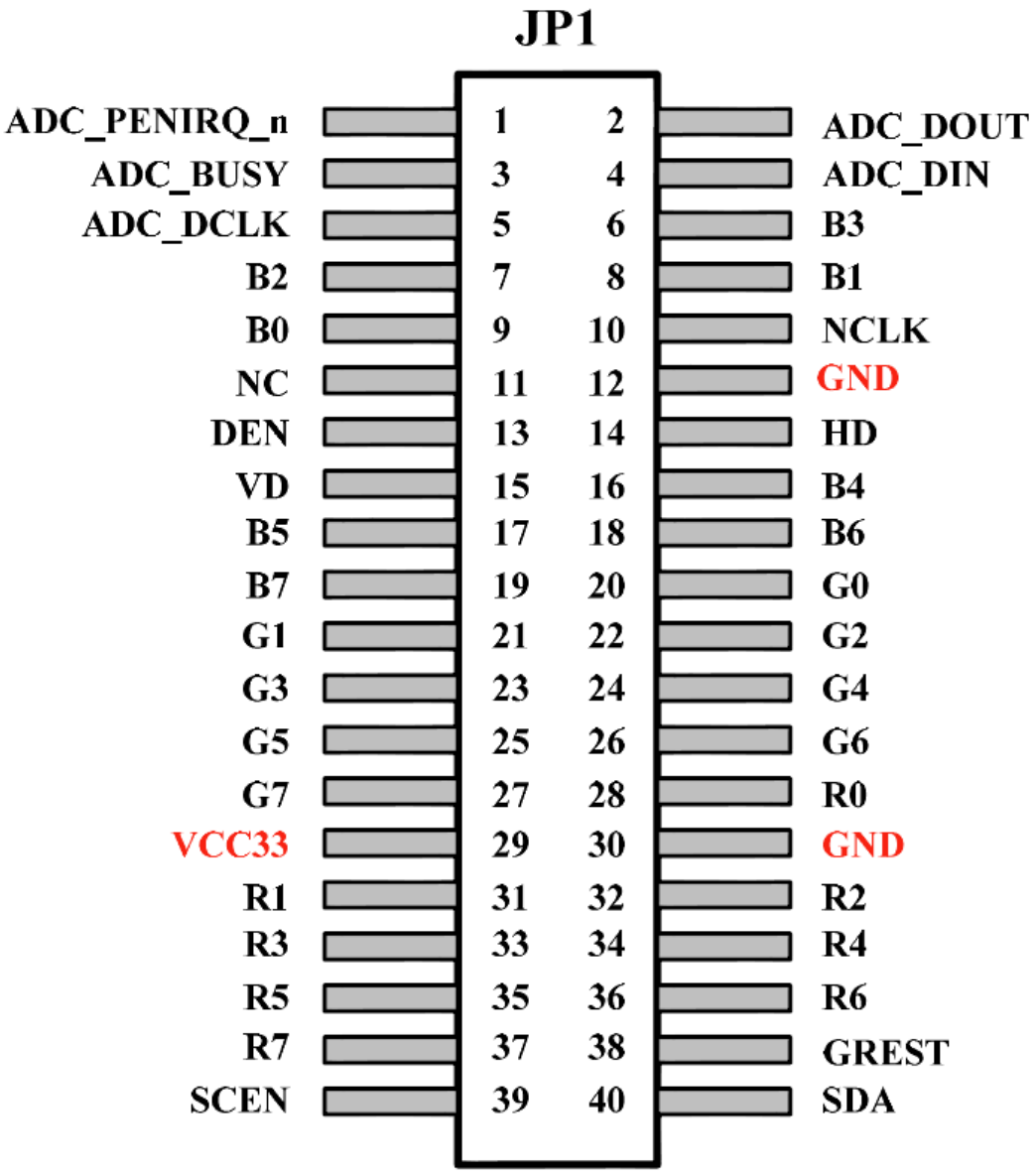


圖 3-8　　TRDB_LTM 顯示模組之腳位圖[34]

表 3-4　　LTM 顯示器模組訊號說明

Pin Number	Name	Direction	Description
1	ADC_PENIRQ_n	Output	ADC pen interrupt
2	ADC_DOUT	Output	ADC Serial interface data out
3	ADC_BUSY	Output	ADC serial interface busy
4	ADC_DIN	Input	ADC Serial interface data in
5	ADC_DCLK	Input	ADC/LCD serial interface clock
6~9	B3~B0	Input	LCD blue data bus bit 3~0
10	NCLK	Input	LCD clock signal
11	NC	N/A	N/A
12	GND	N/A	Ground
13	DEN	Input	LCD RGB data enable
14	HD	Input	LCD Horizontal sync input
15	VD	Input	LCD Vertical sync input
16~19	B4~B7	Input	LCD blue data bit 4~7
20~27	G4~G7	Input	LCD green data bus bit 0
28	R0	Input	LCD red data bus bit 0
29	VCC33	N/A	Power 3.3V
30	GND	N/A	Ground
31~37	R1~R7	Input	LCD red data bus bit 1~7
38	GREST	Input	Global reset，low active
39	SCEN	Input	3-wire serial interface enable
40	SDA	In/Out	3-wire serial interface data

3.2 軟硬體協同設計方法

介紹完硬體開發平台後，對於硬體方面的元件有了大致上的了解，本章節要介紹的是 altera 公司所開發的軟體套件 Quartus II，此設計軟體提供了一個完整的、多平台的設計環境，可以簡單地適應使用者所設計的系統需求。此系統還包含了一個可編程晶片系統 (SOPC)。所以接下來的章節會介紹多主從系統架構、Slave 以及 Master。

本文將參考[35][36]所提出的多主從系統架構的設計方式，實現以軟硬體協同設計之模糊目標物移動方向辨識系統。

3.2.1 多主從系統架構

傳統匯流排的傳輸方式如圖 3-9 所示，當一個 Master 端，如 8051、Nios II 等…使用系統匯流排 (System Bus)存取 PIO 或 Memory 等周邊裝置時，其它的 Master 端便無法使用此系統匯流排去存取周邊裝置，必須等使用中的 Master 端釋放系統匯流排之控制權限後才能去做讀取的動作。

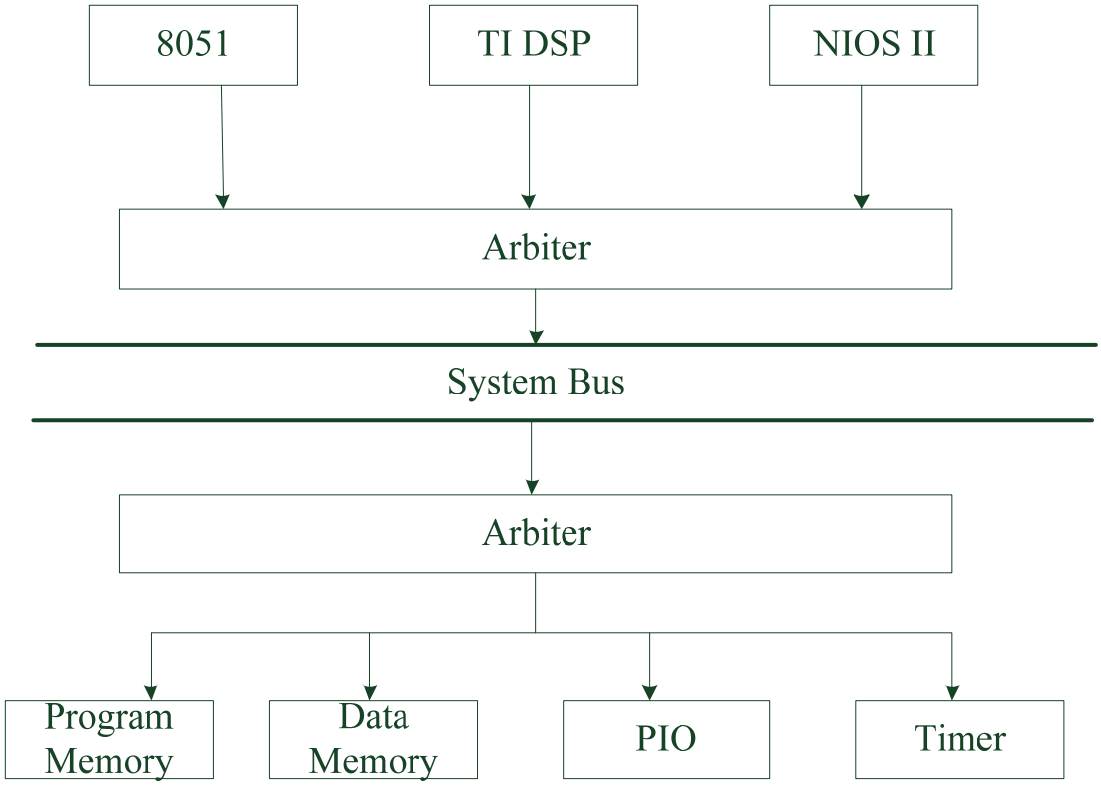


圖 3-9　傳統匯流排傳輸方式

而使用者可利用平行處理架構改善傳統匯流排傳輸方式的處理效能，如記憶體直接存取 (Direct Memory Access, DMA)技術，使用 DMA 控制器取代 CPU 執行記憶體資料搬移的動作，以便讓 Nios II 處理器有時間能同時處理其它的工作。此系統因有多個主處理器在工作，所以需加入多組獨立的匯流排模組來完成[37]。

圖 3-10 所示之 Avalon 匯流排 (Avalon Bus)為 Altera 公司用於 Nios II 處理器所出的介面匯流排，優點為多個主 (Master)端可同步且共同分享從(Slave)端裝置，此同時平行處理架構可以讓處理器在同一個時間裡做資料讀取與寫入的動作。

然而多主從系統架構處理器需同時做存取共同資料記憶體的動作，如使用記憶體分享 (Share Memory)技術亦可解決，但為了避免多主從系統架構處理器產生衝突，因此必須利用仲裁器 (Arbiter)來協調多主從系統架構處理器產生衝突的問題，但在設計上也增加了許多難度，而 SOPC Builder

軟體提供了切換仲裁機制功能，此項功能使得此類設計易於成功。

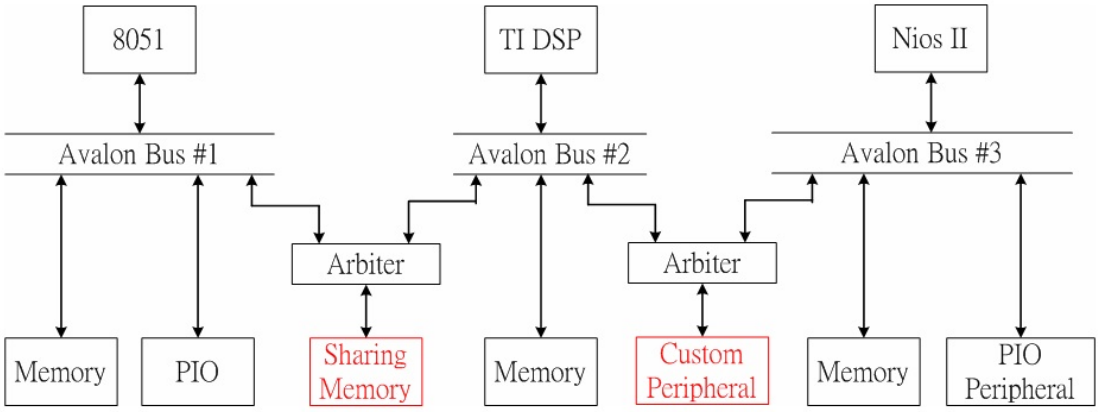


圖 3-10　Avalon Bus 傳輸方式

3.2.2 Master

表 3-5 為 Master 端電路的訊號表，較常用的 Master 端工作模式將在 3.2.2.1 節讀取模式 (Read Mode)、3.2.2.2 節寫入模式 (Write Mode)、3.2.2.3 節多個等待週期的讀取模式 (Wait Request Mode For Write)、3.2.2.4 節多個等待週期的寫入模式 (Wait Request Mode For Read)個別說明。

表 3-5　Master 端訊號表

訊號類型	寬度	方向	敘述
address	1-32	Output	位址線訊號
read、read_n	1	Output	讀取致能訊號
readdata	8, 16, 32, 64, 128, 256, 512, 1024	Input	Master 端從 Avalon Bus 讀取資料的訊號線
write、write_n	1	Output	寫入致能訊號

writedata	8, 16, 32, 64, 128, 256, 512, 1024	Output	Master 端寫入資料到 Avalon Bus 的訊號線
byteenable byteenable_n	1, 2, 4, 8, 16, 32, 64, 128	Output	當所傳輸或寫入的資料大於 8 位元時，必須使用該訊號指定要寫入的資料長度
waitrequest waitrequest_n	1	Input	當 Master 端收到此訊號時，則必須維持輸出訊號不變

3.2.2.1 讀取模式 (Read Mode)

圖 3-11 為 Master 端讀取訊號時序圖，其動作原理敘述如下：

(A) Avalon Bus 在時脈訊號 clk 的第一個上升緣觸發時開始傳輸。

(B) 此時 address、byteenable、read 訊號為有效訊號。

(C) Avalon Bus 已準備好有效的 readdata 資料。

(D) 在下一個 clk 的上升緣觸發之前，Master 端會將 readdata 訊號取走。

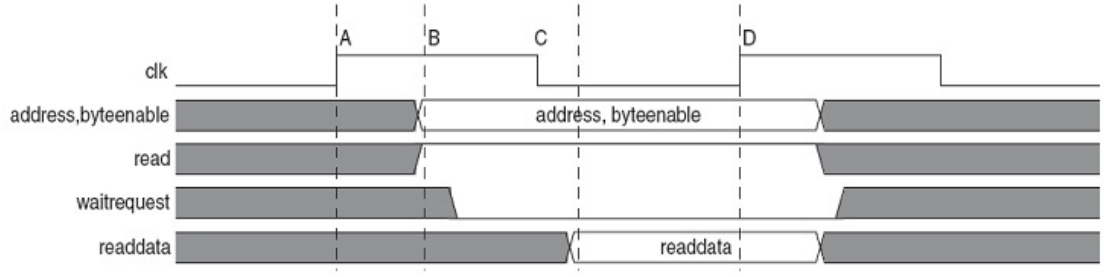


圖 3-11　寫入模式 (Write Mode)

圖 3-12 為 Master 端寫入訊號時序圖，其動作原理敘述如下：

(A) Avalon Bus 在時脈訊號 clk 的第一個上升緣觸發時開始傳輸。

(B) 此時 address、byteenable、write、writedata 訊號為有效訊號。

(C) 在下一個 clk 的上升緣觸發之前，Avalon Bus 會將 writedata 訊號取走。

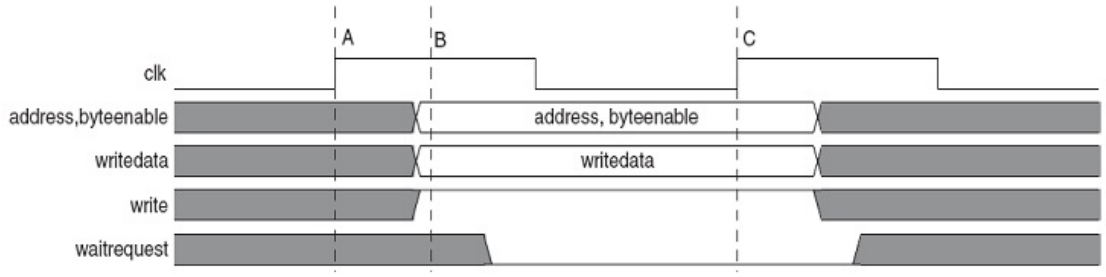


圖 3-12　Master 寫入模式訊號時序圖

3.2.2.2 多個等待週期的讀取模式

圖 3-13 為 Master 端有多個等待週期的讀取模式訊號時序圖，其動作原理敘述如下：

(A) Master 端在時脈訊號 clk 的第一個上升緣觸發時開始傳輸。

(B) 此時 address、byteenable、read 訊號為有效訊號。

(C) Avalon Bus 致能 waitrequest 訊號，要求 Master 端等待。

(D) 在 waitrequest 訊號有效期間內，Avalon Bus 會使所有訊號保持不變。

(E) Avalon Bus 準備好有效的 readdata 資料後，將 waitrequest 訊號設置為無效。

(F) 在下一個 clk 的上升緣觸發之前，Master 端會將 readdata 訊號取走。

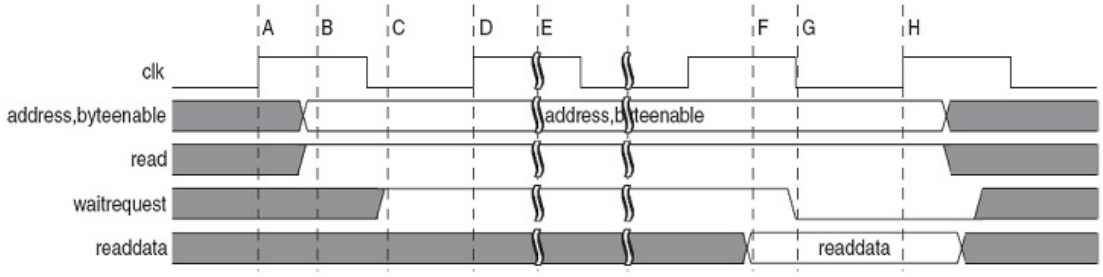


圖 3-13　多個等待週期的 Master 讀取模式訊號時序圖

3.2.2.3 多個等待週期的寫入模式

圖 3-14 為 Master 端有多個等待週期的寫入模式訊號時序圖，其動作原理敘述如下：

(A) Master 端在時脈訊號 clk 的第一個上升緣觸發時開始傳輸。

(B) 此時 address、byteenable、write、writedata 訊號為有效訊號。

(C) Avalon Bus 致能 waitrequest 訊號，要求 Master 端等待。

(D) 在 waitrequest 訊號有效期間內，Avalon Bus 會使所有訊號保持不變。

(E) Avalon Bus 準備好有效的 writedata 資料後，將 waitrequest 訊號設置為無效。

(F) 在下一個 clk 的上升緣觸發之前，Avalon Bus 端會將 writedata 訊號取走。

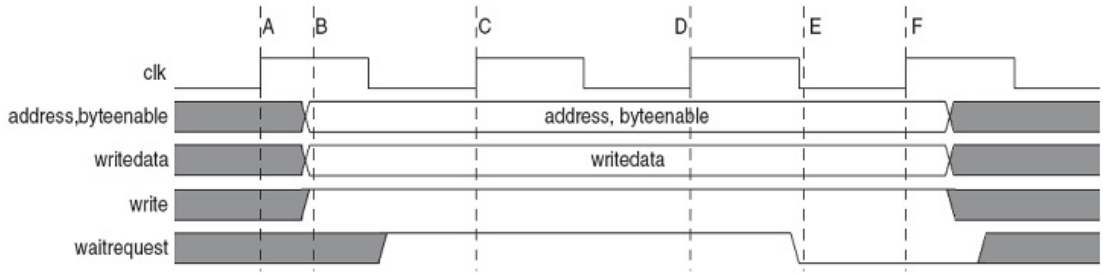


圖 3-14　多個等待週期的 Master 寫入模式訊號時序圖

3.2.3 Slave

實際在設計 Avalon Bus 時，Slave 端電路的設計難度會比 Master 端電路還要易於設計許多，且在設計電路時須按照 Avalon Bus 所規範的訊號時序圖來動作。表 3-6 為 Slave 端電路的訊號表，較常用的 Slave 端工作模式將在 3.2.3.1 節讀取模式 (Read Mode)、3.2.3.2 節寫入模式 (Write Mode)、3.2.3.3 節多個等待週期的讀取模式 (Wait Request Mode For Write)、3.2.3.4 節多個等待週期的寫入模式 (Wait Request Mode For Read)個別說明。

表 3-6　Slave 端訊號表

訊號類型	寬度	方向	敘述
read、read_n	1	Input	讀取致能訊號
write、write_n	1	Input	寫入致能訊號
address	1-32	Input	位址線訊號
readdata	8, 16, 32, 64, 128, 256, 512, 1024	Output	Slave 端從 Avalon Bus 讀取資料的訊號線
writedata	8, 16, 32, 64, 128, 256, 512, 1024	Input	從 Slave 端寫入資料到 Avalon Bus 的訊號線
byteenable byteenable_n	1, 2, 4, 8, 16, 32, 64, 128	Input	當傳輸或寫入的資料大於 8 位元時，須使用此訊號指定要寫入的資料長度。

waitrequest waitrequest_n	1	Output	當 Slave 端無法立即處理 Avalon Bus 的要求時，則須送出此訊號。

3.2.3.1 讀取模式 (Read Mode)

圖 3-15 為 Slave 端讀取訊號時序圖，其動作原理敘述如下：

(A) Avalon Bus 在時脈訊號 clk 的第一個上升緣觸發時開始傳輸。

(B) 此時 address、byteenable、read 訊號為有效訊號。

(C) Avalon Bus 內部會對 Slave 端的位置解碼，並產生 chipselect 訊號。

(D) 當 Slave 端偵測到 chipselect 與 read 訊號時，會致能 Slave 端讀取功能，此時會根據 address 訊號到相對應的暫存器將暫存器值立即送到 readdata。

(E) 在下一個 clk 的上升緣觸發之前，Avalon Bus 會將 readdata 訊號取走。

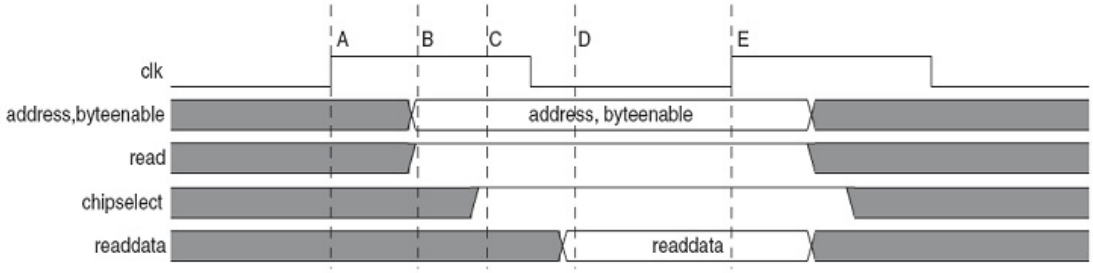


圖 3-15　Slave 讀取模式訊號時序圖

3.2.3.2 寫入模式 (Write Mode)

圖 3-16 為 Slave 端寫入訊號時序圖，其動作原理敘述如下：

(A) Avalon Bus 在時脈訊號 clk 的第一個上升緣觸發時開始傳輸。

(B) 此時 address、byteenable、write 訊號為有效訊號。

(C) Avalon Bus 內部會對 Slave 端的位置解碼，並產生 chipselect 訊號。

(D) 在下一個 clk 的上升緣觸發之前，Slave 端會根據此刻的 address、byteenable、writedata、write、chipselect 訊號，立即將 writedata 資料寫入。

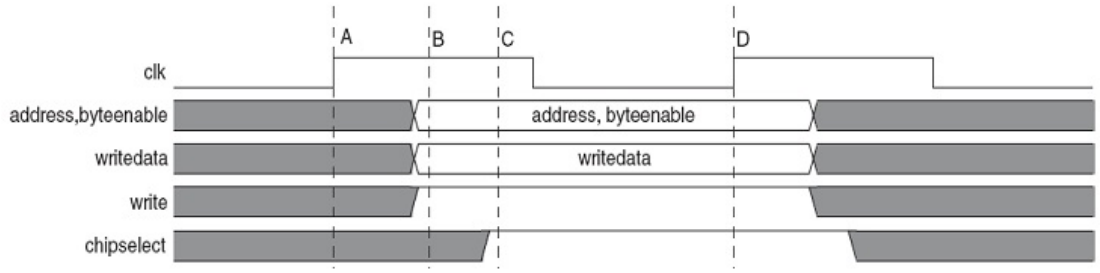


圖 3-16　Slave 寫入模式訊號時序圖

3.2.3.3 多個等待週期的讀取模式

圖 3-17 為 Slave 端有多個等待週期的讀取模式訊號時序圖，其動作原理敘述如下：

(A) Avalon Bus 在時脈訊號 clk 的第一個上升緣觸發時開始傳輸。

(B) 此時 address、byteenable、read、readdata 訊號為有效訊號。

(C) Avalon Bus 內部會對 Slave 端的位置解碼，並產生 chipselect 訊號。

(D) Slave 端會在下一個 clk 的上升緣觸發之前，送出 waitrequest 訊號，要求 Avalon Bus 等待。

(E) 在 waitrequest 訊號有效期間內，Avalon Bus 會使所有訊號保持不變。

(F) 等待 Slave 端準備好有效的 readdata 資料後，將 waitrequest 訊號設置為無效。

(G) 在下一個 clk 的上升緣觸發之前，Avalon Bus 會將 readdata 訊號取走。

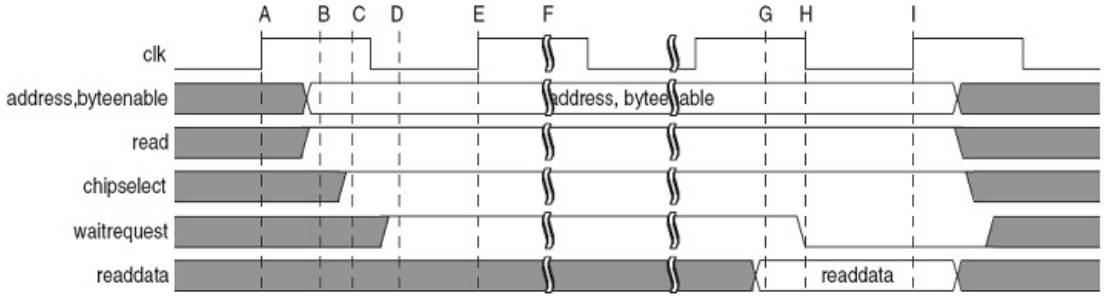


圖 3-17　多個等待週期的 Slave 讀取模式訊號時序圖

3.2.3.4 多個等待週期的寫入模式

圖 3-18 為 Slave 端有多個等待週期的寫入模式訊號時序圖，其動作原理敘述如下：

(A) Avalon Bus 在時脈訊號 clk 的第一個上升緣觸發時開始傳輸。

(B) 此時 address、byteenable、write、writedata 訊號為有效訊號。

(C) Avalon Bus 內部會對 Slave 端的位置解碼，並產生 chipselect 訊號。

(D) Slave 端會在下一個 clk 的上升緣觸發之前，送出 waitrequest 訊號，要求 Avalon Bus 等待。

(E) 在 waitrequest 訊號有效期間內，Avalon Bus 會使所有訊號保持不變。

(F) 等待 Slave 端準備好寫入動作後，將 waitrequest 訊號設置為無效。

(G) 在下一個 clk 的上升緣觸發之前，Slave 端會根據此刻的 address、byteenable、writedata、write、chipselect 訊號，立即將 writedata 資料寫入。

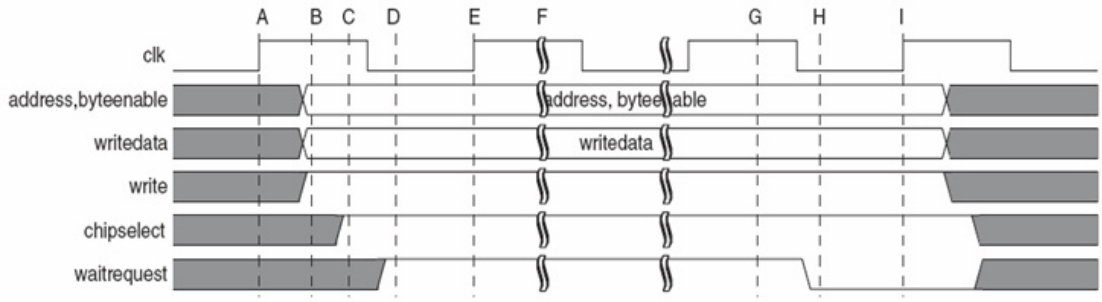


圖 3-18　多個等待週期的 Slave 寫入模式訊號時序圖

第4章 系統架構及軟體模擬

4.1 影像前處理

4.1.1 色彩空間轉換

色彩空間（Color Space）又稱為色域空間，它表示的是一個彩色影像所能夠表現的色彩範圍。YUV 和 RGB 是兩種不同的色彩空間，而計算機彩色顯示器顯示色彩的原理與彩色電視機一樣，都是採用 R（Red）、G（Green）、B（Blue）相加混色的原理，其原理是通過發射出三種不同強度的電子束，使屏幕內側覆蓋的紅、綠、藍磷光材料發光而產生色彩，這種色彩的表示方法稱為 RGB 色彩空間表示。

如果只有 Y 信號分量而沒有 U、V 分量，那麼這樣表示的圖像就是黑白灰度圖像。現在的彩色電視採用 YUV 空間正是為了用亮度信號 Y 解決彩色電視機與黑白電視機的兼容問題，使黑白電視機也能接收彩色電視信號，其 RGB 與 YUV 相互轉換的公式如下：

$$Y = 0.299R + 0.587G + 0.114B \tag{4-1}$$

$$U = -0.147R - 0.289G + 0.436B \tag{4-2}$$

$$V = 0.615R - 0.515G - 0.100B \tag{4-3}$$

$$R = Y + 1.14V \tag{4-4}$$

$$G = Y - 0.39U - 0.58V$$

(4-5)

$$B = Y + 2.03U$$

(4-6)

4.1.2 影像侵蝕

在影像處理中，侵蝕 (Erosion) [38]和膨脹 (Dilation) [38]在形態學中是兩個基本的運算，侵蝕運算的效果主要是去除掉二值化的影像中之前景像素 (即白色的像素)的邊界，此運算可以清除掉一些較少的雜訊，使畫面更清晰。影像侵蝕運算的數學模式如式 (4-7)所示：

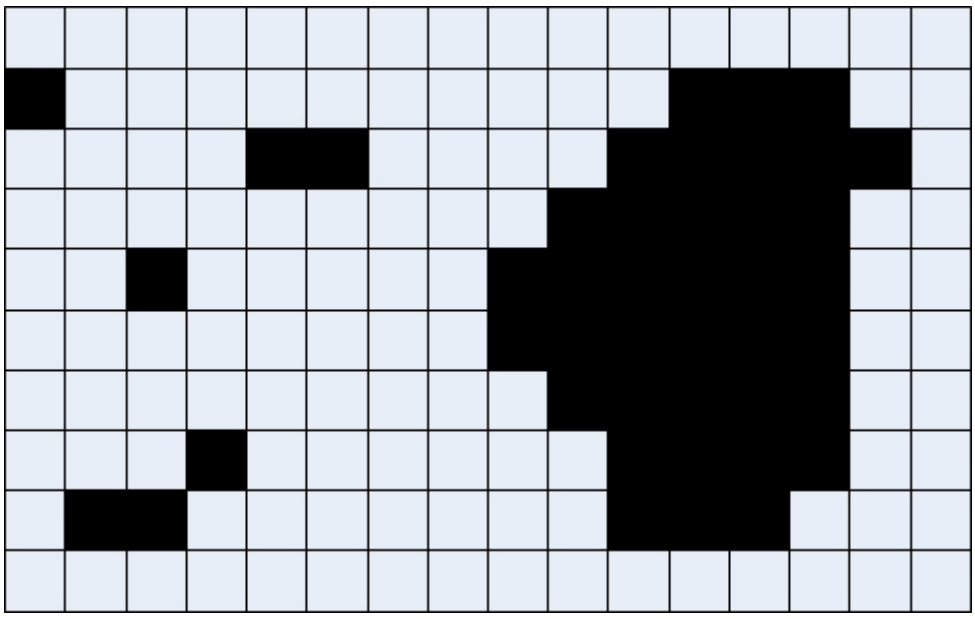
$$eros(x,y) = T(x,y) \ominus u \tag{4-7}$$

其中，$eros(x,y)$ 為經過侵蝕後的影像、$T(x,y)$ 為原始影像、$\ominus$ 為侵蝕運算子、u 為結構元素 (Structure Elements)，本研究所採用的結構元素為 3×3 大小的矩陣。

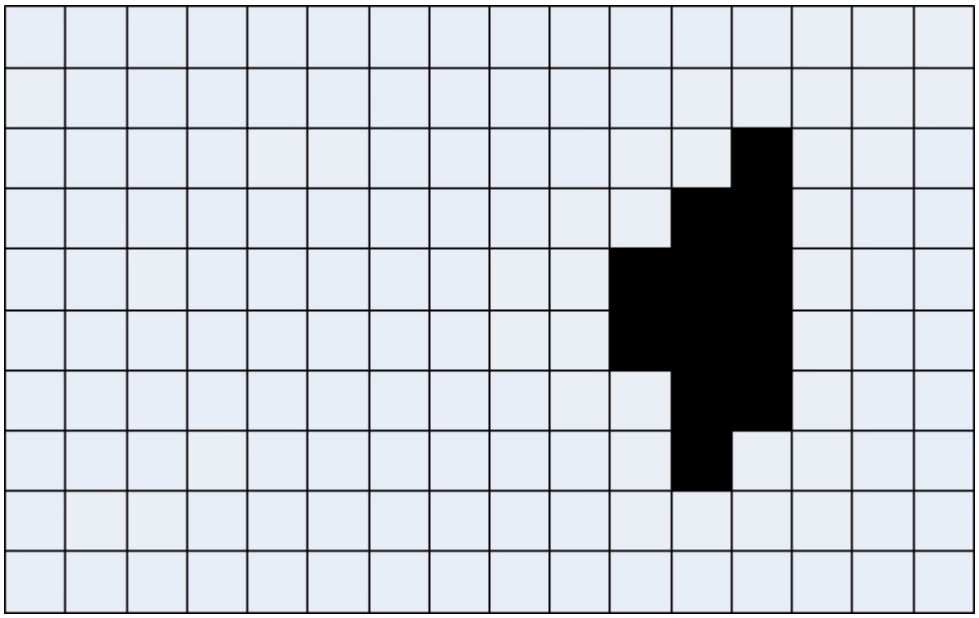
侵蝕運算的的主要概念為對影像遮罩中之像素進行 AND 運算，其演算法流程如下：

1. 擷取影像。
2. 設定結構元素。
3. 設定迴圈控制的邊界，以避免程式執行中發生錯誤。
4. 取出結構內的九個像素值進行 AND 運算。
5. 判斷九個元素內，只要有一個像素值為 0，則將中心點的像素值更改為 0。
6. 遵循迴圈限制數，依序處理影像中每個像素的運算。
7. 輸出膨脹之後的影像。
8. 結束。

如圖 4-1 為形態學的侵蝕運算，其中圖 4-1 (a)為原始影像，圖 4-1 (b)為侵蝕運算後的結果。由侵蝕過後的結果可以得知，侵蝕運算的確能夠將二值化影像中細小的雜訊濾除。



(a)原始影像



(b)原始影像經過侵蝕運算後的結果

圖 4-1　　侵蝕運算示意圖

4.1.3 影像膨脹

介紹完侵蝕運算後，接下來要介紹的是膨脹運算，由於一些環境的因素使得擷取的影像有殘缺不全的狀況，這時候可以使用膨脹運算來填補影像中的破洞或缺角，其公式如式 (4-8)所示。

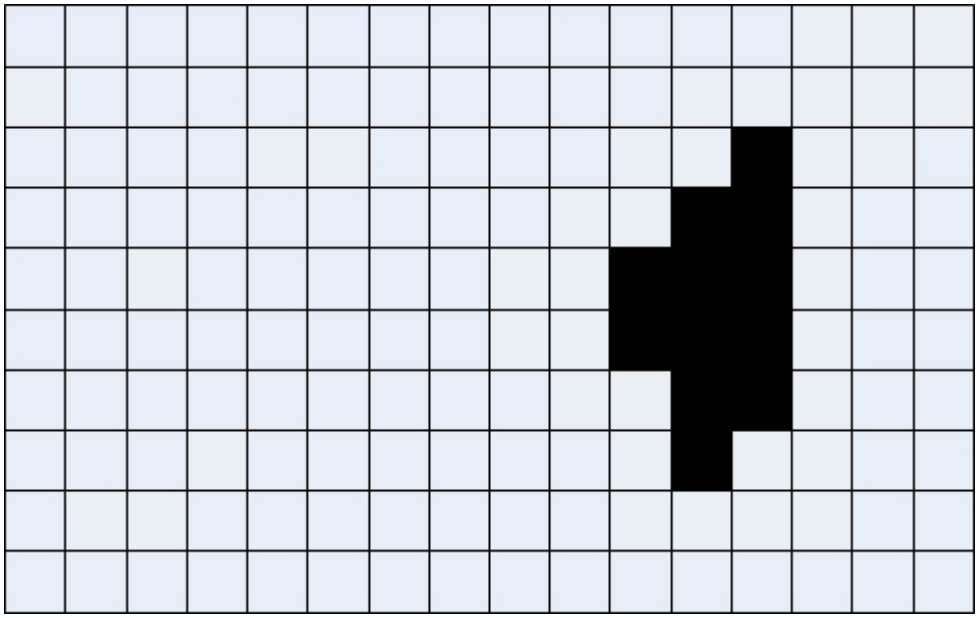
$$eros(x,y) = T(x,y) \oplus u \tag{4-8}$$

其中，$eros(x,y)$ 為經過膨脹後的影像、$T(x,y)$ 為原始影像、$\oplus$ 為膨脹運算子、u 為結構元素 (Structure Elements)，本研究所採用的結構元素為 3×3 大小的矩陣。

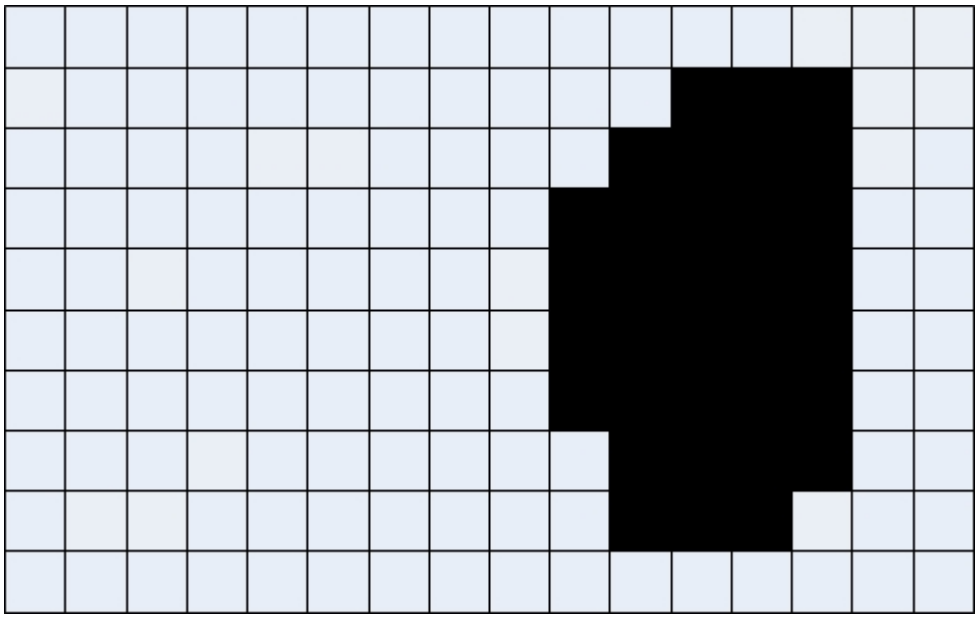
膨脹運算的的主要概念為對影像遮罩中之像素進行 OR 運算，其演算法流程如下：

1. 擷取影像。
2. 設定結構元素。
3. 設定迴圈控制的邊界，以避免程式執行中發生錯誤。
4. 取出結構內的九個像素值進行 OR 運算。
5. 判斷九個元素內，只要有一個像素值為 1，則將中心點的像素值更改為 1。
6. 遵循迴圈限制數，依序處理影像中每個像素的運算。
7. 輸出膨脹之後的影像。
8. 結束。

如圖 4-2 為形態學的膨脹運算，其中圖 4-2 (a)為原始影像，圖 4-2 (b)為膨脹運算後的結果。由膨脹過後的結果可以得知，膨脹運算的確能夠將影像中的缺角或破洞的部分進行填補。



(a)原始影像



(b)原始影像經過膨脹運算後的結果

圖 4-2　　膨脹運算示意圖

4.2 使用模糊理論之歷史軌跡方向辨識系統

本論文提出一種使用模糊理論來辨識目標物移動方向的系統，參考[39]之作法加以改進，此系統是利用連續的 2 維影像，分別經過影像縮小、RGB 轉 Y 空間、影像相減、灰階區間二值化、侵蝕運算、膨脹運算、動態能量遞減、模糊邏輯之方向辨識，可以得到在連續影像中移動物體的方向，可以做為視覺遙控的感測器，其流程圖如圖 4-3 所示。

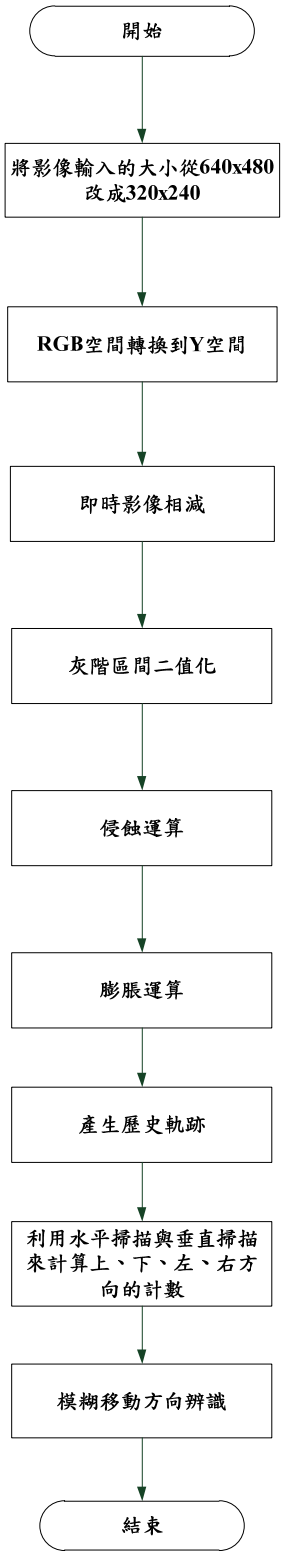
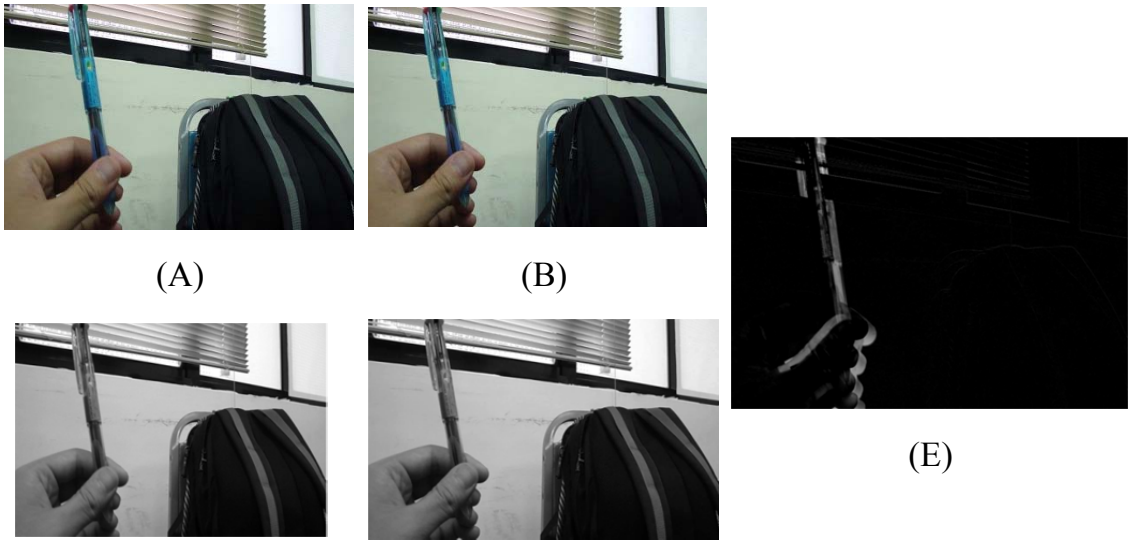


圖 4-3　使用模糊理論之歷史軌跡方向辨識系統架構流程圖

4.2.1 色彩空間轉換與即時影像相減

本論文所使用的色彩空間轉換是 RGB 轉 YUV 的轉換方式，其公式如式 (4-1)到式 (4-3)所示，此步驟可以突顯光流的變化以偵測出物體的移動，為了能更加方便看出物體的移動變化，所以本論文使用 “即時影像相減法” (Temporal Differencing Method)，此方法可藉由連續影像中做及時的相減即可辨別出影像中有差異的部分。色彩空間轉換與即時影像相減的實驗如圖 4-4 所示。



(A) (B)

(C) (D)

(E)

圖 4-4　　連續影像中之色彩空間轉換與即時影像相減

如圖 4-4，為連續影像中之色彩空間轉換與即時影像相減的實驗，其中圖 4-4 (A)、圖 4-4 (B)為兩張連續的 RGB 色彩空間的影像，圖 4-4 (C)、圖 4-4 (D)分別從圖 4-4 (A)、圖 4-4 (B)的 RGB 影像轉成 Y 空間的灰階影像，圖 4-4 (E)為圖 4-4 (C)、圖 4-4 (D)做即時影像相減所得的畫面。

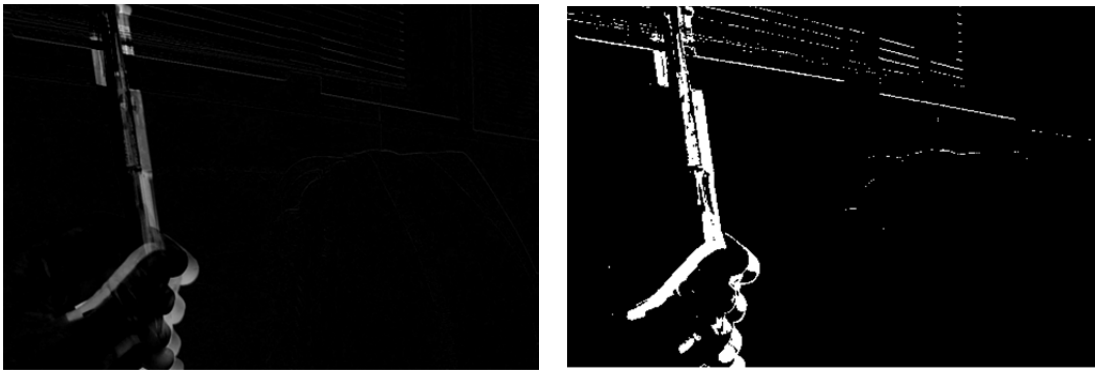
4.2.2 灰階區間二值化

機器視覺的首要工作是擷取一個最佳品質的影像，以避免在後續的分析中發生不必要的錯誤，可以提高機器視覺的成功率。而影像的品質會因

為照明不均、對比、雜訊等影響，造成後續的判斷發生錯誤。在影像處理中，灰階區間二值化是一種常見的濾除雜訊方法。由於圖 4-4 (E)可以看出影像中有許多雜訊，為了後續的影像處理中可以更準確的分析，所以我們使用灰階區間二值化來消除這些雜訊。灰階區間二值化的公式如式 (4-9)所示。

$$Medr_{Frame(x,y)} = \begin{Bmatrix} 255, 120 \le Frame(x,y) \le 200 \\ 0, otherwise \end{Bmatrix} \tag{4-9}$$

其中，$Medr_{Frame(x,y)}$ 為經過灰階區間二值化後的影像、$Frame(x,y)$ 為灰階影像。



(A) (B)

圖 4-5 影像灰階區間二值化之實驗

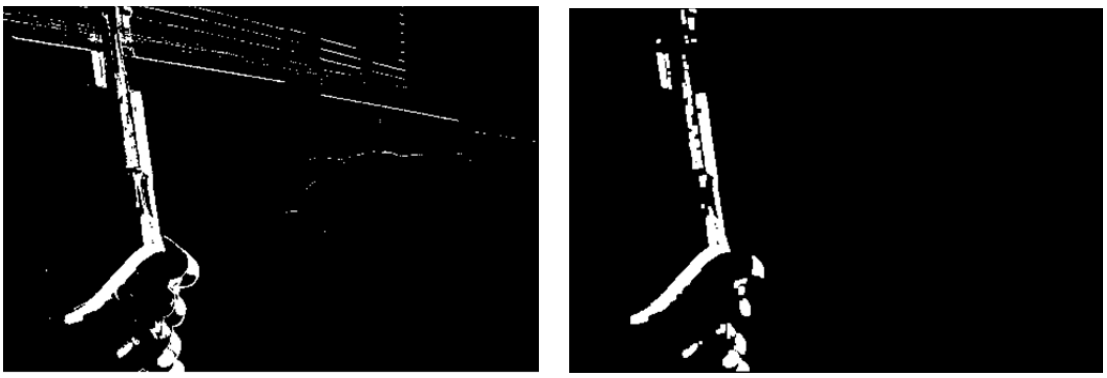
如圖 4-5 (A)為一張灰階的影像，圖 4-5 (B)為經過灰階區間二值化的圖像，我們可以擷取到我們想要的範圍做二值化。

4.2.3 影像開運算 (Opening Operation)

侵蝕是一種消除邊界點，使邊界向內部收縮的過程。可以用來消除小且無意義的物體。膨脹是將與物體接觸的所有背景點合併到該物體中，使邊界向外部擴張的過程。可以用來填補物體中的空洞。先侵蝕再膨脹的過程稱為開運算，用來消除小物體、在纖細點處分離物體、平滑較大物體的邊界的同時並不明顯改變其面積。開運算的公式如式 (4-10)所示。

$$A \circ B = (A \ominus B) \oplus B \tag{4-10}$$

其中，A 為原始的二值化影像，B 為結構元素，$\ominus$為侵蝕運算符號，$\oplus$為膨脹運算符號。



(A) (B)

圖 4-6 影像開運算之實驗

如圖 4-6 為開運算的實驗效果圖，其中圖 4-6 (A)為兩張連續彩色 RGB 影像作 Y 空間轉換，在做影像相減、灰階區間二值化所得的結果，圖 4-6 (B)為圖 4-6 (A)經過開運算所得的解果。比較圖 4-6 (A)與圖 4-6 (B)可得知影像經過開運算後可以消除比較不重要的資訊，並且補齊中間空洞的部分，可以加強影像動量的資訊。

4.2.4 動態能量遞減 (Momentum Decreasing)

當物體在快速運動時，在人眼所看到的影像消失後，人眼仍能繼續保留其影像，約 0.1-0.4 秒左右的圖像，這種現象被稱為視覺暫留現象。人眼觀看物體時，成像於視網膜上，並由視神經輸入人腦，感覺到物體的像，但當物體移去時，視神經對物體的印象不會立即消失，而要延續 0.1-0.4 秒的時間，人眼的這種性質被稱為「眼睛的視覺暫留」[40]。由於人眼有視覺暫留的現象，可以大約判斷出物體移動的方向，所以本節就以人眼的

視覺保留的特性來設計其功能，以便分析物體移動的方向。

為了模仿人眼的視覺暫留現象，我們引入的時間的概念，把不同時段的影像以不同灰階值儲存在同一張影像中，這樣就有如人眼的視覺暫留所看到的影像。其動態能量遞減法的公式如式 (4-11)到式 (4-16)所示：

$hstry^{t}_{(x,y)} = 0$, if $(hstry_{(x,y)}{}^{t-1} \le 50)$

$$hstry^{t}_{(x,y)} = 0 \text{ , if } (hstry^{t-1}_{(x,y)} \le 50) \tag{4-11}$$

$$hstry^{t}_{(x,y)} = hstry^{t-1}_{(x,y)} - 70 \text{ , if } (hstry^{t-1}_{(x,y)} > 50) \text{ and } (hstry^{t-1}_{(x,y)} \le 90) \tag{4-12}$$

$$hstry^{t}_{(x,y)} = hstry^{t-1}_{(x,y)} - 50 \text{ , if } (hstry^{t-1}_{(x,y)} > 90) \text{ and } (hstry^{t-1}_{(x,y)} \le 150) \tag{4-13}$$

$$hstry^{t}_{(x,y)} = hstry^{t-1}_{(x,y)} - 40 \text{ , if } (hstry^{t-1}_{(x,y)} > 150) \text{ and } (hstry^{t-1}_{(x,y)} \le 200) \tag{4-14}$$

$$hstry^{t}_{(x,y)} = hstry^{t-1}_{(x,y)} - 20 \text{ , if } (hstry^{t-1}_{(x,y)} > 200) \text{ and } (hstry^{t-1}_{(x,y)} \le 255) \tag{4-15}$$

$$hstry^{t}_{(x,y)} = hstry^{t-1}_{(x,y)} - 10 \text{ , otherwise} \tag{4-16}$$

其中，其中 $hstry^{t}_{(x,y)}$ 代表當前的歷史軌跡影像像素值，而 $hstry^{t-1}_{(x,y)}$ 則代表上一刻的歷史軌跡像素值。一張彩色RGB影像先後經過RGB轉Y空間、影像相減、灰階區間二值化、開運算、動態能量相減，即可得到歷史軌跡圖，歷史軌跡的實驗如圖 4-7 所示。

(A) (B) (C)

(D) (E) (F)

圖 4-7　動態物體的歷史軌跡

如圖 4-7 為動態物體的歷史軌跡實驗，其中圖 4-7 (A)為在彩色影像中的移動物體、 (B)為 RGB 轉換成 Y 空間灰階影像的移動物體、 (C)為經過即時影像相減後的影像、 (D)為影像經過灰階區間二值化且經過侵蝕運算後的影像、 (E)為再將侵蝕後的影像做膨脹運算即後的影像、 (F)為經過動態能量遞減所產生的歷史軌跡。

4.2.5 基於模糊理論之目標物動態方向辨識

本論文之動態方式的方法是利用先前所提及的歷史軌跡加以分析，可以得到上、下、左、右四個方向的計數，再將這四個計數利用模糊邏輯推論出物體的移動方向。接下來就分兩個部份來詳細說明此演算法，一個部分是探討如何使用歷史軌跡分析上、下、左、右的方向，第二個部分是介紹本論文所使用的模糊邏輯。

1. 使用歷史軌跡判斷權重值

使用歷史軌跡判斷上、下、左、右之方向權重值流程圖如圖 4-8 所示。將產生的歷史軌跡圖使用水平掃描，在水平掃描的過程中，掃描到的像素值不為零時，則此像素值會被記錄為當前的像素值，掃描線繼續掃描，再次掃到不為零的像素值時，此像素值會被紀錄為下一刻的像素值，當當前的像素值小於下一刻的像素值則向右的計數加一，反之，若當前的像素值大於下一刻的像素值，則向左的計數加一。再將歷史軌跡圖使用垂直掃描，

若當前的像素值小於下一刻的像素值，則向下的計數加一，反之，若當前的像素值大於下一刻的像素值，則向上的計數加一。

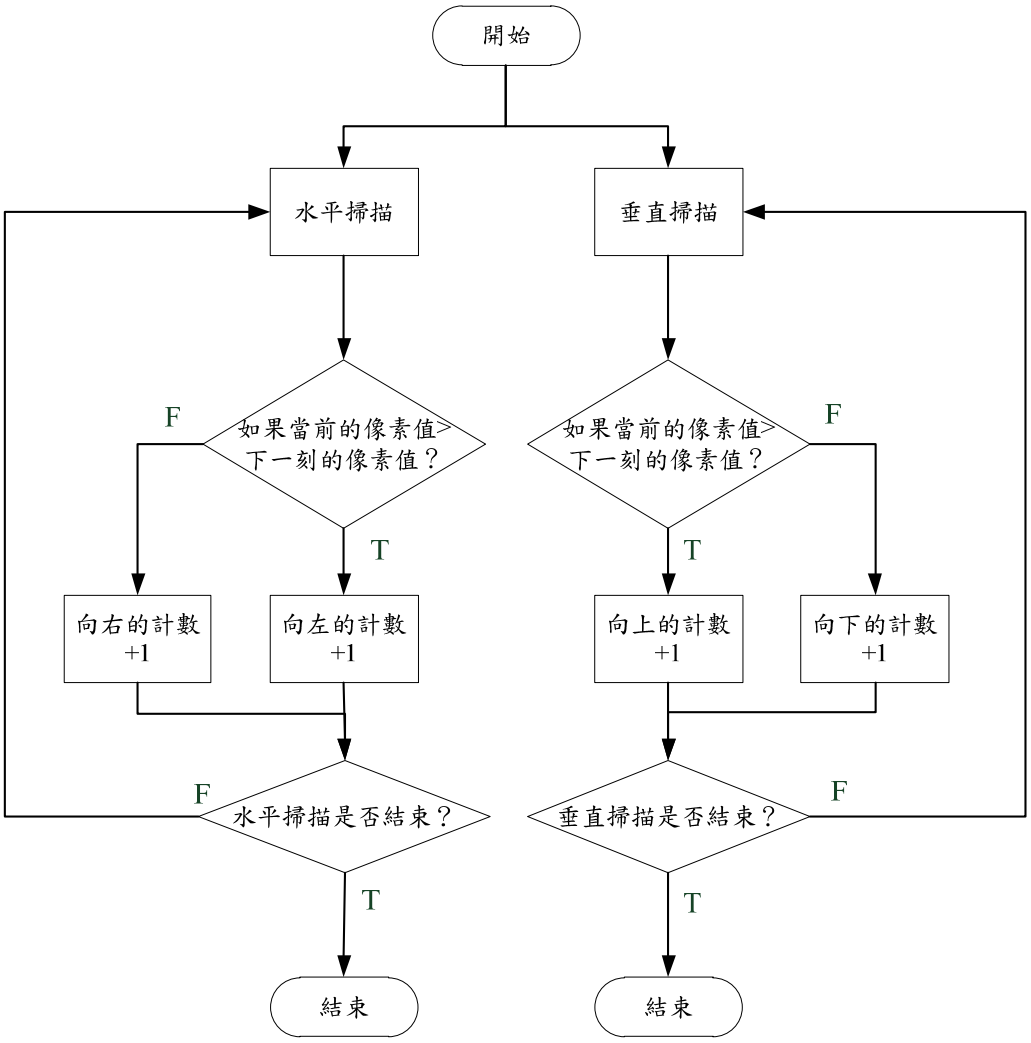


圖 4-8　使用歷史軌跡判斷權重值之流程圖

2. 模糊邏輯

經由歷史軌跡得到四個方向的計數後，首先要將這四個方向的計數正規化，如式 (4-17)到式 (4-20)所示，其中 w_{up} 、 w_{down} 、 w_{left} 、 w_{right} 分別為經過水平掃描及垂直掃描後所得的上、下、左、右的計數，λ_{left} 、 λ_{right} 、 λ_{down} 、 λ_{up} 分別為正規化後所得的結果。

$$\lambda_{left} = \frac{w_{left}}{w_{up} + w_{down} + w_{left} + w_{right}} \tag{4-17}$$

$$\lambda_{right} = \frac{w_{right}}{w_{up} + w_{down} + w_{left} + w_{right}} \tag{4-18}$$

$$\lambda_{down} = \frac{w_{down}}{w_{up} + w_{down} + w_{left} + w_{right}} \tag{4-19}$$

$$\lambda_{up} = \frac{w_{up}}{w_{up} + w_{down} + w_{left} + w_{right}} \tag{4-20}$$

如圖 4-9 到圖 4-12 為本論文所使用的歸屬函數，分別為、λ_{down}、λ_{left}、λ_{right}之歸屬函數。

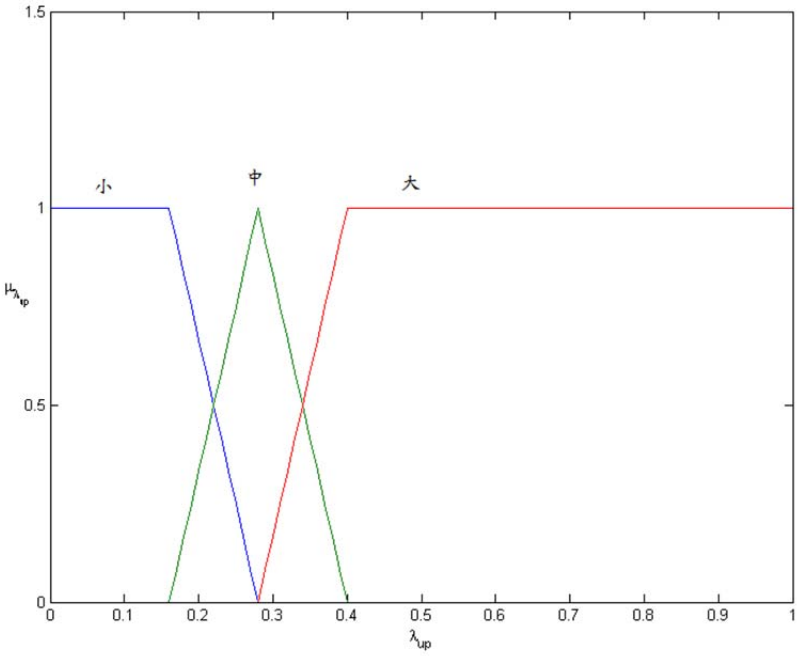


圖 4-9　　λ_{up} 之歸屬函數

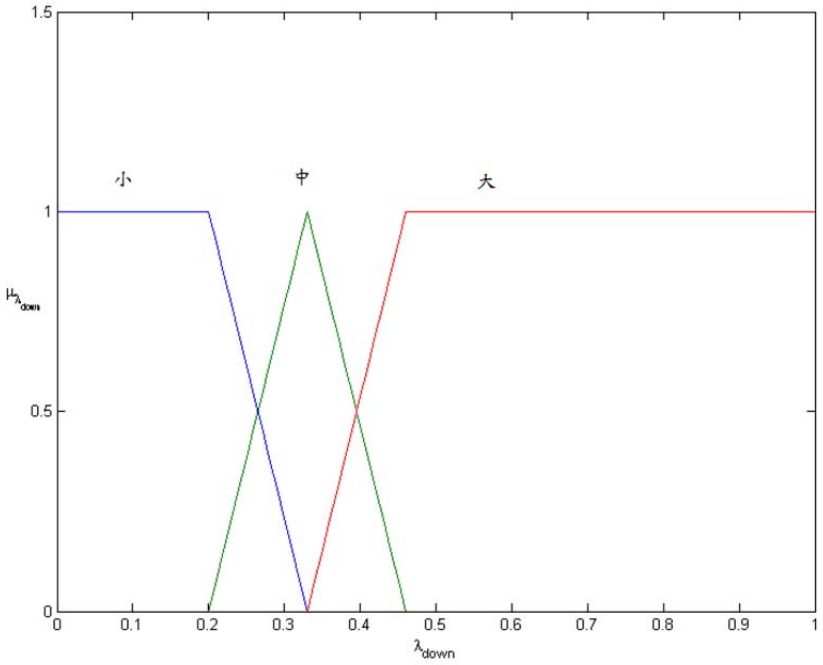


圖 4-10　　λ_{down} 之歸屬函數

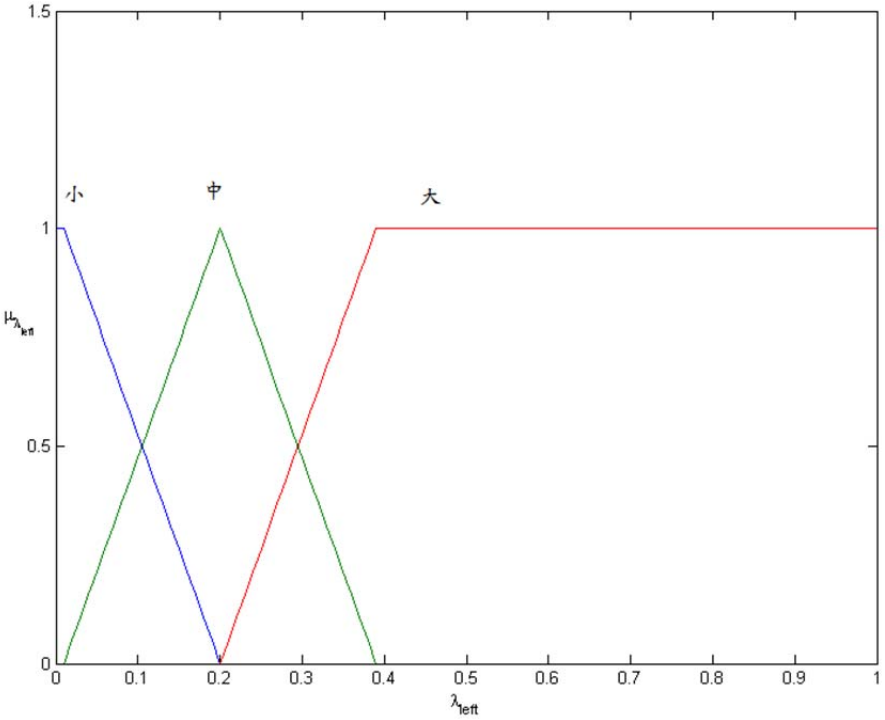


圖 4-11　λ_{left} 之歸屬函數

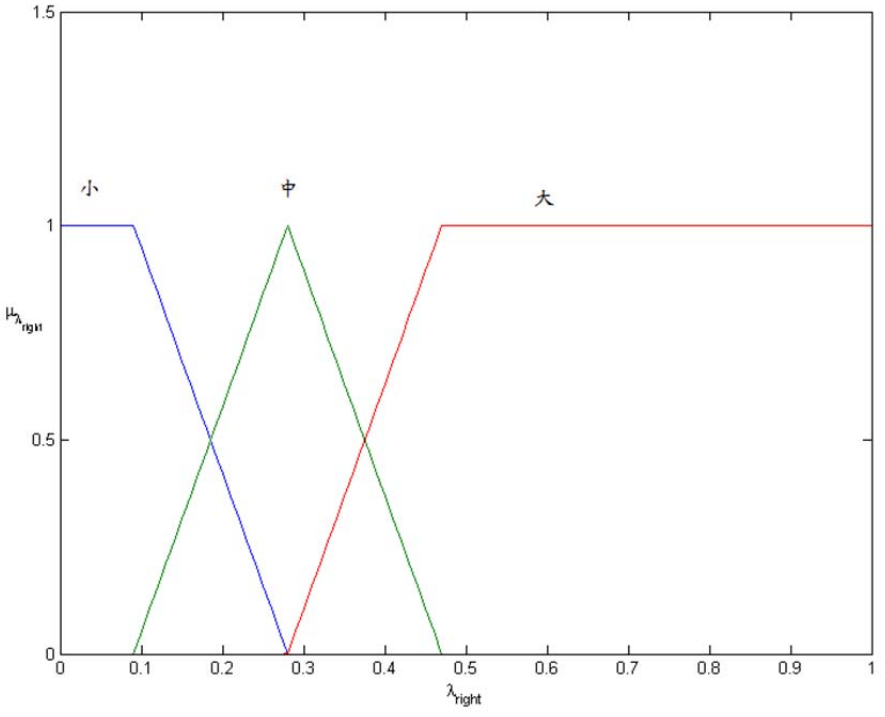
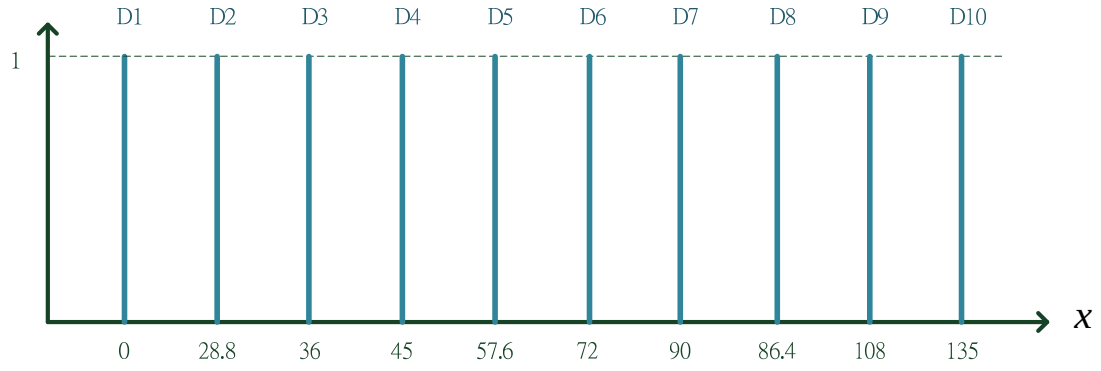
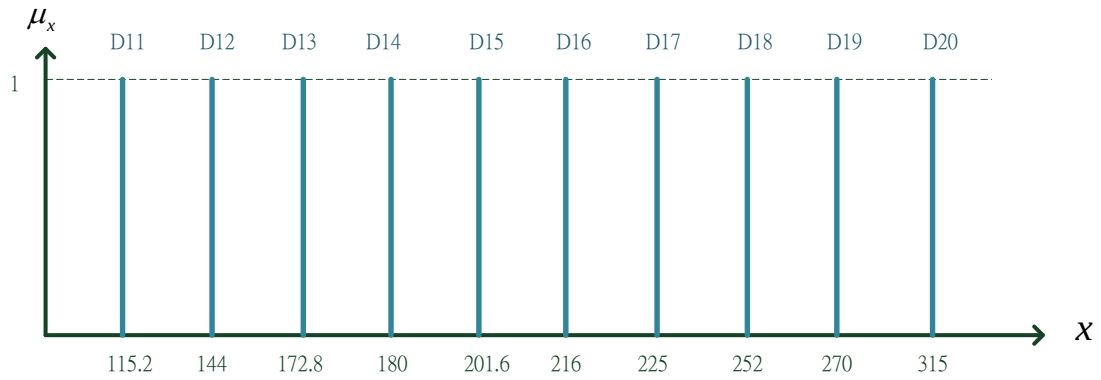


圖 4-12　λ_{right} 之歸屬函數

我們所使用的模糊推論引擎為乘積推論引擎，乘積推論引擎包含了 Madani 乘積蘊含、對所有 t-norm 運算子使用代數乘積替代、對所有 s-norm 運算子使用最大值的方法取代，模糊化使用單點模糊化，解模糊化的部分是使用中心平均值去模糊化，根據多次實驗的數據，我們訂定的規則庫如表 4-1 所示，圖 4-13 為輸出的歸屬函數。



(a)



(b)

圖 4-13　輸出之歸屬函數

表 4-1　　模糊規則庫

規則	前件部	後件部
R1	若λ_{up}為小且λ_{down}為小且λ_{left}為小且λ_{right}為中	則y為D1
R2	若λ_{up}為小且λ_{down}為小且λ_{left}為中且λ_{right}為小	則y為D12
R3	若λ_{up}為小且λ_{down}為小且λ_{left}為中且λ_{right}為大	則y為D1
R4	若λ_{up}為小且λ_{down}為小且λ_{left}為大且λ_{right}為小	則y為D14
R5	若λ_{up}為小且λ_{down}為小且λ_{left}為大且λ_{right}為中	則y為D14
R6	若λ_{up}為小且λ_{down}為中且λ_{left}為小且λ_{right}為小	則y為D16
R7	若λ_{up}為小且λ_{down}為中且λ_{left}為小且λ_{right}為中	則y為D18
R8	若λ_{up}為小且λ_{down}為中且λ_{left}為中且λ_{right}為小	則y為D14
R9	若λ_{up}為小且λ_{down}為中且λ_{left}為中且λ_{right}為中	則y為D13
R10	若λ_{up}為小且λ_{down}為中且λ_{left}為中且λ_{right}為大	則y為D15
R11	若λ_{up}為小且λ_{down}為中且λ_{left}為大且λ_{right}為中	則y為D12
R12	若λ_{up}為小且λ_{down}為大且λ_{left}為小且λ_{right}為小	則y為D19
R13	若λ_{up}為小且λ_{down}為大且λ_{left}為小且λ_{right}為大	則y為D20
R14	若λ_{up}為小且λ_{down}為大且λ_{left}為中且λ_{right}為中	則y為D16
R15	若λ_{up}為小且λ_{down}為大且λ_{left}為大且λ_{right}為小	則y為D17
R16	若λ_{up}為中且λ_{down}為小且λ_{left}為小且λ_{right}為小	則y為D6
R17	若λ_{up}為中且λ_{down}為小且λ_{left}為小且λ_{right}為中	則y為D3

R18	若λ_{up}為中且λ_{down}為小且λ_{left}為小且λ_{right}為大	則y為D1
R19	若λ_{up}為中且λ_{down}為小且λ_{left}為中且λ_{right}為小	則y為D9
R20	若λ_{up}為中且λ_{down}為小且λ_{left}為中且λ_{right}為中	則y為D5
R21	若λ_{up}為中且λ_{down}為小且λ_{left}為中且λ_{right}為大	則y為D2
R22	若λ_{up}為中且λ_{down}為小且λ_{left}為大且λ_{right}為中	則y為D8
R23	若λ_{up}為中且λ_{down}為中且λ_{left}為小且λ_{right}為中	則y為D1
R24	若λ_{up}為中且λ_{down}為中且λ_{left}為小且λ_{right}為大	則y為D1
R25	若λ_{up}為中且λ_{down}為中且λ_{left}為中且λ_{right}為小	則y為D11
R26	若λ_{up}為中且λ_{down}為中且λ_{left}為中且λ_{right}為大	則y為D1
R27	若λ_{up}為中且λ_{down}為中且λ_{left}為大且λ_{right}為小	則y為D12
R28	若λ_{up}為中且λ_{down}為中且λ_{left}為大且λ_{right}為中	則y為D12
R29	若λ_{up}為中且λ_{down}為大且λ_{left}為小且λ_{right}為小	則y為D19
R30	若λ_{up}為中且λ_{down}為大且λ_{left}為小且λ_{right}為中	則y為D15
R31	若λ_{up}為中且λ_{down}為大且λ_{left}為中且λ_{right}為小	則y為D12
R32	若λ_{up}為中且λ_{down}為大且λ_{left}為中且λ_{right}為中	則y為D16
R33	若λ_{up}為中且λ_{down}為大且λ_{left}為中且λ_{right}為大	則y為D18
R34	若λ_{up}為中且λ_{down}為大且λ_{left}為大且λ_{right}為中	則y為D14
R35	若λ_{up}為大且λ_{down}為小且λ_{left}為小且λ_{right}為小	則y為D7
R36	若λ_{up}為大且λ_{down}為小且λ_{left}為小且λ_{right}為大	則y為D4

R37	若λ_{up}為大且λ_{down}為小且λ_{left}為中且λ_{right}為中	則y為D6
R38	若λ_{up}為大且λ_{down}為小且λ_{left}為大且λ_{right}為小	則y為D10
R39	若λ_{up}為大且λ_{down}為中且λ_{left}為小且λ_{right}為小	則y為D7
R40	若λ_{up}為大且λ_{down}為中且λ_{left}為小且λ_{right}為中	則y為D2
R41	若λ_{up}為大且λ_{down}為中且λ_{left}為小且λ_{right}為大	則y為D1
R42	若λ_{up}為大且λ_{down}為中且λ_{left}為中且λ_{right}為小	則y為D8
R43	若λ_{up}為大且λ_{down}為中且λ_{left}為中且λ_{right}為中	則y為D6
R44	若λ_{up}為大且λ_{down}為中且λ_{left}為中且λ_{right}為大	則y為D3
R45	若λ_{up}為大且λ_{down}為中且λ_{left}為大且λ_{right}為中	則y為D9

最後經過解模糊化後可以得到最後的輸出，本論文採用的是中心平均去模糊化的方式，如式 (4-21)所示，其中w^l為$\bar{y}^l$所對應的高度，$\bar{y}^l$為第l個模糊集合的中心，y^*為最後的解模糊化輸出。

$$y^* = \frac{\sum_{l=1}^{M} \bar{y}^l w^l}{\sum_{l=1}^{M} w^l} \tag{4-21}$$

第5章 以軟硬體協同設計之目標物移動方向模糊辨識系統

本論文的系統使用的開發平台是具有軟硬體共同設計之 DE2 開發平台，軟硬體共同設計指的就是在 FPGA 中嵌入 NIOSII 處理器來撰寫軟體程式，且使用硬體描述語言來設計硬體電路，達到軟硬體協同的功能，圖 5-1 為本論文的系統架構圖，圖 5-2 為本系統之使用資源圖，表 5-1 為各個模組所佔用之邏輯單元與記憶體數量。本章將以六個電路分別來做說明：(1) 色彩空間轉換模組、 (2) 即時影像相減模組、 (3) 灰階區間二值化模組、 (4) 歷史軌跡模組、 (5) 上、下、左、右移動方向之計數模組、 (6) 模糊目標物移動方向辨識模組。

表 5-1　　模組所佔用之邏輯單元與記憶體數量表

模組名稱	Logic Cells	Memory Bits
SSRAM_Master	5887	131344
Frame_SW	34	0
RGB to Y	125	0
CMOS Capture	58	0
Raw to RGB	256	28728
SDRAM Control Module	579	24576
Median	6	0
SDRAM Control Module	550	20480
Historical Trajectories	119	668640

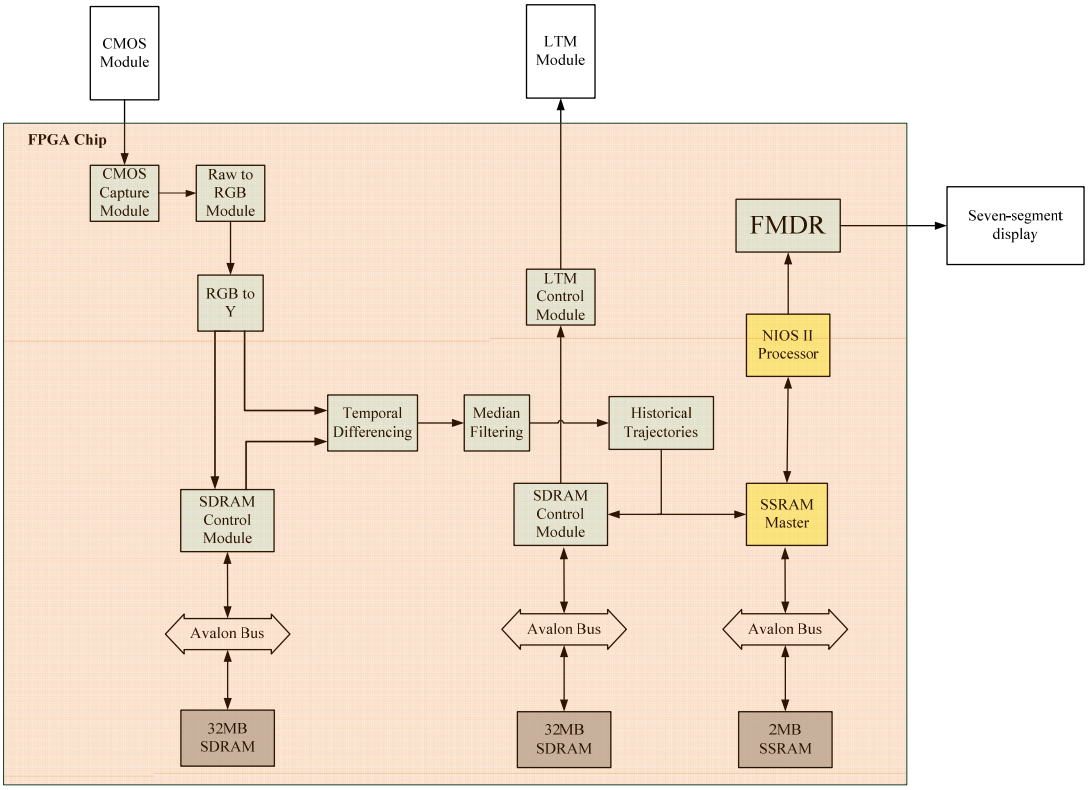


圖 5-1　　以軟硬體協同設計之模糊目標物移動方向辨識之系統架構

Flow Status	Successful - Wed Jul 18 21:48:44 2012
Quartus II Version	9.1 Build 350 03/24/2010 SP 2 SJ Full Version
Revision Name	DE2_70
Top-level Entity Name	DE2_70
Family	Cyclone II
Device	EP2C70F896C6
Timing Models	Final
Met timing requirements	No
Total logic elements	8,231 / 68,416 (12 %)
Total combinational functions	6,455 / 68,416 (9 %)
Dedicated logic registers	5,324 / 68,416 (8 %)
Total registers	5487
Total pins	530 / 622 (85 %)
Total virtual pins	0
Total memory bits	873,768 / 1,152,000 (76 %)
Embedded Multiplier 9-bit elements	4 / 300 (1 %)
Total PLLs	3 / 4 (75 %)

圖 5-2　　系統之使用資源圖

5.1 色彩空間轉換模組

此電路的功能為將 RGB 的影像轉為 Y 空間的影像，公式如式 (4-1) 所示，此模組之邏輯符號圖如圖 5-3 所示，其訊號腳位說明如表 5-2 所示，此模組的輸入為 CMOS 模組擷取到的影像，經由 RAW2RGB 模組從 Bayer Pattern 格式轉換成 RGB 格式。圖 5-4 為 RGB 轉 Y 空間模組之波形驗證。

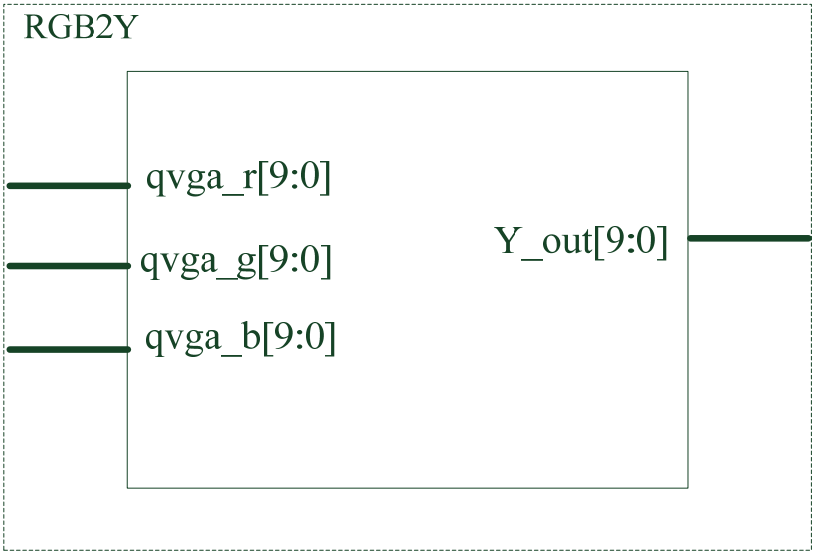


圖 5-3　　RGB 空間轉 Y 空間模組之符號圖

表 5-2　　RGB 空間轉 Y 空間模組之訊號腳位說明

訊號名稱	方向	功能說明
gvga_r	輸入	全彩影像的 R 分量
gvga_g	輸入	全彩影像的 G 分量
gvga_b	輸入	全彩影像的 B 分量
Y_out	輸出	Y 空間的分量

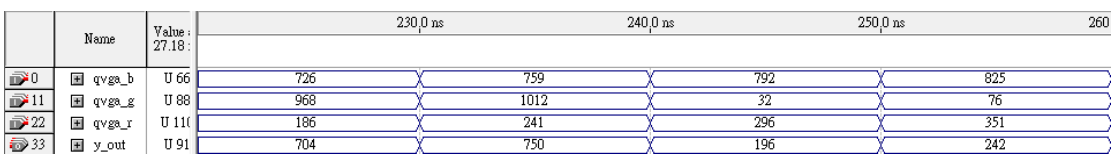


圖 5-4　　RGB 空間轉 Y 空間模組之波形驗證

5.2 影像切換模組及即時影像相減

為了能夠達到影像相減的目的，首先要知道一張畫面何時開始以及何時結束，所以要在這之前先設計一個影像切換模組，功能如表 5-3 所示，符號圖如圖 5-5 所示。

表 5-3　　影像切換模組之功能表

訊號名稱	方向	功能說明
iCLK	輸入	時脈訊號
iRST_n	輸入	重置訊號
iDVAL	輸入	影像有效畫面控制線
oX_CNT	輸出	畫面中的水平座標
oY_CNT	輸出	畫面中的垂直座標
oNDVAL	輸出	影像奇數張控制線
oNDVAL_n	輸出	影像偶數張控制線

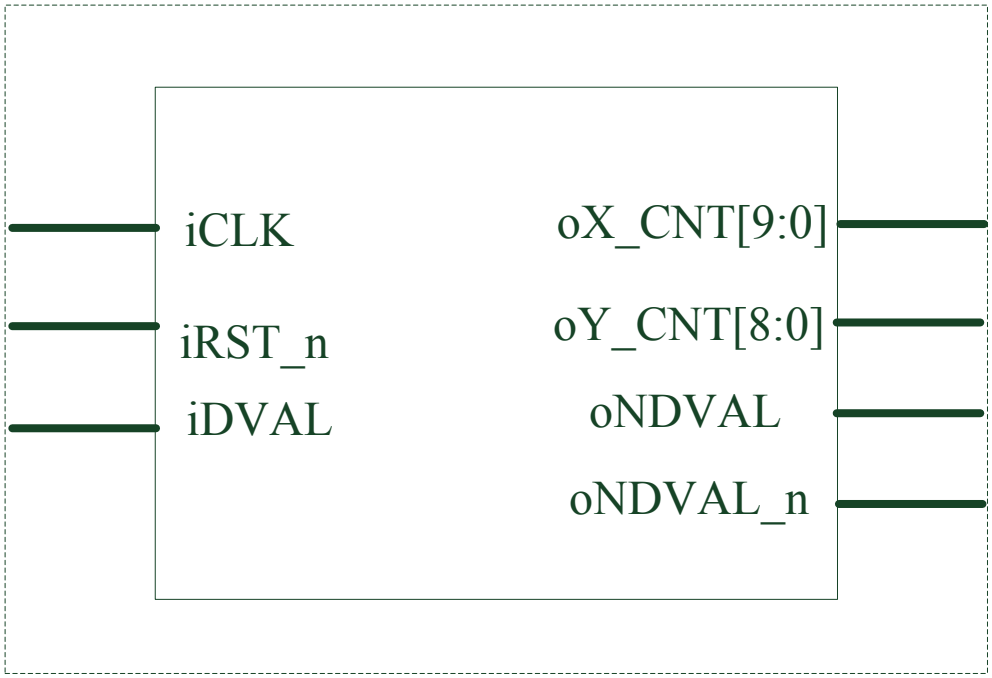


圖 5-5　　影像切換模組符號圖

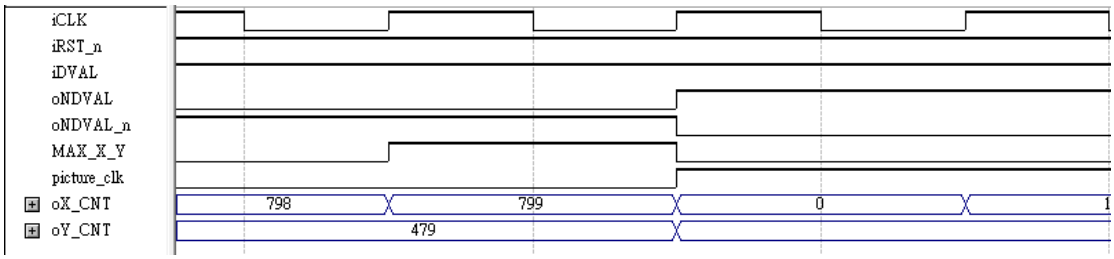


圖 5-6　　影像切換模組之波行驗證

圖 5-6 為影像切換模組之波形驗證，當訊號 iDVAL 為高位準時，則 oX_CNT 及 oY_CNT 開始計數，當 oX_CNT 及 oY_CNT 同時達到最大值，則 MAX_X_Y 為高位準，也就是一張畫面結束的時候。oNDVAL 與 oNDVAL_n 分別為奇數張畫面與偶數張畫面的控制訊號。

即時影像相減法 (Temporal Differencing Method)的硬體電路為將上述所得到的 Y_out，共 10bit，存入 SDRAM 中，等待下一個影像出現時做影像相減。當奇數個影像進來時，oNDVAL 為高位準、oNDVAL_n 為低位準，偶數個影像進來時，oNDVAL_n 為高位準、oNDVAL 為低位準，即用 oNDVAL 控制 SDRAM 的寫入，oNDVAL_n 控制 SDRAM 的讀出，藉由一讀一寫的方法即可得到當前的畫面與前一刻的畫面，再將其相減，即為影像相減的功能。

5.3　灰階區間二值化模組

在此節中，我們將即時影像相減模組的輸出當成灰階區間二值化模組的輸入，其功能為將不重要的資訊刪除並且轉成二值化影像。圖 5-7 為灰階區間二值化模組的測試波形圖。其中訊號 input1 為影像相減後的輸出，output1 為利用式 (4-9)運算後所得的輸出，其輸出結果為二值化的影像。

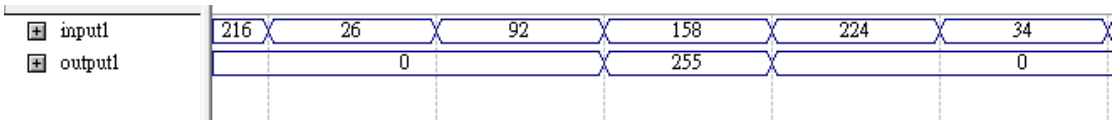


圖 5-7　灰階區間二值化模組的測試波形圖

5.4 歷史軌跡模組

歷史軌跡模組[39]之邏輯符號圖如圖 5-8 所示，其訊號腳位說明如表 5-4 所示，而內部電路設計及相關模組將會分別介紹。

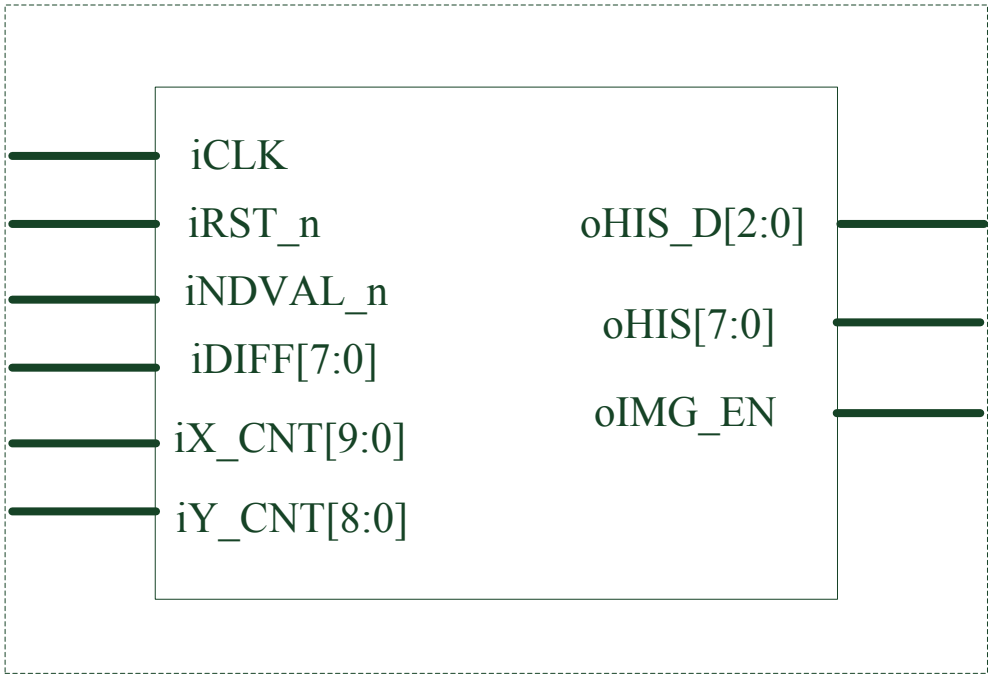


圖 5-8　歷史軌跡模組符號圖

表 5-4　歷史軌跡模組輸出入訊號說明表

訊號名稱	方向	功能說明
iCLK	輸入	時脈訊號
iRST_n	輸入	置能訊號
iNDVAL_n	輸入	影像偶數張控制線
iDIFF	輸入	移動目標物之二值化影像
iX_CNT	輸入	畫面中之 x 座標值
iY_CNT	輸入	畫面中之 y 座標值
oHIS_D	輸出	歷史軌跡 (3bit)
oHIS	輸出	歷史軌跡影像 (8bit)
oIMG_EN	輸出	影像範圍置能訊號

其中 iCLK 為此模組所需要的時脈訊號，iNDVAL_n 則為影像範圍之致能訊號，使用 iNDVAL_n、iX_CNT[0]、iY_CNT[0]這三條觸發訊號，便能夠將SDRAM所讀出來之 800×480 大小影像縮小至原始影像的1/4倍，可以使得計算量減少 4 倍，並且保持足夠的影像資訊。如圖 5-9 為影像縮小的示意圖。

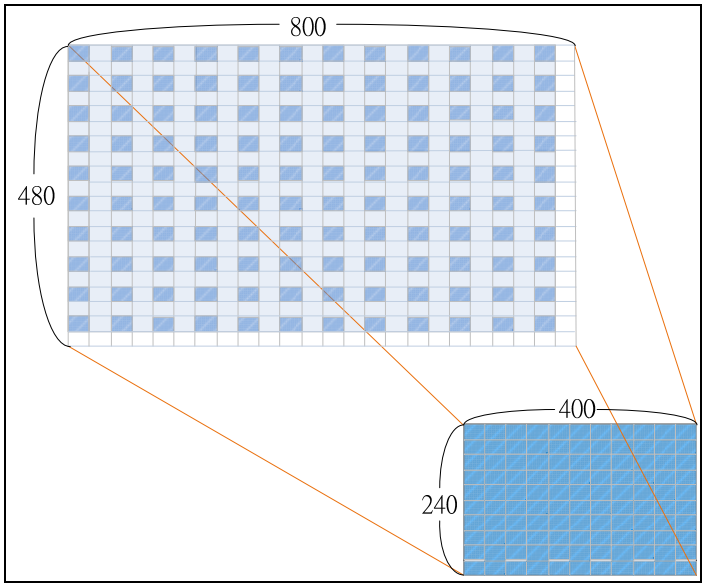


圖 5-9　　影像縮小的示意圖[39]

如圖 5-9 將原始 800×480 大小的影像縮小成400×240後，接者，我們利用七個大小各自為 400×240 且資料寬度為 1 bit 的 LINE_BUFFER，每個 LINE_BUFFER 紀錄其中一張不同時刻的影像，其示意圖如圖 5-10 所示，訊號 din 為灰階區間二值化後的輸出，即為當前的畫面，再將 din 做 and 運算輸入到 LINE_BUFFER_1 中，而 LINE_BUFFER_2 的輸入為上一刻影像的畫面，LINE_BUFFER_3 為上兩刻影像的畫面，以此類推，所以此模組利用七個 FIFO 儲存七張畫面。再經過解碼可以在同一個畫面中顯示影像的歷史軌跡 (oHIS)與影像的編號 (oHIS_D)。

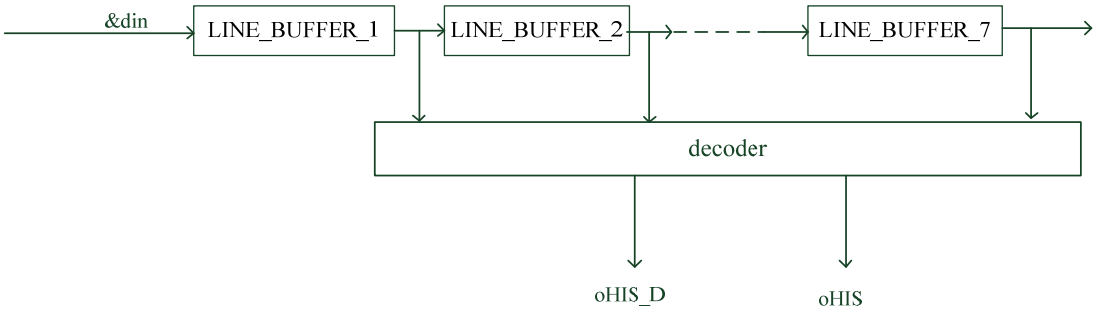


圖 5-10　　歷史軌跡示意圖

LINE_BUFFER 即是線路緩衝器，其邏輯符號圖如圖 5-11 所示，以及訊號腳位說明如表 5-5 所示。LINE_BUFFER 功能在於可對連續的數據進行儲存，防止資料的流失，當 LINE_BUFFER 滿的時候便會依序地讀出，達到暫存的效果。LINE_BUFFER 的使用技巧主要是用於當資料的準備時間超過一個時脈時，且又必須同時將多筆資料與其他模組之間做傳輸溝通時，此時利用 LINE_BUFFER 暫存先前的資料做為緩衝，直到一筆資料準備完成致能寫入訊號 (wrreg)，將資料寫入 LINE_BUFFER 內並準備下一筆資料，直到寫入資料數量達到一定數量時，致能讀取訊號 (rdreq)，在延遲數個時脈後可以由輸出資料 (q)得到依序的資料。而在本歷史軌跡模組中，為了能夠暫存七張歷史動量訊息，所以共使用到七個大小為圖 5-11 所示之 LINE_BUFFER 模組。

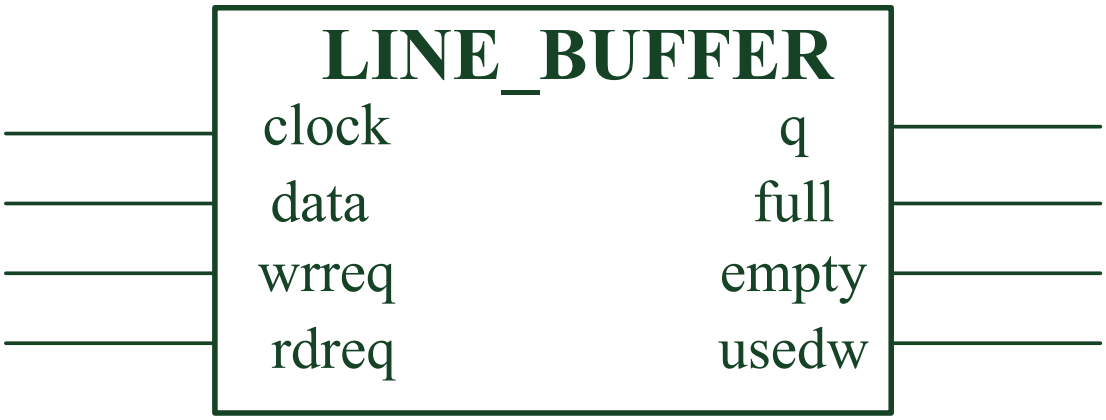


圖 5-11　LINE_BUFFER 模組邏輯符號圖

表 5-5　LINE_BUFFER 模組輸出入訊號說明表

訊號名稱	方向	功能說明
clock	輸入	時脈訊號
data	輸入	初始訊號
wrreq	輸入	寫入訊號

rdreq	輸入	讀出訊號
q	輸出	輸出訊號
full	輸出	溢位訊號
empty	輸出	欠位訊號
usedw	輸出	資料訊號

5.5 使用 SOPC 計算上、下、左、右移動方向之計數模組

在此節中，我們將歷史軌跡模組之其中一條輸出訊號 (oHIS_D)當作此模組的輸入訊號，此訊號為 3bit，其 1~7 的數值代表著不同時刻的畫面，7 代表最新的畫面，1 代表最舊的畫面。本論文之 SOPC 架構如圖 5-12 所示。

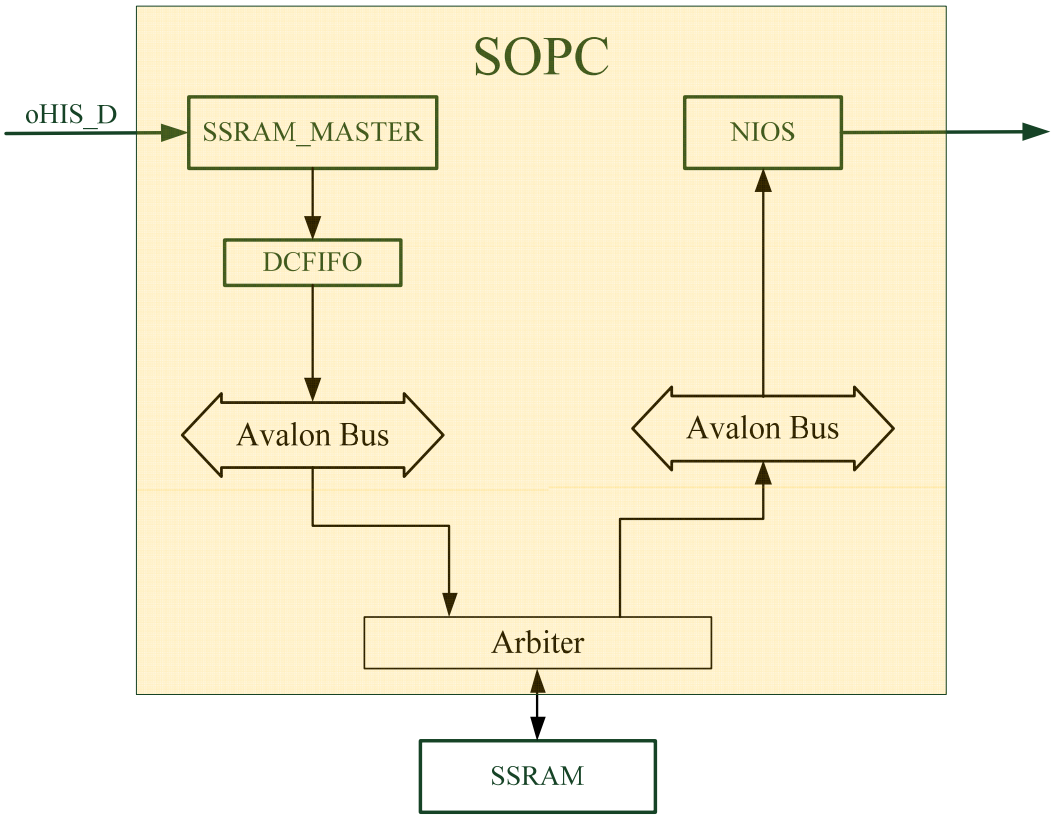


圖 5-12　SOPC 之計數模組架構圖

在本論文中，將歷史軌跡輸出的畫面透過 SSRAM_MASTER 模組且利用 burst 傳輸傳入 DCFIFO，SSRAM 再由 DCFIFO 抓取資料，此方法可

以加速資料的傳輸，NIOS 再從 SSRAM 抓取資料並計算上、下、左、右權重然後輸出。

此系統使用的 DCFIFO 的深度為 2048 筆，burst 傳輸之資料長度為 256 筆，將下來將介紹 DCFIOC 的訊號。

交握式傳輸和 DCFIFO 傳輸是兩種跨時域電路設計最常用的方式，但由於交握式在傳送每一筆資料時，必須等待確認 (ACK)訊號。而 DCFIFO 的寫入和讀取可在同一時間而且不需等待確認 (ACK)訊號。因此如果需要有高速的傳輸資料能力，使用 DCFIFO 會是最佳的選擇。DCFIFO 之電路符號圖如圖 5-13 所示，訊號說明如表 5-6 所示。

圖 5-13　DCFIFO 之電路符號圖

表 5-6　DCFIFO 模組輸出入訊號說明表

訊號名稱	方向	功能說明
data	輸入	寫入 FIFO 的資料。
rdclk	輸入	讀取時脈訊號。
wrclk	輸入	寫入時脈訊號。
wrreq	輸入	寫入需求訊號，當 wrreq 為 high 與 wclk 正緣觸發時將 data 寫入 FIFO。
rdreq	輸入	讀取需求訊號，當 rdreq 為 high 與 wclk 正緣觸發時將 data 從 FIFO 讀

		出。
aclr	輸入	非同步重置寫入與讀取的位址指標
rdfull	輸出	滿的讀取旗標
wrfull	輸出	滿的寫入旗標
rdempty	輸出	空的讀取旗標
wrempty	輸出	空的寫入旗標
rdusedw	輸出	讀取使用之旗標。表示 FIFO 裡的資料數
wrusedw	輸出	寫入使用之旗標。表示 FIFO 裡的資料數
q	輸出	FIFO 的資料輸出

5.6 目標物移動方向模糊辨識模組

此節將從 NIOS 計算出上、下、左、右的計數，用模糊推論邏輯判斷物體的移動方向使用的規則庫如表 4-1 所示，輸入歸屬函數如圖 4-9~圖 4-12 所示，輸出歸屬函數如圖 4-13 所示，推論引擎使用乘積推論引擎。

圖 5-14 為模糊方向辨識模組 (Fuzzy Moving Direction Recognition, FMDR)的邏輯符號圖，表 5-7 為模糊方向辨識模組之輸出入訊號腳，圖 5-15 為波形驗證圖。

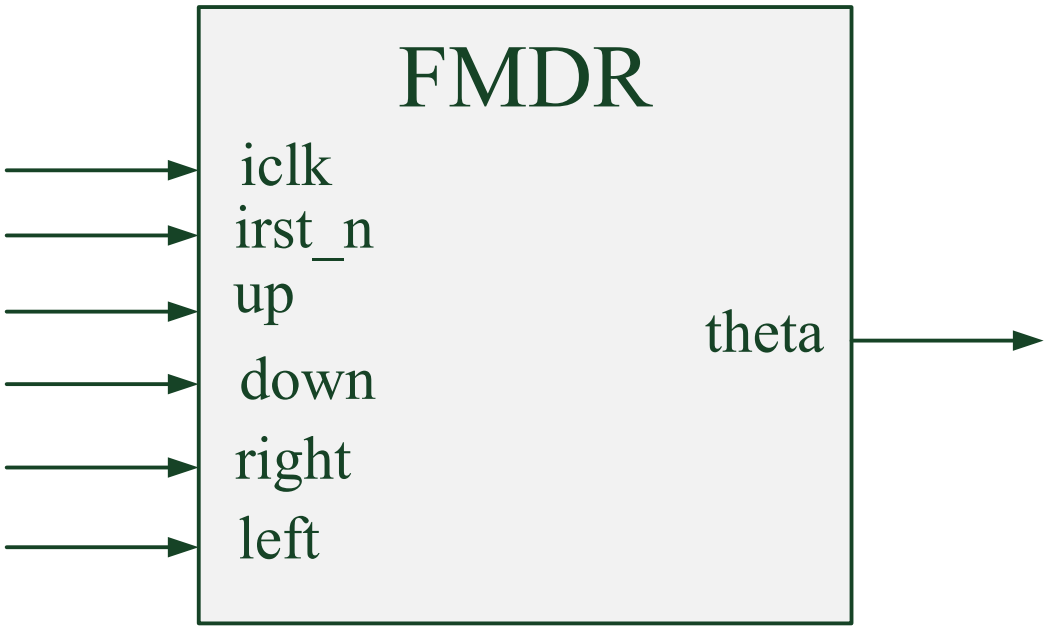


圖 5-14　模糊方向辨識模組之邏輯符號圖

表 5-7　模糊方向辨識模組之輸出入訊號腳

訊號名稱	方向	功能說明
iclk	輸入	時脈訊號
irst_n	輸入	重置訊號
up	輸入	向上的計數訊號
down	輸入	向下的計數訊號
right	輸入	向右的計數訊號
left	輸入	向左的計數訊號
theta	輸出	移動物體之預估角度訊號

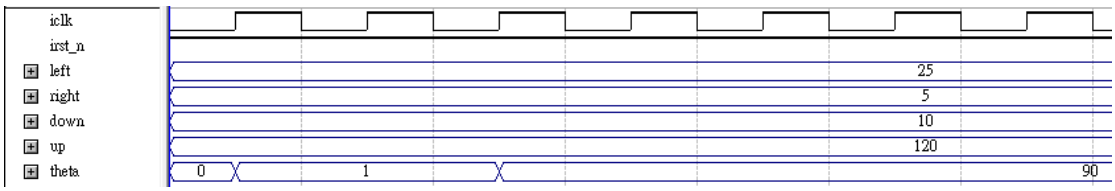


圖 5-15　模糊方向辨識模組之波形驗證圖

圖 5-16 為目標物移動方向模糊辨識模組之模組架構圖，此模組包含兩個模組，一個是 Fuzzy_member 模組，此模組的功能為輸入之歸屬函數，因為本文使用四個輸入歸屬函數，所以有四個 Fuzzy_member 模組。另外一個是 Product_md 模組，此模組的功能為實現乘積推論引擎的功能，由於本文使用了 45 個規則庫，所以有 45 個 Product_md 模組。之後再將每個規則的輸出相加得到 fuzzy 模組的輸出。

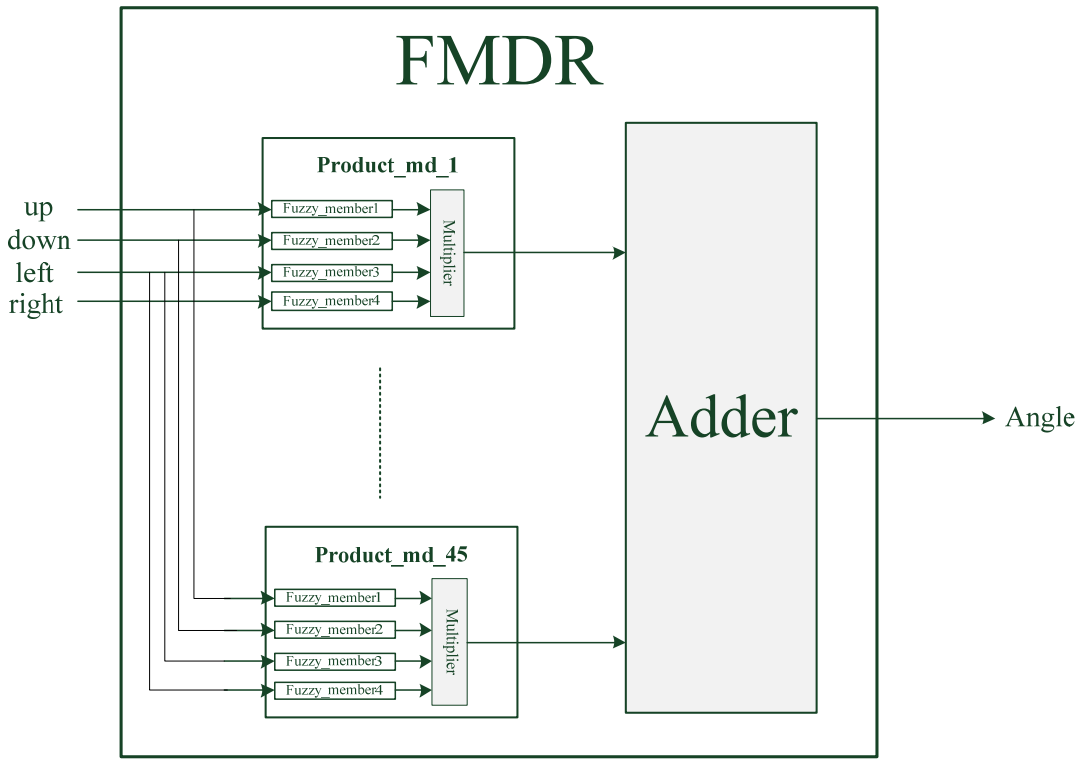


圖 5-16　模糊方向辨識之架構圖

其中 Fuzzy_member1 到 Fuzzy_member4 之模組分別為為將上、下、左、右之正歸化計數代入上、下、左、右的歸屬函數，再將所得的四個結果與規則上的角度做連乘，由於有 45 條規則，所以必須將 45 條分別得到的解果作相加，即為最後的結果。

此系統的設計方式是藉由多次實驗所得的上、下、左、右方向的計數，取上計數的最大值與上的最小值當成輸入歸屬函數的下降點與上升點，左、下、右的方式也是依照實驗的數據取最大值與最小值的方式來決定輸入歸屬函數。

模糊規則庫若依排列組合的方式，總共有 81 條規則，但是本論文刪除掉一些比較不可能的情況，例如上、下、左、右都是屬於大的情況是不存在的，為了避免不必要的計算，所以本論文只設計 45 條規則庫，如表表 4-1 當作本論文的規則庫。

解模糊化經由本論文的實驗結果是以中心平均去模糊化方法可以得到較好的推論結果，而最後由於本論文為了要實現在 FPGA 上，所以本論文使用了計算較方便的乘積推論引擎。

第6章 實驗結果

本章節將分成兩部分，軟體模擬結果與以 DE2-70 多媒體開發版呈現實驗之結果，此實驗的移動物件為單目標物體，若在影像中有兩個以上的目標物被偵測出，則此系統會以資訊較多的物體當作判斷方向的依據，也就是說，在影像中的主運動體會被此系統判斷成移動方向的物體。

6.1 軟體模擬

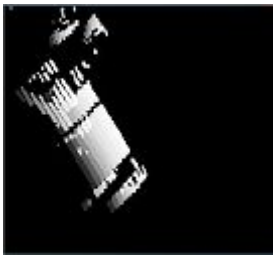
在此實驗中我們利用 webcam 擷取即時動態影像，偵測出移動的物體，再利用歷史軌跡產生移動的軌跡。圖 6-1 (a)~ (p)為歷史軌跡的結果圖，每張圖代表著物體的移動方向，圖 6-1 (a1)~圖 6-1 (a5)之物體移動方向大約為 0 度，圖 6-1 (b1)~圖 6-1 (b5)之物體移動方向大約為 22.5 度，以此類推，圖 6-1 (p1)~圖 6-1 (p5) 之物體移動方向大約為 337.5 度。

圖中越白的區塊代表越接近現在的時間，越黑的區塊代表離現在的時刻越遠，而全黑代表沒有此區塊沒有物體的移動。

四種運動方向之即時動量偵測權重值及估測方向如表 6-1 所示，其中(a4)所算出來的誤差最大，因為角度是以 360 度為一個循環，所以估測方向 357 度其實是非常接近 0 度的方向，

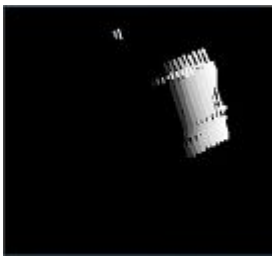
(a1) $\theta = 0^{\circ}$

(b1) $\theta = 17.98^{\circ}$

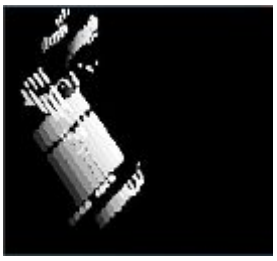


(c1) $\theta = 45^{\circ}$

(a2) $\theta = 0^{\circ}$

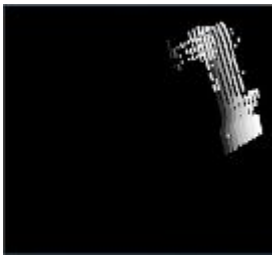


(b2) $\theta = 7.71^{\circ}$

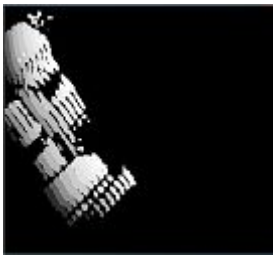


(c2) $\theta = 69.76^{\circ}$

(a3) $\theta = 0^{\circ}$



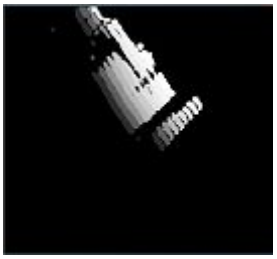
(b3) $\theta = 4.97^{\circ}$



(c3) $\theta = 12.4^{\circ}$

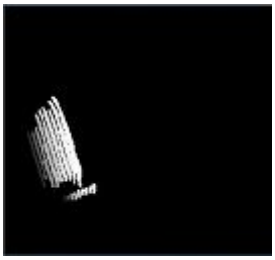
(a4) $\theta = 357.89^{\circ}$

(b4) $\theta = 12.44^{\circ}$



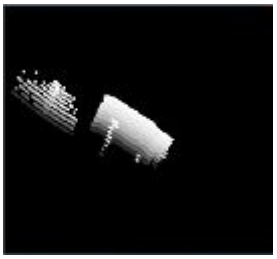
(c4) $\theta = 45^{\circ}$

(a5) $\theta = 0^{\circ}$

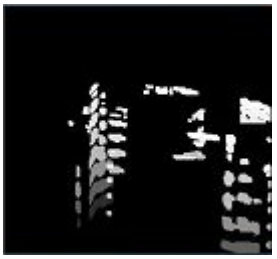


(b5) $\theta = 16.5^{\circ}$

(c5) $\theta = 45.11^{\circ}$

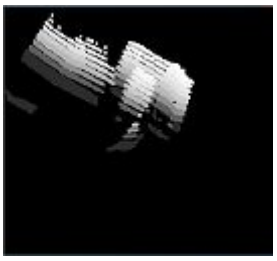


(d1) $\theta = 84.13^{\circ}$



(e1) $\theta = 93.93^{\circ}$

(f1) $\theta = 135^{\circ}$



(d2) $\theta = 45^{\circ}$

(e2) $\theta = 90^{\circ}$

(f2) $\theta = 90^{\circ}$

(d3) $\theta = 66.29^{\circ}$

(e3) $\theta = 90^{\circ}$

(f3) $\theta = 91^{\circ}$

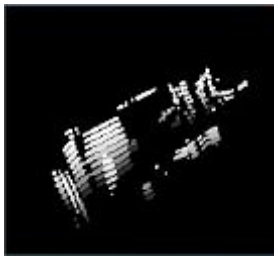
(d4) $\theta = 86.52^{\circ}$

(e4) $\theta = 90^{\circ}$

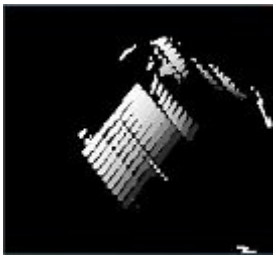
(f4) $\theta = 135^{\circ}$

(d5) $\theta = 89^{\circ}$

(e5) $\theta = 90^{\circ}$



(f5) $\theta = 109.67^{\circ}$



(g1) $\theta = 135^{\circ}$

(h1) $\theta = 180^{\circ}$

(i1) $\theta = 180^{\circ}$

(g2) $\theta = 135^{\circ}$

(h2) $\theta = 163.1^{\circ}$

(i2) $\theta = 180^{\circ}$

(g3) $\theta = 135^{\circ}$

(h3) $\theta = 135^{\circ}$

(i3) $\theta = 180^{\circ}$

(g4) $\theta = 135^{\circ}$

(h4) $\theta = 135^{\circ}$

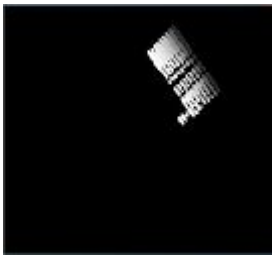
(i4) $\theta = 179.48^{\circ}$

(g5) $\theta = 135^{\circ}$

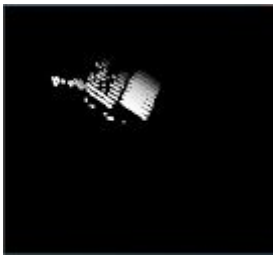
(h5) $\theta = 154.06^{\circ}$

(i5) $\theta = 180^{\circ}$

(j1) $\theta = 225^{\circ}$



(k1) $\theta = 225^{\circ}$



(l1) $\theta = 225^{\circ}$

(j2) $\theta = 225^{\circ}$

(k2) $\theta = 225^{\circ}$

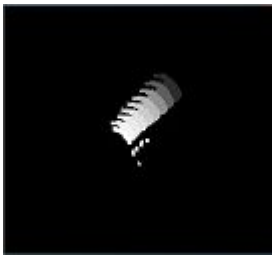
(l2) $\theta = 225^{\circ}$

(j3) $\theta = 180^{\circ}$

(k3) $\theta = 225^{\circ}$

(l3) $\theta = 225^{\circ}$

(j4) $\theta = 180^{\circ}$

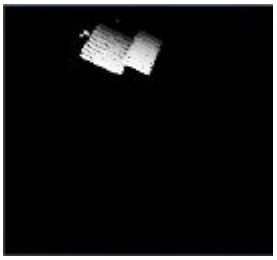


(k4) $\theta = 225^{\circ}$

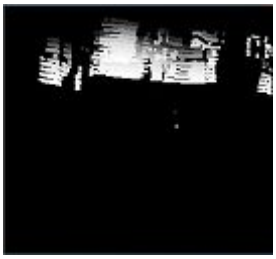
(l4) $\theta = 235.91^{\circ}$

(j5) $\theta = 189.19^{\circ}$

(k5) $\theta = 225^{\circ}$



(l5) $\theta = 237.4^{\circ}$

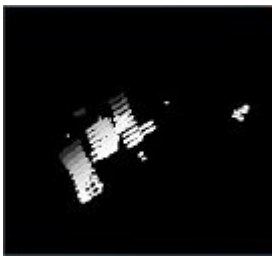


(m1) $\theta = 270^{\circ}$

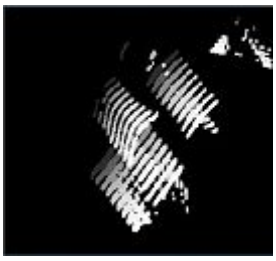
(n1) $\theta = 298.46^{\circ}$

(o1) $\theta = 315^{\circ}$

(m2) $\theta = 270^{\circ}$



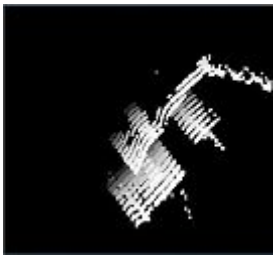
(n2) $\theta = 286.74^{\circ}$



(o2) $\theta = 314.26^{\circ}$

(m3) $\theta = 270^{\circ}$

(n3) $\theta = 281.45^{\circ}$



(o3) $\theta = 347.9^{\circ}$

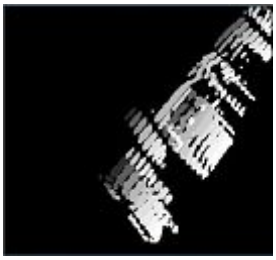
(m4) $\theta = 261.07^{\circ}$

(n4) $\theta = 277.12^{\circ}$

(o4) $\theta = 315^{\circ}$

(m5) $\theta = 270^{\circ}$

(n5) $\theta = 306.62^{\circ}$

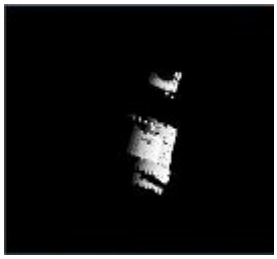


(o5) $\theta = 315^{\circ}$

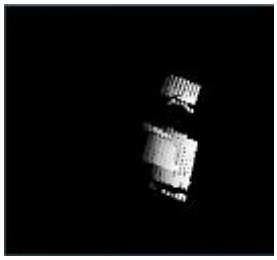
(p1) $\theta = 315^{\circ}$

(p2) $\theta = 0^{\circ}$

(p3) $\theta = 318.38^{\circ}$



(p4) $\theta = 315^{\circ}$



(p5) $\theta = 315^{\circ}$

圖 6-1　　多種方向之歷史軌跡圖

表 6-1　四種運動方向之即時動量偵測權重值及估測方向

History Frame Number	Moving Direction				Estimated angle θ	Error angle
	Up counter	Down Counter	Left Counter	Right Counter		
(a1)	658	704	481	1144	0	0
(a2)	581	644	540	1583	0	0
(a3)	449	586	404	1223	0	0
(a4)	486	550	361	930	357.8957	357.8957
(a5)	191	292	100	286	0	0
(b1)	515	524	205	484	17.98004	4.51996
(b2)	601	584	277	751	7.716049	14.783951
(b3)	751	785	459	962	4.970545	17.529455
(b4)	697	654	411	926	12.44697	10.05303
(b5)	606	594	221	596	16.50943	5.99057
(c1)	988	840	648	1213	45	0
(c2)	1041	905	656	1242	69.76497	24.76497
(c3)	1310	1026	736	1474	12.40585	32.59415
(c4)	780	634	351	838	45	0
(c5)	879	722	476	921	45.11284	0.11284
(d1)	1006	661	318	688	84.138	16.638
(d2)	1466	1102	720	1041	45	22.5
(d3)	1454	1102	697	1031	66.29817	1.20183
(d4)	955	689	304	573	86.52319	19.02319
(d5)	645	489	246	453	89.00156	21.50156
(e1)	851	774	487	664	93.93638	3.93638
(e2)	1646	1243	981	945	90	0
(e3)	759	583	306	306	90	0
(e4)	429	395	136	163	90	0
(e5)	603	566	350	358	90	0
(f1)	895	699	550	417	135	22.5
(f2)	798	643	459	399	90	22.5
(f3)	894	772	496	433	91.00226	21.49774
(f4)	1186	1025	912	738	135	22.5
(f5)	1103	927	795	659	109.6742	2.8258
(g1)	1220	1047	1098	860	135	0
(g2)	925	724	855	493	135	0
(g3)	1013	891	1027	742	135	0

(g4)	506	477	390	329	135	0
(g5)	802	698	952	505	135	0
(h1)	578	624	792	537	180	22.5
(h2)	479	471	667	347	163.1009	5.6009
(h3)	581	610	798	447	135	22.5
(h4)	424	482	619	321	135	22.5
(h5)	765	719	917	567	154.0615	3.4385

6.2 以 DE2-70 多媒體開發版呈現之實驗結果

在此實驗中我們利用 CMOS Camera 擷取即時動態影像，偵測出移動的物體，再利用歷史軌跡產生移動的軌跡，並顯示在 LTM 顯示器上，再利用 NIOS 計算四個方向之權重並當作模糊模組的輸入，進而計算出物體的移動方向並顯示在七段顯示器上。圖 6-2 為本次的實驗環境圖。

圖 6-2　　實驗環境圖

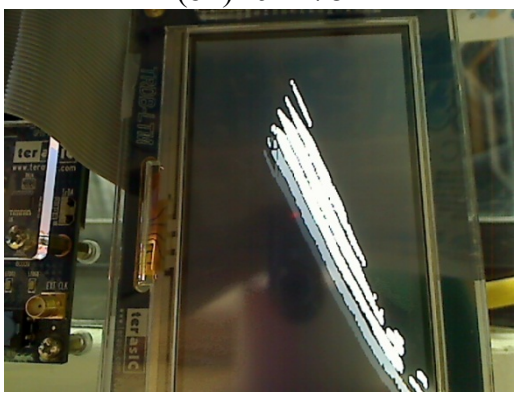
在此實驗中我們利用 D5M 擷取即時動態影像，偵測出移動的物體，

再利用歷史軌跡產生移動的軌跡。圖 6-3 (a)~ (h)為歷史軌跡的結果圖，每個圖代表著每個物體的移動方向，圖 6-3 (a1)~圖 6-3 (a5)之物體移動方向大約為 0 度，圖 6-3 (b1)~圖 6-3 (b5)之物體移動方向大約 45 度，以此類推，圖 6-3 (h1)~圖 6-3 (h5) 之物體移動方向大約為 315 度。

(a1) $\theta = 27^{\circ}$

(b1) $\theta = 78^{\circ}$

(a2) $\theta = 16^{\circ}$



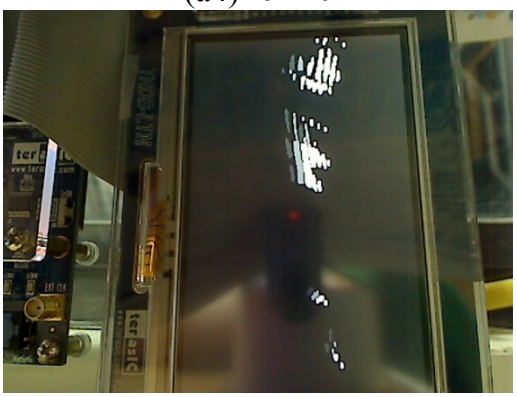
(b2) $\theta = 45^{\circ}$

(a3) $\theta = 21^{\circ}$

(b3) $\theta = 45^{\circ}$

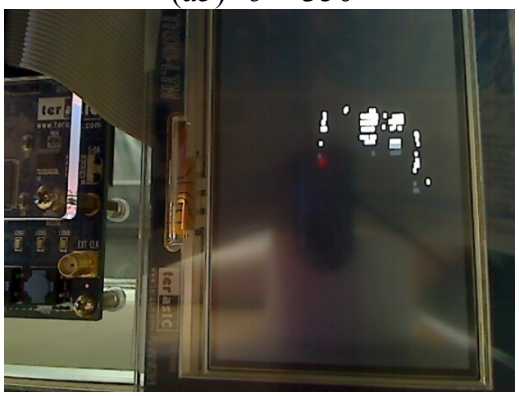
(a4) $\theta = 0^{\circ}$

(b4) $\theta = 65^{\circ}$

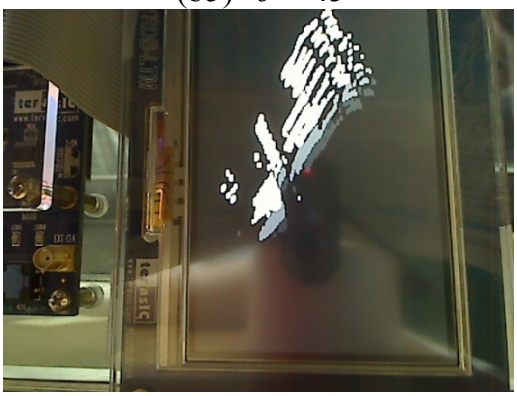


(a5) $\theta = 350^{\circ}$

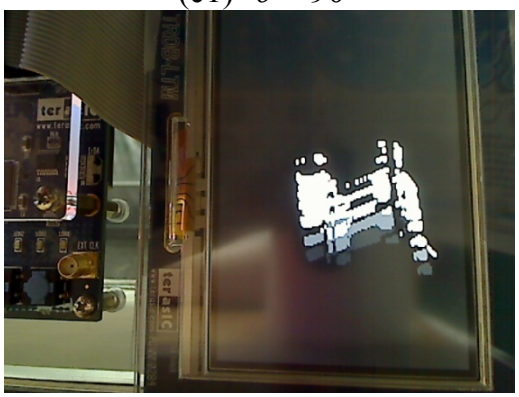
(b5) $\theta = 45^{\circ}$



(c1) $\theta = 90^{\circ}$



(d1) $\theta = 135^{\circ}$

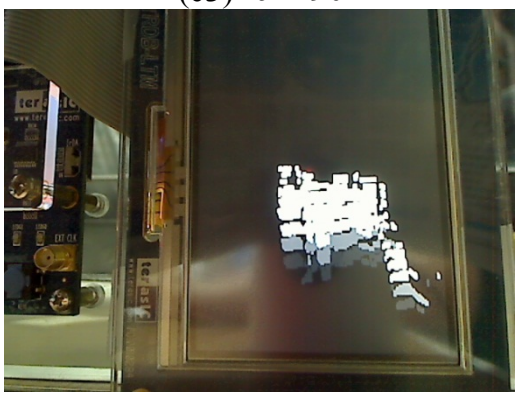


(c2) $\theta = 96^{\circ}$

(d2) $\theta = 135^{\circ}$

(c3) $\theta = 90^{\circ}$

(d3) $\theta = 135^{\circ}$



(c4) $\theta = 90^{\circ}$

(d4) $\theta = 135^{\circ}$

(c5) $\theta = 90^{\circ}$

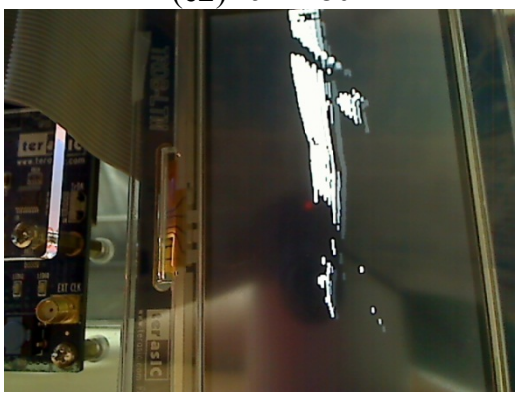
(d5) $\theta = 135^{\circ}$

(e1) $\theta = 180^{\circ}$

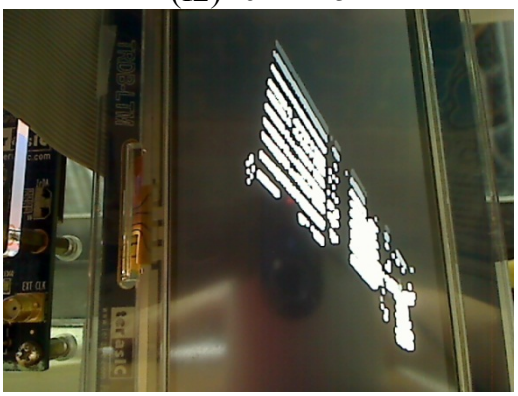
(f1) $\theta = 225^{\circ}$

(e2) $\theta = 180^{\circ}$

(f2) $\theta = 225^{\circ}$



(e3) $\theta = 180^{\circ}$



(f3) $\theta = 225^{\circ}$

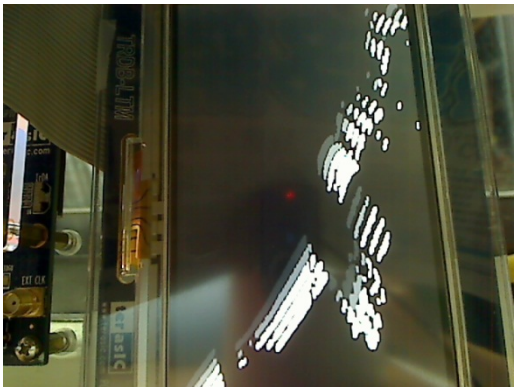
(e4) $\theta = 180^{\circ}$

(f4) $\theta = 243^{\circ}$

(e5) $\theta = 180^{\circ}$

(f5) $\theta = 225^{\circ}$

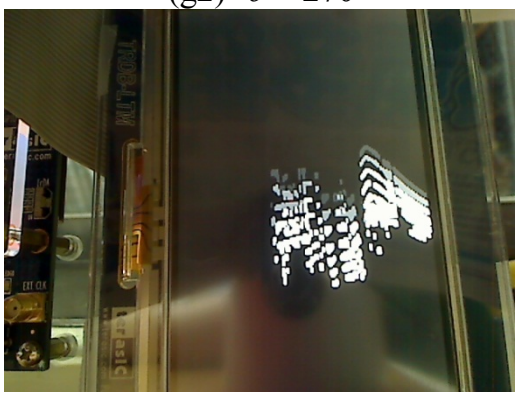
(g1) $\theta = 270^{\circ}$



(h1) $\theta = 297^{\circ}$

(g2) $\theta = 270^{\circ}$

(h2) $\theta = 291^{\circ}$



(g3) $\theta = 270^{\circ}$

(h3) $\theta = 289^{\circ}$

(g4) $\theta = 270^{\circ}$

(h4) $\theta = 281^{\circ}$

(g5) $\theta = 270^{\circ}$

(h5) $\theta = 281^{\circ}$

圖 6-3　　多種方向之歷史軌跡圖

表 6-2　　四種運動方向之即時動量偵測權重值及估測方向

History Frame Number	Moving Direction				Estimated angle θ	Error angle
	Up counter	Down Counter	Left Counter	Right Counter		
(a1)	104	23	52	309	27	27
(a2)	187	118	90	483	16	16
(a3)	129	38	55	440	21	21
(a4)	101	49	68	481	0	0
(a5)	80	67	54	244	350	350
(b1)	425	124	117	312	78	33
(b2)	555	5	7	490	45	0
(b3)	642	15	11	503	45	0
(b4)	635	62	29	349	65	20
(b5)	484	43	8	261	45	0
(c1)	98	5	0	15	90	0
(c2)	523	94	144	124	96	6
(c3)	654	85	118	143	90	0
(c4)	657	66	115	151	90	0
(c5)	479	40	34	83	90	0
(d1)	529	20	360	9	135	0
(d2)	293	14	251	9	135	0
(d3)	166	14	152	14	135	0
(d4)	340	27	360	4	135	0
(d5)	308	26	293	6	135	0
(e1)	162	245	722	106	180	0
(e2)	22	49	366	28	180	0
(e3)	132	143	595	80	180	0

(e4)	46	74	656	27	180	0
(e5)	54	86	499	55	180	0
(f1)	41	655	474	37	225	0
(f2)	48	638	289	57	225	0
(f3)	24	609	416	31	225	0
(f4)	120	484	282	123	243	18
(f5)	69	544	400	71	225	0
(g1)	24	666	142	96	270	0
(g2)	87	675	199	188	270	0
(g3)	58	671	182	136	270	0
(g4)	19	376	82	73	270	0
(g5)	10	601	129	139	270	0
(h1)	110	640	99	563	297	18
(h2)	40	751	39	391	291	24
(h3)	50	727	45	390	289	26
(h4)	98	748	85	445	281	34
(h5)	97	858	88	486	281	34

由表 6-2 可知本論文的系統之估測物體移動之方向與實際方向的誤差在可容許範圍內，接下來我們將探討以 Nios II 純軟體設計目標物移動方向模糊辨識系統，以及以軟硬體協同設計之目標物移動方向模糊辨識系統，在 133 MHz 下在各步驟所耗費的時間，如表 6-3 所示。而 Performance improvement 代表的是軟體時間除以硬體時間所得的結果。

表 6-3　純軟體與全硬體設計之實驗結果

Computational modules	**Time elapsed (μs)**		**Performance improvement**	logic element	fmax
	HW	**SW**			
RGB to Y module	7,680	49,921.47	6.500191406	125/68416	---
Frame_SW module	15,360	130,561.77	8.500115234	35/68416	327.87MHz
Temporal Differencing module	7,680	49,921.47	6.500191406	55/68416	---
Range of image binarization	7,680	46,081.46	6.000190104	4/68416	---
Historical Trajectories module	107,520	645,121.45	6.000013486	118/68416	105.37MHz
Counting module (NIOS)	7,680	18,241.74	2.375	6019/68416	135.35 MHz
Fuzzy module	10	423.7	42.37	5009/68416	20.36 MHz
Total time	153,610	940,273.06	6.121170887		

第7章 結論與未來研究方向

7.1 結論

本論文以多主從架構設計之模糊目標物移動方向辨識系統，並於可程式規劃系統晶片之系統架構下，利用 FPGA 的硬體電路優勢，以軟硬體協同設計之方式實現硬體加速之功能。

本文應用 SOPC 軟硬體協同設計的技術來設計模糊目標物移動方向辨識系統，以 Nios II 軟核心處理器計算權重值，硬體電路進行歷史軌跡之產生，可以大幅降低 Nios II 處理器的負擔，有效降低大量影像資料的處理時間，加快影像的擷取與顯示速度，達到即時動態方向辨識的目的。

本文應用多主從系統架構來設計硬體加速器，並利用軟硬體協同設計的技術來設計實現歷史軌跡與模糊演算法實際應用於動態方向辨識上，藉由軟硬體緊密的合作，可以達到即時的方向辨識效果。

對於實務應用而言，可以先以本方法之軟硬體協同設計方式建立一雛形系統，以驗證其功能，一旦通過測試，即可將權重計算部分全部用硬體加以實現，進一步提升執行之效能。

7.2 未來研究方向

在未來研究方向，對於模糊目標物移動方向辨識之設計上，在移動物體運動方向過於複雜的情況下，需要使用到更精確的辨識法則。在未來的研究中，對於歷史軌跡的移動板塊需要更強健的處理，才能有足夠的資訊來辨識物體移動方向。未來可以加上尺度不變特徵轉換 (Scale-Invariant Feature Transform, SIFT)與加速穩健特徵 (Speeded Up Robust Features, SURF)演算法，來辨識多目標的移動方向。

參考文獻

[1] J. F. Shi, C. J. Hua, and G. H. Li, "A simplifying method of vision attention simulating human vision in machine vision system," *IEEE Transaction on Machine Learning and Cybernetics*, Qingdao, Jan. 2010, pp. 3097-3100.

[2] S. Ma, H. Wang, Z. Hao, L. Bai, and H. Yu, "Application research of machine vision technique in intelligent empty bottle inspection system," *IEEE International Conference on Intelligent Control and Automation*, Jinan, Jul. 2010, pp. 4462-4466.

[3] W. Zhang, Q. Zhao, and L. Liao, "Development of a real-time machine vision system for detecting defeats of cord fabrics," *IEEE International Conference on Computer Application and System Modeling*, Taiyuan, Oct. 2010, pp. 539-543.

[4] M. Klosowski, "Hardware accelerated implementation of wavelet transform for machine vision in road traffic monitoring system," *IEEE International Conference on Information Technology*, Gdansk, May 2008, pp. 1-4.

[5] Y. Cao, H. Chen, X. Yu, H. Zang, and X. Li, "Machine vision-based detection system of a cleaning robot for vertical type air-conditioning duct," *IEEE International Conference on Advanced Computer Control*, Shenyang, Jun. 2010, pp. 545-548.

[6] D. Y. Ge, X. F. Yao, and Q. Zhang, "Development of Machine Vision System Based on BP Neural Network Self-learning," *IEEE International Conference on Computer Science and Information Technology*, Singapore, Sep. 2008, pp. 632-636.

[7] 吳怡明 (98 年)，*手勢辨識應用於遙控音樂播放系統*，國立台灣科技大學電機工程系碩士論文。

[8] 洪兆欣 (95 年)，*以軌跡辨識為基礎之手勢辨識系統*，國立中央大學資訊工

程研究所碩士論文。

[9] 陳奕豪 (99 年)，*基於人臉追蹤之人機介面*，義守大學資訊工程研究所碩士論文。

[10] 劉東樺 (98 年)，*以適應性膚色偵測與動態歷史歸機影像為基礎之即時手勢辨識系統*，大同大學資訊工程研究所碩士論文。

[11] Altera Corporation, *SOPC Builder User Guide*, 2010.

[12] Altera Corporation, *Designing With Nios & SOPC Builder*, 2010.

[13] L. Liu, "A Prototyping IP Hardware for SOPC with Single Instruction Driving," *IEEE International Conference on Communications, Circuits and Systems Proceedings*, Guilin, Jun. 2006, pp. 559-562.

[14] F. Yang, C. Deng, and X. Liu, "The design of motion compensation IP core based on SOPC," *IEEE International Conference on Intelligent Control and Information Processing*, Dalian, Aug. 2010, pp. 453-457.

[15] J. Liu, W. Zhang, G. Xu, and D. Wang, "Design of Online Reconstructable SOPC System Based on TCP/IP," *IEEE International Conference on Measuring Technology and Mechatronics Automation*, Changsha City, Mar. 2010, pp. 1022-1025.

[16] S. A. Li, C. C. Hsu, C. C. Wong, and C.J. Yu, "Hardware/Software Co-design for Particle Swarm Optimization Algorithm," *Information Sciences*, vol. 181, pp. 4582–4596, Oct. 2011.

[17] L. Liu, "A reconfigurable SoPC based on HW-SW co-design," *IEEE International Conference on Industrial Technology*, Chengdu, Apr. 2008, pp. 1-4.

[18] L. A. Zadeh, "Fuzzy sets," *Information and Control* ,vol. 8, pp.338-353, 1965.

[19] L. A. Zadeh, "Fuzzy algorithms," *Information and Control*, vol. 12, no. 2, pp. 94-102, 1968.

[20] R. E. Bellman and L. A. Zadeh, "Decision-maling in a fuzzy environment,"

Management Science, vol. 17, no. 4, pp. 141-164, 1970.

[21] L. A. Zadeh, R. E. Kalman, and N. Declaris, *Toward a theory of fuzzy system*, California, 1971.

[22] L. A. Zadeh, "Similarity relations and fuzzy ordering," *Information Sciences*, vol. 3, no. 2, pp. 177-200, 1971.

[23] L. A. Zadeh, "Outline of a new approach to the analysis of commplex systems and decision processes," *IEEE Transaction on System, Man, and Cybern.* , vol. 3, pp. 28-44, Jan. 1973.

[24] B. Chen and L. L. Hoberock, "Mashine Vision Recognition of Fuzzy Objects Using a New Fuzzy Neural Network," *IEEE International Conference on Robotics and Automation*, Minneapolis, Apr. 1996, pp. 1596-1601.

[25] J. Killing, B. W. Surgenor, and C. K. Mechefske, "A Neuro-fuzzy Approach to Machine Vision Based Parts Inspection," *IEEE International Conference on Fuzzy Information Processing Society*, Montreal, Jun. 2006, pp. 696-701.

[26] S. Singh and J. S. Saini, "Fuzzy FPGA based captive power management," *IEEE International Conference on Power India Conference*, New Delhi, June 2006.

[27] A. K. Aws, M. Radu, and S. X. Yang, "A Neuro-fuzzy Approach to Machine Vision Based Parts Inspection," *IEEE International Conference on Automation and Logistics*, Hong Kong and Macau, Aug. 2010, pp. 539-544.

[28] V. Kumar, K. P. S Rana, V. Kumar, "Real Time Comparative Study of the Performance of FPGA based PID and Fuzzy Controllers for a Rectilinear Plant," *IEEE International Conference on Power Electronics*, New Delhi, Jun. 2011, pp. 1-7.

[29] 汪惠健，*模糊理論與應用*，培生教育出版集團, 2006 年.

[30] 友晶科技網址，http://www.terasic.com.tw/tw/

[31] Terasic Corporation, *TRDB_D5M_UserGuide*, August 2010.

[32] Terasic Corporation, *THDB-D5M Hardware Specification*, 2008.

[33] Terasic, *TRDB_LTM_UserGuide*, Document Version 1.22, 2007.

[34] Terasic Corporation, *LTM User Manual*.

[35] 林玟玲，*以軟硬體協同設計之混合型即時影像物體追蹤系統*，淡江大學電機工程學系碩士班碩士論文 (指導教授：易志孝)，2011。

[36] 李世安，*即時目標影像追蹤之 SoPC 設計*，淡江大學電機工程學系博士班博士論文 (指導教授：翁慶昌)，2008。

[37] Altera Corporation, *Avalon Interface Specifications*, August 2010.

[38] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Prentice Hall, 3rd Edition, 2008.

[39] 劉韋辰，*基於 FPGA 之單移動目標物歷史軌跡方向即時辨識系統*，國立臺灣師範大學應用電子科技學系 (指導教授：許陳鑑)，2011。

[40] 維基百科網址，http://zh.wikipedia.org/wiki/%E8%A6%96%E8%A6%BA%E6%9A%AB%E7%95%99