

國立臺灣師範大學跨域科技產業創新研究學院
AI 跨域應用研究所

碩士論文

Graduate Institute of AI Interdisciplinary Applied Technology

College of Interdisciplinary Industry-Academia Innovation

National Taiwan Normal University

Master's Thesis

可見光視線追蹤器之眼角定位硬體架構

Hardware Architecture for Canthus Localization in
Visible-Spectrum Gaze Trackers

黃秋貴

Chiu-Kuei Huang

指導教授：高文忠 博士

Advisor: Wen-Chung Kao, Ph.D.

中華民國 114 年 7 月

July 2025

致謝

研究所的時光很快地就來到了尾聲，很感謝高文忠教授給我進入實驗室的機會，讓我能在教授的指導以及同學的協助下走到現在，教授時常分享自己的親身經歷，我十分佩服教授在很多事物上把東西從無到有的做出來，而且是做到值得被讚賞的程度，雖然我沒有那麼好的頭腦與能力，但教授讓我們看到他在遇到問題與困境時是怎麼去面對與解決，了解我們未來也可能會面對的環境，並且在研究所的時光給予我們很多的空間去學習與體會。

謝謝在實驗室的夥伴們，特別是璽文與忠謀，幫我從無到有的補充了很多知識，給了很多的建議，在實驗室奮鬥的這些日子也讓我從他們身上看到面對問題時鍥而不捨的精神，跟學習如何解決問題的方法，也特別感謝仁翔、興驊，即使在畢業後，仍能撥空協助我完成研究，以及幾個一起在實驗室度過數個夜晚的成員：祐誠、柏翰、淳嘉、陳灝、少榆、翔元、鈺珊、宇彤、彥廷，雖然有幾位是政宏教授的學生，但因為在同一間實驗室，也一起討論跟研究過許多問題，也因為大家的認真無私與幽默，讓實驗室有著良好與正向的氣氛，大家一起研究跟討論，即使到很晚也不會感到太疲憊，還有謝謝學弟妹們：柏睿、浚祐、傳恩、慶皇，在我們升上碩二後將實驗室的事物承接下來，使我們能更專心去把研究完成。

謝謝我的家人給予的支持和信任，讓我在已經畢業一段時間後，還能義無反顧地回來念研究所。謝謝給過我幫助的每個人，遇到的每件事，都是在給我經驗，以及讓我成長，再次謝謝我的教授，雖然嘴上說後悔回來教書，但還是教了我們那麼多東西，研究所短短的兩年，卻有著非常充實與寶貴回憶。

可見光視線追蹤器之眼角定位硬體架構

學生：黃秋貴

指導教授：高文忠

國立臺灣師範大學 AI 跨域應用研究所



本研究針對僅採單一可見光相機的眼動儀，提出一套高效能、可即時運作的完整視線追蹤硬體系統。由於可見光環境下無法直接取得瞳孔或光斑，本系統改以內眼角作為穩定參考點，整合眼角偵測、眼球模型參數求解、幾何關係推算與動態縮放補償機制，建構完整的視線追蹤硬體架構。在既有高精度眼角偵測演算法的基礎上，我們將完整眼動儀系統硬體化，透過平行運算與管線化設計，在有限現場可程式化邏輯閘陣列資源內實現快速的眼角定位與視線估測，有效解決使用者頭部前後移動時的追蹤誤差問題。實驗結果顯示，眼角偵測模組在現場可程式化邏輯閘陣列平台上的處理速度達每秒 5,115 幀，而完整的眼動儀系統整合後仍能達到每秒 476 幀的處理效能，均具備即時運作能力。系統硬體資源使用率為百分之十三邏輯元件與百分之三十五記憶體，證明本架構能為可見光眼動儀提供可靠且高效的完整硬體解決方案。

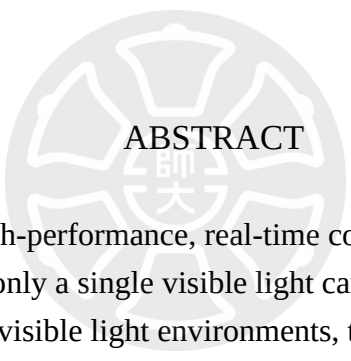
關鍵字：眼動儀、眼角偵測、晶片架構、動態縮放補償、系統整合

Hardware Architecture for Canthus Localization in Visible-Spectrum Gaze Trackers

Student : Chiu-Kuei Huang

Advisors : Dr. Wen-Chung Kao

Graduate Institute of AI Interdisciplinary Applied Technology
National Taiwan Normal University



ABSTRACT

This study proposes a high-performance, real-time complete gaze tracking hardware system for eye trackers using only a single visible light camera. Since pupils or light spots cannot be directly obtained in visible light environments, this system uses the inner eye corner as a stable reference point, integrating eye corner detection, eyeball model parameter solving, geometric relationship calculation, and dynamic scaling compensation mechanisms to construct a complete gaze tracking hardware architecture. Based on existing high-precision eye corner detection algorithms, we implemented the complete eye tracker system in hardware through parallel computing and pipeline design, achieving rapid eye corner localization and gaze estimation within limited field-programmable gate array resources, effectively solving tracking errors caused by users' forward and backward head movements. Experimental results show that the eye corner detection module achieves a processing speed of 5,115 frames per second on the field-programmable gate array platform, while the complete integrated eye tracker system still maintains a processing performance of 476 frames per second, both possessing real-time operation capabilities. The system hardware resource utilization is 13% logic elements and 35% memory, proving that this architecture can provide a reliable and efficient complete hardware solution for visible light eye trackers.

Keywords: eye tracker, canthus detection, hardware engine, dynamic scaling compensation, system integration

目錄

	Page
致謝	i
目錄	iv
表目錄	vii
圖目錄	viii
第一章 緒論	1
1.1 研究背景	1
1.2 研究問題及目的	2
1.3 論文架構	2
第二章 文獻探討	4
2.1 眼角偵測演算法	4
2.1.1 直方圖統計	4
2.1.2 橫向梯度運算	5
2.1.3 自適應閾值計算與二值化	6
2.1.4 最大連通面積計算	7
2.1.5 重心計算	8
2.2 前人模組回顧與本研究整合背景	9
2.2.1 眼球模型硬體模組	9
2.2.2 虹膜匹配硬體模組	10

2.3	眼角與眼球幾何關係計算模組	11
2.4	縮放比估算與參數調整模組	13
2.5	硬體架構設計	14
2.5.1	FPGA 影像處理硬體實作	14
2.5.2	FPGA 捲積硬體實作	15
第三章	研究方法	18
3.1	眼角偵測模組	18
3.1.1	狀態控制	20
3.1.2	直方圖等化模組	21
3.1.3	水平梯度計算模組	23
3.1.4	自適應閾值計算模組	24
3.1.5	二值化模組	25
3.1.6	形態學模組	25
3.1.6.1	全域雜訊侵蝕模組	25
3.1.6.2	ROI 侵蝕與膨脹捲積模組	27
3.1.7	重心計算模組	27
3.2	眼動儀系統整合模組	28
3.2.1	系統控制狀態機	29
3.2.2	眼球模型參數求解模組	30
3.2.3	CORDIC 幾何關係計算模組	31
3.2.4	考慮頭動虹膜匹配模組	32

第四章	實驗結果	34
4.1	實驗環境與設備	34
4.2	眼角偵測模組結果與分析	35
4.2.1	硬體效能分析	35
4.2.2	結果比較	37
4.3	眼動儀系統整合模組結果與分析	38
4.3.1	硬體效能分析	38
4.3.2	結果比較	40
第五章	結果與未來展望	42
5.1	結論	42
5.2	未來展望	43
References		44
自傳		47

表目錄

4.1 FPGA 規格及效能分析	36
4.2 FPGA 規格及效能分析	39
4.3 電腦規格及效能分析	39



圖目錄

2.1 水平梯度計算示意圖，紅點表示當前計算點，左右各取 i 像素進行梯度差分。	6
2.2 眼球的球面模型及其旋轉模型	10
2.3 虹膜邊緣圓匹配的硬體架構（圖中模組以英文標示） . .	11
2.4 以內眼角作為錨點之眼球中心估算幾何架構示意圖 . . .	12
2.5 使用者頭部前後移動下，影像中眼角距離變化之示意圖	13
2.6 改進的 8 方向 Sobel 邊緣偵測架構示意圖	15
2.7 水平（ G_x ）與垂直（ G_y ）梯度計算器硬體架構圖	15
2.8 FPGA 捲積運算加速 IP 核心功能架構圖	17
2.9 多核心並行計算架構圖	17
3.1 眼角偵測演算步驟	19
3.2 眼角偵測系統定位的眼角位置	19
3.3 眼角輸入影像	20
3.4 狀態圖	21
3.5 直方圖等化模組	21
3.6 直方圖統計模組	21
3.7 累計分佈模組	22
3.8 水平梯度計算模組	23
3.9 自適應閾值計算模組	24
3.10 侵蝕捲積模組	26
3.11 侵蝕與膨脹捲積模組	27
3.12 重心計算模組	28
3.13 眼動儀系統整合模塊	29
3.14 狀態圖	30

3.15 眼球模型參數求解模組	31
3.16 狀態圖	33
3.17 考慮頭動虹膜匹配模組	33
4.1 Altera DE2-115 教育開發平台	34
4.2 summary	36
4.3 speedup	36
4.4 電腦計算結果 (圖中白點為眼角位置)	37
4.5 晶片計算結果 (圖中紅點為眼角位置)	37
4.6 不同眼球模型的電腦計算結果 (圖中白點為眼角位置) . .	37
4.7 不同眼球模型的晶片計算結果 (圖中紅點為眼角位置) . .	38
4.8 summary	39
4.9 電腦計算結果	40
4.10 晶片計算結果	40
4.11 電腦計算結果 (離相機較近距離)	41
4.12 晶片計算結果 (離相機較近距離)	41

第一章 緒論

近年來，視線追蹤技術於智慧型互動裝置、人因工程、醫療輔具與虛擬實境等領域中展現廣泛應用潛力，尤以可見光環境下之非侵入式眼動追蹤系統更為受到關注。此類系統不需額外配戴設備，具備高使用者接受度與便利性，惟在自然光源、自由頭動與嵌入式實作等條件下，其穩定性與準確性仍面臨諸多挑戰。因此，如何於資源受限的平台上，同時兼顧即時性、抗雜訊能力與跨模組資料一致性，成為系統設計之關鍵課題。本研究針對此議題，提出一套整合式可見光視線追蹤架構，涵蓋眼角偵測、眼球模型重建、幾何關係推估、縮放比補償與虹膜匹配等模組，並實現在可重構硬體平台上，期能建立具備高實用性與高度可移植性之眼動儀系統。

1.1 研究背景

視線追蹤技術近年來於智慧型人機互動、無接觸介面設計、以及輔助科技等應用中備受重視，特別是在可見光條件下進行之非侵入式視線估測架構，因不需配戴額外裝置、具備操作便利性與高接受度，逐漸成為主流趨勢。傳統方法多仰賴紅外線主動照明與瞳孔中心定位作為視線追蹤依據，但於自然光源下使用時容易受到環境亮度波動與頭部姿態變化的影響，導致偵測不穩定或追蹤中斷。為突破此限制，許多學者開始探討以眼角與虹膜為基礎的可見光視線估測法，並發展對應的幾何模型與補償機制。在可見光眼動儀系統中，眼角位置於各種頭部姿態下仍具有良好穩定性，成為理想的追蹤錨點（anchor point）。前人研究已證實，透過深度學習模型進行影像中眼角特徵點定位，能有效取代傳統基於光學對比的手法，並提供更高準確度與抗雜訊能力。此外，前人於眼球模型與虹膜匹配等模組所提出之演算法雖具備即時性潛力，惟在硬體實作與跨模組整合方面仍屬初步階段。隨著嵌入式系統與 FPGA 技術的發展，將視線追蹤演算法進行硬體加速已成為提升系統效能的重要方向。透過專用硬體設計，不僅能降低系統延遲、提高

處理速度，更能減少對中央處理器的依賴，使眼動儀系統能在資源受限的環境中穩定運作。

1.2 研究問題及目的

基於上述背景，本研究發現現有可見光眼動儀系統面臨數個關鍵問題。首先，由於 AI 模型提供的特徵點座標無法達到像素級的穩定度，需要進一步的精確定位處理。現有的眼角定位演算法雖然具有高精準度與穩定性，但其影像處理流程中各步驟間存在緊密的前後依賴關係，需要多次遍歷影像資料。當影像中含有大量雜訊或處理解析度較高時，計算量會顯著增加，影響系統的即時追蹤能力。其次，在實際應用中，使用者與攝影機之間的相對距離變化將造成影像尺度改變，進而導致眼球模型參數不一致。此外，現有各模組間缺乏有效的資料串接與格式統一機制，限制了整體系統的整合效能。因此，本研究的主要目的為設計並實現高效能的眼角定位硬體架構，透過硬體加速技術大幅提升計算效能，同時保持高精度特性。本研究將建立完整的系統整合架構，包含眼角偵測、眼球幾何推估、縮放補償與虹膜匹配等子系統的硬體映射，確保模組間資料串接的一致性。為解決頭部移動造成的參數變化問題，本研究引入縮放比估算模組作為調整因子，使系統能因應頭部 Z 軸方向的自由移動，動態修正相關幾何參數。最終目標是在 FPGA 平台上實現具備低延遲、高可重組性、與良好資源使用效率的視線估測系統，使可見光眼動儀能在嵌入式平台中穩定且即時運作。透過達成上述目標，本研究期望為可見光眼動儀系統提供一個完整、高效且實用的硬體解決方案。

1.3 論文架構

本論文共分為五章，內容安排如下：第一章為緒論，說明本研究之背景、動機、目的與貢獻。第二章為文獻回顧與整體系統架構介紹，包含眼角偵測技術之發展、眼球模型與虹膜匹配模組之理論基礎與硬體實作情形，並說明本研究所整

合之幾何關係推估與縮放比補償機制。第三章介紹本研究所實作之硬體模組設計與整合流程，詳述各子模組之功能架構、資料流設計與硬體映射策略。第四章為系統實驗與性能評估，包含功能驗證、效能測試與資源使用分析。最後，第五章則為結論與未來展望，總結本研究成果並提出未來可能延伸之研究方向。



第二章 文獻探討

2.1 眼角偵測演算法

眼角偵測演算法是可見光眼動儀系統中至關重要的一環。本論文中採用了 Kao 等人提出的方法以及在 [1]，本節將詳細探討該演算法的各個組成部分，從影像預處理的直方圖統計，到最終精準眼角位置的確定。此演算法不僅能準確地識別眼角位置，還具有抗干擾性強、適應性好的特點，能在不同光照及面部表情變化的情況下保持穩定的識別效果。

2.1.1 直方圖統計

在眼角影像預處理流程中，直方圖等化（Histogram Equalization）被用以提升影像對比度，使眼角特徵於低對比度條件下仍能有效被辨識。由於眼部區域常受限於可見光照明變化，原始影像的像素分布往往集中於低灰階區域，造成梯度計算與閾值判定的困難。透過直方圖等化，可將影像的灰階分布重新調整為較為均勻，進而強化邊界與特徵點對比。

假設輸入影像 I 為大小 $M \times N$ 的灰階影像，其灰階等級為 L ，每一個灰階級 r_k 的像素數可統計得出頻率函數如式 2.1。

$$P_r(r_k) = \frac{n_k}{M * N}, 0 \leq k \leq L \quad (2.1)$$

其中， n_k 為灰階值 r_k 所出現的像素總數。經由上述機率分布，可計算累積分佈函數（Cumulative Distribution Function, CDF）作為像素映射依據，其公式如式 2.2。

$$s_k = T(r_K) = \sum_{j=0}^k p_r(r_j) \quad (2.2)$$

$T(r_k)$ 表示原始灰階值 r_k 對應的映射結果， $s_k \in [0, 1]$ 。為使轉換後影像可涵蓋整個灰階範圍，需將 s_k 線性縮放至 $[0, L - 1]$ ，對應之映射函數如式 2.3。

$$r'_k = (L - 1) * s_k \quad (2.3)$$

透過此方式，低對比的影像像素得以重新分佈，增強區域內部的亮度變化與邊緣細節，進而有助於後續之梯度分析與二值化處理。

2.1.2 橫向梯度運算

在完成影像的對比度增強後，為了精確定位內眼角特徵，對眼角區域進行水平方向的梯度分析。此作法基於觀察得知，內眼角邊界處在水平方向上具有明顯的像素變化，而垂直方向的變化相對較弱，因此藉由水平梯度之變化可作為偵測邊界的重要依據。

在計算每一像素之梯度時，需考量到內眼角邊界具有一定連續性，若僅使用鄰近像素進行計算，可能會因為邊界平滑或微弱變化而導致梯度值不足。因此本研究採用左右距離 i 個像素的方式計算梯度大小，其方法如 Figure 2.1 所示，紅點代表當前欲計算梯度之位置，其左右兩側 i 像素作為比較對象，如式 2.4。

$$G_i(x, y) = k_r * (I(x - i, y) - I(x + i, y)) \quad (2.4)$$

其中， $G_i(x, y)$ 表示位置 (x, y) 處的水平梯度， $I(x, y)$ 為該位置的灰階值， i 為計算梯度時向左及向右的距離， k_r 為根據影像亮度所調整之增益參數。此參數可根據影像中整體亮度進行自適應調整，以防止在亮度過低或過高時產生梯度強度不平衡的情形。

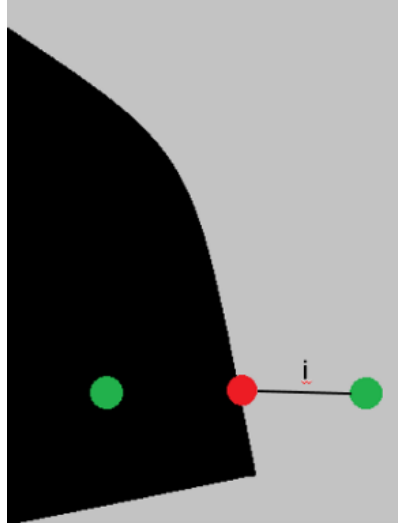


Figure 2.1: 水平梯度計算示意圖，紅點表示當前計算點，左右各取 i 像素進行梯度差分。

2.1.3 自適應閾值計算與二值化

在取得影像的梯度圖後，為能有效萃取眼角邊界資訊，本研究接著進行二值化處理，以進一步將高變化區域從背景中分離。由於眼角影像於不同環境光源條件下會產生強度與分布的差異，固定閾值往往難以適用於所有場景，因此必須設計一個具備自適應能力之閾值計算方法，以提高演算法之穩定性與泛用性。

本研究首先對梯度圖進行直方圖分析，累積統計各灰階強度出現的頻率，記為 $H(i)$ ，其中 $i \in [0, 255]$ 為灰階值。為決定合適的閾值範圍，從高灰階處由大至小累積其像素數量，並尋找一個值 n ，使得從 n 到 255 的總像素數量大於一設定閾值參數 B ，其判定式如式 2.5。

$$B \leq \sum_{i=n}^{255} H(i) \quad (2.5)$$

獲得 n 值後，便可將此範圍內的像素作為候選區間，以其加權平均作為本張影像之二值化閾值 T_h ，如式 2.6。

$$Th = \frac{\sum_{i=n}^{255} i * H(i)}{\sum_{i=n}^{255} H(i)} \quad (2.6)$$

在求得此自適應閾值後，將梯度影像與 Th 進行比較，並將大於 Th 者標記為白（值為 1），小於者標記為黑（值為 0），形成二值化圖像，其處理可表示如式 2.7。

$$B(x, y) = \begin{cases} 1; G(x, y) > Th \\ 0; otherwise \end{cases} \quad (2.7)$$

其中 $B(x, y)$ 為二值化後之影像， $G(x, y)$ 為經由式 2.4 所得之梯度值， Th 為根據式 3.5 所計算之自適應閾值。由於眼角邊界通常伴隨較高的梯度變化，因此在二值化後，邊界區域的梯度訊號將顯著凸顯，為後續眼角位置之提取提供基礎。

然而在某些光源變化劇烈或影像品質不佳的情況下，仍可能出現多處雜訊或非連續邊界導致錯誤偵測，對此本研究於下一步將引入最大連通區域過濾技術以排除非眼角區域，並進一步萃取穩定且唯一的候選點作為眼角位置。

2.1.4 最大連通面積計算

進行影像中眼角區域之正確標記與識別時，連通元件標記（Connected Component Labeling, CCL）為其中一項關鍵技術，其中以傳統的兩次掃描法（Two-Pass Algorithm）最為常見。該方法自 1970 年代由 Rosenfeld 等人提出以來 [2]，廣泛應用於二值化圖像中區域分群與特徵分割等處理流程。其核心概念在於透過兩階段的影像掃描，逐步完成對每個像素所屬區域的標記與整合，最終獲得所有獨立連通區域的完整資訊。在第一階段掃描中，系統由左至右、由上至下對整張影像進行遍歷，對於每一個前景像素（即值為 1 的像素），判斷其左上、

上、右上及左方等鄰近像素的標記狀態；若這些鄰點均未被標記（即為背景），則給予該像素一個新的標記編號；若存在已標記的鄰點，則選取其標記作為目前像素的標記，並於必要時記錄多個標記之間的等價關係，以利後續處理。在第一階段結束後，系統將所有等價標記整併為統一群集，並於第二階段再次遍歷整張影像，根據合併後的標記對所有像素進行最終標示。藉由此種方式，影像中所有具連通性的區塊將被賦予獨立的標籤編號，後續即可進行面積統計、重心計算等特徵萃取操作。

2.1.5 重心計算

在完成最大連通區域的標記後，為獲得精準且具代表性的眼角位置，本研究採用區域重心（Centroid）作為該區塊的空間代表點。此作法的基本假設為，最大連通區域即對應於眼角所在的主要邊界區塊，因此將該區塊中所有有效像素的座標進行加總並取其平均，可得穩定且具抗雜訊能力之眼角位置。此方法不僅計算量低，亦便於以 FPGA 實作，能有效結合平行加總與低延遲除法模組以提升運算效能。具體而言，對於二值化後之影像資料，系統將遍歷整張影像，並針對屬於最大連通區域之像素（即值為 1 的像素） $I(i,j)$ 累加其橫軸與縱軸座標總和，再分別除以總像素數，即可得出該區域的重心座標，如式 2.8 所示。由上式可得眼角之最終重心座標為 $L = (x_c, y_c)$ ，代表本系統最終所推定之眼角座標，並將作為後續幾何模型推算與凝視向量估算之輸入依據。為提升數值穩定性與雜訊抵抗能力，本系統亦於硬體設計中結合位元精度累加器與除法器，以確保在資源受限條件下仍能維持高精度輸出。

$$x_c = \frac{\sum_{i=0}^{\text{ROW}} \sum_{j=0}^{\text{COL}} i \cdot I(i,j)}{\sum_{i=0}^{\text{ROW}} \sum_{j=0}^{\text{COL}} I(i,j)}, \quad y_c = \frac{\sum_{i=0}^{\text{ROW}} \sum_{j=0}^{\text{COL}} j \cdot I(i,j)}{\sum_{i=0}^{\text{ROW}} \sum_{j=0}^{\text{COL}} I(i,j)} \quad (2.8)$$

2.2 前人模組回顧與本研究整合背景

本研究旨在整合多個視線估測模組，建立一套可支援自由頭動之高精度視線追蹤系統。為此，有必要回顧現有系統中關鍵模組之設計理念與硬體實現方式，理解其運作邏輯與參數輸入輸出特性，俾利後續模組介接與資料一致性之維護。本章節將聚焦於兩個與本研究高度相關之核心模組，分別為眼球模型模組與虹膜匹配模組。前者負責依據眼角與虹膜位置建構初始 3D 眼球模型，後者則據此模型進行視線落點推估與比對，兩者皆為視線估測流程中不可或缺之關鍵組件。以下將分別說明此二模組之演算法架構與對應硬體實現方式，並說明其與本研究所設計之縮放模組之整合關係。

2.2.1 眼球模型硬體模組

為達成非侵入式、可於可見光環境下運作之視線追蹤系統，過去 Kao 等人於其研究中提出一套眼球模型建構與比對方法 [3][4][5]，該方法透過建構三維幾何模型以模擬眼球旋轉與虹膜投影，並結合影像特徵點（如內眼角與虹膜中心）進行視線方向估測，如圖 2.2。該圖顯示了眼球模型中的旋轉參數（ θ 、 ϕ ）、眼球半徑 r_E 、虹膜半徑 r_I 、虹膜中心點 I_c 與眼球中心點 E_c 以及與眼角位置 F 之幾何關係，可作為視線估測的主要依據。該模型具備即時性與硬體映射潛力，適用於嵌入式視線追蹤裝置之實作需求。該方法主要輸入為經由前處理模組取得之眼角與虹膜特徵點座標，輸出為眼球半徑、虹膜半徑、虹膜中心位置等參數。為確保估測準確性，模型設計中採用參數搜尋與旋轉矩陣轉換等運算方式，藉以重建虹膜邊緣於不同視角下的形變行為。雖該研究已完整建立演算法架構，惟對應之硬體設計並未於該篇論文中詳述。對此，本研究所屬團隊於後續工作中完成該模型之硬體映射，並設計包含煙花演算法參數產生器、樣本映射模組、評分單元與控制模組等子系統之實作架構。此硬體版本可於 FPGA 上進行即時運算，並具備良好之延遲與資源使用效率，為後續視線估測模組（如虹膜匹配）提供穩定可靠之模型參數輸出。

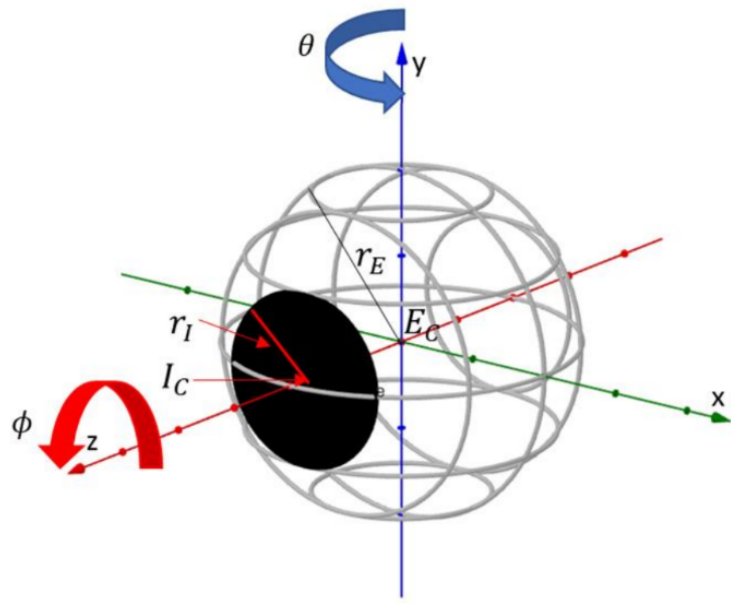


Figure 2.2: 眼球的球面模型及其旋轉模型

2.2.2 虹膜匹配硬體模組

在可見光眼動儀系統中，虹膜邊界位置的準確匹配對於視線方向的判定具有關鍵影響。Kao 與 Lin 等人首先提出了一套演算法來搜尋並定位準確的虹膜位置 [6][7][8]，然而由於此演算法存在計算時間瓶頸，Kao 與 Chiu 等人進一步提出改良演算法 [9]，採用階層式搜尋策略，大幅改善了演算時間。儘管如此，面對高速即時運算的需求仍然不夠快速，因此針對此問題，Kao 與 Liu 等人提出一套以 limbus circle 為核心的虹膜邊界比對機制，並設計對應的專用硬體加速器以支援即時運算 [10]。與傳統以瞳孔中心為基準之紅外線追蹤方式不同，此方法利用眼球模型所生成之虹膜外圍預測軌跡進行比對，透過調整眼球模型的旋轉角度 (yaw, roll)，搜尋最佳擬合之邊界輪廓。該方法首先根據眼球模型輸出參數，生成一系列可能之虹膜邊界（通常為橢圓），每一邊界皆對應一組特定旋轉角度。硬體設計中透過角度產生器 (Angle Counter) 輸出候選角度組合，並以旋轉矩陣運算推估虹膜中心 (Iris Center Calculation) 與 limbus circle 邊界點 (Limbus Circle Calculation) 上之像素位置。針對每一邊界點，系統沿特定放射角度掃

描該點內外的像素亮度，並依據內外亮度差異計算匹配分數（Matching Score Calculation）。其核心假設為：虹膜區域內部通常較暗，外部較亮，因此內外強度落差可作為匹配準確度的依據。在硬體架構上，如圖 2.3 所示，此匹配流程由多組子模組組成，包括角度計數器（Angle Counter）、虹膜中心與邊界點計算器（Iris Center Calculation 與 Limbus Circle Calculation）、匹配分數計算器（Matching Score Calculation）、記憶體模組（RAM）與排序模組（Sort）。各模組以 pipeline 架構串聯，使得系統能於每一個時脈週期內啟動新一組角度之匹配流程，實現高效能之輸出頻率。最終，排序模組將根據所有角度組合的匹配分數選出最佳結果，並用作下一輪比對的初始條件，以加速收斂。根據實驗結果，該硬體模組實作於 FPGA 上可達到每秒 525 張影像之處理速度，且使用資源有限，顯示其於嵌入式視線追蹤應用中的高度可行性。儘管該研究已建立完整之硬體實現，其系統設計仍假設輸入參數為準確的眼球模型資訊，並未涵蓋上游模組（如眼角偵測）與中介補償（如縮放模組）之整合過程。因此，本研究進一步提出縮放係數模組，作為模型輸出與匹配模組之間的橋接器，以實現跨模組參數一致性。

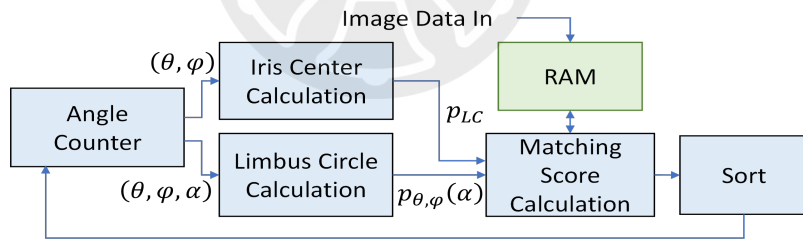


Figure 2.3: 虹膜邊緣圓匹配的硬體架構（圖中模組以英文標示）

2.3 眼角與眼球幾何關係計算模組

為建構可支持頭部自由移動之視線追蹤系統，Kao 等人提出一套以眼角作為錨點的眼球模型建構法 [11]。該方法建立在先前眼球模型模組之參數已完成最佳化的基礎上，亦即如 2.2.1 所述，系統透過煙花演算法於初始化階段求得之眼球半徑、虹膜半徑與眼球中心座標等參數，作為後續幾何關係式建構之依據。此方法觀察到，在真實空間中，眼球中心與內眼角之間的相對位置應為穩定不變的常數項，且與視線方向無關。基於此一幾何假設，系統可進一步透過上述已取得之

模型參數，於初始階段進行眼球中心與眼角之三維相對位置向量計算，並據以建立一套描述其空間關係的幾何函數。該幾何函數能於後續影像處理階段中直接應用，亦即當追蹤系統取得當下畫面中內眼角座標 F 時，眼球中心 C 可藉由加總預先計算之位移向量 ΔX 而得，其計算關係如式 2.9 所示。

$$C = F + \Delta X \quad (2.9)$$

由於眼角位置在影像中較容易取得且不易受眼球轉動影響，因此本方法可避免每一影格重新進行完整眼球模型重建，大幅降低系統之運算負擔，並提升即時性。此幾何關係式為針對單一使用者所建立，能反映個體生理結構差異，且具備通用性與一致性特徵。其幾何架構如圖 2.4 所示，該方法將眼角座標作為幾何映射的輸入點，透過空間轉換推得對應之眼球中心點，最終整合至本研究之追蹤系統流程中，作為中介推估核心。

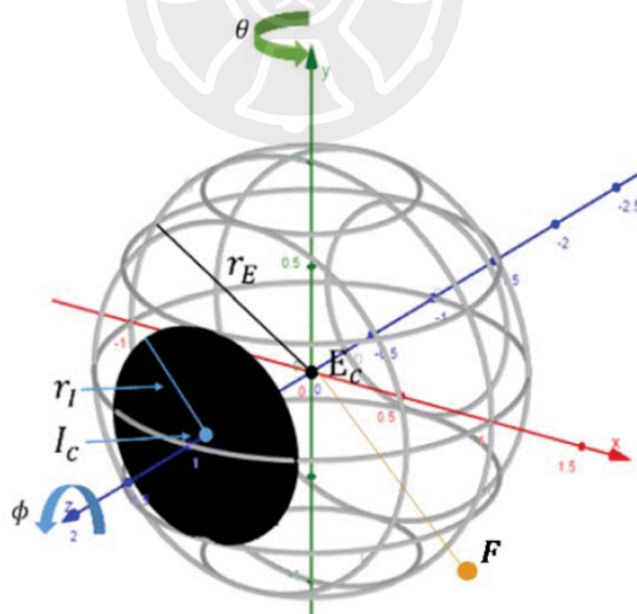


Figure 2.4: 以內眼角作為錨點之眼球中心估算幾何架構示意圖

2.4 縮放比估算與參數調整模組

為因應使用者頭部於 Z 軸方向上的自由移動所帶來之視覺尺度變化，Kao 等人於其研究中提出一套基於眼角距離變化之縮放比（scaling ratio）估算機制 [12][13][14]，用以補償眼球模型輸出參數於不同觀測距離下所產生的誤差。其核心概念假設為：於固定鏡頭參數與環境光源條件下，眼部區域在影像中的投影大小與使用者與鏡頭之間的實際距離呈反比關係。亦即，當使用者靠近攝影鏡頭時，影像中所觀測之眼角間距（inter-canthus distance）將相對增加；反之則縮小，如圖 2.5 所示。

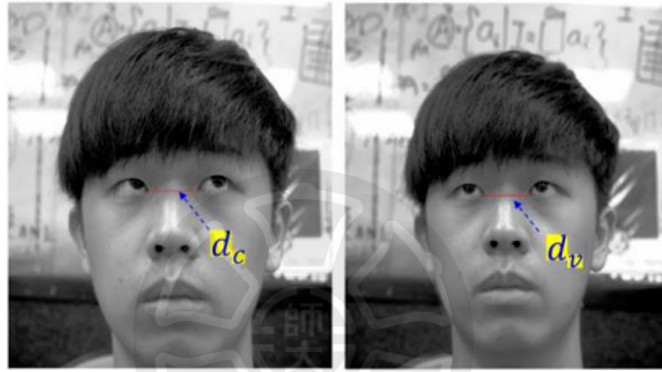


Figure 2.5: 使用者頭部前後移動下，影像中眼角距離變化之示意圖

為量化此變化，本系統於初始化階段建立基準眼角距離 d_f ，並於後續每一影格中即時計算當下影像中的眼角距離 d_v 。透過兩者比值，即可推得縮放比 S ，如式 2.10 所示。該縮放比不直接作用於影像尺寸，而係作為眼球模型與眼球幾何關係函數之參數調整因子。具體而言，眼球模型輸出之眼球半徑、虹膜半徑與眼球中心座標等參數，將分別乘以該縮放比進行修正，以反映真實空間中使用者視角與位置的變化。

$$S = \frac{d_v}{d_f} \quad (2.10)$$

此縮放補償模組位於整體系統架構中的中介位置，承接前端之眼角偵測與眼球幾何關係推估模組輸出，並將修正後之參數輸入至虹膜匹配模組，使後續比對

過程所參照之模型輪廓符合當下使用者實際姿態。此設計不僅提高整體視線追蹤系統於使用者頭部自由移動下之穩定性，亦有助於避免因空間尺度誤差所導致之匹配錯誤與視線估測偏移，顯著提升整體系統於嵌入式應用環境中之實用性與魯棒性。

2.5 硬體架構設計

為了設計出可見光眼動儀眼角偵測演算法的硬體架構並正確實現其功能，我們參考目前已存在的眼動儀硬體架構，作為實現我們眼動儀演算法的參考基礎。我們將晶片設計成能夠快速完成眼角偵測演算法中的多個模組計算，透過平行化與管線化的設計以提高計算效率。

2.5.1 FPGA 影像處理硬體實作

另一篇由 Ravichandran 等人 (2025) 提出的研究 [15]，則聚焦於 Sobel 邊緣偵測運算在 FPGA 上的高效率並行實作。該研究提出改進的 Sobel 演算法，將傳統的 2 方向（水平與垂直）擴展為 8 方向邊緣偵測（ $0^\circ, 45^\circ, 90^\circ, 135^\circ, 180^\circ, 225^\circ, 270^\circ, 315^\circ$ ），如 Figure 2.6 所示。透過多方向的邊緣偵測，能夠更精確地捕捉斜向和複雜邊界特徵，有效提升邊緣定位的準確性。

在硬體架構設計上，如 Figure 2.7 所示，該研究採用專用的水平（Gx）和垂直（Gy）梯度計算單元，實現真正的並行處理架構。系統透過多個並行的乘法累加器（MAC）和比較器，同時完成多個像素的梯度計算，並結合流水線設計減少處理延遲。實驗結果顯示，該系統在處理 1028×720 解析度影像時達到 0.0033 秒的處理速度，相較於其他方法有顯著提升，並在硬體資源使用上比先前設計節省 13.9% 的暫存器和 20.7% 的邏輯區塊。

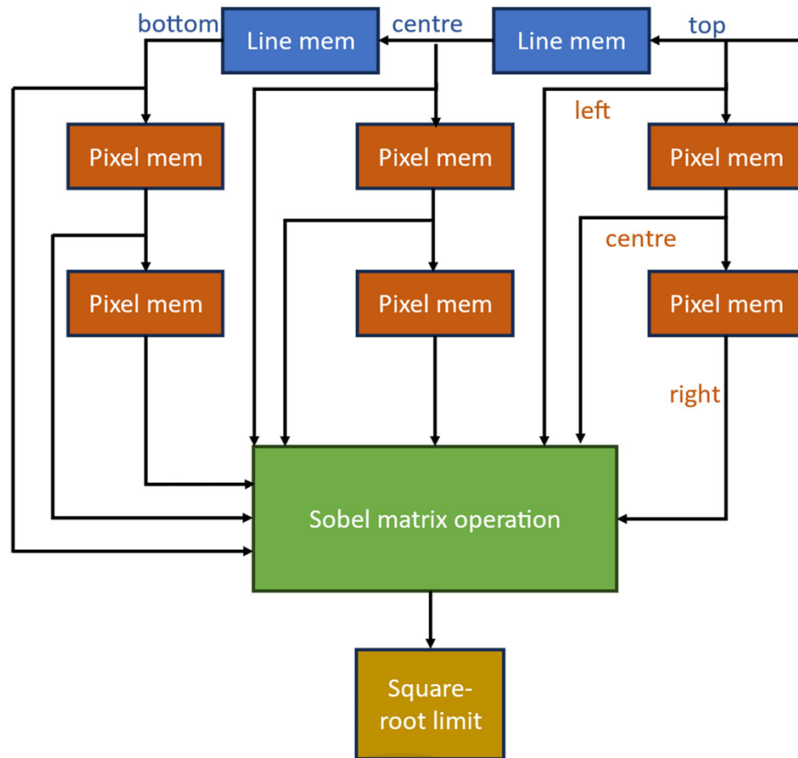


Figure 2.6: 改進的 8 方向 Sobel 邊緣偵測架構示意圖

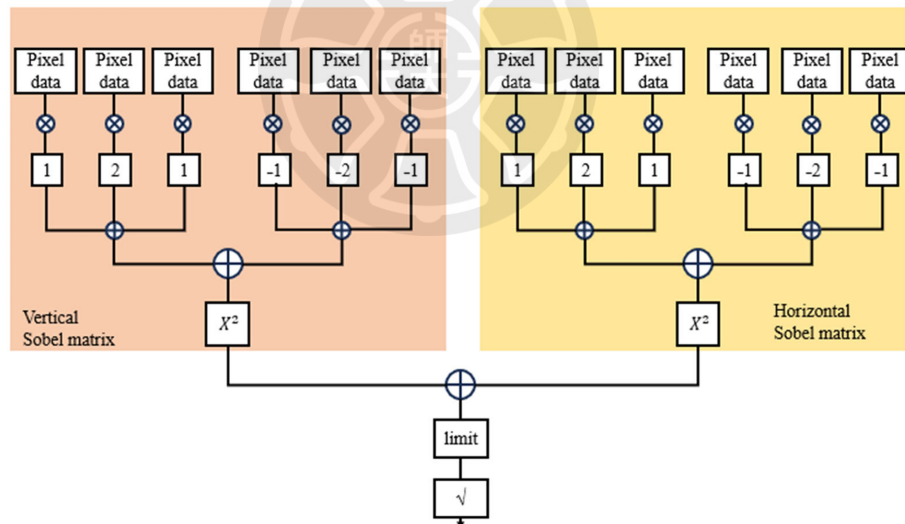


Figure 2.7: 水平 (Gx) 與垂直 (Gy) 梯度計算器硬體架構圖

2.5.2 FPGA 捲積硬體實作

在眾多影像處理演算法中，二維捲積 (2D Convolution) 為圖像辨識與濾波操作的核心基礎，其廣泛應用於邊緣偵測、平滑濾波與深度神經網路 (CNN) 等。然而，捲積本質上為多重乘加 (multiply-and-accumulate, MAC) 操作的組合，隨

著遮罩 (kernel) 尺寸與輸入解析度提升，其硬體實作所需資源與時脈頻率需求亦顯著上升。

為解決此問題，Pham 等人 (2022) 提出一種針對邊緣型 FPGA 設計的捲積加速 IP 核，採用 Verilog 在嵌入式平台上實作，展示出每秒可達 0.224 GOPS (單核) 與 4.48 GOPS (全板) 性能之表現 [16]。其設計核心為 parallel MAC pipeline 結構，如 Figure 2.8 所示，將捲積運算模組化並實現流水線處理，以降低延遲並提升處理吞吐量。

該系統採用多核心並行架構，如 Figure 2.9 所示，通過四個獨立的計算核心同時處理不同通道的特徵圖，每個核心配備專用的 BRAM 和權重載入器。此設計亦考量記憶體佈局與資料流動效率，透過 line-buffer 搭配 window generator 方式解決卷積滑動視窗問題，有效減少資料搬移延遲。實驗結果顯示，該 IP 核心在 Pynq Z2 FPGA 板上僅使用不到 5% 的硬體資源，卻能達到 112 MHz 的工作頻率，證明其在邊緣運算環境下的高效能低資源設計目標。

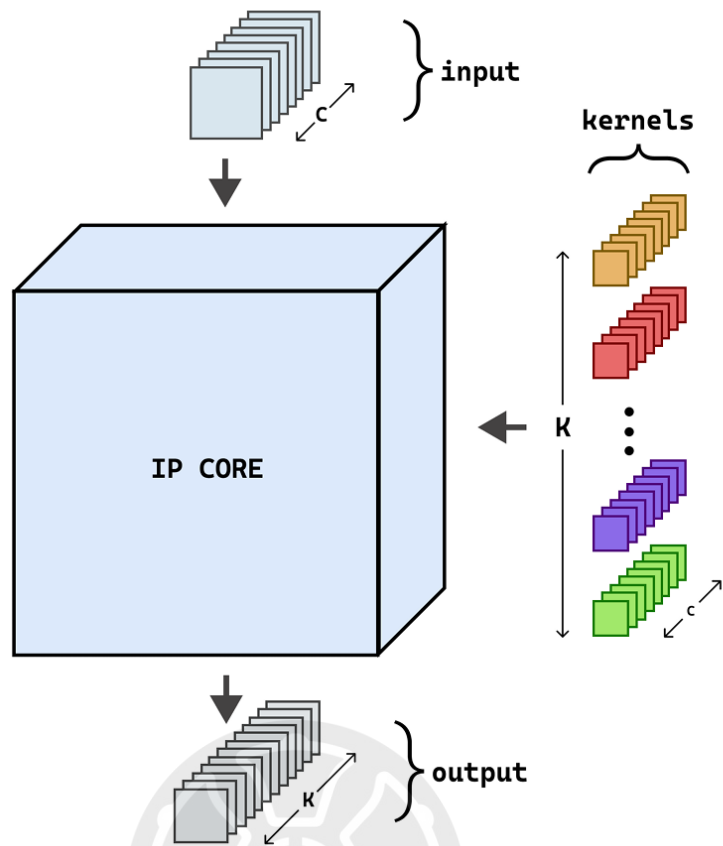


Figure 2.8: FPGA 捲積運算加速 IP 核心功能架構圖

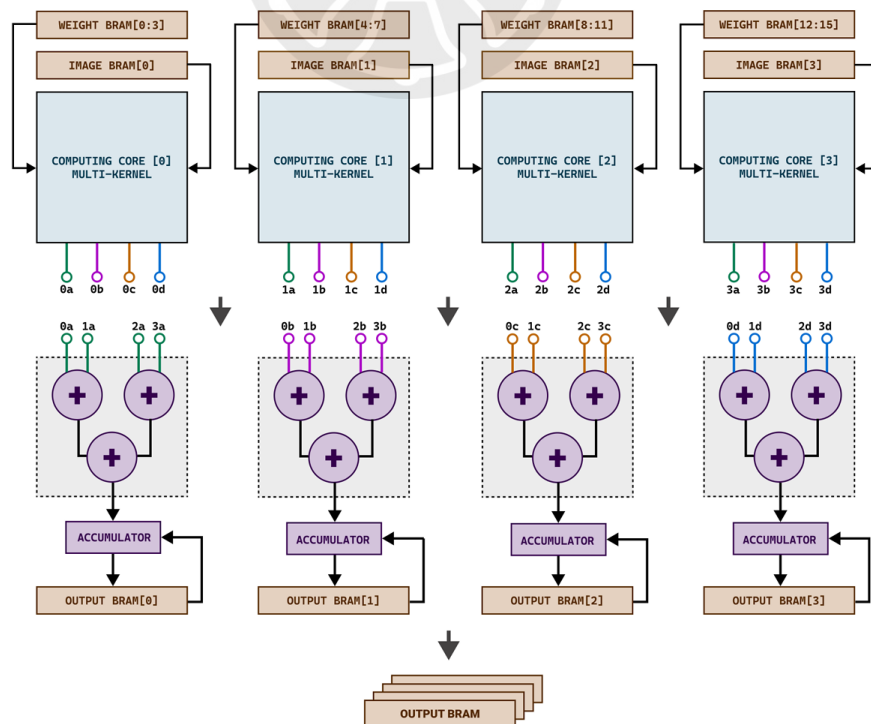


Figure 2.9: 多核心並行計算架構圖

第三章 研究方法

本章將介紹本研究所提出之可見光眼動儀系統中的關鍵處理流程，包含眼角偵測、靜態眼球模型參數導入、眼角與眼球幾何關係推估與眼球模型參數動態縮放校正等模組以及與虹膜匹配的整體系統整合。本系統設計重點為結合眼角資訊與已建構之眼球模型參數，透過眼角距離變化進行動態補償，以因應使用者在相機前方前後移動所造成的比例誤差，進而提升虹膜匹配演算法的穩定性與準確度。整體流程區分為校正階段與驗證階段。校正階段中，系統導入先前於 PC 端計算完成之眼球模型參數（包含眼球中心座標與眼球半徑、虹膜半徑），並利用偵測到的眼角座標，計算出眼球中心與眼角之間的空間幾何關係。同時記錄左右眼角之間的初始距離，作為後續縮放修正的基準值。在驗證階段，系統根據每次輸入影像中偵測到的眼角位置，重新計算兩眼角之間的距離，並與校正階段所記錄之基準距離比較，推得縮放係數 S 。該係數應用於調整眼球中心位置與模型半徑等參數，使得在使用者前後位移時，仍可維持輸入至虹膜匹配模組之幾何資訊正確性。最終，系統輸出經匹配後之視線向量方向。

3.1 眼角偵測模組

在可見光眼動儀的演算法架構中，眼角偵測系統扮演著至關重要的角色。完整的眼動儀系統處理流程包括眼角定位、眼球模型建立、虹膜匹配，最終透過映射公式完成從物理空間到螢幕二維空間的座標轉換。眼角偵測在此系統中作為快速定位 3D 眼球中心座標的錨點，其精確度直接影響後續虹膜匹配的準確性。由於直接從完整影像中定位眼角位置並不實際，系統採用人臉特徵模型來縮小定位範圍，再透過專門的影像處理演算法精確定位眼角位置。眼角偵測系統的核心策略為：首先提高影像對比度以增強特徵，接著透過二值化處理保留眼角的邊界區域，然後運用形態學操作去除雜訊並保留眼角區塊，最後計算該區塊的重心座標作為眼角位置的代表點。實際的處理流程如 Figure 3.1 所示，此演算法共分為

5 個步驟：1. 將影像做直方圖等化，提升全域影像對比度。2. 對影像做水平梯度運算，強化眼角特徵的對比度。3. 計算二值化閾值以及對影像做二值化操作，為後續提取眼角特徵做準備。4. 對二值化影像做最大連通區域的計算，此區域即為眼角特徵區域。5. 對眼角特徵區域做重心計算，其重心即為眼角特徵的代表座標。圖中 $h(x,y)$, $b(x,y)$, $c(x,y)$ 分別代表經過各模組影像處理後所輸出的像素流。系統架構採用流水線設計，確保各處理模組能夠高效協同運作。最終輸出的座標 (x_c, y_c) 可直接映射至相機影像的座標系統，如 Figure 3.2 所示，影像中的紅點即為系統偵測到的眼角座標。在本實驗中，系統根據臉部特徵點模型所標示的內眼角特徵點座標，擷取固定範圍的區域作為輸入。採用 64×64 像素的單通道灰階影像作為輸入格式，如 Figure 3.3 所示，。

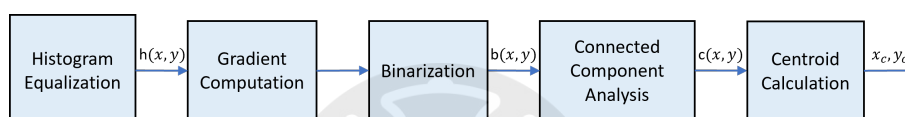


Figure 3.1: 眼角偵測演算步驟

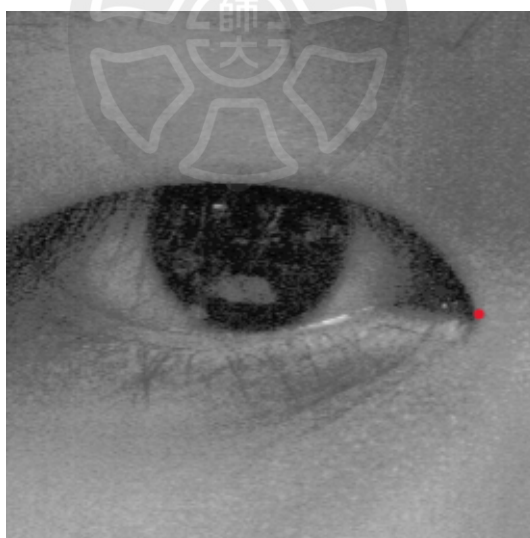


Figure 3.2: 眼角偵測系統定位的眼角位置

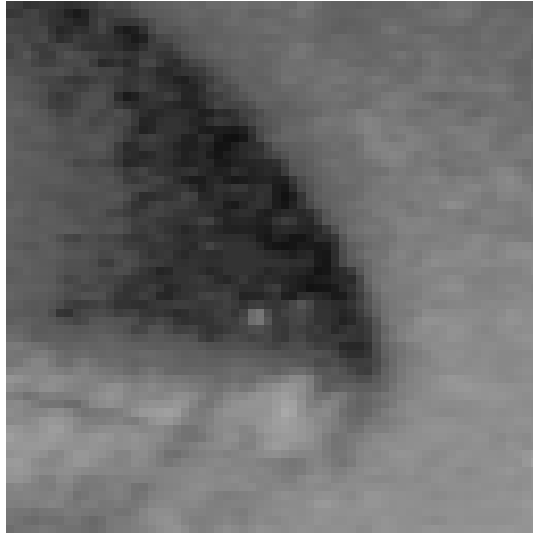


Figure 3.3: 眼角輸入影像

3.1.1 狀態控制

整個眼角偵測系統的狀態轉換如 Figure 3.4 所示。系統啟動後首先進入 IDLE 狀態，將 SRAM 中的影像資料搬移至內部 RAM，透過位址計數器判斷影像資料是否完全載入，完成後進入後續的影像處理狀態。處理流程首先進入 HISTOGRAM 狀態，統計整張影像各像素值的出現頻率，建立灰階值分布直方圖。接著轉換至 CDF（Cumulative Distribution Function）狀態，將像素值個數進行累加運算並儲存累加結果，為後續的直方圖等化處理做準備。為了減少 BRAM 的存取次數並提升處理效率，系統設計採用流水線架構，在 GRADIENT 狀態中同時完成兩項運算：直方圖等化的映射計算以及每個像素的梯度值計算。當整張影像完成梯度處理後，系統回到 HISTOGRAM 狀態重新統計經過梯度處理的影像像素分布。統計完成後進入 THRESHOLD 狀態，基於影像的統計特性計算二值化所需的最適閾值。獲得閾值後，系統轉入 BINARIZE 狀態對整張影像執行二值化處理，將灰階影像轉換為二元影像。隨後進入 MORPHOLOGY 狀態，對二值化影像進行形態學捲積操作，包括侵蝕與膨脹處理。最後在 CENTROID 狀態中，系統計算最大連通區域的重心座標，該座標即為所偵測的眼角位置，完成整個偵測流程並輸出結果。

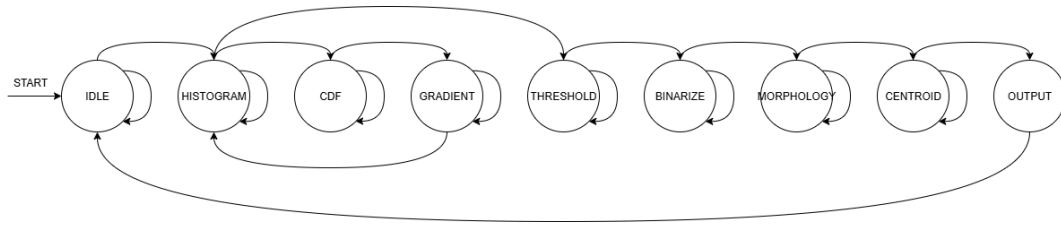


Figure 3.4: 狀態圖

3.1.2 直方圖等化模組

本模組的主要目的為增強影像對比度，使影像亮度分布更加均勻，消除過暗或過亮區域所造成的影響。模組的輸入與輸出皆為完整影像，整體架構如 Figure 3.5 所示。模組內部共分為三個功能區塊：直方圖統計、累計分布計算，以及映射轉換。各區塊協同運作，完成完整的直方圖等化處理流程。

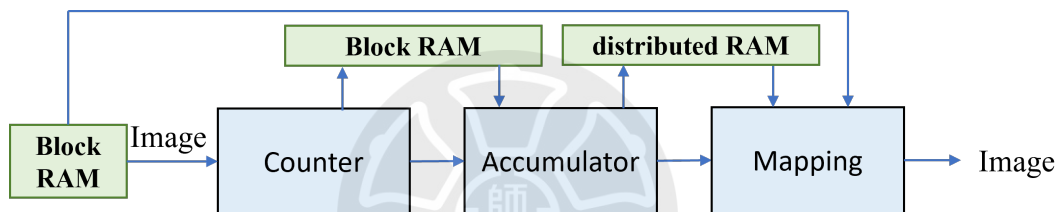


Figure 3.5: 直方圖等化模組

第一個區塊負責計算各像素值的出現頻率。參考 Li 等人所提出的方法 [17]，如 Figure 3.6 所示，電路採用兩個暫存器組成的滑動窗口架構，比較連續的兩個像素值。若像素值相同，則在暫存器中累計數量；直到出現不同像素值時，才將統計結果存入 BRAM。此方法可有效降低 BRAM 的存取次數，提升整體處理效率。

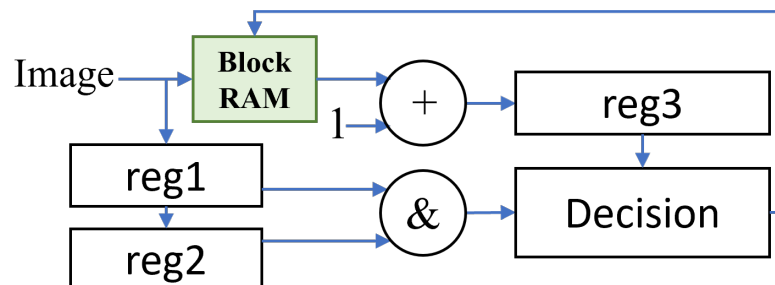


Figure 3.6: 直方圖統計模組

第二個區塊負責將直方圖統計結果轉換為累計分布函數，統計從灰階值 0 至

255 的累計像素數量。為提升硬體實現效率，本模組在存入 distributed RAM 之前即完成直方圖等化的最終計算步驟，使儲存值直接對應於映射所需的輸出灰階值。為簡化硬體電路實現，本設計對傳統直方圖等化映射公式進行優化改進。原始映射公式如式 3.1 所示，其中 f 為輸入灰階值， g 為輸出灰階值， L 為輸入影像總灰階級數， M 為輸出影像總灰階級數， $\sum_{k=0}^f n_k$ 表示從灰階值 0 到 f 的累積像素個數， n 代表整張影像的像素總數。

$$g = (M - 1) \sum_{k=0}^f \frac{n_k}{n} = \frac{(M - 1)}{n} \sum_{k=0}^f n_k, \quad 0 \leq f \leq L - 1 \quad (3.1)$$

經過數學變換，將輸出灰階級數 M 調整為 257（原 256 加 1），使正規化係數恰好為 2 的幕次（ $2^4 = 16$ ），如式 3.2 所示。此項優化使電路可直接透過 4 位元右移運算完成除法操作，大幅簡化硬體架構，如 Figure 3.7 所示。

$$g = \frac{(256 - 1)}{64 * 64} \sum_{k=0}^f n_k \approx \frac{1}{2^4} \sum_{k=0}^f n_k \quad (3.2)$$

經分析驗證，此簡化方案相對於原始公式的誤差僅為 0.39%，該微小誤差在視覺感知上幾乎無法察覺。因此，在保持影像品質的前提下，有效降低了硬體複雜度與功耗。

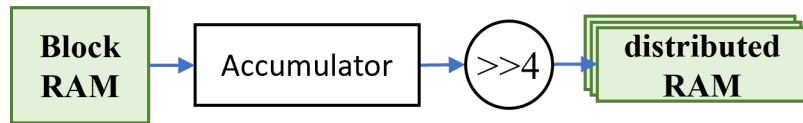


Figure 3.7: 累計分佈模組

第三個區塊負責運用前述步驟建立的查找表（LUT）將輸入影像轉換為直方圖等化後的輸出影像。考量到影像資料儲存於 64 位元寬度的 Block RAM 中，系統可在單一記憶體存取週期內同時讀取 8 個像素資料（每像素 8 位元）。本設計充分利用 distributed RAM 的快速存取特性，實現 8 個像素值的同步映射處理。當從 Block RAM 讀取的 8 個像素值同時作為 8 個 distributed RAM 的地址輸

入時，可在同一週期內並行完成所有像素的灰階值轉換，將映射處理的吞吐量提升 8 倍，顯著縮短整張影像的處理時間。

3.1.3 水平梯度計算模組

水平梯度計算模組的目的在於增強影像的橫向對比度，以便後續正確提取眼角篩選範圍。模組輸入為經過直方圖等化的影像，輸出為經過水平梯度計算的影像結果。硬體架構採用 3 個行緩衝器，以及 8 個並行減法器與比較器來提高資料處理的吞吐量，計算完畢後將結果回存至 BRAM，如 Figure 3.8 所示。系統利用行緩衝器接收前一級模組每個週期輸出的 8 個像素值。由於水平梯度計算需要相鄰像素的資訊，如式 3.3 所示，其中 G_i 表示計算出的當前像素水平梯度值， k_r 為根據影像亮度調整的增益參數， $I(x, y)$ 為當前座標的灰階值。

$$G_i(x, y) = k_r \times (I(x - i, y) - I(x + i, y)) \quad (3.3)$$

為了同時計算 8 個像素的梯度值，系統需要前後週期獲得的部分像素值。因此，模組會等待一個處理週期，待第二個週期輸入另外 8 個像素值後，便可完成第一個週期傳入的 8 個像素值計算，並在下一個週期完成 BRAM 回存動作。在電路運算實現上，系統使用比較器完成 k_r 值的調整，將兩像素相減後的 $score$ 值劃分為三個區間，對應不同的輸出結果：若 $score \leq 0$ 則輸出 0；若 $0 < score \leq 127$ 則將 $score$ 值乘以 2；若 $score > 127$ 則直接設為 255。此分段線性映射可有效增強邊緣特徵，同時避免飽和失真。

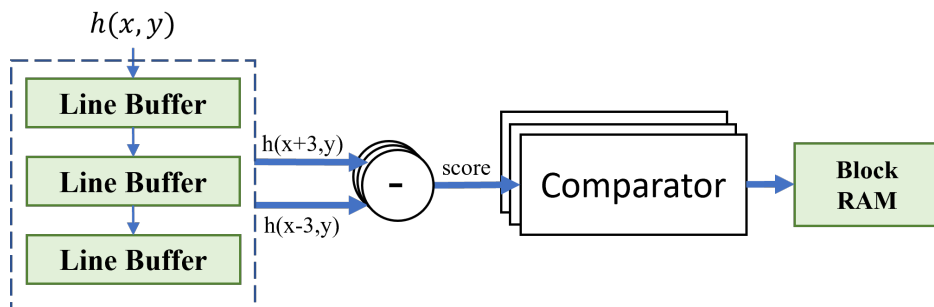


Figure 3.8: 水平梯度計算模組

3.1.4 自適應閾值計算模組

本模組負責計算影像二值化所需的閾值。在進入此模組前，系統需要先取得水平梯度模組所計算的影像灰階值直方圖，因此狀態機會回到 Histogram 狀態進行統計，待統計完畢後進入 Threshold 模組執行計算。閾值計算採用自適應演算法，從高灰階值開始由大至小累積像素數量，尋找滿足特定條件的灰階值 n 。該條件為從 n 到 255 的總像素數量必須大於設定的閾值參數 B ，其判定式如式 3.4 所示。在實際實作中，參數 B 設定為 250，確保能夠捕捉到影像中的主要特徵區域。

$$B \leq \sum_{i=n}^{255} H(i) \quad (3.4)$$

當找到滿足條件的灰階值 n 後，系統根據式 3.5 計算最終的二值化閾值。此公式計算從 n 到 255 範圍內的加權平均值，以灰階值的出現頻率作為權重。

$$Th = \frac{\sum_{i=n}^{255} i \times H(i)}{\sum_{i=n}^{255} H(i)} \quad (3.5)$$

在硬體實現上，如 Figure 3.9 所示，電路同時累加灰階值的個數以及灰階值個數與灰階值的乘積。每個處理週期累加一組數據，直到灰階值累計個數超過設定閾值 B 。當條件滿足時，系統在下一個週期執行除法運算，計算出二值化影像所需的最適閾值。

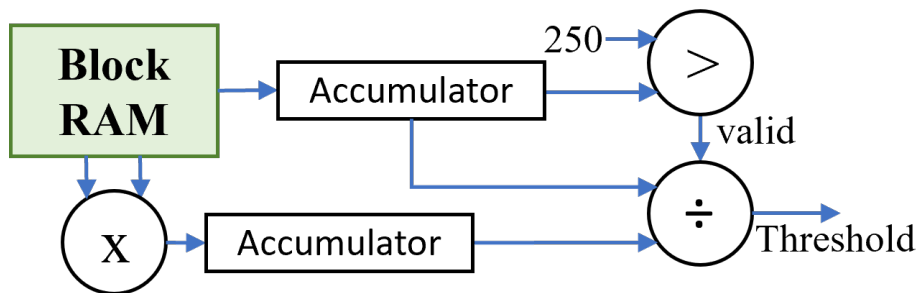


Figure 3.9: 自適應閾值計算模組

3.1.5 二值化模組

二值化模組負責完成整張影像的二值化處理，將灰階影像轉換為二元影像。模組的核心架構由一個比較器及一個儲存閾值的暫存器組成。處理流程首先讀取水平梯度運算模組所回存的影像結果，系統每次可讀取 64 位寬的資料，相當於 8 個像素的資訊。讀取完畢後，模組對這 8 個像素進行並行的閾值比較判斷，將每個像素值與預設閾值進行比較，產生對應的二元結果。完成判斷後，輸出結果會立即送入下一級模組進行後續處理。在二值化過程中，每個像素的位寬由原本的 8 位元壓縮至 1 位元，因此輸出的 8 個像素總位寬為 8 位元。這種位寬壓縮不僅減少了後續模組的資料處理負擔，也提高了系統的整體處理效率。

3.1.6 形態學模組

經過二值化處理後的 ROI (Region of Interest) 常出現結構性瑕疵，包括內部空洞以及由雜訊所導致的邊界凸起，此外，ROI 以外的區域也可能殘留雜訊。因此，本模組的設計目的在於修復這些瑕疵，提升影像品質，並正確地定位與擷取 ROI。模組共分為兩個部分：第一部分為清除全域雜訊的侵蝕處理，第二部分為提升 ROI 影像品質的侵蝕與膨脹處理。

3.1.6.1 全域雜訊侵蝕模組

由於 ROI 在輸入影像中為相對較大的區塊面積，而雜訊相較之下體積小但數量多，因此全域雜訊侵蝕處理模組採用 7×7 的捲積核，可有效過濾小雜訊並保留大塊區域。模組架構包含 6 個行緩衝器與 1 個當前行緩衝器用於保存影像資料流，以及 1 個 7×7 的陣列暫存器作為捲積核，透過加法器計算所有值的和，最後經由比較器決定輸出結果，如 Figure 3.10 所示。當前一級模組完成二值化判定後，資料會送入當前行緩衝器。當緩衝器填滿後，64 位元內容會傳遞至下一行緩衝器，並繼續接收新資料。同時，捲積核陣列暫存器持續滑動更新，直到 Line buffer2 至 Line buffer0 填充完畢且 Current line 已填充資料後，output_enable

訊號才會啟動，此為使用捲積處理的必要延遲。為了避免 7×7 捲積運算成為系統的時序瓶頸，本模組採用三級流水線架構設計。第一級 pipeline 負責 7×7 視窗提取與暫存，將從行緩衝器中提取的 49 個像素值暫存至視窗暫存器中；第二級 pipeline 將每行的 7 個像素值分別進行加總運算，產生 7 個行加總結果；第三級 pipeline 將這 7 個行加總結果進行最終加總，得到完整的捲積和。此設計將原本需要 49 個加法器的大型組合邏輯分散至三個時鐘週期中執行，有效縮短了關鍵路徑延遲，使系統能在更高的工作頻率下穩定運作。雖然增加了三個時鐘週期的處理延遲，但相較於整體 4096 個像素的處理時間，此延遲對系統吞吐量的影響可忽略不計。本模組作為形態學模組的第一個模組，承接二值化模組的 8 位元資料流輸出。為避免將 8 位元資料序列化後逐位處理所造成的效能瓶頸與延遲，設計實例化了 8 個並行的侵蝕捲積處理單元，如 Figure 3.10 中虛線部分所示，實現全位元同步處理。此架構可顯著提升處理吞吐量，並簡化資料通路設計，降低序列控制與暫存所需的額外資源，但相對地會耗費 8 倍的硬體資源與較高的功耗。

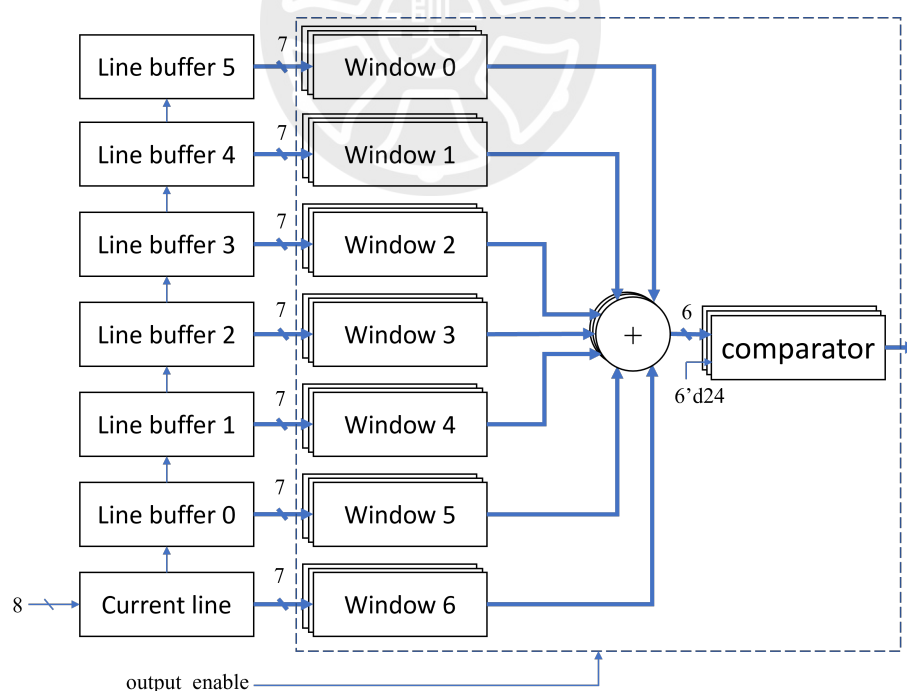


Figure 3.10: 侵蝕捲積模組

3.1.6.2 ROI 侵蝕與膨脹捲積模組

本模組與前一模組結構相似，差異在於使用較小的捲積核作為處理核心，如 Figure 3.11 所示。較小的捲積核能夠對連通區域的結構性瑕疵、內部凹洞或外部邊界凸刺提供良好的修正效果，使圖形變得單純且均勻，確保計算出的重心不會偏移。由於採用 3×3 的捲積核，模組在開始輸出前同樣須等待固定的處理延遲，待 output_enable 訊號啟動後才能產生有效輸出。

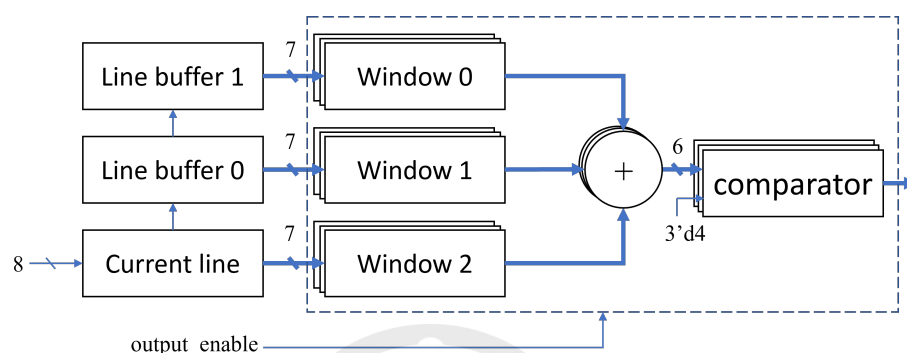


Figure 3.11: 侵蝕與膨脹捲積模組

3.1.7 重心計算模組

重心計算模組為整個眼角偵測演算法的最終階段，負責計算 ROI 的重心座標。該模組以經形態學模組處理後的影像作為輸入，輸出為重心座標，如 Figure 3.12 所示。模組架構由 1 個及開、3 個累加器與 2 個除法器組成。當前一級模組的輸出結果存回 BRAM 後，系統會一次讀取 64 位寬的資料，每個時鐘週期判斷 1 個像素是否進行累加。若該像素值為 1，則將對應的行與列座標值進行累加，同時累加參與計算的像素總數。經過 64 個時鐘週期後，系統會取出下一筆資料進行處理。此設計重用了最初設置的 BRAM 資源，提高了硬體使用效率。完成整張影像掃描後，系統進入下一個狀態，開始執行除法運算。當除法計算完成後，即可獲得精確的眼角座標位置。

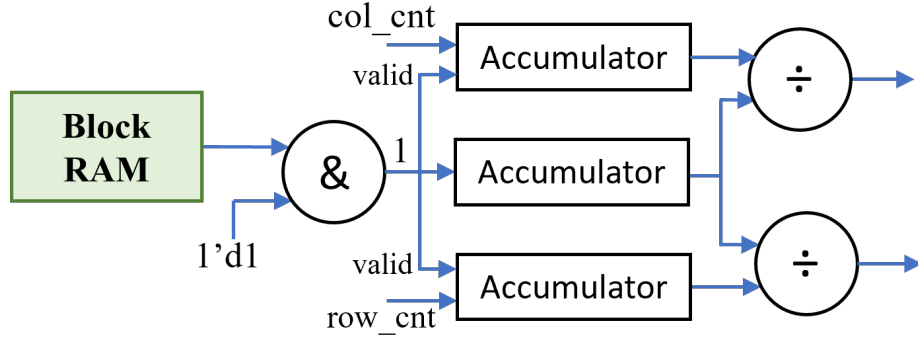


Figure 3.12: 重心計算模組

3.2 眼動儀系統整合模組

在此整合模組中，依據功能劃分為兩大處理階段：其一為在整個眼動儀使用流程中，僅在校正階段執行一次的眼球模型參數求解階段，以及在驗證階段會重複使用到的考慮頭動的虹膜匹配階段。在眼球模型參數求解模組中，我們首先透過左、右眼的眼角偵測分別獲得座標 (x_{L_C}, y_{L_C}) 和 (x_{R_C}, y_{R_C}) ，計算兩座標的幾何距離 d_f 。其次透過左眼眼角偵測所得出的眼角座標，結合眼球模型模組輸出的眼球中心座標 (x_E, y_E) ，傳入眼角眼球幾何關係計算器，得出所對應的 d_c 與 θ_c 。最後將以上參數與眼球模型計算出的眼球半徑 r_E 與虹膜半徑 r_I 一同保存，待下一階段使用。在考慮頭動的虹膜匹配階段中，系統重新檢測當前的左右眼角座標，計算當前眼角間距 d_v ，並與基準距離 d_f 比較得出縮放係數 $S = d_v/d_f$ 。接著進行參數動態調整：計算縮放後的距離 $\hat{d}_c = S * d_c$ ，再通過幾何關係 $(\hat{d}_c \cos \theta_c, \hat{d}_c \sin \theta_c)$ 與當前眼角座標相加，得到調整後的眼球中心座標 (x_E, y_E) 。同時計算調整後的半徑參數 $\hat{r}_E = S * r_E$ 和 $\hat{r}_I = S * r_I$ ，並計算 $\tau = \hat{r}_I/\hat{r}_E$ 。如 Figure 3.13 所示，左側紅色虛線框表示眼球模型參數求解階段（校正階段執行一次），右側紅色虛線框表示考慮頭動的虹膜匹配階段（校正與驗證階段重複執行）。最終將調整後的眼球中心座標、 τ 參數以及眼睛影像一同送入虹膜匹配模組，輸出精準的凝視向量角度 θ 與 ϕ 。

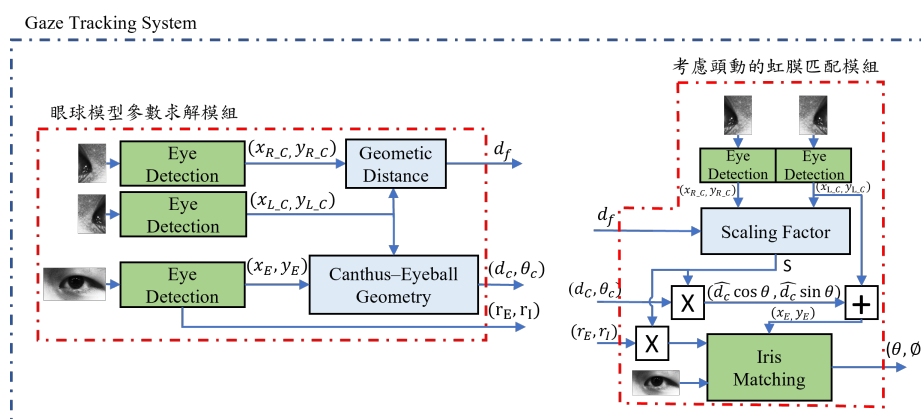


Figure 3.13: 眼動儀系統整合模塊

3.2.1 系統控制狀態機

整個眼動儀系統整合模組的控制核心採用有限狀態機設計，以協調校正與驗證兩個主要階段的執行流程。整個過程的狀態圖可參見 Figure 3.14。系統啟動後首先進入 IDLE 狀態等待外部觸發信號，接收到 start 信號後轉入 LOAD_IMAGES 狀態，負責從外部 SRAM 載入所需的影像資料，包括左右眼角影像各 64×64 像素以及眼睛影像 380×200 像素。影像載入程序透過統一的 SRAM 接口進行，系統會依據預先定義的地址映射分別讀取三種不同類型的影像資料。影像載入完成後，狀態機根據 calibration_mode 控制信號決定進入校正階段或驗證階段的處理流程。當 calibration_mode 信號為高電平時，系統進入校正階段，首先執行 CALIBRATION_SOLVER 狀態。此階段會依序進行左右眼角偵測、眼球模型參數計算以及眼角與眼球中心的幾何關係推算，建立包含基準眼角距離 d_f 、眼角到眼球距離 d_c 、幾何角度 θ_c 以及眼球半徑 r_E 與虹膜半徑 r_I 的完整基準參數集合。參數求解完成後，系統自動轉入 CALIBRATION_MATCHING 狀態，此階段會執行考慮頭動的虹膜匹配演算法，用於校正階段的九點凝視角度計算，每次匹配完成後會輸出對應的 θ 與 ϕ 角度值供 PC 端建立映射方程式。當 calibration_mode 信號為低電平時，系統直接從 LOAD_IMAGES 跳轉至 TRACKING_MATCHING 狀態，進入驗證階段的持續追蹤模式。在此狀態下，系統會重複執行眼角偵測、縮放係數計算、參數動態調整以及虹膜匹配

的完整流程，每處理一幀影像即輸出一組凝視角度結果。無論是校正階段的 CALIBRATION_MATCHING 或驗證階段的 TRACKING_MATCHING，當單次處理循環完成後，系統均會轉入 OUTPUT_RESULTS 狀態，完成結果輸出並設置相應的有效信號，隨後返回 IDLE 狀態等待下一次處理請求。

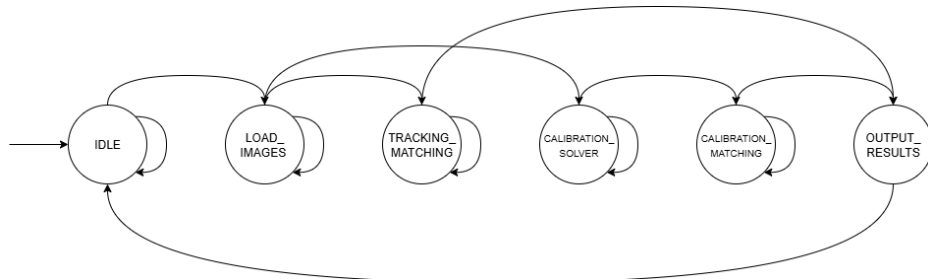


Figure 3.14: 狀態圖

3.2.2 眼球模型參數求解模組

本模組作為校正階段的核心組件，負責整合多個子系統以建立完整的眼球模型基準參數。整個模組採用影像預載入與並行處理架構，透過一次性載入所有影像資料並同步執行多個計算流程。模組內部狀態機設計採用優化的並行處理策略。系統啟動後首先進入影像載入階段，連續從 SRAM 讀取三種影像資料：左眼角影像、右眼角影像以及眼睛影像，並分別儲存至對應的內部記憶體區域。影像載入完成後，左右眼角偵測模組與眼球模型計算同步啟動進行並行處理。當兩個眼角偵測模組並行完成特徵點定位後，系統啟動眼角間距計算，利用已取得的左右眼角座標計算基準距離 d_f 。待眼球模型模組計算完畢後，接著進入眼角與眼球的幾何關係計算階段，這兩個部分我們採用兩個獨立的 CORDIC 演算法實例計算不同的幾何參數。第一個 CORDIC 實例專門計算眼角間距 d_f ，僅需要距離運算而忽略角度輸出；第二個 CORDIC 實例則同時計算眼角到眼球中心的距離 d_c 與方位角 θ_c ，為後續的空間幾何推算提供完整參數。CORDIC 演算法實現採用 12 次疊代的設計，在精度與資源使用間取得良好平衡。每個 CORDIC 實例首先將輸入的座標差值進行象限正規化，接著透過移位與加減運算逐步收斂至目標值。計算完成後，各模組將計算結果分別儲存至對應的內部暫存器中。模組間

的資料同步採用握手協議確保正確性，所有計算模組完成後會產生統一的完成信號，供後續的考慮頭動虹膜匹配模組讀取使用。

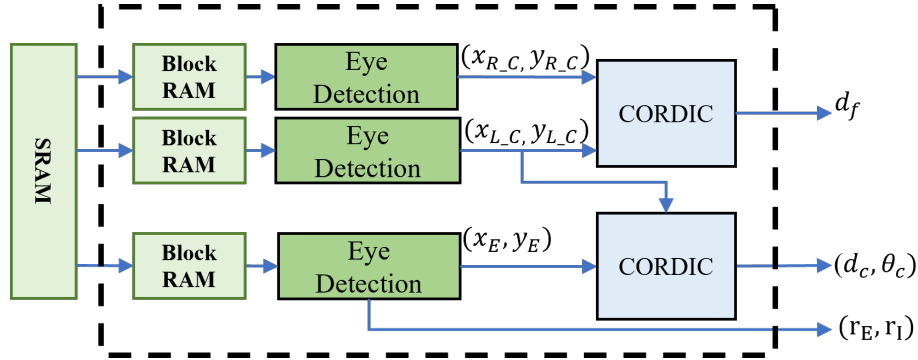


Figure 3.15: 眼球模型參數求解模組

3.2.3 CORDIC 幾何關係計算模組

在眼球模型參數求解流程中，精確計算眼角間距離與眼角到眼球中心的幾何關係是建立基準參數的關鍵步驟。傳統的距離與角度計算需要開方運算與反正切函數，這類超越函數在 FPGA 硬體實現上不僅資源消耗龐大，亦存在計算精度與延遲的權衡問題。為解決此挑戰，本研究採用 CORDIC (Coordinate Rotation Digital Computer) 演算法進行幾何關係計算，透過純粹的移位與加減運算實現高效能的距離與角度同步計算。CORDIC 演算法的核心概念基於向量旋轉原理，透過一系列預定義角度的微小旋轉，將任意向量逐步旋轉至目標方向。本實現採用旋轉模式的 CORDIC 演算法，目標是將輸入向量旋轉至與 X 軸對齊，過程中 X 分量收斂至向量的模長，而累積的旋轉角度即為向量的相位角。演算法在每次迭代中根據當前 Y 分量的正負號決定旋轉方向，若 Y 分量為正則進行順時針旋轉，反之則逆時針旋轉，直至 Y 分量收斂至零。硬體實現中採用 12 次迭代的設計策略，意在計算精度與資源消耗間取得平衡。每次迭代的旋轉角度為 $\arctan(2^{-i})$ ，其中 i 為迭代次數，這些角度值預先儲存於查找表中以避免即時計算。CORDIC 演算法迭代過程完全避免乘法器使用，僅需移位與加減運算。然而為確保距離計算的精度，我們在結果輸出階段採用一個 16 位乘法器進行 CORDIC 增益補償，將理論增益 $K \approx 1.647$ 的影響修正回來。經過 12 次迭代後，角度解析度可達到約

0.02 度，距離計算精度在定點數運算下亦能滿足系統需求。在眼球模型參數求解模組設計中實例化兩個獨立的 CORDIC 計算單元，分別負責不同的幾何參數計算。第一個 CORDIC 實例專門計算眼角間距離 d_f ，輸入為左右眼角座標的差值，僅輸出距離結果而忽略角度計算，以節省不必要的硬體資源。第二個 CORDIC 實例則執行完整的距離與角度計算，輸入為眼角與眼球中心座標的差值，同時輸出距離 d_c 與方位角 θ_c 。雙 CORDIC 並行架構雖然增加了約兩倍的邏輯資源使用，但可同步完成所有幾何參數的計算，縮短整體處理時間。演算法實現過程中特別考慮了象限處理與符號管理。由於 CORDIC 演算法在第一象限具有最佳收斂特性，輸入向量首先進行象限正規化，將座標差值轉換為絕對值並記錄原始象限資訊。計算完成後再根據象限資訊對角度結果進行修正，確保最終輸出角度涵蓋完整的 360 度範圍。

3.2.4 考慮頭動虹膜匹配模組

本模組作為眼動儀系統的核心處理單元，負責在校正與驗證兩個階段中執行動態眼角偵測、縮放係數計算以及虹膜匹配演算法，實現對使用者頭部前後移動的自適應補償。模組設計採用模組化架構，如 Figure 3.17 所示，內部狀態機包含四個主要處理階段，其狀態轉換圖如 Figure 3.16 所示。首先進入眼角偵測階段，系統重新執行左右眼角的特徵點定位，獲取當前影像中的眼角座標。由於影像已在前一階段載入至內部記憶體，兩個眼角偵測模組可同步運作，縮短整體處理時間。眼角偵測完成後，系統轉入縮放係數計算階段，透過比較當前眼角間距與校正階段建立的基準距離 d_f ，推算出頭部位移所對應的縮放係數 S 。縮放係數計算模組首先採用 CORDIC 演算法計算當前左右眼角間的距離 d_v ，接著執行除法運算獲得縮放比例 $S = d_v/d_f$ 。除法運算採用 Altera 浮點除法器 IP 核心實現，具備 14 個時鐘週期的 pipeline 延遲設計，能在維持高時鐘頻率的同時提供精確的浮點除法結果，有效避免了定點數運算可能產生的精度損失與溢出問題。獲得縮放係數後，系統同步進行多項參數的動態調整計算，包括縮放後的眼角到眼球距離

$\hat{d}_c = S * d_c$ 、調整後的眼球半徑 $\hat{r}_E = S * r_E$ 以及虹膜半徑 $\hat{r}_I = S * r_I$ 。參數動態調整階段採用並行乘法運算架構，同時執行多個縮放計算以提升處理效率。眼球中心座標的重新定位透過幾何變換實現，利用基準角度 θ_c 與縮放後距離 $\hat{d}_c$ ，結合三角函數運算計算出調整後的眼球中心位置 $(\hat{x}_E, \hat{y}_E)$ 。三角函數運算採用預先建立的查找表實現，cos 與 sin 值以定點數格式儲存，通過直接查表獲得所需精度。待參數動態調整階段完成後，系統進入虹膜匹配階段，將調整後的眼球中心座標、眼球半徑、虹膜半徑以及眼睛影像一同送入虹膜匹配核心模組。該核心模組整合了先前開發的 limbus circle 匹配演算法，透過角度搜尋與亮度差異評估，精確計算出當前的凝視方向角度 θ 與 ϕ 。處理完成後，系統進入最終的結果輸出階段，將計算結果連同有效性標誌一併輸出，供後續的視線追蹤應用使用。

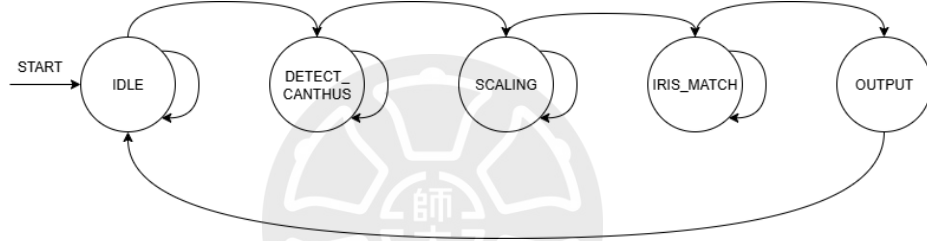


Figure 3.16: 狀態圖

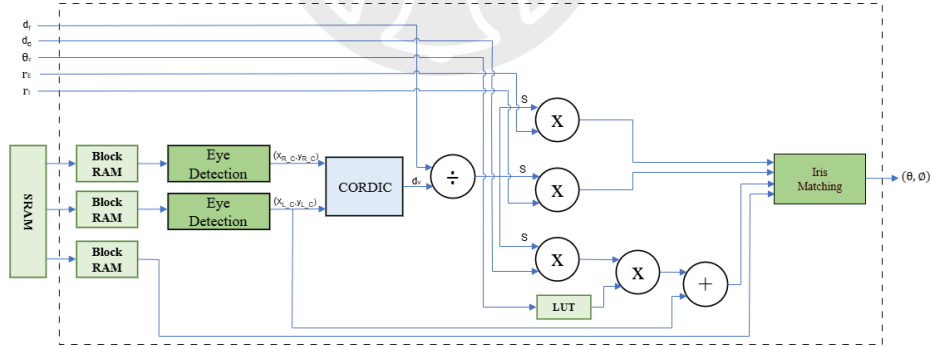


Figure 3.17: 考慮頭動虹膜匹配模組

第四章 實驗結果

本章節會先介紹實驗環境與設備，接著比較幾個計算過程中 FPGA 模組的計算結果與 PC 計算時的差異，以及驗證最終 FPGA 計算出來的輸出結果與效能。

4.1 實驗環境與設備

本論文主要在 FPGA 上實現眼角偵測演算法，透過眼角偵測演算法設計的電路計算出眼角座標，使用的 FPGA 為配備 CycloneIV 系列晶片的 Altera DE2-115 教育開發平台 Figure 4.1，在會 PC 上使用 Quartus II 進行開發與使用 modelsim 模擬波型，上板子時使用 SignalTapII 進行檢查與除錯，最後將 FPGA 上算出的結果角度寫入 SRAM，並由 PC 確認計算結果是否與使用 modelsim 模擬計算時相同，以及在照片中標示出經過 FPGA 實現的眼角偵測算法得到的座標位置，確認該座標位置是否有順利定位到照片中眼角位置。

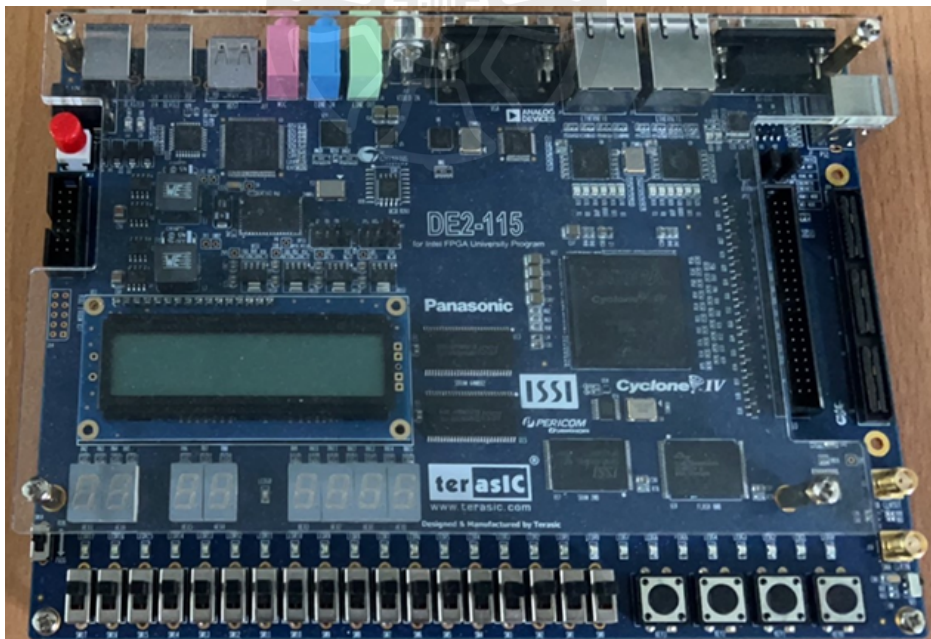


Figure 4.1: Altera DE2-115 教育開發平台

4.2 眼角偵測模組結果與分析

4.2.1 硬體效能分析

Figure 4.2 為最終使用到的硬體，結合效能分析後的結果如 Table 4.1 所示，所使用的 logic elements 為 4,936/114,480 (4%)，而內部 memory bits 為 52,224/3,981,312 (1%)。系統最高可運行於 85.01 MHz，透過 pipeline 以及 parallel 設計後，其計算速度的提升如 Figure 4.3 所示，我們加入平行畫計算觀察到的計算速度提升，可歸納為對全域影像像素做獨立的影像處理所帶來的效益。分析加速效果，pipeline 設計相對於基本版本約有 2 倍的速度提升，這主要得益於各模組間的流水線處理，減少了存取等待時間。而 pipeline 結合 parallel 的設計相對於基本版本約有 1.7 倍的加速效果，顯示平行處理在影像像素級運算中的顯著優勢。由於眼角偵測演算法中的直方圖等化、梯度計算、二值化等步驟都可以對多個像素同時進行處理，平行化設計能夠充分發揮 FPGA 的並行運算能力。整體而言，本研究所設計的硬體架構在資源使用上相當高效，邏輯元件使用率僅 4%，記憶體使用率約 1%，為後續整合其他模組預留了充足的資源空間。在處理速度方面，系統達到 5,115 frames/s 的表現，遠超過一般即時應用的需求。作為可見光眼動儀系統中的眼角偵測模組，此效能表現不僅滿足了高速影像處理的要求，更為後續的縮放係數計算、虹膜匹配計算等模組預留了充足的處理時間餘裕。此優異的效能表現充分證明了本架構在可見光眼角偵測應用中具備高度的可行性與實用性，同時也為整體眼動儀系統的即時運作提供了堅實的硬體基礎。

Flow Summary	
<<Filter>>	
Flow Status	Successful - Mon Jul 14 13:50:14 2025
Quartus Prime Version	20.1.0 Build 711 06/05/2020 SJ Lite Edition
Revision Name	canthus_detection_top
Top-level Entity Name	canthus_detection_top
Family	Cyclone IV E
Device	EP4CE115F29C7
Timing Models	Final
Total logic elements	4,936 / 114,480 (4 %)
Total registers	3433
Total pins	54 / 529 (10 %)
Total virtual pins	0
Total memory bits	52,224 / 3,981,312 (1 %)
Embedded Multiplier 9-bit elements	2 / 532 (< 1 %)
Total PLLs	0 / 4 (0 %)

Figure 4.2: summary

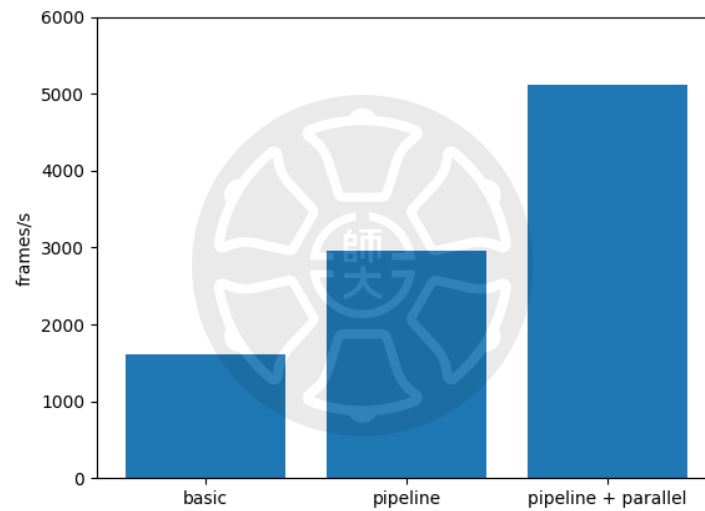


Figure 4.3: speedup

Table 4.1: FPGA 規格及效能分析

Device	Cyclone IV E EP4CE115F29C7
Fmax	85.01MHz
Total logic elements	4936/114480
Total memory bits	52224/3981312
Processing speed	5115 frames/s
Critical path	4×MUX + 8×Adder + 3×Gradient Compute Units (gradient calculator module)

4.2.2 結果比較

Figure 4.4與 Figure 4.5分別為用電腦計算時得到最佳眼角定位點，以及透過 FPGA 計算後得到的最佳眼角定位點結果。

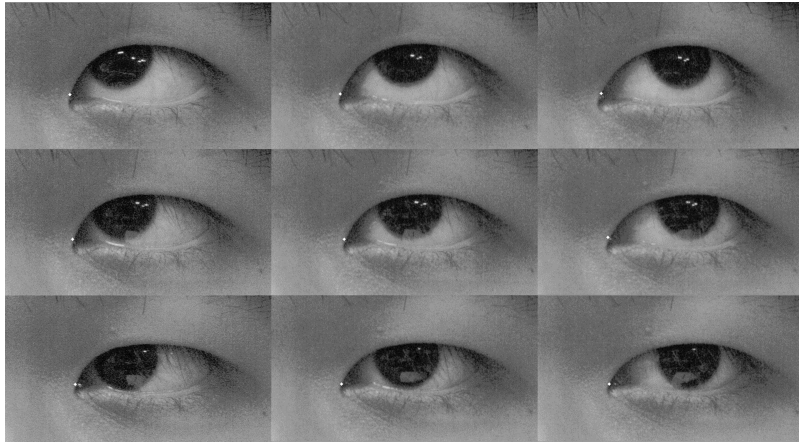


Figure 4.4: 電腦計算結果 (圖中白點為眼角位置)

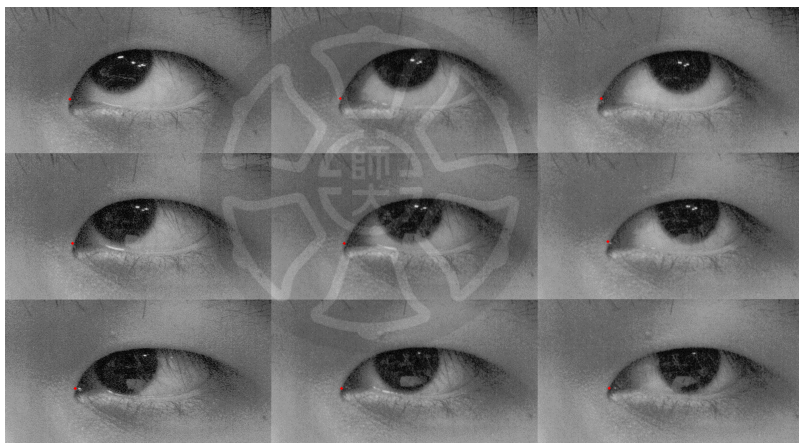


Figure 4.5: 晶片計算結果 (圖中紅點為眼角位置)

跟使用不同的受試者時會有不同的眼球模型，Figure 4.6與 Figure 4.7分別為不同的眼球模型去做計算時在 PC 以及 FPGA 得到的最佳眼角定位點。

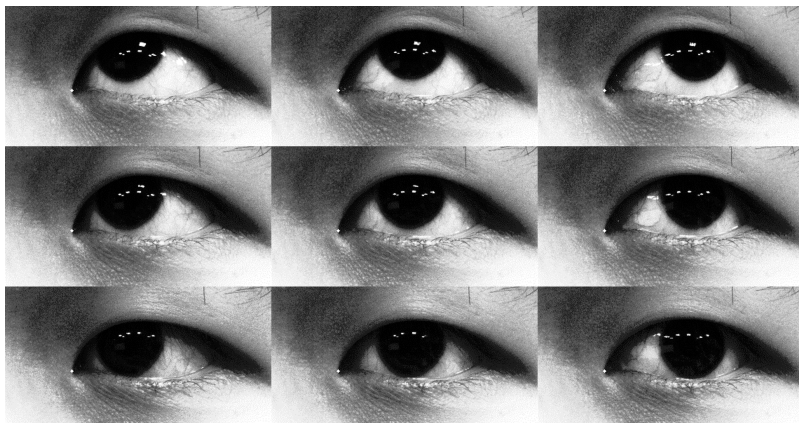


Figure 4.6: 不同眼球模型的電腦計算結果 (圖中白點為眼角位置)

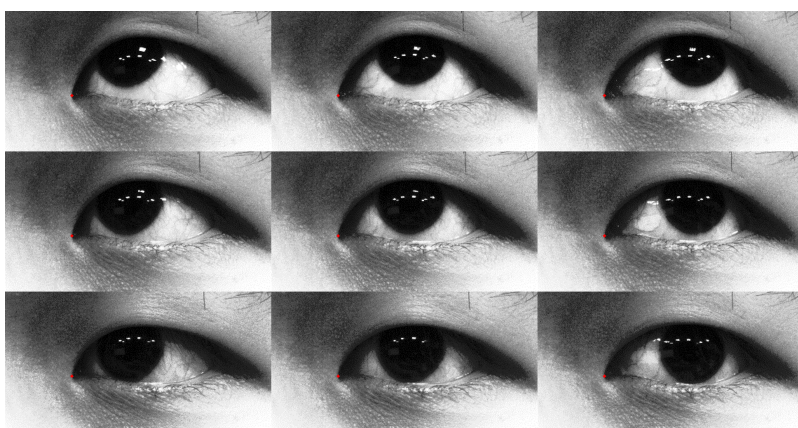


Figure 4.7: 不同眼球模型的晶片計算結果 (圖中紅點為眼角位置)

4.3 眼動儀系統整合模組結果與分析

4.3.1 硬體效能分析

Figure 4.8顯示了最終系統的硬體資源使用情況。經過完整系統整合後的效能分析結果如 Table 4.2 所示，系統總共使用了 15,269 個邏輯元件，占總可用資源 114,480 的 13%，而內部記憶體使用量為 1,401,916 位元，占總可用記憶體 3,981,312 位元的 35%。系統在 FPGA 上的最高運行頻率可達 70.35MHz，顯示出良好的時序性能。雖然邏輯元件的使用量相較於單獨模組有所增加，但仍僅占總資源的 13%，保留了充足的擴展空間。記憶體使用量的大幅提升主要來自於系統整合後各模組的緩衝需求以及影像處理過程中的暫存資料，但整體使用率仍在可接受範圍內。在 50 MHz 的工作頻率下，整合系統每秒能完成 476 張影像的完整處理流程，包括眼角偵測和虹膜匹配。相較於原始虹膜匹配系統在相同頻率下每秒 525 張的處理能力，整合後的性能略有下降，主要原因包括新增的眼角偵測演算法、中間的縮放係數計算所增加的處理延遲，以及 SRAM 介面的頻寬限制。由於 SRAM 介面的位寬僅為 16 位元，在影像載入階段會產生明顯的資料傳輸瓶頸，造成整體處理流程的延遲。然而，當螢幕更新頻率為 60 Hz 時，系統仍能達到即時處理的需求，滿足實際應用的性能要求。進一步與 Table 4.3 所示之電腦端運行效能相比，在相同輸入影像條件下，Intel Xeon W-2235（6 核 12 執

行緒，3.79 GHz，32 GB RAM）僅能達到平均 25.71 FPS，遠低於 FPGA 架構的 476 FPS。這顯示本研究所設計之專用硬體加速架構在資源使用與處理速度間取得了良好平衡，不僅能滿足即時處理需求，更擁有大幅超越通用處理器的運算效能，足以負擔高幀率相機的輸入影像流。

Flow Summary	
Flow Status	Successful - Mon Jul 14 20:53:57 2025
Quartus Prime Version	20.1.0 Build 711 06/05/2020 SJ Lite Edition
Revision Name	gaze_tracker_integration_top
Top-level Entity Name	gaze_tracker_integration_top
Family	Cyclone IV E
Device	EP4CE115F29C7
Timing Models	Final
Total logic elements	15,269 / 114,480 (13 %)
Total registers	9107
Total pins	91 / 529 (17 %)
Total virtual pins	0
Total memory bits	1,401,916 / 3,981,312 (35 %)
Embedded Multiplier 9-bit elements	47 / 532 (9 %)
Total PLLs	0 / 4 (0 %)

Figure 4.8: summary

Table 4.2: FPGA 規格及效能分析

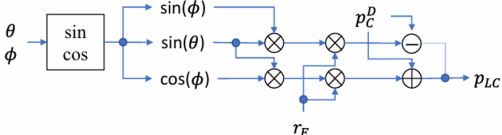
Device	Cyclone IV E EP4CE115F29C7
Fmax	70.35MHz
Total logic elements	15269/114480
Total memory bits	1401916/3981312
Processing speed	476 frames/s
Critical path	 Rotation Angle Selection Module(in iris matching module)

Table 4.3: 電腦規格及效能分析

CPU Model	Intel(R) Xeon(R) W-2235 @ 3.80 GHz
CPU Cores / Threads	6 cores / 12 threads
CPU Architecture	Cascade Lake-W (2nd Gen Xeon W / equivalent to 9th Gen Core)
Base / Max Turbo Frequency	3.79 GHz / 4.60 GHz
Cache	L1: 384 KB, L2: 6.0 MB, L3: 8.2 MB
Memory Capacity	32 GB DDR4-2933 (DIMM, 2 of 8 slots used)
Operating System	Windows 11 Pro 24H2 (OS Build 26100.4652)
Processing Speed	25.71 FPS (average)

4.3.2 結果比較

Figure 4.9與 Figure 4.10分別為用電腦計算時得到最佳虹膜匹配旋轉角，以及透過 FPGA 計算後得到的最佳虹膜匹配旋轉角結果。

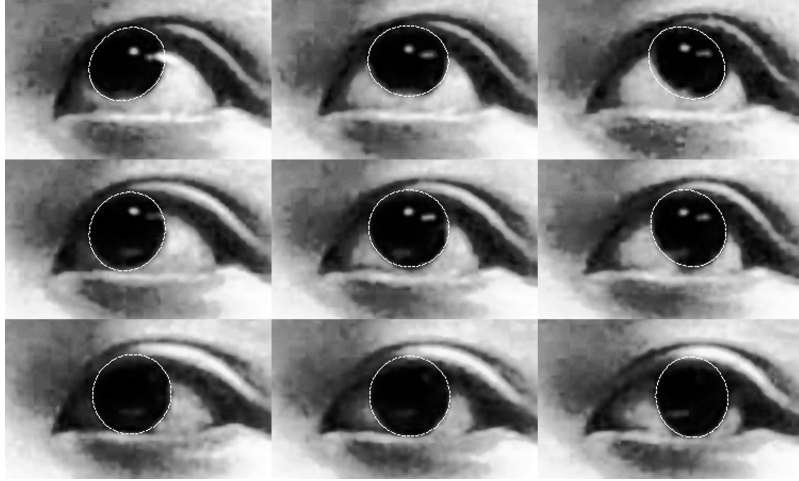


Figure 4.9: 電腦計算結果

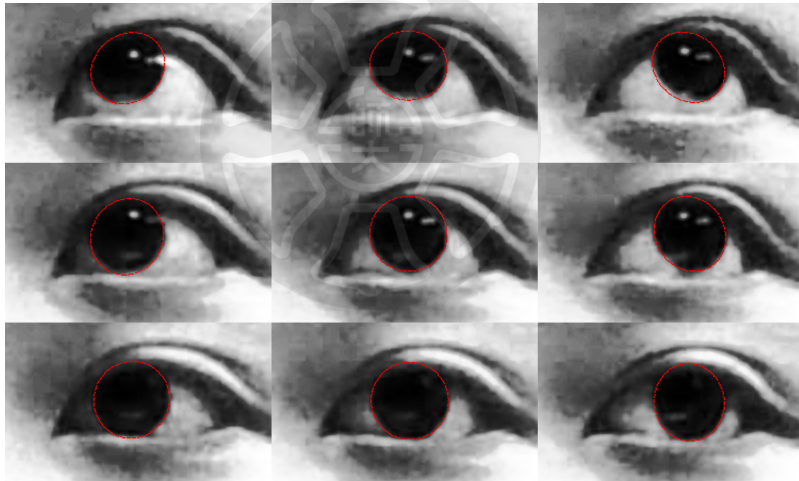


Figure 4.10: 晶片計算結果

Figure 4.11與 Figure 4.12分別為同一受試者於相同眼球模型參數下改變使用距離去做計算時，在 PC 以及 FPGA 得到的最佳虹膜匹配旋轉角度。

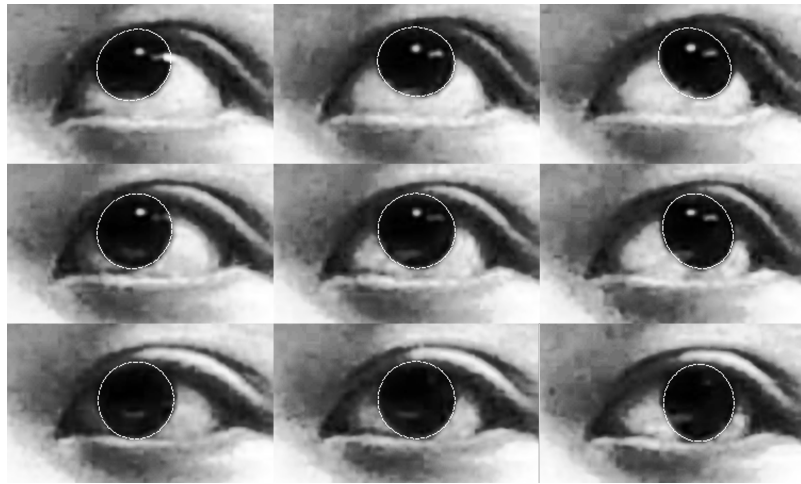


Figure 4.11: 電腦計算結果 (離相機較近距離)

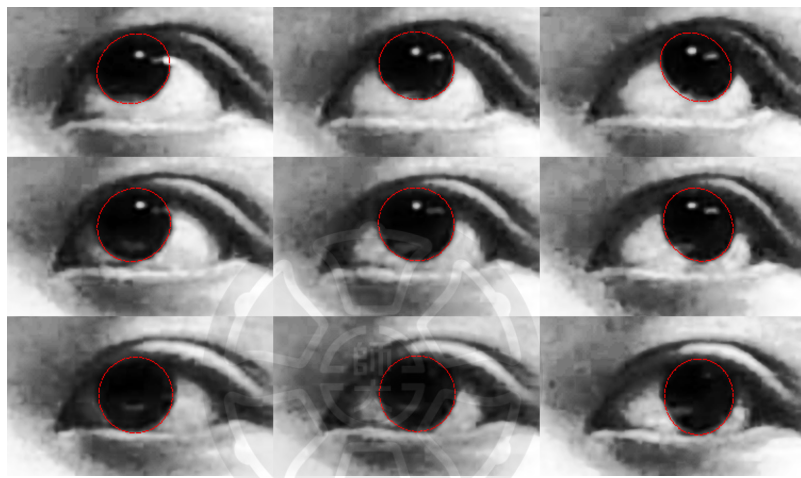


Figure 4.12: 晶片計算結果 (離相機較近距離)

第五章 結果與未來展望

5.1 結論

本研究成功在 FPGA 平台上實現了完整的可見光眼動儀系統，涵蓋眼角偵測、眼球模型參數求解、幾何關係計算以及考慮頭動的虹膜匹配等核心模組。透過硬體加速技術有效提升了系統的即時運算能力，實現了從單一模組到整合系統的完整硬體映射。在眼角偵測模組的設計上，研究成果顯示系統在 Altera DE2-115 FPGA 上達到 5,115 frames/s 的處理速度，遠超過市面上高階眼動儀設備的處理速度（約 2,000 FPS），證明了本架構在眼角偵測應用中的高效能表現。架構設計採用管線化與平行化處理策略，將直方圖等化、梯度計算、自適應閾值、二值化及形態學運算等模組進行最佳化設計。特別是在形態學處理中實作了 8 個並行的捲積單元，有效提升了資料處理的吞吐量。單一眼角偵測模組的硬體資源使用率僅占 4% 的邏輯元件和 1% 的記憶體，展現出優異的資源使用效率。在系統整合層面，本研究進一步實現了完整的眼動儀硬體系統，成功整合眼角偵測、眼球幾何關係推算、動態縮放補償及虹膜匹配等模組。整合後的系統使用 15,269 個邏輯元件（占總資源 13%）和 1,401,916 位元記憶體（占總資源 35%），仍維持良好的資源使用效率。系統整合後在 50 MHz 工作頻率下每秒能完成 476 張影像的完整處理流程，雖然相較於原始虹膜匹配系統的 525 frames/s 略有下降，但當螢幕更新頻率為 60 Hz 時，系統仍能滿足即時處理需求。透過引入縮放比估算機制，系統有效解決了使用者頭部前後移動時的參數校正問題，提升了系統在實際應用環境中的穩定性與準確度。實驗結果證明，整合系統能夠準確定位眼角位置，動態調整眼球模型參數，並維持穩定的視線追蹤效能，為可見光眼動儀的實用化奠定了堅實的硬體基礎。

5.2 未來展望

基於本研究的成果，未來可朝以下幾個方向進行改進與擴展：首先，在硬體效能優化方面，當前系統的主要瓶頸來自於 DE2-115 開發板上外部 SRAM 介面的 16 位元頻寬限制，造成影像載入階段的資料傳輸瓶頸。未來可透過升級至具備更高頻寬記憶體介面的 FPGA 開發平台（如配備 64 位元或 128 位元 DDR3/DDR4 記憶體的開發板），或在現有平台上實現多通道並行存取架構，預期可將系統整體處理速度提升至接近單一模組的性能水準。此外，可探討更先進的記憶體管理策略，例如採用乒乓緩衝或片上記憶體與外部記憶體的混合架構設計，進一步減少資料存取延遲。其次，在演算法改進上，可考慮整合深度學習加速器，將 AI 模型的特徵點偵測也納入硬體加速範圍，實現端到端的硬體眼動追蹤系統。同時可研究動態可重組架構，根據不同應用場景自適應調整硬體資源配置，在校正階段與追蹤階段間動態切換資源分配，進一步提升整體系統效率。第三，在系統擴展性方面，目前系統仍有充裕的邏輯資源（87% 未使用），可進一步整合更多功能模組，如多目標追蹤、瞳孔直徑測量或眨眼偵測等功能。同時可探索多尺度處理架構，支援不同解析度的影像輸入，在保持高精度的同時適應不同應用需求。第四，在商業化應用考量上，可進一步研究 ASIC 實現方案，以大幅降低成本和功耗。結合邊緣運算架構與 5G 通訊技術，開發分散式眼動追蹤系統，使系統能夠應用於虛擬實境、擴增實境、醫療診斷或人機介面等多元領域。最後，可建立標準化的硬體介面和軟體驅動程式，使系統更容易整合至各種嵌入式平台，並開發完整的軟體開發工具包（SDK），降低技術門檻，加速產業應用推廣。透過這些改進方向的實現，本研究成果有望發展成為具備高度商業價值的可見光眼動追蹤解決方案。

References

- [1] W.-C. Kao, J.-Y. Li, S.-C. Lin, and Y.-C. Chiu, “High precision canthus alignment for visible-spectrum gaze tracking system,” in *Proc. 2019 IEEE International Conference on Consumer Electronics - Taiwan (ICCE-TW)*, 2019, pp. 1–2.
- [2] A. Rosenfeld and J. L. Pfaltz, “Sequential operations in digital picture processing,” *Journal of the ACM*, vol. 13, no. 4, pp. 471–494, 1966.
- [3] W.-C. Kao and Y.-C. Chiu, “Eyeball model construction and matching for visible-spectrum gaze tracking systems,” in *Proc. 2018 IEEE 8th International Conference on Consumer Electronics - Berlin (ICCE-Berlin)*, 2018, pp. 1–2.
- [4] W.-C. Kao, R.-X. Ying, and Y.-F. Lee, “Head motion detection and compensation for visible-spectrum gaze tracking systems,” in *Proc. 2023 IEEE International Conference on Consumer Electronics (ICCE)*, 2023, pp. 1–2.
- [5] W.-C. Kao, K.-J. Huang, and Y.-C. Chiu, “Eyeball model construction with head movement compensation for gaze tracking systems,” in *Proc. 2020 IEEE International Conference on Consumer Electronics (ICCE)*, 2020, pp. 1–2.
- [6] W.-C. Kao, C.-Y. Lin, C.-C. Hsu, C.-Y. Lee, B.-Y. Ke, and T.-Y. Su, “Optimal iris region matching and gaze point calibration for real-time eye tracking systems,” in *Proc. 2016 IEEE International Conference on Consumer Electronics (ICCE)*, 2016, pp. 443–444.
- [7] W.-C. Kao and S.-C. Lin, “Iris region matching for visible-spectrum gaze trackers,” in *Proc. 2020 IEEE International Conference on Consumer Electronics (ICCE)*, 2020, pp. 1–2.

- [8] W.-C. Kao *et al.*, “A novel approach to detect & track iris for a different and adverse dataset,” arXiv preprint, 2019.
- [9] W.-C. Kao and Y.-C. Chiu, “Hierarchical search of optimal limbus circle matching for gaze tracking systems,” in *Proc. 2017 IEEE 7th International Conference on Consumer Electronics - Berlin (ICCE-Berlin)*, 2017, pp. 233–234.
- [10] W.-C. Kao, H.-H. Liu, K.-D. Hong, and Y.-F. Lee, “Hardware engine of limbus circle matching for visible-spectrum gaze tracker,” in *Proc. 2024 IEEE International Conference on Consumer Electronics (ICCE)*, 2024, pp. 1–2.
- [11] W.-C. Kao, K.-J. Huang, and Y.-C. Chiu, “Eyeball model construction with head movement compensation for gaze tracking systems,” in *Proc. 2020 IEEE International Conference on Consumer Electronics (ICCE)*, 2020, pp. 1–2.
- [12] W.-C. Kao, P.-T. Cheng, Y.-C. Chiu, K.-J. Huang, H.-Y. Chen, and M. Yen, “Toward free head motion for visible-spectrum gaze tracking systems,” in *Proc. 2021 IEEE International Conference on Consumer Electronics (ICCE)*, 2021, pp. 1–2.
- [13] W.-C. Kao, R.-X. Ying, and Y.-F. Lee, “Head motion detection and compensation for visible-spectrum gaze tracking systems,” in *Proc. 2023 IEEE International Conference on Consumer Electronics (ICCE)*, 2023, pp. 1–2.
- [14] W.-C. Kao, P.-T. Cheng, Y.-C. Chiu, K.-J. Huang, H.-Y. Chen, and M. Yen, “Toward free head motion for visible-spectrum gaze tracking systems,” in *Proc. 2021 IEEE International Conference on Consumer Electronics (ICCE)*, 2021, pp. 1–2.
- [15] S. Ravichandran, H.-K. Su, W.-K. Kuo, D. Dhanasekaran, M. Mahalingam, and J.-P. Yang, “Parallel processing of sobel edge detection on FPGA: Enhancing real-time image analysis,” *Sensors*, vol. 25, no. 12, p. 3649, 2025.

- [16] T. D. Pham, B. G. Bach, L. T. Luu, M. D. Nguyen, H. D. Pham, K. B. Anh, X. Q. Nguyen, and C. P. Quoc, “An FPGA-based solution for convolution operation acceleration,” *arXiv preprint arXiv:2206.04520*, 2022.
- [17] X. Li, Q. Guo, T. Pu, and Y. Zhong, “Real-time image histogram equalization using FPGA,” *Proceedings of SPIE*, 2005, xilinx XC4000 implementation.



自傳

我在選擇研究所階段時，深受國立臺灣師範大學於 2023 年成立的跨域科技產業創新研究學院（CIAI）吸引。該學院下設的 AI 跨域應用研究所聚焦在人工智慧、物聯網、智慧製造、電腦視覺及自然語言處理等領域，並特別強調與企業技術合作與實務結合。這種「即學即用」、理論與產業實務緊密結合的教學理念，是我決定投身該所的重要原因之一。該所不僅與元太科技工業股份有限公司（E Ink）、瑞昱半導體股份有限公司、麗臺科技等共 12 家技術領先企業攜手合作，打造元宇宙實驗室、智慧顯示與影像製造平台等專案，讓學生有機會參與產學合作與實習計畫，同時還提供碩士生高額獎學金機會、實習保障、就業支持等優勢立報傳媒。作為一個對 AI 領域充滿熱情但尚缺乏實務接觸機會的學生，我發現這樣的跨域培育模式正好符合我的求學需求。於是，我毅然決定以該所為目標申請研究生。經過激烈的甄選過程（碩士班錄取率不到兩成、新生註冊率達 100%），我最終如願錄取進入 AI 跨域應用研究所，開始展開結合理論與產業、融合智慧顯示、AIoT 與電腦視覺等多元領域的研究生涯。

進入 AI 跨域應用研究所後，我在高文忠教授的指導下，從一位僅具實作能力的學生，逐步蛻變為具備學術洞察力的研究者。高教授專注於系統晶片設計、軟性顯示器系統及嵌入式影像處理技術，在他的帶領下，我參與了多項研發計畫，實際投入晶片原型開發與影像演算法整合等工作。在這樣強調學術與應用融合的研究環境中，我學會了如何系統性地閱讀前沿論文、設計與執行實驗、撰寫研究報告，並進行跨領域技術的整合與分析。透過這些訓練與實務經驗，我深入理解了嵌入式系統與 AI 技術的整合挑戰與潛力，也逐漸建立起對研究脈絡的掌握能力。這段歷程讓我從單純的程式執行者，成長為能夠獨立思考、分析問題並提出創新解方的學術型人才。