

DT
448.6
1021
100
225280



國立勤益科技大學
電子工程系研究所

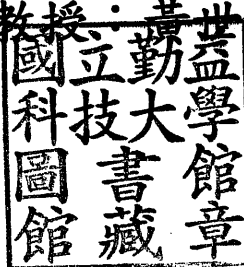
碩士論文

應用 FPGA 實現 CFA 適應性權值色彩插補演算法

Implementation of CFA Adaptive Weighted Color
Interpolation Algorithm Using FPGA

研究生：王保發

指導教授：黃世演



中華民國一〇〇年七月

國立勤益科技大學圖書館



225280

國立勤益科技大學

博碩士論文全文上網授權書

(提供授權人裝訂於紙本論文書名頁之次頁用)

本授權書所授權之論文為授權人在國立勤益科技大學
電子工程系 資訊科技 組 99 學年度第 二 學期取得碩
士學位之論文。

論文題目：應用FPGA實現CFA適應性權值色彩插補演算法
指導教授：黃世演

■ 同意

本人具有著作權之論文全文資料，非專屬、無償授予本人畢業學校圖書館，不限地域、時間與次數，以微縮、光碟或數位化等各種方式重製與利用，提供讀者基於著作權法合理使用範圍內之線上檢索、閱覽、下載及列印。

論文全文上載網路公開之範圍及時間：

校內區域網路	<input checked="" type="checkbox"/> 立即公開
校外網際網路	<input checked="" type="checkbox"/> 立即公開

授權人：王保發

簽名：王保發

中華民國 100 年 8 月 1 日

國立勤益科技大學
研究所碩士班
論文口試委員會審定書

本校 電子工程系 碩士班 王保發 君

所提論文 應用 FPGA 實現 CFA 適應性權值色彩插補演算法

合於碩士資格水準，業經本委員會評審認可。

論文口試委員會：

召集人：黃啓瑞

委員：李煥生

黃世濱

指導教授：黃世濱

所長：林能傑
電子工程系主任

中華民國一〇〇年七月十五日

應用 FPGA 實現 CFA 適應性權值色彩插補演算法

研究生： 王保發

指導教授： 黃世演

國立勤益科技大學

電子工程系研究所

摘要

目前消費型數位相機為單一光感測器架構，透過貝爾格式(Bayer pattern)的色彩濾波器陣列(color filter array, CFA)取樣紅、綠、藍三種顏色(簡稱三原色)排列的像素陣列。光感測器上每一個像素位置只取樣一種色彩，這些色彩像素的排列樣式如同馬賽克一般。

解馬賽克演算法(demosaicing)，或稱色彩插補 (color interpolation)，此方法用來估測未被取樣的另外兩種色彩，來完成全彩影像的重建。由於 CFA 取樣色彩中以綠原色最多，因此優先重建像素的綠色原色，可以獲得良好的影像重建品質。而在重建像素的紅、藍色原色時，典型的作法是利用基於高光譜相關性的方法來估測像素色彩。

為了提升重建品質，常見的作法是利用邊跡偵測來選擇插補公式。例如，自適性次頻帶插補法(adaptive sub-band interpolation, ASCI)使用

二維離散小波轉換，偵測水平及垂直邊跡，根據這些邊跡梯度的強弱程度建立複數個閾值，藉以對應到合適的加權值。因此，ASCI 演算法可以獲得良好的影像重建品質。

然而在硬體設計上，頻域轉換運算需要花費大量的計算成本。為了降低硬體資源，本文採用以較低計算量的方法，取代 ASCI 演算法的頻域轉換運算，並且減少閾值數目來減輕運算負擔。經實驗顯示，已達到與自適性次頻帶插補法品質相近的效果。本文使用現場可規劃邏輯閘陣列(field programmable gate array, FPGA)設計一個全管線化 CFA 解馬賽克硬體模組，以串流 CFA 影像作為輸入，能夠在每一個時脈週期輸出 RGB 三原色的像素。經由模擬測試，本文所提出的色彩插補演算法之峰值信噪比(peak signal-to-noise ratio, PSNR)值為 40.72dB，相較於 ASCI 演算法之 PSNR 值 40.87dB 只相差 0.15dB。本文 CFA 解馬賽克硬體模組經模擬測試，其 PSNR 值可達 40.5dB，證實本研究方法確實達到相近於 ASCI 演算法的成像品質。

關鍵字：色彩濾波器陣列、解馬賽克、色彩插補、現場可規劃邏輯閘陣列

Implementation of CFA Adaptive Weighted Color Interpolation Algorithm Using FPGA

Student: Bao-Fa Wang

Advisor: Shih-Yen Huang

Institute of Electronic Engineering

National Chin-Yi University of Technology

Abstract

Modern digital cameras use a single-chip sensor structure. Sensors use the Bayer pattern color filter array (CFA) to sample red, green, and blue color (named these three color as elementary color) arranged in an array, and each pixel position is only sampling one color. Since each pixel lacks two elementary colors, these pixels make up an image with mosaic format.

Demosaicing, also known as color interpolation, the method estimate the other two missing elementary colors for reconstruct a full-color image. The green elementary color are the most in CFA, therefore, first to reconstruct the missing green elementary color can obtain a good reconstructed quality. In addition, to reconstruct the red or blue elementary color, typically use high spectral correlation-based method.

To improve quality, the traditional method use various interpolation method according edge detect. For example, the adaptive sub-band interpolation (ASCI) applied two-dimensional discrete wavelet transform to detect horizontal and vertical edge, according these edges gradient to select threshold and associate weight.

However, in hardware design, ASCI take a lot of hardware resource. To reduce hardware computing resources, this thesis implemented a lower computational method to detect edge and selected fewer numbers of thresholds. Additionally, design a full pipelined hardware module for CFA demosaicing using field programmable gate array (FPGA). This system, video signal is a streaming data, each clock cycle can output one reconstructed pixel. According a static timing analysis, the maximum clock rate is 104.778MHz. The software simulation results showed that the peak signal-to-noise ratio (PSNR) value is 40.72dB, compared to ASCI's PSNR value 40.87dB is only 0.15dB less. The functional simulation testing for this hardware system, the CFA demosaic module's PSNR value is 40.5dB, only 0.22dB less than the software simulation.

Keywords : Color filter array (CFA), demosaicing, color interpolation, field programmable gate array (FPGA)

誌謝

本論文能夠順利完成，首先要感謝我的指導教授黃世演老師。感謝老師這幾年的悉心教導，不論是在課業研究上的督促與指導，或是在處事態度上的教誨，都使學生受益良多。

也要感謝口試委員黃啟瑞教授以及林灶生教授對本論文的指導與建議，讓本論文能夠更加完整。

另外，還要感謝實驗室學長小白、智遠、小武以及士權的照顧，以及感謝實驗室同學崇志、國志、沛嘉、相垣、任民、育庭、柏君以及郁升在這段日子的陪伴與指教。

最後，我要感謝我的家人，特別是我的父母，對我的關懷與支持，讓我在無後顧之憂的環境下完成學業。

謝謝所有曾經幫助過我的人。謝謝你們!

王保發

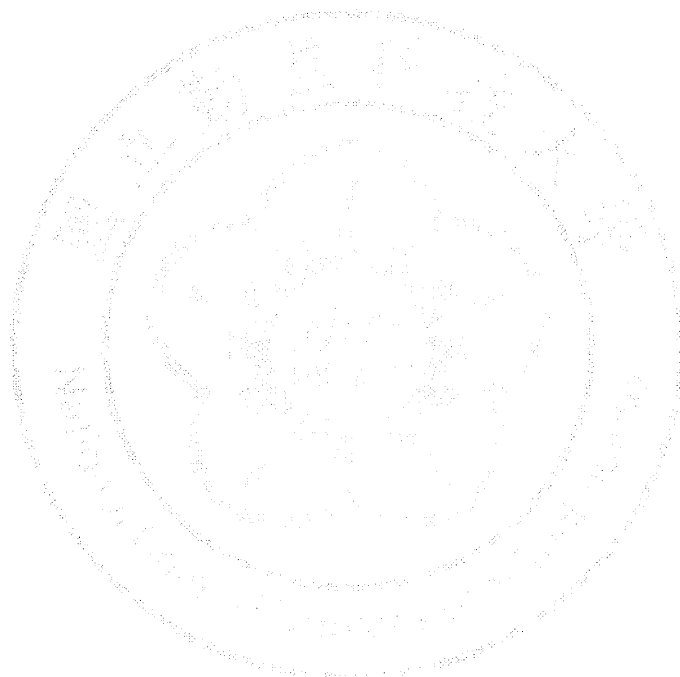
2011年7月

目錄

摘要.....	i
英文摘要.....	iii
誌謝.....	v
目錄.....	vi
圖目錄.....	ix
表目錄.....	xii
第一章 序論.....	1
1.1 研究背景.....	1
1.1.1 數位相機的架構.....	1
1.1.2 色彩濾波器陣列.....	2
1.1.3 CFA 解馬賽克.....	4
1.2 研究動機與目的.....	7
1.3 本文架構.....	8
第二章 相關色彩插補演算法之回顧.....	9
2.1 非適應性插補法.....	9

2.1.1 雙線性插補法.....	9
2.1.2 色彩差異插補法.....	10
2.2 適應性插補法.....	13
2.2.1 二階修正線性插補法.....	13
2.2.2 有效疊代加權邊跡與色彩差異插補法.....	16
2.2.3 中值濾波插補法.....	20
2.2.4 適應性次頻帶關聯插補法.....	22
第三章 CFA 解馬賽克演算法及硬體架構.....	27
3.1 本文提出的適應性權值彩色插補演算法.....	27
3.2 適應性權值解馬賽克演算法之硬體架構設計.....	30
3.2.1 綠色重建模組.....	31
3.2.2 紅/藍色重建模組.....	42
3.2.3 綠色優化模組.....	44
第四章 FPGA 實現與實驗結果.....	49
4.1 開發環境.....	49
4.2 設計流程.....	49
4.3 功能模擬.....	50

4.4 FPGA 實現結果.....	54
4.5 PSNR 比較.....	56
第五章 結論與未來展望.....	64
參考文獻.....	66



圖目錄

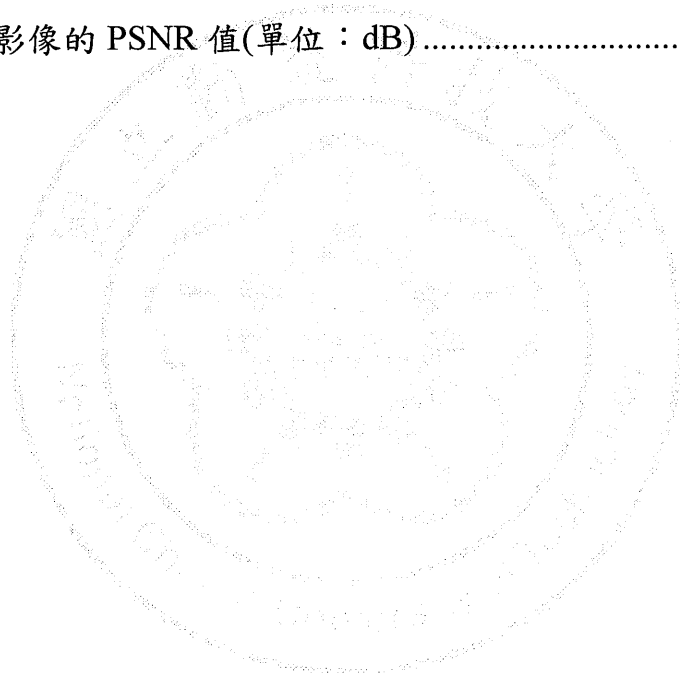
圖 1.1 三組光感測器之系統架構.....	2
圖 1.2 單一光感測器之系統架構.....	2
圖 1.3 貝爾格式.....	3
圖 1.4 經由色彩濾波器陣列取樣的貝爾影像.....	4
圖 1.5 重建後的全彩影像(雙線性插補).....	5
圖 1.6 拉鏈效應的例子.....	6
圖 1.7 假色現象的例子.....	6
圖 2.1 雙線性插補法參照圖.....	10
圖 2.2 色彩平面之間的低差異性.....	11
圖 2.3 色彩差異插補法參照圖.....	11
圖 2.4 二階修正線性插補法參照圖.....	14
圖 2.5 有效疊代加權邊跡與色彩差異插補法流程.....	17
圖 2.6 有效疊代加權邊跡與色彩差異插補法參照圖.....	18
圖 2.7 3x3 視窗範圍中值濾波器插補參照圖.....	21
圖 2.8 適應性次頻帶關聯插補法流程.....	23

圖 2.9 適應性次頻帶關聯插補法參照圖.....	24
圖 3.1 適應性權值彩色插補演算法流程圖.....	28
圖 3.2 較佳 PSNR 值下的 $GDSR$ 與 W 對應關係.....	29
圖 3.3 CFA 解馬賽克架構方塊圖 (本研究色彩插補法).....	31
圖 3.4 綠色重建模組方塊圖.....	32
圖 3.5 二維影像的視窗濾波器運算.....	33
圖 3.6 線緩衝器架構 (3×3 區域).....	34
圖 3.7 線緩衝器架構的運作範例(3×3 區域).....	34
圖 3.8 二維濾波器視窗模組電路方塊圖(3×3 區域).....	35
圖 3.9 串流影像視窗運算後的時序關係.....	37
圖 3.10 影像控制信號模式選擇(TRDB-D5M 影像感測器).....	38
圖 3.11 時序延遲模組(延遲 2 列 2 像素).....	39
圖 3.12 時序延遲(延遲 2 列 2 像素).....	39
圖 3.13 綠色插補模組方塊圖.....	41
圖 3.14 貝爾格式 RGB 之間的位置對應關係.....	41
圖 3.15 紅/藍色重建模組方塊圖.....	42
圖 3.16 紅/藍色插補模組方塊圖.....	43

圖 3.17 紅/藍色的色彩差異插補單元方塊圖.....	44
圖 3.18 綠色優化模組方塊圖.....	45
圖 3.19 綠色更新模組方塊圖.....	46
圖 3.20 3×3 快速中值濾波器架構方塊圖.....	47
圖 3.21 三輸入的排序與極值的架構方塊圖.....	48
圖 4.1 CFA 解馬賽克模組設計流程圖.....	50
圖 4.2 模擬二維濾波器視窗模組(3×3)輸出波形圖.....	51
圖 4.3 模擬二維濾波器視窗模組(3×3)輸出波形圖(局部).....	52
圖 4.4 模擬二維濾波器視窗模組(5×5)輸出波形圖(局部).....	52
圖 4.5 模擬 CFA 解馬賽克測試程序流程圖.....	53
圖 4.6 FPGA 實現 CFA 色彩插補演算法電路的開發環境.....	55
圖 4.7 24 張 Kodak 測試影像.....	58
圖 4.8 各種解馬賽克方法的成像效果(Image5).....	62
圖 4.9 各種解馬賽克方法的成像效果(Image8).....	63

表目錄

表 2.1 ASCI 演算法中的 $GDSR$ 與 W 對應多閾值表.....	25
表 3.1 本研究色彩插補法中的 $GDSR$ 與 W 對應多閾值表.....	29
表 4.1 解馬賽克模組的硬體資源利用度報告.....	54
表 4.2 重建影像的 PSNR 值(單位：dB).....	59



第一章 序論

1.1 研究背景

1.1.1 數位相機的架構

數位相機(DC, digital camera)內部的光感測器(photo sensors)晶片能夠感測光源強度，卻無法直接分辨光源的色彩(即為波長)。若在感測元件外加一層色彩濾波器(color filter)，可得僅一特定波長範圍的光源才能通過此濾波器而抵達感測元件。圖 1.1 為三組光感測器架構的數位相機，利用分光鏡(beam-splitters)將光源分送至各個光感測器，每像素同時利用紅(R)、綠(G)及藍(B)三組色彩濾波器及光感測器，組成全彩(full-color)影像。然而一般消費型數位相機為了降低製造成本與設計複雜度，進而發展出僅使用一個色彩濾波器陣列及單一光感測器架構的數位相機，如圖 1.2 所示。此種架構不需要使用到分光鏡系統以及高達三組感光晶片的成本，但是需經由影像處理程序才可以獲得全彩影像。

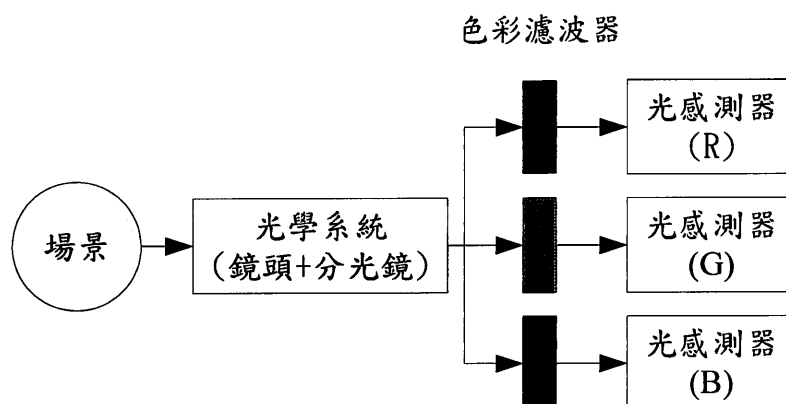


圖 1.1 三組光感測器之系統架構

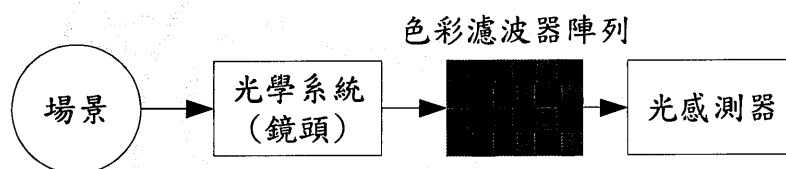


圖 1.2 單一光感測器之系統架構

1.1.2 色彩濾波器陣列

由於像素的色彩是由紅(R)、綠(G)及藍(B)三色組成，稱為三原色。前述圖 1.2 之色彩濾波器陣列(color filter array, CFA)是由不同的原色濾波器以像素為單位，有規則地排列組成的混合型色彩濾波器，因此 CFA 影像中的每一個像素都有相對應的色彩強度。在各種色彩濾波器陣列

排列格式中，目前最為廣泛使用的是貝爾格式(Bayer pattern)[1]，本文即採用此格式，如圖 1.3 所示，貝發格式是以 RGB 三種色彩作為取樣，綠色取樣在整個陣列中所佔比例為紅色及藍色的兩倍，即綠色取樣有 50%，紅、藍色各佔有 25%。圖 1.4(a)為一幅原始影像或場景，透過貝爾色彩濾波器陣列取樣的貝爾影像，如圖 1.4(b)(c)所示。由於色彩濾波器陣列如同棋盤般的排列方式，又稱為色彩濾波器馬賽克(color filter mosaic, CFM)。

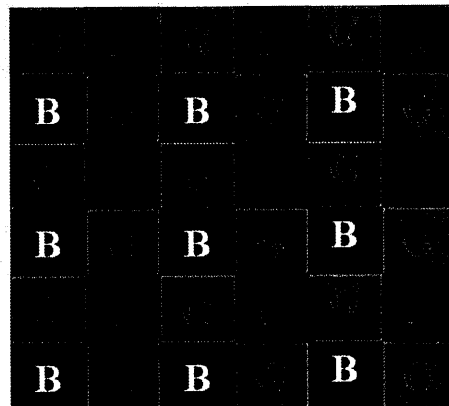
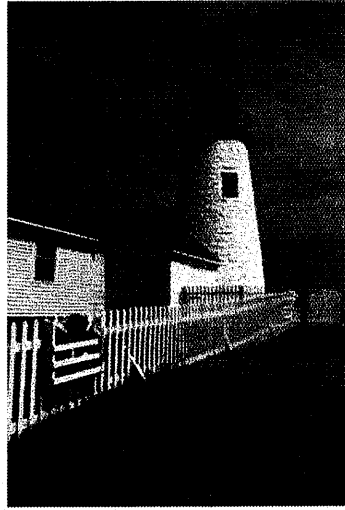


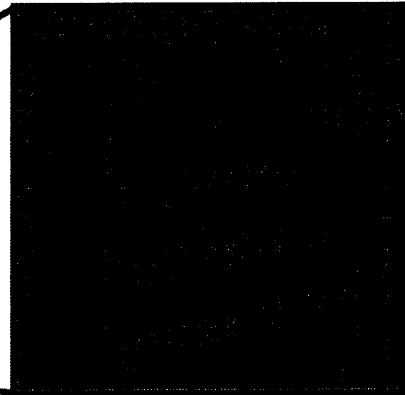
圖 1.3 貝爾格式



(a) 原始影像



(b) 貝爾影像



(c) 局部放大顯示

圖 1.4 經由色彩濾波器陣列取樣的貝爾影像

1.1.3 CFA 解馬賽克

CFA 解馬賽克(demosaicing)是利用色彩插補(color interpolation)方法[2]，重建貝爾影像的每一個像素位置上，另外兩個沒被取樣到的原色值。通常是利用鄰近像素之間的色彩關聯性，估測消失的原色來重建全彩影像。圖 1.5 為一貝爾影像解馬賽克後的全彩影像，它是採用一種簡單且快速的雙線性插補演算法[2]，但影像重建後的某些位置可以發現顏色及紋理失真。通常貝爾影像經由插補重建後可能會產生兩種失真情況：假色(false color)現象以及拉鏈效應(zipper effect)，如圖 1.6 及圖 1.7 所示。因此說明了 CFA 解馬賽克所採用的插補演算法會影響影像的重建品質。

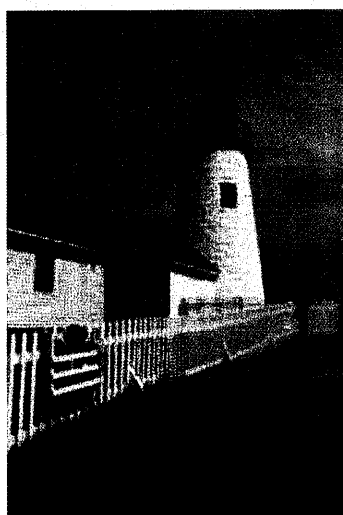
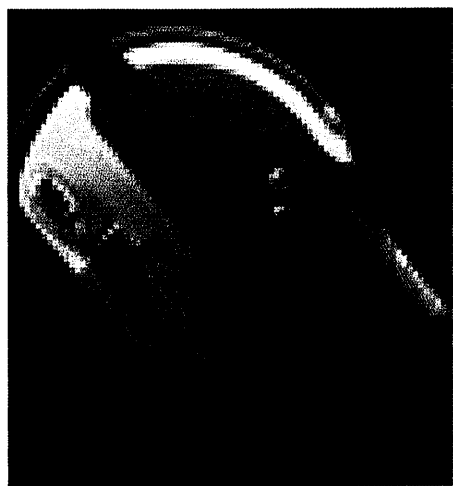


圖 1.5 重建後的全彩影像(雙線性插補)

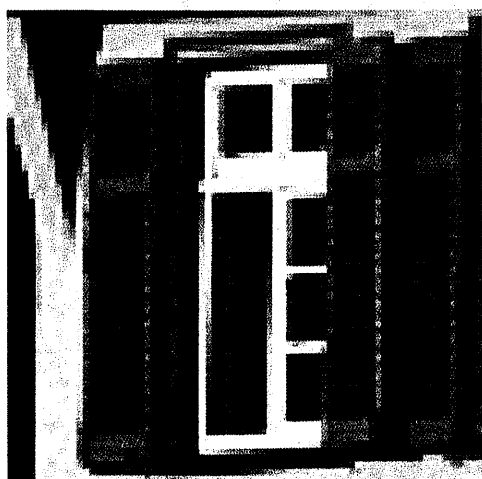


(a) 原始影像

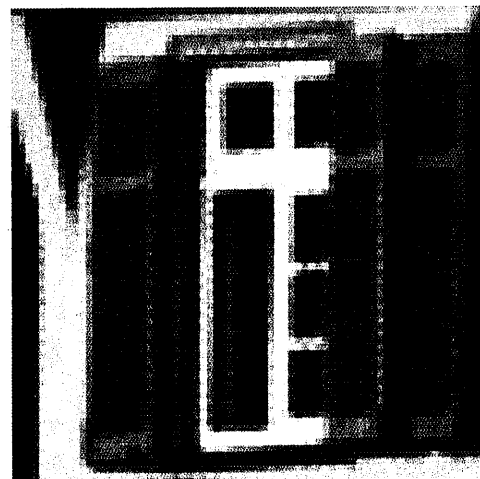


(b) 帶有拉鏈效應的重建影像

圖 1.6 拉鏈效應的例子



(a) 原始影像



(b) 帶有假色現象的重建影像

圖 1.7 假色現象的例子

1.2 研究動機與目的

數位相機內部的影像感測器是以像素串流形式傳送至影像處理晶片，一個輸出成像品質良好的 CFA 解馬賽克方法，其內部主要的影像處理核心必需具備即時處理的能力。傳統的數位信號處理器(DSP, digital signal processor)在演算法及程序規劃上雖具有高彈性化的優點，但是在影像串流資料的處理效能並不理想。輸入影像經過一連串的影像管線(image pipeline)得到的輸出成像，所花費的時間大都是利用像素資料來回存取頁面緩衝區(frame buffer)的時間。若所耗費的時間過長，會導致影像輸出效率低。而現場可規劃邏輯閘陣列(Field Programmable Gate Array, FPGA) 具有硬體平行處理機制的特點，有別於循序執行的處理運算，各項處理運算可以在同一時脈週期內獨立完成。對於具有大量資料流以及即時處理需要的影像管線任務，可作為一種解決方案。針對 CFA 串流影像即時轉換成品質良好的 RGB 串流影像為目標，本文提出一個能夠全管線化(full pipeline)的 CFA 解馬賽克硬體架構，並且利用 FPGA 實現，達到及時輸出彩色影像之功效。

1.3 本文架構

第一章為敘述色彩濾波器解馬賽克的相關背景及本文的研究動機與目的。第二章將探討最為常見的典型演算法。第三章為本文所提出的 CFA 解馬賽克演算法及其全管線化硬體架構。第四章為 FPGA 實現與實驗結果，包含硬體架構的功能模擬，並與其他演算法比較其效能。最後一章為研究的結論並建議未來研究的方向。

第二章 相關色彩插補演算法之回顧

色彩插補演算法可粗略分成兩種類型：一種為非適應性(non-adaptive)插補法，另外一種為適應性(adaptive)插補法。這兩種類型的演算法各自擁有其優勢及缺點。下述各節介紹本文相關的研究方法的色彩插補演算法。

2.1 非適應性插補法

非適應性插補法是使用固定的參數運算來插補每個像素所遺失的原色，因此適合處理影像平滑的區域。常見的非適應性插補法有雙線性插補法 [2]、色彩差異插補法 [3]等演算法。

2.1.1 雙線性插補法

雙線性插補法[2]是利用鄰近範圍內對已知相同色彩像素取平均來估測遺失的像素原色值。如圖 2.1 所示，以重建像素 6 及像素 7 之原色為例。插補位於像素 6 的綠原色 G_6 及紅原色 R_6 ：

$$G_6 = \frac{G_2 + G_5 + G_7 + G_{10}}{4} \quad (2-1)$$

$$R_6 = \frac{R_1 + R_3 + R_9 + R_{11}}{4} \quad (2-2)$$

插補位於像素 7 的紅原色 R_7 及藍原色 B_7 ：

$$R_7 = \frac{R_3 + R_{11}}{2} \quad (2-3)$$

$$B_7 = \frac{B_6 + B_8}{2} \quad (2-4)$$

R_1	G_2	R_3	G_4
G_5	B_6	G_7	B_8
R_9	G_{10}	R_{11}	G_{12}

圖 2.1 雙線性插補法參照圖

2.1.2 色彩差異插補法

在已發展的解馬賽克方法中，有許多演算法[3]-[7]都說明一般自然影像中 RGB 色彩平面之間存在著高光譜相關性(spectral correlation)，

演算法[3]-[7]即基於此特性解馬賽克如圖 2.2 所示。

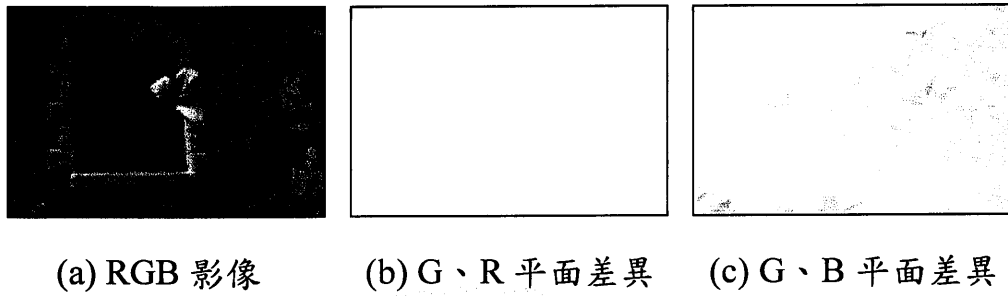


圖 2.2 色彩平面之間的低差異性

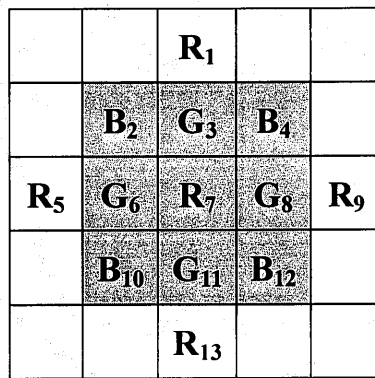


圖 2.3 色彩差異插補法參照圖

色彩差異插補法[3]利用在影像每一個局部區域內，其 RGB 色彩平面之間存在一個近似於常數的差異值假設，而定義了 K_B (G-B)及 K_R (G-R)兩個色彩差異平面：

$$K_B = G - B \quad (2-5)$$

$$K_R = G - R \quad (2-6)$$

並利用此兩色彩差異平面之關係配合雙線性插補來重建影像。演算法以重建綠色平面為優先，隨後重建紅/藍色平面。

如圖 2.3 所示，重建綠原色於紅/藍原色的像素位置，以插補綠原色 G_7 位於像素 7 為例：

首先，分別計算 G_3 、 G_6 、 G_8 及 G_{11} 位置之 K_R 值，以計算 K_{R3} 為例，

K_{R3} 值為 G_3 減去鄰近像素 R 之平均：

$$K_{R3} = G_3 - \frac{R_1 + R_7}{2} \quad (2-7)$$

之後，利用 K_{R3} 、 K_{R6} 、 K_{R8} 及 K_{R11} 之平均近似於 K_{R7} 之假設，插補像素 7 的綠原色 G_7 ：

$$G_7 = R_7 + \frac{K_{R3} + K_{R6} + K_{R8} + K_{R11}}{4} \quad (2-8)$$

以插補紅原色 R_3 位於像素 3 為例：

$$R_3 = G_3 - \frac{K_{R1} + K_{R7}}{2} \quad (2-9)$$

以插補藍原色 B_7 位於像素為例：

$$B_7 = G_7 - \frac{K_{B2} + K_{B4} + K_{B10} + K_{B12}}{4} \quad (2-10)$$

2.2 適應性插補法

適應性插補法是根據不同的情況下經由特定的特徵分析而採取適合的運算方式以求得較佳的結果。例如：二階修正線性插補法[4]、有效疊代加權邊跡與色彩差異插補法[5]、中值濾波插補法[6]及適應性次頻帶關聯插補法[7]。

2.2.1 二階修正線性插補法

二階修正線性插補法[4]具有邊跡偵測的優點，能夠針對水平及垂直邊跡的重建提供較佳的插補運算。演算法也是先重建綠色平面，隨後再重建紅、藍色平面。

		R ₁		
	B ₂	G ₃	B ₄	
R ₅	G ₆	R ₇	G ₈	R ₉
	B ₁₀	G ₁₁	B ₁₂	
		R ₁₃		

圖 2.4 二階修正線性插補法參照圖

如圖 2.4 所示，以插補綠原色 G₇ 位於像素 7 為例，首先定義水平梯度 ΔH 及垂直梯度 ΔV ：

$$\Delta H = |G_6 - G_8| + |2R_7 - R_5 - R_9| \quad (2-11)$$

$$\Delta V = |G_3 - G_{11}| + |2R_7 - R_1 - R_{13}| \quad (2-12)$$

再依據梯度的強弱來決定插補方式：

$$G_7 = \begin{cases} G_7^V & \text{if } \Delta V < \Delta H \\ G_7^H & \text{if } \Delta V > \Delta H \\ \frac{G_7^H + G_7^V}{2} & \text{if } \Delta H = \Delta V \end{cases} \quad (2-13)$$

其中，水平插補 G_7^H 與垂直插補 G_7^V 定義如下：

$$G_7^H = \frac{G_6 + G_8}{2} + \frac{2R_7 - R_5 - R_9}{4} \quad (2-14)$$

$$G_7^V = \frac{G_3 + G_{11}}{2} + \frac{2R_7 - R_1 - R_{13}}{4} \quad (2-15)$$

完成所有綠原色的重建後，再進行重建紅、藍原色。以插補藍原色 B_7 位於像素 7 為例，為了決定左、右對角的插補方向，首先定義左斜梯度 ΔL 及右斜梯度 ΔR ：

$$\Delta L = |B_2 - B_{12}| + |2G_7 - G_2 - G_{12}| \quad (2-16)$$

$$\Delta R = |B_4 - B_{10}| + |2G_7 - G_4 - G_{10}| \quad (2-17)$$

再依據梯度的強弱來決定插補方式：

$$B_7 = \begin{cases} B_7^L & \text{if } \Delta L < \Delta R \\ B_7^R & \text{if } \Delta R > \Delta L \\ \frac{B_7^L + B_7^R}{2} & \text{if } \Delta L = \Delta R \end{cases} \quad (2-18)$$

其中，

$$B_7^L = \frac{B_2 + B_{12}}{2} + \frac{2G_7 - G_2 - G_{12}}{4} \quad (2-19)$$

$$B_7^R = \frac{B_4 + B_{10}}{2} + \frac{2G_7 - G_4 - G_{10}}{4} \quad (2-20)$$

以下分別以插補紅原色 R_3 及 R_6 位於像素 3 及 6 為例：

$$R_3 = \frac{R_1 + R_7}{2} + \frac{2G_3 - G_1 - G_7}{4} \quad (2-21)$$

$$R_6 = \frac{R_5 + R_7}{2} + \frac{2G_6 - G_5 - G_7}{4} \quad (2-22)$$

2.2.2 有效疊代加權邊跡與色彩差異插補法

有效疊代加權邊跡與色彩差異插補法[5]為疊代型演算法，利用疊代運算重覆更新像素直到滿足限定條件。插補程序主要分成三個階段，分別為初始化階段、優化階段及疊代階段。演算法流程圖如圖 2.5 所示。

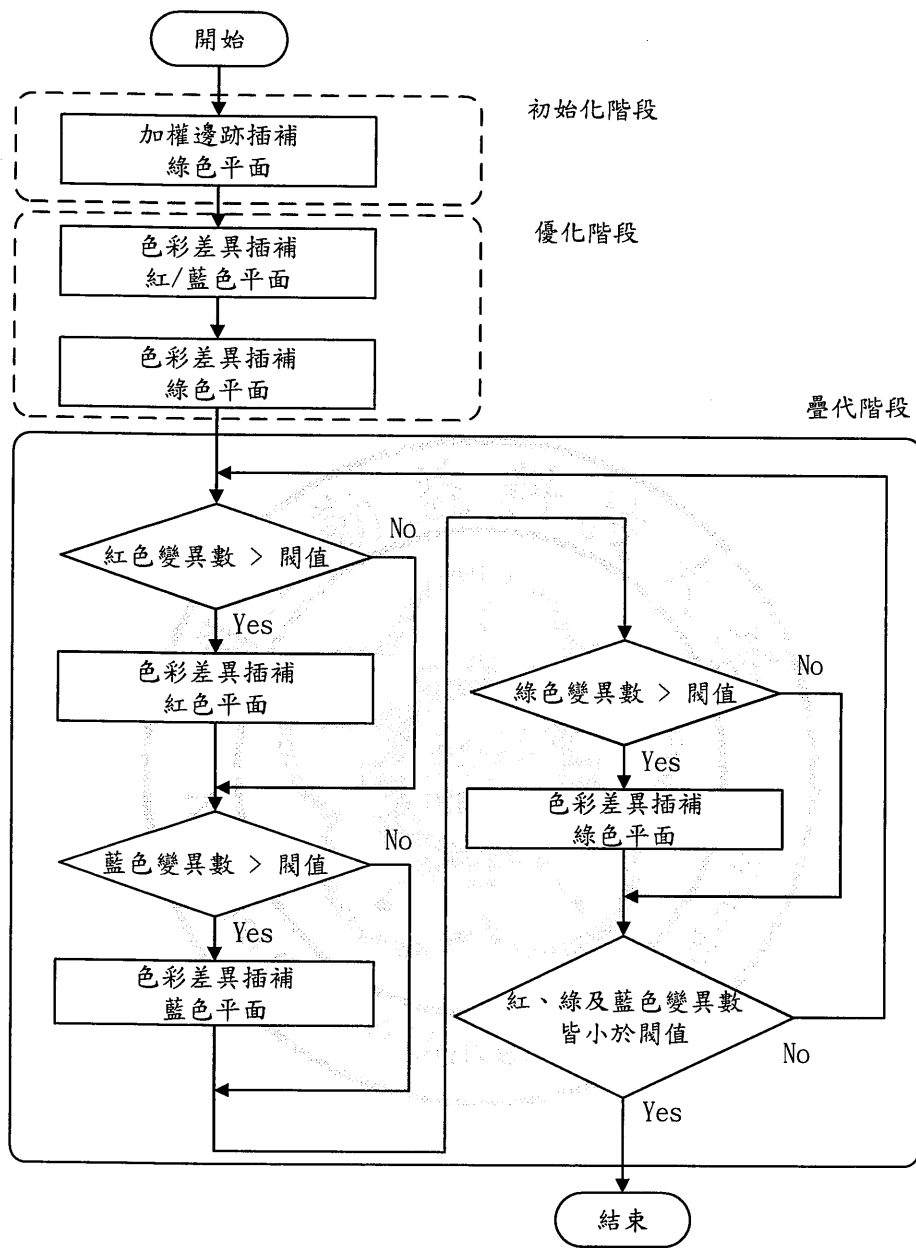


圖 2.5 有效疊代加權邊跡與色彩差異插補法流程

首先在初始化階段是針對綠色平面的重建。如圖 2.6 所示，以插補

G_7 位於 R_7 為例：

		R_1		
	B_2	G_3	B_4	
R_5	G_6	R_7	G_8	R_9
	B_{10}	G_{11}	B_{12}	
		R_{13}		

圖 2.6 有效疊代加權邊跡與色彩差異插補法參照圖

綠色平面的重建同於二階修正線性插補法，先定義水平梯度 ΔH 及垂直梯度 ΔV 同於(2-11)式及(2-12)式所示，再依據梯度的強弱來決定插補方式：

$$G_7 = \begin{cases} G_7^V W_1 + G_7^H W_2 & \text{if } \Delta V < \Delta H \\ G_7^H W_1 + G_7^V W_2 & \text{if } \Delta V > \Delta H \\ \frac{G_7^H + G_7^V}{2} & \text{if } \Delta H = \Delta V \end{cases} \quad (2-23)$$

上式的水平插補值 G_7^H 及垂直插補值 G_7^V 同於(2-14)式及(2-15)式所

示。

插補遺失的綠色像素是由水平及垂直插補值各別給予不同加權所組成，其中 W_1 與 W_2 的關係為 $W_1 + W_2 = 1$ 。 W_1 與 W_2 經由大量的測試影像計算出在 $W_1 = 0.87$ 及 $W_2 = 0.13$ [5]，藉以找到 RGB 平面較小平均的均方差(mean square error, MSE)，如(2-24)式所示。並且建議設定 $W_1 = 7/8$ (即前述 $W_1 = 0.87$ 的近似值)及 $W_2 = 1/8$ 可以獲得較佳的效果。由此可知，當水平梯度 ΔH 小於垂直梯度 ΔV 時，插補遺失的綠色像素是由水平插補獲得較高的加權而垂直插補則為較低加權，反之亦然。只有在水平與垂直梯度相等時，給予相同的權值。

$$(W_1, W_2) = \arg \min_{W_1+W_2} (MSE(R) + MSE(G) + MSE(B)) \quad (2-24)$$

優化階段是採用與色彩差異插補法相同的方法重建紅、藍兩原色，如(2-9)(2-10)式，並且再一次更新綠原色的插補值，如(2-7)(2-8)式。

在疊代階段，紅、綠及藍原色的更新皆採用與色彩差異插補法相同的方法，如(2-7)(2-8)(2-9)(2-10)式。疊代終止條件為紅、綠、藍的變異數(variance)皆低於閾值，以計算紅色變異數 $Var(R)$ 為例，其定義如下：

$$Diff(i) = R_{n+1}(i) - R_n(i) \quad (2-25)$$

$$m_R = \frac{1}{framesize} \sum_{i=1}^{framesize} (Diff(i)) \quad (2-26)$$

$$Var(R) = \frac{1}{framesize} \sum_{i=1}^{framesize} (Diff(i) - m_R)^2 \quad (2-27)$$

其中， $Diff(i)$ 為紅色平面於第 $n+1$ 次及第 n 次疊代時的第 i 個像素差值，而 $framesize$ 為影像的大小。 m_R 為前後兩個紅色平面的差值平均。在文獻[5]中，閾值設定為 1，當紅、綠、藍的變異數皆低於閾值時則完成演算法。

2.2.3 中值濾波插補法

Freeman[8]提出對 RGB 每一個色彩平面進行中值濾波運算插補失去的像素。而 Lu[6]根據色彩平面之間含有高光譜相關的特點，提出一種在初始插補 RGB 運算後執行在 $K_B(G-B)$ 及 $K_R(G-R)$ 平面上的中值濾波插補法。此方法可以有效地減少色彩重建時視覺上失真現象以及提升影像重建的品質。Lu[6]演算法定義如下所示：

$$G'_c = \frac{(R_c + med(G - R) + B_c + med(G - B))}{2} \quad (2-28)$$

$$R'_c = G'_c + med(R - G) \quad (2-29)$$

$$B'_c = G'_c + med(B - G) \quad (2-30)$$

其中， R 、 G 及 B 分別代表紅色、綠色及藍色平面。 $med(*)$ 代表中值濾波運算子，以 3×3 視窗範圍的中值濾波器運算為例，排序視窗內 9 個元素後並取得第 5 順位的元素為中值輸出。如圖 2.7 所示， R_c 、 B_c 分別代表以紅色、藍色為中心點像素， G'_c 代表中心點的綠色二次插補值。文獻[6]建議此方法安排在初始插補階段之後執行，作為影像重建的後處理(post-processing)程序，能夠有效的抑制解馬賽克的假色現象及拉鏈效應。

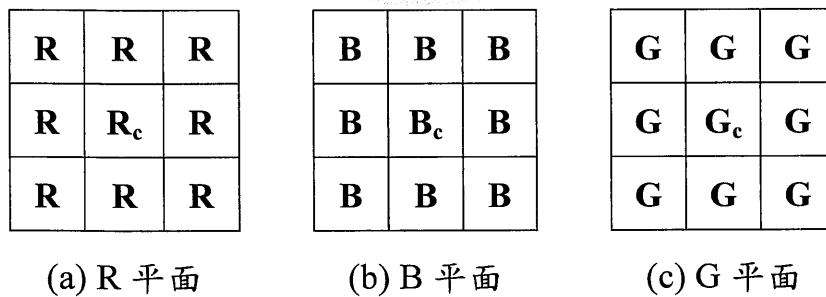


圖 2.7 3×3 視窗範圍中值濾波器插補參照圖

2.2.4 適應性次頻帶關聯插補法

適應性次頻帶關聯插補法 [7] 是利用二維離散小波轉換 (two dimensional discrete wavelet transform, 2D-DT) 將貝爾影像轉換成 4 個次頻帶 (LL、LH、HL 及 HH)，次頻帶 LH 及 HL 分別保留影像的水平及垂直邊跡特徵。在重建綠色平面時利用次頻帶 LH 及 HL 作為影像的水平梯度與垂直梯度的估測項並且根據水平與垂直梯度之間強弱關係，對水平與垂直插補給予不同的加權。其演算法流程如圖 2.8 所示。

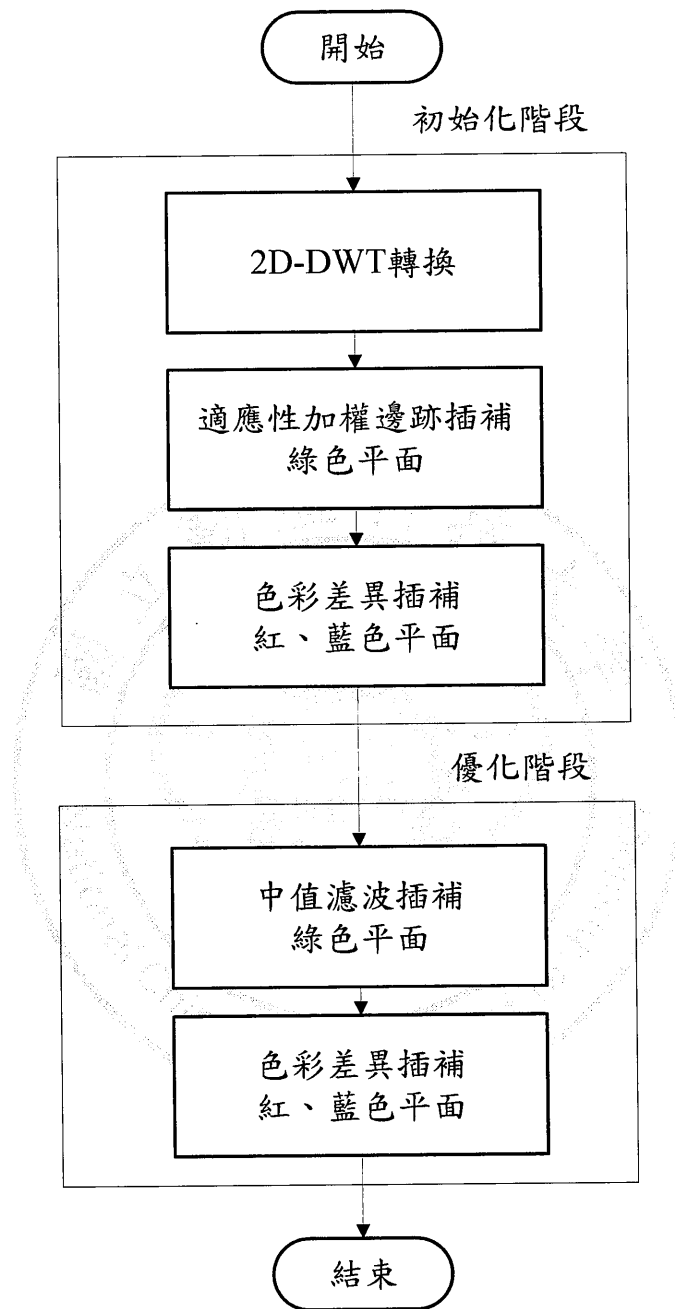


圖 2.8 適應性次頻帶關聯插補法流程

初始階段重建綠色平面，如圖 2.9 所示，以插補綠原色 G_7 位於像

素 7 為例：

		R₁		
	B₂	G₃	B₄	
R₅	G₆	R₇	G₈	R₉
	B₁₀	G₁₁	B₁₂	
		R₁₃		

圖 2.9 適應性次頻帶關聯插補法參照圖

首先定義垂直梯度 GV 與水平梯度 GH ，

$$GV_7 = |LH_3| + |LH_7| + |LH_{11}| \quad (2-31)$$

$$GH_7 = |HL_6| + |HL_7| + |HL_8| \quad (2-32)$$

其中， LH_7 與 HL_7 為圖 2.9 的第 7 像素的垂直與水平邊跡特徵值。

並且定義「梯度差」 GD 、「梯度和」 GS 與「梯度比」 $GDSR$

$$GD_7 = |GH_7 - GV_7| \quad (2-33)$$

$$GS_7 = GH_7 + GV_7 \quad (2-34)$$

$$GDSR_7 = \frac{GD_7}{GS_7} \quad (2-35)$$

其中， $GDSR$ 代表垂直、水平之間邊跡強弱之間的差異度。當 $GDSR$ 值越大表示兩方向之間的梯度差異越大。反之， $GDSR$ 值越小表示兩方向之間的梯度差異越相似。根據此概念，建立 $GDSR$ 與加權值 W 的關係，如表 2.1 所示。

表 2.1 ASCI 演算法中的 $GDSR$ 與 W 對應多閾值表

$GDSR$	≥ 0.75	$0.75 \sim 0.5$	$0.5 \sim 0.25$	$0.25 \sim 0.2$	< 0.2
W	1	0.9	0.8	0.7	0.6

綠色插補方法採用與有效疊代加權邊跡與色彩差異插補法相同方式，唯一不同的地方僅加權值 W 的大小則由 $GDSR$ 決定，如表 2.1。

$$G_7 = \begin{cases} G_7^V W + G_7^H (1-W), & \text{if } GV < GH \\ G_7^H W + G_7^V (1-W), & \text{if } GV > GH \\ \frac{G_7^H + G_7^V}{2}, & \text{if } GH = GV \end{cases} \quad (2-36)$$

垂直插補 G^V 與水平插補 G^H 同(2-14)(2-12)式所示。

初始階段的紅、藍色平面重建採用與色彩差異插補法相同的方法，如(2-9)(2-10)式所示。在優化階段中，綠色平面的更新採用簡化過的中值濾波插補法以降低(2-28)式的計算量[12]。如(2-37)式所示，以插補綠色像素位於紅像素為例：

$$G'_c = R_c + med(G - R) \quad (2-37)$$

最後，利用已更新的綠色平面，使用色彩差異插補法來更新紅、藍色平面。

第三章 CFA 解馬賽克演算法及硬體架構

3.1 本文提出的適應性權值彩色插補演算法

適應性次頻帶關聯插補法使用二維離散小波轉換取得次頻帶特徵，作為估測水平及垂直邊跡梯度及邊跡方向判別的依據，然而使用頻域轉換運算需要花費大量的計算成本，本研究提出以較低計算量的方法來取代適應性次頻帶關聯插補法，在估測水平及垂直邊跡梯度時所需的二維離散小波轉換步驟，可以較少的計算量達到與適應性次頻帶關聯插補法近似的效能，本文所提出的適應性權值彩色插補演算法流程如圖 3.1 所示。

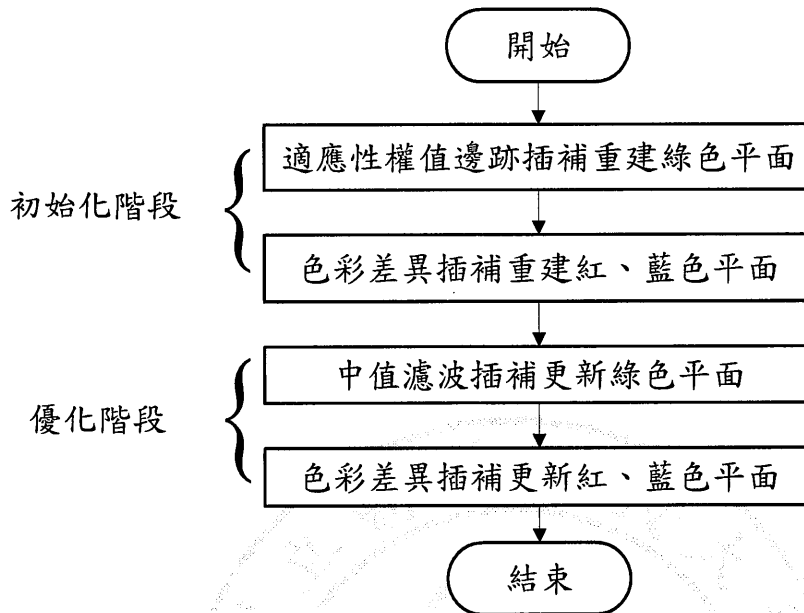


圖 3.1 適應性權值彩色插補演算法流程圖

初始階段的綠色平面的重建採用與適應性次頻帶關聯插補法相同方式，如(2-33)-(2-36)式所示。而水平梯度 GH 及垂直梯度 GV 則採用較為簡單的計算方式((2-11)(2-12)式)來取代 2D DWT 梯度計算。並且重新建立適合本研究演算法的梯度比 $GDSR$ 與加權值 W 之間的多閾值對應關係，如表 3.1 所示。

多閾值對應關係表是使用 Kodak 測試影像圖集[9]對本研究演算法評估梯度比 $GDSR$ 與加權 W 之間的對應關係以取得較佳成像品質。如圖 3.2 所示，為各種梯度比 $GDSR$ 與加權 W 的對應關係所取得最佳 PSNR

值(參見第 4.5 節的 PSNR 值評估)。並且在較佳 PSNR 值的條件下，調整合適的閾值並減少閾值數目，以減輕運算負擔及硬體資源。表 3.1 所列即為精簡後的權值 W 與梯度比 $GDSR$ 的對應表。

表 3.1 本研究色彩插補法中的 $GDSR$ 與 W 對應多閾值表

$GDSR$	≥ 0.6	$0.6 \sim 0.45$	$0.45 \sim 0.25$	< 0.25
W	1	0.85	0.75	0.6

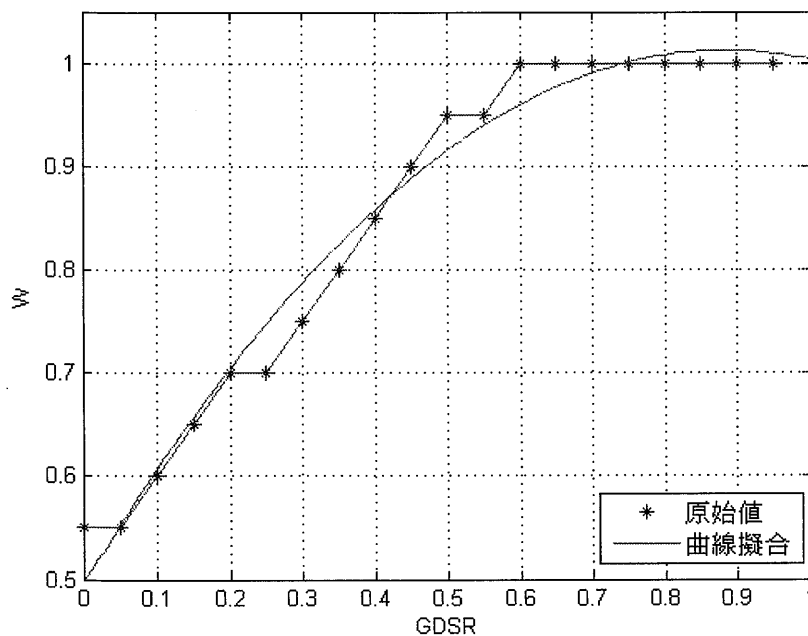


圖 3.2 較佳 PSNR 值下的 $GDSR$ 與 W 對應關係

後續插補紅、藍兩原色的演算過程與適應性次頻帶關聯插補法相同方式。換言之，初始階段的紅、藍色平面重建採用色彩差異插補法，如(2-9)(2-10)式所示。在優化階段中，綠色平面的更新採用簡化的中值濾波插補法，如(2-37)式所示。最後，利用已更新的綠色平面，使用色彩差異插補法更新紅、藍色平面。

3.2 適應性權值解馬賽克演算法之硬體架構設計

適應性權值解馬賽克演算法的硬體架構之執行程序主要分成 4 個步驟，分別為：適應性權值邊跡插補重建綠色平面，色彩差異插補重建紅、藍色平面，中值濾波插補更新綠色平面及色彩差異插補更新紅、藍色平面。設計架構將按照執行步驟分割成 4 個插補電路，由於紅、藍色無論插補或更新都採用色彩差異插補。因此都是採用相同的插補電路模組來實現。圖 3.3 為本研究所提出的適應性權值解馬賽克演算法架構方塊圖，CFA 串流影像經過 CFA 解馬賽克運算電路輸出 RGB 串流影像，串流影像每經過一個影像處理模組都會產生資料輸出延遲，因此模組內部會針對輸入影像控制信號 Frame Valid 及 Line Valid 執行

同步延遲，產生新的影像控制信號能夠正確對應處理後的影像輸出。

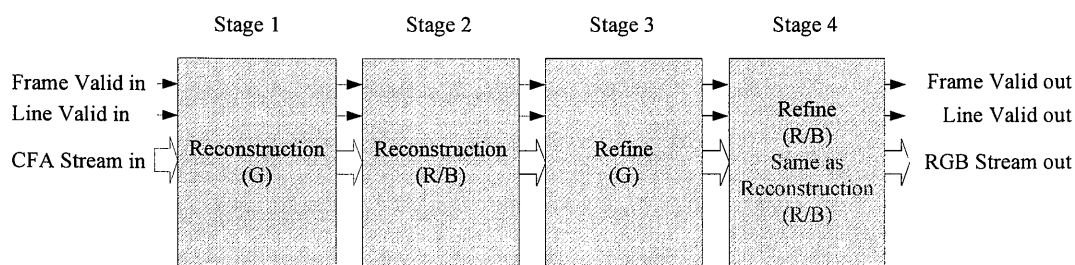


圖 3.3 CFA 解馬賽克架構方塊圖 (本研究色彩插補法)

3.2.1 綠色重建模組

綠色重建模組主要是二維濾波器視窗模組以及綠色插補模組所組成，如圖 3.4 所示。二維濾波器視窗模組能夠將 CFA 像素串流轉換成二維 CFA 視窗像素並且產生相對應的影像控制時序，以提供綠色插補模組執行視窗濾波器運算輸出綠色串流影像。

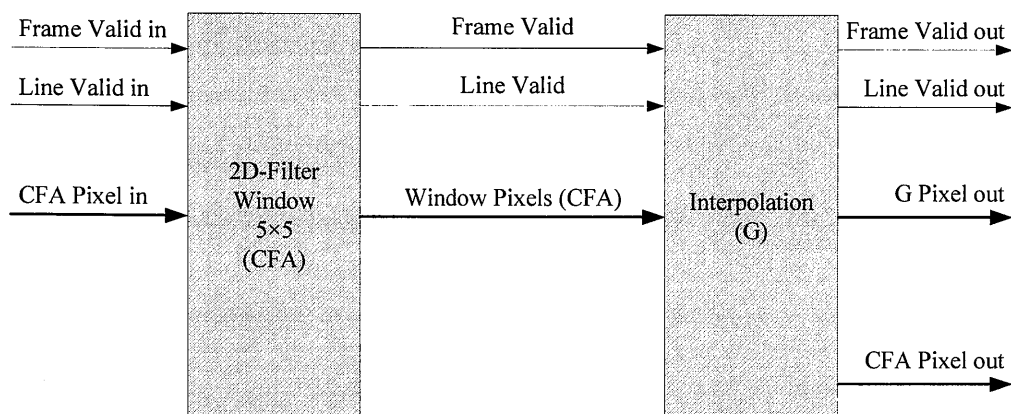


圖 3.4 綠色重建模組方塊圖

3.2.1.1 二維濾波器視窗模組

在影像處理中經常使用視窗濾波 (Window filtering) 來處理二維影像，如圖 3.5 所示。由於影像感測器是以串流形式輸出，無法直接執行視窗濾波，必需先將資料串流轉換成二維陣列輸出，再執行後級視窗濾波器運算。

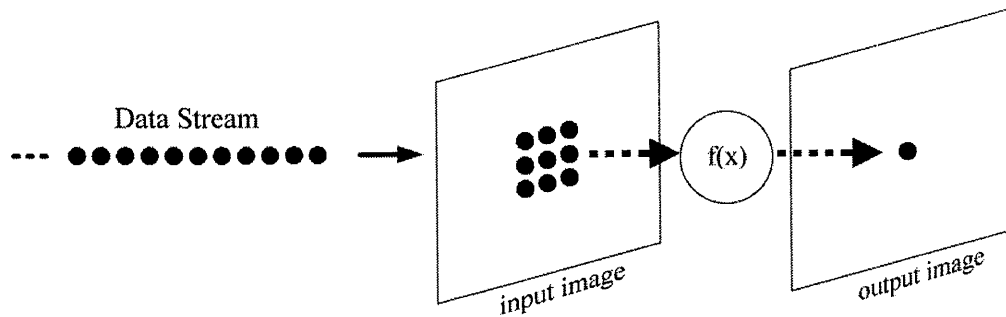


圖 3.5 二維影像的視窗濾波器運算

對於串流影像的二維處理，可以使用線緩衝器(line buffers)架構來完成。圖 3.6 顯示線緩衝器架構是利用資料串流在線緩衝器內部旅行，讓資料達到延時出現的效果。舉例說明線緩衝器的運作方式：圖 3.7 顯示對一幅寬度為 6 的影像執行視窗濾波器運算，當掃描視窗中心移至像素 P22 位置時，像素 P11 成為最後一次出現在掃描視窗內的像素。同時，像素 P33 則為掃描視窗內最新的像素。每次經過一個時脈週期就會有新進來的像素以及離開的像素，就像是在二維影像上執行視窗濾波器運算。

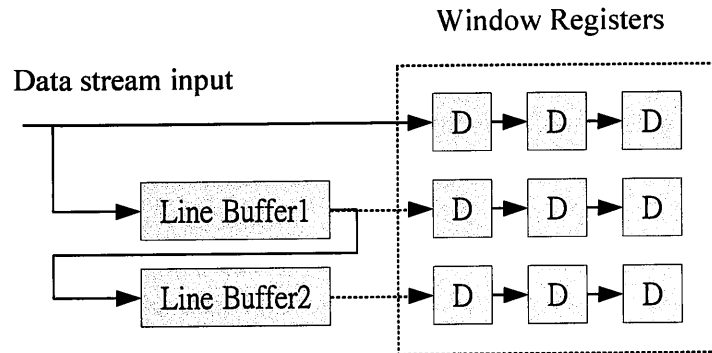


圖 3.6 線緩衝器架構 (3×3 區域)

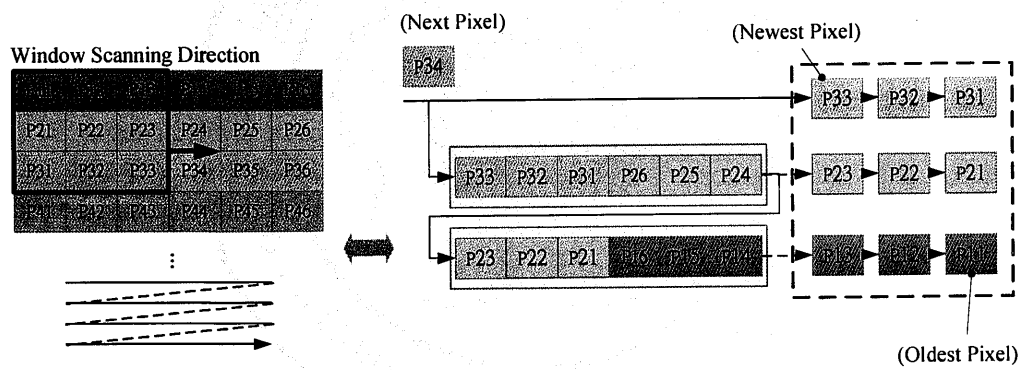


圖 3.7 線緩衝器架構的運作範例(3×3 區域)

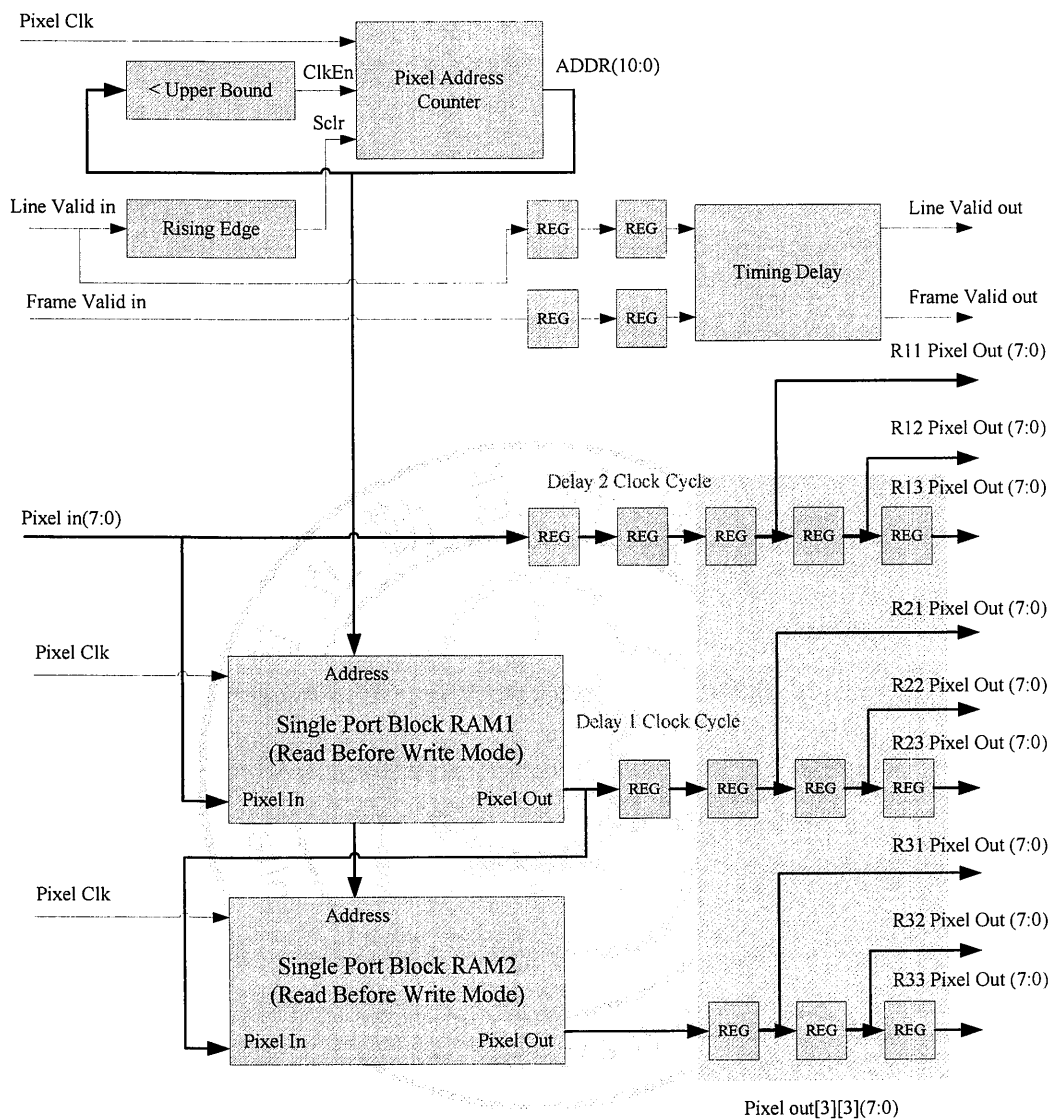


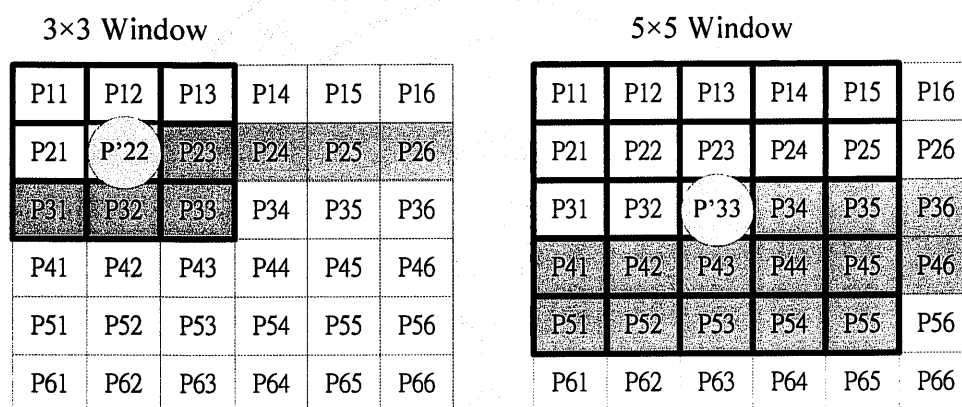
圖 3.8 二維濾波器視窗模組電路方塊圖(3×3 區域)

二維濾波器視窗模組採用與線緩衝器架構相同的方式設計，圖 3.8 為一個以 3×3 存取範圍之二維濾波器視窗模組。模組內部的線緩衝器為 Xilinx 開發工具所提供的單埠記憶體(single port RAM) IP 以及像素

位址計數器所組成。每一個單埠記憶體使用相同定址來源。資料存取採用優先讀取模式，當資料寫入某一位置時，優先讀取該位置上的資料作為輸出。像素位址計數用來計數每一列像素的位置，計數器只有當 Line Valid 信號上升時重置，否則上數到計數上限時停止。在時序上，單埠記憶體的資料讀取延遲為 1 個時脈週期，因此每串接一個單埠記憶體，上一列資料線必需多增加一級移位暫存器作為延遲讀取的抵補機制，以達到正確對應二維影像輸出。在硬體實現上， $M \times N$ 窗掃描架構至少需要 $M-1$ 組線緩衝器來完成。視窗暫存器高度為 3 時需要 2 組線緩衝器，而影像控制信號與像素輸入到視窗暫存器 R11 一樣需要延遲 2 個時脈週期是為了控制輸出信號與像素視窗暫存器 R11 輸出同步。而後級綠色插補模組運算視窗高度為 5，需要 4 組線緩衝器，為了保持影像控制輸入信號與輸入像素在視窗暫存器 R11 輸出同步，需要 4 個延遲時脈週期。

通常二維視窗濾波是以視窗中心點作為輸出像素位置。以圖 3.9(a) 為 3×3 視窗濾波為例，P'22 像素輸出是發生在視窗模組輸入第 P33 像素的時間點，也就是說，在時序上輸出像素落後輸入像素 1 個 Line Valid 及 1 個時脈週期。而圖 3.9(b) 的 5×5 視窗濾波，輸出像素落後輸入像素

2 個 Line Valid 及 2 個時脈週期。由此可得知二維視窗模組的輸入與輸出之間的時序關係，並且透過延遲輸入影像控制信號來取得對應輸出像素的影像控制信號。另外，為了能夠重覆參考輸入影像資訊，二維視窗模組以視窗中心暫存器輸出作為輸入影像的延遲輸出，可與處理後的影像一起執行後級運算。



(a)輸出落後輸入 1 個 Line Valid 及 1 個時脈週期 (b)輸出落後輸入 2 個 Line Valid 及 2 個時脈週期

圖 3.9 串流影像視窗運算後的時序關係

在硬體實現上，延遲影像控信號 Frame Valid 與 Line Valid 可以使用移位暫存器的方式延遲數目較少的時脈週期，而延遲較長的 Line Valid 時間可以利用連續 Line Valid 模式解決，如圖 3.10 所示。本文所使用

影像感測器的影像控制信號 Line Valid 輸出格式分成三種：Default、Continue Line Valid 與 XOR Line Valid 模式[11]。在連續 Line Valid 模式下只要設定經過多少個 Line Valid 週期後使 Frame Valid 信號上升或下降而達到延遲多少個 Line Valid 週期效果。因此，本文設計的時序延遲模組是以連續 Line Valid 模式作為延遲時序信號的解決方案。如圖 3.11 所示，為針對 5×5 的二維濾波器視窗模組輸出影像而設計的時序延遲模組，新產生的 Line Valid 與 Frame Valid 各延遲 2 個 Line Valid 週期及延遲 2 個時脈週期，如圖 3.12 所示。

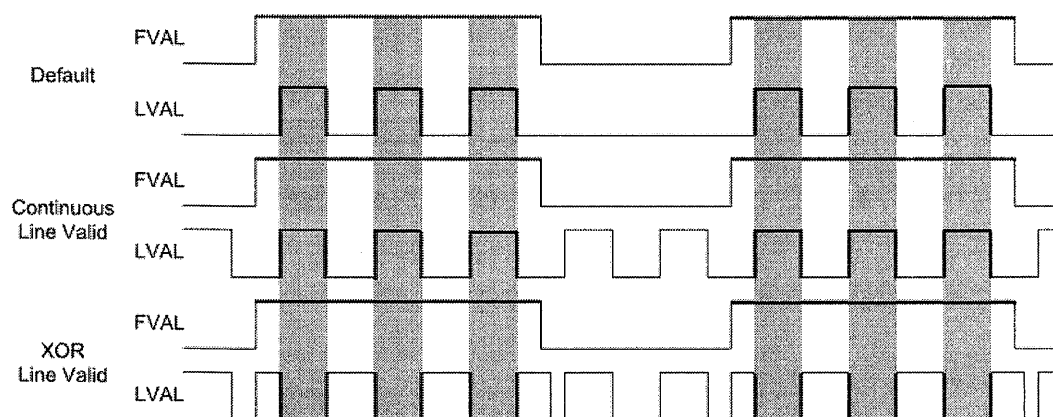


圖 3.10 影像控制信號模式選擇(TRDB-D5M 影像感測器)

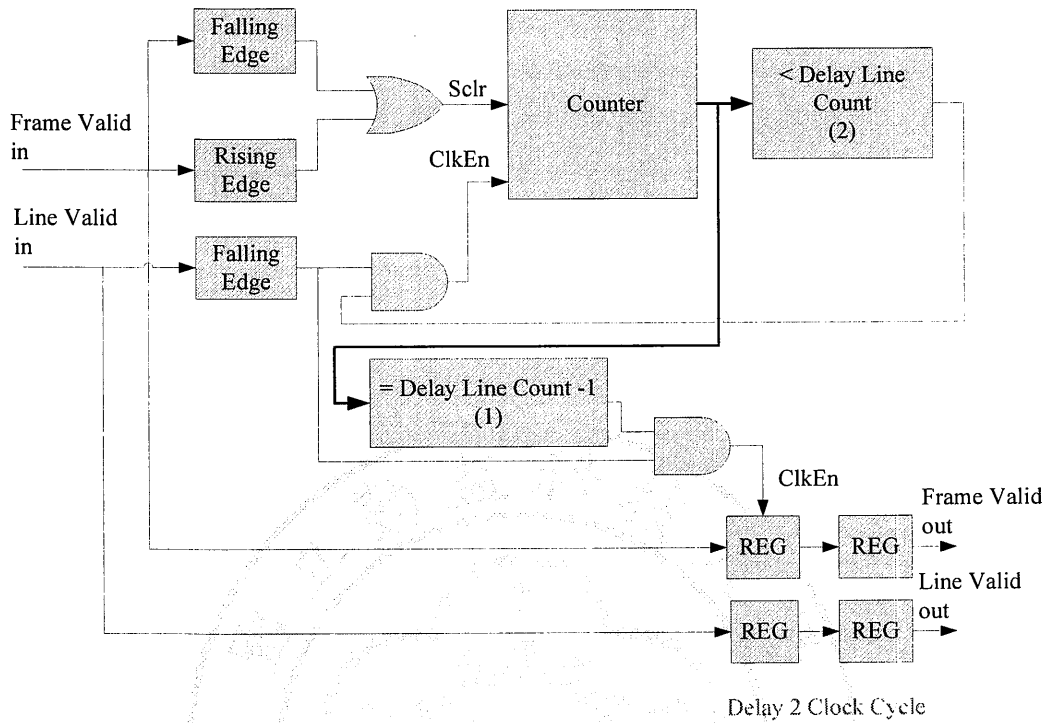


圖 3.11 時序延遲模組(延遲 2 列 2 像素)

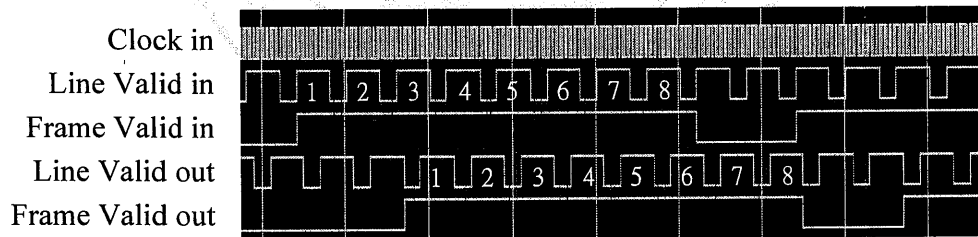


圖 3.12 時序延遲(延遲 2 列 2 像素)

3.2.1.2 綠色插補模組

綠色插補模組為 CFA 解馬賽克演算法初始階段的綠色插補電路，主要是從二維濾波器視窗模組取得的 CFA 視窗像素執行綠色插補運算。如圖 3.13 所示，為綠色插補模組架構方塊圖。輸入為十字型（距離為 5）CFA 視窗像素（參見圖 2.4），垂直邊跡梯度、水平邊跡梯度、垂直邊跡插補及水平邊跡插補單元分別計算 GV、GH、IV 及 IH 值。GDSR 計算單元利用 GV 與 GH 值計算出 GD 與 GS 值，而 GD 與 GS 的除法運算主要使用 Xilinx 開發工具提供的除法器 IP 來完成。而加權分配單元為根據 GDSR 值與方向梯度(GH 與 GV)的大小來分配水平與垂直插補加權(即表 3.1)。插補值與加權的乘法運算使用 Xilinx 開發工具提供的乘法器 IP 來完成。如圖 3.14 所示，由於貝爾格式的綠原色緊鄰紅及藍原色，其中 G0、G1 是為了區分不同像素位置。因此插補位置單元利用影像控制信號(Line Valid 及 Frame Valid)計算目前輸入視窗中心像素的位置類別(G0/R/B/G1)。根據目前視窗中心像素位置類別選擇像素輸出為插補像素(G')或 CFA 實際像素(G)。同步延遲單元(synchronous)主要目的為抵補像素輸出延遲，使輸入影像 CFA 與輸出影像 G 同步出現。也就是延遲輸入經由管線化插補運算所花費的延遲時脈週期數，達到輸出像素與輸出影像控制信號同步。

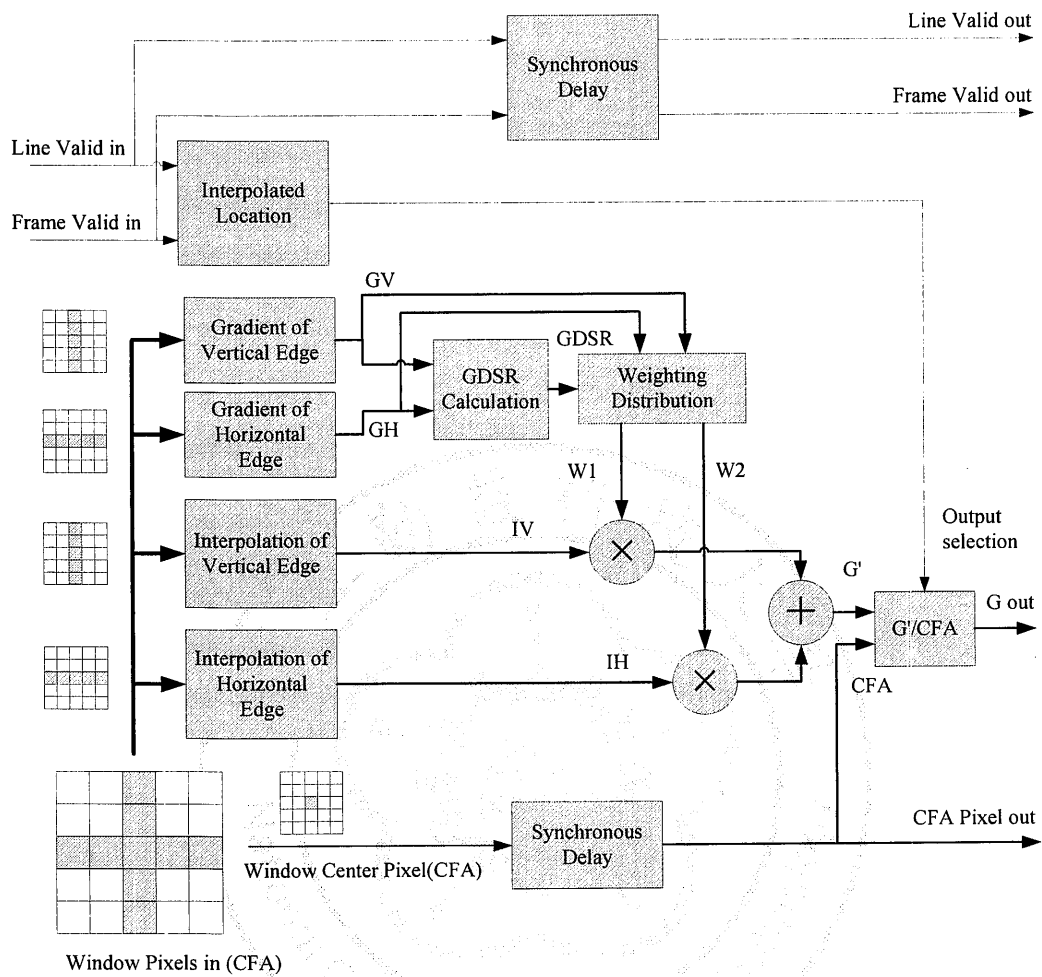


圖 3.13 綠色插補模組方塊圖

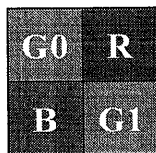


圖 3.14 貝爾格式 RGB 之間的位置對應關係

3.2.2 紅/藍色重建模組

由於紅/藍色插補運算需要參考原始 CFA 影像及綠色插補影像，因此紅/藍色重建模組內的插補模組輸入來源為存放 CFA 及綠色串流兩個二維濾波器視窗模組，如圖 3.14 所示。

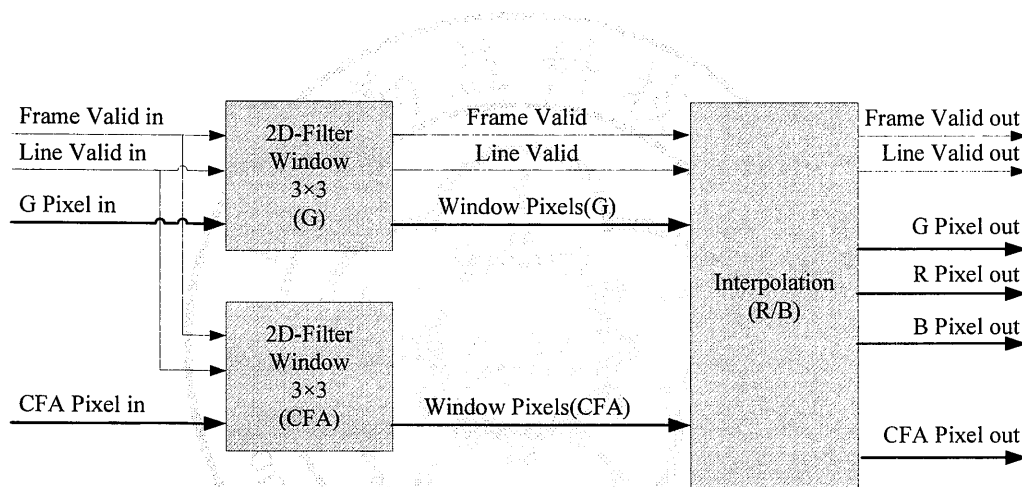


圖 3.15 紅/藍色重建模組方塊圖

圖 3.15 為紅/藍色插補電路，CFA 視窗輸入與綠色視窗輸入皆為 3×3 大小，輸入經由紅/藍色插補單元執行色彩差異插補(參考(2-9)(2-10)式)，分別取得三種插補輸出:對角插補、水平插補及垂直插補，如圖 3.16 所示。紅/藍色插補輸出如同前述之插補位置單元計算視窗中心像素位置類別而決定插補像素(R'/B')輸出或 CFA 實際像素(R/B)輸出。同

步延遲單元負責原始 CFA 與綠色串流與紅、藍色串流同步輸出。

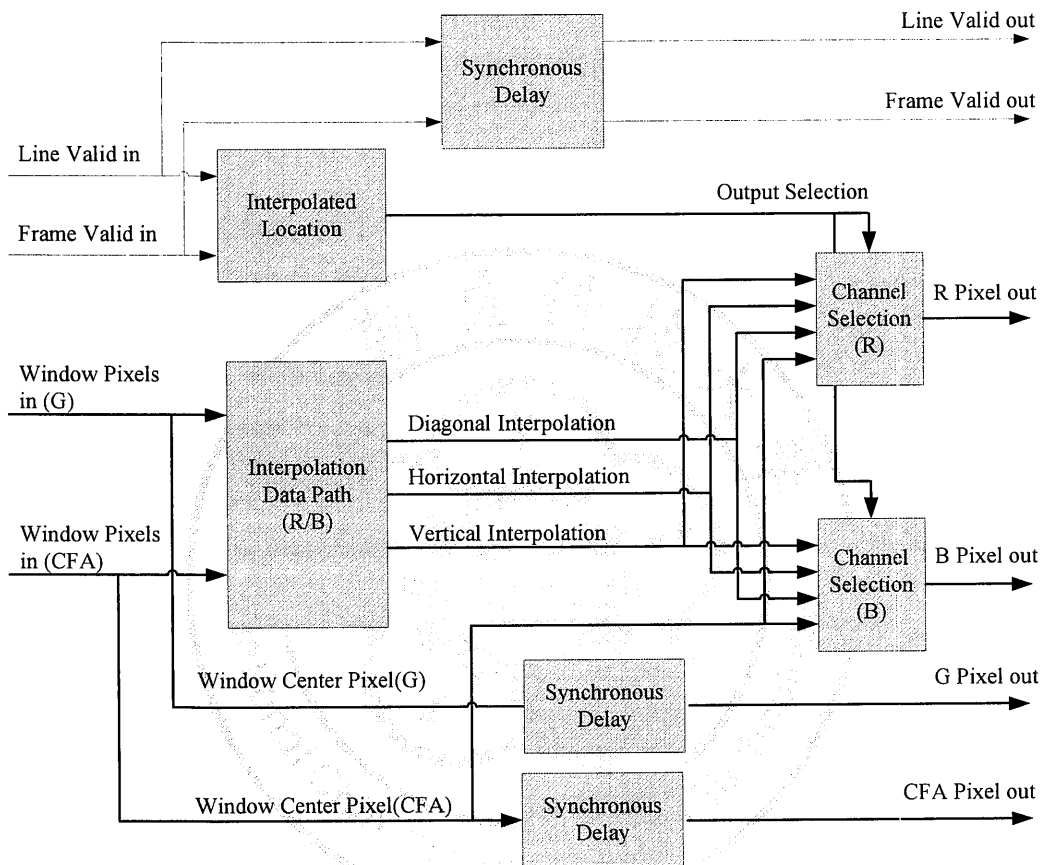


圖 3.16 紅/藍色插補模組方塊圖

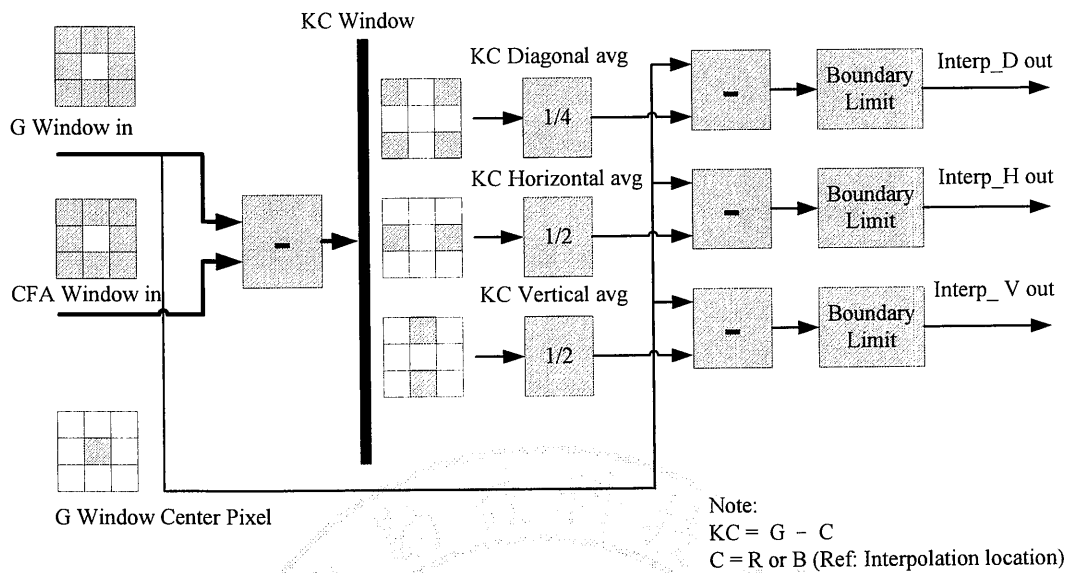


圖 3.17 紅/藍色的色彩差異插補單元方塊圖

3.2.3 綠色優化模組

綠色優化主要是使用中值濾波插補法及色彩差異運算更新綠色影像。圖 3.17 為綠色優化模組方塊圖，綠色優化需要 CFA 影像、KB 及 KR 影像作為參考輸入，其中 KR 與 KB 分別為綠色減去紅以及藍色影像所得。因此總共需要三個二維濾波器視窗模組作為綠色更新模組的輸入。

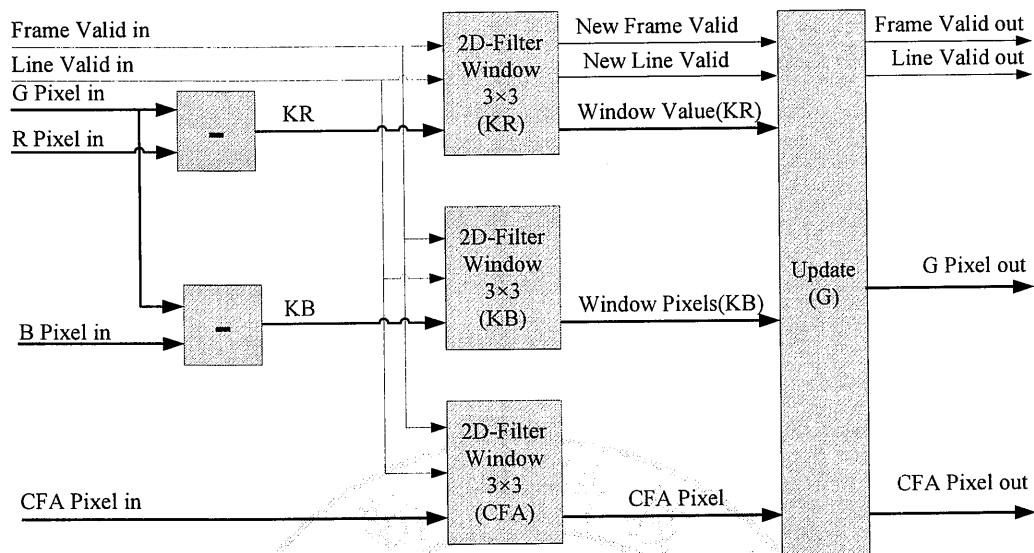


圖 3.18 綠色優化模組方塊圖

圖 3.17 中的更新模組(update)的細部方塊如圖 3.18 所示，CFA 像素(R/B)與執行中值濾波運算後取得的 KR 與 KB 中值相加取得綠色插補值(參考(2-37)式)。由於 KR 與 KB 值為有號數，因此需限定插補值在輸出像素的有效範圍內。

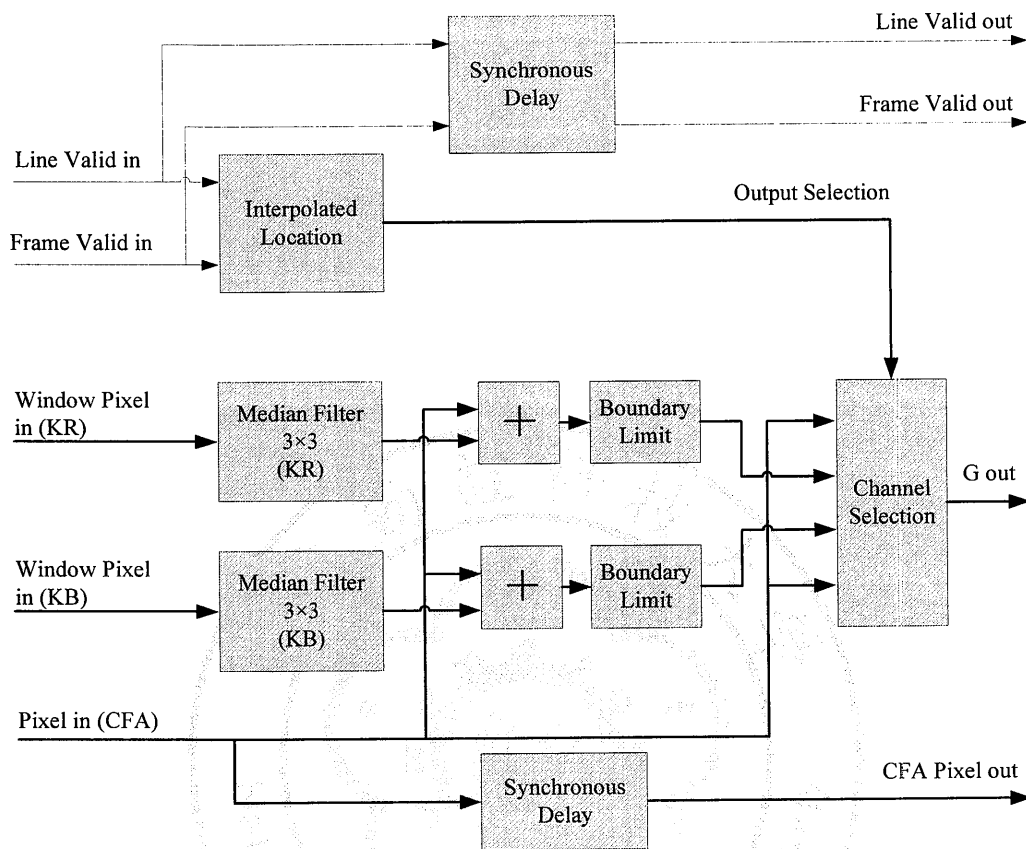


圖 3.19 綠色更新模組方塊圖

綠色更新模組內的 KR/KB 中值濾波器是使用 3x3 視窗(window)快速中值濾波器架構[10]實現的，如圖 3.19 所示。其演算法分為三個階段：

Stage1: 分別找出每一列之最大值(Max)、最小值(Min)及中間值(Med)。

Stage2: 分別找出 Stage1 中的(視窗區域各列最大值中的最小值、視窗區域各列中間值中的中間值、視窗區域各列最小值中的最大值)。

Stage3:找出 Stage2 中的三值的中間值即為視窗區域中間值。

其中 Sort 單元由 3 個比較器組成，如圖 3.20(a)所示。Max 及 Min 單元各為 2 個比較器組成，如圖 3.20(b)所示。因此 3x3 快速中值濾波器架構總共是由 19 個比較器組成。

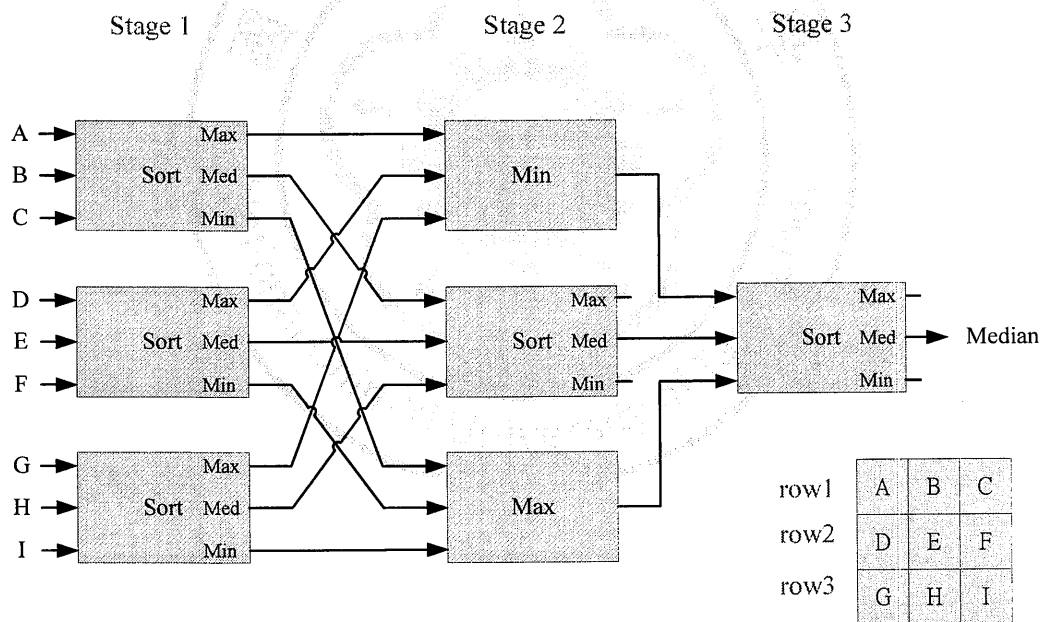
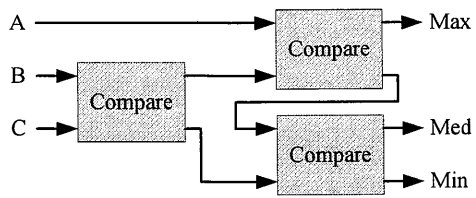
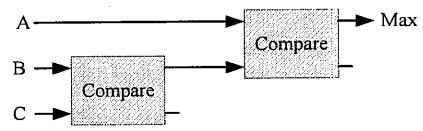


圖 3.20 3x3 快速中值濾波器架構方塊圖

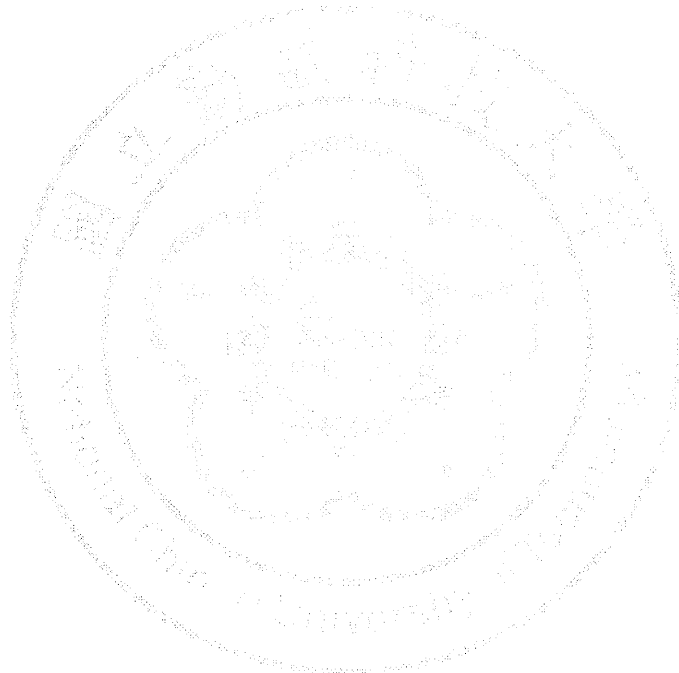


(b) Sort 單元



(c) Max 單元

圖 3.21 三輸入的排序與極值的架構方塊圖



第四章 FPGA 實現與實驗結果

4.1 開發環境

本研究使用 Xilinx ML402 開發板作為開發平台。FPGA 晶片為 Virtex4 系列的 XC4VSX35-FF668-10 晶片，針對數位訊號處理應用的高效能解決方案。晶片內部具有 DSP48 Slice，可以協助開發者快速實現高效能的乘法器與除法器。以及具有許多的 Block RAM 元件，能夠實現各種方式儲存控制。FPGA 開發工具軟體為 Xilinx ISE 11.1，提供 HDL 編輯環境、IP 生成的 Core Generator、ISim 模擬器、XST 合成工具與硬體實現工具等。

4.2 設計流程

圖 4.1 為 CFA 解馬賽克模組實現的設計流程圖，首先為規劃 CFA 解馬賽克模組的設計規格，包含功能、介面以及整體架構等，再按照設計規格使用硬體描述語言(hardware description language, HDL)撰寫模組的暫存器轉換層次(register transfer level, RTL)描述，並且進行功能

模擬測試。然後使用合成(synthesis)工具產生邏輯閘層次連線表(gate level netlist)檔案，提供實現設計工具完成佈局與繞線(place & route)，最後執行硬體測試。

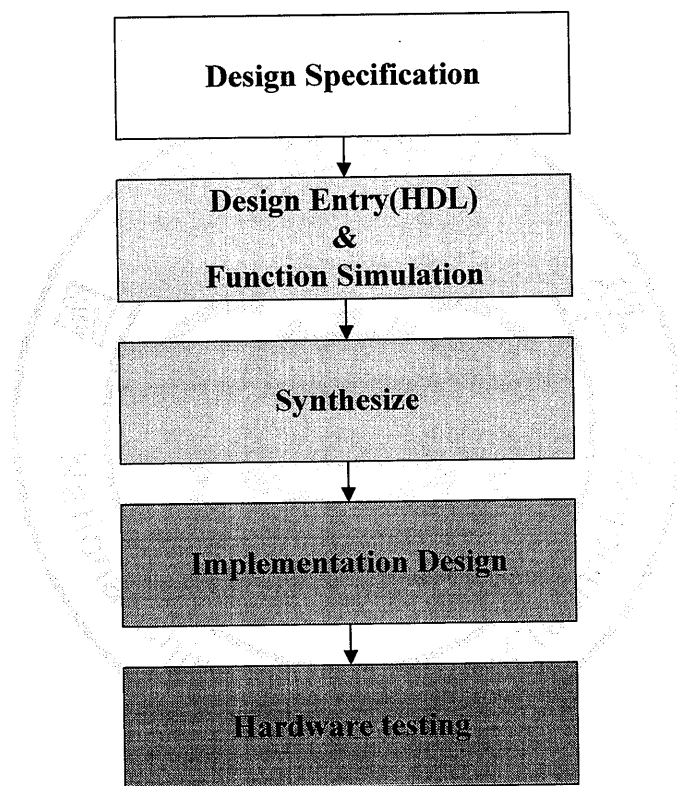


圖 4.1 CFA 解馬賽克模組設計流程圖

4.3 功能模擬

功能模擬的主要目的為檢查撰寫的HDL code是否達到設計目的，

並不包含電路的時序延遲。圖 4.2 為模擬影像感測器輸出一個 6x6 影像至二維濾波器視窗模組的波形圖。輸入端 FV_i(Frame Valid)信號為 1 時，有 6 個有效 LV_i (Line Valid)信號。當有效 LV_i 為 1 時，即有 6 個有效像素從 Pix_i 讀出。D11 至 D33 為 3x3 的視窗暫存器輸出。而輸出時序控制信號 FV_o 及 LV_o 對應視窗中心像素 D22 輸出，如圖 4.3 所示。其中當 D22 輸出為像素 22 時，也就是 D22 輸出時序上第 2 列有效 LV_o 的第 2 個有效像素時，D11 至 D33 輸出為像素位置(2,2) 視窗範圍內的 9 個像素。圖 4.4 為模擬 5x5 輸出的二維濾波器視窗模組的波形圖，輸出時序控制信號 FV_o 及 LV_o 對應視窗中心像素 D33 輸出。

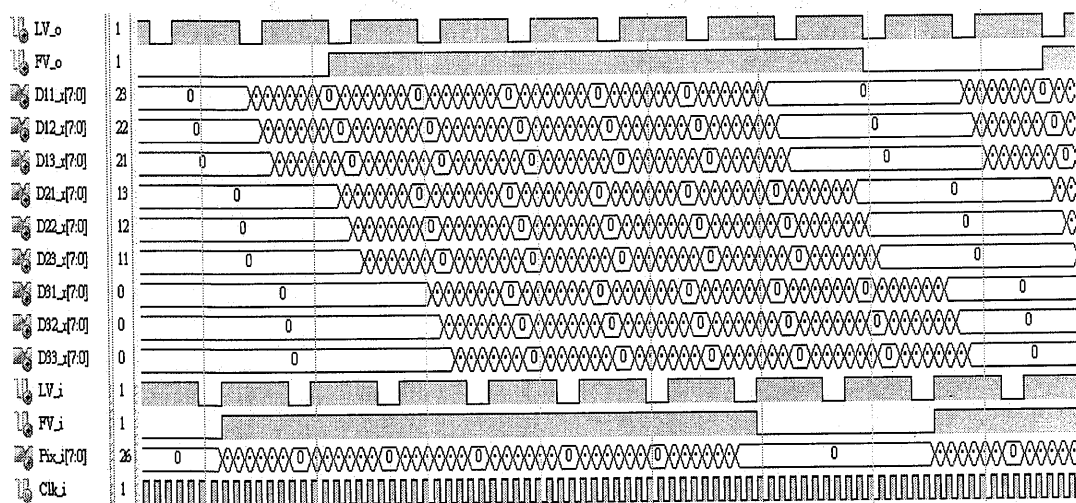


圖 4.2 模擬二維濾波器視窗模組(3x3)輸出波形圖

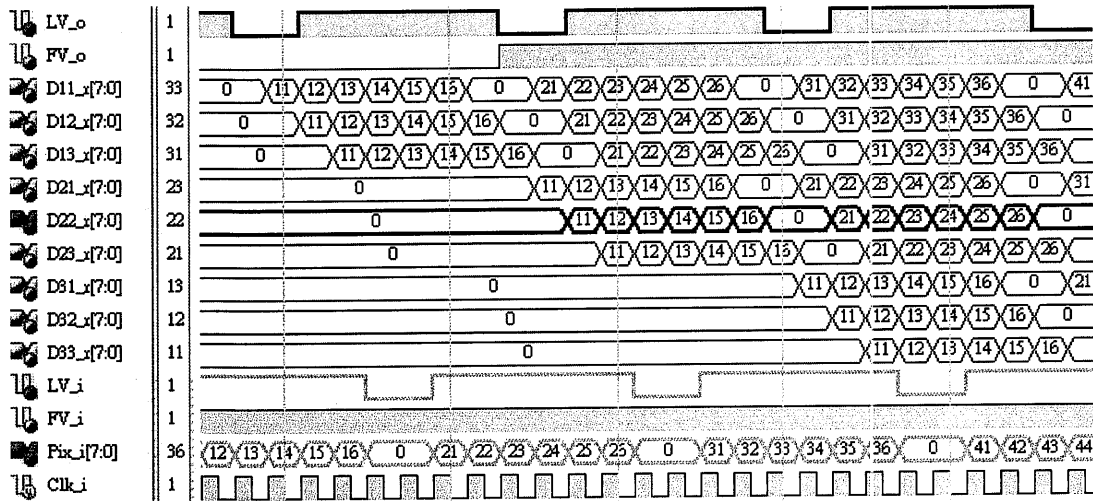


圖 4.3 模擬二維濾波器視窗模組(3x3)輸出波形圖(局部)

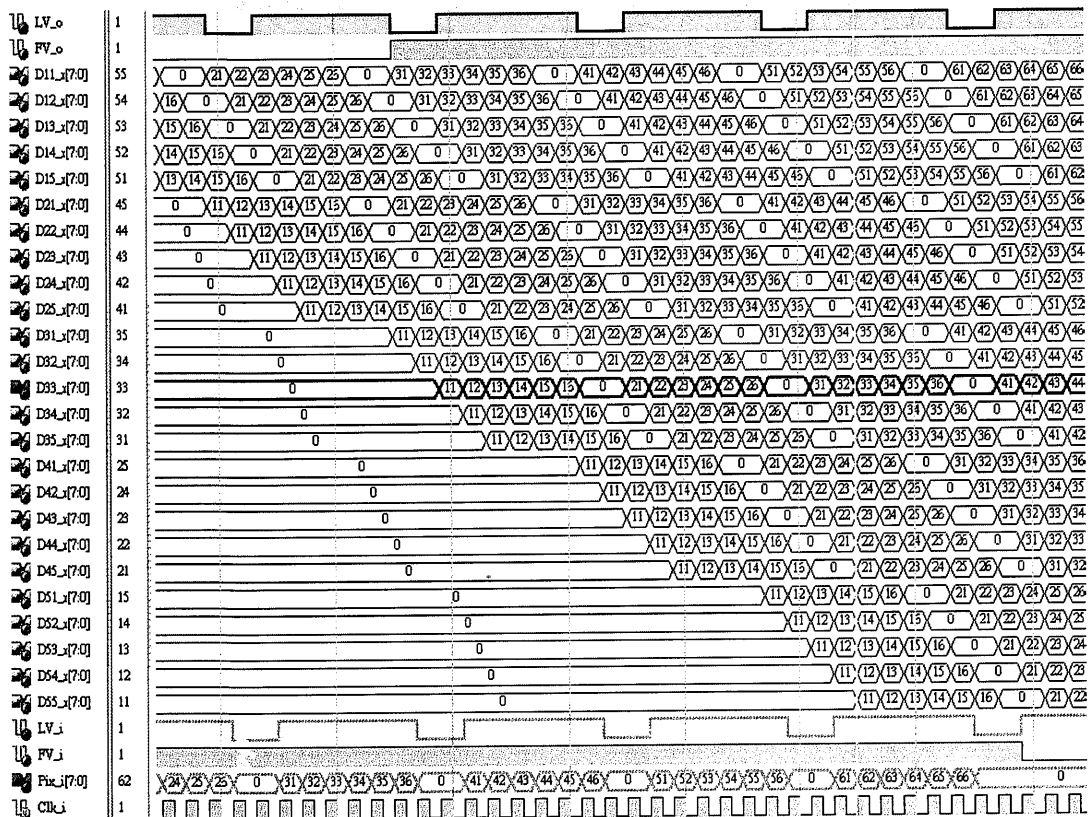


圖 4.4 模擬二維濾波器視窗模組(5x5)輸出波形圖(局部)

前面的子模組是以輸入較少測試樣本透過波形輸出方式完成的功能模擬，然而對於輸入資料量龐大的測試影像來模擬 CFA 解馬賽克模組的輸出，則需使用檔案存取的方式來完成。首先將測試影像轉換貝爾影像後儲存成二進制檔，利用測試程序(testbench)讀取檔案中的貝爾影像，然後模擬影像感測器輸出貝爾像素至 CFA 解馬賽克模組，最後將 CFA 解馬賽克模組輸出的 RGB 影像串流儲存成點陣圖(BMP)檔格式以便於檢視輸出結果。圖 4.5 為模擬 CFA 解馬賽克測試程序流程圖。

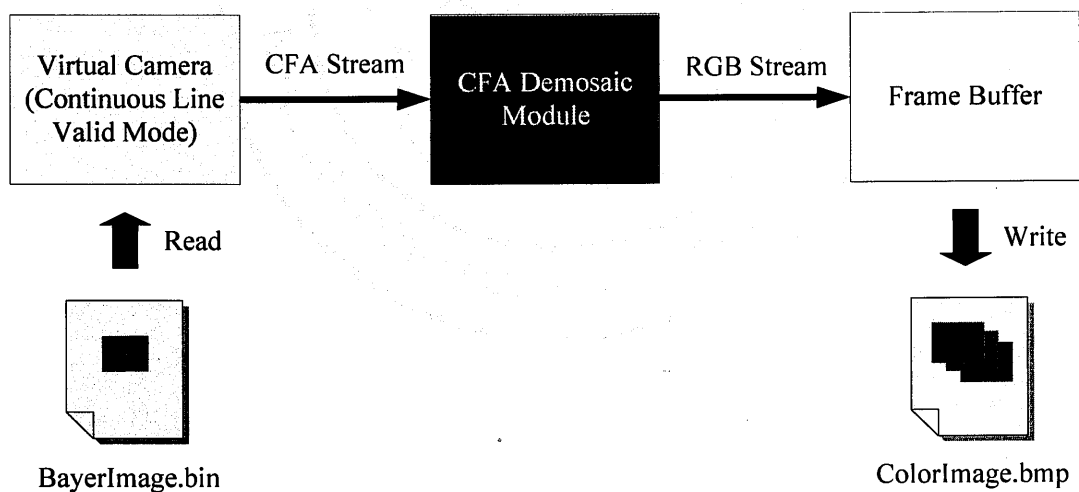


圖 4.5 模擬 CFA 解馬賽克測試程序流程圖

4.4 FPGA 實現結果

在 Xilinx FPGA 晶片中，Slice 為一個基本的可規劃邏輯區塊。Slice 內部有 LUTs(Look-Up Tables)以及多工器作為組合邏輯電路的實現，還有算術邏輯元件以及提供時序邏輯的暫存器元件。而 SLICEM 為可以被配置成分散式記憶體或移位暫存器的 Slice。表 4.1 為解馬賽克模組的 FPGA 硬體資源利用度報告，其中 Line Buffer 是使用 FPGA 內部的記憶體元件 RAMB16 實現的，RAMB16 儲存深度為 16Kbits，1 個 RAMB16 可以實現儲存深度為 2048Bytes 的 Line Buffer。本研究的 Line Buffer 模組是由 2 個 RAMB16 組成，可以處理寬度為 4096 像素以下的影像。乘法器與除法器模組是由 FPGA 內部的 DSP48 Slice 所實現，一個 DSP48 Slice 可以實現 18×18 的乘法運算。

表 4.1 解馬賽克模組的硬體資源利用度報告

Device Utilization Summary (XC4VSX35-FF668-10)			
Logic Utilization	Used	Available	Utilization
Number of BUFGs	1	32	3%
Number of DSP48s	2	192	1%
Number of RAMB16s	34	192	17%
Number of Slices	1803	15360	11%
Number of SLICEMs	106	7680	1%

解馬賽克模組經過佈局及佈線(place & route)後的靜態時序分析(static timing analysis)，可以滿足輸入工作時脈在 100MHz 及工作週期 50% 下的時序約束(timing constraints)，而最大工作時脈為 104.778MHz。圖 4.6 為 FPGA 實現 CFA 色彩插補演算法電路的開發環境。其中影像感測器模組作為 FPGA 開發板的 CFA 影像串流輸入，並且經由 FPGA 內部的色彩插補電路產生全彩影像串流，最終輸出至顯示器顯示。

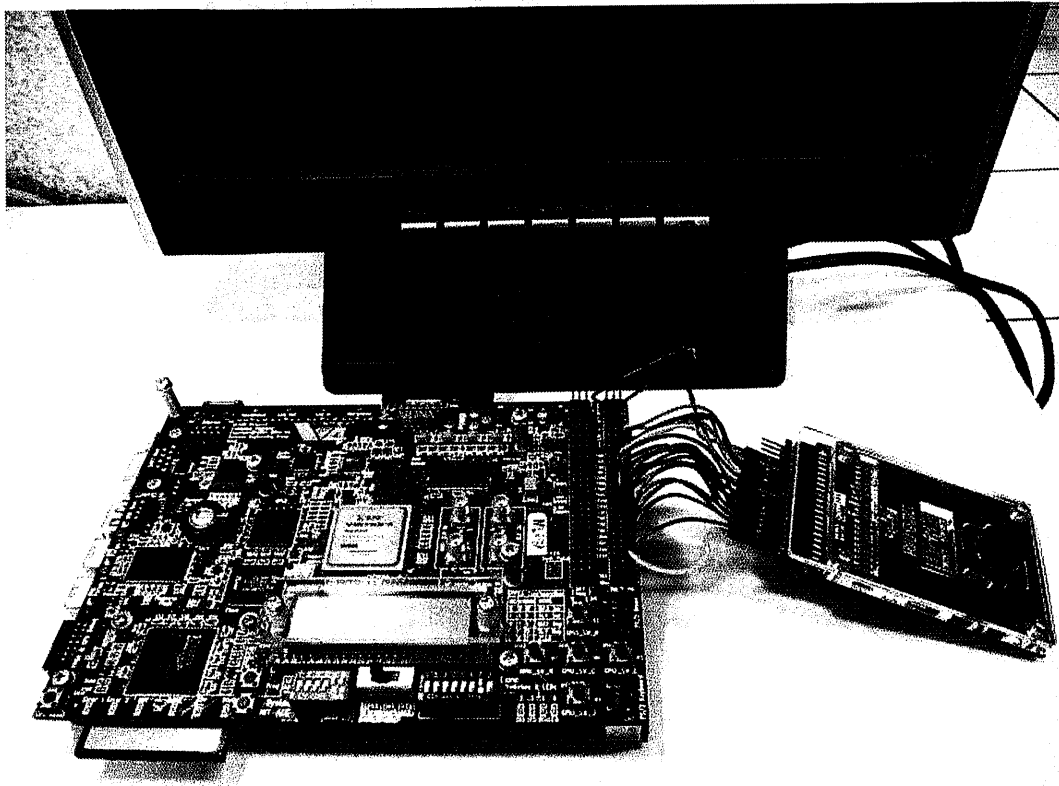


圖 4.6 FPGA 實現 CFA 色彩插補演算法電路的開發環境

4.5 PSNR 比較

為了評估各種解馬賽克方法在影像重建品質上的差異，利用峰值信噪比(peak signal-to-noise ratio, PSNR)值作為影像重建品質的指標。

PSNR 值定義如下：

$$PSNR = 10 * \log \left(\frac{255^2}{MSE} \right) \quad (4-1)$$

$$MSE = \frac{\sum_{n=1}^{FrameSize} (I_n - P_n)^2}{FrameSize} \quad (4-2)$$

在此， MSE 為均方誤差值(mean square error)， I_n 為原始影像第 n 個像素值， P_n 為重建的影像第 n 筆像素值， $FrameSize$ 在此為影像的高與寬之乘積。PSNR(單位為 dB)值越高代表失真越少。

本文採用 24 張全彩 Kodak 數位照片[9]作為測試影像。如圖 4.7 所示，每張影像的大小皆為 768×512 。實驗流程為：先將測試影像當作原始影像並且轉成貝爾格式，再經由解馬賽克後計算每張重建影像的

PSNR 值。表 4.2 為各種解馬賽克方法影像重建後的 PSNR，演算法包含有：雙線性插補(Bilinear)、二階修正線性插補法(ACPI)、色彩差異插補法(ECI)、有效疊代加權邊跡與色彩差異插補法(HEID)、適應性次頻帶關聯插補法(ASCI)以及本文所提出的適應性權值插補法。經由 Matlab 軟體實驗測試，本研究之色彩插補法所得到的 PSNR 值為 40.72dB 與 ASCI 演算法的 PSNR 值 40.87dB，僅相差 0.15dB，說明本文演算法使用較簡單的梯度計算來取代 ASCI 演算法中使用頻域轉換計算梯度，即可達到與 ASCI 演算法成像品質相近的效果，在硬體實現上可以達到較少的硬體成本。CFA 解馬賽克模組經由 FPGA 功能模擬影像輸出也的確能夠獲得優良的成像品質(40.50dB)。圖 4.8 與圖 4.9 為各種解馬賽克方法的成像效果。

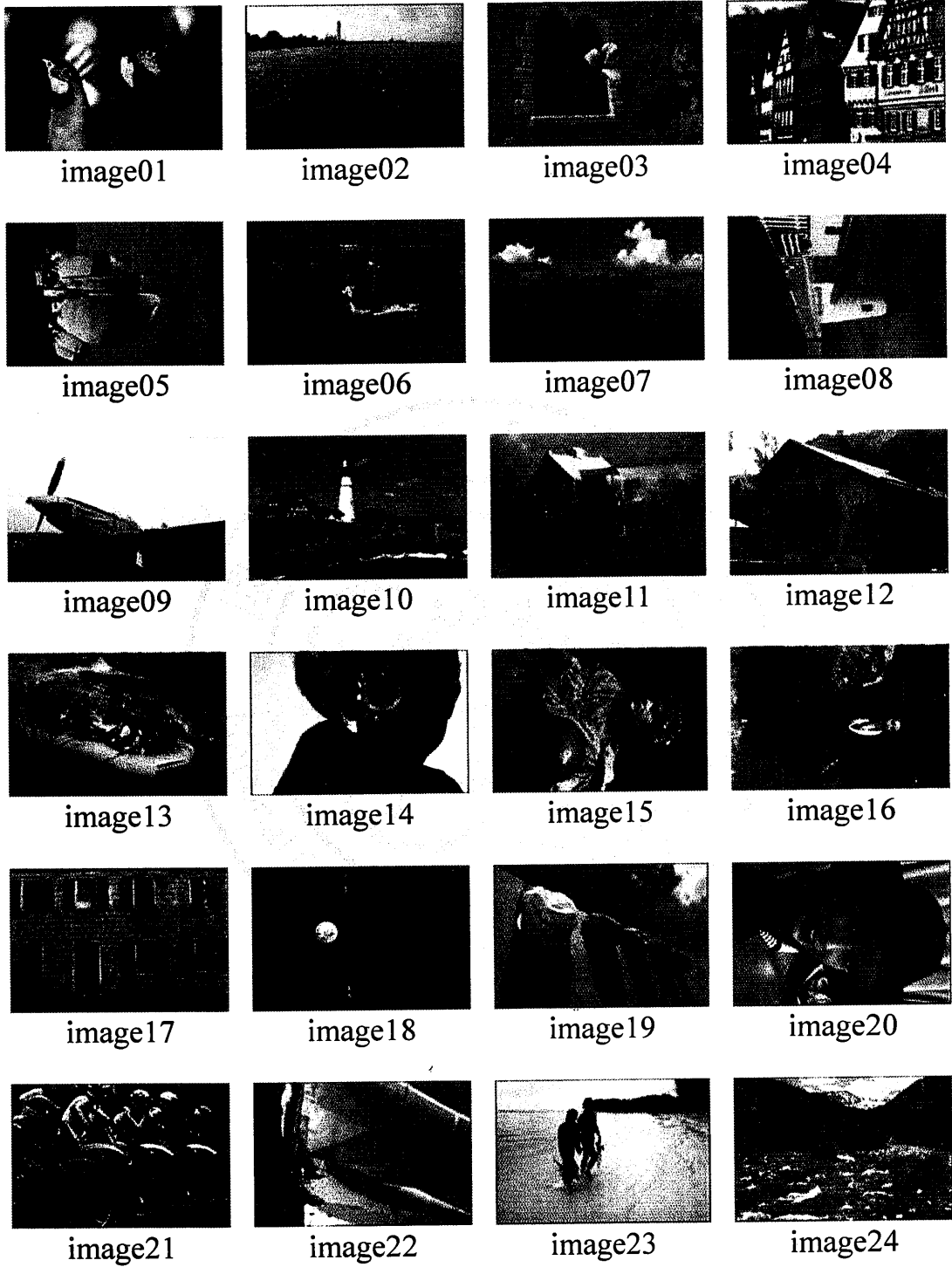


圖 4.7 24 張 Kodak 測試影像

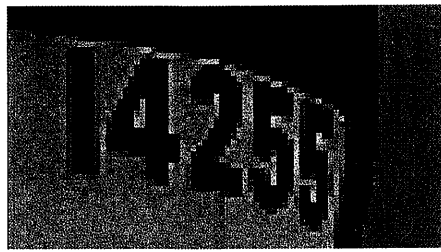
表 4.2 重建影像的 PSNR 值(單位：dB)

Image	Channel	Bilinear [2]	ACPI [4]	ECI [3]	HEID [5]	ASCI [7]	Proposed	Proposed (*1)
1	R	35.63	40.54	41.98	41.96	42.52	42.60	41.28
	G	39.16	44.58	44.96	46.35	47.20	46.97	46.40
	B	35.26	40.21	42.68	43.01	43.38	43.61	42.65
2	R	27.71	32.62	35.55	40.22	41.00	40.29	40.00
	G	32.11	37.34	38.06	43.44	44.05	43.29	43.09
	B	27.88	32.74	35.11	39.23	39.81	39.30	39.01
3	R	33.62	38.85	41.33	42.47	42.92	42.88	42.40
	G	37.48	43.23	43.03	44.33	46.04	45.89	45.49
	B	33.21	38.16	39.85	40.87	41.30	41.26	40.84
4	R	23.51	28.65	30.94	36.01	36.63	36.56	36.42
	G	28.36	34.51	33.95	40.40	40.57	40.33	40.19
	B	23.46	28.61	30.84	36.75	37.11	36.98	36.80
5	R	32.57	37.72	40.03	43.00	43.89	43.70	43.10
	G	36.80	42.74	42.64	46.17	47.14	46.81	46.41
	B	32.62	37.78	39.87	42.43	43.17	43.01	42.38
6	R	29.32	33.99	36.81	39.20	40.08	39.90	39.63
	G	33.32	38.31	39.16	43.28	43.94	43.62	43.40
	B	29.29	34.08	36.99	40.59	41.12	40.91	40.52
7	R	31.36	36.13	38.55	43.40	44.16	43.26	42.70
	G	35.75	40.80	41.07	46.40	46.99	46.08	45.65
	B	31.21	36.00	37.72	42.24	42.89	42.27	41.74
8	R	28.16	33.42	35.63	40.98	41.31	41.34	40.74
	G	32.87	39.40	38.28	44.27	44.07	44.36	44.02
	B	28.13	33.39	35.48	40.32	40.39	40.60	39.88
9	R	31.16	36.15	36.72	41.04	41.97	42.44	41.76
	G	34.48	40.36	40.60	43.95	44.27	45.16	44.70
	B	29.37	33.86	32.32	37.66	40.18	39.79	40.10

Image	Channel	Bilinear [2]	ACPI [4]	ECI [3]	HEID [5]	ASCI [7]	Proposed	Proposed (*1)
10	R	28.71	33.34	36.47	40.29	40.58	40.51	40.06
	G	32.59	37.39	38.49	43.42	43.23	43.24	42.99
	B	28.55	33.18	36.02	39.43	39.61	39.54	39.10
11	R	30.90	35.25	37.60	38.66	39.29	39.24	38.79
	G	34.46	39.10	39.81	41.66	42.39	42.30	42.06
	B	30.26	34.55	36.87	38.21	38.80	38.81	38.39
12	R	26.97	31.25	34.38	36.81	36.90	36.93	36.53
	G	30.17	34.32	36.43	39.68	39.39	39.56	39.35
	B	26.20	29.93	33.26	35.37	35.29	35.34	35.05
13	R	27.36	31.82	31.71	32.13	35.33	35.15	35.51
	G	31.65	37.38	37.23	36.84	39.91	39.97	39.99
	B	28.06	32.79	33.03	33.61	36.08	36.14	36.00
14	R	29.46	33.27	32.25	34.82	36.22	36.17	36.50
	G	34.10	39.52	39.47	41.29	42.37	42.77	42.62
	B	29.67	34.34	34.99	39.24	40.15	40.38	40.10
15	R	31.44	36.23	38.37	41.92	42.16	42.20	41.50
	G	34.57	39.56	40.40	44.34	44.43	44.34	43.99
	B	30.77	35.33	38.22	40.42	40.62	40.63	39.84
16	R	27.17	31.38	34.28	36.33	36.81	36.88	36.26
	G	30.44	34.63	36.13	39.27	39.26	39.63	39.35
	B	26.94	30.87	33.80	35.50	35.82	35.96	35.54
17	R	24.97	29.62	31.10	36.10	37.41	37.16	37.44
	G	29.38	34.82	35.52	41.01	41.06	41.07	40.98
	B	24.91	29.78	32.63	37.86	38.05	38.03	37.85
18	R	30.27	34.00	32.80	34.30	37.34	36.30	37.28
	G	35.43	40.92	40.21	40.62	43.72	42.89	43.50
	B	31.33	35.92	35.56	38.46	41.47	40.73	41.05
19	R	31.19	35.64	33.90	37.82	41.35	40.41	41.55
	G	36.10	42.31	41.78	42.75	45.07	45.28	45.10
	B	32.49	37.24	36.28	39.15	41.03	41.13	40.61

Image	Channel	Bilinear [2]	ACPI [4]	ECI [3]	HEID [5]	ASCI [7]	Proposed	Proposed (*1)
20	R	30.90	34.70	36.71	35.43	37.72	37.06	37.95
	G	35.75	40.76	41.24	41.21	43.04	42.84	43.17
	B	32.20	36.79	34.22	40.53	42.12	42.00	41.88
21	R	25.36	30.75	32.00	34.37	37.10	37.02	37.48
	G	29.18	36.01	36.54	38.41	40.88	41.07	40.94
	B	25.66	31.28	33.40	34.79	36.63	36.78	36.44
22	R	30.10	35.09	37.93	39.22	41.73	41.13	41.66
	G	34.79	41.46	41.74	44.49	45.50	45.66	45.21
	B	31.19	36.34	34.05	40.43	41.21	41.27	40.73
23	R	30.50	35.15	34.01	38.76	42.12	40.87	41.39
	G	35.81	42.47	41.98	44.69	46.59	46.18	45.60
	B	31.64	36.60	38.41	41.09	42.63	42.31	41.71
24	R	22.88	27.05	29.96	35.01	35.15	34.85	35.07
	G	26.48	30.80	32.66	38.33	37.41	37.27	37.19
	B	22.99	27.08	30.46	34.06	33.76	33.71	33.50
Ave.		30.64	35.59	36.73	39.83	40.87	40.72	40.50

註: 1. 本文 CFA 解馬賽克模組的功能模擬結果



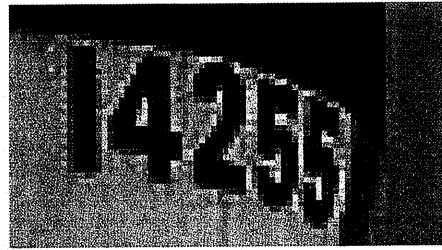
(a) Original



(b) Bilinear



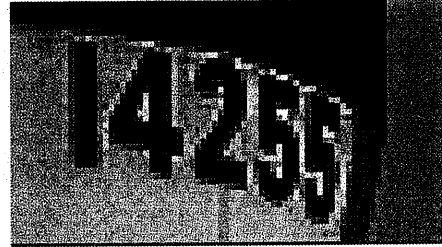
(c) ACPI



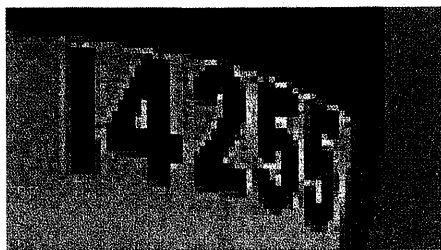
(d) ECI



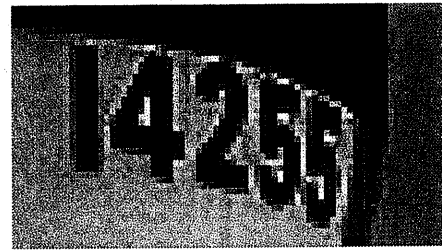
(e) HEID



(f) ASCI

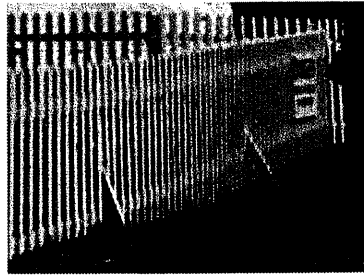


(g) Proposed

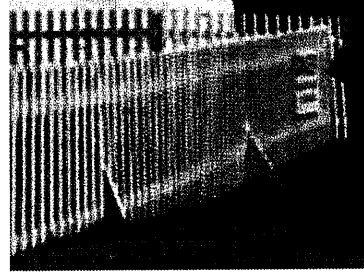


(h) Proposed (Function
Simulation)

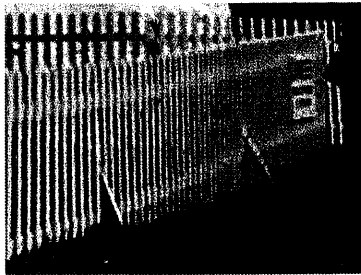
圖 4.8 各種解馬賽克方法的成像效果(Image5)



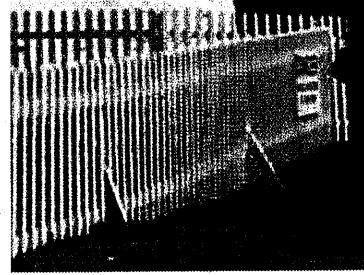
(a) Original



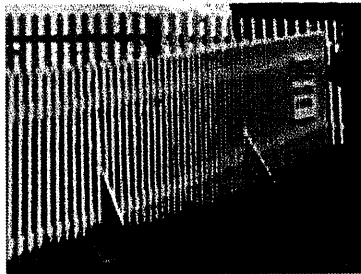
(b) Bilinear



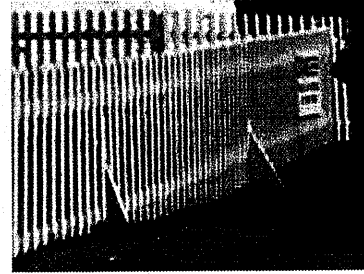
(c) ACPI



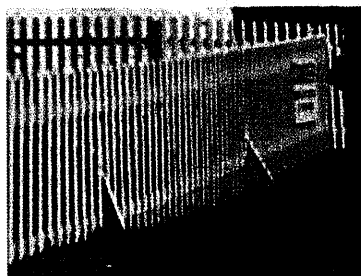
(d) ECI



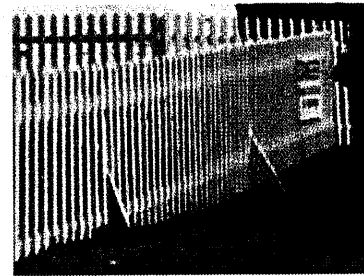
(e) HEID



(f) ASCI



(g) Proposed



(h) Proposed (Function
Simulation)

圖 4.9 各種解馬賽克方法的成像效果(Image8)

第五章 結論與未來展望

由於傳統紅、藍色重建都需要參考到綠色平面資訊，因此綠色平面的重建品質對於整體重建品質有很大的影響。進一步地，為了提升重建品質，傳統上，常採用邊跡偵測再依邊跡方向執行插補運算。適應性次頻帶關聯插補(ASCI)演算法在執行邊跡偵測時，利用二維離散小波轉換取得水平與垂直邊跡特徵，並據以評估水平與垂直梯度，接著依據梯度強弱比例，選擇插補公式及權值。

然而使用頻域轉換運算需要花費大量的計算成本，本論文提出的適應性權值彩色插補演算法可以較低計算量方法取代頻域轉換計算邊跡梯度，期望達到與 ASCI 成像品質相近的效果。經由軟體模擬，本論文所提的插補法之 PSNR 值為 40.72dB 與 ASCI 插補法的 PSNR 值 40.87dB 比較，僅相差 0.15dB，說明本研究方法的確能達到與 ASCI 成像品質相近的效果。

在以 FPGA 實現本演算法的硬體架構中，為了處理 CFA 影像是以即時串流訊號組成，本論文設計了線緩衝器運算、影像控制信號的延遲機制以及管線化的插補運算電路，以達到一個時脈週期輸出一個

RGB 像素的效能，大幅縮短傳統 DSP 循序運算所需的時脈數。

在未來研究方面，建議提高相同運算單元的共享比率，可有效降低硬體資源。例如：綠色重建模組的權值乘法器，紅、藍色重建與優化模組、以及綠色優化模組的 KR、KB 中值濾器等。若能採用鎖相迴路(phase-locked loop, PLL)，可產生比影像感測器快一倍的工作頻率。因此，可在輸出像素週期不變的條件下，以每兩個工作時脈週期處理一個像素。當第一個時脈週期完成運算後，第二個時脈週期就可以把硬體資源提供給其他運算使用，達到硬體資源分享的效果。因此，與本研究所提出的 CFA 解馬賽克模組，在相同的影像輸出速度條件下，可以使用更少的硬體資源，來降低硬體成本。

參考文獻

- [1] B. E. Bayer, "U.S. pat.3971065: Color imaging array," July 1976.
- [2] H. S. Hou, and H. C. Andrews, "Cubic splines for image interpolation and digital filtering," IEEE Trans. Acoustics, Speech, and Signal Processing, vol. ASSP-26, no.6, pp.508-517, Dec. 1978.
- [3] S. C. Pei and I. K. Tam, "Effective color interpolation in CCD color filter arrays using signal correlation," IEEE Trans. Circuits and Systems for Video Technology, vol. 13, pp. 503-513 June 2003.
- [4] J. F. Hamilton Jr. and J. E. Adams, "Adaptive Color Plane Interpolation in Single Sensor Color Electronic Camera," United States Patent, 5629734, May 1997.
- [5] C. Y. Su, "Highly effective iterative demosaicing using weighted-edge and color-difference interpolations," IEEE Transactions on Consumer Electronics, vol. 52, no. 2, pp. 639-645, March 2006.
- [6] W. Lu and Y.-P. Tan, "Color filter array demosaicking: New method and performance measures," IEEE Trans. Image Processing, vol. 12, no. 10, pp. 1194–1210, October 2003.
- [7] 阮任民，應用邊跡關係於彩色影像解馬賽克之研究，國立勤益科技大學，碩士論文，2010。
- [8] W.T.Freeman,"Median filter for reconstructing missing color

samples”, U.S. Patent, No. 4,724,395, 1988.

[9] “Kodak test images”, available at

<http://www.csee.wvu.edu/~xinl/demo/demosaic.html>

[10] Waltz, FM, Hack, R., Batchelor, BG, "Fast, efficient algorithms for 3x3 ranked filters using finite-state machines," Proc. SPIE Conf. on Machine Vision Systems for Inspection and Metrology VII, Vol. 3521, Paper No. 31, November 1998

[11] TRDB-D5M 相關文件

[http://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=Taiwan
&CategoryNo=85&No=282&PartNo=3](http://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=Taiwan&CategoryNo=85&No=282&PartNo=3)

[12] C.-Y. Su and W.-C. Kao, “Effective Demosaicing Using Subband Correlation”, IEEE Trans. Consumer Electronics, vol. 55, no. 1, pp. 199-204, Feb 2009.

