

國立交通大學

電機學院 電機與控制學程

碩士論文

根據雙圖場影像建構適應性混合式解交錯技術之研究

Adaptive Hybrid De-interlacing Technique
With Two Field Image Construction



研究生：李裕鴻

指導教授：董蘭榮 教授

中華民國九十八年六月

根據雙圖場影像建構適應性混合式解交錯技術之研究
Adaptive Hybrid De-interlacing Technique
With Two Field Image Construction

研究生：李裕鴻

Student： Yu-Hong Li

指導教授：董蘭榮

Advisor： Lan-Rong Dung



Submitted to College of Electrical and Computer Engineering
National Chiao Tung University
in partial Fulfillment of the Requirements
for the Degree of
Master of Science
in
Electrical and Control Engineering
June 2009
Hsinchu, Taiwan, Republic of China

中華民國九十八年六月

根據雙圖場影像建構適應性混合式解交錯技術之研究

研究生：李裕鴻

指導教授：董蘭榮 博士

國立交通大學 電機學院 電機與控制學程碩士班

摘要

在視訊解交錯的相關研究中，動作補償是許多高階的解交錯方法之一。由於動作補償在鏡頭的移動與物體的移動時，可以補償出較好的垂直細節。而邊緣線內插法則廣泛的應用於重建影像邊緣並增加邊緣線的細膩。為保有影像移動補償以及降低動作估計所需的記憶體並減少硬體代價，本文提出以雙圖場影像的適應性混合方法的動作補償解交錯技術，本文提出的解交錯是以雙圖場的動作估計補償與改良的邊緣線內插為主體的解交錯。本文以動作估計補償為主要的架構，從前一張影像做動作估計所得的移動資訊來補償，搭配移動向量與差異比對的動作適應性的決策法則，與改良的子像素邊緣線內插方法，會使得輸出的影像具有適應性的補償。適應性的決策法則利用其移動判斷資訊做進一步的動作補償的分類，再混合適當的補償。其適應性決策法以相鄰移動向量與差異值偵測比對動作的資訊，進而改善移動向量不準確之下，所造成的視覺上的缺失。整體來說，本文提出的架構，可由適應性的決策法則再搭配動作估計補償與改良的空間性邊緣線內插得到合適的影像。其實驗結果顯示，而實際的顯示效果也比一般解交錯方法有更進一步的提升。從 PSNR 的比較亦可改善其輸出效能約 2db，可比一般的動作補償解交錯有更好的影像顯示品質，其亦可減少多圖場動作估計的所需的硬體代價。

Adaptive Hybrid De-interlacing Technique

With Two Field Image Construction

Student : Yu-Hong Li

Advisor : Lan-Rong Dung

Degree Program of Electrical and Computer Engineering
National Chiao Tung University

Abstract

This thesis presents an adaptive deinterlacing algorithm that switches the deinterlacing policies between motion compensation and edge-directed interpolation. The motion-compensation-based deinterlacing is one of the most advanced approaches among deinterlacing techniques in that motion compensation can produce better quality on vertical details than the others. However, when the motion estimation is inaccurate the deinterlacing might result in blocking effects. The blocking effects can be dismissed by the edge-directed interpolation. Therefore, this thesis proposes an adaptive hybrid deinterlacing technique for two-field image reconstruction. The hybrid deinterlacing algorithm is based on two-field motion estimation and adaptive motion decision system. The adaptive motion decision system counts the variation of neighborhood motion vector (MV) and SAD to evaluate the error of motion estimation. This can help the deinterlacing to avoid from the interlace artifacts. In the edge-directed interpolation, this thesis presents the sub-pixel interpolation scheme to smooth the edge of object and reduce serration on edges. The experiment results show that the proposed algorithm can produce better video quality than conventional methods by 2 dB of PSNR while the computational complexity is low.

誌謝

承蒙恩師董蘭榮教授在論文研究中所給予我的指導與提攜，當我遭遇困難與瓶頸時給予我適時的指點，讓我得以突破難關，順利完成此論文。在此，獻上至高的謝意。

也非常感謝口試委員陳宏銘老師、林昇甫老師、林進燈老師，感謝您們在百忙之中撥冗參與並給予我精闢的建議，感謝你們寶貴的意見，使我的論文更加的完善。

感謝在週末時一起找教授的同學及學弟們，有你們的一起努力，使我感到研究生涯要更加的努力不懈及堅持。感謝學長盟淳、建發、江修，同學以群，學弟明豪、偉勇、啟彰等，感謝有你們的意見交流，讓我有個充實的研究生活。

同時我也要深深的感謝養育我的父母，讓我在求學過程中無後顧之憂，也要感謝我的女友淑君給予我的鼓勵與支持，謝謝您們！

隨著論文的完成，碩士班的研究生涯也接近尾聲了，一路走來感觸良多，這三年的時間彷彿就像是昨天的事而已，要感謝的人還有很多，僅將本篇論文獻給所有愛我及我愛的人，獻上由衷的感謝，謝謝大家！



目錄

| | |
|----------------------------|------|
| 中文摘要 | i |
| 英文摘要 | ii |
| 誌謝 | iii |
| 目錄 | iv |
| 圖目錄 | vi |
| 表目錄 | viii |
| 第一章 緒論 | 1 |
| 1.1. 研究動機與目的 | 1 |
| 1.2. 論文架構 | 3 |
| 第二章 解交錯技術文獻探討與研究 | 4 |
| 2.1. 交錯式影像與解交錯式概述 | 4 |
| 2.2. 線性內插 | 6 |
| 2.3. 非線性內插 | 11 |
| 2.4. 動作適應性補償 | 15 |
| 2.5. 動作補償 | 18 |
| 2.6. 時間遞迴解交錯濾波 | 22 |
| 2.7. 混合式解交錯演算法 | 23 |
| 2.8. 本章總結 | 26 |
| 第三章 雙圖場適應性混合解交錯演算法設計 | 27 |
| 3.1. 適應性混合解交錯架構改善 | 27 |
| 3.1.1. 減少動作估計的圖場 | 28 |
| 3.1.2. 輸出判斷決策改善 | 29 |
| 3.1.3. 整體架構 | 30 |

| | | |
|--------|-----------------------|----|
| 3.2. | 適應性影像補償之輸出決策 | 32 |
| 3.2.1. | 最近相鄰差異的移動判斷決策改善 | 32 |
| 3.2.2. | 移動區域邊緣向量誤差補償 | 34 |
| 3.3. | 子像素邊緣線內插 | 36 |
| 3.3.1. | 改良式邊緣線子像素內插 | 36 |
| 3.3.2. | 中心差異值比對法的邊緣線內插 | 37 |
| 3.4. | 動作補償解交錯之設定 | 42 |
| 3.5. | 雙圖場適應性混合解交錯演算法 | 45 |
| 3.6. | 邊界區域的補償改善 | 57 |
| 3.6.1. | 邊界區塊的移動向量誤差 | 57 |
| 3.6.2. | 改良邊界區塊的補償 | 58 |
| 3.7. | 本章總結 | 60 |
| 第四章 | 模擬結果與比較 | 61 |
| 4.1. | 影像效能評估 | 61 |
| 4.2. | HD測試影像序與結果分析 | 62 |
| 4.3. | CIF影像序測試結果與分析 | 69 |
| 4.4. | 計算複雜度統計表 | 84 |
| 第五章 | 結論與未來展望 | 87 |
| 5.1. | 結論 | 87 |
| 5.2. | 未來展望 | 88 |
| 參考文獻 | | 89 |
| 自傳 | | 93 |



圖目錄

| | |
|---|----|
| 圖 1 解交錯系統示意圖 | 5 |
| 圖 2 空間性內插與時間性內插法 | 7 |
| 圖 3 線性內插之頻率響應表示圖 | 8 |
| 圖 4 垂直時間性濾波器 | 9 |
| 圖 5 垂直時間性濾波下的快速移動殘影 | 10 |
| 圖 6 邊緣線平均內插演算法 | 12 |
| 圖 7 邊緣方向內插之"News"影像序比較 | 14 |
| 圖 8 一般影像動作偵測的影像差異 | 15 |
| 圖 9 同極性多圖場的動作偵測處理 | 17 |
| 圖 10 同極性動作偵測下的"Table Tennis"影像序比較 | 17 |
| 圖 11 循序與交錯掃描之垂直解析度頻率比較圖 | 18 |
| 圖 12 區塊比對法的動作估計 | 20 |
| 圖 13 動作估計誤差的動作補償結果 | 21 |
| 圖 14 三圖場的動作偵測解交錯架構 | 23 |
| 圖 15 使用Quad-pixel動作偵測圖 | 23 |
| 圖 16 四圖場動作估計的比對方法 | 24 |
| 圖 17 不同的四圖場動作偵測的架構 | 25 |
| 圖 18 多圖場動作估計搜尋區域 | 28 |
| 圖 19 不同的動作判斷機制 | 29 |
| 圖 20 適應性混合動作補償之解交錯主要架構圖 | 31 |
| 圖 21 臨近關聯性決策關係圖 | 32 |
| 圖 22 物體邊緣的移動向量種類 | 34 |
| 圖 23 影像序"Container"移動向量處理誤差的補償 | 35 |
| 圖 24 影像邊緣內插的改良 | 36 |
| 圖 25 子像素邊緣線內插流程圖 | 38 |
| 圖 26 邊緣線方向性 9+6 抽頭內插 | 39 |
| 圖 27 中間值內插與改良邊子像素內插的差異比對圖 | 41 |
| 圖 28 不同的邊緣線內插的效能比對圖 | 41 |
| 圖 29 宏塊大小之效能比較 | 43 |
| 圖 30 子像素圖場差比對法 | 44 |
| 圖 31 臨近關聯性判斷流程關係圖 | 45 |
| 圖 32 連續影像平移示意圖 | 47 |
| 圖 33 區塊比對的前後影像移動向量圖 | 47 |
| 圖 34 水平移動下的"Stefan"移動向量統計圖 | 48 |
| 圖 35 應用於解交錯的移動補償的輸出影像圖 | 50 |
| 圖 36 影像序"Stefan"人物移動區域對應之MV與SAD值 | 51 |

| | | |
|------|------------------------------------|----|
| 圖 37 | 影像"Foreman"的鏡頭物體移動下的差異值比較..... | 52 |
| 圖 38 | 影像"Foreman"的物體移動下的MV與SAD差異值比較..... | 53 |
| 圖 39 | 前後影像的區塊比對圖..... | 54 |
| 圖 40 | 前後影像的區塊比對移動向量圖..... | 54 |
| 圖 41 | 鏡頭縮放移動下的補償..... | 54 |
| 圖 42 | 動作形態與影像邊緣的處理流程圖..... | 55 |
| 圖 43 | 搜尋區域邊界區塊的補償問題..... | 57 |
| 圖 44 | 搜尋區域邊界區塊的移動向量..... | 57 |
| 圖 45 | 影像邊界的移動向量與鄰近位置關係判斷..... | 58 |
| 圖 46 | 解交錯效能比較方法..... | 61 |
| 圖 47 | 影像"Stockholm"輸出結果比較..... | 64 |
| 圖 48 | 影像"Stockholm"房屋邊緣細節的輸出比較..... | 65 |
| 圖 49 | 影像"Shields"輸出比較..... | 66 |
| 圖 50 | 影像"Parkrun"輸出比較..... | 68 |
| 圖 51 | 比對"Stefan"輸出與原圖影像差異..... | 71 |
| 圖 52 | 影像"Stefan"輸出結果比較..... | 72 |
| 圖 53 | 方塊比對誤差處理..... | 73 |
| 圖 54 | 影像"Foreman"邊緣內插的結果比較..... | 74 |
| 圖 55 | 比對"Foreman"輸出與原圖影像差異..... | 75 |
| 圖 56 | 影像"Foreman"第 15 張輸出結果比較..... | 77 |
| 圖 57 | 影像"Table Tennis"輸出結果比較..... | 78 |
| 圖 58 | 影像"Container"輸出結果比較..... | 80 |
| 圖 59 | 影像"Silent"輸出結果比較..... | 82 |
| 圖 60 | 影像"Flag"輸出結果比較..... | 83 |
| 圖 61 | 演算法的指令數百分比..... | 86 |

表目錄

| | |
|---------------------------------|----|
| 表 1 高解析度測試影像序內容分析 | 62 |
| 表 2 高解析HD解交錯影像的PSNR平均比較表 | 63 |
| 表 3 各種方法的PSNR平均比較表 | 69 |
| 表 4 連續影像”Stefan”的PSNR比較表 | 70 |
| 表 5 連續影像”Foreman”的PSNR比較表 | 76 |
| 表 6 指令數計算表 | 85 |



第一章 緒論

在目前解交錯技術中，提出具有動作估計補償的解交錯技術，同時降低對暫存記憶體的需求，增加移動向量的準確性，改善動作判斷的可靠度與邊緣線平均內插的效果，將是本文的研究解交錯技術的方向，本章將說明其研究的動機與目的，與文章組織。



1.1. 研究動機與目的

視訊解交錯的研究從早期的無動作補償的內插，到近期的多圖場動作補償(Motion Compensation)解交錯來還原影像，其中的動作估計是需要大量的記憶體與存取頻寬，多圖場動作估計又需要的大量的計算，因此發展有動作補償的效果的，且降低硬體需求的解交錯，其演算法就必須要從動作估計資訊中，適應性的變化補償方法，以期達較好的顯示品質。

為了重建遺失的影像，在[1]所介紹到一般常用的做法如線平均法(Line Average)，圖場合併(Field Merged)，垂直時間濾波器(Vertical Temporal Filter)，中間值權重濾波器(Weighted Median Filtering)等，這些方法都只是由線性內插出遺失的影像，傳統的內插並不是效能不好，只是這些方法是有使用限制的，因為它無法判斷影像移動的資訊。所以之後的發展是使用動作適應性補償 (Motion Adaptive)來決定影像為動態補償或靜態補償，其缺點是只能判斷出動態區域與靜態區域，靜態區域直接以圖場內插，但動態區

域無法判斷出移動的資訊量。所以比較高階的做法是動作估計補償，但動作估計補償是需要大量的資料傳輸，如果動作估計參考多圖場影像，其硬體代價與記憶體頻寬將會是相當大的，且動作補償應用在解交錯技術中，移動向量的準確性與可靠度是很重要的，如果影像變化很大時，導致移動向量不是很精確的話，就必須要有判斷機制判斷決定的補償方式。

本文利用單向的動作估計資訊，計算出影像的動作資訊，從物體的移動向量與臨近區域的動作比對，得到具有動作適應性補償的資訊，這樣可使得解交錯技術具有動作補償的效果，且可降低記憶體頻寬的需求與降低所需記憶體。而一般的動作估計補償解交錯，只利用動作估計得到的影像，再混合其他的方法，且在移動向量不可靠的狀況下，沒有判斷移動向量的正確與否，直接補償會使得動作補償的效果在解交錯應用中會變的比較差。然而物體的移動會是反應在移動向量上，比較結果的可靠度亦會反應在區塊的差異絕對值加總上，所以本文利用了動作估計比對結果來達到動作的適應性的判斷，可取代動作偵測無法判斷出影像的移動量之缺點，動作估計的資訊也可在讓影像內容變化較大的區域，決策出合適的補償，並由區塊判斷出動作區域與鄰近動作區域的關係，偵測移動物體的動作，可以判斷整體的移動，或區域的移動，並補償各種可能的動作種類。然而當影像快速移動的變化，致使影像無法從前一張影像補償回來時，也可判斷出，並改採用空間性的邊緣線內插改良。所以本文亦改良了一個空間性內插，以邊緣線內插方法為基礎，增加四分之一像素的邊緣線判斷，使物體邊緣線的地方能夠更加的平滑。

本文的研究目的為提出一個雙圖場動作估計的解交錯技術，由前一個圖場動作估計資訊中，找尋適合的補償方法，主要為利用相鄰近的移動向量差異，判斷出影像正確的動作資訊與判斷移動向量的可靠度，從有效的移動資訊補償，降低解交錯中使用動作估計的硬體代價。而在空間性的內插方法提出一個改良的子像素邊緣線內插技術，可針對移動快速的影像與邊緣線變化區域大的影像做補償，它增加了四分之一像素的邊緣線判斷，讓影像物體的邊緣可以更加平滑，並使用中間值差異的比較，增加邊緣線的內插效能，以得到較佳的邊緣內插品質。

1.2. 論文架構

在本文中介紹說明解交錯演算法的研究想法與建構，並模擬比較輸出的結果。

第一章 緒論

為本章所說明的簡介，說明研究動機與章節組織

第二章 解交錯技術的概述與相關研究

針對目前的解交錯演算法，分析其優缺點與效應，了解使用上的限制，與可能出現的問題與解決方法。

第三章 雙圖場適應性混合解交錯演算法

針對上述章節之問題，提出本論文的演算法，分析其移動補償的可靠度，與可改善的範圍，以及可行的解決要點，並對其演算法流程說明。

第四章 整體模擬結果之比對

效能分析與比較，針對上述之優缺點做結果分析，包含評估影像效能的實驗結果與影像的細節比較，最後計算所需要硬體上的複雜度。

第五章 結論與未來展望

分析說明整體的結果，並由分析結果分析訂出未來可進行的方向等。



第二章 解交錯技術文獻探討與研究

本章節將針對[2][3][4][5]中所揭露解交錯的各種演算法，分析其優缺點比較，因為解交錯的方法各有適合的使用條件與限制考量，使得解交錯技術有許多不同的改善。其方法可概分空間性內插，時間性內插，動作適應性補償，動作估計補償等。也說明目前交錯式影像相關的應用與解交錯的各種方法，了解目前解交錯研究問題的要點。



2.1. 交錯式影像與解交錯式概述

交錯影像(Interlaced)的標準格式在[6]有詳細介紹，主要問題為交錯影像在輸入是由場(Field)所構成，有分奇圖場與偶圖場，而解交錯方法是由場(Field)轉換為幀(Frame)，轉換完的交錯式影像稱為循序式(progressive)的影像，如 1080i 轉 1080p，1080 張場畫面轉 1080 張幀畫面。而轉換的方法又稱之為交錯式轉換循序式，轉換的方法是影像輸出最主要的品質關鍵。定義解交錯輸出的信號表示為(2.1)。

$$F_{out}(\bar{x}, n) = \begin{cases} f(\bar{x}, n), (y \bmod 2 = n \bmod 2) \\ f_i(\bar{x}, n), (otherwise) \end{cases} \quad (2.1)$$

$F_{out}(\bar{x}, n)$: 定義為輸出的解交錯式信號， n 是場的編號。

$f(\bar{x}, n)$: 輸入的第 n 影像場， $\bar{x} = (x, y)^T$: 為影像在 (x, y) 的位置上， T 是影像轉置。

$f_i(\bar{x}, n)$: 輸出為內插或是解交錯的第 n 影像場。

當奇圖場奇數線 $n \bmod 2 = y \bmod 2 = 1$ 條件下，輸出原始影像 $F_{out}(\bar{x}, n) = f(\bar{x}, n)$ ，奇圖場偶數線為 $n \bmod 2 \neq y \bmod 2$ 條件下，輸出 $f_i(\bar{x}, n)$ 內插的影像。而在偶圖場下 $n \bmod 2 = y \bmod 2 = 0$ 條件時，在偶圖場偶數線輸出原始影像。而 $n \bmod 2 \neq y \bmod 2$ 條件下，為偶圖場奇數線，輸出內插的影像，其解交錯示意如圖 1 所示。

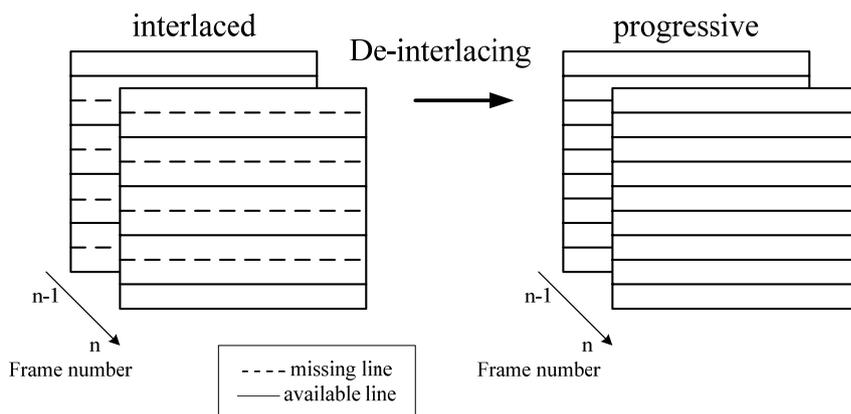


圖 1 解交錯系統示意圖

早期交錯(Interlaced)掃描影像的傳輸的優點，主要原因可以減少一半的傳輸頻寬與硬體的代價，如圖 1 的虛線即代表被抽掉的線。因為早期的電視系統並不需要額外的解交錯系統，配合著電視的掃描方式將接收到的影像就可逐一掃描顯示出，交織成影像。而掃描的影像線分成奇圖場與偶圖場，以兩個場畫面的交錯掃描而構成一個幀畫面，這樣的架構下，一般顯示的幀率(Frame Rate)是可以接受的，但在大片面積閃爍時，人眼對影像閃動會比較不靈敏[1]，但對細節的閃爍卻是較是靈敏的，所以傳統電視的架構整體掃描線的閃爍是可以接受的，因為人眼感受整體閃爍的感覺比較不明顯。

由於影像來源與顯示設備的關係，循序式顯示器直接顯示交錯的信號將會出現閃爍的問題，所以有不同的交錯到循序掃描的轉換被提出。以平面 TFT-LCD 與數位投影機之進展，現在有多數的顯示設備都是循序掃描，但以電視廣播系統的標準依然是交錯掃描式，且很多的影片記錄格式也都是以交錯式影像記錄著。在相同頻寬下，影像解交錯後就可以得到較高的解析度，來源所需要的傳輸的頻寬也可以比較低，所以只要解交錯系統品質優良，其就能有效降低影像閃爍，還原影像將是可以被接受的。

整體來說交錯式影像可降低頻寬，交錯掃瞄利用人眼對整體頻率變化的敏感度較細節頻率變化的低，以奇偶交錯的圖場，掃描出一張畫面，讓人眼比較感受不到整體的閃動，又可降低頻寬。但為配合現在的顯示設備，影像解交錯的基本內插方法又分底下幾種，有空間性內插方法，時間性內插方法，垂直時間濾波器，邊緣線平均方法，混合解交錯法，動作適應性補償，動作估計補償等，底下將針對這幾類補償說明。

2.2. 線性內插

基本的解交錯演算法是由線性內插所構成，本節將對使用線性內插的解交錯技術的方法做說明。包括有空間性的內插，時間性的內插，以及垂直時間內插，然後再延伸到非線性內插，動作適應性補償、動作估計補償、時間遞迴解交錯與混合解交錯法等。

(1) 空間性內插



空間性內插(Spatial Interpolation)，又稱圖場內部內插(Intra-Field Interpolation)，以複製內插(Line Double)或平均內插(line Average)來說，都是空間上的內插，它雖然是很容易實現的方法，但缺點是在影像內容解析度較高的影像會造成解交錯影像的模糊。如圖 2 所示，以 Line Average 為例，取得相鄰的上下點做平均而得，說明如下。

其關係式定義如(2.2)

$$f_{spatial_average}(x, y) = \frac{f(x, y-1) + f(x, y+1)}{2} \quad (2.2)$$

$f_{spatial_average}(x, y)$: 為在 (x, y) 位置的空間性內插的輸出。

$f(x, y-1)$: 為輸入的交錯影像場。

式(2.2)中的，它是由上一條線的像素 $f(x, y-1)$ 加上下一條線的像素 $f(x, y+1)$ ，然後再平均得到的，其所使用的資料是包含了上下兩條線的資料，所以它需要使用到線暫存器(Line Buffer)，來內插出所需要的資料。

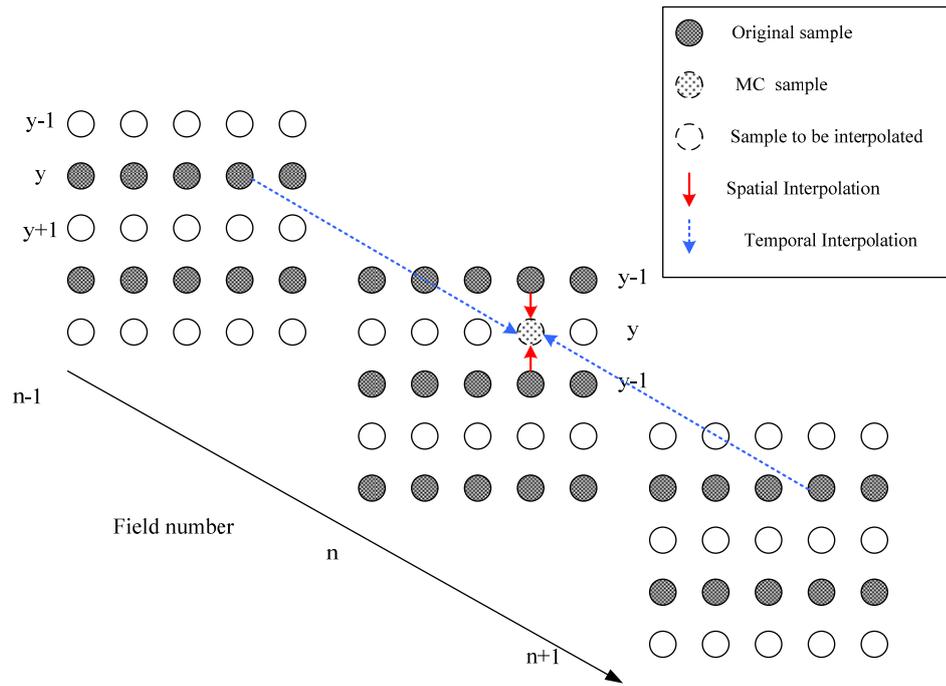


圖 2 空間性內插與時間性內插法

(2)時間性內插

時間性內插(Temporal Interpolation)又稱圖場之間內插(Inter-Field Interpolation, FI)，它的變化有合併法(Merged)或是時間平均法(Temporal Average, TA)，它是取前後場的資料做平均所得，但缺點是影像為動態時，會造成很嚴重的影像殘影(Artifact)或是鋸齒邊緣(Serration)，但影像為靜止的狀況下，前後圖場之合併是最好的內插法，可以完整還原影像的細節，不像空間性內插法會降低解交錯後的影像的品質，如圖 2 所示。

時間平均法的關係式如(2.3)。

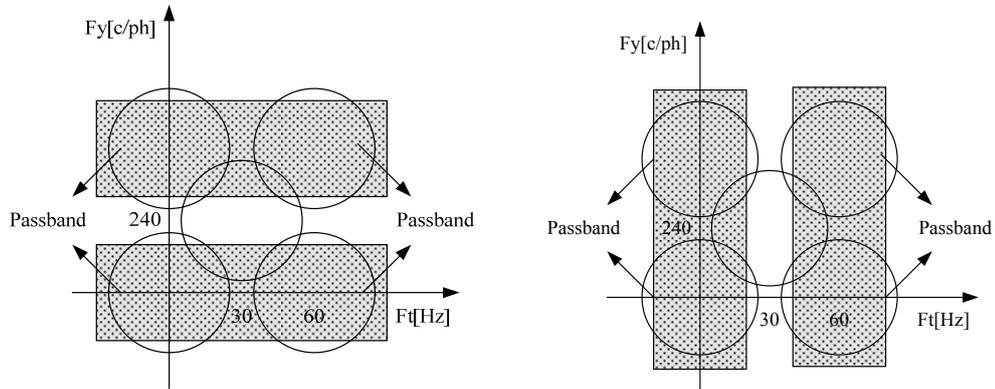
$$f_{temporal_average}(x, y, n) = \frac{f(x, y, n-1) + f(x, y, n+1)}{2} \quad (2.3)$$

$f_{temporal_average}(x, y)$: 為在 (x, y) 位置的時間性內插的輸出。

$f(x, y, n-1)$: 為前一個畫面中 (x, y) 的位置的值。

$f(x, y, n+1)$: 為後一個畫面中 (x, y) 的位置的值。

式(2.3)中由前一張畫面中 (x, y) 的位置，與後畫面的位置資料做平均得到的。它與空間性的內插不同的是它參考的資料是前影像場與後影像場，不是在現在的圖場中，所以它需要圖場記憶體來儲存所要的影像，其大小依照影像而定。



(a) 空間性內插(Spatial Interpolation) (b) 時間性內插(Temporal Interpolation)

圖 3 線性內插之頻率響應表示圖

由線性內插頻率響應分析圖 3 所示[1]，可以了解其效能是有限的。由圖 3 的 x 軸中的 $F_t[\text{Hz}]$ 是代表影像更新率(Frame Rate)，圖 3 的 y 軸為影像的垂直掃瞄出現的掃瞄線週期(Cycle per Picture Height, CPH)，Pass-band 代表空間性內插法會讓信號通過的頻率，圓圈代表交錯影像的頻譜。

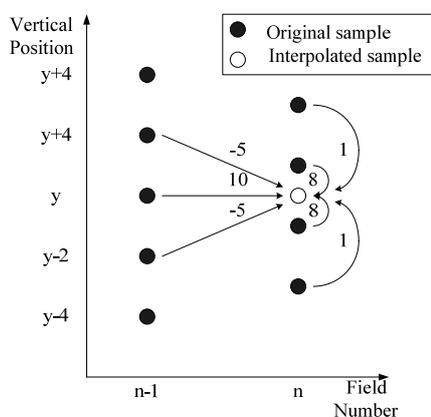
由圖 3 (a) 為空間上的內插，可得知垂直解析度會的頻率響應無法在 $F_y[\text{cph}]$ 的高度上有所提昇，會被限制在一定的高度，這也代表垂直方向的解析度無法很高，所以垂直細節將被空間濾波給平均掉，解析度無法提高，且 Pass-band 在時間上也無法分離開中間的圓圈。而圖 3 (b) 時間性內插在 y 軸的 $F_y[\text{cph}]$ 雖然可以保留較多的垂直細節，但畫面有變化，或是變化速度較快的時候，因為圖場是以平均的方式，所以 x 軸中的 $F_t[\text{Hz}]$ 就無法達到比較高的分辨率。

(3)時間與空間的內插法

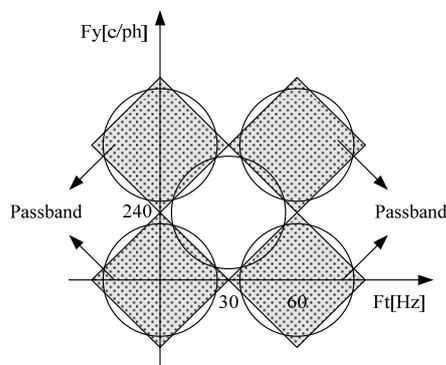
時間與空間的內插法是一個利用前後圖場及上下點做權重濾波的一種方式，是混合時間與空間內插數值的方法。它在靜態的狀況下能有效的補償到適當的值，但動作快速及解析度較高等變化快的高頻影像其畫面通常有會不自然的殘影。以垂直時間濾波解交錯(Vertical-Temporal filter)來說，它採用了數值與位置的權重的關係上，如果它在時間上是低頻的，可藉由前面場的資訊抵消權重，使其成為空間性內插。否則時間的變化是高頻的，其前後場的影像是快速移動的影像時，其輸出會是前後場的平均來得到影像。垂直時間濾波器使用之位置與權重關係如(2.4)。

$$f(x, y, n) = \begin{cases} \frac{8}{18} f(x, y-1, n) + \frac{8}{18} f(x, y+1, n) + \frac{1}{18} f(x, y-3, n) + \frac{1}{18} f(x, y+3, n) + \\ -\frac{5}{18} f(x, y-2, n-1) + \frac{10}{18} f(x, y, n-1) + \frac{-5}{18} f(x, y+2, n-1) \end{cases} \quad (2.4)$$

它的係數可以抵消畫面間垂直與時間方向的變化，使其正確內插。式(2.4)中的 $F(x, y, t)$ 是代表內插的位置點， n 為圖場， y 為垂直方向的位置，當 $n=0$ ，取權重分別為 y 為方向在 $y=-3$ 、 $y=-1$ 、 $y=1$ 、 $y=3$ 的位置，分別乘上權重，再加上前一個畫面的資料與係數，再將這些值的和除上總權重 18 總可得到 $F(x, y, n)$ 位置的內插值，如圖 4 (a) 所表示。



(a) 垂直時間濾波的權重平均

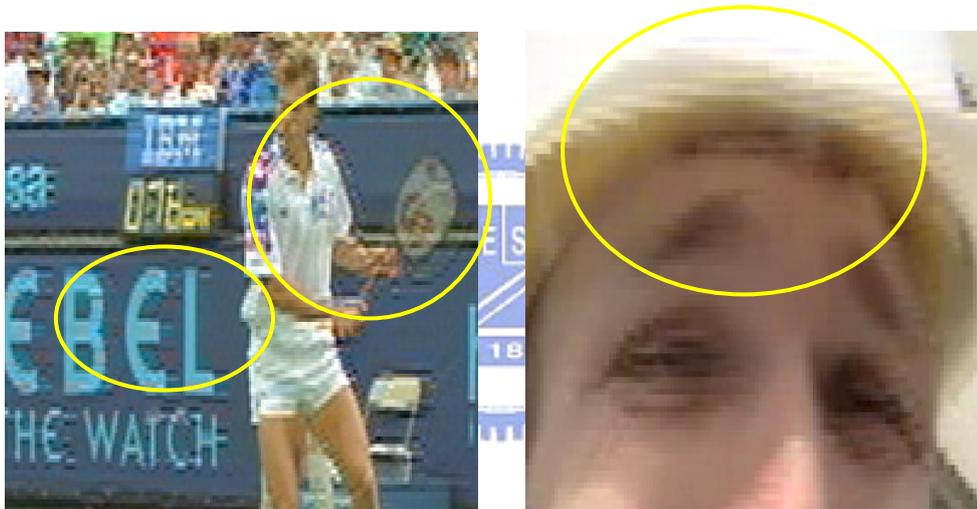


(b) 垂直時間濾波頻率響應

圖 4 垂直時間性濾波器

因為前一個圖場在 $y = -2, y = 0, y = 2$ 位置的資料可以抵消掉，代表其影像的變化是低頻的，也就可以讓現在圖場的 $y = -3, y = -1, y = 1, y = 3$ 位置資料做內插，就會是空間上的內插，但如果影像移動快速，影像間會因為權重無法相互抵消掉，其代表的是高頻信號，結果會是使用前後場影像，為有權重相加起來的時間平均影像。

由垂直時間濾波器的頻譜分布如圖 4 (b)可知，x 軸中的 $F_t[\text{Hz}]$ 是代表影像更新率，圖的 y 軸是影像的垂直大小會出現掃瞄線次數(CPH)，它的頻譜分布不會有空間性內插或時間性內插的頻譜混疊的現象，輸入影像完全落在 Pass-band 上，是所有線性處理中，頻譜分析看來比較好的一種方法，但實際上移動邊緣還是會有些殘影。



(a) 垂直時間濾波下的 Stefan (b) 垂直時間濾波下的 Foreman

圖 5 垂直時間性濾波下的快速移動殘影

以圖 5(a)為例，動態畫面左右移動時，前後場差異較大的時間濾波器因為包含了前場資料的加權，很容易出現影子狀產生，這是因為影像前後場的差異太大，前場資訊沒法相互抵消所造成的，但在靜態畫面下，垂直時間濾波(VT Filter)則會是以平均內插為主的內插。普遍來說，這類型前後影像權重式內插都會因為移動或是物體邊緣，都會造成這種現象，圖 5 (b)所示的也是相類似狀況，是上下移動的影像補償不連續。

雖然理論上垂直時間濾波器(VT Filter)是線性內插方法中，效果較好的方式，但是

實際上會因為它是取加權再平均，其結果會與取樣的點數，取樣的權重，甚至於是取前後圖場的變化影像再做加權處理，但結果是會因為影像的移動，或是垂直解析度很高，無法完全回復，甚至於一些物體的邊緣或是影像水平移動的，都無法從有效的回復，亦會造成一些殘影存在，也因為平均的關係，所以空間上的細節還是會有所損失。

總結線性內插方法與技巧後，線性的內插方法主要是直接對影像做運算，空間性的內插在低頻的信號有良好的效果，它不會造成空間性影像不連續的問題。而時間性的內插雖然可以提高影像解析度，但在移動的影像下，沒辦法有效的對應到空間上的連續，所以就必須考量影像高頻或是移動的判斷等其他因素。線性補償是交錯式影像內插的基礎，雖然是簡單，但因為有限制使用方法，並不是每一個方法都適用在相同狀況下。所以如果沒有正確的判斷，其補償輸出影像時就會出現影像不連續的狀況產生，所以為使影像有更好的解交錯結果，下一章將介紹非線性內插法的解交錯。



2.3. 非線性內插

非線性內插包含有邊緣線平均、中間值濾波、動作適應性、動作補償等。非線性內差方法有加入了判斷的條件，在條件下可得到良好的補償，但還是有其缺點的。本節將針對邊緣線平均與中間值濾波的優缺點說明，與適合的使用條件。

(1)邊緣線方向內插

邊緣線方向平均處理(Edge Line Direction Average)是影像處理中針對邊緣線導向的內插演算法，專門處理邊緣線平均內插，可以計算出影像物體的輪廓或邊界，所以正確判斷邊緣方向就很重要。邊緣方向做差異判斷是決定方向的關鍵，判斷影像物體的邊緣方向後，再依照這的方向的取平均得到內插值，再補償影像邊緣到對應位置上。圖 6 使用了 3×3 的 ELA window 來做說明。

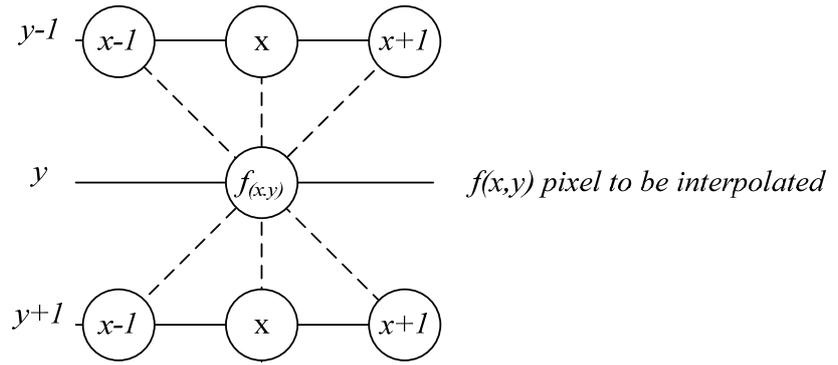


圖 6 邊緣線平均內插演算法

圖 6 所表示是一個 3×3 的 ELA 邊緣線判斷方法，中間的 $f(x,y)$ 是內插影像，以 3×3 說明，所以使用的資料只有上下兩條線的資料。在圖 6 中的 $f(x,y)$ 代表所要內插影像的位置，在式(2.5)中表示之 a 、 b 、 c 分別代表不同的方向，以角度來說明是 -45 度、0 度、45 度角方向，其中 a 為 -45 度角方向的差異絕對值，使用上下兩條線的對角兩點資料，分別是 $(f(x-1, y-1), f(x+1, y+1))$ 來做差異計算。 b 為 0 度角方向的差異絕對值，由 $(f(x, y-1), f(x, y+1))$ 的差異所組成， c 為 45 度角方向的差異絕對值，由 $(f(x+1, y-1), f(x-1, y+1))$ 的差異值所組成。個別的邊緣方向差異計算如(2.5)。

$$ED(d) \Rightarrow \begin{cases} a = |(f(x-1, y-1) - f(x+1, y+1))| & , d = -1 \\ b = |(f(x, y-1) - f(x, y+1))| & , d = 0 \\ c = |(f(x+1, y-1) - f(x-1, y+1))| & , d = +1 \end{cases} \quad (2.5)$$

最小差異值的邊緣方向關係式如(2.6)。

$$d = \text{Arg} \min_{-1 \leq d \leq 1} ED(d) \quad (2.6)$$

邊緣線方向平均內插關係式如(2.7)。

$$f_{ELA}(x, y) = \frac{f(x+d, y-1) + f(x-d, y+1)}{2} \quad (2.7)$$

在計算完個別方向的差異值後，其 a 、 b 、 c 中最小的值為該邊緣方向，它代表的位

置為補償的邊緣方向。如果影像計算完的差異值代表的方向為-45 度角，代表其位置資訊是-1，再將其對應的資料做平均內插。當比較的角度比較少，最小值選擇輸出可直接用邏輯運算來判斷來決定大於或等於，然後挑選出中間值，但角度較多的邊緣也可使用排序來決定大小，進而得到差異最小的值。

(3)邊緣線方向內插與中間值濾波

邊緣線只計算差異會很容易誤判的，因此會在邊緣線內插計算後加入中間值濾波，稱為邊緣線的中間值濾波處理(ELA Median filter)，相關研究如 C.J. Kuo 的[7]與 J. Salo 的[8]，與 C. Hentschel 的[9]，可避免差異值的計算結果不是真的邊緣，還可以讓中間值濾波將計算錯誤的值排除掉。邊緣線與中間值濾波表示式如(2.8)

$$f_{ELA+Median}(x, y) = Median\{f(x, y-1), f(x, y+1), F_{ELA}(x, y)\} \quad (2.8)$$

$f_{ELA+Median}(x, y)$:邊緣線內插與中間值濾波輸出。

$Median\{\}$:中間值濾波處理。

而中間值濾波(Median Filter)是非線性濾波技術，它對於斑點雜訊和椒鹽雜訊的過濾相當有用。而在解交錯邊緣線內插中，它的角色算是一個限制器，可避免內插值錯誤所產生的鋸齒邊，如果與邊緣內插值一起比較的是 $f(x, y-1), f(x, y+1)$ ，那麼該邊緣內插就會被限制在 $f(x, y-1), f(x, y+1)$ 內，可依設計的形式不同給予的對應的值，但最常用的邊緣線內插只參考 $f(x, y-1), f(x, y+1)$ 上下兩點來限制，如果同時加入了很多參考影像反而會沒效果，如 J. Salo 的研究[8]中就有提出多種的中間值濾波的處理方式。

在 M.K. Park 的研究[10]中，其邊緣線平均內插則是使用角度的圖形樣板來比對可能的邊緣，並使用不同的偵測法，但影像邊緣的變化較多樣化，必須有多角度的邊緣樣板比較，才有比較好的補償。而在 M.H. Lee 的研究[11]中多使用上下第三條線資料做運算，用 $f(x, y-3)$ 與 $f(x, y+3)$ 計算更多的差異比較，但 $y = -3, y = +3$ 離要

內插的位置太遠了，實際的內插以 $y = -1, y = +1$ 關係較接近，所以使用到 $y = -3, y = +3$ 的線資料，其參考價值小於鄰近的上下點，效果上也不有有很大的提昇。

相關研究還有以 F. Michaud 的 [12] 研究利用模糊法則偵測邊緣線以及 H.S. Oh [13] 的混合時間與空間的邊緣線方向內插補點。但沒有對其方向做內插，其影像也會有鋸齒邊，又因判斷條件在差異值相同時，或是複雜變化或細膩的圖形時容易判斷錯誤。為了預防方向判斷的錯誤，邊緣方向內插法都通常加入上下像素做中間值濾波，確保當影像是跟上下關聯性較大時，不至於內插錯誤值，但缺點則是會降低解析度。

以圖 7(a) 以 News 影像背景為例，說明 ELA 與中間值實際的內插效果。使用邊緣線內插 (ELA) 內插得到的圖片，在人物的手上使用 ELA 如果沒有判斷正確則會出現圖 7(b) 人為的加工 (artifact) 的雜訊，加入了中間值濾波雖然會改善，但其結果如圖 7(c) 還是有較粗糙的顆粒像素，無法很平順。



圖 7 邊緣方向內插之”News”影像序比較

雖然邊緣方向內插方法在邊緣或線條像素內，可以得到良好的表現，但缺點是當畫素的關連性要是不高，如背景的高頻，會在靜態畫面中內插模糊的畫素，甚至於比直接平均的效果差。加上中間值邊緣線內插的方法，雖然補的畫面由單張看起來還不錯，但連續影片播放卻會出現閃動 (flicker) 的現象，會有固定的頻率在，所以這樣的結果也會使這方法有了使用上的限制。

2.4. 動作適應性補償

動作適應性解交錯(Motion Adaptive Interlaced-to-Progressive conversion, MA-IPC)，它是直接計算影像的差異，經由臨界值取得動態與靜態，由於無法得到移動的資訊，所以在動態補償的效果有限。因為它只計算動作差異(Motion detector, MD)，但沒有計算動作的資訊。其動作偵測定義如下。

定義動作偵測如(2.9)。

$$MD(x, y, n) = Threshold \{ f(x, y, n) - f(x, y, n-1) \}, (y \bmod 2 \neq n \bmod 2) \quad (2.9)$$

動作適應性補償關係式如(2.10)。

$$f_{MotionAdaptive}(x, y, n) = \alpha F_{still}(x, y, n) + (1 - \alpha) F_{motion}(x, y, n) \quad (2.10)$$

由 $MD(x, y, n)$ 為影像動作偵測(Motion Detect)式(2.9)輸出， $Threshold\{\}$ 為臨界值濾波， $f(x, y, n)$ 為現在圖場所要比對動作的位置， $f(x, y, t-1)$ 為前一個圖場所要比對動作的位置，定義 $MotionDetector(x, y) = \alpha$ 值， $f_{still}(x, y, n)$ 為靜態所要使用的方法， $f_{motion}(x, y, n)$ 為式(2.10)動態所使用的方法，以動作偵測到的 α 值決定輸出為補償方法。由此可知臨界值的選取很重要，因為會影響動作偵測後的補償的好壞。

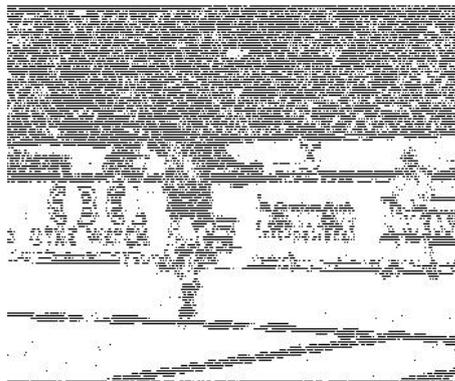


圖 8 一般影像動作偵測的影像差異

在 T. Koivunen 的研究中[14]中使用動作偵測器來計算影像差異，會直接計算點對點的圖場差異，但如果是高頻區域的小部份移動，如圖 8 中的背景，動作偵測器會把高頻內容當成也是移動的區域，且這樣的差異無法使用濾波器過濾高頻誤差，且空間性內插補償，也會使原本的高頻影像變的相當模糊，雖然動作偵測器可以設有臨界值可以調整，但臨界值的設定往往跟影像的內容有關，比較無法全面性的符合各種影片，也沒有辦法分辨出影像差異的移動量等。

然而 $MD(x, y, n)$ 是影像差異再由臨界值運算得到，臨界值的設定是決定動作偵測的關鍵。以圖 8 表示動作偵測經過臨界值計算的輸出，如果設成固定的臨界值，動作偵測將會因為影像內容而有不同，會致使輸出不一定可以正確偵測到。所以也有使用多圖場的動作偵測來確認圖場之間的差異值是否大於臨界值，才決定是否為動態影像，避免參考的動作偵測是錯誤的。當動作偵測畫面差異值時，小於臨界值的會被當成是靜態的，超過臨界將會把判斷結果設為動態的部份。但臨界值的上下，如果有設範圍，則由 α 來決定動態與靜態畫面所佔有的比例。而計算好偵測到的動作區域後，依其動作區域 α 值的設定來內插方式， α 值的設定一般界於 $0 < \alpha < 1$ 間，如果當 $\alpha=1$ ，輸出 $f_{still}(x, y, n)$ ，是靜態則使用時間性內插。如果 $\alpha=0$ ，輸出 $f_{motion}(x, y, n)$ ，是動態影像則使用空間性內插。所以對一個動作偵測器來說，臨界值的設定會很重要，且會與影像內容有關。

其他如 C. Hentschel 的研究中[15]都有使用前後多圖場來做動作偵測的濾波器，如圖 9(a)所示，偵測後才依其特性選擇適合的內插，其相關研究亦有使用四到五個圖場來偵測影像的動作差。再設計其匹配的濾波器，來補償動態與靜態影像的部份，所以臨界值的設定會是動作偵測的關鍵因素。

雖然多圖場偵測的有加入水平方的移動偵測，如圖 9(b)所示。由於影像的移動範圍不一定，水平的移動距離的偵測範圍太小會沒有其效果，但太大的偵測範圍的偵測誤差會很大，這樣會使得多圖場的水平偵測只能在設計的範圍內偵測，其效能也是會被限制的，使得動作偵測在靜態的影像下會比較有效果。

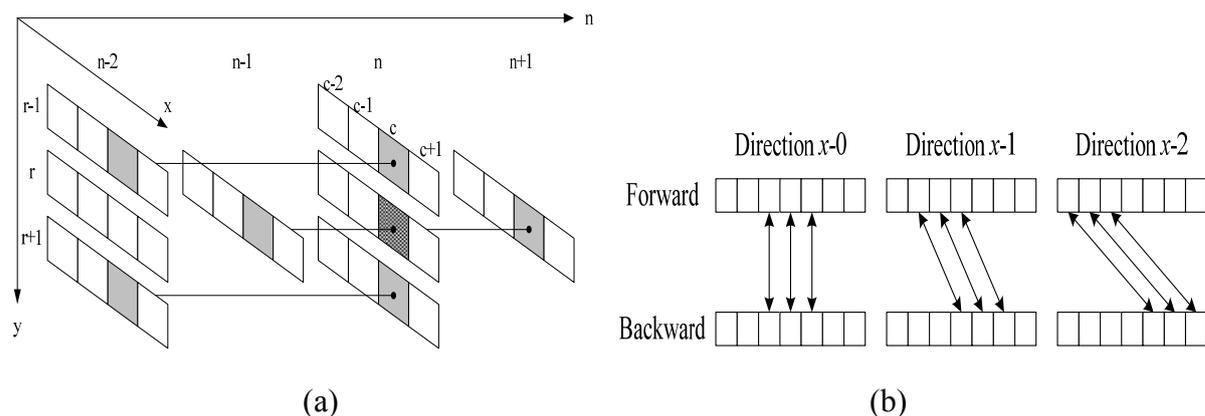


圖 9 同極性多圖場的動作偵測處理

圖 10 是使用同極性的 $f(x, y, n-1)$ 與 $f(x, y, n+1)$ 的差異所偵測的輸出結果，因為乒乓球的變化在每一張畫面下都在不同位置上，且移動都超過乒乓球形體大小的距離，所以會無法偵測到是否為動態，而當成是靜態區域進而產生出影像重疊的樣子。

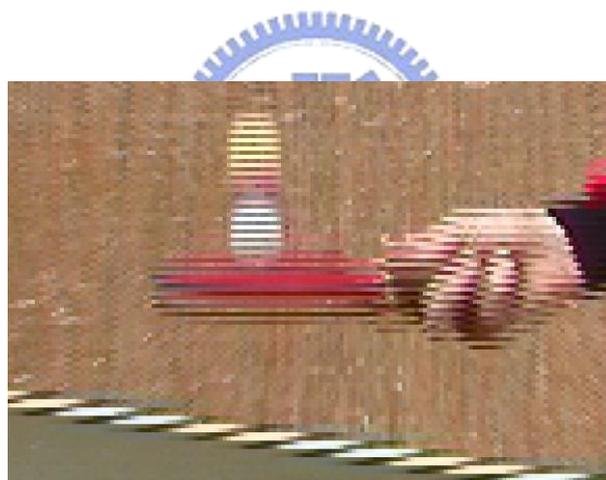


圖 10 同極性動作偵測下的”Table Tennis”影像序比較

整體來說動作適應性(MA-IPC)的方法，主要是將畫面區分為兩部份，一個是偵測靜態畫面，一個是偵測動態畫面，由一個動作偵測器來決定現在是動態畫面或是靜態畫面。但只有動作偵測是不夠的，因為它的偵測不包含動作移動量的資訊，所以後來的研究以動作估計，來計算移動量，不只有偵測的區域是否為靜態畫面或動態畫面。所以要有更好的效果，需加上動作估計補償，再應用動作估計補償在解交錯影像上。底下將進一步的介紹比較高階的動作估計補償。

2.5. 動作補償

由於動作偵測器(MD)並不是真實的找到正確的移動走向，而是只有判斷動態與靜態，為了要找到影像實際的移動走向，必須使用動作估計(Motion Estimation)來計算移動向量，再進行動作補償(Motion Compensation)。有動作估計才能由前場畫面找回原本在影像附近的資料，並提昇垂直的分辨率。但動作補償會因為移動估算的準確性而影響好壞，且比對到的移動向量是否合適?，如果移動向量準確度不高或是物體變形大，將導致補回的影像會發生不連續的狀況，或是移動區塊邊緣的效應，這也都是動作補償所要注意的。

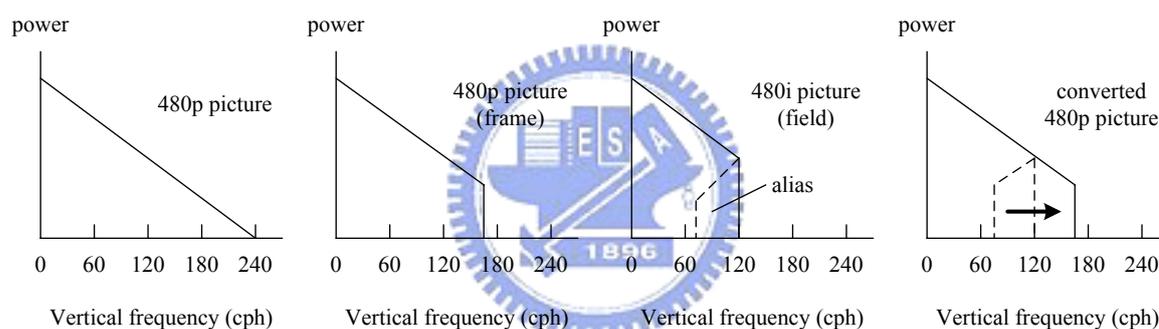


圖 11 循序與交錯掃瞄之垂直解析度頻率比較圖

動作估算補償的好處是因為它可以增加影像解析度。由 K. Sugiyama 與 H. Nakamura [5]的研究中，垂直解析度對應交錯式影像中以圖 11 表示，垂直頻率分布可明顯看出。以 480p 循序掃瞄為例，圖 x 軸是影像垂直掃瞄週期(Cycle per Picture Height, CPH)。加入動作估算補償的解交錯可以增加其垂直方向的解析度，如果只有相鄰圖場(intra-field)的解交錯並無法增加其垂直方向的解析度，因為一般線性內插得到的效果會是只有一半的垂直解析度(120p)，就算是適應性補償(Motion Adaptive)其靜態畫面下，垂直解析度可以達到 170cph，水平動作與垂直動作都無法有效提昇。而使用動作補償會得到比較高的垂直方向的分辨率，並可清楚了解，動作補償的垂直分辨率除了垂直方向的動作的分辨

率會依內容而不同，其他都可以達到比較高的影像分辨率。讓原本的交錯式影像其垂直解析度是循序式(240p)的 70%(170p)。

為使解交錯的系統中的動作估算比對的移動向量有較高的準確性，所以解交錯系統是使用區塊比對的方法計算移動向量，而較少使用像素基底的動作估計。不同的動作估算的技巧，有梯度計算(Gradient Techniques)，像素遞迴(Pel-recursive Techniques)，還有區塊比對方法(Block Matching Techniques)。像素遞迴(Pel-recursive)的動作估計可使影像得到個別的像素最佳移動向量，每一個像素都有自己的移動向量，但實際上，像素比對像素中，要比對到相類似的像素會有太多有著相同的值，且還有雜訊也會影響移動向量的準確性，因為它的計算有因果關係的，且計算的複雜度也相當高。

比較熱門的動作估計比對為區塊比對法(Block-based Matching Algorithm, BMA)，區塊比對的優點是可以得到較準確的移動向量(Motion Vector, MV)，計算複雜度也是可接受的。影像動作估計計算移動向量的方法廣泛應用在很多動態影像處理上，在 F. Dufaux [16]研究中有說明幾種視訊常用技巧。在動作估計動作中，還需要一些規格來設定所需要的參數。動作估計的幾個基本條件，如圖 12 所表示，包含了搜尋法則、比對的條件、搜尋範圍、宏塊大小等。由於完全搜尋(Full Search, FS)會完整的比對所有的區塊，並計算出最小差異的區塊，得到最準確的移動向量。而三步搜尋(Three Step Search, TSS)、新三步搜尋(New Three Step Search, NTSS)、四步搜尋(Four Step Search, FSS)、鑽石搜尋(Diamond Search, DS)等，都是屬於快速搜尋的方法，是以差異分布的原理，快速找到移動向量。

但動作估計應用在解交錯上，最主要是要找出不同圖場所要內插的像素。因此藉由比對前一個圖場來得到相關聯的像素，如圖 12 所示，由於交錯的影像在移動時，其細節會因此消失，為了找回因為移動而消失的影像細節，所以使用現在的影像區塊比對到前一張影像區塊，來找回缺少的移動影像。而影像的移動量稱為移動向量，找出其前後場可重複使用的區塊，動作補償時再把前一張圖場最相似的區塊補償現在的區塊內。搜尋移動會有一個範圍，稱為搜尋範圍(Search Range, SR)。

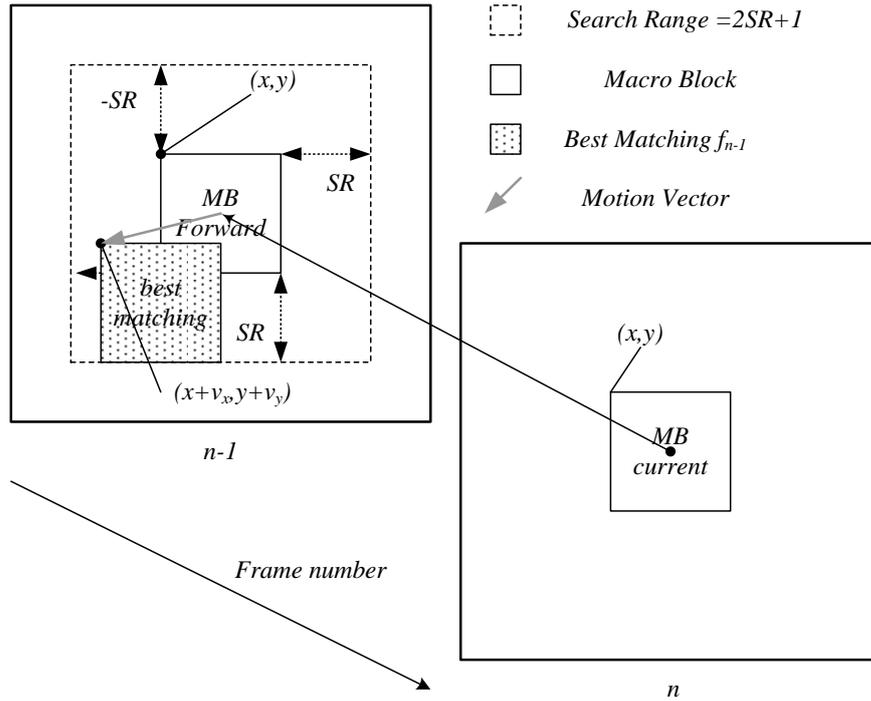


圖 12 區塊比對法的動作估計

而式(2.11)定義移動估算中的移動向量比對方法，使用差異絕對值加總(Sum of Absolutely Differences, SAD)方法。式(2.11)中比對的區塊大小為 $M \times N$ 。

定義在 (x,y) 位置上計算的 $\mathbf{MAD}(v_x, v_y)$ 關係式如(2.11)。

$$\mathbf{MAD}(v_x, v_y) = \frac{1}{MN} \sum_{j=0}^M \sum_{i=0}^N |f_t(i, j) - f_{t-1}(i + v_x, j + v_y)| \quad (2.11)$$

$\mathbf{SAD}(v_x, v_y)$ 關係式如(2.12)。

$$\mathbf{SAD}(v_x, v_y) = MN \cdot \mathbf{MAD}(v_x, v_y) \quad (2.12)$$

範圍內最小的 $\mathbf{SAD}(v_x, v_y)$ 即代表 \mathbf{MV} ，其關係式表示(2.13)。

$$\mathbf{MV} = \text{Arg min}_{v_x, v_y} \{ \mathbf{SAD}(v_x, v_y) \} \quad -SR \leq v_x, v_y \leq SR \quad (2.13)$$

式(2.12)中在計算完搜尋範圍內最小的 $\text{SAD}(v_x, v_y)$ 後，該區塊的移動向量可表示成 $\text{MV}(v_x, v_y)$ ，而 $\text{SAD}(v_x, v_y)$ 為該位移的差異值， SR 為搜尋範圍， $F_t(x, y)$ 為比對的區塊，移動向量輸出後的對應的影像補償為(2.14)式，補償的區塊為移動向量指到的位置，補償到現在的位置上。

定義動作補償輸出 $F_{MC}(x, y, n)$ 的關係式如(2.14)。

$$f_{MC}(x, y, n) = f(x + v_x, y + v_y, n - 1) \quad (2.14)$$

避免快速搜尋可能會犧牲一些準確性，所以使用全區搜尋的動作估計而在解交錯中可降低移動向量的錯誤。如果動作估計是有誤差或不準確，則會有圖 13(a)的狀況，或者是影像中無法找相似的區域時就會像圖 13(b)一樣，補償到的區塊會是不連續的。



(a) Foreman 之區塊不連續

(b) Stefan 之區塊不連續

(c) Silent 之區塊不連續

圖 13 動作估計誤差的動作補償結果

此外為解決解交錯中的動作估計誤差，或是錯誤，如圖 13 所示，許多解交錯採用動作補償中間值濾波(MC Median Filter)可以加以過濾這樣的誤差，如式(2.13)，但這種方式就會變成與邊緣線平均(ELA)或是中間值濾波的效果，但其影像品質也會因為加了中間值濾波而下降，如果不加以判斷該區域的動作估計可靠度或誤差，雖然直接使用中間值濾波輸出結果不會因為動作估計誤差而有不連續區域存在，但加中間值濾波於動作補償中其品質會下降，否則必須要有另外的機制判斷，以確保其品質不會下降。

動作補償之中間值濾波器如式(2.15)

$$f_{MC_Median}(x, y, n) = Median\{f(x + v_x, y + v_y, n - 1), f(x, y - 1, n), f(x, y + 1, n)\} \quad (2.15)$$

$f_{MC_Median}(x, y)$ 是動作補償加上中間值濾波輸出， $f(x + v_x, y + v_y, n)$ 是動作補償的畫面，加入了 $f(x, y - 1, n)$, $f(x, y + 1, n)$ 上下影像，限制輸出的結果要在要在 $f(x, y - 1, n)$ ， $f(x, y + 1, n)$ 範圍內，使輸出不會因為移動向量的誤差而造成影像的不連續的變化，至少輸出在快速移動的狀況下，不會有錯誤。但輸出這樣的狀態在移動區域內容為高頻時，就會因為補償的不定而出現閃爍，也無法增加解析度，因為當影像是移動的，動作估計補償的輸出會被限制在 $f(x, y - 1, n)$ 與 $f(x, y + 1, n)$ 內，如果影像內容都不在限制內，則會造成選擇的影像不固定，這樣會使輸出都是上下資料居多，效果會比較差。

2.6. 時間遞迴解交錯濾波



時間遞迴解交錯(Time-Recursive Filter)參考資料的元素由前場已經補回的資料做動作估計，但缺點為如果前場變化很快，有可能前場已經是錯誤的資訊，那再併入錯誤的資訊將會更無法找到正確的值，在FM Wang的研究中[17]，納入前面以經解完交錯的圖場到現在決策中當參考資料做動作估計，如式(2.16)，原理與動作補償(MC)很相似，也會有中間值的濾波的保護。

時間遞迴解交錯關係式如(2.16)。

$$f_{TR}(x, y, n) = Median\{f_{Deinterlaced}(x, y, n - 1), f(x, y + 1, n), f(x, y - 1, n)\} \quad (2.16)$$

$f_{TR}(x, y)$ 遞迴輸出， $f_{Deinterlaced}(x, y, n - 1)$ 代表動作估計是使用前一為輸出的幀畫面做動作估計，完成後再使用中間值濾波來選擇中間值，其內容會包括前面一個畫面的補償結果再加入，加上使用現在畫面上下線的資料來做限制避免錯誤的累積，其輸出也使用了中間值濾波，整體來說，效能上也會被限制住。

2.7. 混合式解交錯演算法

近期研究解交錯的發展變成了混合式(Hybrid De-interlacing)的作法，以加入很多方法來使影像的品質提昇。例如混合了邊緣線平均內插 ELA，合併圖場的動作補償，適應性水平動作偵測補償，動作補償等。

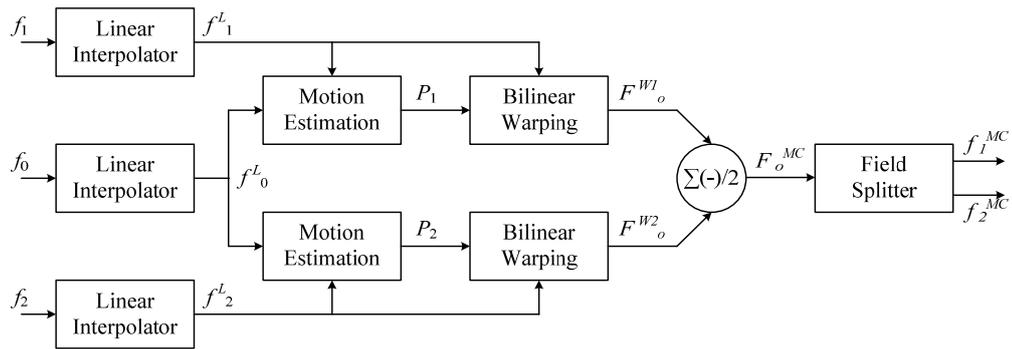


圖 14 三圖場的動作偵測解交錯架構

在A.J. Patti等人的研究中[18]使用三個圖場的動作估計解交錯，如圖 14所示。使用內插得的畫面做動作估計後，再平均動作補償的兩個畫面，再抽掉影像還原動作補償後的畫面，再混合其它的做法解交錯，也是屬於多圖場動作估計。

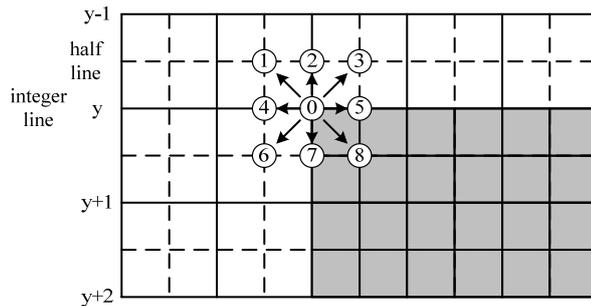


圖 15 使用 Quad-pixel 動作偵測圖

圖 15是O. Kwon等人的研究中[19]，除了邊緣內插外，與前後圖場做動作估計的運算，再使用其移動向量做補償，並加入子像素的動作補償，且混合很多方法。而在YY Jung等人在[20]中的研究混合了動作補償與基本的邊緣內插與平滑向量，但平滑向量可能會將正確的向量過濾掉的可能性，所以要考慮的平滑向量使用時機。

在YL Chang的研究中[21]除了使用多圖場外，更混合了動作補償與全域動作補償與邊緣內插，動作適應性等各式方法，其架構相當的複雜，其判斷的方向包含了多圖場的SAD比對等如式(2.17)~(2.19)。比對兩個搜尋到的多方向移動向量， k 為區域內前後比對的最小SAD總合。以SAD較小的為其移動向量，區域的動作補償再以這個移動向量補償。而全域移動(Globe Motion Estimation)來計算的部份，GME通常需要降低取樣，由計算梯度來判斷影像是否是全域的移動，如攝影機的平移、旋轉、變焦等，但全域的動作估計比較不適合有運動物體的移動的畫面，因為它會降低取樣，會使小物體的移動被忽略，效果可能會有限。

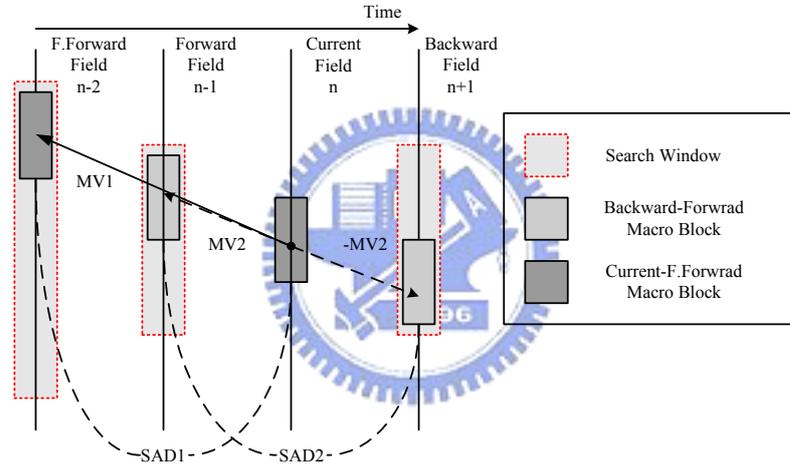


圖 16 四圖場動作估計的比對方法

使用的多圖場的差異值比對準則(2.17)-(2.19)。

$$\mathbf{SAD}_{1k} = \sum_{i,j \in MB} \left| f_{t-2} \left(x + v_{x1}, y + v_{y1} \right) - f_t(x, y) \right| \quad (2.17)$$

$$\mathbf{SAD}_{2k} = \sum_{i,j \in MB} \left| f_{t-1} \left(x - \frac{v_{x2}}{2}, y - \frac{v_{y2}}{2} \right) - f_{t+1} \left(x + \frac{v_{x2}}{2}, y + \frac{v_{y2}}{2} \right) \right| \quad (2.18)$$

$$K = \mathit{Arg} \min_{k \in SR} (\mathbf{SAD}_{1k} + \mathbf{SAD}_{2k}) \quad (2.19)$$

在D. Wang的研究[22]混合了MC與VT Filter，但VT的缺點是會殘存影像。HM Mohammadi等人的研究[23]更是使用到五個圖場的畫面差異偵測，再利用其資訊做垂直動作的補償，亦是多圖場的動作估計。S. Byun等人的研究[24]則是遞迴式的雙圖場動作比對補償，但它也是在輸出使用中間值選擇輸出。J. Kovacevie的研究[25]中則是以權重混合遞迴的方法來完解交錯，但其效果依舊會有遞迴缺點。

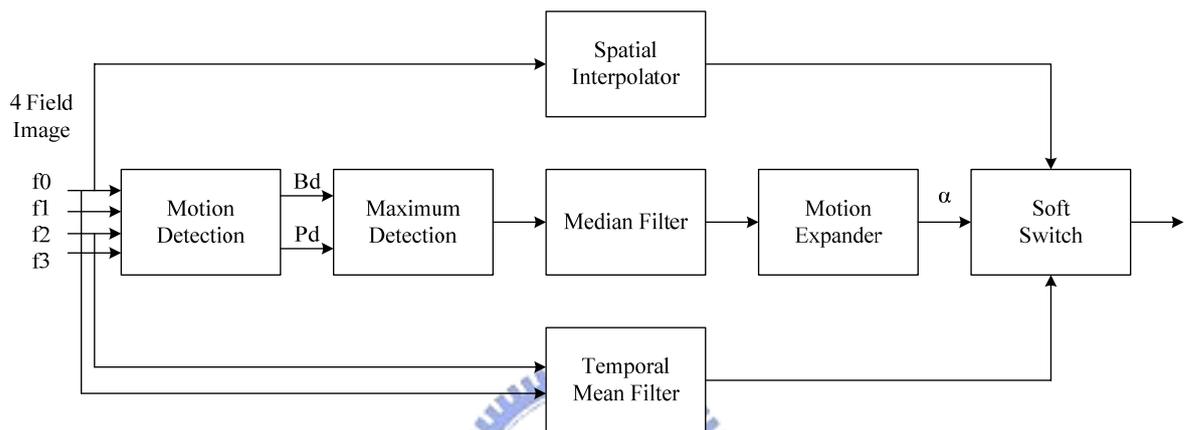


圖 17 不同的四圖場動作偵測的架構

圖 17為D. Han等人的研究[26]，探討當區域亮度改變時對應的動作偵測，但輸出亦有用臨界值，臨界值設定則會影響輸出效益。T.S. Chong 等人的研究[27]則是內容適應性的邊緣線解交錯，使用比對區域內的邊緣線走向的適應性補償方法。G. de Haan 的研究[28]則是使用了適應性遞迴(Adaptive Recursive)的解交錯系統。X.B. Gao 的研究[29]則是以不同搜尋法搭配移動物體的區分做補償。在M. Biswas的研究中[30]，有使用移動向量的關聯處理，但是以判斷出有效或是無效的移動向量，並降低內插誤差。Q. Huang 的研究[31]則是三圖場的動作估計，利用多圖場SAD來當判斷法則，以權重的方法重疊影像。越多的做法越需要準確的決策，來決定輸出的補償方式等適合的使用條件，或是混合輸出的結果再用中間值去選擇統計上的最佳狀況，或是依其特性加入權重比較來處理。但不論是何種方法，都必須是有效的決策出適合的影像，否則在影像上會有不連續的內插出現，所以以決策法針對不同狀況做考量，以便達到更好的品質。

2.8. 本章總結

回顧本章所探討的文獻研究，品質的好壞與所花費的代價會因不同需求有不同的取捨，與硬體的設定。但要降低硬體代價，又要維持相對的品質，其對應的演算法架構與決策機制就相當重要。所以由交錯信號的產生，解交錯還原，其方法有空間性內插、時間性內插與垂直時間濾波器等，可了解線性的方法容易實現但效果有限。進而延伸到非線性內插，有邊緣線平均法、動作偵測、動作估計與補償，或是利用時間遞迴或混合式的解交錯的尋找最佳的內插值。

但不論哪一種方法都有其優缺點與其適用的地方，但影像是變化的，如何有效的降低硬體代價，並考量計算上的負擔就與品質的提昇就非常重要。因為這些基本方法中，基本設定上就會有互相抵觸的地方，例如使用空間性內插，就必須接受垂直解析度無法提高，使用動作補償雖然可以提高垂直解析度，卻是要付出必要代價，所以不同的應用範圍的解交錯，需求也不同，所以動作估計是提高品質的方法，但必須要有相關搭配決策的法則。



第三章 雙圖場適應性混合解交錯演算法設計

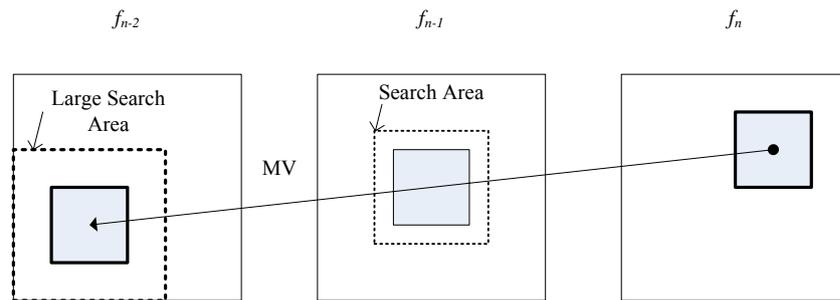
由先前的研究探討中，基本的空間與時間內插方法雖然簡單但效果有限。而動作偵測的方式雖然可以分辨出動態區域與靜態區域，但臨界值的設定動態與靜態是與內容有相關的，且因為只能得知動態區域與靜態區域的區分，無法得到有效的動作資料。而動作補償可以得到移動向量的資訊，但動作估計補償的移動向量準確度也相當重要，只有直接動作補償將會使影像出現不連續交錯的影像線。為解決動作補償的不連續的影像，必須是由動作估算所得到的資訊來決定動作補償的方法，並加入時間與空間的內插補償。另外為降低動作補償的所需要的代價，所以本研究將以降低硬體代價的動作補償解交錯系統為主體，並有效判斷影像動作再加以補償與內插。

3.1. 適應性混合解交錯架構改善

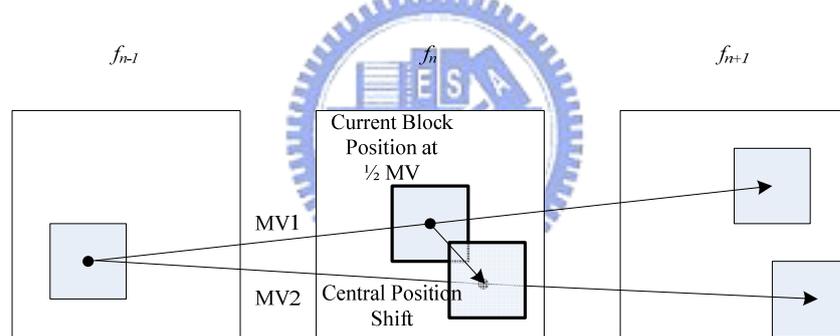
由解交錯研究中，有動作資訊的多圖場動作估計，在架構上將需的很多的記憶體，為保有動作估計的優點，架構上又不需要太多記憶體，因為多圖場記憶體的計算複雜，耗費較多記憶體，且影像內容變化大的時候，其移動向量不一定是線性的移動。所以降低所需要的記憶體，本文以雙圖場的動作估計的架構為基礎，再搭配改良的動作決策判斷，改善移動向量誤差的錯誤補償，增加補償的效能。

3.1.1. 減少動作估計的圖場

在[5] [20] [21][30][31]中所提出的方法都是用前後圖場以上的動作估計，來降低移動向量誤差的與消除圖場極性不同。但計算前後圖場的動作估計如果遇到影像變化比較大，或是計算到的移動向量差異很大，其動作估算的計算可靠度將會降低。而在硬體的代價上，也會比較高。



(a) 三圖場的動作估計搜尋範圍



(b) 前後圖場的移動向量中心點對映

圖 18 多圖場動作估計搜尋區域

如圖 18(a)所示，跨越多圖場的動作估計代表著必須要使用更多的計記憶體，且因為跨越圖場，物體的搜尋範圍也需要比較大，可能的移動向量誤差也會比較大，在這樣的狀況下不但硬體代價會比較高，處理上也比較複雜。而圖 18(b)所說明的是它使用前後圖場做動作估計，再從 $1/2 MV$ 做動作補償，其會有對映的區塊中心點不在現在區塊上的問題，這樣的比對將會更加困難，如果採用前後場都個別做動作估計，其計算量等於是兩倍。所以針對這部份的改善，本文以雙圖場的動作估計為基礎，可降低所需要搜尋範圍與計算量，亦可降低跨圖場的移動量太大的移動向量誤差，再判斷移動的可靠度。

3.1.2. 輸出判斷決策改善

在[20]的研究中輸出的決策方法採用了中間值濾波，表示如式(3.1)。因為中間值選擇只是做影像大小的排列，如果中間值濾波的影像差異很大，中間值選擇的輸出可能會無法控制輸出的品質，因為它是由影像鄰近像素去選擇的，如果相鄰影像變化很大，中間值選擇的機制將使影像品質低落。式(3.1)為中間值選擇決策，關係式如下。

$$F_{MS-MC}(x, y, n) = \begin{cases} f(x, y, n), (y \bmod 2 = n \bmod 2) \\ \text{median}(A, B, C, D, \frac{C+D}{2}), (\text{otherwise}) \end{cases} \quad (3.1)$$

$$A = f(x + v_x, y + v_y, n + 1)$$

其中 $B = f(x - v_x, y - v_y, n - 1)$

$$C = f(x, y - 1, n)$$

$$D = f(x, y + 1, n)$$

$F_{MS-MC}(x, y)$: 在(x,y)位置的中間值選擇輸出的動作補償。

$\text{median}(\cdot)$: 中間值選擇濾波處理。

v_x, v_y : 為該區塊的移動向量。



另外有相關的研究在決策是採用移動向量或是區塊比對差異值，其決策流程如圖 19。

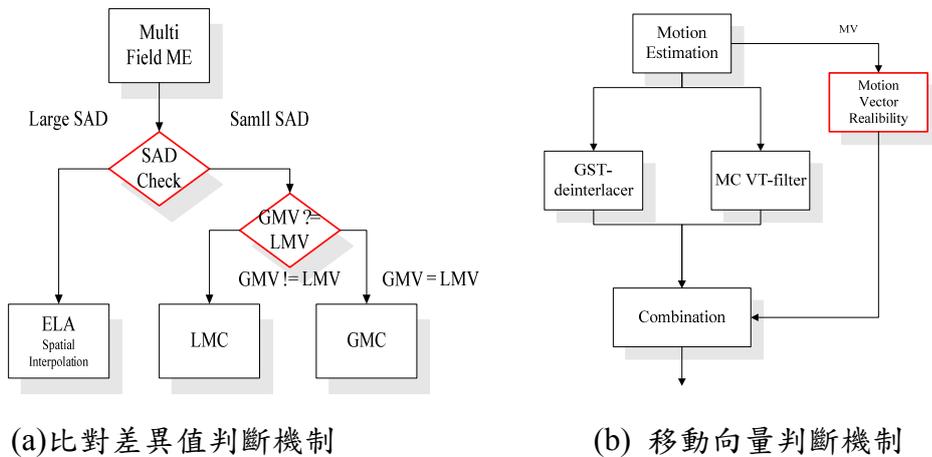


圖 19 不同的動作判斷機制

在圖 19(a)，輸出判斷機制中，在[21]的研究中以 SAD 的大小去確認輸出是使用空間性內插或是空間性內插，如果是使用空間性內插，就需再比對移動向量與全域搜尋移動向量(GMV)是否相同，如果不相同就只有區域的動作補償。但全域動作搜尋對畫面整體移動比較有效果，雖然它在判斷不是全域移動後，有對局部的移動做動作補償，但區域的移動沒有判斷移動向量可靠度再加以補償，可能會有局部的影像不連續現象。

而在[22]的研究中以判斷移動向量的差異大小來做輸出的決策，如圖 19(b)，只以臨近的區域內的移動向量如果小於 ± 2 ，就會認為該區域的移動向量是可靠的，進而使用動作補償或是空間性的內插補償，但這樣移動判斷的誤差容許值會比較大一點，輸出就會受到移動向量誤差影響，由相鄰移動向量判斷得到。

3.1.3. 整體架構

在架構上，多圖場會使得輸出會有時間延遲，也就是現在輸出影像會是上一張影像補償輸出的，會有時間差別。所以採用雙圖場進行動作估計，以前圖場與現在的圖場的動作估計，就不會有這樣的問題，亦可降低記憶體需求與降低傳輸頻寬的改善。

本文研究方向以動作估計為基本架構，可降低所需記憶體，並用前一張畫面與現在畫面的資料作估計來做判斷，研究可行的方法，判斷出影像適應的補償方法。動作估計補償是使用區塊比對方法，其動作補償效能的好壞與移動向量精確度有關。所以當移動向量有因內容變化大而有誤差，必須要有機制判斷影像是否為變化很大的區塊，所以決策上採用移動向量與比對差異值的相鄰關係，由鄰近的關係分析影像動作與移動向量補償的可靠度。整體來說，由雙圖場動作估算所得到的移動向量與區塊比對差異值，做為本文的適應性動作估計的決策因素，讓移動向量的關聯性作畫面動靜上的區分，再予以使用動作補償，並配合改良的空間性的內插，來達到更好的效果。

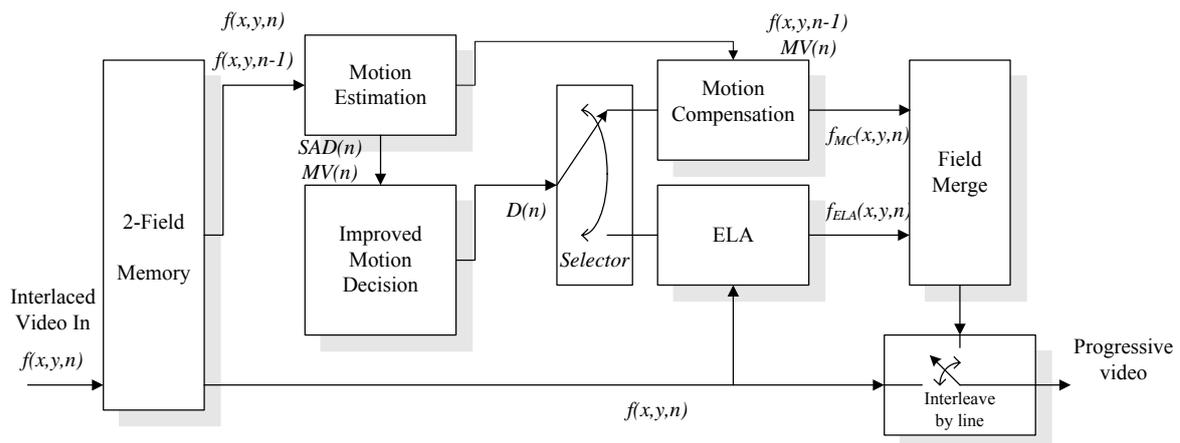


圖 20 適應性混合動作補償之解交錯主要架構圖

且為了有較好的補償效果，增加動作估計的準確度與動作補償的適應性判斷的正確性，與加強邊緣線內插，讓整體的效能在雙圖場的架構上仍有好的補償效果。藉由判斷移動的資訊來達到影像動作的判斷，提高動作補償的準確性，降低多圖場動作估計的記憶體需求，亦可降低整體的存取頻寬，降低計算複雜度。

就動作估計的部份，因為使用動作估計的資訊判斷。也因此需要比較準確的移動向量資訊做判斷，由於快速搜尋的準確度會比較差，所以使用完全搜尋(Full Search)，增加移動向量的比對準確性。定義圖 20 的動作補償輸出的移動向量為 $MV(n)$ ，比對的差異絕對值為 $SAD(n)$ 。決策判斷由整個畫面的 $MV(n)$ 與 $SAD(n)$ 做關聯性比較，因此在動作估計輸出所有的區塊移動資訊後，動作估計會輸出 $MV(n)$ 與 $SAD(n)$ 給動作決策處理，決策法則就以 $MV(n)$ 與 $SAD(n)$ 相鄰關係去判斷輸出該區域的移動資訊是否可靠，進而判斷補償種類，來決定動作補償是 MC 或 ELA 的部份。圖 20 所示中，決策輸出 $D(n)$ ， $D(n)$ 是一個決策參數的矩陣， $D(n)$ 會設定所有的區域要用何種方法補償，所以動作補償電路之後會依照 $D(n)$ ，來決定輸出的種類，最後再混合原始影像成為循序的影像輸出。

3.2. 適應性影像補償之輸出決策

為使影像的動作偵測具有適應性，使用影像的內容的移動的資訊來決定輸出的補償。亦可取代平滑向量的處理，以鄰近的移動向量與區塊差異值變化，決定影像補償的種類。利用移動資訊相鄰的差異做適應性的動作補償，可針對不同狀況的動作做最佳的補償，讓輸出有適應性的補償效果。

3.2.1. 最近相鄰差異的移動判斷決策改善

為改善移動向量誤差的補償，本文嘗試以最近相鄰差異(Nearest Neighbor Difference, NND)的比較，如圖 21，在相鄰近移動向量處理中，一般只有平滑向量(MV smooth)處理較多，但直接平滑向量可能會使輸出的移動補償變的更差。且相關研究較少對相鄰近移動向量(NNMV)與相鄰近比對差異值(NNSAD)一起做判斷，所以本文以建構適應性的補償判斷，利用它們相互關聯性判斷出適當的動作補償，不只是中間值平滑處理的輸出，而是以 NNMV 與 NNSAD 的關係做為輸出的依據，亦改善單方向動作估計的動作補償的誤差時的補償，增加輸出的品質。

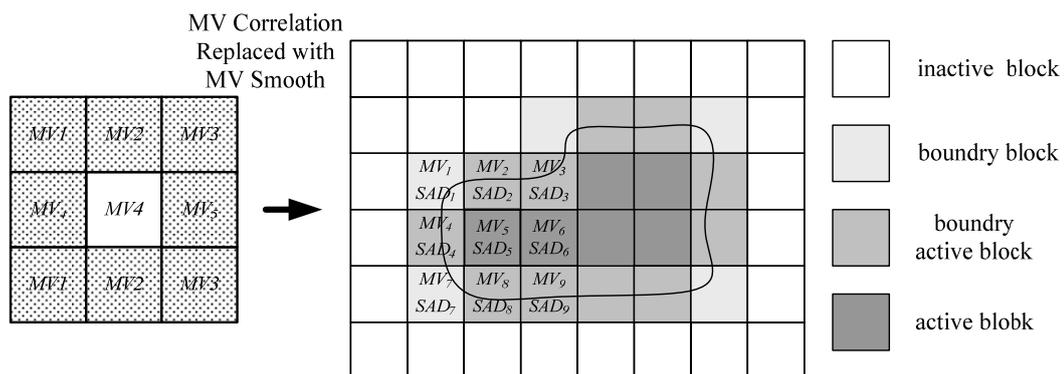


圖 21 臨近關聯性決策關係圖

為了使動作估計解交錯能有正確判斷動作補償的方法，越多的移動資訊越可以做有效的判斷，增加輸出的影像品質。在雙圖場的動作估計中，移動向量資訊的相鄰關係就變的很重要了。當 MV 可靠度不高時，還有 SAD 可判斷差異來決定輸出的品質。

說明圖 21 使用改良式物體的移動關聯性判斷。以定義動作估計的鄰近關聯性資訊，判斷其區塊種類，可使區塊分類的補償更好。定義移動區塊(Active Block, AB)、移動邊緣區塊(Boundary Active Block, BAB)、邊緣區塊(Boundary Block, BB)、非移動區塊(Inactive Block, IAB)。

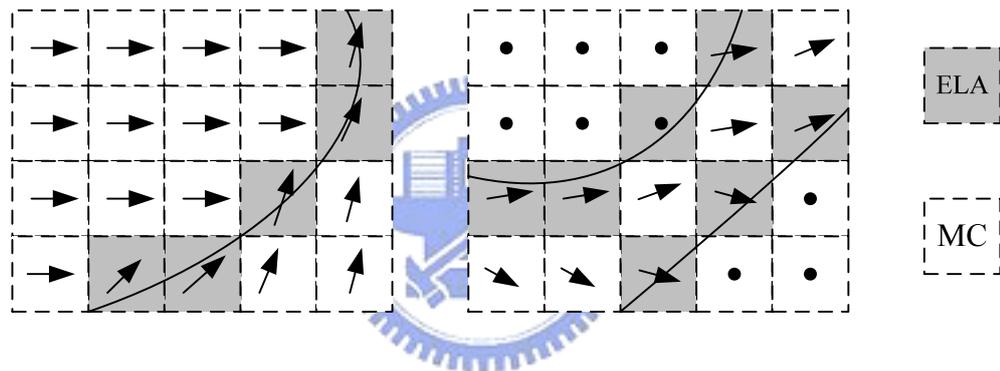
當圖 21 中間的 AB 是區域性的物體移動，就像是鏡頭的平移一樣或是物體平移，移動區域會包含了數個 AB，這代表中間區域的 AB 與前一個畫面的搜尋範圍內有相似度很高的區塊，就可使用 MC 的方式補償。而圖 21 中的 AB 區域旁的物體邊緣，是包含了影像背景的 BAB，BAB 是相鄰 AB 的區塊，如圖 21 所示在 BAB 形體分布，移動向量不一定可以正確找到，加上 SAD 大小差異很大，就需判斷移動區域影像邊緣是否還適合使用 MC 的方式補償。如果 SAD 分布於鄰近區域中，不是所有 SAD 中最大的，移動向量與相鄰的方向一致，這就可代表該區塊是移動物體的邊緣區塊，代表動作補償是會比較好，使用 MC 的補償。反之 SAD 是該區域最大的，過大的 SAD 代表 MC 補償的效果會不好，所以移動的邊緣會以 ELA 內插，以達適應性的補償。

而 BB 為變化物體的邊緣，IAB 即為靜態背景，如果背景是靜態的，又因移動物體的影像或許才佔影像的 1/4，或更小，為避免小物體的移動影像不連續的區域產生，若移動的物體小於區塊內，移動向量會不精確，其 SAD 值也會大於相鄰區塊，亦可判斷出移動物體是局部物體的移動，則由 ELA 的方式補償。如果 MV 與 SAD 變化不大，才會是以 MC 的方式補償。

這部份的動作估計主要以 MV 與 SAD 其相鄰的關係，由該區塊與它的鄰近區塊的移動關係的分布，判斷圖 21 所的 $(MV, SAD)_5$ 與相鄰的 $(MV, SAD)_{2,4,6,8}$ 與 $(MV, SAD)_{1,3,5,7}$ 的移動關聯性。很多狀況下 MC 會因為移動向量不準確或有誤差，而出現區塊狀的線條，這種條件下用 SAD 關聯性判斷就很重要，由 SAD 的大小與相鄰的區塊比較是否差異很多，以改善交錯式影像的動作補償方式。其中不論是 MV 或是 SAD 都以相鄰的關係比較，確保區域性動作的動作補償是可靠的，增加適應性補償效能。

3.2.2. 移動區域邊緣向量誤差補償

由於物體的移動變化是無法預知是大或是小，所以才需對移動物邊緣的移動向量加以判斷，確認其準確性。以圖 22(a)的物體邊緣移動向量示意圖來說，當移動物體大於宏塊，其內部的移動向量會比較一致，這可由移動向量鄰近的分布，看出移動的關連性，這時使用 MC 補償會比較好。但圖 22(b)雖然也是移動，但移動物體接近於比對宏塊，這時比對鄰近的移動向量差將會是有差異的，差異太大又會讓 MC 可能會補償不好，且邊緣的移動會是移動的前景加靜態的背景，這種狀況下使用 ELA 的效果會比較好，可減少不連續影像或是鋸齒邊緣。



(a) 移動區域大於 MB

(b) 移動區域小於 MB

圖 22 物體邊緣的移動向量種類

先前的研究會使用移動向量平滑處理，處理方式亦中間值或平均值的方式，但缺點有動態的移動向量會被鄰近的靜態向量給改變，且鄰近的移動向量分布不集中，平滑移動向量處理誤差就更嚴重了，所以就以 SAD 分布的判斷來說，如果移動物體是不一致的，並且造成很大的 SAD 值時，則代表改以空間性的補償為佳。以圖 21 的 MV_5 會與鄰近的 MV_{1-9} 移動向量做平滑向量濾波器說明，因平滑移動向量處理後，再將其補償回去，是改變其對應的區塊不一定很合適，所以在移動向量上才必須判斷正確性與可靠度。

在 YY Jung 的研究中[20]也有平滑向量處理，但如果沒有加以判斷而直接濾波其動作估計效能將會被降低。因為使用平滑向量的缺點是 SAD 最佳的移動向量有可能會被

改變，所以要有處理判斷移動向量的可靠性。在比較小的物體移動，如果移動向量還有參考價值，採用動作補償會比較好，反之如果移動向量差異很大，就需要空間性的內插。

以圖 23(a)為例，因為該區塊的移動跟內容變化太大，無法完全的找到很合適的區塊，即使採用了平滑向量也找到正確的移動向量，所以才必須以鄰近的移動資訊做判斷，當圖 23(b)中移動向量 MV_a 的移動向量跟相鄰靜態的 MV_s 等差異很大時，代表物體移動的形態已經改變很多，或是移動很快。無法整個區塊補償過來，為避免發生不連續的狀況出現， MV_a 的區域就必須是以ELA補償為主。

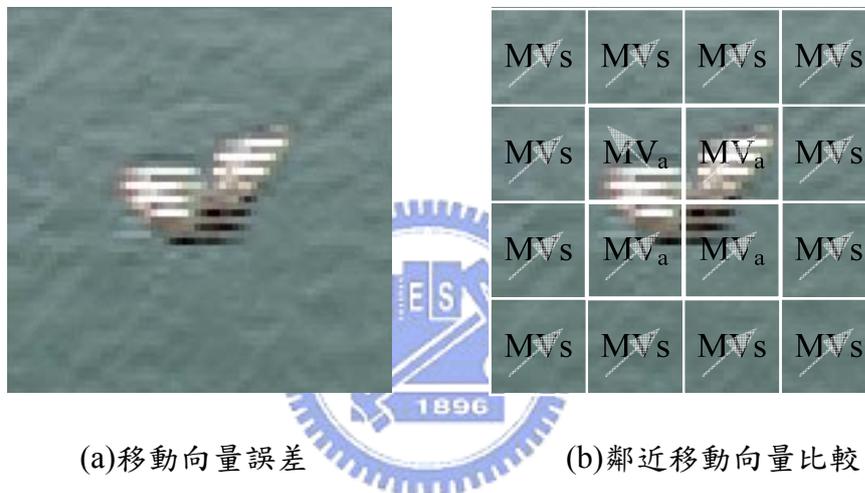


圖 23 影像序”Container”移動向量處理誤差的補償

其主要的判斷關係式表示如式(3.2)，MVD 為移動向量的相鄰差異比較，SADR 為相鄰 SAD 的差異比較， U 為的主要的適應性補償的參數表示。

$$U = \{MVD(v_x, v_y)_{neighborhood}\} \cup \{SADR(v_x, v_y)_{neighborhood}\} \quad (3.2)$$

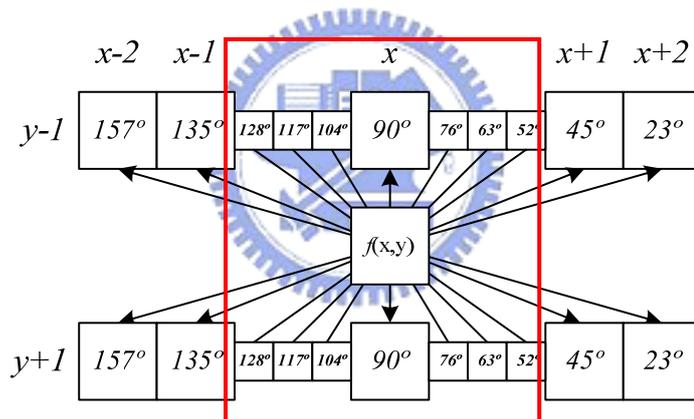
從以上改善比較中可確認，相鄰的MV與SAD關係是非常的重要，它可提供移動關聯性並加以判斷誤差或是了解其關聯性的適應性補償，當移動向量差異很大時，還有SAD輔助，加以正確判斷，使輸出補償可以適應性的補償，使得輸出有較好的補償品質。

3.3. 子像素邊緣線內插

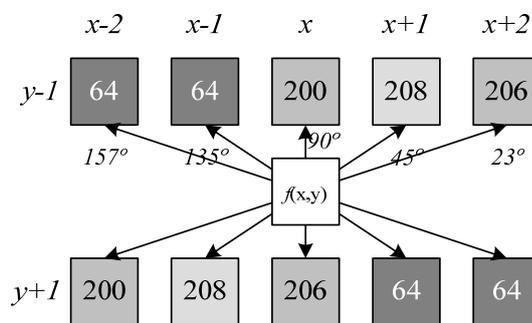
由於大部份解交錯的研究中，所採用的邊緣線內插，其角度通常都小於 45 度，本文的邊緣線處理則增加了 45 度到 90 度的子像素內插，來增加比對邊緣線角度的方法，改變原有的中間值選擇器，使內插輸出更細微。

3.3.1. 改良式邊緣線子像素內插

使用改良差異比較的子像素邊緣線內插(Enhanced Sub-pixel ELA, ES-ELA) ，如圖 24(a)，來取代邊緣線輸出加入的中間值濾波(Median ELA, M-ELA) ，改善邊緣線內插的輸出效果，避免邊緣線內插的中間值濾波可能誤判的狀況。



(a) 邊緣線的子像素計算



(b) 邊緣線的差異比較

圖 24 影像邊緣內插的改良

邊緣線子像素內插判斷中，為了使邊緣的像素都是內插出來的，所以不使用中間值濾波，是因為它不是真的內插的像素，所以才改用比較中心方向內插差異值比較。如果邊緣影像差異的計算不是中間方向，則計算其他方向角度之差異時，會加入 90 度角方向內插值的差異作總和。即使線條在很細的垂直線下，也能夠藉由加入中間值的內插差異比較來增進與中間方向的差異，得到正確的邊緣方向，使邊緣內插效果更好。

而子像素內插的相關研究中，在 K. Jensen 相關研究[33]中也有提到。但由於解交錯所應用的邊緣內插多以 45 度角的影像點開始計算差異值，如圖 24(a)，其垂直方向的角度較少，為求更微小的部份，本文以增加垂直方向的子像素邊緣線內插，增加可以判斷的角度。這樣的好處是當物體的邊緣是大於 45 度角時，補償的細節可以經由方向內插像素的細節更好。所以 ES-ELA 內插像素的細節設定，會增加內插位置在 1/4、2/4、3/4 的點，可增加角度的選擇，平衡兩邊的角度，在 45 度角與 90 度角的像素間，插入 52 度(1/4 位置)、63 度(2/4 位置)、76 度(3/4 位置)的邊緣子像素的計算，以增加垂直方向邊緣角度的偵測比對，會比沒有子像素的 M-ELA 內插到更好的值。

在 G. de Haan 在[34]中的邊緣線有利用到中間值濾波來過濾錯誤的邊緣的內插，但它會使得輸出有可能直接是選到上下點的值，致使細節品質下降。為取代中間值濾波，找到最匹配的邊緣走向，必須不使用中間值濾波器，而以中心差異值計算在取代中間值濾波器過濾，以得到較好的邊緣內插細節。以圖 24(b)來說明正確的內插點是上下的點 x 方向或是 $x+1$ 的方向，但就以差異值來說，卻是線條旁邊的黑色區域的差異值較小，這樣的結果會使得內插出來的像素變成是黑的，會使得線條沒有辦法連續，所以本文以這樣的問題下提出了，中心差異值比對法的邊緣線內插，多比對了方向內插所需計算的個別角度與中間內插的差值，以確保不會內插錯誤影像。

3.3.2. 中心差異值比對法的邊緣線內插

邊緣線內插的演算法設計了 9 個整數點位置邊緣走向內插，外加 6 點子像素的內插，主要是要對垂直線角度的內插的補足，加強垂直線方向的邊緣內插，而使邊緣線的

補償細緻。而水平方向的整數像素角度分別為 45 度、26 度、18 度、14 度，其再增加點數的角度將為 11 度，變化幅度已經不大了，所以取左右四點對區域性水平的邊緣內插已經有很好的效果，加上子像素內插也已經有了 52 度、63 度、76 度。這樣的像素每一個點都是內插出來的，在一些較細微的邊緣部份也會有不錯的品質。對一般的應用，其整數點有到 9 抽頭的邊緣線偵測就已足夠，太多抽頭的邊緣線內插在判斷差異大小也會增加許多代價。

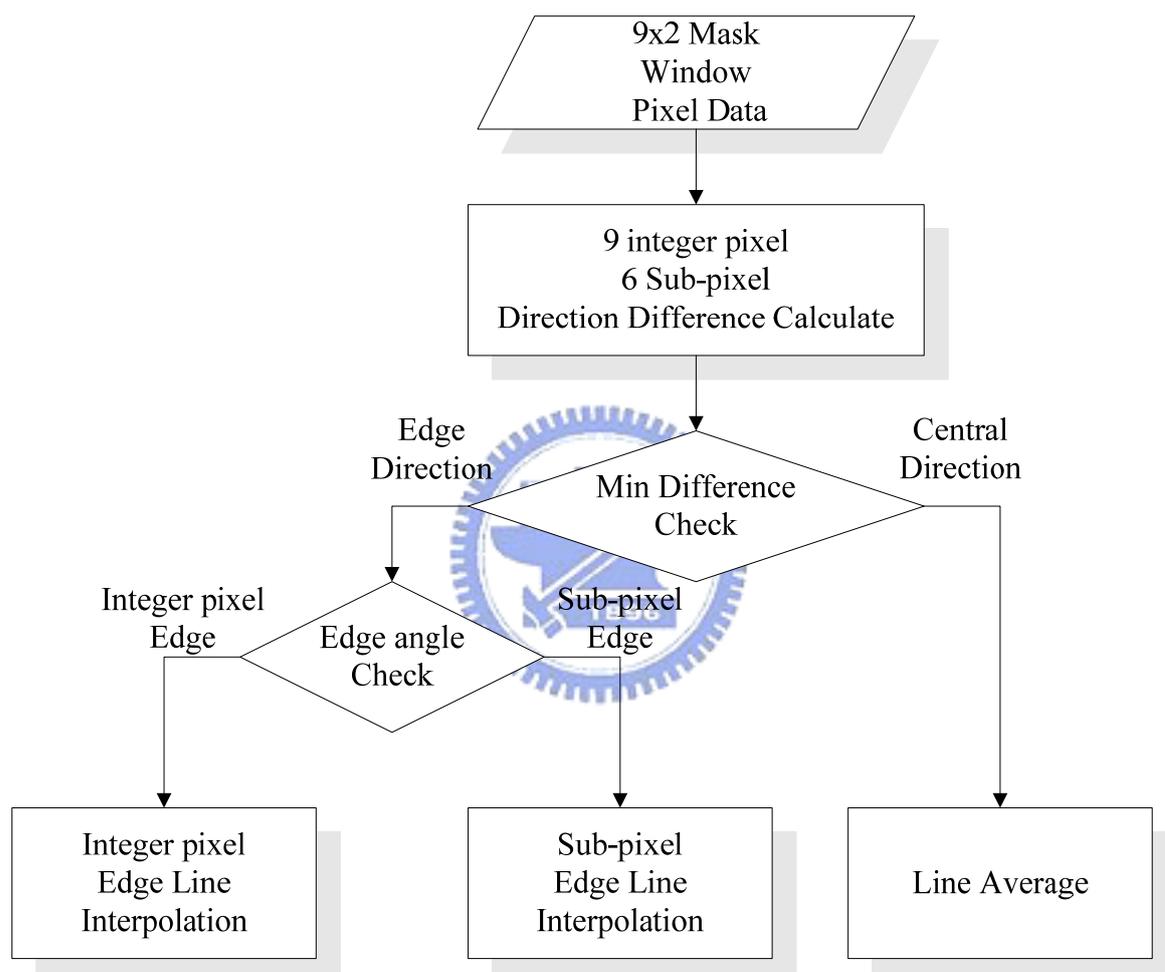


圖 25 子像素邊緣線內插流程圖

圖 25 為邊緣線的處理流程，包含了 9 抽頭邊緣方向性平均內插，6 點子像素內插，以中間差值來計算其邊緣，可使使影像更加細膩，演算法流程如圖。為了不讓邊緣線的內插因為中間值濾波的關係而選到上下點的影像，取而代之的是中間值內插的差異比較，所以得到的結果就一定室內插完成的值，而不是上下點選擇出來的值。

另外因為交錯式影像的關係，上下點資料位置已經是差一條線了，如果還用到跨線在 $y \pm 3$ 線上的資料時，其關聯性會比較低，這也是本文也以僅用上下兩條線的資料做參考，進行空間性內插，得到對應在交錯線的位置上的 52 度、63 度、76 度。其 $y \pm 3$ 平均值的如果是低頻變化，那麼上下兩條線內插就可以代表了，如果是高頻區域， $y \pm 3$ 邊緣很有可能內插到錯誤的機會很高，所以可利用率較低，不列入考量中。

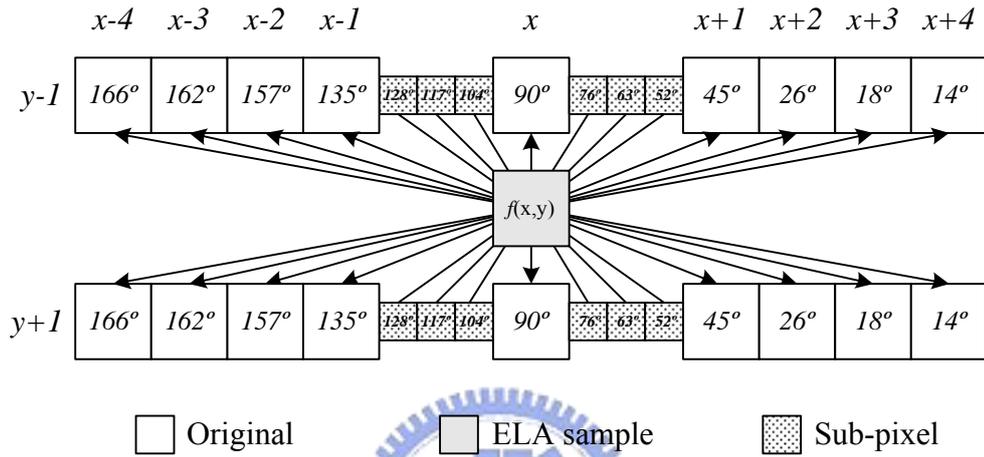


圖 26 邊緣線方向性 9+6 抽頭內插

圖 26 為本文所改良的邊緣線的處理技巧，並說明其代表符號與計算的步驟，首先邊緣線內插的計算會先計算每一個方向的差異值，分為整數像素與子像素的計算。

Step 1

先計算一般像素邊緣，計算各角度的差異 $diff_{direction}(i)$ ，其關係式表示為(3.3)。如果差異最小的不是中間方向，則代表會是該點是邊緣線。在計算完 $diff_{direction}(i)$ 還要計算各個與中間方向像素內插的比對差異值，計算 $diff_{direction}(i)$ 與中心點內插差異後，其目的就是要去取代中間值濾波器，將它的關連性帶入邊緣方向的計算中。其關係式如(3.4)。

$$diff_{direction}(i) = abs(f(x+i, y+1) - f(x-i, y-1)) \quad \forall i = \pm 4, \pm 3, \pm 2, \pm 1, 0 \quad (3.3)$$

$$diff_{center}(i) = abs((f(x, y+1) + f(x, y-1))/2 - diff_{direction}(i)) \quad \forall i = \pm 4, \pm 3, \pm 2, \pm 1 \quad (3.4)$$

Step 2

再計算子像素邊緣的差異值 $diff_{direction_sub}(i)$ ，(3.5)(3.6)所示為 $x=0$ 的位置加上 $x=1$ 或 $x=-1$ 的位置，再取內插的像素值，依照位置取內插的子像素後再比對邊緣線的差異值得 $diff_{direction_sub}(\frac{i}{4})$ 。關係式如式(3.7)。子像素邊緣線差異 $diff_{center_sub}(i)$ 的計算也加入與中間的平均值的差。關係式為(3.8)

$$f_{sub}(x + \frac{i}{4}, y) = f(x-1, y) \times \frac{i}{4} + f(x, y) \times \frac{i+4}{4} \quad \forall i = -3, -2, -1 \quad (3.5)$$

$$f_{sub}(x + \frac{i}{4}, y) = f(x, y) \times \frac{4-i}{4} + f(x+1, y) \times \frac{i}{4} \quad \forall i = 1, 2, 3 \quad (3.6)$$

$$diff_{direction_sub}(\frac{i}{4}) = abs(f_{sub}(x + \frac{i}{4}, y+1) - f_{sub}(x - \frac{i}{4}, y-1)) \quad (3.7)$$

$$diff_{center_sub}(\frac{i}{4}) = abs((f(x, y+1) + f(x, y-1))/2 - diff_{direction_sub}(\frac{i}{4})) \quad (3.8)$$

Step 3

將整數像素的差異 $diff_{direction}(i)$ 與中間平均差 $diff_{center}(i)$ 相加，當成為該方向的差異參考值。(3.9)。同樣的我們將子像素 $diff_{direction_sub}(i)$ 的差異與中間平均差 $diff_{center_sub}(i)$ 相加，當成為該方向的差異參考值(3.10)。

$$diff_{total_pixel}(i) = diff_{direction}(i) + diff_{center}(i) \quad \forall i = \pm 4, \pm 3, \pm 2, \pm 1 \quad (3.9)$$

$$diff_{total_sub}(\frac{i}{4}) = diff_{direction_sub}(\frac{i}{4}) + diff_{center_sub}(\frac{i}{4}) \quad \forall i = \pm 3, \pm 2, \pm 1 \quad (3.10)$$

$$f_{CDS-ELA}(x, y) = Arg \min \left\{ diff_{total_pixel}(i), diff_{total_sub}(\frac{i}{4}) \right\} \quad (3.11)$$

於是我們便得到了個別的差異值，再由式(3.11)將差異值分別的找出最小的值，即為內插的方向，但如果要內插的角度值在 $x=1$ 或 $x=-1$ 位置上則比需加入子像素的判斷。輸出再依其所得到的最小差異，對物體的邊緣方向進行內插。

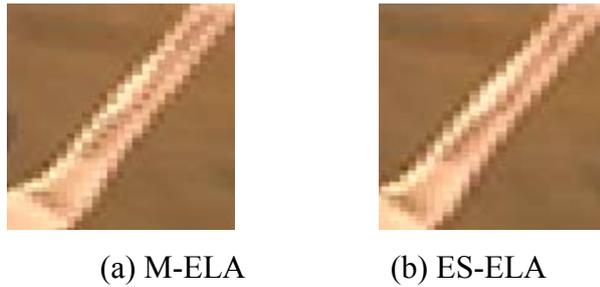


圖 27 中間值內插與改良邊子像素內插的差異比對圖

圖 27 顯示使用子像素中值差異(ES-ELA)與中值濾波(M-ELA)的結果，可看出在鄰近的幾個 Pixel 的變化中，圖 27 (b)本文所提的 ES-ELA，子像素中值差異確實可以將邊緣線的內插正確的判斷，不會有圖 27 (a)的因為中間選值的關係而造成細節上的內插錯誤，致使影像不細緻。

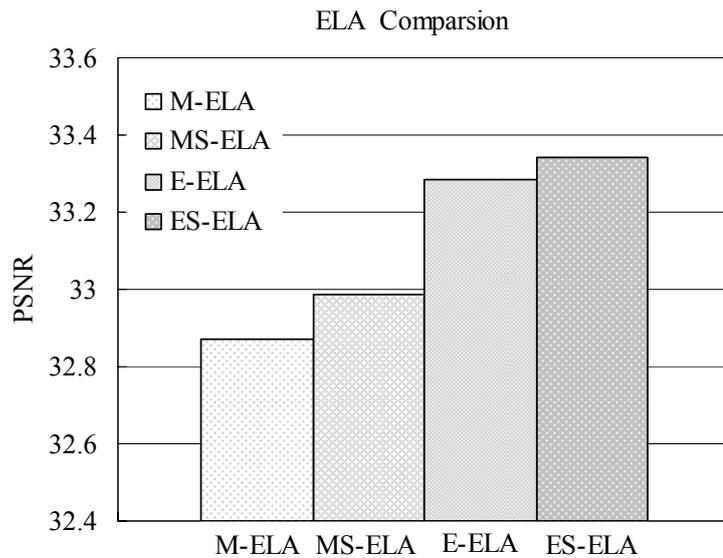


圖 28 不同的邊緣線內插的效能比對圖

而圖 28 再比對 ES-ELA 與 M-ELA 不同方法的比較數據。以 Foreman 為比對平台，它有包含一些邊緣線的背景與人物的邊緣，其相同條件下前 50 個畫面的輸出。圖包含一般的 ES-ELA 與 M-ELA，子像素的中值濾波(MS-ELA)，中值差異比對(E-ELA)，由常用的邊緣線內插做效能比較，其結果顯示，子像素中值差異可有 0.5db 的改善變化，顯示上也有細微的改善。

3.4. 動作補償解交錯之設定

本節說明動作計估計解交錯的相關參數設定，包含了重疊區塊動作估計比對，搜尋範圍與宏塊設定，與使用圖場內插的動作估計等，都是可提昇移動向量的準確性。

(1) 重疊區塊動作估計

為使移動向量更精確一致，許多動作估計都會加入簡單且有效的重疊區塊動作估計比對(Overlap Block Motion Estimation, OBME)，它可以有效的比對到較均勻的移動向量，提高比對的準確度。因為動作估計在比對的時候，如果影像的移動是一致的，會有利動作的判斷。但比對的方式不是很準確，導致移動向量會出現的誤差，就要額外的處理並判斷影像的移動，例如平滑移動向量處理，或平均移動向量等。除可避免不必要的問題產生，使用 OBME 還可增加移動向量的準確性與一致性。



(2) 搜尋範圍與宏塊設定

為了有比較好的影像品質，在多數的研究中，解交錯的動作估計，通常優先考慮使用完全搜尋，由於快速搜尋法比對的移動向量不是很準確或是很一致的話，將會使得動作估計失去其價值，也會將其比對的結果不準確而影響後續的判斷。為求一致的比對結果，本文採取完全搜尋來比對所要的移動向量。

比對準則以差異絕對值總和(SAD)來計算差異的總量，來得到最小差異的方塊。SAD 關係式表示(2.12)。在影像的比對法則中 SAD 與 MAD 的計算，其計算複雜度比較簡單，但如果是 MSE(Mean Square Error)的話，其計算的複雜度會因為 MSE 需要計算到差異平方而增加許多。由於 SAD 只需對差異的絕對值做加總的運算，運算量是比對法中最少的，也有其準確性，所以直接採用 SAD 的比對的方法。

另外由圖 29 的比較得知，16x16、16x8、8x8 以下的宏塊(Macro Block, MB)其 MSE 會比較平均，所以本文使用 16x8 在宏塊在動作比對中。因為宏塊的設定除了得依據影像尺寸來決定外，搜尋範圍的大小與準確度亦有很密切的關係。動作補償中，需要準確的移動向量做補償，因此宏塊大小也會影響到效能表現。小宏塊會不容易找到正確的移動向量，但比對的宏塊要是太大，其計算會比較複雜，太大的宏塊也會使邊緣的影像變的不連續，所以決定比對宏塊要適合比對的影像，其補償的畫質才會比較好。

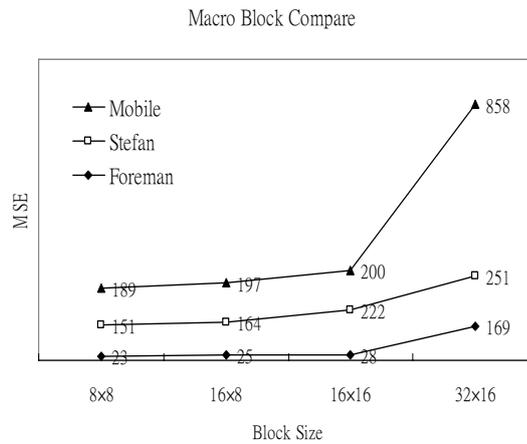


圖 29 宏塊大小之效能比較

搜尋範圍(Search Range, SR)也會與宏塊大小與影像內容有關，所以為了搭配宏塊大小來搜尋移動向量，本文設定搜尋範圍為 $[-16,+15]$ ，因為 ± 16 的搜尋範圍已經是超出宏塊的水平大小，通常影像如果快速移動速度超出一個宏塊，代表影像內容也會是移動快速的，可能會是較模糊的影像，而超出一個宏塊的移動向量，其影像內容變化也很大，直接補償到現在的畫面效果也不一定很好。以 CIF 來說水平大小為 352 個像素，宏塊的水平方向大小有 16 點，是整個 CIF 影像的 $1/22$ ，如果加上移動搜尋區域來說，通常超過的這範圍的影像代表移動很快，移動快的物體通常也會是較模糊的影像。所以本文設定搜尋範圍為 $[-16,+15]$ 的大小在 CIF 的影像上。所以目前的 CIF 的影像，所以先以較小的區塊比對進行研究。

(3)圖場差異的動作估計改進

為解決而奇偶圖場差異，在 K. Sugiyama 與 H. Nakamura [5]的研究中，就有使用了半像素(Sub-Pixel)做動作估計。A.J. Patti 等人的研究中[18]也是相同。另外在 O. Kwon[19]的研究中亦有使用 1/4 像素內插(Quad-Pixel))的畫面做動作估計的比較，使用了子像素內插的動作估計取代圖場的差異，即現在圖場的原始影像，比對前一個圖場線性內插的影像，如果比對的移動向量是零，就可以把前一個場的資料直接補償到現在的圖場內，如圖 30 之動作補償。這是因為子像素內插得到的像素，其位置對應上的就沒有奇偶圖場的差異，又可得到準確的移動向量，不會因為圖場不同而有會有比對的移動向量誤差。可解決圖場不同的問題。所以本文所提的雙圖場影像下，動作估計比對預內插的像素來比對，可解決極性不同的問題，亦可以得到較準確的移動向量。

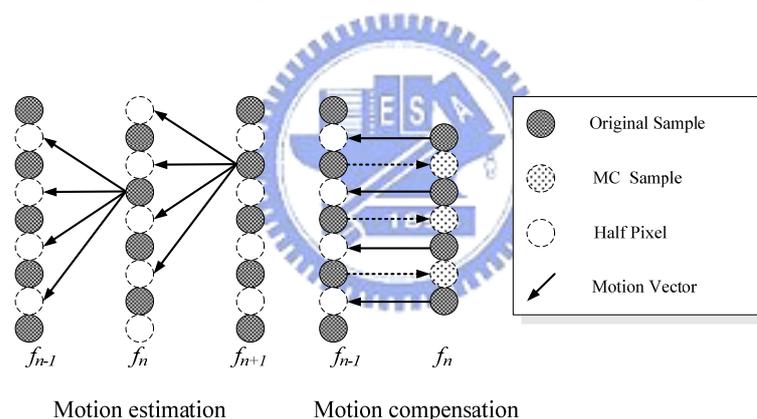


圖 30 子像素圖場差比對法

(4)圖場轉換

由於單方向的動作估計在圖場轉換(Sense Change)上，動作估計無法對圖場轉換的移動向量偵測其變化度是如何，基於這個原因，本文在圖場轉換的方法上，其因為圖場內容變化很大，其對應的移動向量大小與移動方向會很亂，跟相鄰的移動向量做比對，決策部份會使因移動向量不一致以及變化很大，判斷輸出會是空間性的內插，不會是時間性的內插，減少在鏡頭切換下能有錯誤的補償。

3.5. 雙圖場適應性混合解交錯演算法

本文利用相鄰動作 MV 與 SAD 動作資訊的差異做分類處理(Motion classification process)決定動作補償方法，決定動作補償的規則。搭配改良的邊緣線內插補償機制，其判斷的動作分類處理如圖 31。以雙圖場影像動作估計的資訊之建構的適應性混合式解交錯之技術，主體以動作估算補償的解交錯演算法為基礎。再以動作資訊做適應性決策補償，接著說明使用動作資訊補償的決策演算法。

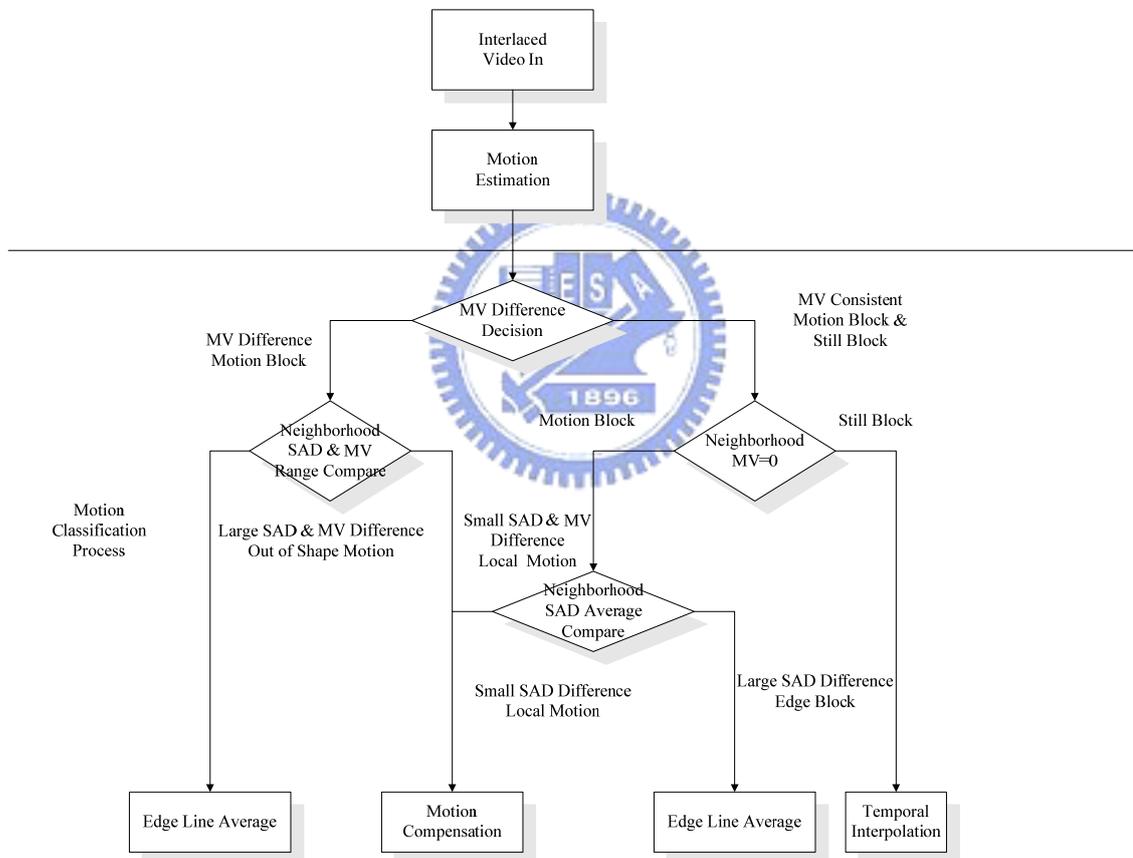


圖 31 臨近關聯性判斷流程關係圖

首先對動作分類處理，對 MV 的移動方向做判斷，再以不同的 MV 移動條件下判斷 SAD 的分布，藉此適應性的判斷出移動向量的可靠度，與其合適的補償方法。

Step 1. 確認區域移動或靜態的影像

影像移動補償的第一步是要確認影像移動是否是全域性的或部份區域的移動。比較區域移動的條件是否成立，這代表中心的 MV_5 與 $MV_{2,4,6,8}$ 的移動四方向都一致，為整體或區域影像的移動。整體的移動如果都一致，在這種條件下，因為周圍的移動向量都相同，代表該區採用 MC 的可靠度很高，但還需要 SAD 值雙重確認，以避免突然出現的物體所造成之 SAD 過大的錯誤。定義平移動作下在 (x, y) 對應的移動向量 (v_x, v_y) 的相鄰位置的決策矩陣 $P_{(x,y)}(n)$ 。

底下定義平移動作下的移動向量式。式(3.12)到(3.17)定義決策動作平移關係式。

$$D_{up}(v_x, v_y) = \{MV_5(v_x, v_y) - MV_2(v_x, v_y)\} \quad (3.12)$$

$$D_{left}(v_x, v_y) = \{MV_5(v_x, v_y) - MV_4(v_x, v_y)\} \quad (3.13)$$

$$D_{right}(v_x, v_y) = \{MV_5(v_x, v_y) - MV_6(v_x, v_y)\} \quad (3.14)$$

$$D_{down}(v_x, v_y) = \{MV_5(v_x, v_y) - MV_8(v_x, v_y)\} \quad (3.15)$$

$$D(v_x, v_y)_{neighborhood} = D_{up}(v_x, v_y) \wedge D_{down}(v_x, v_y) \wedge D_{left}(v_x, v_y) \wedge D_{right}(v_x, v_y) \quad (3.16)$$

$$P_{(x,y)}(n) = \begin{cases} 1 & , D(v_x, v_y)_{neighborhood} \leq 1 \\ 0 & , D(v_x, v_y)_{neighborhood} \geq 2 \end{cases} \quad (3.17)$$

在平移的狀況下， $P(n)=True$ ，區塊比對的動作補償是相當重要的，因為只要移動向量的差異比較值 $D(v_x, v_y)_{neighborhood}$ 小於 1，代表相鄰的移動是同方向的，從前一個畫面補償原始的影像的機會比較大，不會因為移動比較快而使用線性內插導致模糊，所以在交錯影像動作估計上，平移的畫面是可以由區塊比對法則得到好的補償。這也是 MC 補償的優點。而鏡頭的上下移動(Tilt)偵測也是相同，只要鄰近移動向量相同即是同方向的移動，只要是一致移動方向，不論區域性或整體性，動作補償都可有最佳的效能表現。

底下說明在這步驟上在影片所得到結果。以鄰近移動向量的輸出關係可從 Stefan 這個影像序是平移的影像，有 MC 補償的部份，其效果會好較好。

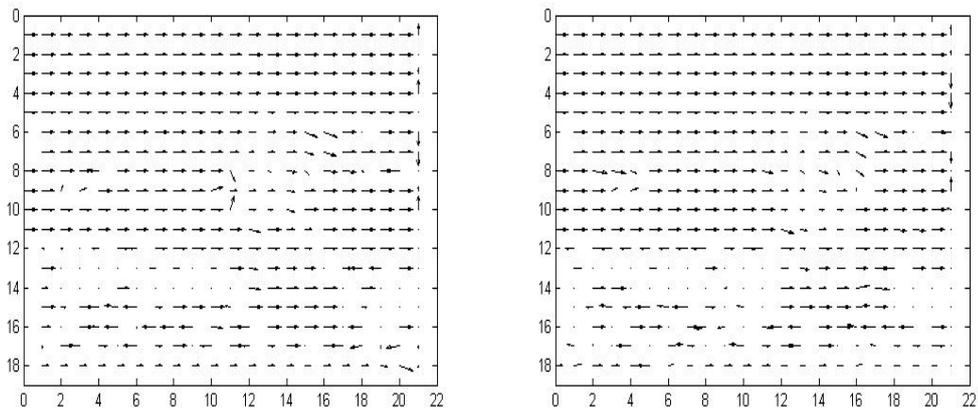


(a) Stefan 第三張

(b) Stefan 第四張

(c) Stefan 第五張

圖 32 連續影像平移示意圖



(a) 移動向量第四張

(b) 移動向量第五張

圖 33 區塊比對的前後影像移動向量圖

圖 32 是連續移動影像鏡頭的水平移動偵測，在畫面平移的狀況下，一般的水平移動偵測如果超出範圍，是無法有效的補償回去。所以 ME/MC 在影像平移或上下移動的補償是可以有最佳的效果，因為平移畫面代表移動方向一致，輸出可從前一個畫面補償原始的影像回去。所以水平移動的動作偵測，在動作估計後可以從整體的移動向量分析看出，當影像是平移時，移動向量也是整體的移動，所以從圖 33 前後張移動向量可看出，其背景是一直移動的，這樣在交錯的影像動作估計上，平移的畫面是可以由區塊比對得到最合適的補償方法。

再觀察移動向量的分布，在 Stefan 第五張的影像，可發現計算前一個圖場的移動向量中，如果是平移鏡頭的話，圖 34 中的移動向量的分布會很集中，所以要偵測影像的平移，可以由移動向量與鄰近的移動向量，比對它的差異來決定影像平移的部份，使用動作補償是否合適。所以當動作移動方向是一致的，其動作補償將是比較好的。

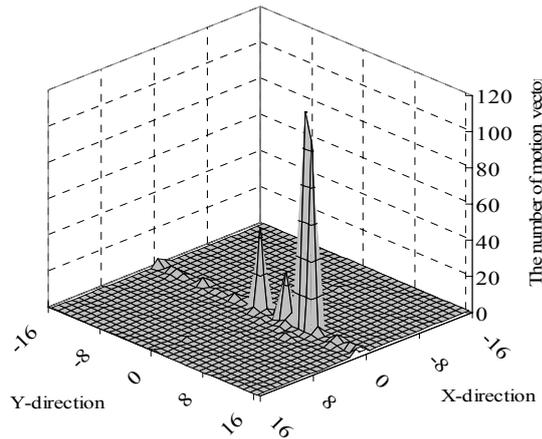


圖 34 水平移動下的”Stefan”移動向量統計圖



Step 2. 確認完全靜態區域

第二步是要找出完全靜態的區域。由判斷鄰近所有的移動向量是否為零，如果完全為零代表是靜態的影像。比較 MV_5 與 $MV_{1,3,5,7}$ 及 $MV_{2,4,6,8}$ 的八方向移動向量是否也是零，確保完全靜態時是使用圖場間內插。因為 MV_5 與 $MV_{1,2,3,4,6,7,8}$ 的移動向量為零的狀況下，其代表為比較大面積的靜態，使用圖場間內插的效果會比較好且可以增加垂直方向的影像解析度。其判斷的方法為 $MV_{1,2,3,4,5,6,7,8}$ 皆為零的時候成立。但為預防有突然出現的物體，如”Stefan”中的網球，但移動向量也沒有在相鄰找到相似的物體，所以當相鄰的八方向的移動向量都是零的條件下，要再加入 SAD 值比較，預防突然出現的物體造成的錯誤，其 SAD 判斷關係式如(3.18)。所以對突然出現的區塊，將會判斷成 ELA 補償。

$$SAD_{th} = \max\{(SAD)_{2,4,6,8}\} + \min\{(SAD)_{2,4,6,8}\} \quad (3.18)$$

Step 3. 確認一般動作補償區域

第三步是要確認小區域性移動的動作補償。在影像移動四方向一致下，或差異很小時，雖然是區域的動作補償，但避免有移動的物體剛好在區塊內，SAD 會變大的確沒偵測到的缺點。在 Step 1.條件成立下，但不完全都是零移動向量的，需判斷 SAD_5 與值是否大於鄰近區塊的 $SAD_{th1}(v_x, v_y)$ 大，如式(3.19)，如果沒有大於 SAD_{th} ，代表 SAD_5 值並沒有很明顯的大於鄰近區塊的 $SAD_{2,4,6,8}$ 值，是可以使用動作補償，代表它是正在移動中的物體，可以使用動作補償，否則過大於 $SAD_{th1}(v_x, v_y)$ 會是 ELA 的補償。因此比較 SAD 相鄰的大小的方法如下， $\mathbf{M}_{1(x,y)}(n)$ 為輸出的比較結果。

定義 SAD_{th1} 與鄰近相比較的關係式如(3.19)

$$SAD_{th1}(v_x, v_y) = \text{Max}\{SAD_{2,4,5,6,8}(v_x, v_y)\} \quad (3.19)$$

定義 $\mathbf{M}_{1(x,y)}(n)$ 為鄰近相比較的結果的矩陣如(3.20)

$$\mathbf{M}_{1(x,y)}(n) = \begin{cases} 1 & , \quad SAD_5(v_x, v_y) > SAD_{th1}(v_x, v_y) \wedge (D(v_x, v_y)_{neighborhood} \leq 1) \\ 0 & , \quad otherwise \end{cases} \quad (3.20)$$

移動物體的中間部份，由於是比較平均的移動向量，所以移動物中間部份可以用 MC 內插。整體而言，移動的物體區域如果是大於宏塊大小，條件滿足下，其移動物體內部就會是 MC 的補償，但部份的移動物邊緣會以 ELA 補償。如式(3.19)判斷的結果，當 $\mathbf{M}_{1(x,y)}(n)=\text{True}$ ，其比對的結果 SAD_5 大於相鄰的 $SAD_{th1}(v_x, v_y)$ ，如式(3.20)，則可排除過大 SAD 之動作補償，降低不良的移動向量補償，增加顯示品質。

說明這個步驟的區域動作物體補償改善。



(a)原始畫面第 18 張

(b)原始畫面第 19 張

(c)區塊比對動作補償第 19 張

圖 35 應用於解交錯的移動補償的輸出影像圖

以 Stefan 的影像序為例，說明移動的前景補償。圖 35(a)(b)分別為 Stefan 原始畫面第 18 張與原始畫面第 19 張，直接使用動作補償，由於原始影像的變化太大無法從前一張畫面直接補償之下，就會出現圖 35(c)的影響不連續影像。基於這樣的原因，在動作變化比較大的區域影像就得對物體的移動向量與區域的 SAD 值比對確認。

在 Stefan 影像中，由於移動的人物區塊部份移動的方向較不一致，加上 SAD 值的判斷，來決定該區域的 SAD 是否變化很大。如果 SAD 分布範圍與鄰近相比還算接近，代表它是一個區域性物體的移動，將會是 MC，否則變形太大的區域將會轉換為 ELA。但如果物體的邊緣可能部份是靜態，部份是移動物體的經過，則判斷 SAD 也可得知該區域的 SAD 是否過大。所以由人物區域性的快速移動向量中，可以看到的前後張影像趨勢已與周遭物體大不相同，可能是移動的很快的物體，也有可能是已經完全找不到合適的區塊，所以加入 SAD 的判斷就是要排除過大的 SAD 差異，確認是合適的補償。

分析 Stefan 第二十張畫面的移動向量 MV，移動的人物區塊部份因為移動的方向較不一致，所以由判斷 SAD 值來決定該區域的 SAD 是否有著變化很大，如果變化範圍與鄰近相比還算接近，代表它是一個區域性物體的移動，而不是畫面的移動，還是會維持

MC，否則變形太大的區域將會轉換為 ELA。而移動物體的邊緣可能部份是靜態，部份是移動物體的經過，這樣的話將會需要加強移動物體的形態處理。

| | | | | | | | | | | | | | | | |
|------|------|------|-----|-----|-----|-----|-----|-------|------|------|------|------|------|------|------|
| | 12 | 13 | 14 | 15 | 16 | 17 | 18 | | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
| MV5 | 7,0 | 7,0 | 7,0 | 8,0 | 8,0 | 8,0 | 8,0 | SAD5 | 2240 | 2682 | 3102 | 2721 | 3146 | 2602 | 2770 |
| MV6 | 7,0 | 7,0 | 5,0 | 4,0 | 7,0 | 8,0 | 8,0 | SAD6 | 2913 | 2831 | 2347 | 2143 | 2320 | 2457 | 2506 |
| MV7 | 7,0 | 7,0 | 4,0 | 5,0 | 6,0 | 7,3 | 8,3 | SAD7 | 1371 | 1760 | 2531 | 2032 | 894 | 886 | 1465 |
| MV8 | 7,0 | 5,0 | 3,0 | 4,0 | 6,0 | 5,2 | 7,3 | SAD8 | 1033 | 2272 | 4667 | 2942 | 2353 | 2576 | 2689 |
| MV9 | 7,0 | 3,6 | 3,0 | 2,1 | 6,1 | 2,2 | 8,1 | SAD9 | 697 | 1328 | 4371 | 2871 | 2887 | 1422 | 707 |
| MV10 | 7,-2 | 4,0 | 4,0 | 4,0 | 5,1 | 6,0 | 8,0 | SAD10 | 717 | 557 | 2278 | 2831 | 2781 | 478 | 213 |
| MV11 | 7,0 | 2,-8 | 4,0 | 3,0 | 5,1 | 8,0 | 8,0 | SAD11 | 1652 | 371 | 2320 | 2317 | 2344 | 2566 | 473 |
| MV12 | 7,0 | 7,0 | 7,2 | 8,0 | 5,0 | 4,0 | 8,0 | SAD12 | 1385 | 419 | 1503 | 2948 | 2073 | 2883 | 573 |
| MV13 | 7,0 | 7,0 | 7,0 | 7,0 | 7,0 | 4,0 | 8,0 | SAD13 | 780 | 1780 | 2519 | 2295 | 2169 | 861 | 709 |
| MV14 | 3,0 | 8,0 | 5,1 | 7,0 | 6,0 | 6,0 | 0,0 | SAD14 | 142 | 639 | 1766 | 1641 | 808 | 926 | 456 |

圖 36 影像序”Stefan”人物移動區域對應之 MV 與 SAD 值

所以從分析 Stefan 第二十張畫面的 SAD 值，如圖 36。由人物區域性的不一致的移動向量中，可以看到的趨勢已與周遭物體大不相同，可能是移動的很快的物體，也有可能是已經完全找不到合適的區塊。以區塊比對是在無法偵測到一致的整體移動向量時，SAD 判斷的結果又大於鄰近的區域很多，就必須是使用空間性的內插了。

以本文所提出的區塊動作比對可以由移動向量分布與 SAD 值的比對，相比於全域的動作估計。如果影像中間有運動物體的移動，全域動作估計比較無法偵測小物體移動，就還是需要以區域動作估計。

Step 4. 確認移動較大的動態區域補償

第四步是要確認移動差異較大的區域，要找出移動邊緣，並採用 ELA 補償。當鄰近移動向量差異大於 2，且 $(SAD)_5$ 大於等於 $SAD_{in}(v_x, v_y) = \text{Max}\{SAD_{2,4,5,6,8}(v_x, v_y)\}$ ，代表其區域的移動變化較大，如式(3.21)與(3.22)，代表區塊 SAD 值變化很大，是代表差異性較大的區塊或是影像邊緣動態前景與靜態背景交界，採用改良的子像素邊緣線平均內插會比較合適的。

定義 SAD_{th2} 與鄰近相比較的關係式如(3.21)

$$SAD_{th2}(v_x, v_y) = Median\{SAD_{2,4,5,6,8}(v_x, v_y)\} \quad (3.21)$$

定義 $M_{2(x,y)}(n)$ 為鄰近相比較的結果的矩陣如(3.22)

$$M_{2(x,y)}(n) = \begin{cases} 1 & , (SAD_5(v_x, v_y) > SAD_{th2}(v_x, v_y)) \wedge (D(v_x, v_y)_{neighborhood} \geq 2) \\ 0 & , otherwise \end{cases} \quad (3.22)$$

再以圖 37 的為例，它是較大物體的區域移動，影像區域邊緣直接使用區塊比對動作補償直接補償的話，會出現圖示的移動區域邊緣的不連續。從人像移動是區域的移動下，得知移動向量如果不一致的部份，就會代表是影像移動區域的邊緣，也代表在移動物邊緣較無法從前一張畫面完整的補償。



(a) 區塊比對的前一張補償後畫面 (b) 區塊比對的前一張補償後畫面

圖 37 影像”Foreman”的鏡頭物體移動下的差異值比較

而移動物體的邊緣向量在圖 37(a)所顯示的它代表影像的背景以靜態居多，前景的動態影像是一個比較大範圍的移動的區域，移動區域的中間部份還可以使用 MC 補償，但移動物體邊緣就沒有辦法。所以從 SAD 分布判斷移動物體邊緣，如果鄰近移動向量差異很大，鄰近 SAD 比較又比中間分布大，亦表示物體除了移動外，變化程度也稍大，

會讓動作補償回去的效果不好，直接的動作補償的會像圖 37 的補償不連續的狀況產生，所以用 SAD 的輔助判斷影像的可靠度就很重要。

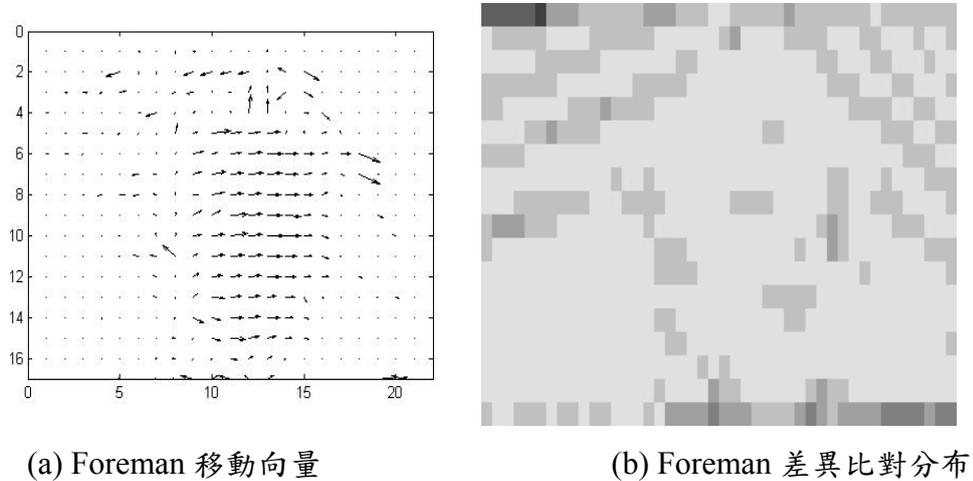


圖 38 影像”Foreman”的物體移動下的 MV 與 SAD 差異值比較

以圖 38(a)所表示人像移動邊緣向量，可比對再圖 38(b)的 SAD 分布圖，顏色越深代表 SAD 越大。可發現因人像物體的移動，其邊緣移動向量相對應在 SAD 的區域上，其 SAD 分布變的比較大，也因為它落在移動區域邊緣，所以它的鄰近 SAD 就會比較大。除此之外，其背景有一些邊緣線的 SAD 其比對的結果也是顯示出 SAD 很大，輸出的品質可能會比較不好，所以背景邊緣線的部份會被判斷成是以 ELA 來補償。

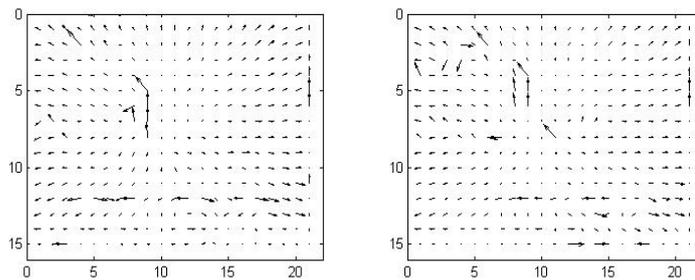
另外 MV 的比對必須是相對的，也就是它與鄰近區域的動作比對，其 MV 可以清楚的顯示出移動物的邊緣，因為移動估計無法比對到很合適的值，所以移動向量就會落在奇怪的位置。因此影像內容有物體移動，會涵蓋到部份的靜態影像，這樣的補償通常會使用空間性內插或是移動形態比較處理，減少可能錯誤的補償。

另外影像動作是鏡頭的縮放(Zooming)，在動作估計比對前後場影像，如果影像會因為鏡頭縮放而使前後場影像放大或是縮小，移動向量的方向是外擴或內縮，而適合的縮放影像補償會是直接的空間性內插。所以這部份為測試動作估計是否能夠找到正確的影像移動方向，確認在決策時能正確判斷補償方法。如圖 39 第 38 張到第 40 張影像鏡頭縮放的畫面，用來說明鏡頭縮放的偵測。底下分析其輸出的移動向量的結果說明。



(a) Tabletennis 第 38 張 (b) Tabletennis 第 39 張 (c) Tabletennis 第 40 張

圖 39 前後影像的區塊比對圖



(a) 移動向量第四張 (b) 移動向量第五張

圖 40 前後影像的區塊比對移動向量圖

以圖 40 的區塊比對的向量分布所示，其影像的動作趨勢大致是向外擴展，所以移動向量會是放射狀的向外增加，而 SAD 值與相鄰比較，也沒有特別的大，也沒有特別小，只有桌球邊緣的區塊 SAD 會比較大。但影像的縮放補償如果要使用區塊的補償，必須是移動較慢的，如影像的中央部份，縮放比例還沒有很大，還有可能使用 MC，但是影像邊界縮放是很快的，使用 ELA 的內插補償會比較適合的。

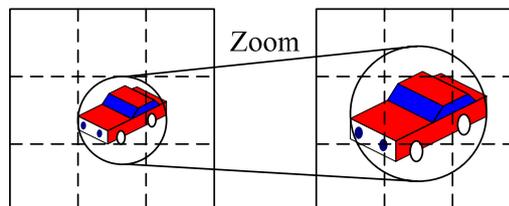


圖 41 鏡頭縮放移動下的補償

以圖 41 說明影像縮放時的補償，中心內容會放大，而 ELA 補償剛好符合空間性

的內插補償，因為放大的影像細節較無法從前面的畫面，再補償到現在的畫面，所以必須是空間性的內插補償。所以本文在 ELA 的改良上才加入子像素的內插，特別是在影像物體縮放的時候，可對影像鄰近點做子像素內插補償，子像素內插的優點會比較細膩，其結果亦優於一般邊緣線內插補償。

動作形態與邊界影像處理

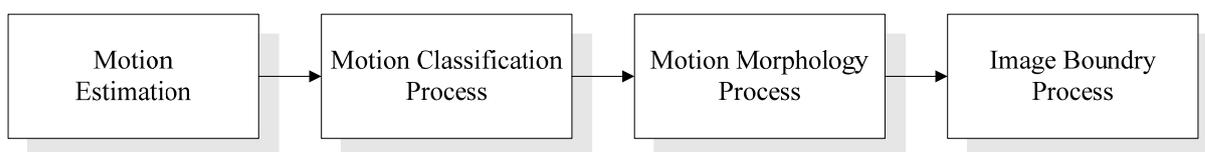


圖 42 動作形態與影像邊緣的處理流程圖

在主要的動作分類處理後，圖 42 為動作形態(Motion Morphology Process)，其為消除移動區域與靜態區域的區塊不連續，稱動作邊緣形態擴展(Morphological Dilation)處理，可消除移動物體的邊緣，這部分的演算法會參考前面已經歸類好的區域，針對移動物體的邊緣設定為差異選擇補償。因為移動估計誤差或所帶來的區塊效應，已利用先前判斷的資訊中處理，這部分主要對動作物體邊緣的再次分類的處理，消除移動區域 MC 補償所帶來的區塊效應。達到更正確的動作補償。

由於影像在物體區塊部份的變化不一，當該區域的背景為是靜態的，只在移動區域邊緣發生，所以必須要有不同的補償方式。由於物體的邊緣區塊的內容如有一小部份會是動態的，有時動作區的大小不一，然而該部份無法造成 SAD 的差異變化，而 MV 也是一致的狀況下，就無法以這兩種資訊判斷出該區塊是否有不連續影像的區域。所對小部份的移動來說，ME 也無法正確的找出合適的方塊，會把它當成靜態的處理，其結果就會看到解交錯的影像有缺了一角的不連續的狀況。所以對移動的區域做移動物體的形態擴展，這個擴展會是選擇為動態邊界的區塊上作移動區域的擴展。

如果該區的 MV 與 SAD 滿足要找的區塊條件，則會把最靠近圖 24 中標示的 BAB

區塊或是 BB 的區域改變為混合補償，該區域也許背景是靜態的，但有一小部份的影像是動態區塊，這時候就要採用影像中間值差異的比對來得到比較適合的補償值，否則仍維持原本補償。

其判斷處理的條件為該區為移動物體邊緣條件成立下，即相鄰的區域已是 ELA 的補償且該區 SAD_5 大於的 SAD_{th3} ，這代表區域可能為移動物體的邊緣，藉此判斷偵測區塊內部份的移動並加以補償。

定義 SAD_{th3} 與鄰近相比較的關係式如(3.23)

$$SAD_{th3}(v_x, v_y) = Median\{SAD_{2,4,5,6,8}(v_x, v_y)\} \quad (3.23)$$

定義 $\mathbf{M}_{3(x,y)}(n)$ 為移動物體形態偵測的結果如(3.24)

$$\mathbf{M}_{3(x,y)}(n) = \begin{cases} 1 & , \quad (SAD_5(v_x, v_y) > SAD_{th3}(v_x, v_y)) \\ 0 & , \quad otherwise \end{cases} \quad (3.24)$$

因為影像背景可能是靜止的，但前景部份會有可能是移動的物體邊緣，導致移動的區塊邊緣有缺角，這個判斷主要就是要補償在移動區域邊緣的小部份缺點。使得影像移動的因小部份的移動向量的問題，所要彌補的缺失。

最後是影像邊界處理(Image Boundary)，邊界處理程序是針對影像邊界的移動做判斷。由決策判斷結果得知該區域是適合動作補償，否則是動作估計無法找到移動向量的部份，採用 ELA 會是比較合適的辦法。當邊界處理好，輸出 $\mathbf{D}(n)$ 參數矩陣給動作補償，設定給動作補償的所採用的方式。依據影像的決策 $\mathbf{D}(n)$ 參數設定，輸出所要的影像。下一節將說明影像邊界改善處理的細節部份。

3.6. 邊界區域的補償改善

因為動作估計在邊界(Boundary)區塊通常沒有辦法像內部區塊一樣，有完整的搜尋範圍，可找到正確的移動向量，所以本節針對邊界影像的動作補償做說明與對應決策。

3.6.1. 邊界區塊的移動向量誤差

如圖 43 所示。在影像移動中，邊界通常是誤差也會比較大，且可能無法找到正確的移動向量，從圖 44 比對的移動向量可看出邊緣移動向量會是有誤差的，所以無法從前一張畫面補償回來。為了有更好的影像的邊界補償，必須對影像邊界的移動向量作可靠度判斷，確認該移動向量是可以適合使用動作補償的，否則必須使用空間性的內插補償，避免錯誤的移動向量造成影像的不連續。

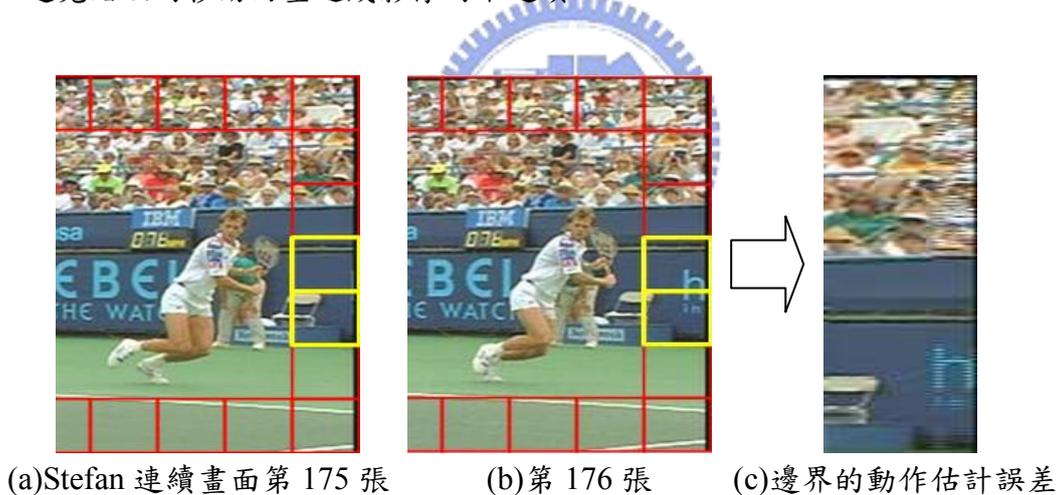


圖 43 搜尋區域邊界區塊的補償問題

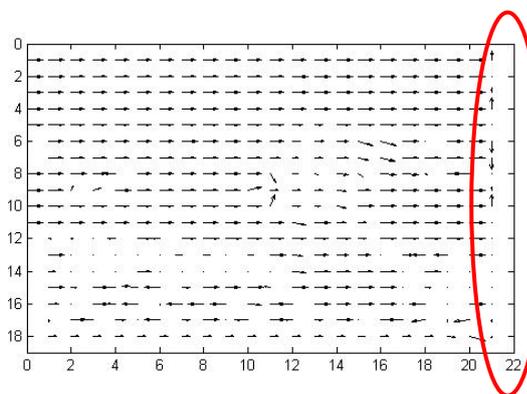


圖 44 搜尋區域邊界區塊的移動向量

3.6.2. 改良邊界區塊的補償

一般邊界的移動向量，移動的比較快或內容變化較大時，其部份的移動向量誤差會很大，為了保留邊界準確的移動向量，邊界的移動也要判斷，並配合適當的補償方法。邊界影像判斷方法如圖 45，鄰近向量依位置而分，以上方中間的邊界為例 MV_5 ，其相鄰的移動向量為 MV_8 ，參考的差異值 $D(v_x, v_y)_{neighborhood}$ 分別為 MV_5 與 MV_8 計算其差異，再判斷移動向量是否相近，如果相近則採用動作補償。如果 MV_5 差異鄰近移動向量很多，則是使用ELA內插，其他的邊界區域以此類推。也就是當影像邊界的移動向量是跟鄰近的移動向量完全相同時才採用MC補償。因為影像邊界不確定出現的新影像是否在上一個畫面可以搜尋到，所以如果MC的補償條件不是很充足下，使用ELA去補償邊緣區塊的動作，會比MC的補償好。而在靜態背景下由於移動向量為零，比較的結果會是直接以圖場間內插補償。

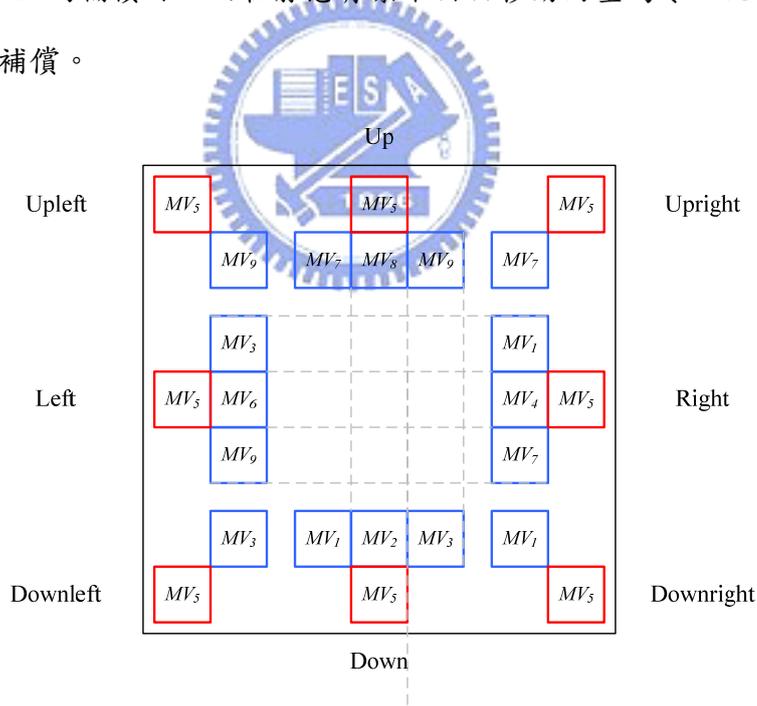


圖 45 影像邊界的移動向量與鄰近位置關係判斷

定義邊界移動向量決策輸出關係式(3.25)

$$B_{(x,y)}(n) = \begin{cases} 1, & D(v_x, v_y)_{neighborhood} \leq 1 \\ 0, & D(v_x, v_y)_{neighborhood} \geq 2 \end{cases} \quad (3.25)$$

其邊界相鄰的移動向量差異為 $D(v_x, v_y)_{neighborhood}$ ， $D(v_x, v_y)_{neighborhood}$ 以其相對位置表示差異關係，分別以 $D(v_x, v_y)_{up,down,left,right,upleft,downleft,upright,dowright}$ 代表，其關係式為(3.26)~(3.33)。

$$D(v_x, v_y)_{up} = \{MV_5(v_x, v_y) - MV_8(v_x, v_y)\} \quad (3.26)$$

$$D(v_x, v_y)_{down} = \{MV_5(v_x, v_y) - MV_2(v_x, v_y)\} \quad (3.27)$$

$$D(v_x, v_y)_{left} = \{MV_5(v_x, v_y) - MV_4(v_x, v_y)\} \quad (3.28)$$

$$D(v_x, v_y)_{right} = \{MV_5(v_x, v_y) - MV_6(v_x, v_y)\} \quad (3.29)$$

$$D(v_x, v_y)_{upleft} = \{MV_5(v_x, v_y) - MV_9(v_x, v_y)\} \quad (3.30)$$

$$D(v_x, v_y)_{downleft} = \{MV_5(v_x, v_y) - MV_3(v_x, v_y)\} \quad (3.31)$$

$$D(v_x, v_y)_{upright} = \{MV_5(v_x, v_y) - MV_7(v_x, v_y)\} \quad (3.32)$$

$$D(v_x, v_y)_{dowright} = \{MV_5(v_x, v_y) - MV_1(v_x, v_y)\} \quad (3.33)$$

由(3.26)~(3.33)各個邊界不同位置的差異計算後，如果 $D(v_x, v_y)_{neighborhood}$ 在對應位置上差異很小，其式(3.25)的 $B_{(x,y)}(n)=True$ ，代表該位置的移動適合動作補償，否則將使用 ELA 的內插。對於影像邊界的補償，只在影像是整體移動的狀況下，才會發生。簡單的說，影像為動態的狀況下，邊界相鄰的移動向量變化很大時，代表可能找不到相似的區塊，但不是所有的邊界都找不到。所以當動作估計在邊界區塊的比對，由相鄰的移動向量做判斷，如果相鄰的移動向量差異都很大，則邊界的補償可能會是不可靠的，則以空間性的補償，否則維持原本的動作補償。

3.7. 本章總結

整體而言，適應性混合方法的動作補償解交錯，以動作估計改善動作偵測的缺點，從動作的資訊比對影像的差異，再採用動作估計的資訊做判斷決策，決定影像補償的方式。以動作估算所得到的 MV 與 SAD 的相鄰關聯性來確認移動向量的可靠度，以增加比對的準確性。有別於單獨判斷 MV 或 SAD 的關係，使用計算鄰近的 MV 和 SAD 差異大小的關係進行決策，加上雙重的比對可增加適應性補償的效果。

因為動作估計所要的影像必須是最接近的，所以影像在搜尋範圍內的話，相鄰區塊的 SAD 值沒有變化很大，則可確認補償的相似度會比較可靠。但整體相鄰的 SAD 值要是分布的很亂，代表該區域的動作可能已經超出搜尋範圍。或者是影像的變化量太大，無法找到準確的移動向量，就會使得移動補償效果不好，所以可由這樣的關係增加本文對動作補償判斷的資訊，並適應性改變影像的動作補償方式，進而改善移動向量不準確下的補償方法。

而移動邊界的區域作處理上，如果邊界的移動向量是和鄰近接近的話，其動作補償會是可靠的。但如果因為動作估計在邊緣會無法找到有效的移動向量，影像區塊無法在前一個圖場所找到，其移動向量的變化會與相鄰的移動向量不同，代表其動作估計會有誤差。所以對邊緣的移動向量判斷上，如果補償的移動向量不可靠，將使用空間性的內插。所以邊緣線內插也加入了子像素與改良的判斷方式，當影像是區域的移動，其邊緣區域會有靜態背景與動態前景，所以使用邊緣線內插的效果將會比較好，配合移動資訊的決策判斷，可使移動的邊緣區域補償比較細緻。

第四章模擬結果與比較

4.1. 影像效能評估

客觀比較指標採用PSNR(Peak signal-to-noise ratio)為主，其客觀量化指標MSE定義如(4.1)，PSNR關係式為(4.2)。評估方式如圖 46所示，評比的方法為輸入循序式影像 $F_{ori}(x, y)$ ，然後依據交錯式影像的規格產生出交錯式的影像 $F_{dei}(x, y)$ ，再經由本文的演算法來進行解交錯的運算，最後與輸入的循序影像比較，計算出量化指標PSNR。

MSE定義如(4.1)及PSNR關係式如(4.2)。

$$MSE = \frac{\sum (F_{ori}(x, y) - F_{dei}(x, y))^2}{MN} \quad (4.1)$$

$$PSNR = 10 \log_{10} \left(\frac{255^2}{MSE} \right) \quad (\text{單位: db}) \quad (4.2)$$

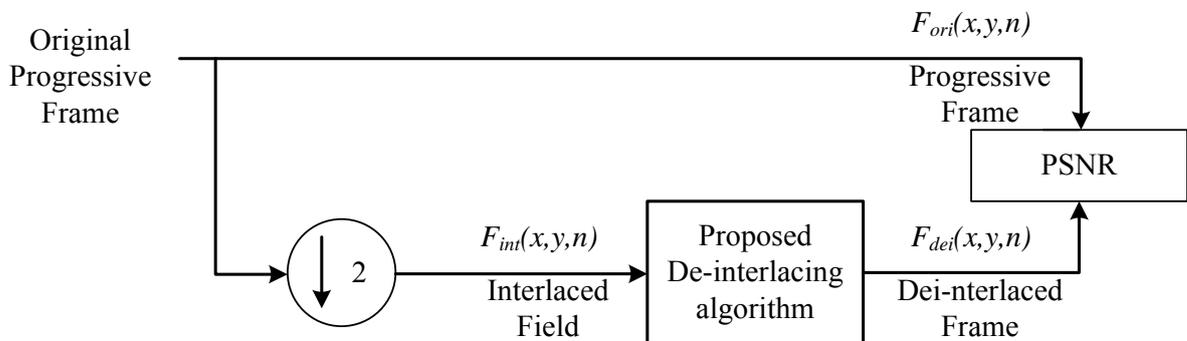


圖 46 解交錯效能比較方法

以PSNR數據為客觀指標，評估演算法效能，比較輸出的影像效果為主觀的評估，所以實驗結果也加入了輸出影像的影像細節上的差異比較，做不同方面的評比。

實驗模擬的平台以 C/C++ 6.0 編寫所有程式。採用 INTEL T2300 1.6G 完成所有的模擬。

4.2. HD 測試影像序與結果分析

高解析度測試影像序(High Definition Video Sequence)內容說明如表 1。

表 1 高解析度測試影像序內容分析

| HD評估影像 | 內容簡述 |
|---|---|
|  | 測試影像序“Parkrun”為在公園裡移動的人且手拿雨傘。背景樹，雪和水的倒影，都是影像細節要求非常的高的影片。而人的影像與背景有點相似，且鏡頭是跟著人移動。 |
|  | 測試影像序“Stockholm”為水平移動拍攝斯德哥爾摩的風景。有細節較高的房屋與窗戶，移動的車子，微小的街景。 |
|  | 測試影像序“Mobcal”為移動的日曆文字和細節很多的瓦薩號照片。背景有兩種類型的壁紙，物體的向下移動與鏡頭緩慢的後移，也算是影像內容細節相當高的影片。 |
|  | 測試影像序“Shields”主播男子穿著格紋外套移動著，外套上有著高頻成份。牆上有詳細的騎士盾牌等細節很多的背景。最後會停格並放大手指部位。 |

(1) 高解析度影像解交錯輸出 PSNR 比較

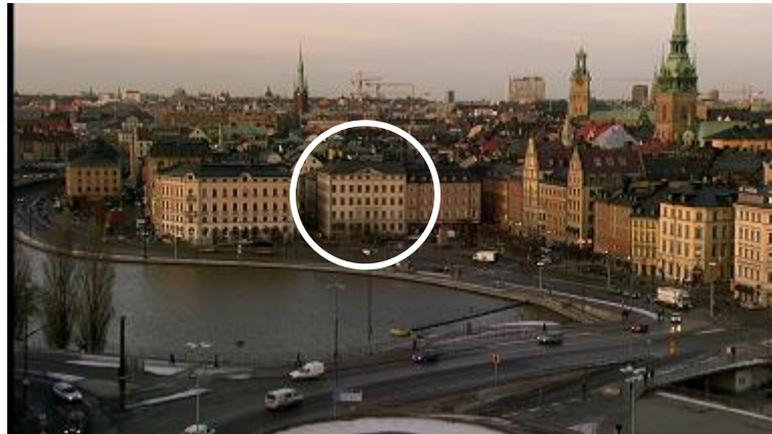
表 2 為 HD 影像測試的輸出影像的 PSNR 比較表，比對常用的做法與本文的方法，比較結果顯示如下，均可改善其輸出的效能。

表 2 高解析 HD 解交錯影像的 PSNR 平均比較表

| | Line Double | Line Average | Median Filter | Merged | 5taps ELA | VT Filter | 2 Field MC | Propose |
|------------------|----------------|-----------------|------------------|--------------|--------------|--------------|---------------|--------------|
| Parkrun | 28.07 | 31.43 | 30.37 | 22.67 | 27.06 | 30.09 | 31.97 | 32.50 |
| Stockholm | 26.12 | 29.08 | 30.75 | 36.70 | 24.85 | 30.69 | 29.72 | 34.38 |
| Shields | 29.94 | 33.05 | 32.15 | 25.81 | 29.14 | 32.7 | 33.61 | 33.99 |
| Mobcal | 23.06 | 25.96 | 24.37 | 19.13 | 21.3 | 24.1 | 26.08 | 27.42 |
| Average | 26.79 | 29.88 | 29.41 | 26.07 | 25.58 | 29.39 | 30.34 | 32.07 |

整體來說，高解析度的影像序主要是細節上的要求比較高，一般的 MC 做法其 PSNR 輸出平均的輸出約為 30 db，而本文的演算法其平均的 PSNR 約為 32 db，大約改善 1~2 db，其詳細的輸出顯示比較說明如下。

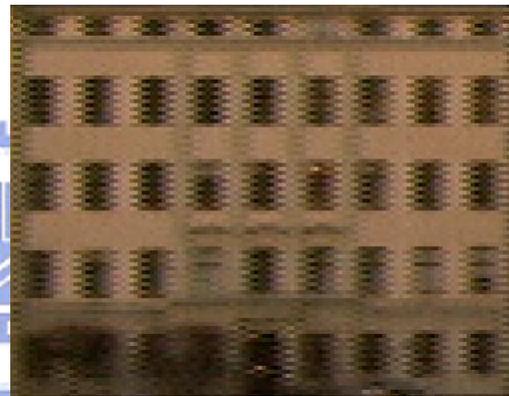
(2)Stockholm



(a) “Stockholm”輸出影像第 27 張



(b) 原使影像放大



(c) 前後圖場內插(FI)



(d)中間值邊緣線內插(M-ELA)



(e)適應性混合解交錯(Proposed)

圖 47 影像“Stockholm”輸出結果比較



(a) 適應性混合解交錯(Proposed)



(b) 垂直時間濾波(VT Filter)

圖 48 影像“Stockholm”房屋邊緣細節的輸出比較

圖 47(a)是“Stockholm”影像的輸出比較，主要觀察影像細節與整體的部份是否有補償得當。因為鏡頭是水平方向的移動，所以影像中間房屋是以動估計補償為主，因為該部份高頻成份較高，且影像的下方因為影像內容中有不規則的物體移動著，所以該部份差異較大的影像補償會是 ELA 為主。由圖 47(b)是原始影像的局部放大，圖 47(c)圖場內插方法，可清楚看見輸出影像是嚴重的鋸齒邊。圖 47(d) 中間值邊緣線內插，這類的輸出雖然可預防邊緣鋸齒邊，但邊緣的陰影還是會因中間值濾波而有顆粒狀的殘影。圖 47(e)是本文的方法，是移動畫面的動作補償，整體結果較為接近原始影像。

由於 HD 的影像解析度高，很細微的部份如果沒補好，放大檢視會很容易看到。圖 48(a)為本文的結果，由影像中央的屋頂比對不同方法的結果是否有補償到平滑邊緣，這部份的邊緣是斜線邊緣，本文的方法在建築物邊緣的一些細微的線條可以很平順的內插，圖 48(b) 是垂直時間濾波器的結果，相對的它的輸出就會造成不好的效應，影像就無法很正確的內插補償回去。

整體來說“Stockholm”的內容雖然細膩，但如果沒補償好，在播放顯示時加上影像緩慢的移動就很容易看到這種鋸齒狀邊緣的缺點或內插誤差產生的人為效應(Artifact)，所以細節上的影像也是要注意的。本文因為適應性的補償，在垂直細節是比較高的時候，動作估計得到的 SAD 也會比較高，移動向量不一致時，則 ELA 會比較優先，可使整個影像的補償輸出看起來良好，其部份比較細微的影像也有不錯的顯示品質。

(3) Shields



(a) “Shields”輸出影像第 27 張



(b)原始影像放大



(c)動作適應性補償(MA-IPC)



(d)垂直時間濾波器(VT Filter)



(e)適應性混合解交錯(Proposed)

圖 49 影像“Shields”輸出比較

圖 49 (a)為影像序“Shields”的解交錯解輸出結果。比對細部影像的差異，圖 49(b)為原始影像的局部放大。圖 49(c)為動作適應性的補償結果，可明顯看出手部邊緣線的鋸齒狀相當明顯。圖 49(d)為垂直時間濾波器，有重疊的影子在，看起來相當不自然。圖 49(e)是本文的方法，背景是移動畫面的動作補償，前景是移動物體的邊緣線補償，整體輸出的結果與原始影像比較，其補償效果也是良好的。

就整體動作而言，“Shields”影像的移動約在 4~5 個像素(pixel)間。因此在移動的狀況下，區域動作偵測的比對如果太小，容易造成很大的誤差，相對的邊緣內插得到的影像也沒有很平滑。而本文有動作估計的補償，在高頻的條件下會得到比較好的影像，而不是得到模糊的影像，而邊緣也有 ELA 針對影像邊緣作加強，所以會有比較好的品質。

(3)Parkrun

圖 50 (a)為影像序“Parkrun”的解交錯解輸出結果。由於“Parkrun”是將鏡頭中心定在移動的人物上面，鏡頭跟隨人移動，所以背景的變化就變的很快，如圖 50 (b)為圖場內插輸出結果，可明顯看出其背景影像由於移動快速，直接合併圖場會使背景出現鋸齒狀。圖 50 (c) 為中間值邊緣線內插，但太細的線條卻會使得中間值的內插錯誤，造成影像邊緣有些微的鋸齒邊。圖 50 (d)為垂直時間濾波器的輸出結果，由於是時間上的平均關係，使得輸出有重疊的影像感，造成影像會在細節上不是很好。圖 50 (e)為本文所提的方法，相比於其他影像的輸出結果，其品質也比傳統方法好。



(a) “Parkrun”輸出影像第 16 張



(b) 圖場內插(FI)



(c) 中間值邊緣線內插(M-ELA)



(d) 垂直時間濾波器(VT Filter)



(e) 適應性混合解交錯(Proposed)

圖 50 影像“Parkrun”輸出比較

4.3. CIF 影像序測試結果與分析

本節比較CIF的測試影像輸出結果。針對幾個比較不同種類的影像序比較其輸出影像品質。要比較的影像種類中，有快速移動的影像，鏡頭平移的影像，鏡頭縮放的影像，鏡頭固定於特定物體的影像，固定人像的影片，形態不固定的影像等，以連續影像的處理測試顯示品質與效能，針對這些影片評估並說明演算法輸出的影像品質。

(1) CIF的PSNR比較表

同樣以 PSNR 為客觀的比較，比較 CIF 測試影片所得到 PSNR 的 db 值結果為主。影像效能 PSNR 比較的評估結果如底下的表 3 所示。

表 3 各種方法的 PSNR 平均比較表

| Video Sequence | Line Double | Line Average | Median Filter | Merge | ELA | VT Filter[2] | 2 Field MC | 2 Field MA | 5 field MC[23] | 4field MC[21] | Proposed |
|------------------------------|--------------|--------------|---------------|--------------|--------------|--------------|--------------|--------------|----------------|---------------|--------------|
| Stefan | 23.19 | 27.45 | 26.26 | 21.01 | 23.09 | 25.89 | 27.15 | 27.48 | 25.01 | 26.12 | 28.12 |
| Foreman | 28.66 | 30.97 | 32.54 | 28.70 | 30.40 | 31.72 | 33.31 | 29.94 | 33.22 | 34.37 | 34.53 |
| Mobile | 21.29 | 25.38 | 25.19 | 23.45 | 20.73 | 26.49 | 25.36 | 25.47 | 24.12 | 25.93 | 26.86 |
| Coastguard | 24.58 | 28.47 | 29.37 | 27.16 | 26.09 | 30.21 | 29.60 | 28.92 | 32.69 | 28.83 | 29.51 |
| Container | 26.00 | 30.01 | 32.25 | 38.81 | 28.10 | 36.12 | 32.75 | 30.84 | 41.14 | 40.17 | 36.94 |
| Silent | 29.05 | 31.41 | 33.51 | 33.19 | 29.33 | 34.22 | 36.30 | 31.96 | 42.11 | 40.37 | 39.31 |
| Mother & Daughter | 32.62 | 36.24 | 38.64 | 37.97 | 33.34 | 38.96 | 40.71 | 37.85 | 39.69 | 44.83 | 41.01 |
| Table Tennis | 24.46 | 28.27 | 29.48 | 28.55 | 25.86 | 31.93 | 33.03 | 30.29 | 27.16 | 35.20 | 33.31 |
| Hall | 26.09 | 29.92 | 31.81 | 33.27 | 26.62 | 34.07 | 34.34 | 32.53 | 39.68 | 40.04 | 37.09 |
| News | 28.58 | 33.86 | 35.31 | 36.23 | 24.76 | 39.68 | 34.28 | 33.87 | 41.14 | 40.20 | 39.48 |
| Average | 26.45 | 30.19 | 31.43 | 30.83 | 26.83 | 32.92 | 32.68 | 30.91 | 34.59 | 35.6 | 34.61 |

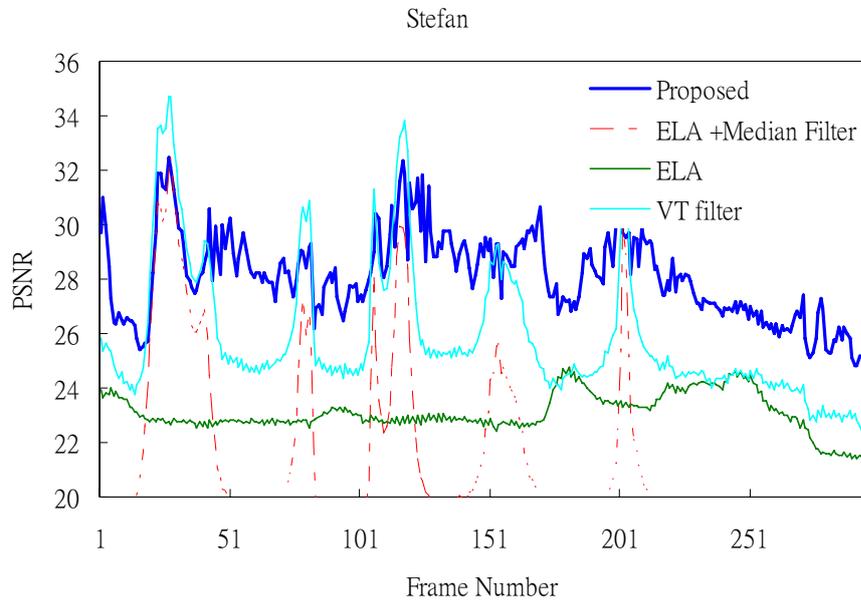
PSNR 的比較總表如表 3 所示，在本文的實驗中以評估傳統的標準做法與其他文獻中所提出的方法做量化數值的比較，比較常被比較的做法有平均內插解交錯(Line Average)，中間值濾波解交錯(Median Filter)，邊緣線平均法(ELA)，垂直時間濾波器(VT

Filter)，及[21][23]等之文獻結果做比較，及本文的方法(Proposed)為雙圖場適應性混合解交錯,其模擬結果顯示雙圖場所設計之動作補償解交錯，在靜態影像的改善較小，但動態影像內容較多的動作補償可以有較好的表現，雖然沒有多圖場的動作估計平均的 PSNR 高，但其硬體代價可降低許多，與一般動作補償解交錯比較也約有 2db 的改善。

(2) Stefan

從分析”Stefan” 連續解交錯影像PSNR變化程度，結果如表 4，可得知不同的方法其輸出的PSNR皆會造成影像品質變化很快，無法有效維持在相同水準，以本文提出的動作補償可縮小影像品質的變化幅度，其他方法如垂直時間濾波器或是中間值濾波，其PSNR的變化幅度都很大，差異變化大的時候有將近有10db的變化，而本文的適應性的動作補償可降低這樣的變化，在影像內容移動較快時，其PSNR可保持較小差距。

表 4 連續影像”Stefan”的 PSNR 比較表



影像後半部的PSNR快速的掉落是因為影後段有快速的移動，而這樣的移動其背景已經開始模糊，ME已經無法找到合適的區塊補償，所以決策部份會判斷成都是ELA的方法完成，雖然比前面影像的品質差一些，但仍比一般的內插效果好。

由於Stefan是一個快速畫面移動的影像序，於圖 51(a)的差異比較圖中可看出，背景的觀眾為高頻信號，因為使用線性內插法會產生較模糊的觀眾影像，圖 51(b)(c)也是相同的狀況，所以這部份使用動作移動補償會有比較高的解析度，由於動作估計有搜尋出前場最接近的像素，所以補償回去可以得到比較好的結果。整體看來，比較各影像圓圈處可明顯看出，圖 51(d)中本文的解交錯演算法在影像在邊緣線或移動背景可以有不錯的表現，雖然Stefan包含單獨物體移動，與高頻成分很高的背景混合，但判斷動作移動資訊，有適當的決策，才有適當的分出補償區域，也才会有比較好的效果。

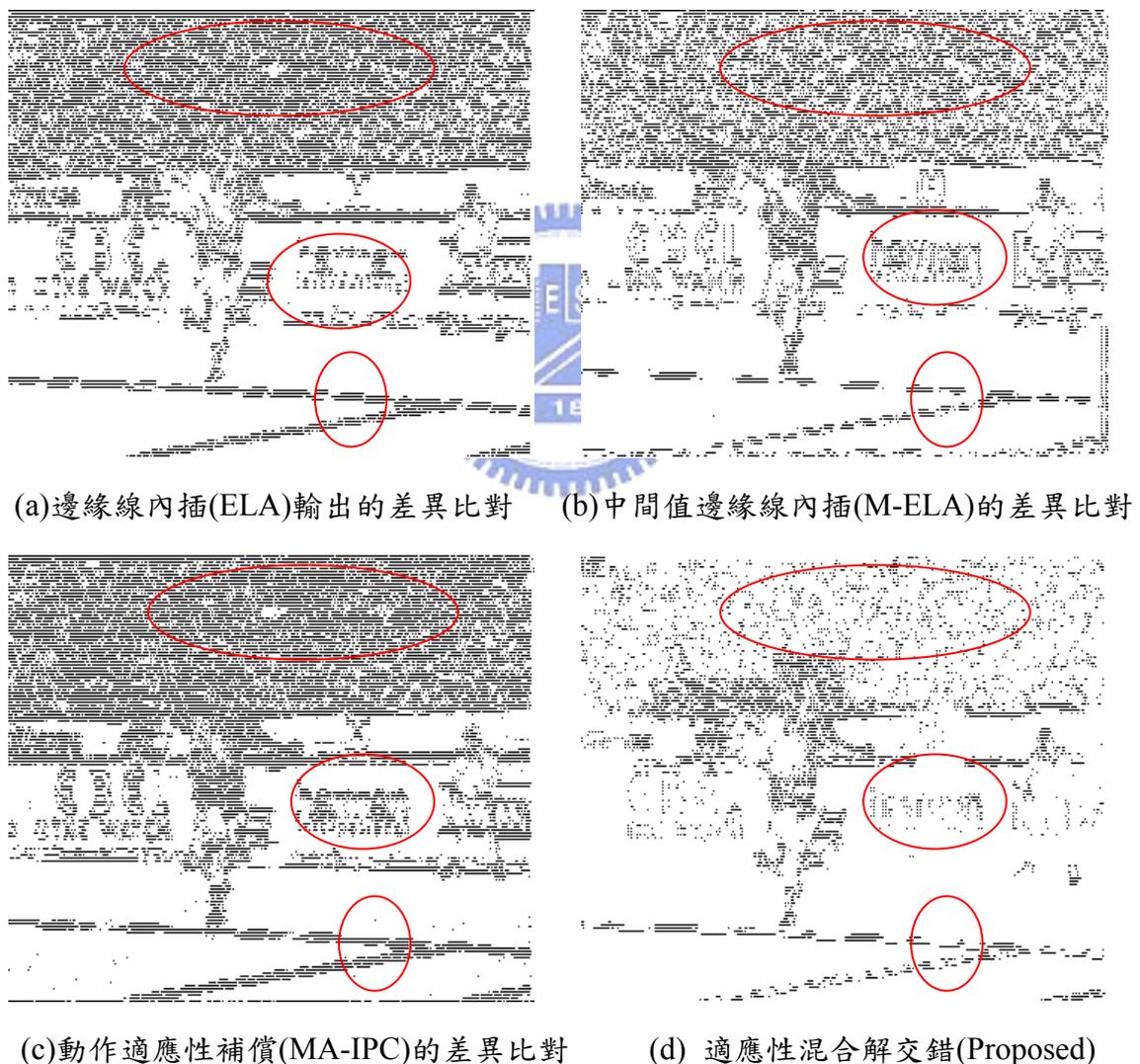


圖 51 比對”Stefan”輸出與原圖影像差異



(a) 原始影像第26張



(b) 動作適應性補償



(c) 邊緣方向中間值濾



(d) 適應性混合解交錯(Proposed)

圖 52 影像“Stefan”輸出結果比較

實際的比較結果如圖 52 所示，以不同方法比較 Stefan 的解交錯輸出，前面表 4 亦也說明其加入適應性的動作補償後，輸出品質可維持高於一般線性內插得到的畫質，且實際輸出影像可看出，其背景高頻部份看起來比較不會模糊。相較於其他方法，不管移動快慢，動作適應性的計算動作差異(Motion Detector)並不能夠得知影像物體的移動方向，僅能知道差異超過臨界，即判斷為動態區域，於是使用一般線性內插，但線性內插的方法又會致使影像模糊，所以一般的圖場動作差異計算上只使用於邊緣線內插較有效，在高頻影像移動下效果就會比較不好。



(a)方塊比對誤差處理



(b) 適應性動作估計(Proposed)

圖 53 方塊比對誤差處理

另外由圖 53(a)中為區塊比對誤差，由於移動內容小於區塊，致使輸出的補償會有不連續的影像發生，所以在這部份的比對上就需要加入相鄰的SAD比對結果。由移動物體的移動量來說，如果移動的方向不一致，再判斷SAD相鄰比較值的差異，SAD如果小，代表該區域比對的影像的差異是很小的，動作補償還會是可靠的。但移動向量很大或不一致，SAD所比對的值又是差異大的，代表其影像的比對是比較差的影像，變形量比較大，所以為避免移動向量不準確的狀況下，採用ELA補償比較可靠，其輸出顯示如圖 53(b)。

整體來說，Stefan 的動作算是局部的移動的物體混合移動背景，其物體的移動的區域可以適應性的變化補償方法，背景則是一般的動作補償，適應性搭配適合的補償，所以輸出顯示品質才會比較好。

(3) Foreman

測試影像”foreman”是一個背景移動較慢，只有前景的頭像在移動，這個影像主要看的是背景牆壁的邊緣是否有鋸齒狀，以及前景人像的移動補償。由於邊緣仍有微小的移動所以要是使用時間上內插法就會有鋸齒狀的邊緣產生，如圖 54(b)。使用一般的邊緣線平均內插配合中間值濾波的方法看起來會有不錯的效果，但在線條的細節上會有微小的差異，如圖 54(c)所示，線條的顆粒會比較大一些。最後是本文的方法，如圖 54(d)，在線條微小細節會比一般邊緣線平均配合中間值濾波好一些。



(a) 原始影像截圖位置



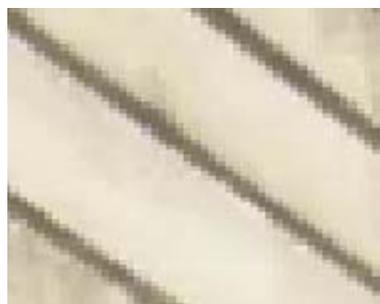
(b) 原始影像的邊緣細節



(c) 一般邊緣線平均法



(d) 邊緣線平均法與中間值濾波



(e) 本文之邊緣線的補償(Proposed)

圖 54 影像”Foreman”邊緣內插的結果比較

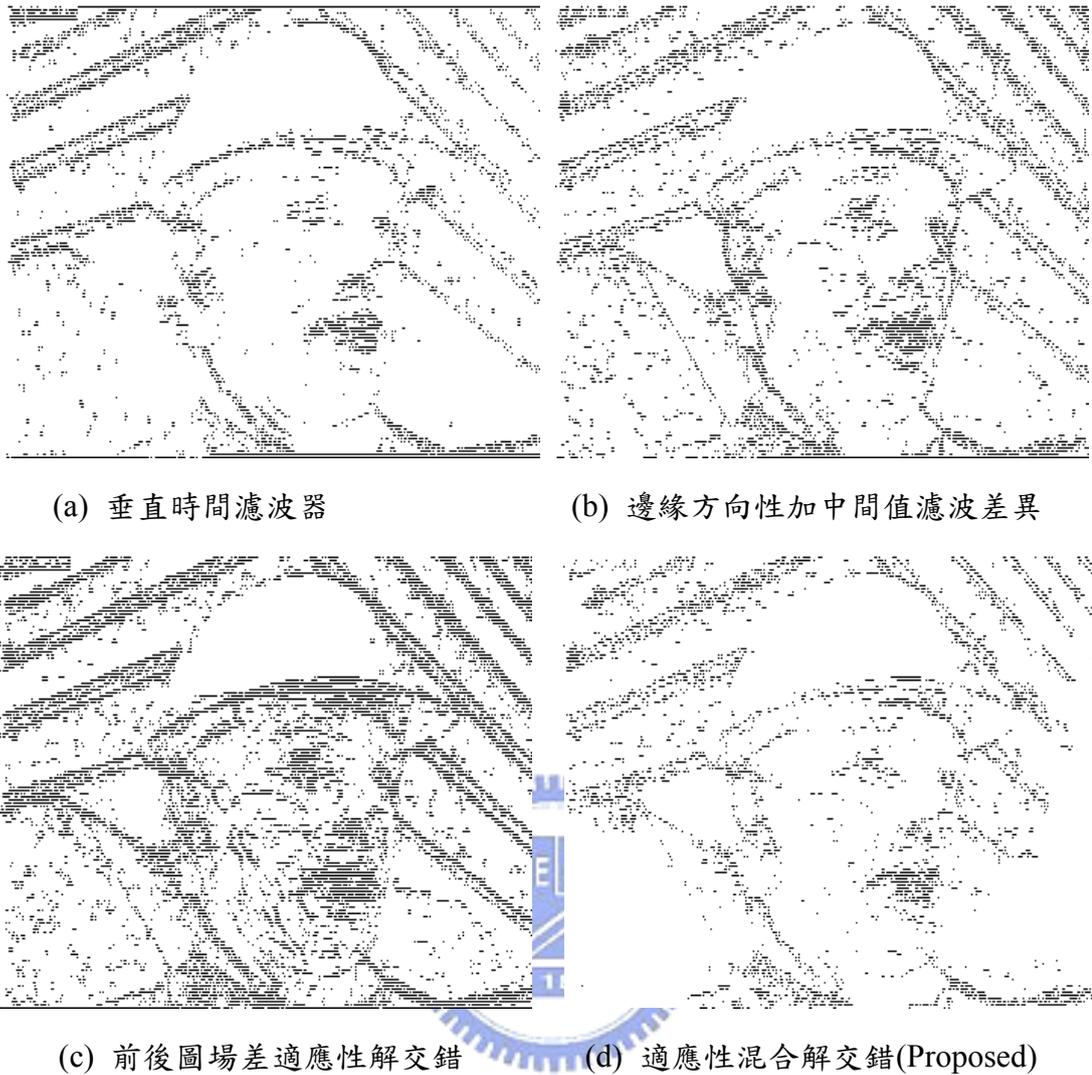
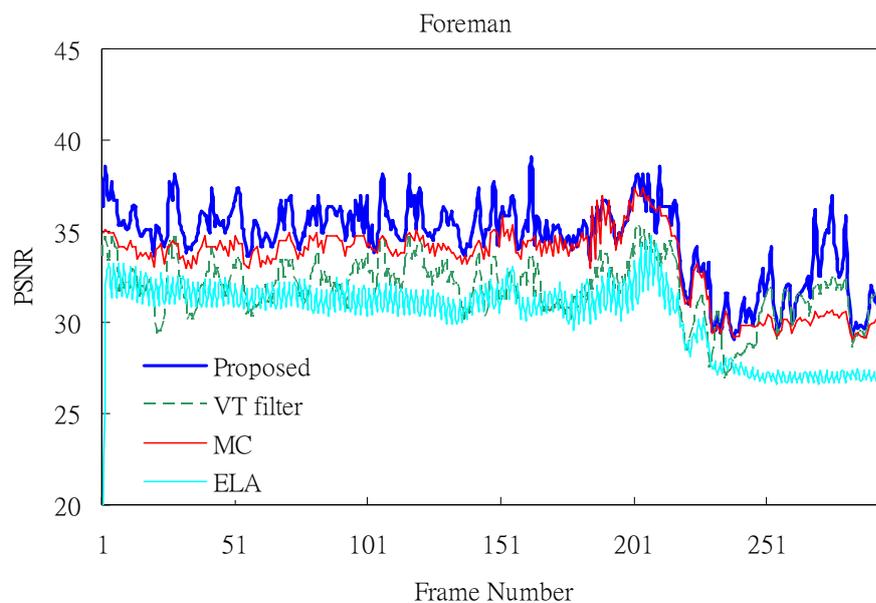


圖 55 比對”Foreman”輸出與原圖影像差異

圖 55 所表示為 Foreman 原始影像與解交錯後的差異圖，Foreman 會因為背景建築物會有邊緣線的緩慢移動，所以完全並不是靜止的，所以在邊緣線也有比較大的差異，也因為這樣所以更要加強 ELA 的補償，以達最好的效果。另外人像的移動也是關鍵，這是因為人像皮膚的影像差距較小，所以要注意動作估計補償時其影像是否有區塊效應在，所以在比較中間移動變化較大的區域，如眼睛鼻子等變化量較大的區域下，採用 ELA，來達比較好的效果。圖 55(a)所表示的，垂直時間濾波器輸出影像會使用前場的資料，與現在的資料作權重作計算，但由於上述邊緣是有為為移動的，加上垂直時間濾波器只有使用垂直同一方向的資料，導致其權重無法抵消，而加權到現在所計算的

值，所以輸出會看到邊緣區域影像都會比較模糊。圖 55(b)所表示的，邊緣線內插加中間值濾波，其缺點是由於中間值濾波器的輸出會有影響內容跳動的現象，導致其邊緣的輸出有時選到 ELA 的補償，有時候選到上下值，這樣的結果便會使輸出邊緣有時補償到對的值，有時補償到更差的值。圖 55(c)為圖場偵測適應性解交錯，其因為邊緣有些微移動導致其出現圖場差，進而使用平均或其他線性內插都會使得其差異都集中在背景的邊緣線。而圖 55(d)為本文的方法有加入了判斷 SAD 值，其 ME 搜尋到的區域其如果 SAD 與相鄰比較上差異很大，雖然移動向量差異很接近，但 SAD 大已經代表補償的效果會不好了，所以需要切換使用 ELA 來得到更適合的補償。

表 5 連續影像”Foreman”的 PSNR 比較表



而 PSNR 之表現可由表 5 顯示出，約在第 200 張畫面後影像鏡頭開始移動，最後畫面鏡頭會停在一個畫面高頻成份比較高的地方，但是影像拍攝時是手持拍攝，影像還會微微震動，加上影像內容高頻成份比較高，所以整體的影像補償結果的 PSNR 值會比較低一些，但不管是動作補償或是 ELA 的作法該區域的顯示效果至少還有 ELA 的內插效果，可檢測演算法在移動太快時，內容模糊的補償會是 ELA 的內插方式，不會看到移動向量錯誤的影像補償所造成的顯示不良，並保有動作偵測補償的優點在。



(a) 原始的



(b) 圖場合併



(c) 邊緣線平均法與中間值濾波



(d) 適應性混合解交錯(Proposed)

圖 56 影像“Foreman”第 15 張輸出結果比較

整體來看，觀察輸出影像來檢測整體影像品質，在圖 56 比較中，各種不同補償的比較圖，可觀察到以圖場合併法是最簡易的方法，如圖 56(b)，它對靜止影像的部份可以完全的補償回去，但是物體只要有移動就會變的很差，圖場合併法只適合靜態部份，動態影像的效果就很差。如果選擇前後場附近區域的影像做處理，因沒有移動資訊的輔助，即使採用權重的濾波處理，其效果皆會變成與空間性的內插相同，會有模糊的影像感，如同圖 56(c)，整體的影像雖沒有像時間性的內插方法一樣，出現不連續的交錯線，但有些不動的字體如果也採用空間性的內插，其結果就會比較模糊。而本文方法的結果在圖 56(d)在字體上因為移動向量為零，也可從前一個圖場補回正常的影像。

(4) Table Tennis



(a) 原始影像



(b) 動作適應性



(c) 中間值濾波器



(d) 適應性混合解交錯(Proposed)

圖 57 影像”Table Tennis”輸出結果比較

測試影像”Table Tennis”的補償主要有移動的手與桌球，再看桌子的邊緣為白線。桌球由於是上下動的，可在圖 57(b)中看見，動作適應性的動作偵測會在移動快速區域中，因為臨界值的設定不好，讓畫面變成前後場內插，致使桌球中間會有錯誤的補點產生，且Table Tennis還有另外一個現象就是鏡頭的縮放，鏡頭的縮放會讓影像邊緣看起來有放射線的產生，其原因為背景被放大或者是縮小了，又被重疊起來，就會出現這種狀況。另外其他方法加入ELA與中間值濾波，就像圖 57(c)，桌子邊緣還是會出現因為中間值濾波上下點不一致而產生的不連續感，另一方面球與拍子也算是移動較快的物體，且原

始影像就有些的殘影，為確保桌球的移動邊緣可以有好的補償，不會因為MC補償成錯誤的值，或因移動太大，且變形量太大的，在本文的處理物體的邊緣決策下，移動區塊的邊緣會加有移動的形態擴張，太相鄰移動區塊的邊緣區塊會被判斷為ELA，以消除不連續的影像，避免動作補償的區塊邊緣會有不連續的現象，如圖 57(d)所示的，並不會有不連續的狀況出現。

分析影像內容的動作，由於球的移動會超過了動作估計的區域。在解交錯中，使用動作估計的搜尋範圍，在本文中使用了 ± 16 的範圍，超過這個搜尋範圍，其影像通常是無法補償的很好，因為影像內容會改變很多，即使是相對的SAD是最小的，其區塊補償也可能會出現影像不連續狀，這與影像壓縮中的動作估計不同，影像壓縮所需要的是找到相類似的區塊，以便消除時間上的冗餘性，解交錯的動作估計是必須找到非常相似的補償回去，否則會有不連續的區塊發生。

除了上面所述移動的前景物體外，其背景因為包含許多高頻成份，部份的背景包含有移動的物體，致使移動向量很亂，SAD也很大，因為在MC的部份，在鏡頭的Zoom in/Zoom out下，由於物體會被放大縮小，這部份的影像放大縮小的動作補償並沒有加入特別的處理來補償這一類的動作，因為即便知道影像鏡頭的Zoom in/out下，放大縮小的補償還是靠空間性內插法使影像來達到補償，所以在這種狀況下，切換為ELA即可彌補因為影像縮小放大的狀況下，MC無法從前一個影像場補償回來的狀況。雖然判斷為ELA的方式補償，會致使輸出影像的PSNR會較低，但比較不會有影像不連續。

整體來說，Table Tennis的補償狀況，在動作補償在縮放畫面的背景無法有效的補償，但影像的背景區域Table Tennis並沒有很重要的細節，所以整體的顯示品質看起來是可接受的。

(5) Container

測試影像“Container”這一個影像序主要是靜態畫面居多，影像中的物體有緩緩移動有移動，唯一有差異的地方是影像後段有快速物體移動，如圖 56(a)所示。

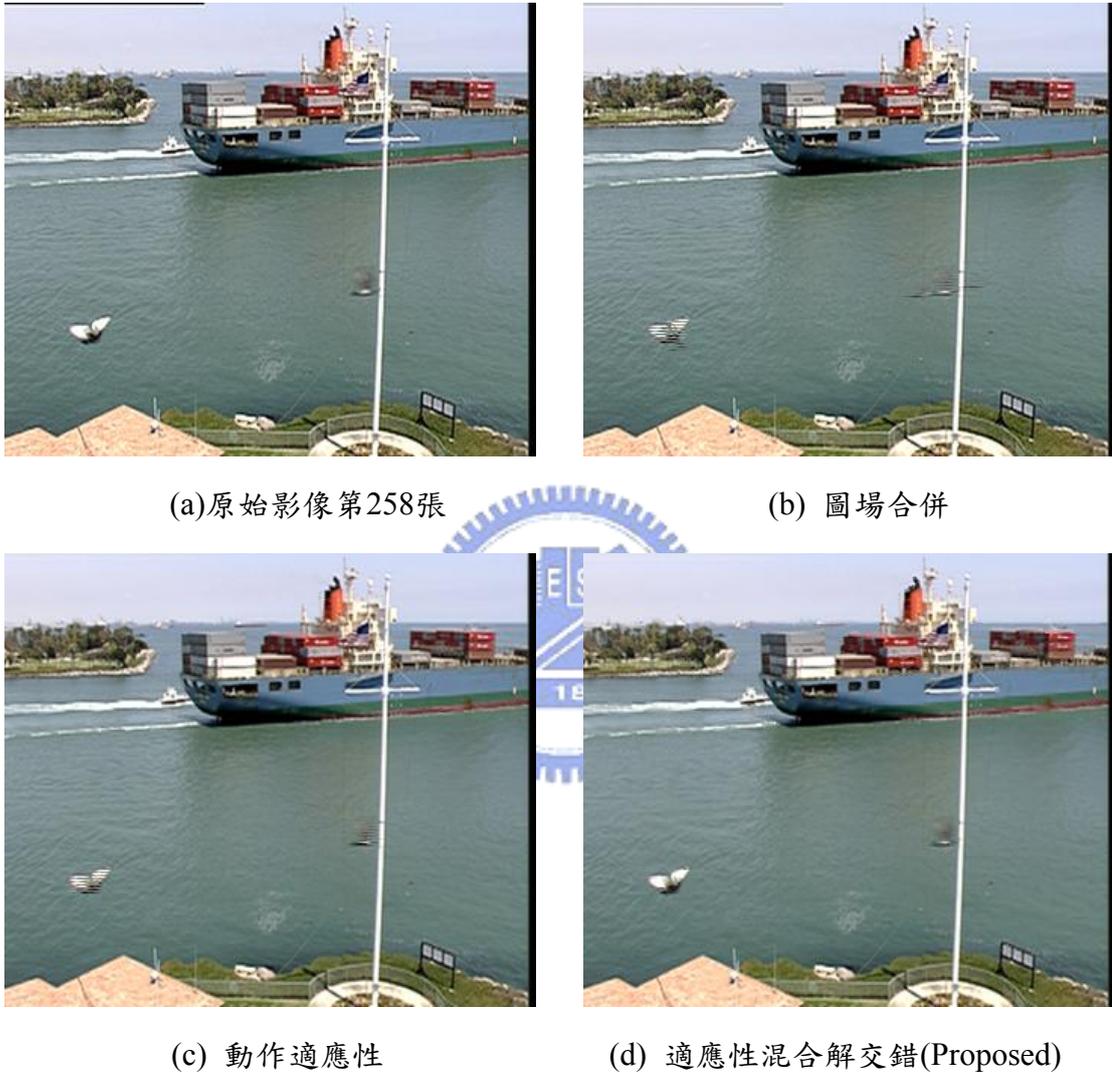


圖 58 影像“Container”輸出結果比較

由於物體瞬間的出現於畫面中，這部份不管以單向或雙向的動作補償都無法從上一個圖場中找到相似的，加上形態已經改變很多，如圖 58(a)中飛行的鳥，其前後的形態大不相同，且移動範圍大，所以要得到正確的補償，需切換到ELA就，降低錯誤的補償。圖 58(b)是圖場合併法，雖然它在背景有良好的補償，但缺點是快速移動的補償就會有

不連續的影像發生。而圖 58(c)，為使用動作適應性補償的方法，但決定影像的動作偵測的結果未達臨界值的話，其判斷也會有錯誤，也無法從錯誤的臨界判斷出影像該適用何種內插補償的方式。由於解交錯影像補償與其內容會有很大的相關，其適應性的動作補償，必須是依據內容自動做調整的判斷法則，以達到最好的影像補償，傳統的解交錯街方法雖皆有其優缺點，但選擇或判斷的補償方法也是相當的重要。

圖 58(d)為本文的輸出結果，該區域會判斷SAD，偵測SAD與鄰近的區塊的差異，再切換為ELA的方法來補償，以確認其輸出的結果是適當的，一般來說解交錯使用ELA空間性的內插，因為依照影像的內容而定，如果不是邊緣線，該區域的物體又是低頻的部份較多，也就是比較模糊的形態，這類的補償使用ELA等空間性內插，其效果都是可接受的。但如果移動的物體其影像是個很複雜的物體，或是高頻成份比較高，採用ELA反而會模糊該區域的影像，而動作估計又無法找從前場到找到這部份的影像，其補償的結果就會很差。但一般來說，移動快速的物體，通常影像都是模糊居多或低頻成份居多，無法維持很多細節在，所以在這種快速移動，形體轉變大的區域，有適應性切換ELA將會是比較可靠的作法。

以這個測試影像來說，使用動作估計補償會因為影像變化太大，移動向量有誤差而無法有效的找到相關區塊，無法從前場找到所需要的資料，這區域就必須轉換到ELA的內插，才不會因為前場資料沒有相關的而找不到相似的影像，由於本文的演算法中有混合MV與SAD的判斷條件，一旦影像區塊出現移動向量變化很大，且SAD值高於臨近區域的，該區塊很有可能是突然出現的影像，且一般的SAD比對結果通常是一個分布，有高有低，但差距不會很大，突然出現的移動物體區域將會破壞這個分布，使該區域的影像在做動作估計的時候，對應的區塊會有較大的SAD值變化程度，所以在移動向量不準時，其反應在相鄰SAD上的差異相對的也比較大，SAD差異會到如果大於周圍區塊SAD的分布群體的話，也就是中間區塊接近鄰近區塊的SAD分布的臨界，即可判定該區塊是突然的出現的物體，所以該區域的SAD很高，也就代表其變化很大，這裡的處理以切換補償方式為ELA，確保不會有不連續的影像，使輸出的品質正常。

(6) Silent



(a) 原始影像



(b) 圖場合併



(c) 動作補償(MC)



(d) 適應性混合解交錯(Proposed)

圖 59 影像“Silent”輸出結果比較

這個影片主要在於移動的手，由於主體與背景以靜態的畫面居多，直接以圖場合併會出現圖 59(b)的鋸齒狀。要把移動的手偵測出，一般的MC會在判斷MV時會出現錯誤，導致手的部份區域沒有辦法完全由MC去補償，如圖 59(c) MC無法找的很合適的區塊，所補回的方塊就會變成有線條狀出現。由於本文在移動區塊邊緣的條件下有判斷，其結果如圖 59(d)所示，由相鄰的SAD比較，差異太大的就不使用MC，改以ELA內插補點，所以在確定移動向量的區域為移動的物體，除了確認SAD外，會再由移動物體的邊緣區塊判斷與擴張移動物體的邊緣區域，這些擴張的區塊會以移動區塊相鄰的位置來決定是ELA補償，讓手勢移動相鄰的區塊影像都可適應性的切換補償方法。

(7) Flag

影像”Flag”是測試前景的移動物體的變化不規則時，其動作補償的偵測是否可以有效的補償。當移動的變形太大，動作估計無法從前面的影像，找到相同的補償區域時，決策部份就需要判斷成ELA的做法，做有效的邊緣內插，以圖 60(b)中，直接用時間性的內插都會產生嚴重的鋸齒狀，因為前後影像內容變化太大，只使用圖場間的內插法會產生這種問題，然而如圖 60(c)的結果，雖然ELA加上中間值濾波器可以解決部份這樣的問題，但其會不規則的選到上下點，但還是有不平順的邊緣或是邊緣線等粗糙的樣子產生。圖 60(d)為本文的結果，可以有較平滑邊緣線的補償。



(a) 原始影像



(b) 合併法(Merged)



(c) 邊緣線中值濾波



(d) 適應性混合解交錯(Proposed)

圖 60 影像“Flag”輸出結果比較

4.4. 計算複雜度統計表

本節分析影像的計算複雜度，相關研究中[32] [35]有比較其演算法的效益與計算複雜度。以影像的大小為 n 計算，一般的平均內插(Line Average)的次數即為 n ，所以複雜度為 $O(n)$ 。相對的以 3×3 windows ELA 是 $O(3n)$ ，因為存取記憶體次數要看 ELA 是取多少資料做計算的，為倍數級的，但整體看來還算是 $O(n)$ ，量級與平均內插 $O(n)$ 相同。而一般動作估計(ME)的方法要處理影像的大小是 n ，乘上 ME 的預估範圍(p)平方的次數乘上的每次計算 MV 的所要算的 SAD 統計，其複雜度將是 $O(p^2 \times n^2)$ 。

約略計算本文演算法所需要的指令次數，包含計算全區搜尋(Full Search)的 ME 所需的指令，分別由宏塊大小為 MB ，在這裡搜尋範圍的定義為 $(p) \times (p) = SR$ ， p 為搜尋距離，影像大小是 n ，計算一個移動向量 MV 所需指令次數。再以一個圖場(Field)有 n 個像素(pixel)，計算其移動向量 MV 的次數表示如(4.3)

$$MV = \frac{n}{MB} \quad (4.3)$$

每得到一個 MV 要做 SR 次的 SAD 計算，與 SR 次的最小 SAD 差值的比較(其包含了 SR 次的分支比較指令，如式(4.4)。每計算一個 SAD 要做 MB 次的差值與差值總合，所以有 MB 次的加與 MB 次的減，得 2 次的 MB 算術指令(Arithmetic)，表示如(4.5)。所以 2 個圖場(field)，計算 ME 會有 $2 \times MB$ 次的資料讀取，和 MB 次的資料寫入，所以得到 3 次的 MB 資料存取(Data Access)，最後在乘上整個影像的宏塊的數目，也就是 MV 的數目，計算出 ME 所花的指令數，表示如(4.6)。

$$\text{Branch Instructions} = (2 \times SR) \times MV \quad (4.4)$$

$$\text{Arithmetic Instructions} = (2 \times MB) \times SR \times MV \quad (4.5)$$

$$\text{Data Access} = (3 \times MB) \times SR \times MV \quad (4.6)$$

以交錯式的 CIF 尺寸的影像來計算為其所需要的指令數， n 為圖場的像素，CIF 的條件下， n 為 352×144 所以是 $n = 50688$ ，在這裡所使用的 MB 的大小為 16×8 ，所以

$MB=128$ ，搜尋範圍是 16×16 所以 $SR=256$ 。再把所有的數值代入並加總，可得 ME 部份的指令數。動作補償(MC)的這部份主要是從 ME 所得到的 MV 指到位置所補償得到，包含了 n 次資料讀取， n 次資料寫入，MC 的 Data Access 為 $2 \times n$ 次的。

邊緣線內插(ELA)的部份包含了 n 次資料讀取 n 次資料寫入，使用 9 抽頭的邊緣線內插(9-taps ELA)，式(4.7)其所需要的資料讀取次數為 $2 \times 9 \times n$ 次的資料存取(Data Access)，差值計算需要 $(9+6) \times n$ 次的算術指令(Arithmetic) 關係式如(4.8)，差異比較所需要的排序指令為 $9! \times n + 6! \times n$ ，得到所需為 $66 \times n$ 次的分支指令，其關係式如(4.9) 確定邊緣線方向後進行內插需要平均計算，所以有 n 次的算術指令(Arithmetic)，最後將會有 n 次資料寫入。

$$\text{Data Access} = (2 \times 9 \times n) + n \quad (4.7)$$

$$\text{Branch Instructions} = (66) \times n \quad (4.8)$$

$$\text{Arithmetic Instructions} = (9+6) \times n \quad (4.9)$$

計算其邊緣線內插(ELA)排序所需分支指令與資料存取部份所需之指令數再與 ME 與 MC 部之運算做總和計算，經整理為表 6。

表 6 指令數計算表

| MIPS | Data Access | Arithmetic Logic | Branch Compare | Total Instruction Counts |
|-------|-------------|------------------|----------------|--------------------------|
| ME | 38.9 | 25.9 | 0.2 | 65 |
| MC | 0.1 | 0 | 0 | 0.1 |
| ELA | 0.9 | 0.7 | 3.3 | 5 |
| Total | 39.9 | 26.7 | 3.5 | 70.2 |

其決策部份與影像邊界處理之判斷部份，指令總數與動作估計部份相比較為微小。另外表 6 所計算的為單一張畫面所得之指令數，以 CIF 30fps，每秒 30 張畫面相乘計算，其總指令數約為 2.1 GIPS。

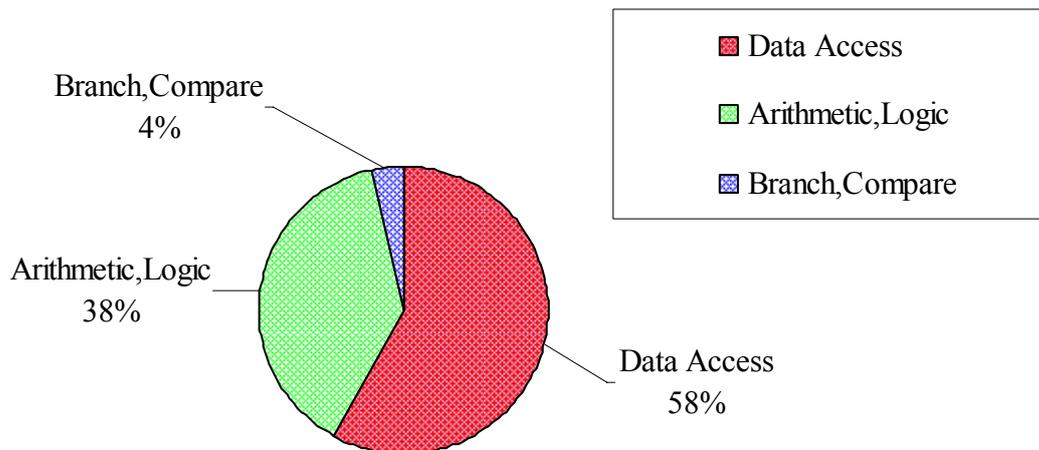


圖 61 演算法的指令數百分比

圖 61 中百分比分布圖中，有動作補償的解交錯系統中，其運算量較大的部份還是在動作估計的存取記憶體上，因為整體的計算量會在存取記憶體比較多，相對的多圖場的動作比對，其整體的運算量將會是倍增的，所以本文雙圖場動作估計，以先降低所要的記憶體與運算量，再對效能上做處理。另外算術指令的部份，由於本文演算法在動作估計的部份是 SAD 的計算，僅加法與減法居多，所以評估上並不會有乘法，因為乘法運算週期的長度會與加法有差。且整體計算動作向量所使用的參考畫面也只有一個，計算 SAD 的部份雖然是以全區搜尋，但這是為了得到可靠的動作向量及計算 SAD 所必須的。其他的決策分支的部份並不會佔到太多指令數，主要由決策輸出決定要使用那種技巧補償而已，最後輸出再混合原交錯影像，完成解交錯，與多圖場的動作估計解交錯指令數相比，可以有接近的品質，但其所花費的指令數可大幅降低 50% 以上。

第五章 結論與未來展望

5.1. 結論

根據第四章的輸出結果比較與效能分析，本文所提之方法為利用移動資訊判斷其可靠度，藉以變化補償方法，適應性的調整解交錯的補償方法，而與先前的方法相比和一般動作補償，其平均的 PSNR 可改進約 2db，其實際的顯示效果也有更進一步的提升。而多圖場解交錯則需要比較高的硬體代價，本文所需要的硬體代價可降低一半，其整體計算量也會比較精簡，整體效益亦比較高。

由本文所提出之雙圖場動作估計解交錯的演算法，可藉由移動資訊做動態區域與靜態區域的區分，再以適應性選擇其適合的補償方法予以補償，以增加影像的細節。而硬體上的代價則是降低記憶體的需求，使用前一個畫面做動作估計，可減少硬體的所需的代價，並以邊緣線內插當成是空間性內插，配合本文所提的決策法則，分辨出物體與背景的移動，並適應性的針對其移動物體做補償。比較於其他方法，除了降低硬體的代價外，本文所提的動作資訊決策，會在移動向量有誤差時，使用空間性的補償方法。可降低移動向量的錯誤補償，所以在動作估計後，加上決策部份的移動向量與比對差異值的組合，來區分動作種類，以決定合適的輸出，達到適應性的補償。

整體而言，本文所建構的動作補償，可有效的區分出所要決策的補償為何，並對影像的內容可以有適應性的補償，可改善動作資訊決策補償的輸出。

5.2. 未來展望

從本文解交錯的研究中，在雙圖場動作補償的架構基礎下，可藉由移動資訊的適應性判斷，提高補償的正確性。最後歸納兩部份的未來的工作與展望。

就品質而言，區塊比對的動作估計補償其效能會限制在區塊的大小，要再讓品質進一步的提昇，可加入更精確的可變區塊大小 (Variable Block Size) 的子區塊動作估計補償處理等，區分出更小區塊的移動向量，以及小區塊的移動向量的可靠度判斷的對策等，做更精確的補償，也都是可探討的研究方向。

以硬體而言，本文嘗試用雙圖場動作估計的解交錯，先降低所需要的記憶體數量，其計算量就可小於多圖場的動作估計，再提昇其準確度，來達到較好的顯示品質。所以在代價上，降低全區搜尋的計算量，又可維持移動向量的準確也是需要去探討的研究。或者是使用快速動作搜尋，快速動作搜尋雖然可降低計算量，但快速搜尋要克服的會是移動向量的準確性等問題。且對解交錯而言，不只有移動向量的準確度，補償影像後的視覺效果也是要考慮的。

就以架構上的區分，外解交錯的技術除了動作估計補償解交錯外，許多研究也有嘗試使用不同的架構，例如使用影像的著色修補(Inpainting)，或是更準確的邊緣線判斷內插等。而從解交錯的技術應用到相關動態影像處理上也是非常的重要，可應用這樣的動作判斷，重建遺失的影像，做有效的內插補點，並研究相關更多動態影像抽補點的動作補償等。

參考文獻

- [1]G. De Haan and E. B. Bellers, “Deinterlacing – An Overview,”in *Proc. of IEEE*, vol. 86, no. 9, pp.1839 – 1857, Sep. 1998.
- [2]G. De Haan and E.B Bellers, *De-interlacing – A Key Technology for Scan Rate Conversion*, Elsevier, 2000.
- [3]M. Zhao and G. de Haan, “Content adaptive vertical temporal filtering for de-interlacing,” in *Proc. ICSE05*, pp. 69 - 73,Jun. 2005.
- [4]S.-K. Kwon, K.-S. Seo, J.-K. Kim, and Y.-G. Kim, “A motion-adaptive de-interlacing method,” *IEEE Transactions Consumer Electronics*, vol. 38, no. 3, pp.145–150, Aug. 1992.
- [5]K. Sugiyama and H. Nakamura, “A method of de-interlacing with motion compensated interpolation,” *IEEE Transactions Consumer Electronics*, vol. 45, no. 3, pp. 611–616, August 1999.
- [6]Keith Jack, *Video Demystified, 3 ed.*, HARRIS, 2001.
- [7]C. J. Kuo, C. Liao, and C. C. Lin, “Adaptive interpolation technique for scanning rate conversion,” *IEEE Transactions Circuits and Systems for Video Technology*, vol. 6, no. 3, pp. 317–321, Jun. 1996.
- [8]J. Salo, Y. Nuevo and V. Hameenaho, “Improving TV picture quality with linear-median type operations,” *IEEE Transactions Consumer Electronics.*, vol. 34, no. 3, pp. 373–379, Aug.1988.
- [9]C. Hentschel “Comparison between median filtering and vertical edge controlled interpolation for flicker reduction,” *IEEE Transactions Consumer Electronics.*, vol. 35, pp. 279–289, Aug. 1989.

- [10]M.K. Park, Moon.G. Kang, K. Nam and S.G. Oh, “New Edge Dependent Deinterlacing Algorithm Based on Horizontal Edge Pattern,”*IEEE Transactions Consumer Electronics*, vol. 49, no. 4, Nov 2003.
- [11]M.H. Lee, J.H Kim, J.S. Lee, K.K. Ryu and D. Song, “A new algorithm for interlaced to progressive scan conversion based on directional correlations and its IC design,” *IEEE Transactions Consumer Electronics.*, vol. 40, pp. 119–129, May 1994.
- [12]F. Michaud, C. T. Le Dinh, and G. Lachiver, “Fuzzy Detection of Edge-Direction for Video Line Doubling,”*IEEE Transactions Circuits and Systems for Video Technology*, vol. 7, no. 3, pp. 539-542, June 1997.
- [13]H.S. Oh, Y. Kim, Y.Y. Jung, et al , “Spatio-temporal edge-based median filtering for deinterlacing,”*IEEE International Conference on Consumer Electronics*, pp.52–53, 2000.
- [14]T. Koivunen, “Motion Detection of an Interlaced Video Signal,”*IEEE Transactions Consumer Electronics*, vol. 40, no 3, pp.753-760, August 1994.
- [15]C. Hentschel, “High quality noise insensitive motion detector using one field memory,”*IEEE Transactions Consumer Electronics.*, vol. 42, no. 3, pp.696–704, Aug. 1996.
- [16]F. Dufaux and F. Moscheni, “Motion estimation techniques for digital TV: A review and a new contribution,” *Proc. IEEE*, vol. 83, pp. 858–876, June 1995.
- [17]F. M. Wang, D. Anastassiou, and A. M. Netravali, “Time-recursive deinterlacing for IDTV and pyramid coding,” *Image Commun. J.*, vol. 2, pp. 365–374, Oct. 1990.
- [18]A. J. Patti, M. I. Sezan, and A. M. Tekalp, “Robust methods for highquality stills from interlaced video in the presence of dominant motion,”*IEEE Transactions Circuits and Systems for Video Technology*, vol. 7, no. 2, pp. 328–342, Apr. 1997.
- [19]O. Kwon, K. Sohn, and C. Lee, “Deinterlacing using Directional Interpolation and Motion Compensation,”*IEEE Transactions Consumer Electronics*, vol. 49, no. 1, pp. 198-203, February 2003.

- [20]Y.Y. Jung, B.T. Choi, Y.J. Park, S.J. Ko,“An Effective De-interlacing Technique Using Motion Compensated Interpolation,” *IEEE Transactions Consumer Electronics*, vol. 46, no. 3, pp. 460 –466, Aug. 2000.
- [21]YL Chang, SF Lin, CY Chen, LG Chen, “Video De-Interlacing by Adaptive 4-Field Global/Local Motion Compensated Approach,” *IEEE Transactions Circuits and Systems for Video Technology*, vol. 15, no. 12, pp. 1569-1582,Dec. 2005.
- [22]D Wang, A Vincent, P Blanchfield, “Hybrid de-interlacing algorithm based on motion vector reliability,”*IEEE Transactions Circuits and Systems for Video Technology*, vol.15, no.8, pp. 1019-1025, Aug. 2005.
- [23]HM Mohammadi, P Langlois, Y Savaria, “A Five-Field Motion Compensated De-interlacing Method Based on Vertical Motion,”*IEEE Transactions Consumer Electronics*, vol.53, no.3, pp.1117-1124, Aug. 2007.
- [24]S. Byun, J. Byun, G. Kim, "A recursive approach for de-interlacing using improved ELA and motion compensation based on bi-directional BMA," *International Conference on Image Processing* , Vol. 3., pp. 1679-1682,Oct. 2004.
- [25]J. Kovacevie, R. J. Safrank, and E. M. Yeh, “Deinterlacing by successive approximation,” *IEEE Transactions Image Processing*, vol. 6, pp. 339–344, Feb. 1997.
- [26]D. Han, C.Y. Shin, S.J. Choi, J.S. Park,"A Motion Adaptive 3-D De-interlacing Algorithm Based on The Brightness Profile Pattern Difference,"*IEEE Transactions Consumer Electronics*, vol. 45 no. 3 , pp 690 -697, Aug. 1999.
- [27]T.S. Chong, O.C. Au, W.S. Chau, T.W. Chan, "A content adaptive de-interlacing algorithm," *IEEE International Symposium on Circuits and Systems*, no. Vol. 5, pp. 4923-4926,May. 2005.
- [28]G. De Haan, “De-interlacing of video data,” *IEEE Transactions Consumer Electronics*, vol. 43, no. 3, pp. 819–825, Aug. 1997.

- [29]X. Gao, J. Gu, J. Li, "De-interlacing algorithms based on motion compensation," *IEEE Transactions Consumer Electronics*, vol.51, no.2, pp. 589-599, May 2005
- [30]M. Biswas and T. Q. Nguyen, "A novel de-interlacing technique based on phase plane correlation motion estimation," in Proc. *IEEE International Symposium on Circuits and Systems*, pp. (11)604-607, May 2003.
- [31]Q. Huang, W. Gao, D. Zhao, and H. Sun, "An efficient and robust adaptive de-interlacing technique," *IEEE Transactions Consumer Electronics*, vol. 52, no. 3, pp. 888-895, Aug. 2006.
- [32]S.F Lin, Y.L. Chang, and L. G. Chen, "Motion adaptive de-interlacing by horizontal motion detection and enhanced ELA processing," in Proc. *IEEE International Symposium on Circuits and Systems*, vol. 49, no. 4, pp.1256-1265, Nov. 2003.
- [33]K. Jensen and D. Anastassiou, "Sub-pixel edge localization and the interpolation of still images," *IEEE Transactions Image Processing*, vol. 4, pp.285-295, Mar. 1995.
- [34]G. de Haan and R. Lodder, "De-interlacing of Video Data Using Motion Vectors and Edge Information," *IEEE Transactions Consumer Electronics*, pp.70-71, Jun. 2002
- [35]P.M. Kuhn, *Algorithms, complexity analysis and VLSI architectures for MPEG-4 motion estimation*, Kluwer Academic, Boston, 1999.

自傳

姓名：李裕鴻，

籍貫：台灣彰化縣

學歷：90年畢業於台灣國立聯合大學電子工程系

兵役：92年役畢

經歷：92~98年服務於天鈺科技，負責TFT驅動電路之數位電路設計研發

專長：數位IC設計，TFT驅動電路之數位電路設計，數位影像處理與電路設計，影像處理之系統驗證與FPGA應用。

