第一章 緒論

1.1 研究動機

現今電視訊號格式，以交錯式掃描 (interlaced scan) 作為電視畫面的視訊傳輸與儲存的主要格式，這種播放格式會產生不好的視覺效果，而且會產生畫面 1.抖動、2.閃爍、3.爪痕現象，這些現象都是由於交錯掃描所造成地。隨著視訊系統的技術快速發展，解交錯式掃描(deinterlacing scan)被提出，它將交錯掃描的格式換成循序掃描(Progressive Scan)的格式，它可以解決交錯式掃描所產生的缺點，並且逐漸廣泛的被採用. 新的影像顯示設備，如 HDTV (High-Definition Television)高畫質電視、液晶顯示器 (Liquid Crystal Display, LCD)、電漿顯示器 (Plasma Display Panel, PDP)與個人電腦顯示器等由本身顯示技術的特性，需利用循序掃瞄的式輸入畫面，因此解交錯式(deinterlacing)的技術已成為主要研究主題了。

由於未來電視傳播系統將會由循序掃描(progressive scan)的格式取代傳統交錯式掃描 (interlaced scan)，而解交錯掃瞄的越來越普及，本論文的動機在於提出一個新的解交錯掃描演算法，解決交錯式掃描的缺點,並且能運用於新的電視傳播系統中。

1.2 研究目標

現今解交錯技術研究發展針對水平運動方面去考慮，忽略垂直運動偵測補償，導致影像品質結果改善有限。在現實環境中，不管是任何物件在動作方面都會結合水平方向的運動和垂直方向的運動，以達成一個完整的動作表現。但解交錯演算法中卻沒有提出針對垂直方向的運動去做偵測插補，只利用了水平方向的運動向量，建構出一張完整的畫面(frame)。這樣的解交錯演算法是不夠健全完善的，不能妥善的處理各種情況下解交錯影像，導致影像品質無法精確的被呈現出來。而本論文的目地在於提出一個完整的移動適應性技術的系統，其中包含了空間資訊技術的演算法和時間資訊技術的演算法。在時間技術的演算法我們使用中位數 EDI 演算法，能有效的預測插補。而時間資訊技術的演算法我們使用了四個場的水平運動偵測與四個場的垂直運動偵測，能有善改影像品質。另外一個影響影像品質缺失的因素就是場景變化。而場景變化的鏡頭經常在電視影片播放過程中發生。但使用的解交錯演算法並沒有針對場景變化的畫面作特別的技術處理，容易發生鋸齒現象、模糊現象和疊影現象。場景變化的情況在我們提出的解交錯演算法中被考慮進去，為了減少不必要的錯誤資訊被使用。

1.3 論文大綱

本論文所研究的重點於運動偵測必須包含了水平跟垂直，不能只著重其中一項，還有提出在空間解交錯的演算法中還必須考慮場景變化之存在，稍後在本文中會詳細的介紹。本文中會介紹現有文獻所提出解交錯演算法的研究方法及成果,並比較其優缺點，最後將我們提出的方法與其他演算法做比較，其中會展現出我們研究結果。本論文安排如下：

第二章主要介紹解交錯的相關技術，包括如下：1.使用空間資訊的處理技術(intrafield deinterlacing)所組成；2.考慮時間資訊的處理技術(interfield deinterlacing)；3.使用移動適應性的技術(motion adaptive deinterlacing)；4.則為移動補償(motion compensated deinterlacing)的方法。

第三章主要介紹我們所提出水平與垂直運動適應性解交錯演算法，其中包含了水平運動偵測補償、垂直運動偵測補償以及空間解交錯補償的方法。章節最後是我們系統實作的方法及結果，可以發現我們提出的演算法可以有效的解決畫面會產生抖動、閃爍、爪痕現象等問題，而 PSNR 也高於其他演算法。

第四章針對水平與垂直運動適應性演算法提出需要改進的地方，並且提出新的演算法解決了水平與垂直運動適應性演算法需要改

進的地方，而演算法本身中也包含了場景變化偵測的技術，文中也會敘述場景變化偵測的重要性。在 4.1 節中會探討一些解交錯的相關技術；4.2 節裡介紹我們提出的場景變化偵測解交錯演算法，包含了移動適應性解交錯技術與場景變化偵測演算法。演算法有效的降低場景變化的畫面所導致的影響，提升影像品質；4.3 節是實驗結果，可以發現我們提出的演算法可以有效的改進水平與垂直運動適應性演算法的不足，而效果和影像信號雜訊比(Peak Signal to Noise Ratios, PSNR)也高於其他演算法。

第五章根據實驗的結果做一個討論，並且提出論文研究的結論。

第二章 相關研究

解交錯演算法將原本垂直解析度只有一半資訊畫面,提升到每個畫面都有全部垂直解析度。舉例來說，在 NTSC 標準中，可將畫面視為水平 720 畫點乘以垂直 480 條掃描線,每次畫面更新只有變動半個畫面，只有奇數場與偶數場交替更新，因此真正更新垂直 240 條掃描線，而解交錯演算法就是要讓每張畫面更新都是垂直 480 條掃描線。解交錯演算法的結構大致上可以區分為四類：1.使用空間資訊的處理技術(intrafield deinterlacing)所組成；2.考慮時間資訊的處理技術(interfield deinterlacing)；3.使用移動適應性的技術(motion adaptive deinterlacing)；4.則為移動補償(motion compensated deinterlacing)的方法。

2.1 空間的解交錯演算法(intrafield deinterlacing)

空間的解交錯演算法(intrafield deinterlacing)[2]-[7]屬於比較簡單的一種，只需單一個場(field)就能建構出一張完整的畫面(frame)，而且計算量的需求通常是很低的。軟體或硬體實所耗費的成本並不高，所以常被拿來使用，如 Bilinear 和 line doubling ，其中 Bilinear 的作法是將遺失點的座標位置上下兩點的值，相加除以二，所求得的值為插補的值。 line doubling 的作法更為簡單，將遺失點的座標位置上下

兩條平行線，選擇上下其中一條然後將相對位置的像素值插補上去，這些技術對於低頻圖像(low frequency image)來說效果不錯。不過，使用在高頻的圖像(high frequency image)裡會造成邊緣的鋸齒狀(jagged effect in the oblique edge)或使影像模糊化(blurring effects)。 Lee 等學者[2]提出 ELA(edge based line average)的演算法來解決 Bilinear 和 line doubling 所造成的問題。 ELA 是依據邊緣的方向差異小作為插補的準則，當邊緣可能被正確地估計時，這種方法提供不錯的效果。相對的，錯誤的邊緣資訊被使用時，會使邊緣破裂不完整，產生不連續的邊緣。Lee 等專家[7]改良傳統的 ELA 演算法，將原本 3x3 比較區塊擴張為 5x3 區塊,增加可判別的角度並且設定門檻值，更準確的判斷插補值。

2.2 時間解交錯技術演算法(interfield deinterlacing)

時間解交錯技術演算法(interfield deinterlacing)是透過直接合併(merge)連續兩個場(field)產生一個完整的畫面(frame)，一般如果是合併的區域是屬於靜止狀況下(static area)，則會得到不錯的效果,相對的合併的區域是屬於運動狀況下(motion area)，則會產生抓痕現象(line crawling effect)。Sun[8]提出一種以最短路徑運動資訊的演算法去重新排列場(field)，得到完整的畫面，這種方法相當的快速，運算量也很

少。

2.3 移動適應性演算法 (motion adaptive deinterlacing)

移動適應性的技術(motion adaptive deinterlacing)[5]，[8]-[16]是合併了空間解交錯技術 (intrafield deinterlacing)和時間解交錯技術(interfield deinterlacing)的優點。Oh 等人[5]提出 2D-ELA 演算法，將原本針對單張場(field)作插補的策略擴充到使用現在要插補的場和它的前後張場，利用三個連續場的資訊來進行插補的預測判斷。在 Han 等人[10]的中針對遺失點的插補方法會經過一連串的偵測和過濾器，如移動偵測、最大值偵測、中位數過濾器、移動擴展器(motion expander)，最後判斷要用空間解交錯或是時間解交錯的演算法做插補，空間解交錯適用 Sobel operator 方法，時間解交錯式採用四個連續場的方法。Lin 等學者[11]提出在空間解交錯方法是採取中位數 ELA，在求解的過程中會得到一組值，而且必須小於門檻值(threshold)，如果沒有小於門檻值(threshold)的話，對所求的值和遺失點座標位置上下兩點作中位數，即可求得插補值。在時間解交錯的方法是使用四個場來偵測移動的相對位置，運用四個場的相關性去找出遺失點可能移動的位置，找到可能移動的位置後，將該點的像素值插補到遺失點上。Lee 等人[12]提出以 edge-based median filter(EMF)和

adaptive minimum pixel difference filter(AMPDF)，為架構的移動適應性解交錯演算法，演算法當中把一張畫面(frame)分成移動區域、背景和邊緣區域三種。使用不同的門檻值來判別出這三種區域，然後根據不同的區域使用不同的方法去找出最佳的值。Chen 等人[13]將畫面區分為背景跟前景，不會移動的物件稱之背景，相對的稱之前景。如果是背景的話可以直接參考前一張的畫面相同位置來拿來作插補,這樣可以省下很大的計算量，如果不是背景的話在以時間資訊的處理技術演算法作插補。在 Kim 等學者[16]中提到使用運動決定回饋參數(motion decision feedback parameter)判斷遺失點插補的時候，是要使用移動適應性解交錯的演算法插補或是使用空間演算法來插補。

2.4 移動補償之解交錯演算法

移動補償之解交錯演算法是利用物體移動軌跡的方向作插補[17]-[21]。藉由物體在時間上的變化適當地找出物體在場與場之間的移動向量，在連續影像中有相當大的相依關係.現有文獻中提出運用移動補償的方法，都以區塊為基礎的比對為基礎，在兩張連續的基數場或是偶數場裡，找出最相近的區塊，然後計算出運動向量，根據運動向量去建構一個新的場，把原本要插補的場和新建構出來的場合

併，即產生一個完整的畫面。Sugiyama 及 Nakamura[17]中提到移動補償的解交錯技術中有三個問題產：1.不同大小的區塊在尋找比對時，會產生不同的效果。2.在移動補償解交錯演算法中，解交錯技術最容易產生插補錯誤遺失點像素值的問題。3.移動向量以區塊作插補容易產生區塊效應導致不連續性的問題。Sugiyama 提出使用 4x4 的區塊來當作移動補償的區塊大小，然後設定收尋範圍，減少不必要的比對，解決了第一個問題。利用插補係數(intra factor)來判斷是要採取空間插補演算法還是時間插補演算法，解決了第二個問題。最後提出利用兩個方向作用的移動補償(bidirectional motion compensation)來解決區塊效應的問題。Li 等人[18]提出相位校正過濾器(phasecorrection filter)演算法，當一個場作移動偵測和移動補償之前要先經過相位校正過濾器的處理，這樣能有效的提升移動偵測的準確率，使在作移動補償演算法時能得到較好的品質，提升影像信號雜訊比(Peak Signal to Noise Ratios, PSNR)。Jung 等人[19]的方法裡，先把前一張場中所有資訊像素重新排列變成垂直高度只有原來的一半，水平寬度沒有變，同樣後一張場也作相同處理，然後經由新產生的前後張場去求得移動向量，把移動向量除以二之後作移動補償，可以產生半張現在場,剛好是整張現在場中所遺失的資訊,將兩張合併及產生一張完整的現在場，在現在場中的移失點所插補值會有錯誤，Jung 另外提出了中位數

過濾器解決這個題。Kown 等學者[20]所提的方法，在空間解交錯技技術是修改了 ELA 演算法，ELA 原本只能判斷 45 度 、0 度、-45 度，把它擴充到可以判別 45 度、26 度、0 度、-26 度、-45 ，在時間解交錯技術是使用前後張場作移動估計，得到移動向量，然後對現在場作移動補償。在現在場中的每個遺失點的像素值設定一個門檻值，作為選擇空間技術演算法產生的插補值或是以時間技術演算法產的插補值判別條件。Chang 等人[21]的作法，利用 4 個場作移動補償，以區塊的比對作移動預測(motion estimation)，在前前張的場(forward forward field)跟現在場(current field)作區塊比對可以得到移動向量(motion vector)，前張的場(forward field)跟後一張的場(backward field)作區塊比對一樣可以得到移動向量(motion vector)，經由兩個移動向量的條件判斷可以得到最佳的移動向量(motion vecter)。最後插補出一張較精確完整的畫面。不過，實現環境中移動補償演算法需要較高的複雜度，沒有可靠的運動向量估計很難得到良好的效果。

上述四種的解交錯演算法各有優缺點，由整體架構評估出來以移動適應性的演算法(motion adaptive deinterlacing)優於其他演算法架構，因此以移動適應性的演算法(motion adaptive deinterlacing)成為現今解交錯技術研究發展的重要主題方向。解交錯的演算法雖能有效的改善影像品質，只針對水平運動方面去考慮，而忽略垂直運動偵測補

償，導致影像品質結果改善有限。在現實環境中，不管是任何物件在動作方面都會結合水平方向運動和垂直方向運動，以達成一個完整的動作表現。但解交錯演算法中卻沒有針對垂直方向的運動去作偵測插補，只利用了水平方向的運動向量，建構出一張完整的畫面(frame)。這樣的解交錯演算法是不夠健全完善的，不能妥善的處理各種情況下的解交錯影像，導致影像品質無法精確的被呈現出來。而本論文中提出一個完整的移動適應性技術系統，其中包含了空間資訊技術演算法和時間資訊技術演算法。在空間技術演算法我們使用中位數 EDI 演算法，能有效的預測插補。而空間資訊技術演算法我們使用了四個場的水平運動偵測與四個場的垂直運動偵測，能有效善改影像品質。

第三章 水平與垂直運動偵測解交錯演算法

3.1 解交錯演算法

本論文提出不只針對水平運動方向的偵測插補外，還要對垂直運動方向作偵測插補。改善畫面抖動、閃爍、爪痕現象，提升影像的品質。除外，在空間的解交錯技術演算法我們使用中位數的 EDI，降低誤判發生的可能性與消除鋸齒的現象，增加影像品質效果。本論文所提之系統架構包含(1)水平運動方向的偵測插補；(2)垂直運動方向的偵測插補；(3)中位數 EDI 空間解交錯演算法。圖 3.1 是我們的流程架構圖，我們的演算法會先從水平運動偵測開始偵測是否有水平運動，如果沒有符合門檻值的話，進行第二步驟垂直運動偵測，當一、二步驟判斷條件都不成立時，則到最後一個步驟是中位數 EDI 演算法。

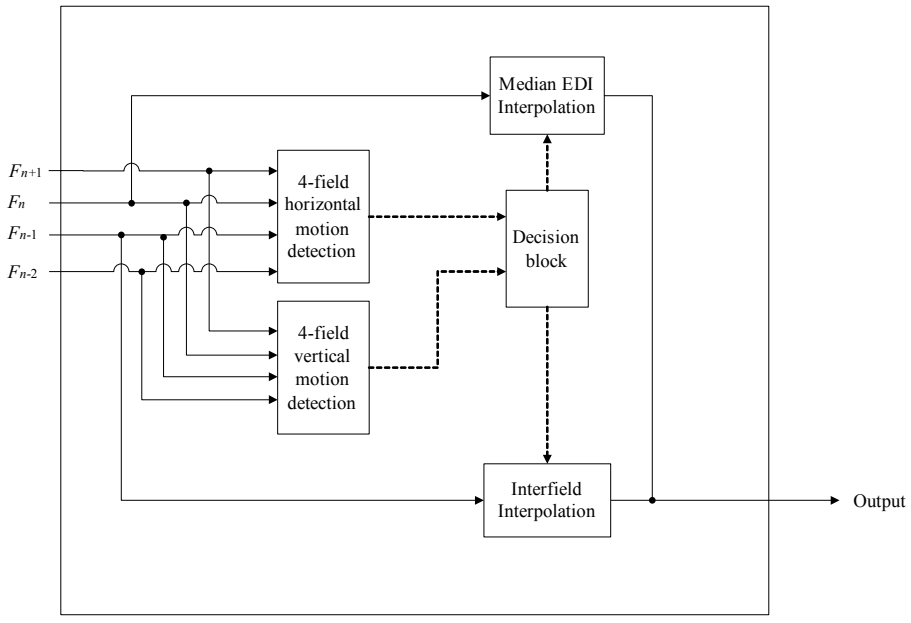

圖 3.1 流程架構圖

3.1.1 運動方向的偵測插補

在說明水平運動跟垂直運動之前，我們將 $F_{n-2}(i,j)$、$F_{n-1}(i,j)$、$F_n(i,j)$和 $F_{n+1}(i,j)$定義為前前張的場、前一張的場、現在場和後一張的場。水平運動方向的偵測演算法中，有演算法提出使用兩個場來偵測水平運動，使用 $F_{n-1}(i,j)$和 $F_{n+1}(i,j)$來判斷 $F_n(i,j)$裡面遺失點該插補的值，但是使用兩個場會產生一些問題，如物件在 $F_{n-1}(i,j)$和 $F_{n+1}(i,j)$之間快速的移動，剛好在 $F_{n-1}(i,j)$和 $F_{n+1}(i,j)$的相同位置被判斷為靜止狀態，但 $F_n(i,j)$裡面應該是運動狀態，因此這時候被誤判為靜止狀態,會產生抓痕效應。在我們提出的水平運動偵測時，我們採取 4 個連續場的演算法作為水平運動偵測，能有效的解決爪痕效應，正確的偵測水平方向的運動。

圖 3.2 是我們水平運動偵測演算法的示意圖，演算法中利用 $F_{n-1}(i,j)$跟 $F_{n+1}(i,j)$採取一個 1x3 區塊比對，可以得到一組誤差最小絕對值，如果這組值小於門檻值 T_1 的話，而且 $F_n(i,j)$跟 $F_{n-2}(i,j)$比對採取 1x1 的區塊，一樣得到一組誤差最小絕對值小於門檻值 T_2 的話，且兩組值的方向也一致的話，就可以進行插補的動作。

使用 4 個場的水平運動偵測能有效的減少錯誤插補的情況發生，並且能正確的偵測出正確的水平運動，產生高品畫質的影像減少了爪痕。尤其針對連續性動作的影片更能表現出水平運動方向偵測插

補的優點。

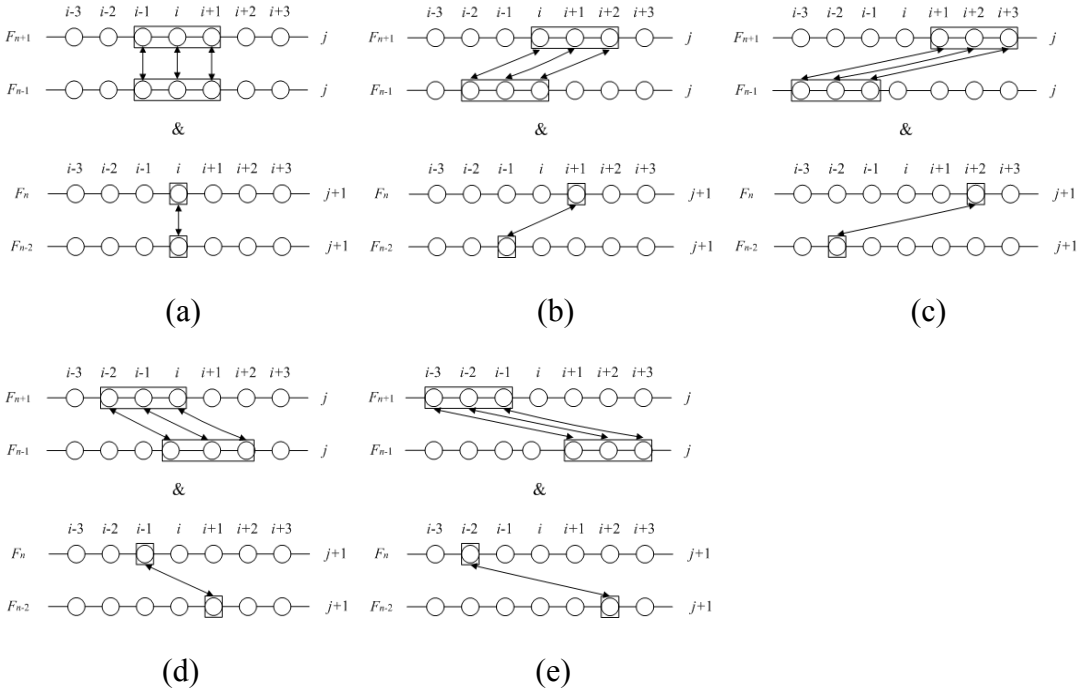

圖 3.2 水平運動方向偵測插補示意圖

物件的運動不只有水平運動，也包含了垂直運動，所以這裡把水平運動跟垂直運動區分出來，我們已經探討了水平運動偵測插補的演算法了，現在要探討垂直運動偵測插補演算法。在垂直運動方向的偵測區分成 1).向上垂直運動偵測；2).向下垂直運動偵測；3).斜方向運動偵測，如果沒有區分出來會導致影像插補時錯亂，造成影像品質下降。垂直運動偵測有效的改善插補點預測錯誤，減少雜訊的產生。

向上垂直及向上斜方向運動偵測採取 4 個連續場的演算法，針對五個方向30°、45°、90°、135°、150°做垂直偵測，圖 3.3 是我們垂直向上運動偵測演算法的示意圖，演算法中利用 $F_{n-1}(i,j)$跟 $F_{n+1}(i,j)$採取

一個 1x3 區塊比對，可以得到一組誤差最小絕對值，如果這組值小於門檻值 T_1 的話，而且 $F_n(i,j)$跟 $F_{n-2}(i,j)$比對採取 1x1 的區塊，一樣得到一組誤差最小絕對值小於門檻值 T_2 的話，且兩組值的方向也一致的話，就可以進行插補的動作。

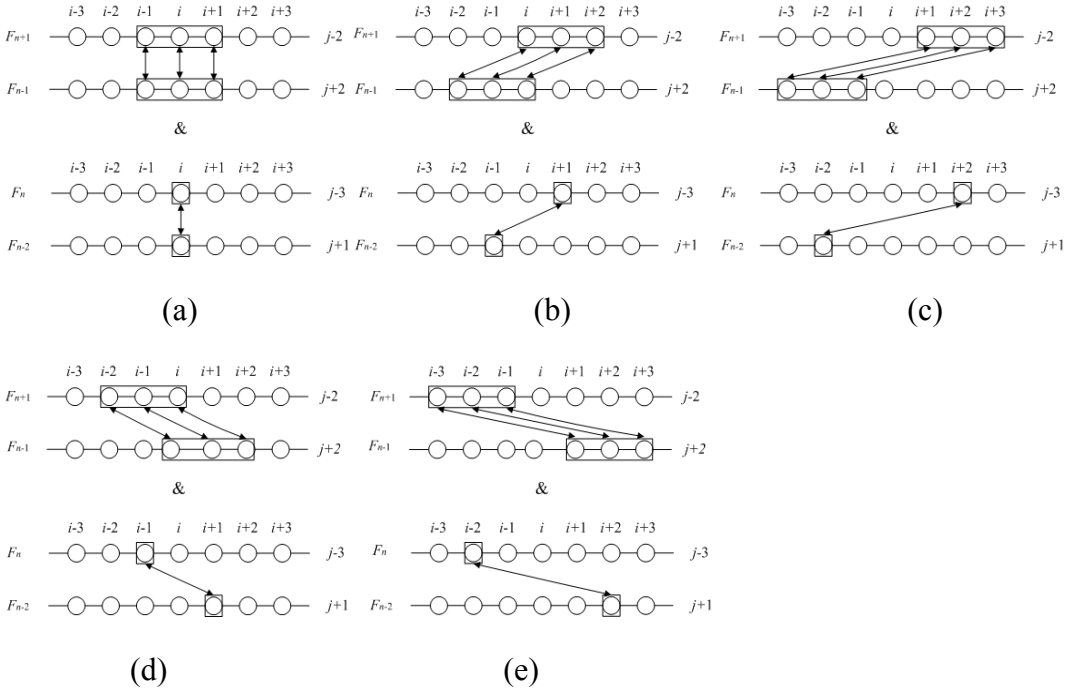

圖 3.3 向上垂直與向上斜方向運動偵測插補示意圖

向下垂直及向下斜方向運動偵測作法和概念跟向上垂直及向上斜方向運動偵測一樣，差別在於區塊比對時要參考的行列數不一樣。

有關於門檻值的設定，門檻值 T_1 介於 30 到 60 之間,而門檻值 T_2 介於 10 到 20 之間的範圍，經過上述的流程我們可以得到一張完整品質不錯的畫面(frame)，解決了物件運動所造成畫面的抖動、閃爍、爪痕現象，降低雜訊的產生，尤其針對連續性動作的影片更能表現出我

們所提出來演算法的優點。

3.1.2 中位數 EDI 空間解交錯演算法

EDI 的空間的解交錯演算法[3]而且效果優一般的 ELA 演算法，可以得到一張高可靠度的畫面(frame)。但有時候會因為邊緣區域(edge region)錯誤的預測插補，導致邊緣破碎不完整鋸齒現象產生，為解決誤判邊緣的方向及有效的消除鋸齒，我們提出一個中位數 EDI 演算法，改善了邊緣區域破碎不完整問題，進而提昇影像品質，圖 3.4 是我們的 EDI 示意圖。