

國立交通大學

電子工程學系 電子研究所碩士班

碩士論文

視覺監控資料錄影與重播之研究

A Study on Recording and Replaying

Video Surveillance Data

研究生：黃昭穎

指導教授：王聖智 博士

中華民國 九十四年 十月

視覺監控資料錄影與重播之研究

A Study on Recording and Replaying Video Surveillance Data

研究生：黃昭穎

Student：Zhao-Ying Huang

指導教授：王聖智 博士

Advisor：Dr. Sheng-Jyh Wang

國立交通大學

電子工程學系 電子研究所碩士班



A Thesis

Submitted to Institute of Electronics

College of Electrical Engineering and Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master of Science

in

Electronics Engineering

October 2005

Hsinchu, Taiwan, Republic of China.

視覺監控資料錄影與重播之研究

研究生：黃昭穎

指導教授：王聖智 博士

國立交通大學

電子工程學系 電子研究所碩士班

摘要



在本篇論文裡，我們提出了一個完整的即時監控系統的架構，除了包含了錄影、即時播放及重播等基本功能外，更針對壓縮儲存的方式提出了更進一步的探討。另一方面，由於攝影機的掃描方式為交錯式(Interlace)訊號，因此我們提出了一個去交錯(de-interlace)的方法，也針對了影像的縮放進行探討，希望能在重播時得到較佳的影像品質。

在壓縮儲存方面，我們採用軟體的壓縮方式，採取以 MPEG4 simple profile 為主體的壓縮方式。為了能使壓縮率更加提高，我們針對訊號與雜訊進行分析，在完全沒有訊號的情形下，採取 Frame Skip 的機制；而當有訊號發生的時候，我們設計出 Block-Change 的方式，根據訊號佔據畫面的大小來調整錄製的畫面；最後，當畫面進入長時間沒有訊號的情形下，我們會將調大 MPEG4 的 QP 值，讓壓縮率進一步的提高。在去交錯的部分，我們提出的方式主要的架構是以 MA-IPC (Motion adaptive interlace-to-progress conversion)為主體，在空間域部分我們提出了一個新的方法，以分群為概念，綜合了 LA(Line Average)以及 Median 的方式，進一步的提高去交錯後的影像品質。最後在影像縮放部分，我們根據影像內容，在平滑區與邊界區做不同的處理，綜合了 NN (Nearest Neighborhood)、Bilinear 以及 Win-scale 的方法，並且額外提出了一個 model-based NN 的概念，以達到影像縮放時較好的影像畫面品質。

A Study on Recording and Replaying Video Surveillance Data

Student : Zhao-Ying Huang

Advisor : Dr. Sheng-Jyh Wang

Department of Electronics Engineering, Institute of Electronics

National Chiao Tung University

Abstract

In this thesis, a real-time surveillance system is proposed. Besides real-time monitoring, recording and replaying, we also focus on the enhancement of compression rate. Additionally, because the capture frames are in the interlaced form, we propose a new de-interlacing method. Moreover, we also propose a new image scaling method to improve the quality of scaled images.

In compression and recording, we adopt a software solution that is based on MPEG4 simple profile. To further enhance the compression rate, three additional methods are adopted. If there is no change in the monitored scene, we perform frame-skip. If there are changes in the scene, we will detect the corresponding parts of the images and save only the information of those parts. Moreover, if the video has been under no change for a while, we'll also change the value of QP (Quantization Parameter) to further enhance the compression rate.

In replaying, a new de-interlacing method is proposed which is based on MA-IPC (Motion adaptive interlace-to-progress conversion). In the spatial domain, we propose a new method based on grouping. Based on grouping, the proposed method combines the advantages of LA (Line Average) method and the Median method. Moreover, a content-based image scaling method is also proposed which is a combination of NN (Nearest Neighborhood), Bilinear and Win-scale. We also propose a new idea named model-based NN to improve the quality of scaled images.

誌謝

感謝我的指導教授
王聖智老師
他是個好老師、好教授。

感謝實驗室的學長姊
已經畢業的和快要畢業的
給予我研究上的指導

感謝同學
同個實驗室和其他實驗室的
常常給我鼓勵

感謝實驗室的學弟妹
認真研究的態度深深感動了我

感謝家人
忙得沒時間照顧他們
他們還是不時傳來溫暖

感謝實驗室整體的氣氛
也感謝實驗室常常會漏水的冷氣
以及不知道幾屆以前留下來的彈簧床
有了你們實驗室就像一個小型的天堂

感謝我的電腦
她陪伴了我不知道有多久
不管何時總是默默的陪伴著我
感謝感謝感謝

目錄

第一章 簡介.....	1
第二章 先前的研究.....	2
2.1 壓縮技術-- MPEG4 視訊編碼.....	2
2.1.1 運動補償(Motion Compensation)	3
2.1.2 離散餘弦轉換 (Discrete-Cosine-Transform)	3
2.1.3 鋸齒狀掃描 (Zigzag Scan)	3
2.1.4 變換長度編碼 (Run-Level-Coding).....	4
2.2 去交錯(DE-INTERLACING)	4
2.2.1 去交錯基本成因.....	4
2.2.2 僅以一個field 的資訊量去交錯.....	6
2.2.3 MA-IPC(Motion adaptive interlaced-to-progressive conversion).....	8
2.2.4 MC-IPC(Motion compensated interlaced-to-progressive conversion)	10
2.3 影像縮放 (IMAGE SCALING)	10
2.3.1 NN (Nearest Neighbor)	11
2.3.2 Bilinear	11
2.3.3 Win-scale.....	11
2.3.4 其他影像縮放方式.....	13
第三章 監控系統基本架構.....	14
3.1 概念.....	14
3.2 影像的擷取	14
3.3 硬體架構與軟體平台	15
3.4 系統的基本架構	16
3.4.1 進一步的系統架構.....	17
第四章 壓縮儲存.....	18
4.1 概念.....	18
4.2 壓縮儲存資料的分析.....	18
4.3 變動的判斷與雜訊的分析.....	19
4.3.1 雜訊的分析.....	20
4.3.2 雜訊門檻的初始設定.....	21
4.4 FRAME SKIP 機制.....	22
4.4.1 最大影像省略數值.....	23
4.5 QP 微調	25

4.6 區塊的變化(BLOCK-CHANGE)機制.....	27
4.7 動態調整門檻機制	28
4.8 整體系統架構.....	29
4.9 小結	31
第五章 去交錯(DEINTERLACING)	32
5.1 概念.....	32
5.2 測試環境	32
5.3 一個 FIELD 去交錯	33
5.4 時間域上提出的去交錯方式	41
5.5 小結	45
第六章 影像縮放(IMAGE SCALING)	46
6.1 概念	46
6.2 邊界的分析	47
6.3 提出的改進	48
6.3.1 邊界的判定.....	48
6.3.2 建構新的模型.....	49
6.3.3 邊界點的分群的方法.....	52
6.4 測試環境與實驗結果	52
6.5 小結	54
第七章 結果與討論.....	56
7.1 系統介面與實驗結果	56
7.2 結論	60



圖目錄

圖 2.1 MPEG4 編碼流程圖.....	2
圖 2.2 運動補償示意圖.....	3
圖 2.3 鋸齒狀掃描.....	4
圖 2.4 (A) 直接播放未去交錯影像，(B) 為部分放大後的影像.....	5
圖 2.5 (A) 為漸進式掃描示意圖，(B) 為交錯式掃描示意圖.....	5
圖 2.6 利用空間域資訊來作為去交錯的參考點.....	6
圖 2.7 利用空間域與時間域資訊來作為去交錯的參考點.....	9
圖 2.8 內插示意圖，C0 到 C3 為周圍像素點的灰階值，而要內插出 P 點位置的灰階值.....	11
圖 2.9 WIN-SCALE 方法下觀看像素點的方法.....	12
圖 2.10 當影像從 4*4 像素點要內插出 5*5 像素點時的情形.....	12
圖 3.1 影像的擷取流程.....	14
圖 3.2 系統基本架構圖.....	16
圖 3.3 壓縮、解壓縮、與影像加強方塊圖.....	17
圖 4.1 實驗室內監控攝影的場景.....	18
圖 4.2 壓縮(COMPRESSION)系統方塊圖.....	19
圖 4.3 (A)圖為監控攝影的場景，(B)為監控場景的灰階影像，(C)則為前後兩張靜止影像灰階資訊相減後的差量值，此差量值被放大十倍後秀出，圖片中越亮代表差量越大，(D)圖把影像切割成 3*3 等分的九個區塊。.....	21
圖 4.4 影像省略張數暫存器與影像對應關係.....	22
圖 4.5 畫面中可能對應的四種影像與其影像省略張數暫存器對應關係。(A)圖為一直判定為變動的影像，(B)圖為一直判定為變動後判定為靜止的情形，(C)圖為一直判定為靜止的影像，(D)圖為一直判定為靜止後判定為變動的情形.....	23
圖 4.6 當最大影像省略張數設為 10 而且畫面一直靜止時所儲存的影像.....	24
圖 4.7 最大影像省略張數調整流程圖.....	25
圖 4.8 最大影像省略張數調整流程圖與 QP 改動流程.....	26
圖 4.9 BLOCK-CHANGE 示意圖，(A)圖為原始影像，(B)把原始影像分為九個相等的區塊，(C)圖判定中間區塊有變動產生時的情形，(D)圖僅對中間區塊及其附近的畫面調整其影像內	

容，其餘部分則拷貝之前影像的內容	27
圖 4.10 動態調整雜訊門檻機制流程圖.....	28
圖 4.11 整個系統動態調整參數機制，(A)圖為判別影像是否為變動狀態的機制，(B)圖則是從(A)圖中承接 FRAME SKIP 機制，(B)圖內部為判斷影像是否為背景資訊，若為背景則啟動從新調整門檻機制，(C)圖是承接(B)圖的改動參數，(C)圖中主要是判斷是否要對最大影像省略數值以及 QP 做調整。	30
圖 5.1 影像加強部分區塊圖，本章所要討論的屬於其中的去交錯部分	32
圖 5.2 利用 FOREMAN SEQUENCE 前後張錯開的 TOP FIELD 和 BOTTOM FIELD 疊出來的效果.....	33
圖 5.3 五種方法在 FOREMAN SEQUENCE 內 PSNR 的比較圖	34
圖 5.4 此圖為 FOREMAN SEQUENCE 第一張圖形，左圖為原始影像與去掉 BOTTOM FIELD 影像以及五種方法去交錯後的圖形，右圖為左圖的左上角部分放大圖.....	37
圖 5.5 此圖為 FOREMAN SEQUENCE 第 251 張圖形，左圖為原始影像與去掉 BOTTOM FIELD 影像以及三種方法 PSNR 較好的方式去交錯後的圖形，右圖為左圖的左上角部分放大圖.....	38
圖 5.6 此圖為 FOREMAN SEQUENCE 第 251 張圖形，分別經過(A)ELA、(B)MEDIAN 以及(C)LA，去交錯後和原始影像的比較圖，亮的部分表示差量大於 30 的像素點，黑的部分表示差量小於 30 的部分.....	39
圖 5.7 去交錯時參考的資訊(A)圖為系統流程圖(B)圖為對應的像素點的灰階值	40
圖 5.8 各種去交錯方式的 PSNR 比較圖，其中 PROPOSED 為提出的方法。	41
圖 5.9 去交錯時參考的資訊(A)圖為整體去交錯系統流程圖，(B)圖為對應的像素點的灰階值。	42
圖 5.10 FOREMAN SEQUENCE 七種去交錯方式的 PSNR 比較圖	43
圖 5.11(A)圖為 FOREMAN 原始影像，(B)圖為空間域上提出的去交錯方式所還原出的影像，(C)圖為空間域加上時間域的去交錯方式所還原出的影像，(D)圖為(A)與(B)圖差量*10 的影像資訊，(E)圖則為(A)與(C)圖差量*10 的影像資訊，(F)圖則是(D)、(E)兩圖間的差量*10 的影像資訊圖。	44
圖 6.1 影像加強部分區塊圖，本章所要討論的屬於其中的影像縮放部分	46
圖 6.2 本圖為針對 FOREMAN 影像先用 NN 取樣到 240*192 個點之後在分別用 BILINEAR 和 NN 內插回 352*288 個像素點，上圖為內插後的影像圖，下圖則是上圖最右上角長寬各別放大四倍後的結果.....	47
圖 6.3 利用 FOREMAN 模擬分析門檻值的設定。左上圖為原始影像，依序往右邊為設定門檻值為 5、8、10、20、30，利用相鄰四點的最大最小值的差量算這區塊的斜度，若差量大於此門檻則為亮的，反之則為暗的區塊。	49
圖 6.4 邊界的三種情況，(A)斜的、(B)橫的、(C)直的邊界。	50

圖 6.5 各種邊界所對應的端點值，(A)斜的，角落值不確定、(B)橫的、(C)直的邊界。	50
圖 6.6 可能的邊界對應的端點值，(A)斜邊、(B)橫的、(C)一個像素點寬的斜邊。	51
圖 6.7 NN 方法對應此三種端點值的處理方式，(A)斜邊、(B)橫的、(C)對角。	51
圖 6.8 提出的模型，(A)斜邊、(B)橫的、(C)對角。	51
圖 6.9 本圖是針對 FOREMAN 影像先用 NN 取樣到 240*192 個點之後在分別用四種內插方式補回 352*288 個像素點。	52
圖 6.10 本圖為針對 FOREMAN 影像先用 NN 取樣到 240*192 個點之後在分別用四種內插方式補回 352*288 個像素點，並且把最右上角部分長寬各放大 4 倍。	53
圖 6.11 本圖為針對 FOREMAN 影像先用 NN 取樣到 240*192 個點之後在分別用四種內插方式補回 352*288 個像素點，並且把最左上角部分長寬各放大 4 倍。	53
圖 6.12 四種內插方法的 PSNR 比較圖，先用 NN 取樣到 240*192 個點之後在分別用四種內插方式補回 352*288 個像素點。	54
圖 7.1 操作介面，左端為即時間控並且同時壓縮儲存，右端為回放的介面	56
圖 7.2 交錯式影像訊號	57
圖 7.3 去交錯式機制下所產生的影像訊號	58
圖 7.4 影像縮放機制，左上角為原始壓縮時影像大小 CIF 格式，其他的則是用其他影像縮放方法放大為 640*480，方法依序為 NN、6-TAP FIR、BILINEAR、WIN-SCALE 以及 PROPOSED METHOD	59

表目錄

表 3.1 壓縮方式與效率比較	31
表 5.1 各種 SEQUENCE 與 PSNR 比較圖	44



第一章 簡介

隨著科技越來越發達，我們能夠享受到科技所帶來方便的生活，其中一樣就是有關於多媒體快速的發展。視訊已經在所帶來的便利已經慢慢充斥在生活之中，在現今都市化的結構中，忙碌的生活之中，大家各司其職，有的時候我們希望能夠常常監控某些我們無法兼顧到的場景，如家裡是否有遭小偷入侵、道路的上有沒有人亂飆車、商店的物品是否有按照順序排好等等，原本可能需要耗費大量的人力資源才可以確認這些地方，現今只需要一個取像設備與配應的硬體與軟體監控平台，透過多媒體的傳輸以及數位化的系統，讓我們有辦法能夠把影像記錄下來，當我們希望的時候能夠隨時調閱現在的影像或是已經發生的影像。

我們可以在所需要的地方架設擷取系統來拍攝我們所需要的畫面，根據不同情況做不同的處理，一般來說，監控系統需要建立一個監控的平台，從影像的拍攝、影像的擷取、影像的辨識、影像的壓縮、影像的儲存、儲存資料的重播、影像的增益等等都需要作一系列統整的處理，在數位化的系統之下，如何在影像的品質、壓縮的效率、資料量大小、運算速度等等各種因素中取得平衡，將是主要面臨的問題。

接下來在第二章，我們將先簡介一些在監控中遇到的問題以及一般的解決方式，主要分為成三大部分，包含影像壓縮儲存、交錯式(Interlace)影像轉為漸進式影像(Progressive)問題以及影像縮放(Image Scaling)三個重點，之後在第三章介紹整個硬體架構與軟體平台並且簡述本監控系統主要架構，而第四章將對影像的壓縮進一步的討論，第五章根據交錯式影像問題提出自己的解決方法，並討論影像品質的改進，第六章則是針對影像的縮放提出一些討論和比較，最後在第七章作出結論。

第二章 先前的研究

這一章主要分成三個部分：第一部分主要要討論的是壓縮相關的問題，進一步將會介紹 MPEG4 編碼規格;第二部分包含交錯式影像的成因和解決之道；第三部分則是探討有關影像縮放的各種問題以及相關的方法。

2.1 壓縮技術 -- MPEG4 視訊編碼

龐大的資料量一直是監控系統中一個難以解決的問題，壓縮則是可以把多餘的或是不重要的資訊量拿掉的方法，除掉冗餘的資訊，僅僅留下有用的資訊，可以節省儲存設備的寶貴空間。

在目前的視訊監控系統中，主要包含了幾種主要的壓縮方式，其中包括 MPEG2、MPEG4、H.264/AVC、Motion-JPEG 等等，各有其優勢與劣勢的地方，而在建立我們的監控系統平台時，在考量運算速度與影像品質的情況下，最後採用了 Optimized MPEG4 Simple Profile 來當作我們壓縮系統的基本骨架，接下來將會介紹 MPEG4 的基本架構

下圖為 MPEG 編碼器之方塊圖：

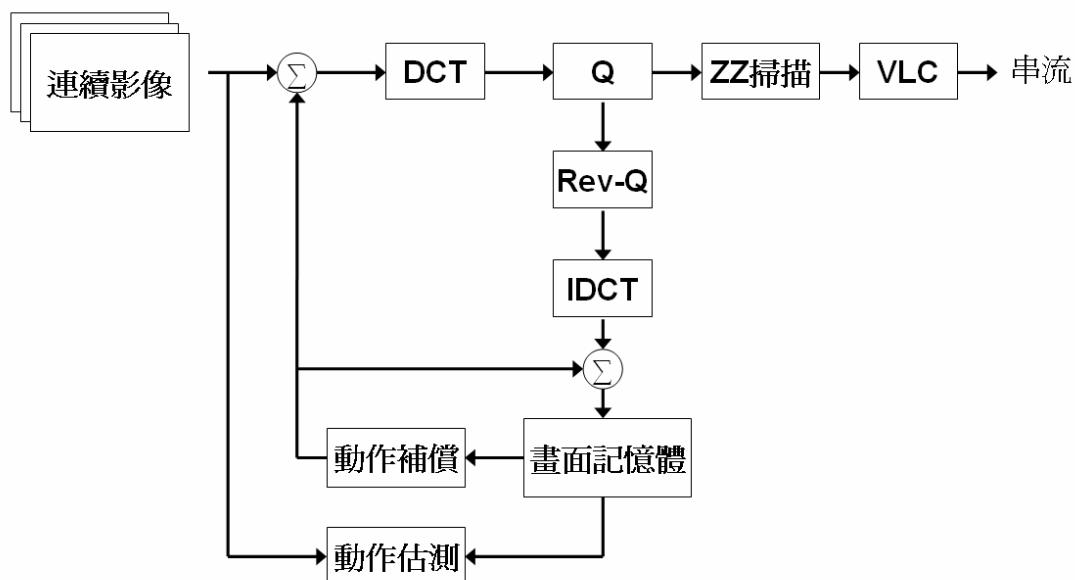


圖 2.1 MPEG4 編碼流程圖[1]

下面將簡介 MPEG-4 編碼中的主要部份作基本的介紹。

2.1.1 運動補償(Motion Compensation)

運動補償是相當基本的視訊編碼過程，由於一般視訊具有大量的時間性冗餘 (Temporal Redundancy)，經由運動補償則可以大幅減低資料量，達到壓縮的目的。運動補償的過程簡述如下：

區塊基礎 (Block-based)

把畫面切割為許多小單位，通常為正方塊。

運動估測 (Motion Estimation)

對於每一個方塊，在前一張畫面中尋找最相似的方塊，然後記錄其位置。所記錄的位置，即是估測出來運動前的位置。

運動補償 (Motion Compensation)

把目前的方塊的影像數值減去運動估測對應的方塊影像數值。

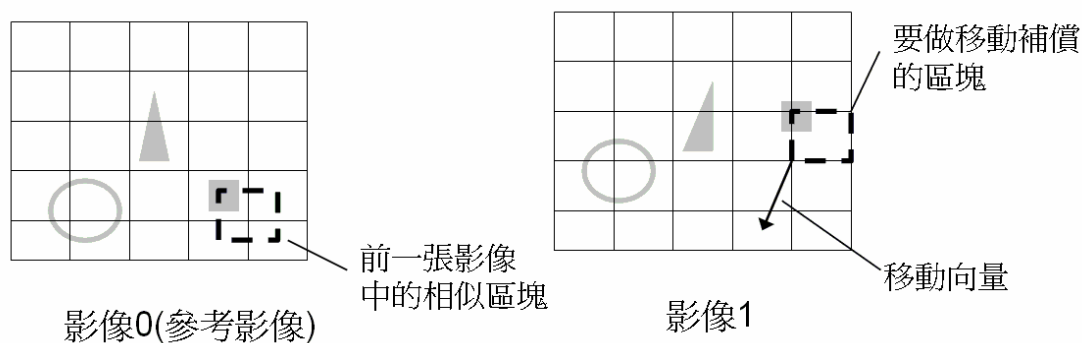


圖 2.2 運動補償示意圖[1]

2.1.2 離散餘弦轉換 (Discrete-Cosine-Transform)

離散餘弦轉換是影像壓縮與視訊壓縮常見的方法，由於人眼對於高頻訊號比較不敏感，所以影像經過離散餘弦轉換之後，高頻的係數可以比較粗略地量化，如此一來，可以大幅增加壓縮比，而且對於影像品質的影響很小。

2.1.3 鋸齒狀掃描 (Zigzag Scan)

二維的離散餘弦轉換係數，經過量化後，高頻部分數值很小，幾乎皆為零，能量集中在低頻係數。而鋸齒狀掃描將係數由低頻排列至高頻，高頻的零被集中

排列，這樣子的排列順序經由變動長度編碼(Run Level Coding)，可達到可觀的壓縮比。

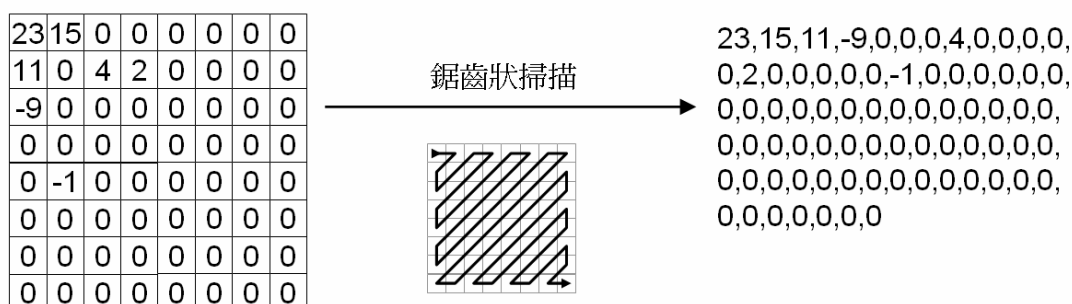


圖 2.3 鋸齒狀掃描[1]

2.1.4 變換長度編碼 (Run-Level-Coding)

鋸齒狀掃描出來的序列，能量集中在前面，後面大多是零，MPEG-4 採用變換長度編碼對序列壓縮。變換長度編碼能非常有效地壓縮鋸齒狀掃描後的序列，延續前一小節的例子，我們只要用 7 個變換長度編碼符號(Symbol)即可完成長度為 64 的序列的編碼：

(Run, Level, End) → (0,23,0) (0,15,0) (0,11,0) (0,-9,0) (3,4,0) (5,2,0) (5,-1,1)

而各個不同的符號，根據機率的統計，再施以霍夫曼編碼(Huffman coding)。

2.2 去交錯(De-interlacing)

在數位漸進式(progressive)影像系統和傳統交錯式(interlace)的影像系統之間，常常會存在有轉換的問題，主要的成因就是在於兩個系統之間有著資料的時間差。當我們用數位漸進式掃描的設備上面，要播放交錯式影像的輸入時，則需要進行去交錯的動作，提高畫面的品質。

2.2.1 去交錯基本成因

下圖為去直接把交錯式影像時 2 個 field 合併的結果

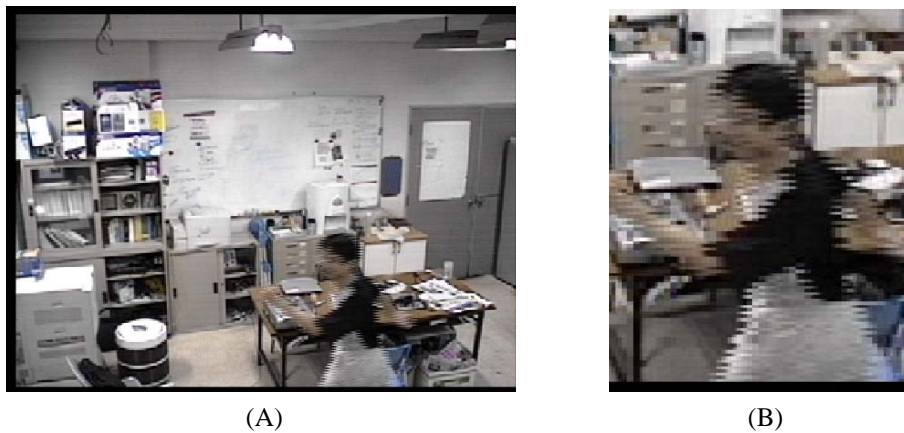


圖 2.4 (A)直接播放未去交錯影像，(B)為部分放大後的影像

假如畫面為交錯式掃描，而畫面有變動發生時，比如移動、轉動、縮放、覆蓋等等情況時，假若不去交錯的話直接在漸進式掃描的設備上播放，將會產生鋸齒狀、模糊狀，各種損害影像品質的不良影響，主要的原因就在於交錯式掃描是以 field 為單位而漸進式掃描是以 frame 為單位，如圖 2.4 所示

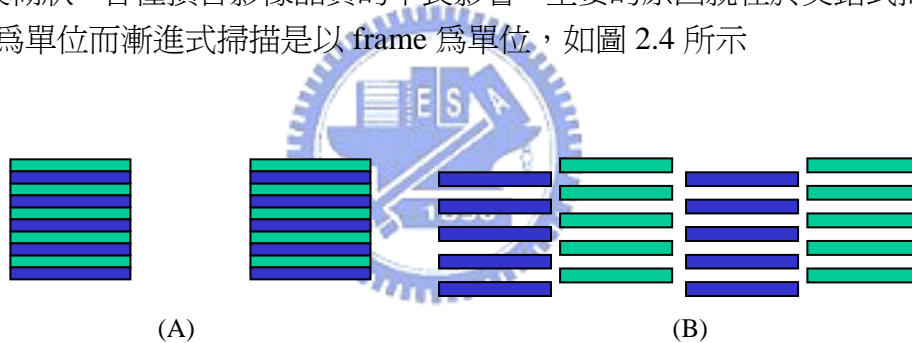


圖 2.5 (A)為漸進式掃描示意圖，(B)為交錯式掃描示意圖

漸進式掃描系統是以一整個畫面為單位一張一張擷取與播放，而交錯式掃描系統卻是把一個完整的畫面切割成兩個一上一下互相交錯的 field，一次以一個 field 為單位來擷取與播放。假若以每秒鐘 30 張的交錯式系統的攝影機來拍攝的話，可以視為每 1/60 秒擷取一個 field 的影像，而當我們在漸進式掃描影像設備上播放時，有變動的部分就會發生時間軸不一致的情形，從而產生鋸齒狀或是模糊的不良影響，所以在播放之前我們會需要作一個去交錯的動作，把交錯式的影像轉換成漸進式的影像。

我們假定 P 為漸進式影像，而 $P(i, j, t)$ 為時間點 t 在 (i, j) 位置的像素點值； I 為交錯式影像而 $I(i, j, t)$ 為時間點 t 在 (i, j) 位置的像素點值，當我們要還原出影像資訊的時候可以視為

$$I(i, j, t) = 0 \text{ for } i \bmod 2 \neq t \bmod 2$$

$$P(i, j, t) = \begin{cases} I(i, j, t), & \text{for } i \bmod 2 = t \bmod 2 \\ N(f), & \text{Otherwise} \end{cases} \quad (2.1)$$

$N(f)$ 就是我們要利用去交錯的方式內插出來的像素值，去交錯從解決方法所用資訊來看一般可以分成三個大類，分別是只用一個 field 去還原，也就是僅考慮時間點 t 的資訊、MA-IPC(Motion adaptive interlaced-to-progress conversion)，加入了前後時間點的資訊與 MC-IPC(Motion compensation interlaced-to-progress conversion)，利用運動補償來還原三大類，以下將一一介紹。

2.2.2 僅以一個 field 的資訊量去交錯

顧名思義，這類的方法主要就是只用一個 field 的資訊量去還原出一整張 frame 的影像，一般說來，由於只有一個 field 的資訊量，所以還原的效果會比其他兩種方法來的差，但是相對的速度以及硬體方面卻也是最快最簡單的。常見的方法有以下幾種 LD (line doubling)、LA(line average)、ELA(edge based line average)、中位數(median)、平均數(mean)等等，而其他的方法則主要是架構在這些方法之上。



圖 2.6 利用空間域資訊來作為去交錯的參考點

上圖為一個 field 去交錯的示意圖，當我們以一個 field 為基準要內插出另外一個 field 的資訊量時，我們可以利用的資訊主要是間隔一個垂直軸方向的資訊，如上圖所示，X 是我們希望補償的像素點的灰階資訊量，而我們常常利用它四周圍已經存在的像素點來作為參考點，如 ABCDEF 所代表的為附近點的灰階值。

一、 LD (Line Doubling)[2]

這個方法可以說是最簡單也最直覺的方法，直接複製整塊已經有的 field 資訊到遺失的資訊上，幾乎完全不需要運算量，但是一般來說影像品質也是最差的，其還原的方法為

$$X = B \quad (2.2)$$

二、 LA(Line Average)

這個方法單純用 PSNR 來看是非常高的，因為他的方法是直接取他最近的兩個點的平均值，也就是他的上面和下面的均值來當成遺失點的值，但是從影像來看，在邊界不是直線的部份，這個方法將會產生嚴重的鋸齒狀波紋，但是當上下兩點為同一個區塊時，這個方法可以有效地內插出中間遺失的點。

$$X = \frac{1}{2}(B + D) \quad (2.3)$$

三、 ELA (Edge Based Line Average)[3]

這個方法跟 LA 很像，差別只是在於 LA 是只有取上下兩個點來做平均，而 ELA 則是多取了兩條線(左上右下以及右上左下)的線來作為可能的均值。取哪一條則是利用其線上兩端點的差量大小來區分，取三條線中差量最小的那條線，利用均值的方式，補出中間遺失的值，這個方法在邊界部份可以得到不錯的效果，但是在影像複雜的部份(如樹葉天空)容易造成區塊狀的產生。

$$\begin{aligned} R &= |A - F| & S &= |B - E| & T &= |C - D| \\ \text{If } (\min(R, S, T) = R) & & X &= (A + F) / 2 \\ \text{Elseif } (\min(R, S, T) = S) & & X &= (B + E) / 2 \\ \text{Else} & & X &= (C + D) / 2 \end{aligned} \quad (2.4)$$

四、 中位數(Median)[4]

這方法主要是取上、下、左上、左下、右上、右下共計六個點的值來取其中位數，取出最中間兩個值來做平均，這個方法的整體效果很不錯但是運算量要相對地來得高一些。

$$X = \text{Median}(A, B, C, D, E, F) \quad (2.5)$$

五、 平均數

這方法主要是取上、下、左上、左下、右上、右下共計六個點的值來取其平均值來補遺失點的值，因為六個點可能分屬不同區塊影像內容，所以效果不見得會來的好。

$$X = \text{Mean}(A, B, C, D, E, F) = (A + B + C + D + E + F) / 6 \quad (2.6)$$

六、 其他方法

其他的方法主要是從上述五種方法演化而來的，像是強化形 ELA[5]，多找出幾條線來做，可能的方向是從原本六個點內差出其他點，或是另外多找其他存在的點來做平均，不然就是找出例外值把差排除在外，或者是利用不同比重的平均值來作為預測值等等方法，而這些方法將會額外增加運算複雜度。

有一個方法就是利用 Median 為基準的去交錯方式[6]，姑且稱之為 Group Median，它的方法是先判斷附近點哪些比較可靠，並且把 $(B+E)/2$ 的資訊一併拿進來算 Median 的值，由於它判斷的個數為奇數點，所以不必再經過平均的動作。

$$P = |B - F| + |A - E| \quad Q = |C - E| + |B - D|$$

$$\begin{aligned} \text{If } (P < Q) & \quad X = \text{Median}(A, B, E, F, (B + E) / 2) \\ \text{Elseif } (P > Q) & \quad X = \text{Median}(B, C, D, E, (B + E) / 2) \\ \text{Else} & \quad X = \text{Median}(A, B, C, D, E, F, (B + E) / 2) \end{aligned} \quad (2.7)$$

整體來說，只用一個 field 來做去交錯，速度上是最快的，但是由於資訊量並不足夠，所以效果頂多達到一個極限，並沒有辦法得到非常好的效果。



2.2.3 MA-IPC (Motion adaptive

interlaced-to-progressive conversion)

除了當個空間域上的資訊外，在影像轉換中，還存在著時間域上的資訊，而 MA-IPC 的基本做法主要就是把影像切割成前景和背景，當此像素點被判別為有運動而成為前景時，則使用空間域上面的資訊去內插；而當此像素點被判別為背景資訊時，則直接使用空間域上面的資訊去內插。

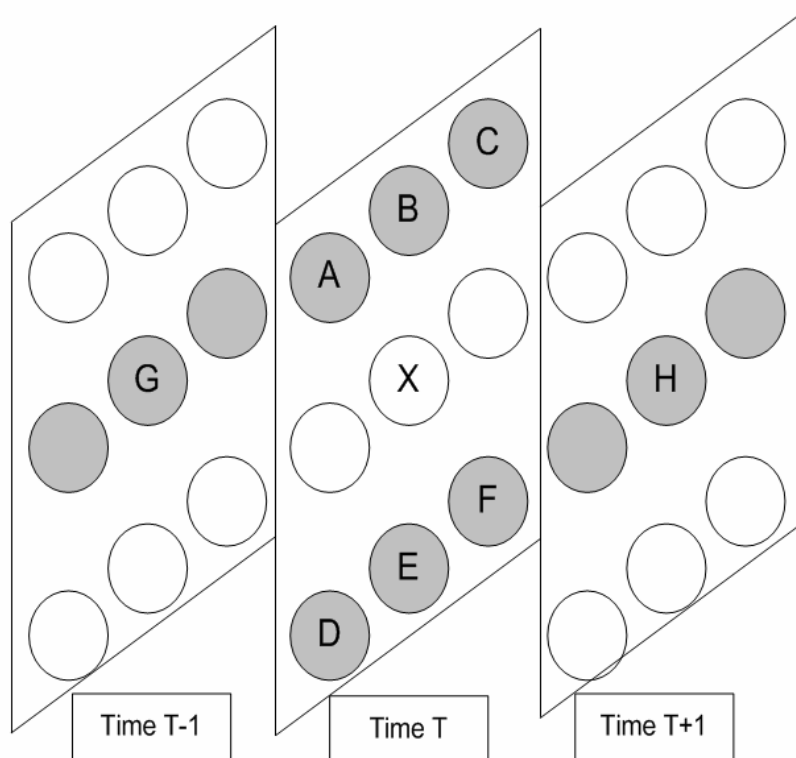


圖 2.7 利用空間域與時間域資訊來作為去交錯的參考點

交錯式影像輸入時，在時間軸為基數點時，影像更新上半個畫域(Top field)，而當時間點為偶數時間點時，影像將會更新另外半個畫域(Bottom field)，因此當我們要還原出時間點 T 的整張影像時，除了本身時間點的半個畫域資訊外，很自然地就會參考到前後兩張的資訊，如圖 2.7 所示，當我們想要內插出時間點 T 中像素點的灰階值 X 時，除了利用同一張畫面內的資訊外，也可以利用前後張的資訊量來加以還原，圖 2.7 中 ABCDEFGHIJKL 為各個參考點的灰階值。

最基本的 MA-IPC 方式就是直接使用 LA(Line Average)來當做去交錯的機制，當判定為前景部分時則使用空間域上的 LA；而當判定為背景時，則使用時間域上的 LA。

$$\begin{aligned} \text{If (foreground)} \quad & X = (B + D) / 2 \\ \text{else} \quad & X = (H + K) / 2 \end{aligned} \quad (2.8)$$

另外有兩個方式是根據這個方法所提出，一個是結合了 ELA 和 temporal 的資訊來補償[7]，稱之為空間與時間域上的 ELA 去交錯方式

$$\begin{aligned} Q1 &= |A - F| & Q2 &= |B - E| & Q3 &= |C - D| \\ Q4 &= |G - L| & Q5 &= |H - K| & Q6 &= |I - J| \end{aligned}$$

$$\begin{aligned}
\text{If } (\min(Q1, Q2, Q3, Q4, Q5, Q6) = Q1) & \quad X = (A + F) / 2 \\
\text{Elseif } (\min(Q1, Q2, Q3, Q4, Q5, Q6) = Q2) & \quad X = (B + E) / 2 \\
\text{Elseif } (\min(Q1, Q2, Q3, Q4, Q5, Q6) = Q3) & \quad X = (C + D) / 2 \\
\text{Elseif } (\min(Q1, Q2, Q3, Q4, Q5, Q6) = Q4) & \quad X = (G + L) / 2 \\
\text{Elseif } (\min(Q1, Q2, Q3, Q4, Q5, Q6) = Q5) & \quad X = (H + J) / 2 \\
\text{Else} & \quad X = (H + I) / 2
\end{aligned} \tag{2.9}$$

另一個方式則是挑出特殊情況下會造成畫面補償不好時做例外的處理[8]，這個方式是先把影像分成三個種，分別是沒有運動(zero-motion)、全域運動(global motion)或是整條線狀運動(line motion)，若是有的話就先加以簡單的補償，在MA-IPC 補不好的情況下把 MC-IPC 的優點考慮進來，可以說是兩種方式的結合。

2.2.4 MC-IPC (Motion compensated

interlaced-to-progressive conversion)

這個方法主要也是利用前後兩張 field 的資料以及現在這張 field 的資料來加以還原出整張影像，與 MA-IPC 不同的是，它是利用切塊來找尋運動向量的方式補償出遺失的資訊，而 MA-IPC 並沒有去找尋物體的運動向量。一般來說，MC-IPC 補償出影像的效果與運動向量的精準度有絕對的關係，若運動向量找得越準確，去交錯的效果會越好，但相對的運算複雜度也會提高，基本上這個方法為三種方法中效果最好的，但是運算複雜度也是最高的。

2.3 影像縮放(Image Scaling)

數位化影像有很多的優點，其中之一就是很方便的對影像內容做處理，由於影像輸入大小不見的是我們想要的(ex. QCIF、CIF、VGA、XGA)，我們會希望將影像轉換成我們希望的大小，這時候影像縮放技術就會被用到。基本的影像縮放可以被視為是重新取樣二維的資訊。從以前到現在，已經有很多的方法被提出來討論，下面將就幾個常見的方式做適當的討論與分析。

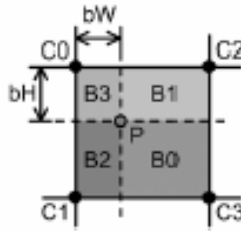


圖 2.8 內插示意圖，C0 到 C3 為周圍像素點的灰階值，而要內插出 P 點位置的灰階值[10]

2.3.1 NN (Nearest Neighbor)

這是最簡單的影像縮放的方法，要內差出的點直接找最近的值來替代，這可以視為 0 階的內差方式，他能夠有很好的高頻成分被保留，但是相對的，aliasing 所造成的影像會很嚴重，使得影像的品質會大打折扣。

$$\begin{aligned}
 & \text{If } (bW \leq (C2 - C0) / 2 \quad \&\& \quad bH \leq (C1 - C0) / 2) \quad P = C0 \\
 & \text{Elseif } (bW > (C2 - C0) / 2 \quad \&\& \quad bH \leq (C1 - C0) / 2) \quad P = C2 \\
 & \text{Elseif } (bW \leq (C2 - C0) / 2 \quad \&\& \quad bH > (C1 - C0) / 2) \quad P = C1 \\
 & \text{Else} \quad P = C3
 \end{aligned} \tag{2.10}$$



2.3.2 Bilinear

這是最普遍的影像縮放方式，其內差點的值隨著和相鄰點的距離而呈一階線性內差，影像會有逐漸被模糊掉的趨勢。為了方便起見我們把 C0 點到 C2 點與 C0 點到 C1 點的距離皆定為 1，而利用 Bilinear 來內插出 P 點的公式如下

$$\begin{aligned}
 & P = B0 * C0 + B1 * C1 + B2 * C2 + B3 * C3 \\
 & \begin{cases} B0 = (1 - bW) * (1 - bH) \\ B1 = (1 - bW) * bH \\ B2 = bW * (1 - bH) \\ B3 = bW * bH \end{cases} \tag{2.11}
 \end{aligned}$$

2.3.3 Win-scale[9]

這個方法可以視為 NN 和 Bilinear 的綜合體[10]，在不同的影像縮放大小的情況下，所配合比例也不盡相同，主要的概念為接近參考點的話用 NN，而在一段距離之外用 bilinear，這樣可以保留 NN 高頻的特性，相對的 aliasing 也不會很嚴重。

Win-scale 主要的方法在於把一個個像素點的觀念推展成一個個均勻分佈的區塊，然後一張影像就可以根據其所佔像素點的長寬個數來看成一塊塊均勻分佈的區塊所組成。

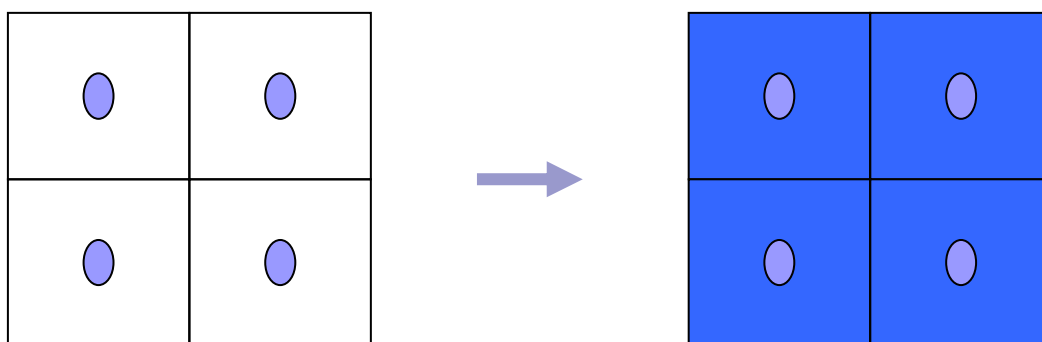


圖 2.9 Win-scale 方法下觀看像素點的方法

而當我們要做影像縮放時，可以以原始影像大小為起始值，以要縮放的大小為代測值，直接縮放一定比例後貼到起始的影像上，這樣就會產生區塊的重疊部分，而我們就可以利用重疊部分的比例關係來當成加權的比重，算出我們所要內插的像素點的資訊。

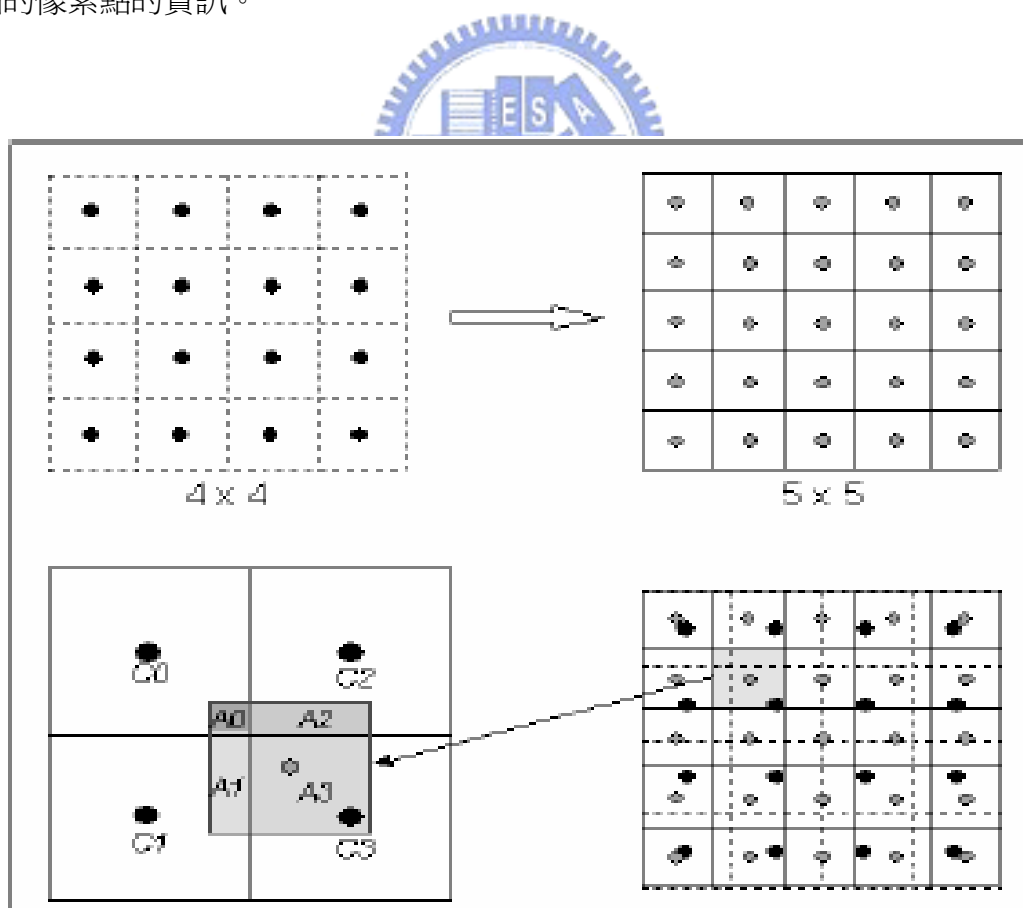


圖 2.10 當影像從 4*4 像素點要內插出 5*5 像素點時的情形[9]

如圖 2.10 所示，當我們要把 4*4 大小的原始影像內插成 5*5 的大小時，我們

先把 4*4 大小的影像資訊展開成長寬為 1 的 16 個正方形區塊，把原始影像視為這 16 個等分的區塊所組成，而當我們要內插出 5*5 的影像時，我們先建立出一個 5*5 的區塊，並且按照要內插的長寬比壓縮成原始參考影像大小，之後再貼到原始大小影像上面(如 2.10 右圖所示)，這麼一來，希望內插出來的像素點區塊與原始影像的參考區塊之間就會有明顯的重疊部分，而這個部分就是我們拿來做為比重算出代測點的影像內容，如 2.10 左圖所示，代測點被原始影像的四個點所圍住，這四個點分別為 C0、C1、C2、C3，而他們之間的面積重疊的部分為 A0、A1、A2、A3，而我們的代測點 P 的資訊就可以利用下列的示子算出其數值

$$\begin{aligned}
 P &= SF * (A0 * C0 + A1 * C1 + A2 * C2 + A3 * C3) \\
 &= w0 * C0 + w1 * C1 + w2 * C2 + w3 * C3 \\
 SF &= 1 / \text{area of filter window}
 \end{aligned}
 \tag{2.12}$$

其中 SF 為 normalize factor 讓 $w0+w1+w2+w3=1$ 。

當我們強制設定 SF=1 的時候，也就是假設代測點往外長的的面積大小強制設定成和原始影像的長寬一樣大時，這時候證明 Win-scale 所算出來的結果會與 Bilinear 做法相仿[5]，而當把 SF 趨近於無窮大的時候，則此時的內插方式又會與 NN(Nearest Neighbor)相同，所以 Win-scale 方法可以視為 NN 與 Bilinear 的結合，利用繪畫出面積的長寬與原始影像長寬比，來進一步調整其 NN 與 Bilinear 的比重。以原始的 SF 大小來看，可以保留住 NN 對於邊緣附近的銳利度，而另一方面又利用線段性一階線性內插來算出偏離原像素點位置的代測值，保留住了兩種方式的優點。

2.3.4 其他影像縮放方式

一、多項式類，諸如 Bi-cubic、Polynomial

二、頻譜分析方式，raise-cosine[4]

上述方式的複雜度都比單純的 0 階或是一階線性內差複雜的多，除了上述提到的方式之外，其他還有一些僅是針對 2 或是 2 的倍數來做縮放的特殊化處理。

第三章 監控系統基本架構

本章說明整個監控系統的整體概念，硬體架構，軟體平台，並指出與一般的監控系統不同的地方。

3.1 概念

由於科技的發達，攝影機越來越普及，監控系統越來越廣為人所接受，一個適當的平台顯得越來越重要，一個基本的監控系統至少要具備三種功能，包含錄影、重播、監視的功能，這樣除了能夠即時看即時的影像外，也能夠隨時調閱之前的影像，查詢是否有異常事件發生。

隨著越來越多人投入這塊市場，漸漸地也會發現一些需要處理的問題，比方說去交錯，儲存裝置容量不夠等等，而本篇論文主要研究的目的即在即時監控的情況下，除了改善影像品質外，能夠把壓縮的效果進一步地提高。

3.2 影像的擷取

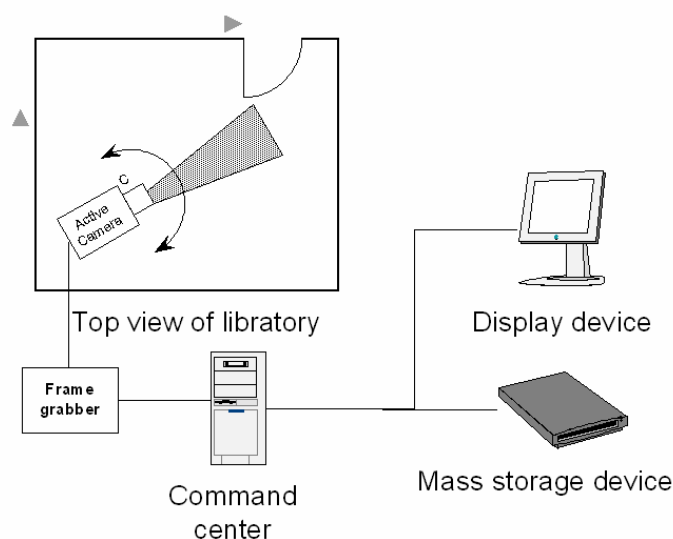


圖 3.1 影像的擷取流程

如圖 3.1 所示，我們需要有一台取像設備來擷取影像資訊，接著透過影像擷取卡把擷取到的影像資訊抓下來傳送到電腦端，經過電腦處理後，把影像的資訊透過顯像設備展示出來，或者是儲存進儲存裝置內等待以後需要的時候重播。

3.3 硬體架構與軟體平台

■ 電腦配件硬體部分

- 主機規格：P4 3.0G CPU、256MB*2 DDR2 RAM
- 影像擷取卡：PicProdigy-Color
- 攝影機型號：D7720 SERIES HIGH SPEED DOME CAMERA

■ 電腦配件軟體部分

- 發展平台: LV-SDS (Leutron Vision Software Development Suite)
- 壓縮規格: Optimized MPEG-4 14496-2 Simple Profile

■ 影像儲存部分

- 影像大小為 CIF(352*288)格式
- 速度為即時錄影(每秒 15 或 30 張)
- 儲存格式: MPEG4 壓縮格式，以十分鐘為單位儲存檔案
- 儲存裝置: SATA 硬碟儲存裝置

3.4 系統的基本架構

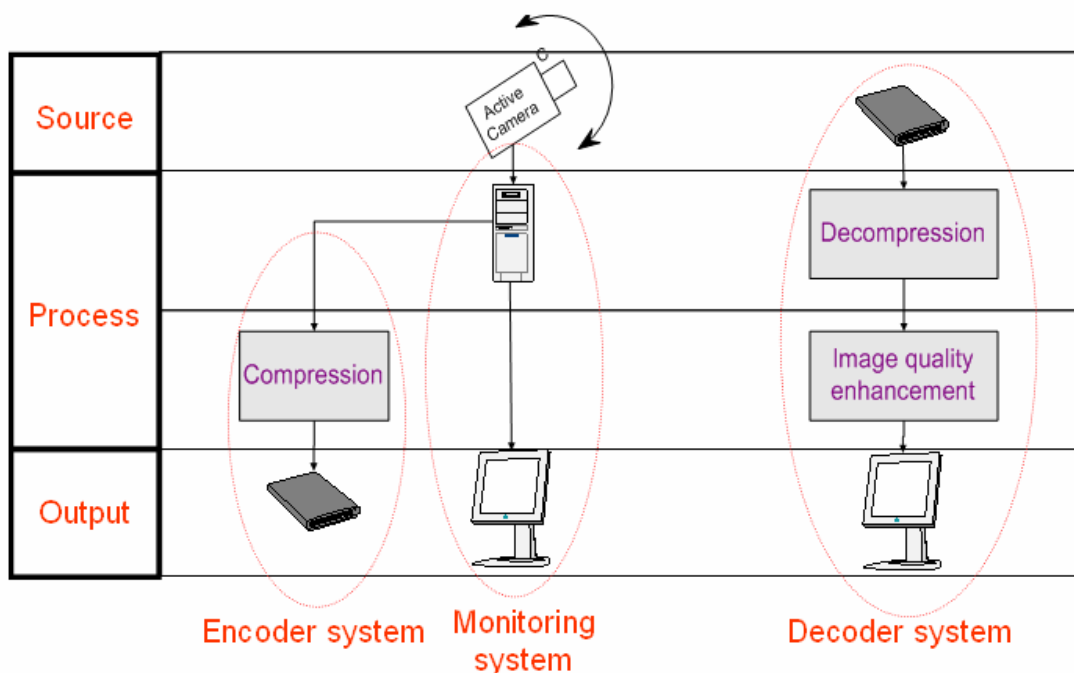


圖 3.2 系統基本架構圖

整個系統可以分做兩大部份，第一個部份包含了影像擷取，影像壓縮儲存以及即時影像的播放，而第二個部份則是已經錄過的影像的重播。當影像從攝影機擷取下來之後，需要經由傳輸線傳送到影像擷取卡上，再透過影像擷取卡傳送到電腦去做進一步的處理，資料進入電腦之後，一方面會直接顯示在螢幕上，另一方面則是經過壓縮處理之後儲存到儲存裝置內；另一方面，當要調閱之前的資料時，要先從儲存裝置內找到該時間的影像資料，再透過顯示裝置顯示出來。

當在影像擷取卡擷取完影像的之後，我們發覺單純僅僅用 MPEG4 simple profile 去做壓縮後，資料量還是很大，因此我們提了一套建構於 MPEG4 之上的壓縮方法，先對畫面作分析，進一步的把畫面作壓縮的動作，此外，由於拍攝的畫面會因為擷取攝影機是屬於交錯式掃描的影像擷取因而造成在播放端時需要經過去交錯的一道手續，最後當錄影重播時，由於壓縮時影像大小只有 QCIF 的大小，再經由壓縮儲存後影像品質在重播時不夠清楚，因此會額外再做影像放大的動作。

3.4.1 進一步的系統架構

針對於監控系統的部分，我們特別針對三個區域去進行討論，分別為壓縮機制、解壓縮機制、以及影像的加強這三大部分，而其中解壓縮部分與壓縮部分是相對應的，所以我們之後主要要討論的部分主要是在於壓縮解壓縮以及影像的加強部分，而又細分為三大塊，主要是壓縮機制，去交錯機制以及影像的縮放機制三類，將會分別在第四、第五、第六章節中做更進一步的探討。

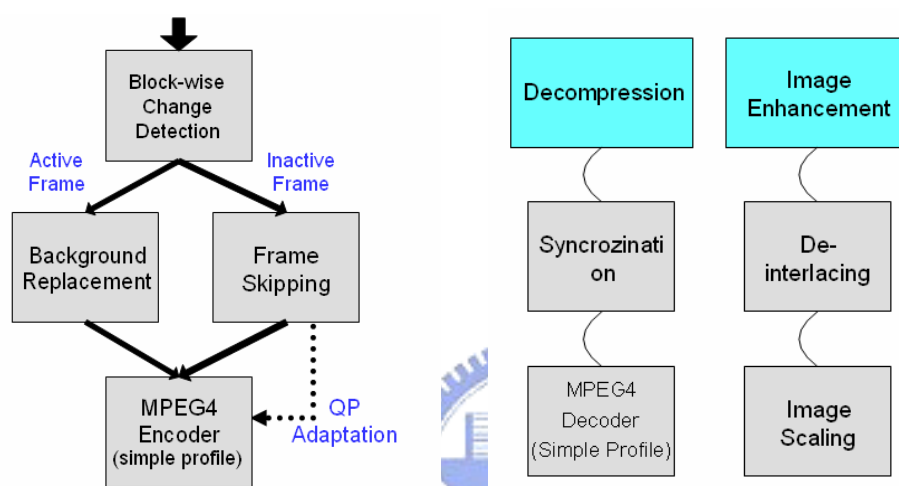


圖 3.3 壓縮、解壓縮、與影像加強方塊圖

如圖 3.3 所示，在影像壓縮方面，由於速度、品質、壓縮率等等考量下，採取了 MPEG4 simple profile 的方式作為基礎，在這個基礎之上，針對影像不動的區域僅產生 noise 的地方，我們不希望耗費資訊量去存它，因此，當判定整張影像沒有變化時，我們會做 frame skip 的動作，讓這個時間點幾乎完全不需要耗費到儲存空間，而當影像畫面有變化時，把影像分區塊的來做分析，僅儲存有變化區塊的資料量，這樣其他區塊的雜訊才不至於一塊存進來，進一步節省資料的儲存空間，最後在進行 MPEG4 壓縮前，會再進一步做 QP 的調整，當畫面變動頻繁時，把 intra/inter QP 調到 4/4 的值，而當畫面一值趨於完全不動的情況之下，會逐漸把 intra/inter QP 的值調整到 8/31 的值，讓壓縮率再進一步的提高。

而在解碼端，在 MPEG4 解碼之前，需要把 frame skip 的張數算清楚，這樣時間才不會錯亂掉，之後再丟進 MPEG4 decompress，而當解碼完成後才進一步做影像的增益，把交錯式的影像轉換成完整的影像，之後再看看需不需要放大顯像，這邊的放大固定比率從 CIF(352*288)大小放大為像素點 640*480 的大小。

第四章 壓縮儲存

本章說明本篇論文提出壓縮儲存的方法，從基本概念，推展至實作上的重點。

4.1 概念

在監控系統的領域之中，影像儲存一直是一個存在的問題，因為儲存裝置的大小有限，但是高品質的錄影卻要佔據十分龐大的空間，在壓縮標準裡面，本系統採用了 MPEG4 simple profile，主要的優點就是速度、品質、壓縮率三者方面的考量，然而在用完了 MPEG4 的壓縮方式之後，我們嘗試著向 MPEG4 並沒有考慮到而仍然會壓縮資料的地方—雜訊—伸出了我們的大刀，希望能夠達到畫面實際上有變動時，本系統才會把變動儲存起來，而當畫面沒有變動時，我們就不要浪費我們寶貴空間，直接跳過，這麼一來將可以將壓縮的效率進一步的提高。

4.2 壓縮儲存資料的分析

當我們利用 MPEG4 simple profile 來壓縮輸入影像後，可以發現它的壓縮率從 30 倍到 100 倍不等，以一個每秒 30 張、記錄十分鐘的影像來看，YUV420 的原始資料大小為 2.7GByte，而當經過壓縮過後，運動頻繁的影像被壓縮成 100 多 MB，而靜止的影像則會被壓縮至 30MB 左右，下面為實驗用拍攝的場景。



圖 4.1 實驗室內監控攝影的場景

從 MPEG4 壓縮的角度來看，壓縮的影像可以被區分為兩個部分，I frame 和

P frame，其中 I frame 僅利用本身同一張影像的資訊量去做壓縮，而 P frame 則是利用時間軸上的資訊，針對前面一張的影像與現在影像的差量來儲存。對靜止狀態的影像而言，30MB 的資訊一方面來自於 I frame 記錄場景的資訊，另一方面，則是 P frame 的資料，而 P frame 的資料主要成份則是雜訊。

靜止的影像對於我們來看，這些資訊是完全不需要的，只需要儲存第一張影像，後面的跟它重覆即可，因此我們採用了 frame skip 的機制，這樣一來對於靜止的影像我們只需記錄一張 I frame 即可；另一方面，對於有運動狀態的影像來看，其中並不是所有的影像內容都在動，我們希望只記錄那些有變動的部分，而其他沒有變動的部分屬於雜訊的部分則不需要記錄下來，可以利用之前已經記錄的資訊補過來即可，因此我們提出了 Block-Change 機制來達到這個效果。

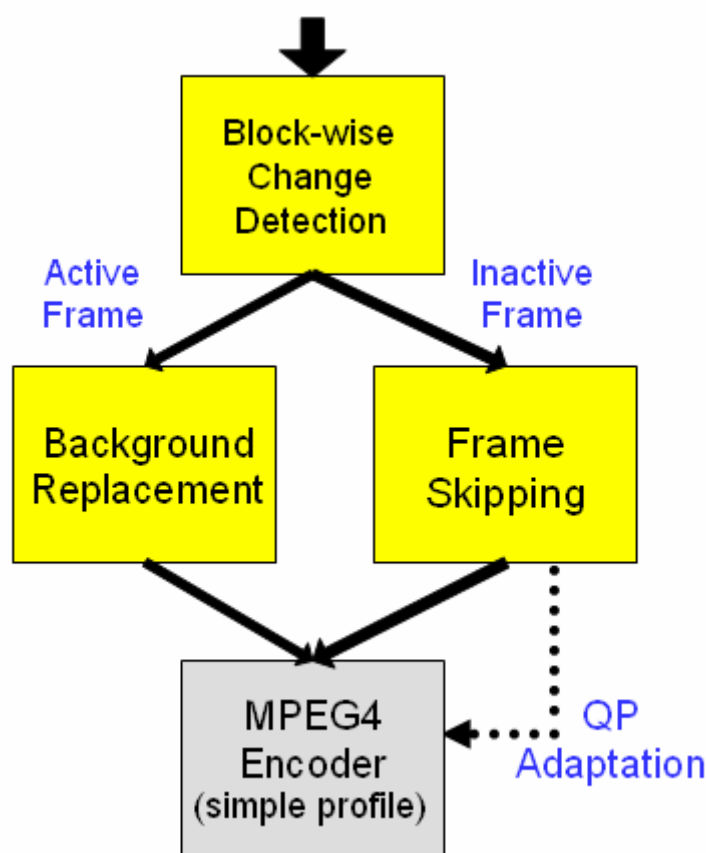


圖 4.2 壓縮(Compression)系統方塊圖

4.3 變動的判斷與雜訊的分析

在判斷變動方面，爲了不影響壓縮速度，我們採用了十分簡單又有效的方法，Frame difference。我們把前後兩張畫面影像中同位置像素點上面的亮度值拿來相減，當畫面沒有運動的情況之下，相減出來的值應該爲 0，而當畫面在運動的情形下，對應的像素點相減出來的值應該不爲 0，藉此來判斷畫面中是否有物

體運動。但不幸的，影像訊號經過攝影機輸入、訊號線傳輸、擷取卡的擷取等等，都會產生一些誤差，使得就算畫面中沒有物體在運動的情形之下，前後像素點的亮度相減之後也常常不為 0，針對這個問題，我們統計了這個雜訊的統計特性，藉由對雜訊的分析找出適當的方法來判斷是否影像內部有物體在運動。

4.3.1 雜訊的分析

下圖為在畫面不動的情形下，所拍攝出來的影像與其亮度值的差異量，我們可以發現，這個雜訊的分配是不均勻的，與影像內容有關，其中可以看出深藍色的屏風與棕色的桌面部分雜訊比較強烈，而相對於其他的地方，雜訊分佈的就顯得比較均勻。

由於雜訊與影像內容有關，因此若仍然是用整張影像來看的話，勢必沒有辦法有效地統計雜訊的特性，所以我們把偵測變動的機制改成一塊一塊的區塊來判斷。根據我們影像的內容，我們把它均分為均等的九塊，並且利用這九塊區塊前後張影像亮度值的差量總合，算出其每一塊的平均值與標準差，並且以各個區塊的平均值加上三倍標準差來當作各個區塊雜訊值的門檻，當我們區塊的中各像素點灰階值的差量的和 SAD(Sum of absolute difference)小於此門檻時，代表這兩張影像中，此區塊並沒有運動的資訊，反之則為有。

$$SAD = \sum_x \sum_y |f_t(x, y) - f_{t-1}(x, y)| \quad T = \text{mean}(SAD) + 3 * \text{std}(SAD) \quad (4.1)$$

其中 $f(x, y)$ 代表位置點為 (x, y) 上的灰階資訊， T 為門檻值，利用時間軸上記錄同一區塊的 SAD 值，並且對其統計其特性，算出平均值與標準差，利用平均值加上三倍標準差來當做此區域的門檻值。

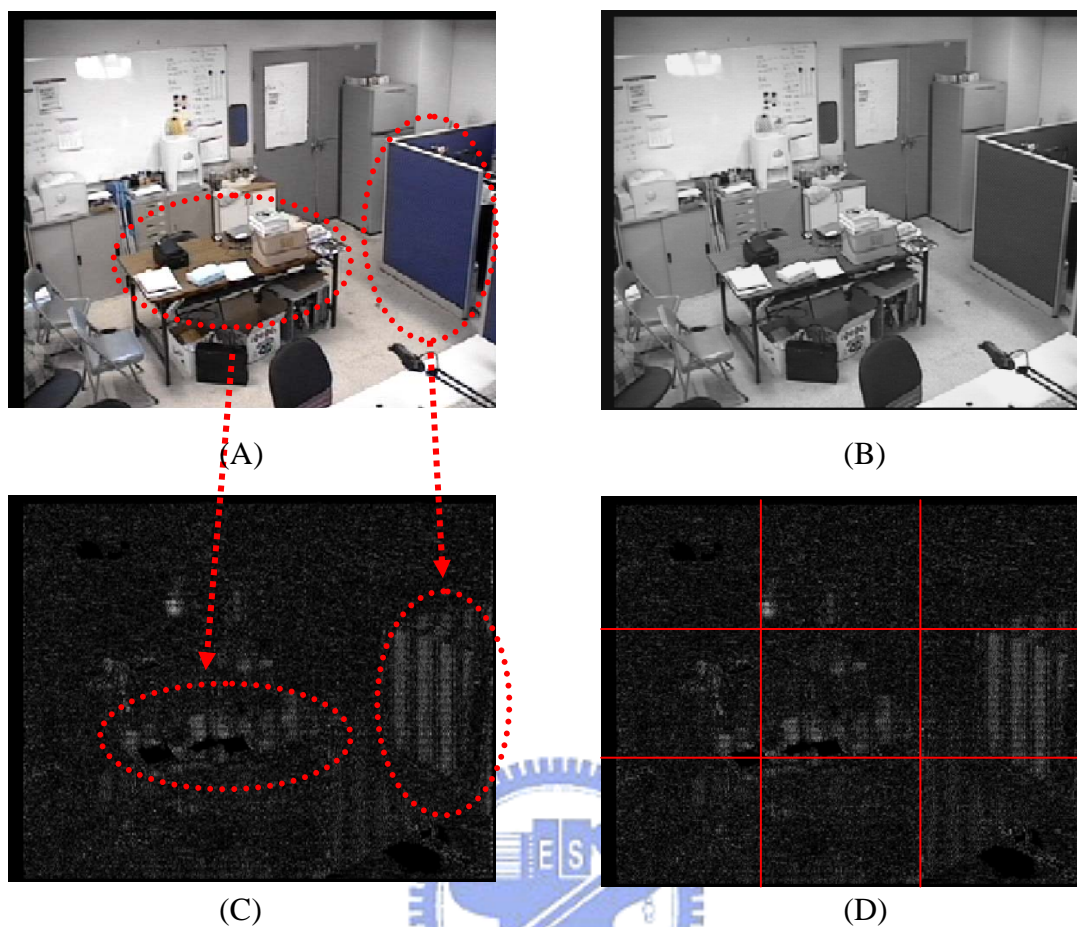


圖 4.3 (A)圖為監控攝影的場景，(B)為監控場景的灰階影像，(C)則為前後兩張靜止影像灰階資訊相減後的差量值，此差量值被放大十倍後秀出，圖片中越亮代表差量越大，(D)圖把影像切割成 3*3 等分的九個區塊。

4.3.2 雜訊門檻的初始設定

我們知道雜訊的門檻必須是是利用靜止狀態時影像差量的統計特性來判斷，因此，在一開始先對希望拍攝的影像背景，拍攝一段靜止的時間，測量其靜止狀態時雜訊的統計特性，分別以九個區塊來算出其中的均值和標準差，而由於背景有可能會隨著時間而變動，可能是光影的變化或是物體的運動，造成此門檻值並不能代表所有的情形，所以我們的初始門檻設定會比算出來的均值和標準差來得高的數值來設為初始的平均數和變異數，而且當判定整張影像為靜止狀態時，會重新根據現有影像算出另一組的平均數和變異數值，來當成現有影像的門檻值，運算的方法將會在下節提到。

4.4 Frame Skip 機制

當我們利用分區來統計雜訊特性之後，接著希望能夠當整張影像完全沒有運動資訊的時候，能夠把整張影像略過，利用前一張影像的資訊直接補進這一張影像即可，這麼一來就可以大大地增加壓縮率。

上一節我們提到了由於雜訊分佈的不均勻，因此我們把整張影像切割成 3x3 一共 9 塊區塊，而判斷影像是否有變動則是利用每一個區塊是否有變動來看，當所有的 9 個區塊都沒有變動時，我們就判定這張影像與前一張影像是沒有變動的，有的只是雜訊的擾動，因此我們就會把這張影像給省略掉。

爲了重播時的同步，當我們判定影像有屬於變動狀態而丟進 MPEG4 壓縮之時，也會同步記錄在這張影像之前有幾張影像被省略掉，而且把這個影像省略張數的資訊記錄在額外的檔案裡面，當要做重播動作時，再把資訊調出來做同步的動作。

爲了方便對影像做分析，我們另外產生一個暫存器，這個暫存器稱做“影像省略張數暫存器”，大小爲 10 個，用來儲存之前的影像省略的張數，如下圖所示

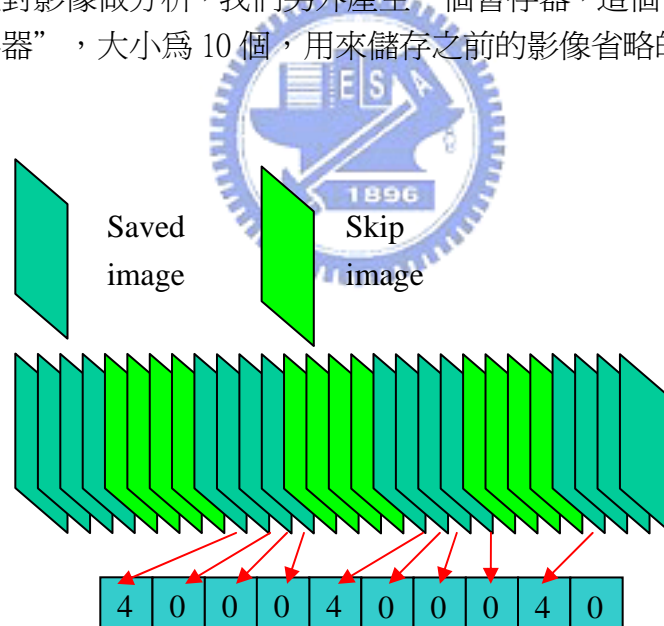


圖 4.4 影像省略張數暫存器與影像對應關係

這麼一來就可以很容易發現影像的特性，我們把影像區分爲四大類，分別是一直完全沒有變動的狀態；第二個是在一段時間沒有變動的狀態下，突然有物體的變動；第三個則是在頻頻變動的狀態下，突然沒有變動的產生；最後則是從頭到尾一直有物體在運動。下面爲四種狀態下對應的圖形

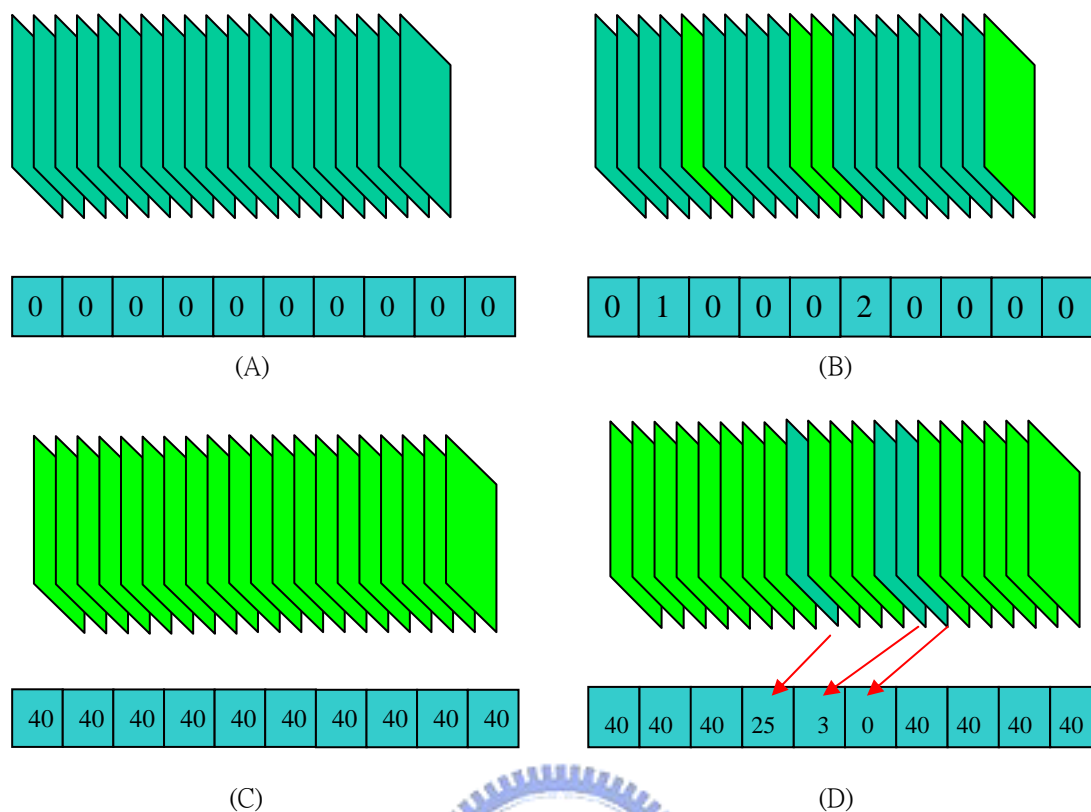


圖 4.5 畫面中可能對應的四種影像與其影像省略張數暫存器對應關係。(A)圖為一直判定為變動的影像，(B)圖為一直判定為變動後判定為靜止的情形，(C)圖為一直判定為靜止的影像，(D)圖為一直判定為靜止後判定為變動的情形

我們利用了分析場景的統計特性來做為判斷影像是否有變動發生，但是假如此門檻設得太過偏高時，會誤判為影像一直沒有在變動的狀態；而另一方面，當門檻設得太低時，會造成誤判為影像一直處在變動的狀態。對於門檻設得太高的情況而言，我們設計了兩個對應機制來處理，第一個是強制每隔幾張必定會儲存一張的機制，其對應的參數稱為“最大影像省略數值”，而第二個方式則是動態地調整各個門檻值，利用靜止影像時不用記錄影像多餘的時間，統計此時影像的雜訊特性，以更新原本的門檻值，關於此機制將會在 4.7 節中加以進一步的討論。

4.4.1 最大影像省略數值

當影像一直被判定在動卻突然間判定為不動的情形發生時，極有可能是變動得極為微小以至於被機制誤判為沒有變動，因此，我們希望能夠每隔一小段時間就去記錄影像畫面中的差值。另一方面，當場景一直被判定為沒有變動的狀態時，也能希望一次可以省略多一些畫面，因此我們將會根據場景畫面的特性來調整此一參數。

起始的最大影像省略數值設定為 10，在每秒鐘取像 30 張影像的攝影機來

看，10 張影像所代表的只是三分之一秒鐘，這也就是說當我們發覺現在的影像雖然判定為靜止影像的情況下，我們還會經過每三分之一秒記錄其差量，來避免誤判所帶來的遺憾。

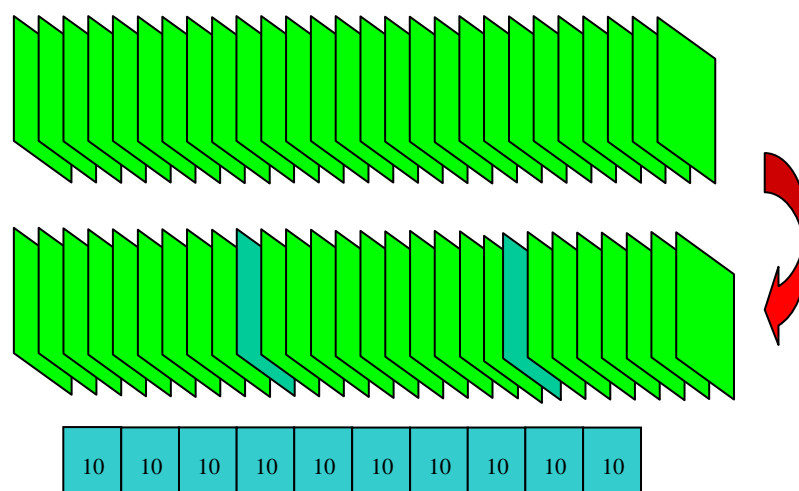


圖 4.6 當最大影像省略張數設為 10 而且畫面一直靜止時所儲存的影像

當我們判定為有變動狀態時，這個時候有可能是真的發生物體的變動，但是也有可能只是雜訊突然地加大，為了判定此一情況，我們分析剛剛提到的“影像省略張數暫存器”裡面所儲存的數值，來幫助我們判定是為畫面真正有變動或是只是存在較強烈的雜訊。我們利用在 10 個數值裡面至少有 3 個數值小於 5 來判定，也就是說在前面儲存的影像資訊中，至少要有三個影像之前的省略張數小於 5，也就是說在短時間六分之一秒內至少要被判定為變動的情況要有三次，當此條件成立時，代表我們判定畫面中真正有物體在運動，而當條件不滿足時，我們判定畫面中變動的資訊並不充分。當判斷為真的物體在變動狀態時，我們將會把最大影像省略數值設為起始的數值，也就是為 10，強迫我們至少三分之一秒要記錄影像的資訊。

若是我們一直判斷畫面為靜止狀態時，表示這個靜止狀態應該是很可靠的，這時我們不希望僅僅是每三分之一秒就強迫要記錄其差量，因此當全部影像省略張數暫存器裡面所儲存的數值都相等於最大影像省略數值時，我們把最大省略數值放大為兩倍，隨著時間的經過，當最大省略數值超過一千時我們設定最多每一千張一定強迫去記錄一張數值，約每 33 秒存一張影像，而以十分鐘的影像而言，我們的系統一共只會存 20 張左右，而在 I Frame 與 P Frame 個數比例設為 1 比 29 的系統內，僅會儲存到一張 I frame。下圖為此機制的 Flow Chart：

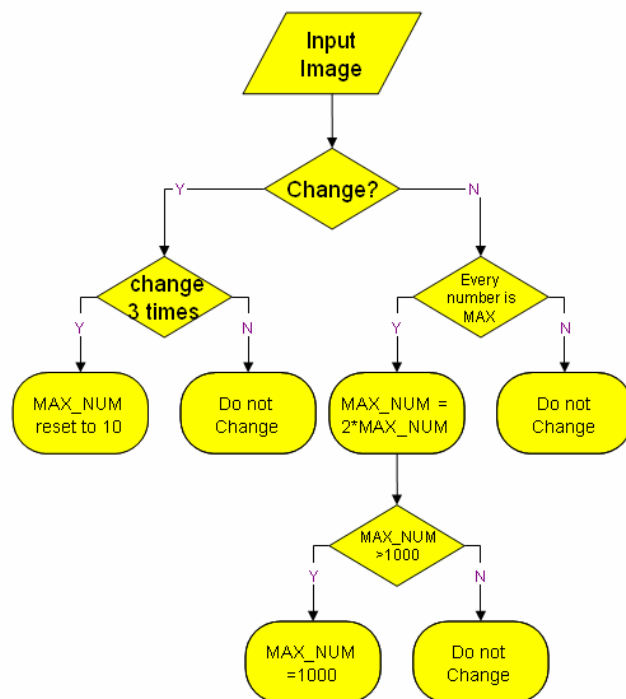


圖 4.7 最大影像省略張數調整流程圖

上圖中，輸入影像一開始利用區塊的 SAD 來判斷是否所有區塊皆沒有變動，接下來若是有變動的話，則判斷儲存的影像省略張數中是否在十個數字內有三個以上小於 5，若成立代表畫面變動的機曾很大，這時將重新設定最大影像省略數值為 10，也就是每秒強迫最少記錄三張影像。另一方面，當判斷為沒有變動時，則查看影像省略張數中，十個數字是否全部都等於最大影像省略數值，若成立，則代表畫面一直沒有變動的機曾很大，此時會把最大影像省略數值放大兩倍，然而若是放大兩倍後的最大影像省略數值大於上限值 1000 時，則將其值設定為 1000。

4.5 QP 微調

另一方面，當畫面被判斷為一直處於靜止狀態時，這個時候所記錄的影像資訊十分有可能只是雜訊，因此我們會進一步地希望再去對這雜訊做調整，把 Inter QP 調大，這樣在儲存 P frame 時，所儲存的差量將會被量化地更加粗略，讓資訊量降低。而對於一直沒在動的影像而言，這種影像幾乎是我們完全不需要去注意的，因此我們也會調整 Intra QP 的值，把畫面調整成可以接受的畫質即可，進一步地連 I frame 影像的資訊量也降低。值得注意的是，當調整 Intra QP 時，連帶的會影響到畫面的品質，所以只有當我們十分確定畫面為靜止影像而且是十分不需

要去注意的時候，才會去調動 Intra QP 的數值。

這部份我們的做法和調整最大影像省略數值相近，當暫存器內 10 個數值裡面至少有 3 個數值小於 5 時，判定為影像是屬於變動的狀態，因此我們會還原 Intra QP 以及 inter QP 為起始值 4，不同的是，只有當最大影像省略數值調為大於 300 時，Inter QP 才被設為 31；而當最大影像省略數值調為 1000 的同時，才會把 Intra QP 加大到 8。而當要調整最大影像省略數值為 1000 之前，最少要連續判定影像為靜止狀態 3.5 分鐘的時間。

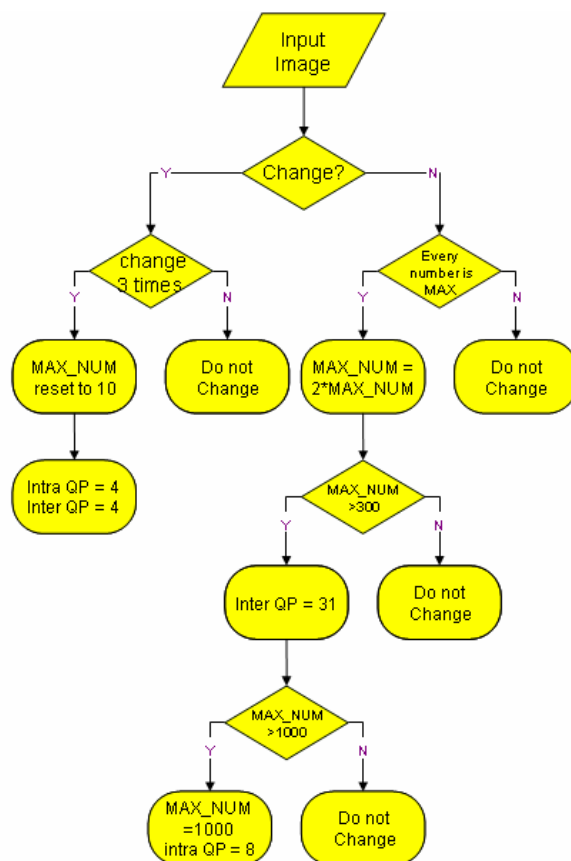


圖 4.8 最大影像省略張數調整流程圖與 QP 改動流程

圖 4.7 記錄了如何改動 QP 的流程，配合之前調整最大影像省略張數的架構，再判斷為合理的靜態影像時，先後改動 Inter QP 以及 Intra QP 的數值，當 Inter QP 改為 31 的時候，強迫記錄下來的影像雜訊的差量將會被量化地更小，以提高壓縮率。而改動 Intra QP 是當已經判定畫面為靜態影像很久了之後，確定畫面不是我們所想要記錄的資訊，因此調升 Intra QP 為可以接受的品質即可，這樣可以進一步地連帶把 I frame 的影像資訊更進一步地壓縮。

當最大影像省略數值調整為 1000 時，當影像一直判定為不動的情形下，十分鐘的影像資訊將會被儲存大小為 12KByte 的資訊，其中僅包含了一張 I frame 來當成整段影像的起始畫面。

4.6 區塊的變化(Block-Change)機制

討論完靜止影像部分之後，接著來考慮判定為變動狀態的畫面，這個時候代表至少有一個區塊被判定為有變動的狀態，當變動區域不大時，常常只有一兩個區塊被判定是有變動的狀態，而其他的區塊則仍然是判定為靜止狀態。當這個情形發生時，由於 MPEG4 simple profile 一次記錄畫面是以一張影像為單位，不可能把畫面切割成只記錄變動的部分，但若只因為一小部分的變動而記錄整張畫面的差量的話，會把靜止狀態下的雜訊也一並記錄起來，關於這個部分的資訊則是我們完全不想要的，因此，在判定完有變動的區塊後，我們只更新有變動區塊內與其附近的值，而不是整張影像的畫面都更新，這樣一來可以減少背景的雜訊所帶來的資料量。

下圖為其示意圖，當我們判定中間區塊有變動產生時，這時我們會把中間區塊以及附近的資料重新更新。

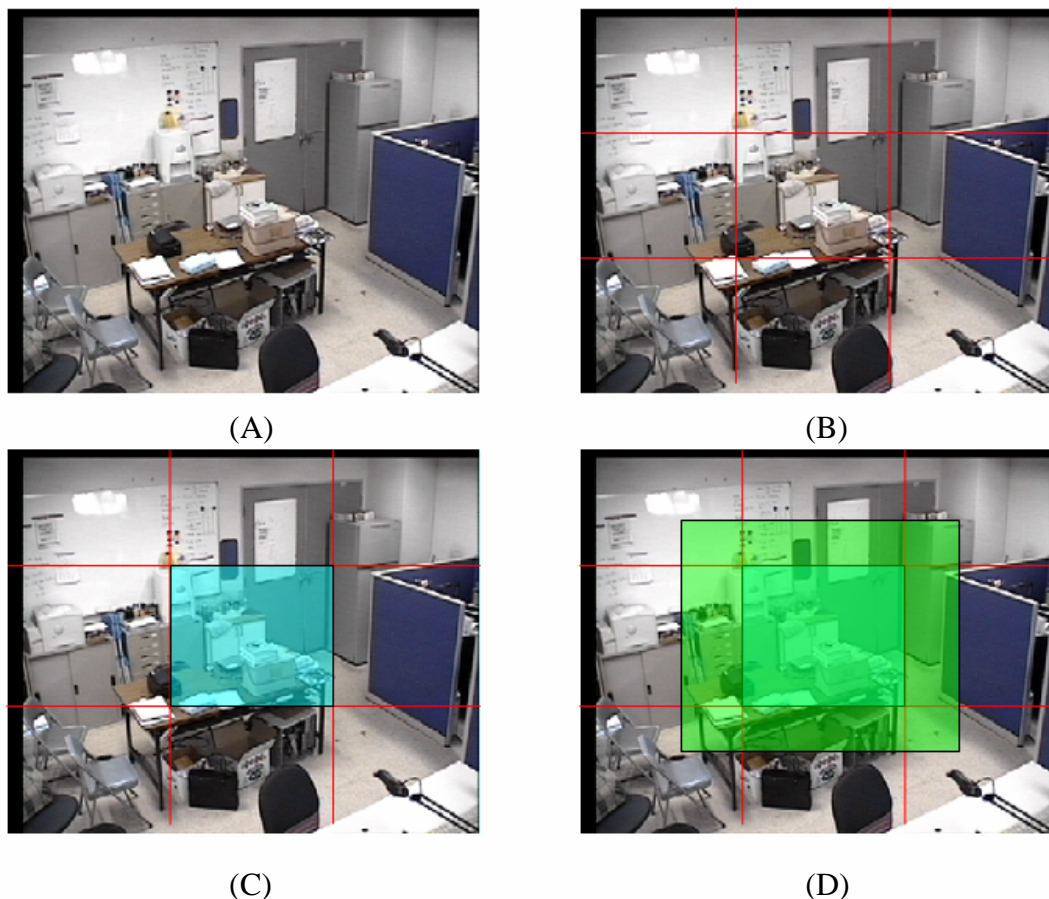


圖 4.9 Block-Change 示意圖，(A)圖為原始影像，(B)把原始影像分為九個相等的區塊，(C)圖判定中間區塊有變動產生時的情形，(D)圖僅對中間區塊及其附近的畫面調整其影像內容，其餘部分則拷貝之前影像的內容

當我們使用 Block-Change 機制時，由於其他區塊因為判定沒有變動產生，所以產生的資訊是來自於雜訊的部分，而針對這些部分，直接拿前面記錄的影像畫面，拷貝過來就可以了，這麼一來，當記錄影像為 P frame 的時候，由於前後兩張畫面完全一樣，所以靜止部分的資料量為 0，假如沒有做這件事情的話，則會把雜訊的部分給記錄下來。

4.7 動態調整門檻機制

我們利用了分析場景的統計特性來做為判斷影像是否有變動發生，但是假如此門檻設的不夠正確，會產生一些不良的後果，當門檻設定太高時，造成常常在靜止畫面內，而當門檻設定太低時，造成常常判定有畫面在變動。

當我們判定為靜止狀態而省略張數還沒到最大時，這個時候並不會進入 MPEG4 壓縮機制，因此可以利用此時多餘的運算時間拿來重新計算場景的雜訊模型，更進一步的貼近我們希望的門檻值，因此我們初始的門檻值會設定地比較大，之後再慢慢調為適應於此場景的門檻值，以下為此設計機制：

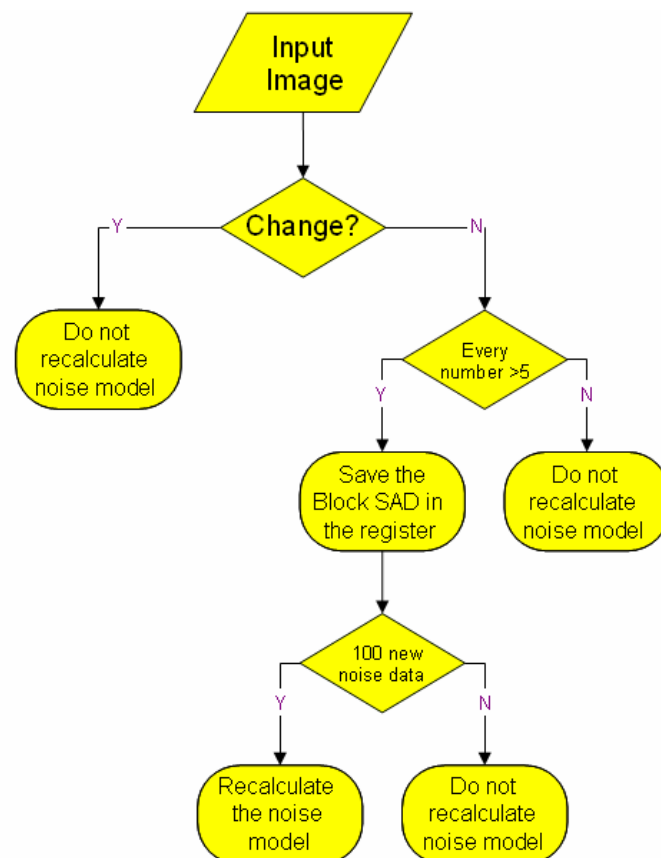


圖 4.10 動態調整雜訊門檻機制流程圖

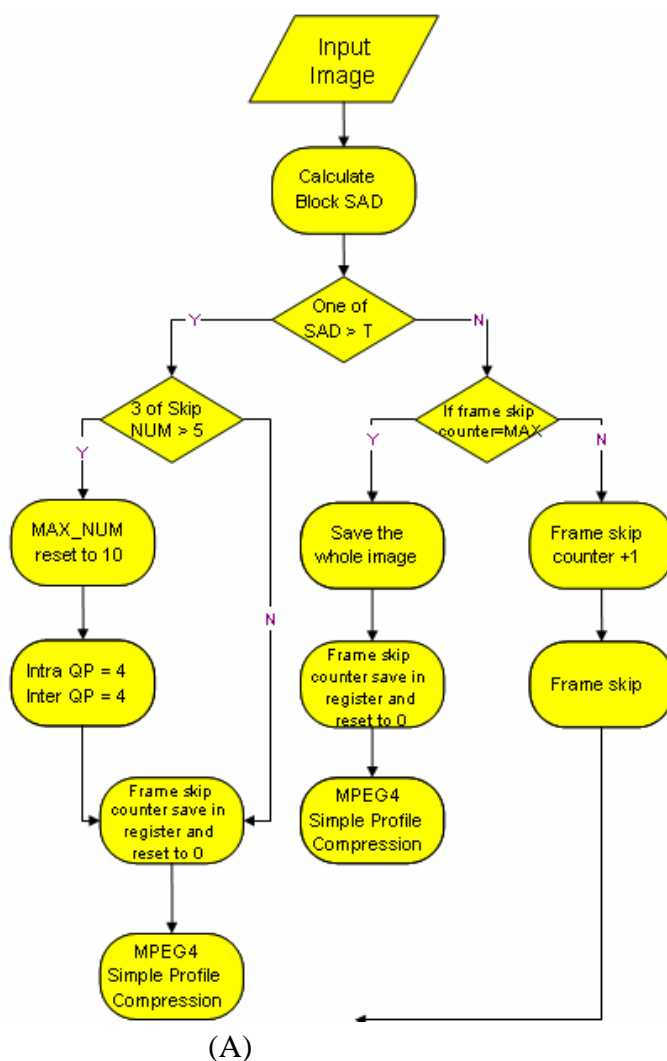
當我們判定為可靠的靜止影像，也就是所有的影像省略數值都大於 5，這個時候開始記錄各個區塊的 SAD 值，每當記錄一百個值之後，就會重新算出各個

區塊的平均值和標準差，並以各塊的平均值加上三倍標準差拿來做為我們新的門檻值。

當一開始的起始門檻設得太低的時候，會導致於就算畫面一直沒有動，也會判定為有影像在變動，因而丟進 MPEG4 Simple Profile 內壓縮儲存，完全沒有多餘的時間去運算雜訊的模型。另一方面，當一開始的起始門檻設得太高的時候，會把許多微小的變動忽略掉，把它當成是雜訊所帶來的影響，以至於算出來的雜訊模型會偏高，為了以防萬一，我們會限定平均值和標準差的上下限，當超出這個可接受範圍時，會重新設定為初始值，這麼一來，所算出來的門檻才會逐漸接近真正的雜訊模型。

4.8 整體系統架構

整個壓縮的機制如下，



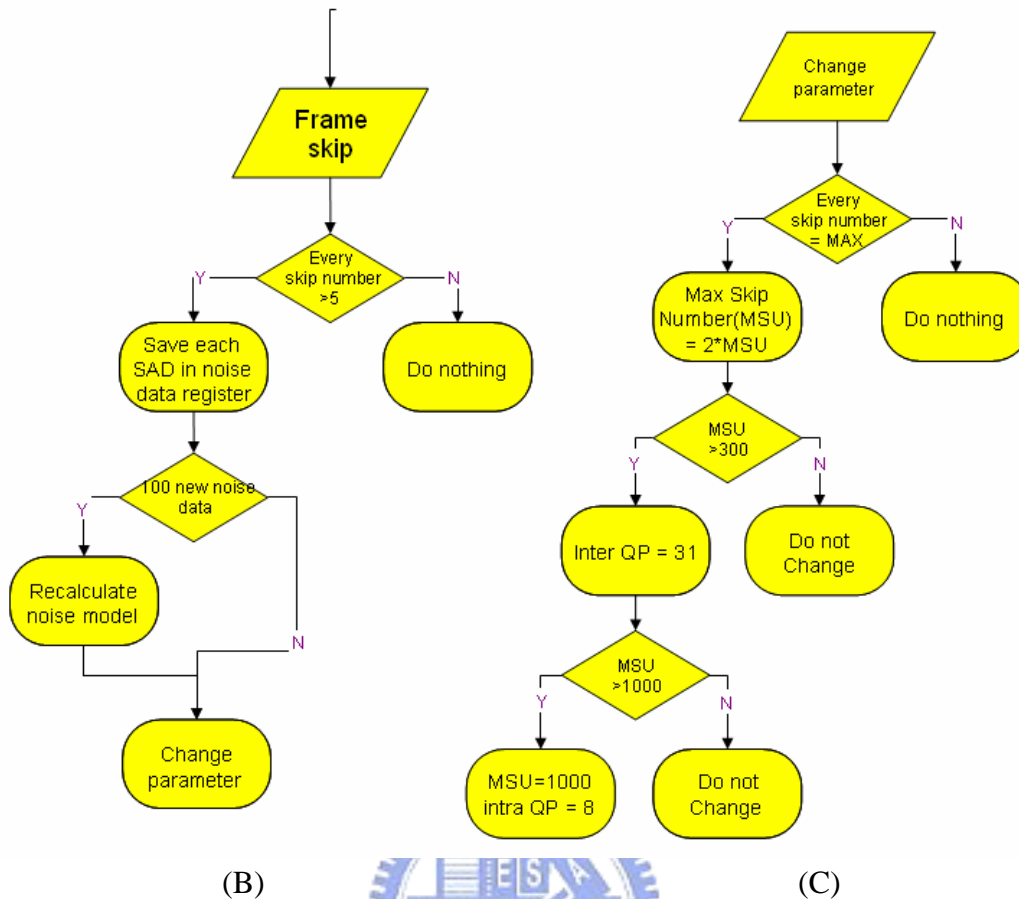


圖 4.11 整個系統動態調整參數機制，(A)圖為判別影像是否為變動狀態的機制，(B)圖則是從(A)圖中承接 Frame skip 機制，(B)圖內部為判斷影像是否為背景資訊，若為背景則啟動從新調整門檻機制，(C)圖是承接(B)圖的改動參數，(C)圖中主要是判斷是否要對最大影像省略數值以及 QP 做調整。

一開始把影像畫面的輸入等分為 9 個區塊，並且以區塊為單位進一步地算出前一張影像與現在這張影像的 SAD 值，並把此 SAD 拿來與區塊的門檻值相比，若全部區塊都被判定為靜止狀態時，判定整張影像為靜止狀態，若影像省略張數還未到最大值，把影像省略張數加一，並且進一步丟入 QP 和 MSU(最大影像省略數值)的判斷式去，而只要一個區塊判定為變動狀態時，就丟入 Block-Change 機制，分析影像省略暫存器，若是判定場景為變動狀態(三個影像省略張數小於 5)，這時將會重新設定 MSU 和 QP(MSU=10、inter QP=4、intra QP=4)，之後再進一步丟入 MPEG4 裡面去進行壓縮，而當全部區塊皆判定為靜止狀態但是影像省略張數已經相等於最大值時，此時強迫所有的影像都要記錄，再丟入 MPEG4 裡面進行壓縮的動作。

4.9 小結

經過這三重壓縮過後，十分鐘的影像 18000 張 frame 從 YUV420 的原始影像大小 2.7GB 壓縮最小到 12KB，若是只用 MPEG4 壓縮大約僅能壓縮到 30MB，而每一個壓縮所產生的功用如下表所示

壓縮方式	Frame Skip	Block-Change	QP adaptation
針對影像	靜態影像	動態影像	靜態影像
改善資料	I and P frame	P frame	I and P frame
壓縮倍率	最大可到 1000 倍	示影像雜訊大小以及影像變動範圍，對每張 P frame 而言若一半影像判定不動約省 0.3KByte	Inter QP = 31，每張 P frame 約省 0.6 KByte 而 Intra QP = 8 約可省一半的 I frame 資訊量

表 3.1 壓縮方式與效率比較

可以看到我們根據影像的內容，記錄我們需要的資訊，並且把冗餘的資訊更進一步的排除，達到更加的壓縮效率。

第五章 去交錯(De-interlacing)

本章說明本篇論文提出去交錯的方法，從基本概念，推展至實作上的重點。

5.1 概念

當傳統的交錯式影像格式的影像拿到漸進式掃描裝置上播放的時候，若是不經過去交錯的方式把交錯式訊號轉為漸進式訊號，畫面在有變動的地方會有很多缺陷發生，而去交錯的方式，也都有經過很多人的討論和分析，而接下來的章節將會一一做分析與討論。

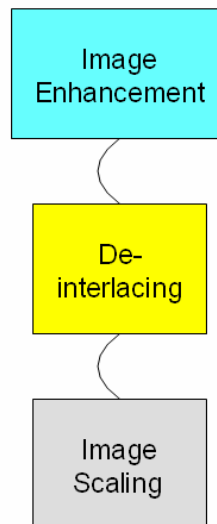


圖 5.1 影像加強部分區塊圖，本章所要討論的屬於其中的去交錯部分

5.2 測試環境

由於沒有客觀的評估環境，我們採用的是一般測試影像中比較常用 Foreman sequence 來作為我們測試的平台，由於 Foreman sequence 為漸進式的影像資訊，所以我們要設法模擬出交錯式影像的資訊，而當只有利用到一個 field 的資訊來作為去交錯的原始資料時，這個時候的模擬方式相對地簡單，把 sequence 內所有的 bottom field 丟掉，僅利用 top field 的資訊來還原整張影像。而當要用到前後張時間域上面的資訊時，我們的做法是採取二分之一取樣的方式，也就是奇數張的

影像丟掉 top field，而偶數張影像丟掉 bottom field，最後則以 top field 為主，還原出整張影像的資訊，這樣一來可以當做每隔三十分之一秒影像擷取一個 field 的資訊，每秒更新 15 張影像，真正的攝影機為每六十分之一秒擷取一個 field，每秒更新 30 張影像，在取樣頻率上少掉一半，但這樣子已經大致可以表現出交錯示訊號所會面臨的問題與處理的方法。



Frame 89-90



Frame 91-92



Frame 93-94



Frame 95-96

圖 5.2 利用 Foreman sequence 前後張錯開的影像模擬交錯式訊號

5.3 一個 field 去交錯

由於我們是要把 de-interlacing 的機制建構於即時影像的監控上面，所以運算複雜度為其中很重要的考慮因素，若是選用太複雜的機制如 MC-IPC，雖然效果可能會很好，但是畫面因此而沒有辦法達到即時播放的話反而失去了它的意義，因此我們希望可以結合幾種簡單的方法，並且分析他們的優缺點，在不會增加大量的運算複雜度之下，能夠得到比較好的效果。

因此，對於一個 field 的去交錯方式時，我們模擬了各式各樣不同但簡單的

方法，包含 LD(Line Doubling)、LA(Line Average)、ELA(Edge Based Line Average)、Median、Mean 等五種方法，並且與原始影像做 PSNR 的比較，下圖則為 Foreman sequence 中，五種方法的 PSNR 的比較圖

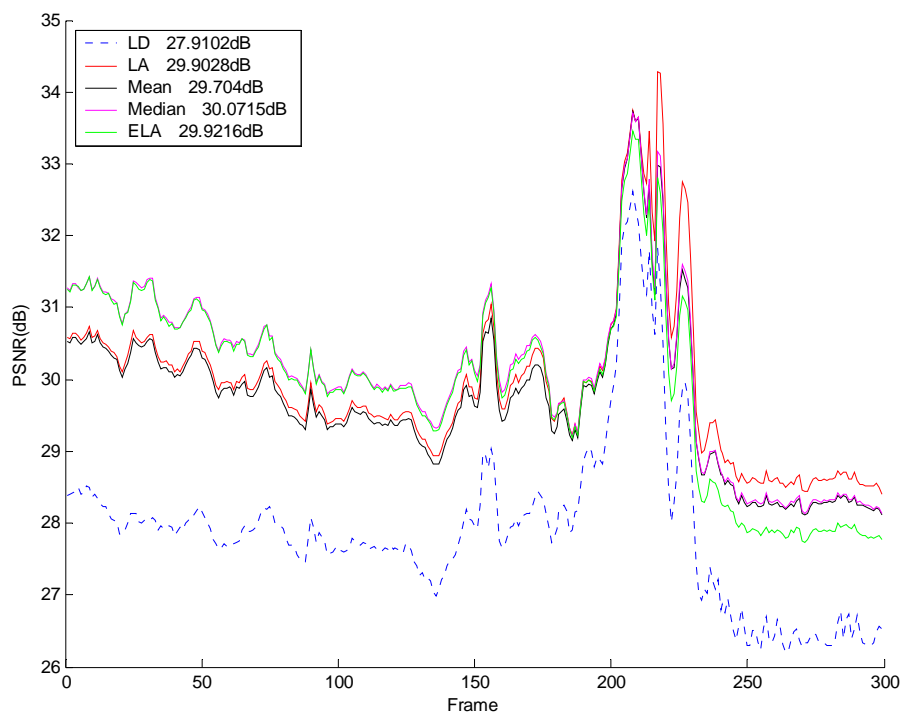


圖 5.3 五種方法在 Foreman sequence 內 PSNR 的比較圖

由圖 5.3 可以發現到在 foreman 前半段時候(在 frame 200 之前)，Median 方法與 ELA 方法以 PSNR 來看是五個方法裡面最高的，而在 foreman 後半段時(再 frame 230 以後)，LA 卻是 PSNR 裡面最好的，而相對於之前 PSNR 比較高的 Median 和 ELA 的方法，LA 最高，Median 次之，ELA 為三個裡面最差的。LD 的方法雖然是最簡單，但是也是 PSNR 最低個一個。接著我們來分析一下各個方法的優缺點以及實際上利用一個 field 去交錯方式後所產生的圖形。

- LD → 很明顯地，由於僅僅把 top field 原封不動地拷貝給 bottom field，雖然有十分高的銳利度，但是畫面會產生嚴重的 aliasing 影響，而且在斜邊的地方補成一格一格的，十分破壞畫面品質。不過它是所有方法中最簡單的一個。
- LA → Line Average 方法是利用上下兩個點的資訊，取平均後，補出中間的點，由於上下兩個點是所有的 top field 內離待補點最接近的點，所以補出來的 PSNR 值會非常高。但是，也由於它僅僅是利用上下兩的點的資訊來補，所以他完全沒有考慮到斜邊的影像，像是 Foreman sequence 中，背景房屋的部分用 LA 補出來的值往往就不是我們想要的

數值，雖然它有很高的 PSNR 但是視覺效果會被大打折扣，所以，一般不會直接採用 LA 去還原出整張漸進式影像。

- Mean → 這個方法是利用附近六個點的資訊來補出中間點待補點的資訊，我所採用的方式是全部都是等比重的方法，這可以說是考慮比較多點的情況下，利用平均補出中間點，在六個參考點皆為同一個區域時，也就是待補點在平滑區的時候，參考六個點來補可以讓整體的平滑性更進一步。但是當六個點分屬不同區域時，這個方法依然是等比重的做平均，因此很有可能補出不正確的資訊，使得 PSNR 降低，如圖形中房屋的邊緣部份。
- Median → 這個方法首先先要排序，把附近的六個點由高到低排列，之後再取中間兩個點的資訊來平均，算出待補點的資訊。雖然它依舊是沒有參考到邊界的資訊，但是卻會無形中把屬於點數比較少的資訊排除掉，進一步地判斷出適合的影像還原值。
- ELA → 這個方法，主要就是參考邊界的資訊，利用三個方向上相減的大小值，選取最小的一邊來當成兩邊都是同一個區域內的值，因此中間的待補點可以利用這個邊上兩點的資訊來內插出中間的資訊。在區塊比較大，而且斜向 Edge 常而且明顯時，它能夠內插出不錯的效果。

下面為五種方法僅以一個 field 去交錯後所得到的圖形



原始 foreman 影像的灰階資訊



左上角放大四倍的圖形



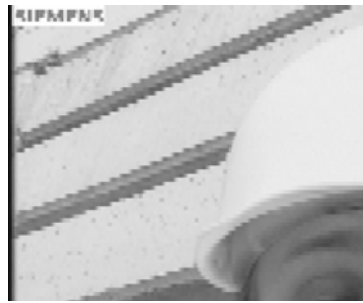
去掉 bottom field 的圖



左圖左上角放大四倍



LD 去交錯後的圖



左圖左上角放大四倍



LA 去交錯後的圖



左圖左上角放大四倍



Mean 去交錯後的圖



左圖左上角放大四倍



Median 去交錯後的圖



左圖左上角放大四倍



ELA 去交錯後的圖



左圖左上角放大四倍

圖 5.4 此圖為 Foreman sequence 第一張圖形，左圖為原始影像與去掉 bottom field 影像以及五種方法去交錯後的圖形，右圖為左圖的左上角部分放大圖

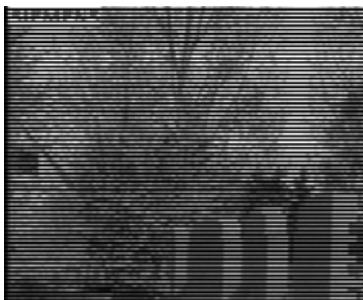
由圖 5.4 首先分析一下在左上角的斜邊邊界部分，由於各種方式的特性，所以不難發現到在屋簷部分 ELA 與 Median 的方式所產生的影像比較接近於原始影像的屋簷部分，而 LD、LA 與 Mean 在屋簷部分的鋸齒狀情形相較於 ELA 和 Median 來說嚴重很多，所以在前半段的部分，ELA 和 Median 的 PSNR 會比其他的方式來得好。



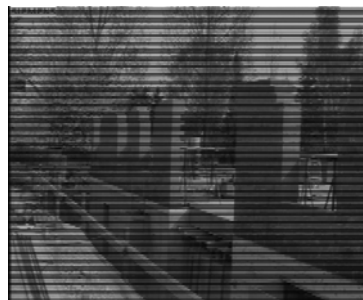
原始 foreman 影像的灰階資訊



左上角放大四倍的圖形



去掉 bottom field 的圖



左圖左上角放大四倍



ELA 去交錯後的圖



左圖左上角放大四倍



Median 去交錯後的圖



左圖左上角放大四倍



LA 去交錯後的圖



左圖左上角放大四倍

圖 5.5 此圖為 Foreman sequence 第 251 張圖形，左圖為原始影像與去掉 bottom field 影像以及三種方法 PSNR 較好的方式去交錯後的圖形，右圖為左圖的左上角部分放大圖。

接下來看一下 ELA、LA 與 Median 在後半段的圖形。從 PSNR 來看，在 Foreman 後半段(frame 230 以後)，LA 的 PSNR 最高，Median 其次，ELA 最慘。從去交錯後的圖形來看，由於在 Foreman 後半段，背景部分為十分複雜的樹葉和樹枝，而在數後面是一整片的天空，用 ELA 簡單的判別邊界方向性的結果會除了造成不少誤判外(比如說樹枝判成天空，天空判成樹枝)，也容易造成一個一個區塊狀個樹葉。再來是 Median，由於它是採用附近六個點的資訊取中位數來加以平均，所以當附近點的資訊量多的一方反而不屬於同一區塊時，也會造成誤判的情形。

最後是 LA，它雖然完全不管方向性，但它所拿的是上下兩個點的資訊，也就是離它最近的兩點的平均值來補，反而有時候比較容易找到正確的資訊

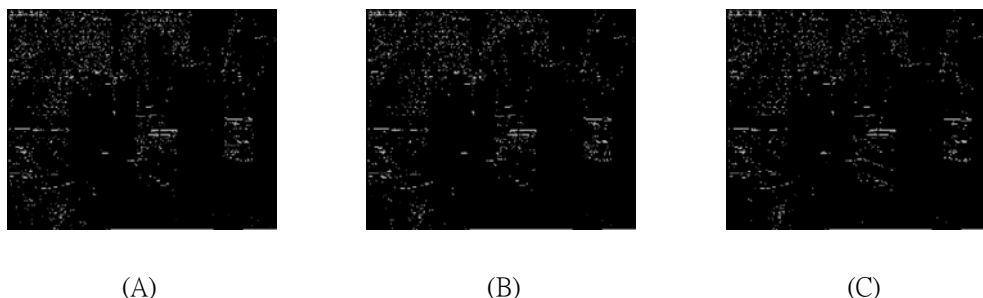


圖 5.6 此圖為 Foreman sequence 第 251 張圖形，分別經過(A)ELA、(B)Median 以及(C)LA，去交錯後和原始影像的比較圖，亮的部分表示差量大於 30 的像素點，黑的部分表示差量小於 30 的部分。

由圖 5.6 可以得知，LA 在 Foreman 後半段的補償值比 ELA 和 Median 來得好。當在平滑區的時候，當參考附近六個參考點皆屬於同一區塊時，LA 是利用最近兩點的資訊來補，這個本身就是極為可靠的資訊。而當在邊界時，若邊界的方向為垂直狀的，用 LA 補將極為適合，但是在有長條型斜向邊界時，LA 會造成鋸齒狀的波紋。ELA 是利用邊界的方向性，透過附近六個點三條線的方式，找尋最可靠的邊界方向，以此邊界的兩端點來做平均，當在平滑區時這個資訊不會錯，而當在長條型斜向邊界時，這個方法就可以發揮地很好，在很細的邊界上，如樹枝而背景又是一大片同樣顏色的天空時，這時候找到的資訊就有可能會被背景所拉走。Median 的情形與 ELA 差不多，比較好的情況是它是利用亮度值排序，只要有四個點同屬於同一個區塊，則內插出來的值就會在這個區塊內，而在普遍的情形中，這樣可以達到很好的補償方式。

在分析過後，發覺 LA 可以應付大部分的情形，只要上下兩點是屬於同一個區塊的時候，LA 可以補償出極好的效果，因此我們的所提出的方法主要就是希望發揮 LA 簡單有效的特點，並且把它的缺點補起來，首先，先判定上下兩點是否屬於同一個區塊，如果是的話則直接利用 LA 來做內插，如果發覺是不同區塊的話，則再利用 median 來做內插的動作，下面為此方式的示意圖：

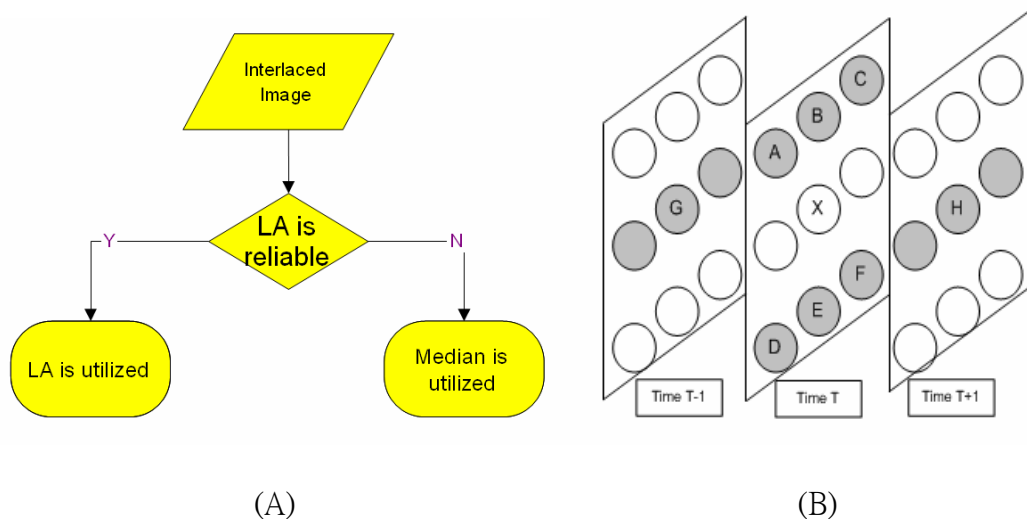


圖 5.7 去交錯時參考的資訊(A)圖為去交錯空間域上系統流程圖，(B)圖為對應的像素點的灰階值。

至於怎麼樣判別上下兩點是屬於同一個區塊，我們採用的方式如下

$$M = \max(A, B, C, D, E, F) - \min(A, B, C, D, E, F)$$

$$\text{If } (\text{abs}(B - E) < M / 2) \quad X = (B + E) / 2 \quad (5.1)$$

$$\text{else} \quad X = \text{median}(A, B, C, D, E, F)$$

當我們發覺上下兩個點差量比起全部六個參考點最大減最小值的一半來得小時，我們把它判定為上下兩點為同一個區塊中的兩點。之所以取這樣子的做法主要有兩個原因，第一個是先觀看附近六點的最大差量，越大代表附近點越有可能分屬於不同區塊，因此要小心處理。另一方面，當差量越大時可以分做兩個部分來看，一個是單純只分為兩個區塊的情形，這個情形下利用一半的最大差量可以很輕易判別上下點是否屬同一區塊，但另外的一個可能是影像畫面很複雜，這個時候可能六個點散佈在不同的六個區塊中，而我們以一半的最大差量來判斷，可以包含比較多種的可能，而且當使用 LA 做為去交錯的方式時，可以保持直線的完整性，而當上下點差異太大，大於一半的最大差量時，代表此時上下兩點極有可能為邊界的兩端，所以需要另外用可以找出邊界的方法來補。

從另外一個角度來想，因為門檻值的設定與六點的最大差異值相關，可以是做為錯誤容忍度會隨著附近點的變動大小而更改，假如真的把不同區塊誤判成同一區塊，也不會造成太大的損傷。並不會被像是 ELA 或是 median 的方式，有可能被其他區塊拉走，造成線的斷層。另外，因為上下兩點從距離來看最靠近中間點，因此這條線是十分有可能存在的，而當上下兩點判定分屬不同區塊的時候，我所採用的方式是利用 median 找出其中位數，當上下兩點不同塊時，點數越高的一方的確是越有可能是中間點所要補的區塊，就算六點分屬於不同的六個區塊時，我們也選最中間的兩個點的平均值來當做內插的值，這樣一來錯誤就會壓

小，不像 ELA 只是找最像的邊線來平均。因此，我們採用 median 來做為我們的後續處理方法。

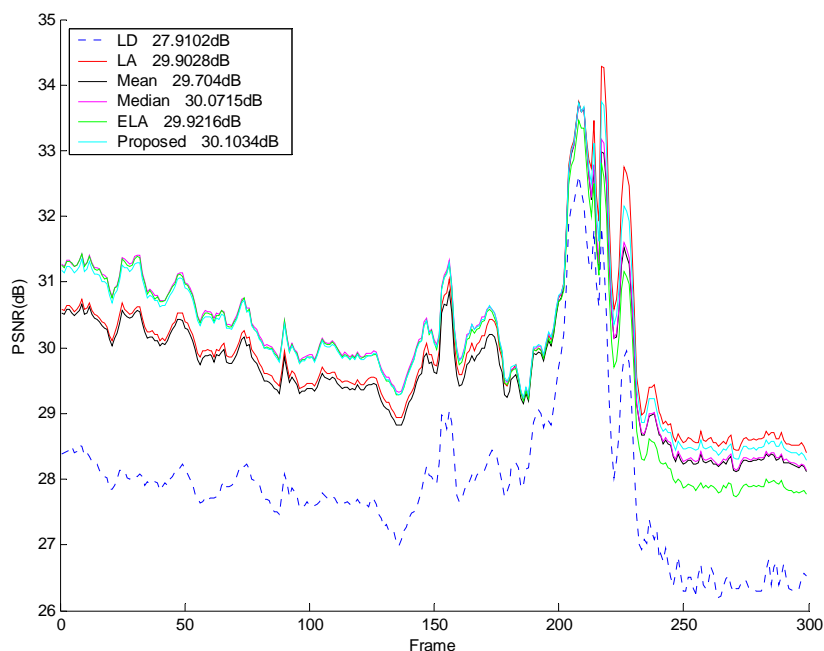


圖 5.8 各種去交錯方式的 PSNR 比較圖，其中 Proposed 為提出的方法。

從圖 5.8 中可以看到，我們所提出的方法，在前半段可以貼近 median 的 PSNR 值，而在後半段也遠比 Median 和 ELA 來的高，可以說是結合了 LA 和 Median 的優點，雖然後半段還比 LA 低一點，但是整體平均的 PSNR 值是所有方法中最高的。

5.4 時間域上提出的去交錯方式

討論完一個 field 去交錯方式後，接下來要討論的是時間域上面的資訊，我們採用的方式主要是根據 MA-IPC，也就是並沒有做 motion compensation 的動作，主要的原因就是希望能夠以簡單的方法達到一定的效果。而由於雜訊的干擾，所以再利用時間域上資訊去做去交錯時，額外提出了一些保護的動作。我們先把時間域的補償分做兩部分，第一部分當前後兩張的 bottom field 值相差不到雜訊的量時，我們判別可能是背景部分的資訊，所以可以利用前後兩個 field 來還原出中間 field 遺失的資訊。而當還原出來時，另外針對 noise 的大小還要判別是否這

個值是否可靠，假如是會產生如鋸齒狀的波紋的圖形，也要設法把它給排除掉，所以最後的流程圖如圖 5.8 所示

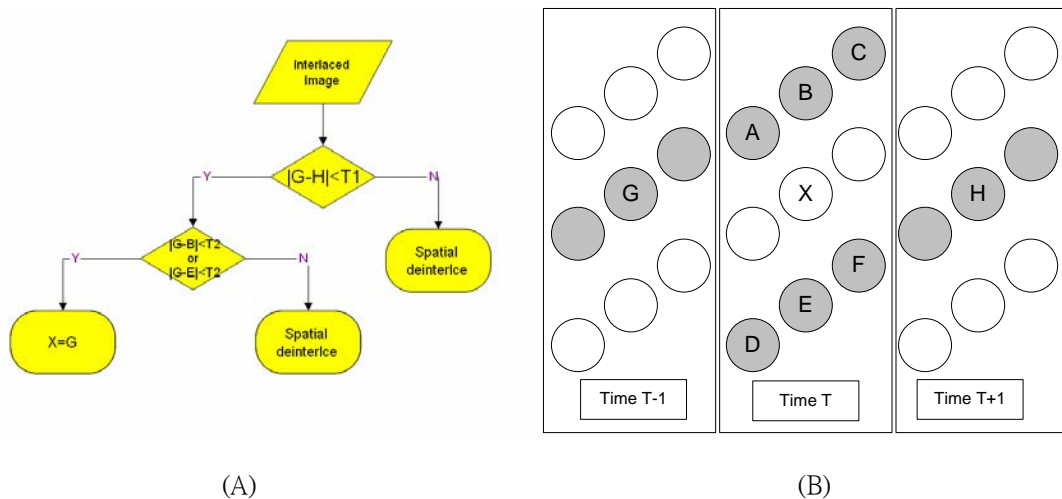


圖 5.9 去交錯時參考的資訊(A)圖為整體去交錯系統流程圖，(B)圖為對應的像素點的灰階值。

我們一開始先判斷影像資訊視為背景還是前景的資訊，這個可以利用簡單的 $G-H$ 來判斷。若 $G-H$ 小於雜訊的門檻值，我們可以判定 X 點很有可能是屬於背景的部分，這個時候直接用 G 和 H 的資訊來補出 X 是很合理的，一般 temporal deinterlacing 最簡單的方法是用 $(G+H)/2$ 的方式來補，而我們實際上在使用時，是直接補進 G 的值，也就是之前整張影像擷取時的資訊，並沒有再加入 H 。而另一方面，假如補出來的值不屬於上下區塊內的值的話，我們會認為這個值是有可能出錯的，倒不如直接用空間域上的補法。因為補錯一個點的資訊會造成影像上十分明顯的缺陷，因此在這方面我們利用 B 和 E 之間與補出來的值是否相似，也就是相差值是否為可容忍的雜訊，若差異太大，那就直接用空間域上的資訊去補。

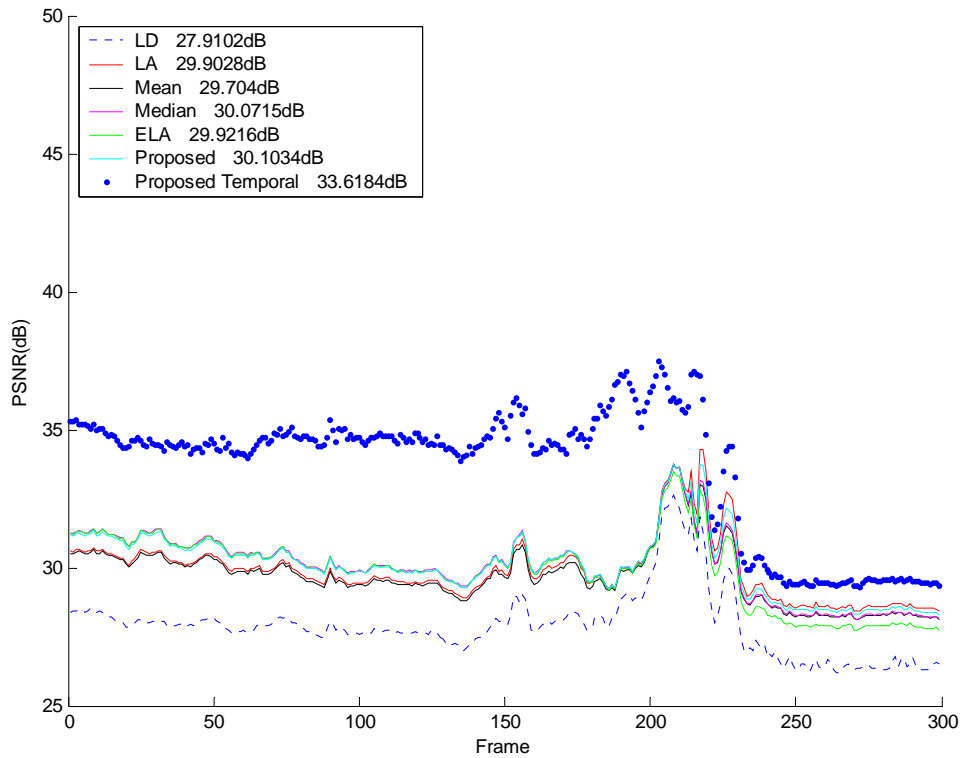


圖 5.10 Foreman sequence 七種去交錯方式的 PSNR 比較圖

上圖的 Proposed Temporal 即為時間域上與空間域上所提出的去交錯方法，可以看到與僅用空間域上資訊的方法相比，在前半段背景比較單純，運動不大的地方，PSNR 值可以提升 3dB 左右，而在變動比較劇烈的後半段，只有一點多 dB 的提升。上圖中兩個參數 T1, T2 分別設為 1 和 20，因為 Foreman 影像雜訊的影響比較小，而在實際的監視系統中，則是分析雜訊在來加以設定 T1 以及 T2。



(A)



(B)



(C)

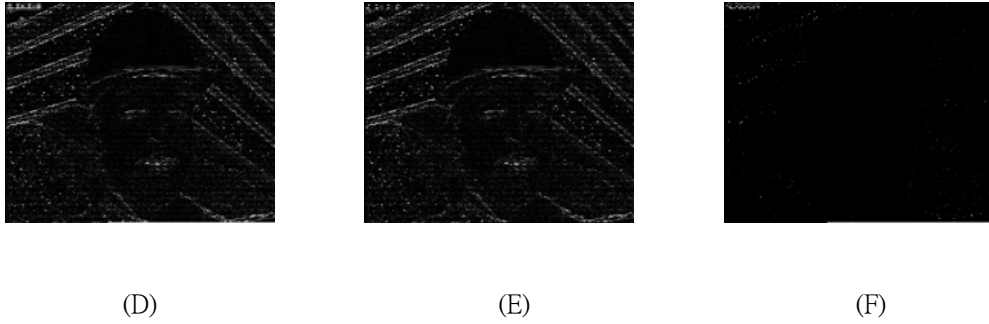


圖 5.11 (A)圖為 Foreman 原始影像，(B)圖為空間域上提出的去交錯方式所還原出的影像，(C)圖為空間域加上時間域的去交錯方式所還原出的影像，(D)圖為(A)與(B)圖差量*10 的影像資訊，(E)圖則為(A)與(C)圖差量*10 的影像資訊，(F)圖則是(D)、(E)兩圖間的差量*10 的影像資訊圖。

圖 5.11 中，可以看到僅利用空間域的方法與加入時間域的方法，在平滑區所補部分的差量其實不大，但在邊緣區域，像是屋簷附近，加入時間域的資訊能夠使得還原的效果更好，特別是在左上角的文字部分，當只用空間域資訊去補時，由於有一整個畫域(field)的資訊全部遺失，因此可以看到字形中一個像素寬的部分被剔除掉了，如英文字母中 E 的底部，因此僅用空間域資訊補出來的效果十分不好，而當加入時間域的資訊後，雖然並沒有全部補回來，但是比僅用空間域補出來的效果好多了，因此在算 PSNR 時，畫面品質提升了 3 個 dB 有餘。

值得注意的一點是，在我們用時間域資訊來補償時，因為我們有第二階段的判斷，而判斷方法為看看上下點是否相近，因此會造成就算使用時間域上的資訊，一個像素點寬的線也有可能被我們剔除掉，就像是 Foreman 左上角的英文字 E，若要把它還原的話，必須要有另外的額外機制可以來判斷線狀的值是很穩定的，但是攝影機擷取的影像不僅雜訊大而且常常會晃動，因此我們系統對於類似的額外點保持剔除的態度。

表 5.1 各種 sequence 與 PSNR 比較圖

Methods	Videos									
	Akiyo	Bridge	Container	Foreman	Mobile	Mother	News	Paris	Stefan	Hall
LD	33.96	24.16	24.38	27.91	21.31	33.1	28.23	23.41	23.23	25.73
LA	39.87	25.14	28.69	29.9	25.56	38.53	33.58	26.55	27.62	31.57
Mean	36.51	25.08	27.97	29.7	24.21	37.22	30.79	24.54	26.9	30.04
Median	38.26	25.14	28.13	30.07	24.43	37.93	31.94	24.85	26.85	30.53
ELA	38	25.02	27.77	29.92	23.67	37.74	31.51	25.23	26.3	30.39
Proposed	39.71	25.17	28.41	30.1	24.99	38.25	32.94	26.26	27.12	31.19
Proposed_T	40.37	27.79	28.88	33.62	25.02	38.76	34.06	26.81	27.13	31.53

表 5.1 中，Proposed 方法為直接採用 Spatial 資訊去作 de-interlacing 的方法，而 Proposed_T 則是包含了 Temporal 的資訊。可以發現到，在大多 Video sequence 裡面，Proposed_T 的 PSNR 都是最高的，除了影像變動比較劇烈(Mobile、Stefan)，或是影像雜訊比較高的 Video(Hall)。而在這些影像中，LA 方法是 PSNR 最高的，其實可以發現到，若是光比 PSNR 的話，LA 都會非常高，因為它是拿上下兩點的均值來內插，所以除非有長條的斜向邊界存在，不然直接拿 LA 來運算就已經可以得到非常好的效果，但是當有長條的斜向邊界時，LA 的 PSNR 雖然也不低，但是整體畫面的品質就十分不好，如 Foreman 的前半段所示。

5.5 小結

去交錯(De-interlacing)是一個用來轉換的工具，把傳統交錯式(Interlaced)掃描的影像轉換成漸進式(Progressive)掃描的影像，讓影像在漸進式掃描的顯現裝置上能夠對齊時間軸，使得播放出來的影像有一致性，我們根據影像內容的特徵來做去交錯的動作，在背景的地方我們主要是用到時間域上的資訊，而在前景的地方則是用到空間域上面的資訊，而在時間域上，並沒有用到 Motion Compensation 而是直接對同樣空間位置的值拿來補償，而在空間域去交錯上面，則是利用了分群的概念，先判定 LA 適不適合，再決定是要採取 LA 或是 median 的方式來去除交錯的影像。



第六章 影像縮放 (Image Scaling)

本章說明本篇論文提出影像縮放的方法，從基本概念，推展至實作上的重點。

6.1 概念

在本系統中，由於有經過 MPEG4 的壓縮，影像的大小已經變成了 CIF 大小 (352*288) 的格式，而這樣的大小如果能夠把它加以放大，這樣對於細微的動作與紋理能有比較好的影像品質，因此對影像縮放領域做一系列的探討，並根據邊界上影像的特出性，提出了自己的想法，在最後並做分析與比較。

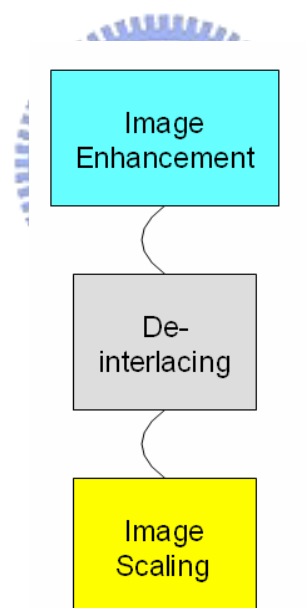


圖 6.1 影像加強部分區塊圖，本章所要討論的屬於其中的影像縮放部分

由於內插方式是要建構於監控系統之上，所以複雜度不能太高，因此我們分析最爲簡單的兩個內插方式: Bilinear 以及 NN(Nearest Neighbor)，希望能夠找到他們所欠缺的部分，並且加以改進。

6.2 邊界的分析

以傳統的影像縮放技術而言，在邊界問題的處理上常常會造成影像的模糊(Bilinear)，或是造成影像的區塊狀(Nearest Neighbor)，尤其是當放大倍率越大時這個問題將越加地明顯。

下圖為 Bilinear 和 Nearest Neighbor 方法放大 Foreman 影像後所產生的圖形。



圖 6.2 本圖為針對 Foreman 影像先用 NN 取樣到 240*192 個點之後在分別用 Bilinear 和 NN 內插回 352*288 個像素點，上圖為內插後的影像圖，下圖則是上圖最右上角長寬各別放大四倍後的結果

當我們放大屋簷的來觀看，會看到 Bilinear 將屋簷模糊化了，使得影像的銳利度降低，而 NN 方法雖然保存著銳利度，但是內插的方式卻不是沿著邊界的方向資訊，而是用方形的模型來做為內插的標準，造成在斜向邊界上的會產生鋸齒狀的波紋。

6.3 提出的改進

由於 Bilinear 以及 NN 各有其優點與其缺點，Bilinear 在平滑區能夠做出簡單有效的內插，而且不會破壞原始影像的平滑程度，但是在邊界地方卻很容易造成影像的模糊；另一方面，NN 能夠有效的保持住邊界的銳利程度，但是會產生嚴重的 aliasing 的結果，而且在邊界的地方，其內插方式是是以方塊狀為模型，因此容易在邊界產生鋸齒狀波紋。

因此，我們提出一個方法，首先先對影像內容區分為平滑區，邊界區，以及中間區，而在平滑區之中，利用一階線性內插 Bilinear 可以對平滑區做有效的內插，並不會破壞到平滑的區域，而若是處於難以判斷為平滑區或是邊界區的中間區時，我們利用 Win-scale 的方法來直接做內插。Win-scale 可以視為在端點附近為 NN 而過一段距離後用一階線性內插，這個距離與縮放的比例有關，放大比例越大則越像 NN，放大比例越接近為 1 則越像 Bilinear 的方法，用此方法的原因是我們無法判定這塊區域是屬於平滑區還是屬於邊界的部分，在平滑區我們希望能有平滑的特性，而在邊界能有銳利的特性，因此我們選用 Win-scale 來當做我們的內插方式。

最後，當處於邊界區時再簡單的分類出邊界的方向，再根據方向進一步的做影像的縮放。



6.3.1 邊界的判定

由於我們是要架構在即時監控系統上面，因此方式的複雜度是我們所關切的，複雜的運算會使得速度降低，因此我們希望一切的運算能夠用距離待補點最近的四個影像點就可以達到，就像 Bilinear 和 NN 一樣，因此對於邊界與否的判定，我們直接用離待補點附近四點的最大和最小值相減，並觀察在平滑區和在邊界區時，最大和最小的差量為多少。

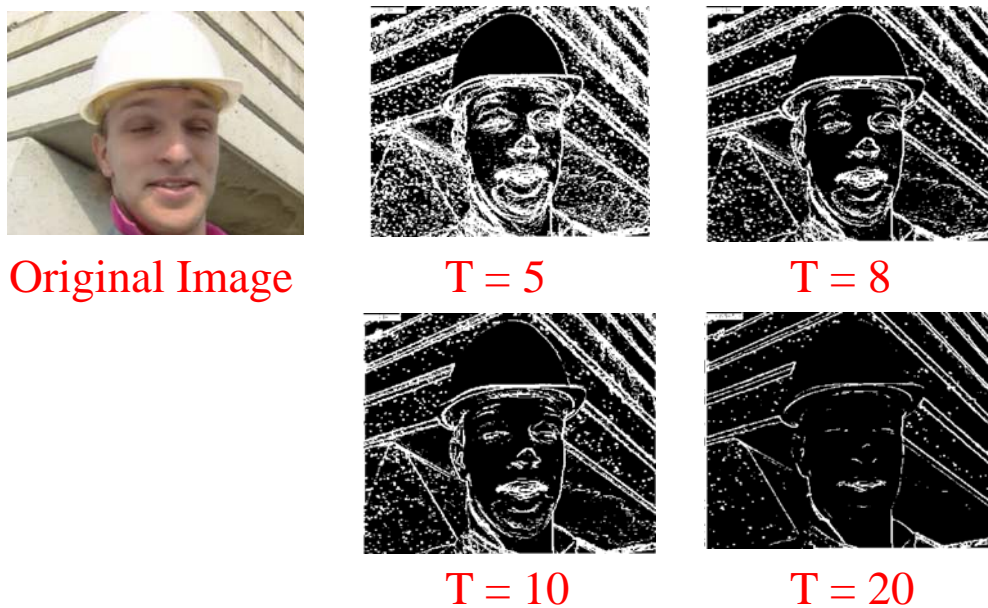


圖 6.3 利用 Foreman 模擬分析門檻值的設定。左上圖為原始影像，依序往右邊為設定門檻值為 5、8、10、20，利用相鄰四點的最大最小值的差量算這區塊的斜度，若差量大於此門檻則為亮的，反之則為暗的區塊。

如圖 6.3 所示，當門檻值為 10 以下的部分，可以視為平滑區的部分，而大於 20 的部分則可以代表為邊界的部分，而介於 10 到 20 的中間部分則為無法判定的邊界區。

因此以 Foreman 影像而言，可以得到一個判斷是否為邊界的條件

- 平滑區：此區域四週的端點最大與最小值差量小於 10
- 中間區：此區域四週的端點最大與最小值差量介於 10 到 20 之間
- 邊界區：此區域四週的端點最大與最小值差量大於 20

6.3.2 建構新的模型

我們將在平滑區使用 Bilinear 來做為內插的方式，因為可以達到平滑的效果，而在中間區時，我們利用 Win-scale 來做為內插的方式，因為 Win-scale 保有平滑與銳利的特性，當在邊界區的時候，希望在內插完之後能夠保持其銳利度與方向性，因此我們採用改良型的 Nearest Neighbor。

接下來我們分析從四個端點可以看到的資訊，首先分析邊界的方向性，一般可以視為直的、橫的、斜的三類，傾斜的角度各種方向都有可能，簡單起見我們考慮 45 度的方向性來當做代表：

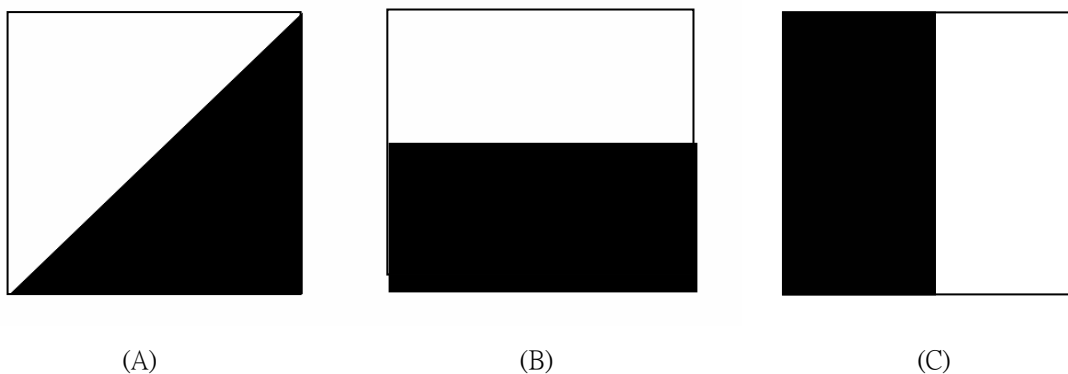


圖 6.4 邊界的三種情況，(A)斜的、(B)橫的、(C)直的邊界。

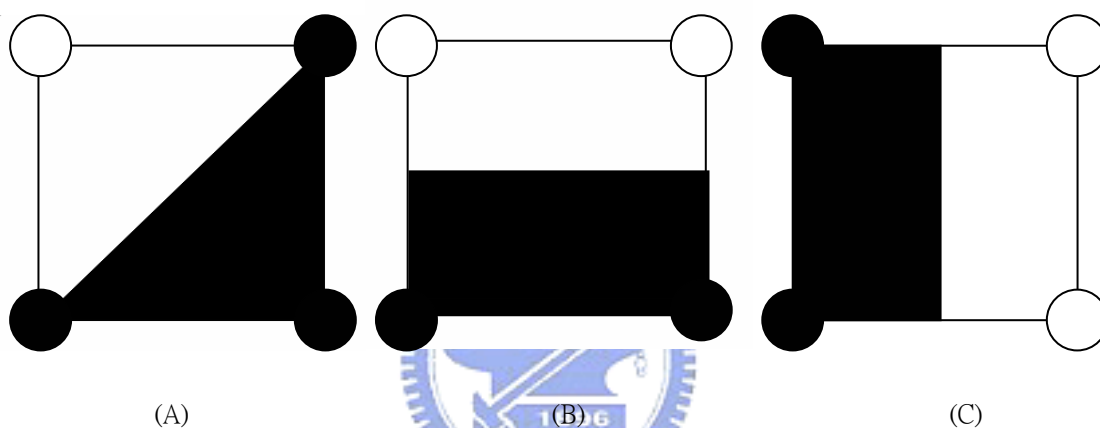


圖 6.5 各種邊界所對應的端點值，(A)斜的，角落值不確定、(B)橫的、(C)直的邊界。

圖 6.4 展現出各個邊界可能的情况，圖 6.5 則代表其對應的端點值，其中，斜邊的部分由於不清楚靠近哪一方，所以有可能為白色或是黑色。

在討論完邊界的方向後，我們試著去分析內插時在邊界區域的特性，為了方便起見，把物體分為兩個區塊，一塊是黑的另一塊是白的。分析周圍四點可以分成三類，第一類為其中三個點為一組，另一點為一組，而這種情形往往代表著此邊界區屬於斜邊的部分，第二類為垂直或是水平的邊界，其中相鄰的兩點為一組，另一邊相鄰的兩點為另一組，而第三類可以視為一個像素點寬的線，這一類為對角的兩點為一組，另外兩點屬於另一組，如下圖所示

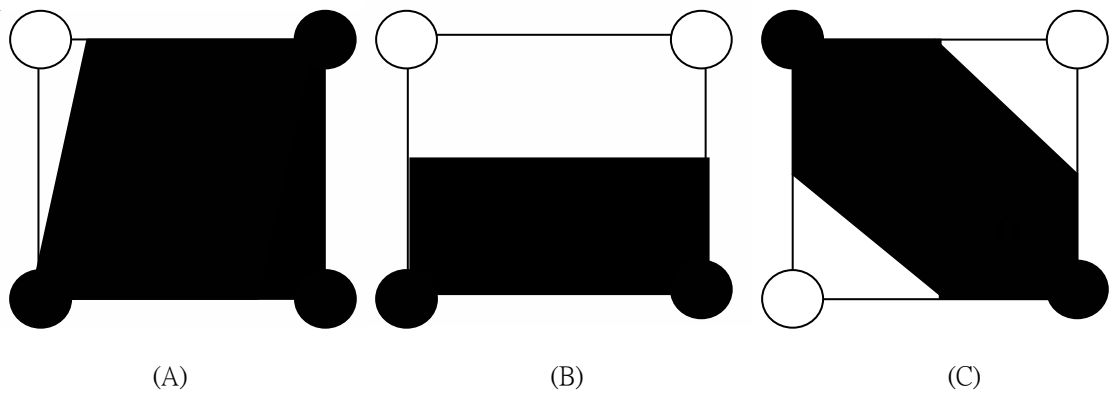


圖 6.6 可能的邊界對應的端點值，(A)斜邊、(B)橫的、(C)一個像素點寬的斜邊。

以 NN 來做內插時，對於上面的端點，它將會內插出如下圖的形式：

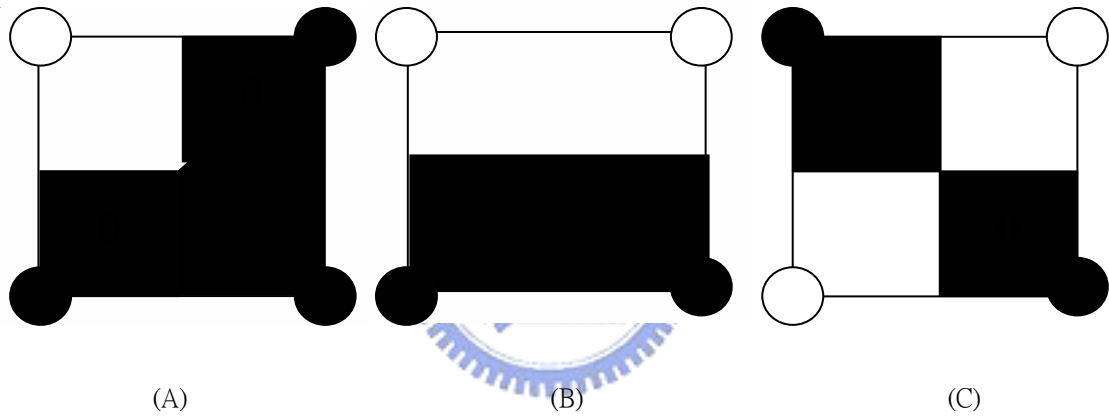


圖 6.7 NN 方法對應此三種端點值的處理方式，(A)斜邊、(B)橫的、(C)對角。

圖 6.7 為 NN 對應邊界時四個端點與內差點之間的關係，把最為靠近的點的值直接待進來，完全不考慮其邊界的方向性，有鑑於此，我們提出了另一套模型，希望能夠針對此不足的地方做補強。

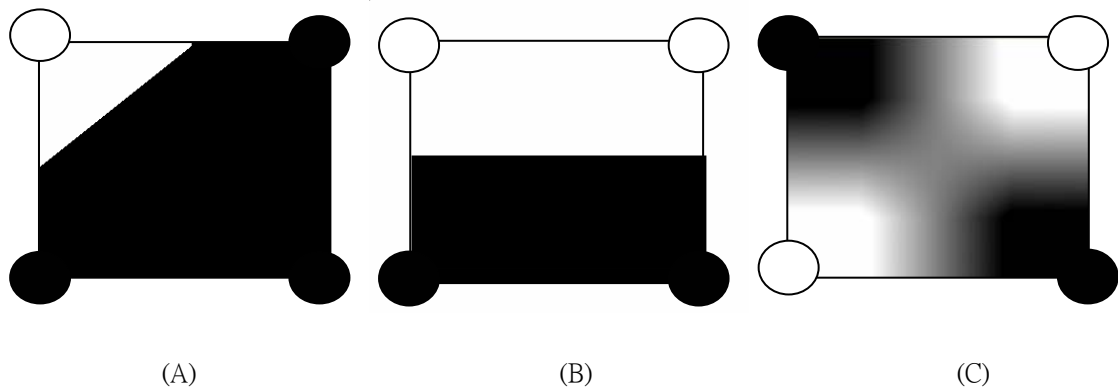


圖 6.8 提出的模型，(A)斜邊、(B)橫的、(C)對角。

圖 6.8 為提出的模型與對應邊界四個端點之關係，當有三個點為同一個群落時，我們提出的模型為從一個點的群落網外延伸，到另外兩端點的中點畫一條線連起來，若內差點在此三角形內，則直接用此點的值去補，而當落再此三角形外面時，則利用另外三點的資訊去做內插。若是直線或是斜線的情況，則與 NN 相同，若是對角線為同一群落時，因為僅以四個點來參考並沒有多餘的資訊告訴我們可能是哪個方向的線段，因此我們使用 Bilinear 來做為我們內插的方式。

6.3.3 邊界點的分群的方法

當我們判定為邊界區域後，直接把四個點分類為兩群，利用最大和最小的平均值來當作區分的門檻，這麼一來就可以套進提出的模型裡面做進一步的內插。

6.4 測試環境與實驗結果

我們主要是利用 Foreman sequence 來做為我們測量的平台，首先先利用 NN 來降頻到 240*192 各種情形，在利用四種方法 NN、Bilinear、Win-scale、Proposed Method 來內插還原出 352*288 的影像，下面為各種圖示



圖 6.9 本圖為針對 Foreman 影像先用 NN 取樣到 240*192 個點之後在分別用四種內插方式補回 352*288 個像素點。



Bilinear



NN



Win-scale



Proposed

圖 6.10 本圖為針對 Foreman 影像先用 NN 取樣到 240*192 個點之後在分別用四種內插方式補回 352*288 個像素點，並且把最右上角部分長寬各放大 4 倍。



Bilinear



NN



Win-scale



Proposed

圖 6.11 本圖為針對 Foreman 影像先用 NN 取樣到 240*192 個點之後在分別用四種內插方式補回 352*288 個像素點，並且把最左上角部分長寬各放大 4 倍

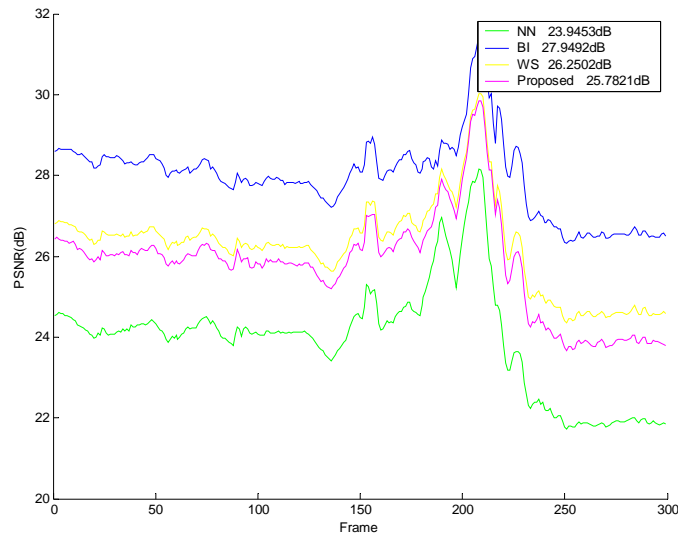


圖 6.12 四種內插方法的 PSNR 比較圖，先用 NN 取樣到 240*192 個點之後在分別用四種內插方式補回 352*288 個像素點

圖 6.10 中可以看到，原本鋸齒狀的波紋轉變成爲比較斜向的邊緣了，但是由於我們的 model 主要是根據 45 度的斜向來判斷，所以當斜邊方向爲 45 度倍數時，能夠達到比較好的效果，但是當斜邊的角度不爲 45 度的倍數時，我們提出的方法會用 45 度與水平和垂直軸組出其斜邊的角度，這樣一等於是把斜邊的效應累加到一定程度在利用特定 45 度倍角的斜邊組回來如圖 6.11，而原本的 NN 則是只有垂直和水平軸可以組回來，若是要更貼近斜邊的角度讓模型進一步的準確的話，則需要更多的運算或是要參考更多的點，才有可能，這麼一來運算的複雜度也會更加提高，因此我們沒有再往下繼續討論下去。而圖 6.12 爲四種方法的 PSNR 比較，可以看到 Bilinear 最高，其次是 Win-scale、Proposed，最後是 NN，由於要保持銳利度，所以很多的內差點都直接拿旁邊的點來直接套，造成失去了影像的平滑特性，而用提出來的方法內插在長條形邊界上會有不錯的效果，但是對於只有一個點狀的脈衝或是邊界的端點而言，或因爲模型的關係轉成菱形形狀，像左上角 SIMENS 的字跡那樣，字跡旁邊角落就有可能產生 45 度斜角的切邊，而這個與原始影像不合，使得 PSNR 降低。

6.5 小結

當初發展影像縮放技術主要是要應用在即時重播上，把原本的 CIF 大小的影像轉換成 640*480 像素大小的影像，因此利用簡單的方法 Bilinear、NN 以及利用他們組成的 Win-scale 這三種方法來做爲我們方法的基本核心，利用影像的特性做分類，分別對應不同的內插方式，讓平滑區有平滑的特性，邊界區有銳利的

感覺，但是由於僅僅考慮到四個點的資訊，所以造成方向性上無法提出更多的方向，以至於效果僅能說差強人意，甚至再高倍率放大時還有鑽石塊狀的區域產生，這種影像的視覺效果就十分不好，不過在放大倍率不到 2*2 倍的情形下，內插出來的效果還算可以接受，並不會造成太大的缺陷，而且也能把邊界方向描繪地還不錯。



第七章 結果與討論

本章說明本篇論文經過整體的壓縮、影像的縮放以及去交錯系統後，模擬得到的效果與改進，做一個總結。

7.1 系統介面與實驗結果

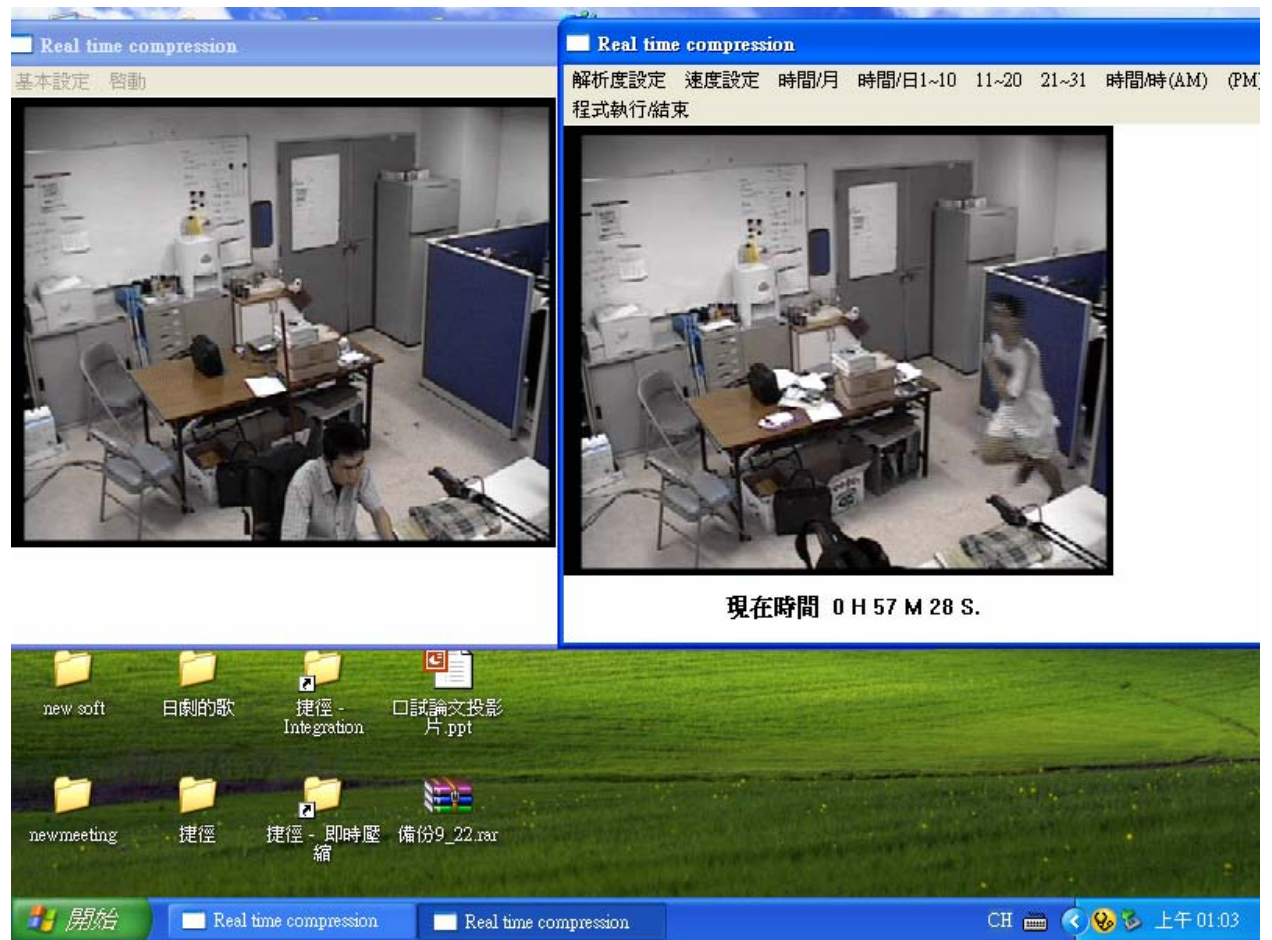


圖 7.1 操作介面，左端為即時監控並且同時壓縮儲存，右端為重播的介面

在本系統可以分為壓縮端，即時監控端，以及重播三大部分，其中壓縮方式是以 MPEG4 Simple Profile 為主體，再配上 Frame Skip、Block-Change、QP Adaptation 的機制，讓壓縮效率進一步的提升，最高可以達到每十分鐘的影像僅花 12KByte 儲存，整整壓縮了 22 萬 5 千倍，當我們一面在錄影的時候也可以對現場的場景做即時的監控，在最後的重播端部分，我們做了兩個改善影像品質的動作，分別為去交錯機制(De-interlacing)與影像縮放(Image Scaling)機制，讓我們影

像的品質進一步的提升。

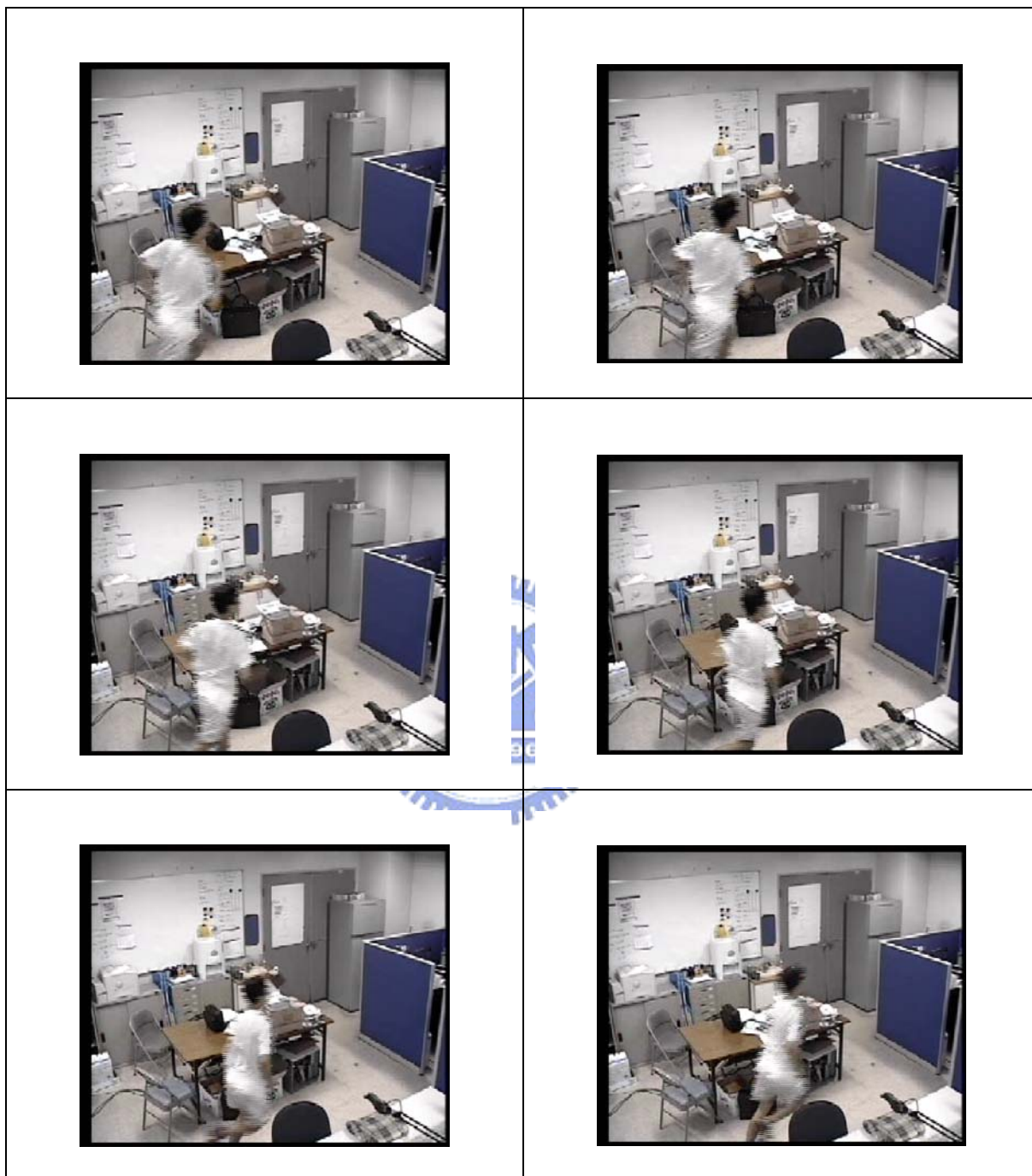


圖 7.2 交錯式影像訊號

圖 7.2 並沒有使用去交錯機制直接播放一段跑步的影片，可以看到由於交錯式訊號的時間點沒有對齊，所以在人的周圍有很明顯的鉅齒狀波紋。

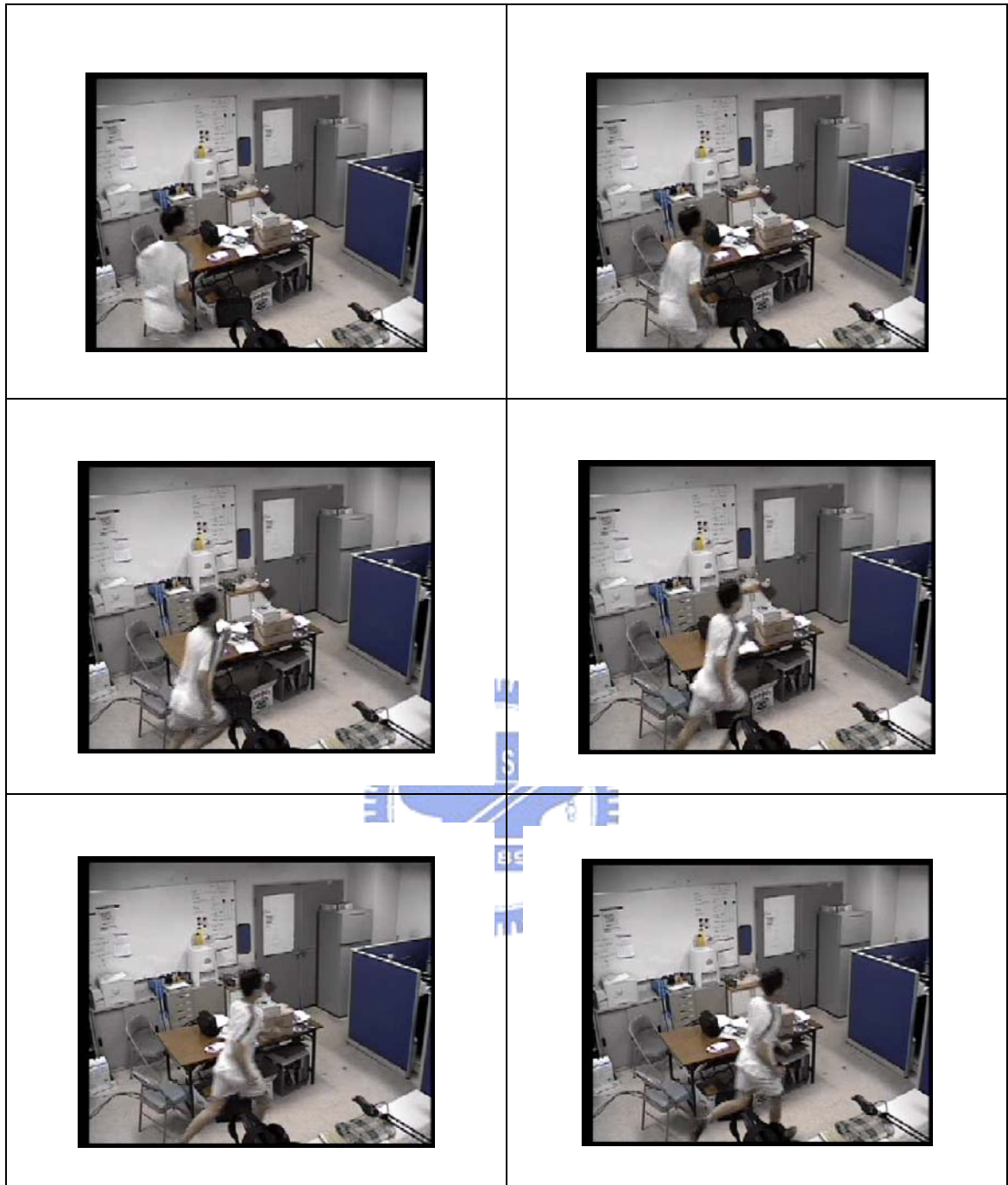


圖 7.3 去交錯式機制下所產生的影像訊號

圖 7.3 使用去交錯機制後播放另一段跑步的影片，可以看到雖然交錯式訊號的時間點沒有對齊、但是由於有經過去交錯的機制，所以整個人並不會感覺到有鉅齒狀的波紋，而是一個完整的個體。



圖 7.4 影像縮放機制，左上角為原始壓縮時影像大小 CIF 格式，其他的則是用其他影像縮放方法放大為 640*480，方法依序為 NN、6-tap FIR、Bilinear、Win-scale、Proposed Method

圖 7.4 為各種影像的放大機制，除了在論文中討論過的 NN、Bilinear、Win-scale 以及所提出來的方法之外，其中還有一個簡單的 6-tap FIR filter。不過由於它只能夠內插出中間點來，所以並不在本篇論文內加以討論，上述的四個方法中 NN 會有區塊狀產生，Bilinear 會使得影像變模糊，Win-scale 則是介在兩者中間，至於提出的方法，可以看到影像有些微的變清晰，不過不是很明顯，可能的原因是在於長寬放大倍率並沒有超過兩倍，所以斜向邊界的內插效果並沒有很明顯。

7.2 結論

我們針對監控系統上所發現的問題做了進一步的探討，其中主要針對三個部分，分別為影像壓縮機制、影像去交錯機制、以及影像縮放機制。

在影像壓縮機制上，本系統的方法是架構於 MPEG4 Simple Profile 之上，除了利用壓縮標準進行壓縮外，更進一步的分析影像內容，為了分離雜訊與訊號，提出了三個方法，包含 Frame Skip、Block-Change 和 QP 調整機制，當畫面判定有在動的時候，找出不動的區塊把之前的影像貼過來，這樣會節省 P frame 的資料量，而當畫面沒在動時，我們丟棄影像資訊，直接利用前一張補過來，而當場時間畫面都沒有變動時，我們將會加大 QP 的參數，進一步的壓縮資料。

在去交錯機制上，在背影部分我們利用時間軸上的資訊來補，而對於前景部分我們提出了一個方法，結合了 LA(Line Average)以及 Median 的優點，對於直線段為可靠的部分利用 LA 來做，而在直線段為不可靠的時候則利用 Median 的群組特性，來做為我們的去交錯方式。

而在影像縮放的部分，我們根據影像的內容不同對應不同的縮放機制，在屬於平滑區的部分，利用 Bilinear 的方式做內插，而在邊界的部分，我們利用所提出的 Model Based NN 的概念來做內插，而介在邊界和平滑區的部分，我們直接用 Win-scale 的方法來做為我們的內插方式。

本篇論文提出了一套即時監控的機制，並且對於其中的問題做探討和處理，根據實驗結果顯示，這些方法應用在即時監控下為一可行的方法。

參考文獻

- [1] Jyh-jyh Huang, "Visual Improvement on MPEG-4 Video Coding with Frequency Information Selection", Department of Electronics Engineering, Institute of Electronics National Chiao Tung University Res. 2004
- [2] R. S. Prodan, "Multidimensional Digital Signal Processing for Television Scan Conversion", Phillips J. Res. 41, pp. 576-603, 1986
- [3] T. Doyle, "Interlaced to Sequential Conversion for EDTV applications", in Proc. 2nd Int. Workshop Signal Processing of HDTV, pp. 412-430, Feb. 1988
- [4] H. Rabtanen, "Color Video Signal Processing with Median Filters", IEEE, Trans. on Consumer Electrons, Vol.38, No. 3, pp. 157-161, Aug, 1992
- [5] S.C. Tai, C.S. Yu, F.J. Chang, "A Motion and Edge Adaptive Deinterlacing Algorithm", Multimedia and Expo, 2004. ICME '04. 2004 IEEE International Conference on Vol. 1, pp. 659 – 662, June 2004
- [6] M.Q. Phu, P.E. Tischer, H.R. Wu, "A Median based Interpolation Algorithm for Deinterlacing", ISCAS '04, Intelligent Signal Processing and Communication Systems, pp.390-397, Nov. 2004
- [7] C. T. Hsu, M. J. Chen, C.H. Huang, "High Performance Spatial-Temporal De-interlacing Technique Using Inter-field Information", ISCAS '04, Circuits and Systems, Vol. 2, pp.II-213-16, May 2004
- [8] S. Yang, Y. Y. Jung, Y. H. Lee, R.H. Par, "Motion compensation assisted motion adaptive interlaced-to-progressive conversion", IEEE, Trans. on Circuits and Systems for Video Technology, Vol. 14, pp. 1138-1148, Sep. 2004
- [9] C. H. Kim, S. M. Seong, J. A. Lee, L.S. Kim, "Winscale: an image-scaling algorithm using an area pixel model", IEEE, Trans. on Circuits and Systems for Video Technology, Vol. 13, pp. 549-553, June 2003
- [10] E. Aho, J. Vanne, K. Kuusilinna, T.D. Hamalainen, "Comments on "Winscale: an image-scaling algorithm using an area pixel Model", IEEE, Trans. on Circuits and Systems for Video Technology, Vol. 15, pp.454-455, March 2005