

第一章 緒 論

MPEG-4 與 H.264 串流視訊壓縮技術是近年來的壓縮技術主流，因為它們有著高壓縮比的特點，使得經壓縮完的檔案變得具有高可攜性。然而，它們也有著一個共通的缺點，壓縮完的檔案對於傳輸錯誤的發生，如封包遺失、位元遺失、位元反相...等，相當地敏感；在解碼端，由於 MPEG-4 與 H.264 兩種標準皆有使用到其它的 Frame (或稱 Inter Frame) 來完成預測及編碼[1][2]，因此，某一位元的遺失，可能導致錯誤一直漫延到後面的數張畫面，甚者，導致後續的整個畫面無法成功地解碼重建。

雖然我們可以不理會這些壞掉的畫面而將它略過，只需要重新找到同步點，繼續執行解碼的動作，即可繼續正確解碼。然而，對於消費者的視覺享受而言，這並不能使他們感到滿意。因此，有許多學者提出了一些機制來修復這些已經損毀的畫面，以維持畫面的連續性，這些方法當中，最常被使用的方法有錯誤回復(Error Resilience)及錯誤隱藏(Error Concealment) [3]兩種技術。

錯誤回復技術是一個比較複雜且比較耗力的方法。此技術主要是針對編碼器做修改，以達到錯誤發生時，能藉由其他的資訊將損毀的影像修補回來；然而，錯誤往往都是無法預測的，一旦確定了編碼的方式，但若遇到了無法預期的錯誤，甚至是高錯誤發生率或高資料遺失率時，則將會發生無法解碼的現象，因此，錯誤回復的方法比較起來較沒有彈性[3]。

錯誤隱藏技術相對於錯誤回復技術而言，則較為容易實現，此技術主要是實現於解碼端。一般而言，錯誤的發生大多存在於傳輸階段，主要的原因是一般的網路多數屬於錯誤傾向的網路，例如：無線傳輸、或

UDP 傳輸架構等，故無法確保其服務的品質(QoS, Quality of Service)，因此當錯誤已經發生，若在解碼端先行判斷錯誤的種類，再進行特別的錯誤隱藏，比起錯誤回復的方法，較為實用。

1.1 視訊影像壓縮標準

MPEG-4 及 H.264 為最近兩年業界及學術界的熱門研究及研發標準。主要是因為現今的資訊科技越來越發達，電腦工業的進步一日千里，許多以前因為軟硬體技術瓶頸而無法達到之效能，在現今的高速電腦上，大多已經可以順利完成了。視訊影像編碼，在早期的電腦上，即為一種具高運算量的技術，因此在九零年代，這些編碼大多須要藉由特殊電腦如 PowerPC 等來完成。以下本論文將先介紹關於 H.264 及 MPEG-4 兩種視訊編碼標準。

1.2 視訊影像壓縮原理

MPEG[19]為 Motion Picture Expert Group 的簡寫，為 ISO/IEC 底下的一個協會，主要訂定國際壓縮、解壓縮、處理、動態影像表述式、聲音及上列的組合標準。目前該團體訂定出來的標準有 MPEG-1、MPEG-2、MPEG-4、MPEG-7 及 MPEG-21。MPEG-1 及 MPEG-2 主要應用於 VCD 及 DVD，而 MPEG-4 主要的應用除了 VCD 及 DVD 外，亦即將應用在數位電視(Digital Television)上，MPEG-4 除了強調高壓縮比之外，更強調與使用者互動之功能，並加入與先前影像壓縮標準所未擁有的以物件為基底的壓縮技術。

VCEG 為 Video Coding Experts Group 之簡寫，為國際電信組織 (ITU-T) [20]的子成員。該團體訂制了一系列的國際視訊影像通訊標準，

如 H.261、H.263、H.263+、H.263++以及最近很熱門的 H.26L。VCEG 所訂制的國際標準，與 MPEG 的不同點，是應用在視訊通訊上，如視訊會議、網路電話...等。早期 MPEG 與 VCEG 皆各自訂定各家的標準，然而，從 2001 年開始，VCEG 與 MPEG 合作，一起訂制 H.26L 標準，MPEG 稱為 MPEG-4 / Part-10，而 VCEG 則沿用 H 系列的編號，自稱為 H.264。VCEG 所訂制的標準大多不易被看見，主要的原因是因為 H.26x 系列的標準大多被包裝在系統層，因而一般人較少聽見。

圖 1 為 H.264 壓縮標準之架構圖[4]。視訊壓縮的主要方法是去除三種冗餘性(Redundancy)來達到壓縮的目的，它們分別為時間冗餘性(Temporal redundancy)、空間冗餘性(Spatial Redundancy)及統計冗餘性(Statistical Redundancy)。

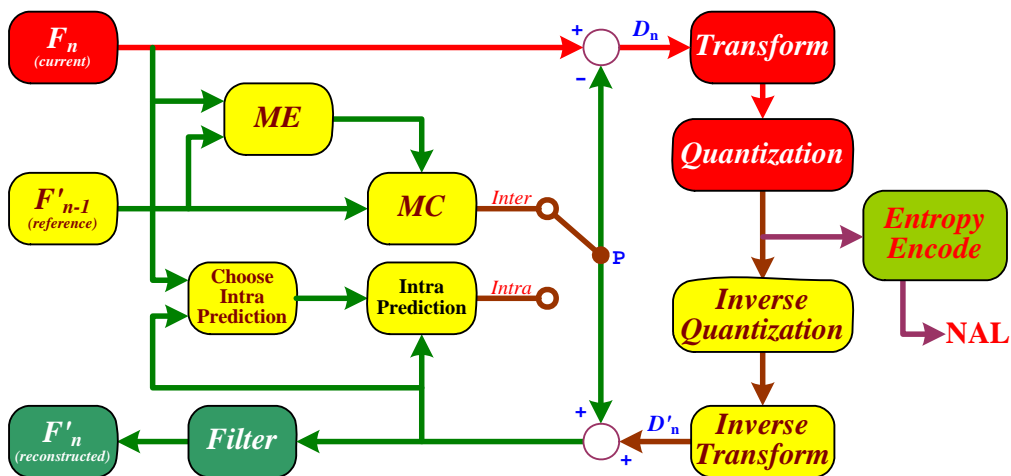


圖 1 H.264 編碼系統架構

空間冗餘性是經由一些色彩轉換(Color Transform)後再交給離散餘弦轉換將色彩分成高頻與低頻兩個部份，接著在高頻成分中再加入適當的量化處理，由於人眼對高頻的敏感度遠低於在低頻的變化，因此在高頻成份中加入量化可達到減少資料量，進而達到壓縮效果。

第二部份緊接著的是要除去時間的冗餘性。一般在差一個取樣時間單位的畫面之間的影像內容有許多是具有高相似性的，因此在以 Block 為基底的畫面裡，利用移動估測(Motion Estimation)及移動補償(Motion Compensation)可以大幅的減低資料量，記錄中僅需要記錄與搜尋範圍中所對應到最相似的 Block 最為相似的移動向量以及該 Block 與移動向量所對應 Block 之像素差值，因此，每一個 Block 中所記錄的值愈接近零值，愈能達到良好的壓縮效果。在 MPEG-4 中，每一個 Macroblock 的大小皆固定為 16×16 的大小，然而，在 H.26L 中，為了使影像重建的效果較先前的標準來得好，H.26L 標準訂定了更多種 Macroblock 的分割分式[2][14]，如圖 2 所示。若加以組合分割方法，可以有 16 種分割方式。

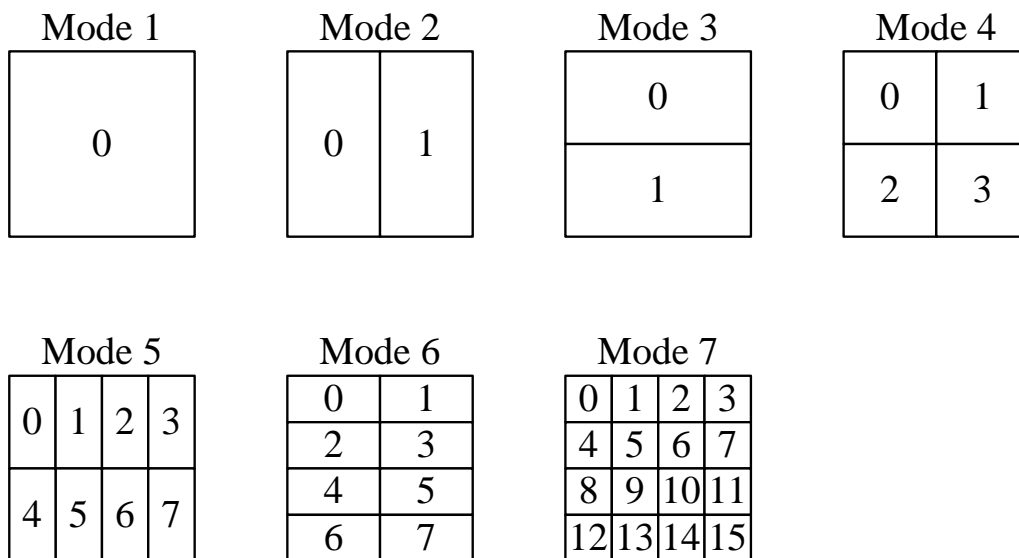


圖 2 H.26L Macroblock 分割模式

另一方面，移動補償則是與移動估計之逆向動作，將移動向量所指的 Block 與差值相加，而將影像還原，得到重建後的影像即為在解碼端所見到的影像。

最後，利用可變長度編碼來消除資料在統計的冗餘性質。可變長度編碼主要的目的是欲將較長出現的碼以較短的對應碼來進行編碼，而較少出現的碼，則給予較長的對應碼。因此，可以得到較好的編碼效果。

1.3 視訊影像格式及資料結構

現今常見的動態影像格式與我們所熟知的靜態影像之色彩平面並不相同，靜態影像多以 RGB 為三個主要的色彩平面，而動態影像則將 RGB 影像轉為人眼較敏感的亮度(Luminance)以及較為不敏感的彩度(Chrominance)來進行處理。在大小方面，現在的視訊壓縮支援如表 1 所示之尺寸大小。

表 1 各種影像格式之畫面大小

格式	Luminance	Chrominance
4CIF	704 × 576	352 × 288
CIF	352 × 288	176 × 144
QCIF	176 × 144	88 × 72
VGA	640 × 480	320 × 240
QVGA	320 × 240	160 × 120

影像格式除了與大小有關之外，與取樣方式亦有關係。由於人眼對影像中色相較不敏感，故在色相平面上做下取樣(down-sampling)亦可減少影像的資料量。常見的取樣方式有 4:2:2 以及 4:2:0 兩種，簡稱為 YUV422 及 YUV420，其中 YUV420 又被稱為 I420 格式。4:2:2 的取樣方式如圖 3 所示，在亮度平面上以完整點取樣，而在彩度平面則在寬度上以每兩點取樣一點方式取樣。4:2:0 取樣方式如圖 4 所示。此取樣法

在亮度平面上之取樣點和 4:2:2 相同，而在彩度平面則在寬度及高度上皆以每兩點取一點的方式進行取樣。

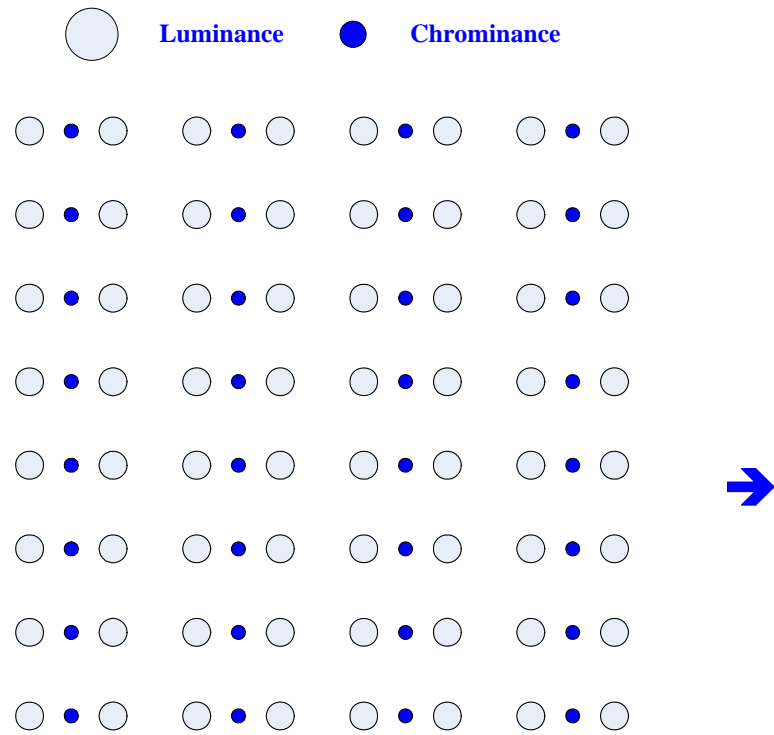


圖 3 YUV 4:2:2 取樣格式

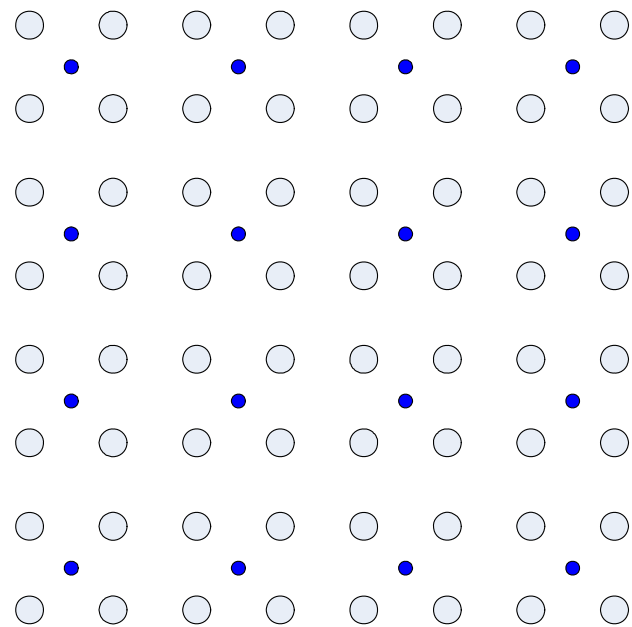


圖 4 YUV 4:2:0 取樣格式

1.4 H.264 編碼標準簡介

H.264 編碼標準與 MPEG-4 標準一樣，在不同層次的應用上，有不同的 Profile 需求，其中共有三種不同的 Profile，分別為 Baseline Profile、Main Profile 以及 Extended Profile[2][4]。

Baseline Profile 的 Profile ID 為 66，主要應用在視訊會議及視訊電話上，主要特點如下：

- ◆ 支援 I(Intra)及 P(Inter)兩種畫面的編碼。
- ◆ 加入 De-Block Filter。
- ◆ 支援 Field-Base 編碼。
- ◆ 使用 Zig-Zag 掃瞄。
- ◆ 使用四分之一點的 Inter Prediction。
- ◆ 使用 CAVLC(Context-Adaptive Variable Length Coding)。
- ◆ 樹狀結構的 motion segmentation，最小可分割至 4×4 的大小。
- ◆ 支援任意區段順序(Arbitrary Slice Order；ASO)及可變區塊順序(Flexible Macroblock Order；FMO)兩種編碼模式。
- ◆ 影像編碼格式為 YUV420。

Main Profile 的 Profile ID 為 77，主要的應用為廣播視訊系統，其特點如下：

- ◆ 支援 I、P 及 B(bi-direction)畫面。
- ◆ 熵編碼使用 CABAC(Context Adaptive Arithmetic Coding)。
- ◆ 支援交錯(Interlaced)模式編碼。
- ◆ Inter-Coding 使用 weighted prediction。

Extended Profile 的 Profile ID 為 88，主要應用在串流多媒體之編解碼，其特點如下：

- ◆ 支援 SP(Switch-P)及 SI(Switch-I)模式編碼。
- ◆ 改良型錯誤回復，如 Data-Partition。
- ◆ 不支援交錯模式編碼。

圖 5 為 Baseline、Main 及 Extended 三種 profile 之特點示意圖。

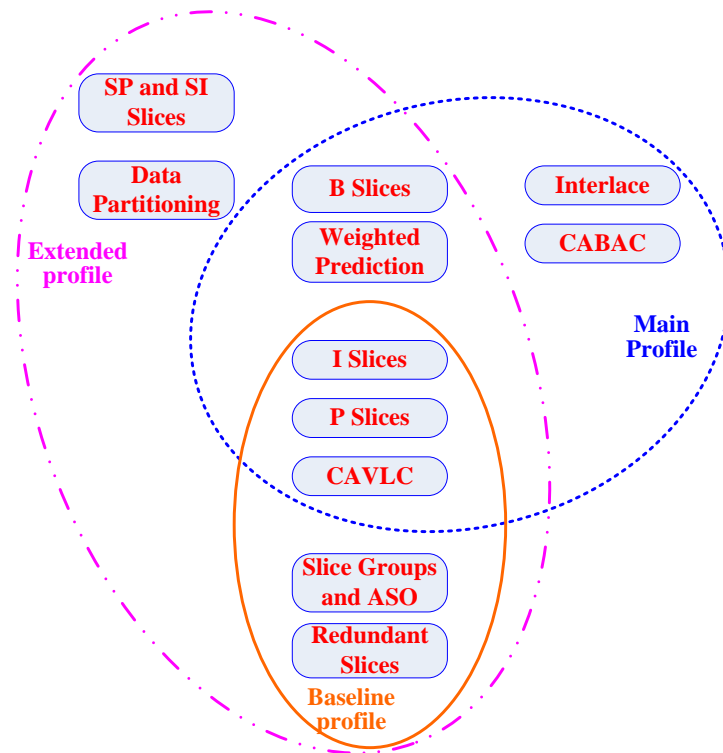


圖 5 H.264 Baseline, Main and Extended Profiles[4]

1.5 可變區塊順序 (FMO) 編碼

在無線通訊中，存在著兩個很大的課題，即頻寬的限制和雜訊的干擾。為了抵抗雜訊的干擾，H.264 標準中定義兩種抵抗的方法，分別為錯誤回復(Error Resilient)及錯誤隱藏(Error Concealment)，此部份將在後面章節說明。除了使用錯誤更正的工具之外，在 H.264 中，另外定義了一些工具，可以在錯誤發生時，讓這些因為錯誤而造成之令人討厭的視覺損害降至最低，其中的一種方法即為可變區塊序列編碼(Flexible Macroblock Order；FMO)。

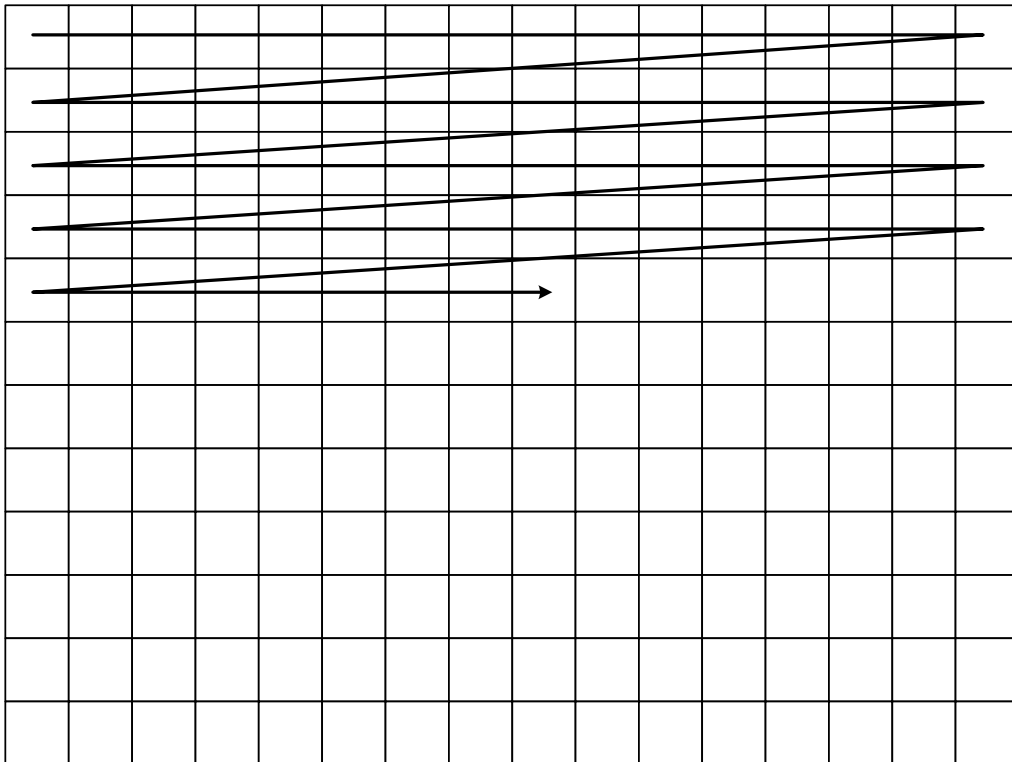


圖 6 傳統編碼掃描模式

圖 6 為傳統編碼所使用的方法，稱為光柵式掃描模式，其區塊編碼之順序為由左而右，由上而下的掃描，這樣的好處是可以讓每一個相鄰的區塊之間的相似性減小，使得壓縮的效率能夠提高；然而，若其中某一個區塊發生錯誤，將會導致一連串的區塊發生無法解碼的情形。圖 7 為交錯模式編碼的示意圖，此方法將一張畫面分割成兩個部份，第一個部份為掃描線編號為偶數(even field)之部份，另一部份則為掃描線編號為奇數(odd field)的部份，此方式的優點為若其中一個 field 發生錯誤時，可以利用另一個 field 插補後得之。

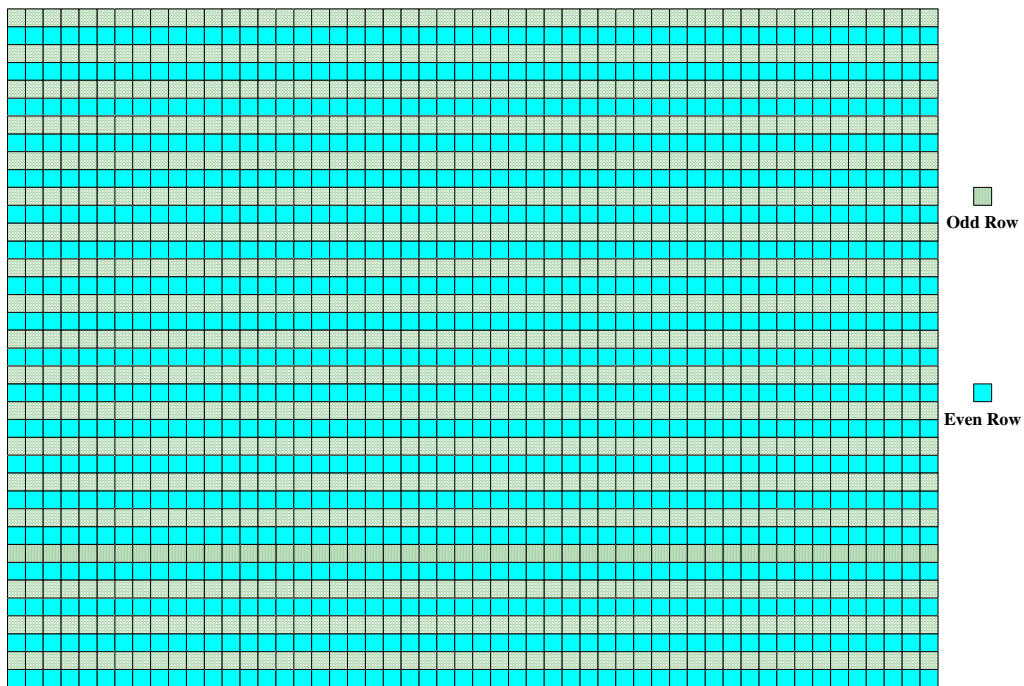


圖 7 交錯模式編碼

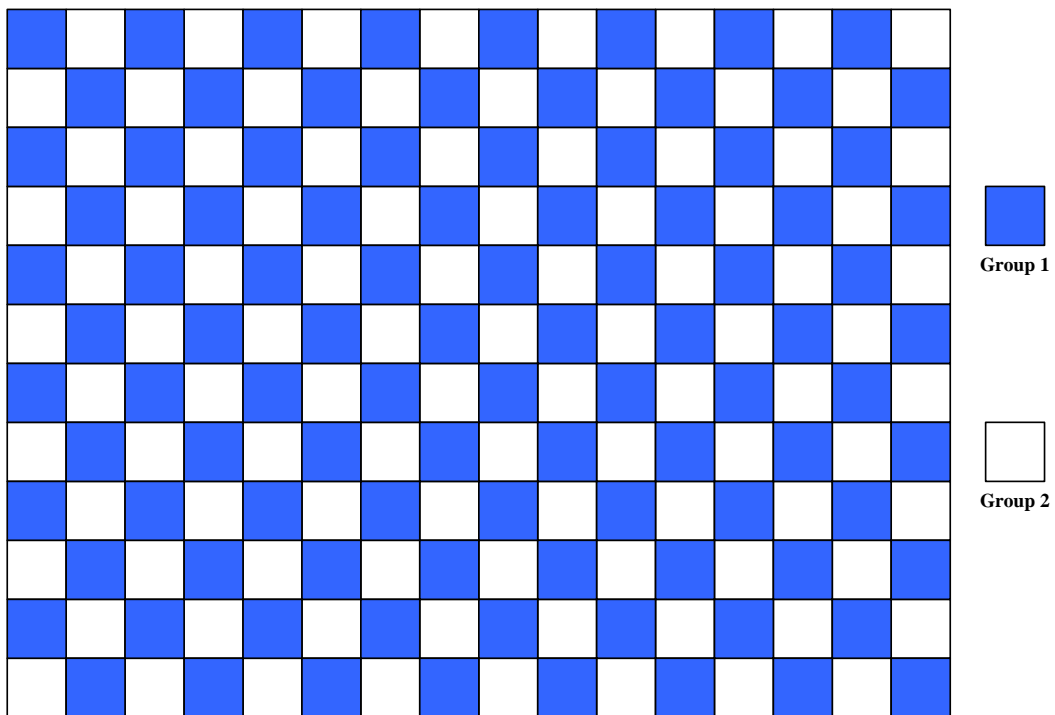


圖 8 FMO Type-1

圖 8 為 H.264 特有的掃瞄模式 FMO，此種掃瞄方式將畫面中所有的區塊分成幾個 group，在 baseline profile 中，至多可以分成 8 個 group，再依序將這幾個 group 送入編碼器進行編碼，若其中的某一個 group 發

生錯誤，則可利用其他 group 中的資料來完成補償，由於現有的編碼器多是以 block 為基底，因此，使用 FMO 可以使得 Codec 的修改較交錯模式來的少，而達到不錯的錯誤抵抗的效果。圖 9 為在 H.264 中另一種 FMO 的分類方式，稱之為 Type-2，此種模式下即是將 block 分成三個 group 再進行編碼，如此一旦某一個 group 的 block 發生錯誤，可以利用其它兩個 group 的 block 來進行修復的動作。

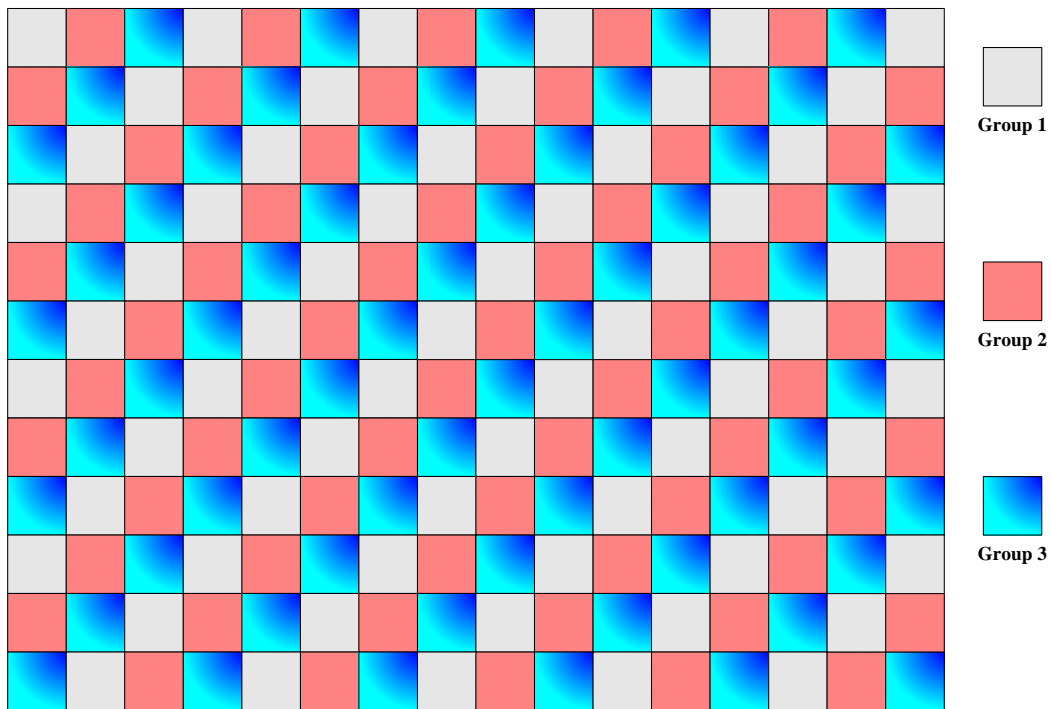


圖 9 FMO Type-2

1.6 研究動機及目的

近年來，多媒體因電腦科技的進步而蓬勃發展。早期的電腦硬體科技在有限的網路頻寬下及硬體計算速度的限制下，傳送一張影像是一個很大的負擔，更別說要再加上一些聲音或是動態影像的傳輸了；然而，現今的網路技術已經改善許多，再加上中央處理器技術進入奈米等級使得計算速度大幅提昇，Cable 及 DSL(Digital Subscriber Line)使得傳輸的

頻寬變大，傳輸視訊影像已經不是很大的問題了；另一方面，由於許多視訊壓縮的技術如 MPEG-4 及 H.26L 的提出，影像資料壓縮後更變得很小，使得視訊影像的傳輸機率變得更大了。

從另一方面來看，視訊壓縮技術大量地採用可變長度編碼技術 (Variable Length Coding；VLC)，使得壓縮後的資料對於錯誤的發生相當的敏感。可變長度編碼技術是利用資料出現的機率特性，將較常出現的資料用一個長度較短的碼來代表，而較少出現的資料則用一個長度較長的碼來代表，藉此達到高效率的壓縮編碼；除此之外，在可變長度編碼中的每一個編碼皆是互斥且唯一的，然而，也因此使得在傳輸的過程中，只要發生資料位元遺失或資料位元反相，則可能造成部份資料無法解碼還原影像的錯誤，甚至在隨後的所有資料流皆無法解碼還原的遺憾。

為了抵抗這種因傳輸而造成的錯誤，前人提出錯誤回復方法來預測並避開錯誤的發生，亦或者是在偵測到錯誤的發生後，利用及錯誤隱藏補償技術將其修正回可接受的程度。

錯誤回復技術是在將影像進行壓縮編碼時，事先針對可能會發生資料錯誤的情況，在串流資料(bitstream)裡加入預測位元的機制，或是插入一個同步位元，以避免當發生無法還原的錯誤時，該錯誤會一直傳輸下去而影響到隨後的資料。錯誤隱藏技術則是在收到串流資料後，對於已經發生錯誤的資料，藉由偵測出發生錯誤的區段，利用該錯誤區段在畫面中周圍的資料，以時間相關或空間相關的特性將原先錯誤的區段補償回來。換句話說，錯誤回復是在編碼時即已嵌入一些位元來補救錯誤的發生，而錯誤隱藏則是在解碼端利用已經被正確解碼出來的資訊將錯誤還原。

本論文結構如下：第二章為文獻回顧，重點式地介紹過去許多學者所提出的做法；第三章說明提出的方法，以一個較好的架構來完成錯誤回復；第四章為提出的方法與其他方法之實驗結果及比較，第五章則為本論文之結論以及未來可進行的方向說明。