

類 MPEG4 視訊壓縮標準編解碼器之軟體實作 Implementation of MPEG4-like Video Codec

劉東昀 蔡建戊 吳家麟
Dung-Yung Liu, Chien-Wu Tsai and Ja-Ling Wu

台灣大學資訊工程研究所
Department of Computer Science and Information Engineering
National Taiwan University
Taipei, Taiwan, R. O. C.

摘要

由於 MPEG-4 主要是為迎合各種應用對視音訊資料壓縮方法的迫切需求而產生的，這些應用如數位儲存媒介，網際網路，有線無線通訊等。雖然目前此標準尚未定案，但其所含觀念及技術相當重要且符合現在及未來多媒體應用的需求。本論文即以 MPEG-4 的視訊部分做為 MPEG-4 相關研究的開端，並採 client-server 架構實作出一能於網際網路上執行的類 MPEG-4 視訊編碼器及一 Java-based 的視訊解碼器雛型，以展示其強大的使用者互動性，物件可調整性壓縮處理特性等功能。

關鍵詞：MPEG-4，視訊編碼，可調整性壓縮，資料壓縮。

Abstract

The MPEG-4 specification was developed in response to the growing need for a coding method that can facilitate access to audiovisual objects for various applications such as digital storage media, internet, various forms of wired or wireless communication etc. Although the specification is still in the working draft stage, the concepts and technologies of the specification are very important and suitable for the multimedia applications. This paper is devoted to the research and implementation of the video part of MPEG-4. We develop a prototype of MPEG4-like video encoder and a prototype of Java-based video decoder, which can be executed on Internet, in the client-server model by using Java Applet. The prototype can show us the strong functionality of MPEG-4 that includes user interactivity and manipulation of object scalability coding etc.

Keywords : MPEG-4, video coding, scalability coding, data compression.

1. 前言

在 1990 年底，MPEG(Motion Picture Experts Group) 制訂了 MPEG-1 標準 [1]，以位元率 1.5Mbps/sec 做為影音壓縮法的目標，可提供 VHS 視訊和 CD 音訊品質之服務。隨後，MPEG2 標準 [2][11] 也被制訂出來，以更高的位元率 100Mbps/sec 為目標，可提供直播衛星和高畫質電視等需要高品質影音訊號的應用。

MPEG 一向以資料壓縮為基礎理論來開發更多的應用領域，而傳統的視訊壓縮則是以一連串影像為壓縮對象，也就是對於一整張畫面(frame)做資料壓縮。如此會將整張畫面不重要的部份，如單調的背景，也一起壓縮進去而佔有一定程度的資料量，而不利於低位元率環境之應用。另一方面對於其他形態的影音資料，如文字 [14] 並沒有一個統一的標準，因而使互動式電視缺乏一套有系統且有效率的標準。因此，MPEG 的成員，達成一個共識：MPEG-4 將以物件為基礎的(object-based)壓縮方式來打破目前的限制，並開發更多的多媒體應用。

MPEG-4 將包含四個重要的部分：系統(System)、視訊(Natural Video)、音訊(Natural Audio)、以及電腦合成資料(Synthetic-Natural Hybrid Coding; SNHC)。系統方面：定義影音資料及其他資料如何融合在一起(Multiplexing)和在網路上如何傳輸，另外又定義了 MSDL(MPEG-4 System Description Language)用來傳送和連結物件的額外資訊。視訊部分：會把目標位元率設定於 5-64kbps 和 2Mbps 以上，以達到不同應用上所需求的頻寬。音訊部分：目標位元率設定於 2-64kbps，將會由數個不同的音訊壓縮演算法組成，以達到整個目標位元率的範圍。SNHC 部分：主要以電腦合成的資料為範疇，包括了 VRML 和真人語音系統(Text To Speech, TTS)。MPEG-4 正式發表標準前，此部分將會個別合併到視訊和音訊部分。

由於現在 CPU 的計算能力越來越強，多彩多姿的多媒體應用蓬勃發展。過去認為用軟體很難達成高速執行的演算法，在今天都變的易如反掌。因此，提供了軟體取代硬體的機會。也就是說可以用純軟體來做視訊解壓縮。除此之外，有鑑於網路的日漸

普及，跨平台應用的需求日漸升高。希望此軟體不只僅能在一個作業平台執行，更希望其具有跨平台執行的能力，且又有程式下載能力。因此選擇 Java 來實作解壓縮程式，應該是正確的選擇。雖然目前速度仍是有限，不過目前 Java 已對於多媒體應用做加速之努力，如推出 JMF(Java Media Framework)，再加上 CPU 不斷的昇級，要達到滿意的速度將是可預期的。

MPEG-4 以物件為基礎的觀念不只應用於所要壓縮的資料本身，其基本精神是希望解碼過程中所用到的程式庫也能從遠端下載，也就是說程式本身也是物件而可流通於網路上，目前的 Java 程式(byte code)恰具有這種特色。因此，本論文以 Java 實作出一 MPEG-4 的雛型，以展示其使用者互動性，物件處理特性等觀念。

本論文第二節介紹 MPEG-4 視訊壓縮的驗證原型，第三節討論類 MPEG-4 視訊解碼程式之實作，第四節討論類 MPEG-4 視訊編碼程式之實作，第五節為結論。

2. MPEG-4 驗證原型

2.1 簡介

MPEG-4 視訊驗證原型(Verification Model [4][5][7][8])，到今年四月已經更新到 7.0 版[5]。每次的更新都會將實驗結果較好的方法加入新版而取代舊有的方法，目前已發表了較正式的工作草案(Working Draft 3.0[6])，MPEG-4 視訊標準大致的架構已經成型。

在此驗證原型中，一個畫面將可視為是由多個視訊物件(Video Object)所組成，如此將可提供很高的使用者互動性，用以開發更多的應用。使用者可自由存取操作這些視訊物件，而組成自己想要的畫面。

每一個視訊物件還可分層(Layer)：基礎層(Base Layer)提供較低的解析度，加強層(Enhancement Layer)提供較高的解析度。每一層由真正影像的實體：視訊物件平面(Video Object Plane, VOP)所構成，視訊物件平面則內包含了物件形狀(shape)和材質(textual)的資料。

2.2 壓縮原理

傳統的視訊壓縮是將一連串的視訊看做一張張的連續矩形畫面，在 MPEG-4 中，視訊不一定是矩形的，可以任意形狀，矩形只是其中的特例。

2.2.1 使用 YCrCb color space 另加入 alpha channel

和一般數位視訊壓縮一樣，由於人的視覺系統對亮度最敏感，MPEG-4 也是使用 YCrCb 為其色彩空間。除此之外，由於有了物件的概念，更把透明度(alpha channel)加入考慮，用來表現物件若隱若現的

特性或是特效。為了壓縮效率考量，MPEG-4 把物件的透明度又分為兩類，一種是二元的透明值(binary alpha value)另一種則是灰階的透明值(gray scale alpha value)。二元的透明值言下之意就是說只有 0 和 1：0 表示透明也就是物件之外，1 表示不透明也就是屬於物件中。灰階的透明值就是可以把透明度應用在 0 到 255 之間。這兩種不同特性將會被分開考慮，用不同的方式壓縮，以達到較好的壓縮倍率。

2.2.2 形狀壓縮(Shape coding)

由於 MPEG-4 將以物件做為壓縮的對象，所以物件形狀之相關資訊也將進行壓縮。Alpha plane 也就是描述物件形狀的重要資訊，可分為二元形狀的壓縮和灰階形狀的壓縮。可採用方法有：四元樹壓縮法(Quartree)，MMREAD(Modified Modified Relative Address Detection)壓縮法，上下行數值壓縮法(Context-based Arithmetic Encoding, CAE)。因在本論文實作中是採用 MMREAD 方法，僅就此部分稍做說明。

MMREAD 起緣自 FAX 標準中的 G3 二維模式。為增進效率，簡化原有的定義，只定義了三個點，如圖 1。此演算法共有三種編碼模式：

(1) 垂直模式(vertical mode)

在 a0 和 b1 正下方點的距離小於等於 5 的時候，即可使用垂直模式，這是此演算法最好的狀況。做法是把 a0 到 b1 正下方點的距離編碼即可。

(2) 水平模式(horizontal mode)

如果在 a0 和 b1 正下方點的距離大於 5 但又不超過到一行寬度以及 b1 沒有偵測到的時候，就採用水平模式。

做法是先寫出水平模式的旗標，如果有偵測到 b1，則把自 a0 開始沒有改變的長度(Unchanged length, ULB)送出，如果沒有偵測到 b1，則把自 a0 開始剩下來的長度(Residual length, RLB)送出。

(3) 垂直穿越模式(vertical pass mode)

在前兩模式都不適用時，就使用垂直穿越模式，這是此演算法最差的狀況。

做法是如果 b1 沒有偵測到，則先送出一個水平模式(ULB = width)，RLB = a1 - a0 - Width。如果 b1 偵測到的話，RLB = a1 - a0。視 RLB 相當於幾個區塊寬(Width)就寫出幾個垂直模式(V0)，最後再用水平模式送出剩餘的長度(RLB)。

2.2.3 可調整性壓縮(scalability coding)

傳統的視訊壓縮，處理不同解析度的方法是個別將不同解析度的資料壓縮成個別的位元串列，而 MPEG-4 是採用如同 MPEG2 裡的可調整性壓縮，只要給定一個壓縮檔，視解碼程式的複雜度決定所要解碼的解析度。基本的可調整性壓縮可以分為時間上的可調整性(temporal scalability)和空間上的可調整

性(spatial scalability)。時間上的可調整性可以改變某一個視訊物件的播放速率，空間上的可調整性可以改變某一個視訊物件的解析度。

如果以兩個階層的架構，base layer 提供低解析度的資料，enhancement layer 提供高解析度的資料。而 MPEG-4 中的加強型態又可分為加強整個 VO，和只加強某一部分。

2.2.4 材質壓縮(Textual Coding)

2.2.4.1 離散餘弦轉換(DCT)

由於現在要壓縮的物件是不定形狀的，所以有幾種不同的方法用來適合現在這種改變：一種是填補式離散餘弦轉換(Padding DCT)，另一種是隨形改變離散餘弦轉換(Shape Adaptive DCT； SADCT)。

Padding DCT: 就是將物件以外的部份填補上其鄰近的顏色。經過 padding 後，再把整個資料進行 DCT。而在此的 DCT 也可選用 MPEG1 的 Frame DCT，或 MPEG2 的 Field DCT 模式。

SADCT: 不再是需要以 8 點為固定的輸入，而是 8 以下不定長度的 DCT。SADCT 目前仍在測試中，相關的 VLC table 和 syntax 尚未定出來。

2.2.4.2 位移向量計算方法(Motion Techniques)

由於不定形狀的關係，在目前的這個 VOP 須採用修改的區塊搜尋(Modified Block matching，or Polygon matching)。在前一個參考的視訊物件平面將做 padding，目前的視訊物件平面就只對其為物件裡面的圖點做位移向量計算，就位移向量的精準度而言，正如 MPEG-1/2 一樣，有整點和半點之分，以提高精準的位移估算。此外，也可如 H.263[3][10]一樣加入 8x8 區塊個別的位移向量。

2.2.4.3 量化(Quantization)

可選用 H.263 或 MPEG1/2 量化方法，用以彈性調整所需要的品質和效率。

2.2.4.4 Run-length Coding

經過 DCT 及量化後，區塊中的高頻係數出現零的數量很多，可再經過一個 Run-length Coding，只把非零係數編碼，可更節省編碼位元數。目前 MPEG-4 採用和 H263 相同的方法，即皆採用 zigzag scan 來做掃描零的順序。而 MPEG2 的 alternate-vertical scan 及另一種 alternate-horizontal scan 之效果仍在測試中。

2.2.4.5 可變長度編碼(Variable Length Coding)

MPEG-1/2 或 H.263 所採用的原理都是源自霍夫曼碼(Huffman code)，如此將可給定一組資料和編碼的對應表。目前 MPEG-4 採用的是 H.263 的可變長度編碼表。

3. 解碼程式之實作

本章將描述解碼程式之實作。MPEG-4 的視訊解碼程式主要可分三個部份，第一個部分是分工器(DEMUX)，第二個部分是視訊物件解碼器，最後一部份是構圖器(Composer)。

所以，其核心程式在於第二部分的視訊物件解碼器。此解碼程式將會負責傳統的視訊解碼功能，包括可變長度碼解碼器(VLC decoder)，反量化(inverse quantization)，反離散餘弦轉換(IDCT)，以下將實作上的架構予以說明。

3.1 架構

由於 MPEG-4 以 Video Object 為壓縮的對象，其位元串列也是分成 Object 個別放置且個自獨立。因此可先利用一個分工器(DEMUX)把不同的 Video Object 分送到個自的解碼器個別解壓縮，最後再送到構圖器顯示出來。所以本實作的方法是利用多執行緒(Multi-thread)完成整個解碼過程，如圖 2。

3.2 分工器

如上述，分工器把不同的物件分送到各自的解碼器個別解壓縮。所以分工器負責產生新的視訊物件解碼器，並且把視訊物件的位元串列從原來的位元串列抽離出來，分開送到所屬的視訊物件解碼器中。利用此方法可把不同視訊物件的位元串列完全分開處理，簡化解碼器之製作。

3.3 視訊物件解碼器

由於主要的影像資料都是包含在 VOP 中，所以每一個視訊物件解碼器皆以 VOP 為中心。以其資料性質可分為 shape decoding，motion decoding，和 texture decoding。整個流程如下圖 3 所示。

Shape decoding

目前採用 MMREAD 的演算法。

Motion decoding 和 Texture decoding

包括可變長度碼解碼器，反量化，反離散餘弦轉換，和向量補償(Motion compensation)。MPEG-4 目前多採用 H.263 的演算法，請參考第二節說明。

Scalability decoding

視使用者需要決定：解碼器是預設只解基礎層，加強層都將被省略掉，當使用者要求較高解析度時，再進一步解碼，用以充分達到解碼效率和兼顧使用者的互動性。

3.4 構圖器

經過這些視訊物件解碼器後，將產生數張視訊物件的影像，構圖器將這些視訊物件的影像，根據它們的相對位置和順序，依次呈現在畫面上。然而由

於每個視訊物件解碼的速度快慢不一，所以本論文實作的構圖器會使解碼較快的視訊物件解碼器先停住，等到其他視訊物件解碼器都解碼出視訊物件後，才讓它繼續解碼，如此不至於使視訊物件解碼器間漸行漸遠。

除此之外，構圖器也負責使用者的互動性，這些互動性包括可允許使用者更改視訊物件擺放的位置，放大縮小，也可接受使用者要求去解加強層以增加解析度，所以構圖器也必需負責把使用者的要求送到視訊物件解碼器。

3.5 Java applet decoder

MPEG-4 另一個特色是要有解碼程式可下載 (Decoding Downloadable) 的能力。也就是網路電腦 (Network Computer ; NC) 的概念，只就需要的軟體和資料下載於接受端。除此之外，MPEG-4 為了滿足各式應用的不同需求，其解碼器將採用彈性化的設計，解碼器由解碼演算法組成，演算法再由組成解碼器各個解碼工具(tools)組成。目前 MPEG-4 把解碼器分為三種彈性程度：

1. Flex_0(nonprogrammable)
只支援標準的解碼工具
2. Flex_1(flexible)
支援標準的解碼工具，並可修改其設定。
3. Flex_2(extensible)
可支援任何的解碼工具。

Flex_2 的解碼器，可接受遠端編碼器的下載工具，用以解碼不同壓縮法的資料，這也是我們實作解碼器的目標，在這個架構下解碼工具必須能跨越不同的作業平台，因此選用了 Java 語言，利用其跨平台的特性，達到解碼程式可下載的能力。

3.6 Java applet decoder 的執行狀況

由於利用 Java 跨平台的特性，只要有支援 Java 的網頁瀏覽器皆可執行本解碼器如圖 4。

使用者加強	Pentium-200 畫面/每秒	Sun Ultra Sparc 畫面/每秒
No	1.1	1.2
加強前景	0.6	0.7
加強背景	0.5	0.5
全部加強	0.4	0.4

表 1 Java applet 的執行速度

表 1 的為此解碼器的執行速度。測試的視訊物件，前景為 CLAIRE(128x128)，背景為 GARDEN(176x144)。測試時的使用者加強是先後在解析度和畫面數(sp+tm)加強。

4. 編碼程式之實作

本章將描述編碼程式之實作。MPEG-4 的視訊編碼程式可分為兩個部份。第一個部分是視訊物件編碼器，第二個部分是融合器(MUX)。

融合器主要的工作僅在將視訊物件編碼器所輸出的位元串列融合在一起。所以其核心程式在於第一部分的視訊物件解碼器，此解碼程式將會負責傳統的視訊編碼功能，包括可變長度碼編碼器，量化，離散餘弦轉換，以及位移向量計算等。除此之外，為了達成可調整性壓縮，也做了特定的安排，以下將實作上的架構予以說明。

4.1 架構

本實作的編碼器的輸入為已經分解出形狀的視訊物件和在畫面上所在位置，所以在圖 5 的 VO Definition 表示前處理過後只需要定義物件的識別碼，而每一個視訊物件解碼器將會把視訊物件分為一個基礎層及三個加強層。最後融合器扮演最後輸出位元串列的角色。

4.2 可調整性壓縮

本壓縮程式與以往視訊壓縮程式最大不同是可輸入數個視訊，將它們看做不同的視訊物件，予以個別獨立之壓縮，除此之外又加上可調整性壓縮，使用者或解碼器可依其需要解碼，以下將說明如何完成此壓縮法。

1. 空間上的可調整性(Spatial Scalability)

空間上的可調整性可以改變某一個視訊物件的解析度，其做法如圖 6。

最低解析度的當做基礎層，以上都是加強層，如此分層即可達到不同解析度的壓縮，其在位元串列上的視訊物件平面安排可如圖 7 圖 8 兩種方式。

目前實作上是採用放置 P-VOP 於加強層的方法，這種方法不但在壓縮端易處理外，在解壓縮端也較方便處理，且又可使解壓縮端隨意的解加強層，不必解壓完整的加強層。此法缺點是壓縮倍率沒有 B-VOP 高。

2. 時間上的可調整性(Temporal Scalability)

時間上的可調整性可以改變某一個視訊物件的播放速率，如同空間上的可調整性其視訊物件平面的安排可如圖 9 圖 10 兩種方式：

目前實作上也是和空間上的可調整性一樣採用加強層是 P-VOP 的方法並且只放一張 P-VOP，時間上的可調整性不必如空間上的可調整性做縮放影像大小的工作，僅將某些時間的影像放到加強層。

4.3 軟體 MPEG-4 視訊物件編碼程式執行效率

實作上的比較 Quadtree 和 MMREAD 在二元形狀壓縮的壓縮倍率和執行速度：

用來測試的視訊物件，前景為 CLAIRES(128x128)，背景為 GARDEN(176x144)，個別有 113 個畫面。圖 11 僅取前 28 張畫面的情形，可以發現壓縮倍率 MMREAD 優於 Quadtree 兩倍以上。整體的效能如表 2。

	Quadtree	MMREAD
Bitstream Size(byte)	598,489	316,297
Frame rate	3.0	6.6

表 2 Quadtree 和 MMREAD 的比較

表 3 的測試環境為 Pentium-200 MMX。測試的視訊物件，前景為 CLAIRES(128x128)，背景為 GARDEN(176x144)。分別就加強層的數目比較編碼效能

	2 VO(1 Layer)	2 VO(4 Layer)
Bitstream size (bytes)	321,296	800,270
此位元串列可供使用者的互動程度	只能移動縮放物件	不但能移動縮放物件，還有三個加強層可達到 Scalability
畫面率	6.6	2.4
解碼畫面率	0.41	0.4

表 3 編碼程式之編碼效能

5. 結論

就 MPEG-4 的目標而言，是為多媒體應用提供一個整合的標準。比如，虛擬實境(VRML)，合成音樂(MIDI)，以及自然影像及電腦圖學兩者的合成 [15]。所以其應用不限定於幾個固定的應用，而是整個多媒體應用都將兼顧。因此，就其廣度方面將是多媒體應用一個相當重要的標準。同時，在本文中實作出一能於網際網路上執行的類 MPEG-4 視訊編碼器及一 Java-based 的視訊解碼器雛型，證實了 MPEG4 強大的功能及可行性，若未來能加速 Java 程式執行將更能揮其威力。

參考資料

[1] ISO/IEC JTC1 CD 11172, "Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to about 1.5 Mbits/s," December 1991
 [2] ISO/IEC Draft International Standard 13818, "Information Technology-Generic Coding of Moving Pictures and Associated Audio," March 1994
 [3] Draft ITU-T Recommendation H.263, "Video CODEC for low bitrate communication," October 1995

[4] ISO/IEC JTC1/SC29/WG11, "MPEG-4 Video Verification Model Version 2.1," May 1996
 [5] ISO/IEC JTC1/SC29/WG11, "MPEG-4 Video Verification Model Version 7.0," April 1997
 [6] ISO/IEC JTC1/SC29/WG11, "Working Draft 3.0 of ISO/IEC 14496-2," April 1997
 [7] Leonardo Chiariglione, "MPEG and Multimedia Communications," IEEE Trans. Circuits and System for Video Tech. Vol. 7, No.1, February 1997
 [8] Thomas Sikora, "The MPEG-4 Video Standard Verification Model," IEEE Trans. Circuits and System for Video Tech. Vol. 7, No.1, February 1997
 [9] Junavit Chalidabhongse and C.C. Jay Kuo, "Fast Motion Vector Estimation Using Multiresolution-Spatio-Temporal Correlations," IEEE Trans. Circuits and System for Video Tech., April 1996
 [10] 蕭登元, "The Realtime Implementaion of H.261 Video CODEC," 臺大資訊所碩士論文, June 1996
 [11] 翁詠泉, "Implementaion of Software MPEG-2 Video CODEC and Its Related Research," 臺大資訊所碩士論文, June 1996
 [12] Fenando Pereira, Rob Koenen, "Very low bit-rate audio-visual applications," Signal Processing: Image Communication, September 1994
 [13] Gertjan Keesman, Imran Shah, Rene Klein-Gunnewiek, "Bit-rate control for MPEG encoders," Signal Processing: Image Communication, December 1993
 [14] Jiro Katto, "Results of Core Experiment M1," ISO/IEC JTC1/SC29/WG11/1825, February 1997
 [15] Caspar Horne (editor), "SNHC Verification Model 3.0," JTC1/SC29/WG11 N1545, February 1997

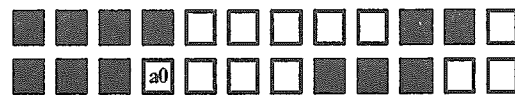


圖 1: a0 為原點，一開始從 16x16 區塊的左上角開始。a1 為從 a0 到下一個改變點(由 0 變 1，由 1 變 0)。b1 為從 a0 上一行的下一個位置開始的下一個改變點。

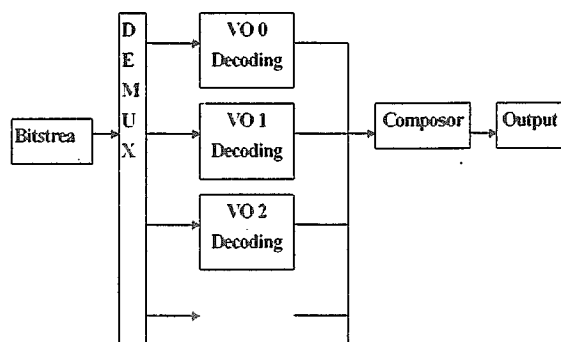


圖 2: 解碼器架構

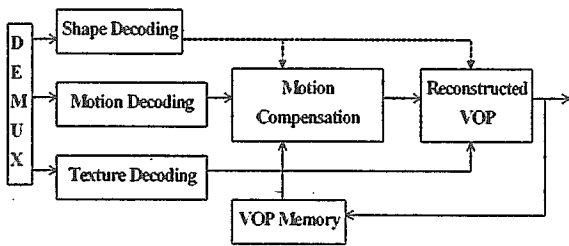


圖 3 Video Object Decoder

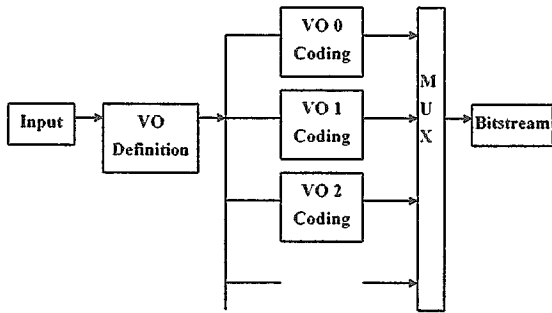


圖 5 編碼器架構

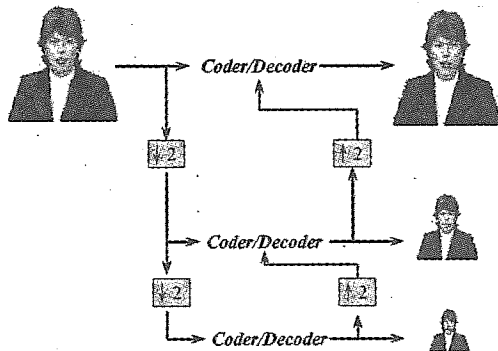


圖 6: Spatial Scalability

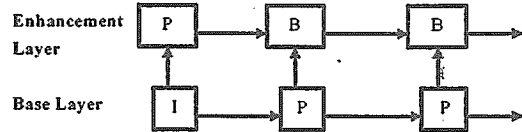


圖 7: 使用 B-VOP 放置於加強層

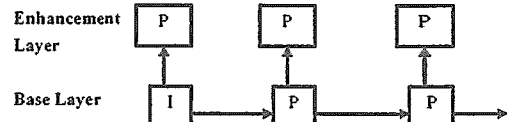


圖 8: 使用 P-VOP 放置於加強層

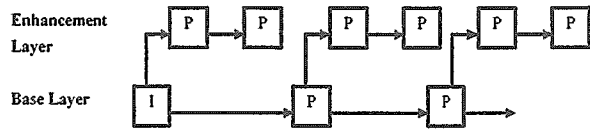


圖 9: 使用 P-VOP 放置於加強層

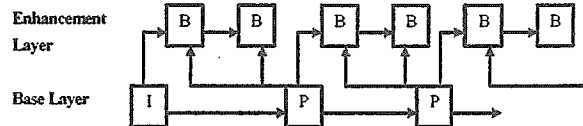


圖 10: 使用 B-VOP 放置於加強層

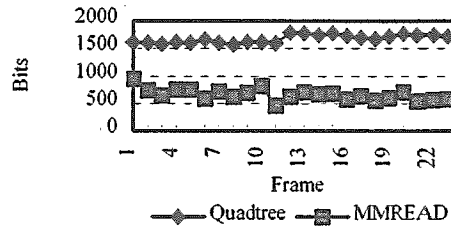


圖 11 Quadtree 和 MMREAD 在二元形狀壓縮的壓縮倍率



圖 4: 利用網頁瀏覽器執行解碼器的狀況。使用者可直接拉動物件，擺在想要的位置，並藉由左邊的按鈕，改變物件大小，物件解析度(SP)，以及物件畫面數(TM)。