

發明專利說明書

(本說明書格式、順序及粗體字，請勿任意更動，※記號部分請勿填寫)

※申請案號：94127316

※申請日期：94.8.11

※IPC 分類：H04N 7/28

G06F 17/10 (2006.01)

一、發明名稱：(中文/英文)

利用隨機位元重排獲致精細度可縮放的執行上下文適性二進制算術編碼的方法

METHOD FOR PERFORMING CONTEXT ADAPTIVE BINARY ARITHMETIC CODING WITH STOCHASTIC BIT RESHUFFLING FOR FINE GRANULARITY SCALABILITY

二、申請人：(共 1 人)

姓名或名稱：(中文/英文)

國立交通大學

NATIONAL CHIAO TUNG UNIVERSITY

代表人：(中文/英文)

張俊彥/CHANG, CHUN-YEN

住居所或營業所地址：(中文/英文)

新竹市大學路 1001 號

1001 TA-HSUEH RD., HSINCHU, TAIWAN R.O.C.

國籍：(中文/英文)

中華民國/R.O.C

三、發明人：(共 3 人)

姓名：(中文/英文)

1. 彭文孝/PENG, WEN-HSIAO

2. 蔣迪豪/CHIANG, TIHAO

3. 杭學鳴/HANG, HSUEH-MING

國 籍：(中文/英文)

1. ~ 3. 中華民國/R.O.C

四、聲明事項：

主張專利法第二十二條第二項 第一款或 第二款規定之事實，其事實發生日期為： 年 月 日。

申請前已向下列國家（地區）申請專利：

【格式請依：受理國家（地區）、申請日、申請案號 順序註記】

有主張專利法第二十七條第一項國際優先權：

美國 2005.06.21 11/158,034

無主張專利法第二十七條第一項國際優先權：

主張專利法第二十九條第一項國內優先權：

【格式請依：申請日、申請案號 順序註記】

主張專利法第三十條生物材料：

須寄存生物材料者：

國內生物材料 【格式請依：寄存機構、日期、號碼 順序註記】

國外生物材料 【格式請依：寄存國家、機構、日期、號碼 順序註記】

不須寄存生物材料者：

所屬技術領域中具有通常知識者易於獲得時，不須寄存。

國 籍：(中文/英文)

1. ~ 3. 中華民國/R.O.C

四、聲明事項：

主張專利法第二十二條第二項 第一款或 第二款規定之事實，其事實發生日期為： 年 月 日。

申請前已向下列國家（地區）申請專利：

【格式請依：受理國家（地區）、申請日、申請案號 順序註記】

有主張專利法第二十七條第一項國際優先權：

美國 2005.06.21 11/158,034

無主張專利法第二十七條第一項國際優先權：

主張專利法第二十九條第一項國內優先權：

【格式請依：申請日、申請案號 順序註記】

主張專利法第三十條生物材料：

須寄存生物材料者：

國內生物材料 【格式請依：寄存機構、日期、號碼 順序註記】

國外生物材料 【格式請依：寄存國家、機構、日期、號碼 順序註記】

不須寄存生物材料者：

所屬技術領域中具有通常知識者易於獲得時，不須寄存。

五、中文發明摘要：

本發明揭露一種利用隨機位元重排架構以執行上下文為本的二進制算術編碼的方法，其在改善基於 MPEG-4 精細度可縮放 (fine granularity scalability, FGS) 的位元平面編碼。該方法包括下列步驟：以 MPEG-4 AVC (Advanced Video Coding) 中的 4×4 整數轉換係數取代 8×8 DCT；將各轉換係數分割為有效位元 (significant bit) 與精化位元 (refinement bit)；根據轉換區塊中的能量分布與鄰近區塊的空間相關性，設定有效位元上下文；利用一估測的拉普拉斯分布 (Laplacian distribution) 導出精化位元的編碼機率；以及利用位元平面之間的上下文來分割各有效位元平面，以節省側邊資訊位元。

六、英文發明摘要：

The disclosure relates to a method for performing context based binary arithmetic coding with a stochastic bit-reshuffling scheme in order to improve MPEG-4 fine granularity scalability (FGS) based bit-plane coding. The method comprises steps of: replacing 8 x 8 DCT with 4 x 4 integer transform coefficient in MPEG-4 AVC (Advance Video-Coding); partitioning each transform coefficient into significant bit and refinement bit; setting up significant bit context based on energy distribution within a transform block and spatial correlation in adjacent blocks; using an estimated Laplacian distribution to derive coding probability for the refinement bit; and using the context across bit-planes to partition each significant bit-plane for saving side information bit.

七、指定代表圖：

(一)本案指定代表圖為：第 7 圖。

(二)本代表圖之元件符號簡單說明：

A1-AN、B1-BN、C1-CN、D3：暫存器

八、本案若有化學式時，請揭示最能顯示發明特徵的化學式：

九、發明說明：

【發明所屬之技術領域】

本發明係關於一種利用隨機位元重排獲致精細度可縮放的執行上下文適性二進制算術編碼的方法，特別是關於一種利用隨機位元重排架構、執行上下文為本的二進制算術編碼的方法，其目的在改善基於精細度可縮放(fine granularity scalability, FGS)的位元平面編碼。

【先前技術】

隨著網際網路和無線通道上多媒體應用的快速成長，可縮放視訊編碼(scalable video coding, SVC)的重要性也與日俱增。在這些應用中，以變動頻寬經由容易出錯通道傳送的視頻資訊，在不同的網路中係由各種的設備所利用。為了在異質環境中，能滿足多媒體的各種應用，MPEG-4 委員會已制定了精細度可縮放規格(Fine Granularity Scalability, FGS)，詳細可參看 W. Li, "Overview of Fine Granularity Scalability in MPEG-4 standard," *IEEE Trans, Circuits Syst. Video Technol.*, vol. 11, no. 3, pp. 301-317, 2001，該規格提供一種在層方式中基於離散餘弦轉換(Discrete Cosine Transformation, DCT)的可縮放方法。該基本層以一種非可縮放 MPEG-4 先進簡易設定檔(advanced simple profile, ASP)來編碼，而其加強層係以嵌入式位元平面編碼來作層內編碼，以獲致精細度可縮放。類似的位元平面編碼架構，也被採用於最近的 MPEG-4 Part 10 Amd. 1 中的可縮放視頻編碼標準(Scalable Video Coding Standard)內。

在目前的位元平面編碼，加強層圖框首先被施以 8×8 DCT 轉換。爲了提供精細度可縮放，轉換係數係以一個位元平面接一個位元平面的方式被編碼。從最有效位元平面 (most significant bit-plane) 到最低有效位元平面 (least significant bit-plane)，DCT 區塊的編碼被排序爲光柵掃描的順序。一 DCT 區塊的每一位元平面，被獨立地用 <RUN, EOP> 符號表示且依霍夫曼表 (Huffman tables) 來編碼。然而，目前的編碼方式擺出二個問題點：

a. 低劣的編碼效率

第一個問題點是低劣的編碼效率。此問題點來自三個因素：第一是帶有不同資訊加權的各位元，被以無差別的方式編碼。第二是存在於位元平面之間以及空間相鄰區塊之間的相關性，沒有被利用。最後是，霍夫曼編碼 (Huffman coding) 不能有效地與統計的變化相匹配。在現有的解決方案中，一位元平面內所有位元的不同加權係以 <Run, EOP> 符號被連帶地分組。此外，一區塊中每一位元平面的編碼是不相關的。更進一步，各轉換區塊係獨立編碼的。這些因素加總起來造成低劣的編碼效率。

b. 低劣的主觀品質

第二個問題點是當加強層被部份解碼時，確定性光柵掃描引起的品質差異。目前對各位元平面，MPEG-4 FGS 執行編碼係以一區塊接一區塊方式進行的。所有區塊係依光柵掃描而排序的。一區塊的編碼，只有在前一區塊的編碼完成後才能進行。當加強層被部份解碼時，此種順序可能只利用一

額外的位元平面對被解碼圖框的上層部份作精化。不均勻的精化引起主觀品質的低下。

有些研究者對於 DCT 為本的位元平面編碼提出了改良編碼效率的方案。現存的解決方案，例如 N.K. Laurance and D. M. Monro, “Embedded DCT Coding With Significance Masking”. *IEEE Int'l Conference on Acoustics, Speech, and Signal Processing*, 1997 以及 D. Nister and C. Christopoulos, “An Embedded DCT-Based Still Image Coding Algorithm”, *IEEE Int'l Conference on Acoustics, Speech, and Signal Processing*, 1998 所提出者，皆提到基於上下文的二進制算術編碼，用以達到有效的 DCT 為本的位元平面編碼。前者，亦即 N.K. Laurance 等所提出的，係利用 DCT 轉換的能量分布來改進編碼效率。此外，後者的 D. Nister 等則進一步在設計他們的上下文時考慮存在於相鄰區塊的共處係數 (co-located coefficients) 的空間相關性。然而，這些先前技術沒有考慮位元平面之間的相關性。故尚存在有可進一步改良編碼效率的空間。此外，這些先前技術仍然在各位元平面編碼時使用光柵掃描順序。不均勻品質分布的問題點，依然存在。

進而，在 W. H. Peng, C. N. Wang, T. Chiang and H. M. Huang, “Context-based binary arithmetic coding with stochastic bit reshuffling for FGS”, *IEEE Int'l Symposium on Circuit and System*, Vancouver, May 2004 的文件中，也提到這領域的一個先前技術。然而，此論文中的位元重排架

構限制在相同的位元平面作重新排序。

此外，有幾件專利揭露利用位元重排架構與係數重新排序架構以改良編碼效率的技術，例如美國專利公開號碼 2003/0133499, Y. Chen et al., “Fine granular scalable video with embedded DCT coding of the enhancement layer”；美國專利號碼 6,275,531, W. Li, “Scalable video coding method and apparatus”；美國專利號碼 6,625,321, J. Li et al., “Embedded image coder with rate-distortion optimization”；美國專利公開號碼 2004/00177949, W. Lin et al., “Apparatus and method for performing bitplane coding with reordering in a fine granularity scalability coding system”；以及美國專利號碼 6,501,397, Radha et al., “Bit-plane dependent signal compression”。但是，這些技術係在編碼前執行 8×8 DCT 因而需要複雜的浮點算術運算，以及他們的編碼流程是由 MSB(Most Significant Bit)位元平面開始到 LSB(Least Significant Bit)位元平面，因而沒有考慮爲了內容覺察重排(content aware reshuffling)的訊息率失真(rate-distortion)資料的更新。

因此，很有需要開發一種編碼方法既可以簡化操作，又可在位元編碼順序容許更多的彈性，進而可改良 FGS 爲本的位元平面編碼的編碼效率。

【發明內容】

針對前述問題點，本發明之目的在提供一種利用隨機位元重排獲致精細度可縮放的執行上下文適性二進制算術編

碼的方法，以改良基於 FGS 的位元平面編碼的編碼效率與主觀品質。

為達到此目的，根據本發明，其在提供一種利用隨機位元重排獲致精細度可縮放的執行上下文適性二進制算術編碼的方法，包括下列步驟：以 MPEG-4 AVC(Advanced Video Coding，亦即 H.264 規格)中的 4×4 整數轉換係數取代 8×8 DCT；將各轉換係數分割為有效位元(significant bit)與精化位元(refinement bit)；根據轉換區塊中的能量分布與鄰近區塊的空間相關性，設定有效位元上下文；利用估測的拉普拉斯分布(Laplacian distribution)導出精化位元的編碼機率；以及利用位元平面之間的上下文來分割各有效位元平面以節省側邊資訊位元(side information bit)。

再者，根據上述本發明之方法，該利用位元平面之間的上下文來分割各有效位元平面以節省側邊資訊位元的步驟，其包括使用較高位元平面的 EOSP(End-Of-Significant-Bit-Plane，即「有效位元平面終點」)位置將各有效位元平面分割為二部份以節省側邊資訊位元。

又者，根據上述本發明之方法，再包括下列步驟：由估測訊息率失真(rate-distortion)來決定各位元的編碼順序，其中該所有編碼位元依照估測失真縮減量與估測位元率二者的比值而以降幂排序。

又者，根據上述本發明之方法，該各位元的估測訊息率失真，係由利用離散拉普拉斯分布去模型化轉換係數而求得。

又再者，根據上述本發明之方法，再包括下列步驟：利用二進制熵來對位元率估測值作編碼；以及利用最大似然原理 (maximum likelihood principle) 以提供拉普拉斯分布參數的估測。

此外，根據上述本發明之方法，再包括下列步驟：將一動態編碼流程利用於隨機位元重排。

本發明的前述以及其他目的、特徵以及優點，將可由最佳實施例的詳細說明而顯得更加清晰，該詳細說明會參看伴隨的圖式來進行。

【實施方式】

A. 術語

第 1 圖定義一些我們所提到的術語例如「一圖框的 MSB 位元平面」、「一區塊的 MSB 位元平面」以及「一係數的 MSB」。為簡單起見，我們假設整個加強層圖框有 3 個轉換區塊，且每一區塊有 4 個係數。垂直軸表示轉換係數的二進位數。一係數的 MSB 位元表示其最有效位元 (most significant bit)。一圖框的 MSB 位元平面表示該一包含有整個圖框內最大係數的 MSB 位元者。另一方面，一區塊的 MSB 位元平面表示該一包含有一區塊內最大係數的 MSB 位元者。

B. 位元分類與位元平面分割

各轉換係數的所有位元被分割為 3 種型式包括：有效位元、精化位元及符號位元。從一區塊的 MSB 位元平面到 LSB 位元平面，一係數的有效位元係指先前於 (且包括) 其 MSB 位元的那些位元。另一方面，精化位元係指那些在 MSB 之後

的位元。符號位元則表示一係數的符號。

此外，對一轉換區塊的各位元平面，我們提出二個側邊資訊符號 (side information symbol)。他們是 End-Of-Significant-Bit-Plane (EOSP) 與 Part_II_ALL_ZERO。該 EOSP 係在一非零有效位元之後被編碼，以指出一位元平面的有效位元編碼的終了。爲了極小化 EOSP 位元數，我們根據前一位元平面的最後非零有效位元、稱爲 **LastS**，而將一位元平面的有效位元分割爲二部份。第一部份 (Part I) 指在 **LastS** 之前的有效位元的群組，係以鋸齒狀排序；而第二部份 (Part II) 則涵蓋其餘的有效位元。第 2 圖表示我們的有效位元平面分割的一個範例。第 2 圖中每一行顯示出轉換係數的二進制表示。由上到下，每一列代表一位元平面。爲了位元分割，位元平面 2 的 **LastS** 是在 AC3。對位元平面 2 而言，在 AC3 之前的有效位元被歸類爲 Part I，其餘的有效位元則被歸類爲 Part II。在我們的演算法中，我們以傳送 Part II 內非零有效位元之後的 EOSP 符號，來節省 EOSP 位元。

至若目前編碼位元平面的最後非零有效位元、實際上是在 **LastS** 之前發生，我們需要將 Part II 內的所有零的有效位元傳送出去。爲避免對這些冗餘位元作解碼，我們用一 Part_II_ALL_ZERO 符號通知解碼器關於所有的零位元。特別是，該符號在各位元平面的 Part II 編碼之前就要傳送。

表 1. 計算 MSB_REACHED 符號的上下文指數的程序

```

//Summation of MSB REACHED status in the nearest four adjacent blocks ContextIndex=
MSB_REACHED[UpperBlock]+MSB_REACHED[BottomBlock]+MSB_REACHED
[LeftBlock]+MSB_REACHED[RightBlock]

//UpperLeft to BottomRight diagonal direction
IF (MSB_REACHED[UpperLeftBlock]=1 && MSB_REACHED[BottomRightBlock]=1)
    ContextIndex+=2*(MSB_REACHED[UpperLeftBlock]
                    +MSB_REACHED[BottomRightBlock])
ELSE
    ContextIndex+=(MSB_REACHED[UpperLeftBlock]
                  +MSB_REACHED[BottomRightBlock])

//UpperRight to BottomLeft diagonal direction
IF (MSB_REACHED[UpperRightBlock]=1 && MSB_REACHED[BottomLeftBlock]=1)
    ContextIndex+=2*(MSB_REACHED[UpperRightBlock]
                    +MSB_REACHED[BottomLeftBlock])
ELSE
    ContextIndex+=(MSB_REACHED[UpperRightBlock]
                  +MSB_REACHED[BottomLeftBlock])

//Truncation
IF (ContextIndex>=4)
    ContextIndex=4

```

表 2. 有效位元的上下文指數

指數 (Index)	說明 (Description)	範圍 (Range)
Run	編碼位元與前一非零有效位元之間以鋸齒狀順序的距離	0~7
有效狀態的加權總和	計算程序與表 1 同，但我們用共處係數的有效狀態取代 MSB_REACHED	0~4
頻帶位置	編碼位元的鋸齒狀指數	0~10

C. 上下文設計

1. MSB_REACHED

MSB_REACHED 符號是一個側邊資訊位元用以指出一區塊的 MSB 位元平面是否已到達。至於 MSB_REACHED 符號的上下文指數 (context index) 計算，我們須參考鄰近 9 個區塊的 MSB_REACHED 狀態。表 1 列出我們的詳細計算程序。

2. 有效位元

表 2 列出對一有效位元編碼時我們所參考的上下文 (context)。特別是，為計算表 2 中的加權總和指數 (weighted summation index) 時，我們用共處係數的有效狀態來取代表 1 中的 MSB_REACHED，然後再依循表 1 中計算指數相同的程序。此外，為了在成本與性能之間作取捨，每一個被參考到的上下文指數值，皆被適當地截尾。第 3 圖表示我們的有效位元上下文模式的一個例子，其中 “Run” 的值是 1、有效狀態的加權總和是 4、而頻帶位置是 8。

3. 精化位元

我們利用估測拉普拉斯模型 (estimated Laplacian model) 來導出精化位元的編碼機率。再沒有附加的上下文模

型被創造以供使用於精化位元。正如後面將會提到，一精化位元的機率模型，可由估測拉普拉斯模型導出。

4. 符號位元

對於符號位元，我們利用一相等的機率模型。

表 3. EOSP 位元的上下文指數

指數 (Index)	說明 (Description)	範圍 (Range)
EOSP 偏移	編碼位元與預測 EOSP 位置之間以鋸齒狀順序的偏移	-7 ~ 7
一區塊的位元平面指數	從一區塊的 MSB 位元平面到 LSB 位元平面，位元平面指數是由零以 1 作增量	0 ~ 4

5. End-Of-Significant-Bit-Plane (EOSP, 有效位元平面終點)

表 3 彙總我們所參考 EOSP 位元的上下文模型。首先，我們以最靠近 4 個區塊內的 EOSP 方位 (position) 的平均值來預測 EOSP 位置 (location)。利用該預測位置，我們將目前非零有效位元與預測 EOSP 位置之間的偏移 (offset) 定義為 EOSP 位元的上下文。注意該偏移可能是一個負值。此外，我們也把位元平面包括在參考上下文的一部份。從一區塊的 MSB 位元平面到 LSB 位元平面，位元平面指數是由零以 1 作增量。為節省記憶體，位元平面指數大於 4 的要被截尾。

6. Part_II_ALL_Zero

我們取一區塊的位元平面指數當作 Part_II_ALL_Zero 的上下文模型。從一區塊的 MSB 位元平面到 LSB 位元平面，位元平面指數是由零以 1 作增量。為節省記憶體，位元平面指數大於 4 的要被截尾。

表 4. 被提案的、以光柵掃描編碼順序的 CABAC 演算法

Nblock: Number of total 4x4 blocks in a frame
DC0, AC1~45 : DC and AC coefficients in zigzag scan
<pre> MAIN: FOR bit-plane=MSB to LSB of the entire frame FOR 4x4_Block_Index=1 to Nblock (Line 2) FOR Coeff_Index=DC0 to AC15 (Line 3) IF (MSB_REACHED[4x4_Block_Index]) IF (Coding Bit == Significant Bit) Encode/Decode_Significant/Sign/EOSP_Bit() ELSE Encode/Decode_One_Refinement_Bit() ELSE IF (Coeff_Index==DC0) Encode/Decode_One_MSB_REACHED_Bit() IF (MSB_REACHED[4x4 Block Index]) Encode/Decode_Significant/Sign/EOSP_Bit() SubRoutine: Encode/Decode_Significant/Sign/EOSP_Bit() IF (Is Part II significant bit) IF (Is first Part II significant bit) Encode/Decode_Part_II_ALL_Zero_Bit() IF (Part II not all zero) Encode/Decode_One_Significant_Bit() IF (non-zero significant bit) Encode/Decode_One_Sign_Bit() Encode/Decode_One_EOSP_Bit() ELSE Encode/Decode_One_Significant_Bit() IF (non-zero significant bit) Encode/Decode_One_Sign_Bit() </pre>

D. 利用確定性光柵掃描的編碼流程

表 4 顯示我們所提案的利用光柵掃描順序的位元平面編碼的虛擬碼。對各加強層圖框，我們從 MSB 開始作位元平面編碼一直到 LSB 為止。對每一位元平面，編碼係以一區塊接一區塊的方式進行。在每一區塊內，所有係數係以鋸齒狀的順序被造訪。在各區塊編碼之前，MSB_REACHED 符號首先被編碼用以告知目前區塊是否已到達其 MSB 位元。當 MSB_REACHED 為真，我們即啟動位元平面編碼。在編碼過程中，不同位元型式係作不同的編碼。若輸入位元被認為是一有效位元，我們將檢查目前有效位元所屬的分割部份。對於有效位元屬於 Part I，只要將該有效位元與符號位元一起編碼即可。對於有效位元屬於 Part II，在符號位元之後我們要追加一 EOSP 位元的編碼。特別是，在每一個 Part II 有效位元編碼開始之前，我們會對一 Part_II_ALL_Zero 位元作編碼。當 Part_II_ALL_Zero 或 EOSP 為真，我們即停止目前區塊的編碼而前進到次一區塊。若輸入位元是一精化位元，我們只要將該精化位元本身編碼即可。從實驗結果，我們發現和 MPEG-4 FGS 方式比較有 0.5~1dB PSNR 的改良。比較詳細的實驗結果，將在後面呈現出來。

E. 隨機位元重排

為改善主觀品質，一種隨機位元重排架構被提案出來。我們依一隨機訊息率失真感測值來執行位元重排。我們對每一位元指定二個估測因子即：(1)平方誤差縮減量、 ΔD ，與(2)編碼成本、 ΔR 。平方誤差縮減量表示對平方誤差改良的

貢獻度，而編碼成本表示所需的位元率。對於各位元已知的該二個因子，方程式(1)定義了我們的編碼順序判別式，其中記號 $\hat{E}()$ 表示所取的估測值，上標表示位元識別，下標表示編碼順序。簡而言之，我們將所有編碼位元依 ΔD 估測值對 ΔR 估測值的比率重新排列為降冪順序。

$$\frac{\hat{E}(\Delta D^i_1)}{\hat{E}(\Delta R^i_1)} > \frac{\hat{E}(\Delta D^j_2)}{\hat{E}(\Delta R^j_2)} > \frac{\hat{E}(\Delta D^k_3)}{\hat{E}(\Delta R^k_3)} > \dots > \frac{\hat{E}(\Delta D^l_N)}{\hat{E}(\Delta R^l_N)} \quad (1)$$

F. 參數估測

1. 離散拉普拉斯分布

爲了估測每一位元的 ΔD 與 ΔR ，我們求助於他們的期望值。爲了計算期望值，我們需要轉換係數的機率分布。由於解碼器並沒有準確的機率分布，我們採用一種其可將解碼器負擔作極小化的模型。特別是，在加強層我們使用 MPEG-4 AVC/H.264 中的 4×4 整數轉換。對每一個 4×4 轉換係數，我們將其模型化爲一具有拉普拉斯分布的隨機變數。方程式(2)係將離散拉普拉斯分布的機率模型做公式化而得到的結果，其中 C_n 表示第 n 個鋸齒順序的係數， k_n 表示其結果，而 α_n 是待決定的拉普拉斯參數。

$$P(C_n = k_n) = \frac{1 - \alpha_n}{1 + \alpha_n} * \alpha_n^{|k_n|} \quad (2)$$

2. 拉普拉斯參數的估測

方程式(3)表示我們所設計之拉普拉斯參數的最大似然估測器(maximum likelihood estimator)。如式中所示，爲了估測 α_n ，我們首先計算在第 n 個鋸齒位置的共處係數絕對值的平均值，然後將其代入方程式(3)即可。在我們的方式

中，估測係在編碼器進行然後傳送到解碼器。對於 4×4 整數轉換，我們只需要 16 個參數用於亮度成份，以及另外的 16 個參數用於色訊部份。

$$\alpha_n = \frac{-\frac{1}{\text{AvgK}} + \sqrt{\left(\frac{1}{\text{AvgK}}\right)^2 + 4}}{2} \quad \text{其中} \quad \text{AvgK} = \frac{\sum_{j=1}^M |k_n^j|}{M} \quad (3)$$

3. ΔD 的估測

我們由期望的平方誤差縮減量來估測每一係數位元的 ΔD 值。第 4 圖中，我們出示一係數的預定義拉普拉斯分布並以二個範例來說明 ΔD 估測。在左手邊，我們出示有效位元的範例，其中該係數的 MSB 被編碼為零。在右手邊，我們出示精化位元的情形，其中該係數的 MSB 被編碼為非零。

從已編碼的 MSB 位元，我們可分辨該存放實際數值的不確定區間。對於有效位元之場合，我們知道實際值是在 MSB 編碼之後的區間 B。同樣地，對於精化位元之場合，實際值被放置於區間 A。已知這些區間，每一係數位元的編碼可進一步減少不確定區間。例如在第 4 圖中，要被編碼的有效位元即可決定實際值是在區間 B1+、B0 或 B1-。同樣地，要被編碼的精化位元即可決定實際值是在區間 A1 或 A0。

不確定區間的減少即代表期望平方誤差的減少。對各區間，我們定義期望平方誤差為該區間內的變異數 (variance)。亦即，我們對每一位元的 ΔD 估測，可寫為變異數的縮減量。方程式 (4) 即定義了我們對第 4 圖中有效位元範例的 ΔD 估測、 $\hat{E}(\Delta D)$ ，其中 $E(X|Y)$ 代表在已知 Y 條件下的 X 的條件期望值；且 $P(X|Y)$ 表示一條件機率。在方程式 (4)

中，第 1 項表示在區間 B 內的變異數，而三個減項分別表示在區間 B1+、B1-及 B0 的加權變異數。由於我們在編碼前不能決定實際值放在哪一區間，故每一減項的加權因數表示各該子區間的機率。依循相同的程序，所有其他位元的 ΔD 估測皆可被算出。方程式 (5) 定義了第 4 圖中精化位元的 ΔD 估測。

為使我們的有效位元重排具有內容覺察 (content aware)，我們將方程式 (4) 中的子區間機率用上下文機率模型來取代。方程式 (6) 定義了我們對一有效位元的內容覺察 ΔD 估測。在方程式 (6) 中， $P_{ctx}()$ 表示一有效位元關連的上下文機率。

由於我們沒有對精化位元發展出任何的上下文模型，故對一精化位元的 ΔD 估測，我們是利用方程式 (5)。

$$\begin{aligned} \hat{E}(\Delta D) \text{ of significant bit} \equiv & \\ & E(C_n^2 | C_n \in B) \\ & - P(C_n \in B1+ | C_n \in B) * [E(C_n^2 | C_n \in B1+) - E^2(C_n | C_n \in B1+)] \\ & - P(C_n \in B1- | C_n \in B) * [E(C_n^2 | C_n \in B1-) - E^2(C_n | C_n \in B1-)] \\ & - P(C_n \in B0 | C_n \in B) * [E(C_n^2 | C_n \in B0)] \end{aligned} \quad (4)$$

$$\begin{aligned} \hat{E}(\Delta D) \text{ of refinement bit} \equiv & \\ & [E(C_n^2 | C_n \in A) - E^2(C_n | C_n \in A)] \\ & - P(C_n \in A1 | C_n \in A) * [E(C_n^2 | C_n \in A1) - E^2(C_n | C_n \in A1)] \\ & - P(C_n \in A0 | C_n \in A) * [E(C_n^2 | C_n \in A0) - E^2(C_n | C_n \in A0)] \end{aligned} \quad (5)$$

$$\begin{aligned} \text{Improved } \hat{E}(\Delta D) \text{ of significant bit} \equiv & \\ & E(C_n^2 | C_n \in B) \\ & - P_{ctx}(1) * [E(C_n^2 | C_n \in B1+) - E^2(C_n | C_n \in B1+)] \\ & - P_{ctx}(0) * [E(C_n^2 | C_n \in B0)] \end{aligned} \quad (6)$$

4. ΔR 的估測

為要估測 ΔR ，我們利用方程式 (7) 所定義的二進熵

(binary entropy), 其中 $P(1)$ 表示一輸入位元的非零機率。

$$H_b(P(1)) \equiv -P(1) \cdot \log_2 P(1) - (1-P(1)) \cdot \log_2(1-P(1)) \quad (7)$$

在方程式(8)中，我們出示一有效位元的 ΔR 估測。第一項表示利用關連上下文機率作為引數(argument)而求出的一有效位元的二進熵，而第二項表示一符號位元的估測成本。在方程式(8)中，我們假設平均每一符號位元消耗掉一個位元。此外，符號位元的成本係由有效位元的非零機率來予以加權的。特別是，非零機率係由關連上下文機率模型而導出的。

在方程式(9)中，我們出示一精化位元的 ΔR 估測。由於精化位元沒有特別的上下文機率模型，在二進熵計算時我們使用由拉普拉斯模型所導出來的非零機率。特別是，非零機率即是對應子區間的機率。例如，在第4圖中的精化位元的例子， $P_{model}(1)$ 即等於 $P(A1|A)$ 。

$$\hat{E}(\Delta R) \text{ of significant bit} \equiv H_b(P_{ctx}(1)) + P_{ctx}(1) * 1 \quad (8)$$

$$\hat{E}(\Delta R) \text{ of refinement bit} \equiv H_b(P_{model}(1)) \quad (9)$$

G. 隨機位元重排的編碼流程

為達到隨機訊息率失真的最佳化，編碼器與解碼器兩方皆須實施兩個動態編碼串列以保持編碼的優先度。一個是為有效位元，另一個是為精化位元。該串列中的每一位元被配置一暫存器用以記錄位元位置、位元型式、編碼上下文以及估測訊息率失真資料等。

在我們的演算法中，對於編碼順序我們擺明有兩個限

制，以避免對額外側邊位元與冗餘位元的編碼。這兩個限制是：(1)對每一係數，編碼是從 MSB 到 LSB 依序執行，及(2)對一轉換區塊的每一位元平面，Part II 中有效位元的編碼順序總是跟隨鋸齒狀掃描。

有了這些限制，第 5 圖顯示出我們編碼流程的概觀。為簡單起見，我們只畫出有效位元串列。精化位元的流程，除了不必更新其訊息率失真資料外，其餘是相同的。開始時，在初始化步驟將一加強層圖框的 MSB 位元平面的所有 DC 係數位元放在串列內。然後，我們按照估測訊息率失真資料，將串列內的位元作重排。由重排結果，我們檢索出最優先位元並執行二進制算術編碼。編碼後，我們對關連的上下文機率模型作更新。由於上下文機率模型已被更新且共處有效位元的上下文在編碼後會改變，我們須對串列中落伍的訊息率失真資料作更新。更新後，我們利用一仿擴散機制把更多位元包括進來以供重排。第 5 圖中的步驟(b)到(e)一直被重複，直到所有輸入位元皆被編碼為止。

表 5. 目前編碼位元與接著要被包括進來作重排的位元之間的連結

位元型式	編碼值	接著要被包括進入編碼串列的位元
part I 的有效位元	0	在較低位元平面的相同係數的有效位元 (part I)
	1	在較低位元平面的相同係數的精度化位元
part II 的有效位元	0	<ul style="list-style-type: none"> 在較低位元平面的相同係數的有效位元 (part I) 次一鋸齒狀係數 (在相同位元平面) 的有效位元 (part II)
	1 非 EOSP	<ul style="list-style-type: none"> 在較低位元平面的相同係數的精度化位元 次一鋸齒狀係數 (在相同位元平面) 的有效位元 (part II)
	1 EOSP	<ul style="list-style-type: none"> 在較低位元平面的相同係數的精度化位元 依鋸齒狀順序 (在較低位元平面) 的次一係數的有效位元 (part II)
精度化位元	0 或 1	在較低位元平面的相同係數的精度化位元

1. 訊息率失真資料的更新

更新估測訊息率失真資料，可以保證在估測時總是使用最新的上下文機率模型。特別是，我們只更新有效位元的訊息率失真資料。精度化位元的沒有被更新，因為他們是由估測拉普拉斯機率模型導出來的。而估測拉普拉斯機率模型在整個重排過程中是固定的。

在我們的演算法中，串列中的有效位元要被更新的有：

第 1 類：那些與目前編碼的位元使用相同上下文的有效位元。

第 2 類：那些與鄰近區塊具共處係數的有效位元。

第 3 類：在目前編碼位元之後的 Part I 有效位元(以鋸齒狀的順序)。

2. 將新位元包括進來

由於並非所有輸入位元可以被同時放入重排的串列，我們提出一種仿擴散架構(diffusion-like scheme)將更多位元包括進來以供重排與編碼。基本的構想是，首先將已編碼位元周圍的尚未編碼位元包括進來。表 5 定義了我們介於目前編碼位元與接著要被包括進來的位元之間的地圖規則。

第 6 圖顯示一個範例。依序從第 6(a)圖到第 6(d)圖，我們出示一轉換區塊的首先 4 個編碼狀態。為簡單起見，假設該轉換區塊只有 4 個係數。每一行代表一個係數的二進制表示，而每一列表示一位元平面。從左到右，這些係數是依鋸齒順序的。對於每一係數，空四邊形表示一個尚未編碼的位元。另一方面，有二進制數字的表示其編碼值。陰影四邊形代表目前編碼的位元，而虛線四邊形表示在串列內的位元。對於在串列內的位元，我們用 S_I 、 S_{II} 或 R 表示他們的位元型式。 S_I 表示 Part I 的有效位元而 S_{II} 表示 Part II 的有效位元。另一方面， R 代表精化位元。

第 6(a)圖表示在 BP3 上的 DC0 係數的位元被辨識為最優先位元且被編碼為零。在編碼之後，根據表 5 中的第 3 規則我們將 BP3 上的 AC1 的有效位元與 BP2 上的 DC0 的有效位元包括進入串列內。再經過重排，第 6(b)圖表示當 BP3 上的 AC1 的有效位元被編碼的情形。同樣地，我們再將 BP3

上的 AC2 的有效位元與 BP2 上的 AC1 的有效位元包括進入串列內。其後，第 6(c)圖表示 BP2 上的 AC1 的有效位元比 BP2 上的 DC0 的有效位元較早被編碼的情形。由定義，用在 AC1 的有效位元的“Run”上下文，其會依存於 DC0 的有效狀態。然而，只要是一直保有較好的估測訊息率失真性能，編碼時我們就容許使用非實際上下文。根據表 5 中的第 1 規則，在編碼後我們將 BP1 上的 AC1 的量化位元包括進來。最後，第 6(d)圖表示當 AC2 上的 MSB 是有最高優先度且被辨識為 BP3 的 EOSP。根據表 5 中的第 5 規則，我們再將 BP2 上的 AC2 的量化位元與 BP2 上的 AC3 的有效位元包括進入串列內。依從表 5 的規則，任何人可完成所有其他輸入位元的編碼。

H. 動態記憶體組織

更新與重排有效位元需要一些密集的計算。爲了降低計算量，我們提案一種動態記憶體組織架構來管理有效位元串列中的暫存器。我們觀察到並非該一串列中的所有有效位元皆需要更新與重排。因此，在我們的架構中只對那些有影響的位元重排或更新，而對其餘的保持不碰。

爲了快速識別那些與目前被編碼的位元使用相同上下文的有效位元，我們使用一連結串列將具有相同上下文的有效位元歸成一組。在第 7 圖的右手邊，我們所舉的例子以圓圈表示一有效位元對應的暫存器，而每一四邊形表示一個上下文群組。按照定義，暫存器 A3、B3、C3 與 D3 有相同的上下文。爲了識別每一組中的最優先位元，我們進一步

按照 $\hat{E}(\Delta D)/\hat{E}(\Delta R)$ 的比率而將暫存器重新排列。從左至右，優先度(即 $\hat{E}(\Delta D)/\hat{E}(\Delta R)$ 的比率)是依降冪排列。

利用我們的分組架構，該一串列中的最優先位元可由比較各群組中的最優先位元而求出。為執行此種比較，我們對各群組指定一上下文群組指標指到該一群組中的最優先位元。根據各群組的最優先位元而將上下文群組指標重新排序，即可識別出該一串列中的最優先位元。特別是，為保持該順序，我們利用一連結串列結構來實現該上下文群組指標。第 7 圖中，該等上下文群組指標係顯示在中央部份。從上到下，優先度是以降冪排列的，亦即 $A_3 > A_1 > A_5 > \dots > A_N$ 。在本例中， A_3 不僅是 C_{tx3} 群組的最優先位元，也是該一串列中的最優先位元。

在最優先位元被識別並編碼之後，我們必須更新那些上下文已改變的有效位元。由於編碼是動態的，我們無法事先決定哪些有效位元必須被更新。其結果是，每當一個非零有效位元被編碼，我們依從前節所定義的種類 2 與種類 3 來導出那些過時的有效位元的位元位置。根據所導出的位元位置，在編碼之前，我們以反轉已編碼位元的狀態而來重新計算他們的上下文。從過時的有效位元的上下文指數，我們可將搜尋限制在那些上下文群組。為避免對所有上下文群組指標作耗竭性的搜尋，我們建構一個圖形查表(look-up table)其以上下文指數為輸入，並產生對應的上下文群組指標當作輸出。第 7 圖中的左手邊顯示該一個圖形查表。實際上，該表係以指到上下文群組指標的一些指標所形成的矩陣來實

現的。

爲了說明整體的更新與再排序的流程，我們利用第 7 圖當作一個範例，其中我們假設自從初始化以來在每一群組中的有效位元與上下文群組指標已被重新排序過。亦即在實際編碼時，每當一有效位元被編碼時，我們僅單純地修改與重排一部份的暫存器。在本範例中，A3 是串列中的最優先位元。當 A3 被編碼時，B3 與 D3 要被更新，因爲他們與被編碼的位元有相同上下文。此外，第 7 圖中，我們假設在 A3 編碼之後，有二個有效位元其上下文已被改變。我們假設該二個位元的上下文在 A3 編碼之前分別是 Ctx5 與 Ctx2。另一方面，在 A3 編碼後該對應的上下文假設爲 Ctx4 與 Ctx6。由於該二個有效位元的上下文已經改變，我們利用最新上下文機率模型 (latest context probability models) 來更新暫存器，並將關連暫存器由一群組移至另一群組。編碼前的上下文指數用以鑑別上下文群組、再找出要被更新的暫存器。另一方面，編碼後的上下文指數用以決定我們要存放被修改暫存器之目的地群組。由所有這些上下文指數，我們利用圖形查表快速存取上下文群組以供檢索與更新。注意在一群組內的檢索係由比較位元的位址而完成的。如第 7 圖所示，暫存器 B5 被更新並被從 Ctx5 群組移到 Ctx4 群組的 B4 位置。同樣地，暫存器 A2 被更新並被從 Ctx2 群組移到 Ctx6 群組的 A6 位置。特別是，當一被修改的暫存器被放到一目的地群組時，其在連結串列內的位置係由更新過的訊息率失真資料來決定。例如，暫存器 B5 被移至 B4，因爲他們之間的優先

順序是 $A4 > B4(B5) > C4$ 。而在更新與移動之後，Ctx3 群組、Ctx2 群組以及 Ctx6 群組的最優先位元也已被改變。如此，我們重新決定這些群組在上下文群組指標連結串列中的順序。特別是，我們首先從連結串列內將那些群組指標移除。然後，我們經由比較更新過的最優先位元而依序將他們嵌回適當的位置。利用此種方法，我們可以使那些上下文沒有改變的群組，保持不被動到。在我們的實施例中，上下文群組指標的連結串列被實施為雙方向，如此我們可以隨機且快速地將任何一個節點從串列中移除。

I. 實驗結果

1. 客觀的品質比較

在我們的客觀品質比較中，我們使用 H.264 JM4 當作基本層以及 RFGS 當作加強層。特別是，我們使用 PSNR 作為客觀品質的量度。基準線系統使用 MPEG-4 FGS 為本且利用再訓練表的位元平面編碼，並跟隨同一的測試條件。第 8 圖表示各種的 PSNR 比較。如圖所示，我們提案的演算法 (proposed CABAC) 比基準線系統 (MPEG-4 VLC based bit-line coding) 平均改善 PSNR 約 0.5~1.5dB。

2. 主觀的品質比較

第 9 圖係表示各種不同方式的主觀品質比較。這些供比較的方式有：(a) MPEG-4 FGS 為本的位元平面編碼並利用光柵掃描，(b) 被提案的 CABAC 並利用光柵掃描，(c) 被提案的 CABAC 並利用在位元平面內的重排，以及 (d) 被提案的 CABAC 並利用本發明所顯示的加強型隨機重排架構。

如所預期，第 9(a)圖及第 9(b)圖顯示出確定性的光柵掃描方式有不均勻的品質分布。上半部提供了均勻的品質，但下半部卻有明顯的區塊反常。相對地，第 9(c)圖及第 9(d)圖顯示出整個圖框有更均勻的品質。特別是，第 9(d)圖顯示出加強型隨機位元重排比限制型的在位元平面內的重排，其所呈現的品質更佳。其他一序列的實驗結果，也顯示類似的性能。

J. 位元重排的應用

位元重排的觀念可依各種目的例如訊息率失真的最大化、主觀的品質改良以及關心區域(region-of-interest)的功能性而被擴展到對係數、區塊、區域或週期等層次的重排。此種已揭露的構想可採用不同的優先順序且執行各種不同調整的重排，以達到特定的目的。

在此，我們要將所揭露的位元重排擴展到訊息率失真的執行與對 MPEG-4 Part 10 Amd.1 SVC 的循環區塊編碼的主觀品質改良。第 10 圖表示 MPEG-4 Part 10 Amd.1 SVC 中的循環區塊編碼的一種符號表示。如圖所示，各轉換區塊係用 (EOB, Run, Level) 或編碼週期的一精化符號，而被利用同等編碼。較少非零係數的區塊，會在較多非零係數的區塊之前完成編碼。然而，從訊息率失真的最大化的預估，能量較多的區塊應被賦予較高的編碼優先度。所揭露的位元重排架構，可被擴展且被應用於區別不同符號、區塊或區域的重要性。

第 11 圖表示優先循環區塊編碼的一種實施形態。如圖

所示，不同區塊在一編碼週期中可能被施以不等的編碼。特別是，爲了避免傳送優先資訊(額外負擔)，編碼器與解碼器可參考到相同上下文以獲得區塊或符號的優先指定，此有如所揭露的參數估測架構要獲得位元重排資訊一樣。此外，優先循環區塊編碼也可用於關心區域的功能性。關心區域內的區塊，可以較高的優先度來編碼。特別是，優先資訊可被嵌入於位元流或由上下文參考值來暗示。

本發明揭露一種新奇的基於上下文的具有隨機位元重排架構的二進制算術編碼系統，其目的在對 MPEG-4 FGS 爲本的位元平面編碼的編碼效率與主觀品質兩方面皆作改良。本發明提出的架構，不僅可利用於 MPEG-4 FGS，也可使用於高等的 FGS 系統，例如由 H. C. Huang, C. N. Wang and T. Chiang, "A Robust Fine Granularity Scalability Using Trellis Based Predictive Leak," *IEEE Trans. on Circuits System for Video Technology*, vol. 12, no. 6, pp. 372-385, Jun, 2002 所提出的 Robust FGS (RFGS) 系統，以及由 F. Wu, S. Li, and Y. Q. Zhang, "A Framework for Efficient Progressive Fine Granularity Scalable Video Coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, pp. 332-344, 2001 所提出的 Progressive FGS (PFGS) 系統等等。尤有甚者，隨機位元重排的概念也可應用到其他的嵌入熵編碼系統(embedded entropy coding schemes)。

以上說明了本發明的最佳實施例，然而熟知本技術領域之人士明顯可知，在不脫離本發明之精神與專利範圍之情況

下，可對上述實施例作各種改變或修改而予以實施之。

【圖式簡單說明】

第 1 圖係在解說最有效位元平面與最有效位元的定義；

第 2 圖係表示根據本發明之有效位元平面分割的一個範例；

第 3 圖係表示根據本發明之有效位元之上下文指數的一個範例；

第 4 圖係表示離散拉普拉斯分佈與 ΔD 估測的一個範例；

第 5 圖係表示根據本發明之隨機訊息率失真最佳化之動態編碼流程的概觀；

第 6 圖係表示根據本發明之在轉換區塊內包含新位元的一個範例，其中首先 4 個編碼位元分別為：(a)在 BP3 的 DC0 係數的有效位元，(b)在 BP3 的 AC1 的有效位元，(c)在 BP2 的 AC1 的精度位元，以及 (d)在 BP3 的 AC2 的有效位元；

第 7 圖係表示根據本發明之隨機位元重排的動態記憶組織的概觀；

第 8 圖係表示 (a)Foreman，(b)Tempete，(c)Mobile，以及 (d)News 的 PSNR (客觀品質) 的比較情形，其中基本層的 $Q_p=38$ ，圖框速率 = 10 frames/s，圖框大小 = CIF(352×288)；

第 9 圖係表示主觀品質與 Foreman 序列的比較情形，其中基本層的 $Q_p=38$ ，圖框速率 = 10 frames/s 且加強層被截尾為 384 kbits/s；

第 10 圖係表示 MPEG-4 Part 10 Amd.1 SVC 中的循環區

塊編碼的一種符號表示；以及

第 11 圖係表示優先循環區塊編碼的一種實施形態。

【主要元件符號說明】

A1-AN、B1-BN、C1-CN、D3：暫存器

十、申請專利範圍：

1. 一種利用隨機位元重排獲致精細度可縮放的執行上下文適性二進制算術編碼的方法，包括下列步驟：

以 MPEG-4 AVC(Advanced Video Coding)中的 4×4 整數轉換係數取代 8×8 DCT；

將各轉換係數分割為有效位元(significant bit)與精化位元(refinement bit)；

根據轉換區塊中的能量分布與鄰近區塊的空間相關性，設定有效位元上下文；

利用估測的拉普拉斯分布(Laplacian distribution)導出精化位元的編碼機率；以及

利用位元平面之間的上下文來分割各有效位元平面以節省側邊資訊位元。

2. 如申請專利範圍第 1 項之方法，其中該利用位元平面之間的上下文來分割各有效位元平面以節省側邊資訊位元的步驟，其包括使用較高位元平面的 EOSP(End-Of-Significant-Bit-Plane，即「有效位元平面終點」)位置將各有效位元平面分割為二部份以節省側邊資訊位元。

3. 如申請專利範圍第 1 項之方法，再包括下列步驟：由估測訊息率失真(rate-distortion)來決定各位元的編碼順序，其中該所有編碼位元依照估測失真縮減量與估測位元率二者的比值而以降冪排序。

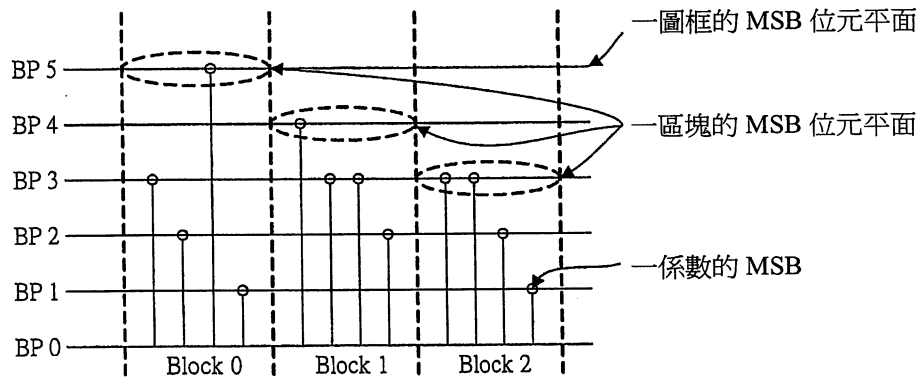
4. 如申請專利範圍第 3 項之方法，其中該各位元的估測訊息率失真係由利用離散拉普拉斯分布以模型化轉換係數而

求得。

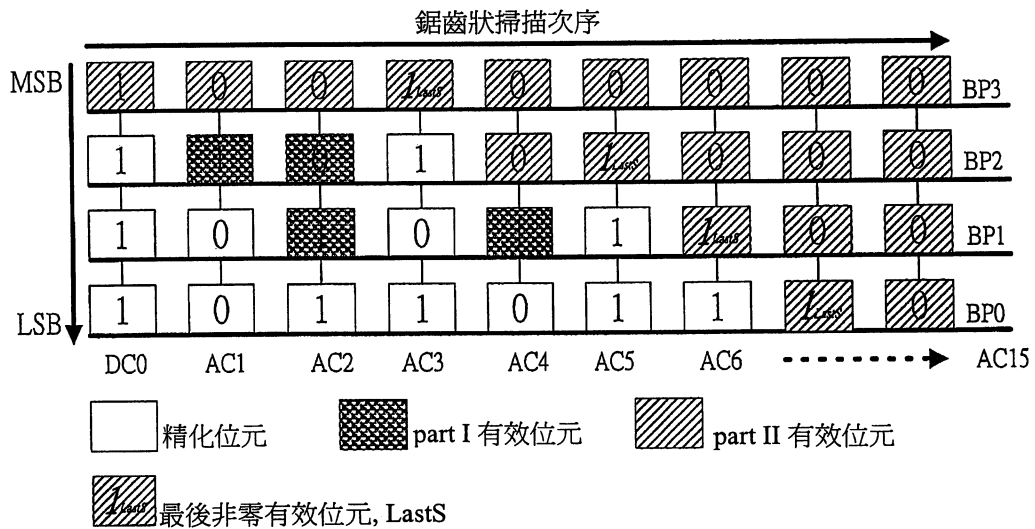
- 5.如申請專利範圍第 1 或 3 項之方法，再包括下列步驟：利用二進制熵來對位元率估測值作編碼；以及利用最大似然原理(maximum likelihood principle)以提供拉普拉斯分布參數的估測。
- 6.如申請專利範圍第 1 項之方法，再包括下列步驟：將一動態編碼流程利用於隨機位元重排。
- 7.如申請專利範圍第 6 項之方法，其中該位元重排可依各種目的例如訊息率失真最佳化、主觀品質改良、以及注目區域功能性等而被延伸至係數、區塊、區域或週期等層次的重排。
- 8.如申請專利範圍第 1 項之方法，再包括下列步驟：利用不同優先度指定與執行重排於不同精細度，以滿足各特定應用。

十一、圖式：

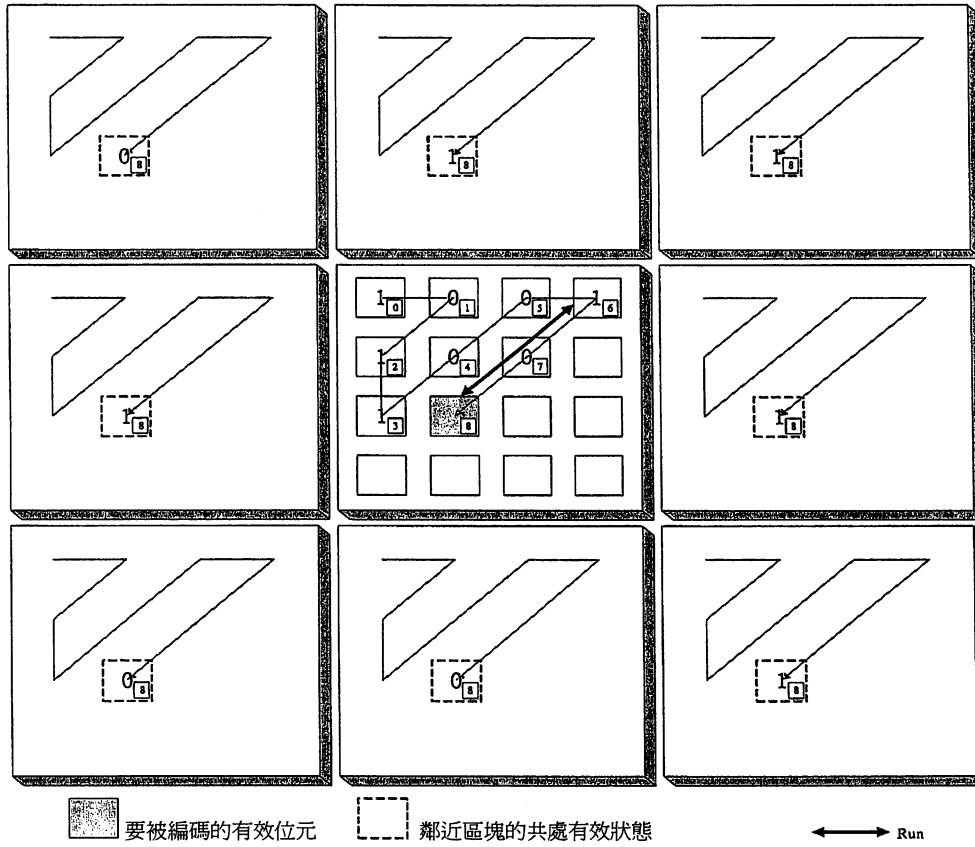
第 1 圖



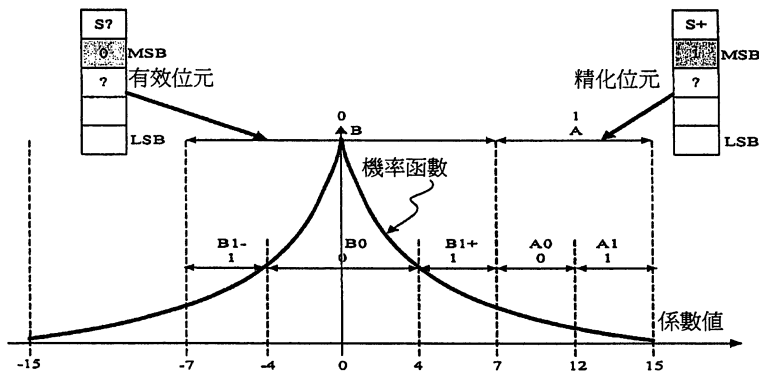
第 2 圖



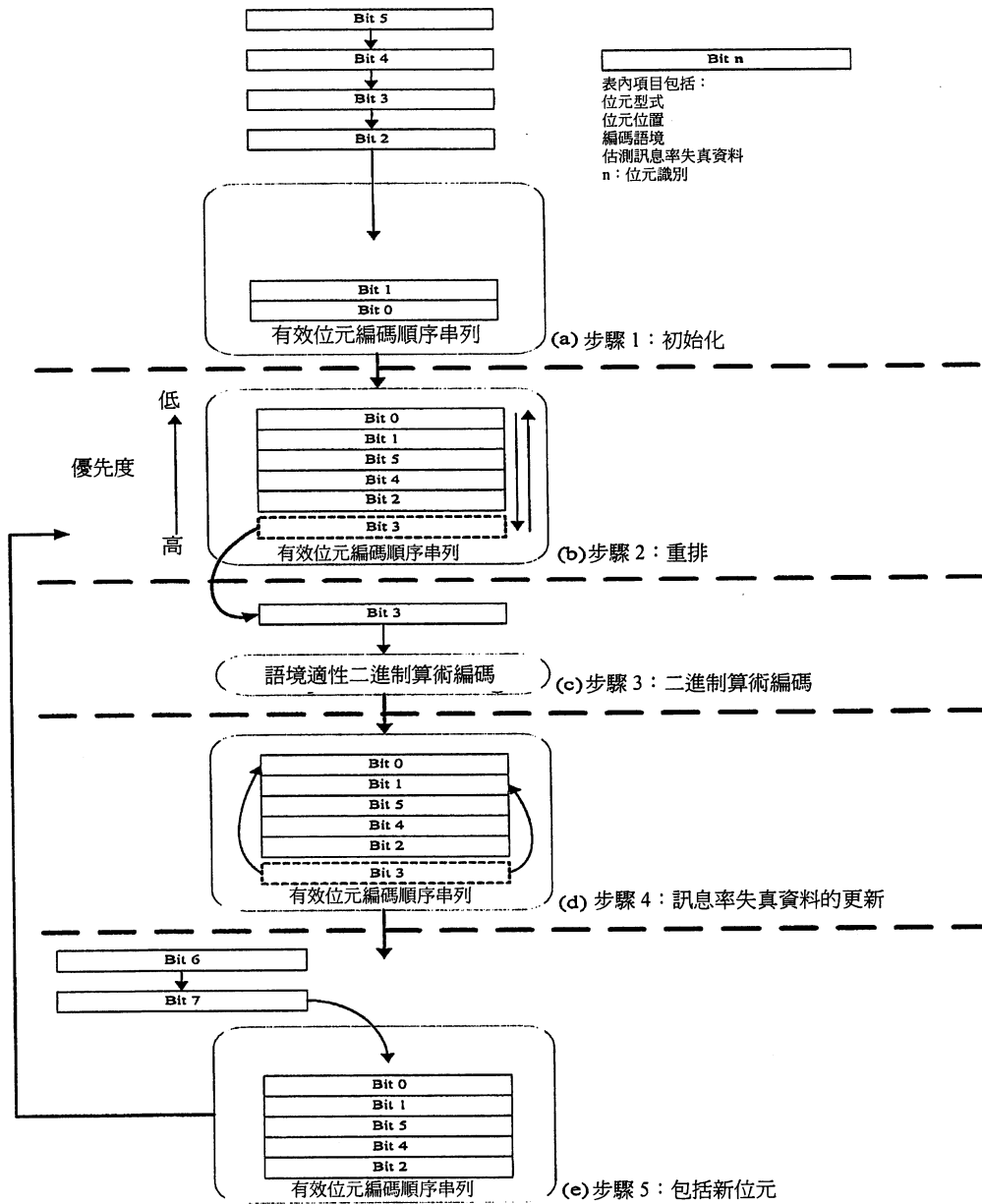
第 3 圖



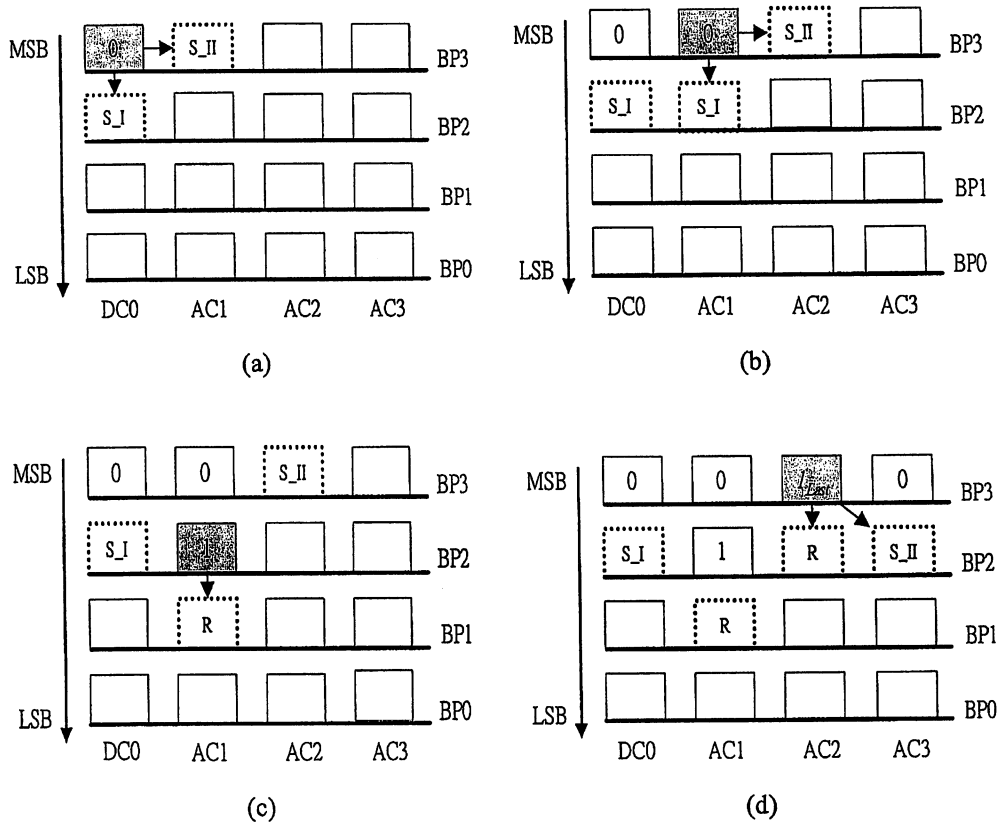
第 4 圖



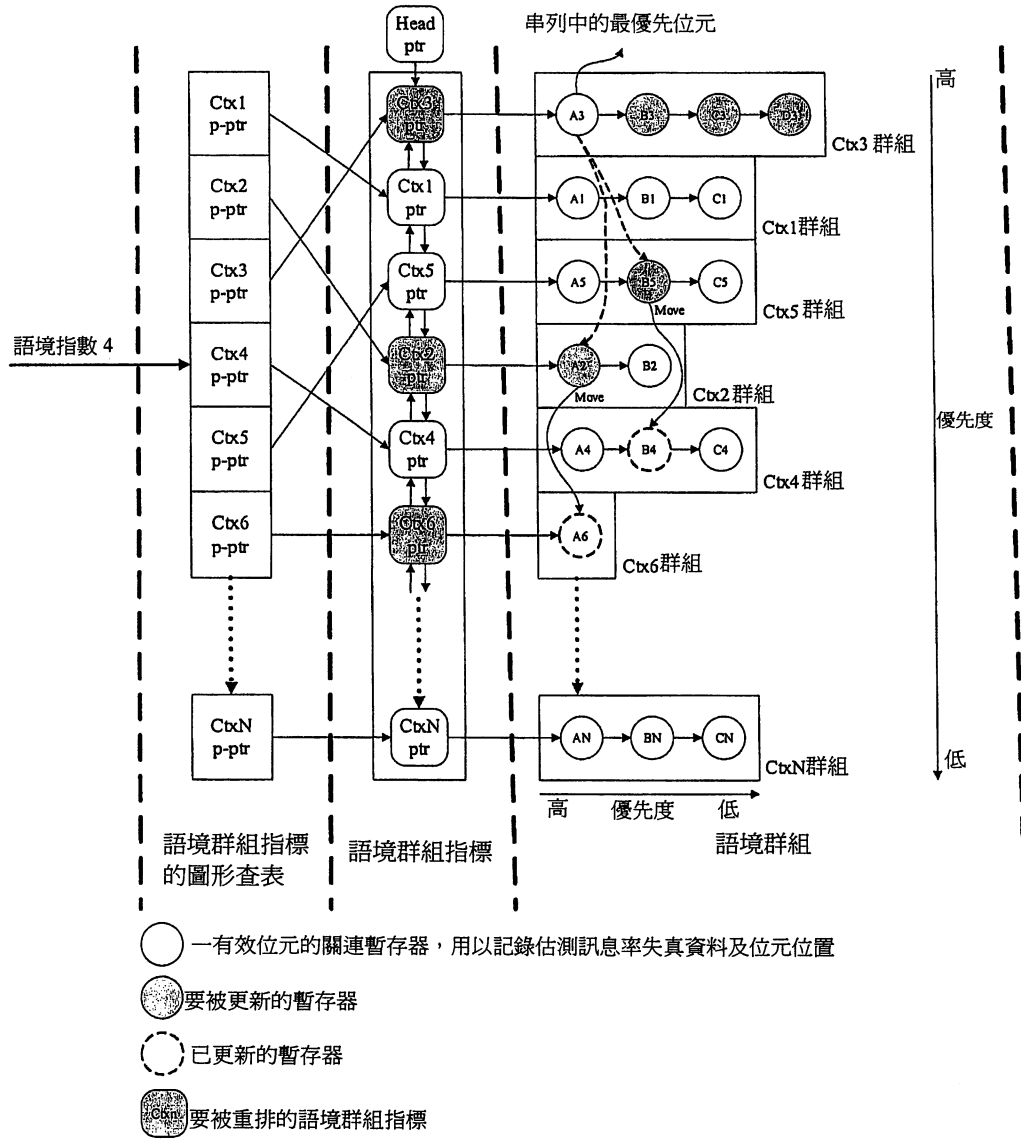
第 5 圖



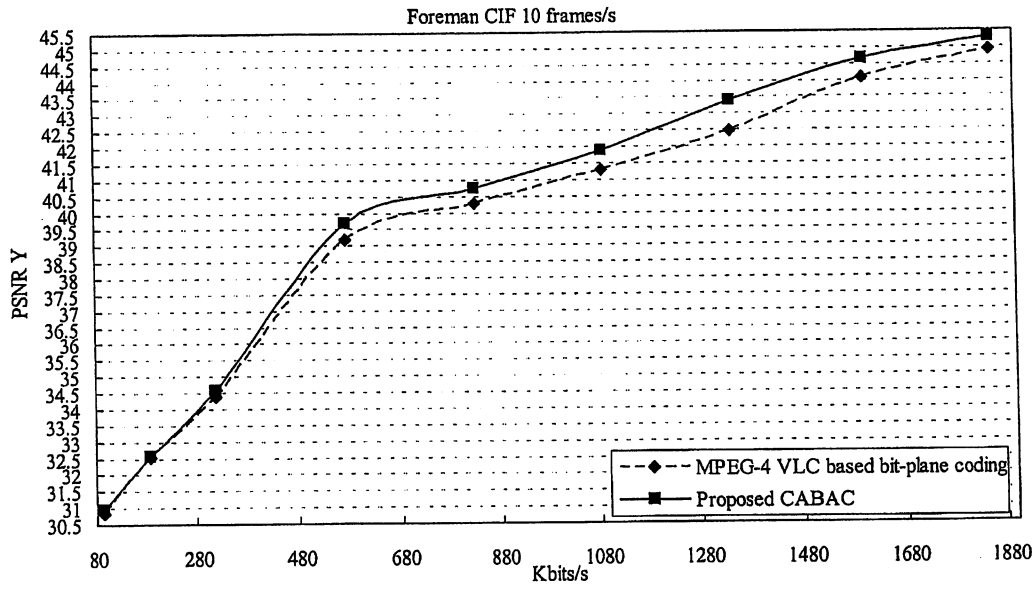
第 6 圖



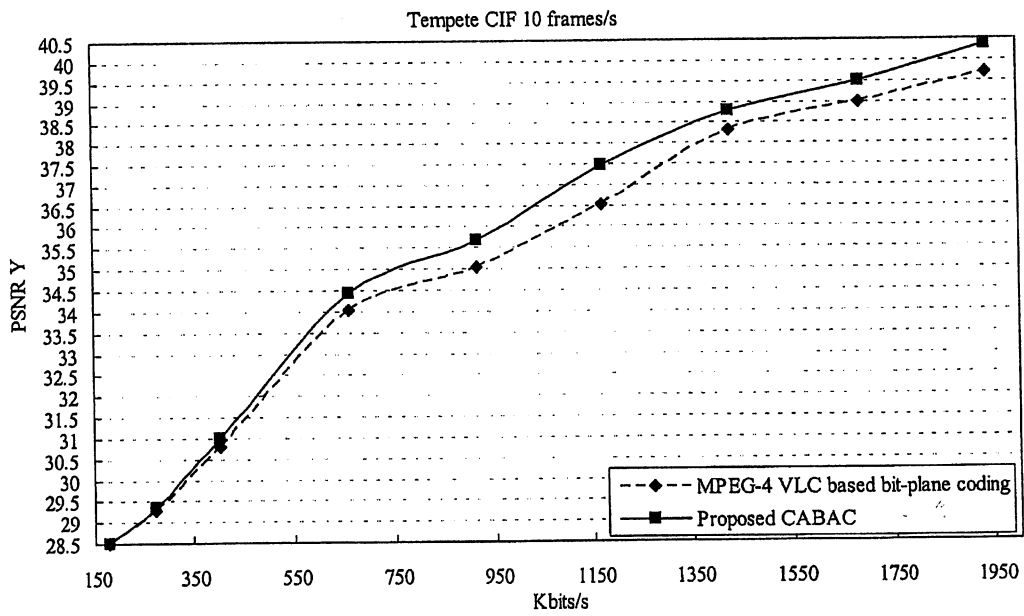
第 7 圖



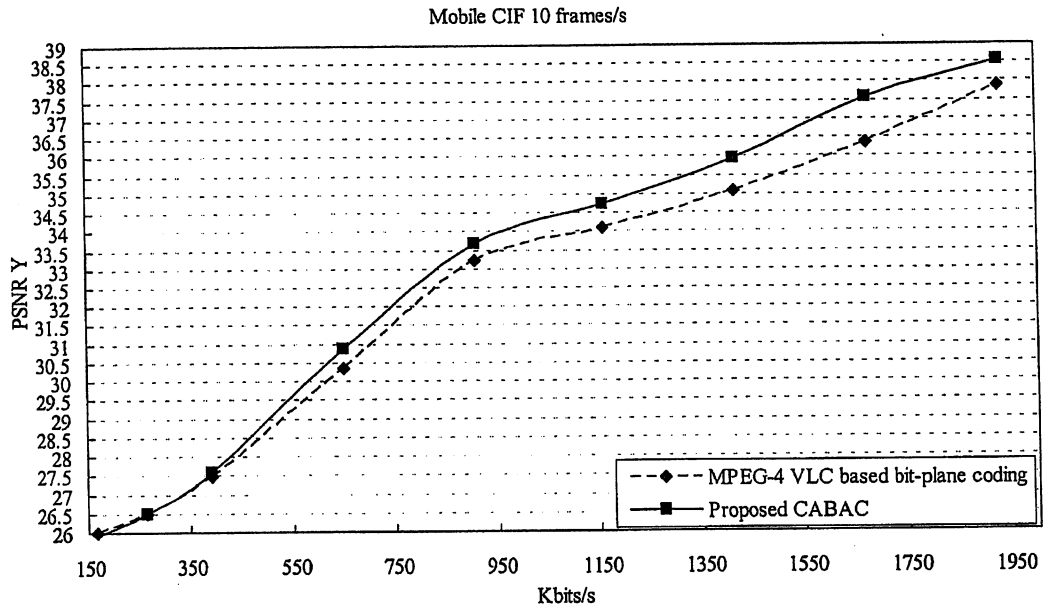
第 8(a)圖



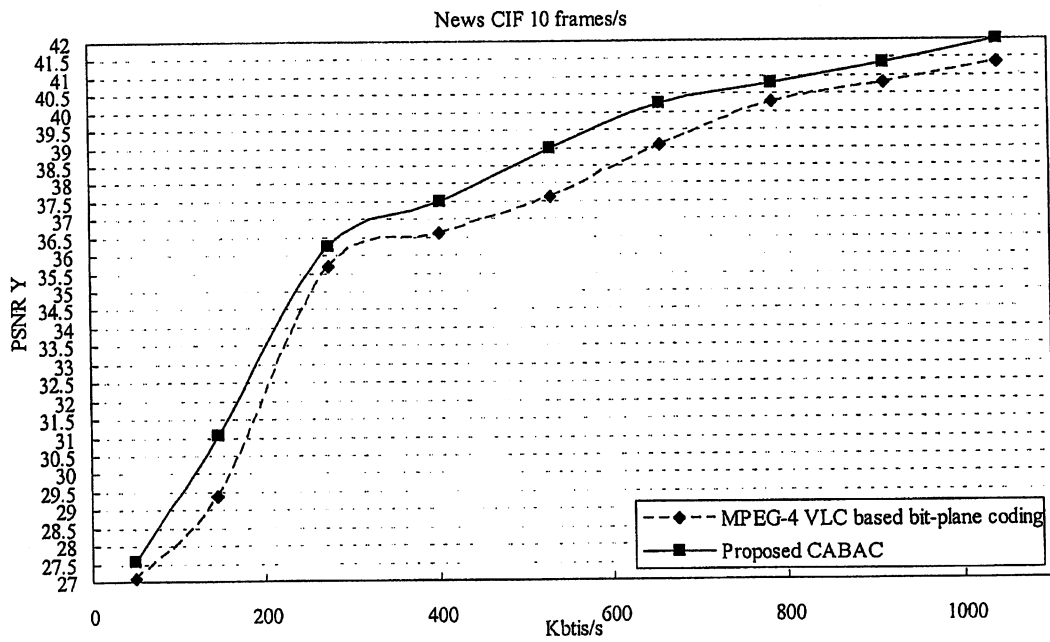
第 8(b)圖



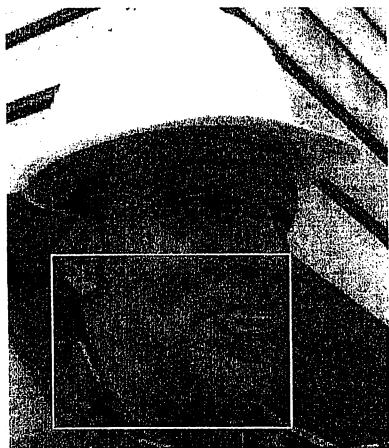
第 8(c)圖



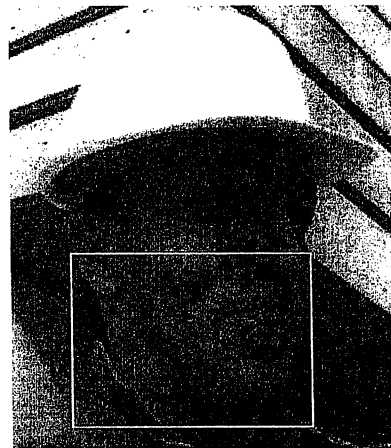
第 8(d)圖



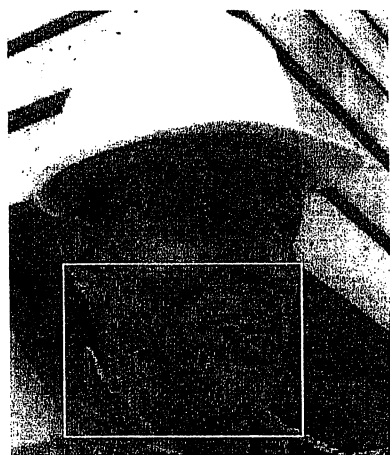
第 9 圖



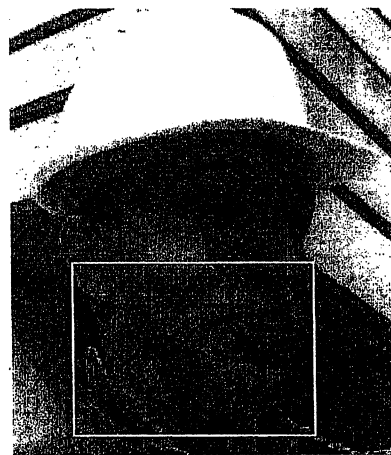
(a) MPEG-4 FGS 爲本的位元平面編碼+光柵掃描



(b) 被提案的 CABAC+光柵掃描

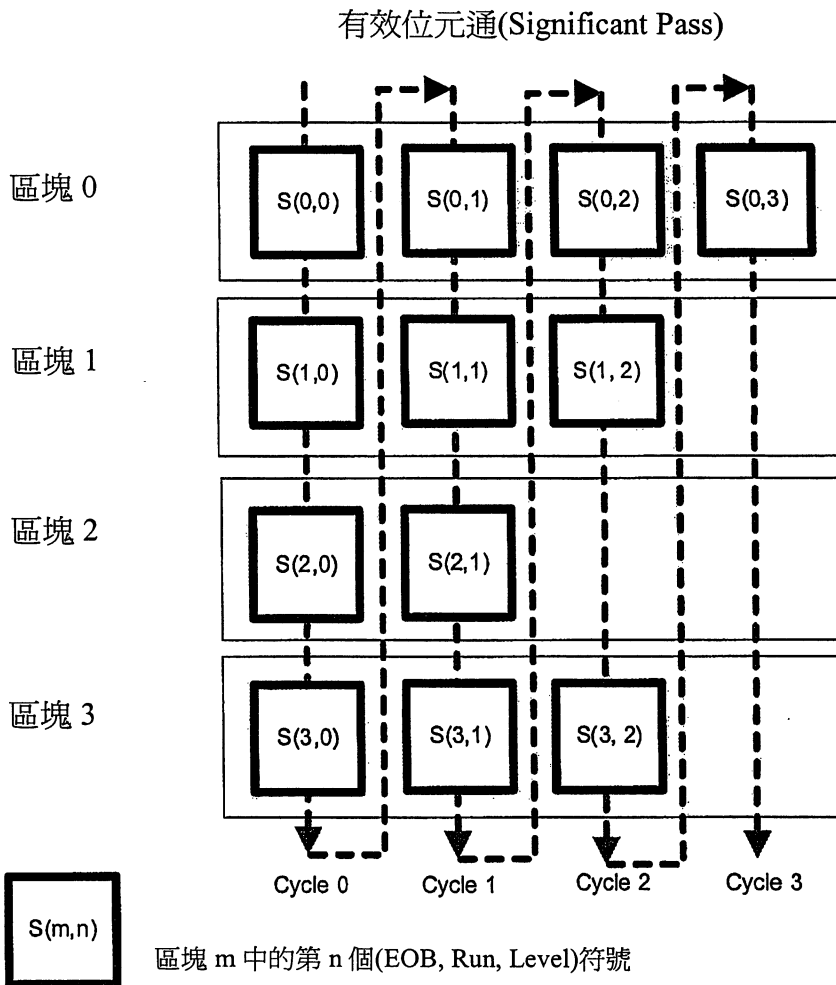


(c) 被提案的 CABAC+位元平面重排



(d) 被提案的 CABAC+加強型重排

第 10 圖

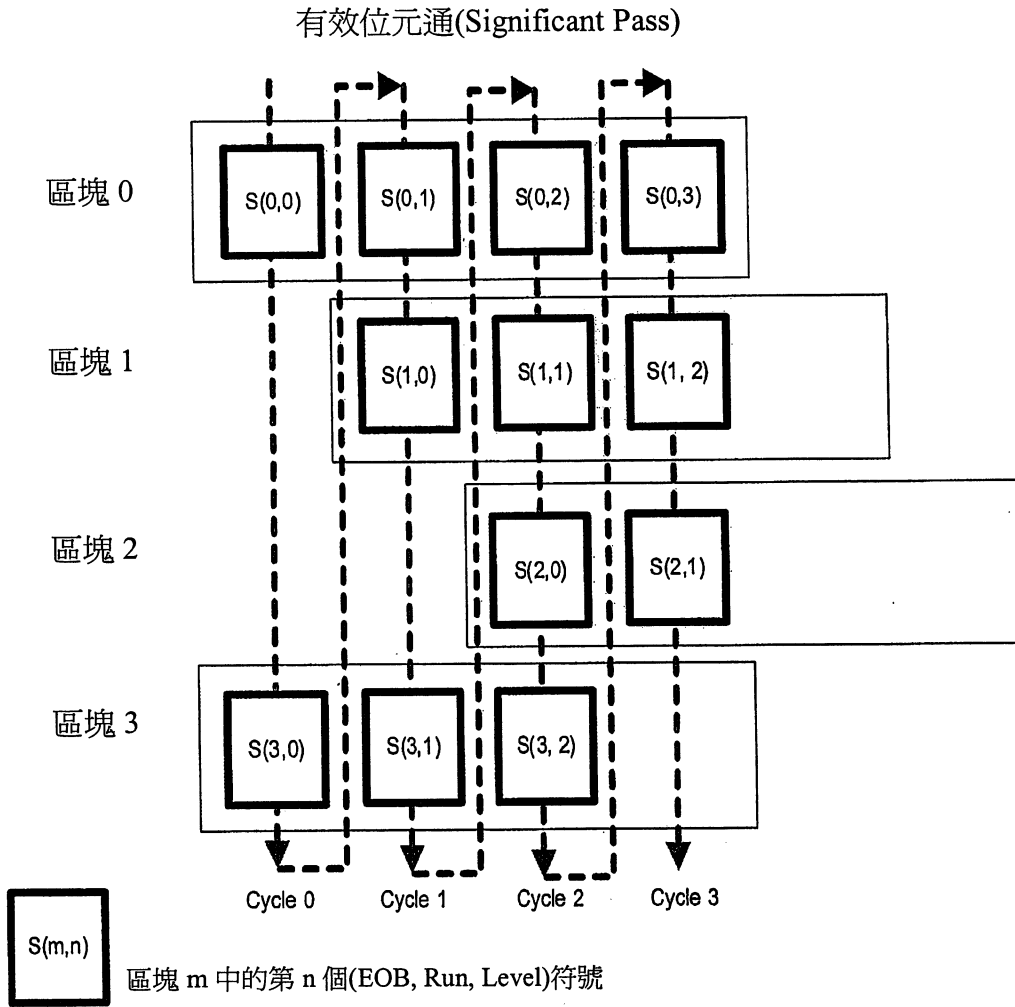


$$S(3, 0) = (\text{EOB}, \text{Run}, \text{Level}) = (0, 0, 10),$$

$$S(3, 1) = (\text{EOB}, \text{Run}, \text{Level}) = (0, 1, 2),$$

$$S(3, 2) = (\text{EOB}, \text{Run}, \text{Level}) = (1, 0, 0)$$

第 11 圖



$$S(3, 0) = (\text{EOB}, \text{Run}, \text{Level}) = (0, 0, 10),$$

$$S(3, 1) = (\text{EOB}, \text{Run}, \text{Level}) = (0, 1, 2),$$

$$S(3, 2) = (\text{EOB}, \text{Run}, \text{Level}) = (1, 0, 0)$$