

發明專利說明書

595230

(填寫本書件時請先行詳閱申請書後之申請須知，作※記號部分請勿填寫)

※ 申請案號：92104361 ※IPC分類：H04N 7/32

※ 申請日期：92.2.27

壹、發明名稱

(中文) 在視訊影像編碼中作為動態預測的裝置與方法

(英文) _____

貳、發明人 (共 4 人)

發明人 1 (如發明人超過一人，請填說明書發明人續頁)

姓名：(中文) 王俊能

(英文) Chung-Neng Wang

住居所地址：(中文) 高雄縣大樹鄉九曲村神農街 16 號

(英文) _____

國籍：(中文) 中華民國 (英文) R.O.C.

參、申請人 (共 1 人)

申請人 1 (如發明人超過一人，請填說明書申請人續頁)

姓名或名稱：(中文) 國立交通大學

(英文) National Chiao Tung University

住居所或營業所地址：(中文) 新竹市大學路 1001 號

(英文) _____

國籍：(中文) 中華民國 (英文) R.O.C.

代表人：(中文) 張俊彥

(英文) Chang, C. Y.

續發明人或申請人續頁 (發明人或申請人欄位不敷使用時，請註記並使用續頁)

發明人 2

姓名：(中文) 羅正弘

(英文) Jeng-Hung Luo

住居所地址：(中文) 台北縣中和市德壽里三鄰和平路 200 巷 35 弄 8 之 1 號

(英文)

國籍：(中文) 中華民國

(英文) R.O.C.

發明人 3

姓名：(中文) 蔣迪豪

(英文) Tihao Chiang

住居所地址：(中文) 台北市同安街 48 巷 3 號 3 樓

(英文)

國籍：(中文) 中華民國

(英文) R.O.C.

發明人 4

姓名：(中文) 李鑑明

(英文) Gen-Ming Lee

住居所地址：(中文) 台南縣仁德鄉仁和村 472 號

(英文)

國籍：(中文) 中華民國

(英文) R.O.C

發明人 5

姓名：(中文)

(英文)

住居所地址：(中文)

(英文)

國籍：(中文)

(英文)

發明人 6

姓名：(中文)

(英文)

住居所地址：(中文)

(英文)

國籍：(中文)

(英文)

捌、聲明事項

本案係符合專利法第二十條第一項 第一款但書或 第二款但書規定之期間，其日期為：_____

本案已向下列國家（地區）申請專利，申請日期及案號資料如下：

【格式請依：申請國家（地區）；申請日期；申請案號 順序註記】

1. 美國； 2002/11/20； 10/301,415

2. _____

3. _____

主張專利法第二十四條第一項優先權：

【格式請依：受理國家（地區）；日期；案號 順序註記】

1. 美國； 2002/11/20； 10/301,415

2. _____

3. _____

4. _____

5. _____

6. _____

7. _____

8. _____

9. _____

10. _____

主張專利法第二十五條之一第一項優先權：

【格式請依：申請日；申請案號 順序註記】

1. _____

2. _____

3. _____

主張專利法第二十六條微生物：

國內微生物 【格式請依：寄存機構；日期；號碼 順序註記】

1. _____

2. _____

3. _____

國外微生物 【格式請依：寄存國名；機構；日期；號碼 順序註記】

1. _____

2. _____

3. _____

熟習該項技術者易於獲得，不須寄存。

肆、中文發明摘要

一種在視訊影像編碼中作為動態預測的方法，此方法建構了一種備有三階二元值層的金字塔型結構。在這個基於三階二元值層的金字塔型結構的動態預測法中，狀態更新模組之暫存器儲存最終的運動向量的重覆出現次數。穩態查核模組根據此重覆出現次數來決定本方法是在穩態模式或是在正常模式。在正常模式裡，第一階二元值層是在 ± 3 像素之精調視窗內來執行此搜尋，以決定一第一階運動向量。在第二階二元值層裡，搜尋範圍是根據 6 個候選的運動向量計算出來的。藉由查核搜尋範圍裡的每個點，第二階二元值層的搜尋產生了一第二階運動向量。最後，在第三階二元值層中 ± 2 像素之精調視窗內，根據第二階二元值層所獲得之運動向量，產生了一最終的運動向量。在穩態模式裡，細調模組在 ± 1 像素之精調視窗內執行搜尋，並且產生了一最終的運動向量。

伍、英文發明摘要

陸、(一)、本案指定代表圖爲：第 1a 圖

(二)、本代表圖之元件代表符號簡單說明：

- | | |
|---------------|-------------|
| 101 二元化金字塔型結構 | 103 第一階搜尋模組 |
| 105 第二階搜尋模組 | 107 第三階搜尋模組 |
| 109 細調模組 | 111 濾波模組 |
| 113 二元化模組 | 115 解析度降低模組 |
| 117 穩態查核模組 | 119 狀態更新模組 |

柒、本案若有化學式時，請揭示最能顯示發明特徵的化學式：

玖、發明說明

【發明所屬之技術領域】

本發明係關於視訊影像編碼的動態預測(motion estimation for video encoding)，特別是關於視訊影像編碼且依據一具有全二元化金字塔型結構(pyramid structure with all binary representation)的動態預測。

【先前技術】

在一個多媒體系統裡，視訊影像編碼模組包含幾個主要元件，這些主要元件包括離散餘旋轉換(Discrete Cosine Transform, DCT)/反離散餘旋轉換(Inverse DCT)、動態預測(motion estimation)、動態補償(motion compensation)、量化(quantization)、反量化(inverse quantization)、位元率控制(bit rate control)、和可變長度編碼(variable length coding, VLC)，其中最花費計算量的部分就是動態預測。一般而言，對一個已最佳化的多媒體系統而言，動態預測約消耗其整個計算功率的50%。所以，在一個多媒體系統裡的即時視訊影像編碼，欲將成本降低，進一步將動態預測最佳化是非常重要的。

當下已經發展出多種快速的運動向量搜尋方法，包括三步搜尋法(three-step search)、二維自然基底搜尋法(2-D logarithmic search)、共軛方向搜尋法(conjugate directional search)、起源搜尋法(genetic search)、菱形

ⓧ 續次頁

搜尋法(diamond search)、利用累積投影之特性為基礎的區塊動態預測法，和具有交錯像素-像素解析度下降(pixel-decimation)之圖案的次取樣運動場估算(sub-sampled motion field estimation)。在運動向量精準性的投注下，這些各式各樣的搜尋技術減小了複雜度。因為這些快速搜尋技術所選取的，可能是某個區域中平均絕對差值(mean absolute difference, MAD)最小(local minimum)的區塊運動向量。因此，與傳統的全搜尋法(full search)的全域最小值(global minimum)相比較之下，這些各式各樣的快速搜尋技術皆僅獲得次佳(suboptimal)的運動向量。

傳統的多解析度(multi-resolution)動態預測技術起始於一個小很多的視窗，從低解析度層到高解析度層去執行搜尋。運動向量的精確性在每一層漸漸被改善並且複雜度減低，但是其搜尋區域卻等同於複雜性最高的全搜尋法的搜尋區域。為了進一步減小複雜性，傳統的二元化動態預測法則藉由減小位元深度(bit depth)，大幅降低計算複雜性和匯流排頻寬(bus bandwidth)。依據一種二元化金字塔型結構，Song等人在*IEEE Trans. on Circuits and Systems for Video Technology*, vol. 10, no. 7, pp.1015-1028, Oct. 2000之期刊的文獻“New fast binary pyramid motion estimation for MPEG2 and HDTV encoding”中揭露一種快速的二元化動

態預測法則，亦即快速的二元化金字塔型動態預測 (FBPME)。此FBPME的金字塔型結構包含一個最低解析度，亦即最小的畫面大小(smallest picture size)，的八位元值層(8-bit data layer)，和三個具有影像細節資訊的二元值層。FBPME以在二元值層的XOR布林區塊匹配(block matching)條件去執行動態搜尋，並且在八位元值層去執行平均絕對差值的運算。此XOR布林區塊匹配是使用比平均絕對差值的運算還要簡單和快速的XOR運算來實現。

然而，此FBPME結構使用了一層八位元值層，此八位元值層導致必須使用兩個不同的失真計算模組(distortion computation module)來執行XOR運算和平均絕對差值運算。因此，FBPME的硬體實現需要較大的編碼大小(code size)和較多的硬體複雜性(hardware complexity)。此FBPME結構也需要較複雜的前置處理步驟(pre-processing)，包括濾波(filtering)、解析度下降(decimation)、二元化(binarization)和內插(interpolation)。XOR運算和平均絕對差值運算所需的硬體複雜度，以及在多層搜尋技術裡所需的較複雜的前置處理步驟，導致了FBPME在硬體實施時更多的電力消耗。

由Natarajan、Bhaskaran、和Konstantinides等人提出的另一個傳統的二元化動態預測法則是以基於一個具有傳統搜尋技術的簡單的1-位元轉換(one-bit transform)技術。此動

態預測法則提供了目前區塊和參考區塊並進行單層動態預測。然而，此單層二元值表示法未使用任何的階層式結構(hierarchical structure)。階層式結構時的優勢是若處理較低解析度之畫面，其所能得到的運動向量會比基於單層的二元值表示法的動態預測法所得到的運動向量更精準。

【內容】

本發明克服上述傳統動態預測的缺點，其主要目的之一為提供一種視訊影像編碼之具有全二元化的動態預測的方法。依此，建構了一種具有三階的影像的二元值層(binary layer)。首先，以一依位元(bit-wise)執行的差和為基礎的條件在第一階二元值層搜尋，來找出一個第一階運動向量。然後使用 6 個候選的運動向量，來決定一個在第二階二元值層的運動向量。最後，根據此第二階運動向量，在第三階二元值層執行第三階搜尋，並且產生一個最終的運動向量。

本發明中，二元化金字塔型結構的建構包括濾波、二元化，和解析度降低。精確的邊緣資訊係根據影像在一小區域中的空間變化(spatial variation)擷取出來的，不需使用任何的八位元值層，即可提供全二元化邊緣資訊(edge information)。在第一階搜尋中，以一 ± 3 像素之精調視窗來執行此搜尋。在第二階搜尋中，本發明根據存在於區塊之

間的空間-時間的相依性，從目前的畫面和上一個畫面估算出6個候選的運動向量，利用這6個候選的運動向量計算出每一個大小為 8×8 的區塊的二維運動偏移量 $([R_{\min}^x, R_{\max}^x], [R_{\min}^y, R_{\max}^y])$ 的範圍。沿著6個運動向量的平均向量，第二階的精調視窗所以涵蓋了大小為 $(R_{\max}^x - R_{\min}^x + 1) \times (R_{\max}^y - R_{\min}^y + 1)$ 範圍的主要搜尋區域。本發明然後以一 $(R_{\max}^x - R_{\min}^x + 1) \times (R_{\max}^y - R_{\min}^y + 1)$ 像素之精調視窗執行全搜尋XOR布林區塊匹配。類似地，被選取的候選運動向量將被傳遞到下一二元化階。在第三階中，以 ± 2 像素之精調視窗來執行搜尋。在每一階中，利用XOR區塊匹配，最佳運動向量的搜尋和決定是以一個最小的依位元執行的差和的條件為基礎。

本發明的另一個目的是提供一種視訊影像編碼之動態預測的裝置。依此，此裝置包含一個二元化金字塔型建構模組、一個第一階搜尋模組、一個第二階搜尋模組，和一個第三階搜尋模組。每一階搜尋模組包括一資料載入模組(data loading module)、一位元對位模組(bit alignment module)和一 XOR 區塊匹配模組。二元化金字塔型建構模組更包含一濾波模組(filtering module)、一二元化模組，和一解析度下降模組(decimation module)。每一 XOR 區塊匹配模組更包括一查表次模組(table lookup sub-module)和一個依位元執行的差和次模組 SoD (bit-wise sum of difference sub-module)。

□ 續次頁

本發明的動態預測適用於管線結構(pipelined architecture)。動態預測的方法可實現在多種結構中，這些結構包含如x86的通用結構、使用英代爾(Intel)的MMX技術的單指令多資料(single instruction multiple data, SIMD)結構、以及心跳陣列(systolic array)。本發明的管線結構包含三個主要的共同模組，包括整合建構(integrated construction)、緊湊的記憶體(compact storage)、以及平行的區塊匹配。

本發明使用 MPEG-4 視訊參考編碼器和一個大小為 16×16 的巨集區塊來作區塊匹配，以說明實施效率。根據此實驗的結果，本發明不但具有低運算複雜性和低記憶體頻寬消耗的優點，並且對於搜尋範圍的增大並不敏感。系統設計者可以選擇較好的二元化方法，使視覺品質能更進一步地獲得改善。另外，針對不同大小與型態的暫存器之特定平台，也能夠開發各種不同的最佳化方法。因此，本發明的動態預測法比其它動態搜尋計算法具有較大的彈性。由運算數目來看，本發明的動態預測法也很適合在通用處理系統中進行軟體實施。它也能被實施在 ASIC 設計的平行管線，並且在硬體設計階段更允許在矽區(silicon area)面積、電力消耗和視覺品質間作取捨。

茲配合下列圖式、實施例之詳細說明及專利申請範

X 續次頁

圍，將上述及本發明之其他目的與優點詳述於后。

【實施方式】

本發明之全二元化估算基本上包含一多階二元化金字塔型結構和數個搜尋模組。為了在估算運動向量的精確度和計算的複雜度之間作取捨，本發明之較佳實施例使用一個三階金字塔型結構作為動態預測。在底下的描述中，各階分別表示為第一階、第二階和第三階。

圖 1a 為根據本發明作為動態預測之三階全二元化金字塔型結構的區塊圖。作為動態預測之圖 1a 的裝置包含一個二元化金字塔型結構 101、一個第一階搜尋模組 103、一個第二階搜尋模組 105、一個第三階搜尋模組 107、一個細調模組 (fine tuning module) 109、一個穩態查核模組 (static-state checking module) 117 和一個狀態更新模組 (static update module) 119。每一階的搜尋模組包含一資料載入模組、一位元對位模組，和一 XOR 區塊匹配模組。二元化金字塔型結構 101 更包含一濾波模組 111、一二元化模組 113 和一解析度降低模組 115。使用簡單的解析度降低方法，在 XOR 區塊匹配中用到的區塊大小在三個階層中分別為 4×4 、 8×8 和 16×16 。圖 2 至圖 4 分別說明第一階、第二階和第三階的搜尋模組。

參考圖 1b 圖和圖 1c 的穩態查核模組和一個狀態更

新模組，其中包含一計數器 121 和三個比較器 122、123 和 124。當處理在原始視訊影像(source video)中的第一個畫面時(first frame)，所有區塊的計數器歸零。使用兩個常數 $S1$ 和 $S2$ ， $S1 > S2$ ，來控制靜態模式的出現頻率以保持影像品質。為了處理每一區塊 i ，穩態查核模組中的比較器 122 檢查計數器 i 的內容是否大於 $S1$ 。若是，則動態搜尋進入穩定狀態，且在細調模組中在 ± 1 的範圍內細步調整運動向量。否則動態搜尋通過正常的狀態三階的二元化搜尋。動態預測之後，狀態更新模組中的比較器 123 檢查目前畫面(current frame) j 每一區塊 i 的最終運動向量(final motion vector)是否等於前一個畫面(previous frame) $j-1$ 的區塊 i 中儲存的最終運動向量，其中前一個畫面的區塊與目前的區塊的座標相等。若兩個運動向量相等，則比較器 124 檢查計數器 i 121 的內容是否小於 $S2$ 。如果小於 $S2$ ，則計數器 i 121 的內容加 1。否則，計數器 i 121 的內容歸零。上述的步驟應用到目前畫面的每一個區塊。必須注意的是圖 1b 圖和圖 1c 中使用相同的計數器 i 121。

為了進一步降低計算的複雜性，可以選擇性地加入一計數器以記錄每一巨集區塊(macroblock)的運動向量重覆出現次數(repeat occurrence)。若經過 4 個畫面之後，運動向量的值維持不變，則本發明認定目前的巨集區塊為穩定狀態且跳過動態搜尋，而只在圖 1a 中所示的細調模組 109

中作小範圍 ± 1 的調校。在全二元化表示式，XOR 運算和小範圍搜尋視窗等條件之下，動態預測的速度就能被改進而只是犧牲一點點重建影像(reconstructed image)的品質。

參考圖 2，第一階搜尋模組包含一資料載入模組 201、一位元對位模組 203，和一平行(parallel) XOR 區塊匹配模組 205。位元對位模組 203 含一 ± 3 像素之精調視窗(pixel refinement window)。參考圖 3，第二階搜尋模組 105 也包含一資料載入模組 301、一位元對位模組 303、一平行(parallel) XOR 區塊匹配模組 305 和一運動偏移模組 307。

參考圖 3 中的運動偏移模組 307，在進入第二階細調搜尋之前，第一階搜尋和上一個畫面估算出 6 個候選的運動向量，而運動偏移模組利用這 6 個候選的運動向量計算出每一個大小為 8×8 的區塊的二維運動偏移量的範圍 $([R_{\min}^x, R_{\max}^x], [R_{\min}^y, R_{\max}^y])$ 。以推導出的運動偏移量為基礎，位元對位模組 303 有一大小為 $([R_{\min}^x, R_{\max}^x], [R_{\min}^y, R_{\max}^y])$ 的像素之精調視窗。

參考圖 4a，第三階搜尋模組含一資料載入模組 401 一位元對位模組 403，和一平行 XOR 區塊匹配模組 405。圖 2 至圖 4 中的每一平行 XOR 區塊匹配模組 205、305 和 405 更包含一查表次模組和一個依位元執行的差和次

模組 SoD，如圖 4b 所示。實施於圖 1 中所示之裝置中的動態預測方法包含 4 個主要步驟。首先，第一步驟係建造多階二元化金字塔形結構。第二步驟係以一 3×3 像素之精調視窗來執行全搜尋 XOR 布林區塊匹配。第三步驟係對每一 8×8 區塊，由上一個和目前的畫面估算 6 個候選的運動向量，以計算其二維運動偏移量的範圍 $([R_{\min}^x, R_{\max}^x], [R_{\min}^y, R_{\max}^y])$ ，並且在特定的區域執行布林區塊匹配作第二階的精調(refinement)。最後，第四步驟係在最後一階以 ± 2 像素之精調視窗來執行全搜尋 XOR 布林區塊匹配。

第一步驟更包含 3 個次步驟，濾波、二元化和解析度降低。三個次步驟分別在濾波模組 111、二元化模組 113 和解析度降低模組 115 中執行，以建造三階的二元化金字塔形結構。

每一個原始像素值與一臨界值作比較以產生二元值表示法(binary representation)，此臨界值是由數個鄰近的像素值得來。在以八位元值表示法(8-bit representation)的階段，經過濾波的影像重覆地對每一個影像維度以每兩點取一點的方式來完成下一層的影像。

在二元化金字塔型結構的建造過程中，精確的邊緣資料係根據影像像素在一小區域中的空間變化(spatial

variation)擷取出來的。本發明之濾波模組的設計不需使用如 FBPME 方法中所需的八位元值層，即可提供了全二元值邊緣資訊(all binary edge information)。像素間之空間變化可用各種濾波器擷取出來。假設以一 8 位元表示的目前畫面 F ，經低通(low-pass)濾波後，產生一個新的畫面 \bar{F}_l 。換言之，

$$F_L = F ,$$

$$\bar{F}_l = H(F_l) ,$$

其中 $l=1..L$ ， L 為使用的塔層總數。本發明中，將 L 設定為 3。畫面 \bar{F}_l 係與第 l 層相同大小之原始畫面模糊化後的版本(blurred version)。圖 5 說明每一塔層以二元化格式的建造。

然後用經過濾波的畫面 \bar{F}_l 來產生目前畫面的二元化表示式。二元化表示式係依一臨界值 T_b 而建造的。此臨界值被定義以便經精確地表示二元化格式的邊緣資訊。邊緣資訊可從原始影像和其經過低通濾波器產生的影像兩者之間的差值來找到。為了計算二元化表示式，本發明採用一種新的差異運算(differencing operation)，此差異運算使用一個由畫面 \bar{F}_l 計算出的特定臨界值，並且此畫面 \bar{F}_l 提供鄰近像素區域的平均空間變化的資訊。所以，第 l 階的二元化表示式可由底下 1-位元轉換公式求得：

$$S_l(x, y) = T_{1-bit}(F_l, \bar{F}_l) = \begin{cases} 1 & \text{若 } (F_l(x, y) \geq T_b = \bar{F}_l(x, y)) \\ 0 & \text{否則的話} \end{cases}$$

在解析度降低步驟的最後過程中，包含大部份目前階層的原始影像的空間資訊的低通畫面 \bar{F}_l 被用來產生下一塔層的輸入畫面。為了計算下一階層，畫面 \bar{F}_l 被以每兩點取一點的方式來產生新的畫面 \bar{F}_{l-1} ，如下列公式所描述：

$$F_{l-1} = \downarrow(\bar{F}_l)。$$

在第二步驟中，在第一搜尋模組 103 內以一 ± 3 像素之精調視窗執行全搜尋 XOR 布林區塊匹配，以找出一 4×4 區塊大小的候選的初始運動向量。候選的初始運動向量再投射到 8×8 區塊大小的下一個二元化階層。

本發明的匹配準則(criterion)為依位元執行的差和運算 SoD:

$$SoD(u, v) = \sum_{(x, y) \in Block} [S_{l,t}(x, y) \oplus S_{l,t-1}(x+u, y+v)]，$$

其中 $S_{l,t}(x, y)$ 表示在第 l 層中的目前的二元化區塊，

$S_{l,t-1}(x+u, y+v)$ 表示從目前區塊的左上角偏移 (u, v) 量的參考

二元化區塊。因為畫面資料全部用二元化格式表示，所以差值的計算用一個簡單的 XOR 運算即可。

本發明以一 ± 3 的搜尋範圍執行第一階全域搜尋，而不用 ± 16 ($R_{Lv1} = R/4 - 1$)的搜尋範圍，其中 R 為目標搜尋範圍(target search range)。以一個小的視窗仍然可以得到涵蓋16倍的實際搜尋區域的運動向量。

在第三步驟中，以存在於區塊中的時空相依性(spatial-temporal dependencies)為基礎，本發明利用上述的XOR準則，在目前和先前的畫面中，在6個候選的向量中選取最佳的 8×8 運動向量。此6個候選的向量對應右上(UR)、上(U)、左(L)、第一階的候選的初始運動向量、時間的先前區塊、以及中央之(C)6個運動向量。候選的中央向量也就是零運動向量(zero motion vector)。

根據本發明，每一維度的6個候選向量的最小和最大偏移量係由圖3所示之運動偏移模組307選取作為第二階的精調視窗的邊界。因此，運動偏移模組307對每一 8×8 區塊定義二維運動偏移的範圍

$([R_{min}^x, R_{max}^x], [R_{min}^y, R_{max}^y])$ 。第二階的精調視窗，沿著6個運動向量，涵蓋了大小為 $(R_{max}^x - R_{min}^x + 1) \times (R_{max}^y - R_{min}^y + 1)$ 的主要搜尋區域的範圍。因此，為了改善第二階，本發明以一 $(R_{max}^x - R_{min}^x + 1) \times (R_{max}^y - R_{min}^y + 1)$ 像素精調視窗執行全域搜尋XOR布林區塊匹配。同樣地，結果的候選運動向量傳到第三階以作為搜尋範圍為 ± 2 的進一步改善。

在步驟中 4 中，使用一個 ± 2 搜尋視窗作精調的搜尋，以導出區塊大小為 16×16 的最終運動向量。在本發明中，最終運動向量的精確度受到第一階二元化過程的效率所影響。在二元化過程中所使用的臨界值 T_b 與平均空間特徵和濾波後的邊緣資訊有關。如上所述，具較高的截斷頻率(cutoff frequency)的低通濾波器可去除二元化表示 $S_{i,t}(x,y)$ 中平均空間變異性較小的部份而保留較空間變異性較大的部份。空間變異性較大的部份意指影像中的高頻資訊。保留的額外的高頻資訊可增加全域二元化邊緣資訊的精確度。在實際的實施例中，編碼效率和濾波器的複雜性必須作個取捨。

為了說明本發明的動態預測適用於管線結構，以下例舉幾種不同的結構作說明。這些結構包含如 x86 的通用結構、使用英代爾的 MMX 技術的 SIMD 結構、以及心跳陣列。為了簡化起見，底下所描述的所有實施例中，濾波模組 111 中用到的濾波器為

$$H_A = \frac{1}{4} \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}。$$

為了證明本發明在全搜尋法的有效性，本發明使用 x86 系統和 C 語言作模擬。由於每一階的儲存單元和初始位置皆不同，區塊匹配的加速係以本發明所提出的方法一階一階地完成。一階一階的實施方法在區塊匹配中消

耗較低的記憶體頻寬。

本發明的管線結構包含三個主要的共同模組，分別為整合的架構、緊湊的記憶體、以及平行的區塊匹配。對於每一模組，記憶體的大小和形式被個別結構所定義的資料倉(data bins)/暫存器(register)所限制，其中各個結構提供多種不同大小的資料儲存裝置/暫存器。為了實施 x86 系統中每一種類的模組，使用 C 語言以提供三種基本的資料倉，包含 INT(32 位元)、SINT(16 位元)和 UCHAR(8 位元)。在使用英代爾的 MMX 技術的 SIMD 結構中，每一模組最大的資料倉為 64 位元 MMX 的暫存器。除了可獲得較大的資料倉之外，這類結構的最佳化處理和 x86 結構是相似的。為了在兩種結構中進一步的減少處理單元間記憶體頻寬(memory bandwidth)和減少運算次數，應該使用較大的資料倉以單一封包方式來儲存數個連續區塊的二元化表示式。

三階二元化金字塔形整合結構的第一個模組包含濾波、二元化和解析度降低等三個步驟。這個模組可用邊界分割(boundary partitioning)和平行處理(parallel processing)兩個次模組來加強。邊界分割用來避免去決定目前的資料是否位於畫面邊界上。第一個模組，即邊界分割，係將畫面分割成九個區域，包括四個角落、上下兩個橫邊、左右兩個直邊，和內部區域。第二個模組，即平行處理，

續次頁

由載入數個像素集合來達成。該像素集合係依序儲存在較大的暫存器中。圖 6 說明使用濾波器 H_A 來建造低通畫面 \bar{F}_l 的一個例子。像素 $F_l(x, y)$ 表示在塔層的第 l 階的處理畫面中 (x, y) 坐標上的像素。使用濾波器 H_A 的低通畫面的推導等同於包含所有距離查核點為 1 的鄰近像素之平均值的計算。

假設畫面資料儲存在 x86 系統 C/C++ 環境底下的資料倉中，叫做 SINT。圖 6 也說明處理像素對 $(F_l(0,1) F_l(1,1))_{32}$ 的一個例子。首先，本發明由參考畫面載入 4 個鄰近的像素對 $(F_l(1,0) F_l(0,0))_{32}$ ， $(F_l(1,2) F_l(0,2))_{32}$ ， $(F_l(0,1) 0)_{32}$ 和 $(F_l(2,1) F_l(1,1))_{32}$ 至 4 個 32 位元的 INT 暫存器 $(R_1)_{32}$ ， $(R_2)_{32}$ ， $(R_3)_{32}$ 和 $(R_4)_{32}$ 中。載入後，由於像素資料沒有上限溢位 (overflow) 或向下溢位 (underflow) 的問題，故可以直接計算加總 (summation) 和進位 (rounding) 運算。這是由於 8 位元寬的資料儲存在 16 位元的資料倉中的緣故。在載入和加總運算之後，其結果再放回到一個 32 位元的 INT 暫存器 $(R)_{32}$ 中：

$$\begin{aligned} (R)_{32} &= (R_1)_{32} + (R_2)_{32} + (R_3)_{32} + (R_4)_{32} + 0x00040004 \\ &= (F_l(1,0) + F_l(1,2) + F_l(0,1) + F_l(2,1)) [F_l(0,0) + F_l(0,2) + 0 + F_l(1,1)]_{32} + 0x00040004 \end{aligned}$$

其中值 0x00040004 係作為同步處理像素的進位用。每一

續次頁

像素的臨界值可由抽取暫存器 $(R)_{32}$ 內像素對的值而得到：

$$\begin{aligned}(R_{LSB} \gg 2)_{16} &= ((R)_{32} \gg 2) \& 0x3FFF \\ (R_{MSB} \gg 2)_{16} &= (R)_{32} \gg (2+16)\end{aligned}$$

其中‘ \gg ’表示邏輯右移(logical SHIFT RIGHT)運算。得到的臨界值係用來作二元化用。以這個最佳化的方法，在建造的階段約可提高 30% 的速度。

如果可以取得更大的資料倉，速度可以用一個相當於能夠同時載入單一資料倉的像素數目相等的因子而增加。在使用英代爾的 MMX 技術的 SIMD 結構中，二元化塔層也能以類似上述 x86 架構的邊界分割和平行處理的方式建造。然而，由於 64 位元結構的緣故，MMX 技術可達到的平行度約為 2 倍。以上述兩公式為基礎，速度的改進為 63%。

畫面資料的二元化表示法可達到塔層所要得到的緊湊儲存空間。在第一階段中，每一 4×4 區塊的橫列只佔 4 個位元。因此，一個 4α 位元的資料層能夠儲存數個依序的 α 區塊相同的橫列。在第二階段中，一個大小為 $(R_{\max}^x - R_{\min}^x + 1) \times (R_{\max}^y - R_{\min}^y + 1)$ 的連續且依序的搜尋範圍以 6 個候選的向量定義。在第二階搜尋中，採用數個區塊組裝(pack)成單一資料倉作為第一階段的區塊。因此，對於這兩個電腦結構，本發明在第二階段將每一連續 α 個 8×8 大小的區塊的橫列組裝成大小為 8α 的單一資料。使用與

第一階段相同的觀念，本發明在第三階段將每一連續 α 個 16×16 大小的區塊組成的橫列組成一大小為 16α 的單一資料倉。註，在兩個結構中最寬的資料倉只是用來加速區塊的匹配。

圖 2 至圖 4 說明包括載入，位元對位和平行 XOR 區塊匹配等步驟之使用全二元化邊緣資訊作為快速區塊匹配的區塊圖。載入模組將每一組序列資料置於較大的對應資料倉中以減少記憶體存取動作。由於二元化的畫面資料已經緊湊且依序地儲存，故記憶體的存取就變成只是簡單的擷取(fetch)指令。因此，載入模組分別將目前資料和搜尋視窗中的參考資料載入晶片記憶體。

如圖 2 所示，對於第一階每一組 4 個連續的 4×4 大小的區塊，本發明一個接一個地將同一列的目前 4 個區塊載入一個特定的 SINT 暫存器中，且在 x86 結構中將參考區塊的對應列載入一 INT 暫存器。因此，接續的平行區塊匹配可使用 4 個 SINT 暫存器以提高速度至約為區塊對區塊的匹配法(block-by-block matching scheme)的 4 倍。在英代爾的 SIMD 架構中，使用 64 位元的資料倉和相同的流程來處理第一階每一組 8 個連續的 4×4 大小的區塊。因此，可使用 4 個 MMX 暫存器以提高平行度(parallelism)至約為區塊對區塊的匹配法的 8 倍。

位元對位模組將搜尋視窗中的參考資料與目前資料

☒ 續次頁

同步。位元對位之後，採用(平行)XOR 區塊匹配，查表和 SoD 比較次模組，以得到最小 SoD 值的運動向量。此

XOR 運算應用來計算失真量(distortion)。為了推導 XOR 模組之後的區塊 SoD，在查表次模組中，藉由計數在臨時的暫存器(temporary register)中位元值為 1 的數目，使用查表運算。最後，本發明在 SoD 比較次模組中比較全部的 SoD 值，在個別的 4 個區塊中，一一決定那一個運動向量是最佳的。在目前畫面的搜尋範圍內全部區塊都已完成匹配之後，最小 SoD 值的運動向量即為最終的運動向量。

除了可提供特殊的指令能對 64 位元暫存器，同時處理更多的連續區塊和 SIMD 結構中的暫存器之外，兩種結構的最佳運動向量搜尋流程是相似的。因此，底下只詳細說明 x86 系統中每一搜尋階層相關的步驟如何獲得如此的平行度。平行 XOR 區塊匹配模組的最佳化步驟將在第一階中詳細說明。其他階層的相關描述將因其相似性而被省略。

在第一階段中，因為每一個區塊皆已依序指定初始位置，所以本發明能在 x86 結構中同時計算 4 個區塊的 SoD 值，並且在 MMX 技術中以一系列一列的方式同時導出 8 個區塊的 SoD 值。底下 5 個步驟概述達到平行度的最佳化：

☒ 續次頁

步驟 1(載入)：因為本發明同時處理目前畫面大小為 4×4 的 4 個鄰近區塊，每一列都須要 16 個位元資料來計算 SoD 值。以搜尋範圍為 ± 3 的平行區塊匹配，需要載入資料倉的參考畫面的資料應該大於等於 22 個位元。

步驟 2(位元對位)：對於初始載入的資料，本發明將參考區塊與圖 2 所示之第一階相對於目前區塊 +3 個水平偏移量作對位。為了移到下一查核點，藉由將暫存器中的位元向右移一，而重覆使用重疊區的位元。

步驟 3(XOR)：圖 2 說明按列匹配法，其中每一四方形代表一個位元。因為 SINT 包含 16 個位元，因此只有參考暫存器的較低的 16 個位元用來與目前資料作 XOR 比較。XOR 的結果暫時儲存在另一個 SINT 暫存器中。對於 4 個區塊，分別需要 3 個 4 位元的移位(SHIFT)和 3 個且(AND)運算以得到 SoD 值，其中且運算係作為遮罩(mask)用。

步驟 4(查表)：Diff 係一個 SINT 暫存器，其中暫存第 n 個區塊的目前列的對應 SoD 值是以下列公式來算出：

$$SoD(u, v)_n = T[Diff \gg [4 \times (4 - n)] \& (0000\ 0000\ 0000\ 1111)]$$

每一個區塊的總 SoD 值等於每一列的 SoD 值的加總。

步驟 5(SoD 比較)：經由比較全部的 SoD 值，本發明能決定那一個運動向量對 4 個區塊是最佳的。

在第二階中，本發明使用與第一階相似的封包方法，將兩個連續的區塊儲存在一個 32 位元的資料倉中。對於圖 3 中的運動偏移模組 307 所得到的每一個事先定義的查核點，第二階的動態預測以一系列一列的方式計算出區塊差。對於目前的列，本發明不僅需要將前置定義的搜尋範圍內所需要的位元載入暫存器中，也需要將暫存器位移 β 個位元以便和參考資料作對位。 β 為目前運動向量所指示的水平偏移量。對於連結的查核點，需要額外的移位運算來對位參考資料。圖 3 說明第二階搜尋的位元對位過程。在第三階中，在 x86 結構中包含載入和位元模組的最佳化說明如下。在載入模組中，一個 32 位元的暫存器 A 儲存區塊 X 和上一個區塊($X-1$)相同列的參考資料。另一個 32 位元的暫存器 B 儲存區塊($X+1$)相同的列。這兩個暫存器皆含有特定搜尋窗中參考資料的部分位元，如圖 7 所示。為了對位參考資料，水平的預測運動向量在初始時被設定，例如設定為-10。暫存器 A 向左移 10，而暫存器 B 則向右移 6，以達到初始搜尋位置。本發明以一個簡單的 XOR/OR 運算將兩個暫存器的內含值結合。在最後的步驟中，參考位元被對位，以匹配初始搜尋位置+2。當移到同一列的下一個位置時，為了作匹配，本發明需要一個額外的移位運算來將這對資料作同步。

□ 續次頁

底下描述本發明在心跳陣列結構的實施例。每一塔層的全二元化表示法將儲存空間由 N^2 個位元組(byte)減少到 N^2 位元，因此，可用幾組列向量或行向量來儲存。由於向量係連續地儲存，本發明經由二元化資訊管線有效率地擷取資訊，並且對每一個匹配處理，重疊其處理的時間。因為在搜尋過程中目前區塊是固定的，在 ASIC 的實施例中，參考資料能以管線的形式擷取和傳遞。簡言之，利用心跳陣列來設計本發明的動態預測的硬體實施例是有好處的。

在圖 2、圖 3 和圖 4a 中，在第一階段、第二階段和第三階段搜尋裡，平行的 XOR 區塊匹配模組是以心跳陣列來作最佳化。圖 8 說明使用心跳陣列，對每一查核點的欄，平行區塊匹配的空間-時間表示式，其中每一個粗點(bold dot)表示一個處理單元，且 XOR 區塊匹配的空間-時間表示式定義為 $SoD(u, v) = \sum_{(x, y) \in Block} [S_{i,t}(x, y) \oplus S_{i,t-1}(x+u, y+v)]$ 。

圖 8 中的 $S_{i,t}(..)$ 表示目前畫面在時間 t 第 l 階的二元化表示式。 $S_{i,t-1}(..)$ 表示前一個時間畫面之相同的塔層上的參考資料。為了區塊匹配，區塊的大小設為 $N \times N$ ，而搜尋範圍設為 $\pm R$ 。從目前區塊到對應的參考區塊的運動向量以 (u, v) 表示之。

目前區塊和參考區塊之二元化資料可轉置到處理器

☒ 續次頁

中，並依一系列一列的方式將 XOR 區塊匹配法的差值作加總，以算出最終的 SOD 值。對於目前或參考畫面的每一區塊而言，每一列的二元化資料皆儲存成一維的向量。由目前和參考區塊的每一對向量送到處理單元 (processing element, PE) 以便計算 SoD 值。目前區塊更進一步經由心跳陣列送到下一個處理單元。因此，對於查核點的每一欄，本發明得到 $(2R+1)$ 個最終的 SoD 值，如圖 8 所示。

為了涵蓋所有大小為 $(2R+1)^2$ 的查核點，本發明以管線方法計算和比較 SoD 值，其中管線的每一個處理單元，在搜尋視窗中同一欄裡，處理參考區塊的一特定的列。由於管線的緣故，本發明能即時依序地處理搜尋視窗內的每一欄。亦即，本發明檢查位於搜尋視窗中第一欄的每一查核點，並由這此查核點中挑選出最小的 SoD 值者，以找出最佳的候選向量。利用陣列中所有的處理單元，管線掃描接續的 $2R$ 個欄，以取得所有搜尋點中，具最小的 SoD 值的最終候選向量。因此得到目前區塊的最終運動向量。

管線結構需要 $(2R+1)$ 個處理單元， $(2R+N)*(2R+1)$ 個週時 (cycle)， $(2R+N)*(2R+1)$ 個記憶體存取運算以取得參考資料，和 $N*(2R+1)$ 個記憶體存取運算以便由晶片上記憶體載入目前資料來計算每一個大小為 N^2 位元的區塊的

ⓧ 續次頁

$(2R+1)^2$ 個 SoD 值。其中，需要額外的 $2R$ 個週時(cycle)，用來作管線的初始化(initialization)，並且，每一個記憶體存取運算擷取 N 個位元的參考區塊和 N 個位元的目前區塊。建造 $(2R+1)$ 個處理單元的總閘數(gates counts)是小，但對記憶體的存取效率卻造成挑戰。

為了進一步減少記憶體擷取的潛在問題，本發明利用心跳陣列提出一二維(2-D)平行區塊匹配結構。此二維結構同時擷取參考視窗內所有 $(2R+N)^2$ 個位元到晶片上記憶體，以避免載入連續的參考區塊中重疊的位元，而造成浪費。然後，本發明從大小為 $(2R+N)$ 個位元的每一列，將每一組 N 個序列位元(serial bits)反多工(de-multiplexes)至對應的管線。每一對區塊的所有 SoD 值皆在一個處理單元中計算。此處理單元的實施例將在圖 9 中詳細的電路來作說明。

圖 9 說明平行二維區塊匹配的詳細的實施例。圖 9 中的平行二維區塊匹配的裝置包含一個記憶體₁模組(storage module) 901、一個延遲模組(delay module) 903、 $(2R+N)^2$ 個處理單元、一個多工模組(MUX) 905、一個晶元(on-chip)記憶體模組 907，和一個記憶體₂模組 909。記憶體₁模組 901 儲存目前區塊的二元化資料。延遲模組 903 接收來自目前區塊的二元化資料的列，並且在下一個時脈延遲(clock cycle delay)將該列送出。參考圖 9，

X 續次頁

$(2R+N)^2$ 個處理單元被安排成 $(2R+N)$ 列乘 $(2R+N)$ 欄的二維陣列，並且接收來自目前區塊和參考區塊的二元化資料的列、平行運算多位元的 XOR 運算、利用解碼器計算出位元值為 1 的個數，以及最後累積位元值為 1 的總數作為 SoD 值。多工模組 905 接收來自參考區塊的二元化資料的 $(2R+N)$ 個位元，並且將每 N 個位元分配到處理單元的每一列。MUX 接收到的 $(2R+N)$ 個位元從記憶體₂ 模組 909 被擷取並存至晶元(on-chip)記憶體模組 907。

如圖 9 中所示，被安排成 $(2R+N)$ 列乘 $(2R+N)$ 欄之二維陣列的 $(2R+N)^2$ 個處理單元更包含一個 XOR 模組 911、解碼器(decoder)模組 913、一個加總(summation)模組 915、兩個延遲(delay)模組 917 和 918，以及一個開關(switch)模組 919。

解碼器 913 用來計算每一筆資料中 1 的個數。標示 'D' 的區塊為延遲單元(delay elements)。儲存在內迴路(inner loop)的延遲單元中的 SoD 值累加解碼器 913 的輸出值。同時匹配的區塊之全部 SoD 值再以一比較器作比較，圖 9 中沒有標示此比較器。在處理完搜尋區域內所有位置之後，就得到具最小 SoD 的運動向量。圖 9 中的 $S_{li}(,..)$ 表示目前畫面在前一個時間畫面之相同的塔層上的參考資料。為了區塊匹配，搜尋範圍設為 $\pm R$ 。從目前區塊到對應的參考區塊的運動向量以 (u, v) 表示之。在解碼器 913 之旁

ⓧ 續次頁

的延遲模組'D'為搜尋範圍內對應的位置儲存累加的 SoD 值。

處理單元平行地執行數個位元的 XOR 運算、利用解碼器算出位元值為 1 的個數，並且最後累計位元值為 1 的總個數作為 SoD 值。處理單元、目前區塊和參考區塊彼此間的關係能以圖 9 之區塊圖來表示。輸入到每一個處理單元的區塊的維度與目前塔層的區塊維度有關。例如，第一階的區塊維度為 4，而到第三階則增加到 16。在二維平行結構中，本發明需要 $(2R+1)^2$ 個處理單元、 $(2R+N)$ 個週時、 $(2R+N)$ 個記憶體存取運算以取得參考資料和 N 個記憶體存取運算，以便由晶元記憶體載入目前資料以得到搜尋範圍為 R 且每一區塊大小為 N^2 位元的 SoD 值。每一次的記憶體存取皆從晶元記憶體擷取參考區塊的 $(2R+N)$ 個位元和目前區塊的 N 個位元。若與 1-D 管線作比較，即可發現計算 SoD 值的速度為 $(2R+1)$ 倍，並且每一筆參考資料只從晶元記憶體中擷取一次，而這是將位元載入到系統中所需要最少的次數。由於本發明的動態預測，對每一階只需要一個小的搜尋範圍，並且實現二維管線結構所需增加的閘數也在一個合理的範圍。雖然本發明提出了一個實施例，但在將動態預測的區塊圖映照(map)到實際的硬體上，諸如 ASIC 或 FPGA 晶片，可能會更有效率。

圖 10 說明傳統的全搜尋('FS')法以及本發明含與不含硬體加速器的動態預測法的計算複雜性和匯流排頻寬。畫面的大小為 352x288 個像素而畫面的速率為每秒 30 個畫面(fps)。底下首先分析本發明在 x86 結構下的動態預測之計算複雜性，此結構分別具有和不具有圖 2 至圖 4 所使用心跳陣列之平行 XOR 區塊匹配模組的實施例。此分析係基於一個單一的區塊如何來計算其運動向量，以及對一個特定的運動速率，其記憶體每秒的存取次數。符號 W 、 H 、 $\pm R$ 和 F_r (fps) 分別代表畫面寬度、畫面高度、搜尋範圍以及每秒的畫面數。區塊大小設定為 16x16，這個值是在視訊壓縮標準中常用到的。

為了容易分析在第二階範圍 $([R_{\min}^x, R_{\max}^x], [R_{\min}^y, R_{\max}^y])$ 內的動態搜尋的複雜性，使用最大的動態搜尋範圍作為最壞情況的考量。對於每一維度，全部畫面的最大的動態搜尋範圍皆依下列公式找出。對於目前序列每一畫面 i 中的每一區塊 j ，第二階搜尋之最大的動態搜尋範圍的邊界為

$$r_{\max}^x = \max\{(R_{\max}^x)_{i,j}\},$$

$$r_{\min}^x = \min\{(R_{\min}^x)_{i,j}\},$$

$$r_{\max}^y = \max\{(R_{\max}^y)_{i,j}\},$$

$$r_{\min}^y = \min\{(R_{\min}^y)_{i,j}\}.$$

對每一個區塊，使用絕對差之和(sum of absolute difference, SAD)的傳統的全搜尋法需要在搜尋視窗內處理全部 $4R^2$ 個搜尋點。每個位置需要 $16^2 \times 3$ 個運算，其中 3 個運算為一個減法、一個絕對值和一個加法運算。因此，為了取得一個單一運動向量，標示為 'FS' 的全搜尋的計算複雜性每一巨集區塊約為 $\delta_{FS} = 3072 \times R^2$ 個運算。以理論上最好的情況為基礎，在 x86 系統中使用 32 位元暫存器作全搜尋，為了計算一個運動向量，每一巨集區塊 (macroblock) 約需 $\bar{\delta}_{FS} = \frac{1}{4} \delta_{FS} = 768 \times R^2$ 個運算。

對於相同的搜尋區域，以本發明所提出的結構，但不用二維心跳陣列來計算一個運動向量，則每一巨集區塊的總運算數(標示為 $ABME_H_A$)為

$\delta_{ABME_H_A} = \delta_C + \delta_{Lv1} + \delta_{Lv2} + \delta_{Lv3}$ ，其中 δ_C 、 δ_{Lv1} 、 δ_{Lv2} 和 δ_{Lv3} 分別表示建造和二元化塔層的每一階動態搜尋的運算數。若搜尋視窗的大小為 $4R^2$ 且以 XOR 運算來匹配大小為 16×16 的巨集區塊，則第一階的搜尋範圍減少到 $(R/2-1)^2$ 且從第一階到第三階的區塊大小分別為 4^2 、 8^2 和 16^2 。每一個像素的二元化過程平均需要 4.5 個運算。因此，建造二元化塔層每一區塊需要 $\delta_C = (4^2 + 8^2 + 16^2) \times 4.5 = 1512$ 個運算。

至於區塊匹配過程，由於每一階資料儲存的技術都不同，每一階所需要的運算都個別分析。在第一階裡，每一個儲存在 4 SINT 倉中 4 個區塊的包裝，形成一個匹配單元(matching unit)，此匹配單元包含目前列中的資料

☒ 續次頁

，而這 4 個區塊依序儲存在目前列中。為了完成 4 個 SoD 值的推導，對搜尋視窗中每一區塊列需要 15 個運算。此 15 個運算包含一個暫存器移位運算作 XOR、3 個且運算、3 個移位運算來求得 4 個依位元的 SoD 值、4 個查表運算，和 4 個加法運算以同步累加 4 個巨集區塊的 SoD 值。因此，每一巨集區塊在第一階的總運算數為

$$\delta_{Lv1} = (15 \times 4) \times \left(\frac{R}{2} - 1\right)^2 \times \frac{1}{4} = 15 \times \left(\frac{R}{2} - 1\right)^2。$$

第二階的計算分析類似第一階，但是具有較大的區塊和動態的搜尋範圍，後者係由目前畫面和先前的畫面的 6 個候選的運動向量所算出。在第二階裡，每一個儲存在 2 SINT 倉中兩個區塊的包裝，形成一個匹配單元，此匹配單元包含目前列中的資料，而這兩個區塊依序儲存在目前列中為了完成兩個 SoD 值的推導，對搜尋視窗中每一區塊列，需要 7 個運算。此 7 個運算包含一個暫存器移位運算作 XOR、一個且運算、一個移位運算來求得兩個依位元的 SoD 值、兩個查表運算，和兩個加法運算以同步累加兩個巨集區塊的 SoD 值。搜尋位置的數目為 $(r_{\max}^x - r_{\min}^x + 1) \times (r_{\max}^y - r_{\min}^y + 1)$ 且 8 SINT 倉涵蓋兩個大小為 8×8 的第二階區塊的全部的列。因此，第二階搜尋，每個區塊需要 $\delta_{Lv2} = 8 \times 7 \times (r_{\max}^x - r_{\min}^x + 1) \times (r_{\max}^y - r_{\min}^y + 1) \times \frac{1}{2}$ 個運算。

第三階的計算分析類似第二階，但是具有較大的區塊

和固定的搜尋範圍。搜尋位置的數目為 25 並且 16 SINT 倉涵蓋一個大小為 16×16 的第三階區塊的全部的列。因此，在第三階搜尋裡，每個區塊需要 $\delta_{Lv3} = 16 \times 3 \times 25 = 1200$ 個運算。

建造二元化塔層，每個區塊需要 $\delta_C = (4^2 + 8^2 + 16^2) \times 4.5 = 1512$ 個運算。在第一階裡，每一個巨集區塊的總運算數 δ_{Lv1} 為 $15 \times (R/2 - 1)^2$ 。在第二階和第三階裡，每一個區塊的總運算數 δ_{Lv2} 和 δ_{Lv3} 分別為 $28 \times (r_{\max}^x - r_{\min}^x + 1) \times (r_{\max}^y - r_{\min}^y + 1)$ 和 1200。因此，'ABME_HA' 的總運算數為

$$\begin{aligned} \delta_{ABME_HA} &= \delta_C + \delta_{Lv1} + \delta_{Lv2} + \delta_{Lv3} \\ &= 2712 + 15 \times (R/2 - 1)^2 + 28 \times (r_{\max}^x - r_{\min}^x + 1) \times (r_{\max}^y - r_{\min}^y + 1) \end{aligned}$$

在第一階，第二階和第三階的平行 XOR 區塊匹配模組使用二維心跳陣列，稱為 'ABME_HA_HW'，此用法將每一巨集區塊的總運算數減少到

$$\begin{aligned} \delta_{ABME_HA_HW} &= \delta_C + \delta'_{Lv1} + \delta'_{Lv2} + \delta'_{Lv3} \\ &= 1512 + \left(\frac{R}{2} - 1 + 4\right) + (r_{\max}^x - r_{\min}^x + 8) + (5 - 1 + 16) = 1542 + \left(\frac{R}{2}\right) + (r_{\max}^x - r_{\min}^x + 1) \end{aligned}$$

比較 δ_{ABME_HA} ， $\delta_{ABME_HA_HW}$ 和 $\bar{\delta}_{FS}$ 三者的值，可發現本發明優於全域搜尋法，此全域搜尋法在各種不同的搜尋範圍中，與圖 10 所示的計算複雜度是一致的。

底下分析從目前畫面和參考畫面載入資料的總記憶

體頻寬(total memory bandwidth)。全搜尋法的總頻寬消耗為每秒 $\zeta_{FS} = (\zeta_{FS_Cur} + \zeta_{FS_Ref}) \times F_r$ 個位元組，其中 ζ_{FS_Cur} 和 ζ_{FS_Ref} 分別為從目前畫面和參畫面擷取資料的記憶體頻寬。假設目前區塊被同時載入含 16×16 UCHAR 倉的晶片記憶體。因此要擷取大小為 $W \times H$ 的目前畫面的資料需要 $\zeta_{FS_Cur} = W \times H$ 個位元組。

假設參考區塊被同時載入含 $(16 + 2R)^2$ UCHAR 的晶片記憶體。當移動到下一個區塊時，重疊區的資料被重覆使用，而頻寬需求為 $16 \times (16 + 2R)$ 。因此，為了從參考畫面完全載入資料需要額外的

$\zeta_{FS_Ref} = \frac{H}{16} \times \left[(16 + 2R)^2 + \left(\frac{W}{16} - 1 \right) \times (16 + 2R) \times 16 \right]$ 個位元組。第一項表示第一個搜尋視窗，由於記憶體困在管線的初始階段，因此需要更多的運算。其餘的搜尋視窗，因為與其先前的視窗重疊，故只需較少的運算即可。

在使用二維心跳陣列之含與不含硬體加速器的 x86 系統中，本發明的動態預測之總計記憶體頻寬分別為 $\zeta_{ABME_HA} = (\zeta_{Cur} + \zeta_{Lv1} + \zeta_{Lv2} + \zeta_{Lv3}) \times F_r$ 和 $\zeta_{ABME_HA_HW} = (\zeta_{Cur} + \zeta'_{Lv1} + \zeta_{Lv2} + \zeta'_{Lv3}) \times F_r$ 個位元組。 ζ_{Cur} 項表示載入目前畫面所須的頻寬消耗。其餘的項 ζ_{Lv1} 、 ζ'_{Lv1} 、 ζ_{Lv2} 、 ζ_{Lv3} 和 ζ'_{Lv3} 分別表示擷取畫面所需的頻寬。

在使用二維心跳陣列之不含硬體加速器的 x86 系統中，載入目前畫面的每一個區塊的記憶體頻寬為

$$\zeta_{Cur} = \left(\frac{1}{16} + \frac{1}{4} + 1 \right) \times HW \times \frac{1}{8} \approx 0.164 \times W \times H。$$

基於暫存器中之資料的可重覆使用性，在每一畫面的搜尋視窗中載入資料，對每一塔層所需要的記憶體頻寬為

$$\zeta_{Lv1} = \left(\frac{H}{4 \times 4} \right) \times \left[\left(4 + \frac{R}{2} - 1 \right) \times \left(4 \times 4 + \frac{R}{2} - 1 \right) + \left(\frac{W}{4 \times 4} \times \frac{1}{4} - 1 \right) \times \left(4 + \frac{R}{2} - 1 \right) \times 16 \right] \times \frac{1}{8},$$

$$\zeta_{Lv2} = \left(\frac{H}{2 \times 8} \right) \times \left[\left(8 + r_{\max}^y - r_{\min}^y + 1 \right) \times \left(8 \times 2 + r_{\max}^x - r_{\min}^x + 1 \right) + \left(\frac{W}{2 \times 8} \times \frac{1}{2} - 1 \right) \times \left(8 + r_{\max}^y - r_{\min}^y + 1 \right) \times 16 \right] \times \frac{1}{8}$$

,

$$\zeta_{Lv3} = \left[(16+5)^2 + \left(\frac{W}{16} - 1 \right) \times 16 \times (16+5) \right] \times \frac{H}{16} \times \frac{1}{8}。$$

其中，為了易於計算第二階每一畫面的記憶體頻寬需求，搜尋範圍的最壞情況設定為在 x 和 y 維度皆為[-3,3]。

當第一、第二和第三階的匹配模組皆以二維心跳陣列結構來實施時，在搜尋視窗中載入參考資料所需要的記憶體頻寬，每畫面分別為

$$\zeta'_{Lv1} = \left(\frac{H}{4 \times 4} \right) \times \left[\left(4 + \frac{R}{2} - 1 \right)^2 + \left(\frac{W}{4 \times 4} - 1 \right) \times \left(4 + \frac{R}{2} - 1 \right) \times 4 \right] \times \frac{1}{8},$$

$$\zeta'_{Lv2} = \left(\frac{H}{2 \times 8} \right) \times \left[\left(8 + r_{\max}^y - r_{\min}^y + 1 \right) \times \left(8 + r_{\max}^x - r_{\min}^x + 1 \right) + \left(\frac{W}{2 \times 8} - 1 \right) \times \left(8 + r_{\max}^y - r_{\min}^y + 1 \right) \times 8 \right] \times \frac{1}{8}$$

，並且 $\zeta'_{Lv3} = \zeta_{Lv3}$ 。與全域搜尋法相較之下，本發明很明顯

☒ 續次頁

地降低記憶體頻寬。當搜尋範圍增大時，全域搜尋法的記憶體頻寬也隨之劇增。然而相對地，由於 ζ_{Lv1} 和 ζ'_{Lv1} 的增加，本發明的運動估算法的記憶體頻寬只是輕微增加而已。另外，全域搜尋法的記憶體頻寬對各種不同的R值有高敏感性，而本發明的運動估算法對於搜尋範圍的變動較不具敏感性。這是因為第一階的畫面大小是最小的，且來自連續區塊的資料能儲存在一個較大的倉中。這些較佳的性能與圖 10 的觀察不謀而合。

為了比較本發明和全搜尋法的實施效率，本發明使用 MPEG-4 參考視訊編碼器和一個大小為 16×16 的巨集區塊來作區塊匹配。示於圖 11 的實施效率的比較，其分析方法係基於下列因素：視訊序列(video sequences)、編碼條件(encoding conditions)、各種不同的解析度降低濾波器(decimator filter)、動態預測式，和以 PSNR 值表示之重建視訊的視覺品質。

CIF 格式的視訊序列包含 Coastgaurd、Forman 和 Akiyo，並且含 Container 和 Mother-Daughter 之 QCIF 格式的視訊序列用來作測試。6 個序列表現了每種空間和運動行為(motion activity)的特性。本發明更進一步測試兩個 CCIR 601 序列，包含 Table Tennis(桌球)和 Stefan，其中含有快速移動的物件(object)。在大尺寸的圖形中含有快速移動的物件被採用，以檢驗本發明之運動估算法的實

☐ 續次頁

施性能和計算的負載。每一原始序列由 300 個畫面組成。

至於編碼條件，每一序列皆依 MPEG 委員會所建議的條件下進行編碼。目標畫面(target frame)的速率為每秒 10 個畫面(fps)，且不同序列的位元速率的範圍為 10 kbps 至 2 Mbps。為了找出精確的運動向量，每一序列的搜尋範圍為 16。CCIR-601 序列的搜尋範圍增大至 32。

至於第二階搜尋的動態搜尋範圍，由經驗得知，大小為 $([r_{\min}^x, r_{\max}^x], [r_{\min}^y, r_{\max}^y]) = ([-3,3], [-3,3])$ 的範圍足以提供滿意的第二階運動向量。因此可作為圖 11 中的複雜性估算。

至於解析度降低濾波器，本發明使用三個二維濾波器和三個一維可分離的濾波器(separable filter)來分析各種不同的濾波器如何影響動態預測的編碼效率和二元化。三個二維濾波器分別記為 H_A 、 H_B 和 H_C ，並且分別為：

$$H_A = \frac{1}{4} \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}, H_B = \frac{1}{4} \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}, H_C = \frac{1}{4} \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}。$$

從濾波器 H_B 或 H_C 取得的畫面 $S_i(x, y)$ 比從濾波器 H_A 取得的畫面保留較多高頻的資訊。

☐ 續次頁

三個一維濾波器為三個在不同截斷頻率 20%，25% 和 30% 奈氏(Nyquist)頻率的 13-tap Hamming 濾波器。為了進一步降低濾波器的複雜性，每一個濾波器的係數形式為 $k \cdot 2^{-m}$ 的線性相位 FIR 濾波器。對於具 M-tap 的一維濾波器，亦即 $H = [h_1, h_2, \dots, h_M]$ ，其低通畫面的計算為

$$\bar{F}_l(x, y) = H(F_l(x, y)) = \sum_{d=1}^M h_d * F_l\left(x + d - \frac{M+1}{2}, y\right) = \left(\sum_{d=1}^M k_d * F_l\left(x + d - \frac{M+1}{2}, y\right)\right) \cdot 2^{-m}$$

，其中 $m = \log_2\left(\sum_{d=1}^M k_d\right)$ 。例如，係數形式為 $k \cdot 2^{-m}$ 的 5-tap 濾

波器 $H = [1, 4, 6, 4, 1] \cdot 2^{-4}$ 能夠只需用移位和加法運算來實施。

這兩種運算在各種現今電腦結構中只佔一個運算週時。因此，這類的濾波器能夠顯著的加快速度，雖然它並非以特定的濾波器設計方法來設計。三個一維濾波器為：

$$H_{20} = [-1, 0, 4, 15, 33, 49, 56, 49, 33, 15, 4, 0, -1]/256,$$

$$H_{25} = [-1, -2, 0, 11, 32, 55, 66, 55, 32, 11, 0, -2, -1]/256, \text{ 且}$$

$$H_{30} = [-1, -2, -4, 4, 30, 62, 78, 62, 30, 4, -4, -2, -1]/256 \text{ 。}$$

根據圖 10 的複雜性分析和圖 11 的實施效率的比較，與全搜尋法相較之下，即可發現本發明不但具有低運算複雜性和低記憶體頻寬消耗的優點，並且對於搜尋範圍的增大並不敏感。本發明也證明以比較好的二元化方

法，視覺品質能更進一步地獲得改善。依此特性，本發明可提供具有彈性的構造。系統設計者可依他們的系統能夠提供的下列因素來選擇二元化方法：可獲得的記憶體、計算能力、顯示器的解析度，或是資料匯流排的頻寬。例如，無線通訊的行動電話的特性為低計算能力、較差的顯示器解析度，以及較少的記憶體。因此，應使用最小的濾波器以得到最佳的執行速度且保留低解析度之顯示器下仍可接受的視覺品質。反之，對於較快速的機器，諸如現今的個人電腦或高檔的數位信號處理(DSP)系統，能夠使用較佳頻率響應的濾波器，因為它的計算能力負擔得起較複雜的計算。另外，為了不同大小的暫存器的特定平台，能夠開發各種不同的最佳化方法。因此，本發明的動態預測法比其它動態搜尋計算法具有較大的彈性。

本發明也舉例說明適用於數種硬體結構，諸如 x86、使用 MMX 技術的 SIMD 和心跳陣列，的特定平台最佳化。由運算數目來看，本發明的動態預測法也很適合在通用處理系統作軟體實施。它也能被實施在 ASIC 設計的平行管線，且在硬體設計階段允許在矽區(silicon area)、電力消耗和視覺品質間作取捨。因此，本發明的動估算法對在軟、硬體平台上的多媒體系統是靈活且有效率的。

唯，以上所述者，僅為本發明之較佳實施例而已，當

□ 續次頁

不能以此限定本發明實施之範圍。即大凡依本發明申請專利範圍所作之均等變化與修飾，皆應仍屬本發明專利涵蓋之範圍內。

【圖式簡單說明】

圖1a為根據本發明作為動態預測之三階全二元化金字塔型結構的區塊圖。

圖1b為根據本發明之動態預測的穩態查核模組。

圖1c為根據本發明之動態預測的狀態更新模組。

圖2為圖1之動態預測的第一階搜尋模組。

圖3為圖1之動態預測的第二階搜尋模組。

圖4a為圖1之動態預測的第三階搜尋模組。

圖4b說明本發明之XOR區塊匹配模組包含一查表次模組和一個依位元執行的差和次模組SoD。

圖5說明本發明之每一層二元值層的建構。

圖6說明二元化過程的平行處理。

圖7說明第三階搜尋之初始搜尋位置的位元調整。

圖8說明使用心跳陣列，本發明對於查核點的每一欄，平行區塊匹配的空間-時間的表示法。

圖9說明平行二維區塊匹配的詳細的實施例。

圖10說明傳統的全搜尋('FS')法以及本發明含與不含硬體加速器的動態預測法的計算複雜性和匯流排頻寬。

圖11說明本發明之動態預測相對於全搜尋之實施效率的

比較，其分析方法係基於下列因素：各種的編碼條件、濾波器，和以PSNR值表示的視覺品質。

圖號說明

101 二元化金字塔型結構	103 第一階搜尋模組
105 第二階搜尋模組	107 第三階搜尋模組
109 細調模組	111 濾波模組
113 二元化模組	115 解析度降低模組
117 穩態查核模組	119 狀態更新模組
121 計數器	122、123和124 比較器
201 資料載入模組	203 位元對位模組
205 (平行) XOR區塊匹配模組	
301 資料載入模組	303 位元對位模組
305 (平行) XOR區塊匹配模組	307 運動偏移模組
401 資料載入模組	403 位元對位模組
405 (平行) XOR區塊匹配模組	
901 記憶體 ₁ 模組	903 延遲模組
905 多工模組	907 晶元記憶體模組

 續次頁

909 記憶體₂ 模組

911 XOR 模組

913 解碼器模組

915 加總模組

917 和 918 延遲模組

919 開關模組



申請專利範圍

1. 一種在視訊影像編碼中作為動態預測的裝置，包含有：
 - 二元化金字塔型結構建造模組，用來將視訊影像轉換成第一階、第二階和第三階二元值層；
 - 穩態查核模組，根據最終運動向量的重覆出現次數來決定該裝置係在穩態模式或正常模式；
 - 第一階搜尋模組，接收該第一階二元值層，且在第一階以一 ± 3 像素之精調視窗執行平行 XOR 區塊匹配，且為該正常模式產生一第一階運動向量；
 - 第二階搜尋模組，接收該第二階二元值層，且利用 6 個候選的運動向量來為該正常模式決定一個第二階運動向量；
 - 第三階搜尋模組，接收該第二階運動向量，且在第三階以一 ± 2 像素之精調視窗執行平行 XOR 區塊匹配，且為該正常模式產生一最終運動向量；
 - 細調模組，以一 ± 1 像素之精調視窗執行平行 XOR 區塊匹配，且為該穩態模式產生一最終運動向量；以及
 - 狀態更新模組，記錄和更新該最終運動向量的重覆出現次數。
2. 如申請專利範圍第 1 項所述之作為動態預測的裝置，其中該二元化金字塔型結構建造模組更包含一濾波模組、一二元化模組和一解析度降低模組。

3. 如申請專利範圍第 1 項所述之作為動態預測的裝置，
其中該第一階搜尋模組包含
 - 一資料載入模組，用來接收該第一階二元值層；
 - 一位元對位模組，與該資料載入模組耦合，用來在第一階以一 ± 3 像素之精調視窗與目前區塊和參考區塊對位；
 - 以及
 - 一平行 XOR 區塊匹配模組，用來根據最小的依位元執行的差和運算值的準則產生一第一階運動向量。

4. 如申請專利範圍第 3 項所述之作為動態預測的裝置，該
平行 XOR 區塊匹配模組更包含一查表次模組和一依位元
執行的差和次模組 SoD。

5. 如申請專利範圍第 1 項所述之作為動態預測的裝置，其
中該 6 個候選的運動向量分別對應右上區塊、上區塊、
左區塊、第一階目前區塊的初始運動向量、前一時間的
區塊、以及中央之 6 個運動向量。

6. 如申請專利範圍第 3 項所述之作為動態預測的裝置，其
中該第二階搜尋模組包含：
 - 一運動偏移模組，依據該 6 個候選的運動向量計算出運
動向量偏移量的搜尋範圍；
 - 一資料載入模組，接收該第二階二元值層與該運動偏移
模組之該搜尋範圍；

一位元對位模組，用來與目前區塊和參考區塊對位；以及
一平行 XOR 區塊匹配模組；

其中該位元對位模組和該平行 XOR 區塊匹配模組，由查
核對應到由該運動偏移模組計算出之該搜尋範圍之運
動向量，決定該第二階運動向量。

7. 如申請專利範圍第 6 項所述之作為動態預測的裝置，該
第二階搜尋模組之該平行 XOR 區塊匹配模組更包含一
查表次模組和一依位元執行的差和次模組 SoD。

8. 如申請專利範圍第 6 項所述之作為動態預測的裝置，其
中該第三階搜尋模組包含一資料載入模組，用來接收該
第三階二元值層和該第二階運動向量；

一位元對位模組，與該資料載入模組耦合，用來在第三
階以一 ± 2 像素之精調視窗與目前區塊和參考區塊對位；以及
一平行 XOR 區塊匹配模組，用來依據最小的依位元執
行的差和運算值的準則產生一最終運動向量。

9. 如申請專利範圍第 8 項所述之作為動態預測的裝置，該
第三階搜尋模組之該平行 XOR 區塊匹配模組更包含一查
表次模組和一依位元執行的差和次模組 SoD。

10. 如申請專利範圍第 1 項所述之作為動態預測的裝置，其
中該狀態更新模組更包含一計數器，用來記錄前一個畫

續次頁

面中該最終運動向量之重覆出現次數，若一相同的最終運動向量在該前一個畫面中重複，則該計數器加 1，若該計數器大於一預設之常數或得到一不同的最終運動向量，則該計數器歸零。

11. 如申請專利範圍第 10 項所述之作為動態預測的裝置，其中用來記錄該前一個畫面中該最終運動向量之重覆出現次數之計數器，若該計數器大於一預設之常數，該穩態查核模組設定該裝置為在該穩態模式。
12. 如申請專利範圍第 1 項所述之作為動態預測的裝置，其中該二元化金字塔型結構建造模組、該第一階搜尋模組、該第二階搜尋模組和該第三階搜尋模組係以管線結構來實施。
13. 如申請專利範圍第 1 項所述之作為動態預測的裝置，其中該二元化金字塔型結構建造模組、該第一階搜尋模組、該第二階搜尋模組和該第三階搜尋模組係以心跳陣列來實施。
14. 如申請專利範圍第 1 項所述之作為動態預測的裝置，其中該第一階搜尋模組、該第二階搜尋模組或該第三階搜尋模組包含一搜尋電路，該搜尋電路具有：
一第一儲存單元，儲存目前區塊的二元化資料；

- 一延遲電路，與該第一儲存單元耦合；
 - 一第二儲存單元，儲存參考區塊的二元化資料；
 - 一晶元記憶體模組，接收來自該第二儲存單元的資料；
 - 一多工模組，與該晶元記憶體模組耦合且提供複數個輸出；以及
 - 一二維心跳陣列，具有複數個心跳細胞列，每一心跳細胞具有一第一輸入端，用來接收由該延遲電路來的資料，在一心跳細胞列中，除第一心跳細胞外，每一心跳細胞具有一第二輸入端，與同一心跳細胞列中前一個心跳細胞的輸出端耦合；
- 其中每一心跳細胞列中的第一心跳細胞具有一第二輸入端，與該多工模組的輸出端耦合。

15. 如申請專利範圍第 14 項所述之作為動態預測的裝置，
- 其中每一心跳細胞包含有：
- 一解碼器；
 - 一 XOR 模組，具有一輸出端，與該解碼器的輸入端連結，一第一輸入端，與該心跳細胞的第一輸入端連結，和一第二輸入端，與該心跳細胞的輸出端連結；
 - 一第一延遲模組，連結在心跳細胞的第二輸入端和輸出端之間；
 - 一第二延遲模組；
 - 一加總模組，具有一第一輸入端，與該解碼器的輸出端耦合，和一輸出端，與該第二延遲模組連結；以及

一開關模組，用來控制該第二延遲模組之輸出端和該加總模組之第二輸入端間之連結。

16. 一種在視訊影像編碼中作為動態預測的方法，包含下列步驟：

- (a) 建造一二元化金字塔型結構，係將視訊影像轉換成第一階、第二階和第三階二元值層；
- (b) 根據最終運動向量的重覆出現次數來決定該方法係在穩態模式或正常模式；
- (c) 若該方法係在穩態模式，以一 ± 1 像素之精調視窗執行平行 XOR 區塊匹配，並且產生一最終運動向量；
- (d) 在該第一階二元值層執行第一階搜尋，係在第一階以一 ± 3 像素之精調視窗執行平行 XOR 區塊匹配，並且若該方法係在正常模式，則產生一第一階運動向量；
- (e) 若該方法係在正常模式，根據 6 個候選的運動向量在該第二階二元值層執行第二階搜尋，並且決定一個第二階運動向量；
- (f) 若該方法係在正常模式，根據該第二階運動向量在該第三階二元值層執行第三階搜尋，且在第三階以一 ± 2 像素之精調視窗執行平行 XOR 區塊匹配，並且產生一最終運動向量；以及
- (g) 記錄和更新該最終運動向量的重覆出現次數。

17. 如申請專利範圍第 16 項所述之作為動態預測的方法，

x 續次頁

其中該步驟(a)包含濾波、二元化和解析度降低。

18. 如申請專利範圍第 16 項所述之作為動態預測的方法，其中該步驟(d)係在第一階以一 ± 3 像素之精調視窗與目前區塊和參考區塊對位，並且根據最小的依位元執行的差和運算值的準則，使用一平行 XOR 區塊匹配來完成。

19. 如申請專利範圍第 16 項所述之作為動態預測的方法，其中該 6 個候選的運動向量分別對應右上區塊、上區塊、左區塊、第一階目前區塊的初始運動向量、前一時間的區塊、以及中央之 6 個運動向量。

20. 如申請專利範圍第 16 項所述之作為動態預測的方法，其中該步驟(e)包含：

(e1) 依據該 6 個候選的運動向量計算出運動向量偏移量的搜尋範圍；以及

(e2) 根據最小的依位元執行的差和運算值的準則，使用 XOR 區塊匹配，藉由查核在該搜尋範圍內對應到查核點之運動向量，來決定該第二階運動向量。

21. 如申請專利範圍第 16 項所述之作為動態預測的方法，其中該步驟(f)係以在第三階以一 ± 2 像素之精調視窗與目前區塊和參考區塊對位，以及依據最小的依位元執行的

差和運算值的準則使用一平行 XOR 區塊匹配，來產生一最終運動向量來完成。

22. 如申請專利範圍第 16 項所述之作為動態預測的方法，其中該步驟(g)利用一計數器來記錄前一個畫面中該最終運動向量之重覆出現次數，若一相同的最終運動向量在該前一個畫面中重複，則該計數器加 1，若該計數器大於一預設之常數或得到一不同的最終運動向量，則該計數器歸零。

23. 如申請專利範圍第 22 項所述之作為動態預測的方法，其中用來記錄該前一個畫面中該最終運動向量之重覆出現次數之計數器，若該計數器大於一預設之常數，則該方法係在該穩態模式。

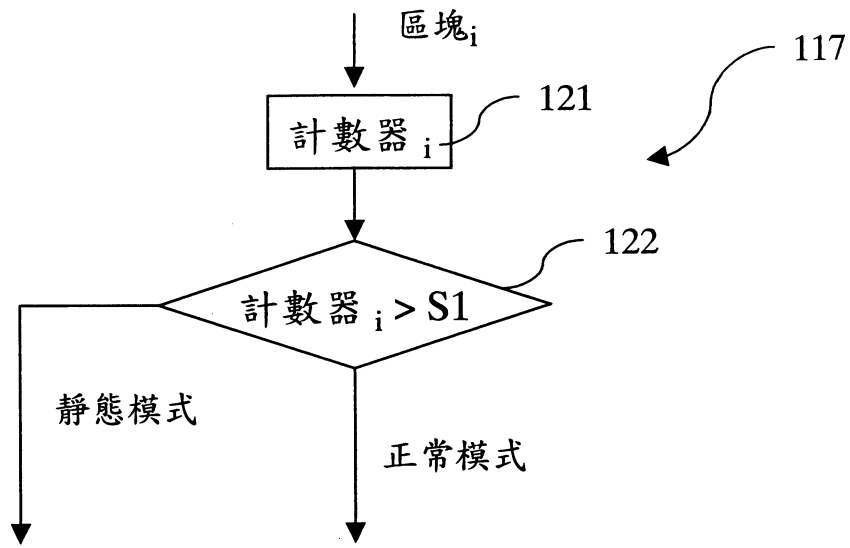


圖1b

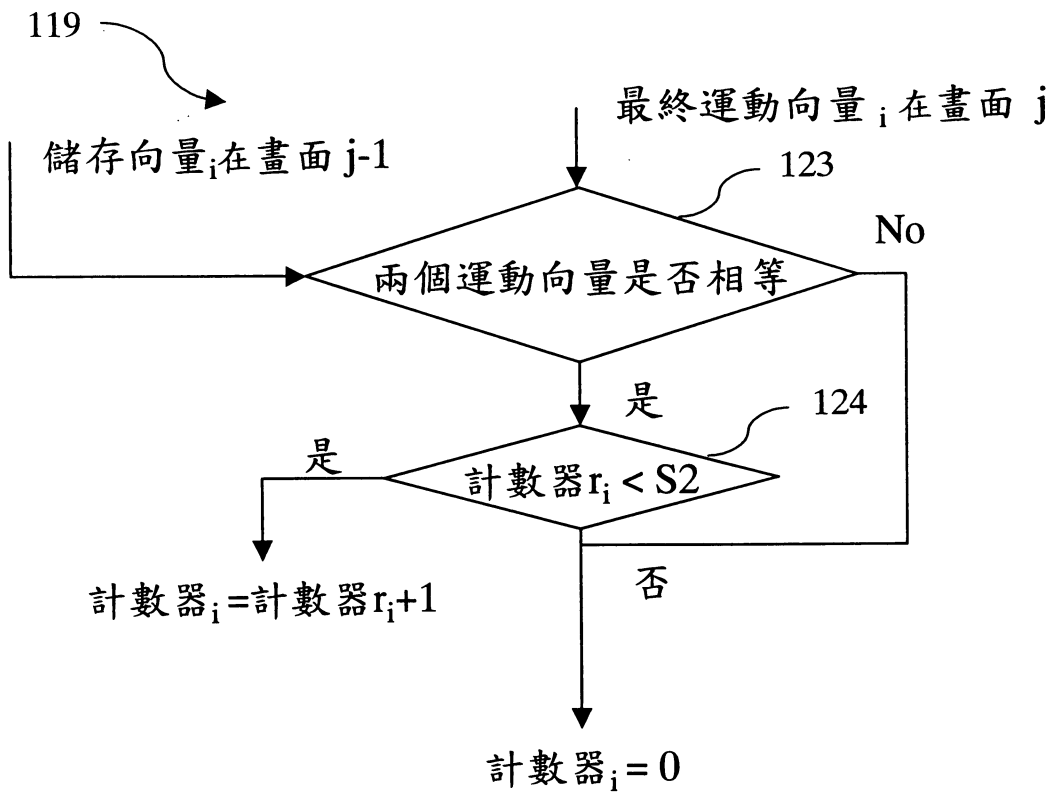


圖1c

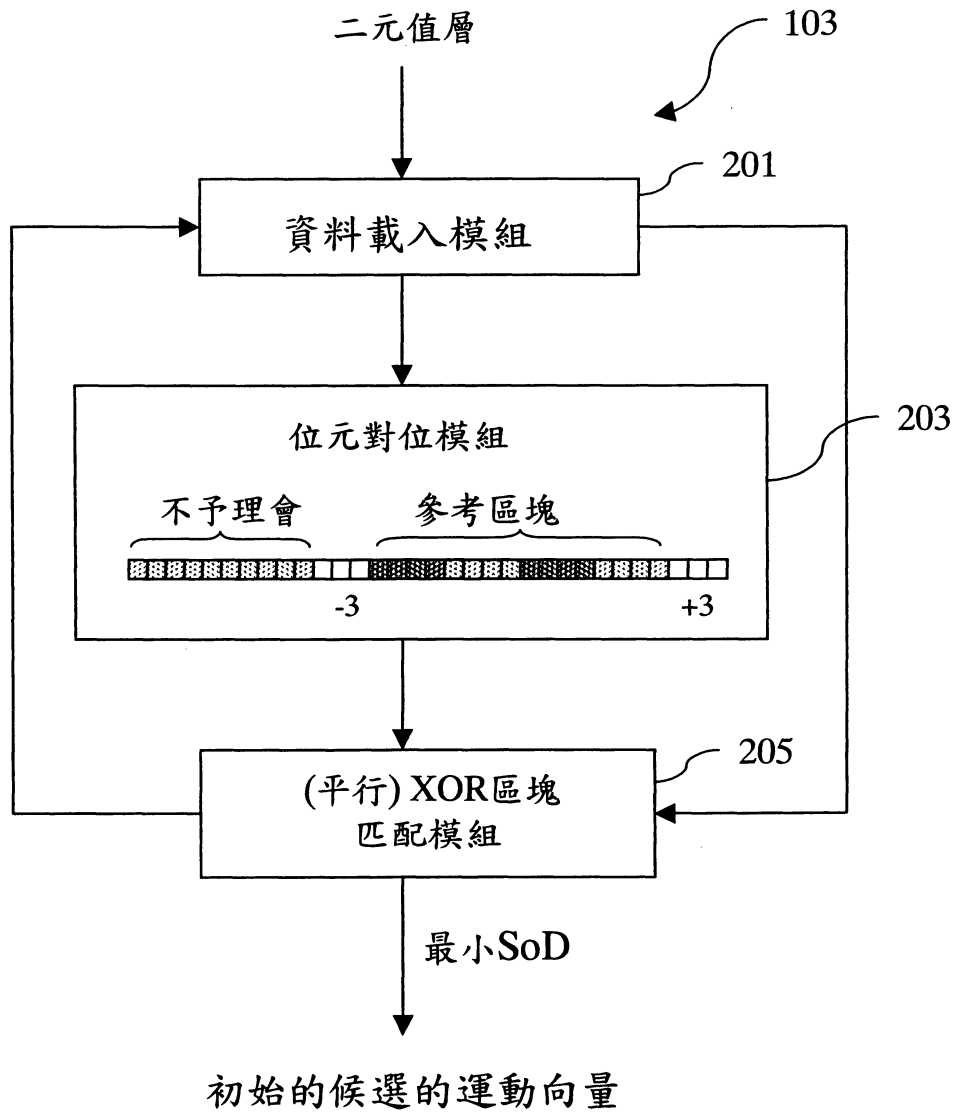


圖2

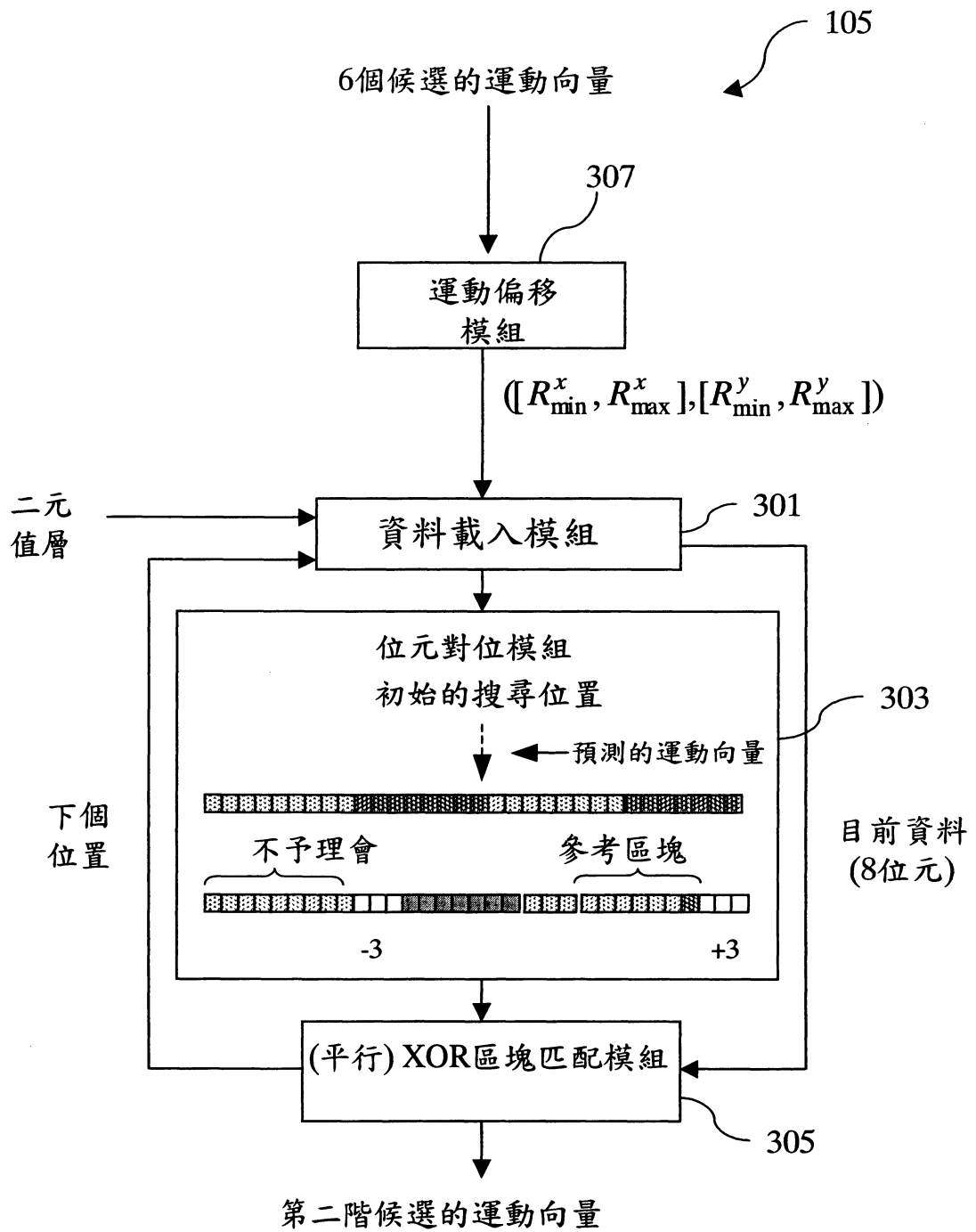


圖3

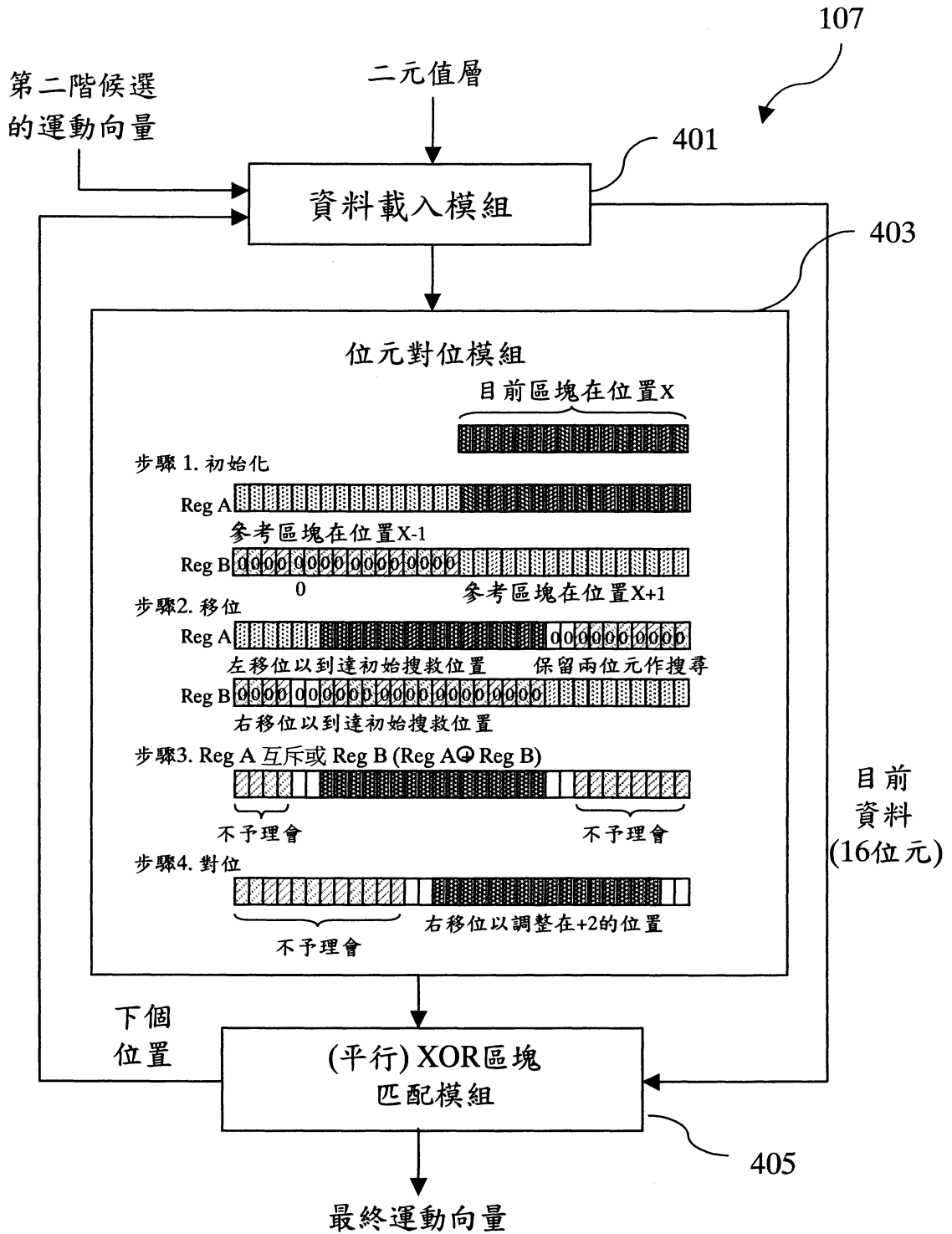


圖4a

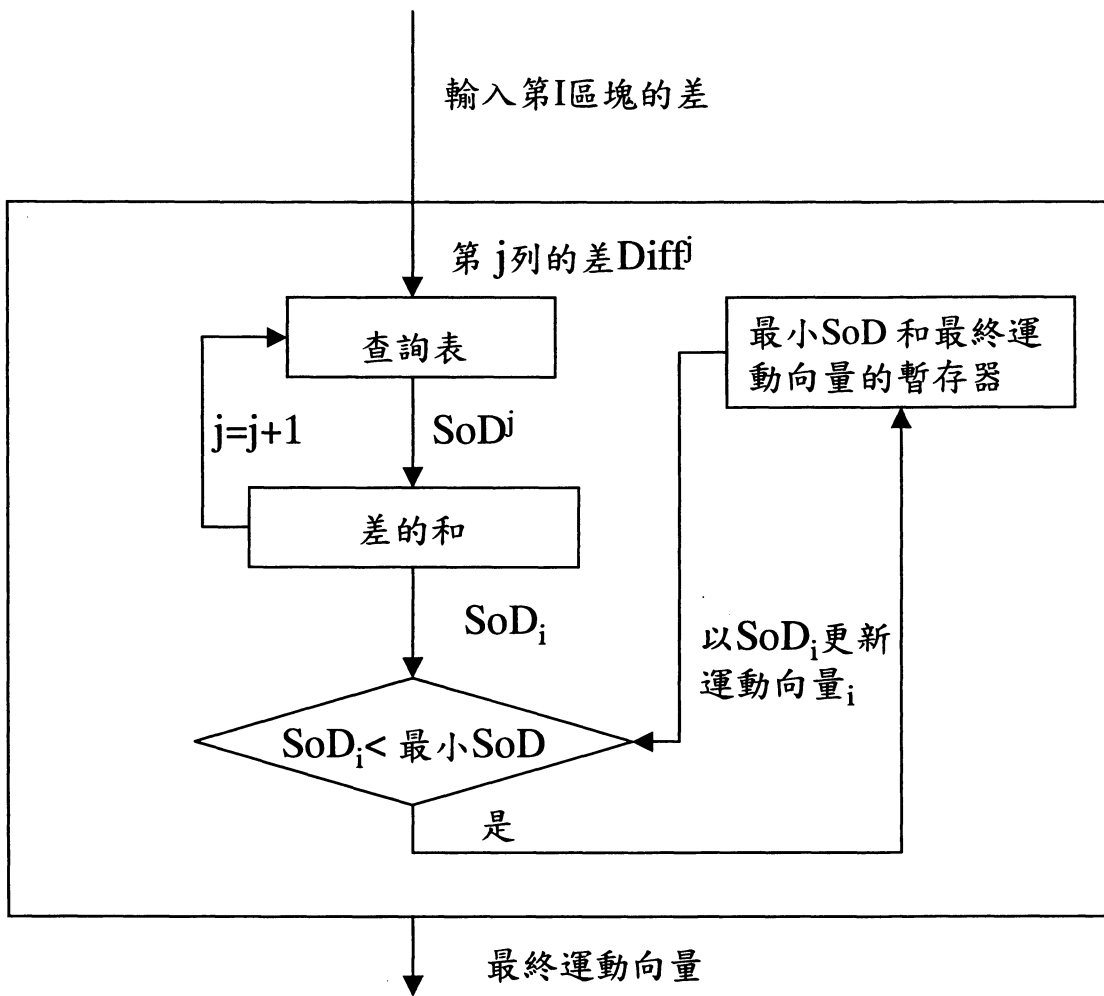
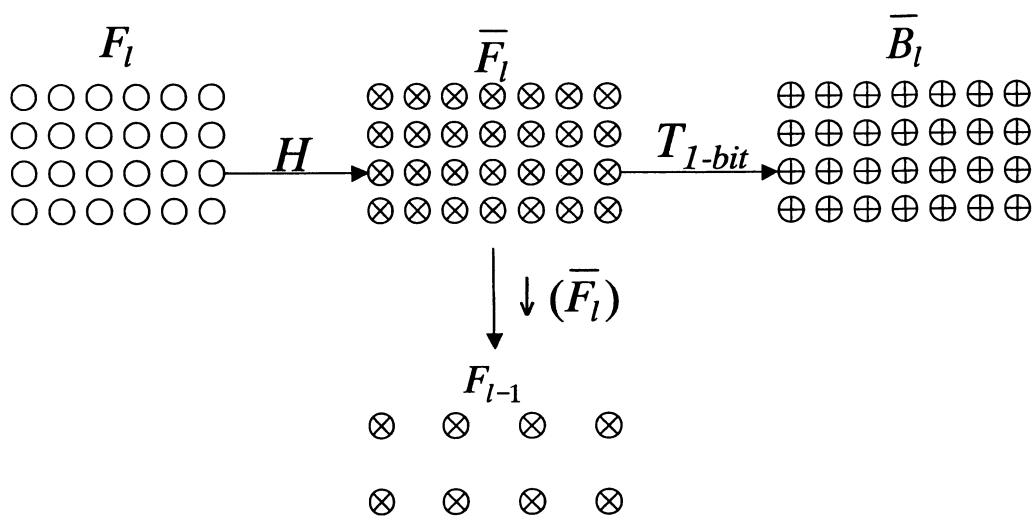


圖4b



- 原始八位元資料
- ⊗ 低通過濾後的八位元資料
- ⊕ 二元值資料

圖 5

畫面邊界

0	$F_i(0,0)$	$F_i(1,0)$	$F_i(2,0)$	$F_i(3,0)$	$F_i(4,0)$
0	$F_i(0,1)$	$F_i(1,1)$	$F_i(2,1)$	$F_i(3,1)$	$F_i(4,1)$
0	$F_i(0,2)$	$F_i(1,2)$	$F_i(2,2)$	$F_i(3,2)$	$F_i(4,2)$
0	$F_i(0,3)$	$F_i(1,3)$	$F_i(2,3)$	$F_i(3,3)$	$F_i(4,3)$

圖6

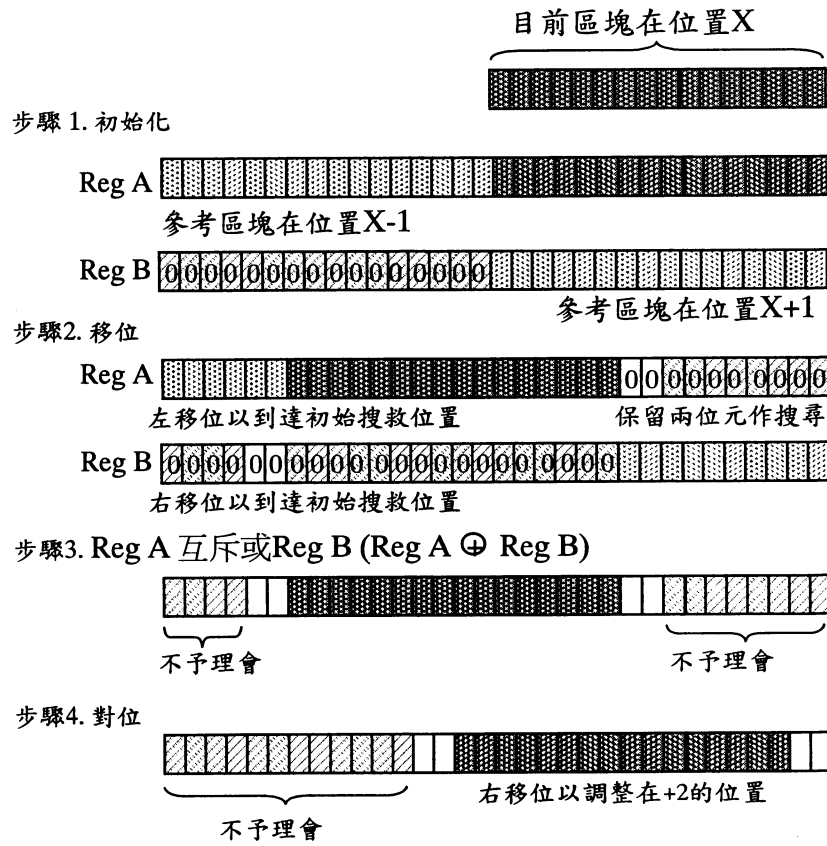


圖 7

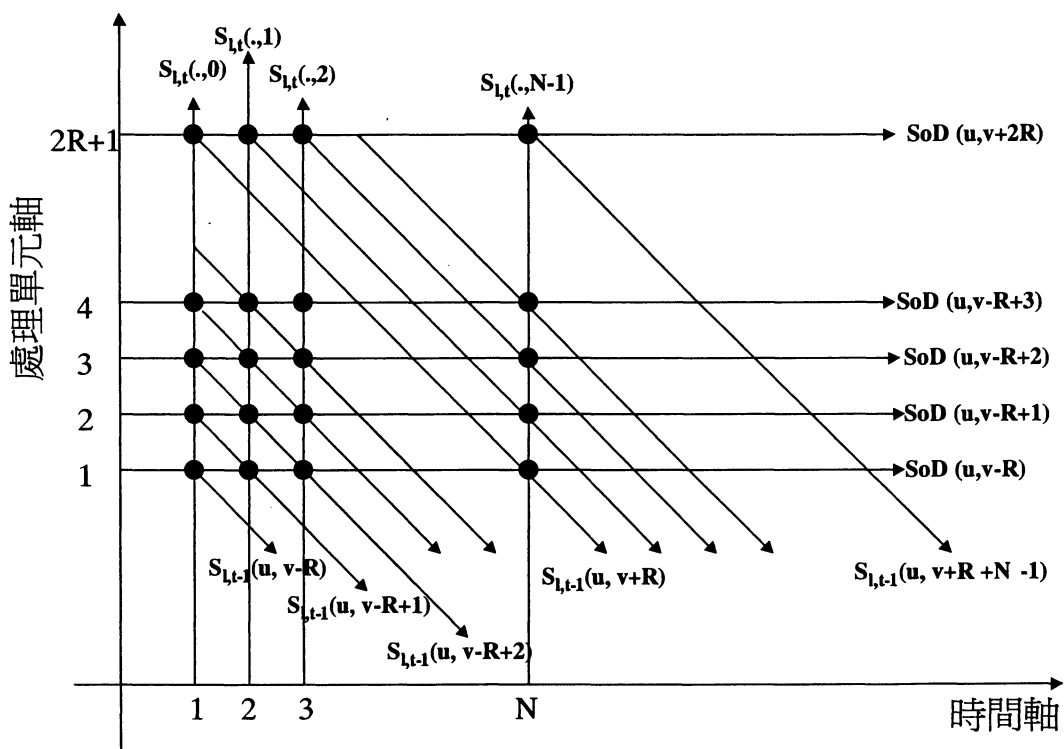


圖8

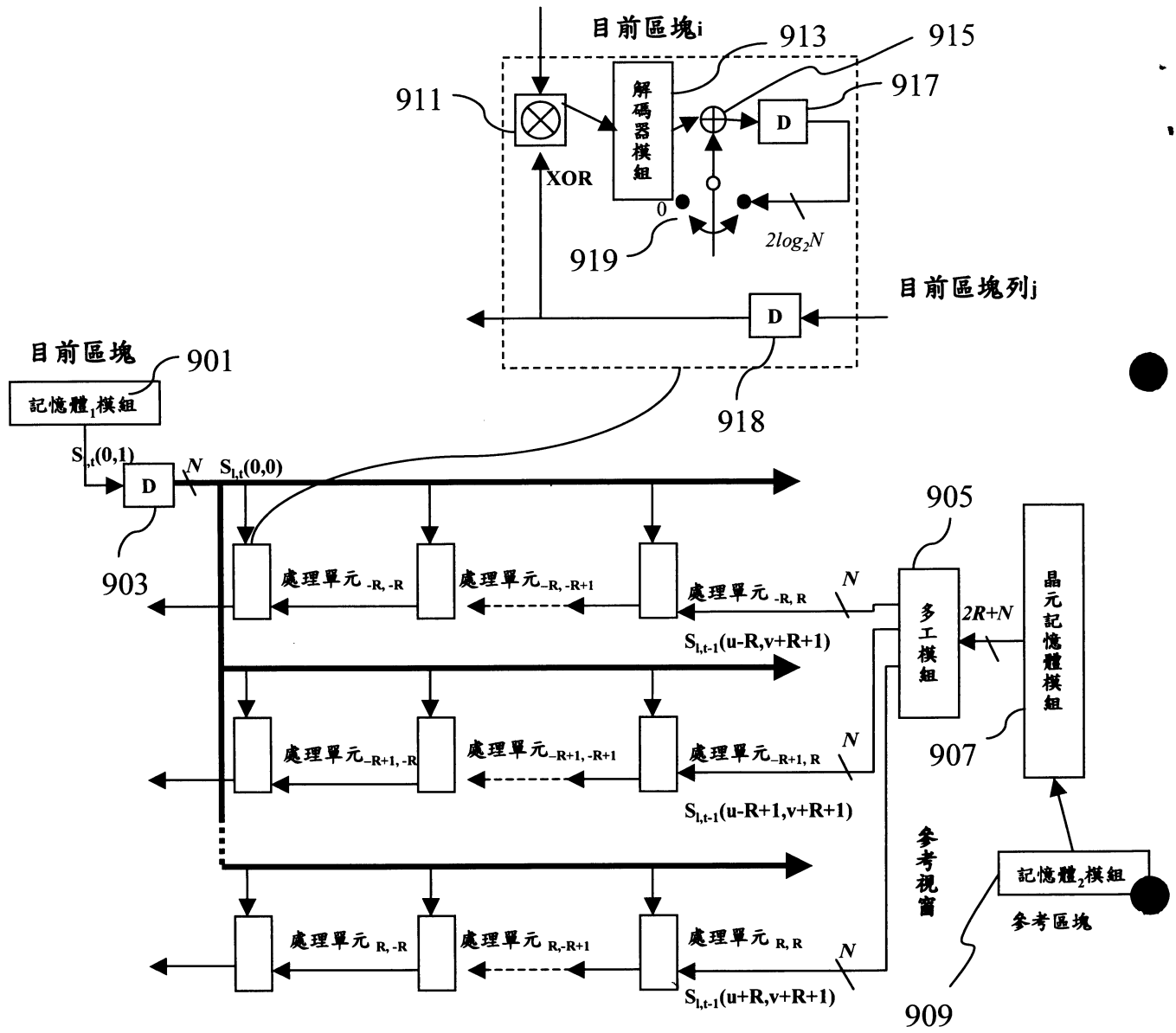


圖 9

視訊序列 (目標位元率, 畫面大小)	方法	Y_PSNR (dB)	U_PSNR (dB)	V_PSNR (dB)	總位元數	ΔY_PSNR (dB)
Mother-Daughter (24kbps, QCIF)	FS	34.86	40.59	41.12	238928	
	ABME_H _A	34.73	40.28	41.05	239072	-0.13
	ABME_H _B	34.74	40.30	40.99	239120	-0.12
	ABME_H _C	34.75	40.34	41.06	239904	-0.11
	ABME_H ₂₀	34.77	40.31	41.01	239016	-0.09
	ABME_H ₂₅	34.78	40.34	41.04	239200	-0.08
	ABME_H ₃₀	34.74	40.26	41.04	239064	-0.12
Container (10kbps, QCIF)	FS	29.84	37.39	36.64	98832	
	ABME_H _A	29.60	37.35	36.43	99216	-0.24
	ABME_H _B	29.62	37.47	36.43	99952	-0.22
	ABME_H _C	29.60	37.26	36.40	99144	-0.24
	ABME_H ₂₀	29.60	37.28	36.44	99304	-0.24
	ABME_H ₂₅	29.63	37.41	36.45	99256	-0.21
	ABME_H ₃₀	29.58	37.31	36.47	99176	-0.26
Coastguard (112kbps, CIF)	FS	27.10	38.98	41.51	1116152	
	ABME_H _A	26.73	39.13	41.64	1115072	-0.37
	ABME_H _B	26.87	39.07	41.82	1114904	-0.23
	ABME_H _C	26.90	39.03	41.54	1114568	-0.20
	ABME_H ₂₀	26.89	39.02	41.58	1115520	-0.21
	ABME_H ₂₅	26.95	39.19	41.81	1114800	-0.15
	ABME_H ₃₀	26.92	39.04	41.93	1113264	-0.18
Foreman (112kbps, CIF)	FS	30.19	36.79	37.53	1114624	
	ABME_H _A	29.56	37.05	37.83	1114912	-0.63
	ABME_H _B	29.76	37.08	37.89	1114864	-0.43
	ABME_H _C	29.78	37.12	37.95	1114784	-0.41
	ABME_H ₂₀	29.72	37.08	37.92	1114864	-0.47
	ABME_H ₂₅	29.82	37.06	37.87	1114792	-0.37
	ABME_H ₃₀	29.81	37.16	38.14	1114928	-0.38
Akiyo (112kbps, CIF)	FS	40.77	44.06	45.16	1115760	
	ABME_H _A	40.72	43.98	45.01	1115872	-0.05
	ABME_H _B	40.73	44.00	45.08	1115992	-0.04
	ABME_H _C	40.74	44.08	45.15	1115808	-0.03
	ABME_H ₂₀	40.74	44.02	45.11	1115600	-0.03
	ABME_H ₂₅	40.76	44.03	45.13	1115824	-0.01
	ABME_H ₃₀	40.72	43.96	45.04	45.04129	-0.05

圖 10

視訊序列 (目標位元率, 畫面大小)	方法	Y_PSNR (dB)	U_PSNR (dB)	V_PSNR (dB)	總位元數	ΔY_PSNR (dB)
Stefan (512kbps, CCIR601)	FS	29.55	34.22	33.96	5095128	
	ABME_H _A	29.36	34.33	34.03	5095072	-0.19
	ABME_H _B	29.45	34.40	34.11	5095440	-0.10
	ABME_H _C	29.46	34.39	34.10	5095344	-0.09
	ABME_H ₂₀	29.42	34.37	34.10	5095344	-0.13
	ABME_H ₂₅	29.45	34.39	34.11	5095640	-0.10
	ABME_H ₃₀	29.44	34.40	34.11	5095160	-0.11
Table Tennis (2Mbps, CCIR601)	FS	34.65	39.60	39.86	19901976	
	ABME_H _A	34.28	39.53	39.74	19901784	-0.37
	ABME_H _B	34.49	39.63	39.90	19901688	-0.16
	ABME_H _C	34.50	39.64	39.91	19901680	-0.15

圖 10 (續)

搜尋範圍	方法	複雜度(Complexity)		匯流排頻寬 (Bus Bandwidth)	
		運算次數 / Macroblock	對全搜尋法 之加速比	Mbytes / sec	百分比
16	FS	196608	1	12.39	100 %
	ABME_H _A	4819	40.80	1.26	10.17 %
	ABME_H _A _HW	1599	122.96	1.27	10.25 %
32	FS	786432	1	20.04	100 %
	ABME_H _A	7459	105.43	1.32	6.60 %
	ABME_H _A _HW	1607	489.38	1.34	6.67 %
64	FS	3145728	1	38.50	100 %
	ABME_H _A	18499	170.05	1.47	3.82 %
	ABME_H _A _HW	1623	1938.22	1.50	3.90 %

圖 11