



(21)申請案號：099131632

(22)申請日：中華民國 99 (2010) 年 09 月 17 日

(51)Int. Cl. : **H04N7/64 (2006.01)**(71)申請人：國立成功大學(中華民國) NATIONAL CHENG KUNG UNIVERSITY (TW)  
臺南市東區大學路 1 號

(72)發明人：雷曉方 LEI, SHEAU FANG (TW)；林俞伸 LIN, YU SHENG (TW)

(74)代理人：吳冠賜；蘇建太

(56)參考文獻：

CN 1523896A

US 2007/0014358A1

"A 2-stage Reference Picture Selection H.264 Encoder and a novel Non-Casual Whole Frame Concealment Algorithm" by S.Monsen etc.  
2008/12/31

審查人員：李國福

申請專利範圍項數：11 項 圖式數：14 共 0 頁

(54)名稱

於可調式影像解碼的全畫面錯誤掩蓋方法

METHOD OF FRAME ERROR CONCEALMENT IN SCABLE VIDEO DECODING

(57)摘要

本發明提供一種於可調式影像解碼的全畫面錯誤掩蓋方法，其係用於重建一遺失幀，該遺失幀分成複數個方塊，首先偵測一第零參考幀及一第一參考幀之間是否有一遺失幀，當有一遺失幀時，獲取該遺失幀的一方塊及該方塊的位置，再分別計算方塊在第零參考幀及第一參考幀的一第零時間差距及一第一時間差距，然後判斷第零時間差距是否小於第一時間差距，若是，設定該第一參考幀及一第一對應參考幀為現行參考幀組，並以該現行參考幀組計算補償方塊位置，最後擴展該方塊，以產生一擴充區域，依據該擴充區域以對該方塊進行重建。

The invention provides a method of frame error concealment in scable video decoding for reconstructing a lost frame. The lost frame has a plurality of blocks. First, it detects whether there is a lost frame between a zero reference frame and a first reference frame. When there is a lost frame between the zero reference frame and the first reference frame, it gets the block of the lost frame and the position of the block. Then, it calculates the time differences between the block and the zero reference frame and between the block and the first reference frame for generating a zero time difference and a first time difference, respectively. If the zero time difference is smaller than the first time difference, it sets the first reference frame and a first corresponding reference frame as a current reference frame set for calculating the position of a compensation block. Finally, it expands the block for generating a expansion area, and reconstructs the block based on the expansion area.

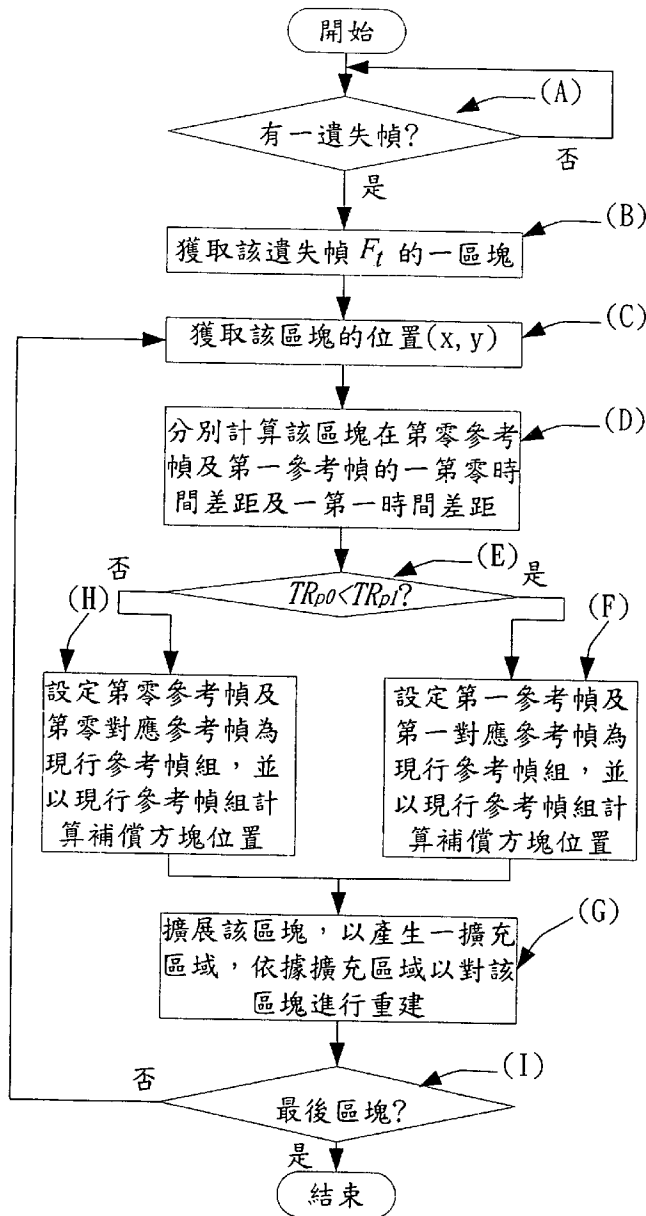


圖 5

# 發明專利說明書

(本說明書格式、順序，請勿任意更動，※記號部分請勿填寫)

※申請案號：99171637

※申請日：99.9.17

※IPC分類：

H04N7/64

(2006.01)

一、發明名稱：(中文/英文)

於可調式影像解碼的全畫面錯誤掩蓋方法

Method of frame error concealment in scable video decoding

二、中文發明摘要：

本發明提供一種於可調式影像解碼的全畫面錯誤掩蓋方法，其係用於重建一遺失幀，該遺失幀分成複數個方塊，首先偵測一第零參考幀及一第一參考幀之間是否有一遺失幀，當有一遺失幀時，獲取該遺失幀的一方塊及該方塊的位置，再分別計算方塊在第零參考幀及第一參考幀的一第零時間差距及一第一時間差距，然後判斷第零時間差距是否小於第一時間差距，若是，設定該第一參考幀及一第一對應參考幀為現行參考幀組，並以該現行參考幀組計算補償方塊位置，最後擴展該方塊，以產生一擴充區域，依據該擴充區域以對該方塊進行重建。

### 三、英文發明摘要：

The invention provides a method of frame error concealment in scalable video decoding for reconstructing a lost frame. The lost frame has a plurality of blocks. First, it detects whether there is a lost frame between a zero reference frame and a first reference frame. When there is a lost frame between the zero reference frame and the first reference frame, it gets the block of the lost frame and the position of the block. Then, it calculates the time differences between the block and the zero reference frame and between the block and the first reference frame for generating a zero time difference and a first time difference, respectively. If the zero time difference is smaller than the first time difference, it sets the first reference frame and a first corresponding reference frame as a current reference frame set for calculating the position of a compensation block. Finally, it expands the block for generating an expansion area, and reconstructs the block based on the expansion area.

四、指定代表圖：

(一)本案指定代表圖為：圖 5。

(二)本代表圖之元件符號簡單說明：

步驟(A)~步驟(I)

五、本案若有化學式時，請揭示最能顯示發明特徵的化學式：

無

## 六、發明說明：

### 【發明所屬之技術領域】

本發明係關於影像處理之技術領域，尤指一種於可調式影像解碼的全畫面錯誤掩蓋方法。

### 【先前技術】

H.264/AVC視訊編碼中引進了網路抽象層 (network abstract layer, NAL)的概念。網路抽象層的應用提高了影像編碼位元串的網路可親性，不論是採用Byte-Stream Format或Packet-Transport的系統都能很簡單的運用。而可調式視訊編碼 (scalable video coding, SVC)為H.264/AVC的延伸標準，自然也承襲網路抽象層這種特性。

在現今網路上的傳輸，其可靠度往往受到許多因素影響，例如通道頻寬的不穩定、通道的雜訊等都可能造成傳送的封包接收不完整甚至遺失。而封包的遺失便造成編碼影像的接收不完整，因而在進行影像解碼時就會使得畫面中的部分影像或全部影像有所缺失。在某些低位元率的影像傳輸系統，如3G通訊網路，由於其影像編碼後的資料量通常不高，因此封包的丟失容易會造成整個影像畫面的丟失(whole frame lost)。

錯誤掩蓋方法(frame error concealment)是利用已經接收到的正確的資訊來修補或掩蓋掉因為傳輸的錯誤所產生的影像遺漏或錯誤的部分。錯誤掩蓋方法係屬於解碼

端的後處理(post processing)。而其中一種重要的錯誤掩蓋方法便是時域上的錯誤掩蓋方法。

時域置換方法(Temporal Replacement, TR)係為一種習知的時域錯誤掩蓋方法，其係將丟失的方塊的運動向量(motion vector)直接設為零向量，也就是該方塊直接以參考幀上相同的空間位置的對應方塊(co-located block)的內容來做置換，此方法通常適用於掩蓋的區域沒有動作的狀況。

另一種習知的時域錯誤掩蓋方法係利用邊界匹配演算法(boundary matching algorithm, BMA)以改善時域置換方法的缺點，其係將丟失的方塊的運動向量由周圍方塊的運動向量中選取，以選取一個能夠使得補償後的方塊與周圍方塊的邊界失真最小的當作重建的運動向量。

前述以邊界匹配演算法(boundary matching algorithm)為基礎的錯誤掩蓋方法，都需要周圍的方塊的資訊做為判斷丟失方塊重建運動向量的依據，然而當運用在一個低位元率的影像傳輸環境時，比如說行動網路，則往往傳輸的錯誤會造成整個影像畫面的丟失，在此情形周圍方塊的資訊也同樣遺失，而不可得知。

針對前述問題，在SVC的標準草案討論會議中則針對全畫面影像丟失提出兩種基本的方法，分別為幀複製方法(frame copy, FC)及時域直接方法(temporal direct, TD)。

幀複製方法(FC)的方法是當有一個畫面的丟失發生時，直接把參考幀的內容當作現在這個畫面的內容直

接複製過來，因此在實現上十分容易。但是幀複製方法(FC)只適用在丟失的畫面與其參考幀的時間差距小，且畫面變動不大的影像中，若兩者的差異大則運用幀複製方法(FC)來進行錯誤掩蓋將會造成很大的誤差而使得重建的畫質降低許多。

在全畫面影像丟失的情況下，當前畫面的所有資訊，不論是運動向量、畫面殘值、方塊編碼模式等資訊都視為不可得知。然而若可以利用影像在時域上的相關性，計算估計出丟失的運動向量，再利用運動補償的方法就可從參考畫面重建出丟失的畫面，時域直接方法(TD)即是利用這個概念來對全畫面的影像丟失進行錯誤掩蓋。

如圖 1 所示，在時間  $t$  有一個 B 幀丟失，記做  $F_t$ ，其第一表單(list 1)的參考幀記作  $F_{t1}$ ， $F_{t1}$  的第零表單(list 0)參考幀記作  $F_{t1}^0$ 。 $F_t$  與  $F_{t1}$  兩幀的時間差距為 TRd， $F_t$  與  $F_{t1}^0$  兩幀的時間差距為 TRb，而 TRp0 為  $F_{t1}$  與  $F_{t1}^0$  兩幀間的時間差，亦即  $\text{TRp0} = \text{TRb} + \text{TRd}$ 。假設針對  $F_t$  上的任一個方塊  $B_L$ ，在  $F_{t1}$  上可以找到與  $B_L$  有相同的空間位置的對應方塊  $B_{C1}$ ，其中可以找到一個由  $F_{t1}$  指向  $F_{t1}^0$  的運動向量，此即  $F_{t1}$  的第零表單(list 0)運動向量  $\bar{v}_{C,0}$ 。由於  $\bar{v}_{C,0}$  在時序上跨過  $F_t$ ，由於在短時間內影像的移動具有連續性，因此可以利用  $\bar{v}_{C,0}$  在時序上進行內插來得出  $F_t$  指向  $F_{t1}^0$  的運動向量  $\bar{v}_0$  以及  $F_t$  指向  $F_{t1}$  的運動向量  $\bar{v}_1$ 。其中， $F_t$  指向  $F_{t1}^0$

的運動向量  $\bar{v}_0$  及  $F_t$  指向  $F_{t1}$  的運動向量  $\bar{v}_1$  可用公式(1)表示：

$$\bar{v}_0 = \frac{TR_b}{TR_{p0}} \times \bar{v}_{C,0}, \quad \bar{v}_1 = \frac{TR_d}{TR_{p0}} \times \bar{v}_{C,0}。 \quad (1)$$

當  $F_{t1}^0$  為  $F_t$  的第零表單(list 0)，再藉由公式(1)計算出  $F_t$  中每個方塊的  $\bar{v}_0$  與  $\bar{v}_1$ ，便可利用雙向預測與運動補償以便將丟失的畫面重建回來。

前述時域直接方法(TD)提供了一個利用時序上的關係來重建出丟失的畫面的運動向量，藉此達成錯誤掩蓋的方法。然而時域直接方法(TD)在某些情況下仍然有可以改進的部分。

考慮一個圖片組(group of pictures, GOP)大小為 8 的階層式 B 畫面架構，其幀間預測的關係如圖 2 所示。圖 2 係圖片組為 8 的階層式 B 畫面結構之示意圖。當幀 6 (Frame 6)在傳送的時候遺失，根據時域直接方法(TD)的方法， $F_t = \text{Frame 6}$ ， $F_{t1} = \text{Frame 8}$ ， $F_{t1}^0 = \text{Frame 0}$ 。在此情況下，由於  $F_{t1}^0$  與  $F_{t1}$  的時間跨度( $TR_{p0} = 8$ )相較於  $F_{t1}$  與  $F_t$  之時間跨度( $TR_d = 2$ )差距太大，因此利用公式(1)所計算出來的運動向量將會有較大的誤差。

圖 3 係習擴充時域直接方法的運動向量推導之示意圖。如圖 3 所示， $F_t$  的第零表單(list 0)參考幀記作  $F_{t0}$ ，而  $F_{t0}$  的第一表單(list 1)參考幀記作  $F_{t0}^1$ ，對  $F_t$  上的任一方塊  $B_L$ ，我們同樣可在  $F_{t0}$  上找到一個相對應的方塊  $B_{C0}$ ，

其中可以找到一個  $\bar{v}_{C,l}$  為從  $F_{t0}$  指向  $F_{t0}^l$  之運動向量； $F_t$  與  $F_{t0}$  兩幀的時間差距為  $TR_b$ ， $F_t$  與  $F_{t0}^l$  兩幀的時間差距為  $TR_d$ ，而  $TR_{p1}$  為  $F_{t0}$  與  $F_{t0}^l$  兩幀間的時間差，則可利用  $\bar{v}_{C,l}$  在時序上進行內插來得出  $F_t$  指向  $F_{t0}$  的運動向量  $\bar{v}_0$  以及  $F_t$  指向  $F_{t0}^l$  的運動向量  $\bar{v}_l$ 。其中， $F_t$  指向  $F_{t0}$  的運動向量  $\bar{v}_0$  及  $F_t$  指向  $F_{t0}^l$  的運動向量  $\bar{v}_l$  可用公式(2)表示：

$$\bar{v}_0 = \frac{TR_b}{TR_{p1}} \times \bar{v}_{C,l}, \quad \bar{v}_l = \frac{TR_d}{TR_{p1}} \times \bar{v}_{C,l}。 \quad (2)$$

考慮前例， $F_t = \text{Frame } 6$ ， $F_{t0} = \text{Frame } 4$ ， $F_{t0}^l = \text{Frame } 8$ 。此時  $F_{t0}$  與  $F_{t0}^l$  便擁有一個較小的時間跨度 ( $TR_{p1} = 4$ )，因此能夠計算出一個誤差較小的運動向量。而擴充時域直接方法 (extend temporal direct method, ETDM) 便是在做運動向量的估計之前，先行計算  $TR_{p0}$  與  $TR_{p1}$ ，倘若  $TR_{p0}$  較小則利用公式(1)來做運動向量之估計，反之若  $TR_{p1}$  較小則利用公式(2)來做運動向量之估計，因而對TD做出了改進。

前述之習知之時域直接方法 (TD) 與擴充時域直接方法 (ETDM) 利用了影像時域上的連續性，在時域上利用前後參考幀的運動向量以計算出丟失的畫面的運動向量，藉此來進行錯誤掩蓋。然而，不論是時域直接方法 (TD) 與擴充時域直接方法 (ETDM)，對每一個丟失的畫面中的

方塊，其運動向量都是由前後參考幀的相對應位置的方塊的運動向量得之。

圖 4 係習知動作補償概念之示意圖。考慮如圖 4 的情況，在丟失的畫面中有一個方塊  $B_L$ ， $x$  與  $y$  分別是此方塊  $B_L$  最左上角的像素在空間中的水平與垂直位置，以  $B_L(x,y)$  來表示，而在第一表單(list 1)參考幀中與  $B_L$  有相同空間位置的方塊稱為  $B_L$  的相對應方塊，以  $B_C^1(x,y)$  表示，其中上標的 1 代表的是第一表單(list 1)參考幀。

在時域直接方法(TD)與擴充時域直接方法(ETDM)中，由  $B_L(x,y)$  指向第零表單(list 0)參考幀的運動向量  $\bar{v}_0$  與指向第一表單(list 1)參考幀的運動向量  $\bar{v}_1$  可由  $B_C^1(x,y)$  的第零表單(list 0)參考幀之運動向量  $\bar{v}_{C,0}$  算出，由公式(1)可以得出：

$$\bar{v}_0 = \frac{TR_b}{TR_{p0}} \times \bar{v}_{C,0}, \quad \bar{v}_1 = \frac{TR_d}{TR_{p0}} \times \bar{v}_{C,0}。$$

以  $v_{0x}$ 、 $v_{0y}$  代表  $\bar{v}_0$  的水平與垂直分量， $v_{1x}$ 、 $v_{1y}$  代表  $\bar{v}_1$  的水平與垂直分量。因此  $B_L(x,y)$  由向量  $\bar{v}_0$  指向第零表單(list 0)的參考方塊其最左上角像素的水平與垂直位置分別為  $x+v_{0x}$ 、 $y+v_{0y}$ ，則此方塊就以  $B_C^0(x+v_{0x}, y+v_{0y})$  表示。而由向量  $\bar{v}_1$  指向第一表單(list 1)的參考方塊其最左上角像素的水平與垂直位置分別為  $x+v_{1x}$ 、 $y+v_{1y}$ ，則此方塊以  $B_C^1(x+v_{1x}, y+v_{1y})$  表示。

$B_L(x, y)$  便藉由  $B_C^0(x+v_{0x}, y+v_{0y})$  及  $B_C^1(x+v_{1x}, y+v_{1y})$  進行雙向運動補償 (bi-directional motion compensation)，也就是方塊  $B_L(x, y)$  的每一個像素都是由  $B_C^0(x+v_{0x}, y+v_{0y})$  及  $B_C^1(x+v_{1x}, y+v_{1y})$  內的對應像素做平均而得。而這個補償的動作其實就意味著將方塊  $B_L(x, y)$  與方塊  $B_C^0(x+v_{0x}, y+v_{0y})$  及方塊  $B_C^1(x+v_{1x}, y+v_{1y})$  視為在不同時序上擁有相同內容的方塊，而此方塊移動的軌跡，以方塊之左上角像素為代表，即從  $(x+v_{0x}, y+v_{0y})$  經過到  $(x+v_{1x}, y+v_{1y})$ 。

前述說明是將時域直接方法 (TD) 與擴充時域直接方法 (ETDM) 所得出的參考運動向量所代表的移動軌跡進行分析，也就是經過錯誤掩蓋後的移動軌跡。因為上述運動補償用的運動向量  $\bar{v}_0$  與  $\bar{v}_1$  皆是由  $\bar{v}_{C,0}$  得出，因此再來將針對  $\bar{v}_{C,0}$  所代表的移動軌跡進行分析。

以  $v_{C,0x}$ 、 $v_{C,0y}$  代表  $\bar{v}_{C,0}$  的水平與垂直分量，由於  $\bar{v}_{C,0}$  是由第一表單 (list 1) 參考幀上的方塊  $B_C^1(x, y)$  指向第零表單 (list 0) 的運動向量，因此  $B_C^1(x, y)$  在第零表單 (list 0) 參考幀上的參考方塊其左上角像素的位置為  $(x+v_{C,0x}, y+v_{C,0y})$ ，而此方塊就以  $B_C^0(x+v_{C,0x}, y+v_{C,0y})$  表示。而由  $B_C^1(x, y)$  指向丟失的畫面的運動向量，可以藉由對  $\bar{v}_{C,0}$  做內插，得出該向量為：

$$-\bar{v}_l = \frac{TR_d}{TR_{p0}} \times \bar{v}_{C,0},$$

其水平與垂直分量分別為 $-v_{lx}$ 、 $-v_{ly}$ ，因此 $B_C^l(x,y)$ 在丟失畫面的參考方塊其左上角像素位置為 $(x-v_{lx}, y-v_{ly})$ ，該方塊就以 $B_L(x-v_{lx}, y-v_{ly})$ 表示。

同樣的，也可將方塊 $B_C^0(x+v_{C,0x}, y+v_{C,0y})$ 與方塊 $B_C^l(x,y)$ 還有方塊 $B_L(x-v_{lx}, y-v_{ly})$ 視為不同時序上擁有相同內容的方塊，而此方塊移動的軌跡，以方塊左上角的像素為代表，即從 $(x+v_{C,0x}, y+v_{C,0y})$ 經過 $(x-v_{lx}, y-v_{ly})$ 到 $(x,y)$ 。而此軌跡乃是由 $\bar{v}_{C,0}$ 這個已知的運動向量建構出來的，因此可以視為是實際的移動軌跡。

由前述說明可知，經過時域直接方法(TD)或擴充時域直接方法(ETDM)錯誤掩蓋的移動軌跡與實際的移動軌跡實際上是不完全相同的。當這兩個軌跡所包含的方塊在畫面中是屬於同一個物件，或者兩者的移動是一致的，則此時畫面上不同位置的運動向量差異不大，利用經過時域直接方法(TD)或擴充時域直接方法(ETDM)錯誤掩蓋的移動軌跡來做運動補償將不會對畫面的重建造成影響。但當兩個軌跡包含的方塊是屬於不同的物件，且物件的移動並不一致，有相對的位置變化，則時域直接方法(TD)或擴充時域直接方法(ETDM)計算出來的運動向量與正確的運動向量將會存在差異，因此若利用經過時域直接方法(TD)或擴充時域直接方法(ETDM)錯誤

掩蓋的移動軌跡來做運動補償將可能重建出一個錯誤的畫面，因而使得重建的效果變差。

表 1 係針對時域直接方法 (TD) 或擴充時域直接方法 (ETDM) 所計算出來的運動向量與未經過損失的正確的運動向量進行比較並統計兩者之間的差異。假設擴充時域直接方法 (ETDM) 計算出來的運動向量為  $MV_{ETDM}$ ，正確的向量為  $MV_{CRCT}$ ，將  $MV_{ETDM}$  與  $MV_{CRCT}$  的距離平方，與一個門檻值 (threshold, 以  $R_{TH}$  表示) 進行比較，統計兩者距離平方大於此門檻的比例。如表 1 所示，當門檻值  $R_{TH}$  設為 256 時，亦即  $MV_{ETDM}$  與  $MV_{CRCT}$  的距離大於 16，依照各影片的特性，最高有 47.23% 的比例兩者距離大於 16，而當門檻值  $R_{TH}$  設為 64，亦即  $MV_{ETDM}$  與  $MV_{CRCT}$  的距離大於 8，此時最高有 58.40% 的比例兩者距離大於 8，且最低也有 17.52% 的比例。由此數據可以得出時域直接方法 (TD) 或擴充時域直接方法 (ETDM) 所計算出來的向量與正確的向量的確有相當的差距。故習知全畫面錯誤掩蓋方法仍有改善空間。

$R_{TH}$ \ sequence	BUS	CITY	CREW	FOOTBALL
256	18.97%	10.09%	30.58%	47.23%
64	23.92%	24.21%	45.06%	58.40%
$R_{TH}$ \ sequence	FOREMAN	ICE	MOBILE	SOCCER
256	16.09%	12.27%	4.36%	31.07%
64	30.44%	19.17%	17.52%	42.57%

表 一

**【發明內容】**

本發明之主要目的係在提供一種於可調式影像解碼的全畫面錯誤掩蓋方法，其係一全新架構，同時，其可較習知技術獲得正確的重建結果。

依據本發明之一特色，本發明提出一種於可調式影像解碼的全畫面錯誤掩蓋方法，其係運用於一影像幀序列解碼中以重建在時序上位於一第零參考幀及一第一參考幀之間的一遺失幀，該遺失幀分成複數個方塊，該第零參考幀屬於一第零表單，該第一參考幀屬於一第一表單，該第零參考幀對應於該第一表單內具有一第零對應參考幀，該第一參考幀對應於該第零表單內具有一第一對應參考幀，該方法包含(A)係偵測該第零參考幀及該第一參考幀之間是否有一遺失幀；(B)當步驟(A)中偵測該第零參考幀及該第一參考幀之間有一遺失幀時，獲取該遺失幀的一方塊；(C)獲取該方塊的位置；(D)分別計算該方塊在該第零參考幀及該第一參考幀的一第零時間差距及一第一時間差距；(E)判斷該第零時間差距是否小於該第一時間差距；(F)當判定該第零時間差距小於該第一時間差距時，設定該第一參考幀及該第一對應參考幀為現行參考幀組，並以該現行參考幀組計算補償方塊位置；以及(G)擴展該方塊，以產生一擴充區域，依據該擴充區域以對該方塊進行重建。

### 【實施方式】

圖5係本發明一種於可調式影像解碼的全畫面錯誤掩蓋方法的流程圖，其係運用於一影像解碼裝置的影像幀序列解碼中以重建在時序上位於一第零參考幀  $F_{t0}$  及一第一參考幀  $F_{t1}$  之間的一遺失幀  $F_t$ ，該遺失幀分成複數個方塊。

圖6係本發明圖片組(group of pictures, GOP)的關係之示意圖。圖7係本發明圖片組(GOP)的另一關係之示意圖。如圖6所示，例如當幀6(frame number 6)為遺失幀  $F_t$  時，第零表單(list 0)則為幀號碼在幀6之前的幀之集合，第一表單(list 1)則為幀號碼在幀6之後的幀之集合。於圖6中，第零表單(list 0)為幀0至幀5之集合，第一表單(list 1)為幀7至幀8之集合。

該第零參考幀  $F_{t0}$  屬於第零表單(list 0)，該第一參考幀  $F_{t1}$  屬於第一表單(list 1)，該第零參考幀  $F_{t0}$  對應於該第一表單(list 1)內具有一第零對應參考幀  $F_{t0}^1$  (幀號碼為8之幀)，該第一參考幀  $F_{t1}$  對應於該第零表單(list 0)內具有一第一對應參考幀  $F_{t1}^0$  (幀號碼為0之幀)。

而圖7的情形，為熟於該技術者可由圖6推知，不予贅述。

首先於步驟(A)中，偵測該第零參考幀  $F_{t0}$  及該第一參考幀  $F_{t1}$  之間是否有一遺失幀  $F_t$ 。其可判斷該第零參考幀  $F_{t0}$  及該第一參考幀  $F_{t1}$  之間的幀號碼是否連續，當該第

零參考幀  $F_{t0}$  及該第一參考幀  $F_{t1}$  之間的幀號碼非連續時，表示有一遺失幀  $F_t$ ，此時執行步驟(B)，反之，重回步驟(A)。

當步驟(A)中偵測該第零參考幀  $F_{t0}$  及該第一參考幀  $F_{t1}$  之間有一遺失幀  $F_t$  時，於步驟(B)中獲取該遺失幀  $F_t$  的一方塊。

於步驟(C)中獲取該方塊的位置(x,y)。

於步驟(D)中分別計算該方塊在該第零參考幀  $F_{t0}$  及該第一參考幀  $F_{t1}$  的一第零時間差距  $TR_{p0}$  及一第一時間差距  $TR_{p1}$ 。其中，該第零時間差距  $TR_{p0}$  為  $F_{t1}$  與  $F_{t1}^0$  的時

間差距，該第一時間差距  $TR_{p1}$  為  $F_{t0}^1$  與  $F_{t0}$  的時間差距。

於一實施例中，第零時間差距  $TR_{p0}$  可計算  $F_{t1}$  與  $F_{t1}^0$  的幀號碼差距，該第一時間差距  $TR_{p1}$  可計算  $F_{t0}^1$  與  $F_{t0}$  的幀號碼差距。

於步驟(E)中，判斷該第零時間差距  $TR_{p0}$  是否小於該第一時間差距  $TR_{p1}$ 。

當判定該第零時間差距  $TR_{p0}$  小於該第一時間差距  $TR_{p1}$  時，於步驟(F)中設定該第一參考幀  $F_{t1}$  及該第一對應參考幀  $F_{t1}^0$  為現行參考幀組，並以該現行參考幀組計算補償方塊位置。

圖8係本發明當第零時間差距  $TR_{p0}$  小於該第一時間差距  $TR_{p1}$  時之示意圖。於步驟(F)中設定該第一對應參考幀  $F_{t1}^0$  為該第零表單(list 0)的參考幀，設定該第一參考幀  $F_{t1}$  為該第一表單(list 1)的參考幀，並分別計算該方塊對應於該第零表單(list 0)及該第一表單(list 1)相對方塊的運動向量( $\bar{v}_0$ 與 $\bar{v}_1$ )，並依據該運動向量( $\bar{v}_0$ 與 $\bar{v}_1$ )設定該方塊位置、該方塊對應於該第零表單(list 0)之方塊位置、及該方塊對應於該第一表單(list 1)之方塊位置。

該方塊對應於該第零表單(list 0)相對方塊的運動向量  $\bar{v}_0$  及該方塊對應於該第一表單(list 1)相對方塊的運動向量  $\bar{v}_1$  分別為：

$$\bar{v}_0 = \frac{TR_b}{TR_{p0}} \times \bar{v}_{C,0}, \quad \bar{v}_1 = \frac{TR_d}{TR_{p0}} \times \bar{v}_{C,0}。$$

當中， $\bar{v}_{C,0}$  為該第一參考幀  $F_{t1}$  對應於該第零表單(list 0)的運動向量， $TR_b$  為該遺失幀  $F_t$  與該第一對應參考幀  $F_{t1}^0$  的時間差距， $TR_d$  為該遺失幀  $F_t$  與該第一參考幀  $F_{t1}$  的時間差距。並設定該方塊位置為  $(x - v_{1x}, y - v_{1y})$ ，該方塊對應於該第零表單(list 0)之方塊位置為  $(x + v_{C,0x}, y + v_{C,0y})$ ，該方塊對應於該第一表單(list 1)之方塊位置為  $(x, y)$ 。 $v_{1x}$ 、 $v_{1y}$  代表  $\bar{v}_1$  的水平與垂直分量， $v_{C,0x}$ 、 $v_{C,0y}$  代表  $\bar{v}_{C,0}$  的水平與垂直分量。

於步驟(G)中，擴展該方塊以產生一擴充區域，依據該擴充區域以對該方塊進行重建。

當判定該第零時間差距  $TR_{p0}$  非小於該第一時間差距  $TR_{p1}$  時，於步驟(H)中，設定該第零參考幀  $F_{t0}$  及該第零對應參考幀  $F_{t0}^I$  為一現行參考幀組，以該現行參考幀組計算補償方塊位置，並執行步驟(G)。

圖9係本發明當第零時間差距  $TR_{p0}$  非小於該第一時間差距  $TR_{p1}$  時之示意圖。於步驟(H)中設定該第零參考幀  $F_{t0}$  為該第零表單(list 0)的參考幀，設定該第零對應參考幀  $F_{t0}^I$  為該第一表單(list 1)的參考幀，並分別計算該方塊對應於該第零表單(list 0)及該第一表單(list 1)相對方塊的運動向量 ( $\bar{v}_0$  與  $\bar{v}_1$ )，並依據運動向量 ( $\bar{v}_0$  與  $\bar{v}_1$ ) 設定該方塊位置、該方塊對應於該第零表單(list 0)之方塊位置、該方塊對應於該第一表單(list 1)之方塊位置。

該方塊對應於該第零表單(list 0)相對方塊的運動向量  $\bar{v}_0$  及該方塊對應於該第一表單(list 1)相對方塊的運動向量  $\bar{v}_1$  分別為：

$$\bar{v}_0 = \frac{TR_b}{TR_{p1}} \times \bar{v}_{C,1}, \quad \bar{v}_1 = \frac{TR_d}{TR_{p1}} \times \bar{v}_{C,1},$$

當中， $\bar{v}_{C,1}$  為該第零參考幀  $F_{t0}$  對應於該第一表單(list 1)的運動向量， $TR_b$  為該遺失幀  $F_t$  與該第零參考幀  $F_{t0}$  的時間差距， $TR_d$  為該遺失幀  $F_t$  與該第

零對應參考幀  $F_{t_0}^l$  的時間差距。並設定該方塊位置為  $(x - v_{0x}, y - v_{0y})$ ，該方塊對應於該第零表單(list 0)之方塊位置為  $(x, y)$ ，該方塊對應於該第一表單(list 1)之方塊位置為  $(x + v_{C,1x}, y + v_{C,1y})$ 。  $v_{0x}$ 、 $v_{0y}$  代表  $\bar{v}_0$  的水平與垂直分量， $v_{C,1x}$ 、 $v_{C,1y}$  代表  $\bar{v}_{C,1}$  的水平與垂直分量。

於步驟(I)中，判斷該方塊是否為該遺失幀  $F_t$  的最後方塊，若是，結束該全畫面錯誤掩蓋方法，若否，獲取下一方塊，並重回步驟(C)。

圖10係本發明動作補償後的重疊與未包含區域之示意圖。由於本發明是利用丟失畫面的參考幀上的相對應方塊之運動向量的移動軌跡來決定重建的方塊的位置，因此在畫面重建的時候，就會如圖10所示會有部分方塊重疊(overlap area)以及部分像素未包含(uncover area)的狀況。為了解決這個問題，在步驟(G)中對運動補償的部分進行了修改。

為了處理重疊區域的運動補償情形，必須了解正在補償的像素位置是否已於前面方塊進行運動補償時被補償過，且現在的像素位置是否位於一擴充區域(expansion area)亦會影響補償的結果。因此利用一個矩陣  $PxCheck[pic\_width][pic\_height]$  來記錄每個像素目前的狀態。

其中， $PxCheck[x][y] = 0$ ，表示在  $(x, y)$  位置的像素尚未被重建； $PxCheck[x][y] = -1$ ，表在  $(x, y)$  位置的像素

已被重建，但屬於一擴展區域(extend area);  $PxCheck[x][y] = 1$ ，表在在(x, y)位置的像素已被重建，且非擴展區域(extend area)。於開始時，將所有像素的 $PxCheck[x][y]$ 均初始化為0。

圖11係本發明步驟(G)之詳細流程圖。該方塊由 $M \times N$ 個像素所組成。在H.264的標準中，該方塊的大小可以有 $4 \times 4$ 、 $8 \times 4$ 、 $4 \times 8$ 、 $8 \times 8$ 、 $16 \times 8$ 、 $8 \times 16$ 、 $16 \times 16$ 像素。為了避免重建時有像素未被包含，於步驟(G)中將該方塊由 $M \times N$ 個像素擴充為 $(M + \text{abs}(v_{0x}) + \text{abs}(v_{1x})) \times (N + \text{abs}(v_{0y}) + \text{abs}(v_{1y}))$ 個像素的該擴充區域(expansion area)，當中 $v_{0x}$ 為運動向量 $\vec{v}_0$ 在x軸的大小， $v_{0y}$ 為運動向量 $\vec{v}_0$ 在y軸的大小，當中 $v_{1x}$ 為運動向量 $\vec{v}_1$ 在x軸的大小， $v_{1y}$ 為運動向量 $\vec{v}_1$ 在y軸的大小。

於步驟(G1)中，獲取該擴充區域(expansion)的一像素，其中該擴充區域係由該方塊及一擴展區域(extend area)。圖12係本發明擴充區域之示意圖，由圖12可知，擴展區域(extend area)係指非該方塊的斜線部分。

於步驟(G2)中，獲取該像素的位置(x,y)。

於步驟(G3)中，判斷該像素的位置(x,y)是否位於該擴展區域。

其中，於步驟(G3)中擴展區域的判斷則是由運動向量的方向來決定，圖13、圖14所示為擴展區域的分布。圖13係本發明經錯誤掩蓋後的方塊與參考方塊的關係之示意圖，其係以第零表單(list 0)參考幀為基

礎 ( $TR_{p0} > TR_{p1}$ )。圖 14 係本發明經錯誤掩蓋後的方塊與參考方塊的關係之示意圖，其係以第一表單 (list 1) 參考幀為基礎 ( $TR_{p0} \leq TR_{p1}$ )。

依照重建方塊與相對應方塊的相關位置，可分為 8 種情形，其可歸納如表 2 所示，其中情形 0 (case 0) 到情形 3 (case 3) 為以第零表單 (list 0) 參考幀為基礎 ( $TR_{p0} > TR_{p1}$ )。情形 4 (case 4) 到情形 7 (case 7) 以第一表單 (list 1) 參考幀為基礎 ( $TR_{p0} \leq TR_{p1}$ )。

參考幀	情形	落在擴展區域之條件
第零表單 (list 0) $TR_{p0} > TR_{p1}$	0	$i < \text{abs}(v_{ix}) \parallel i \geq (M + \text{abs}(v_{ix})) \parallel j < \text{abs}(v_{iy}) \parallel j \geq (N + \text{abs}(v_{iy}))$
	1	$i < \text{abs}(v_{ox}) \parallel i \geq (M + \text{abs}(v_{ox})) \parallel j < \text{abs}(v_{iy}) \parallel j \geq (N + \text{abs}(v_{iy}))$
	2	$i < \text{abs}(v_{ix}) \parallel i \geq (M + \text{abs}(v_{ix})) \parallel j < \text{abs}(v_{oy}) \parallel j \geq (N + \text{abs}(v_{oy}))$
	3	$i < \text{abs}(v_{ox}) \parallel i \geq (M + \text{abs}(v_{ox})) \parallel j < \text{abs}(v_{oy}) \parallel j \geq (N + \text{abs}(v_{oy}))$
第一表單 (list 1) $TR_{p0} \leq TR_{p1}$	4	$i < \text{abs}(v_{ox}) \parallel i \geq (M + \text{abs}(v_{ox})) \parallel j < \text{abs}(v_{oy}) \parallel j \geq (N + \text{abs}(v_{oy}))$
	5	$i < \text{abs}(v_{ix}) \parallel i \geq (M + \text{abs}(v_{ix})) \parallel j < \text{abs}(v_{oy}) \parallel j \geq (N + \text{abs}(v_{oy}))$
	6	$i < \text{abs}(v_{ox}) \parallel i \geq (M + \text{abs}(v_{ox})) \parallel j < \text{abs}(v_{iy}) \parallel j \geq (N + \text{abs}(v_{iy}))$
	7	$i < \text{abs}(v_{ix}) \parallel i \geq (M + \text{abs}(v_{ix})) \parallel j < \text{abs}(v_{iy}) \parallel j \geq (N + \text{abs}(v_{iy}))$
( $i, j$ ) 為相對於當前補償範圍最左上角之座標		

表 2

依據表 2，步驟 (G3) 即可判斷該像素的位置 ( $x, y$ ) 是否位於該擴展區域。

若步驟 (G3) 判定該像素的位置 ( $x, y$ ) 位於該擴展區域 (非 4X4 的部分)，於步驟 (G4) 中，再判斷該像素是否已有重建值，且位於先前重建方塊的非擴展區域。其係判斷  $PxCheck[x][y]$  是否為 1。

若判定該像素已有重建值且位於該先前重建方塊的非擴展區域，於步驟(G5)中，再判斷該像素是否為該擴充區域(expansion area)最後一像素，若是，結束步驟(G)，若否，獲取下一像素，並執行步驟(G2)。

若步驟(G4)判定該像素沒有重建值，且非位於先前重建方塊的非擴展區域，於步驟(G6)中，再判斷該像素是否尚未被重建。其係判斷 $PxCheck[x][y]$ 是否為0。

若步驟(G6)判定該像素尚未被重建，於步驟(G7)中，則對該像素進行重建。步驟(G7)包含步驟(G71)及步驟(G72)。於步驟(G71)中，設定 $PxCheck[x][y]$ 為-1，於步驟(G72)中，將該像素之值設定為 $(pel_{c_0} + pel_{c_1} + 1) \gg 1$ ，並執行步驟(G5)。當中， $pel_{c_0}$ 為區塊 $B_{c_0}$ 相對應的像素值， $pel_{c_1}$ 為區塊 $B_{c_1}$ 相對應的像素值， $\gg$ 為位元右移運算(right shift)。

若步驟(G6)判定該像素已被重建，表示該像素已有重建值且位於該先前重建方塊的擴展區域，於步驟(G8)中，則對該像素進行重建並與該先前重建方塊的擴展區域之重建值平均，再執行步驟(G5)。

於步驟(G8)中將該像素之值設定為 $(2 \times pel + pel_{c_0} + pel_{c_1} + 3) \gg 2$ ，當中， $pel$ 為該像素先前的重建值， $pel_{c_0}$ 為區塊 $B_{c_0}$ 相對應的像素值， $pel_{c_1}$ 為區塊 $B_{c_1}$ 相對應的像素值， $\gg$ 為位元右移運算(right shift)。

若步驟(G3)判定該像素的位置(x,y)非位於該擴展區域，於步驟(G9)中，再判斷該像素是否已有重建值，且

位於該先前重建方塊的非擴展區域。其係判斷  $PxCheck[x][y]$  是否為 1，若是，執行步驟(G8)，若否，設定  $PxCheck[x][y]$  為 1，並執行步驟(G72)。

由於時域直接方法(TD)與擴充時域直接方法(ETDM)是以經過錯誤掩蓋的移動軌跡來進行運動補償，而這並不能在所有的情況中都獲得正確的重建結果，當畫面中的物件不是一致的移動時，時域直接方法(TD)與擴充時域直接方法(ETDM)利用經過錯誤掩蓋的移動軌跡將會對重建後的結果產生誤差。本發明的錯誤掩蓋方法在對丟失的畫面進行運動補償時，選擇了確實的移動軌跡而非經過錯誤掩蓋的移動軌跡來判斷補償的對象方塊，可較習知的時域直接方法(TD)與擴充時域直接方法(ETDM)獲得正確的重建結果。本發明的錯誤掩蓋方法較時域直接方法(TD)在信噪比(PSNR)上有平均 0.32dB 的改善，而與擴充時域直接方法(ETDM)相比，本發明的錯誤掩蓋方法在信噪比(PSNR)上有平均 0.2dB 的改善。

由前述說明可知，本發明提供一種全新架構的於可調式影像解碼的全畫面錯誤掩蓋方法，其可較習知的時域直接方法(TD)與擴充時域直接方法(ETDM)獲得正確的重建結果。

由上述可知，本發明無論就目的、手段及功效，在在均顯示其迥異於習知技術之特徵，極具實用價值。惟應注意的是，上述諸多實施例僅係為了便於說明而舉例

而已，本發明所主張之權利範圍自應以申請專利範圍所述為準，而非僅限於上述實施例。

### 【圖式簡單說明】

圖1係習知時域直接方法的運動向量推導之示意圖。

圖2係圖片組為8的階層式B畫面結構之示意圖。

圖3係習知擴充時域直接方法的運動向量推導之示意圖。

圖4係習知動作補償概念之示意圖。

圖5係本發明一種於可調式影像解碼的全畫面錯誤掩蓋方法的流程圖。

圖6係本發明圖片組的關係之示意圖。

圖7係本發明圖片組的另一關係之示意圖。

圖8係本發明當第零時間差距小於該第一時間差距時之示意圖。

圖9係本發明當第零時間差距非小於該第一時間差距時之示意圖。

圖10係本發明動作補償後的重疊與未包含區域之示意圖。

圖11係本發明步驟(G)之詳細流程圖。

圖12係本發明擴充區域之示意圖。

圖13、圖14係本發明為擴展區域的分布之示意圖。

### 【主要元件符號說明】

步驟(A)~ 步驟(I)

步驟(G1)~ 步驟(G9)

## 七、申請專利範圍：

1. 一種於可調式影像解碼的全畫面錯誤掩蓋方法，其係運用於一影像幀序列解碼中以重建在時序上位於一第零參考幀及一第一參考幀之間的一遺失幀，該遺失幀分成複數個方塊，該第零參考幀屬於一第零表單，該第一參考幀屬於一第一表單，該第零參考幀對應於該第一表單內具有一第零對應參考幀，該第一參考幀對應於該第零表單內具有一第一對應參考幀，該方法包含：

(A) 係偵測該第零參考幀及該第一參考幀之間是否有一遺失幀；

(B) 當步驟(A)中偵測該第零參考幀及該第一參考幀之間有一遺失幀時，獲取該遺失幀的一方塊；

(C) 獲取該方塊的位置；

(D) 分別計算該方塊在該第零參考幀及該第一參考幀的一第零時間差距及一第一時間差距；

(E) 判斷該第零時間差距是否小於該第一時間差距；

(F) 當判定該第零時間差距小於該第一時間差距時，設定該第一參考幀及該第一對應參考幀為現行參考幀組，並以該現行參考幀組計算補償方塊位置；以及

(G) 擴展該方塊，以產生一擴充區域，依據該擴充區域以對該方塊進行重建。

2. 如申請專利範圍第1項所述之於可調式影像解碼的全畫面錯誤掩蓋方法，其更包含：

(H) 當判定該第零時間差距非小於該第一時間差距時，設定該第零參考幀及該第零對應參考幀為一現行參

考幀組，以該現行參考幀組計算補償方塊位置，並執行步驟(G)。

3. 如申請專利範圍第2項所述之於可調式影像解碼的全畫面錯誤掩蓋方法，其更包含：

(I)判斷該方塊是否為該遺失幀的最後方塊，若是，結束該全畫面錯誤掩蓋方法，若否，獲取下一方塊，並重回步驟(C)。

4. 如申請專利範圍第3項所述之於可調式影像解碼的全畫面錯誤掩蓋方法，其中，於步驟(F)中設定該第一對應參考幀為該第零表單的參考幀，設定該第一參考幀為該第一表單的參考幀，並分別計算該方塊對應於該第零表單及該第一表單相對方塊的運動向量，並依據該運動向量設定該方塊位置、該方塊對應於該第零表單之方塊位置、該方塊對應於該第一表單之方塊位置。

5. 如申請專利範圍第4項所述之於可調式影像解碼的全畫面錯誤掩蓋方法，其中，該方塊對應於該第零表單相對方塊的運動向量  $\vec{v}_0$  及該方塊對應於該第一表單相對方塊的運動向量  $\vec{v}_1$  分別為：

$$\vec{v}_0 = \frac{TR_b}{TR_{p0}} \times \overline{v_{c,0}}, \quad \vec{v}_1 = \frac{TR_d}{TR_{p0}} \times \overline{v_{c,0}},$$

當中， $\overline{v_{c,0}}$  為該第一參考幀  $F_{t1}$  對應於該第零表單的運動向量， $TR_b$  為該遺失幀  $F_t$  與該第一對應參考幀  $F_{t1}^0$  的時間差距， $TR_d$  為該遺失幀  $F_t$  與該第一參考幀  $F_{t1}$  的時間差距。

6. 如申請專利範圍第5項所述之於可調式影像解碼的全畫面錯誤掩蓋方法，其中，其中，於步驟(H)中設定該第零參考幀為該第零表單的參考幀，設定該第零對應參考幀為該第一表單的參考幀，並分別計算該方塊對應於該第零表單及該第一表單相對方塊的運動向量，並依據運動向量設定該方塊位置、該方塊對應於該第零表單之方塊位置、該方塊對應於該第一表單之方塊位置。

7. 如申請專利範圍第6項所述之於可調式影像解碼的全畫面錯誤掩蓋方法，其中，該方塊對應於該第零表單相對方塊的運動向量  $\vec{v}_0$  及該方塊對應於該第一表單相對方塊的運動向量  $\vec{v}_1$  分別為

$$\vec{v}_0 = \frac{TR_b}{TR_{p1}} \times \vec{v}_{C,1}, \quad \vec{v}_1 = \frac{TR_d}{TR_{p1}} \times \vec{v}_{C,1},$$

當中， $\vec{v}_{C,1}$  為該第零參考幀  $F_{t0}$  對應於該第一表單 (list 1) 的運動向量， $TR_b$  為該遺失幀  $F_t$  與該第零參考幀  $F_{t0}$  的時間差距， $TR_d$  為該遺失幀  $F_t$  與該第零對應參考幀  $F_{t0}^l$  的時間差距。

8. 如申請專利範圍第7項所述之於可調式影像解碼的全畫面錯誤掩蓋方法，其中，於步驟(G)中將該方塊由  $M \times N$  個像素擴充為  $(M + \text{abs}(v_{0x}) + \text{abs}(v_{1x})) \times (N + \text{abs}(v_{0y}) + \text{abs}(v_{1y}))$  個像素的該擴充區域，當中  $v_{0x}$  為運動向量  $\vec{v}_0$  在 x 軸的大小， $v_{0y}$  為運動向量  $\vec{v}_0$  在 y 軸的大小，當中  $v_{1x}$  為運動向量  $\vec{v}_1$  在 x 軸的大小， $v_{1y}$  為運動向量  $\vec{v}_1$  在 y 軸的大小。

9. 如申請專利範圍第8項所述之於可調式影像解碼的全畫面錯誤掩蓋方法，其中，該步驟(G)更包含：

(G1) 獲取該擴充區域的一像素，其中該擴充區域係由該方塊及一擴展區域所組成；

(G2) 獲取該像素的位置；

(G3) 判斷該像素的位置是否位於該擴展區域；

(G4) 若步驟(G3)判定該像素的位置位於該擴展區域(非4X4的部分)，再判斷該像素是否已有重建值，且位於先前重建方塊的非擴展區域；

(G5) 若判定該像素已有重建值且位於該先前重建方塊的非擴展區域，再判斷該像素是否為該擴充區域最後一像素，若是，結束步驟(G)，若否，獲取下一像素，並執行步驟(G2)。

10. 如申請專利範圍第9項所述之於可調式影像解碼的全畫面錯誤掩蓋方法，其更包含下列步驟：

(G6) 若步驟(G4)判定該像素沒有重建值，且非位於先前重建方塊的非擴展區域，再判斷該像素是否尚未被重建；

(G7) 若步驟(G6)判定該像素尚未被重建，則對該像素進行重建，並執行步驟(G5)；

(G8) 若步驟(G6)判定該像素已被重建，表示該像素已有重建值且位於該先前重建方塊的擴展區域，則對該像素進行重建並與該先前重建方塊的擴展區域之重建值平均，再執行步驟(G5)。

11. 如申請專利範圍第10項所述之於可調式影像解碼的全畫面錯誤掩蓋方法，其更包含下列步驟：

(G9)若步驟(G3)判定該像素的位置(x,y)非位於該擴展區域，再判斷該像素是否已有重建值，且位於該先前重建方塊的非擴展區域，若是，執行步驟(G8)，若否執行步驟(G7)。

八、圖式 (請見下頁)：

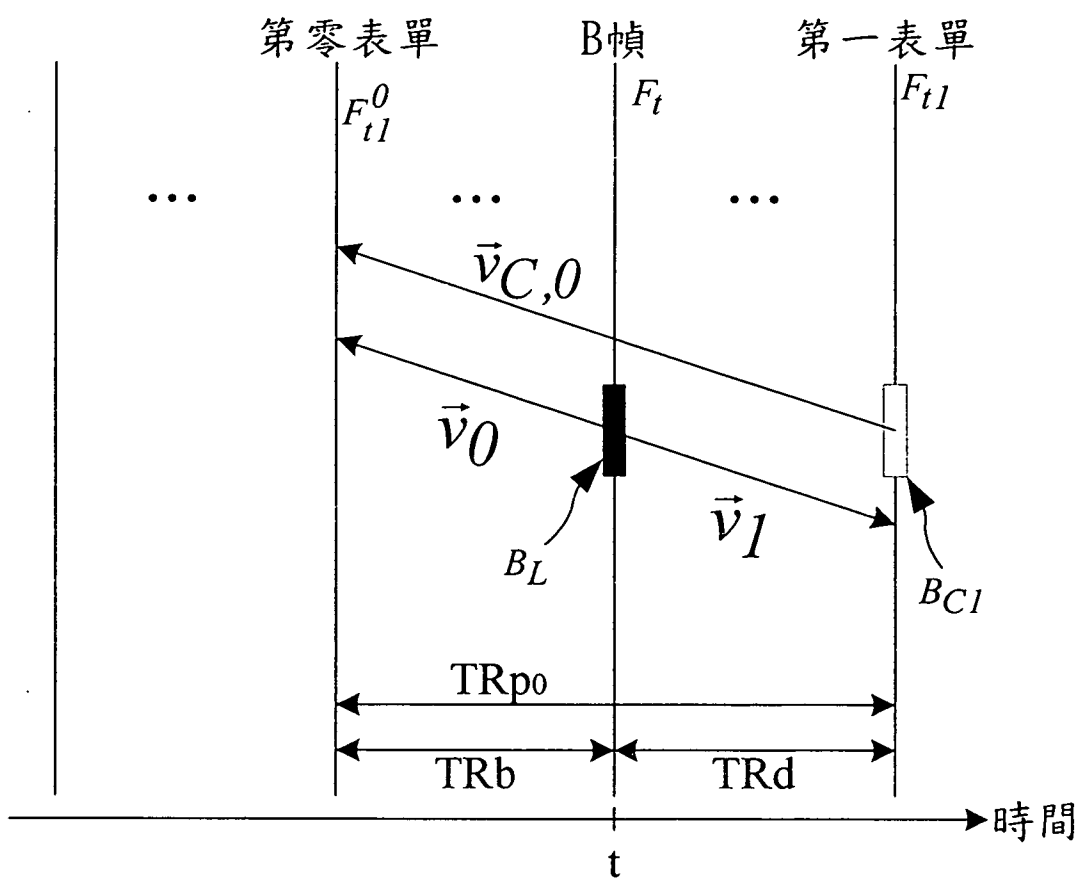


圖 1

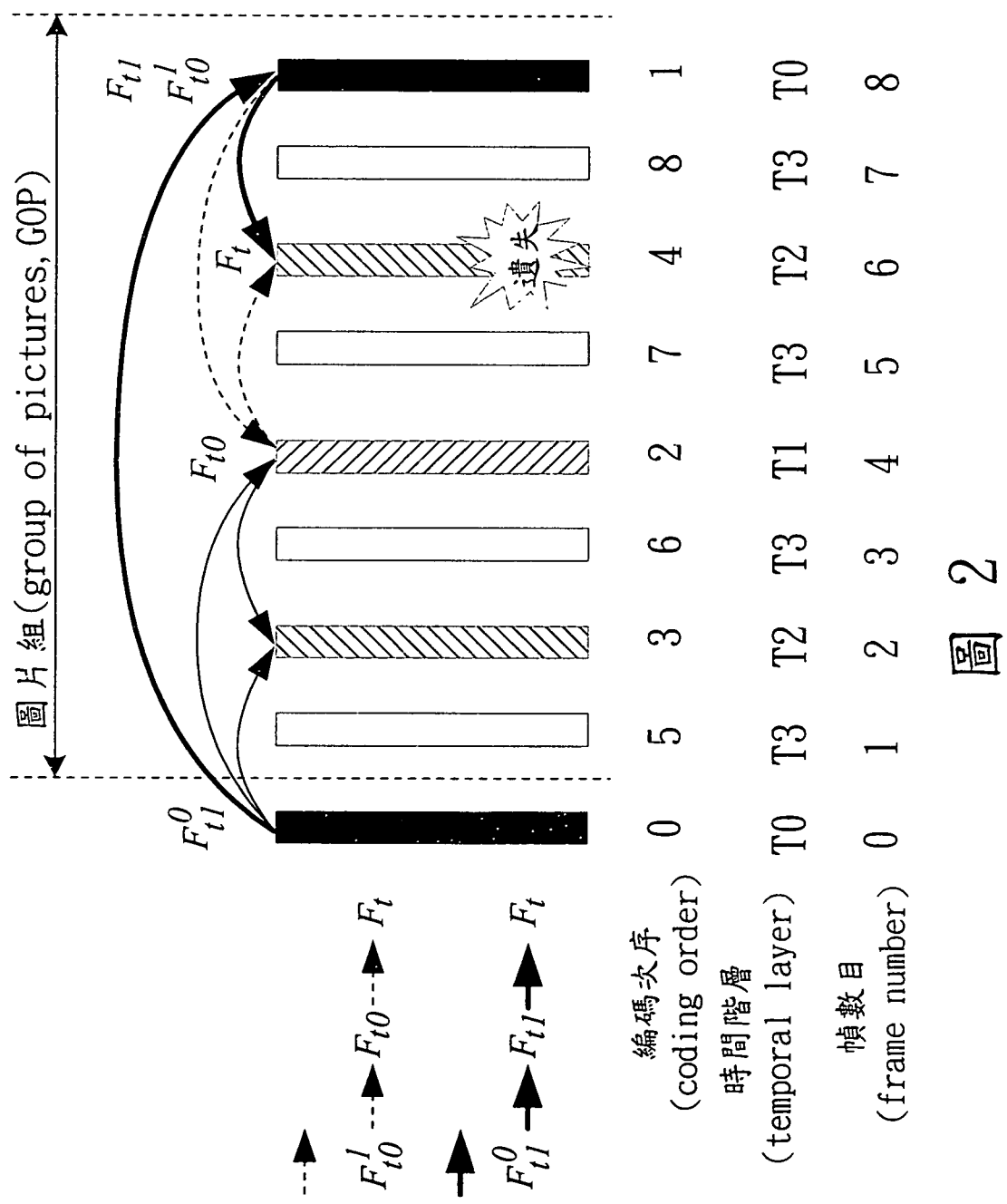


圖 2

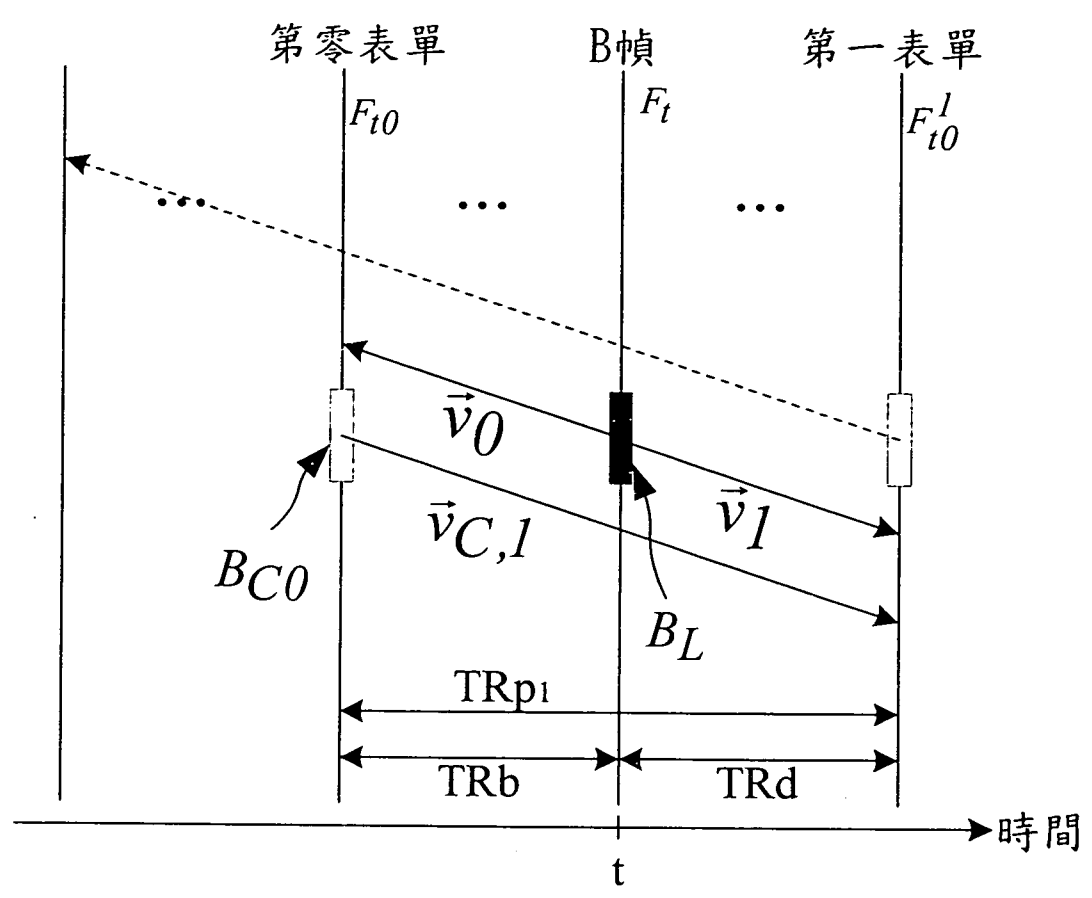


圖 3

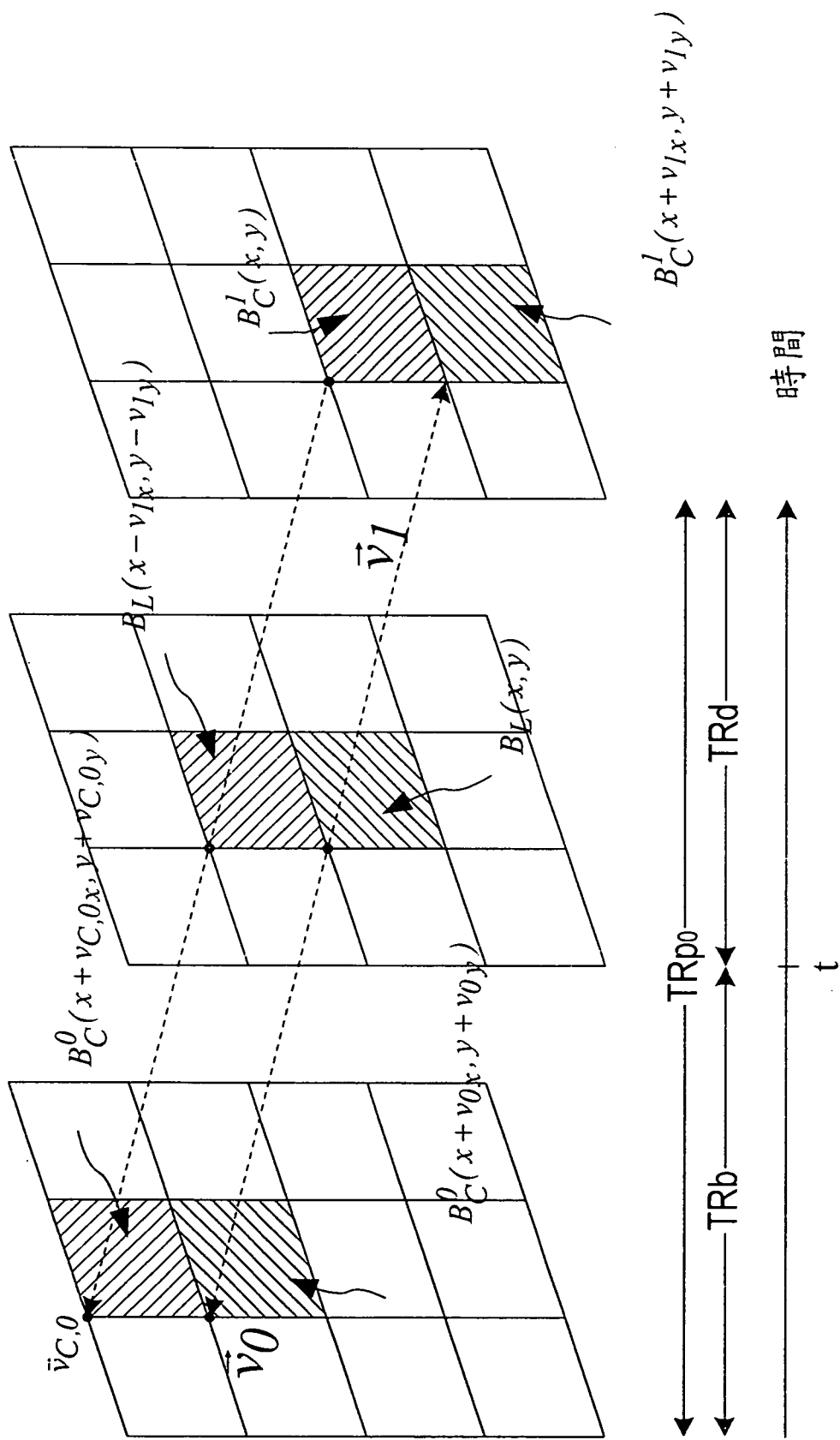


圖 4

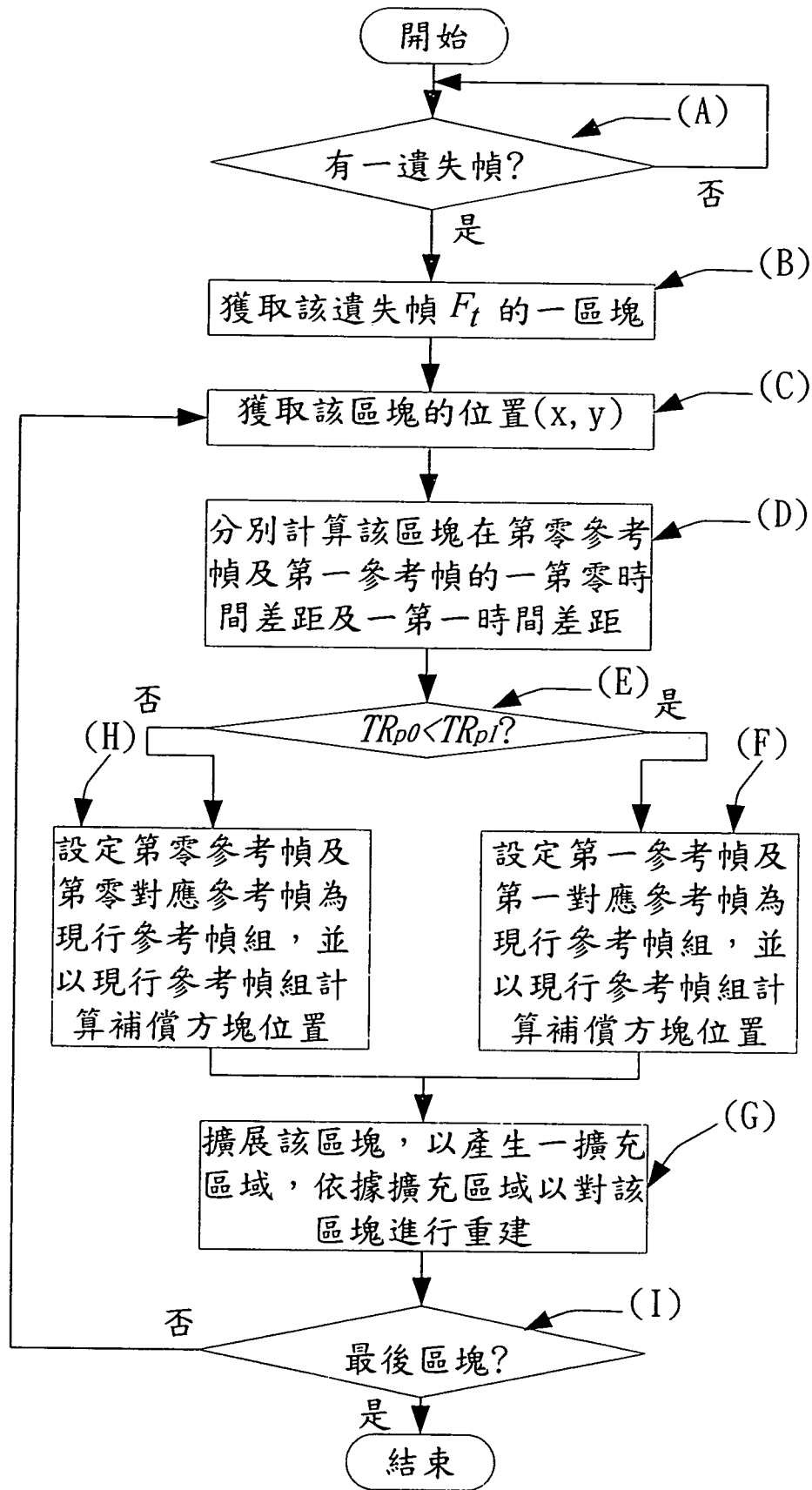


圖 5

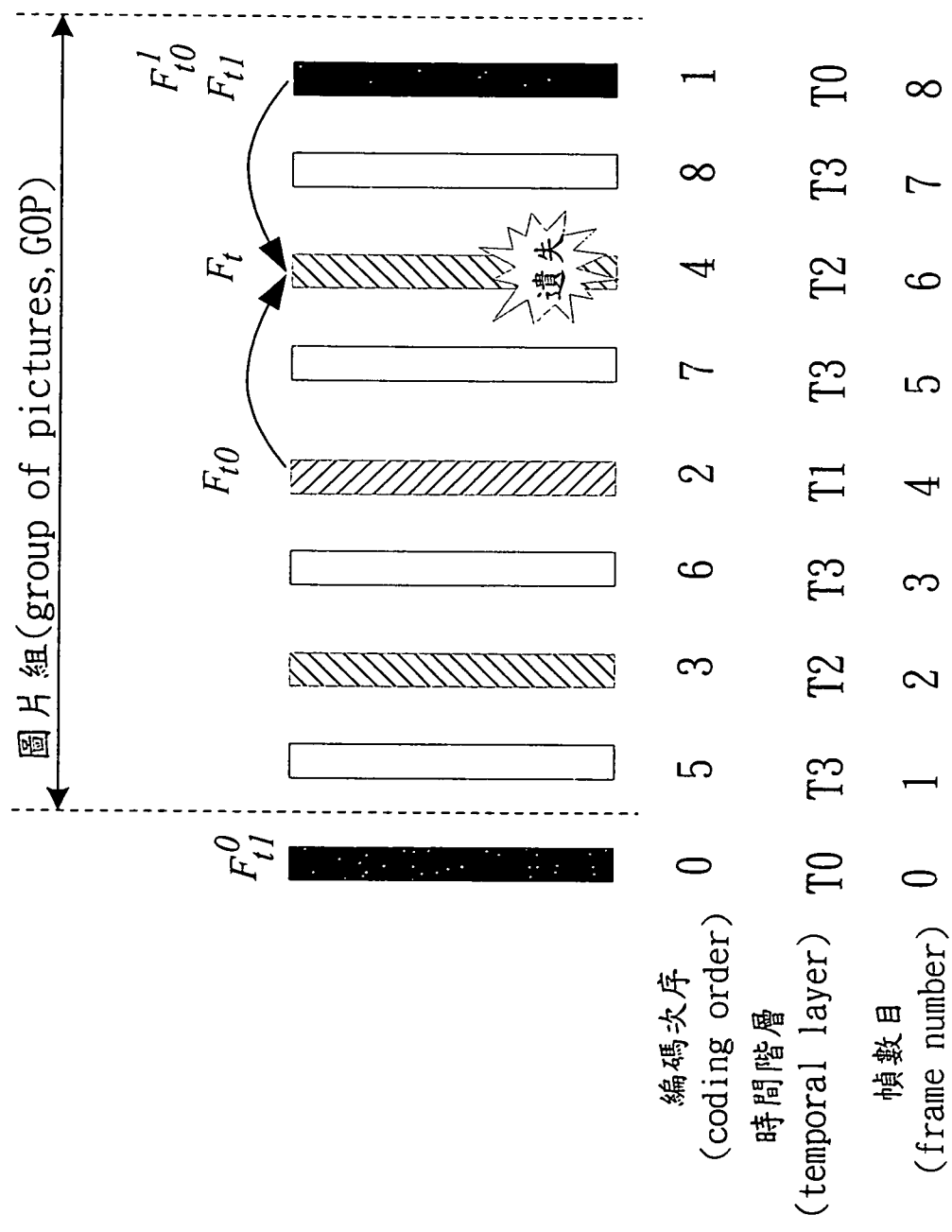


圖 6

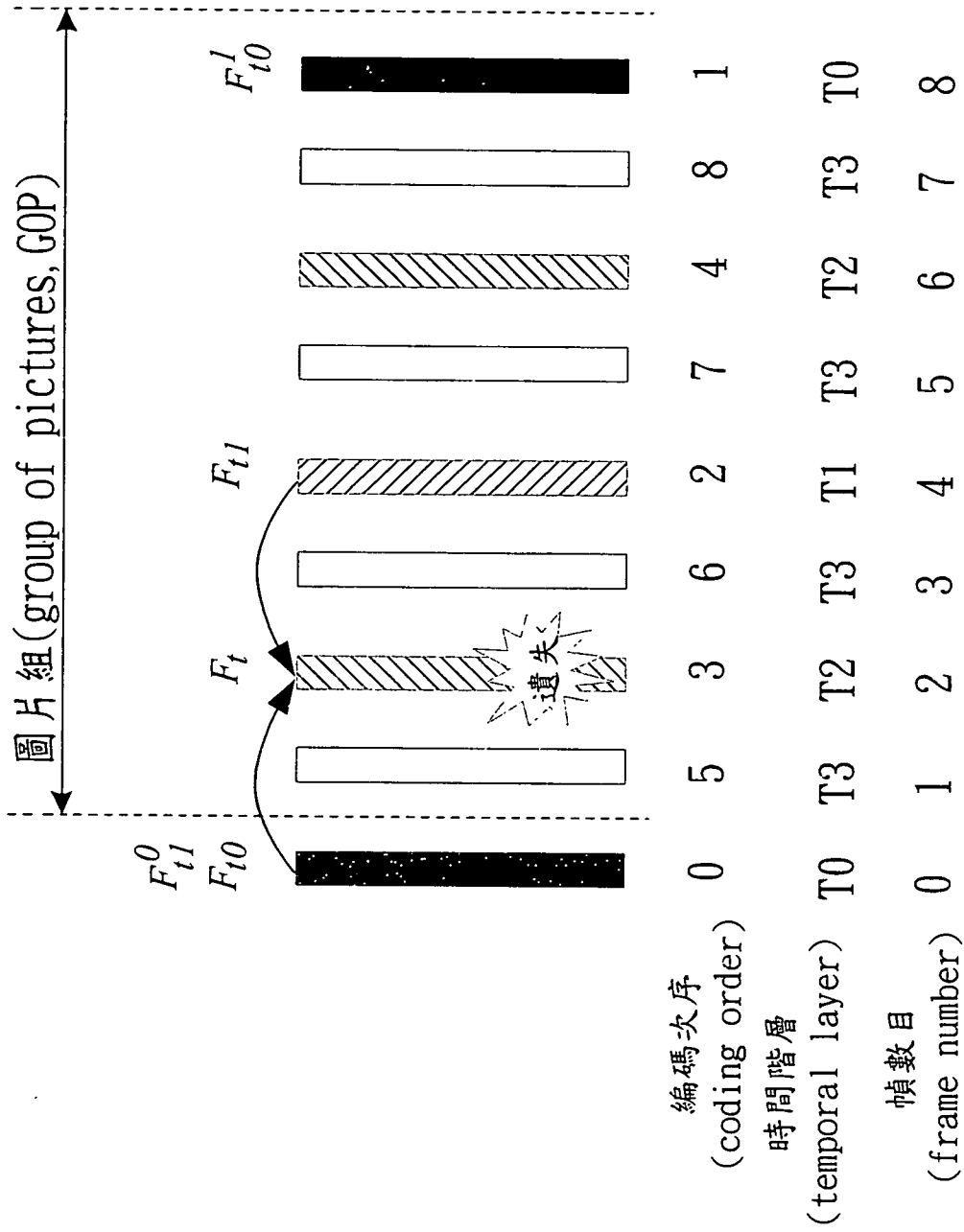


圖 7

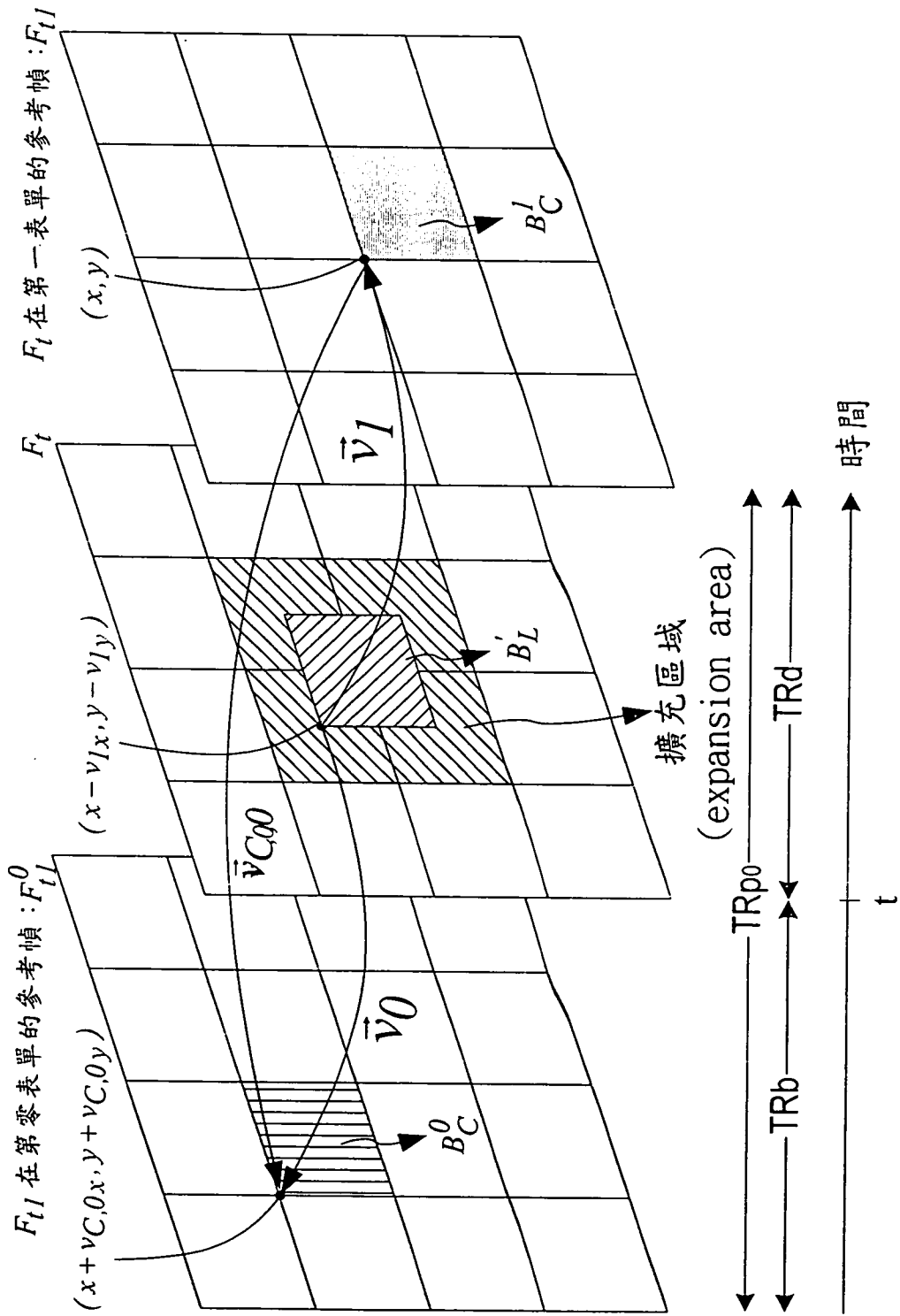


圖 8

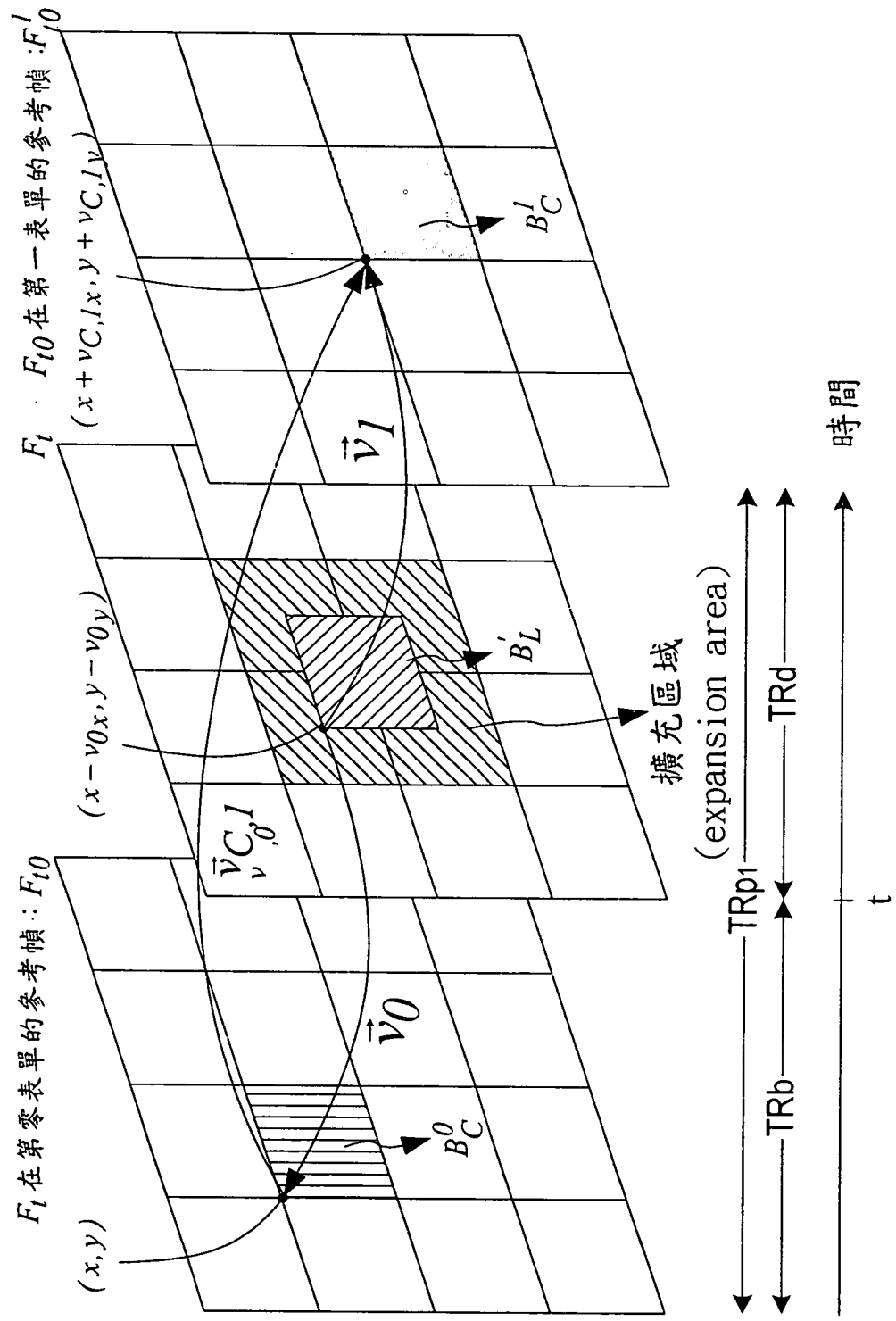


圖 9

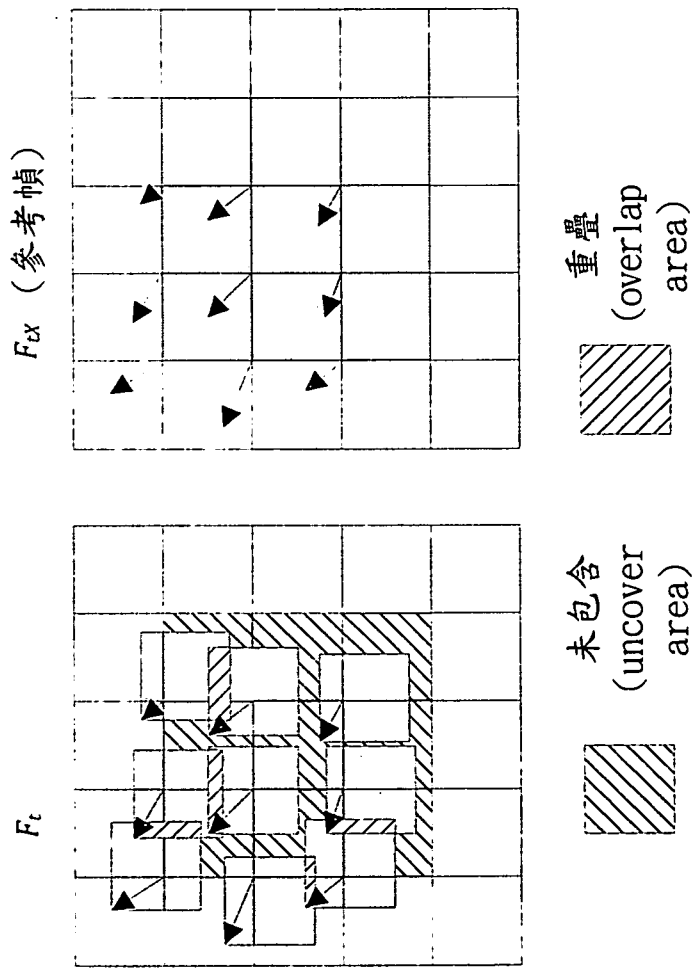


圖 10

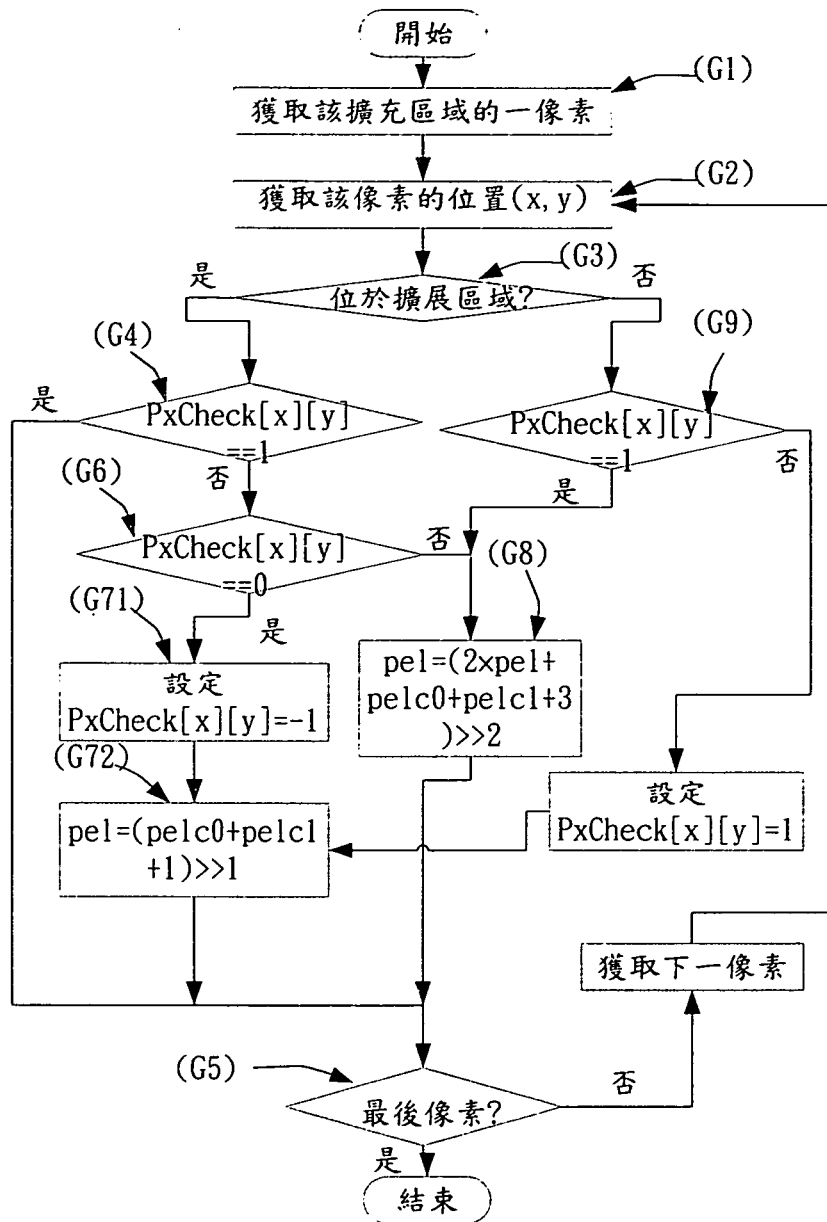


圖 11

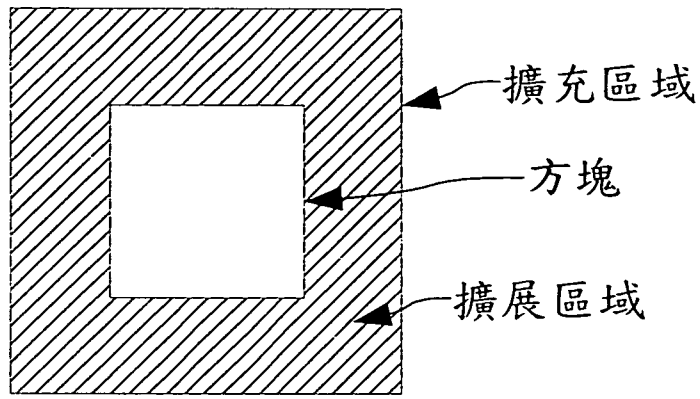
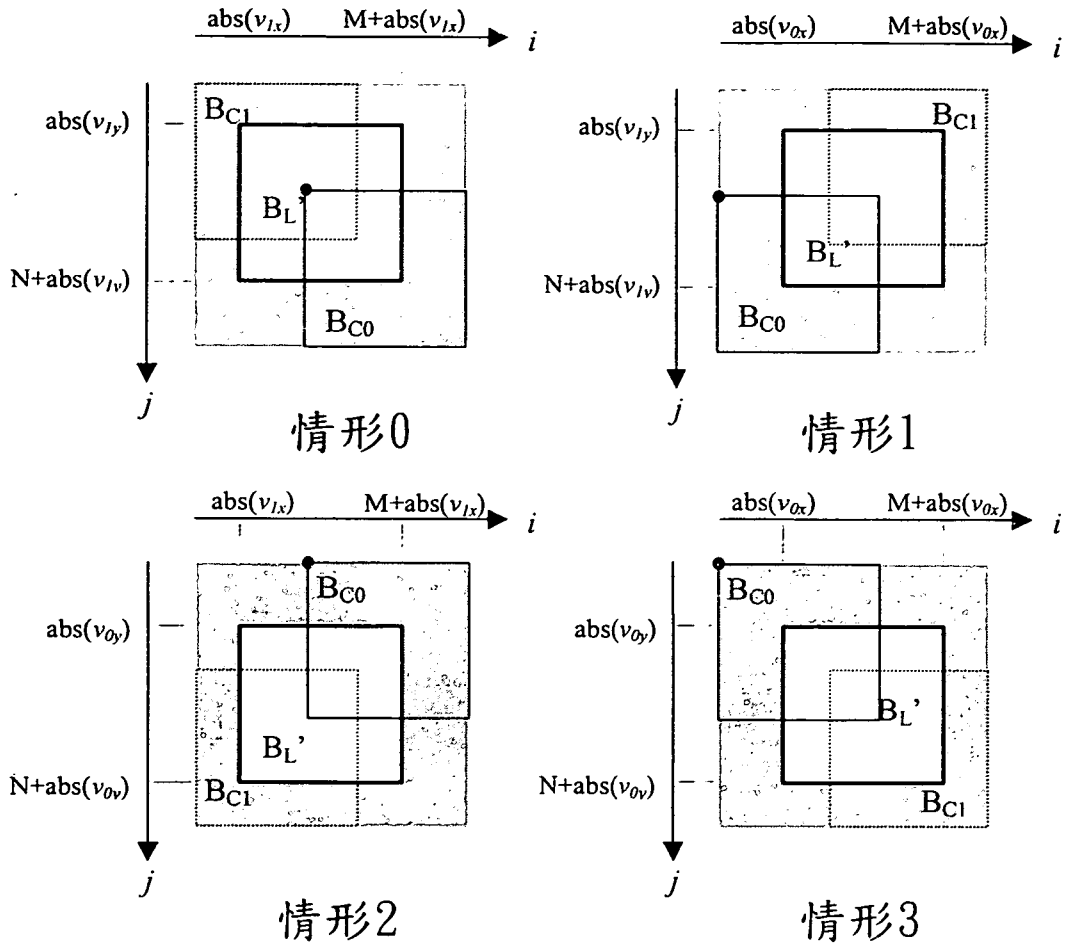
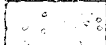


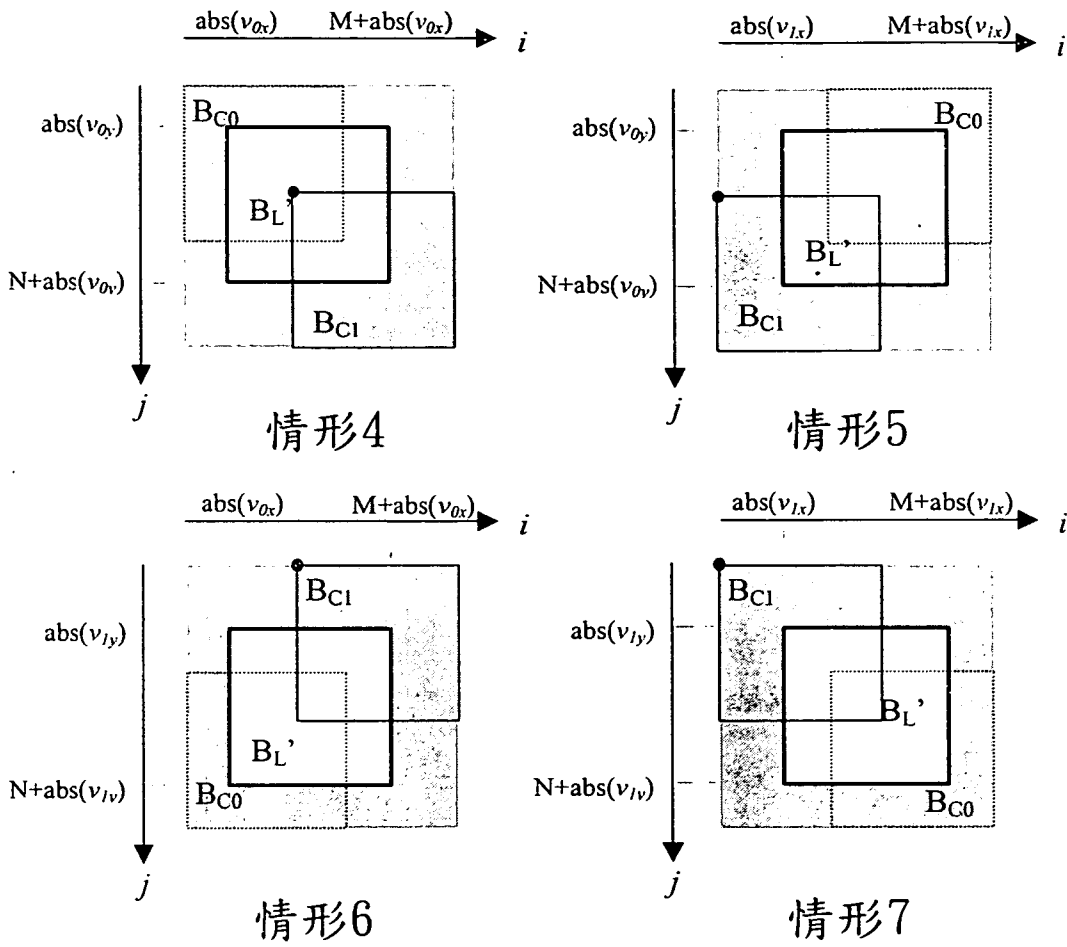
圖 12



 : 擴展區域

• : 現行區塊左上角位置(x, y)

圖 13



□ : 擴展區域

• : 現行區塊左上角位置(x, y)

圖 14