

應用於多點視訊會議之視訊轉換編碼技術

林嘉文

前言

在視訊會議中，針對各種網路環境及視訊設備用戶之不同需求往往需進行頻寬及畫面格式轉換，而視訊轉換編碼技術則是因應此等轉換需求所發展的一種技術。本文將介紹各種常用之視訊轉換編碼器架構及分析其效能。並說明如何利用多點視訊會議之特性，在視訊轉換編碼器中利用動態位元分配技術，以獲得較佳之畫面品質。

簡介

網路媒體服務，如：視訊會議 [2]、隨選視訊，以及遠距教學等在網際網路上之應用已日形普遍，逐漸和日常生活相結合。而在這些網路多媒體之應用中，常需調整視訊資料之頻寬，以配合不同用戶網路之需求。如圖 1 所示，在異質網路 (heterogeneous network) 上，此種頻寬調適允許不同子網路 (sub-network) 上之不同用戶可以根據他們可用的通道頻寬容量，獲得不同等級的服務品質 (Quality of Service ; QoS)。在一些即時視訊編碼應用中，可利用視訊編碼器的速率控制 (rate control) 技術達到此種頻寬調整需求 [9]；但是在很多其他應用，如隨選視訊，因其存放之視訊資料為已經過壓縮之視訊位元流 (video bitstream)，並無法直接使用此速率控制技術進行頻寬轉換。

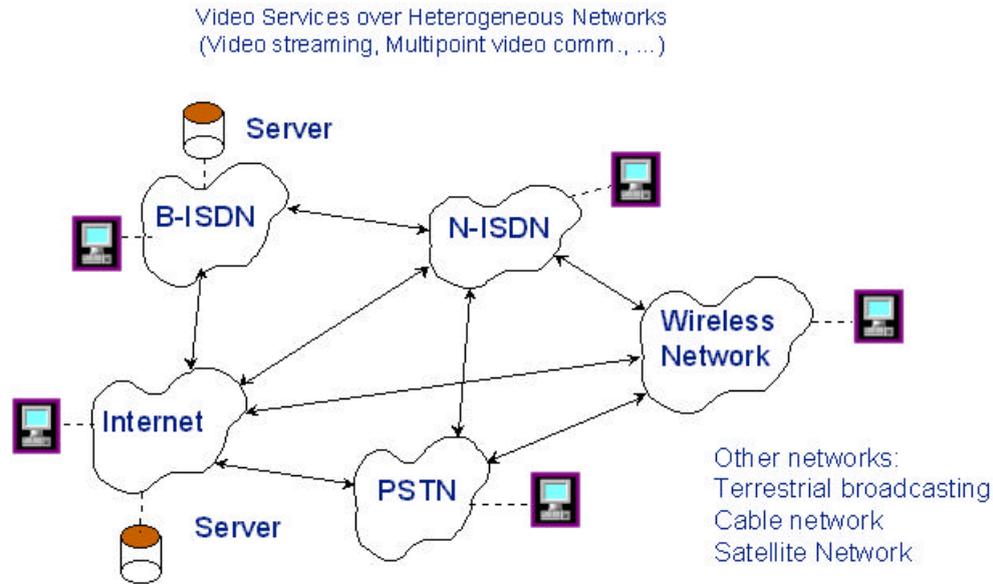


圖 1 異質網路上之視訊通訊

視訊會議為傳統語音通訊服務之自然延伸。由於數位視訊處理技術、低成本視訊編解器、及網路技術的急速發展及完善國際標準的推波助瀾，視訊會議應用正日益廣泛。如圖 2 範例所示，在一個廣域網路上所舉行的視訊會議，與會者先撥接到一個多點視訊控制伺服器(Multipoint Control Unit；MCU)，在此伺服器中之視訊橋接器會將各方送來之已壓縮視訊位元流之內容組合起來，再將所組成之新的視訊內容（例如以分割畫面顯示），送回每一個與會者。在回送組合之視訊內容時，由於資料量增大數倍，所以往往需要利用視訊轉換編碼技術 [1]，將組合後之視訊內容進行再壓縮，以配合用戶端之頻寬限制。圖 3 則顯示了一種實際的視訊會議系統範例。此系統具有 H.323 [3]及 H.324/I [4]用戶介面使得區域網路用戶及 ISDN 用戶皆可連上此系統進行視訊會議。在此視訊會議伺服器中包含一個多點會議控制器(Multipoint Controller；MC)模組負責多點視訊會議議程建立及管理、控制；並包含一個多點處理器(Multipoint Processor；MP)模組負責綜合視音訊資料，及頻寬調整等工作。視訊轉換編碼器即包含於 MP 模組中。

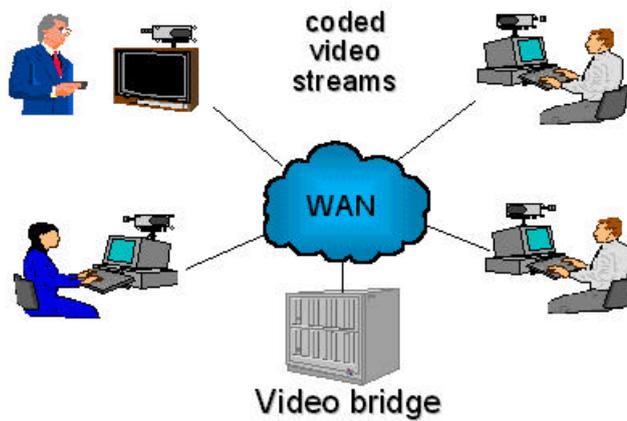


圖 2 集中型多點視訊會議

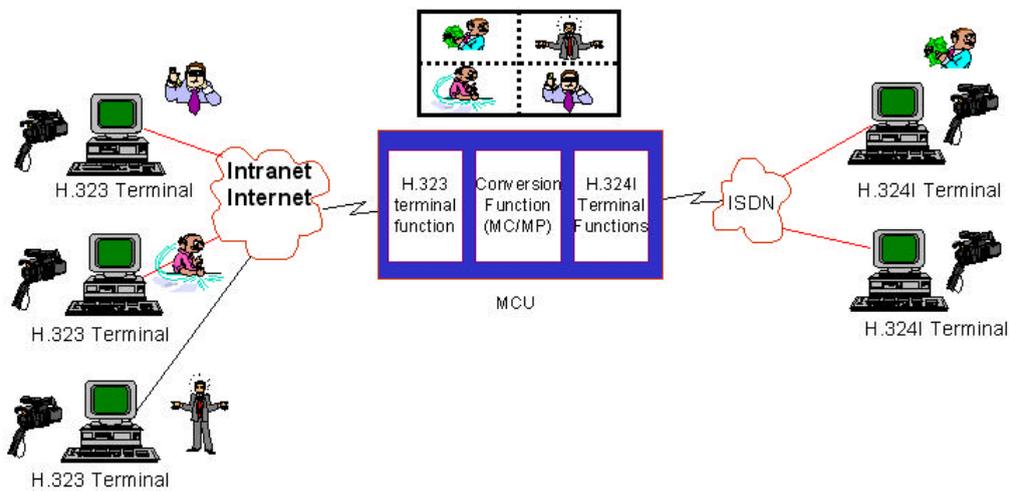


圖 3 一個 H.323/H.324 MCU 範例

基本視訊轉換器架構

一般而言，視訊轉換編碼器在頻寬轉換之應用為降低頻寬，若需提高視訊位元流之頻寬則僅需加上一些填補位元即可。而最簡單之降低頻寬的方法，如圖 4 所示，為利用一個開迴路之轉換編碼器直接將高頻的 DCT 係數移除或是執行一個再量化之程序 [9-10]。此種做法之運算複雜度極低，但在以開迴路形式刪改視訊位元流內容後，會造成在前端視訊編碼器及後端視訊解碼器重建影像內容不吻合(mismatch)的問題，而產生隨時間累積之漂移誤差(Drift)，因而造成令人無法

忍受的嚴重畫質失真。

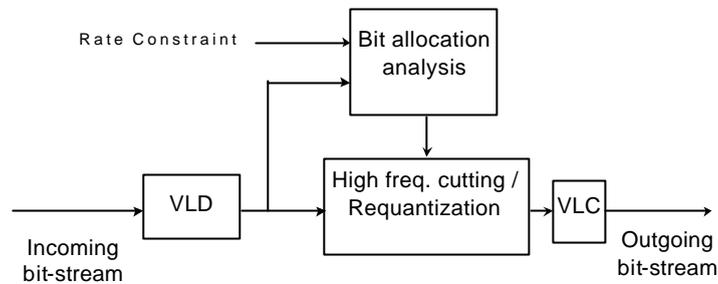


圖 4 開迴路視訊轉化編碼器架構

串接式視訊轉換編碼器

另一種轉換編碼器之架構如圖 5，為直接串接一個視訊解碼器及一個視訊編碼器。前級視訊解碼器先將視訊位元流解碼成像素值，而後級視訊編碼器再利用速率控制技術將輸出位元率轉換成所需頻寬。此種架構不會產生因重建影像不吻合所導致的漂移誤差，所以可得到較佳之視訊品質，但其運算複雜度太高，並不適合在需即時處理之應用。圖 6 顯示了以 H.263 [6] 視訊編解碼器實現此一架構之功能方塊圖。

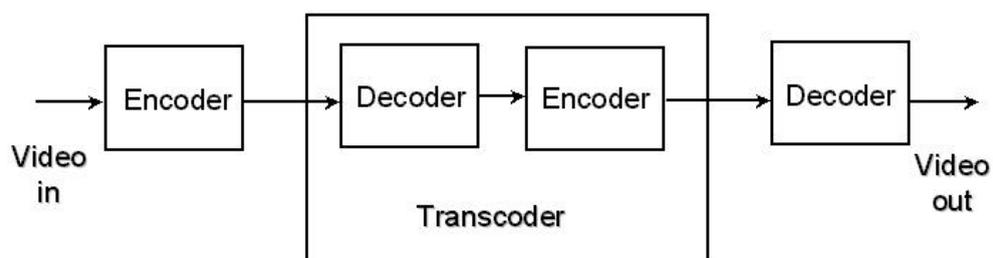


圖 5 串接式視訊轉換編碼器

在此 $S[I_n^{(i)}, \underline{V}_n]$ 為運動補償所須之資料搬移運算，亦即將第 i 個畫面儲存記憶體的第 n 個畫面中的區塊搬動 \underline{V}_n 之位置。

由於在前級解碼器以接收了相當多有用之資訊，如編碼模式、量化步階、位元分配資訊、運動向量資訊等可作為後級編碼器之參考。善加利用這些資訊，可大幅簡化所須之運算複雜度。在[11]中提出了一種運動向量再利用(motion vector reuse)使得轉換編碼器架構可以大幅簡化之方法。如果在後級編碼器重複利用前級解碼器所解碼得之運動向量資訊（亦即 $\underline{V}_n^{(1)} = \underline{V}_n^{(2)} = \underline{V}_n$ ），則不僅可以在後級編碼器節省運動估測計算，亦可進一步節省其他部分之 DCT/IDCT 運算急畫面內容儲存記憶體等。利用運動向量再利用之法，(5)式可簡化為

$$\text{DCT}(R_n^{(2)}) = \text{DCT}(R_n^{(1)}) - \text{DCT}(S[E_{n-1}^{(2)}, \underline{V}_n^{(2)}]) \quad (6)$$

利用(6)式，我們可得到一個簡化的架構如圖 7 所示。和串接式架構相較，如果皆重複使用所接收之運動資訊，此簡化架構可節省一個 IDCT 運算及一個畫面儲存記憶體。

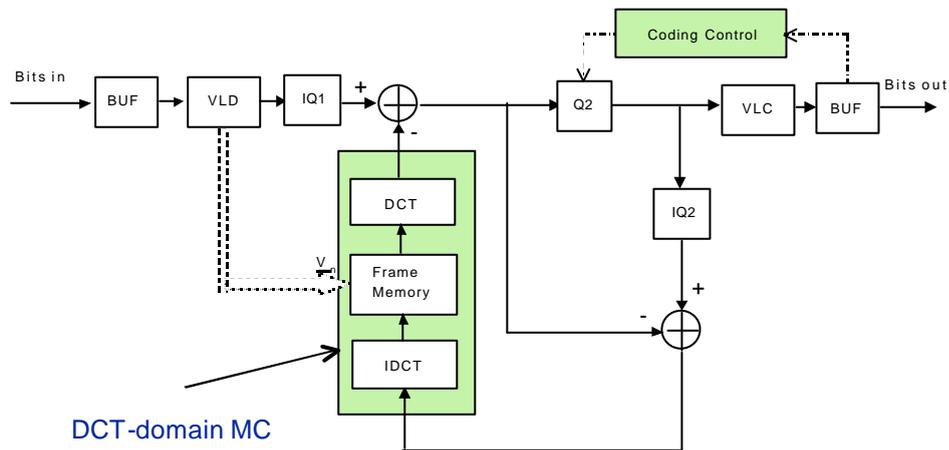


圖 7 串接式視訊轉換編碼器功能方塊圖

在圖 7 中，後級編碼器之 DCT 係數之量化誤差經由 IDCT 運算解碼成像素值

後存放至畫面儲存記憶體。然後在像素域進行運動補償運算，並將所得之預測值利用 DCT 運算轉換成 DCT 係數，如框中所示。此整體運算程序共需一個 IDCT 運算、一個 DCT 運算及一個畫面位移運算，為主要運算複雜度所在。

DCT 域簡化型視訊轉換編碼器架構

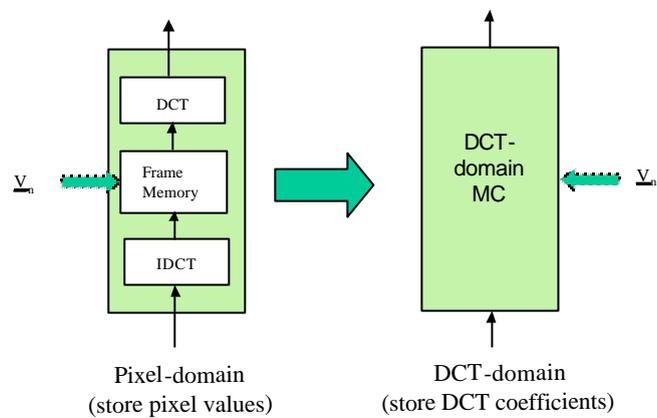


圖 8 像素域和 DCT 域運動補償運算

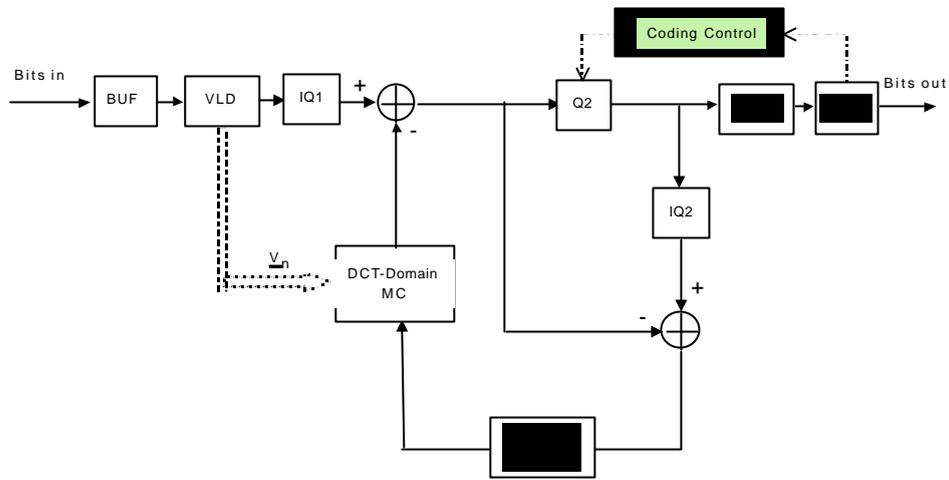


圖 9 DCT 域簡化型視訊轉換編碼架構

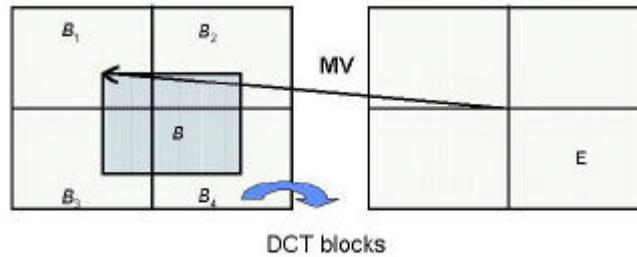


圖 10 DCT 域運動補償運算圖示說明

上述之圖 7 運算，也可以直接在 DCT 域中進行，以避免所需之 DCT 及 IDCT 運算，如圖 8 所示。並形成了如圖 9 之 DCT 域簡化型視訊轉換編碼架構。圖 10 中則說明了在 DCT 域運算之做法，乃直接由相鄰的 4 個參考區塊的 DCT 係數值及運動向量值，計算出經位移後之目標區塊之 DCT 係數值，其中每個區塊大小為 8×8 。此種做法之數學原理如下所示：

一個 8×8 區塊之像素值 $\mathbf{x} = \{x(n,m)\}_{n,m=0}^7$ 可經由 2-D 之 DCT 運算轉換成 8×8 之 DCT 係數 $X = \{X(u,v)\}_{u,v=0}^7$ ，如(7)式所示

$$X(u, v) = \frac{c(u)}{2} \frac{c(v)}{2} \sum_{n=0}^7 \sum_{m=0}^7 x(n, m) \cos\left(\frac{2n+1}{16} \cdot u\mathbf{p}\right) \cos\left(\frac{2m+1}{16} \cdot v\mathbf{p}\right) \quad (7)$$

若定義一個 1-D 8 點之 DCT 矩陣 $S = \{s(u, n)\}_{u, n=0}^7$ ，其中

$$s(u, n) = \frac{c(u)}{2} \cos\left(\frac{2n+1}{16} \cdot u\mathbf{p}\right) \quad (8)$$

則可得到其矩陣-向量表示式如下

$$X = S\mathbf{x}S^t \quad (9)$$

[12]中首先提出一種 DCT 域計算方法以解決圖 8 之問題。在此方法中將目標區域 \mathbf{b} 表示為 4 個組成區塊 \mathbf{b}_{14} ， \mathbf{b}_{23} ， \mathbf{b}_{32} ，及 \mathbf{b}_{41} 的和，而此四個成分區塊則源自於其四個相鄰區塊 $\mathbf{b}_1 \sim \mathbf{b}_4$ ，如圖 11 及圖 12 所示。

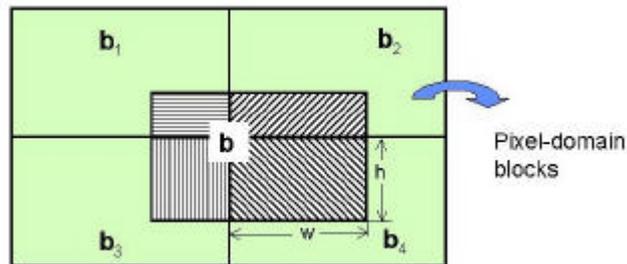


圖 11 位移目標區塊 \mathbf{b} 和其相鄰參考區塊 $\mathbf{b}_1 \sim \mathbf{b}_4$ 之幾何關係

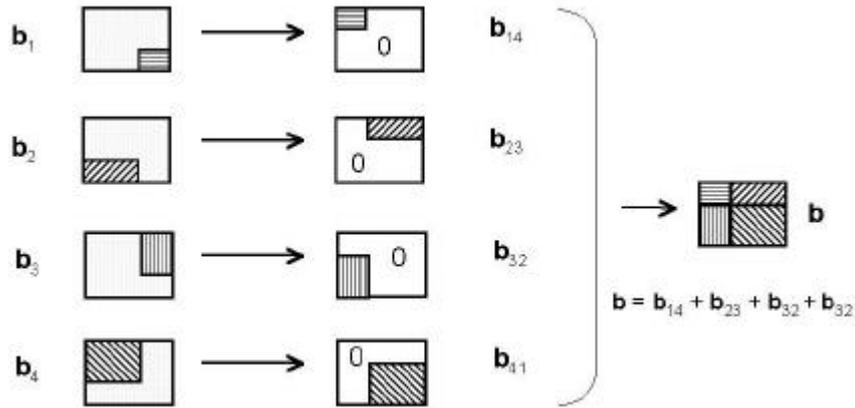


圖 12 將鄰近參考區塊 $b_1 \sim b_4$ 利用幾何轉換以形成目標區塊 b

在[12]中利用一種幾何轉換技巧推導出了 b_{14} , b_{23} , b_{32} , 及 b_{41} 個別的 DCT 係數矩陣 B_{14} , B_{23} , B_{32} , 及 B_{41} 和 $b_1 \sim b_4$ 的 DCT 矩陣 $B_1 \sim B_4$ 之間的關係。使得目標區塊 b 之 DCT 係數矩陣 B 可由 $B_1 \sim B_4$ 利用一些矩陣運算求得，而無需任何的 DCT 及 IDCT 運算。在此以成分區塊 b_{41} 為例說明此運算程序。 b_{41} 和其對應之參考區塊 b_4 之幾何轉換關係如(11)式及圖 13

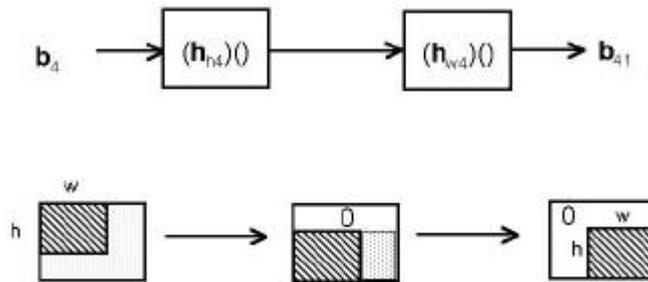


圖 13 幾何轉換圖示

$$b_{41} = h_{h_4} b_4 h_{w_4}, \text{ 在此 } h_{h_4} = \begin{bmatrix} 0 & 0 \\ I_{h_4} & 0 \end{bmatrix} h_{w_4} = \begin{bmatrix} 0 & I_{w_4} \\ 0 & 0 \end{bmatrix} \quad (11)$$

在此 I_{h_4} 及 I_{w_4} 分別為 $h_4 \times h_4$ 和 $w_4 \times w_4$ 大小之單位矩陣。利用 DCT 轉換的乘法分配性，(亦即 $\text{DCT}(\mathbf{ab}) = \text{DCT}(\mathbf{a})\text{DCT}(\mathbf{b})$)，可得到

$$B_{41} = H_{h_4} B_4 H_{w_4} \quad (12)$$

在此 $B_{41} = \text{DCT}(\mathbf{b}_{41})$, $B_4 = \text{DCT}(\mathbf{b}_4)$, $H_{h_4} = \text{DCT}(\mathbf{h}_{h_4})$, 及 $H_{w_4} = \text{DCT}(\mathbf{h}_{w_4})$

經由此種幾何轉換，目標區塊之 DCT 係數 B 可表示為

$$B = \sum_{i=1}^4 H_{h_i} B_i H_{w_i} \quad (13)$$

直接計算(13)式共需 8 個矩陣乘法和 3 個矩陣加法。若設 $h_4=h$, $w_4=w$, 表示顯示了各個幾何轉換矩陣之值，並顯示了後述之等式關係： $\mathbf{h}_{h_1} = \mathbf{h}_{h_2} = \begin{bmatrix} 0 & I_{8-h} \\ 0 & 0 \end{bmatrix}$,

$$\mathbf{h}_{h_3} = \mathbf{h}_{h_4} = \begin{bmatrix} 0 & 0 \\ I_h & 0 \end{bmatrix} , \quad \mathbf{h}_{w_1} = \mathbf{h}_{w_3} = \begin{bmatrix} 0 & 0 \\ I_{8-w} & 0 \end{bmatrix} , \text{ and } \mathbf{h}_{w_2} = \mathbf{h}_{w_4} = \begin{bmatrix} 0 & I_w \\ 0 & 0 \end{bmatrix}$$

表 1 \mathbf{h}_{h_i} 和 \mathbf{h}_{w_i} 等矩陣

相鄰區塊	位置	\mathbf{h}_{h_i}	\mathbf{h}_{w_i}
\mathbf{b}_1	右下角	$\begin{bmatrix} 0 & I_{8-h} \\ 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 \\ I_{8-w} & 0 \end{bmatrix}$
\mathbf{b}_2	左下角	$\begin{bmatrix} 0 & I_{8-h} \\ 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & I_w \\ 0 & 0 \end{bmatrix}$
\mathbf{b}_3	右上角	$\begin{bmatrix} 0 & 0 \\ I_h & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 \\ I_{8-w} & 0 \end{bmatrix}$
\mathbf{b}_4	左上角	$\begin{bmatrix} 0 & 0 \\ I_h & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & I_w \\ 0 & 0 \end{bmatrix}$

利用這些等式關係，(13)式之計算可簡化為 6 個矩陣乘法及 3 個矩陣加法。而 H_{h_i} 及 H_{w_i} 之係數皆可預先計算儲存後查表，無須額外之計算。

由於 $B_1 \sim B_4$ 通常為稀疏矩陣，所以可用一些稀疏矩陣運算的技巧再進一步簡化其計算複雜度。在 [13-14] 中提出了一種矩陣分解之技巧，而在 [] 中將 H_{h_i} 及

H_{w_i} 以簡化的 2 的幕級數加以近似以降低約 80% 之計算量。在 [19] 中則提出一種資訊分享的方法(shared information)可進一步降低 50% 之計算量。此外，若是水平或垂直之運動向量為零時，計算量可再大幅降低，所以整體的運算複雜度可較像素域之簡化型架構為低。

效能比較及評估

我們實際模擬比較了前述 3 種轉化編碼器架構之效能。模擬時使用了兩種國際標準測試影像序列。對每一序列利用量化步階 $Q=5$ 先行編碼後，再利用 $Q=7, 10, 15$ 及 18 利用 3 種不同架構進行轉換編碼。表 2 中顯示了經由各種架構及量化步階值轉化編碼所得之平均峰值信號雜訊比(Peak Signal-to-Noise Ratio; PSNR)及位元率。圖 14 顯示了 PSNR 對位元率之模擬結果，而圖 15 及 16 則分別顯示了利用 $Q=10$ 進行轉換編碼後，兩種測試序列對 3 種不同架構每一畫面的 PSNR 及位元率。由上述之模擬結果，我們發現了一個有趣的現象，即在數學分析上雖然 3 種架構是等效的（都基於運動向量再利用之方法），但其實體效能卻有明顯的差異性。其中串接式架構的效能最佳，而像素域簡化型架構的效能則明顯為最差。分析其主要原因為，在進行架構簡化時，再參考文獻中假設了運動補償是一種像性運算，DCT 及 IDCT 可以彼此互消，且 DCT/IDCT 具備加法及乘法之分配性等。但在實際的情形中，由於 DCT 及 IDCT 的有限字元長度所造成的誤差，將像素值限制於 $0\sim 255$ 間及 DCT 係數限制於一定限之箝位(clipping)運算，及因為半像素(half-pixel)運動補償所產生之非線性誤差皆可能導致明顯的漂移誤差效應[8]。

表 2

經由 3 個不同轉換編碼架構的平均 PSNR 及位元率比較。前端編碼器所使用的 Q 值為 5，而轉換編碼器分別使用 $Q=7, 10, 15$ 及 18 。

(a) 串接型視訊轉換編碼架構

Sequence	Quant	Average PSNR (dB)	Average Bit-rate (Kbits/s)
Foreman	Q=7	35.823	198.47
	Q=10	32.918	120.34
	Q=15	30.552	73.47
	Q=18	29.458	59.40

Carphone	Q=7	37.207	175.05
	Q=10	34.213	105.53
	Q=15	31.674	63.21
	Q=18	30.540	50.21

(b) DCT 域簡化型視訊轉換編碼架構

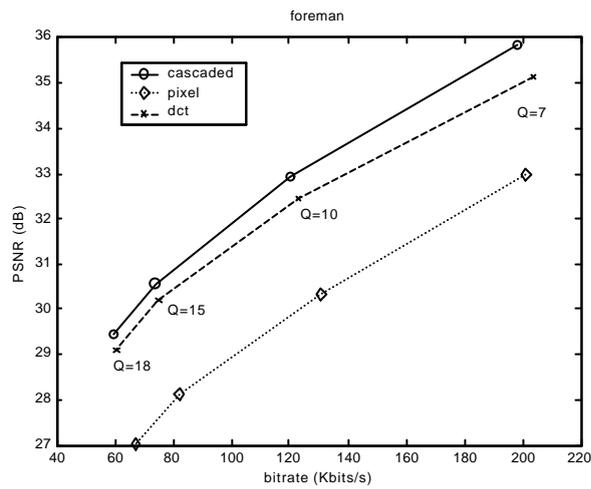
Sequence	Quant	Average PSNR (dB)	Average Bit-rate (Kbits/s)
Foreman	Q=7	35.119	203.74
	Q=10	32.436	123.00
	Q=15	30.191	74.92
	Q=18	29.100	60.34
Carphone	Q=7	36.450	180.82
	Q=10	33.612	107.58
	Q=15	31.230	64.16
	Q=18	29.900	50.88

(c) 像素域簡化型視訊轉換編碼架構

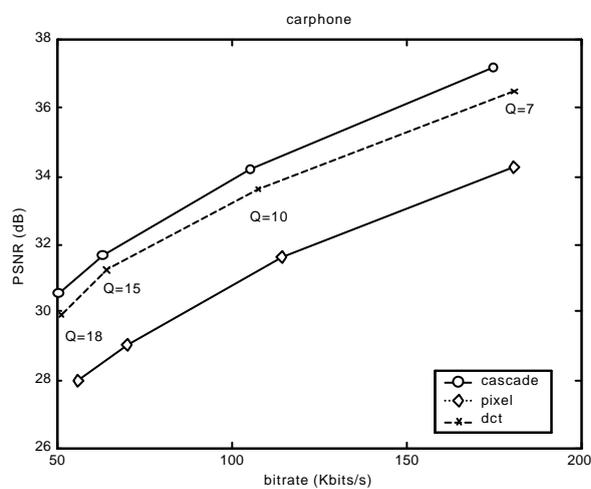
Sequence	Quant	Average PSNR (dB)	Average Bit-rate (Kbits/s)
Foreman	Q=7	32.977	201.37
	Q=10	30.326	130.64
	Q=15	28.135	82.18
	Q=18	27.054	66.83
Carphone	Q=7	34.233	180.83
	Q=10	31.618	114.32
	Q=15	29.075	70.26
	Q=18	27.963	56.06

由於 DCT 域簡化型架構系利用矩陣運算以取代 DCT/IDCT 運算，其漂移誤差遠較像素域簡化型架構為低。此種漂移效應在 H.263 等位元率的應用之影響尤為顯著，主要是在此種低位元率的應用上，很少加入 I 畫面，大部分的畫面皆是利用運動補償的方法由前一畫面預測得到，以致造成漂移誤差會隨著時間增加而產生嚴重的累積，如圖 14 所示。

在 MPEG 的應用中由於相隔一個 GoP (Group of Pictures)即會插入一個更新用的 I 畫面，所以此漂移誤差的影響並不那麼明顯。此外上述的兩種簡化型架構係基於運動向量再利用的策略所推導而得，所以在一些需要重新計算運動向量值的應用如畫面率轉換，大小轉換及其他格式轉換等，皆無法利用此等簡化型架構。

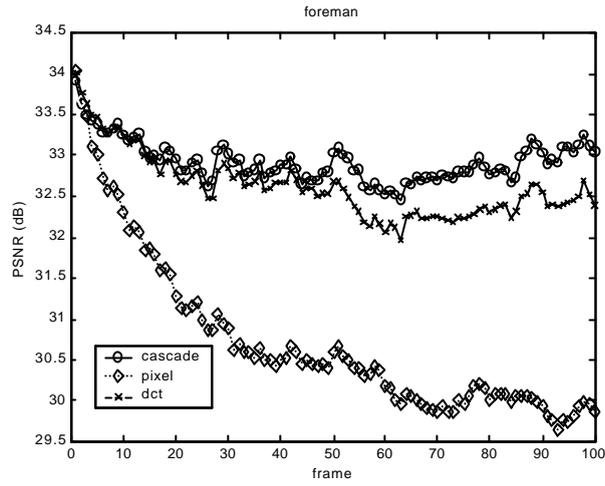


(a)

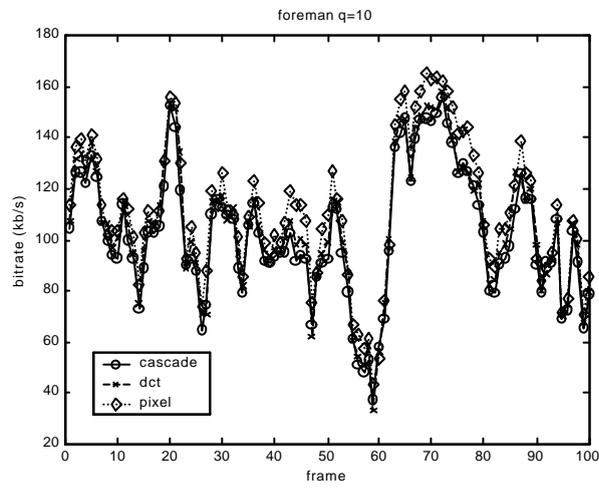


(b)

圖 14 經由 3 個不同轉換編碼架構的平均 PSNR 及位元率比較。前端編碼器所使用的 Q 值為 5，而轉換編碼器分別使用 Q=7，10，15 及 18。(a)“foreman”測試序列；(b)“carephone”測試序列。

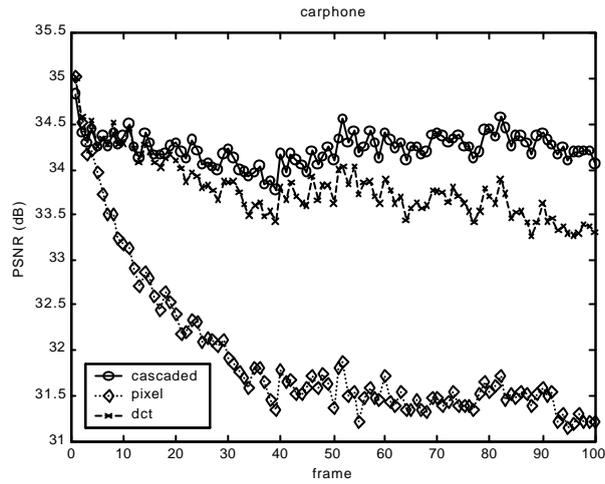


(a)

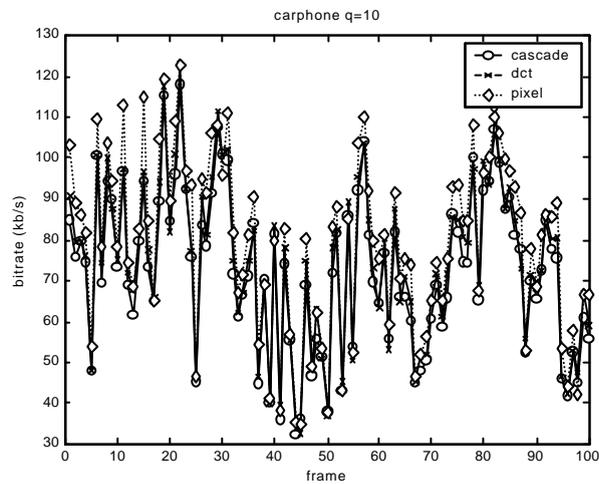


(b)

圖 15 “foreman”測試序列經由 3 種不同架構進行轉換編碼後的每一畫面 PSNR 及位元量比較。前級編碼器使用之 Q 值為 5，轉換編碼使用之 Q 值為 10。
(a)PSNR；(b)位元量。



(a)



(b)

圖 16 “carphone”測試序列經由 3 種不同架構進行轉換編碼後的每一畫面 PSNR 及位元量比較。前級編碼器使用之 Q 值為 5，轉換編碼使用之 Q 值為 10。
(a)PSNR；(b)位元量。

多點視訊轉換編碼架構

當使用於多點視訊會議應用時，需利用多點轉換編碼器以執行頻寬及格式轉換。圖 17 顯示一種針對多點視訊會議的架構實現方式，稱之為型式 I。此架構中，前級編碼器先接收所有與會者之視訊位元流，將其部分解碼後，利用視訊組

合器將所有遠方影像組合成整合畫面（例如分割畫面）。每一個與會者可以看到不同的與會者及顯示畫面。針對每一個與會者不同的頻寬及畫面格式需求，需使用不同的轉換編碼器進行轉換。這種架構的缺點是由於一個與會者的影像畫面，可能會與多個與會者所共享，型式 I 並無法利用到此種共享的特性而節省運算複雜度。

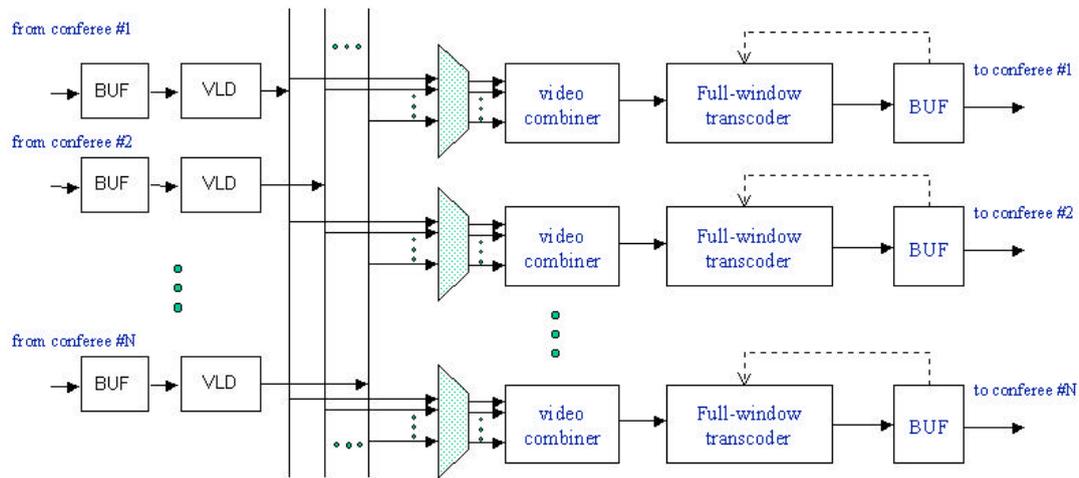


圖 17 型式 I 多點視訊轉換編碼架構

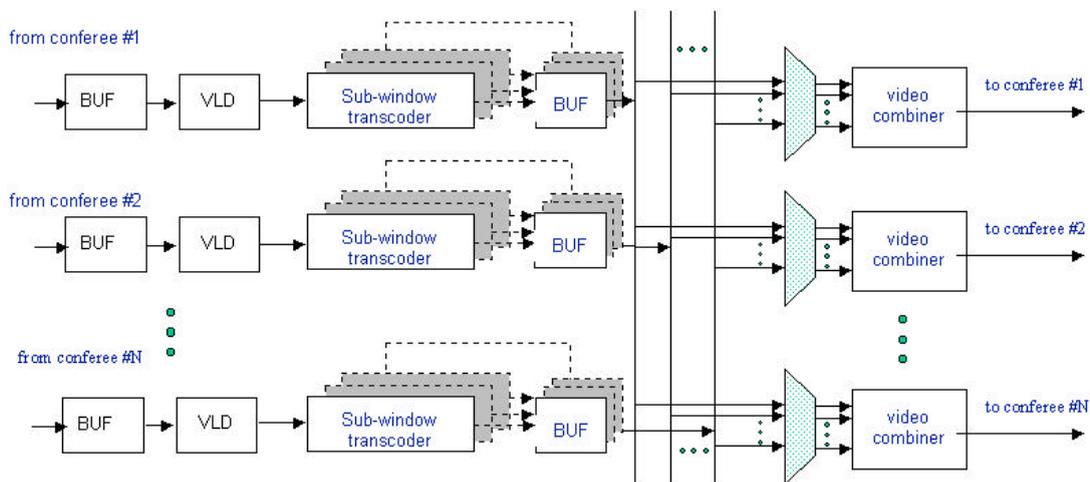


圖 18 型式 II 多點視訊轉換編碼架構

圖 18 所示之型式 II 則改善了此一缺點。此架構先對每一個輸入之視訊位元流進行部分解碼後，先對每一個個別的視訊子畫面進行轉換編碼後再選擇所要的子畫面組合成全畫面。若有多個與會者共享一子畫面的情形時，此種架構和型式 I

相較，具有較低的運算複雜度。但此架構亦有其缺點，當共享同一子換面的與會者具有不同的頻寬需求時，可能仍需使用多個轉換編碼器以產生不同位元率的版本。為了解決此一問題，如果所使用的編碼標準支援可伸縮性(scalable)的特性(例如 MPEG-4, H.263+)，則可使用單一的轉換編碼器產生具備多重速率的輸出，再針對每一與會者之需求，選擇適當的輸出畫面加以組合。圖 19 顯示此一架構。利用此架構可獲得最佳之效率。

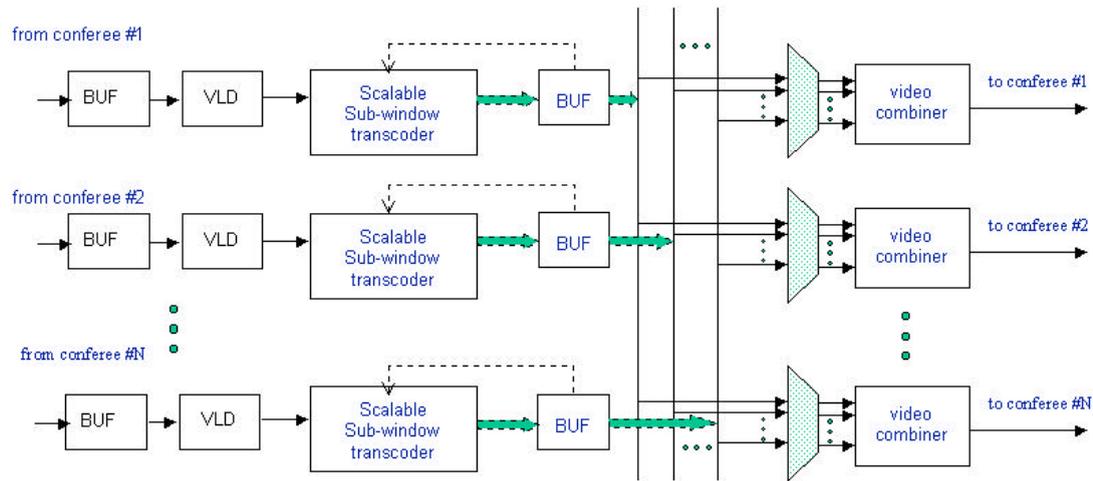


圖 19 型式 III 多點視訊轉換編碼架構

在多點視訊會議中，每一個時間點通常僅有 1~2 位與會者會有較大的動作（通常是發言者），其他的人則相對地呈現較小的動作。由於具有較大動作的子畫面需要較多的位元量以維持其畫面品質，而相對的，動作小的子畫面所需的資料量遠低於前者。利用此種畫面的相關性，可以利用動態位元分配的策略，以達到較佳的畫面品質。

圖 20 顯示了型式 II 之轉換編碼器加上動態位元分配後之架構。轉換編碼器可利用接收運動向量資訊判斷每個畫面的運動量大小，並據此分配位元量。圖 21 顯示了使用此種動態位元分配後的效能比較。很明顯地，右下角動作較大的子畫面可以達到相當明顯的畫面品質改善，而其他畫面在減少位元量後之品質降低事實上難以察覺。圖 22 顯示了一個實際多點視訊會議系統的使用者介面範例，下方的視窗為本地畫面，而上方視窗為經過轉換編碼後之遠方畫面。

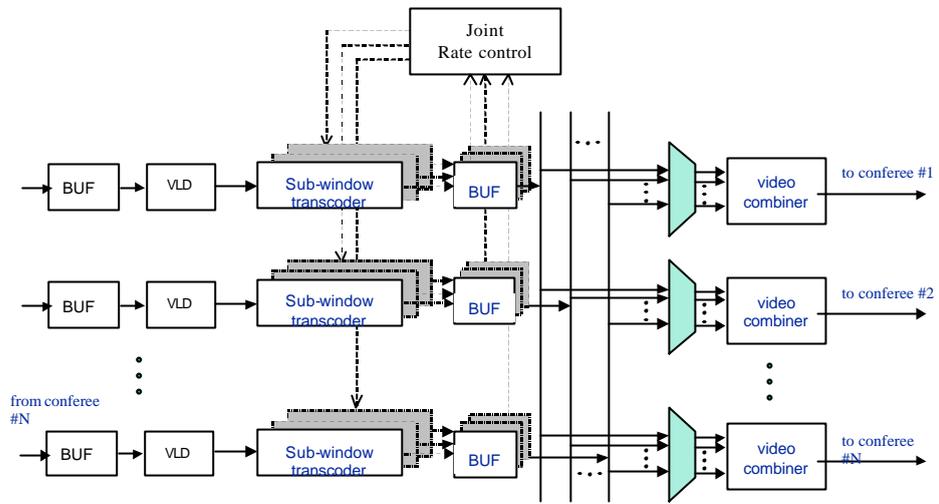


圖 20 具動態位元分配功能之視訊轉換編碼架構



(a)

(b)

圖 21 畫面品質比較：(a)未使用動態位元分配；(b)使用動態位元分配



圖 22 一個實際多點視訊會議系統的使用者介面

結論

本文中介紹了各種常用的單點及多點視訊轉換編碼器架構，並分析比較了其效能及優缺點。在決定使用何種架構前需針對其用途、成本及所需畫面品質要求做一最佳取捨。

在多點視訊中利用視訊畫面的鄰關係，可以利用動態位元分配技術，在很小的運算需求下，即可大幅提昇畫面品質。

參考文獻

- [1] Tzong-Der Wu, Jenq-Neng Hwang and Ming-Ting Sun, "Video Transcoding," invited book chapter for Multimedia Image and Video Processing, L. Guan, S.Y. Kung, and J. larsen ed., CRC press.

- [2] Ming-Ting Sun and I-Ming Pao, "Multipoint Video Conferencing," Visual Communication and Image Processing, Marcel Dekker, C.W. Chen and Y.Q. Zhang ed., 1997.
- [3] ITU-T Recommendation H.323, "Visual Telephone Systems and Terminal Equipment for Local Area Networks Which Provide A Non-Quaranteed Quality of Service".
- [4] ITU-T Recommendation H.324, "Terminal for Low Bit Rate Multimedia Communication".
- [5] ITU-T Recommendation H.261, "Video Codecs for Audiovisual Services at $p \times 64$ kbits". March 1993.
- [6] ITU-T Recommendation H.263, "Video Coding For Low Bit Rate Communication". May 1997.
- [7] ISO/IEC JTC1/SC29/WG11 "Coding of Moving Pictures and Associated Audio MPEG98/W2194". (MPEG-4), March 1998.
- [8] J. Youn, M.T. Sun, and J. Xin, "Video transcoder architectures for bit-rate scaling of H.263 bit streams," *Proc. ACM Multimedia*, Orlando, FL, Nov. 1999.
- [9] A. Eleftheriadis and D. Anastassiou, "Constrained and General Dynamic Rate Shaping of Compressed Digital Video," ICIP '95, 1995.
- [10] H. Sun, W. Kwok and J.W. Zdepski, "Architecture for MPEG Compressed Bitstream Scaling", *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, No. 2, April 1996.
- [11] G. Keesman et al., "Transcoding of MPEG Bitstream," *Signal Proc.. Image Commun.*, pp. 481-500, 1996.
- [12] S.-F. Chang and D. G. Messerschmitt, "Manipulation and Compositing of MC-DCT Compressed Video," *IEEE Journal Selected Areas in Commun.*, vol. 13, pp. 1-11, Jan. 1995.
- [13] M. Merhav and V. Bhaskaran, "Fast Algorithms for DCT-Domain Image Down-Sampling and for Inverse Motion Computation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, pp. 468-476, Jun. 1997.
- [14] M. Merhav and V. Bhaskaran, "Fast Inverse Motion Compensation Algorithms for MPEG-2 and for Partial DCT Information," *HP Technical Report #HPL-96-53*, 1996.

- [15] P. A. A. Assuncao and M. Ghanbari, "Optimal Transcoding of Compressed Video," in *IEEE International Conference on Image Processing*, Chicago, Oct. 1997.
- [16] P. A. A. Assuncao and M. Ghanbari, "A Frequency Domain Video Transcoder for Dynamic Bit Rate Reduction of MPEG-2 Bit Streams," *IEEE Trans. Circuits Syst. Video Technol.*, Vol. 8, No. 8, pp. 953-567, Dec. 1998.
- [17] Y. Arai, T. Agui, and M. Nakajima, "A Fast DCT-SQ Scheme for Images," *Trans. of the IEICE*, E 71(11): 1095, Nov. 1988.
- [18] B.-L. Yeo, "Efficient Processing of Compressed Images and Video," *Ph.D. Dissertation*, Princeton University, USA, Jan 1996.
- [19] J. Song and B.-L. Yeo, "A Fast DCT Domain Inverse Motion Compensation Algorithm Based on Shard Information in a Macroblock," *IBM Research Report RC 10880*, June 1997.
- [20] O. H. Werner, "Generic Quantizer for Transcoding of Hybrid Video," in *Picture Coding Symposium*, Berlin, Sep. 1997
- [21] O. H. Werner, "Real-Time Transcoding of MPEG-2 Video Bit Stream," in *International Broadcast Convention*, Amsterdam, pp. 206-301, Sep. 1997.