行政院國家科學委員會專題研究計畫　成果報告

子計劃四:無線網路串流聲訊研究及聲視訊子系統整合(I)

計畫類別： 整合型計畫
計畫編號： NSC91-2219-E-009-011-
執行期間： 91 年 08 月 01 日至 92 年 07 月 31 日
執行單位： 國立交通大學電子工程學系

計畫主持人： 杭學鳴

計畫參與人員： 楊政翰，陳繼大，蔡家揚，侯思瑋，李仰哲，王俊能

報告類型： 完整報告
報告附件： 出席國際會議研究心得報告及發表論文
處理方式： 本計畫可公開查詢

中　華　民　國 92 年 10 月 23 日

行政院國家科學委員會補助專題研究計畫 ■ 成 果 報 告
□期中進度報告

無線網路串流聲訊研究及聲視訊子系統整合 (1/3)
Wireless Streaming Audio Research and Audio/Video Subsystem Integration (1/3)

計畫類別：□ 個別型計畫 ■ 整合型計畫
計畫編號： NSC 91-2219-E009-011
執行期間： 91 年 8 月 1 日至 92 年 7 月 31 日

計畫主持人：杭學鳴
計畫參與人員：楊政翰，陳繼大，蔡家揚，侯思瑋，李仰哲，王俊能

成果報告類型(依經費核定清單規定繳交)：□精簡報告 ■完整報告

本成果報告包括以下應繳交之附件：
□赴國外出差或研習心得報告一份
□赴大陸地區出差或研習心得報告一份
■出席國際學術會議心得報告及發表之論文各一份
□國際合作研究計畫國外研究報告書一份

處理方式：除產學合作研究計畫、提升產業技術及人才培育研究計畫、列管計畫及下列情形者外，得立即公開查詢
□涉及專利或其他智慧財產權，□一年□二年後可公開查詢

執行單位：國立交通大學電子工程學系

中 華 民 國 92 年 10 月 15 日

行政院國家科學委員會專題研究計畫成果報告

無線網路串流聲訊研究及聲視訊子系統整合（1/3）

Wireless Streaming Audio Research and Audio/Video Subsystem Integration (1/3)

計畫編號: NSC 91-2219-E009-011
執行期限: 91 年 8 月 1 日至 92 年 7 月 31 日

主持人: 杭學鳴　國立交通大學電子工程學系教授
計畫參與人員：楊政翰，陳繼大，蔡家揚，侯思瑋，李仰哲，王俊能
國立交通大學電子研究所

中文摘要

本子計畫之主要目標在研究與製作寬頻無線網路環境中的串流聲訊系統。主要完成目標為聲訊(Audio & Speech)在寬頻無線網路中的有效傳輸方式，含可調式(scalable)編碼法，整合通道與訊源編碼，錯誤糾正與補償等演算法之研究。其次考慮通道錯誤偵測與錯誤控制技術以及系統層對達成品質保證(QoS)目標的影響。並配合群體計畫的 DSP 展示系統，則同時進行 MPEG-4 Audio Coding (AAC 或其延伸)與 3GPP AMR Speech coding 的 DSP 實現。第三年亦將協助總計畫，在本計畫內先將視訊與聲訊 DSP 整合成訊源子系統。

關鍵詞：AMR，AAC，串流聲訊

Abstract

The goal of this research project is to study, simulate and design effective streaming audio algorithms/systems transmitted in the wideband wireless environment. This is often achieved by three means: sclable source coding schemes, adaptive channel coding techniques and QoS protocol support from the network. We will focus mostly on the former two methods.

In the first year, we study and design the MPEG AAC (advanced audio coding) audio codec and BSAC (Bit-Sliced Arithmetic Coding) scalable coding techniques. In the future, we work on the DSP implementation of AAC and 3GPP AMR (adaptive multi-rate) coding. In the third year, we will integrate the audio and video components together with the system layer to form the source subsystem to be merged into the Group Project test bed.

關鍵詞：AMR，AAC，streaming audio

目錄 Table of Contents

1. C.-H. Yang and H.-M. Hang, “Efficient bit assignment strategy for perceptual audio coding,” ICASSP 2003, Hong Kong, April 2003.
2. C.-H. Yang and H.-M. Hang, “Cascaded trellis-based optimization for MPEG-4 Advanced Audio Coding,” to be presented in Audio Engineering Society Convention 2003, New York, Oct. 2003.
3. 出席國際學術會議報告 IEEE ISCAS 2003

A. 背景

前一期執行電信國家型計畫時，已經實踐了一些 Speech Coding 與 Convolutional Coding 演算法，故在本期第一年中我們先探討 AAC Audio Coding 編碼法與可調式(scalable)切片式算數編碼(Bit-Sliced Arithmetic Coding，BSAC)。

AAC 最先是 MPEG-2 中的聲訊編碼標準，在 1997 年 12 月制定完成。MPEG-2 AAC 聲訊編碼標準捨棄與 MPEG-1 聲訊編碼標準的相容性，加入了 Temporal Noise Shaping (TNS) 及預測 (Prediction) 這兩個獨立的新模組，因此 AAC 能提供比 MP3 更好的壓縮率及聲訊品質[1][2][3]。MPEG-4 AAC (version 2)是 ISO/IEC MPEG 於 1999 年 12 月制定完成之新一代聲訊編碼標準[4][5]。MPEG-4 AAC 聲訊編碼是以 MPEG-2 AAC 為基礎，並加入了數個獨立的新模組，Long Term Prediction (LTP) [6]、Perceptual Noise Substitution (PNS) [7]、Transform-Domain Weighted Interleave Vector Quantization (Twin-VQ) [8]等。這些新的模組將有助於更低位元率的聲訊壓縮。除此之外 MPEG-4 AAC 更引進“可調整之彈性（scalability）”的概念，也就是利用 Bit-Sliced Arithmetic Coding (BSAC)這個新模組，讓編碼端可以依情況 (傳輸通道之頻寬等) 來對壓縮位元率及聲訊頻寬作調整，並調整編碼層次。解碼端則可以依情況 (接收到的有效資料串多寡) 來調整解碼的層次，進而得到不同聲訊品質。舉例來說，若編碼端以 128 kbps 進行聲音壓縮，解碼端可以隨著通道狀況的不同，用 32kbps、64kbps、96kbps 及 128kbps 進行解碼[3]。因此 MPEG-4 AAC 除了可提供更高壓縮率、更好品質外，亦更適合網路或無線通道的傳輸。我們擬選取適當的演算法加以 DSP 實現。

B. 研究步驟

1. AAC General Coding 編碼法之研究：

多媒體壓縮標準如聲訊壓縮，只界定解碼端。編碼端許多參數選擇，是設計工程師的職責。好的參數選擇產生較好的效果。圖 1 是 MPEG-4 AAC 聲訊編碼標準的整體架構。在 MPEG-4 AAC 聲訊編碼中，參數選擇中最重要的一類是控制位元率，以達到較佳聲訊品質與較高壓縮比，也就是圖.1 中 Rate/Distortion (R/D) Control 的部分。經由前置處理所得到的頻線係數(Spectral Coefficients)先經過量化器(Quantizer)量化，量化過的頻譜係數再利用霍夫曼(Huffman)編碼。參數 “比例因子(Scale Factor, SF)”控制著量化器的步階(Step Size)，因此也決定著量化誤差(noise-to-masking ratio, NMR) 。

另外 MPEG-4 AAC 提供了 12 組“霍夫曼編碼書(Huffman Code Book, HCB)”以供編碼。Rate/Distortion Control 就是藉由選擇不同的 SF 和 HCB 數值來控制編碼位元率以及聲訊品質。SF 和 HCB 這兩項參數最後也需要被編碼並傳送到解碼端。在 Rate/Distortion Control 演算法上，我們嘗試兩種想法，分別發表兩篇論文，這兩種想法有部分亦可結合在一起。

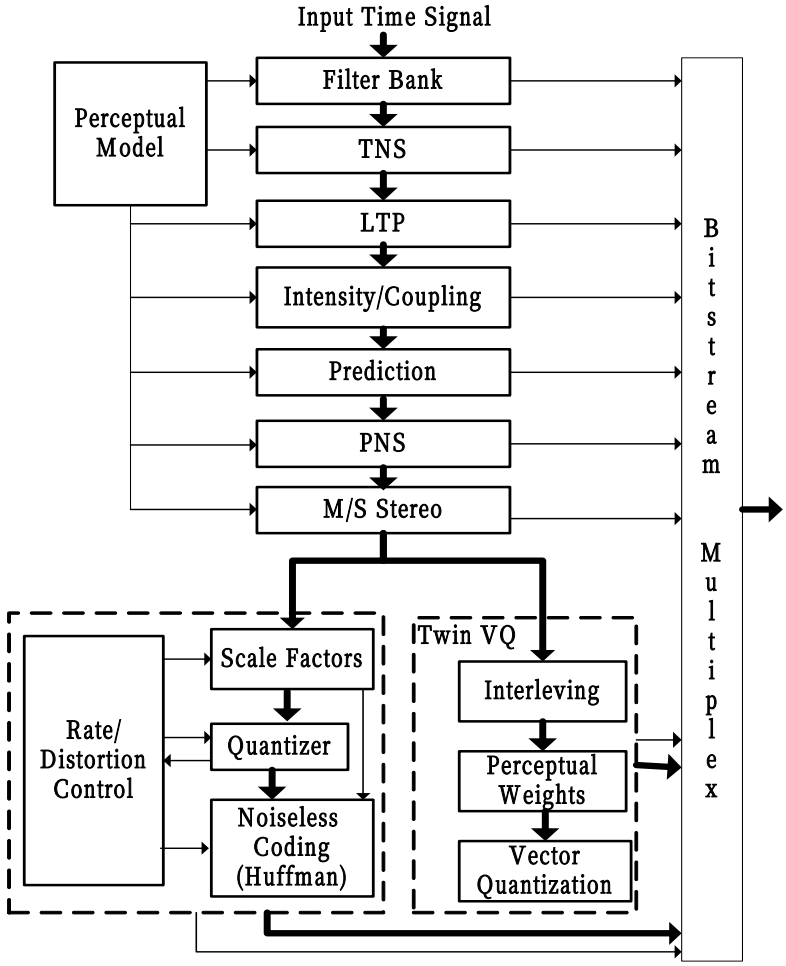


圖 1. MPEG-4 AAC 聲訊編碼整體架構

1.1 有效位元分配方式:

在聲訊壓縮中傳統位元分配方法是，觀察各頻帶(Band) 的 NMR 值（誤差值），將位元優先配給 NMR 值最大者。但如從整體效率角度，這並不是成效最好的方法，例如下例：

Band	NMR (dB)	NMR-Gain/bit
A	3.5	0.5
B	3	1.5

如上表，Band A 的 NMR 值較大，但給 Band A 1 bit 只降低 0.5 dB。若給 Band B 1 bit 則可降低 1.5 dB。如果聽覺上，不論哪一個 Band，降低 1.5 dB 都會比降低 0.5 dB 好，則選 Band B 較有效。

不同於傳統的位元分配方法，我們將“位元使用效率”這個概念引入，並根據此概念提出下方之新的位元分配之原則。新的位元分配原則可以比傳統方法更有效的控制位元使用效率。

“Give bits to the band with the maximum NMR-Gain/bit” *或*

“Retrieve bits from the band with the maximum bits/NMR-Loss”.

根據我們所提出之新的位元分配原則，我們針對 MPEG-4 AAC 聲訊編碼設計一套新的位元分配方法，Max Bits/NMR-Loss (BNL)位元分配方法。

Max Bits/NMR-Loss 位元分配方法大至可分為以下四個步驟:

1. 前置處理(Pre-Processing)：主要是用來初始化一些在接下來步驟中所需要用的參數，如: 參考位元($bits_{ref}$)，參考 NMR(NMR_{ref})等…。

2. Bits/NMR-Loss 分析：對於各 band，藉著調整 SF 的數值，我們可以得到一組新位元($bits_{new}$)及新 NMR(NMR_{new})。經過 Bits/NMR-Loss 的分析後，對於各 band，我們就可以找出最大 Bits/NMR-Loss 數值和相對應的最佳 SF 數值。

   $$\text{Bits/NMR-Loss} = (bits_{ref} - bits_{new}) / (NMR_{new} - NMR_{ref})$$

3. 分析所有 band 的 Bits/NMR-Loss，並選出擁有最大 Bits/NMR-Loss 數值的 band，然後將此 band 的 SF 數值調整到最佳數值。

4. 計算新的總編碼位元，如果大於限定之編碼位元則更新所有參數(如: $bits_{ref}$、NMR_{ref} 等)並回到步驟(2)。

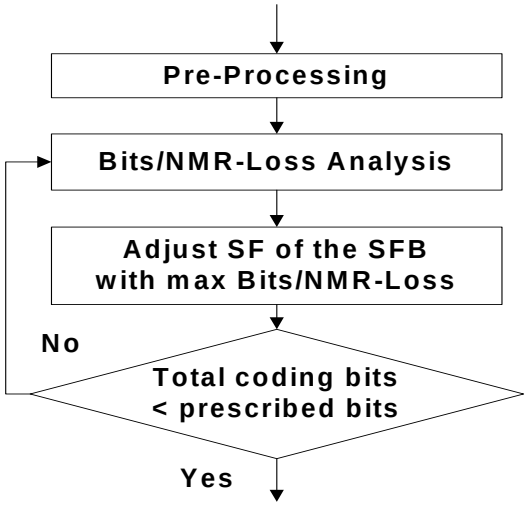
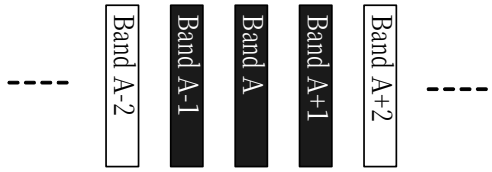


圖 2. Max Bits/NMR-Loss 位元分配方法

步驟(2)是 Max Bits/NMR-Loss 位元分配方法中計算量最大的部分，如下例：MPEG-4 AAC 聲訊編碼有 49 個 band，如果各 band 的候選 SF 數目為 10，那總共要執行 Bits/NMR-Loss 計算 490 次。為了減少新位元分配方法的計算量，我們同時提出了一個步驟(2)的快速演算法。經過統計上的分析，我們發現，除了少數特定的 band 之外，其餘 band 的 “最佳 SF” 和 “最大 Bits/NMR-Loss”數值在前後 iteration 的相似性極高。因此對於這類 band，我們可以省去它的大部分 Bits/NMR-Loss 計算。以下圖說明，如果我們在這個 iteration 調整了 Band A 的 SF 數值，那麼在下個 iteration 我們只需要從新分析 Band A 及與其相鄰的 2 個 band (Band A-1、A+1)的 Bits/NMR-Loss。其餘 band 的“最佳 SF”等參數可以延續到下個 iteration 使用，如此一來便可以大幅的減少步驟(2)的計算量。



1.2 快速位元分配參數搜尋法:

SF 和 HCB 這兩項參數的組合控制著編碼位元率以及聲訊品質。想要得到最佳的效果，最直接的方法就是比較 SF 和 HCB 這兩項參數所有可能的組合並選出最佳的組合，就也就是所謂的窮舉搜尋法。但由於窮舉搜尋法的計算量非常大，不適合於實際應用。前人觀察到編碼參數前後間之重複關係，提出利用 trellis-based search 的方法來降低計算量。由於在前人所提出之方法中，SF 和 HCB 這兩項參數是利用 trellis-based search 的方法同時決定，所以其演算法又稱為 *joint trellis-based* (JTB)，而其效果可以逼近窮舉搜尋法[9][10]。

雖然 JTB 演算法的計算量已遠較窮舉搜尋法為低，但是仍然過高，因此我們提出一個計算量更低之快速演算法 ***cascaded trellis-based***(CTB)。CTB 演算法同樣是利用 trellis-based search 的方法來決定參數，但是它和 JTB 演算法最大的不同點在於 SF 和 HCB 兩項參數是分在兩個不同步驟決定的。下圖 4 為 CTB 演算法的架構，大致可分為四個步驟。

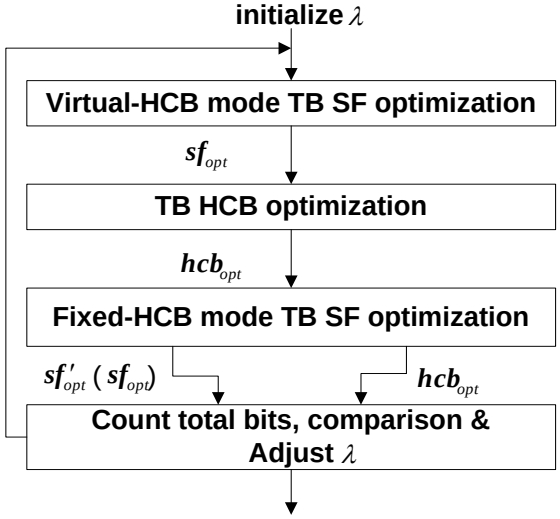
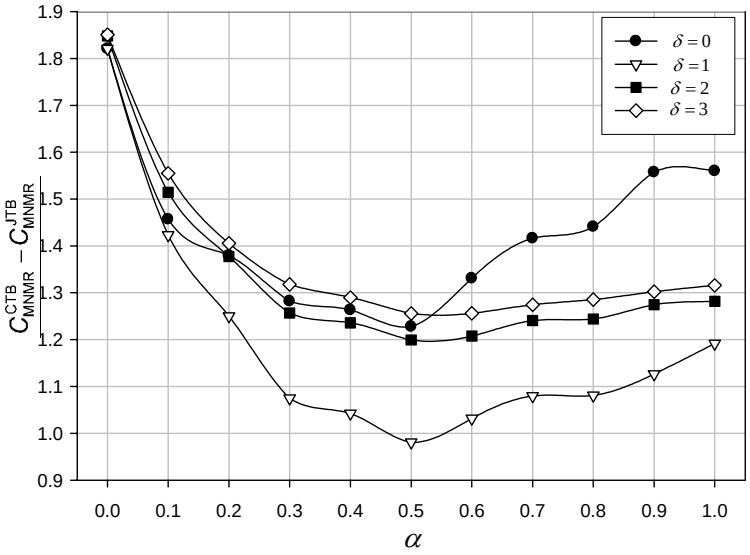


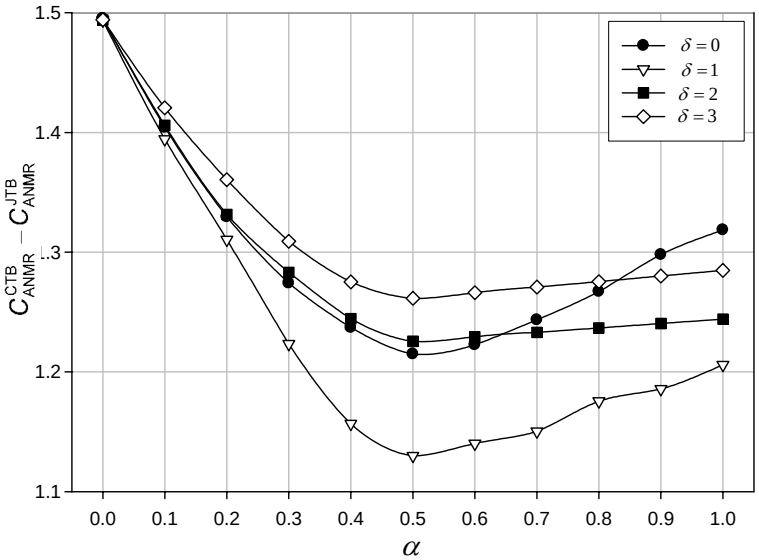
圖 4. CTB 演算法架構

1. 在沒有可參考之 HCB 的條件下，利用 trellis-based search 的方法決定一組最佳的 SF，$\boldsymbol{sf}_{opt}$。
2. 在給定一組 SF，$\boldsymbol{sf}_{opt}$，的條件下(來自步驟(1))，利用 trellis-based search 的方法決定一組最佳的 HCB，$\boldsymbol{hcb}_{opt}$。
3. 在給定一組最佳的 HCB，$\boldsymbol{hcb}_{opt}$ 的條件下(來自步驟(2))，利用 trellis-based search 的方法再次決定一組新的最佳 SF，$\boldsymbol{sf}'_{opt}$。此步驟是為了修正一些在步驟 1 中因為沒有可參考之 HCB 時對於 SF 做的偏差決定。
4. 計算在使用 $\boldsymbol{sf}'_{opt}$ 和 $\boldsymbol{hcb}_{opt}$ 這組參數來編碼頻線係數時所需要的總位元率，並與一限定位元率比較，視情況再做適當的調整。

上述流程是 CTB 演算法的完整模式，如果要再進一步減化計算量，我可以省去上述流程中的步驟(3)，這樣便是 CTB 演算法的簡化模式。分開決定 SF 和 HCB 兩項參數可以大量減少計算量，但也可能大幅降低品質，其關鍵點在於找出適當的虛擬霍夫曼編碼書模型(virtual HCB model)，而此模型主要是用於 CTB 演算法中的步驟(1)。在 JTB 演算法中，因為 SF 和 HCB 兩參數是一起決定，所以對於任一候選 SF，有 12 組 HCB 與之搭配形成 12 組延申候選組合。然而在 CTB 演算法中，對於任一候選 SF，只有一組 HCB 與之搭配。步驟(3)中用來搭配之 HCB 是由步驟(2)求得之真實 HCB。然而在步驟(1)中，我們則需要利用 virtual HCB model 來決定一組用來搭配的 virtual HCB，而這個 virtual HCB 適當與否將影響整個 CTB 演算法的效能。為了建立適當的 virtual HCB model，我們從統計資料中找出篩減候選 HCB 數目的規則性，並利用它來找出 virtual HCB model 中兩個重要的變數，編碼位元偏移(δ)和虛擬霍夫曼編碼書比重(α)。下圖 5 顯示 CTB 演算法和 JTB 演算法之間的誤差隨著不同的 δ 和 α 數值而改變，而越小的誤差值表示越好。



(a)



(b)

圖 5. CTB 演算法和 JTB 演算法之誤差值 v.s (δ,α)

2. 切片式算數編碼(Bit-Sliced Arithmetic Coding，BSAC)研究：

在MPEG-4 第一版的比率編碼方法中,只粗略的提供了幾個特定的位元率進行編碼 (例如，24 kbps 編碼率的基礎層，另外再加上一至二個 16 kbps 編碼率的增進層)，仍舊存在許多尚待改進的空間。因此，配合 Fine Granularity Scalability(FGS)的概念，MPEG-4 在第二版的聲訊壓縮標準中提供了編碼率精細可調式的新工具 Bit-Sliced Arithmetic Coding(BSAC)[11]。每個可調間距大約為 1 kbits/s/ch。這功能對一些頻寬容易變動的通訊系統，例如網際網路或行動通訊來說，是非常有用的。

我們首先研究切片式算數編碼的音質效能及其對於傳輸錯誤的敏感度。接著，我們提出兩種方法試圖改善切片式算數編碼的編碼效率。

由於 BSAC 和傳統 AAC 的編碼架構大致相同，只有在最後的部分用 BSAC 方法取代原先用在頻線係數和 SF 上的無失真編碼，因此我們比較了 BSAC 和傳統 AAC 的音質效能之後，並對實驗結果進行分析，並提出造成兩者效能差異的可能原因。因為算數編碼是一種對傳輸錯誤很敏感的編碼方式，所以我們也研究了切片式算數編碼中的錯誤傳遞問題。

BSAC 編碼過程中有兩個重要的步驟，一是將頻線係數由低到高頻的分佈切割成不同可調層，另一個則是依照頻線係數的特性替算數編碼決定適合的機率模型。因此我們就從這兩方要來改善編碼效率。

MEPG-4 BSAC 中的機率模型大致是兩兩一組。每次可以從兩組機率模型中選出一組用來做算數編碼。但是由於 MEPG-4 BSAC Verification Model 中都只固定使用其中一組，因此我們將選擇機率模型的機制打開，並嘗試兩種不同的選擇機率模型的方式，(1)在 R/D loop 之外選擇機率模型，(2) 在 R/D loop 之內選擇機率模型。此外，我們也設計並測試經由實際聲音訊號所產生的機率模型來取代 MEPG-4 BSAC 原有的機率模型。

另一個改善編碼效能的方法是改變可用位元的分配及可調層的切割。可調層的切割的改變如下表。基礎層(Base layer, BL)的 coefficient 數目不做更動，而把原來增進層(enhancement layer, EL)分割規則由(12、12、8)改為(16、16)。而在分析不同可調層位元使用情況(下圖 6) 。我們可以發現位元使用多集中在低頻可調層部分，較高頻可調層(35 以上)幾乎不需要位元。因此我們在可用位元的分配上嘗試多種調整的方式，大致可以分為兩大方向，(1)增加基礎層的位元分配，(2)增加每個增進層位元分配。因為位元分配的順序是由低頻可調層到高頻可調層，而在總位元率不變的條件下，這兩種方式的主要觀念都是分配更多的位元數給較低頻的可調層。以下表為例，在調整過可調層的切割方式後為了和原始的 BSAC 達到相似的效果，增進層位元分配由原來的(1、1、1)調整為(1.5、1.5)。

Original	BL	EL1	EL2	EL 3
# coeff.	160	12	12	8
位元分配 kbps/ch	16	1	1	1

Modified	BL	EL1	EL2
# coeff.	160	16	16
位元分配 kbps/ch	16	1.5	1.5

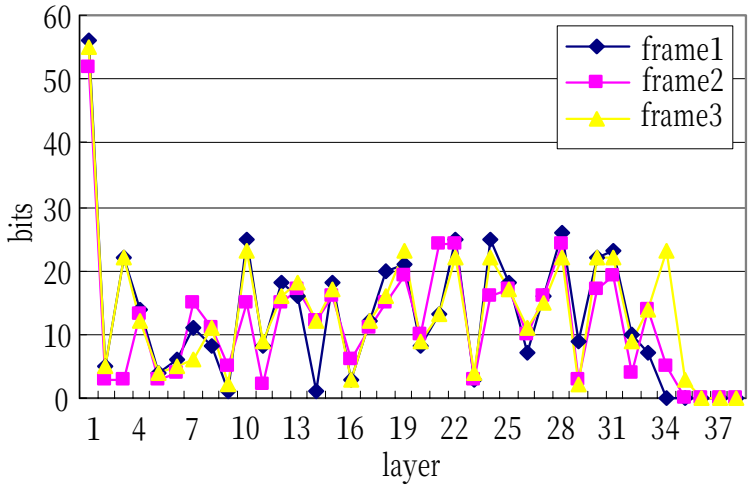


圖 6. 不同可調層位元使用情況

C. 實驗與結果

1. AAC 編碼法 Rate/Distortion Control：

這裡將列出我們所提出的兩種 Rate /Distortion Control 演算法的複雜度分析和效能的實驗結果。比較的對象一是 MPEG-4 AAC Verification Model (VM-TLS)，另一則是 JTB 演算法。又由於最佳化的條件不同，JTB 演算法又可以細分為 1. 針對 Average NMR 條件做最佳化的 JTB-ANMR 和 2. 針對 Maximum NMR 條件做最佳化的 JTB-MNMR 兩種。

表 1. Max-BNL 演算法之複雜度分析

Algorithm	Complexity Ratio	Storage Ratio
JTB	1	1
Max-BNL	1/17	1/120
Fast Max-BNL	1/150	1/120

由圖 7 中兩種不同的客觀品質量測結果證明，我們所提出的(Fast) Max-BNL 位元分配方法比 MPEG-4 AAC Verification Model 要好上約 3dB，而與 JTB-ANMR 演算法的效果類似。同時表 1 列出了以 JTB 演算法為基準的實際複雜度比較，我們可以看到 Fast Max-BNL 演算法的計算量只有 JTB 演算法的 1/150。另外，運算中用於 trellis-based search 上所需的記憶體數量也只有 JTB 演算法的 1/120，這對於某些只允許有限且少量記憶體的應用(如：DSP 實現)來說是較為有利的。

表 2. CTB 演算法之複雜度分析

Algorithm	Complexity Ratio	Storage Ratio
JTB	1	1
完整模式 CTB (Two-Loop)	1/71	1/12

簡化模式 CTB (One-Loop)	1/142	1/12

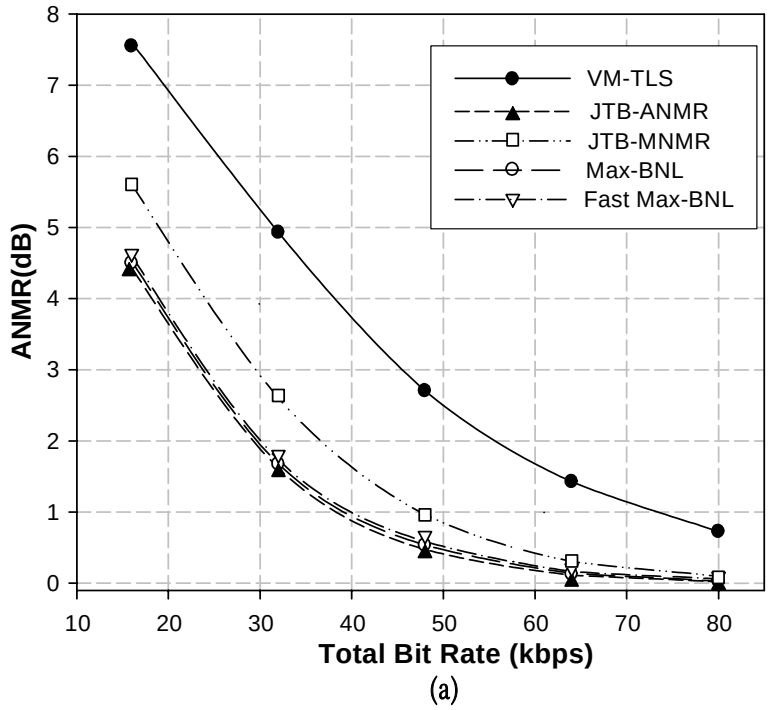
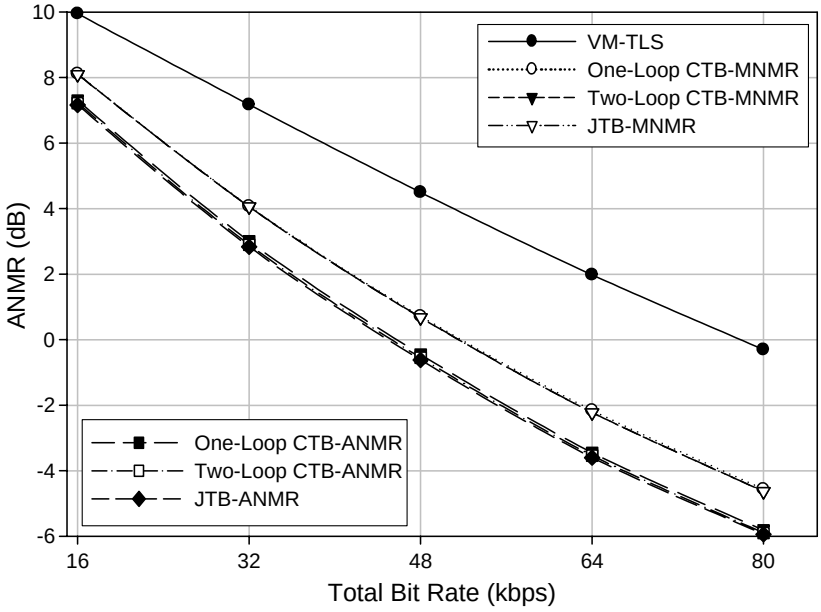
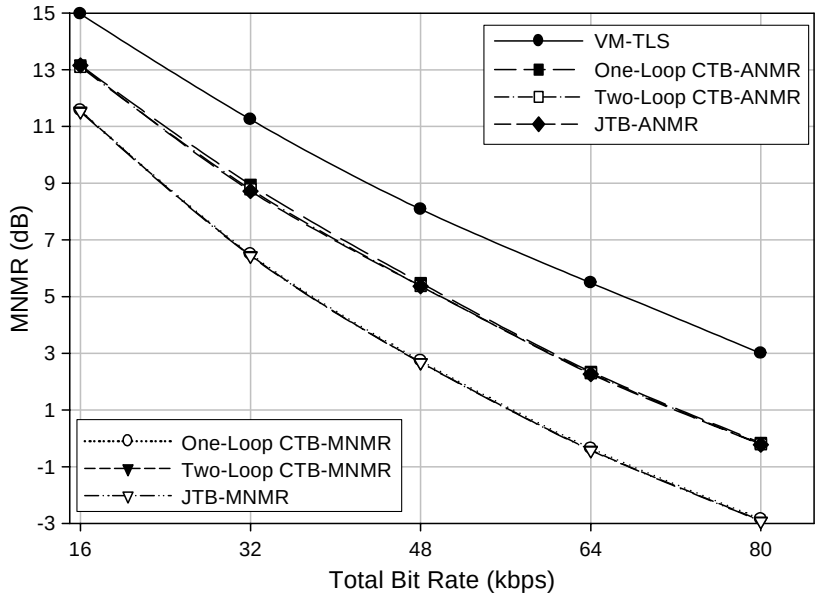


圖 7. Max-BNL 演算法之 Rate-Distortion 分析



(a)



(b)

圖 8. CTB 演算法之 Rate-Distortion 分析

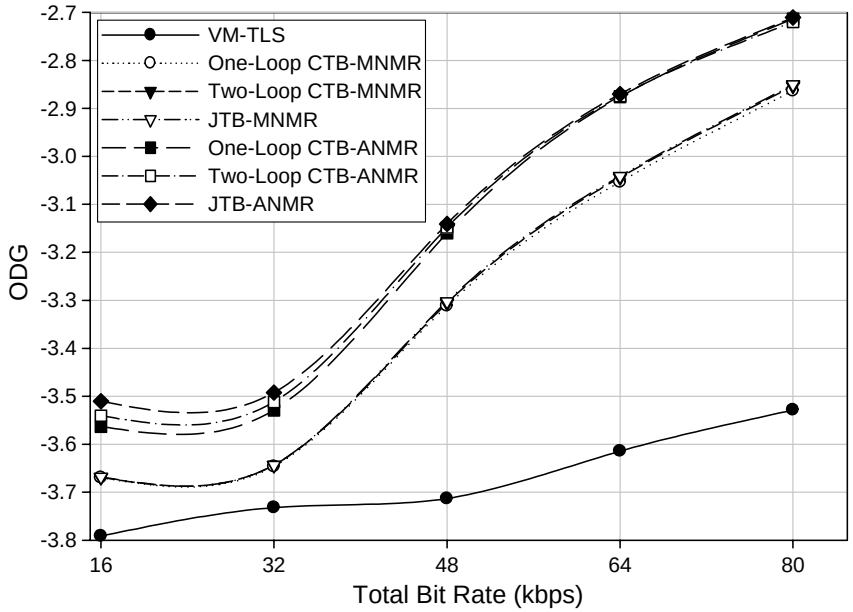
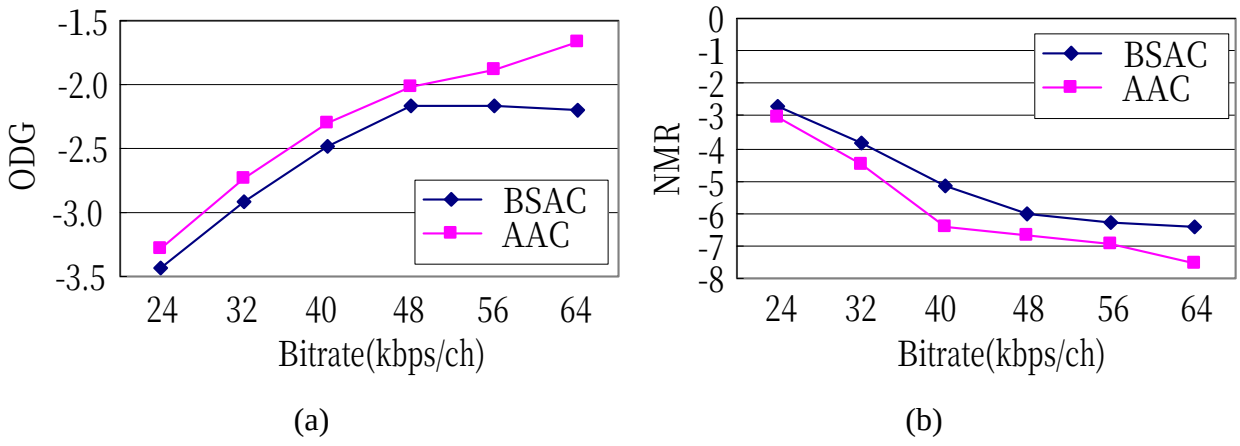


圖 9. CTB 演算法之 ODG 分析

Trellis-based search 方法的複雜度取決於 trellis 中 stage 的數目和每個 stage 包含 state 的數目。Stage 代表著 MPEG-4 AAC 中的 band，而 state 則代表著不同的候選參數。在相同的 SF 候選數目和 HCB 候選數目條件下，CTB 演算法和 JTB 演算法的複雜度比較結果列於表 2。我們可以看到完整模式 CTB 演算法的計算量只有 JTB 演算法的 1/72，而簡化模式 CTB 演算法的計算量更只有 JTB 演算法的 1/142。而運算中用於 trellis-based search 上所需的記憶體數量也只有 JTB 演算法的 1/12。圖 8 是兩種不同客觀品質量測的結果，圖 9 則是利用模擬主觀品質 Objective Difference Grade (ODG) 量測的結果。我們可以發現，不論從那種量測結果都顯示 CTB 演算法的壓縮效能和 JTB 演算法無甚差別，而且都比 MPEG-4 AAC Verification Model 要好很多。

2. 切片式算數編碼：

下圖 10 是 BSAC 和 AAC 聲訊品質的比較，我們可以明顯的看出兩者之間的差異，特別在較高的位元率時，兩者的 NMR 可差到約 1dB。為了找出造成如此差異的原因，我們分析了 BSAC 和 AAC 的位元分配情況，其如果列於表 3 中。BASC 中的 side information 大至包含 SF、stereo 和機率模型等幾個部分，而 AAC 中的 side information 包含 SF 和 HCB 等資訊。BASC 為了達到 FGS 的效果花了相對較多的位元在 side information 上，真正用於較重要的 spectral data 的位元比 AAC 少約了 6.5%，因此也損失了部分聲訊品質。

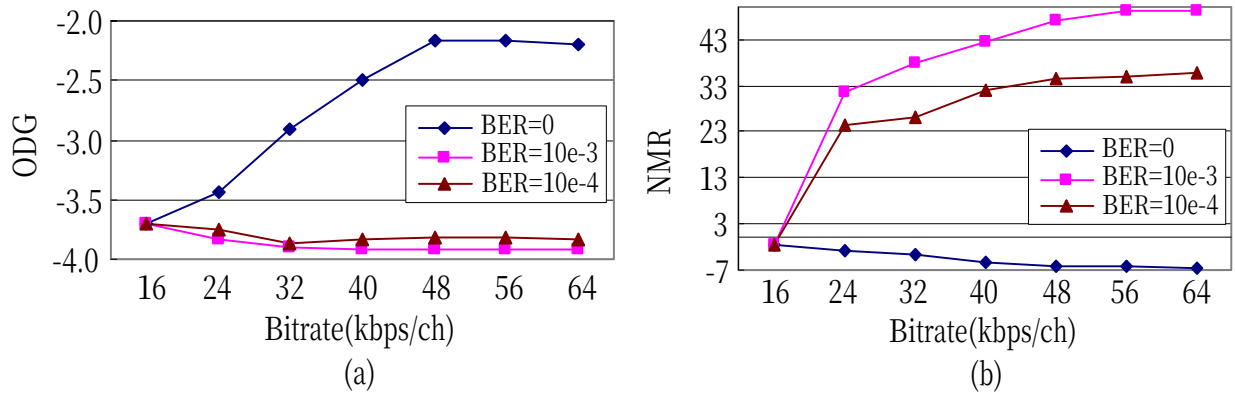


(a) (b)

圖 10. BSAC 和 AAC 效能比較

表 3. BSAC and AAC 位元分配情況分析

	BSAC	AAC
Spectral data	65.95%	72.60%
Side information	34.05%	27.40%



(a) (b)

圖 11. BSAC 中位元錯誤造成品質衰減之分析

我們由圖 11 可以發現，位元錯誤對 BSAC 的造成很大的影響，聲訊品質衰減的十分嚴重。另外由於錯誤傳遞使得聲訊品質衰減的情況在高位元率時反而比在低元率時來得嚴重。

表 4. 不同的機率模型選擇流程分析

	無機率模型選拔	R/D loop 外選擇機率模型	R/D loop 之內選擇機率模型
位元串 size	240	236	240
size 減少	—	1.66%	0%

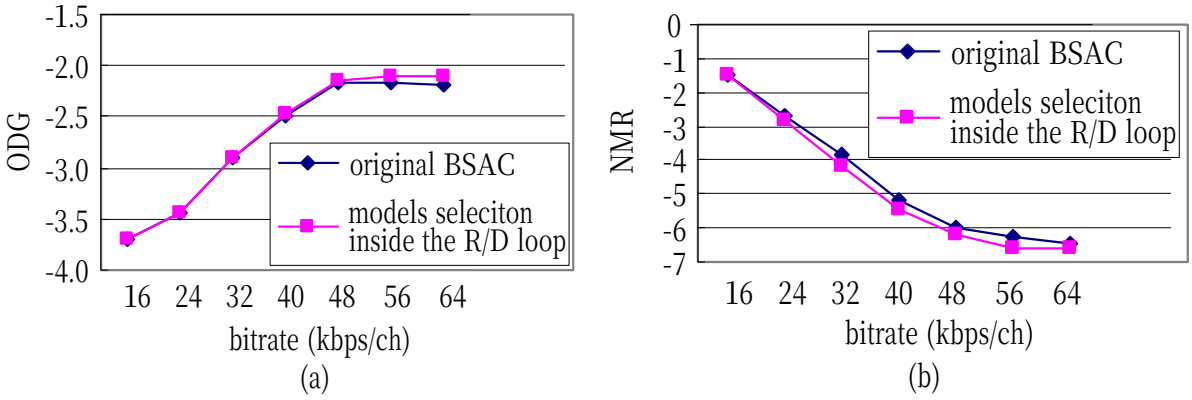


圖 12. 在 R/D loop 之內選擇機率模型之效能分析

因為在 R/D loop 外選擇機率模型只會影響壓縮後之位元串大小，所以由表 4 來看，在 R/D loop 外選擇機率模型對於減少位元串大小的改善有限。而在 R/D loop 內選擇機率模型則會影響品質，由表 4 和圖 12 來看，在 R/D loop 內選擇機率模型對於整體的聲訊品質改善也同樣很有限。

圖 13 是使用了新機率模型的效能分析。我們發現，就算利用受測訊號所產生之機率模型來作算數編碼，效能上得到的改善也不太多。

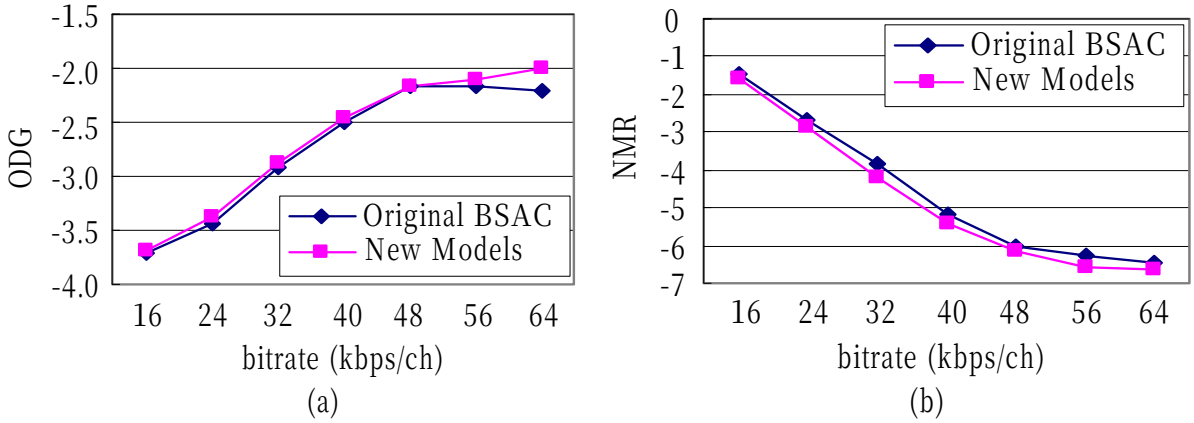


圖 13. 使用新機率模型之效能分析

圖 14 是在固定基礎層位元分配在 16kbps/ch 條件下，改變每個增進層位元分配的結果。由於總位元固定，增加每個增進層位元分配就會減少可編碼增進層的數目。由結果我們可以發現，分配較多位元給每個增進層可以得到較好的效能。圖 15 是在固定每個增進層位元分配在 2kbps/ch 條件下，改變基礎層位元分配的結果。由結果我們可以發現，分配較多位元給每個基礎層可以得到較好的效能。上面兩項結果都顯示了，對於 BSAC，分配更多的位元數給較低頻的可調層可以得到比較明顯的效能改善。

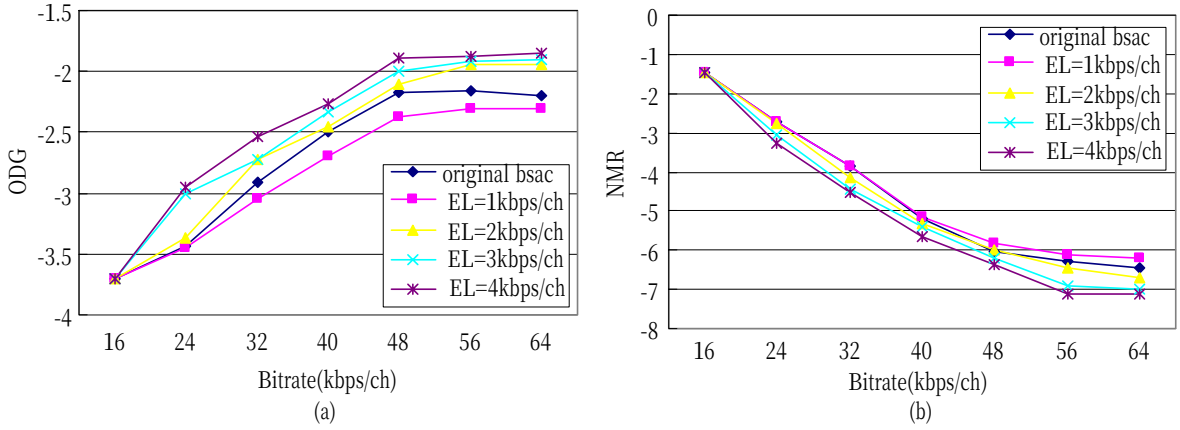


圖 14.改變可調層位元分配之效能分析

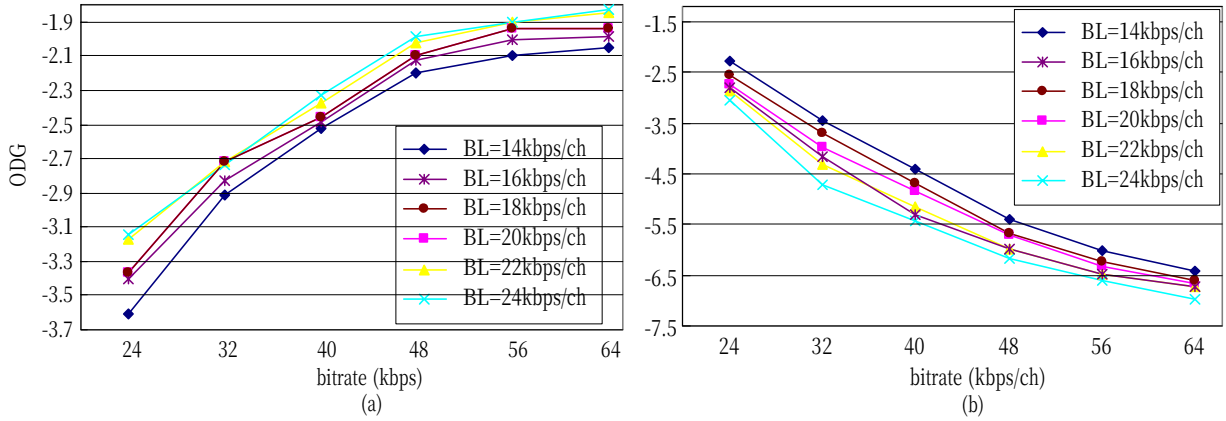


圖 15.改變基礎層位元分配之效能分析

D. 結論

在 AAC 編碼法 rate control 方面，我們提出兩個有效位元分配計算方式。第一個方法是基於我們提出之原則

"Give bits to the band with the maximum NMR-Gain/bit" 或

"Retrieve bits from the band with the maximum bits/NMR-Loss".

實驗結果此法與窮舉搜尋法達到類似效果，但計算量為百分之一。

第二個方法是使用快速位元分配參數搜尋法。因此我們將演算法切割為段落，分開計算可以大量減少運算量，但也可能大幅降低品質。其關鍵點在找出適當的位元編碼模型 (virtual Huffman code book model)。實驗數據顯示，我們可降低計算量達 trellis-based search 的百分之一，而壓縮效果無甚差別。

在 AAC 編碼法 rate control 方面，我們首先研究切片式算數編碼的音質效能及其對於傳輸錯誤的敏感度。接著，我們提出兩種方法試圖改善切片式算數編碼的編碼效率。比較切片式算數編碼和進階音訊編碼(Advanced Audio Coding, AAC)的音質效能之後，我們對實驗結

果進行分析，並提出造成兩者效能差異的可能原因。

在改善編碼效率方面，我們研究了在切片式算數編碼過程中會用到的機率模型。我們也設計並測試經由實際聲音訊號產生的機率模型。另一個改善編碼效能的方法是改變每個可調層分配到的位元數。主要觀念在於分配更多的位元數給較低頻的可調層。這個方法將可以看到比較明顯的效能改善。

E. 參考文獻

[1] ISO/IEC 13818-3: 1997, *Information technology – Generic coding of moving pictures and associated audio information – Part 3: Audio*

[2] ISO/IEC 13818-7: 1997, *Information technology – Generic coding of moving pictures and associated audio information – Part 7: Advanced Audio Coding (AAC)*

[3] M. Bosi, et al., "ISO/IEC MPEG-2 Advanced Audio Coding", *J. Audio Eng. Soc., Vol.45, No 10,* October 1997.

[4] ISO/IEC 14496-3, 1999, *Information technology –Coding of audio-visual objects – Part 3: Audio*

[5] J. Herre and B. Grill, "Overview of MPEG-4 audio and its applications in mobile communications", *5th International Conference on Signal Processing,* Vol 1, pp. 11 –20, 2000.

[6] J. Ojanpera, and M. Vaananen, "Long Term Predictor for Transform Domain Perceptual Audio Coding", *107th AES Covention*, New York 1999.

[7] B. Schulz, "Improving Audio Codecs by Noise Substitution", *J. Audio Eng. Soc., Vol.44, No. 7/8,* July/August 1996.

[8] N. Iwakami, T. Moriya, and S. Miki, "High-quality audio-coding at less than 64 kbit/s by using transform-domain weighted interleave vector quantization (TwinVQ)**",** *IEEE Int'l Conf. Acoustics, Speech, and Signal Proc.,* Vol. 5, pp. 3095 -3098, 1995.

[9] A. Aggarwal, S.L. Regunathan, and K. Rose, "Trellis-based optimization of MPEG-4 advanced audio coding," *Proc. IEEE Workshop on Speech Coding*, pp. 142-4, 2000.

[10] A. Aggarwal, S.L. Regunathan, and K. Rose, "Near-optimal selection of encoding parameters for audio coding," *Proc. of ICASSP*, vol. 5, pp. 3269-3272, Jun 2001.

[11] S. H. Park, *et al.*, "Multi-layer bit- sliced bit rate scalable MPEG-4 audio coder", *103rd Convention of the AES*, New York, Sep. 1997.

F. 計畫成果自評

無線通訊為國家重點發展的科技項目，而多媒體服務是寬頻無線網路的最重要應用。然而在無線網路上傳送串流多媒體數據有許多困難，本專題研究將承繼我們過去的經驗與前人的成果，進一步設計發展解決方式。所發展出的技術、經驗及成品極具實用價值，可促進國內工業研發技術開發。

參與工作人員(研究生與博士後)在學理上習得聲訊與語音編碼技術與國際標準。針對寬頻無線網路，設計開發可調式編碼等演算法，成員得到此課題研究與開發產品的經驗與知識。畢業後進入產業，直接有助於產業界開發新產品，提昇我國工業技術能力，達到人才培育之目的。期間研究成果論文兩篇，已發表於國際學術會議，並準備延伸成為期刊論文。

綜合評估：研究內容與原計畫大致相符, 已達成學術研究創新與人才培育之預目標。整體成效良好。研究成果頗具學術與應用價值, 已發表學術論文兩篇以及碩士學位論文一冊如下表。

Publications:

[1] C.-H. Yang and H.-M. Hang, “Efficient bit assignment strategy for perceptual audio coding,” ICASSP 2003, Hong Kong, April 2003.

[2] C.-H. Yang and H.-M. Hang, “Cascaded trellis-based optimization for MPEG-4 Advanced Audio Coding,” to be presented in Audio Engineering Society Convention 2003, New York, Oct. 2003.

[3] Szu-Wei Hou 侯思瑋, *Performance Analysis* and *Improvement on MPEG-4 Bit-Sliced Arithmetic Coding for Audio*, MS Thesis, NCTU, June 2003.

G. 附錄

1. C.-H. Yang and H.-M. Hang, “Efficient bit assignment strategy for perceptual audio coding,” ICASSP 2003, Hong Kong, April 2003.
2. C.-H. Yang and H.-M. Hang, “Cascaded trellis-based optimization for MPEG-4 Advanced Audio Coding,” to be presented in Audio Engineering Society Convention 2003, New York, Oct. 2003.
3. 出席國際學術會議報告 IEEE ISCAS 2003

報告論文（前一期成果） K.-T. Shih, C.-Y. Tsai, and H.-M. Hang, “Real-time implementation of H.263+ using TI TMS320C6201 digital signal processor”, ISCAS 2003, Bangkok, Thailand, May 2003.

<附件一>

EFFICIENT BIT ASSIGNMET STRATEGY FOR PERCEPTUAL AUDIO CODING

*Cheng-Han Yang and Hsueh-Ming Hang***

Department of Electronics Engineering
National Chiao Tung University Hsinchu, Taiwan, R.O.C.
**hmhang@cc.nctu.edu.tw; Fax: (886)-3-5723283

ABSTRACT

For the purpose of efficient audio coding at low rates, a new bit allocation strategy is proposed in this paper. The basic idea behind this approach is **"Give bits to the band with the maximum NMR-Gain/bit"** or **"Retrieve bits from the band with the maximum bits/NMR-Loss"**. The notion of "bit-use efficiency" is suggested and it can be employed to construct a bit assignment algorithm operated at band-level as compared to the traditional frame-level bit assignment methods. Based on this strategy a new bit assignment scheme, called Max-BNLR, is designed for the MPEG-4 AAC. Simulation results show that the performance of the Max-BNLR scheme is significantly better than that of the MPEG-4 AAC Verification Model (VM) and is close to that of TB-ANMR [3], which is the (nearly) optimal solution. Moreover, the Max-BNLR scheme has the advantages of low computational complexity comparing to TB-ANMR.

1. INTRODUCTION

Many highly efficient and high quality audio coding schemes have been developed and proposed to meet the growing demand of multimedia applications. The MPEG-4 Advanced Audio Coding (AAC) is one of the most recent audio coder specified by the ISO/IEC MPEG standards committee [1]. It is a very efficient audio compression algorithm aiming at a wide variety of applications, such as Internet, wireless, and digital broadcast arenas [2]. For the applications where the bandwidth is very limited, the low rate audio coding with good quality becomes essential.

The procedure of bit assignment is one of the most important elements in audio coding. Particularly, when bits are scare, how to make the best use of the limited bits is critical in producing the best quality audio. Up to now, the popular strategies on bit assignment are as follows ([2][3][5]).

1. "Give bits to the band which has the largest value of NMR (perceptual distortion)."
2. "Give bits to the bands of which the distortion is larger than the masking threshold".

In these strategies, the bit-use (giving away bits) is considered at frame-level and only the value of distortion is taken into consideration at band-level. Hence, it is hard to control the bit-use efficiency (the NMR improvement due to adding one bit) at band level and thus results in a less efficient compression scheme.

In this paper, we suggest the notion of *bit-use efficiency* and propose a new strategy to improve the bit-use efficiency, which can be evaluated at band-level. Moreover, a new bit assignment scheme based on this new strategy is proposed for MPEG-4 AAC.

The organization of the paper is as follows. Section 2 describes the aforementioned new strategy. A new AAC bit assignment scheme is delineated in section 3. Finally, the complexity analysis and the simulation results are presented in section 4.

2. EFFICIENT BIT-USE STRATEGY

How to make use of the bits more efficiently is always the key issue in audio coding. The traditional strategies, "Giving bits to the band with the largest NMR" or "Giving bits to the bands of which the distortion is larger than masking threshold", do not necessarily provide the best bit-use efficiency. For example, there are two candidate bands, A and B, and their NMR characteristics are listed in the table below. Which band should the first available bit be assigned to? In this table, NMR-Gain/bit means the gain in NMR by allocating one bit to this particular band. A more precise definition of NMR-Gain/bit will be given in section 3.

Band	NMR (dB)	NMR-Gain/bit
A	3.5	0.5
B	3	1.5

Following the traditional strategy, we would assign this one bit to band A; however, considering the bit-use efficiency, this one bit should be assigned to band B so that the overall NMR reduction is maximized. The essence of this new strategy can be summarized by the following statements.

"Give bits to the band with the maximum NMR-Gain/bit" or ***"Retrieve bits from the band with the***

maximum bits/NMR-Loss", where bits/NMR-loss is the number bits we save if we give away one unit of NMR.

3. MAX BITS/NMR-LOSS BIT ASSIGNMENT SCHEME

In this section, a new bit assignment scheme designed for MPEG-4 AAC based our new strategy is described. First, we define NMR-Gain/bit and bits/NMR-Loss by the following equations.

$$NMR-Gain/bit = (NMR_{ref} - NMR_{new})/(bits_{new} - bits_{ref}) \quad (1)$$

and

$$bits/NMR-Loss = (bits_{new} - bits_{ref})/(NMR_{new} - NMR_{ref}). \quad (2)$$

Figure 1 is the block diagram of the Max bits/NMR-Loss based bit assignment scheme. Each step in Figure 1 will be elaborated in the following sub-sections.

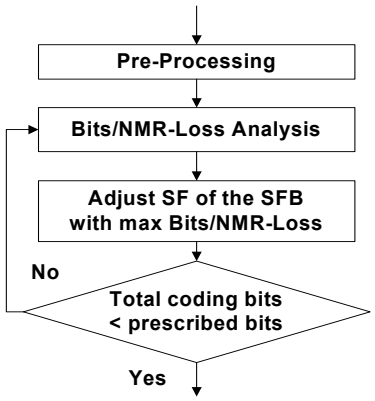


Figure 1. Max bits/NMR-Loss bit assignment scheme

3.1. Pre-Processing

The pre-processing step is to initialize two of the major parameters in the *bits/NMR-Loss* analysis: the reference NMR and the reference bits. There are no particular values associated with these parameters and thus the design of the pre-processing is case-dependent. In our implementation, we set the reference *NMR*=1 (0dB) for all the scale factor bands (SFB) at the beginning of processing a frame. After that, the reference scale factor (SF) for each SFB and the reference bits are calculated based on the input audio data.

3.2. Bits/NMR-Loss Analysis and SF Adjustment

In this scheme, only one SF value (of one SFB) is adjusted in one adjustment iteration. The detailed process is described below.

1. Initialization. Get the reference bits (B_{ref}), and the reference SFs (sf_{ref}) and NMRs (NMR_{ref}) for all SFBs (*N_SFB* SFB in total) from the pre-processing step. Start the max *bits/NMR-Loss* analysis from the first SFB and thus set the SFB index i=1.
2. Find the local max *bits/NMR-Loss* ratio of the ith SFB, $BNLR_i$, by computing

   $$BNLR_i = \max_{sf}\{(B_{ref} - B_{sf})/(NMR_{sf,i} - NMR_{ref,i})\}$$

   $$\forall sf \ and \ sf_{ref,i} < sf \le sf_{\max,i}$$

   The B_{sf} is the new value of the total coding bits in the current frame if the SF value (of the ith SFB) is changed from $sf_{ref,i}$ to $sf_{sf,i}$. The $sf_{\max,i}$ is the SF value that quantizes all the spectral coefficients in the ith SFB to zero. The local optimal SF (of the ith SFB), $sf_{opt,i}$, is the SF with the maximum *BNLR*. The local optimal coding bits of the ith SFB, $B_{opt,i} = B_{sf_{opt,i}}$, is also recorded.
3. If i<*N_SFB*, update i to i+1 and go to step 2.
4. Find the global maximum *bits/NMR-Loss* ratio, $BNLR_{global}$, by computing

   $$BNLR_{globe} = \max_i\{BNLR_i\} \quad \forall i, 0 \le i < N\_SFB$$

   The global optimal SFB, sfb_{global}, is the SFB that has the $BNLR_{global}$. Then, the global optimal SF, sf_{global}, is the local optimal SF of the sfb_{global}-*th* SFB. Similarly, the global optimal coding bits, B_{global}, is the coding bits of the sfb_{global}-*th* SFB.
5. Set the SF of the sfb_{global}-*th* SFB to sf_{global}. Update parameters for the sfb_{global}-*th* SFB; that is, $sf_{ref,sfb_{global}} = sf_{global}$ and $NMR_{ref,sfb_{global}} = NMR_{sf_{global},sfb_{global}}$.
6. Compare B_{global} to the prescribed rate, B. If $B_{global} > B$, update B_{ref} to B_{global} and go to step 2.

Note that, in performing the local maximum *bits/NMR-Loss* ratio analysis in step 2, only the SF of one SFB that is being examined is modified. The SF of the other SFBs are kept unchanged.

3.3. Trellis-Based Optimization on HCB

Total coding bits calculation in step 2 in the **Bits/NMR-Loss Analysis** (in sub-section 3.2) is one of the most computational-intensive processes. When the SF for each SFB is determined, the quantized spectral coefficients are also fixed. Before calculating the total coding bits, the HCB for each SFB has to be chosen first. The MPEG-4 AAC Verification Model (VM) has a simple algorithm for this purpose; however, a more efficient algorithm is needed for HCB decision. Thus, we adopt the Viterbi-based approach in this paper.

The problem for finding the optimal HCB can be reformulated as minimizing the following cost function:

$$C_{HCB} = \sum_i b_i + R(h_{i-1}, h_i), \quad (3)$$

where b_i is the coding bits of the quantized spectral coefficients for the *i*th SFB, h_i is the HCB for the *i*th SFB, and *R* is the run-length coding function (bits needed) for coding HCB. We find that the contribution of h_i to C_{HCB} depends only on the *previous* choice, h_{i-1}. Therefore, the minimization of C_{HCB} can be achieved by finding the optimal path through the trellis using the Viterbi algorithm.

A trellis is thus constructed for minimizing C_{HCB}. Each stage in the trellis corresponds to an SFB and each state at the *i*th stage represents a HCB candidate for this scale factor band. In other words, for the *i*th stage, if a path passes through the *m*th state, the *m*th HCB is employed for encoding the *i*th SFB. The Viterbi search procedure is outlined below.

The *k*th state at the *i*th stage is denoted by $S_{k,i}$ and the minimum accumulative-partial cost ending at $S_{k,i}$ is denoted by $C_{k,i}$. The transition cost from $S_{n,i-1}$ to $S_{m,i}$ is $R(h_{n,i-1}, h_{m,i})$.

1. Initialize $C_{m,0} = 0, \forall m$. Initialize *i*=1.
2. *Search*. $\forall m$, the best path ending at $S_{m,i}$ is found by computing
$$C_{m,i} = \min_n \{C_{n,i-1} + b_{m,i} + R(h_{n,i-1}, h_{m,i})\}$$
3. If $i < N$, set $i = i+1$ (SFB) and go to step 2.

3.4. Fast algorithm for Bits/NMR-Loss Analysis

The most time-consuming computation in this bit assignment scheme is the trellis-based HCB optimization for coding bits calculation in step 2 (Search). For each SF modification in step 2, the new value of total coding bits needs to be recalculated. Therefore, for one SF adjustment iteration, we need to perform $(sf_{\max,i} - sf_{ref,i})$ times trellis-based HCB optimization processes for the local *bits/NMR-Loss* analysis. Hence, the total number of calculations for finding the global maximum *bits/NMR-Loss* is

$$\sum_{i=1}^{N\_SFB} (sf_{\max,i} - sf_{ref,i}). \quad (4)$$

There are at least two ways to reduce computations. One is to reduce the complexity of the trellis-based HCB optimization; the other is to reduce the number of trellis-based HCB optimization.

By analyzing the local optimal parameters, $sf_{opt,i}$ and $BNLR_i$, we find some interesting properties.

1. The average value of the difference between the local optimal SFs of the *m*th and the *(m+1)*th iterations, $sfdiff_{ave}$, is often close to zero.

$$sfdiff_{ave} = \frac{1}{(N\_SFB - 3)} \times \sum_{i \notin S} abs(sf_{opt,i}^{m+1} - sf_{opt,i}^{m}),$$

where $S = \{sfb_{global}^m - 1, sfb_{global}^m, sfb_{global}^m + 1\}$ and sfb_{global}^m is the global optimal SFB of the *m*th SF adjustment iteration.

2. The average value of the difference between the local max *bits/NMR-Loss* ratio of the *m*th and the *(m+1)*th iteration, $BNLRdiff_{ave}$, is typically quite small.

$$BNLRdiff_{ave} = \frac{1}{(N\_SFB - 3)} \times \sum_{i \notin S} abs(BNLR_i^{m+1} - BNLR_i^m)$$

Using these two properties, we can drastically reduce the number of *bits/NMR-Loss* analyses (trellis-based HCB optimizations). We only need to perform the *bits/NMR-Loss* analysis on three SFBs after the first SF adjustment iteration.

4. SIMULATION RESULTS

The computational complexity and objective quality based on our simulations are summarized in this section. The bits assignment schemes used in comparison are as follows.

(1) The MPEG-4 VM of AAC (VM-TLS) without modification.

(2) The modified MPEG-4 VM of AAC (VM-TLS-M), in which the HCB decision algorithm is replaced by the TB-HCB optimization procedure described in section 3.3.

(3) The trellis-based ANMR optimization (TB-ANMR) and the MNMR optimization (TB-MNRM), which are implemented as described in [3] and [4].

(4) The normal and fast max bits/NMR-Loss schemes (max-BNLR).

Ten audio files with sampling rate 44.1K are used as test sequences. Two of them are extracted from MPEG SQAM [6], and the others are from EBU [7].

4.1. Computational complexity

The storage and computational complexity of one iteration in various schemes are summarized in Table 1.

Table 1. Complexity Analysis

	Search complexity	Storage
VM-TLS	1	--
VM-TLS-M	$12^2 \times N\_SFB$	$12 \times N\_SFB$
TB-ANMR TB-MNMR	$(60 \times 2)^2 \times 12^2 \times N\_SFB$	$60 \times 2 \times 12 \times N\_SFB$
Max-BNLR	$N\_SFB \times \text{Ave\_SF} \times 12^2 \times N\_SFB$	$12 \times N\_SFB$
Fast Max-BNLR	※ (a) ($N_SFB \times$Ave_SF $\times 12^2 \times N\_SFB$) (b) $3 \times \text{Ave\_SF} \times 12^2 \times N\_SFB$	$12 \times N\_SFB$
※ (a) is only for the first iteration; all the rest are using (b)		

In this table, Ave_SF is the average number of SF tested for the max BNLR analysis for each SFB and its typical value is around 17 or so. Table 2 is the statistics collected from the simulations on audio sequences. It is clear that in terms of computational requirement:
Fast Max-BNLR<< Max-BNLR<< TB-ANMR(MNMR)

Table 2. Statistics on Computational Complexity

	Average iteration /frame	**Average TB HCB optimi-zations/ frame**	**Average TB HCB optimi-zations/ iteration**	**Complexity ratio**
TB-ANMR (MNMR)	12	14400*12	14400	1
Max-BNLR	50	10103	10103/12 = 842	1/17
Fast Max-BNLR	50	1153	1153/12 = 96	1/150

4.2. Objective results

Two common objective quality measurements, average noise to mask ratio (ANMR) and maximum noise to mask ratio (MNMR) [5], are adopted in the performance comparison. Note that, in evaluating distortion, the NMR is set to 0 dB if the original NMR value is less than 0 dB. The rate-distortion curves of six bit assignment schemes are shown in Figures 2 and 3. (Note: TB-ANMR and TB-MNMR are similar algorithms aiming at two different target NMRs.) We can find that the ANMR performance of the Max-BNLR scheme is almost as good as that of TB-ANMR. There is almost no loss of ANMR performance in using the fast algorithm for Max-BNLR either. The MNMR values of TB-ANMR, Max-BNLR and Fast Max-BNLR are also similar. The characteristic of the proposed Max-BNLR scheme is closer to that of TB-ANMR as compared to TB-MNMR. Again, TB-ANMR and TB-MNMR are the optimal solutions tuned for their target cost functions, ANMR and MNMR, respectively [3][4].

4. CONCLUSIONS

In this paper, we propose a new concept, bit-use efficiency, for improving audio coding performance. Furthermore, a new bits assignment scheme based on this new concept (strategy) is proposed for MPEG-4 AAC, named Max-BNLR. Simulation results show that the Max-BNLR scheme has a performance close to TB-ANMR and is much better than the MPEG VM. In addition, its computational complexity is much lower than that of TB-ANMR.

5. REFERENCES

[1] ISO/IEC JTC1/SC29, "Information technology – vary low bitrate audio-visual coding," *ISO/IEC IS-14496 (Part 3, Audio),* 1998

[2] M. Bosi, *et al.*, "ISO/IEC MPEG-2 advanced audio coding," *Journal of Audio Engineering Society,* vol. 45, pp. 789-814, October 1997.

[3] A. Aggarwal, S.L. Regunathan, K. Rose, "Trellis-based optimization of MPEG-4 advanced audio coding," *Proc. IEEE Workshop on Speech Coding*, pp. 142-4 2000.

[4] A. Aggarwal, S.L. Regunathan, K. Rose, "Near-optimal selection of encoding parameters for audio coding," *Proc. of ICASSP*, vol. 5, pp. 3269-3272, Jun 2001.

[5] H. Najafzadeh and P. Kabal, "Perceptual bit allocation for low rate coding of narrowband audio," *Proc. of ICASSP*, vol. 2, pp. 893-896, 2000.

[6] "The MPEG audio web page." http://www.tnt.uni-hannover.de/project/mpeg/audio.

[7] European Broadcasting Union, *Sound Quality Assessment Material: Recordings for Subjective Tests,* Brussels, Belgium, Apr. 1988.

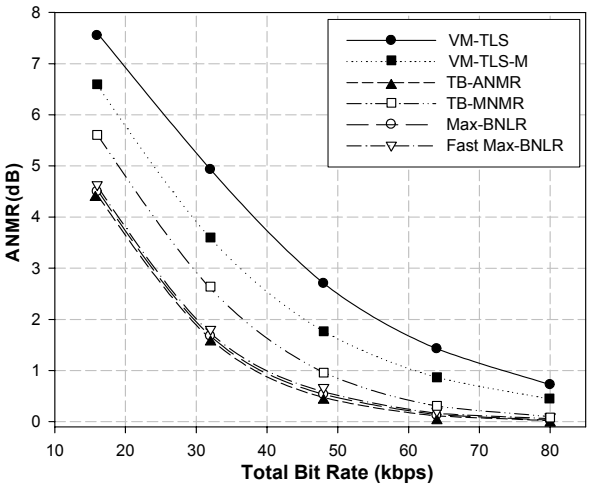


Figure 2. ANMR rate-distortion analyses

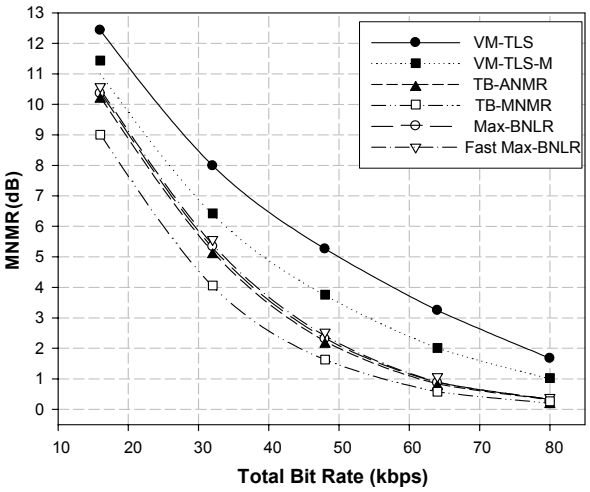


Figure 3. MNMR rate-distortion analyses

<附件二>

Audio Engineering Society
Convention Paper

Presented at the 115th Convention
2003 October 10–13 New York, New York

Cascaded Trellis-Based Optimization For MPEG-4 Advanced Audio Coding

Cheng-Han Yang [1], Hsueh-Ming Hang [1]

[1] Department of Electronics Engineering, National Chiao Tung University, Hsinchu, Taiwan, R.O.C.

hmhang@mail.nctu.edu.tw; Fax: (886)-3-5723283
u8911831@cc.nctu.edu.tw; Fax: (886) -3-5731791

ABSTRACT

A low complexity and high performance scheme for choosing MPEG-4 Advanced Audio Coding (AAC) parameters is proposed. One key element in producing good quality compressed audio at low rates in particular is selecting proper coding parameter values. A joint trellis-based optimization approach has thus been previously proposed. It leads to a near-optimal selection of parameters at the cost of extremely high computational complexity. It is, therefore, very desirable to achieve a similar coding performance (audio quality) at a much lower complexity. Simulation results indicate that our proposed cascaded trellis-based optimization scheme has a coding performance close to that of the joint trellis-based scheme, and it requires only 1/70 in computation.

1. INTRODUCTION

To meet the demand of various multimedia applications, many high-efficient audio coding schemes have been developed. The MPEG-4 Advanced Audio Coding (AAC) is one of the most recent-generation audio coders specified by the ISO/IEC MPEG standards committee [1]. It is a very efficient audio compression algorithm aiming at a wide variety of different applications, such as Internet, wireless, and digital broadcast arenas [2].

One key element in an AAC coder is selecting two sets of coding parameters properly, the scale factor (SF) and Huffman codebook (HCB) in the rate-distortion (R-D) loop. Because encoding these parameters is inter-band dependent, i.e., the coded bits produced for the second band depend on the choice of the first band, the choice of their proper values so as to minimize the objective quality becomes fairly difficult. As discussed in [3][4], the poor choice of parameters for rate control is one shortcoming of the current MPEG-4 AAC Verification Model (VM) and therefore its compression efficiency is not as expected at low bit rates.

Some methods such as vector quanitzers rather than scalar quantizers have been suggested to reduce the side information [5][6]. They would alter the syntax of the standards. In this paper, we focus on finding the parameters in the existing AAC standard that produce the (nearly) optimal compressed audio quality for a given bit rate.

In [3] and [4], a joint optimization scheme, which takes the inter-band dependence into account, is proposed for choosing the encoding parameters for all the frequency bands. This joint optimization is formulated as a trellis search and is, therefore, called *trellis-based optimization*. Although the complexity of this joint trellis-based optimization scheme can be reduced by adopting the Viterbi algorithm, its search complexity is still extremely high and is thus not suitable for practical applications.

In this paper, we propose a *cascaded trellis-based* (CTB) optimization scheme for selecting the proper encoding parameters. Our scheme retains the good audio quality offered by the joint trellis-based (JTB) optimization while its search complexity is drastically decreased.

The organization of this paper is as follows. In section 2, a brief overview of MPE-4 AAC is provided. The proposed CTB scheme with several variations for choosing the optimal coding parameters is described in sections 3, 4 and 5. The algorithm complexity analysis and the simulation results are summarized in section 6.

2. OVERVIEW OF AAC ENCODER

The basic structure of the MPEG-4 AAC encoder is shown in Figure 1. The time domain signals are first converted into the frequency domain (spectral coefficients) by the modified discrete cosine transform (MDCT). For tying in with the human auditory system, these spectral coefficients are grouped into a number of bands, called scale factor bands (SFB). The pre-process modules, which are the optional functions, can help removing the time/frequency domain redundancies of the original signals. The psychoacoustic model calculates the spectral coefficient masking threshold, which is the base for deciding coding parameters in the R-D loop. The R-D loop, our focus in this paper, is to determine two critical parameters, SF and HCB for each SFB so as to optimize the selected criterion under the given bit rate constraint. The SF is related to the step size of the quantizer, which determines the quantization noise-to-masking ratio (NMR) in each band. The quantized coefficients are then entropy-coded by one of the twelve pre-designed HCBs. In addition, the indices of SFs and HCBs are coded using differential and run-length codes respectively and are transmitted as side information.

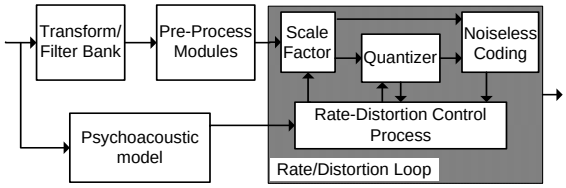


Fig. 1. Basic structure of the MPEG-4 AAC encoder

3. CASCADED TRELLIS-BASED OPTIMIZATION

The JTB optimization approach can substantially enhance the coding performance at low bit rates [3][4]. However, this approach also results in a very high computational complexity. The coding parameters in the JTB scheme, SF and HCB, are optimized simultaneously by using the trellis search. The states at the *i*th stage in the trellis for the JTB scheme represent all combinations of SF and HCB for the *i*th SFB. Different from the JTB scheme, our scheme, so-called *cascaded trellis-based scheme* (CTB), finds the proper coding parameters, SF and HCB, in two consecutive steps. The search complexity can thus be drastically reduced, while the advantage of trellis-based optimality is mostly retained.

The way that the trellis search performs depends on the optimization criterion it adopts. There are two frequently used criteria, the average noise-to-mask ratio (ANMR) and the maximum noise-to-mask ratio (MNMR) [7]. Both criteria will be used in this paper.

3.1. Trellis-Based ANMR Optimization on SF

The constrained optimization problem for the ANMR criterion is formulated as below.

$$\min \sum_i w_i d_i \quad \text{s.t.}$$

$$\sum_i \left(b_i + D(sf_i - sf_{i-1}) + R(h_{i-1}, h_i)\right) \le B$$

, where w_i is the inverse of the masking threshold and d_i is the quantization distortion. Under this criterion, we minimize the sum of the perceptually weighted distortion. The coding parameters, SF and HCB, for the *i*th SFB is denoted by sf_i and h_i. Symbol D is the differential coding function performed on SF and symbol R is the run-length coding function performed on HCB. The returned function values in both cases are bits to encode the arguments. Parameter b_i is the bits for coding the quantized

spectral coefficients (QSCs) and the parameter B is the prescribed bit rate for a frame.

As described in [3], the ANMR optimization problem can be reformulated as minimizing the unconstrained cost functions, C_{ANMR}, with the Lagrangian multiplier λ.

$$C_{ANMR} = \sum_i w_i d_i + \lambda \cdot (b_i + D(sf_i - sf_{i-1}) + R(h_{i-1}, h_i)) \quad (1)$$

Different from that in the original JTB scheme, the optimization problem in our CTB scheme is reformulated as minimizing two unconstrained cost functions, C_{SF_ANMR} and C_{HCB}, as follows.

$$C_{SF\_ANMR} = \sum_i w_i d_i + \lambda \cdot (b_i + D(sf_i - sf_{i-1})) \quad (2)$$

$$C_{HCB} = \sum_i b_i + R(h_{i-1}, h_i) \quad (3)$$

The minimization of C_{SF_ANMR} is described in this sub-section, and the minimization of C_{HCB} will be described in section 3.3.

Similar to the approach in the JTB scheme, the goal for finding proper SFs that minimize C_{SF_ANMR} can be achieved by finding the optimal path through the trellis. Each stage in the trellis corresponds to an SFB. (There are *N_SFB* stages in total.) However, different from JTB, each state at the *i*th stage in our scheme only represents a SF candidate for the *i*th SFB. In other words, at the *i*th stage, if a path passes through the *m*th state, it means that the *m*th SF candidate is employed to encode the *i*th SFB. For a given value of λ, the Viterbi search procedure described in [3] is modified as stated below.

The *k*th state at the *i*th stage is denoted by $S_{k,i}$ and the minimum accumulative-partial cost ending at $S_{k,i}$ is denoted by $C_{k,i}$. The state-transition cost, $T_{l,i-1\to k,i}$, from $S_{l,i-1}$ to $S_{k,i}$ is $\lambda \cdot D(sf_{k,i} - sf_{l,i-1})$.

1. Initialize $C_{k,0} = 0, \forall k$ and i =1.
2. Search for, $\forall k$, the best path ending at $S_{k,i}$ by computing
$$C_{k,i} = \min_l \{ C_{l,i-1} + w_i d_{k,i} + \lambda \cdot b_{k,i} + T_{l,i-1\to k,i} \} \quad (4)$$
3. If $i < N\_SFB$, set $i = i+1$ and go to step 2.

3.2. Trellis-Based MNMR Optimization on SF

The constrained optimization problem for the MNMR criterion is formulated below.

$$\min \left(\max_i w_i d_i \right) \quad \text{s.t.}$$

$$\sum_i \left(b_i + D(sf_i - sf_{i-1}) + R(h_{i-1}, h_i) \right) \le B$$

, where $\max_i w_i d_i$ is the maximum NMR in a frame. Again, using the unconstrained format, the cost function in the JTB scheme [4] becomes

$$C_{MNMR} = \sum_i b_i + D(sf_i - sf_{i-1}) + R(h_{i-1}, h_i) \quad (5)$$

Different from the cost function in the JTB scheme [4], the MNMR optimization problem in our CTB scheme is reformulated as the minimization of two cost functions, C_{SF_MNMR} and C_{HCB} (Eqn.(3)), under the constraint: $w_i d_i \le \lambda, \ \forall i$, for some constant value of λ,

$$C_{SF\_MNMR} = \sum_i b_i + D(sf_i - sf_{i-1}) \quad (6)$$

Similar to the trellis-based ANMR optimization on selecting SF, a trellis is constructed for minimizing C_{SF_MNMR} and each state at the *i*th stage only represents a SF candidate for the *i*th SFB. The Viterbi search procedure described in [4] is modified as stated below. The state-transition cost, $T_{l,i-1\to k,i}$, from $S_{l,i-1}$ to $S_{k,i}$ is $D(sf_{k,i} - sf_{l,i-1})$.

1. Initialize $C_{k,0} = 0, \forall k$ and i =1.
2. Find the valid states for the *i*th stage, $S_{k,i}, \forall k$. A state is valid if the NMR ($w_i d_{k,i}$) corresponding to that state parameter is $\le \lambda$.
3. Search for, $\forall k$, the best path ending at the valid state $S_{k,i}$ by computing
$$C_{k,i} = \min_l \{ C_{l,i-1} + b_{k,i} + T_{l,i-1\to k,i} \} \quad (7)$$
4. If $i < N\_SFB$, set $i = i+1$ and go to step 2.

As pointed in [3][4], in the trellis for selecting the optimal SF (for both ANMR and MNMR), each state is further split into two consecutive states. In the first state, the spectral coefficients are quantized using the assigned valid SF, and in the second state, all quantized values of spectral coefficients are set to zero.

3.3. Trellis-Based Optimization on HCB

The HCB optimization is performed under the condition that the SF for each SFB has already been determined. With a determined SF, QSCs (quantized spectral coefficients) for each SFB are fixed and thus the b_i term in the cost function C_{HCB} (Eqn.(3)) only depends on HCB. The optimization procedure here is to find the HCBs that minimize the cost function C_{HCB} and this can be achieved again by finding the optimal path through the trellis.

A trellis is thus constructed for minimizing C_{HCB}. Each stage in this trellis corresponds to an SFB (There are *N_SFB* stages in total.) and each state at the *i*th stage represents a HCB candidate for the *i*th SFB. In other words, at the *i*th stage, if a path passes through the *m*th state, the *m*th HCB candidate is employed for encoding the *i*th SFB. The state-transition cost, $T_{n,i-1\to m,i}$, from $S_{n,i-1}$ to $S_{m,i}$ is $R(h_{n,i-1}, h_{m,i})$. The Viterbi search procedure for finding optimal HCBs is as follows.

1. Initialize $C_{m,0} = 0, \forall m$. Initialize *i*=1.
2. Search for, $\forall m$, the best path ending at $S_{m,i}$ by computing
$$C_{m,i} = \min_n \{ C_{n,i-1} + b_{m,i} + T_{n,i-1\to m,i} \} \quad (8)$$
3. If *i* < *N_SFB*, set *i* = *i*+1 and go to step 2.

3.4. Cascaded Trellis-Based Optimization

The block diagram of the CTB optimization scheme is shown in Figure 2 and the processing steps are described below.

1. Initialize λ.
2. For a given λ, a set of optimal SF, $\boldsymbol{sf}_{opt}$, is determined by the trellis-based SF optimization procedure using the Virtual-HCB Mode.
3. For the given $\boldsymbol{sf}_{opt}$ obtained from step 2, a set of optimal HCB, $\boldsymbol{hcb}_{opt}$, is determined by the trellis-based HCB optimization procedure.
4. For the given $\boldsymbol{hcb}_{opt}$ obtained from step 3 and λ, a set of recalculated optimal SF, $\boldsymbol{sf}'_{opt}$, is obtained from the Fixed-HCB Mode trellis-based SF optimization procedure.
5. *Adjust rate*. For the given optimal $\boldsymbol{sf}'_{opt}$ (or $\boldsymbol{sf}_{opt}$) and $\boldsymbol{hcb}_{opt}$, the total coding bit rate is calculated and compared to the prescribed bit rate (***B***). Adjust λ and go to step 2 if the constraint is not met.

In the preceding procedure, the trellis-based ANMR (MNMR) optimization on SF is applied to steps 2 and 4.

As described in the preceding procedure, the trellis-based optimization procedure on SF is operated in two different modes, Virtual-HCB Mode and Fixed-HCB Mode. In these two modes, the value of $b_{k,i}$ in Eqn.(4) and Eqn.(7) is determined in different ways, and these will be described in section 4.

The preceding procedure is called the *full optimization mode* (or *Two-Loop mode*), because the optimization procedure on SF is done twice. The second optimization procedure on SF (step 4) can help in recovering some improper SF values determined in step 2. Furthermore, the CTB optimization can also been operated in a simpler optimization mode (so-called *One-Loop mode*), in which step 4 is not included.

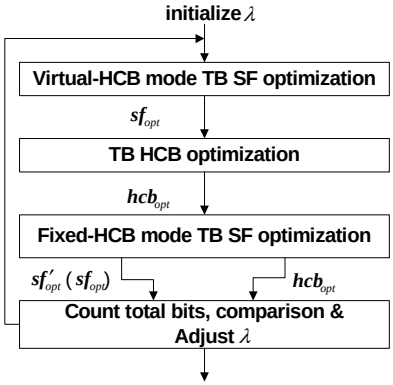


Fig. 2. Cascaded trellis-based optimization scheme

4. FIXED AND VIRTUAL HCB MODE FOR SF OPTIMIZATION

For an identified $C_{k,i}$ in Eqn.(4) and Eqn.(7) in sections 3.1 and 3.2, $w_i d_{k,i}$ or $D(sf_{k,i} - sf_{l,i-1})$ is unique for a given state or state transition. However, the value of $b_{k,i}$ depends not only on the state parameter $sf_{k,i}$; it also depends on the choice of HCB. In the JTB optimization scheme, for each candidate value of SF, all possible *b* values, corresponding to 12 pre-designed HCBs, are evaluated. But in our SF optimization scheme, we have to determine one proper value of $b_{k,i}$ for the state $S_{k,i}$. According to our implementation, the trellis-based ANMR or MNMR SF optimization scheme can operate in two modes, the Fixed-HCB mode and the Virtual-HCB mode, for determining the value of $b_{k,i}$.

In the Fixed-HCB mode, a set of fixed HCBs, $[h_1^f\ h_2^f \ldots\ldots h_{N\_SFB}^f]$, is determined beforehand. For all the states at the *i*th stage, the QSCs, $\boldsymbol{q}_{k,i}$, are encoded using h_i^f; thus, $\forall k$, $b_{k,i} = h_i^f(\boldsymbol{q}_{k,i})$.

In the Virtual-HCB mode, a Virtual HCB, $h_{k,i}^v$, is employed for state $S_{k,i}$. Thus, $h_{k,i}^v$ needs to be pre-constructed to help us in determining $b_{k,i}$ and it can be constructed in several ways. For example, $h_{k,i}^v$

may be one of the 12 pre-designed HCBs or a compound codebook. Consequentially, the more accurate of the $b_{k,i}$ and $h_{k,i}^{v}$ we can estimate, the higher accuracy of SF optimization we can achieve. In order to improve the accuracy of the estimated values of $b_{k,i}$ and $h_{k,i}^{v}$, we did some analysis on the JTB optimization scheme.

For a given value of λ, by applying the JTB scheme, we can find a set of optimal parameters, $\boldsymbol{sf}_{opt}^{\boldsymbol{JTB}}$, $\boldsymbol{hcb}_{opt}^{\boldsymbol{JTB}}$ and $\boldsymbol{b}_{opt}^{\boldsymbol{JTB}}$ that minimize the cost function C_{ANMR} (Eqn.(1)) or C_{MNMR} (Eqn.(5)). For comparison purpose, we also construct an ideal set of bits for coding QSCs, $\boldsymbol{b}_{\boldsymbol{min}}^{JTB}$. For the ith SFB, $b_{\boldsymbol{min},i}^{JTB}$ is the minimum value of bits for coding $\boldsymbol{q}_{opt,i}^{\boldsymbol{JTB}}$ using 12 pre-designed HCBs and is formulated as:

$$b_{\min,i}^{JTB} = \min_m \left\{ h_m(\boldsymbol{q}_{opt,i}^{JTB}) \right\}$$

, where $\boldsymbol{q}_{opt,i}^{\boldsymbol{JTB}}$ is QSCs quantized by $sf_{opt,i}^{\boldsymbol{JTB}}$.

The histogram of the differences between $\boldsymbol{b}_{opt}^{\boldsymbol{JTB}}$ and $\boldsymbol{b}_{\min}^{\boldsymbol{JTB}}$, denoted by $\boldsymbol{b}_{opt-\min}^{JTB}$, is shown in Figure 3. We can find that over 91% of $\boldsymbol{b}_{opt-\min}^{JTB}$ is less than 3 for both ANMR and MNMR criterions. From the other viewpoint, we tend to choose the HCB that results in nearly the minimum QSCs bits.

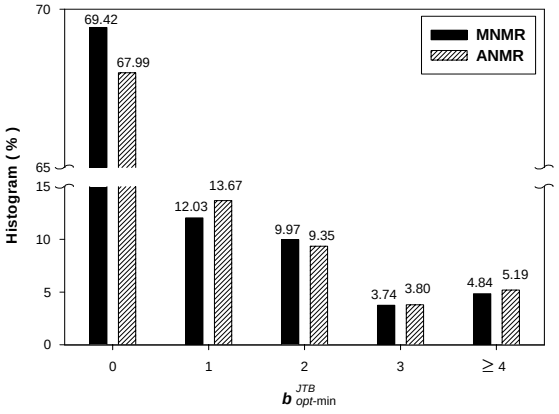


Fig. 3. Comparison against $\boldsymbol{b}_{opt-\min}^{JTB}$

Observing this characteristics of $\boldsymbol{b}_{opt}^{\boldsymbol{JTB}}$, we derive a rule in determining $h_{k,i}^{v}$ and $b_{k,i}$. For state $S_{k,i}$, the candidate $h_{k,i}^{v}$ values are the set of HCB that satisfies the proposed rule in Eqn.(9); namely,

$$h(\boldsymbol{q}_{\boldsymbol{k,i}}) \le \min_m \left\{ h_m(\boldsymbol{q}_{\boldsymbol{k,i}}) \right\} + \delta \qquad (9)$$

Then, $b_{k,i}$ is formulated as:

$$b_{k,i} = \frac{1}{\left|h_{k,i}^{v}\right|} \sum_{n \in h_{k,i}^{v}} h_n(q_{k,i}) + \alpha \cdot R_v(h_{l,i-1}^{v}, h_{k,i}^{v}) \qquad (10)$$

R_v is the run-length coding function performed on the Virtual HCB and it is similar to R in our implementation.

$$R_v(h_{l,i-1}^{v}, h_{k,i}^{v}) = \begin{cases} 0, & if\ (h_{l,i-1}^{v} \cap h_{k,i}^{v}) \neq \phi \\ 9, & else \end{cases} \qquad (11)$$

, where α is a weight for including $R_v(h_{l,i-1}^{v}, h_{k,i}^{v})$ in $b_{k,i}$. Finally, we have to determine suitable values for δ and α. The simulation results of the normalized difference, (C_{ANMR}^{JTB} - C_{ANMR}^{CTB}) or (C_{MNMR}^{JTB} - C_{MNMR}^{CTB}), versus different values of δ and α are shown in Figure 4 and 5.

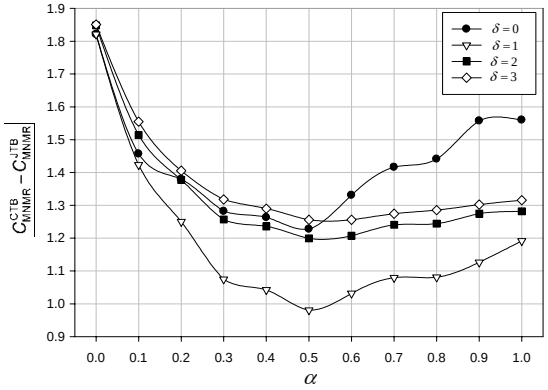


Fig. 4. (C_{MNMR}^{JTB} - C_{MNMR}^{CTB}) v.s (δ,α)

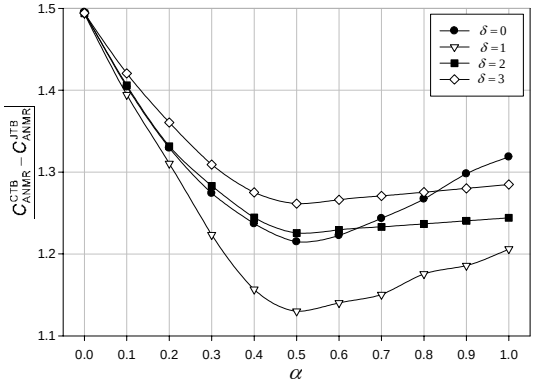


Fig. 5. (C_{ANMR}^{JTB} - C_{ANMR}^{CTB}) v.s (δ,α)

In this notation, C_{ANMR}^{JTB} (or C_{MNMR}^{JTB}) is the minimal C_{ANMR} (or C_{MNMR}) derived from the JTB scheme and C_{ANMR}^{CTB} (or C_{MNMR}^{CTB}) is the minimal C_{ANMR} (or C_{MNMR}) derived from the CTB scheme. We find that for a

wide range of δ values, we can achieve better performance when $R_v(h^v_{l,i-1}, h^v_{k,i})$ is included in $b_{k,i}$ (α > 0). As shown in Figure 4, the CTB scheme can achieve the nearly best performance when δ=1 and α=0.5. Therefore, we choose 1 for δ and 0.5 for α in our implementation.

5. FAST SEARCHING ALGORITHM

The computational complexity of the trellis-based optimization scheme depends on the searching range (number of states) of each stage in the trellis. Hence, reducing the candidate states at each stage is an effective way in reducing the complexity. Base on this idea, we propose fast searching algorithms for the trellis-based optimization schemes on SF and HCB.

5.1. Fast Searching Algorithm for HCB Optimization

In MPEG-4 AAC, SFs are differentially coded and HCBs are coded by run-length coding. Run-length coding can be viewed as a special case of differential coding; therefore, the procedure of trellis-based optimization on HCB is similar to that on SF. However, the output of run-length coding has only two possible values, either 0 or 9. In looking for $C_{k,i}$ in Eqn.(8), the cost of run-length coding is as follows.

$$R(h_{n,i-1}, h_{m,i}) = \begin{cases} 0, & if\ n = m \\ 9, & else \end{cases} \quad (12)$$

In HCB optimization, each state at the *i*th stage represents a HCB candidate. As shown in Figure 6(a), for finding the optimal path ending at $S_{m,i}$, all the HCB candidates at (*i-1*)th stage have to been taken into consideration. In MPEG-4 AAC, there are 12 pre-designed HCBs, so the searching complexity for finding all the optimal paths ending at the *i*th stage is 12×12.

The number next to the arrow in Figure 6 is the state-transition cost. Except for the path $S_{m,i-1} \rightarrow S_{m,i}$, the state-transition costs of the other paths ending at $S_{m,i}$ are all the same (equal to 9). Therefore, in calculating $C_{k,i}$ in Eqn.(8), among these 11 paths, the path with the smallest $C_{n,i-1}$ will result in the smallest $C_{m,i}$. Based on this property, a fast searching algorithm is proposed and is divided into two steps.

1. Among the 12 candidate states at (*i-1*)th stage, the state with the minimum cost, $C_{\min,i-1}$, is chosen and treated as the virtual thirteenth state, $S_{\min,i-1}$.

$$C_{\min,i} = \min_n \{ C_{n,i-1} \}$$

2. As shown in Figure 6(b), while finding the optimal path ending at $S_{m,i}$, we only have to consider two paths, path ($S_{m,i-1} \rightarrow S_{m,i}$) and path ($S_{\min,i-1} \rightarrow S_{m,i}$). The rest of searching procedure is the same to that in section 3.3.

The searching complexity (in terms of branch metric calculation) of this fast algorithm is approximately 12 + 2×12. The first "12" term is the computational complexity for determining $S_{\min,i-1}$. Note that the performance (accuracy) of the fast searching algorithm is the same to that of the full searching algorithm.

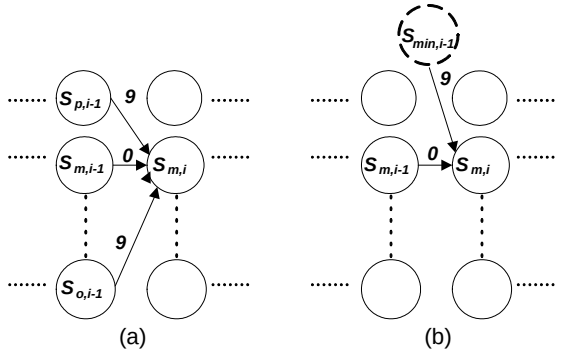


Fig. 6. Trellis representation of HCB optimization

5.2. Fast Searching Algorithm for SF Optimization

In SF optimization, each state in the trellis represents a candidate value of SF. Searching over a larger set of SF candidates can result in better performance, but at the cost of higher searching complexity.

In general, the state numbers (*sn*) for all the stages in the trellis are the same and the searching complexity for each stage transition in this uniform *sn* trellis is *sn* × *sn*. In this section, we propose two non-uniform (adaptive) *sn* algorithms, in which the *sn* for each stage in the trellis can vary to reduce the searching complexity. The first one is called "global minimum reference SF restricted non-uniform *trellis*", or "Gm_Nu" in short, and the second one is called "local minimum reference SF restricted non-uniform *trellis*", or "Lm_Nu". In both cases, a reference SF is first identified and then the number of candidates is reduced against this reference.

In the first step, we define the reference SF, sf_i^{ref}, for *i*th SFB as the largest SF among all the valid states at the *i*th stage. Then we can find a global

minimum reference SF, $sf_{G\cdot Min}^{ref}$, which is the minimum SF among all the reference SFs. In the Gm_Nu algorithm, we restrict the SF candidates at the *i*th stage in the range of $[sf_i^{ref}, sf_{G\cdot Min}^{ref} - \varepsilon]$. Thus, the *sn* at the *i*th stage, $sn_{Gm,i}$, equals to $(sf_i^{ref} - sf_{G\cdot Min}^{ref} + 1 + \varepsilon)$.

Next, we define the *n*th-order local reference minimum SF at the *i*th stage, $sf_{L\cdot Min,i}^{ref}$, where

$$sf_{L\cdot Min,i}^{ref} = \min_{i-n \le j \le i+n} \left\{ sf_j^{ref} \right\} \quad (13)$$

In the Lm_Nu algorithm, we restrict the SF candidates at the *i*th stage in the range of $[sf_i^{ref}, sf_{L\cdot Min,i}^{ref} - \varepsilon]$. Therefore, the *sn* for the *i*th stage, $sn_{Lm,i}$, equals to $(sf_i^{ref} - sf_{L\cdot Min,i}^{ref} + 1 + \varepsilon)$. In both cases, ε is a parameter to control the searching range for all stages. In the simulations in section 6, the values of *n* in Eqn.(13) and ε are both set to 1.

6. SIMULATION RESULTS

In this section, we will discuss the computational complexity and the coded audio quality in our experiments. Three types of bits allocation algorithms have been tested and compared as described below.

(1) The MPEG-4 VM of AAC (VM-TLS).
(2) The joint trellis-based ANMR and MNMR optimization schemes, abbreviated as JTB-ANMR and JTB-MNRM respectively, described in [3] and [4].
(3) The cascaded trellis-based ANMR and MNMR optimization schemes, abbreviated as CTB-ANMR CTB-MNMR respectively, described in section 3.

Ten two-channel audio sequences with sampling rate 44.1kHz are tested. Two of them are extracted from MPEG SQAM [8], and the others are from EBU [9].

6.1. Complexity Analysis

The computational complexity analyses for the aforementioned several coding schemes are summarized in Table 1. The value in "Computation" column is the searching complexity in calculating one stage transition in the trellis in terms of branch metric computation. For the convenience of comparison, the full-search JTB is set as the reference (ratio=1) and all the other schemes are rated based this base. Also shown in Table 1 is the storage requirement. Again, it is measured in terms of one branch metric computational needs.

We can find from Table 1 that the *n*-Loop CTB scheme is approximately (142/*n*) times faster than the JTB scheme. Moreover, the storage requirement for the trellis search in the CTB scheme is much smaller than that in the JTB scheme.

For the JTB scheme, the fast HCB optimization algorithm can reduce the complexity to 1/4. Note that sn_{Gm}^{ave} and sn_{Lm}^{ave} in Table 1 are the average *sn* in the Gm_Nu and Lm_Nu algorithms and are calculated by Eqn.(14) and Eqn.(15).

$$sn_{Gm}^{ave} = \left(\frac{1}{NB\_SFB} \sum_{i=1}^{NB\_SFB} (sn_{Gm,i-1} \cdot sn_{Gm,i}) \right)^{1/2} \quad (14)$$

$$sn_{Lm}^{ave} = \left(\frac{1}{NB\_SFB} \sum_{i=1}^{NB\_SFB} (sn_{Lm,i-1} \cdot sn_{Lm,i}) \right)^{1/2} \quad (15)$$

The simulation results show that typical sn_{Gm}^{ave} is approximately 12 and sn_{Lm}^{ave} is about 5. Hence, the Gm_Nu algorithm can reduce the complexity to 1/25 and the Lm_Nu algorithm can reduce the complexity to 1/144.

Table 1. Complexity Analysis

	Computation	**Ratio**	**Storage**
JTB	$(60\times 2)^2 \times 12^2$	1	60×2×12
n-Loop CTB	$n\times(60\times 2)^2 + 12^2$	n /142	60×2
JTB + Fast HCB	$(60\times 2)^2 \times 36$	1/4	60×2×12
JTB + Fast HCB + Gm_Nu	$(sn_{Gm}^{ave}\times 2)^2 \times 36$	1/100	60×2×12
JTB + Fast HCB +Lm_Nu	$(sn_{Lm}^{ave}\times 2)^2 \times 36$	1/576	60×2×12
n-Loop CTB + Gm_Nu + Fast HCB	$n\times(sn_{Gm}^{ave}\times 2)^2 + 36$	n /3600	60×2
n-Loop CTB + Lm_Nu + Fast HCB	$n\times(sn_{Lm}^{ave}\times 2)^2 + 36$	(n+0.4) / 20736	60×2

6.2. Objective Quality Analysis

The rate-distortion curves of these bit allocation schemes are shown in Fig. 7 and 8. Two major evaluative methodologies, ANMR and MNMR, are used for distortion. We can find that the performance of the CTB scheme is similar to that of the JTB scheme. The ANMR performance loss is less than 0.2dB for One-Loop CTB-ANMR and less than 0.1dB for Two-Loop CTB-ANMR (the lowest three curves in Fig. 7). The MNMR performance loss is less than 0.1 dB for both One- and Two-Loop CTB-MNMR (the lowest three curves in Fig. 8). Both of them are much better than the MPEG-4 VM (the top line).

The differences of performance between the fast searching algorithms and the original CTB-MNMR scheme are shown in Fig. 9 and 10. In light of the complexity analyses on Gm_Nu and Lm_Nu, and the uniform *NB_SF* fast algorithms, with *NB_SF*=12 and 5, are chosen for comparison. There is nearly no performance loss for the Gm_Nu algorithm (ANMR or MNMR Difference ≈ 0). The advantage of the non-uniform algorithms over the uniform algorithms at about the same complexity is clearly shown in Figs. 9 and 10.

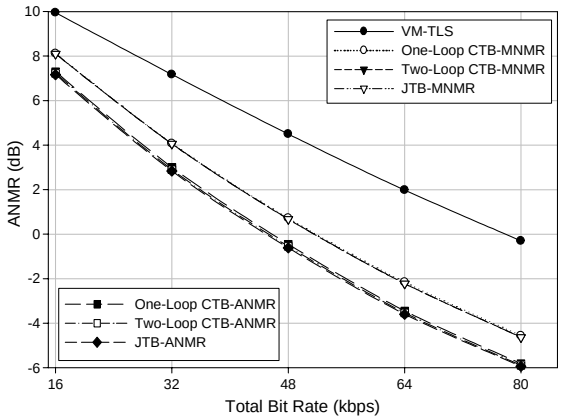


Fig. 7. ANMR Rate-Distortion Analysis

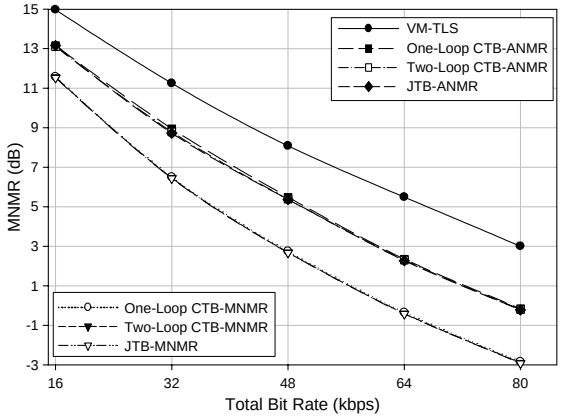


Fig. 8. MNMR Rate-Distortion Analysis

6.3. Subjective Quality Analysis

Listening test by human ears is the traditional method to subjectively evaluate the audio quality and is also the most recognized subjective quality test. However, such subjective test is expensive, time consuming, and difficult to reproduce. Informal listening tests on the aforementioned schemes show that it is hard to differentiate between JTB and various CTB schemes. In addition, a "*simulated*" subjective test, Objective Difference Grade (ODG), has been conducted. ODG is a measure of quality designed to be comparable to the Subjective Difference Grade (SDG). It is calculated based on the difference between the quality rating of the reference and test (coded) signals. The ODG has a range of [-4, 0], in which –4 stands for very annoying difference and 0 stands for imperceptible difference between the reference and the test signals [10][11]. The ODG results for various search schemes discussed in this paper are shown in Fig. 11 and the reference signal is the original audio sequence. According to the collected test data (Fig. 11), the difference between JTB and CTB schemes is quite small. The ODG results, which are relative to the CTB-MNMR scheme, for various fast searching algorithms are shown in Fig. 12. Again the performance of the non-uniform *NB_SF* algorithms is better than that of uniform *NB_SF* algorithms at about the same computational complexity.

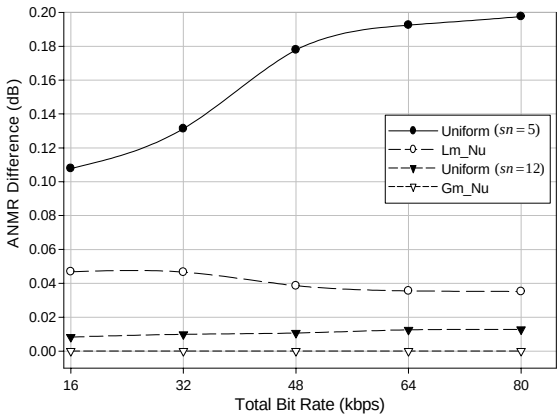


Fig. 9. ANMR Difference Analysis

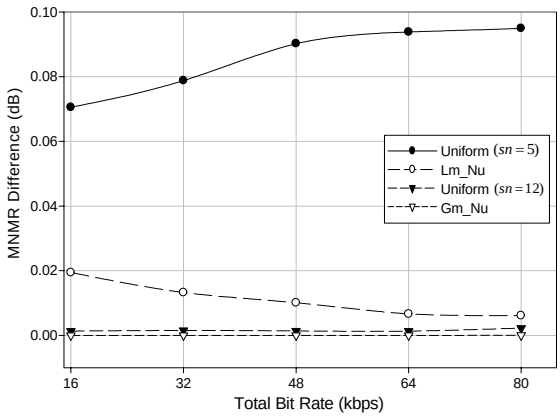


Fig. 10. MNMR Difference Analysis

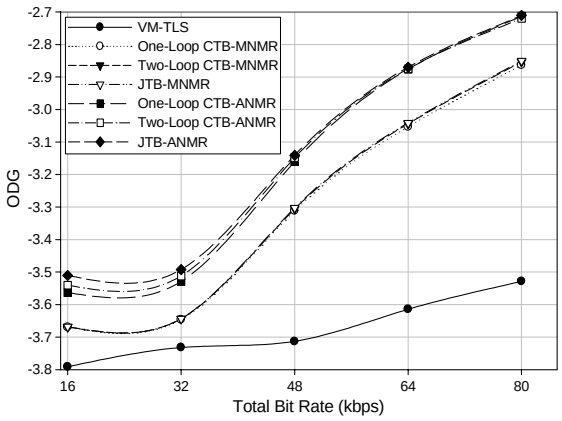


Fig. 11. ODG for VM-TLS, JTB and CTB

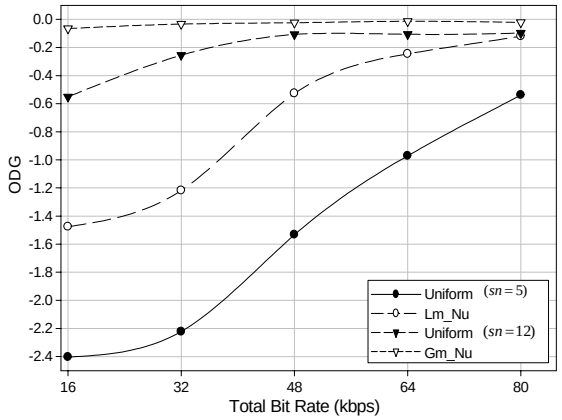


Fig. 12. ODG for Various Fast Searching Algorithms

7. CONCLUSIONS

In this paper, we propose a CTB optimization scheme for the MPEG-4 AAC coder, in which the optimization procedures for finding coding parameters, SF and HCB, are separated in two consecutive steps. Based on the complexity analysis, the proposed CTB scheme is approximately 71 to 142 times faster than the JTB scheme. Moreover, the simulation results show that both the objective and subjective quality of the CTB scheme is close to that of the JTB scheme. In addition, we also propose a lossless fast searching algorithm for trellis-based HCB optimization, which is about 4 times faster. Furthermore, two non-uniform searching algorithms, Gm_Nu and Lm_Nu, are proposed for trellis-based SF optimization. The simulation results show that the non-uniform searching algorithms can achieve better performance than uniform searching algorithms under the same complexity.

8. ACKNOWLEDGMENT

This work was supported by National Science Council, Taiwan, R.O.C., under Grant NSC-91-2219-E-009-011.

9. REFERENCES

[1] ISO/IEC JTC1/SC29, "Information technology – vary low bitrate audio-visual coding," *ISO/IEC IS-14496 (Part 3, Audio)*, 1998.

[2] M. Bosi, *et al.*, "ISO/IEC MPEG-2 advanced audio coding," Journal *of Audio Engineering Society*, vol. 45, pp. 789-814, October 1997.

[3] A. Aggarwal, *et al.*, "Trellis-based optimization of MPEG-4 advanced audio coding," *Proc. IEEE Workshop on Speech Coding*, pp. 142-4 2000.

[4] A. Aggarwal, *et al.*, "Near-optimal selection of encoding parameters for audio coding," *Proc. of ICASSP*, vol. 5, pp. 3269-3272, Jun 2001.

[5] T. V. Sreenvias and M. Dietz, "Vector quantization of scale factors in advance audio coder (AAC)," *Proc. of ICASSP*, vol. 4, pp. 3641-3644, May 1998.

[6] H. Najafzadeh and P. Kabal, "Improving perceptual coding of narrowband audio signals at low rates," *Proc. of ICASSP*, vol. 2, pp. 913-916, March 1999.

[7] H. Najafzadeh and P. Kabal, "Perceptual bit allocation for low rate coding of narrowband audio," *Proc. of ICASSP*, vol. 2, pp. 893-896, 2000.

[8] "The MPEG audio web page." http://www.tnt.unihannover. de/project/mpeg/audio.

[9] European Broadcasting Union, *Sound Quality Assessment Material: Recordings for Subjective Tests* Brussels, Belgium, Apr. 1988.

[10] Draft ITU-T Recommendation BS.1387: "Method for objective measurements of perceived audio quality," July. 2001.

[11] A. Lerchs, "EAQUAL software", Version 0.1.3 alpha, http://www.mp3-tech.org/

<附件三>

行政院國家科學委員會補助國內專家學者出席國際學術會議報告

92 年 6 月 5 日

報告人姓名	杭學鳴	服務機構及職稱	國立交通大學電子工程學系教授
會議時間地點	92 年 5 月 26 至 92 年 5 月 28 泰國曼谷市	本會核定補助文號	NSC 91-2219-E-009-011
會議名稱	(中文) 國際電子電機協會電路與系統會議 (英文) IEEE International Symposium on Circuits and Systems		
發表論文題目	(中文) 利用 TI TMS320C6201 數位訊號處理器實現即時 H.263+編解碼器 (英文) Real-Time Implementation of H.263+ Using TI TMS320C6201 Digital Signal Processor		

一、參加會議經過

本次 IEEE ISCAS 位於曼谷舉行，為期三日，本次參與會議主要是為了報告一篇關於利用數位訊號處理器實現 H.263+編解碼器的論文。曼谷距台彎約四個小時飛航時程，會議舉辦地點位於曼谷市中心 Imperial Queen's Park Hotel。此次會議約有一千八百篇來稿，約錄取 1200 篇，內容豐富。由於 SARS 疫情影響，各個報告會場的出席情況並不踴躍，甚至有出現流會的情況；而亦由於 SARS 的關係，自台灣方面來的出席者必須每三天做簡單的身體檢查，諸如測量體溫，及一些基本 SARS 徵狀檢測，但由於會場有常駐醫師，此保健措施並未造成不便。

ISCAS 為 IEEE Circuits and Systems Society 的年度最重大之學術會議。議題涵蓋廣泛，除了電路設計方面課題外，近年來無線通訊與多媒體亦為重要課題。由於本會具甚高學術地位，論文篩選頗為嚴格，論文入選為甚高榮耀。我多次擔任論文審查委員及分組會議(Technical Session)主席。會議期間亦同時召開 IEEE Circuits and Systems Society 各委員會，我目前在其中 Visual Signal Processing and Communications 委員會。

此次會議多媒體相關場次甚多，諸如 multimedia transmission, security and watermarking, MPEG-4/H.264 video coding, wavelets, motion estimation, video transcoding 等等。除了參加演講場次(lecture session)外，亦參觀了在會場的論文海報，針對幾項有興趣的主題，諸如多媒體與通訊相關，與作者討論，獲益良多。

此行主要目的是和博士班學生蔡家揚報告一篇利用數位訊號處理器實現 H.263+的論文，場次為在 28 日下午的低傳輸碼率視訊編碼(low bit-rate video coding

session) 。此外，另一篇與其他數位老師同學合作的論文 FGS-Based Video Streaming Test Bed for MPEG-21 Universal Multimedia Access with Digital Item Adaptation 由蔣迪豪教授學生報告。兩篇報告均十分順利，亦有多人提問，可見大家甚感興趣。

二、與會心得

這次的報告由於 SARS 影響，很明顯整體與會人數較往年會議減少，特別中國大陸專家更是幾乎無人前往，也由於人數少的關係，造成許多會議流會或是報告人數不足，頗為可惜，許多有興趣的題目也無法與論文作者討論。不過即使如此，從此次會議的論文亦可大致上看出近年幾個熱門的研究方向，諸如多媒體通訊以及無線通訊，另外生物相關方面研究也很熱門。

在多煤體方面，見到許多應用於無線傳輸應用，如結合通道編碼(Channel Coding)研究和可調式(scalable)編碼方式的研究。而新一代的視訊標準 H.264(AVC)也見到許多相關論文，由於 H.264 具有高復雜度與優良編碼效率(coding efficiency)特性，效能要較現有 MPEG-4 更好。而在無線通訊上也看到許多有趣的課題，這次所見的會場海報有許多與無線網路(wireless LAN)相關主題，多半是針對 802.11a 與 802.11b 的研究，也有少許與 802.11g 相關。在會場也看到一些關於無線網路的行動性(mobility)的探討，無線網路若具有良好行動特性，則實用性則大為提高。

學術研究課題隨時間而變，參加學術會議，可以了解最新學術研究趨勢與成果。其次，每回在中碰到許多過去的老師，朋友和工作同仁，互道短長，也是專業以外的重要收穫。

這次的 ISCAS 為近年辦得甚好的一次，不論是硬體設備，或是會場的招待，都可以體認到主辦單位的用心。可惜由於 SARS 的影響，影響出席意願。

三、攜回資料名稱及內容

這次會議提供一本手冊紀錄會議各個場次的時程，並附一片光碟記載會議論文。

REAL-TIME IMPLEMENTATION OF H.263+ USING TI TMS320C6201 DIGITAL SIGNAL PROCESSOR

K.-T. Shih, C.-Y. Tsai, H.-M. Hang
Department of Electronics Engineering
National Chiao Tung University
Hsinchu, Taiwan, R.O.C.
cytsai.ee90g@nctu.edu.tw, hmhang@mail.nctu.edu.tw

ABSTRACT

In this paper, we use a digital signal processor (DSP) to implement a real-time H.263+ codec. We use fast algorithms to reduce the codec computational complexity. Furthermore, the C programs are modified to take advantages of the DSP architecture and its C compiler features to reduce the on-chip memory and to increase the processing speed. In addition, a simple but effective rate-control algorithm is implemented to maintain the target bit rate. We can encode about 20 QCIF frames/second using one TI DSP. And the average decoding speed is about 26 QCIF frames/second.

1. INTRODUCTION

With the growing popularity of multimedia demand, video transmission over wireless network becomes very desirable in the near future. Because of the limited bandwidth of wireless channel, video signals have to be highly compressed. ITU-T H.263+ [1] is a specification for video compression targeting at very low bit-rate applications. Our codec is developed based on the H.263+ simulation software provided by Telenor Research and University of British Columbia [2].

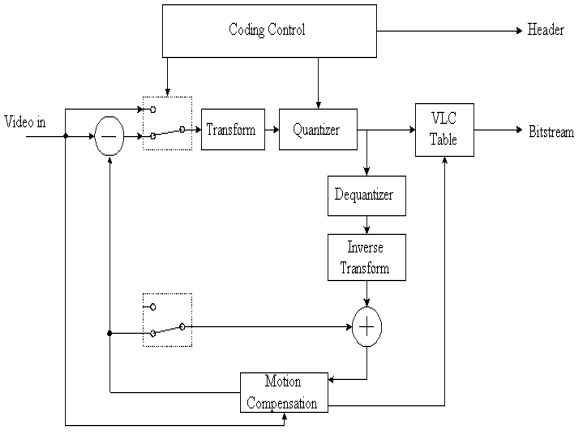


Figure 1. Block diagram of the basic H.263+ encoder [3].

Figure 1 shows the basic H.263+ encoder. The key elements are motion-compensated prediction, discrete cosine transformation (DCT), quantization, and variable-length coding (VLC). Motion-compensated prediction is used to remove temporal redundancy. The purpose of DCT and quantization is to reduce spatial redundancy. The VLC technique reduces syntax redundancy. All together, H.263 needs a very low bit rate, 64K bits/s or less, to transmit videophone type pictures. Because of high computational requirement of motion-compensated prediction and DCT, we use fast motion search algorithms and fast discrete cosine transforms to achieve real-time implementation.

The TI TMS320C62xx fixed-point DSP has a rather good performance. Its instruction cycle time is 5 ns (200 MHz clock). It adopts the advanced VelociTI very long instruction word (VLIW) architecture that enables sustained throughput of up to eight instructions in parallel and thus allows the processor running much faster. Therefore, the maximum computation power is 1600 million instructions per second (MIPS).

This paper is organized as follows. Section 2 describes the fast algorithms we use, diamond search, DIF DCT. Rate control algorithm is described in Section 3. In Section 4, the DSP implementation techniques and results are presented. Finally, a summary is given in the last section.

2. FAST ALGORITHM

2.1 Diamond Search

The full search algorithm for motion estimation examines all search points inside the search area. Therefore, the amount of its computation is proportion to the size of the search area. Although it finds the best possible match, it requires a very large computational power. Hence, many fast algorithms are proposed to reduce computation at the price of slightly performance loss. The basic principle of these fast algorithms is dividing the search process into a few sequential steps and choosing the next search direction according to the current search result [6].

The diamond search algorithm [7] starts with zero-vector candidate. Then, it moves to the most promising search point and does another search after the current step is completed. This procedure is repeated until it cannot move further and the local optimum is reached. The diamond search assumes that the matching function is monotonic along any direction away from the optimal point. But in reality the monotonic matching function assumption is sometimes invalid and thus the diamond search algorithm is suboptimal [6].

The procedure of the diamond search is described below.

Step 1: Compute the sum-of-absolute-difference (SAD) between the current macroblock and the same location macroblock in the previous reconstructed frame, called SAD0. This value is the prediction error when the current macroblock is

predicted using the zero-vector. Then, we set the current best vector (0,0) to be the search center.

Step 2: Four search points are chosen to center around the current best vector in a shape of diamond as shown in Figure 2. Their locations are (-1,0), (0,1), (1,0), (0,-1), respectively. Then, we compute the SAD of every search point and compare them to SAD0. If SAD0 is the minimum, the center represents the best motion vector, stop; otherwise, continue.

Step 3: Move the search center to the best vector that has the minimum SAD computed in Step 2 and replace SAD0 with this SAD value. Then, go to Step 2.

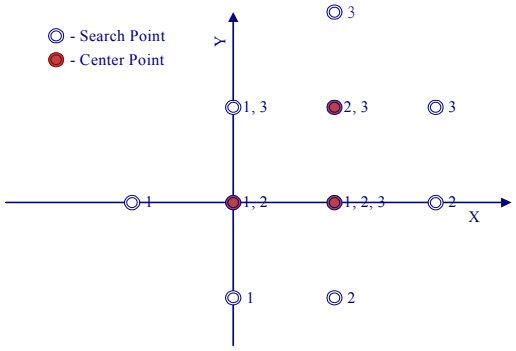


Figure 2. An example of diamond search procedure [11]

Figure 2 shows an example of the diamond search. Each stage contains four search points and a central point. They are all labeled by the same number. Three stages are presented in Figure 2. The central point moves to the right in the second stage. Then, it moves to the top in the third stage.

2.2 Decimation-In-Frequency (DIF) DCT

The original DCT algorithm in the software uses the sparse matrix factorizations. It requires 26 real additions and 16 real multiplications for an 8-point DCT. The amount of computation is not large, but this process is non-recursive and needs a complicated index mapping [8]. These two disadvantages leads to a serious delay in DSP implementation. Therefore, we need another fast DCT algorithm. The DIF DCT algorithm requires about the same number of additions and multiplications comparable to the original algorithm. But its process is recursive and needs only a moderate index mapping. So it fits better to the chosen DSP.

The key concept of the DIF DCT algorithm is to divide an N-point DCT to two N/2-point DCTs by rearranging of the input samples in the frequency domain. This concept results in a desirable recursive modularity of a fast decomposition. In our case, an 8-point DCT is decomposed into two 4-point DCTs. This method divides a DCT process into two parts with equal computing load when implemented on the DSP. Therefore, the DSP compiler can optimize the for-loop with ease. Moreover, this algorithm also can save three-fourths of the DCT code size. A 2-D 8-point DCT is obtained by computing two 1-D 8-point DCT along horizontal and vertical axes. The forward and inverse DCT used are specified in [8].

Because our DSP arithmetic unit uses fixed-point operations, floating-point computations are very inefficient when we implement DCT on this DSP. Therefore, we convert this floating-point algorithm to a fixed-point one. We multiply C_k by 2^{14}. Then, we round these values to their nearest integers. We use the new C_k to compute DCT and then divide the final DCT coefficients by 2^{14}. The multiplication and division of 2^{14} are realized simply by left-shift and right-shift by 14 bits on the DSP. It increases nearly no additional load. However, this fixed-point algorithm has the accuracy problem.

H.263+ Annex A specifies the desirable inverse transform accuracy [1]. We use it to examine the fixed-point DIF DCT, the floating-point DIF DCT, and the fixed-point original DCT. (We modify the original DCT into the fixed-point DCT in a similar manner as described in the above.) The results are shown in Tables 1 to 3. Annex A specifies that the overall mean error should not exceed 0.0015 in magnitude and the overall mean square error should not exceed 0.02. The floating-point DIF DCT satisfies this specification, but the fixed-point DIF DCT and the fixed-point DCT cannot meet the specification due to rounding errors in the fixed-point computing process. The errors are accumulated and propagated to the next frames in the interframe coding. Therefore, frames become blurred if we encode a number of P-frames (inter-coded frames).

Data Range	Overall Mean Absolute Error	Overall Mean Square Error
L=256,H=255	0.861547	1.589669
L=5,H=5	0.854330	1.545261
L=300,H=300	0.734516	1.354213

Table 1. The accuracy of the fixed-point DIF DCT

Data Range	Overall Mean Absolute Error	Overall Mean Square Error
L=256,H=255	0.000014	0.000014
L=5,H=5	0	0
L=300,H=300	0.000011	0.000011

Table 2. The accuracy of the floating-point DIF DCT

Data Range	Overall Mean Absolute Error	Overall Mean Square Error
L=256,H=255	1.262053	3.570425
L=5,H=5	1.222489	3.271673
L=300,H=300	1.076280	3.047527

Table 3. The accuracy of the fixed-point DCT

Figure 4 shows this situation. Figure 4 depicts the 100-th P-frame of “Salesman” encoded using different DCTs. The floating-point DCT is used in (a), while the fixed-point DIF DCT is used in (b). Clearly, the salesman’s right hand and the box blur more significantly in (b). The errors are generated in the I-frame stage and then are diffused by motion vectors into the P-frames. Hence, the moving part has the strong blurred effect. In order to avoid this effect, we force our encoder to perform one INTRA-frame coding for every ten P-frames. Consequently, the DCT errors are not accumulated large enough to distort the video quality. There is another solution stated in the Annex W [5]. The fixed-point DCT specified in Annex W has a better accuracy, but it, on the other hand, increases computations significantly. In practice, to eliminate transmission error propagation in particularly wireless environment, frequent frame refresh or intra-frame coding is necessary. Intra-coding strategy is thus adopted.

(a)

(b)

Figure 3. The 100-th P-frame of "Salesman" encoded by using (a) floating-point DCT and (b) fixed-point DIF DCT.

3. RATE CONTROL

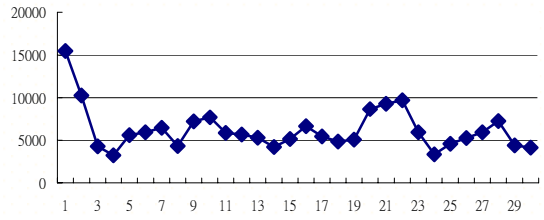
To maintain a constant output bitrate, a rate control algorithm must be used in our encoder. There are two major considerations: the delay produced by bits accumulated in the encoder buffer and the bit allocation issue, which affect the coded picture quality. In addition, including a rate control into our encoder implementation requires a significant amount of additional computation and memory (for program and data). Hence, we choose a less complicated rate control algorithm described in TMN5 [12].

At the beginning, we have a target frame rate f_{target} and a target bitrate R . Consequently, we can calculate the target number of bits per picture B. When finishing encoding one frame, we obtain $\Delta_1 B$, which is the difference of bits spent on the coded picture and the target bits B at frame level. If the available bits per frame (B) are distributed uniformly over all macroblocks, we know the target number of bits per macroblock. Thus, we can calculte $\Delta_2 B$, which is the difference of coded bits spent on a macroblock and its target bits at macroblock level. Essentially, we use these two values, $\Delta_1 B$ and $\Delta_2 B$ to adjust the quantization parameters to allocate bits based on the following formula [12]:

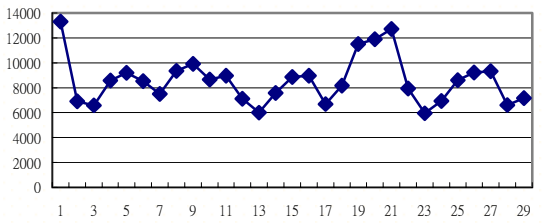
$$QP_{new} = QP_{i-1}\left(1 + \frac{\Delta_1 B}{2B} + \frac{12\,\Delta_2 B}{R}\right),$$

where QP_{i-1} is the quantization stepsize for the previous frame and QP_{new} is the quantization stepsize for the current macroblock to be coded. Moreover, we need to take care of the delay due to buffering and maintain the buffer fullness as near constant as we can. Therefore, if the buffer fullness exceeds a threshold, the encoder will skip the next frame until it goes down to the acceptable region.

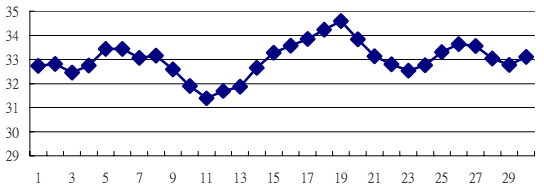
Figure 4 shows the simulation results using the aforementioned (TMN5) rate control algorithm. In this experiment, the target bit rate is 53kbps, and the buffer threshold is 8kbps for skipping frames. Because one intraframe is enforced for every ten frames, the bit rate is somewhat higher at multiples of the tenth frame. Also, the PSNR values are lower for these frames since intraframe coding requires more bits. Although the frame coding mode (inter/intra) is changed regularly, the output bit rate and the buffer level remain nearly constant. In general, the rate control mechanism works quite well.



(a)



(b)



(c)

Figure 4. QCIF foreman sequence encoded with TMN5 rate control (a) bits used per frame, (b) encoder buffer fullness, and (c) PSNR per frame.

4. DSP IMPLEMENTATION

We implement our system on the Blue Wave Systems PCI/C6600 applications board. It provides a software support library for data transfer between the host PC and the DSP. This library is called "Generic Host Interface Library" (GenrHL) [10]. It includes two types of data transfer: "message system" and "mailbox interrupt". The former one provides efficient bulk of data transfer between host PC and DSP. The latter one provides interrupt between host PC and DSP and is useful for synchronization. Besides, This board has two TI TMS320C6201 fixed-point processors on it. Each DSP has 64kbytes internal program memory, 64kbytes internal data memory, and 16Mbytes external SDRAM. We will describe our implementation in detail in the next subsection.

4.1 Optimization for DSP Architecture

- ***Intrinsic operator:*** The C6000 compiler provides intrinsics, special functions that map directly to inlined C62x instructions to optimize C codes. All instructions that are not easily expressed in C codes are supported as intrinsics [13][15]. For example, we can use the intrinsic operator "_abs" to calculate the saturated absolute value.
- ***Wider memory access for smaller data widths:*** In order to maximize data throughput, it is often desirable to use a

single load or store instruction to access multiple data values consecutively located in memory. For example, C6x have instructions with associated intrinsics, such as "_add2()", "_mpyhl()", "_mpylh()", etc, that operate on 16-bit data stored in the high and low parts of a 32-bit register. When operating on a stream of 16-bit data, we can use word accesses to read two 16-bit values at a time, and then use another C6x intrinsics to operate on the data [13].

- ***Memory management:*** TI TMS20C6201 DSP has only totally 128kbytes internal memory. Therefore, memory management becomes very important. In program memory management, we delete unused codes and re-write some functions to reduce the program code size. Furthermore, we use the compiler options to optimize the execution speed. In data memory management, we put all dynamic allocated memory sections into the external SDRAM and put frequently used data in the internal data memory.

4.2 Implementation Results

We replace the motion vector search algorithm and the DCT algorithm with the fast algorithms described before. In addition, we optimize the program using the optimization techniques described in the previous section. The final results are shown in Table 4. Our code size is about 46kbytes without rate control and 58kbytes with it.

	Original	Optimized
I- frame coding	67.48MIPS	2.89MIPS
P-frame coding	229.52MIPS	6.44MIPS

Table 4: Encoding speed comparison between the original source codes and the optimized codes.

On the other hand, the decoder is easy to implement on DSP as compared to the encoder. Hence, we do not describe in details on its implementation. We simply give the final results here. The average decoding speed is about 26 QCIF frames per second, and the code size is about 27kbytes.

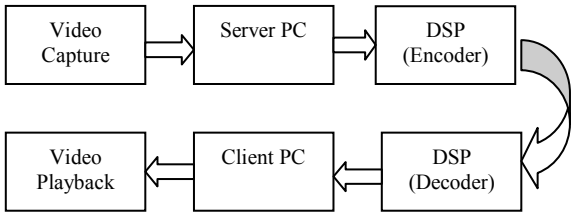


Figure 5. System block diagram of the pipeline-like structure.

4.3 Multitasking System

Multitasking programming for operating system means the computation power of CPU is not dedicated to a single application but can be distributed to multiple tasks simultaneously [14]. In our system design, each block in Figure 5 is an individual thread. Therefore, the system has a pipeline-like structure. Furthermore, each interface between different blocks requires a synchronization object to control the data flow. This server-client based system can achieve about 14 QCIF frames per second encoding and decoding speed.

5. SUMMARY

In this paper, we implement the basic H.263+ encoder using the TI TMS320C6201. To reduce the computation load, we bring in the diamond search algorithm and the fixed-point DIF DCT algorithm. However, the fixed-point DIF DCT contains the rounding errors, which affect the video quality. One simple and practical approach to reduce this effect is forcing one intra frame per ten or so inter-coded frames. In addition, we modify our C codes to take the advantages of the TI DSP architecture and the compiler's features. At the end, we can encode about 20 QCIF frames/second using one TI DSP. And the average decoding speed is about 26 QCIF frames/second.

6. REFERENCES

[1] ITU-T Study Group 16, *Video Coding for Low Bit Rate Communication*, 1998.

[2] Telenor research official ftp site: ftp://bonde.nta.no/pub/tmn/software.

[3] C. Côté, B.Erol, M.Gallant, and F.Kossentini "H.263+: Video Coding at Low Bit Rates". *IEEE Trans. Circuit Syst. Video Technol*., vol. 8, no. 7, Nov 1998, pp. 849-866.

[4] Texas Instruments "TMS320C6x Technical Brief." *Texas Instruments*, 1999.

[5] ITU-T Study Group 16, *H.263 Draft Annex U, V, and W*, Feb. 2000.

[6] H.-M. Hang and J. W. Woods, *Handbook of Visual Communications*, Academic Press, 1995

[7] ITU-T Study Group 16, *Video Codec Test Model, Near-Term, Version 8* (TMN8), 1997.

[8] K. R. Rao and R. Yip, *Discrete Cosine Transform- Algorithms, Advantages, Applications*, Academic Press, 1990.

[9] Blue Wave Systems, *Blue Wave Systems PCI/C6600 Applications Board Technical Reference Manual*, 1999

[10] Blue Wave Systems, *Blue Wave Systems Host and MPC860 C GenrHL User Guide for C6x Boards*, 1999

[11] M. L. Woo, *Real-Time Implementation of H.263+ Using TI TMS320C62xx*, MS Thesis, Institute of Electronics, National Chiao Tung University, June 2000.

[12] ITU-T Study Group 15, *Video Codec Test Model, TMN5*, Jan. 1995.

[13] Texas Instruments "TMS320C6000 Programmer's Guide" *Texas Instruments*, 1999

[14] J. R. Woo, *DSP-based Real-Time H.263 Encoding /Decoding and Transportation for Video Conferencing*, MS Thesis, Institute of Electronics, National Chiao Tung University, June 2000.

[15] Texas Instruments, *C Implementation of the TMS320C62xx Intrinsic Operators*. Texas Instruments, Application Report, Literature Number: SPRA616, Dec. 1999.