

MPEG-2 音频实时压缩编解码的一种快速算法¹

刘常澍 马 欣 李文元 张毓忠

(天津大学电子信息工程学院 300072)

摘 要 该文介绍采用一片 TI 公司的数字信号处理芯片 TMS320C31 实现了 MPEG-2 音频 Layer-1,2 实时压缩编解码器。为了达到实时的目的,对 MPEG 建议的子带分析和子带合成方案分别提出了一种新的快速算法,采用该算法的运算量分别是 MPEG 标准建议算法运算量的 1/5 和 1/10。所有算法都经过了软件模拟和硬件实时仿真,通过仿真器装载到一片 TMS320C31 上实现了实时编解码运算。

关键词 子带分析滤波,子带合成滤波,心理声学模型

中图分类号 TN911.31

1 概 述

MPEG-2 音频压缩编码标准已被 DAB(digital audio broadcast)等消费电子领域广泛采纳。它以心理声学理论为基础,充分利用人耳的听觉特性,不传送人耳感觉不到的信息,消除信号中的不相干成分,从而达到压缩编解码的目的。MPEG 音频编码方案是一种子带编码,它按帧处理数据,其帧结构如图 1 所示。Header 包含了一帧的所有控制信息,如同同步字、采样频率、编码比特率等;比特分配信息包含了 32 个子带的音频样值各自所用的编码比特数;音频数据部分是各子带音频样值的编码。



图 1 MPEG 音频帧结构

MPEG 建议的音频压缩编解码方案中,子带分析/合成滤波的运算量很大,按照标准建议的算法,要想实现实时运算,就必须采用多片 TMS320C31 或采用更高速度的 DSP(digital signal processor)芯片,增加硬件设计的复杂度及成本。我们对 MPEG 建议的子带分析和子带合成方案分别提出了一种新的快速算法,采用该算法可以把子带分析滤波的运算量降低到原来的 1/5,子带合成滤波的运算量降低到原来的 1/10,从而实现只采用一片 TMS320C31 就能实时地完成 MPEG-2 从子带分析、音频模型计算到帧打包的压缩编解码全过程,简化了硬件设计。

2 MPEG-2 音频编解码的实时实现及其快速算法

MPEG 音频压缩编解码过程如图 2 所示。

在编码部分,先将每一帧输入的 PCM (pulse coded modulate) 信号进行子频带分析,将宽频带的音频信号分成 32 个等带宽的子频带,然后利用心理声学模型计算出每个子频带的信掩比 (SMR, signal mask ratio),根据 SMR 对每个子带进行动态比特分配,即对人耳听觉敏感的子频带分配较多的编码比特,对人耳听觉不敏感的区域分配较少编码比特或不分配

¹ 1999-06-24 收到, 1999-11-25 定稿

国家自然科学基金资助项目

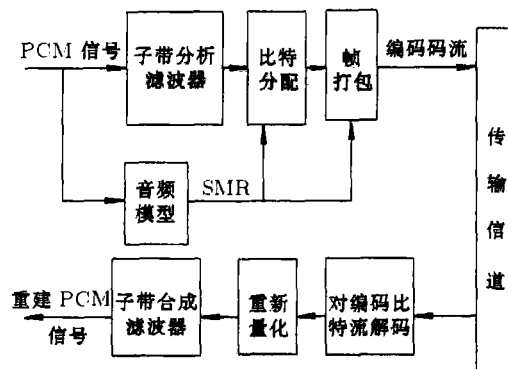


图 2 MPEG 音频压缩编解码过程

比特。在每一个子频带中分别量化、编码，然后按照标准帧的格式组成一帧，发送出去。在数据接收端，解码器根据接收到的比特流，找出同步信息按帧对齐，根据 Header、比特分配信息对音频数据进行重新量化后，经子带合成滤波恢复出原始 PCM 信号。

实现 MPEG-2 实时压缩编解码，耗时最多的是子带分析 / 合成滤波。对 32 个音频样点进行处理就要做约 $32 \times 64 = 2048$ 次带余弦的乘加运算，运算量非常大，目前已提出了几种快速算法 (参考文献 [3, 4])。下面我们提出一种新的快速算法，这种算法利用了余弦对称性质和 FFT，比文献 [3, 4] 中所建议的算法节省大约一半的运算时间。

2.1 子带分析滤波的快速算法 MPEG 建议的子带分析滤波方案如图 3 所示。其中， X 为 512 点的分析矢量； C 为 512 点的分析窗系数； Y 为 64 点的中间矢量； M_{kn} 为矩阵变换系数， $M_{kn} = \cos[(2k+1)(n-16)\pi/64]$ ； S 为计算出的 32 点频域样值。这种方案采用了临界采样、余弦调制多相滤波器组，把时域样值转换成频域的子带值。它的优点有：(1) 时间分辨率高，为 $8\text{ms}/12=0.67\text{ms}$ 。(2) 块边界效应小。缺点是运算量大，如果采样频率为 48kHz ，那么处理一帧 1152 个样值的可用时间为 $1152/48\text{kHz}=24\text{ms}$ 。

如果用一片晶振为 60MHz 的 TMS320C31 运算，它的运算速度为 30MIPS (每秒执行百万条指令)，对一帧 1152 个样值进行子带分析滤波要耗时 75.9ms ，要想达到实时 MPEG-2 编码，就必须采用多片 TMS320C31 或更高速度的 DSP 芯片。子带滤波快速算法的关键是 MDCT(modified discrete cosine transform):

$$S[k] = \sum_{n=0}^{63} Y[n] \cos[(2k+1)(n-16)\pi/64], \quad k = 0, \dots, 31 \quad (1)$$

的快速算法，使 (1) 式的 MDCT 变换的循环次数降低，就是子带分析快速算法的关键。

首先，(1) 式可以写为如下形式

$$S[k] = \sum_{n=-16}^{47} Y[n+16] \cos[(2k+1)n\pi/64], \quad k = 0, \dots, 31 \quad (2)$$

由余弦函数的性质:

$$\cos[(2k+1)(n-16)\pi/64] = -\cos[(2k+1)(n-16+64)\pi/64] \quad (3)$$

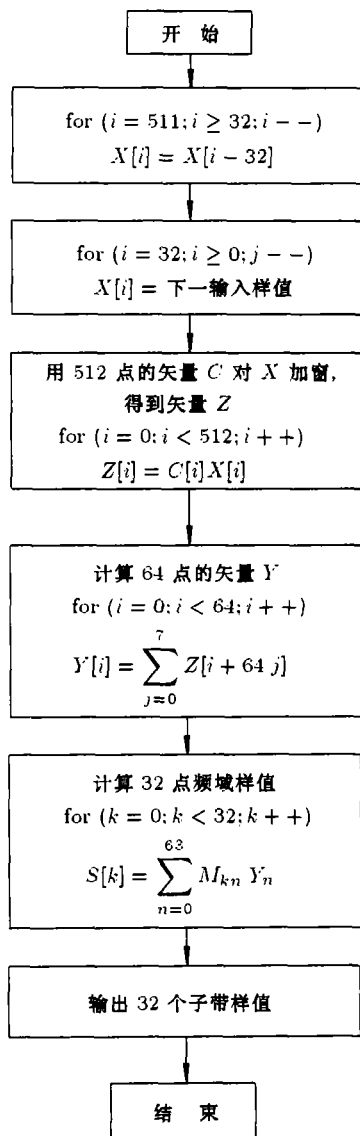


图 3 MPEG 子带分析滤波器流程

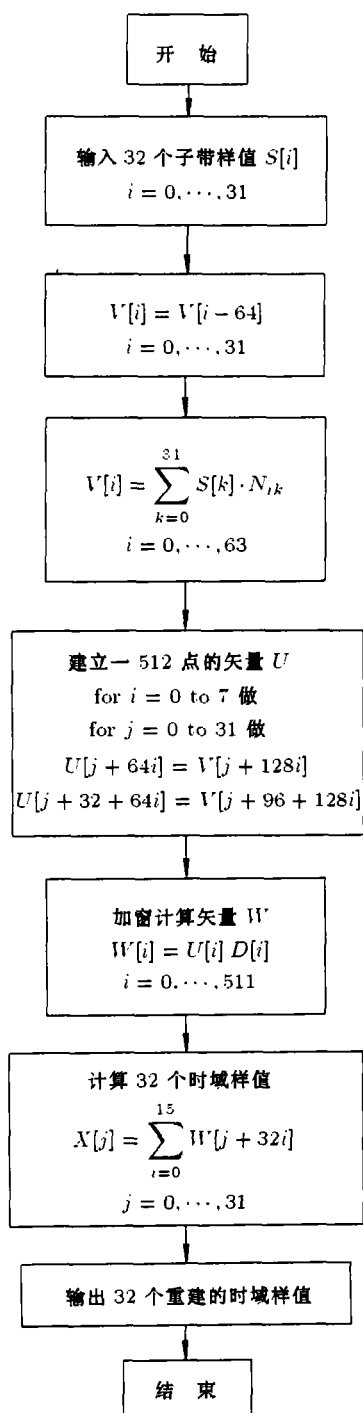


图 4 子带合成滤波算法流程

将 $Y[n]$ 做如下移位, 得到一新的序列 $X_1[n]$:

$$X_1[n] = Y[n + 16], \quad n = 0, \dots, 47 \quad (3a)$$

$$X_1[n] = -Y[n - 48], \quad n = 48, \dots, 63 \quad (3b)$$

那么, 就得到 $S[k]$ 的如下表示

$$S[k] = \sum_{n=0}^{63} X_1[n] \cos[(2k+1)n\pi/64], \quad k = 0, \dots, 31 \quad (4)$$

注意到 (4) 式中的 $\cos[(2k+1)n\pi/64]$ 是 $e^{-j\pi(2k+1)n/64}$ 的实数部分, 对 (4) 式做如下变换:

$$\begin{aligned} S[k] &= \sum_{n=0}^{63} X_1[n] \cos[(2k+1)n\pi/64] \\ &= \operatorname{Re} \left\{ \sum_{n=0}^{63} X_1[n] e^{-j\pi(2k+1)n/64} \right\} \\ &= \operatorname{Re} \left\{ \sum_{n=0}^{63} X_1[n] e^{-jn\pi/64} e^{-j2\pi kn/64} \right\}, \quad k = 0, \dots, 31 \end{aligned} \quad (5)$$

因为 $e^{-jn\pi/64}$ 只与 n 有关, 而与 k 无关, 可以把 $X_1[n]e^{-jn\pi/64}$ 看成为一个 64 点的复数序列, 则 (5) 式就是一个标准的 64 点 DFT 形式, 可以通过快速傅里叶变换 (FFT) 进行计算, 取其结果的实数部分, 就得到 32 个 $S[k]$ 值, 运用这种算法计算 32 个 $S[k]$ 值, 所需的乘法运算次数为 $2 \times 32 \times \log_2 64 = 384$ 次, 加法运算次数为 $4 \times 32 \times \log_2 64 = 768$ 次, 用一片 C31 对一帧 1152 个音频样值进行子带分析, 约需 12.8ms.

子带分析滤波快速算法的步骤如下:

- (1) 向 512 点的分析矢量 X 输入 32 个新的音频样值, 舍弃 32 个旧的样值.
- (2) 利用矢量 C 对 X 加窗, 得到矢量 Z :

$$Z[n] = C[n]X[n], \quad n = 0, \dots, 511 \quad (6)$$

- (3) 计算 64 点的中间变量 Y

$$Y[n] = \sum_{j=0}^7 Z[n + 64j], \quad n = 0, \dots, 63 \quad (7)$$

- (4) 对矢量 Y 进行移位处理

$$X_1[n] = Y[n + 16], \quad n = 0, \dots, 47; \quad X_1[n] = -Y[n - 48], \quad n = 48, \dots, 63. \quad (8)$$

- (5) FFT 预处理

$$X_2[n] = X_1[n]e^{-jn\pi/64}, \quad n = 0, \dots, 64 \quad (9)$$

- (6) 做 64 点 FFT

$$X_3[k] = \sum_{n=0}^{64} X_2[n]e^{-j2\pi kn/64}, \quad k = 0, \dots, 32 \quad (10)$$

- (7) 取 X_3 的实数部分, 即为子带分析的结果:

$$S[n] = \operatorname{Re}\{X_3[n]\}, \quad n = 0, \dots, 32 \quad (11)$$

(8) 输出 32 点子带样值。

2.2 子带合成滤波的快速算法 子带合成滤波是子带分析滤波的反过程, 它一次把频域的 32 个子带样值恢复成 32 个时域 PCM 样值输出, MPEG 标准建议的子带合成滤波器方案如图 4 所示。其中, S 为 32 个子带样值序列; V 为 1024 点的合成矢量, V 每次移位 64 点, 丢弃 64 点旧数据, 移入矩阵运算 $V[i] = \sum_{k=0}^{31} S[k]N_{ik}$, 求得的 64 点新数据, N_{ik} 是子带合成矩阵系数 $N_{ik} = \cos[(2k+1)(i+16)\pi/64]$; U 是由 V 得到的一个 512 点的矢量; D 为 512 点的合成窗函数; X 为恢复出的 32 点时域 PCM 样值。

类似于子带分析滤波, 子带合成滤波算法关键是 IMDCT(inverse modified discrete cosine transform)

$$V_i = \sum_{k=0}^{31} S[k] \cos[\pi(16+i)(2k+1)/64], \quad i = 0, \dots, 63 \quad (12)$$

首先考察 V_i , 它有如下性质:

$$V(16) = 0 \quad (13a)$$

$$V(16-i) = -V(16+i), \quad i = 0, \dots, 15 \quad (13b)$$

$$V(48-i) = V(48+i), \quad i = 0, \dots, 15 \quad (13c)$$

也就是说, 64 点的矢量 V 中, 有一半数据可以由另一半数据来表示, 只求出其中的 32 点就可以得到整个 64 点的 V_i 值。

为了能够对 (12) 式使用 FFT 变换, 设 $S_{32} \sim S_{63}$ 为 0, 于是得到一个 64 点的序列 $S[k]$, 那么 V_i 式可以写为下式:

$$\begin{aligned} V_i &= \sum_{k=0}^{63} S(k) \cos[\pi(16+i)(2k+1)/64] \\ &= \operatorname{Re} \left\{ \sum_{k=0}^{63} S(k) e^{-j\pi(16+i)(2k+1)/64} \right\}, \quad i = 0, \dots, 63 \end{aligned} \quad (14)$$

由于对 i 的移位 16 只与结果矢量 V 的移位有关, 所以抽取矢量 G_i 进行分析:

$$G_i = \operatorname{Re} \left\{ \sum_{k=0}^{63} S(k) e^{-j\pi i(2k+1)/64} \right\} = \operatorname{Re} \left\{ \sum_{k=0}^{63} S(k) e^{-j2\pi ki/64} e^{-j\pi i/64} \right\}, \quad i = 0, \dots, 63 \quad (15)$$

(15) 式中, $e^{-j\pi i/64}$ 与 k 变化无关, 所以有

$$G_i = \operatorname{Re} \left\{ e^{-j\pi i/64} \sum_{k=0}^{63} S(k) e^{-j2\pi ki/64} \right\}, \quad i = 0, \dots, 63 \quad (16)$$

对 $S[k]$ 做 64 点 FFT 变换, 然后进行后处理, 乘以 $e^{-j\pi i/64}$ 就得到 G_i , 对 G_i 移位得到矢量 V 。

考虑到余弦函数的对称性

$$\cos[2\pi(16+i)(2k+1)/64] = \cos[2\pi(16+i)(64-2k-1)/64] \quad (17)$$

所以有

$$G_i = \text{Re} \left\{ e^{-ji/64} \sum_{k=0}^{63} [S(2k) + S(64 - 2k - 1)] e^{-j\pi ki/32} \right\}, \quad i = 0, \dots, 31 \tag{18}$$

这样, 把 $S[2k] + S[64 - 1 - 2k]$ 组成一个新的 32 点的序列, 对它做 32 点的 FFT 变换, 求出矢量 G 的 32 个点。如前所述, G 和 V 的关系是偏移 16 位的关系, 并且根据 (13a), (13b), (13c) 式的对称性, 得出 V_i 的分段表示:

$$\begin{aligned} V[i] &= -G[i + 16], \quad i = 0, \dots, 15 \\ V[16] &= 0 \\ V[i] &= G[48 - i], \quad i = 17, \dots, 48 \\ V[i] &= G[i - 48], \quad i = 49, \dots, 63 \end{aligned} \tag{19}$$

采用这种算法, 对 32 个子带样值进行子带合成只需做 32 点的 FFT 变换, 乘法运算次数为 $2 \times 16 \times \log_2 32 = 160$ 次, 加法运算次数为 $4 \times 16 \times \log_2 32 = 320$ 次。用一片 C31 对 1152 个子带样值进行子带合成耗时约 6.5ms。

子带合成滤波快速算法的步骤从略。

3 结 论

我们先用 C 语言编程实现了 MPEG-2 压缩编解码算法, 然后用 TMS320C31 系列浮点编译器把 C 语言转成输出代码, 通过仿真器装载到 C31 上执行。但由于系统实时性要求高, C 编译器效率低, 不能达到实时要求, 所以采用了汇编语言编程实现 MPEG-2 压缩编解码算法。表 1 和表 2 是用汇编语言编写的编解码算法在 TMS320C31 上执行时处理一帧数据各部分占用的运算时间。当选择最高采样率为 48kHz 时, 处理一帧数据的可用时间为 24ms。从表中我们可以看到, MPEG-2 编码过程总耗时为 20.36ms, 解码过程总耗时为 7.26ms, 均在一帧可用时间之内, 可以达到实时编解码运算。

表 1 MPEG-2 编码过程执行时间

任务	子带分析	音频模型	比特分配	量化	帧打包	总计
执行时间 (ms)	12.8	4.7	1.29	0.65	0.92	20.36

表 2 MPEG-2 解码过程时间

任务	重新量化	子带合成	总计
执行时间 (ms)	0.76	6.5	7.26

参 考 文 献

[1] MPEG,"ISO/IEC 13818-3, Generic coding of moving pictures and associated audio, part3, Nov, 1993.

[2] T. Imai, J. Shiokaxa, H. Chiba, MPEG-1 audio real-time encoding system, IEEE Trans. on Consumer Electronics, 1998, CE-44(3), 888-894.

[3] Chen Yu-Chi, Tsai Chien-Wu, Fast time-frequency transform and their applications to real-time software impenentation of AC-3 audio CODEC, IEEE Trans. on Consumer Electronics, 1998, CE-44(2), 413-423.

- [4] Masahiro Iwadare, Akihiko Sugiyama, A 128 kB/S HI-FI audio CODEC based on adaptive transform coding with adaptive block size MDCT, IEEE J. on Selected Areas in Communications, 1992, 44(2), 138-144.
- [5] Chen Jing, Tai Heng-Ming, Real-time implementation of the MPEG-2 audio CODEC on a DSP, IEEE Trans. on Consumer Electronics, 1998, CE-44(3), 866-871.

A FAST ALGORITHM FOR REAL-TIME MPEG-2 AUDIO CODING AND DECODING

Liu Changshu Ma Xin Li Wenyuan Zhang Yuzhong

(School of Electronics Information Engineering, Tianjin University, Tianjin 300072, China)

Abstract A real-time MPEG-2 layer1,2 audio CODEC realized on one TMS320C31 chip, which is a new DSP product of TI, is developed in this paper. In order to realize code & decode in real-time, new fast algorithms for MPEG-2 audio sub-band analysis filter and sub-band synthesis filter are presented respectively. Using these fast algorithms, the computational load of sub-band analysis/synthesis can be reduced to 1/5 and 1/10 that of the original algorithms respectively. All the algorithms have been simulated on software and hardware. Through a simulator, the programmes are loaded to a TMS320C31 and realize code and decode in real-time.

Key words Sub-band analysis filter, Sub-band synthesis filter, Psychology acoustics

刘常澍: 男, 1946 年生, 副教授, 现主要从事电路与系统的研究.
马欣: 女, 1971 年生, 教师, 现主要从事数字音频编解码的研究.
李文元: 男, 1946 年生, 副教授, 现主要从事语言处理的研究.
张毓忠: 男, 1936 年生, 教授, 现主要从事数字音频编解码的研究.