고정소수점 연산구조에 기초한 MPEG-4 CELP coder 구현

이 우 종, 이 재 식, 박 지 태, 장 태 규, *이 전 우
중앙대학교 전자전기공학부. *한국전자통신연구원

A Fixed-point implementation of MPEG-4 CELP coder

W.J. Lee, J.S. Lee, J.T Park, T.G. Chang, *J.W Lee
School of Electrical Engineering Chung-Ang University, *ETRI
E-mail : woojong74@hanmail.net

Abstract

본 논문에서는 음성압축 앨고리즘인 MPEG-4 CELP coder를 16 bit DSP 구현에 필요한 고정소수점 연산구조로 구현하였다. 기본 앨고리즘 중에 LSP 계수를 구하는 방법인 Chebyshev series method 대신 고정소수점 구현에 유리한 Real root method 앨고리즘을 사용하였다. 또한 cosine, log 등 DSP 명령어가 지원하지 않는 수학 함수들은 미리 계산하여 테이블 적용기법을 사용하였고 고정 소수점 연산에 불리한 나눗셈 연산을 최대한 배제하였다. 고정 소수점 연산 구조로 변환한 후 부동 소수점 연산구조와의 비교를 통하여 오차를 최소화하도록 하였다. 구현한 음성코더를 남,여 각 5문장에 적용했을 때 부동 소수점 연산구조에 비교해 음질의 열화가 없음을 확인하였다.

I. 서론

음성 압축 앨고리즘은 이동 멀티미디어 단말 시스템 개발에 있어 필수적이며 이를 DSP 프로세서 혹은 RISC 프로세서로 구현하는 것이 일반적인 추세이다. 이에 본 논문에서는 멀티미디어 단말을 위한 부동 소수점 기반의 압축 앨고리즘인 MPEG-4 CELP coder를 16 bit 고정 소수점 기반으로 구현하였다. 이를 위해 LSP 계수를 계산하는 앨고리즘을 Chebyshev series method 대신 Real root method 방법을 사용하였고 정밀도(Precision)와 동적 범위(Dynamic Range)를 고려하여 스케일링 기법을 사용하였으며 DSP 명령어가 지원하지 않는 수학 함수들은 미리 계산하여 테이블링 기법을 적용하였다. 인지(Perception) 테스트를 통하여 부동 소수점 코더와 음질의 열화 상태를 비교하였다.

II. MPEG-4 CELP coder의 개요

MPEG-4 CELP가 기존의 CELP와 다른점은 유연성(flexibility)이다. 일반적으로 CELP 방법은 특정 타겟에 대해 단일 bit rate으로 압축하는 방법인데 반해 MPEG-4 CELP는 단일 기본 코딩기법을 통하여 여러 응용분야에 적용이 가능하다. MPEG-4 CELP coder는 Multiple bit rate, Bit-Rate Scalability, Bandwidth Scalability, Complexity Scalability와 같은 기능을 제공한다. Complexity Scalability는 16kHz sampling mode에서 지원되며 Table 1과 같이 디코더를 3가지 Level중에 선택하여 구현할 수 있다.

Complexity Level	Description
C3	Improved LPC interpolation (LSP) : Full decoding
C2	Simplified LPC interpolation (LAR)
C1	Reduced order LPC synthesis filter

Table 1. Possible Decoder complexity levels

또한 MPEG-4 CELP coder는 SampleRateMode, QuantizationMode, FineRateControl 이 세 가지 파라미터의 조합으로 8개의 모드를 제공한다. 그림 1에 8개의 모드를 나타내었다. 본 논문에서는 일반적인 저비트율 CELP 방식인 모드 Ⅷ을 기준으로 구현하였다.

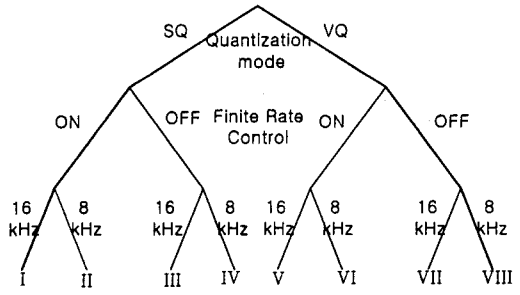

그림 1. Possible modes in which the MPEG-4 coding scheme can operate

III. 고정 소수점 연산구조 구현

3.1 고정 소수점 음성코더 개발 과정

그림 2는 고정 소수점 기반의 음성코더 개발 과정을 나타내었다. 일반적으로 음성 코더는 초기에 부동 소수점 연산구조에 기반하여 개발되고 검증을 실시한 후 DSP 어셈블리로 변환하기 전에 중간 과정인 고정소수점 구조로의 변환이 필수적이다. 고정소수점 연산구조로의 변환과정에서 DSP 프로세서가 제공하지 않는 수학 함수들은 테이블 기법을 적용하여야 하며, 음성 모델 파라미터에 대해 scale factor를 미리 정의하여야 한다. 각 모듈별로 고정 소수점 연산 구조로 변환하고 이를 통합하여 인지(perception) 테스트를 실시하였다.

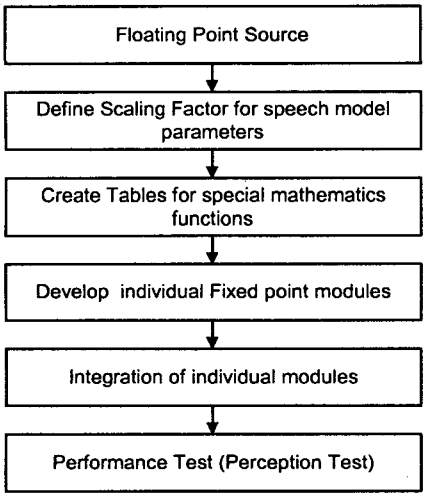

그림 2. 고정 소수점 음성코더의 개발 차트

3.2 음성 파라미터 스케일링

고정 소수점 DSP는 제한된 dynamic range내에서 데이터를 다루기 때문에 음성 파라미터에 대한 scale factor들이 미리 정의 되어야 한다. 적절한 스케일링은 각 파라미터의 dynamic range를 극대화 시켜주며 라운딩 에러를 최소화 시켜준다. 또한 overflow나 saturation을 방지해 준다.

3.3 수학 함수들에 대한 테이블링 적용기법

표준 DSP 명령어가 제공하지 않는 수학함수들은 미리 계산하여 테이블링 하는 것이 필수적이다. 따라서 구현할 코더에서 사용되는 cosine 함수와 log(1+x) 함수, 10^x 함수를 테이블링 하였다. cosine 함수는 512개로 테이블링 하였고, log(1+x) 함수는 [1 ~ 600]의 입력범위에 대해 600 엔트리를 가지도록 하였으며 결과값의 범위가 [0.3 ~ 2.7789]이기 때문에 4로 스케일링하여 사용하였다. 10^x 함수는 입력범위가 [1.0 ~ 3.0]이고 이를 384 엔트리를 가지도록 테이블링 하였다.

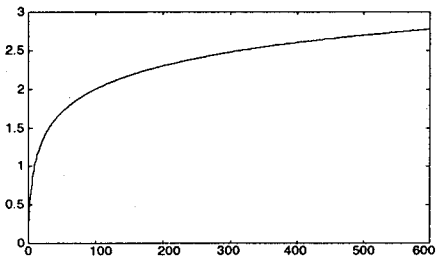

그림 3. log(1+x)의 입출력 범위

3.4 정규화(Normalization)

고정소수점 연산구조에서 가장 주의해야 할 사항은 에러의 축적이다. 이의 해결방법 중의 한가지가 입력 데이터의 정규화이다. 이것은 곱하기 연산이나 나누기 연산에서 정밀도(Precision)를 극대화 해준다. 정규화 명령어는 일반적으로 DSP 명령어에 포함되어 있으며 MSB(most significant bit)의 zero가 없어질 때까지 왼쪽으로 쉬프트 한다.

3.4 모듈안에서의 파라미터 스케일링

고정 소수점 연산구조를 구현하기 전에 모든 Scale Factor들을 미리 정의하는 것이 필요하다. 또한 부동 소수점 모듈을 통하여 Dynamic range의 계산이 필요하다. 그림 4는 부동 소수점으로 구현되어 있는

Synthesis Filter이고 이를 고정 소수점 연산구조로 변환시킨 것이 그림 5이다. 그림 5에서는 4로 스케일된 LPC 계수를 입력으로 받아서 결과는 원래의 스케일로 복원하여 내준다.

```
for (j=0; j < n; j++) {
    s = (float)0.0;
    for (i=0; i < np; i++)
        s = s - a[i] * pm[i];
    xr[j] = di[j] + s;
    for (i=2; i < np+1; i++)
        pm[np-i+1] = pm[np-i];
    pm[0] = xr[j];
}
```

그림 4. 부동소수점 Synthesis Filter

```
for (j=0; j < n; j++) {
  Fixs = (long)0;
  for (i=0; i < np; i++)
      Fixs=L_SUB(Fixs,L_MULT(Fixa[i],
                  Fixpm[i]));
  xr[j] = di[j] + s;
  Fixxr[j]=ADD(Fixdi[j], ROUND(L_SHL(Fixs,2)))
  for (i=2; i < np+1; i++)
      Fixpm[np-i+1] = Fixpm[np-i];
  Fixpm[0] = Fixxr[j];
}
```

그림 5. 고정소수점 Synthesis Filter의 스케일링

만일 중간 결과가 범위 내에서 값이 너무 크다면 중간 스케일링(Intermediate Scaling)이 요구된다. 곱하고 축적(multiply and accumulate)하는 연산에 대해 overflow나 saturation을 방지하기 위하여 중간에 적당한 값으로 스케일 해준다. 그림 6은 Levinson-Durbin recursion을 구하기 위하여 Auto-correlation 계수를 구하는 모듈의 일부분이다. 이것은 두 입력을 곱하고 축적하는 전형적인 구조이다. 이를 고정 소수점으로 변환할 때 그림 7에서와 같이 두 입력을 16 bit으로 받아서 곱한 값을 32 bit으로 저장하고 오버플로우를 방지하기 위하여 1 bit을 내려서 축적한다. 이렇게 축적된 값은 다시 16 bit으로 반올림하여 결과값으로 내준다.

```
for (l = 0; l < N - i; l++)
    temp += *pin1++ * *pin2++;
*acf++ = temp;
```

그림 6. 부동 소수점 축적 연산구조

```
for (l = 0; l < N - i; l++) {
    Acc1 = L_MULT(*pin1++ , *pin2++);
    Acc1 = L_SHR(Acc1,1);
    Acc0 = L_ADD(Acc0,Acc1);
}
*Fixacf++ = ROUND(Acc0);
```

그림 7. 고정 소수점 축적 연산구조의 중간 스케일링

3.5 LSP 계수를 위한 앨고리즘 변환

부동 소수점 연산 기반의 코더에서는 LPC 계수를 LSP 계수로 변환하기 위하여 Chebyshev series method를 이용하였다. (1)식을 Chebyshev 다항식을 이용하여 확장하면 x=cos(w)식에 의해 실수 구간 [-1 1] 사이에 근이 존재하게 된다.

$$P(z) = 2e^{j5w/2}[A_0\cos(\frac{5}{2}w)+A_1\cos(\frac{5-2}{2}w)+\ldots+\frac{1}{2}A_{5/2}] \quad (1)$$
$$Q(z) = 2e^{j5w/2}[B_0\cos(\frac{5}{2}w)+B_1\cos(\frac{5-2}{2}w)+\ldots+\frac{1}{2}B_{5/2}]$$

이 방법을 고정 소수점 연산으로 구현하면 오차가 Real root method 방법으로 구현한 것보다 크고 사인 함수, 코사인 함수 그리고 역 코사인 함수 총 3개의 함수에 대해 테이블링이 요구된다. 이 3개의 테이블링에 의해 오차가 누적되는 결과를 초래한다. 이에 반해 Real root method 방법은 (1)식에 x=cos(w)를 대입하여 식을 정리하면 (2)식을 얻을 수 있고

$$P_{10}(x) = 16A_0x^5+8A_1x^4+(4A_2-20A_0)x^3+(2A_3-8A_1)x^2 +(5A_0-3A_2+A_4)x+(A_1-A_3+0.5A_5)$$
$$Q_{10}(x) = 16B_0x^5+8B_1x^4+(4B_2-20B_0)x^3+(2B_3-8B_1)x^2 +(5B_0-3B_2+B_4)x+(B_1-B_3+0.5B_5) \quad (2)$$

근은 (3)식에 의해 구할 수 있다.

$$LSF(i) = \frac{\cos^{-1}(x_i)}{2\pi T}, \quad \text{for } 1\le i\le p \quad (3)$$

Real root method는 코사인 테이블 한 개로 구현이 가능하여 오차측면과 메모리 측면에서 Chebyshev series method에 비해 성능이 우수함을 확인하였다.

3.5 성능 테스트

음성 코더의 성능 실험은 주로 객관적인 테스트(objective test)와 주관적인 테스트(subjective test)로 실시된다. 본 연구에서는 주관적인 테스트인 인지(perception) 테스트를 실시하였다. 시료는 남자 음성 5

문장과 여자 음성 5문장을 사용하였다. 이를 부동 소수점 기반의 코더와 고정 소수점 기반으로 구현한 코더를 가지고 원본과의 음질 열화를 비교하였다. 그림 8은 여자 음성을 시료로 하여 부동 소수점 기반의 코더와 본 연구에서 구현한 고정 소수점 기반의 코더를 통과한 음성 파형을 출력한 것이다. 그림 (a)는 여성 화자의 음성 파형이고 (b)는 부동 소수점 기반의 코더를 통과한 음성 파형이고 (c)는 고정 소수점 기반의 코더를 통과한 음성 파형이다. 인지 테스트 결과 대부분의 시료에서 고정 소수점 기반의 코더를 통과한 음성은 부동 소수점 코더를 통과한 음성과 비교하여 음질의 열화가 없음을 확인하였다.

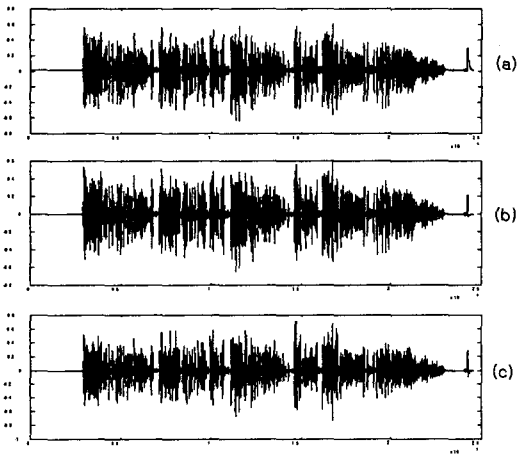

그림 8. MPEG-4 CELP coder를 통과한 음성파형
(a) 원본 음성, (b) 부동 소수점 코더를 통과한 음성파형, (c) 고정 소수점 코더를 통과한 음성파형

V. 결론

본 논문에서는 부동 소수점 기반의 음성 압축 알고리즘인 MPEG-4 CELP coder를 DSP 어셈블리로 구현이 가능하도록 고정 소수점 기반으로 구현하였다. 고정 소수점 연산에서 고려해야 할 정밀도(Precision)와 동적범위(Dynamic Range)를 보장하기 위하여 정규화(Normalization)와 스케일링 방법을 사용하였다. 그리고 DSP 명령어가 지원하지 않는 수학 함수들은 미리 계산하여 테이블링 기법을 적용하였다. 인지 테스트 결과 부동 소수점 코더와 비교하여 대부분의 시료에서 음질의 열화가 없음을 확인하였다. 그러나 10개의 시료 중에 일부는 부동소수점 코더와 비교하여 음질의 열화가 약간 발생하는 것도 확인하였다. 앞으로 알고리즘의 개선과 코드, 테이블, 스케일링의 최적화에 관한 연구가 수행될 필요가 있다.

참고문헌

[1] ISO/IEC 14496-3 Part 3: Audio Subpart 3 : CELP

[2] A.M.Kondoz, *Digital Speech*, John Wiley & Sons, 1994

[3] Ludy Liu, "Fixed Point Vocoder Implementation," *IEEE Circuits and Systems, Proceedings of the 40th Midwest Symposium on*, pp. 710 -715 vol.2. 1998

[4] L.R.Rabiner, R.W. Schafer, *Digital Processing of Speech Signals*, Prentice-Hall, 1978

[5] Peter Kabal, Ravi, "The computation of Line Spectral Frequencies Using Chebyshev Polynomials," *IEEE Trans on ASSP*, pp. 1419-1425, December 1986.