

인터넷 화상 전화용 음성 코덱을 위한 MPEG4-CELP 부호화기의 구현

김병수*, 김도형*, 강경욱**, 홍진우**, 정재호*

*인하대학교 전자공학과

**한국전자통신 연구원

Implementation of MPEG4-CELP Vocoder for Speech Codec of Internet Video Phone

B.S.Kim*, D.H.Kim*, K.O.Kang**, J.U.Hong**, J.H.Chung*

*Dept. of Electronic Engr. Inha Univ.

**Electronics and Telecommunications Research Institute

E-mail : g1991185@inhavision.inha.ac.kr

Abstract

인터넷이 일상생활에 다양하게 활용되면서 인터넷 채널을 통한 정보의 형태는 문자와 이미지 외에 음성, 오디오 신호 및 동영상 부분까지 확대되고 있다.

본 논문에서는 MPEG4-CELP를 인터넷 화상 통신의 음성 코덱용으로 사용하기 위한 최적화 기법 및 알고리즘의 개선을, DSP칩이 내장된 보드가 아닌 인터넷의 터미널로 사용되고 있는 펜티엄 프로세서를 장착한 PC에 초점을 맞추어 수행하였다.

MPEG4-CELP VM C소스를 분석 및 프로파일(Profile)한 결과를 토대로 파라미터 추출을 위해 많은 연산을 수행하는 부호화에 대해서 CPU상에 부하를 많이 주는 함수들을 제 1차 최적화 대상 함수들로 선정하고, CPU에 부하를 많이 주지는 않으나 호출되는 회수가 많은 함수를 2차 최적화 대상 함수로 선정해, C소스 레벨의 소프트웨어 파이프 라이닝(Software Pipelining) 기법들을 적용하여 최적화를 수행하였다. 또한 1차 최적화 대상 함수의 경우에는 소프트웨어 파이프라이닝의 적용과 함께 연산량 감소를 위한 알고리즘 변형까지 수행하였다.

위의 과정을 거쳐 최적화 된 MPEG4-CELP는 펜티엄III 450MHz PC에서 음성을 부호화 하는데 원 VM소스에 비해 약 2배정도의 시간이 단축되는 것을 확인하였다.

1.서론

음성 코덱을 실제분야에 적용하기 위해서는 코덱을 통하여 나오는 합성음의 음질과 부호화 및 복호화의 속도 두 가지의 문제점을 해결해야 한다. 현재 무선 통신 및 인터넷 화상 통신의 음성 코덱으로 사용하고 있는 대부분의 코덱들은 선형예측상수(LPC)를 기반으로 하

는 보코더(Vocoder)이다. 보코더는 송신측에서 인간의 발성 과정을 성도와 여기신호 생성부로 분류하고 두 가지를 모델링하여 수신측에 보내 두 파라미터를 이용하여 음성을 재합성한다.[1] 이 두 가지 파라미터의 정확한 추출은 수신측에서 생성되는 합성음의 질을 결정짓는 중요한 요소가 되지만 정확한 파라미터의 추출과정은 많은 연산량을 필요로 하여 보코더의 수행 속도를 저하시키는 요인이 된다.

성도모델에 입력 신호로 사용되는 여기신호의 추출 방법은 크게 분석 합성법(Analysis and Synthesis)과 합성에 의한 분석법(Analysis by Synthesis)[2]으로 나누어지며 여기신호의 추출 방법과 형태에 따라 여러 보코더로 분류된다.

1982년 제안된 합성에 의한 분석법은 여러 가지 여기신호를 성도모델에 입력으로 부과하여 원음성과 가장 오차가 적은 합성음을 생성시키는 여기신호를 최적의 여기신호로 선정하는 방법이다.[3] 합성에 의한 분석법은 많은 여기신호의 시도를 통하여 최적의 것을 선정함으로써 높은 질의 합성음을 생성시킨다는 장점이 있지만 여기신호 선정 과정에서 많은 연산을 수행함으로써 속도가 느린 단점이 있다. 멀티 펄스(Multi-Pulse) 보코더나 EVRC, CELP계열의 보코더는 합성에 의한 분석법에 의해 여기신호를 추출하는 보코더의 대표적인 예라고 할 수 있다.

본 논문에서 제시하고자 하는 MPEG4-CELP도 합성에 의한 분석법에 의해 여기신호를 추출한다. 음성의 주기적인 여기 성분은 매 프레임마다 업데이트되는 적응 코드북에서 선정을 하고 그 나머지 여기성분은 멀티 펄스나 레귤러 펄스(Regular-Pulse) 두 가지 방식에 의해 추출한다. MPEG4-CELP 부호화기에서는 여기신호의 선정 때문에 많은 연산량을 요구하며 이로 인해 속

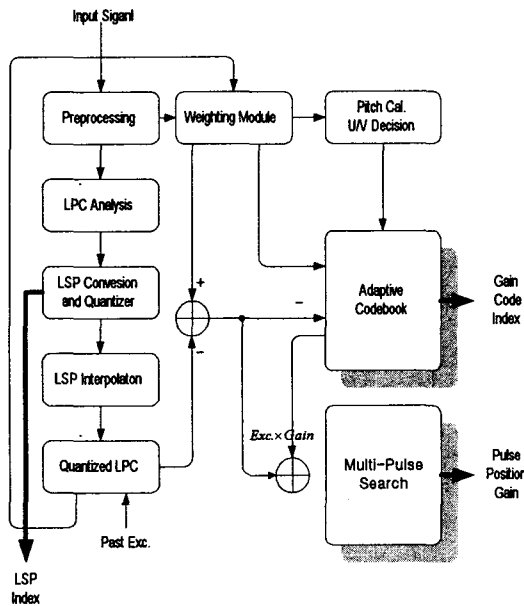


그림 1. MPEG4-CELP 부호화기의 구조

도의 저하를 수반한다.

2. MPEG4-CELP 부호화기의 개요 및 구조

MPEG4-CELP 부호화기는 8kHz와 16kHz로 샘플링된 음성을 입력으로 하여 여러 전송률 중에서 하나를 선정하여 음성을 압축한다. 그리고 여기방식도 멀티펄스(MPE) 방식과 레귤러펄스(RPE) 방식을 제공한다.

멀티펄스 모드는 8kHz와 16kHz로 샘플링된 두 신호 모두에 대해서 지원을 하며 레귤러 펄스 모드는 16kHz로 샘플링된 음성에 대해서만 지원을 한다. 본 논문에서는 두 가지 여기 방식중 8kHz로 샘플링된 음성을 지원하는 멀티펄스 방식의 부호화기의 최적화에 초점을 맞추었다.

그림1은 MPEG4-CELP 부호화기의 구조를 간략화시킨 것이다. 입력음성은 음성 신호의 저주파 대역에 걸쳐서 존재하는 DC성분 제거를 위해 프리프로세싱(preprocessing) 단계를 거쳐 일반적인 보코더와 동일하게 레빈슨-딜빈(Levinson-Durbin) 알고리즘을 통해 성도 모델링을 위한 10차의 선형예측계수(LPC) 추출을 한다. 이것은 다시 LSP로 변화되어 두 스테이지의 스플리트 양자화에 의해서 양자화 되며, 양자화된 LSP는 전 프레임의 LSP와 보간되어 웨이팅(weighting)모듈을 구성한다. 웨이팅 모듈은 선형예측상수에 의해서 결정되는 주파수 영역에서의 포먼트 부분을 약화시키는 역할을 하고 이 모듈을 거친 음성은 여기신호를 추출하는 기준 신호로 이용된다.[4] 피치는 개방 루프(Open-Loop)를 이용하여 정규화된 자기상관

함수(normalized autocorrelation)값이 최대가 되는 지점을 선택하고 이 값을 기준으로 하여 전 프레임에서 업데이트된 적응 코드북에서 음성의 주기적 성분에 해당하는 여기신호를 추출하고 이득을 구한다.

일반적인 CELP의 경우 적응 코드북 검색 이후 적응 코드북에서 선정되어진 코드와 이득을 곱한 값과 원음성의 차를 이용해서 고정 코드북 검색을 통하여 잔여 여기신호 성분을 선정하지만 MPEG4-CELP의 멀티펄스 모드에서는 적응 코드북 검색이후 웨이팅된 기준신호와 적응 코드북에서 얻어진 여기신호와 이득을 곱해 두 신호를 뺀 값을 이용하여 고정 코드북 검색 대신 전송률에 따라서 서브 프레임당 8~11개 사이의 펄스의 위치와 이득을 구한다. 이 방식은 합성에 의한 분석법 도입 당시 가장 먼저 시도되었던 멀티펄스 보코더[5]와 유사한 방법으로 정해진 수의 펄스의 위치와 이득으로 고정 코드북을 대신한다. 고정 코드북을 대신한 멀티펄스의 도입으로 인하여 MPEG4-CELP 멀티펄스 모드의 부호화 속도가 느려진다. 앞의 과정 중 웨이팅 모듈 생성 단계부터는 서브 프레임의 수에 따라 행해지게 되며 한 프레임의 길이와 서브프레임의 수 및 길이 역시 전송률에 따라 변하게 된다. 이런 과정을 거쳐 추출된 파라미터는 정해진 전송률에 따라서 비트 패킹(packing)된다.

3. MPEG4-CELP의 최적화

MPEG4-CELP의 소프트웨어를 최적화 하기 위해서는 그림2 에서와 같이 부호화기의 실행 시간을 프로파일링해 그 중 호출 빈도 수가 높거나 많은 시간을 점유하는 함수들을 최적화 대상 함수로 선정하여 그 함수들을 중심으로 소스를 분석하고 최적화 대상 함수들을 일반적으로 사용되는 소프트웨어 파이프라이닝의 적용과 알고리즘 변형을 수행하면서 지속적인 프로파일링을 하여 그 결과를 참고로 최적화 과정을 수행하였다.

적용한 일반적인 소프트웨어 파이프라이닝 기법은 다음의 세 가지이다.

- i) Redundant Loops Removing
- ii) Trip Counter Control
- iii) Loop Unrolling

i)의 방법은 가장 기초적인 소프트웨어 파이프라이닝 방법으로 프로그램 내에서 필요 없이 수행되는 루프를 제거하거나 같은 인덱스를 가지고 동작하는 여러 개의 루프를 하나로 묶는 것이다.

ii)의 방법에서 trip counter란 수행되는 루프의 반복 회수를 조정하는 계수를 말하며, 루프 반복의 한계를 결정하는 수를 trip count라고 한다. 루프를 정의하는 문장에서 " $i < N$ "과 같이 trip count의 조건을 확인하는 계산을 수행하게 된다. 이러한 조건의 확인도 수행 시간을 소비하게 되는 것이다. 대개의 컴퓨터들은 " $i < N$ "과 같은 복잡한 비교를 하는 instruction 을 제공하지는

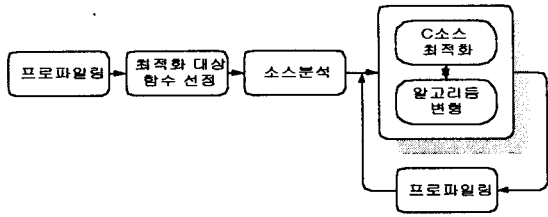


그림 2. MPEG4-CELP 최적화 과정

않지만 “neqz”(not equal to zero)와 같은 간단한 비교를 하는 instruction을 제공한다. Windows의 경우도 예외는 아니어서 register에 있는 내용과 숫자를 직접 비교하는 “CMP”라는 instruction을 제공한다. 따라서, trip counter가 간단한 비교를 통하여 수행될 수 있도록 조정하는 것이 바람직하다. 이를 위하여 일반적으로 증가하는 인덱스를 갖는 trip counter를 감소하는 인덱스를 갖도록 조정함으로써 코드의 실행시간을 감소시킬 수 있다.

iii)의 방법은 반복 횟수가 정해져 있는 루프들을 하나씩 모두 풀어서 코드를 작성하는 방법이다. 이것은 위에서 언급한 다른 방법보다 효과가 매우 높다. 소스의 for문이나 while등의 문장은 order에 해당되는 숫자만큼 주어진 문장을 수행하는 것 외에 order만큼의 조건 판단과 counter 증가를 수행한다. 이것은 매우 redundant한 작업이라 할 수 있다. 반복 횟수가 고정되어 있는 경우 루프를 풀어 조건 판단이나 counter증가를 제거함으로써 명령어의 개수를 줄여 프로그램의 실행 속도를 증가시킨다.

이상의 세 가지 소프트웨어 파이프라이닝 방법을 기초로 하여 소스를 최적화시켜 부호화 속도를 향상시킬 수 있으나 현재 제공되는 컴파일러의 성능이나 CPU속도의 발달로 인해 큰 속도의 향상은 얻기가 힘들다. 그래서 이 세 가지 방법 이외에 부호화 과정 중 매 프레임마다 초기화되고 다시 연산이 되어지는 값들 중에서 일정하게 유지되는 값들은 테이블 형태로 저장을 하여 부호화 초기시 읽어 버퍼에 저장하여 사용함으로써 쓸모 없는 연산량을 제거할 수 있다.

위와 같은 소프트웨어 파이프라이닝 방법과 테이블의 활용 방법 이외에 부호화 속도를 증가시키기 위한 연산량 감소를 목적으로 다음과 같은 알고리즘을 적용하였다.

MPEG4-CELP VM 소스의 웨이팅 모듈과 선형예측 계수를 이용하여 음성을 필터링 하는 nec_syn_filt(), nec_pw_filt() 함수에서 서브 프레임의 수만큼 메모리를 업데이트하는 부분을 서브 프레임당 한번씩만 수행하도록 프로그램을 수정함으로써 최소 39번에서 79번의 루프를 제거하여 표1에서 볼 수 있는 것과 같이 nec_pw_filt()함수는 3배 이상의 속도 향상을 보였으며 nec_syn_filt()함수는 5배 이상의 속도 향상을 얻을 수 있었다.

음성의 주기적인 성분을 표현하는 피치의 계산은 CELP 부호화기에서는 식(1)과 같이 정규화된 자기상관 함수를 이용하여 값이 최대가 되는 지점을 선택한다.

$$R(T_i) = \frac{\left\{ \sum_{k=0}^{N-1} s[ptr - T_i + k]s[ptr + k] \right\}^2}{\sum_{k=0}^{N-1} s[ptr - T_i + k]s[ptr - T_i + k]} \quad (1)$$

T_i 는 피치를 검색하는 범위 값으로 MPEG4-CELP 멀티 펄스 8kHz모드에서는 최소 17부터 최대 144값을 가지게 된다. N 의 값은 한 프레임의 샘플수이다. 식(1)에서 분모의 경우 첫 번째 T_i 의 경우만 모든 샘플을 수행하고 다음 T_{i+1} 에서부터는 식(2)와 같이 T_i 에서 계산된 값을 저장하고 있다가 다음 샘플의 제곱과 맨 앞의 샘플의 제곱을 각각 더하고 뺄으로써 같은 결과는 나타내지만 곱셈 연산량을 감소시켜 부호화기의 속도를 증가시킬 수 있다.[6]

$$C(k, k)_{T_i} = C(k, k)_{T_{i-1}} + \{s[ptr - T_i + N - 1]\}^2 - \{s[ptr - T_i - 1]\}^2 \quad (2)$$

식(2)의 $C(k, k)_{T_i}$ 는 T_i 때의 자기상관함수 값을, $C(k, k)_{T_{i-1}}$ 은 초기의 자기상관함수 값을 나타낸다.

프로파일 결과 MPEG4-CELP 부호화기의 속도를 저하시키는 가장 큰 요인은 적응 코드북 검색과 멀티 펄스의 생성 부분이다. 적응 코드북의 경우 계산한 피치를 기준으로 하여 전 프레임에서 업데이트 되어진 적응 코드북의 검색을 식(3)에 의해서 17번을 수행하여 최적의 여기신호를 선정한다.

$$E_{\min}(l) = \sum_{n=0}^{N-1} (s_w[n] - g_{\nu}d[n])^2, \quad t_{op} - 8 < l < t_{op} + 8 \quad (3)$$

이 부분의 검색 회수를 감소시키기 위해 검색 지점을 2씩 증가시켜 9번의 검색만을 수행함으로써 정확성이 떨어지는 코드의 인덱스를 찾아낼 가능성을 내포하지만 부호화 속도는 향상되었다.

적응 코드북 검색 후 수행되는 멀티 펄스의 위치를 찾는 부분에서 원래의 MPEG4-CELP는 다음 펄스의 위치를 찾기 위해 전 펄스의 위치를 기준으로 최대 32곳의 위치를 검색한다. 하지만 최적화 된 CELP에서는 이것을 전 펄스의 위치를 기준으로 하여 4곳만을 검색하여 현재 펄스의 위치를 찾게 함으로써 많은 연산량을 감소시킬 수 있다. 그러나 여기신호의 검색 부분의 연

표 1. 프로파일 결과

(단위 : ms)

	Original Codec		Optimized Codec	
	time	function	time	function
	1799.29	nec_enc_mp	997.26	nec_enc_mp
	731.67	nec_pw_filt	230.28	nec_acb_gen
	428.73	nec_acb_gen	184.95	nec_pw_filt
	402.59	nec_syn_filt	129.92	nec_zero_filt
	244.89	nec_zero_filt	107.45	nec_mode_dec
Total	4654.94		2228.69	

산량 감소로 인해 부호화 속도는 향상되었으나 멀티 펄스 선정 범위의 축소와 적응 코드북에서 피치를 기준으로 17곳을 검색 하던 것을 홀수 부분을 누락 시킴으로써 최적의 여기신호 선정 과정이 약화됨으로 인해 합성음의 질은 열화 되었다

표1 은 펜티엄III 450MHz PC에서 2.89초 길이의 8kHz로 샘플링된 음성을 전송률 12kbps로 부호화 하는데 걸리는 시간을 원소스와 최적화 후의 것을 프로파일링 하여 비교한 것이다.

합수들은 모든 합수를 열거하지 않고 상위 다섯 개의 합수들만 열거하였다.

4. 실험 및 결과

최적화 후의 MPEG4-CELP 부호화기의 성능 실험은 12kbps의 압축률에 대해서 원 부호화기와 최적화 후의 부호화기의 부호화 속도를 프로파일러로 측정하였으며 합성음의 음질을 비교하기 위해 MPEG4-CELP에서 제공하는 동일 복호화기를 사용하여 합성한 음성을 원래의 음성과 신호 대 잡음비(SNR)를 이용하여 비교하였다. 부호화 속도 측정과 합성음의 음질 비교를 위하여 사용한 음성은 모두 8kHz로 샘플링된 한국어 문장-여성화자 3명, 한국어 문장-남성화자 3명, 그리고 영어문장-여성화자 3명 총 9명에 대하여 실험을 하였으며 각각의 문장은 서로 다른 길이의 다른 문장을 사용하였다. 실험을 한 PC와 운영체제 및 프로파일러의 종류는 다음과 같다.

- i) PC사양 : 펜티엄III 450MHz
- ii) 운영체제 : 윈도우 98
- iii) 프로파일러 : MS Developer 6.0

실험을 통하여 최적화 후의 CELP부호화기의 속도는 원 부호화기에 비해 약 2배정도 향상되었음을 볼 수 있으나, 수정된 CELP부호화기는 연산량 감소를 위해 여기신호를 생성하는 적응 코드북과 멀티 펄스 선정 부분의 검색 횟수를 감소 시켜 최적의 여기신호 선정의 정확도가 원소스에 비해 감소되기 때문에 신호대 잡음비 수치상 음질의 열화가 발생한 것을 관찰할 수 있다. 그러나 전반적인 신호대 잡음비가 높은 값을 가지므로써 스피커를 통해 들었을 경우 두 합성음의 구분이 어려웠다.

5. 결론 및 앞으로의 방향

본 논문에서는 펜티엄III 450MHz PC에 맞추어 CELP를 최적화하여 음성을 실시간으로 부호화하는 것을 보였다. 하지만 동일한 PC에서도 CPU의 부하상태나 하드디스크의 속도 및 메모리의 크기에 따라서 부호화의 속도가 변하는 것을 볼 수 있었으며 속도향상을 위해 여기신호 분석부분의 검색 횟수를 감소시킴으로 신호대 잡음비 수치상 음질의 열화를 보였다.

인터넷의 채널 영향과 전송속도 및 현재 일반적으로

표 2. 신호대잡음비와 프로파일링 결과

	Length(ms)	SNR(dB)		Profiling Result(ms)	
		최적화전	최적화후	최적화전	최적화후
FK1	2675	41.93	40.44	4371.2	2028.2
FK2	1466	40.17	38.73	2374.3	1305.7
FK3	2621	38.81	38.24	4237.1	2041.9
MK1	6112	35.02	34.13	9844.2	4735.9
MK2	5303	38.33	37.34	8506.2	4643.1
MK3	3560	39.09	38.09	5738.6	2759.1
FE1	2859	46.67	42.09	4604.6	2134.4
FE2	2579	43.04	39.30	4146.1	2029.2
FE3	2919	45.12	39.12	4695.9	2265.9

F:여성화자 ,M:남성화자 ,K:한국어 ,E:영어

사용되고 있는 PC의 CPU속도 등을 고려하여 인터넷 화상 통신용 음성 코덱으로 MPEG4-CELP를 사용하기 위해서는 본 실험에 사용한 PC보다 더 낮은 사양의 CPU에서도 실시간 부호화가 가능해야 한다.

펜티엄 프로세서에 알맞은 소프트웨어 패키징 기법의 연구와 더불어 여기신호 분석부분 알고리즘의 개선을 통하여 부호화 속도를 향상시키고 음질의 열화를 감소시키는 방향으로 연구를 진행할 예정이다.

6. 참고문헌

[1] L.R.Rabiner and R.W.Schafer, "Digital Processing of Speech Signals," Prentice Hall, pp.98-106, 1978
 [2] Ravi P. Ramaachandran and Richard Mammone, "Modern Methods of Speech Processing," Kluwer Academic Publishers, pp.51-74, 1995
 [3] J.H.Chung, "A Nwe Homomorphic Vocoder Framework Using Analysis by Synthesis Excitation Analysis," Georgia Institute of Technology, 1991
 [4] A.M.Kondoz, "Digital Speech Coding for Low Bit Rate Communication Systems," John Wiley & Sons, pp.141-146, 1994
 [5] Bishnu S.Atal and Joel R.Remde, "A New of LPC Excitation for Producing Natural Sounding Speech at Low Bit Rates," Proc. ICASSP, pp.614-617, 1982
 [6] Takahiro Unno, Thomas P. Barnwell III, and Kwan Truong, "An Improved Mixed Excitation Linear Prediction(MELP) Coder," Proc. ICASSP, pp.245-248, 1999

=====

본 연구는 한국전자 통신연구원의 연구지원에 의하여 수행되고 있습니다.

=====