

論文98-35T-6-17

검색(Cache) 메모리를 이용한 음성 부호화기 코드북 검색에 관한 연구

(A Study on Searching for Vocoder Codebook using
Cache Memory)

金錫燦 * , 全慶一 *

(Kim Seog Chan and Jeon Kyung Il)

요약

음성 신호의 특성과 LD-CELP 알고리즘의 코드북 인덱스 검색 현황을 조사한 결과, 전 단계에서 검색된 코드북 인덱스가 다음 단계에서도 쓰이게 되는 것을 알 수 있었다. LD-CELP 알고리즘은 음성의 분석 구간을 짧게하여 음성의 명료도가 좋은 편이나, 최적의 코드를 검색하기 위해 많은 계산량이 발생하는 단점이 있다. 본 논문에서는 LD-CELP에서 검색된 코드북 인덱스를 저장하는 검색 메모리를 두어, 코드북의 검색 과정을 일부 생략하는 방법을 제안한다. 이 방법으로 샘플 음성을 시뮬레이션한 결과, 코드북 검색 계산량을 줄여 기존의 LD-CELP 방식보다 코드북 검색 시간을 3.2%-11.7%까지 줄일 수 있었다.

Abstract

In the analysis of the audio signal characteristics and the codebook indices of LD-CELP, there are many cases of detecting codebook indices that are used previous. LD-CELP algorithm achieves good quality of audio because it has analyzed a short term of audio signal. In spite of these advantage, the method has a drawback in which searching time of best codebook index increase due to a complicated calculation for codebook index.

This paper is proposed to decreasing the searching time of codebook index using a searching memory. As a simulation of the proposed method, searching time for codebook index is reduced 3.2%-11.7% as compared with LD-CELP.

I. 서론

음성 부호화 방식에는 크게 파형 부호화(waveform coding) 방식과 신호원 부호화(source coding) 방식 그리고 이 둘을 결합한 혼합 부호화(hybrid coding) 방식이 있다. 혼합 부호화 방식은 Atal과 Schroeder

에 의해 제안된 CELP(Code-Excited Linear Predictive) 부호화기가 있으며 여러 가지의 변형된 형태의 알고리즘이 연구되고 있다^[1,2,3,4,10].

ITU-T G.728에서 제안된 LD-CELP(Low-Delay CELP) 방식은 전송 대역폭이 16kbps이며, 분석 구간을 짧게하여 음성의 명료도가 좋다^[5]. LD-CELP에서의 코딩 지연은 버퍼링 지연과 연산 지연으로 나눌 수 있는 데, 버퍼링 지연은 음성 신호를 분석하기 위해 프레임 단위로 신호를 모을 때 발생하고, 연산 지연은 프레임 단위로 모아진 신호를 처리하는데 걸리는 시간으로 최적의 코드를 검색하기 위해 많은 계산량이 요구된다^[6,7].

* 正會員, 仁荷大學校 電子計算工學科
(Dept. of Computer Eng., In-Ha Univ.)
* 正會員, 仁荷工業專門大學 電子科 教授
(Dept. of Electronics, In-Ha Tech. Junior College)
接受日字: 1998年4月20日, 수정완료일: 1998年6月12日

기존의 Atal과 Schroeder가 제안한 CELP 방식^[1]이 예측 파라미터와 여기 신호 벡터를 수신 측에 전달하는 것과는 달리 LD-CELP는 양자화된 여기 신호 코드북 인덱스만을 전달한다. 기존의 CELP 방식과 같이 합성에 의한 분석방법을 취하고 있으나 성능 향상을 위해 후방 참조 적응 방식(backward adaptive)을 도입하였다^[8,9,10]. 이때 사용되는 여기 이득, 합성 필터 그리고 인식 가중화 필터들의 계수는 이전의 신호 벡터에 의해 생성된다. 여기 신호는 단위 벡터 당 5차원 구조를 가지며 음성 신호 샘플 5개씩을 하나의 벡터 단위로 처리한다. 또한 기존의 LTP(Long - Term Prediction) 방식 대신에 채널 에러에 대한 안정성을 증대시키기 위해 50차의 STP(Short - Term Prediction) 방식을 사용하였으며, 합성 필터에 사용되는 예측 계수는 매번 양자화된 4개의 신호 벡터마다 한번씩 선형 예측 방법에 의해 수정된다. 여기 이득은 한 벡터마다 수정되고, 인식 가중화 필터는 예측 계수와 같은 주기로 수정된다. 따라서 합성 필터의 예측 계수와 인식 가중화 필터의 계수는 2.5ms의 수정 주기를 갖는다.

그림 1은 LD-CELP의 부호화기 블록도이다. 64kbps의 입력 신호가 5개씩 하나의 벡터 단위로 구성되어 LPC 분석 과정을 거치고 인식 가중화 필터를 지나게 되면 가중화된 신호 벡터가 만들어진다. 한편 전 단계의 여기 벡터 신호에 여기 이득 값 만큼 보정 계산을 행한 후, 50차 후방(backward) 선형 예측 분석 방법을 사용하는 합성 필터를 통과시키면, 제로 입력 응답 신호 벡터가 생성된다. 인식 가중화 필터를 거친 음성 신호와 제로 입력 응답 신호 벡터의 차에 의해 벡터 양자화된 코드북 검색 목표 벡터가 만들어지게 된다. 이렇게 생성된 두 개의 신호 벡터는 다음 단계인 코드북 검색 모듈을 지나게 된다^[4,5].

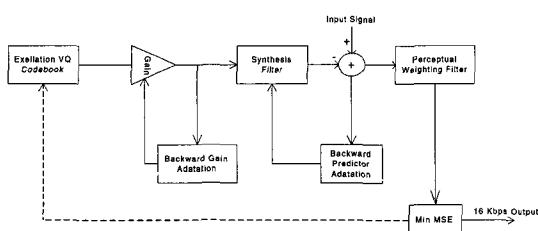


그림 1. LD-CELP 부호화기 블록도
Fig. 1. LD-CELP Encoder block diagram.

II. LD-CELP 코드북 검색 분석

1. 음성 신호에 대한 코드북 인덱스 검출 특성

그림 2에서와 같이 음성 신호는 120 ~ 200 개의 샘플 구간 내에서 변화하지 않으며, 특히 유성음의 경우 앞과 뒤의 신호 프레임이 강한 상관 관계가 있다. LD-CELP 방식에서는 신호의 단위 벡터 간격을 0.625ms로 설정하여 분석을 수행하므로 24 ~ 40개의 벡터 구간 내에서는 일정한 주기를 가지고 중복되거나 거의 비슷한 여기 신호가 추출된다.

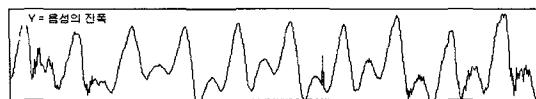


그림 2. 음성의 신호 파형

Fig. 2. Audio signal waveform.

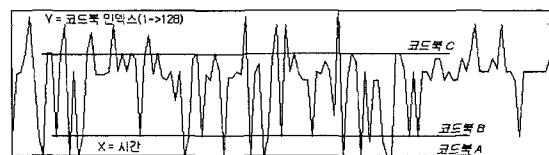


그림 3. 코드북 인덱스 검출

Fig. 3. Detection of codebook index.

그림 3은 3250ms 길이를 가진 대화 음성을 LD-CELP 알고리즘을 수행하여 코드북을 추출해 본 결과이며, 일정 구간에서 같은 코드북 인덱스가 반복적으로 검출됨을 확인할 수 있다. A, B, C 각 연장 선상에 연결된 꼭지점들은 같은 코드북 인덱스를 의미하며, 하나의 연장선상에 많은 꼭지점이 연결될 수록 해당 구간 안에서의 국부성이 높음을 나타낸다. 이 구간 내에서 연속된 신호 벡터는 서로 강한 상관 관계가 있게 되며, 하나의 신호 벡터로부터 검출된 코드북 인덱스는 그로부터 24~40 개까지의 신호 벡터 내에서 재 사용될 확률이 매우 높게 된다.

2. 코드북 인덱스 검출의 국부성

그림 4는 특정 코드북 인덱스가 얼마나 많이 참조되었는가를 보여주고 있다. 음성 신호의 유성음이 많이 모여 있을 경우 특정한 몇몇 코드북이 집중적으로 쓰여 타원형 안에 있는 꼭지점의 높이가 높아진 것을 알 수 있다.

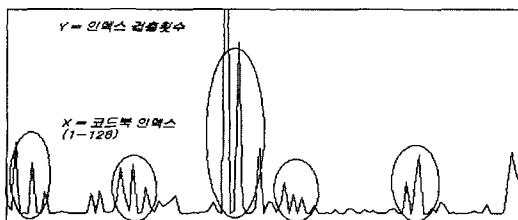


그림 4. 코드북 인덱스 사용 횟수

Fig. 4. The frequencies of used codebook index.

3. 코드북 인덱스 검출 간격 분포 특성

그림 5와 그림 6에서는 코드북 인덱스가 얼마나 시간 간격을 두고 검출되게 되는지를 살펴본 도표이다. 그림 5에서 짧은 선이 많이 모인 곳일 수록 코드북 인덱스가 주로 15ms내의 시간 간격으로 여러 번 나타남을 알 수 있고, 타원형으로 그린 점들의 집합이 왼쪽에 나타날 수록 코드북의 검출 시간 간격의 국부성이 이 매우 높게 나타난다. 그림 6의 그래프에서 몇몇 특정한 코드북 인덱스는 5ms 이내의 간격을 두고 집중적으로 나타남을 보이고 있다.

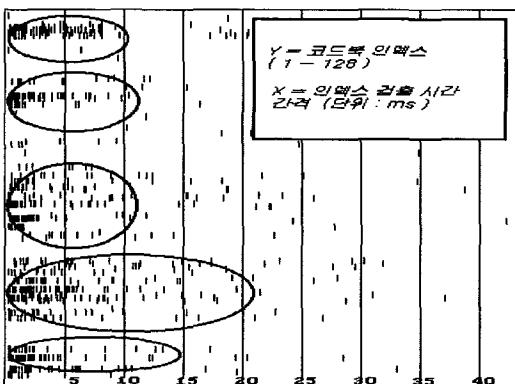


그림 5. 코드북 인덱스 검출 간격 분포도

Fig. 5. Detecting interval distribution of codebook index.

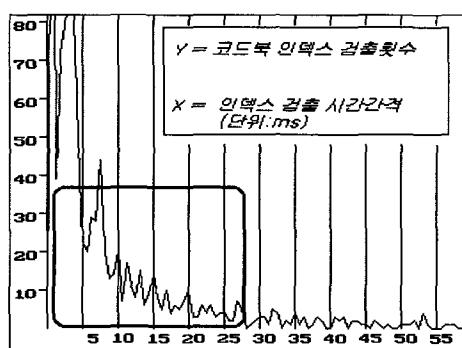


그림 6. 코드북 인덱스 검출 간격 누적도

Fig. 6. Detecting interval cumulative distribution of codebook index.

4. 검색 메모리를 이용한 코드북 인덱스 검색 방식

LD-CELP의 코드북에서 총 1024개의 코드 벡터는 검색의 효율을 증대시키기 위해 7비트의 음성(shape) 코드북과 3비트의 이득 코드북으로 이루어져 있다. 따라서 음성 코드북은 128개의 벡터이고 이득 코드북은 신호 표시에 1bit를 할당하고 크기를 나타내는데 2비트를 할당하고 있다.

7비트 음성 코드북에서 j 번째 코드 벡터를 y_j 라 하고, 3bit 이득 코드북에서 i 번째 코드 벡터를 g_i 라 할 때, H 는 합성 필터와 인식 가중화 필터를 통해 생성되는 임펄스 응답 벡터 $h(n)$ 의 행렬이라 놓으면 필터의 출력은 다음과 같다.

$$\tilde{x}_{ij} = H\sigma(n)g_iy_j \quad (1)$$

$x(n)$ 은 목표 검색 벡터이고, \tilde{x}_{ij} 가 목표 검색 벡터를 이득치 만큼 스케일링한 벡터 신호라면, i 와 j 에 가장 알맞은 코드북 검색은 다음과 같은 자승에러 최소화(D)을 만족해야 한다.

$$\begin{aligned} D &= ||x(n) - \tilde{x}_{ij}||^2 \\ &= \sigma^2(n) ||\hat{x}(n) - g_i H y_j||^2 \end{aligned} \quad (2)$$

여기서 $\hat{x}_{ij} = H\sigma(n)g_iy_j$, $\hat{x}(n) = x(n) / \sigma(n)$, $\sigma(n)$: 이득

식 (2)를 전개하면 다음과 같다.

$$D = \sigma^2(n) [||\hat{x}(n)||^2 - 2g_i p^T(n)H y_j + g_i ||H y_j||^2] \quad (3)$$

$||\hat{x}(n)||^2$ 과 $\sigma^2(n)$ 의 값이 코드북 검색 동안 고정되어 있으므로 D 를 최소화하면 다음과 같다.

$$\hat{D} = -2g_i p^T(n)y_j + g_i^2 E_j \quad (4)$$

여기서 $p(n) = H^T \hat{x}(n)$, $E_j = ||H_j||^2$

E_j 는 j 번째 음성 코드 벡터를 필터링한 에너지를 나타내며 4개의 코드 벡터가 수행될 동안 일정하다. 또한 $P_j = p^T(n)y_j$ 라 놓으면 식 (4)는 다음과 같이 간소화 할 수 있다.

$$\hat{D} = -2g_i P_j + g_i^2 E_j$$

d_i 를 같은 부호를 갖는 g_i 와 g_{i+1} 의 중간 값이라 할 때, 임의의 최적 이득 $\hat{g}_i < d_i$ 는 $P_j < d_i E_j$ 와 같게 되어 코드북 검색을 수행하게 된다.

그림 7은 검색 메모리(searching memory)를 이용한 코드북 인덱스 검출 블록도이다.

$$H = \begin{array}{|c|ccccc|} \hline & h(0) & 0 & 0 & 0 & 0 \\ \hline h(1) & h(0) & 0 & 0 & 0 & 0 \\ h(2) & h(1) & h(0) & 0 & 0 & 0 \\ h(3) & h(2) & h(1) & h(0) & 0 & 0 \\ h(4) & h(3) & h(2) & h(1) & h(0) & 0 \\ \hline \end{array}$$

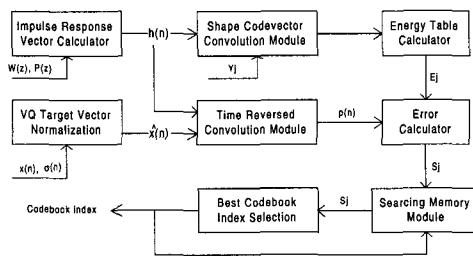


그림 7. 검색 메모리를 이용한 코드북 검색 블록도
Fig. 7. Codebook searching block diagram using searching memory.

임의의 짧은 구간 내에서의 높은 신호 벡터의 상관 관계를 이용하여 코드북 검색 메모리에 cache를 도입 한다. 그림 7은 cache를 이용한 일단 한번 검출된 코드북 인덱스 코드는 검색 메모리에 저장하여 두고, 그 이후 24~40 개의 신호 벡터가 처리될 동안 메모리에 상주하게 된다. 그리고 신호 벡터로부터 코드북 검출 시에 검색 메모리를 먼저 참조하여, 전단계에 쓰인 코드북을 탐색하고 적절한 인덱스로 판단될 때, 코드북 탐색 절차를 끝냄으로써 탐색 시간을 줄이게 된다. 검색 메모리 음성 코드북에 대한 인덱스는 코드북 추출의 특성상 두 가지 형태로 구분한다.

검색 메모리의 크기는 각각 cache_type_1과 cache_type_2에 2비트씩 다음과 같이 할당한다.

```

if 1 <= 음성 코드 인덱스 <= 114 then
    cache_type_1 = 음성 코드 인덱스 ;
else if 115 <= 음성 코드 인덱스 <= 128 then
    cache_type_2 = 음성 코드 인덱스 ;
  
```

코드북 인덱스가 임의의 구간 동안 특정한 값을 기준으로 양방향으로 나타나는 특성을 이용하여 검색 메모리의 수정 주기는 1.25ms로 설정한다. 신호 특성에 따라 15ms 동안 해당되는 인덱스가 없으면 그 코드 인덱스는 검색 메모리에서 삭제 시킨다. D의 비교 최

소치 임계값은 3으로 설정한다.

III. 모의 실험 및 결과 분석

본 논문에서 제안한 검색 메모리를 이용한 코드북 검색 성능 평가를 위해 실험은 IBM Pentium - 75MHz 컴퓨터로 시뮬레이션을 하였으며, 실험 데이터는 20대의 남성 화자가 임의로 발성한 것을 샘플링 주파수 8KHz를 8비트 PCM화하여 사용하였다. off line 시뮬레이션 특성상 성능 평가는 전체 구간이 끝난 후, 전체 시뮬레이션의 계산 시간을 인덱스로 생성하였다.

다음은 실험 음성 샘플이다.

Data 1: 안녕하세요 여기는 프로그램 언어 연구실입니다.

Data 2: 졸업과 입학을 축하 합니다.

Data 3: 사용하기가 무척 편리합니다.

Data 4: 오늘은 날씨가 무척 맑습니다.

Data 5: 우와 기분 좋다.

Data 6: 단일 유성음.

Data 7: 단일 무성음.

Data 8: 주변 잡음.

표 1. LD-CELP와 검색 메모리를 사용한 LD-CELP의 결과 값

Table 1. Computation times of LD-CELP and cached LD-CELP.

	LD-CELP	Cached-LD-CELP	개선율(%)
Data1	113.04	107.52	10.4
Data2	102.645	99.56	6.8
Data3	87.36	85.28	7.6
Data4	92.29	151.25	3.2
Data5	117.24	110.54	11.7
Data6	102.15	94.54	18.1
Data7	107.65	103.22	9.3
Data8	114.16	116.01	-3.4

<표 1>과 같이 음성 신호의 내용이 조용하고 평이 할 때에는 탐색 시간이 감소되었고, 같은 종류의 대부분의 음성에서 고른 비율로 나타났다. 특히 유성음이 많이 포함 될수록 탐색 시간이 더욱 감소되었다. 단일 유성음의 경우 최상의 성능을 보여 코드북 탐색 시간이 18%까지 향상되었다. 무성음이 많을수록 검색 메

메모리 내용의 재사용이 비교적 낮게 나타났고, 신호의 선형 특성이 없는 주변 잡음일 경우는 코드북 인덱스가 매우 불규칙하게 나타나서 성능이 오히려 -3% 정도 까지 저하됨을 볼 수 있었으며, 이것은 검색 메모리를 적용함에 있어서 사용된 몇 가지 if 문과 메모리 참조에 의한 오버 헤드로 추정된다. 또한 음이 유성음과 무성음이 바뀌는 곳은 파형의 변동이 심하여 MSE의 값이 3보다 훨씬 커서 검색 메모리를 사용하지 않았다.

IV. 결 론

본 논문은 음성 신호 처리에 대한 계산량을 감소시키기 위해 음성 신호의 선형 특성을 이용하여 검색 메모리용 cache 메모리를 코드북 검색에 이용하였다. 검색된 코드북 인덱스를 검색 메모리에 저장한 다음 신호 벡터의 코드북 검색 시에 우선적으로 검색 메모리를 참조하는 방법이다. 코드북 인덱스의 검색 메모리 상주 기간은 15ms로 사용하였으며, 제안한 방법으로 컴퓨터 시뮬레이션으로 수행해 본 결과 일반적인 음성 신호에서 3.2% - 11.7%의 검색 시간의 단축을 가져왔다. 향후, Hitting rate를 보다 높여 안정되고 균일한 수행성을 보장 하려면 더욱 정확한 검색 메모리의 수정 주기 설정과 검색 메모리에 대한 적절한 크기의 할당이 이루어져야 하며, 이를 위해 더 많은 종류의 음성 분석과 그에 대한 비교 평가가 연구되어져야 할 것이다.

참 고 문 헌

- [1] M. R. Schroeder and B. S. Atal, "Code

Exited Linear Prediction: High quality Speech at very low bit rates," Proc. Int. conf. Acoust., Speech, Signal Processing, pp937 - 940, 1985.

- [2] D. O Shaughnessy, *Speech Communication: Human and Machine*, Addison-Wesley Publishing Company, Inc. 1987.
- [3] J. Chen and M. S. Rauchwerk. "An 8kb/s low-delay speech coding algorithm," IEEE Speech Coding Workshop, pp 3-4, September 1991.
- [4] J. Chen, Y. C. Lin and R. V. Cox. "A fixed-point 16kb/s LD-CELP algorithm," Proc. of ICASSP, pp.21-24, 1991.
- [5] *Coding Of Speech At 16 kbit/s Using Low-Delay Code Excited Linear Prediction*, ITU-T, Recommendation G.728, 1992.
- [6] A. N. Inc, *Digital Speech Processing(speech coding, synthesis, and recognition)*, Kluwer Academic Publishers, 1992.
- [7] A. M. Kondoz, *Coding For Low Bit Rate Communications Systems*, John Wiley & Sons Ltd., 1994.
- [8] 강홍구, 서정태, 차일환, 윤대희, "구조화된 다중 펄스열을 이용한 낮은 전송율 음성 부호화기," 한국통신학회 논문집, 제20권 제7호, pp.1907-1917, 1995.
- [9] 신철선, 정재호, "가변 저 비트율 음성 부호기 개발 및 실시간 구현," 한국통신학회 논문집, 제20 권 제9호, pp.2525-2537, 1995.
- [10] 손종서, 김시현, 강지양, 성원용, "FS-CELP 음성 부호화기의 고정 소수점 성능 분석 및 구현," 한국통신학회 논문집, 제21권 제2호, pp. 365-374, 1996.

저 자 소 개



全 慶一(正會員)

1945년생. 1978년 인하대학교 전자 공학과 공학학사. 1980년 인하대학교 대학원 전자공학과 공학석사. 1989년 아주대학교 대학원 전자공학과 공학 박사. 1980년~현재 인하공업전문대학 전자과 교수. 관심분야: 디지털 시스템, 마이크로프로세서응용, 인공지능 등



金錫燦(正會員)

1961년생. 1985년 인하대학교 전자 계산과 이학사. 1987년 인하대학교 전자계산과 석사. 1995년 금성신전연 구소 선임연구원. 1996~현재 인하대학교 전자계산공학과 박사과정. 관심 분야: 마이크로프로세서응용, 실시간 시스템, 웹디자인 Communication Language 등