

벡터 양자화를 위한 고속 탐색 알고리즘에 관한 연구

A Study on the Fast Search Algorithm for Vector Quantization

지 상 현*, 강 상 원*, 김 용 석**, 이 남 일**
(Sang-Hyun Chi*, Sang-Won Kang*, Yong-Seok Kim**, Namil Lee**)

*한양대학교 전자전기제어계측공학과, **삼성전자 통신망 총괄 코아 기술 연구 그룹
(접수일자: 2002년 11월 11일; 수정일자: 2003년 5월 9일; 채택일자: 2003년 5월 19일)

본 논문에서는 NNVQ (nearest neighbor vector quantization)를 위한 고속 탐색 알고리즘을 제안하였다. 제안된 알고리즘은 입력 벡터 내 지정된 요소의 부호화 왜곡을 임계 값과 비교해서 최적 코드워드가 될 가능성이 없는 코드워드를 탐색 대상 코드워드에서 제외함으로써 코드북 탐색의 범위를 줄인다. 이를 통해 기존의 풀서치 (full-search) 알고리즘과 동일한 SD (spectral distortion) 성능을 얻으면서 부호화 과정에서의 처리시간과 복잡도를 감소시킨다. 제안된 알고리즘을 Gauss-Markov 소스에 대해 설계된 일반적인 벡터 양자화기와 3GPP에서 표준화된 AMR (adaptive multi-rate) 음성 코덱의 LSP 양자화에 각각 적용하여 실험하였다. 실험결과 제안된 알고리즘은 SD 성능감쇄 없이 탐색 대상 범위를 감소시킴으로써 부호화시 복잡도를 50% 이상 줄일 수 있었다.

핵심용어: 벡터 양자화, 고속 탐색, 음성 부호화, AMR

부고분야: 음성처리 분야 (2.2)

In this paper, we propose a fast search algorithm for nearest neighbor vector quantization (NNVQ). The proposed algorithm rejects those codewords which can not be the nearest codeword and reduces the search range of codebook. Hence it reduces computational time and complexity in encoding process, while it provides the same SD performance as the conventional full search algorithm. We apply the proposed algorithm to the adaptive multi-rate (AMR) speech coder and a general vector quantizer designed by LBG algorithm. Simulation results show effectiveness of the proposed algorithm.

Keywords: Vector quantization, Fast search, Speech coding, AMR

ASK subject classification: Speech signal processing (2.2)

1. 서론

벡터 양자화는 벡터 내 요소간의 상관도를 이용함으로써 같은 비트의 스칼라 양자화에 비해 더 작은 양자화 에러를 제공하기 때문에 데이터 압축 분야에서 가장 널리 사용되는 양자화 방법이다. 그러나 벡터 양자화 방법은 기본적으로 두 가지 단점을 가지고 있다. 첫번째는 벡터의 차원이 증가함에 따라 부호화시 요구되는 복잡도가 기하급수적으로 증가한다는 것이고, 두 번째는 벡터 차원이 증가함에 따라 코드워드 저장에 필요한 메모리 공간이 지수적으로 증가한다는 것이다.

복잡도에 대한 문제를 해결하기 위하여 많은 연구가 진행되어 왔다. 대표적인 방법은 PDS (partial distance search) 방법[1]과 삼각부등식을 이용하는 방식[2]을 들 수 있다. PDS 방법은 곱셈과 덧셈 연산을 대폭 감소시킬 수 있으나 비교 연산은 오히려 크게 늘어나는 단점이 있다. Orchard가 제안한 삼각부등식을 이용하는 방법은 각 코드벡터와 기준 벡터간의 거리에 관한 정보를 추가로 저장해야 한다는 단점이 있다. 본 논문에서는 메모리 증가와 성능감쇄 없이 벡터 양자화시 최적 코드벡터를 고속으로 찾을 수 있는 고속 부호화 알고리즘을 제안하였다. 제안된 방식은 주어진 코드북을 이용해서 입력 훈련용 벡터들의 최적 왜곡 (distortion) 값들을 구하고, 그러한 값들로부터 임계 값을 결정해서 입력 벡터의 한 요소 값과 최적 벡터의 해당 요소 값간의 왜곡이 이러한 임계

책임저자: 강상원 (swkang@hanyang.ac.kr)
425-822 경기도 안산시 사1동 1271번지
한양대학교 전자컴퓨터공학부
(전화: 031-400-5215; 팩스: 031-418-8476)

값보다 크면, 그러한 요소를 포함하는 코드워드를 탐색 범위에서 제외하여 코드북 내 탐색 대상 코드벡터를 결정한다. 또한 탐색 대상 코드벡터의 결정을 효율적으로 하기 위해 코드북 내 코드워드의 배열 순서를 해당 요소 값의 크기에 따라 순차적으로 배열시킨다.

본 논문의 구성은 다음과 같다. 제 2장에서는 NNVQ (nearest neighbor vector quantization) 부호화에 대하여, 제 3장에서는 설계된 고속 NNVQ 부호화에 대하여 기술하였고, 제 4장 및 5장에서는 제안된 알고리즘을 Gauss-Markov 신호에 대한 일반적인 벡터 양자화와 3DPP의 협대역 표준 음성 부호화 방식인 AMR 음성 코덱 나 LSP 양자화에 적용한 실험 결과를 기술하였고, 마지막으로 제 6장에서 결론을 맺는다.

II. NNVQ

NNVQ는 분할 셀이 코드북과 왜곡 크기 (measure)에 의해 완전히 결정되는 특징을 가지고 있다. 벡터 양자화는 일반적으로 NNVQ와 문헌적으로 동일한 의미로 사용된다. 이러한 NNVQ는 부호화 과정에서 코드북과 왜곡 크기 외 셀의 구조와 관련된 어떠한 정보도 요구하지 않는다. 그래서 부호화 과정이 매우 간단하다. NNVQ 방식에서 분할 셀은 다음 식으로 정의된다.

$$C_i = \{ \bar{x} : d(\bar{x}, \bar{y}_i) \leq d(\bar{x}, \bar{y}_j), \text{ for all } 1 \leq j \leq L \} \quad (1)$$

여기서 C_i 는 분할 셀이고, $d(\bar{x}, \bar{y})$ 은 왜곡 크기 값이며, \bar{x} 는 입력벡터, \bar{y}_i 는 코드 벡터, 그리고 L 은 코드북 크기를 의미한다. 즉, 셀 C_i 는 다른 코드 벡터로 부호화할 때보다 \bar{y}_i 로 부호화 하였을 때 더 작은 왜곡 값을 갖는 입력 벡터 \bar{x} 로 구성된다. 모든 셀들이 분할 형태를 이루기 위해서는 경계상의 값들은 반드시 하나의 셀에만 할당되어야 한다. 위의 정의는 입력 벡터 \bar{x} 를 왜곡 크기 $d(\bar{x}, \bar{y}_i)$ 가 최소가 되는 셀 C_i 의 요소로 할당함으로써 간단히 적용할 수 있다.

III. 고속 NNVQ 부호화

차원당 비트할당율이 α 이고, 차원 (dimension)이 k 라 가정하면, 벡터 양자화는 k 차원 벡터공간 R^k 을 이 공간의 유한 부분집합 $Y = \{ \bar{y}_i : i = 1, \dots, L \}$ 로 배치하는 것

이며, 여기서 $L = 2^{k\alpha}$ 이다. 일반적인 풀서치 (full-search) 방식으로 입력 벡터 \bar{x} 를 부호화하기 위해서는 먼저 N 개의 코드워드 각각에 대하여 입력 벡터와의 왜곡 값들을 구하고, 그 값들을 서로 비교해서 입력 값에 가장 근사한 코드워드를 구한다.

입력과 출력 벡터 사이의 거리를 나타내는 왜곡 크기 $d(\bar{x}, \bar{y})$ 의 가장 간단한 예로써, 두 벡터 사이의 자승 유클리디언 왜곡이라 불리어지는 자승 에러 왜곡 크기를 들 수 있다.

$$d(\bar{x}, \bar{y}_i) = \|\bar{x} - \bar{y}_i\|^2 = \sum_{j=1}^k (x_j - y_{ij})^2, \quad i=1, \dots, L \quad (2)$$

위 식에서 x_j 와 y_{ij} 은 각각 입력 벡터 \bar{x} 와 codeword \bar{y}_i 의 j 번째 요소이다.

자승 유클리디언 왜곡 크기에서 각 왜곡 값을 구하는데 k 개의 곱하기 연산과 $2k-1$ 개의 더하기 연산이 요구된다. 그래서 각 입력 벡터에 대하여 kL 개의 곱하기 연산, $(2k-1)L$ 개의 더하기 연산, 그리고 L 개의 비교 연산이 요구된다. 그러므로 부호화 과정의 복잡도는 탐색되는 코드북 크기 L 에 비례함을 알 수 있다. 따라서 탐색 되는 코드북의 사이즈가 작으면 부호화 과정의 복잡도 역시 작아진다.

탐색 코드북의 크기를 줄이기 위한 방법으로서 벡터의 MSE는 벡터내 요소의 MSE보다 항상 크다는 사실을 이용한다. 그래서 훈련 벡터 MSE 값들의 통계 치들로부터 임계 값을 결정하고 입력 벡터내 특정 요소와 최적 코드벡터의 해당 요소간의 왜곡 값이 임계 값을 초과하지 않는 코드 벡터들을 선정한다. 즉 특정 요소간의 왜곡 값이 임계 값보다 큰 코드 벡터들은 입력벡터의 최적 코드 워드 후보가 될 수 없다. 이런 성질을 이용하여 입력벡터 \bar{x} 와 코드벡터 \bar{y}_i 간의 거리를 구하는 대신, 특정 요소간의 거리만을 구함으로써 적은 계산량으로 많은 수의 코드벡터들을 탐색 대상에서 제외시킬 수 있다. 탐색 대상 범위를 결정시 사용하는 벡터내 특정 요소는 벡터내 요소들 중 탐색구간을 가장 많이 감소시키는 요소를 실험적으로 구한다.

모든 \bar{y}_i 에 대하여 임계 값은 $thrs = \rho \times \sqrt{dist}$ 라 하고, 여기서 ρ 는 조절 인자이다.

$$dist = \max_{\text{for all } \bar{x}} \left\{ \min_{i=1, \dots, L} \|\bar{x} - \bar{y}_i\|^2 \right\} \quad (3)$$

즉, $dist$ 는 모든 입력벡터에 대한 최소 자승 에러 값들 중 최대 값이다. 만약 주어진 입력벡터 \vec{x} 의 특정 요소 x_j 에 대하여 다음 식이 만족한다면

$$|x_j - y_{ij}| < thr_s$$

i.e. $x_j - thr_s < y_{ij} < x_j + thr_s$, (4)

위의 식을 만족하는 코드벡터 \vec{y}_j 는 최적 코드워드가 존재하는 범위 내의 벡터이다. 그런데 y_{ij} 가 식 (4)의 범위 밖에 존재한다면, \vec{y}_j 는 입력 벡터 \vec{x} 의 최적 코드워드 후보 벡터로 볼 수 없다. 이와 같이 정해진 탐색구간내의 코드벡터들에 대하여 풀서치 알고리즘을 적용한다. 탐색구간의 시작점과 끝점은 위 식에서 알 수 있듯이 임계값과 입력벡터의 요소 값의 크기에 의해 결정된다.

합당한 $dist$ 값은 충분히 많은 훈련 벡터들을 이용함으로써 구할 수 있다. 제안된 알고리즘을 효율적으로 적용하기 위해, 0과 1사이의 값을 갖는 조절인자 ρ 를 이용하여 thr_s 값을 조절할 수 있다.

요소간 왜곡 값과 thr_s 값을 이용해서 풀서치를 위한 탐색범위 결정을 간단하게 하기 위해, 코드벡터의 지정된 요소들의 크기에 따라 코드북을 내림차순으로 재정렬한다. 재정렬된 코드북을 이용해서 탐색범위를 결정하는 개념이 그림 1에 나타나 있다. 실제로 특정 요소의 크기 순으로 재정렬된 코드북의 시작점과 끝점을 효율적으로 결정하기 위하여 이진 (binary) 탐색 방법[3]을 사용하였다. 이러한 이진 탐색 방법을 이용하여 탐색구간을 결정하는데, $2 \times \log_2 L$ 개의 추가적인 비교 연산이 요구되지만, 탐색범위 감소에 의한 연산량 감소에 비하면 추가된 연산은 매우 적다.

제안된 알고리즘을 이용하여 벡터 양자화를 수행하기 위해서는 임계값 thr_s 을 훈련용 벡터들을 이용해서 통계적으로 결정하고, 아래와 같이 코드북을 재정렬하여야 한다.

- Step 1 : 코드북 Y 의 재정렬에 사용될 특정 요소 j 를 결정. 훈련 벡터들을 이용한 실험 중 최대의 탐색구간 감소를 보이는 요소를 특정 요소로 결정.
- Step 2 : 특정 요소 j 의 크기에 따라 코드북을 내림차순으로 재정렬.

코드북이 재정렬된 후 제안된 알고리즘의 부호화 과정은 다음과 같다.

- Step 1 : 입력벡터 \vec{x} 와 임계값 thr_s 를 이용하여 코드북 탐색구간의 시작점과 끝점을 이진 탐색 방식으로 결정.
- Step 2 : 결정된 구간내의 코드벡터들에 대하여 부호화를 수행한다.

IV. AMR 음성 코덱에의 적용

제안된 알고리즘을 AMR 음성 코덱의 LSP 양자화에 적용하였다[5]. AMR 음성 코덱은 채널의 상태에 따라 변화하는 8개의 전송 모드가 존재하며, 전체 모드에 대한 LSP 양자화는 크게 SMQ (split matrix quantizer)와 SVQ (split vector quantizer) 방식으로 나눌 수 있으며, SVQ는 할당되는 비트에 따라 다시 3개로 분류될 수 있다. 각 양자화는 1차 MA 예측기를 사용하며 LSP 양자화

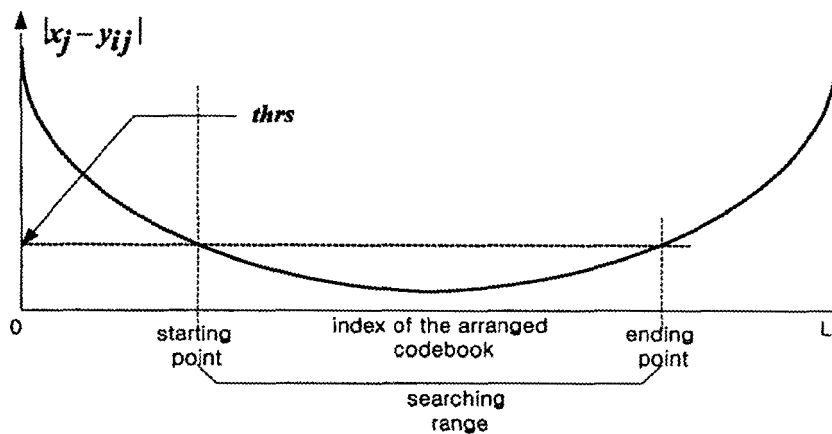


그림 1. 재정렬된 코드북에서 탐색 범위의 결정
Fig. 1. Determination of the search range using the arranged codebook.

시 다음식과 같은 에러 기준 값으로 가중된 자승 유클리디언 왜곡 크기값을 이용한다.

$$E_{LSP} = \sum_{i=1}^{10} [L_i w_i - L_i^* w_i]^2 \quad (5)$$

이거시 w_i 는 가중치이며, 가중치의 결정은 아래와 같다.

$$w_i = 3.347 - \frac{1.547}{450} d_i, \text{ for } d_i < 450, \\ = 1.8 - \frac{0.8}{1050} (d_i - 450), \text{ otherwise.} \quad (6)$$

where, $d_i = L_{i+1} - L_{i-1}$ with $L_0 = 0$ and $L_{11} = 4000$

그림 2와 3은 AMR 음성 코덱에서 사용되는 LSP 양자화기 SMQ와 SVQ의 블록 다이어그램을 나타낸 것이다. 각 부 벡터내 특정 요소는 충분히 긴 훈련용 신호를 사용하여 평균 탐색 범위를 최소화시키는 요소를 선택한다. 또한 각 부 벡터별 임계 값을 구한다. 이러한 개념을 이용

하여, SMQ에서는 각 부 매트릭스 내에서 $r_2^{(2)}, r_3^{(2)}, r_6^{(1)}, r_7^{(2)}$, 와 $r_{10}^{(2)}$ 를 선택하였고, SVQ에서는 각 부 벡터 내에서 r_3, r_5 그리고 r_7 을 특정 요소로 선정하였다: $r_b^{(a)}$ 는 a번째 LSF 예측 에러 벡터의 b번째 요소값을 의미한다. AMR 음성 코덱의 LSP 코드북은 결정된 특정 요소의 크기 값을 기준으로 내림차순으로 재정렬시켰다. 이렇게 재정렬된 코드북과 임계값을 이용하여 고속 부호화 과정이 수행되었고 실험결과는 다음 장에 기술되어 있다.

V. 실험 및 결과

제안된 알고리즘을 두 개의 양자화기에 적용하여 실험하였다. 첫 번째는 LBG 알고리즘을 사용하여 설계된 일반적인 양자화기이다. 이때, training 벡터는 음성 신호의 모델링으로 많이 사용되고 있는 $\rho=0.86$ 인 1차 Gauss-

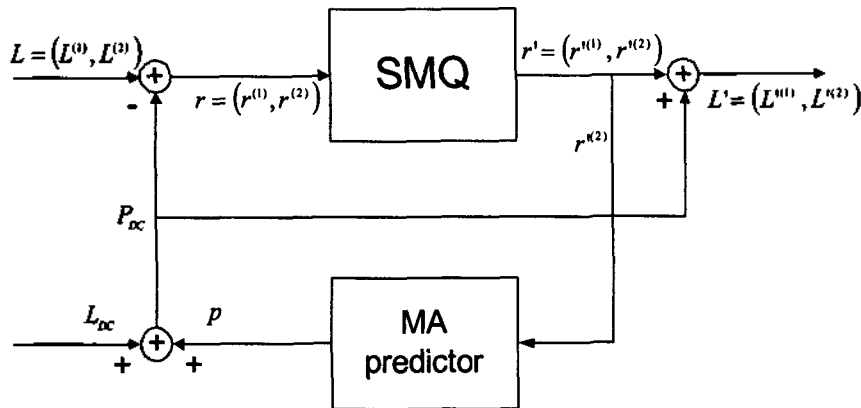


그림 2. 예측 SMQ 구조에 대한 블록 다이어그램
Fig. 2. Structure of the predictive split matrix quantizer.

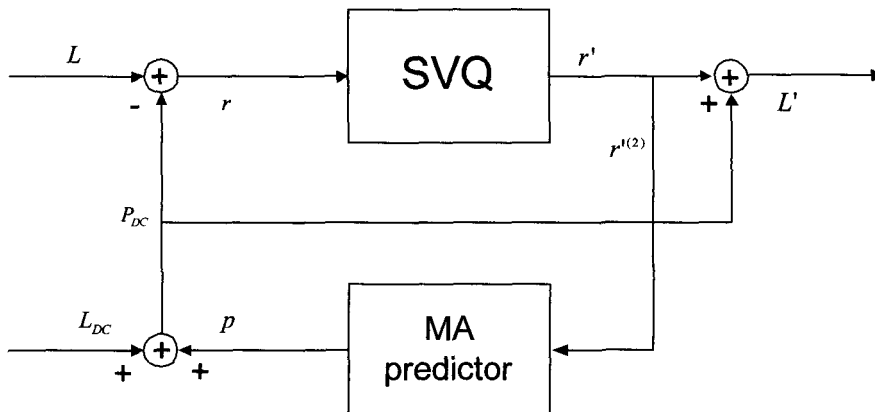


그림 3. 예측 SVQ 구조에 대한 블록 다이어그램
Fig. 3. Structure of the predictive split vector quantizer.

Markov 소스가 이용되었다. (0.86은 음성신호의 1차 자기 상관계수로 널리 사용되고 있는 값이다.) 두 번째는 기존의 AMR 음성 코덱에서 사용되고 있는 LSP 양자화기에 적용하여 실험하였다.

일반적인 벡터 양자화기 경우의 실험은 벡터의 차원을 2차에서 4차로 변화시키는 동시에 차원 당 할당되는 비트를 각각 2비트 또는 3비트 할당하여 탐색 범위의 감소량을 측정하는 방법으로 이루어졌다. AMR 음성 코덱에 적용한 경우에는 NTT에서 제작 배포한 음성을 입력신호로 사용하여 LSP계수의 벡터 양자화시 탐색 범위의 감소량을 계산하였다.

임계 값은 다음 수식에 의하여 결정하였다.

$$thr_i = p_i \times \sqrt{dist} \tag{7}$$

여기서 p_i 는 조절 인자이고, $dist$ 는 벡터의 최적 에러 값의 최대값을 의미한다.

Gauss-Markov 신호에 대해서는 두 개의 다른 임계 값 $Thrs_1$ 및 $Thrs_2$ 을 적용하였다. $Thrs_1$ 은 제안된 알고리즘의 출력 인덱스 값과 풀서치 알고리즘에 의한 출력 인덱스 값간의 인덱스 에러가 존재하지 않는 경우의 임계

표 1. 조절 인자 p 에 따른 인덱스 에러율과 SNR 성능
Table 1. The SNR performance and the index error for the control factor p .

1.00	11.1	18.30	0
0.50	5.56	18.30	0
0.30	3.34(= thr_1)	18.30	0.001
0.20	2.23	18.30	0.004
0.10	1.11(= thr_2)	18.29	0.010
0.05	0.56	18.22	0.162
0.01	0.11	17.49	6.348

표 2. 두 임계 값에 대한 평균 탐색범위
Table 2. The average search range of the codebook for two thresholds thr_1 and thr_2 .

2	16	10.59	33.82	5.39	77.73
	64	30.58	52.22	24.83	61.20
	256	152.08	46.14	113.81	59.70
3	64	40.76	36.78	15.40	75.94
	512	208.26	59.32	133.49	73.93

값이고, $Thrs_2$ 는 인덱스 에러가 0.1% 존재할 경우의 임계 값을 나타낸다. 실험을 위해 50,000 개의 훈련용 샘플을 사용하였고, 100,000 개의 테스트용 샘플을 사용하였다. 제안된 방식에 의한 효율적인 탐색 시작점과 끝점을 찾기 위하여 LBG 알고리즘[4]에 의하여 설계된 코드북을 특정 요소의 크기 값에 따라 내림차순으로 정렬하였다.

표 1은 2차원 벡터에 차원 당 3 비트가 할당되었을 경우에 임계 값 $Thrs_1$ 및 $Thrs_2$ 을 결정하는 예를 보여주고 있다. $Thrs_1$ 은 제안된 알고리즘과 풀서치 알고리즘의 SNR 성능 차이가 없는 최소 값으로 선택되었고, $Thrs_2$ 는 실험적으로 SD (spectral distortion)값이 소수점 이하 셋째 자리까지 유효값으로 보았을 때 SD 값이 변함없는 값을 기준으로 SNR 성능 차이가 0.01 dB 미만으로 발생하는 한도에서의 최대값으로 선택되었다. 제어 요소 $p=1$ 인 경우는 풀서치 알고리즘에 의한 출력 인덱스와 동일한 결과를 얻을 수 있다. 이는 벡터내 한 요소에 대한 최적에러 값의 최대값은 벡터에 대한 최적 에러 값의 최대값보다 항상 작기 때문이다.

표 2는 표 1에서 결정된 두 개의 임계 값을 이용하여 할당 비트와 벡터의 차원을 변화시키면서 실험한 결과를 나타낸다. 풀서치 알고리즘과 비교시, 제안된 알고리즘에 임계 값 thr_1 을 적용하였을 경우 코드북 탐색 범위가 최소 33.82%에서 최대 59.32%까지 감소하였고, thr_2 을 적용하였을 경우 코드북 탐색 범위가 최소 59.70%에서 최대 77.73%까지 감소하였다.

AMR 코덱의 LSP 벡터 양자화에 적용하기 위하여 1,682,080 개의 음성 샘플 (10,513 프레임)을 사용하였다. 표 3과 4의 결과는 풀서치 알고리즘의 SNR성능과 동일한 성능을 얻었을 경우의 평균 탐색 범위와 탐색 범위의 감소율을 나타낸다.

12.2 kbps 모드에 제안된 알고리즘을 적용 하였을 경우 코드북 탐색 범위가 프레임 당 58.07% 감소함을 표 3에서 알 수 있다. 나머지 7개 모드에 적용할 경우 코드북 탐색

표 3. 12.2 kbps 모드에 대한 실험 결과
Table 3. Test results for 12.2 kbps mode.

1차 부행렬	128	62.42	51.24
2차 부행렬	256	85.26	66.70
3차 부행렬	부호 0	256	87.59
	부호 1	256	174.27
4차 부행렬	256	127.06	51.37
5차 부행렬	64	54.27	15.21
프레임	1216	509.97	58.07

표 4. 나머지 7개 모드에 대한 실험 결과
Table 4. Test results for other 7 modes except 12.2 kbps.

모드	코드북 크기	평균 탐색 범위	감소율 (%)
13.2, 7.4, 6.7, 5.9 kbps 모드	1280	492.90	61.50
5.1, 5, 4.75 kbps 모드	640	244.09	61.87
7.35 kbps 모드	1536	551.94	64.07

범위를 최소 61.05%에서 최대 64.07%까지 감소할 수 있음을 표 4에서 알 수 있다.

VI. 결론

본 논문에서는 NNVC를 위한 고속 부호화 알고리즘을 제안하였다. 제안된 알고리즘은 기존의 풀서치 알고리즘에 비하여 SD 성능 감소와 추가적인 메모리 요구없이 탐색 대상 코드북의 크기를 줄임으로써 부호화 과정의 복잡도를 감소시켰다. Gauss-Markov 소스의 경우, 두 개의 임계값 $Thrs_1$ 과 $Thrs_2$ 을 적용하여 실험한 결과 풀서치 탐색방법에 비하여 코드북 평균 탐색 범위가 $Thrs_1$ 인 경우 최소 33.82%에서 최대 59.32%까지 감소하였고, $Thrs_2$ 인 경우 최소 59.70%에서 최대 77.73%까지 감소하였다. AMR 코덱의 LSP 모듈은 Oak DSP 구현시 인코더 전체 계산량 23.3 MIPS (million instructions per second)의 9.23%에 해당하는 4.5 MIPS의 계산량을 요구한다. (본 연구실에서 개발한 DSP 솔루션을 기준으로 하였다.) LSF 모듈 계산량의 82.22%에 해당하는 3.7 MIPS의 계산량을 필요로 하는 LSP 양자화 부분에 제안된 알고리즘을 적용한 결과, 코드북 평균 탐색 범위가 최소 58.07%에서 최대 64.07%까지 감소되어 약 2.22 MIPS의 계산량 감소 효과를 얻었다.

감사의 글

본 연구는 1999년 삼성전자(주)와 한양대학교 교내 연구비에 의하여 수행되었음.

참고 문헌

1. C. D. Bei and R. M. Gray, "An improvement of the minimum distortion and palteren matching," *IEEE Trans. Commun.*, COMM-33, 1132-1133, Oct, 1985.
2. M. T. Orchard, "A fast nearest neighbor search algorithm," *Proc. ICCASP*, 2297-2300, May 1985.
3. A. Lowry, S. Hossain, and W. Millar, "Binary search trees for vector quantization," *Proc. ICASSP*, 2205-2208, May 1987.
4. Y. Linde, A. Buzo and R. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Commun.*, 84-95, Jan, 1980.
5. ETSI, "Digital cellular telecommunications system; Adaptive Multi-Rate speech transcoding," Feb, 1999.

저자 약력

• 지 상 현 (Sang-Hyun Chi)



2000년 2월: 한양대학교 전자컴퓨터공학부 졸업 (공학사)
2002년 2월: 한양대학교 전자전기 제어계측공학과 졸업 (공학석사)
2002년 9월~ 현재: 삼성종합기술원, 연구원

• 강 상 원 (Sang-Won Kang)

한국음향학회지 제20권 제4호 참조

• 김 용 석 (Yong-Seok Kim)



1983년 2월: 성균관대 전자공학과 졸업
1998년 1월: 삼성전자 입사
1998년 3월~2000년 2월: 성균관대 겸임교수
1991년 5월: 발명의 날 '대통령상'
2000년 2월: 삼성그룹 '기술상 대상'
2000년 4월: 정보통신의 날 '국무총리상'
2002년 12월: 사내 명예박사
2003년~ 현재: 삼성전자 통신연구소 모델링과
Lab장/수석연구원

• 이 남 일 (Namil Lee)



1998년 2월: 한양대학교 제어계측공학과 (공학사)
2001년 2월: 한양대학교 전자전기 제어계측공학과 (공학석사)
2001년 3월~ 현재: 삼성전자 통신연구소 IMT-2000
단말연구팀 연구원