

벡터의 발생 순서를 이용한 효율적인 벡터양자화

김동환, 윤재선, 홍광석

성균관대학교 전기전자 및 컴퓨터공학부 휴먼컴퓨터 연구실

An Effective Vector Quantization using Generating Sequence of the Vector

Dong-Hwan Kim, Jeh-Seon Youn, Kwang-Seok Hong

HCI Lab, Electrical & Computer Engineering, SungKyunKwan University

kdh2k@popsmail.com, sunhci@ece.skku.ac.kr, kshong@yurim.skku.ac.kr

요약

벡터양자화는 신호의 압축에 이용되는 일반적인 방법이다. 그러나 유클리드 거리 등을 이용한 거리 계산량이 많아서 코드북 크기나 압축율의 제한이 있게 된다. 따라서 PDS(partial distance search)와 같은 벡터양자화 부호화의 계산량을 줄이기 위한 많은 방법들이 제안되고 있다. 본 논문에서는 이웃한 음성신호는 급격히 변하지 않고 서서히 변해가는 성질에 착안하여 현재의 벡터 다음에 발생하는 벡터를 조사하여 인덱스를 저장한 후 이를 다음 벡터의 벡터양자화때 참고함으로써 불필요한 계산을 줄이는 방법이다.

제안한 방법으로 음성신호에 대해 실험한 결과 전탐색의 결과와 비교하여 빠른시간에 큰 오차없이 벡터양자화 부호화를 할 수 있었다. 이 방법은 PDS와 같은 이미 제안되어 있는 많은 방법들과 같이 이용하면 더욱 효과적인 벡터양자화 부호화를 할 수 있을 것이다.

I. 서론

벡터양자화는 신호처리, 음성인식 등에 이용되는 효율적인 압축기법이다. 벡터양자화의 부호화는 코드북에서 가장 거리가 가까운 코드벡터를

찾아내어 찾아낸 벡터의 인덱스로 정보를 표현하게 된다. 가장 일반적인 거리척도로 유클리드(Euclidean) 거리를 주로 사용한다.

벡터양자화가 이론적으로는 가장 효과적인 압축방법이지만 부호화에서 거리가 가까운 코드벡터를 찾는 거리계산량이 많아서 코드북 크기와 압축률의 제한이 있게 된다.

따라서 코드북 탐색시 많은 계산량의 문제를 해결하기 위하여 트리구조를 갖는 TSVQ (tree structured vector quantization), K-d[k-dimensional] tree[1], 다음 상태에 대한 함수를 이용하는 FSVQ(finite-state VQ)[1] 등이 있고, 전탐색과 유사한 PDS[1], RPDS(rotated PDS)[2], FNNS(fast nearest neighbor search), ENNS(equal-average neighbor search), EENNS(equal-average equal-variance nearest neighbor search)[3] 등과 같은 다양한 탐색방법이 제안되었다.

본 논문에서는 음성신호는 단시간 구간 분석의 연결로 관찰하여 볼 때 그 특징이 급격하게 변화하지 않고 서서히 변화하는 성질[4]에 착안하여 현재 발생된 벡터에 연이어 발생하는 벡터의 종류를 조사한 후 이 벡터들의 인덱스를 테이블로 구성하였다. 부호화때 이 테이블을 참조하여 코드북을 탐색함으로써 코드북 탐색의 시간을 줄이는 방법이다. 따라서 제안한 방법은 FSVQ 계열

에 속한다고 할 수 있다. 제안한 방법의 타당성을 확인하기 위하여 음성 코퍼스를 새로이 구축하여 실험하였다.

제안한 방법은 모든 코드벡터와 비교하는 것이 아니라 인덱스 테이블에 있는 벡터만을 탐색하기 때문에 PDS와 같은 다른 방식의 탐색기법과 함께 사용하면 더욱 빠른 탐색이 가능하게 될 것이다.

II. 음성신호의 특성

음성신호를 스펙트로그램으로 관찰해 보면 주파수 특성이 서서히 변하고 있다는 것을 알 수 있다. 그것은 음성 발생기관이 관성의 원리에 의해 급격히 변하지 않는 성질과 유관할 것이다. 이러한 성질을 알아보기 위해서 음성의 스펙트럼과 스펙트럼 포락에 대하여 살펴본다. 그림 2.1에 음성 “안녕하세요”에 대한 스펙트럼을 나타내었다.

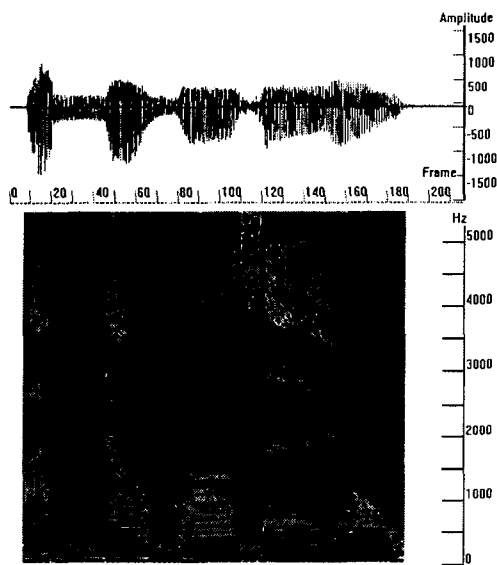


그림 2.1 “안녕하세요”에 대한 스펙트로그램

그림 2.1에서 발생되는 음운의 경계부분을 제외하면 대부분의 구간에서 시간상으로 인접한 곳에서의 특성이 유사하다는 것을 볼 수 있다.

음성신호를 스펙트럴 포락의 시간변화로 관찰해 보면 그 특성이 서서히 변한다는 성질을 좀더 뚜렷하게 알 수 있다. 그림 2.2에 음성 “아”에 대

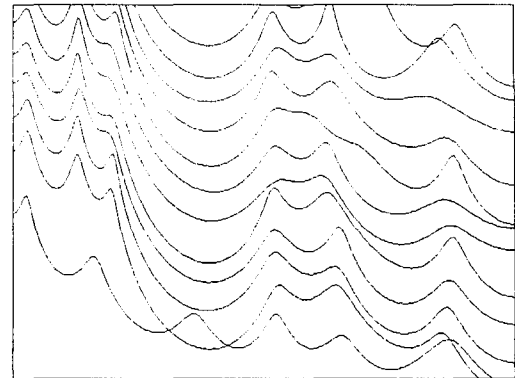


그림 2.2 “아”에 대한 스펙트럼 포락의 시간변화
한 스펙트럴 포락의 시간변화를 나타내었다.

그림 2.2는 연속된 12 프레임에 대한 스펙트럴 포락으로서 산과 골의 변화가 이웃한 프레임과 유사하다는 것을 관찰할 수 있다.

이와같은 음성신호의 성질은 음성의 특징벡터 발생에 영향을 주어 현재 출현한 벡터 다음에 출현하는 벡터는 유사한 벡터를 갖게 될 것이다. 이러한 성질을 살펴보기 위하여 벡터의 발생을 조사하여 그림 2.3에 표현하였다.

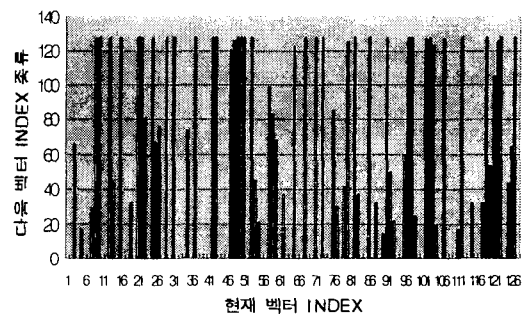


그림 2.3 벡터의 발생수

그림 2.3은 128개의 코드벡터(인덱스는 0부터 127로 표기)를 갖는 코드북을 구성하고 이를 이용하여 조사한 것으로서 현재 시점의 벡터 다음에 연이어 나오는 벡터 종류의 결과를 살펴보면, 하나도 없는 것 부터 전부 나오는 경우도 있는데 다음에 나오는 벡터 수의 평균은 49.30469 이고 표준편차는 55.16968 이다. 표 2.1은 그림 2.3을 수치로 표현한 것이다.

표 2.1 다음 벡터 발생 수

0	1	2	3	4	5	6	7	8	9
0	0	66	0	17	0	0	29	127	128
10	11	12	13	14	15	16	17	18	19
0	0	128	43	0	128	0	0	32	0
20	21	22	23	24	25	26	27	28	29
128	128	81	0	128	67	76	0	128	0
30	31	32	33	34	35	36	37	38	39
128	0	0	0	74	0	128	0	0	0
40	41	42	43	44	45	46	47	48	49
0	128	128	0	0	0	121	126	128	128
50	51	52	53	54	55	56	57	58	59
128	0	128	45	21	0	0	99	83	68
60	61	62	63	64	65	66	67	68	69
0	37	0	0	123	0	0	128	0	0
70	71	72	73	74	75	76	77	78	79
128	0	128	0	0	85	30	0	41	125
80	81	82	83	84	85	86	87	88	89
0	128	37	0	0	128	0	32	0	15
90	91	92	93	94	95	96	97	98	99
128	50	22	0	0	60	128	128	25	0
100	101	102	103	104	105	106	107	108	109
0	127	128	124	19	0	127	0	0	0
110	111	112	113	114	115	116	117	118	119
17	128	0	0	32	0	0	32	128	53
120	121	122	123	124	125	126	127		
105	126	128	0	43	65	128	0		

이와같은 특성을 이용하여 본 논문에서는 현재 매핑된 벡터에 연이어 나오는 벡터의 인덱스를 표로 만들어 두고 코드북 탐색시 표를 참고하여 탐색하는 새로운 벡터양자화 부호화 방법을 제안한다.

III. 벡터의 발생순서를 이용한 벡터양자화

그림 3.1에 제안한 벡터양자화 부호화의 개념을 도시하였다. 음성신호가 들어오면 분석하여 얻은 특징벡터를 코드북과 매핑할 때 모든 코드벡터와 거리비교를 하는 것이 아니라 표에 제시된 코드벡터와 비교함으로써 매핑 시간을 줄이게 된다. 제안한 방법은 인덱스를 저장한 표를 사용하기 때문에 부호화시 메모리를 추가한다는 단점이 있지만 불필요한 탐색을 줄이는 장점이 있게 된다.

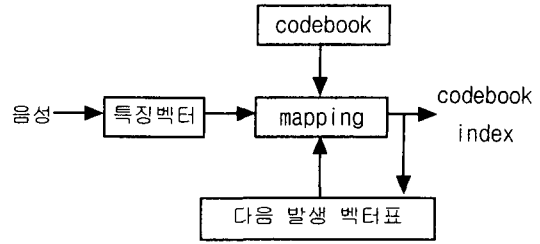


그림 3.1. 제안한 VQ 부호기

IV. 실험 및 결과

실험에 사용한 음성데이터는 PBW가 포함된 1,001개 단어를 60명이 발성한 것과 성명 1,145 단어를 62 명이 발성한 음성을 사용하였다. 음성은 16bit 11.025kHz로 샘플링하여 멜 캡스트럼 분석을 하였다. PBW가 포함된 단어데이터의 벡터 수는 총 5,998,718개이고, 성명데이터의 경우 총 6,665,698개의 벡터를 포함한다.

그림 4.1에 PBW 데이터를 이용하여 만든 코드북에 제안한 표를 추가하여 만든 부호화기를 이용하여 성명 데이터를 부호화 한 결과와 전 탐색 하였을 때와의 차이가 나는 인덱스 수를 나타내었다.

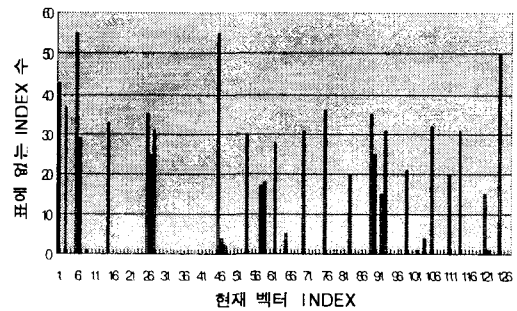


그림 4.1 PBW 벡터 표와 어긋나는 성명데이터 벡터 수

그림 4.1에서 PBW 벡터 표와 어긋나는 성명 데이터 벡터 수의 평균은 6.375 이고 표준편차는 13.28429 이다. 표 4.1은 그림 4.1을 수치로 표현한 것이다.

표 4.1 표에 없는 벡터의 발생 수

0	1	2	3	4	5	6	7	8	9
43	0	37	0	0	55	29	0	1	0
10	11	12	13	14	15	16	17	18	19
0	0	0	0	33	0	0	0	0	0
20	21	22	23	24	25	26	27	28	29
0	0	0	0	0	35	25	31	0	0
30	31	32	33	34	35	36	37	38	39
0	0	0	0	0	0	0	0	0	0
40	41	42	43	44	45	46	47	48	49
0	0	0	0	0	55	4	2	0	0
50	51	52	53	54	55	56	57	58	59
0	0	0	30	0	0	0	17	18	0
60	61	62	63	64	65	66	67	68	69
0	28	0	0	5	0	0	0	0	31
70	71	72	73	74	75	76	77	78	79
0	0	0	0	0	36	0	0	0	0
80	81	82	83	84	85	86	87	88	89
0	0	20	0	0	0	0	0	35	25
90	91	92	93	94	95	96	97	98	99
0	15	31	0	0	0	0	0	21	0
100	101	102	103	104	105	106	107	108	109
0	1	0	4	0	32	0	0	0	0
110	111	112	113	114	115	116	117	118	119
20	0	0	31	0	0	0	0	0	0
120	121	122	123	124	125	126	127		
15	1	0	0	50	0	0	0		

제안된 부호화기의 성능을 평가하기 위하여 성명 음성 데이터를 이용하여 실험하였을 때 표에 나타나지 않는 어긋나는 벡터 수를 구하여 전체 벡터수로 나누면 다음과 같다.

$$\frac{\text{어긋나는 벡터 수}}{\text{전체 벡터 수}} = \frac{4011}{6665698} = 0.0006017$$

이 결과는 제안한 방법이 전탐색의 결과와 근소한 차이를 보인다는 것을 알 수 있다. 그러나 탐색시간은

$$\frac{49.30469}{128} = 0.3875366$$

으로 전탐색의 약 38.7%의 시간이 소요된다는 장점을 갖는다.

V. 결론

본 논문에서는 코드북 탐색시 음성신호의 성질을 이용하여 제한적으로 탐색을 하여가는 효율적인 코드북 탐색 방법을 제안하고 그 유효성을 확인하였다.

제안한 방법은 부호화에서 인덱스 표를 위한 약간의 메모리가 부가되지만 미미한 성능 저하와 함께 탐색시간은 전탐색의 약 38.7% 정도만 소요된다. 따라서 제안된 VQ 부호화 알고리즘은 기존의 부호화 방법과 결합하여 사용하면 더욱 효율적인 부호화 방법이 될 것이다.

향후 연구는 코드북 레벨수를 늘려서 다양한 코드북 크기에서의 성능확인을 하고, 스펙트럼 왜곡이나 S/N비 등의 평가척도를 이용한 평가가 이루어져야 할 것이다.

※ 본 연구는 “스피치인 테크놀러지”의 연구비 지원으로 이루어졌습니다.

참고문헌

- [1] Allen Gersho, Robert M. Gray, "Vector Quantization and Signal Compression", KAP, 1991.
- [2] J. McNames, "Rotated Partial Distance Search for Faster Vector Quantization Encoding", IEEE Signal Processing Letters, Vol.7, No.9, 2000.
- [3] 백성준, 이대룡, 전범기, 성평모, "고속 VQ 부호화 알고리즘", 한국음향학회지, 15권 5호, 1996.
- [4] S.W. Ha, S.H. Yoon, K.W. Chung, K.S. Hong, "Vector Quantization using Signal Property", SICOPS'96, 1996.