

벡터 양자화에서 벡터의 특성을 이용한  
단축 탐색 방법

고 종석, 김 재균, 김 성대  
한국 과학 기술원, 전기 및 전자 공학과

A Simple Algorithm for Fast Codebook Search  
in Image Vector Quantization

KOH Jong-Seog, KIM Jae-kyoon, and KIM Seong Dae  
Dept. of Electrical Engineering, KAIST

ABSTRACT

We present a very simple algorithm for reducing the encoding (codebook search) complexity of vector quantization (VQ), exploiting some features of a vector currently being encoded. A proposed VQ of 16 (=4x4) vector dimension and 256 codewords shows a slight performance degradation of about 0.1-0.9 dB, however, with only 16 or 32 among 256 codeword searches, i.e., with just 1/16 or 1/8 search complexity compared to a full-search VQ. And the proposed VQ scheme is also compared to and shown to be a bit superior to tree-search VQ with regard to their SNR performance and memory requirement.

I. 서 론

벡터 양자화기 (Vector Quantizer ; VQ)의 큰 문제점 중의 하나가 바로 최적 벡터를 찾기 위한 복잡도이다. 지금까지 TSVQ (Tree Search VQ) [1,2] 또는 multi-stage VQ [3] 등의 여러가지 방법이 이러한 탐색 (search) 복잡도를 줄이기 위해 제안되었다. 이진 (binary) TSVQ의 경우 2배의 기억소자를 필요로 하고 약간의 성능 감소를 수반하기 때문에 본 논문에서는 이 두가지 문제를 개선시킨 SCVQ (split Codebook VQ)를 제안한다. 참고 문헌 [4]에 언급된 바와 같이 multi-stage VQ는 근본적인 알고리즘상의 준최적성 때문에 TSVQ에 비해서조차 성능이 떨어지므로 더이상 비교대상으로 삼지 않는다.

II. SCVQ Algorithm

본 논문에서 제안하는 기본적인 SCVQ (Split codebook VQ)의 특징은 부호화하려는 벡터의 특성 (feature)을 이용하여 원래 모두 탐색하려던 codebook의 크기를 줄이는 데에 있다. 이렇게 함으로써 계산을 통한 탐색 수를 현격히 줄이게 되는 것이다. 반면에 탐색하는 codebook의 크기가 작아지므로 최적의 벡터를 찾지 못할 가능성이 높아져 성능 감소를 수반하게 된다. 여기에서 원래 탐색하려던 codebook을 super codebook, SCVQ에 의해 탐색하려는 작은 codebook을 partial codebook 이라 부르기로 한다.

그림 1을 참고로 하여 partial codebook의 분리 선택 과정을 알아 본다. 크기가  $N$  (즉 code vector 의 수가  $N$ ) 인 super-codebook 에서 미리 결정된 분리선택 방법에 따라 크기가  $M (\leq N)$  인 partial codebook을 분리 선택한다. 그러나 이러한 분리선택방법이 복잡하지 않도록 현재 부호화하려는 벡터의 간단한 특성만을 이용한다. 그 특성으로는 벡터 성분들의 평균 또는 분산 등을 모두 이용할 수 있다. 본 논문에서는 첫 단계로서 간단한 덧셈만을 필요로 하는 평균을 이용하였다. 이와 같은 조건을 가진 SCVQ 알고리즘은 다음과 같이 두 단계로 나타낼 수 있다.

2-1. VQ Codebook의 설계 및 재배열 과정

- (1) 일반화된 Lloyd algorithm [5] 을 이용하여 크기가  $N$  인 super codebook을 설계한다.
- (2) 벡터 성분의 평균값의 크기 순으로 super codebook을 재배열한다.

- (3) 재배열된 super codebook과 그 벡터 성분의 평균값을 저장한다.

2-2. VQ 탐색 과정 (그림 2참조)

- (1) 하나의 입력 벡터  $X$  을 받아 그 성분들의 평균값  $m_x$  를 계산한다.  
 (2) 위의 2-1절의 (3)번 과정에서 저장되어 있는 평균값으로부터  $m_x$  와 가장 유사한 값의 index  $J$  를 찾는다. 이 과정은 스칼라 양자회기로 간단히 처리할 수 있다. 하드웨어로 구현할 때 영상의 pel 해상도와 같은  $2^8$  address bits 와  $\log_2 N$  outputs 를 갖는 ROM 하나로 구현할 수 있다.

$$J = Q_s(m_x)$$

- (3) 다음식에 의하여 크기가  $M$  인 partial codebook의 최대 탐색 영역  $J_L$  과  $J_U$  를 결정한다.

$$J_L = J - \frac{M}{2}$$

$$J_U = J + \frac{M}{2} - 1$$

If  $J_L < 1$ , then  $J_L = 1$  and  $J_U = M$

If  $J_U > N$ , then  $J_L = N - M - 1$  and  $J_U = N$

이 과정 역시 위의 (2)번 과정을 처리하는 과정에서 동시에 얻을 수 있다.

$$J_L = Q_{s1}(m_x), \quad J_U = Q_{s2}(m_x)$$

- (4) 분리 선택된 partial codebook을 완전 탐색 (full search) 하여 준최적의 index  $I$  를 구한다.

$$I = Q_v(X)$$

III. 컴퓨터 실험 및 고찰

이상의 탐색 과정을 그림 2와 같이 나타낼 수 있다. VQ decoding 과정은 보통의 VQ 에서와 마찬가지로, 전송된 index  $I$ 로부터  $I$  번째 vector를 재생시킨다.

실험 영상으로써 4장의 256x256 크기의 8 bits/pel 영상을 이용하였다. 표 1과 같은 통계 특성을 갖는 GIRL 과 COMP 영상을 실제 실험 영상으로 사용하였다. COMP

영상은 표1의 나머지 4 영상을 합하여 하나로 만든 512x512 영상이다.

본 논문에서 제안한 SCVQ 알고리즘이 보다 큰 training set에 대해서도 유사한 성능을 갖는다는 것을 보이기 위해 COMP 영상을 사용하였다. 일반화된 Lloyd algorithm [5]을 이용하여 각각 GIRL 과 COMP 를 training set로 하여 super codebook을 설계하였다. coding은 GIRL/GIRL, COMP/COMP, GIRL/COMP, COMP/GIRL 의 4가지 조합으로 하였다. 여기서 사진의 앞은 설계에 사용한 training set, 뒤는 coding에 사용한 training set 를 나타내며 후자의 두 경우를 각각 under training, over training 이라 부른다. 벡터의 크기는 4x4 영상 block을 벡터화하여 16 으로 하였고 super codebook의 크기는 256 으로 고정하였다. 결국 bit rate는  $R = (\log_2 256) / 16 = 0.5$  bits/pel 이다.

한편 본 논문과 같은 의도로 제안된 TSVQ (Tree Search VQ)의 성능을 함께 비교하였다. 각 node 당 각각 2개, 16개의 branch를 갖는 binary TSVQ 와 hexa TSVQ를 적용하여 SCVQ 와 비교하였다.

표 2에서 우선 SCVQ의 성능을 보면 4가지의 설계 및 coding 조합의 경우 모두 같은 경향을 갖는다는 것을 알 수 있다. 특히 256 벡터중에서 64개의 탐색만으로 완전 탐색 (full search)의 성능을 갖는다는 것을 알 수 있다. 16 개 및 32 개 벡터 탐색의 경우 약 0.1-0.9 dB의 성능 감소를 가져오지만 탐색의 복잡도를 1/8 - 1/16 로 줄일 수 있다는 것을 알 수 있다. 이와 같은 성능은 같은 탐색 복잡도를 갖는 binary TSVQ와 hexa TSVQ 보다 항상 0.1-0.4 우수한 성능을 갖는다는 것을 표2로부터 알 수 있다. codebook을 저장하기 위한 memory도 SCVQ가 더 작은 크기의 memory로 해결될 수 있다.

현재 partial codebook의 분리 선택에 이용하는 방법으로 벡터의 분산을 이용하는 방법과 평균 및 분산을 동시에 이용하는 방법을 연구하고 있다. 또한 벡터 예측에 의한 차 벡터에 적용하는 방법도 시도되고 있다.

REFERENCES

- [1] Buzo,A., Gray,A.H., Gray,R.M., and Markel,J.D. :  
'Speech coding based upon vector quantization', IEEE  
Trans., ASSP-28, pp.562-574, October 1980.
- [2] Wong,D., Juang,B.-H., and Gray,A.H., Jr. : 'An 800  
bit/s vector quantization LPC vocoder', IEEE Trans.,  
ASSP-30, pp.770-779, October 1982.
- [3] Juang, B.-H. and Gray,A.H.,Jr : 'Multistage vector  
quantization for speech coding', IEEE Int'l Conference  
Proc. on ASSP, pp.597-600, April 1982.
- [4] Makhoul,J., Roucos,S., and Gish,H. : 'Vector  
quantization in speech coding', Proc. of IEEE, Vol.73,  
pp.1551-1588, November 1985.
- [5] Linde,Y., Buzo,A., and Gray,R.M. : 'An algorithm for  
vector quantizer design', IEEE Trans., COM-28,  
pp.84-95, January 1980.
- [6] Abut,H., Gray,R.M., and Rebolledo,G. : 'Vector  
quantization of speech and speech-like waveforms',  
IEEE Trans., ASSP-30, pp.423-435, June 1982.

Table 1. Statistics of test images

	GIRL	GIRL2	HOUSE	CRONKITE	COMP
Image size	256x256	256x256	256x256	256x256	512x512
Mean	73.6	99.3	139.6	88.6	100.3
Standard Deviation	42.6	53.6	61.2	54.5	58.7
Horizontal Correlation	0.973	0.982	0.984	0.987	0.980
Vertical Correlation	0.962	0.982	0.973	0.991	0.977

Table 2. Performance comparison of SCVQ and TSVQ

Design image/ Coding image	Partial codebook size of SCVQ						TSVQ	
	8	16	32	64	128	256	binary	hexa
GIRL/GIRL	29.3	30.7	31.4	31.5	31.5	31.5	30.4	31.2
	75.5	94.2	99.5	100.0	100.0	100.0	-	-
COMP/COMP	28.9	30.3	31.1	31.2	31.2	31.2	30.0	31.2
	80.1	95.7	99.6	100.0	100.0	100.0	-	-
GIRL/COMP	27.3	28.0	28.4	28.5	28.6	28.6	27.6	28.2
	73.0	92.4	98.8	99.9	100.0	100.0	-	-
COMP/GIRL	28.3	29.3	29.9	30.1	30.1	30.1	29.0	29.5
	76.2	94.6	99.5	100.0	100.0	100.00	-	-
Search Complexity	1/32	1/16	1/8	1/4	1/2	1	1/16	1/8
Memory Requirement	1	1	1	1	1	1	2	16/15

(The upper value of each table entries represents SNR in dB,  
and the lower represents the search accuracy in percentage.)

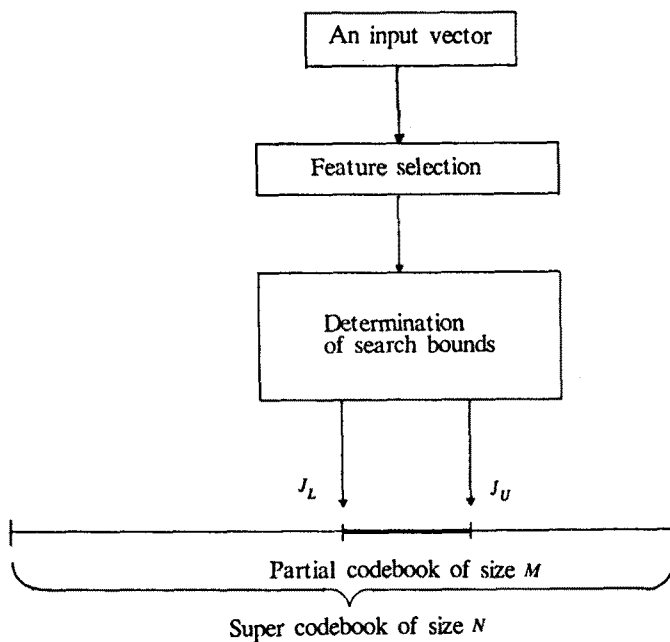


Fig. 1. Codebook split / selection method

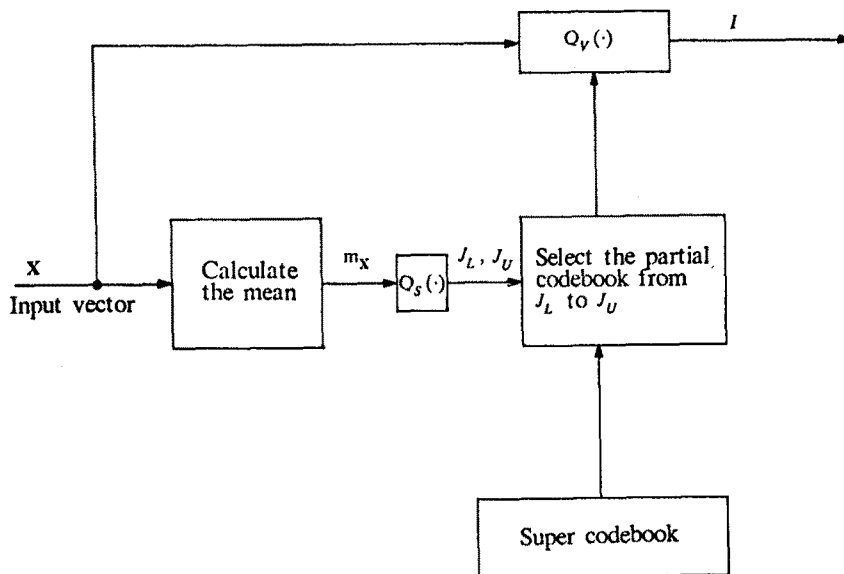


Fig. 2. Block diagram of SCVQ (Split Codebook VQ)