

## 신경망을 이용한 벡터 양자화의 코드북 설계

正會員 朱 相 顯\* 正會員 元 致 善\*\* 正會員 申 宰 浩\*\*

### A Codebook Design for Vector Quantization Using a Neural Network

Sang-Hyun Joo\*, Chee-Sun Won\*\*, Jaeho Shin\*\* *Regular Members*

#### 要 約

벡터양자화를 위한 신경망의 사용은 그것의 적응적 설계 특성으로 더 좋은 코드북을 설계할 수 있을 것으로 기대되며, 또한 설계된 코드북의 코드워드는 자동정렬되어 실시간 탐색을 가능케 한다.

신경망의 이러한 장점을 살리기 위하여 본 논문에서는 KSFM(Kohonen's Self-organizing Feature Map)을 수정하고, K-means 알고리즘을 결합한 새로운 코드북 설계 알고리즘을 제안한다. 실험결과로 부터 제안된 알고리즘의 성능향상과 실시간 처리를 위한 코드북의 부분탐색 가능성을 확인하였다.

#### ABSTRACT

Using a neural network for vector quantization, we can expect to have better codebook design algorithms for its adaptive process. Also, the designed codebook puts the codewords in order by its self-organizing characteristics, which makes it possible to partially search the codebook for real time process.

To exploit these features of the neural network, in this paper, we propose a new codebook design algorithm that modifies the KSFM(Kohonen's Self-organizing Feature Map) and then combines the K-means algorithm. Experimental results show the performance improvement and the ability of the partial search of the codebook for the real time process.

#### I. 서 론

영상을 디지털화하여 효율적으로 전송 또는 저장하는 문제는 도래할 정보통신망을 갖춘 멀티미디어

시대의 선결 과제다. 이러한 문제의 해결방안으로서 는 전송로의 전송능력을 향상시킬 수도 있지만, 그것 보다는 신호의 압축을 통해 전송율을 향상시키는 것이 더욱 경제적이다. 신호의 압축 방법중에서 벡터 양자화 기법은 영상의 화질을 적절히 유지하면서도 압축율이 좋은 특성 때문에 전송 및 저장을 목적으로 하는 영상 데이터의 압축에 널리 이용되고 있다. 그

\*生産技術研究院

\*\*東國大學校 電子工學科

Dept. of Electronics Engineering Dongguk University

論文番號: 94-27

러나 기존의 알고리즘을 사용하여 코드북을 설계할 때 다음 두가지의 제한점을 따르게 된다. 첫째, 설계시에 학습데이터로 사용되는 영상들은 각각의 특징을 갖고 있으며 그들에 의해 만들어진 코드북은 그들의 특성을 민감하게 반영할 수 밖에 없다. 그러나 설계된 코드북에 의해 실제 사용되는 영상은 설계에 참여하지 않은 영상이 대부분이며 학습영상의 수를 많이 취할수록 영상이 갖는 특성을 보다 더 반영할 수 있다. 게다가 학습영상 전체를 일괄(batch) 처리하는 기존의 알고리즘을 이용할 경우 막대한 **실시시간**이 소요되며 설계시간과 학습영상의 수 간에는 적당한 절충을 해야한다. 둘째, 의용(medical) 화상과 같이 고화질을 요구하는 경우에는 무손실(lossless) 압축이 행하여지지만, 압축율은 약 1:2 정도에서 그치고 만다. 무손실은 아니더라도 약간의 손실만을 허용하는 경우에는 의당 코드북의 크기를 크게해야 하지만 그 크기에 따른 계산량의 증가때문에 그 크기에도 제한이 따르는 단점을 갖고 있다.

따라서 최근에는 신경망 구조를 이용한 컴퓨터 개념이 도입되면서 신경 회로망의 병렬처리 특성과 환경 변화에 적응적인 학습특성을 이용해 기존의 벡터 양자화를 더욱 빠르고 유연성 있게 처리할 수 있는 방법이 제안되고 있다<sup>(1)(2)(3)</sup>. 병렬처리 구조의 신경망을 이용하면 각 학습 데이터를 적응적으로 학습시켜 코드북을 설계하므로 보다 빠르고 많은 학습 영상을 사용할 수가 있으며, 설계된 코드북은 공간적으로 유사한 코드워드가 서로 인접하는 구조를 갖기 때문에 탐색시간이 크게 줄어들 것으로 기대된다. 좀 더 구체적으로 말하면, KSFМ(Kohonen's Self-organizing Feature Map) 알고리즘으로 영상의 벡터양자화를 위한 코드북을 설계하면 결정된 대표벡터들 중에 유사한 벡터들끼리 서로 모여 자동정렬되므로, 코드북 탐색에 시간을 크게 절약할 수 있다<sup>(4)</sup>. 이와 같은 장점은 많은 코드워드를 갖는 코드북의 경우 더욱 뚜렷이 나타난다. 그러나 크기가 큰 코드북 설계에 KSFМ을 적용하면 모든 노드가 골고루 학습되지 못하고 특정 노드에 집중되는 현상이 발생한다. KSFМ은 일반적인 경쟁학습(competitive learning) 방법에서 발생하는 "특정노드 집중학습" 문제를 선택된 노드(winner) 뿐만이 아니라 이웃(neighbor) 노드들도 동시에 학습시킴으로써 해결하고 있으나, 코드워드의 수(즉, 노드의 수)가 많아지면 더욱 강렬한 해결책이 요구된다. Krishnamurthy 등은 FSCL(Frequency Sensitive Competitive Learning) 방법

을 통하여 집중 학습문제를 해결하였다<sup>(2)</sup>. 즉, 각 노드마다 계수기(counter)를 두어 각 노드들의 선택된(winner) 횟수를 기억하고, 그것의 합수를 인자의 선택시 반영하여 자주 선택된 노드의 선택기회를 인위적으로 낮추는 신경망 구조이다. 그러나 FSCL은 KSFМ과는 달리 이웃노드들을 선택된 노드와 동시에 학습시키지 않으므로 설계된 벡터 코드워드간에 유사성을 갖는 공간적 배열을 기대할 수 없다. 본 논문에서는 FSCL에서의 이웃(neighborhood) 개념을 가미하여 KSFМ의 "특정노드 집중학습" 문제를 해결하고, 동시에 결정된 코드워드 간에 유사성을 바탕으로 한 공간배열을 이룰 수 있는 새로운 방법을 제안하고, 이 알고리즘을 MKSFМ(Modified KSFМ)이라 부르기로 한다. 여기서 계수기의 역할은 코드워드를 인위적으로 분산시키는 역할을 하게 되며, 학습에 참여한 영상의 특징을 다소 완화시켜 보다 포괄적인 코드북을 설계하는데 있다. 따라서 학습에 참여하지 않은 영상의 압축에 보다 유리할 것으로 기대되며, MKSFМ에 의해 설계된 코드북은 성능향상을 위해 K-means 알고리즘에 의해 마무리처리(fine tuning) 과정을 거침으로써, MKSFМ의 공간정렬특성을 그대로 유지하면서도 성능의 향상을 꾀할 수 있어 보다 우수한 코드북의 설계가 가능하다고 본다. 실제로 KSFМ과 K-means 알고리즘의 결합은 단어인식(word recognition) 분야에 적용되어 성능향상을 이룬 바 있다<sup>(6)</sup>.

## II. 코드북 설계 알고리즘

코드북 설계과정은 우선 학습영상을 선택한 후 블럭내 데이터가 L개가 되도록 적당한 크기로 블럭을 나누어 학습데이터 집합을 구성한다. 이렇게 구성된 학습벡터들로부터 L 차원의 대표벡터(코드워드)를 구하는 것이 코드북의 설계과정이다. 본절에서는 KSFМ<sup>(7)</sup>과 FSCL<sup>(2)</sup>의 특징을 결합한 MKSFМ 알고리즘을 제안한다. MKSFМ으로 학습된 신경망은 다시 K-means 알고리즘에 의해 마무리 처리과정을 거침으로써 MKSFМ의 공간정렬특성을 그대로 유지하면서 성능향상을 이룰수 있는 장점을 갖고 있다.

### 1. K-means 알고리즘

K-means 알고리즘은 주어진 학습데이터를 분할하여 평균 사승 오차(MSE)가 최소가 되도록 정해진 수(K 개)의 대표벡터를 찾는 일련의 반복과정이다.

K 개의 코드워드를 갖도록 할 때, 임의로 K 개의 코드워드를 초기화한 후 입력벡터와 거리가 최소가 되는 노드를 찾아 각 노드에 할당되는 입력벡터들의 평균으로 새로운 코드워드의 값을 결정하며, 코드워드의 전체 변화를 추적하여 임계값 이하일 때까지 반복적으로 수정을 가하는 알고리즘이다. 본 논문에서는 성능비교를 위해 기존(신경망이 아닌) 컴퓨터 구조에서 K-means 알고리즘을 사용하여 코드북을 설계하며, MKSFM의 신경망과 결합하여 최종 코드워드에 학습 데이터의 자연적 집중구조를 반영할 수 있도록 마무리처리에도 사용한다. K-means 알고리즘의 코드북 설계과정은 아래와 같다.

- 1단계 : 모든 가중벡터(codeword)의 초기화.
- 2단계 : 코드워드  $w_{i,j}$ 를 기준으로 nearest neighbor rule에 의해 K 개의 partition  $\{P(w_{i,j}) : 1 \leq i \leq M, 1 \leq j \leq N; K = M \times N\}$ 를 구함.  
 즉, n번째 주어진 입력벡터  $x_n$ 에 대해  $x_n \in P(w_{i,j})$  if  $d(x_n, w_{i,j}) \leq d(x_n, w_{i,j+1})$ .
- 3단계 : 각 i, j 노드에 할당된 입력벡터들을 평균하여 이 평균값들로 가중벡터를 다시 지정.  
 $w_{i,j} \leftarrow \text{Cent}(P(w_{i,j}))$
- 4단계 : 전체왜곡의 변화가 임계값 이하일 때까지 2~3단계 반복.

**2. KSFM 알고리즘**

Kohonen은 스스로 학습할 수 있는 신경망 알고리즘을 발표하였으며,<sup>[7]</sup> KSFM은 벡터양자화에 아주 적합하게 적용될 수 있다. KSFM에서는 입력벡터와 초기화된 코드워드간의 Euclidean 거리를 계산하여 출력 노드를 결정하며 처음에는 출력 노드를 중심으로 전체 노드에 걸쳐 학습이 되도록 하지만, 시간이 지남에 따라 이웃반경을 줄여 가면서 서로 비슷한 벡터가 공간적으로 가까운 노드에 정렬되도록 하는 중요한 특징을 갖고 있어 수신단에서 복호화할 때 이들 특징을 이용해 전탐색(full-search)이 아닌 부분탐색(partial-search) 방법이 사용될 수 있어 코드워드 탐색시간을 크게 절약할 수 있다<sup>[8]</sup>. KSFM은 아래의 단계를 통해 코드북을 형성한다.

- 1단계 : 가중벡터의 임의 초기화.
- 2단계 : 학습 데이터에 대하여 가중벡터와의 거리계산.
- 3단계 : 입력벡터  $x(n)$ 과 가중벡터  $w_{i,j}(n)$  간의 최소 Euclidean 거리를 갖는 노드를 선택하여

winner라 정하고  $w^*_{i,j}$ 로 표시한다.

4단계 : 선택된 노드의 weight 조정과

$$w^*_{i,j}(n+1) = w^*_{i,j}(n) + \alpha[x(n) - w^*_{i,j}(n)],$$

$$0 < \alpha < 1$$

이웃반경내에 있는 노드의 weight 조정.

$$w_{i,j}(n+1) = w_{i,j}(n) + \alpha[x(n) - w_{i,j}(n)],$$

$$0 < \alpha < 1$$

5단계 : 전체왜곡의 변화가 임계값 이하일 때까지 2~4 단계 반복.

**3. MKSFM 알고리즘**

FSCL은 선택되는 노드에 계수기의 함수를 거리 측정시 반영하여 특정 노드의 집중학습 문제를 완화하였다. 본 논문에서 제안하는 MKSFM 알고리즘은 FSCL에서의 계수기를 KSFM에 부가한다. 즉, MKSFM에서는 KSFM 알고리즘은 제3단계에서 실행한 입력벡터  $x$ 와 각 노드간의 거리를  $D_{i,j} = U(i,j) \times d(x, w_{i,j})$ 로 계산한다. 여기서  $U(i,j)$ 는 i, j번째 노드의 누적된 계수값이고,  $d(x, w_{i,j})$ 는 입력벡터  $x$ 와 i, j번째 노드의 가중치  $w_{i,j}$ 의 Euclidean 거리로써 KSFM의 제3단계의 거리측정 함수와 같다.  $U(i,j)$ 의 역할은 자주 선택된 노드의 선택기회를 줄이는 것으로 다음장에서 설명하는 i, j번째 노드의 선택횟수의 변수로 함수화 된다.

이외같이 누적계수의 가중치로 수정된 Euclidean 거리를 KSFM의 거리 대신 사용한다. 또한 MKSFM을 적용한 후 학습이 끝나면 이들을 초기 코드워드로 간주하여 K-means 알고리즘을 적용한다. 즉 K-means의 초기 코드워드가 공간적으로 정렬된 상태이기 때문에 K-means 적용 후, 성능향상을 이룬 후에도 공간정렬 특성을 그대로 유지하면서 학습데이터의 자연적 군집을 반영할 수 있도록 마무리 처리하였다. 다만 K-means 알고리즘의 추가적용은 코드북 설계시간의 지연을 가져올 수 있지만 신경망의 병렬성과 K-means 알고리즘의 빠른 수렴특성, 그리고 코드북 설계가 off-line에서 이루어진다는 점에서 큰 문제가 되지는 않는다.

**III. 시뮬레이션 및 고찰**

**1. 코드북 설계**

본 시뮬레이션에서는,  $512 \times 512 (M = N = 512)$  크기의 lena 영상으로부터  $4 \times 4$ 의 학습 벡터를 취하여 코드북을 설계하였으며, 코드북의 코드워드를 16

부터 256 까지 가변시켜 가면서 PSNR(Peak Signal to Noise Ratio)과 RMSE(Root Mean Square Error)를 비교하여 본 것이다. 여기서 사용된 임계치는 100이며, 성능의 지표가 되는 PSNR과 RMSE는 식 (1), (2)와 같이 정의된 것이다.

$$PSNR = 10 \log_{10} \frac{255^2}{\frac{1}{M \times N} \times \sum_{i=1}^M \sum_{j=1}^N (x_{ij} - y_{ij})^2} \quad (1)$$

$$RMSE = \sqrt{\frac{1}{M \times N} \times \sum_{i=1}^M \sum_{j=1}^N (x_{ij} - y_{ij})^2} \quad (2)$$

KSFM의 제4단계에서 사용한 학습계수  $\alpha$ 를 0.7로 하고, 계수기의 함수로는  $U(i, j) = n^{e-1}$ 을 사용하였을 때 각 비트율에 따른 PSNR 및 RMSE를 표1에 제시하였다. 한편 MKSFM 알고리즘에서는 사용되는  $U(i, j)$ 와 학습계수의 선택에 따라 성능이 크게 좌우되며 학습계수와 이웃의 범위는 적절한 방법으로 형성될 수 있으며, 본 시뮬레이션에서는 초기 이웃반경이 2 차원 코드북의 크기가  $M \times M$  일때 그림 1과 같이 반복횟수에 따라 줄어들도록 하였다. 또한 계수기의 함수  $U(i, j)$ 와 학습계수  $\alpha$  값의 선택도 성능에 영향을 미

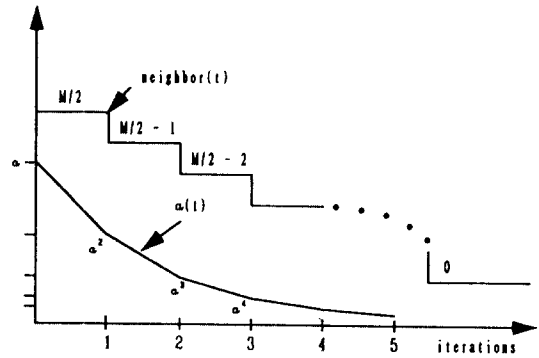


그림 1. 사용된 학습계수  $\alpha$ 와 이웃반경. ( $M \times M$ )  
 Fig. 1. Used  $\alpha$  and radius of neighborhood in the caes of  $M$  by  $M$  grid line.

치며 그들의 변화에 따른 성능의 변화를 표2에 제시하였다.

표1에서 알 수 있듯이 코드북의 크기가 클수록 마무리 처리에 의한 MKSFM 알고리즘의 성능향상이 두드러짐을 알 수 있다. 예를 들어, 코드북의 크기가 16(0.25 b/p)인 경우 거의 성능향상이 없지만 256(0.5 b/p)인 경우 약 0.6 dB의 마무리처리가 이루어짐을

표 1. 코드북 크기에 따른 K-means, KSFM, MKSFM 및 MKSFM + K-means의 성능 비교.

Table 1. The performance comparison with increasing codebook size of the K-means, KSFM, MKSFM and MKSFM + K-means.

code book size	bit Rate (b/p)	K-means		KSFM		MKSFM		MKSFM + K-means	
		PSNR	RMSE	PSNR	RMSE	PSNR	RMSE	PSNR	RMSE
16	0.250	27.28	11.03	26.84	11.61	27.14	11.21	27.27	11.04
32	0.313	28.43	9.66	27.78	10.41	28.21	9.91	28.51	9.57
64	0.375	29.44	8.56	29.20	8.84	29.15	8.90	29.61	8.43
128	0.438	30.51	7.61	30.41	7.69	30.00	8.06	30.62	7.51
256	0.500	31.02	7.17	30.86	7.31	30.92	7.26	31.51	6.78

표 2.  $\alpha$ 와  $U(i, j)$ 의 변화에 따른 성능비교.

Table 2. The performance comparison with various  $\alpha$ 's and  $U(i, j)$ 's.

initial alpha	KSFM		MKSFM		MKSFM		MKSFM + K-means	
			$U(i, j) = n$		$U(i, j) = n^{e-1}$		$U(i, j) = n^{e-1}$	
	PSNR	RMSE	PSNR	RMSE	PSNR	RMSE	PSNR	RMSE
0.1	23.34	17.37	23.75	16.57	24.26	15.62	31.62	6.69
0.3	27.63	10.60	27.50	10.76	28.17	9.95	31.72	6.61
0.5	29.34	8.70	29.16	8.88	29.28	8.76	31.72	6.62
0.7	30.86	7.31	30.62	7.51	30.92	7.26	31.51	6.78
0.9	31.51	6.78	30.86	7.31	31.03	7.16	31.43	6.84

알 수 있다. 표1에서 K-means 알고리즘에 의한 결과는 전통적인 컴퓨터 구조를 바탕으로 얻은 결과이며, MKSFM + K-means 알고리즘은 신경망 구조에 적용된 것으로 코드워드가 공간배열되는 특성을 갖지만 기존 컴퓨터 구조에 의한 K-means 알고리즘에서는 그런 특성이 없다. 표2에서는 코드북의 크기를 256으로 하고 전송율을 0.5 bpp으로 고정시켰으며 학습계수  $\alpha$ 는 0과 1사이에서 변화시켰다. 노드  $i, j$ 가 선택된 횟수  $n$ 에 대한 함수는  $U(i, j) = n$ 와  $U(i, j) = n^e$ 를 사용하여 비교하였다. 또한 MKSFM과 마부리 처리를 한 경우에 대해서도 비교하였다. 본 실험의 결과에 따르면  $\alpha$ 의 값이 작을 수록 성능이 떨어지지만 결국 K-means 알고리즘에 의한 마부리처리 과정을 거치게 되면 비슷한 성능을 갖게 됨을 알 수 있어  $\alpha$  값의 선택은 성능변화에 그리 민감하지 않음을 보여 주고 있다.

그림 2는  $\alpha = 0.1$  부터  $\alpha = 0.9$  까지의 MKSFM 후 K-means에 의한 마부리 처리되는 과정을 추적한 것인데,  $\alpha$ 의 값이 작을수록 빨리 수렴하지만 결국 마부리 처리 이후 비교적 근사한 값으로 수렴함을 알 수 있다. 따라서 본 알고리즘을 사용하여 코드북을 설계할 때  $\alpha$ 의 값을 작게 취하는 것이 설계 시간을 단축시킬 수 있을 것이다.

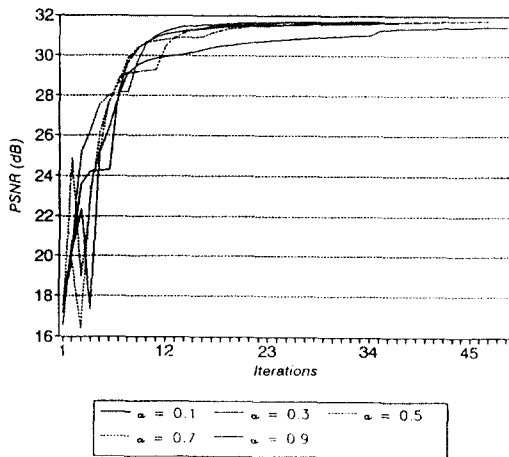


그림 2.  $\alpha$ 의 변화(0.1~0.9)에 따른 MKSFM + K means의 성능 비교.  
 Fig. 2. The performance comparison for various  $\alpha$ 's of the MKSFM + K-means. ( $\alpha = 0.1 \sim 0.9$ )

## 2. 코드북 탐색

제안된 알고리즘에 의해 설계된 코드북은 2차원으로 구성되었으며 3차원으로도 확장이 가능하다. 차원이 높아질수록 보다 빠른 탐색이 가능하며, 신경망을 이용한 코드북이 어떻게 공간적으로 배치되는가를 가시화하기 위해 1차원 코드워드를 갖는 2차원 코드북을 설계하였다.(표3 참조) 사용된 영상은 lenna 영상(512×512)이며, 코드북의 크기는 8×8로 하였다. 그림에서 볼 수 있듯이 신경망에 의해 설계된 코드북은 2차원 공간내에 비슷한 값들끼리 몰려 공간정렬을 이루는 것을 알 수 있으며 KFSM, MKSFM, MKSFM + K-means에 의해 각각 구성된 코드북을 표4에 나타내었는데 MKSFM 이후에 K-means에 의한 마부리처리를 한 후에도 공간정렬 특성을 그대로 간직함을 알 수 있다. 또한 KFSM 코드북에서는 39에서 210까지의 값을 갖는 반면 MKSFM에서는 38에서 212까지의 값을 가지므로 더욱 폭이 넓은 코드북이 설계됨을 의미한다. 즉 KFSM 코드북은 학습된 영상의 복원시 유리하지만 학습에 사용되지 않은 영상의 복원시에는 MKSFM 보다 불리하다는 것을 말해 주며, 이러한 현상을 뒷받침하기 위해 실제 512×512 크기의 lenna 영상을 학습영상으로 취하고 부분탐색을 위해 코드북의 크기를 15×15 크기(0.44 bpp, 대칭성을 고려한 크기임, 코드북의 크기가 15×15가 아닌 경우에는 중복탐색 또는 탐색영역의 제한등의 기법이 사용될 수 있음)로 설계한 후 각 알고리즘에 대한 전탐색 및 부분탐색을 1개의 학습영상과 4개의 비학습 영상에 대하여 실험하여 보았다. 그 결과 학습에 참여한 영상(lenna)의 경우 KFSM 코드북의 성능이 우수하지만 학습에 참여하지 않은 비학습영상(bank, girl, jaguar, cheer)에서는 MKSFM 코드북이 더 우수한 성능을 보였다. 부분탐색은 그림3과 같이 먼저 9개의 위치에서 최초의 에러를 갖는 노드를 결정된 후 그 노드를 중심으로 16개의 위치를 다시 비교하고, 여기서 선택된 노드를 중심으로 5×5의 전탐색을 하는 방법을 취하였다. 결국 225개의 노드를 전부 계산해야 하는 전탐색에 비해 9 + 16 + 25 = 50개의 노드만을 비교함으로써 약 78%의 계산량을 감소시킬 수 있다. 본 알고리즘의 특징대로라면 50개의 노드를 비교하는 것도 많은 것처럼 보이지만 실제로 코드북을 설계하면 각 노드간의 오차가 전구간에 걸쳐 완전히 균일하지 못하기 때문에, 선택된 특정노드를 중심으로 일정반경만을 탐색하는 경우 오차가 더 작으면서도 반경내에 들지 못하여 계산에서 제외되

표 3. 1차원 코드워드의 공간정렬.

Table 3. Spatial organizations of one dimensional codewords.

KSFM		MKSFM		MKSFM + K-means	
	1 2 3 4 5 6 7 8		1 2 3 4 5 6 7 8		1 2 3 4 5 6 7 8
1	39 47 60 71 81 83 87 89	1	38 49 63 76 92 105 110 114	1	36 48 62 76 92 105 110 114
2	52 56 67 75 85 86 89 91	2	44 53 64 78 97 113 115 118	2	43 53 64 78 97 113 115 119
3	78 77 84 87 97 99 95 93	3	58 65 74 88 112 124 126 122	3	58 66 74 88 112 124 126 122
4	119 119 119 117 109 108 100 99	4	68 72 86 102 123 133 136 138	4	68 72 86 102 123 133 136 138
5	143 141 137 133 121 111 104 102	5	81 84 95 117 131 140 142 144	5	81 84 95 117 131 140 143 144
6	162 155 150 139 130 114 107 105	6	90 94 108 129 150 156 158 155	6	90 94 108 129 150 157 158 155
7	196 175 157 146 135 119 112 108	7	92 95 115 141 153 167 184 201	7	92 96 116 142 153 167 184 202
8	210 185 169 152 140 125 115 110	8	93 99 120 147 161 174 193 212	8	93 99 120 147 162 174 193 213

(a)KSFM 코드북

(b)MKSFM 코드북

(c)MKSFM + K-means 코드북

표 4. 학습영상과 비학습영상에 대한 전탐색 및 부분탐색.

Table 4. Full search and Partial search for trained and untrained images.

	KSFM		MKSFM		MKSFM + K-means	
	FS	PS	FS	PS	FS	PS
lenna,512	31.19	29.72	30.90	29.83	31.43	30.27
bank,512	23.27	22.83	23.42	23.11	23.80	23.40
girl,512	23.65	23.52	24.38	24.25	24.71	24.50
jaguar,512	22.11	21.59	22.26	21.84	22.83	22.38
cheer,512	25.87	25.53	26.19	25.79	26.54	26.11

(at 0.44 bpp)

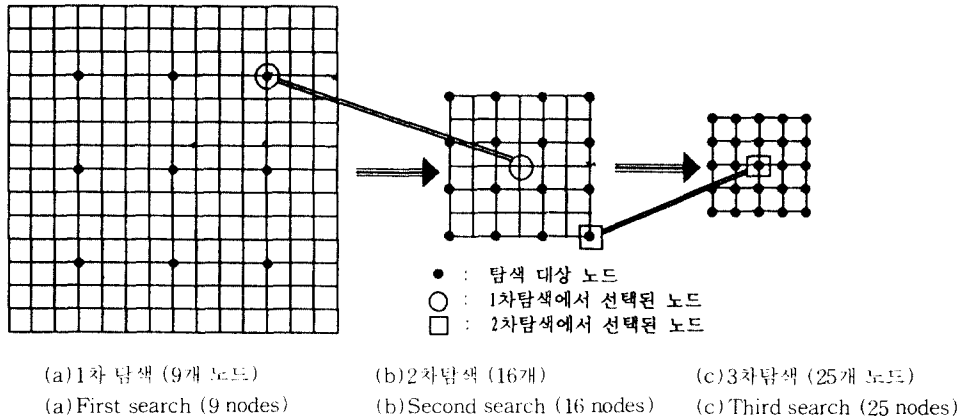


그림 3. 부분탐색에 사용된 방법.

Fig. 3. Steps for partial search.

는 경우도 발생하게 된다. 이러한 경우를 방지하기 위해 중복된 검색이 반드시 필요하며 경우에 따라서는 탐색알고리즘을 어떻게 하느냐에 따라 계산량을 더 줄일 수도 있을 것으로 기대된다.

그림4는 K-means, KSFM, MKSFM 및 MKSFM + K-means 알고리즘을 적용했을 때 0.44 bpp에서의 결과 영상이며 MKSFM 후 K-means 알고리즘이 기존의 방법 보다 더 나은 결과를 얻을 수 있었다.

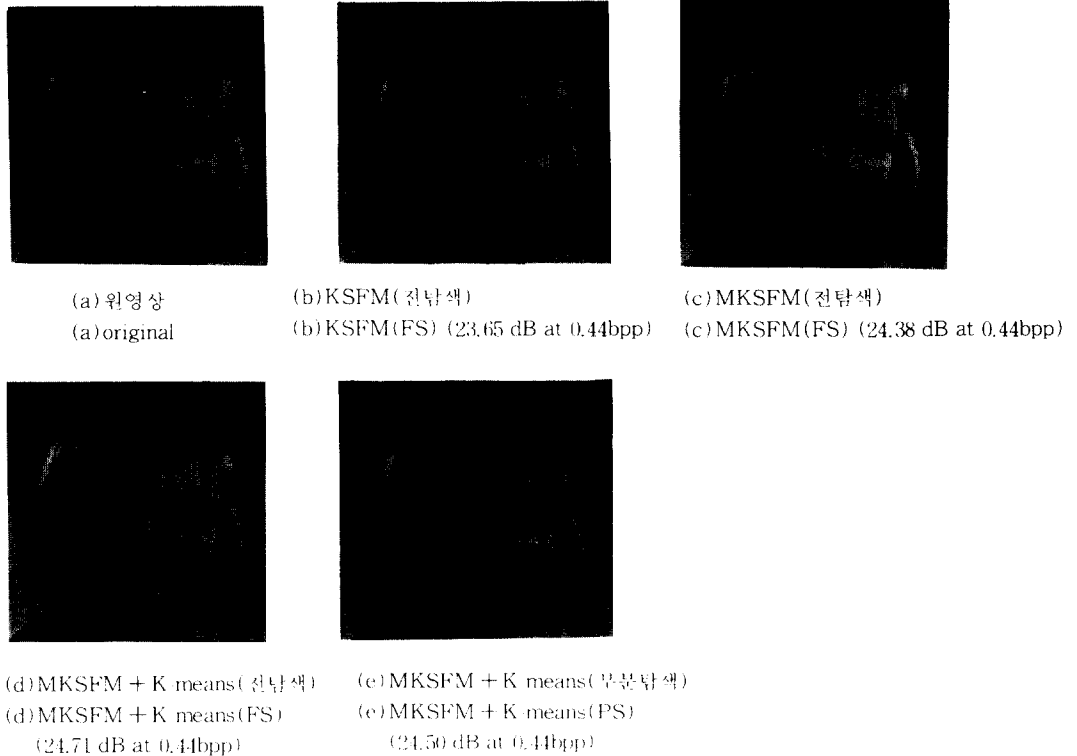


그림 4. K means, KFSM 및 MKFSM + K means 알고리즘에 대한 전탐색 및 부분탐색의 복원영상과 원영상. (512×512, 0.44 bpp)

Fig. 4. The original and the reconstructed images by FS and PS for each algorithm. (512×512, 0.44 bpp)

#### IV. 결 론

본 논문에서는 KFSM 구조에 FSCL의 계수기법을 도입한 수정된 KFSM(MKFSM)을 사용하여 많은 노드를 갖는 신경망 구조에서 가능한 모든 노드가 알고리즘 학습될 수 있도록 하는 동시에, 학습된 노드들 간에 공간적인 상관관계도 유지되도록 하였다. 또한 참고문헌 6에서는 KFSM과 K-means를 결합한 신경망 구조를 “단어인식”의 분야에 적용하여 성능향상을 이룬 것에 착안하여 제안된 MKFSM과 K-means의 Fine-tuning 과정을 결합하고 그것을 “벡터양자화의 코드북 설계”의 분야에 적용하였다.

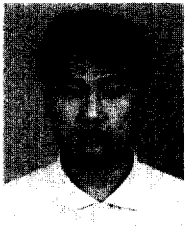
시뮬레이션 결과, 3E1에서와 같이 코드북 크기가 증가할수록 제안된 방법이 기존의 KFSM이나 전탐

색인 컴퓨터 구조에서의 K-means의 결과보다 성능 향상이 있음을 확인하였고, 표2에서 학습율  $\alpha$ 가 코드북의 성능에는 그다지 큰 영향을 미치지 못함을 확인하였으며, 표 3에서는 Fine tuning 후에도 공간정렬 특성을 그대로 간직하여 전탐색에 비해 약 78%의 탐색시간이 절약됨을 보였으며, 표4에서는 제안한 알고리즘에서 부분탐색에 의한 영상복원과 전탐색에 의한 영상복원이 그 성능면에서 유사함을 보여 주었다.

#### 참 고 문 헌

1. Y.Linde, A.Buzo and R.M.Gray, "An Algorithm for Vector Quantizer Design," *IEEE Trans. Communications*, vol.COM-28, pp.84-95, Jan.

- 1980.
2. C.C.Lu and Y.H.Shin, "A Neural Network Based Image Compression System," IEEE Trans. on Consumer Electronics, vol.38, no.1, pp.25-29, Feb. 1992.
  3. A.K.Krishnamurthy, S.C.Ahalt, D.E.Melton, and P.Chen, "Neural Network for Vector Quantization of Speech and Images," IEEE J. on Selected Areas in Communications, vol.8, no.8, pp.1449-1457, Oct. 1990.
  4. N.M.Nasrabadi and Y.Feng, "Vector Quantization of Images based upon the Kohonen Self-organizing Feature Maps," Proc. IEEE Int. Conference on Neural Networks, at San Diego, California, pp.1101-1108, 1988.
  5. K.K.Truong, "Multilayer Kohonen Image Codebooks with a Logarithmic Search Complexity," IEEE ICASSP, vol.4, pp.2789-2792, 1991.
  6. Z.Huang and A.Kug, "A Combined Self-organizing Feature Map and Multilayer Perception for Isolated Word Recognition," IEEE Trans. Signal Processing, vol.40, no.11, pp.2651-2657, Nov. 1992.
  7. T.Kohonen, "Self-organized Formation of Topologically Correct Feature Maps," Biological Cybernetics, vol.43, pp.59-69, 1982.



朱相顯(Sang-Hyun Joo) 學生會員  
1965년 2월 1일생  
1989년 2월 : 東國大學校 工科大學  
電子工學科(工學士)  
1994년 2월 : 東國大學校 大學院 電  
子工學科(工學碩士)  
現在 : 生産技術研究院 研究員



元致善(Chee-Sun Won) 正會員  
1959년 11월 24일생  
1982년 2월 : 高麗大學校 工科大學  
電子工學科(工學士)  
1986년 1월 : Univ. of Massachu-  
setts(MS)  
1990년 1월 : Univ. of Massachu-  
setts(Ph.D)

1989년 11월 ~ 1992년 8월 : 금성사 전임연구원  
現在 : 東國大學校 工科大學 電子工學科 助教授



申宰浩(Jaeho Shin) 正會員  
1952년 8월 29일생  
1979년 2월 : 서울大學校 工科大學  
電子工學科(工學士)  
1982년 2월 : 서울大學校 大學院 電  
子工學科(工學碩士)  
1987년 2월 : 서울大學校 大學院 電  
子工學科(工學博士)

現在 : 東國大學校 工科大學 電子工學科 副教授