論文 89－26－11－29

# 神經回路網을 이용한 코드북의 順次的 更新 알고리듬

# (An Algorithm to Update a Codebook Using a Neural Net)

丁 海 默,* 李 周 熙,* 李 忠 雄*

(Hae Mook Jung, Joo Hee Lee, and Choong Woong Lee)

**要 約**

本 論文에서는 神經回路網을 이용하여, 連續的인 프레임에 대해서 順次的으로 코드북을 更新시켜 나갈 수 있는 알고리듬을 제안하였다. 신경회로망은 Kohonen의 自律學習 신경회로망모델(self-organizing feature maps)을 근간으로 하여, 각 클러스터의 圓心을 順次的으로 갱신하되, Kohonen이 사용한 經驗的인 방법 대신 정확한 계산에 의해 圓心(centroid)을 更新시켜 나감으로써, 最適量子化를 이룰 수 있도록 하였다. 이 모델을 이용하면 한번 반복으로 LBG 알고리듬과 거의 같은 성능을 얻을 수 있기 때문에, 연속적인 화면에 대해서 코드북을 구성해 나갈 수 있으며, 하드웨어로 構成하여 實時間으로 동작시키기에 용이한 長點을 갖고 있다.

**Abstract**

In this paper, an algorithm to update a codebook using a neural network in consecutive images, is proposed. With the Kohonen's self-organizing feature map, we adopt the iterative technique to update a centroid of each cluster instead of the unsupervised learning technique. Because the performance of this neural model is comparable to that of the LBG algorithm, it is possible to update the codebooks of consecutive frames sequentially in TV and to realize the hardware on the real-time implementation basis.

## I. 序 論

벡터 量子化는 一種의 패턴 매칭(pattern matching) 방법으로써, 종래의 모든 量子化器가 하나의 샘플을 가지고 그것에 대해 양자화를 하는 것이었던 반면에, k개의 샘플 또는 k개의 매개변수(parameter)를 하나의 k차원 벡터로 취급하여, 이것을 양자화하는 것이다.[1] 따라서, 코드북은 하나의 畵像을 가장 잘 나타낼 수 있는 代表 벡터들로 이루어지는데, 이 코드북을 구성하는 가장 일반적인 방법으로는 Linde, Buzo, Gray에 의해 개발된 LBG알고리듬[2, 3]이 널리 알려져 있다. 이 LBG 알고리듬은 反復的인(iterative) 방법으로써, 코드북 구성에 많은 시간이 걸리기 때문에, Equitz는 所要時間을 약 24배 정도로 줄이는 高速 알고리듬[2]을 提案하였다.

그러나, 이러한 알고리듬들은 하드웨어로 구성하기에는 어려운 알고리듬들이다. 하드웨어로 구성하여 實時間 處理를 하기 위해서는, 트레이닝 벡터를 입력시키면서 코드북(codebook)을 更新해야 하는데, 이런 방법의 일종으로 Nasrabadi는 神經回路網을 이용하여 코드북을 구성하는 알고리듬[5]을 제안하였다.

이 알고리듬은 Kohonen의 自律學習 神經回路網 모델(self-organizing feature map)을 이용하여, 트레이

*正會員, 서울大學校 電子工學科
(Dept. of Electronics Eng., Seoul Nat'l Univ.)

接受日字 : 1989年 7月 24日

닝 벡터를 하나씩 입력하면서 코드벡터를 更新시켜 나가는 알고리듬이다. 이러한 順次的 更新方法을 이용하면, 連續的인 畵像(예를 들면, 텔레비젼의 프레임(frame))에 대해서 계속 코드북을 갱신시켜 나갈 수 있으며, 하드웨어로 구성하기에 용이한 長點을 갖고 있다. 그러나 Nasrabadi가 제안한 이 알고리듬은 經驗적인 학습모델인 Kohonen의 학습모델을 그대로 이용했기 때문에, 根本的으로 stationary하지 않은 畵像 新號에 대해서는 맞지 않는다. 따라서, Nasrabadi는 화상신호를 stationary한 成分과 stationary하지 않은 에지(edge)成分으로 나누어 신경 회로망에 적용하였으며, 화상 블록의 크기도 2×2로 하여 전송율을 2.15bit/pixel 以下로 낮출 수가 없었다. 이와같은 이유로 畵質이나 傳送率 面에서 Nasrabadi가 제안한 알고리듬이 LBG 알고리듬보다 性能이 훨씬 떨어진다.

본 논문에서는 기존의 여러 벡터 양자화 방법에서와 같은 傳送率과 畵質을 유지 하기 위해서 Kohonen의 학습모델을 이용하되 경험적인 학습보다는 각 코드벡터가 대표하는 클러스터의 圓心(centroid)을 順次的으로 更新시켜 나감으로써 最適量子化를 이룰 수 있는 방법을 提案한다. 이 모델을 이용하면 한번 반복으로 LBG 알고리듬과 거의 동일한 성능을 얻을 수 있으며, 또한 初期 코드북(intial codebook)을 임으로 잡아도 收斂되기 때문에, 畵面의 切替(scene change)가 일어나도 항상 最適化로 수렴되어 나가는 장점을 갖고 있다.

## Ⅱ. Kohonen의 自律學習 모델을 벡터 量子化에 利用時의 問題點

Kohonen은 인간의 頭腦와 비슷한 인공 신경회로망을 이용하여 그림 1과 같은 자율 학습 신경회로망 모델을 세웠다.[6]

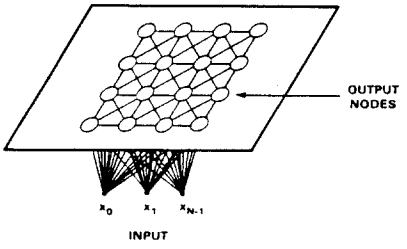



그림 1. Kohonen의 신경회로망 모델
(R. P. Lippmann [6]에서 인용)

**Fig. 1.** Kohonen's self-organizing feature map
(from R. P. Lippmann [6]).

이 자율학습 모델에서 각 뉴우런(neuron)은 웨이트벡터(weight vector) W(t)를 갖고 있으며, 이 웨이트벡터와 입력 벡터 X(t)를 비교하여, 입력 벡터와 가장 類似한 클러스터(cluster)로 입력 벡터 X(t)를 分類하는 작업을 한다. 학습과정을 수행하기 前에 모든 웨이트벡터들은 임의의 값으로 初期化(initialization)되고 입력 벡터들이 신경회로망에 하나씩 차례대로 입력되면서 웨이트벡터들이 다음과 같은 규칙에 따라 更新되어 나간다.

1) 入力 벡터 X(t)와 가장 가까운 웨이트 벡터 W(t)를 갖는 뉴우런 j*를 선택한다. 이때, 뉴우런 j*는 다른 뉴우런들과의 경쟁에서 이겼다고 말하며, 거리 $d_j$는

(1) 식과 같이 定義된다.

$$d_j = \sum_{i=0}^{N-1} (X_i(t) - W_{ij}(t))^2, \quad j = 1, 2, \cdots\cdots, M \qquad (1)$$

이 식에서 $X_i(t)$는 N차원 입력 벡터 X(t)의 i번째 성분이며, $W_{ij}(t)$는 $X_i(t)$로 부터 뉴우런 j까지의 웨이트이다. M은 뉴우런의 갯수이고, t는 시간을 표시한다.

2) 뉴우런 j*를 중심으로 일정한 距離 내에 있는 뉴우런들의 집합을 $NE_{j*}(t)$라 하고 $NE_{j*}(t)$에 속하는 뉴우런들의 웨이트를 (2)식과 같이 變更한다.

$$W_{ij}(t+1) = W_{ij}(t) + \eta(t)\,(X_i(t) - W_{ij}(t)), \quad j \in NE_{j*}(t) \qquad (2)$$

여기서, $\eta(t)$는 0에서 1까지의 값을 갖는 경험적인 이득계수로서, 시간이 경과함에 따라 값이 줄어들도록 한다. 또한, $NE_{j*}(t)$의 반지름도 시간에 따라서 서서히 줄어 들게하여 학습율이 어느정도 높아지게 되면 $NE_{j*}(t)$는 뉴우런 j*만을 포함하게 된다. 이는 학습을 많이 할수록 학습의 정도를 작게 하여 안정된 最適化를 이루게 하기 위함이다. 이러한 경험적인 계수는 입력 패턴들의 통계적 성질에 따라서 달라질 수 있으며, 또한 부적당한 값을 취하면 학습이 전혀 이루어지지 않을 수도 있게 된다.

이 모델은 입력 벡터에 대해서 각 뉴우런의 웨이트를 갱신시켜 줌으로써 학습을 시킬 수 있기 때문에 Nasrabadi는 이 모델을 이용하여 벡터 양자화에서의 코드북을 생성하는 알고리듬[4]을 제안하였다. 이때, 입력 벡터가 트레이닝 벡터가 되며, 각 뉴우런의 웨이트 벡터가 코드북의 코드벡터가 된다. Nasrabadi가 제안한 알고리듬에서는 초기 코드벡터를 웨이트벡터로 잡고 트레이닝 벡터를 입력하면서 웨이트벡터를 위와 같은 Kohonen의 법칙과 같이 갱신시켜

주면, 최종적으로 우리가 원하는 코드벡터를 웨이트 벡터로부터 얻을 수 있지만, Kohonen의 모델은 경험적인 학습 모델이기 때문에 이 코드벡터들이 반드시 최적화된다는 保障은 없다. 따라서, Nasrabadi의 방법에서 이 웨이트벡터들을 최적화로 이끌기 위해서는 다음과 같은 몇 가지 條件이 필요하게 된다.

첫째, 트레이닝 벡터들의 統計的 性質을 알고 經驗的인 係數를 그것에 맞추면서, 웨이터벡터가 최적화가 되도록 誘導해야 한다. 왜냐하면, (2)식의 $\eta(t)$와 $NE_j*(t)$의 반지름은 경험적인 것이기 때문에 트레이닝 벡터들의 성질에 크게 좌우되며 값에 따라 學習率은 달라지게 되기 때문이다.

둘째, 트레이닝 벡터의 성질을 알고 그것에 맞춰서 입력 순서를 잘 선택해야 한다. 그렇지 않으면 전혀 학습이 안될 수가 있는데, 이런 현상을 k차원의 超救體(hypersphere)上에서 살펴보면 그림 2와 같다. 여기서, 웨이트벡터의 위치를 o기호로 트레이닝 벡터의 위치를 x기호로 나타내었다. 예를 들어, 그림 2의 (a)와 같이, 트레이닝 벡터가 어느 정도 區分되어 몇 개의 클러스터(cluster)로 이루어져 있는 경우에는, 학습이 진행될수록 각 뉴우런의 웨이터벡터들이 그 클러스터의 원심쪽으로 이동되면서 최종적으로 그 뉴우런은 자신이 맡은 클러스터를 代表하게 되지만, 그림 2의 (b)와 같이 트레이닝 벡터들이 산만하게 분산되어 있는 경우는 학습을 시키기가 어렵다. 그림 2의 (b)에서 트레이닝 벡터 $X_1$이 입력되었다고 가정한다. 그때 가장 가까운 거리에 있는 웨이트벡터는 $W_1$이므로 $W_1$을 (2)식에 의하여 화살표 $P_1$만큼 $X_1$벡터 쪽으로 끌어 오게 된다. 이 화살표 $P_1$의 크기는 $\eta(t)$에 의해서 정해지며, 이때 웨이트벡터 $W_1$은 화살표 $P_1$의 크기만큼 학습되었다고 할 수 있다. 다음 트레이닝 벡터 $X_2$가 입력되었을 경우, $W_1$이 학습되기 전에, $X_2$와 가장 가까운 웨이트벡터는 $W_2$였지만 $W_1$이 $X_1$에 의해서 학습이 되었기 때문에 이제는 $X_2$에 가장 가까운 웨이트 벡터는 $W_2$가 아니고 $W_1$이 된다. 따라서, 이 $W_1$이 화살표 $P_2$만큼 다시 $X_2$쪽으로 끌려와 학습이 되고나면, $W_1$은 처음 $X_1$에 의하여 클러스터 A를 대표하도록 誘引되었지만, 다시 $X_2$에 의하여 클러스터 B쪽으로 誘引된다. 만약, 트레이닝 벡터의 입력 순서가 $X_1$, $X_2$, $X_3$ 순서가 아니고, $X_1$, $X_3$, $X_2$ 순서로 되었다면, $X_1$에 의해 $P_1$만큼 학습된 $W_1$ 벡터는 다시 $X_3$에 의하여, $P_3$만큼 학습되어 $X_2$와는 멀어지므로 다시 $X_2$에 의해서 끌리는 일 없이 안정적으로 클러스터 A를 대표할 수 있게 된다. 즉, 트레이닝 벡터의 입력순서가 학습 과정을 좌우하게 된다. 이런 식으로 모든 웨이트 벡터들이 방향성이 없이 트레이닝 벡터에 따라 이리저리 끌려 다니면, 전혀 학습이 이루어지지 않을 뿐더러 최적화는 바라볼 수도 없게 된다. 그런데, 화상의 벡터 양자화에서는 에지 패턴이 두드러지기 때문에, 트레이닝 벡터들이 stationary 하지 않아서 위와 같이 웨이트벡터들이 방향성 없이 끌려다니는 現象이 일어나게 된다. 이런 문제 때문에, Nasrabadi는 畵像을 stationary한 부분(낮은 주파수 성분)과 에지 성분(높은 주파수 성분)으로 나누어서 따로 학습시켰다. 또한, 트레이닝 벡터들의 위치가 산만하게 퍼지는 것을 방지하기 위하여, 화상블록의 크기를 기존의 4×4에서 2×2로 줄였기 때문에 코드북크기가 384개로 증가 되었으며, 그 결과 2.15bit/pixel까지 전송율이 높아졌다. 이는 기존의 벡터양자화와 비교해 볼 때 전혀 장점이 없으며 LBG알고리듬 보다 성능이 훨씬 떨어지게 된다.

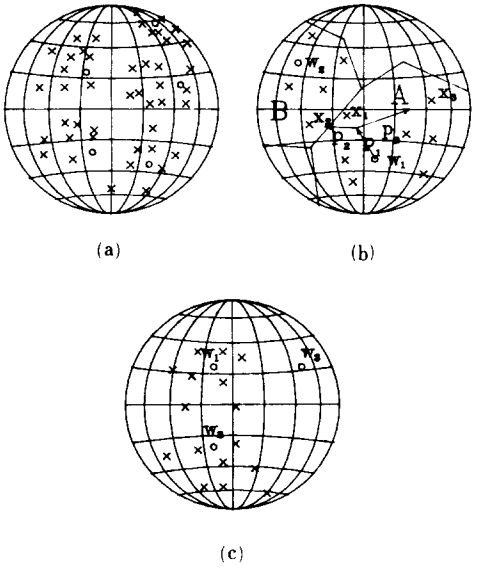



그림 2. 초구체상에서의 벡터들의 위치
(a) 트레이닝 벡터가 몇 개의 클러스터로 구분된 경우
(b) 트레이닝 벡터가 산만하게 퍼져있는 경우
(c) 동떨어진 웨이트벡터가 있는 경우

Fig. 2. Positions of the codevectors and training vectors in the hypersphere.
(a) the case in which training vectors are clustered properly.
(b) the case in which training vectors are distributed sporadically.
(c) the case of a distant codevector.

셋째, 초기 코드북을 잘 잡아 그림 2 (c)의 $W_3$ 와 같은 동떨어진 웨이트벡터가 없도록 해야 한다. 만약, $W_3$와 같이 동떨어진 웨이트벡터가 있었다면 이 $W_3$는 어떤 트레이닝 벡터에 대해서도 선택되지 못하기 때문에 학습을 할 수 있는 기회가 주어지지 않는다. 따라서, 전혀 쓰지 않는 코드벡터가 생겨나고 되고 이는 화질을 떨어뜨리는 가장 주된 요인이 된다. 실제로 화상 벡터 양자화에서 초기 웨이트 벡터를 임의적으로 잡았을 때, 특정한 웨이트벡터들만이 학습되고 나머지 웨이트벡터는 전혀 변화가 없게 되는 경우가 발생되며, 심한 경우에는 웨이트벡터 256개 중의 약 몇 개 정도만이 학습되고, 나머지는 전혀 학습되지 않는 경우가 있게 된다. 따라서, 順次的으로 圓心(centroid)을 갱신시키되, 위와 같은 경험적인 방법이 아니라, Ⅲ절과 같은 정확한 계산에 의하여 갱신시켜야 하며, 그림 2의 (c)에서처럼 동떨어진 웨이트벡터를 트레이닝 벡터들 쪽으로 끌어주는 알고리즘이 필요하다.

## Ⅲ. 最適區分化에 依한 順次的 更新 방법

LBG 알고리듬은 비슷한 성질을 갖는 트레이닝 벡터들을 모아 코드벡터 갯수 만큼의 클러스터로 나누고 각 클러스터의 원심을 구해서 코드벡터로 삼는 作業과 다시 그 코드벡터가 대표하는 가장 최적인 클러스터들로 트레이닝 벡터를 區分하고, 다시 그 클러스터의 원심을 구해서 코드벡터를 갱신하는 作業을 반복한다. 이 LBG 알고리듬으로 구성한 코드북은 적어도 국부적인 최적(local optimum)이 보장된다. 그러나 본 논문에서 제안할 順次的 更新 방법으로 얻어진 코드북이 최적이라는 것을 증명할 수 없기 때문에 본 논문에서는 LBG 알고리듬과 비교하여 성능이 같거나 더 좋으면 最適이라고 가정한다.

LBG 알고리듬에서처럼 트레이닝 벡터들을 먼저 클러스터로 나누고 그 클러스터의 원심을 구하는 방법도 있지만, 각 클러스터의 원심을 트레이닝 벡터가 입력됨에 따라 (3)식과 같이 갱신시켜 나갈 수도 있다. 예를 들어, 트레이닝 벡터 X′가 入力 되었을 때, 뉴우런 j가 다른 뉴우런들과의 경쟁에서 이겼다고 가정하면, 뉴우런 j가 대표하는 클러스터 $C_j$의 원심 $m_j$는 다음과 같이 更新된다.

$$m_j' = \frac{\sum_{x \in C_j} X + X'}{n_j + 1} = \frac{\sum_{x \in C_j} X}{n_j + 1} + \frac{X'}{n_j + 1} = m_j + \frac{X' - m_j}{n_j + 1} \qquad (3)$$

여기서, $m_j$는 갱신되기 전의 원심이며, $n_j$는 갱신되기 전에 그 클러스터가 대표하고 있던, 트레이닝 벡터의 갯수이다. (3)식을 보면 (2)식과 형태가 유사한 것을 알 수 있는데, (3)식에서는 $\eta(t)$대신 $1/(n_j+1)$이 이득계수 역할을 한다. 그러나, (3) 식에서의 이득계수는 (2) 식과는 달리 경험적인 것이 아니라, 그 클러스터의 정확한 圓心을 구하도록 유도된 것이며, 새로운 트레이닝 벡터가 그 클러스터에 기여하는 만큼만 원심을 갱신시킴으로써, 먼저 클러스터를 區分한 後에 그 클러스터의 원심을 구한 것과 同一하게 된다. 따라서, 이득계수에 따라 學習率이 달라지는 경우가 없으며 그 클러스터에 속한 트레이닝 벡터의 수가 많아질수록 $n_j+1$이 커지므로 이득계수는 작아진다고 할 수 있다.

LBG 알고리듬에서 이 원심은 코드벡터를 뜻하므로 트레이닝 벡터를 입력하면서, 신경회로망의 웨이트벡터를 (3) 식과 같이 갱신해 나가면 웨이트벡터는 클러스터의 원심이 되므로 LBG 알고리듬에서의 원심과 같게 된다. 여기서, 각 웨이트벡터는 클러스터의 원심을 나타내지만 각 클러스트가 最適(optimum)으로 區分化(partition)가 되었다는 保障이 없다. 그러므로, 최적화된 코드벡터를 얻기 위해서는 다음과 같은 최적 구분화 작업을 해야 한다.

초기 코드북을 잘 잡으면, 2 차원의 경우에 그림 3과 같이 트레이닝 벡터들의 分布 密度 函數(point density function[7]) 값이 큰 곳에서는 초기 코드벡터가 稠密하게 分布되고, 分布 密度 函數 값이 작은 곳에서는 초기 코드벡터가 드물게 分布하게 된다.

그림 3은 多次元에 대해 일반화한 Lioyd의 최적 양자화기 구성 알고리듬[2, 3]을 2 차원에서 살펴 본 것으로 스플리팅(splitting) 技法[2, 3]을 이용하면 그림 3과 같이 초기 코드북을 잡을 수 있다. 스플리팅 기법은

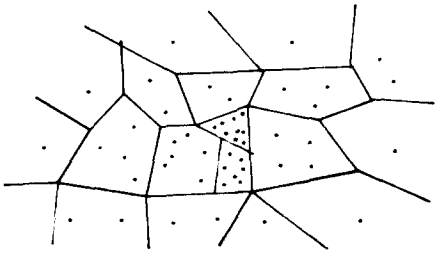


**그림 3.** 2 차원 벡터의 확률 밀도 함수에 따른 Dirichlet 구분화

**Fig. 3.** Dirichlet partition according to the point density function.

Smith의 壓伸(companding) 알고리듬[8]을 k 차원으로 확대한 것이라고 類推할 수 있다. 그러나, 프레임마다 이 스플리팅 기법으로 초기 코드북을 구해줄 수는 없기 때문에, 그림 2의 (c)에서와 같은 동떨어진 초기 웨이트벡터가 존재 하게 된다.

만약, 동떨어진 웨이트벡터가 존재하면 그 웨이트벡터는 전혀 更新이 되지 않으므로 쓰지않는 코드벡터가 생겨 코드북의 낭비를 가져 온다. 앞으로 쓰지 않는 코드벡터를 廢棄 코드벡터라고 指稱한다. 특히, 畵面의 切替(scene change)가 일어나면, 前 프레임의 코드북은 現 프레임의 초기 웨이트벡터가 되므로, 이런 동떨어진 웨이트벡터가 많게 된다. 일단 프레임이 바뀌어서 배경은 고정되고 화면내의 물체만 움직이는 경우는 화면의 통계적 성질이 비슷하고 변하는 부분이 별로 없기 때문에, 또 다른 화면의 절체가 일어나지 않는 한 이 동떨어진 웨이트벡터들은 갱신되지 않고 남아 있으며, 따라서 최적화에 이를 수 없다. 이런 現象을 막기 위하여 Kohonen의 신경 회로망에서는 $NE_{j^*}(t)$를 정하여 뉴우런 $j^*$가 경쟁에서 이겨도 $NE_{j^*}(t)$의 모든 뉴우런의 웨이트벡터들을 變更시켜 줬다. 그러나, 이것은 그림 2 (b)에서와 같은 현상이 발생할 수 있다. 그 이유는 $NE_{j^*}(t)$내의 뉴우런들이 서로 비슷한 웨이트벡터들을 갖고 있어야 하지만 트레이닝 벡터들이 超救體 上에서 산만하게 퍼져 있는 경우에는 뉴우런 j는 물론 그 이웃인 $NE_{j^*}(t)$내에 있는 뉴우런들까지 방향성을 갖지 못하고, 이리저리 끌려다니는 현상이 발생하기 때문이다.

最適化를 이루기 위해서는 각 클러스터에서 최적 원심이 무엇인가를 알고, 그에 맞도록 트레이닝 벡터들을 미리 구분화해야 하는데, 順次的 更新 알고리듬에서는 원하는 결과를 미리 알 수 없을 뿐만 아니라, 트레이닝 벡터의 통계적 성질도 모르기 때문에 區分化하는데 어려움이 있다. 다만, 初期 코드북을 스플리팅 技法을 이용하여 구성하면 트레이닝 벡터들의 確率 密度 函數에 맞도록 초기 코드북을 적절히 잡아주므로 적정한 구분화 기법 없이도 최적화로 收斂해 갈 수 있지만, 임의적으로 초기 코드북을 잡는 경우에는 폐기 코드벡터의 수가 많아지면서 수렴하지 않는다.

여러가지 양자화기에 대해서 어느 정도의 양자화 잡음의 下限(lower bound)을 구할 수 있다. Benett[9]는 1차원 양자화기에서의 量子化 雜音을 유도하였고 Gersho[10]는 Zador가 유도한 양자화 잡음의 下限을 k 차원 볼록 多角體(convex polytope)를 도입하여, k차원의 양자화 잡음 $D_1(N)$과 $D_2(H_Q)$를 유도하였다. 또한, Conway와 Sloane[10]은 여러가지 多角體에 대하여 Gersho가 유도한 양자화 잡음의 하한을 계산하였다. 이때의 결과는 코드벡터의 갯수 M이 무한대로 근접하는 근사적인 경우에 대해서 유도한 것이기 때문에 실제 유한개의 코드벡터를 갖는 경우에는 맞지 않지만, 결론적으로 Gersho는 다음의 두가지 사실을 언급했다.

첫째, M——>∞인 경우, 구분화된 각각의 클러스터는 최적 양자화기의 양자화 잡음에 똑같은 기여를 한다. 즉, 각 클러스터의 量子化 雜音은 同一하다.

둘째, M——>∞인 경우, 제한된 엔트로피(entropy)를 갖는 최적 양자화기는 거의 선형 양자화기(uniform quantizer)로 되어 간다.

위의 사실에서, 코드북이 最適化되면, 트레이닝 벡터의 分布 密度가 높은 곳에서는 그 클러스터에 속한 트레이닝 벡터의 갯수는 많지만, 그 클러스터의 自乘 誤差의 平均은 작고, 分布 密度가 낮은 곳에서는 갯수는 작지만, 自乘 誤差의 平均은 크다. 따라서, 각 클러스터의 전체 量子化 雜音은 同一하게 된다. 그러나, 코드벡터가 有限個인 경우에는 위의 두 사실이 완전히 맞아 들어가지는 않으며, 다만 그런 쪽으로 區分化를 해나가야 한다는 방향만을 제시해 주고 있다. 즉, 최적 양자화에서는 그림 3과 같이 분포 밀도가 높은 곳에서는 이웃한 코드벡터들 간의 거리가 가깝고, 分布 密度가 낮은 곳에서는 거리가 멀게 된다. 따라서, 하나의 클러스터에 일정량 이상의 트레이닝 벡터들이 속해지면, 그 클러스터를 계속 分離해 나가는 방법이 필요하게 된다. 그러므로, Kohonen의 방법 대신에 Bienenstock, Cooper, Munro에 의해서 제안된 방법[11]에 기초를 둔 새로운 방법을 제안한다. Bienenstock이 제안한 방법은 한 뉴우런이 경쟁에서 이기지 못하면 자신의 감도(sensitivity)를 증가시키는 방법이다. 반대로 너무 많이 경쟁에서 이기는 뉴우런은 자신의 감도를 감소시킨다. 이 방법은 각 뉴우런마다 臨界値(threshold)가 있다고 가정하고 이 臨界値를 초과하지 않는 입력에 대해서만 그 뉴우런이 경쟁에 참여할 수 있다고 가정하면 쉽게 구현된다. 만약, 뉴우런이 이기지 못하면 임계치를 증가시키고 이길 때는 임계치를 감소시킴으로써, 최종적으로는 모든 뉴우런이 반응하게 되는 방법이다. 그러나, 이 방법은 약간의 단점이 있다. 즉, 입력이 그 뉴우런의 임계치가 넘는 경우에 경쟁에서 그 뉴우런을 제외하면 그 뉴우런이 경쟁에서 이길 수 있는데도 불구하고 그 뉴우런을 경쟁에서 제외함으로써, 다른 덜 최적인 뉴우런이 경쟁에서 이기는 경우가 발생된다. 따라서, 본 논문에서는 위와 같은 臨界値를 이용하는 방법보다는 한 뉴우런이 너

무 많이 경쟁에서 이기는 경우에는 경쟁을 할 때 다른 뉴우런에 대해 이기기 힘들도록 조정을 해 준다. 즉, 競爭의 基準이 (1) 식과 같은 뉴우런의 웨이트벡터와 입력 트레이닝 벡터의 自乘 誤差(squared error)이고, 이 자승오차가 가장 작은 뉴우런이 경쟁에서 이긴다고 가정하면, 경쟁에서 많이 이긴 뉴우런은 그 뉴우런의 웨이트벡터가 대표하는 클러스터 $C_j$의 갯수 $n_j$가 클 것이므로 이 $n_j$에 비례하는 조정함수 $B(n_j)$를 (1) 식의 $d_j$에 곱해줌으로써, 경쟁에서 불리하게 한다. 즉, 경쟁에서 이길수록 앞으로의 경쟁에서 더욱 더 불리하게 되므로, 이길 가능성이 그만큼 더 줄어들게 된다.

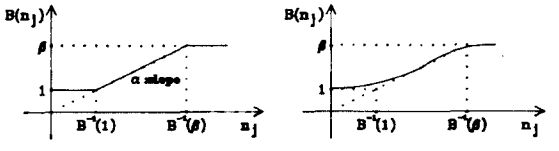



**그림 4.** 뉴우런의 조정함수
(a) 램프 조정함수　(b) sigmoid 조정함수

**Fig. 4.** Adjusting function of the squared error.
(a) ramp adjusting function.
(b) sigmoid adjusting function.

이 램프(ramp) 調整函數의 특성과 비슷한 것으로 그림 4의 (b)와 같은 sigmoid 함수가 있다. 이와 같은, 함수들은 뉴우런의 기본구조에서 非線形 函數(nonlinear function[6])와 근본적으로 같은 것이다. (2) 식과 같이 갱신을 하면서 그림 4와 같은 조정함수를 이용하면, 초기 코드북이 어떤 상태에 있어도 코드벡터를 최적으로 이끌 수가 있다.

LBG 알고리듬에서는 초기 코드북을 어떻게 잡느냐에 따라 수렴 상태가 달라진다. 또한, 트레이닝 벡터들 중에서 초기 코드북을 잡지 않고, 임의적으로 초기 코드북을 잡으면 LBG 알고리듬에서도 쓰지않는 코드벡터가 나올 수 있기 때문에, 그런 경우는 LBG 알고리듬을 돌릴 때 폐기 코드벡터를 버리고 가장 큰 클러스터를 分離하는 기능을 넣어야 하며, 전체적인 最適化를 위해서는 초기 코드북을 스플리팅 技法을 이용해서 構成하는 것이 바람직하다.

본 논문의 방법은 초기 코드북을 어떻게 잡아도 연속적으로 클러스터들이 분리되어 나가기 때문에 최적화로 수렴되어 나간다. 극단적인 경우로, 초기 코드북의 그레이레벨(gray level)을 모두 100으로 잡더라도 클러스터들이 자동적으로 분리되어 나간다.

다음 장에서는 改善된 Kohonen의 神經回路網 학습모델로 코드북을 構成하는 알고리듬을 살펴 본다.

## Ⅳ. 코드북 構成을 위한 神經回路網 모델

뉴우런들 간의 競爭의 基準이 되는 歪曲函數 $D_j$는, 超救體 上에서의 距離의 自乘, 즉, (1) 식의 $d_j$에 調整函數 $B(n_j)$를 곱해준 것으로 定義하고, (5) 식과 같다.

$$D_j = d_j \cdot B(n_j),\ j=1, 2, \cdots\cdots, M. \qquad (4)$$

여기서, M은 코드벡터의 갯수이다.

그림 4의 (a)에서와 같은 램프 調整函數에서의 기울기 $\alpha$와 限界値 $\beta$는 초기 코드북을 어떻게 잡느냐에 따라 성능의 차이가 나고, $\alpha$와 $\beta$를 크게 잡을수록 클러스터가 분리되는 경향이 커지며, $\alpha$와 $\beta$가 작을수록 클러스터들 사이의 벡터들의 이동이 줄어 들고 안정화되어 간다. 기울기 $\alpha$는 보통 0.1~10 정도이고, 한계치 $\beta$는 1~10000 정도까지 허용이 된다. 따라서, 이 $\alpha$와 $\beta$값으로 최적화의 수렴속도를 조정할 수 있는데, 이는 適應 信號 處理(adaptive signal processing)의 LMS 알고리듬에서와 마찬가지로 類推할 수 있다.

이 $\alpha$와 $\beta$값의 선택은 前 프레임의 코드벡터 중 폐기 코드벡터의 갯수에 따라 결정하면 된다. 즉, 폐기 코드벡터의 갯수가 많아지면, $\alpha$와 $\beta$의 값을 크게 해주어서 分離 기능을 강화시키고, 폐기 코드벡터의 갯수가 적은 경우에는 $\alpha$와 $\beta$의 값을 작게 함으로써 최적화쪽으로 클러스터들을 안정화시킨다.

본 논문의 신경회로망 모델을 이용하면 코드북을 만든 후 전송할 레이블(label)을 위한 탐색작업을 따로 할 필요가 없다. 왜냐하면, 하나의 트레이닝 벡터가 입력되었을 때 경쟁에서 이긴 뉴우런의 인덱스(index) j가 바로 그 트레이닝 벡터의 레이블이 되기 때문이다. 일단, 레이블이 정해진 후에도 다른 트레이닝 벡터가 뉴우런 j가 대표하는 클러스터의 원심을 (3) 식과 같이 다시 갱신시켜 코드벡터에 약간의 변화를 줄 수 있지만, 區分化가 最適으로 되면 그 차이는 무시할만하며, 실제로 SNR이 0.2dB 이상 차이가 나지 않는다.

本 論文의 神經回路網모델의 알고리듬을 정리하면 다음과 같다.

1) 임의의 코드북으로 초기 웨이트벡터를 잡는다. 따라서, 연속적인 프레임에서는 매 프레임마다 웨이트벡터를 임의적으로 잡지 않고, 그 前 프레임에서

만들어진 코드벡터를 그대로 현 프레임의 초기 웨이트벡터로 이용한다.

2) 각 뉴우런이 대표하는 클러스터 $C_j$의 원소갯수 $n_j$을 0으로 초기화 한다.

3) 트레이닝 벡터를 입력한다.

4) 모든 뉴우런의 웨이트벡터와 입력 트레이닝 벡터사이의 거리 자승 $d_j$를 (1) 식과 같이 구하고, 각 뉴우런 j의 $n_j$에 따라 (4) 식과 같이 $B(n_j)$를 곱해준 $D_j$를 구한다.

5) 가장 작은 $D_j$를 갖는 뉴우런 j*를 찾는다. 이 j*가 현재 입력된 트레이닝 벡터의 레이블이 된다.

6) j*번째 뉴우런의 웨이트벡터를 (3) 식과 같이 갱신한다. $n_{j*} \leftarrow n_{j*}+1$.

7) 한 프레임의 트레이닝 벡터가 모두 입력되면 2)로 가서 새로운 프레임에 대해서 다시 시작하고 한 프레임내에서 입력될 트레이닝 벡터가 아직 남았으면 3)으로 가서 반복한다. 한 프레임마다 각 뉴우런 j의 $n_j$를 전부 0으로 초기화하는 것은 프레임의 절체가 일어났을 때, 前 프레임의 $n_j$가 다음 프레임까지 영향을 미치는 것을 방지하기 위함이다. 또한, 이렇게 매 프레임마다 $n_j$를 0으로 초기화해도 코드벡터가 이미 어느 정도 형성되었다면, $n_j$의 값에 크게 영향을 받지 않는다.

## V. 實驗 및 結果

실험은 256×256크기의 LENA화상과 JAGUAR화상을 대상으로 행하였고, 코드북의 크기가 256인 경우 먼저 LENA화상에 대하여 LBG알고리듬으로 코드북을 얻은 후 그 코드북을 JAGUAR화상의 초기 코드북으로 놓고, JAGUAR화상의 트레이닝 벡터를 입력하면서 초기 코드북이 어떻게 更新되어 나가는가를 살펴 보았다. 또한, 실제의 연속적인 텔레비젼 프레임에 대하여 畵面의 切替가 일어나는 경우에 신경회로망이 최적화 방향으로 수렴되어 나가는 과정을 살펴보았다.

그림 5의 LENA와 JAGUAR화상을 코드벡터의 갯수가 256인 경우 LBG 알고리듬으로 코드북을 구성하면 복원된 LENA 화상의 SNR은 약 28.9dB, JAGUAR는 약 26.4dB가 된다. 이 LBG 알고리듬으로 얻은 화상의 SNR을 기준으로 제안된 알고리듬을 비교한다.

표 1은 LBG 알고리듬으로 만든 LENA 화상의 코드북을 초기 웨이터벡터로 놓고 JAGUAR 화상을 트레이닝 벡터로 하여 입력할 때, 반복횟수에 따른 SNR의 변화와 폐기 코드벡터의 갯수를 비교한 것이다.

(a) (b)

그림 5. 元화상
(a) LENA 화상 (b) JAGUAR 화상
Fig. 5. Original image.
(a) LENA image. (b) JAGUAR image.

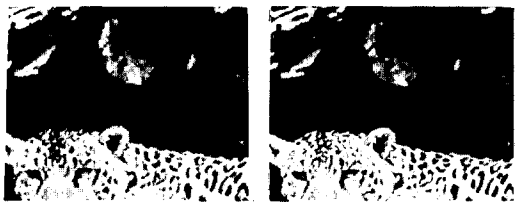
표 1. 반복횟수에 따른 폐기 코드벡터의 갯수와 SNR의 변화(램프 調整函數를 이용하는 경우)
Table 1. SNR and number of disused codevectors vs. iteration number when using a ramp adjusting function.

| 반복횟수 | | 1회 | | 2회 | | 3회 | |
|---|---|---|---|---|---|---|---|
| $\alpha$ | $\beta$ | 폐기코드벡터 수 | SNR | 폐기코드벡터 수 | SNR | 폐기코드벡터 수 | SNR |
| 0.1 | 2.0 | 23 | 26.07 | 21 | 26.33 | 21 | 26.36 |
| 0.1 | 8.0 | 21 | 26.02 | 20 | 26.28 | 18 | 26.33 |
| 0.5 | 2.0 | 4 | 26.23 | 2 | 26.43 | 2 | 26.44 |
| 0.5 | 80.0 | 0 | 25.80 | 0 | 26.01 | 0 | 26.01 |
| 0.9 | 2.0 | 4 | 26.32 | 2 | 26.42 | 2 | 26.49 |
| 0.9 | 8.0 | 0 | 25.89 | 0 | 25.90 | 0 | 25.91 |

SNR : dB

그림 4(b)의 sigmoid함수는 램프 함수와 특성이 비슷하기 때문에 생략하였다. 표 1에서 알 수 있듯이 반복 횟수에 따라 SNR이 꾸준히 증가하는 것을 알 수 있다. 이는 횟수가 반복될수록 전체적인 최적화로 나아가기 때문이다. 램프 조정함수의 경우는 $B(n_j)$를 곱하더라도 계속 경쟁에 참여시킴으로써 클러스터들 간의 이동이 있을 수 있기 때문에 반복횟수에 따라 SNR이 증가하며, 2~3번 반복으로 LBG 알고리듬과 거의 비슷한 성능을 얻을 수 있는 것을 알 수 있다. 또한, 한번 반복으로도 LBG알고리듬으로 얻은 SNR보다 약 0.1~0.4dB 정도 밖에 떨어지지 않는다. $\alpha$와 $\beta$의 값을 잘 선택해 줌으로써, 한번 반복으로 LBG 알고리듬과 거의 동일한 SNR을 얻을 수 있는데, $\alpha$가 0.9이고 $\beta$가 2.0인 경우가 SNR이 가장 좋게 나타났다. LENA화상의 코드북으로 JAGUAR 화상을

복원하는 경우의 SNR은 약 23dB 정도밖에 되지 않기 때문에, 한번 반복으로 약 3dB이상이 증가한 것을 알 수 있다. LENA화상의 코드북으로 직접 구성한 JAGUAR화상과 한번 반복해서 얻은 JAGUAR화상이 그림 6 에 있다.



(a)　　(b)

**그림 6.** (a) LENA 코드북으로 복원한 JAGUAR 화상(23dB)
(b) 한번 반복으로 얻은 코드북으로 복원한 JAGUAR 화상(26dB)

**Fig. 6.** (a) JAGUAR image reconstructed by LENA codebook.
(b) JAGUAR image reconstructed by the codebook obtained after one iteration.

표 1에서 알 수 있듯이 $\alpha$가 0.1인 경우는 분리 기능이 별로 없기 때문에 폐기코드벡터의 갯수가 약 20개 정도가 된다. 이때, SNR은 반복 횟수에 따라 서서히 증가하는데 이 폐기 코드벡터들의 갯수를 0으로 만들기 전에는 LBG 알고리듬과 같은 SNR을 얻을 수 없다. 그러나, $\alpha$가 0.1이라도 $\beta$가 2.0인 경우와 8.0인 경우를 비교하면, $\beta$가 8.0인 경우에 있어서는 $B(n_i)$의 한계가 6.0만큼 높아졌으므로, 그에 따른 폐기 코드벡터의 수는 감소하였지만, SNR은 약간 떨어진다. 이러한 SNR의 감소는 분리 기능을 크게 하면 구분화 작업이 안정되지 못하기 때문에 일어나는 것으로, 그에 따른 SNR의 감소는 무시할 수 있다. 이런 현상은 $\alpha=0.5$, $\beta=8.0$인 경우와 $\alpha=0.9$, $\beta=2.0$인 경우를 비교해도 알 수 있는데, $\alpha=0.9$, $\beta=2.0$인 경우는 폐기 코드벡터의 수가 2～4개 정도 있게 된다. 그러나, 클러스터들 간의 안정화를 이룰 수 있으므로 $\alpha=0.5$, $\beta=0.8$인 경우에 비해서 SNR은 약간 증가하게 된다.

한편, LFNA 코드북으로 JAGUAR 화상을 직접 복원하는 경우는 256개의 코드벡터중 150개만이 쓰였던 것과 비교해서 어떤 경우도 폐기 코드벡터의 갯수가 반복 횟수에 비례하여 大幅 감소하는 것을 알 수 있다.

결론적으로 $\alpha$와 $\beta$의 값은 前 프레임의 폐기 코드벡터의 갯수가 얼마나 생겼는가 하는 것에 따라서 임의적으로 변경시켜 줌으로써 최적화에 이를 수 있다. 즉, 폐기 코드벡터의 갯수가 많아지면 $\alpha$와 $\beta$의 값을 크게 해줘 분리 기능을 강화하고 폐기 코드벡터의 갯수가 작아지면 $\alpha$와 $\beta$의 값을 작게 함으로써 안정화를 기한다. 이때 $\alpha$와 $\beta$값은 프레임 단위로 고정시켜야 하며 프레임 內에서는 變更시키지 않는다.

이제 連續的인 畵像에 대해서 提案된 알고리듬을 적용하는 경우를 살펴본다. 그림 7과 같은 連續적인 8프레임에 대해서 컴퓨터 시뮬레이션을 하였다. 8프레임 중 4프레임은 연속적인 FAN 화상이고, 4프레임은 연속적인 HAND 화상으로 4개의 연속적인 FAN 화상 이후에 畵面의 切替가 일어났다고 가정한다. 이 8프레임에 대한 초기 웨이트벡터는 제안된 신경회로망을 이용해서 얻은 JAGUAR 화상의 코드북을 이용하였다. 표 2는 $\alpha=0.9$, $\beta=8.0$으로 고정시키고, 연속적인 프레임에 따라 폐기 코드벡터의 갯수, 제안된 알고리듬의 SNR, 각 프레임에 대해서 LBG 알고리듬으로 얻은 SNR을 비교한 것이다.

JAGUAR 화상은 전체적으로 어두운 부분이 많고, FAN 화상은 전체점으로 밝기 때문에 화상의 DC레벨이 크게 차이가 난다. 이런 경우는 극단적인 경우지만 실제로 발생될 수 있기 때문에 살펴 보았다. DC레벨 차이 때문에 표 2에서 알 수 있듯이 폐기 코드벡터의 갯수가 매우 많아졌으며 SNR의 급격한 감소를 가져 온다. 또한, 계속적으로 반복을 하더라도 이 프레임에 대해서는 $\alpha=0.9$, $\beta=8.0$으로는 분리가 일어나지 않기 때문에, $\alpha$와 $\beta$를 크게 해 주어야 함을 알 수 있다. 따라서, 표 3에 $\alpha$와 $\beta$값을 폐기 코드벡터 수에 比例해서 調整하는 경우의 SNR과 폐기코드벡터의 갯수, 그때의 $\alpha$와 $\beta$값을 제시하였다. 이 표에서 알 수 있듯이 畵面의 切替가 일어 나면 쓰지 않는 코드벡터의 수가 122개로 되었으므로, 프레임 2에서는 $\alpha$를 10, $\beta$를 10000으로 크게 증가시킨 결과 폐기 코드벡터의 갯수가 15개로 크게 줄었으며, SNR은 37.08dB로 크게 증가 하였다. 일단, 폐기 코드벡터의 수가 3개로 감소한 후의 안정화를 이루기 위해서 프레임 4에서 $\alpha=0.9$, $\beta=3.0$으로 감소시켰으며, 그 결과 LBG 알고리듬과 거의 동일하거나, 더 좋은 SNR로 수렴되는 것을 알 수 있다.

화면 절체후 SNR은 LBG 알고리듬보다 약 2 dB 정도 떨어지나, 이러한 畵面의 절체 후의 畵質의 저하는 우리눈이 인식할 수 없기 때문에 문제가 되지

표 2. 연속적인 프레임에서의 SNR의 변화 ($\alpha$=0.9, $\beta$=8.0로 고정된 경우)

**Table 2.** SNR variations in consecutive images when $\alpha$ and $\beta$ are fixed with 0.5 and 8.0.

| | 프레임 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|---|
| LBG | SNR | 35.31 | 37.20 | 35.42 | 35.97 | 35.26 | 34.06 | 34.34 | 34.55 |
| 제안된 방법 | SNR | 33.98 | 36.20 | 34.78 | 34.86 | 33.87 | 32.81 | 33.08 | 33.36 |
| | # | 122 | 115 | 103 | 90 | 77 | 60 | 55 | 54 |

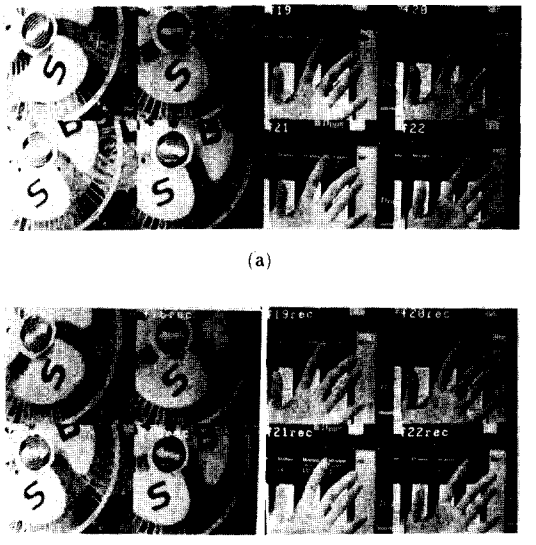
#: 폐기 코드벡터 갯수 SNR:dB

표 3. 연속적인 프레임에서의 SNR의 변화 ($\alpha$와 $\beta$값은 무효 벡터의 수에 비례해서 조정한 경우)

**Table 3.** SNR's when adapting $\alpha$ and $\beta$ according to the disused codevectors.

| 프레임 / 성능 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| $\alpha$ | 0.5 | 10.0 | 4.0 | 0.9 | 0.5 | 3.0 | 0.5 | 0.3 |
| $\beta$ | 2.0 | 10000 | 100 | 3.0 | 3.0 | 100 | 2.0 | 1.5 |
| SNR | 33.98 | 37.08 | 35.45 | 36.05 | 34.33 | 33.04 | 34.26 | 34.48 |
| 폐기코드벡터수 | 122 | 15 | 3 | 3 | 3 | 43 | 0 | 0 |

SNR:dB



(a)

(b)

그림 7. (a) 8 개의 연속적인 프레임
(b) 신경회로망으로 갱신시키면서 얻은 코드북으로 복원한 프레임($\alpha$와 $\beta$를 표3과 같이 변화시켰을 때)

**Fig. 7.** (a) 8 consecutive frames.
(b) 8 frames reconstructed by the codebooks obtained from the neural net when $\alpha$ and $\beta$ is changed according to Table 3.

않는다.[12] 이 신경회로망을 이용해서 프레임마다 얻어진 코드북으로 복원된 각 프레임은 그림 7과 같은데, 코드북의 크기가 256이기 때문에 약간의 블록성 雜音이 있지만, 깨끗하게 復原된 것을 알 수 있다.

제안된 알고리듬으로 구한 SNR은 코드북 생성 후에 다시 탐색(searching) 작업을 하지 않고 경쟁에서 이긴 뉴우런의 인덱스를 그대로 그 트레이닝 벡터의 레이블로 잡고 계산한 것이다. 나중에 레이블 探索 작업을 다시해서 화상을 복원하면 약 0.2dB정도 SNR의 증가를 가져오지만, 이것은 肉眼으로 구별할 수 없다.

## Ⅳ. 結 論

神經回路網을 이용하여 연속적인 프레임에서 코드북 갱신을 할 때는, 經驗的인 學習보다는 클러스터의 圓心을 更新시켜 나감으로써 最適化에 이를 수가 있었으며, 畵面의 切替가 일어나도 약 2프레임 後면 LBG 알고리듬과 同一한 성능을 얻을 수 있었다. 특히 제안된 알고리듬에서 이용된 調整함수는 동떨어진 코드벡터를 트레이닝 벡터쪽으로 誘引하는 性能이 뛰어나서 코드북의 效率性을 높여주므로 SNR의 현격한 증가를 가져온다. 畵面의 切替가 일어나는 경우 화상의 DC 레벨이 틀려지므로, 그때에는 분리 기능을 높여 주어야 클러스터의 분리가 활발해 진다. 일단, 어느 정도 分離가 되고 나면 기울기나 限界置등은 화상에 따라서 약간의 차이가 있지만 媒介變數의 변화에 따른 SNR의 변화는 약 0.5dB 以內이기 때문에 크게 문제가 되지 않는다.

또한, 제안된 알고리듬을 쓰면 코드북을 만든 후에 뉘시 레이블(label)을 探索하는 作業이 필요 없으며 코드북을 更新할 때 각 트레이닝 벡터에 대해서 競爭에서 이긴 뉴우런의 인덱스(index)가 레이블이 된다. 이렇게 구한 레이블로 화상을 복원하는 경우 나중에 다시 탐색하는 경우에 비해서 SNR이 0.2dB 이상 떨어지지 않는다.

특히, 이 방법은 實時間 具現이 가능하기 때문에 갱신된 코드북 전송을 위한 데이타 량을 줄이는 문제만 해결되면, 전송률이 0.5bit/pixel을 약간 넘는 수준에서 벡터 양자화를 구현할 수 있고, 코드북을 모두 전송하더라도 1.0bit/pixel 이면 가능하다.

## 參 考 文 獻


[1] A. Gersho, "On the structure of vector quantizers," *IEEE Trans. on Information Theory*, vol. IT-28, no. 2, pp. 157-166, Mar. 1982.

[2] Y. Linde, A. Buzo, R.M. Gray, "An Algorithm for Vector Quantization Design", *IEEE Trans. on Comm.* vol. COM-28, no. 1, pp. 84-95, Jan. 1980.

[3] R.M. Gray, "Vector quantization," *IEEE ASSP Magazine*, Apr. 1984.

[4] W. Equitz, "Fast Algorithm for vector quantization picture coding," Ph.D Dissertation of Stanford Univ., Jun. 1987.

[5] N.M. Nasrabadi, Y. Feng, "Vector quantization of images based upon the kohonen self organizing feature maps," *Proc. of IEEE International Conference on Neural Networks*, vol. 1, pp. 101-108, San Diego CA, Jun. 1988.

[6] R.P. Lippmann, "An introduction to computing with neural nets," *IEEE ASSP Magazine*, vol. 4, no. 2, apr. 1987.

[7] A. Gersho, "Asymptotically optimal block quantization," *IEEE Trans. on Information Theory*, vol. IT-25, no. 4, Jul. 1979.

[8] B. Smith, "Instantaneous companding of quantized signals," *Bell Syst. Tech. Journal*, vol. 52, pp. 1037-1076, Sept. 1973.

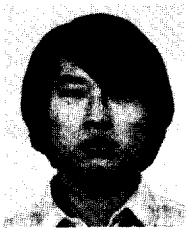
[9] W.R. Bennet, "Spectra of quantized signals," *Bell Syst. Tech. Journal*, vol. 29, pp. 446-472, Jul. 1948.

[10] J.H. Conway and N.J.A. Sloane, "Voronoi regions of lattices, second moments of polytopes, and quantization," *IEEE Trans. on Information Theory*, vol. IT-28, no. 2, Mar. 1982.

[11] R. McClelland, The PDP Research Group, Parallel Distributed Processing, vol. 1, pp. 179-180, MIT Press, 1986.

[12] A.J. Seyler, "The coding of visual signals to reduce channel capacity requirement," *Proc. IEE*, vol. 119, pt. c, pp. 676-684, 1962.


## 著 者 紹 介

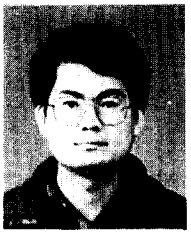



**丁 海 默**(正會員)

1962년 10월 9일생. 1985년 서울대학교 전자공학과 졸업. 1987년 서울대학교 대학원 전자공학과 졸업. 공학 석사학위 취득. 1987년 3월부터 현재 서울대학교 전자공학과 박사과정 재학중. 주 관심 분야는 통신방식, 화상 처리 및 코딩과 HDTV 처리 기술 등임.

●

**李 忠 雄** (正會員) 第 26卷 第 5號 參照.
현재 서울대학교 전자공학과 교수



**李 周 熙**(準會員)

1965년 5월 2일생. 1988년 서울대학교 전자공학과 졸업. 1988년 3월부터 현재 서울대학교 대학원 전자공학과 석사과정 재학중. 주 관심 분야는 통신 방식, 화상 처리 및 코딩 등임.