

論文95-32B-3-7

2 단 Self-Organizing Feature Map 을 사용한 변환 영역 영상의 벡터 양자화

(Image VQ Using Two-Stage Self-Organizing Feature Map in the Transform Domain)

李東學*, 金穎煥*

(Dong-Hahk Lee, and Young Hwan Kim)

요 약

본 논문에서는 신경망 모델을 이용한 변환 영역 영상의 분류 벡터 양자화 방법을 제안한다. 제안 방법에서는 벡터 양자화에 앞서 벡터 양자화의 목표와 동일한 특성을 가지고 있으며, 패턴 인식 특성이 있는 self-organizing feature map (SOFM) 을 이용하여 분류기에 사용할 클래스의 특징을 추출하였다. 기존의 방법과 달리 클래스 특징을 미리 정하지 않고, 주어진 학습 영상으로부터 클래스의 특징을 추출하여 사용하므로 영상질을 향상할 수 있다. 또한 신경망 모델의 적응적 특성을 이용하면, 이 방법은 처리하는 영상의 특성이 달라질 때 새로운 벡터 양자기 설계에 쉽게 적용할 수 있다. 각 클래스의 코드북은 일반화된 BFOS 알고리즘으로 비트 할당을 한 후 최상의 복원 영상질을 얻기 위하여 역시 SOFM 으로 설계하였다. 흑백 영상들에 대한 실험에서는 복원시 좋은 영상질을 얻었다. 또한 다른 분류 벡터 양자화 방법과는 비슷한 수준의 PSNR 을 얻었으며, JPEG 베이스라인 시스템 보다는 PSNR 이 높았다.

Abstract

This paper presents a new classified vector quantization (VQ) technique using a neural network model in the transform domain. Prior to designing a codebook, the proposed approach extracts class features from a set of images using self-organizing feature map (SOFM) that has the pattern recognition characteristics and the same as VQ objective. Since we extract the class features from the training images unlike previous approaches, the reconstructed image quality is improved. Moreover, exploiting the adaptivity of the neural network model makes our approach be easily applied to designing a new vector quantizer when the processed image characteristics are changed. After the generalized BFOS algorithm allocates the given bits to each class, codebooks of each class are also generated using SOFM for the maximal reconstructed image quality. In experimental results using monochromatic images, we obtained a good visual quality in the reconstructed image. Also, PSNR is comparable to that of other classified VQ technique and is higher than that of JPEG baseline system.

* 正會員, 浦項工科大学校 電子電氣工學科
(Dept. of Elec. Eng., Pohang Univ. of Science

and Technology)
接受日字 : 1994年 2月 12日

I. 서론

손실 영상 압축 (lossy image compression) 의 주된 목적은 복원시 인지할 수 없을 정도의 영상 질 (image quality) 손실을 허락하면서 비트율을 감소시키는 것이다^[1,2]. 일반적으로 비트율을 감소하기 위하여 인간의 시각 특성 뿐 아니라 영상에서 나타나는 중복성 (redundancy)^[1] 을 이용하고 있다. 손실 영상 압축에 사용하는 방법으로 DPCM (differential pulse code modulation)^[1,2], 변환 영역 부호화^[1,2,3,4,5], 벡터 양자화 (vector quantization)^[1,2,4], 하이브리드 부호화^[2,4], 그리고 영상 처리 방법들의 발달에 대응하여 이들의 적응적 방법들이 개발되었다. 이 방법들 중에서 벡터 양자화는 특히 고압축을 요구하는 영상의 부호화에 많이 사용되어 왔다^[1,2,4]. 그러나 전 탐색 (full-search) 벡터 양자화를 사용할 경우 코드북 설계 복잡도와 탐색 복잡도가 벡터 차원에 대하여 지수 함수적으로 증가하고^[2], 복원된 영상의 윤곽 특징이 나빠지는 현상이 발생한다^[3,6]. 이를 보완하기 위하여 분류 벡터 양자화 (CVQ) 방법^[6] 이 제안되었다. 이 방법은 전체 $N \times N$ 영상을 $n \times n$ 부분블럭 (subblock) 으로 나눈 후 각 부분블럭들을 특성에 따라 분류기로 분류하여 부호화를 수행한다. 여기에서 N 과 n 은 행과 열의 화소 수이다. 이와 같이 분류기는 부분블럭을 특성에 따라 주어진 클래스 중의 하나로 구분하는 기능을 하며, 복원 영상의 질을 향상하기 위해서는 이의 설계가 중요하다. 기존의 방법에서는 부분블럭의 특성을 상세히 조사하기 어려워 정해진 특징에 대하여 부분블럭내에서 이웃 화소의 상관 관계, 즉 경사 (gradient) 와 경험적으로 구한 수개의 문턱값을 사용하여 부분블럭을 분류하였다. 널리 사용하고 있는 대표적인 부분블럭의 특징은 수평, 수직, 대각 방향 블럭과 뒷배경 블럭이 있다. 이와는 다른 방법으로 패턴 분류 기능을 가진 신경망 모델 중에서 다층 퍼셉트론 (multi-layer perceptron)^[10] 을 부분블럭 분류에 사용한 방법이 있다^[11]. 그러나 이 구조는 지도 학습 (supervised learning) 을 하므로 학습전에 목표 출력이 필요하다. 따라서 학습에 앞서 경사와 문턱값을 이용하여 부분블럭의 특징을 미리 정해야 한다. 이와 같이 공간 영역에서의 분류 벡터 양자화는 수개의 문턱값의 사용과 경사를 구해야 하는 복잡한 분류기 설계 과정으로 인하여 최근 들어서는 변환 영역 부호화와 분류 벡터 양자화가 결합한 방법으로 DCT/CVQ^[2,4,5,7,8,9] 가 계산의 복잡도와 성능의 trade-off 을 제공하므로 많은 주목을 받게 되었다. 이 방법은 각 부분블럭에 대하여 DCT (discrete cosine transform)^[4]

를 수행한 후 분류 벡터 양자화를 하는 것이다. 기존의 DCT/CVQ 에서 사용하는 분류기는 부분블럭에서 미리 정해진 ac 에너지 분포 혹은 ac 계수 크기의 분포와 같은 DCT 의 특성을 이용하여 정해진 클래스 특징에 대하여 부분블럭을 분류하였다^[3,5,7,9]. 그러나 제안 방법에서는 설계전에 목표 출력을 미리 정하지 않고, 부분블럭의 특성을 조사하기 위하여 신경망 모델인 SOFM^[10,12] 을 이용하여 분류기를 설계한다. 이 모델은 식 (1) 을 목표로 입력 벡터 패턴의 특징을 나타내는 최적의 특징 맵을 구성하는 특성이 있으며, 이는 벡터 양자화의 목표와 동일하므로 벡터 양자화에 앞서 주어진 입력 데이터로부터 특징을 추출하는데 사용할 수 있다.

$$\min \sum_i^M \sum_j^{N_i} \sum_k^L (x_{ijk} - w_{ik})^2 \quad (1)$$

여기에서 M 은 출력 노드 수이고, N_i 는 클래스 i 에 속하는 부분블럭 수이며, L 은 입력 노드 수이다. 그리고 x 는 입력 벡터이고, w 는 시냅스의 웨이트를 나타낸다. 따라서 벡터 양자화에 앞서 분류기에 사용할 클래스 특징을 추출하는데 SOFM 을 사용함으로써 앞서 기술한 단점을 보완할 수 있다.

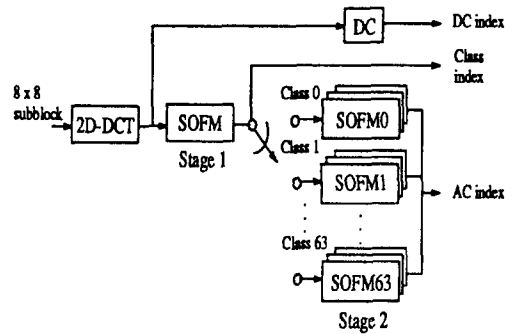


그림 1. 제안하는 방법의 블럭 다이어그램
Fig. 1. Block diagram of the proposed approach.

본 논문에서는 2 단 SOFM 을 이용한 새로운 분류 벡터 양자화 방법을 제안한다. 제안 방법은 벡터 양자화에 앞서 벡터 양자화의 복잡도와 영상질 향상을 위하여 첫번째 SOFM 으로 부분블럭을 분류할 수 있는 클래스 특징을 추출하였다. 그리고 분류 벡터 양자화의 목표는 주어진 클래스 수에 대하여 각 클래스의 특성에 따라 비트 할당을 달리하여 전체 평균 왜곡^[6] 이 작아지도록 하는 것이므로 제안 방법에서는 여러 클래스 수들에 대하여 학습한 결과를 토대로 전체 평균 왜곡을 사용하여 클래스 수를 선택하였다. 각 클래스는 수정 점근적 왜곡을 함수^[7] 를 이용하여 일반화된

BFOS 알고리즘^[13] 을 가지고 비트 할당을 하였으며, SOFM 으로 코드북을 설계하였다. 그림 1 에 제안하는 벡터 양자화 방법의 블록 다이어그램을 나타내었다. 첫번째 단은 부분력을 분류하는 분류기 설계를 위한 SOFM 이며, 두번째 단의 SOFM 은 코드북 설계를 위한 것이다.

본 논문의 구성은 다음과 같다. II 장에서는 SOFM 을 이용한 분류기의 설계에 대하여 기술하고, III 장에서는 각 클래스의 벡터 분할, 비트 할당, 코드북 설계에 관하여 기술한다. IV 장에서는 흑백 영상에 대해서 제안하는 구조의 성능을 평가하고, 다른 방법들과 비교한다. 끝으로, V 장에서 결론을 내린다.

II. 분류기 설계

일반적으로 부분력의 특징을 추출할 때 입력으로 공간 영역 혹은 변환 영역 데이터를 사용한다. 그러나 부분력에 변환을 수행하면 영상 데이터가 비상관화 (decorrelation) 되므로 부분력의 특징이 공간 영역에서와 같이 화소간의 상관 관계에 따라 결정되는 것이 아니라 DCT 계수에 의하여 결정된다. 그러므로 공간 영역에서 부분력을 분류할 때 수개의 문턱 값을 사용하고, 경사를 구해야 하는 복잡한 과정과는 달리 DCT 계수 값을 이용함으로써 부분력을 쉽게 분류할 수 있다. 또한 공간 영역의 분류 벡터 양자화에 비해 DCT 영역의 분류 벡터 양자화가 같은 비트율에서 2 dB 정도 우수하며^[13], 복원 영상의 질도 향상됨이 밝혀져 있다^[3,6]. 그러므로 본 논문에서도 DCT 영역의 데이터를 학습 데이터로 사용하여 SOFM 으로 분류기에 사용할 클래스 특징을 추출한다.

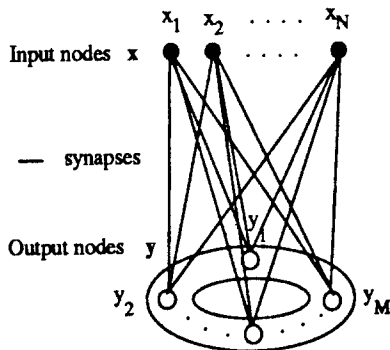


그림 2. 클래스 특징 추출을 위한 SOFM 모델
Fig. 2. The SOFM model for the class feature extraction.

SOFM 으로 클래스 특징을 추출하기 위하여 먼저 512×512 흑백 영상을 현재 널리 사용하고 있는 8×8 크기의 부영상으로 나눈다. 이 부분력에 2 차원 DCT 를 수행하면 부분력의 평균 밝기를 나타내는 dc 계수와 부분력의 특징을 나타내는 63 개의 ac 계수가 생긴다. 본 논문에서는 부분력 분류에 필요한 클래스의 특징을 추출하기 위하여 SOFM 의 입력으로 ac 계수를 사용하며, 출력 노드의 수는 2 의 제곱승인 8, 16, 32, 64, 128 개에 대하여 SOFM 을 학습한다. 클래스 특징 추출 SOFM 의 구조를 그림 2 에 나타내었다. 이 SOFM 을 학습하기에 앞서 제안 방법에 적용하기 위한 파라미터를 결정하기 위하여 각 출력 노드 수에 대하여 초기 이득항 값 η_0 를 0.01, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.5, 0.7, 0.9 로 하여 학습한 뒤 각 출력 노드 수에서 η_0 값에 대하여 식 (1) 을 비교하였다. 초기 이득항이 0.05~0.15 사이일 때 식 (1) 이 다른 값일 때보다 작았다. 그러므로 제안 방법에서는 이 영역의 값에서 초기 이득항 값으로 0.1 을 선택하였으며, 사용하는 이득항은 식 (2) 에 나타내었다.

$$\eta(t) = 0.1 \exp(-t/10^3) \quad (2)$$

여기에서 10^3 은 시상수이며, t 는 반복 회수 (iteration number) 를 나타낸다. 그리고 클래스 특징 추출을 위한 SOFM 의 neighborhood 는 식 (3) 을 사용하였다.

$$NE_w(t) = I \exp(-t/(5 \times 10^2)) \quad (3)$$

여기에서 I 는 초기 neighborhood 의 수이고, 제안 방법에서는 출력 노드 수의 약 2/3 로 하였으며, 5×10^2 은 시상수를 나타낸다.

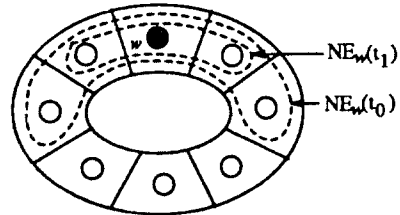


그림 3. 원형 neighborhood
Fig. 3. A circular neighborhood.

학습시 update 되는 출력 노드의 수는 $NE_w(t)$ 의 가장 근접한 정수값이며, 승리 노드를 기준으로 양쪽이 같은 수의 노드가 된다. 그러나 1 차원 배열된 출력 노드에서는 학습하는 동안 시냅스의 웨이트가 update

될 때 가장 자리 노드의 불공평성이 존재하므로 그림 3에 나타난 것과 같은 원형 neighborhood를 사용하였다. 또한 neighborhood 내에서 이웃 노드의 상관관계를 고려하여 시냅스 웨이트의 update 되는 양은 승리 노드에서 거리에 반비례하도록 하였다.

분류 벡터 양자화의 목표는 부블럭들 중에서 특성이 다른 부블럭을 여러 개의 클래스로 분류한 후 각 클래스별로 비트 할당을 달리하여 전체 평균 왜곡^[3,6]이 작아지도록 하는 것이다. 그러므로 본 논문에서는 여러 클래스 수들에 대한 학습 결과를 토대로 전체 평균 왜곡을 이용하여 클래스 수를 선택한다. 전체 평균 왜곡(D_{avg})은 식 (4)에 나타내었다.

$$D_{avg} = \sum_{i=1}^M P_i D_i \quad (4)$$

여기에서 M 은 클래스 수를 나타내고, P_i 는 i 번째 클래스에 부블럭이 속할 확률이다. 그리고 D_i 는 i 번째 클래스 점근적 왜곡의 최저 한계^[7]이다. 이 함수를 이용하면 코드북을 직접 설계하지 않고도 벡터 양자화시 발생하는 왜곡을 예측할 수 있으므로 설계하는 벡터 양자화의 성능을 평가할 수 있다. 그러나 최근 들어서는 실제적으로 사용하는 1.0 bpb (bit per pixel) 이하의 비트율에서 점근적 왜곡을 함수가 실험 값과 잘 일치하지 않아 수정한 함수가 사용되고 있다^[17].

각 클래스 수에 대하여 전체 평균 왜곡을 계산하기 위하여 각 클래스 별로 벡터를 분할하고, 비트 할당 알고리즘으로 각 클래스의 벡터별로 비트를 할당하였다. 벡터 분할 방법으로는 DCT 계수들이 *i.i.d.* Laplacian 소스^[3,7]라고 가정할 때 수정 점근적 왜곡을 함수가 벡터 기하 평균 분산에 비례하는 특성을 이용한 방법이 있다^[17]. 그러므로 이 방법은 왜곡을 작게 하기 위하여 벡터 기하 평균 분산이 작아지도록 벡터를 분할하였다. 제안 방법에서는 이를 기초로한 수정 방법을 사용하였으며, 이는 다음 장에서 상세하게 기술한다. 그리고 분할된 벡터별로 주어진 비트율에서 각 클래스의 벡터별로 일반화된 BFOS 알고리즘^[13]을 이용하여 비트 할당을 하였다. 이 알고리즘은 각 클래스의 주어진 비트에서 왜곡의 기울기를 계산하여 이 경사의 기울기가 작은 클래스에 대하여 주어진 비트를 감소시키면서 최적의 코드북 크기를 결정하는 방법이다. 제안 방법에서는 주로 사용하는 비트율인 1.0 bpb 이하에서 각 클래스 수에 대하여 비트 할당을 한 후 이 결과를 이용하여 전체 평균 왜곡을 계산하였다. 그림 4에는 주어진 비트율에서 계산한 전체 평균 왜곡으로부터 각 클래스 수에 대하여 8개 클래스의 전체 평균 왜곡을

기준으로 정규화한 전체 평균 왜곡을 나타내었다.

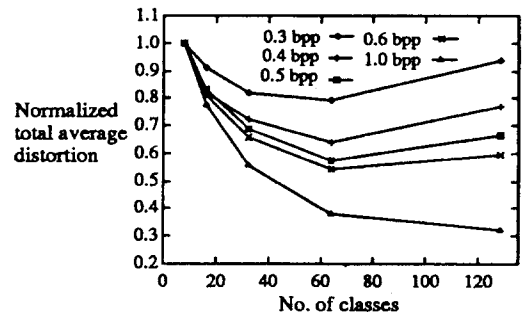


그림 4. 8개 클래스의 평균 왜곡으로 정규화한 전체 평균 왜곡

Fig. 4. A total average distortion normalized by 8-class average distortion.

그림 4에서 보는 바와 같이 클래스 수가 늘어나면 늘어날수록 클래스를 구분하기 위한 오버헤드^[5]는 증가하지만 실질적으로 부블럭들이 특성별로 분류가 상세하게 이루어지므로 전체 평균 왜곡이 감소하였다. 그러나 64개의 클래스에서는 클래스 수를 배로 늘려도 주어진 비트율에서 전체 평균 왜곡이 오히려 증가하거나 감소하는 정도가 작았다. 이는 클래스 수를 증가시켜 부블럭을 더 이상 분류하는 것은 벡터 양자화의 성능 향상에 도움이 되지 않는 것을 의미하므로 제안 방법에서는 64개의 클래스 수를 사용하여 분류 벡터 양자기를 설계한다. 64개의 클래스에 대하여 클래스 특징 추출 SOFM을 학습한 후 저장된 시냅스의 웨이트를 가지고 18개의 영상에 대하여 분류한 결과 전체 부블럭 수의 약 30%를 차지하는 클래스가 하나 존재하며, 이 클래스는 뒷배경 클래스에 해당한다. 이에 비해 나머지 클래스는 영상의 윤곽 특징^[3,5]을 나타내는 클래스들이다.

III. 코드북 설계

1. 벡터 분할

일반적으로 8×8 블럭을 하나의 벡터로 양자화를 수행하기에 입력 벡터 차원이 너무 크다. 그러므로 DCT 부블럭을 작은 차원의 수개의 벡터로 나누어 양자화를 수행하는 곱부호 (product code) 방법^[12]을 사용한다. 기존의 벡터 분할 방법 중에서 효율적인 방법으로 수정 점근적 왜곡을 함수를 기초로 한 방법이 있다^[7]. 이 방법으로 분할된 벡터는 저주파 뿐 아니라 고주파 영역의 계수를 같이 포함하고 있으므로 이 방법은 상세한 영상 (high-detailed image)에 잘

동작한다. 8×8 DCT 블록에 대하여 여러 영상들을 가지고 실험적으로 좋은 결과를 보인 벡터 분할을 그림 5 (a) ^[7] 에 나타내었다.

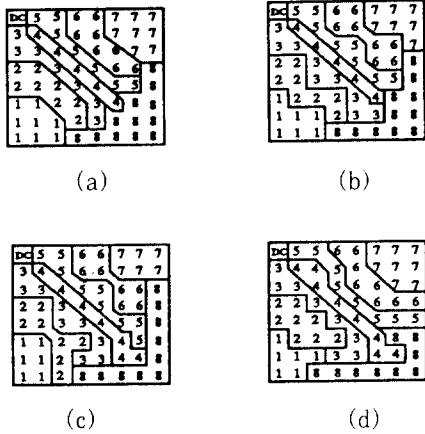


그림 5. 벡터 분할
Fig. 5. Vector partition.

그러나 분류 벡터 양자화에서는 부블럭의 특성에 따른 분류로 인하여 이 벡터 분할 방법을 모든 클래스에 동일하게 사용하면 비효율적이다. 그러므로 제안 방법에서는 클래스의 특성에 따라 서로 다른 벡터 분할을 하기 위하여 그림 5 (a) 를 초기 상태로 하여 다음 두가지 제한 조건으로 벡터 분할을 수정한다. 첫째, 벡터들 사이의 차원을 비슷하게 하기 위하여 부블럭내에서 행 방향으로 스캔할 때 현재 행과 열이 가지는 벡터 그룹의 변화는 이웃하는 8 개 계수의 벡터 그룹으로만 제한한다. 둘째, 후보들을 이동하여 벡터 기하 평균 분산이 서로 비슷한 값을 가지도록 한다. 즉, 벡터 기하 평균 분산의 표준 편차가 작아지도록 한다. 이렇게 되면 비트 할당시 비슷한 비트가 할당되므로 각 클래스의 벡터 양자기끼리 부호화 복잡도가 비슷해진다. 이 제한 조건을 이용한 벡터 분할 알고리즘을 그림 6 에 나타내었으며, 이 알고리즘을 이용하여 각 클래스의 특성에 따라 벡터 분할을 달리 하였다. 64 클래스에 대한 분할 결과의 일부를 그림 5 (b),(c),(d) 에 나타내었으며, 같은 번호를 가지는 성분들은 하나의 벡터로 양자화된다. 그리고 0.3~0.9 *bpp* 에 대하여 모든 클래스를 그림 5 (a) 로 분할하여 식 (4) 를 계산한 후 이를 기준으로 수정 방법과 비교한 결과 평균 2.1 % 정도 전체 평균 왜곡이 감소하였다.

2. 비트 할당과 코드북 설계

분류와 벡터 분할이 끝나면 수정 점근적 왜곡을 함수 ^[7] 를 이용하여 주어진 비트로 전체 평균 왜곡이 작아지도록 각 클래스에 대하여 비트 할당을 수행한다.

본 논문에서는 8×8 DCT 블록에서 ac 계수들은 분할된 벡터별로 수정 점근적 왜곡을 함수를 가지고, 일 반화된 BFOS 알고리즘을 이용하여 비트 할당을 하였다. 그리고 평균 밝기를 나타내는 dc 계수는 인지에 민감하므로 8 비트로 균일 양자화를 수행하였다 ^[5,8].

모든 클래스에 대하여 각 계수의 분산을 계산;
모든 클래스에 대하여 벡터 분할을 그림 5 (a) 로 초기화;
벡터 기하 평균 분산의 표준 편차를 계산. gsd;
foreach(클래스){

```
do {
    move = 0; /* 벡터 그룹의 변화 회수 */
    foreach(DC 열을 제외한 8x8 블록의 열) {
        foreach(DC 행을 제외한 8x8 블록의 행) {
            이웃하는 벡터 그룹중에서
            현재 행과 열의 벡터 그룹과 다른 벡터 그룹을 후보로 선택;
            벡터들의 차원과 기하 평균 분산 기억;
            foreach(후보, k) {
                현재 행과 열의 분산을 후보 벡터 그룹에 포함;
                벡터들의 차원, 기하 평균 분산 update;
                벡터 기하 평균 분산의 표준 편차를 계산.
                mgstd[k];
                벡터들의 차원, 기하 평균 분산을 원래 값으로 치환;
            }
            최소 mgstd[k] 및 벡터 그룹 기억;
            if(gsd > mgstd[k]) {
                현재 행과 열의 벡터 그룹을 기억된 벡터 그룹으로 치환;
                move++;
                gsd = mgstd[k];
                벡터 기하 평균 분산 계산;
            }
            else
                현재 행과 열의 벡터 그룹 유지;
        }
    }
} while(move != 0);
```

그림 6. 벡터 분할 알고리즘
Fig. 6. Vector partition algorithm.

평균 0.55 *bpp* 일 때 클래스의 빈도수, 비트 할당 결과, 왜곡을 표 1 에 나타내었다. 뒷배경 클래스, 클래스 0 보다 나머지 클래스, 즉, 윤곽 특징을 나타내는 클래스에 더 많은 비트가 할당되었으므로 벡터 양자화 후 이 클래스들에 속하는 부블럭들이 충실히 복원된다. 각 클래스별로 비트 할당이 이루어지면 이 할당된 비트 수 만큼의 코드 벡터를 가지는 코드북을 설계해야 한다. 널리 사용된 코드북 설계 알고리즘으로는 Linde-Buzo-Gray (LBG) ^[14] 알고리즘이 있으며,

표 1. 평균 0.55 bpp에서 얻은 비트 할당 결과

Table 1. Bit allocation results obtained at 0.55 *bpp* (average bit rate).

Class	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Prob.(%)	29.06	0.30	0.37	0.23	0.54	0.15	0.20	0.31	0.35	0.29	0.22	0.46	0.19	0.36	0.21	0.39
Bit	8	52	55	50	56	46	48	48	52	59	49	54	48	53	42	54
Dist.	123.5	67.7	70.4	75.0	64.5	76.8	70.9	90.6	95.8	40.9	66.9	78.1	73.4	72.7	96.5	73.8
Class	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Prob.(%)	0.22	0.80	0.51	0.72	0.65	0.6	0.85	0.39	0.48	0.50	0.49	0.92	0.69	0.57	0.18	0.27
Bit	36	52	56	61	58	53	54	54	54	55	64	60	65	56	42	59
Dist.	76.9	94.6	77.8	77.3	75.9	96.1	98.7	81.5	81.8	78.5	52.7	89.8	62.8	80.4	66.4	41.2
Class	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
Prob.(%)	0.23	0.38	0.32	1.35	1.72	0.77	0.84	2.18	7.11	1.70	0.94	0.59	8.78	1.17	6.52	0.72
Bit	42	55	61	59	56	56	57	46	12	53	54	67	11	55	16	68
Dist.	115.2	79.6	40.3	117.1	104.2	107.7	102.7	184.8	267.4	150.8	126.3	37.2	251.0	138.9	299.5	44.3
Class	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
Prob.(%)	7.10	1.11	0.47	2.01	1.74	0.69	0.74	1.92	1.12	1.15	1.31	0.39	0.97	0.63	0.34	0.47
Bit	17	55	56	52	54	60	60	48	64	64	59	51	56	51	53	56
Dist.	310.8	132.8	80.3	162.1	124.7	67.3	86.3	166.2	82.9	71.9	107.8	87.7	105.0	92.9	74.5	89.6

최근들어 신경망 모델을 이용한 방법으로 SOFM 과 FSCL (Frequency-Sensitive Competitive Learning)^[15] 이 있다. 참고 문헌 [16] 에서는 코드북 설계시 LBG 알고리즘과 SOFM 에 대하여 여러 항목들을 비교한 결과 SOFM 의 성능이 우수하였다. 참고 문헌 [15] 에는 코드북 설계시 LBG 알고리즘과 FSCL 의 성능을 비교하였다. 그러나 주어진 비트율에서 영상의 벡터 양자화시 MSE (mean square error) 를 비교한 실험 결과에서는 FSCL 이 LBG 알고리즘보다 큰 MSE 를 가졌다. 그러므로 제안 방법에서는 SOFM 을 가지고 코드북을 설계하였다.

각 SOFM 의 입력은 각 클래스에서 분할된 벡터의 ac 계수이다. 이에 비해 출력 노드의 수는 각 클래스별로 할당된 비트수에 따라 달라진다. 코드북 설계도 역시 패턴을 분류하는 것^[2] 이므로 코드북을 설계할 때 사용하는 파라미터에서 이득항은 분류기 설계시 사용한 식 (2) 를 사용하였다. 그리고 $NE_n(t)$ 는 출력 노드 수가 많을 때 대각 방향으로 이웃 노드의 상호관계를 고려하기 위하여 직사각형 neighborhood^[10] 를 사용하였으며, 출력 노드의 행과 열의 neighborhood 변화는 식 (3) 을 사용하였다. 이 파라미터들을 이용하여 각 클래스의 SOFM 을 학습하고 나면 각 SOFM 의 최종 시냅스 웨이트들은 각 클래스의 코드북을 나타낸다.

IV. 실험 결과

신경망 모델을 이용한 제안 방법의 성능을 평가하기

위하여 256 개의 그레이 레벨을 가지는 512×512 흑백 영상에 대하여 실험을 하였다. 벡터 양자화에 앞서 영상을 8×8 크기의 부영상으로 나누어 DCT 를 수행하였으며, 코드북 설계를 위한 학습 영상으로 18 개의 영상을 사용하였다. 제안 방법의 성능 평가를 위하여 4 개의 벤치 마크 실험 영상 (LENA, BRIDGE, PEPPER, BABOON) 을 선택하였다. 여기에서 LENA 와 BRIDGE 는 학습 집합에 포함되지 않는 영상이고, PEPPER 와 BABOON 은 학습 집합에 포함된 영상이다.

벡터 양자화 후 복원된 영상의 질은 다음과 같이 정의되는 PSNR (peak signal to noise ratio) 로 측정하였다.

$$\text{PSNR} = 10 \log_{10} [255^2 / (\frac{1}{M} \sum_{i=1}^M (x_i - y_i)^2)] \text{ dB} \quad (5)$$

여기에서 x_i, y_i 는 원영상과 복원 영상의 화소의 그레이 레벨 값이고, M 은 부호화하는 영상의 총 화소의 수이다. 그리고 분류 벡터 양자화에서 압축된 영상을 나타내기 위해 사용하는 화소당 평균 비트수는 부블럭이 어느 클래스에 속하는지를 나타내는 오버헤드 정보의 평균 비트율이 포함되어야 한다. 이를 포함한 비트율은 다음과 같다.

$$\text{Bit rate} = \frac{\text{Bit}_0 * C_0 + \text{Bit}_1 * C_1 + \dots + \text{Bit}_{63} * C_{63}}{N^2 * T_R} + \text{overhead} \quad (6)$$

식 (6) 에서 $\text{Bit}_i, i = 0, 1, \dots, 63$ 은 클래스별로 할당된 비트수를 나타내며, $C_j, j = 0, 1, \dots, 63$ 은 각 클래

스로 분류된 부분력의 수이다. 그리고 N 은 부분력의 행과 열의 크기이며, T_b 는 주어진 영상의 총 부분력의 수이다. 식 (6) 의 두번째 항인 overhead 는 부분력이 어느 클래스인지 나타내는 분류 코드의 평균 비트율이다. 제안 방법에서는 N 이 8 이고, T_b 는 4,096 이므로, overhead 는 $(4,096 \cdot 6) / 512^2 = 0.09375$ *bpp* 이다.

표 2. 학습 집합에 포함되지 않은 영상의 PSNR 비교

Table 2. PSNR comparison of the nontraining image.

Image		Prop.	Ref.[7]	JPEG
LENA	Bit rate(<i>bpp</i>)	0.429	0.416	0.430
	PSNR(dB)	32.85	32.82	32.72
	Bit rate(<i>bpp</i>)	0.500	0.473	0.501
	PSNR(dB)	33.81	33.43	33.70
	Bit rate(<i>bpp</i>)	0.604	0.600	0.605
	PSNR(dB)	34.85	34.74	34.71
BRIDGE	Bit rate(<i>bpp</i>)	0.520	0.518	0.522
	PSNR(dB)	25.96	25.94	25.64
	Bit rate(<i>bpp</i>)	0.656	0.648	0.659
	PSNR(dB)	26.91	26.84	26.50
	Bit rate(<i>bpp</i>)	0.775	0.768	0.777
	PSNR(dB)	27.72	27.64	27.12

표 2 에는 학습 집합에 포함되지 않은 영상의 PSNR 을 비교하였다. 표 2 의 Prop. 는 2 단 SOFM 을 이용한 제안 방법이고, 참고 문헌 [7] 은 부분력 분류를 위하여 9 개의 DCT 계수를 사용하고,

LBG 알고리즘으로 코드북을 설계한 방법이다. 그리고 JPEG^[17] 은 정지 영상 압축의 표준으로서 8 ×

8 DCT 를 사용한 변환 부호화 방법이다. 표 2 에 있는 *bpp* 는 식 (6) 를 이용하여 계산한 값이며, 제안 방법은 참고 문헌 [7] 에서 사용한 클래스 수와 코드북 크기가 같지 않으므로 비트율이 달라질 수 있다. 그러나 본 논문에서는 참고 문헌 [7] 의 방법, JPEG 베이스라인 시스템과 제안 방법을 비교하기 위하여 근접한 비트율이 되도록 하였다. 참고 문헌 [7] 과 비교한 결과 LENA 와 BRIDGE 영상은 제안 방법이 비슷한 비트율에서 평균 0.12 dB 가 높았다.

표 3. 학습 집합에 포함된 영상의 PSNR 비교

Table 3. PSNR comparison of the training images.

Image		Prop.	JPEG
PEPPER	Bit rate(<i>bpp</i>)	0.343	0.347
	PSNR(dB)	33.81	32.53
	Bit rate(<i>bpp</i>)	0.401	0.404
	PSNR(dB)	34.64	33.45
BABOON	Bit rate(<i>bpp</i>)	0.485	0.485
	PSNR(dB)	25.70	24.63
	Bit rate(<i>bpp</i>)	0.604	0.609
	PSNR(dB)	26.98	25.41

이에 비해 JPEG 베이스라인 시스템과 비교하면 거의 같은 비트율에서 평균 0.30 dB 가 높았다. 그림 7 에는 0.429 와 0.604 *bpp* 에서 복원한 LENA 영상을 나타내었다. 그림 7 (a) 에서는 약간의 블러화 현상과 윤곽의 흐림이 보이나 비트율이 증가한 그림 7 (b) 에서는 블러화 현상과 윤곽 특징의 나빠짐 현상이 감소하였다. 그림 8 에는 0.520 과 0.775 *bpp* 에서 BRIDGE 영상의 복원 결과를 나타내었으며, 그림 8 (b) 의 복원 영상에서는 영상 질이 향상됨을 볼 수 있



(a)

(b)

(c)

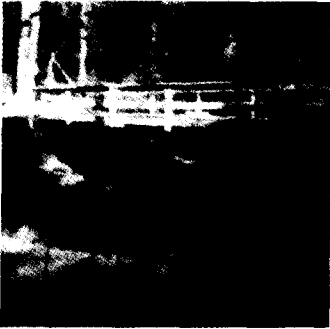
그림 7. LENA 영상 (a) 0.429 *bpp* (b) 0.604 *bpp* 에서 복원한 영상 (c) 원 영상

Fig. 7. LENA image (1) Reconstructed with 0.429 *bpp* (b) Reconstructed with 0.604 *bpp* (c) Original Image.

다.



(a)



(b)



(c)

그림 8. BRIDGE 영상 (a) 0.520 *bpp* (b) 0.775 *bpp*에서 복원한 영상 (c) 원 영상
Fig. 8. BRIDGE image (a) Reconstructed with 0.520 *bpp* (b) Reconstructed with 0.775 *bpp* (c) Original image.

학습 집합에 포함된 영상의 PSNR 비교는 표 3에 나타내었으며, JPEG 베이스라인 시스템으로 부호화하였을 때와 비교한 결과 PEPPER 영상의 경우는 제안 방법이 평균 1.24 dB 가 높았으며, BABOON 영상의 경우도 평균 1.32 dB 가 높았다. 이와 같이 제안

방법은 1.0 *bpp* 이하의 영상 압축 실험에서 좋은 성능을 보였다.

V. 결 론

본 논문에서는 신경망 모델인 SOFM 을 이용하여 영상을 압축하기 위한 DCT/CVQ 의 새로운 방법을 제안하였다. 제안하는 방법은 SOFM 을 2 단으로 구성하여 클래스 특징 추출과 각 클래스별로 코드북을 설계하였다. 기존의 방법과는 달리 클래스 특징을 정하지 않고, 클래스 특징을 조사하기 위하여 신경망으로 직접 추출하여 사용하였다. 분류기에 사용할 클래스 수는 전체 평균 왜곡을 이용하여 결정하였다. 추출된 클래스 특징을 이용하여 DCT 부블럭들을 분류한 결과 약 30 % 의 부블럭들은 뒷배경 클래스에 속하였으며, 나머지 부블럭들은 영상의 윤곽 특징을 나타내는 특징 클래스들로 분류되었다. 흑백 영상을 이용한 실험 결과에서는 참고 문헌 [7] 의 방법과 비슷한 비트율에서 평균 0.12 dB 가 높았으며, 복원한 영상에서 블러화 현상과 윤곽 특징 나빠짐 현상이 현저히 감소하였다. 그리고 정지 영상 압축 표준인 JPEG 베이스라인 시스템과 PSNR 을 비교한 결과 학습 집합에 포함되지 않은 영상에 대해서는 평균 0.30 dB 가 높았다. 제안 방법은 영상의 특성이 달라질 때 새로운 분류기 설계 시 클래스 특징을 정하지 않아도 되므로 새로운 분류 벡터 양자기 설계에 쉽게 적용할 수 있는 장점이 있다. 또한 이웃한 부블럭들을 가지고 입력된 부블럭의 특징을 예측할 수 있으므로 제안 클래스 설계 방법은 예측 벡터 양자화 혹은 FSVQ (finite-state VQ) 에 응용될 수 있다.

참 고 문 헌

- [1] M. Rabbani and P. W. Jones, *Digital Image Compression Techniques*. Bellingham, Washington: SPIE Optical Engineering Press, 1991.
- [2] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Boston, MA: Kluwer Academic Publishers, 1991.
- [3] D. S. Kim and S. U. Lee, "Image vector quantizer based on a classification in the DCT domain," *IEEE Trans. on Communications*, vol. COM-39, pp. 549-556, Apr. 1991.

- [4] K. R. Rao and P. Yip, *Discrete Cosine Transform Algorithms, Advantages, Applications*. San Diego, CA: Academic Press Inc., 1990.
- [5] J. Y. Nam and K. R. Rao, "Image coding using a classified DCT/VQ based in two-channel conjugate vector quantization," *IEEE Trans. in Circuits and Systems for Video Technology*, vol. 1, pp. 325-336, Dec. 1991.
- [6] B. Ramamurthi and A. Gersho, "Classified vector quantization of images," *IEEE Trans. on Communications*, vol. COM-34, pp. 1105-1115, Nov. 1986.
- [7] J. W. Kim and S. U. Lee, "A transform domain classified vector quantization for image coding," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 2, pp. 3-14, Mar. 1992.
- [8] K. Aizawa, H. Harashima, and H. Miyazawa, "Adaptive discrete cosine transform coding with vector quantization for color images," in *Proc. IEEE Int. Conf. ASSP*, 1986, pp. 985-988.
- [9] Y.-S. Ho and A. Gersho, "Classified transform coding of images using vector quantization," in *Proc. IEEE Int. Conf. ASSP*, 1989, pp. 1890-1893.
- [10] R. P. Lippman, "An introduction to computing with neural nets", *IEEE ASSP Magazine*, vol. 4, pp.4-22, Apr. 1987.
- [11] C.-C. Lu and Y. H. Shin, "A neural network based image compression system," *IEEE Trans. on Consumer Electronics*, vol. CE-38, pp.25-29, Feb. 1992.
- [12] T. Kohonen, "The self-organizing map," *Proc. of the IEEE*, vol. 78, pp.1464-1480, Sept. 1990.
- [13] E. A. Riskin, "Optimal bit allocation via the generalized BFOS algorithm," *IEEE Trans. on Information Theory*, vol. IT-37, pp.400-402, Mar. 1991.
- [14] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. on Communications*, vol. COM-28, pp.84-95, Jan. 1980.
- [15] A. K. Krishnamurthy, S. C. Ahalt, D. E. Melton, and P. Chen, "Neural networks for vector quantization of speech and images," *IEEE Journal on Selected Areas in Communications*, vol. 8, pp.1449-1457, Oct. 1990.
- [16] J. D. M. Auliffe, L. E. Atlas, and C. Rivera, "A comparison of the LBG algorithm and Kohonen neural network paradigm for image vector quantization," in *Proc. IEEE Int. Conf. ASSP*, 1990, pp.2293-2296.
- [17] G. K. Wallace, "The JPEG still picture compression standard," *IEEE Trans. on Consumer Electronics*, vol. 38, pp.xviii-xxxiv, Feb. 1992.

— 저 자 소 개 —



李東學(正會員)

1964년 11월 15일생. 1988년 2월 경북대학교 전자공학과 졸업(공학사). 1991년 2월 포항공과대학교 전자전기공학과 졸업(공학석사). 1991년 3월 ~ 현재, 동 대학원 전자전기공학과 박사

과정 재학중. 주관심 분야는 회로 시뮬레이션, VLSI 설계, 영상 신호 처리임.

金穎煥(正會員) 第 31卷 A編 第 12號 參照.

현재 포항공과대학교 전자전기공학과 조교수.