

움직임 보상 특성과 신경망을 이용한 차영상 분류 벡터 양자화

정회원 이경환*, 최정현*, 이법기*, 정원식*, 정태연*, 김덕규*

CVQ for MCDI Using the Characteristics of Motion Compensation and Neural Network

Kyeong-Hwan Lee*, Jung-Hyun Choi*, Bub-Ki Lee*, Won-Sik Chcong*, Tae-Yeon Jung*,
Duk-Gyoo Kim* *Regular Members*

요 약

동영상 부호화에서, 인접 프레임들 간의 시간적인 중복성이 제거된 움직임 보상된 차영상 (MCDI; motion compensated difference image)에 대한 효율적인 부호화를 위해 차영상 블록의 분산을 이용한 분류 벡터 양자화 (CVQ; classified vector quantization) 기법이 제안되었다. 본 논문에서는 MCDI 블록의 분산이 움직임 벡터 (motion vector)의 크기와 보상 블록 (compensated block)의 분산과 밀접한 관계가 가진다는 특성을 이용하여 새로운 MCDI CVQ 방법을 제안하였다. 또한 코드북 설계시 학습 속도가 빠르고, 코드벡터 초기화 및 학습 과정 중에서 사용될 수 있는 분류 경계지점 학습 벡터의 최종 코드북에 대한 영향이 적은 FSCL (frequency sensitive competitive learning) 신경망을 사용하였다. 따라서 부호화시 부코드북 (sub-codebook)을 선택하는 부가 정보가 필요 없으며, 움직임 보상 과정과 병렬적으로 보상 블록의 분산을 계산하므로 부호화시간을 줄일 수 있다. 모의 실험 결과, 분류 비트가 필요한 기존의 방법에 비해 제안한 방법이 우수한 성능을 나타내었다.

ABSTRACT

CVQ (classified vector quantization) using the variance of the motion compensated difference block was presented as an efficient MCDI (motion compensated difference image) coding method. In this paper, we found that this difference block variance is closely related to both the motion vector magnitude and the corresponding compensated block variance. From this property, we propose a new MCDI CVQ method, that generates no side information for what sub-codebook is selected and reduces the coding time since the variance of the compensated block can be computed in parallel with motion compensation process. FSCL (frequency sensitive competitive learning) neural network is used in VQ codebook design for its fast training speed and low damage from training vectors located at the class boundaries, which can be used in codebook initialization or on the process of the training. Simulation results show that the proposed method exhibits a good performance when compared with conventional methods.

I. 서론

영상 신호는 프레임내 (intra-frame)의 공간적 상관성 (spatial correlation)이 높으며, 특히 동영상 신호는 인접 프레임간 (inter-frame)의 시간적 상관성 (temporal correlation) 또한 매우 높다. 동영상 부호화는, 인접 프레임간의 시간적 중복성 (temporal redundancy)을 제거하는 움직임 추정 과정과, 그 후 움직임 보상된 영상과 원영상과의 차영상 (MCDI;

호는 인접 프레임간 (inter-frame)의 시간적 상관성 (temporal correlation) 또한 매우 높다. 동영상 부호화는, 인접 프레임간의 시간적 중복성 (temporal redundancy)을 제거하는 움직임 추정 과정과, 그 후 움직임 보상된 영상과 원영상과의 차영상 (MCDI;

* 경북대학교 전자전기공학부 (khlee@palgong.knu.ac.kr)
논문번호 : 99367-0908, 접수일자 : 1999년 9월 8일

motion compensated difference image)의 공간적 중복성 (spatial redundancy)을 제거하는 MCDI 부호화 과정으로 이루어진다. 움직임 추정 기법들 중 화소 단위의 추정 방법 (PRA; pixel recursive algorithm)은 알고리즘이 복잡하고 계산량이 많아서 잘 사용되지 않고, 대부분의 경우 하드웨어 실현이 간단하고 계산량이 적은 블록 정합 방법 (BMA; block matching algorithm)으로 움직임을 추정한다^[1]. 그러나 BMA 방법은 블록 단위의 움직임 추정 방법이기 때문에, MCDI 블록 내의 화소들 사이에는 공간적 상관성이 다소 남아 있게 된다^[2]. 그러므로 MCDI의 효율적인 부호화가 복원 영상의 화질 측면에서 매우 중요한 역할을 한다.

MCDI 부호화 기법들 중, DCT (discrete cosine transform)를 이용한 방법에서는 MCDI의 특성 때문에 모든 주파수에 걸쳐서 변환 계수값이 고루 분포되므로, 부호화 효율이 떨어지며 구획화 현상 (blocking effect)이 생긴다^[3]. 이를 보완하기 위해 Furner 등은 MCDI 부호화에 벡터 양자화 (VQ; vector quantization)를 이용하는 방법을 제안하였는데^[4], 블록들마다의 특성을 고려하지 않고 모든 MCD 블록들에 대해 단일 코드북 (codebook)을 사용하였으므로 화질을 크게 향상시킬 수 없었다.

분류 벡터 양자화 (CVQ; classified VQ) 방법에서는, 훈련 벡터 (training vector)를 그 특성에 따라 몇 개로 분류하고 각각에 대해 다른 부코드북 (sub-codebook)을 만든다^[5]. 이 방법에서는 부호화할 때 분류기 (classifier)로써 입력 벡터의 특성에 맞는 부코드북을 선택하여 양자화함으로써 부호화 효율을 향상시킬 수 있다. Akansu 등은 MCDI를 부호화할 때 블록의 에너지에 따라 CVQ하는 방법을 제안하였다^[6]. 이 방법을 사용하면 각 블록의 에너지를 나타내는 분산 (variance)의 크기에 따라 만든 여러 개의 부코드북들을 사용하여 MCDI를 부호화함으로써, 화질을 향상시켰다. 그러나, 움직임을 추정한 후 MCDI 블록의 분산을 계산해 주어야 하므로 부호화 시간이 많이 걸리며, 분류 정보에 대한 부가 비트의 전송이 필요하다.

MCD 블록의 특성은 움직임 벡터 (motion vector)의 크기와 관계가 큰데, 일반적으로 움직임이 큰 영역일수록 현재 프레임과 이전 프레임의 화소들간의 시간적 상관성이 떨어지게 되어 MCDI 블록이 큰 분산을 가진다^[7]. 한편, MCDI 블록과 보상 블록 (compensated block)도 많은 상관성을 가진다. 다시 말하면, 보상 블록이 에지 부분일 경우 움직임 추정

오차는 주로 큰 값을 가지게 되어, 보상 블록의 분산이 크면 MCDI 블록의 분산도 커지는 경향이 있다^[8].

이러한 특성들을 이용하여, 본 논문에서는 움직임 벡터와 보상 블록의 특성을 이용한 MCDI 부호화 방법을 제안하였다. 움직임 벡터의 크기와 보상 블록의 분산을 곱하여, 이 값을 코드북의 분류 정보로 하여 MCDI를 CVQ하였다. 부호화기에서 움직임 보상과 상관없이 이미 복원된 이전 프레임 블록들의 분산을 미리 계산하므로 기존의 방법보다 부호화 시간을 줄일 수 있으며, 움직임 벡터와 보상 블록을 부호기측에서도 알고 있으므로 분류 정보에 대한 부가 비트가 전송될 필요가 없다.

또한 코드북 설계시 GLA (generalized Lloyd algorithm) 방법이 주로 사용되는데^[9], 이 방법은 학습벡터들을 모두 저장한 다음 초기화된 각 코드 벡터로부터 코드북이 지역적으로 최적화 (locally optimization) 되도록 반복하는 방법이므로, 최종 코드북은 초기 코드벡터들에 많은 영향을 받는다. 따라서 분류 경계지점의 벡터가 초기 코드벡터로 사용되면 코드북의 효율성이 떨어지므로, 본 논문에서는 초기화 또는 코드북 학습 중에서 사용될 수 있는 발생빈도가 낮은 분류 경계지점의 벡터에 의한 영향을 적게 받는 FSCL (frequency sensitive competitive learning) 신경망을 사용하였다^{[10][11]}. 이 방법은 또한 설계 전과정 동안 학습벡터들을 저장할 필요가 없어 기억용량을 줄일 수가 있고 한 번의 학습벡터의 입력에 하나의 코드벡터만이 갱신되므로 학습시간이 짧은 장점을 가진다.

모의 실험 결과, 제안한 방법의 분류기로 차영상 분산을 이용한 분류기를 적절하게 대체할 수 있음을 확인할 수 있었으며, 시험 영상에 대해서 우수한 성능을 나타내었다.

II. 움직임 보상된 차영상의 벡터 양자화

BMA 방법에서는, 블록내의 화소들이 동일한 움직임을 가진다는 가정 하에, 영상을 작은 블록으로 나누어 블록단위로 이전 프레임의 주어진 탐색 영역 내에서 움직임을 추정한다. 움직임 벡터는 현재 프레임의 블록과 최소 오차를 가지는 이전 프레임 블록과의 변위로 결정되고, 빠른 탐색을 위해 블록 정합 척도로는 평균 절대 오차 (MAD; mean absolute difference)가 널리 사용된다.

동영상에서, 움직임 추정을 통하여 현재 프레임은

이전 프레임의 블록들로 보상하더라도 이들 사이의 오차 성분은 무시할 수 없을 정도의 큰 값을 가지므로, 우수한 복원 영상을 얻기 위해서는 이 차영상을 따로 부호화 해주는 것이 필수적이다. 또한 영상 신호는 이웃 화소들간의 공간적 상관성이 크기 때문에, 대부분의 부호화 방법들에서는 효율적인 정보 압축을 위하여 영상을 블록 단위로 부호화한다. 특히 동영상 부호화에서 BMA 방법으로 움직임을 추정하면, 블록내의 모든 화소들이 동일한 움직임 벡터를 가진다는 가정하에 움직임 보상이 블록을 기반으로 행해지므로, MCDI에는 블록 단위의 화소들간의 중복성이 많이 남아있게 된다. MCDI를 부호화하는 방법들 중, DCT 등 변환 부호화 방법은 MCDI의 특성상 변환 계수값들이 저주파에서 고주파까지 고루 분포되어서 압축률을 높이기 어렵고 높은 압축률로 부호화할 경우 구획화 현상이 생기는 등 비효율적이다. 따라서 높은 압축률에서도 비교적 블록내의 공간적 중복성을 잘 제거하여 좋은 성능을 나타내는 VQ를 이용한 MCDI 부호화 방법이 제안되었다⁴⁾.

MCDI에는, 움직임이 비교적 정확하게 추정되어서 오차가 적은 부분이 있는 반면, 시간에 대한 화소값의 변화가 아주 클 경우나 움직임이 너무 커서 탐색 영역을 벗어나는 경우에는 오차가 큰 부분이 생긴다. 이러한 특성을 지니는 MCDI를 부호화할 때, 단일 코드북을 사용하여 VQ하면 부호화 효율이 떨어진다. 더욱이 통계적으로 분산이 작은 블록이 많은 일반적인 MCDI를 훈련 영상 (training image) 으로 하여 코드북을 만들었을 경우, 분산이 작은 코드벡터 (code vector)들이 많이 만들어져서, 분산이 큰 블록에 대한 부호화 오차가 커지므로 전체 영상의 화질을 떨어뜨리는 원인이 된다.

여러 개의 부코드북을 사용하여 양자화하는 CVQ는 블록 벡터를 그 특성에 맞게 효율적으로 부호화

하는 방법이다. MCDI 블록의 분산을 분류값으로 두고 MCDI를 CVQ하는 방법이 제안되었으며⁶⁾, 그림 1에서는 이러한 분류기를 가지는 동영상 부호기의 블록도론 나타내고 있다. 이 방법은 좋은 화질을 나타내지만, MCDI 블록을 부호화할 때마다 블록의 분산을 구해야 하므로 부호화 시간이 많이 소요되고, 또한 분류기가 어떤 코드북을 선택하였는지를 나타내는 부가적인 분류 정보 (classifying information)가 수신측에 전송되어야 하므로 비트율이 높아지는 단점이 있다.

III. 제안한 MCDI 부호화

1. 움직임 보상 특성을 이용한 MCDI CVQ

동영상 부호화에서 움직임 추정으로 시간적인 중복성이 제거된 후에도, 움직임이 클 경우에는 차영상에 공간적인 중복성이 많이 남아 있게 되어서, MCDI 블록은 적지 않은 분산값을 가지게 된다. 즉, 움직임이 큰 영역일수록 현재 프레임과 이전 프레임의 화소들간의 시간적 상관성이 떨어지게 되어 MCDI 블록의 에너지가 커지는 경향이 있으며⁷⁾, 이는 움직임 벡터의 크기가 MCDI 블록의 분산에 영향을 미치기 때문이다. 그러나, 작은 분산값의 평탄 영역에 있는 현재 블록과 보상 블록들은 이러한 경향을 따르지는 않는다.

또한, 현재 블록이 이전 프레임의 에지 부분으로 보상될 경우, 작은 움직임에 대해서도 상대적으로 많은 오차가 생기므로 MCDI 블록의 에너지가 커지며⁸⁾, 이는 보상 블록의 분산과 MCDI 블록의 분산은 서로 큰 상관성을 가지기 때문이다. 즉, 현재 블록의 움직임이 클 경우 MCDI 블록은 큰 분산값을 가지나, 평탄 영역에 있는 현재 블록과 보상 블록들은 이러한 경향을 따르지는 않으며, 또한 보상 블록의 분산이 클 경우에는 움직임이 작더라도 MCDI 블록은 큰 분산값을 가지는 경향이 있다.

본 논문에서는 위의 특성들을 근거로 움직임 벡터의 크기와 보상 블록의 분산을 곱하여 이를 분류값(classifying value)이라고 정하고, 이 분류값을 이용하여 MCDI를 CVQ하였다. 분류값은 다음과 같다.

$$\begin{aligned} & \text{if } \text{Var}(\text{Com. block}) = 0, \\ & \quad \text{Classifying value} = \text{Mag}(\text{Motion vector}) \\ & \text{else if } \text{Mag}(\text{Motion vector}) = 0, \\ & \quad \text{Classifying value} = \text{Var}(\text{Com. block}) \end{aligned}$$

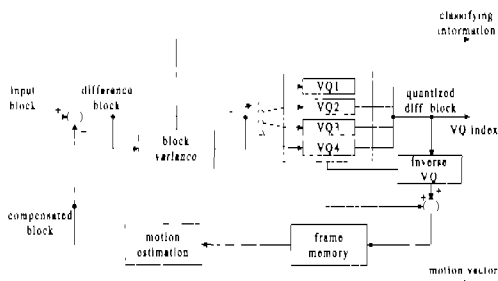


그림 1. 차영상 분산 분류기를 이용한 MCDI CVQ 동영상 부호기

else,
 Classifying value =

$$\text{Mag}(\text{Motion vector}) \cdot \text{Var}(\text{Com. block})$$
 (1)

이때 $\text{Var}(\cdot)$ 은 블록 벡터 요소 (element)의 분산율, $\text{Mag}(\cdot)$ 은 크기를, Com. block은 보상 블록을 각각 뜻한다.

그림 2는 FOOTBALL 영상에 대한 제안한 분류값과 MCDI 블록의 분산과의 관계를 보인 것인데, 기존 방법의 분류값인 MCDI 블록의 분산을 직접 구하지 않더라도 이를 적절히 대처할 수 있음을 알 수 있다. 그림 3에서는 제안한 분류값에 대한 MCDI 블록들의 발생빈도로 통계적 특성을 나타내었다. 제안한 분류기를 이용한 MCDI CVQ 동영상 부호기의 블록도도 그림 4에 나타내었다.

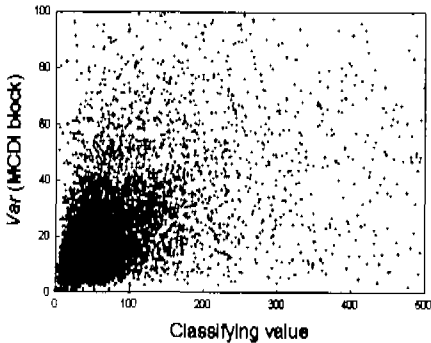


그림 2. 제안한 분류값과 MCDI 블록의 분산과의 관계

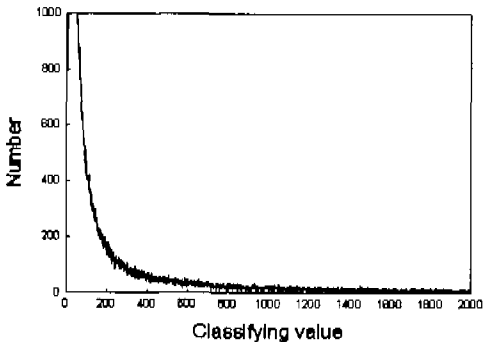


그림 3. 제안한 분류값의 통계적 특성

이 부호기에서는 분류 정보가 부가적으로 전송되지 않고, 보상 블록의 분산이 움직임 보상 과정과 병렬적으로 구해져서 이로부터 분류값을 구할 수 있다. 또한, 일반적으로 움직임 추정에서의 기본 블록의 크기가 VQ에서 블록 벡터의 크기보다 크므로, 움직임 벡터의 크기는 VQ 부호화를 할 때마다 매

번 구하지 않아도 되어 이로인한 계산량에 큰 영향을 미치지 않는다. 반면 부호화 시간의 측면에서 볼 때, 제안한 방법의 부호화 시간은 기존의 CVQ를 이용한 방법보다는 빠르고, 단일 코드북의 VQ를 이용한 방법과는 차이가 없다.

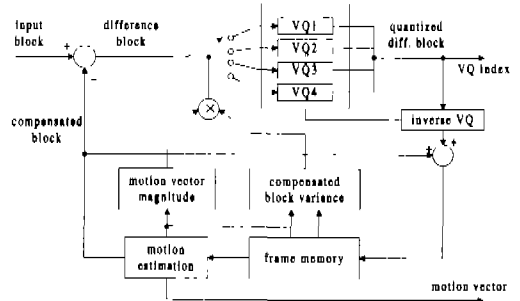


그림 4. 제안한 분류기를 이용한 MCDI CVQ 동영상 부호기

2. FSCL 신경망을 이용한 코드북 설계

일반적인 VQ에서의 코드북 설계방식으로 GLA 방법이 널리 사용된다. 이 방법은 흔히 LBG 알고리즘으로 알려져 있으며, 먼저 코드벡터들을 초기화시킨 후, 학습 벡터들을 모두 받아들인 다음 NN (nearest neighbor) 조건에 따라 최소 거리 (distance)의 코드벡터가 같은 것들끼리 학습 벡터의 집합 (cluster)을 만들고 중심 (centroid) 조건에 의해 집합의 평균으로 새로이 코드벡터를 갱신한다. 이러한 과정은 전체 학습 벡터들과 각각이 속한 집합의 코드벡터들의 거리가 미리 정한 수준까지 수렴할 때까지 반복되고, 반복과정의 모든 끝나면 최종 코드북이 만들어진다^{[8][9]}.

즉 이 방법은 초기 코드벡터로부터 코드북이 지역적 최소화되도록 반복하는 설계방법이므로, 최종 코드북은 초기화에 많은 영향을 받게되어 초기화물 부적절하게 하면 각 집합에 속한 입력벡터의 수가 많은 차이를 보여 코드북의 효율성이 떨어지게 되고, 학습 전과정에 학습 벡터들을 모두 저장하고 있어야 하므로 많은 기억용량이 필요하며, 한번의 반복 과정내에 모든 학습 벡터들과 각각이 속한 집합의 코드벡터들 간의 거리를 계산하여야 하므로 코드북 설계시간이 많이 걸리는 단점이 있다.

제안한 방법의 분류값은 기존 분류기의 MCDI 블록의 분산을 효과적으로 대체하는 예측값이므로, 임의의 클래스 (class)와 이웃 클래스 분류 경계지점의 벡터가 코드벡터로 되면 코드북의 효율성이 나빠진다. 따라서 본 논문에서는 학습 과정중에 전체

왜곡 (distortion)을 최소화하면서 코드북의 엔트로피 (entropy)를 최대화시키는 FSCL 신경망은 이용하여 코드북을 설계하였다^[11]. 즉, 이 방법은 초기화나 학습시 분류 경계지점의 벡터가 사용되어도 전체 학습 벡터에서 차지하는 빈도가 작다면 최종 코드북에 영향을 주지않아 효율적인 코드북을 만들 수 있다.

코드북 학습에 사용된 신경망은 그림 5에서와 같이 입력층 (input layer)과 출력층 (output layer or winner-take-all layer) 두개의 층으로 되어 있는데, M개의 신경단위 (node)를 학습시킨다면 코드벡터를 $W_i(0), i = 1, 2, \dots, M$ 을 초기화 시키고, 학습에 쓰이는 벡터의 수가 N개라면 이들을 차례로 입력하여 각 신경단위의 코드벡터들과 거리계산을 통해 최소거리의 승리한 신경단위를 정하고, 승리한 신경단위의 코드벡터를 학습 벡터쪽으로 갱신시킨다. t 시점의 입력벡터 X에 대해 각 대표벡터들의 거리를 측정했을 때 최소거리의 신경단위가 j라면, 다음 시점 t+1에서의 해당 코드벡터는

$$W_j(t+1) = W_j(t) + \epsilon(t)(X - W_j(t)) \quad (2)$$

와 같이 갱신된다. 이때 $\epsilon(t)$ 는 학습비율로서 학습 과정이 경과함에 따라 0.9에서 0으로 선형적으로 감소하는 함수이다. 반복과정이 모두 끝나면 코드벡터들은 최종 코드북의 코드벡터가 된다. 이러한 신경망을 CL (competitive learning) 신경망이라고 하는데, 이를 그대로 사용하면 일부 신경단위에서만 편중하여 갱신이 이루어질 수 있어 여전히 초기화 문제가 남아있게 된다. 따라서 이를 개선하여 거리계산시 해당 신경단위의 승리한 횟수 $u_i(t)$ 를 곱하여 다음과 같은 변형된 거리계산법을 사용한다.

$$d^*[X, W_j] = d[X, W_j(t)] \times u_j(t) \quad (3)$$

이때 $u_i(t)$ 는 이전의 학습과정 중에 i번째 신경단위가 선택된 승리 횟수이므로, 이 방법으로 학습을 시킬 경우 입력 빈도가 높은 벡터들 쪽으로 많은 코드벡터가 만들어 진다. 이와 같이 의도적으로 적게 선택된 신경단위를 승리할 수 있게 만들어 코드북 전체가 발생빈도가 높은 입력 벡터들로 적응해 나가도록 하여 코드북의 효율성을 높인 방법을 FSCL (frequency sensitive competitive learning) 신경망 방법이라고 한다. 즉, FSCL 신경망의 학습 과정은 입력되는 학습벡터들에 대해 신경단위들이 전체 왜곡을 최소화 하면서 코드북의 엔트로피를

최대화 시키려는 과정이며, 따라서 일부의 분류 경계지점의 벡터들이 선계된 코드북에 영향을 적게 주므로 제안한 방법에 적합한 코드북 설계 방법이다.

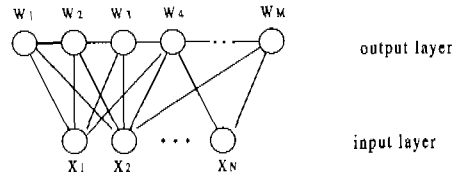


그림 5. FSCL 신경망의 구조

IV. 실험 결과 및 고찰

제안한 방법의 성능 비교를 위해 모의 실험은 행하였다. 움직임 추정은 16×16 크기의 블록 단위로 전역 탐색 블록 정합 방법 (full search BMA)으로 행하였으며, 탐색 영역은 -16 ~15로 하였다. VQ에 사용될 벡터의 크기는 16차원, 즉 4×4 블록으로 하였으며, 704×480 크기의 FLOWER GARDEN, MOBILE 및 TABLE TENNIS 각각 5 프레임, 모두 15 프레임을 혼련 영상으로 사용하여 각 코드북을 설계하였다.

움직임 탐색에 사용된 거리계산법으로 다음과 같은 MAD를 사용하였다.

$$d[X, Y] = \frac{1}{K} \sum_{k=1}^{K-1} |x_k - y_k| \quad (4)$$

이때, X, Y는 거리를 계산할 K차원 블록 벡터이고 x_k, y_k 는 각 블록의 요소값이다. 또한 VQ 코드북 설계 및 부호화 과정에는 다음과 같은 평균 자승 오차 (MSD; mean squared difference)를 사용하였다.

$$d[X, Y] = \frac{1}{K} \sum_{k=1}^{K-1} (x_k - y_k)^2 \quad (5)$$

또한 복원된 영상의 성능을 평가하기 위해, 널리 사용되는 PSNR (peak signal-to-noise ratio)를 사용하였다.

$$PSNR = 10 \log_{10} \left(\frac{255^2}{FMSD} \right) \text{ [dB]} \quad (6)$$

이때 FMSD는 원 프레임과 각 방법으로 복원한 프레임간의 전체 MSD이다.

한편 CVQ를 행할 때 클래스를 나누는 문턱값은,

코드북 설계에 쓰인 영상 15 프레임의 분류값을 구하여 그것의 통계적 특성을 보고 각 클래스에 똑같은 개수의 벡터가 속하도록 문턱값을 조절하여 최종 문턱값을 구했는데, 기존 방법의 분류기는 9, 25, 100으로 두었으며 제안한 분류기는 80, 300, 1200으로 하였다.

먼저, 단일 코드북 VQ 방법의 코드북을 128 크기로 만들었고, 기존의 CVQ 방법 및 제안한 방법의 각 부코드북(sub-codebook)들도 128 크기로 만들었다. 표 1에서는, 움직임이 큰 영상인 FOOTBALL, 카메라의 움직임이 있는 FLOWER GARDEN 영상의 혼련에 쓰이지 않은 부분, 비디오폰 영상인 SUSIE 각 20 프레임에 대한 각 방법들로 부호화를 했을 때의 비트율과 평균 PSNR을 나타내었다. 제안한 방법은 동일한 비트율을 가지는 단일 코드북을 이용한 VQ 방법이나 DCT 방법보다 좋은 화질을 보이고, 분류 비트가 첨가되어 부코드북들의 크기는 같으나 비트율이 높은 기존의 CVQ 방법에 비해 FOOTBALL의 경우 22.2%의 비트율을 감소시키면서 0.7%의 PSNR 감소만을 보여 좋은 성능을 보임을 확인할 수 있다.

CVQ와 또 다른 장점은 각 클래스(class)의 코드북 크기별 분류값의 생동폭(dynamic range)이나 분포 특성에 따라 효과적으로 결정하여 비트율을 낮추고 코드북 탐색시간을 줄일수가 있다는 것이다. 제안한 CVQ 방법의 부코드북의 크기별 64, 128, 128, 256의 크기와 32, 64, 128, 256의 크기로 실험하였으며, 유사한 비트율에서의 비교를 위하여 기존의 방법의 부코드북의 크기는 16, 32, 32, 64로 하였다. 표 2에서는 기존의 CVQ 방법과 제안한 CVQ 방법의 복원 영상의 평균 PSNR 결과를 나타내고 있다. 제안한 방법은 기존의 방법에 비해 FOOTBALL 영상의 경우 4.6~12.8%의 비트율을 줄이면서 1.1~4.7%의 PSNR의 증가를 보이는 등 우수한 결과를 나타냄을 볼 수 있다.

표 1. FOOTBALL, FLOWER GARDEN, SUSIE 영상에 대한 비트율과 평균 PSNR 비교

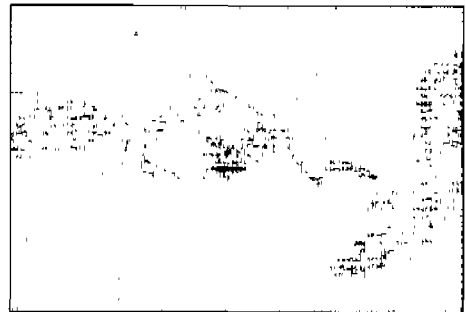
DCT	0.4375	0	27.78	28.38	37.67
VQ	0.4375	0	28.53	28.55	38.29
CVQ	0.4375	0.125	29.66	30.24	39.25
Proposed	0.4375	0	29.45	29.91	38.82

표 2. 부코드북의 크기를 가변시켰을 때의 비트율과 평균 PSNR 결과

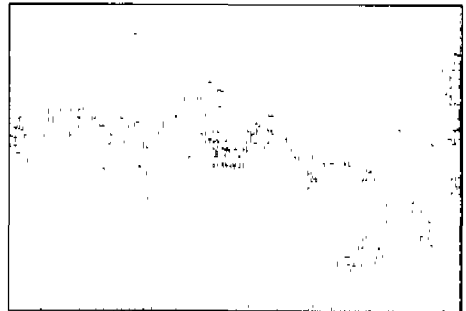
CVQ		0.4425	28.75	0.4298	29.57	0.3936	38.42
Proposed	Scheme 1*	0.4218	30.09	0.4496	30.21	0.3849	38.62
	Scheme 2**	0.3858	29.17	0.4277	29.85	0.3246	38.48

*; the sizes of each sub-codebook is 64, 128, 128 and 256

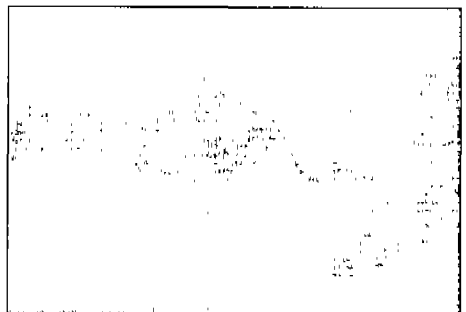
**; the sizes of each sub-codebook is 32, 64, 128 and 256



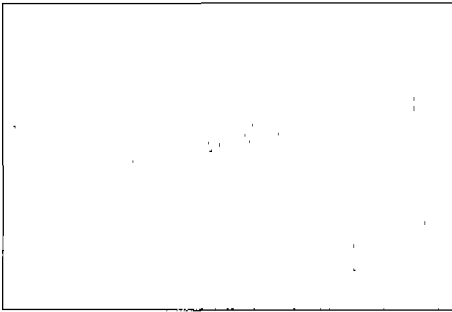
(a)



(b)



(c)



(d)

그림 6. FOOTBALL 3번째 프레임과
 (a) DCT (0.4375bpp, 26.84dB),
 (b) VQ (0.4375bpp, 28.66dB),
 (c) conventional CVQ (0.4420bpp, 28.90dB),
 (d) proposed CVQ (0.4193bpp, 29.33dB)로 복원한 영
 상의 차영상

그림 6에서는 FOOTBALL 원영상 한 프레임과
 사 방법으로 부호화하여 복원한 영상의 동일 프레
 임 사이의 차영상을 나타내고 있는데, 제안한 방법
 이 가장 낮은 비트율임에도 가장 작은 왜곡을 보인
 을 알 수 있다.

V. 결 론

본 논문에서는, 움직임 벡터와 보상 블록을 이용
 한 분류기틀 제안하고, 이 분류기틀로써 MCDI 블록
 을 CVQ하는 새로운 MCDI 부호화 방법을 제안하
 였다. 기존의 MCDI 블록의 분산으로 CVQ할 때
 전송되던 클래스에 대한 부가 정보없이 움직임 보
 상 과정에서 수신측에서 알고있는 움직임 보상 특
 성인 움직임 벡터의 크기 및 보상 블록의 분산은
 이용하여 분류값을 예측하였다. 또한 VQ 코드북 설
 계시 분류 경계지점의 벡터가 초기화나 학습 과정
 에서 사용되어 최종 코드북의 코드벡터가 되는 것
 을 방지하기 위해 FSCL 신경망을 이용하였다. 모의
 실험 결과 제안한 방법은 부가 정보가 없으므로 비
 트율을 줄이면서 우수한 PSNR 성능을 나타냄을 확
 인할 수 있었다. 또한 보상 블록의 분산을 계산하는
 과정과 움직임 보상을 이용한 차영상을 만드는 과
 정을 병렬로 처리함으로써, 실제의 부호기 하드웨어
 설계에서 부호화 시간을 적절히 줄일 수 있을 것으
 로 기대된다.

참 고 문 헌

- [1] A. N. Netravali and J. O. Limb, "Picture Coding: A Review," Proc. of IEEE, vol. 68, no. 3, pp. 366-406, Mar. 1980.
- [2] H. G. Musmann, P. Prisch, and H. J. J. Grallert, "Advances in Picture Coding," Proc. of IEEE, vol. 73, no. 4, pp. 523-548, Apr. 1985.
- [3] P. Strobach, "Trec-structured Scene Adaptive Coder," IEEE Trans. on Communication, vol. 38, no. 4, pp. 477-486, Apr. 1990.
- [4] R. R. Furner, R. W. Christiansen, and D. M. Chabrics, "Motion Compensated Vector Quantization," ICASSP, pp. 989-992, 1986.
- [5] B. Ramamuthi and A. Gersho, "Classified Vector Quantization of Image," IEEE Trans. on Communication, vol. COM-34, no. 11, Nov. 1986.
- [6] A. N. Akansu and M. S. Kadur, "Adaptive Vector Quantization of Video Signals with motion compensation and spatial masking," Proc. of IEEE Int'l Symposium Circuits and Systems, pp. 1378-1381, May, 1989.
- [7] A. N. Netravali and B. G. Haskell, *Digital Pictures*: Plenum, 1995.
- [8] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*: Kluwer Academic Publishers, 1992.
- [9] Y. Linde, A. Buzo, and R. M. Gray, "An Algorithm for Vector Quantizer Design," IEEE Trans. on Communication, vol. COM-28, pp. 84-95, Jan. 1980.
- [10] S. C. Ahalt, A. K. Krishnamurthy, P. Chen, and D. E. Melton, "Competitive Learning Algorithms for Vector Quantization," Neural Networks, vol. 3, pp. 277-290, 1990.
- [11] J. E. Fowler Jr., M. R. Carbonara, and S. C. Ahalt, "Image Coding Using Differential Vector Quantization," IEEE Trans. on Circuits and systems for video technology, vol. 3, no. 5, Oct. 1993.

이 경 환(Kyeong-Hwan Lee) 통신학회 논문지 제24권 제1B호 참조	정회원
최 정 현(Jung-Hyan Choi) 통신학회 논문지 제24권 제1B호 참조	정회원
이 범 기(Bub-Ki Lee) 통신학회 논문지 제24권 제1B호 참조	정회원
정 원 식(Won-Sik Cheong) 통신학회 논문지 제24권 제1B호 참조	정회원
정 태 연(Tae-Yeon Jung) 통신학회 논문지 제24권 제8B호 참조	정회원
김 덕 규(Duk-Kyoo Kim) 통신학회 논문지 제24권 제1B호 참조	정회원