동적 주소 사상을 이용한 벡터 양자화

배성호 † · 서대화 † · 박길훈 †

요 약

본 논문에서는 인접블록들간의 높은 상관성을 이용한 동적 주소 사상에 의한 벡터 양자화 방법을 제안한다. 제안한 방법에서는 부호화할 입력블록에 대한 벡터 양자화의 주소를 사이드 메지 오자를 이용하여 재정렬된 부호화에서의 새로운 주소로 사상하는 주소 변환 함수를 정의하여 비트율을 효율적으로 감소하였다. 이러한 방법은 주소 변환 함수에 의한 새로운 주소가 주소 분리값이 아닌 내부로 사상된 경우는 새롭게 사상된 주소를 부호화하고, 그렇지 않은 경우에는 재정렬되지 않은 부호벡터 주소를 부호화하는 방법이다. 실험을 통하여, 제안한 방법에서의 복원영상의 화질은 일반적인 벡터 양자화 방법에서의 복원영상의 화질과 동일하고 비트율은 약 45~50% 감소함을 확인하였다.

Vector Quantization Using a Dynamic Address Mapping

Sung-Ho Bae † · Dae-Wha Seo † · Kil-Houm Park †

ABSTRACT

In this paper, we propose a vector quantization method which uses a dynamic address mapping based on exploiting the high interblock correlation. In the proposed method, we reduce bit-rate by defining an address transform function, which maps a VQ address of an input block which will be encoded into a new address in the reordered codebook by using side match error. In one case that an original address can be transformed into a new transformed address which is lower than the threshold value, we encode the new address of the transformed codeword, and in the other case we encode the address of the original codeword which is not transformed. Experimental results indicate that the proposed scheme reduces the bit-rate by 45~50% compared with the ordinary VQ method for image compression, at the same quality of the reconstructed image as that of the ordinary VQ system.

1. 서 론

최근 디지털 영상 신호를 압축하여 저장하거나 전송하는 영상 부호화 기법들에 대한 많은 연구가 진행되고 있다. 이러한 영상 부호화 기법들 중 벡터 양자화 (VQ: Vector Quantization)는 낮은 전송률에서도 효율적인 성능을 나타내는 압축 기법으로 하드웨어 기

† 경희대학교 공과대학 전자공학과
논문결과: 1996년 1월 18일, 심사완료: 1996년 5월 30일

술의 발전과 더불어서 점차 실용화가 가까워지고 있다[1]. 또한, Shannon의 Rate Distortion 이론[2]에 의하면 스칼라 대신 벡터로 조합된 신호를 부호화하는 것이 낮은 전송률에서도 좋은 성능을 얻을 수 있으므로 영상 신호를 벡터 양자화하여 큰 데이터 압축 효과를 얻을 수 있다. 벡터 양자화는 메모리가 없는 벡터 양자화로 메모리가 있는 벡터 양자화와 달리 메모리가 있는 벡터 양자화는 인접한 블록의 정보를 이용하지 않고 독립적으로 블록들을 양자화하는 방식이므로 일반 영상과 같이 인접블록들간의 상
관성이 큰 정보원을 부호화할 때에 불특갈간의 상관성과 연속성을 이용하자 못하게 된다. 반면에, 메모리가 있는 벡터 양자화는 인접블록간의 상관성을 이용하여 부호화하여 압축 효율을 크게 높일 수 있다. 벡터 양자화는 양자화되는 단위 벡터의 차원들로 추적된 적절한 성능을 나타낸다고 알려져 있다[2]. 그러나 실제 단위 벡터의 차원은 CPU 시간과 메모리 저장 측면에서 기하급수적으로 증가하는 복잡도에 의해 제한되어야 한다. 따라서 이 문제를 극복하는 방법은 인접블록들의 부호벡터들(occupant) 주소를 상관없이 부호화함으로써, 인접블록간에 잔여하는 상관성을 이용하는 방법이 필요하다. 이러한 인접블록간의 상관성을 적절히 이용하여, 평균 황록을 일반적인 벡터 양자화 시스템에서의 값을 유지하면서 평균 비율을 감소시키는 대표적인 방법에는 FSVD(spell finite state VQ)[3], PVQ(predictive VQ)[4]와 AVQ(address VQ)[5] 등이 제안되었다.

일반적인 영상에서는 인접화소들 간에 화소값의 급격한 변화가 없는 화소값들간의 연속성을 가지므로, 인접블록들간에 서로 결합하고 있는 화소값들도 연속성을 가지게 된다. 그러므로 본 논문에서는 인접블록들간에 서로 결합하고 있는 화소값들의 오차를 제한하는 사 이드 매치 오차[6]가 작은 순으로 부호화 내의 부호벡터를 재정렬하여, 압축율에 대응하는 부호벡터 주소가 새로운 주소로 사상하는 주소 변환 함수를 이용한 벡터 양자화 방법을 제안한다. 제안한 방법에서는 부호화한 입력블록에 인접한 이미 부호화된 인접블록들과 부호화된 부호벡터와의 사이드 매치 오차(side match error)가 작은 순으로 부호화된 빌딩의 경우, 인접블록들간에 서로 결합하고 있는 화소값들의 연속성을 이용하여, 새롭게 사상한 부호벡터 주소의 확률 분포가 낮은 주소로 집중한 좌표들이 높은 특성을 이용하여 비율을 효율적으로 줄였다. 이러한 방법은 새롭게 사상한 부호벡터 주소가 주소 분할의 여하인 낮은 주소로 사상한 경우에 새롭게 사상한 부호벡터 주소를 부호화했고, 그렇지 않은 경우에는 일반적인 벡터 양자화 방법과 동일하게 재정렬되지 않은 부호화된 부호벡터 주소를 부호화하여, 벡터 양자화 주소의 손실이 없으므로 복원 영상의 화질은 일반적인 벡터 양자화 방법과 동일하다. 또한, 제안한 방법에서는 주소 변환 함수에 의하여 주소 분할값이 하이인 낮은 주소로 사상이 가능한 내 개의 부호벡터 주소들 을 집합하여 계층적으로 부호화함으로써 부호화 효율을 더욱 높였다.

2. 기존의 벡터 양자화 방법

벡터 양자화는 K차원 유클리드 공간 $R^K$로부터 $R^K$의 유한한 부분집합 $Y$로의 사상(mapping) $Q$로 정의한다.

$$Q : R^K \rightarrow Y$$  (1)

여기서 $Y = (y_i : 1 \leq i \leq N)$는 대표벡터의 집합이며, $N$은 대표벡터의 수이다. 벡터 $X$에서 $X$의 사상에 는 여러 가지 알고리즘들이 사용되며, 이는 Obtaining 알고리즘과 같이 주어진다.

$$d(X, \hat{X}) = \sum_{i=1}^{k} (X_i - \hat{X}_i)^2$$  (2)

일반적으로 외영상 $X$와 복원영상 $\hat{X}$에 대한 각각의 화질의 비교는 평균 자수 오차를 이용한 식 (4)와 같은 PSNR가 사용된다.

$$MSE = \frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} (X(i,j) - \hat{X}(i,j))^2$$  (3)

$$PSNR = 10 \log_{10} \left( \frac{255^2}{MSE} \right) [dB]$$  (4)

여기서 $M$은 영상의 행의 크기이며, $N$은 열의 크기이다.

벡터 양자화는 영상 부호화에 이용할 경우 몇 가지 문제점이 발생한다. 그 중 하나는 부호화의 복잡성이 있다. 양자화 벡터화 벡터의 차원을 증가시키면 양자화의 비율을 감소시킬 수 있으므로 성능은 좋아지나 벡터의 차원이 증가함에 따라 부호화의 복잡성은 기하급수적으로 증가하게 된다. 또한, 모든 차원으로서 일반적인 자동 오차를 사용하자면 그 영상에서의 영상의 비율을 계산하지 못하여 재구현 현상이 나타나고, 복잡한 구조적으로 부호화하는 과정에서 생기는 오차로 인해 블록화 현상이 나타나 눈에 띄는 경향이 나타난다. 이러한 현상을 줄이기 위하여 Ramamurti와 Gersho는 영상의 각 벡터를 여러 개의 부류로 분류하는 분류 벡터 양자화(CVQ)를 제안
하였다[7]. CVQ는 일반적으로 희런 영상의 각 불록을
예지블록과 평탄불록으로 분류한다. 그리고 예지블
록에 대해서는 예지의 위치와 방향에 따라 더 세밀하
게 분류한다. 각 불록에 따라 별도의 부호화를 설계
하므로 각 입력블록을 할당된 불록에 해당하는 부호
책만 탐색하여 부호화시 계산량이 감소되고, 예지의
위치와 방향이 보존되므로 시각적으로 우수한 영상
을 얻을 수 있다. 그러나 CVQ 방법에서는 분류기의
설계가 다소 복잡해지는 단점이 있다.

백터 양자화를 영상 부호화에 이용할 경우 나타나
는 또 다른 문제점은 백터 양자화의 불안정성이다.
이 분산성은 부호책 설계시 사용된 영상으로부터
추출된 백터와 통계적 특성이 다른 백터를 부호화할
때 나타난다. 이러한 통계적 특성 중 가장 큰 영향을
미치는 것은 다음과 같이 정의되는 백터의 표본 평균
\( M_k \).

\[
M_k = \frac{1}{k} \sum_{i=1}^{k} X_i
\]

(5)

여기서 \( X_i \)는 입력백터이고 \( k \)는 백터의 차원이다. 이라
한 평균치를 따로 부호화하기 위하여 Baker와 Gray
는 MSVQ(mean shape VQ)를 제안하였다[8]. MSVQ
에서는 먼저 입력백터의 표본 평균을 구하여 입력백
터의 각 화소값들에서 표본 평균을 제거한 잔여 백터
를 백터 양자화하여 전송하고, 표본 평균은 각각의
양자화하여 전송하는 방법이다. 여기서 평균을 전송
하기 위하여 일반적인 예측 부호화를 사용한다. 그러
나 평균의 전송으로 인한 비트율이 증가되는 단점이
있다.

MSVQ의 수정은 입력백터의 각 화소값들로부터
스칼라 양자화된 표본 평균을 제거한 잔여 백터를 백
터 양자화하는 MRVQ(mean residual VQ)가 있다[9].
MRVQ는 표본 평균의 양자화 오차가 잔여 백터에 더
해짐으로써, 불록의 왜곡을 줄이는 장점이 있다.

Hang와 Haskell[9]은 입력블록의 표면(surface)의
근사화에 있어 보간 백터 양자화(interpolative VQ)방
법을 제안하였다. 이 방법은 입력블록여보간된 표
면을 제거한 잔여 백터를 백터 양자화함으로써 시각
적으로 우수한 부호화량을 가졌었다.

일반적인 백터 양자화 방식은 입력백터를 추려진
직사각형에 의해 그 값이 최소가 되는 부호책 내의
부호책과 대응시키는 것이다. 백터 양자화를 영상
부호화에 적용할 경우 일반적인 백터 양자화 방법에
서 사용되는 입력백터는 4x4 크기의 입력불록으로 나
타내고, 부호책의 크기가 128일 때, 입력불록에 부가
되는 비트가 7 비트가 할당되므로 영상영상에 대한
비트율은 0.4375 [bpp]이며, 일반적인 실험 영상에서
PSNR는 28 ~ 30 [dB]로 나타난다.

일반적인 백터 양자화 방법들은 비트율을 감소
하기 위해서는 불록들의 상관성을 이용하거나 입력
블록들의 크기를 크게 하는 것이 필요하다. 비트율을
감소하기 위한 방법으로는 TVQ(transform VQ)[11],
HVV(hierarchical VQ)[12], FSVQ[3], AVQ[5]등이 있다.

TVQ는 입력블록의 크기를 8x8이나 16x16으로하
여 입력블록을 cosine 변환과 같은 선형 변환에 의한
변환 영역으로 나누어 고주파수의 계수들을 제거
함으로써 비트율을 감소하였다. 또한 부호책 영역에서
백터 양자화를 할 경우 불록이 현상을 감소하였다.

HVV는 성체도에 따라서 불록의 크기를 달리하여
부호화하는 방법이다. 이러한 방법은 이미 성체에 있
는 불록은 부호책 크기를 작게하여 부호화하고 평탄
성체에 있는 불록은 부호책 크기를 크게하여 부호화
함으로써 평연량의 화질의 영향이 없이 비트율을
효과적으로 감소시키는 방법이다.

FSVQ는 이미 부호화된 불록들을 이용하여 입력백
터에 대하여 상대 공간을 정의하여 각 상대에 따른
부호화 과정을 형성하는 방법이다. 따라서 각 상대에
따른 부호책 크기를 일정하게 하고 상대 공간을 크게
하면, 계산량을 줄일 수 있으며, 또한 일반적인 백터
양자화 방법보다 전체 대표백터 양이 증가하여 복원
영상의 화질의 양상을 기대할 수 있다. 그러나 FSVQ
의 단점은 다음 상대를 정의하는 방식에 따라 성능이
 좌우되며, 상대를 잘못 정의함으로써 부호화 에의
누적을 초래한다.

AVQ는 백터 양자화된 인접블록들간에 남아 있는
상관성을 이용하여 인접블록들의 부호책 부호책을
주어지며, 가장 적절한 인접블록들의 주요 조합이 지정
되어 있는 주요 부호책을 참조하여 주로 백터를 양자
화하는 효율적인 방법이다. 주로 백터 양자화는 두 개
의 부호책으로 구성되어 있다. 첫 번째 부호책은 수정
된 PMRVQ( predicted mean residual VQ)[5]를 이용
한 부호책이며, 두 번째 부호책은 내 개의 인접블
독특한 보호해의 추수의 조합으로 나타내는 추수 보호해이다. AVQ는 블록의 추수를 각각 보호하기 않고, 인접블록들간에 전이하는 상관성을 이용하여, 주소벡터를 보호하여 일반적인 벡터 양자화 방법보다 두배 정도의 보호효율을 높이는 정점이 있다. 그러나 AVQ는 양자화 과정에서 모든 가능한 추수 조합을 포함하여야 하므로, 주소벡터 보호해의 요소들을 재정렬하기 위하여서는 많은 수의 비교가 이루어져야 하며, 주소벡터 보호해를 저장하는데 필요할 메모리 양이 방대하다는 문제점을 갖고 있다. 또한 양자화된 블록들의 주소벡터가 주소 보호해에 존재하지 않으면 재정렬이 되지 않은 보호해의 주소인 블록들 각각의 추수를 보호해야 하므로 보호효율이 떨어진다. 그러므로, AVQ는 주소벡터를 보호해할 수 있는 활동이 낮은 영상에 대해서는 약간 효율이 떨어진다.

3. 동적 주소 사상을 이용한 벡터 양자화 방법

제한된 동적 주소 사상을 이용한 벡터 양자화 시스템의 보호하기와 복호하기는 (그림 1)로 표현될 수 있다. (그림 1)에서 첫째로 나타낸 주소 변환기와 역 변환기를 제외하면 이는 일반적인 벡터 양자화 시스템에 구조가 된다. (그림 1)의 보호하기에서 $\alpha(\cdot)$는 각 입력벡터 $x_n \in R^{M \times M}$에서 $\beta(\cdot)$는 가장 가까운 벡터를 보호해에 해당하는 입력벡터와 가장 가까운 벡터를 보호해에서 찾아 그 주수 $i_{h, q}$를 출력하는 양자화기이며, 복호하기에서 $\beta(\cdot)$는 전송된 주수 $i_{h, q}$를 받은 벡터 $x_n$로 출력하는 역양자화기이다. 이 때, $M$은 벡터 양자화되는 입력블록의 수평 및 수직 크기이며, $l = \{1, 2, \ldots, N\}$은 주수 집합이며, $N$은 보호해의 크기다. 일반적인 벡터 양자화 시스템에서는 원래의 주수 $i_{h, q}$가 전송되지만, 주소 변환기를 도입하는 경우에는 입력벡터의 양자화 주수 $i_{h, q}$가 아닌 새로운 주수 $i_{h, q}$가 되도록 하는 주소 변환함수 $f(\cdot)$를 이용하여 새로운 주수 $i_{h, q}$가 전송된다.

본 논문에서 제한한 주소 변환기의 목적은 일반적인 벡터 양자화 시스템에서와 같은 해석을 유지하면서 새롭게 사상된 주소의 동적 범위가 감소되도록 원래의 주수를 새로운 주수로 변환하여 비트율을 감소하는데 있다. 이러한 목적이 이루어지기 위하여 주소 변

.hit_image: (a) 부호화기, (b) 복호화기

(Fig. 1) Block diagram of VQ system with dynamic index mapping.

(a) encoder (b) decoder

환기에 요구되는 조건은 다음과 같다.

1) 주소 변환기는 새로운 주수의 확률 분포가 낮은 주수에 집중되도록 해야한다.
2) 주소 변환 함수 $f(\cdot)$는 역함수 $f^{-1}(\cdot)$가 존재할 수 있도록 일일일 할 수 있어야 한다.
3) 변환 방법은 복호화기에서도 알 수 있어야 하므로 이미 보호된 이전 블록들의 정보만을 이용하여야 한다.

위의 조건을 만족하는 주소 변환을 위하여 (그림 2)
에서 나타낸 화소들은로부터 식 (6)과 같은 사이드 매치 오차를 이용하여 주소를 변환시키는 방법을 제안한다.

\[
E_5(Y^p,q) = \left[ \left( \sum_{m=0}^{1} (X_{5m}^{p-1,q} - Y_{0m}^{p,q})^2 + \sum_{m=0}^{1} (X_{5m}^{p-1,q} - Y_{0m}^{p,q})^2 \right)^{\frac{1}{2}} \right]
\)

식 (6)에서 유한 차원의 부호부트너와의 부호밸더이고, \(X^{p-1,q}\)은 이미 부호화된 복토를 나타낸다.

\[
\begin{array}{cccc}
X^{p-1,q} & & & \\
X_0^p & X_1^p & X_2^p & X_3^p \\
X_0^q & X_1^q & X_2^q & X_3^q \\
\end{array}
\]

(그림 2) 사이드 매치
(Fig. 2) Side Match.

일반적인 영상에서는 인접블록들간에 서로 결하고 있는 화소값들에 연속성을 인하여 이미 부호화된 인접블록들의 부호밸더와 부호확 내의 부호밸더들 중 서로 결하고 있는 화소값들간의 오차를 사이드 매치 오차가 작은 부호밸더임수록 현재 부호화한 인
력블록의 부호밸더로 부호화된 확률이 높다. 그러므로 사이드 매치 오차가 작은 순으로 부호확 내의 부호밸더를 재정렬할 경우에는, 부호화할 부호밸더 주소의 확률 분포가 낮은 주소로 집중하므로 앵트로피가 작아지는 결과를 가져올 수 있다. 그러므로 응용 문제에서 사용한 주소 변환 함수 \(f(\cdot)\)는 사이드 매치 오차가 작은 순으
로 부호화를 재정렬하여, 전송한 원래의 주소 \(i_{k,q}\)가 재정렬된 부호확 내의 새로운 주소 \(i_{k,q}'\)로 사상되는

\[
\begin{array}{c}
\lambda \\
0 \\
\vdots \\
1 \\
\vdots \\
\end{array}
\]

\[
\begin{array}{c}
\text{encodable address} \\
\vdots \\
\text{non-encodable address} \\
\vdots \\
\end{array}
\]

(그림 3) 사이드 매치 오차를 이용한 재정렬된 부호확
(Fig. 3) Reordered codebook with side match error.
부호벡터 주소를 부호화하는데 많은 비트 할당이 필요하지만, 새롭게 사용된 부호벡터 주소가 주소 분격값 이하인 주소로 사상 가능한 확률이 높아지므로 재정렬하지 않은 부호책의 원래의 부호벡터 주소를 부호화할 확률이 낮아진다. 반대로, 주소 분격값이 낮은 새롭게 사용된 부호벡터 주소를 부호화하는데 적은 비트 할당이 필요하지만, 새롭게 사용된 부호벡터 주소가 주소 분격값이 이하인 주소로 사상 가능한 확률이 낮아지게 되어 재정렬하지 않은 부호책의 원래의 부호벡터 주소로 부호화할 확률이 높아진다. 그러므로 임력영상은 임력블록으로 분할하였을 때 임력영상에 대하여 주소 분격값 이하인 새로운 주소로 사상 가능한 임력블록의 개수와 새롭게 사용된 주소로 부호화하는데 필요한 비트 할당 수와는 trade-off 관계가 있으며, 주소 분격값의 결정 요인이 된다.

또한, 제안한 방법에서는 재정렬된 부호책에서 주소 분격값 이하인 주소로 묶이는 데 개의 부호벡터들 의 주소를 결합하여 상위블록(higher block)에 대한 부호벡터들의 주소로 부호화하였다. 이러한 방법은 부호책의 재정렬 유, 무를 판별하기 위하여 각 임력블록당 한 비트의 부가비트를, 상위블록당 부가비트로 한 비트를 전송함으로써 부호화 효율을 높일 수 있다. 본 논문에서 제안한 방법에서의 부호화 과정을 정리하면 다음과 같다.

단계 1: 새 개의 임력블록 각각에 대해서 이미 부호화된 임력블록들의 부호벡터들을 이용하여, 사이드메시 오차가 작은 순으로 부호책을 재정렬한다.

단계 2: 현재의 임력블록과 자승 오차가 가장 작은 보호벡터 주소가 재정렬된 부호책에서 정해진 주소 분격값 이하인 경우인가를 판별한다.

단계 3: 새 개의 임력블록 각각에 대하여 새롭게 사상된 부호벡터 주소가 모두 정해진 주소 분격값 이하일 경우에는, 새 개의 임력블록들에 대응되는 부호벡터 주소들이 모두 주소 분격값 이하인 낮은 주소로 변환 가능하다는 부가비트 한 비트와 새개의 임력블록에 대응되는 재정렬된 부호책의 부호벡터 주소를 묶어 비트별로 전송한다. 새 개의 임력블록들 중 새롭게 사상된 부호벡터 주소가 하나라도 정해진 주소 분격값을 초과할 경우에는, 새 개의 임력블록들 모두 부호화될 때 부가비트 한 비트와 각 임력블록당 재정렬 유, 무를 판별하는 비트와 재정렬 유무를 나타내는 재정렬된 부호책의 부호벡터 주소를 비트별로 전송하거나 재정렬되지 않은 부호책의 부호벡터 주소를 비트별로 전송한다.

제안한 방법은 임력블록들간의 사이드 메시 오차가 작은 부호책 내의 부호벡터가 부호화된 확률이 높으므로 새롭게 사상된 부호벡터 주소는 낮은 주소로 분포될 확률이 높은 특성을 이용하여, 전송할 주소를 주소 분격값보다 낮은 주소로 사상하는 새로운 부호벡터 주소범위를 부호화하여 비율을 효율적으로 줄였으며, 새로운 부호벡터 주소가 주소 분격값보다 낮은 주소로 사상되지 않는 경우에는 일반 백터 압축 방법과 동일하게 원래의 부호벡터 주소를 부호화함으로써 일반적인 백터 압축의 방법과 동일한 복원영상의 화질을 갖는다.

그러나 제안한 방법은 일반적인 백터 압축과 방법보다 부호화소스 부호화서 임력영상에서 분할된 한 개의 임력블록에 대하여 부호책의 크기만큼 사이드 메시 오차를 구하여야 하고, 사이드 메시 오차를 이용하여 부호책을 재정렬해야하는 계산이 필요하다.

4. 실험 및 고찰

실험에서는 부호책 내의 부호벡터의 크기를 4x4로 하고 부호책의 크기를 128로 하여, 256의 바이크를 갖는 다섯 장의 512x512 크기의 흰색 영상들로부터 Ramamurthi와 Gersho가 제안한 CVQ 방법을 적용하여 부호책을 설계하였다. 임력영상은 분할한 전체 임력데이터들에 대하여 128 크기의 부호책을 재정렬하는 시간은 IBM PC 펑험 120 기종에서 2초가 소요되었다. 제안한 부호화 방법의 성능을 평가하기 위하여, 비순전 영상인 LENA 영상, BOAT 영상, PEPPER 영상, LADY 영상 등에 실험 영상을 사용하여 임력블록들을 재정렬한 상위블록들의 크기와 재정렬된 부호책에서의 전송 가능한 주소 분격값을 변화시켜 실험하였다. (표 1)과 (표 2)에서는 상위블록의
크기를 각각 8x8, 16x16 크기로 변환하여 주소 분석 값을 변화시켜 가변적으로 실험한 비트를 결과이다. 실험 결과에서 나타난 것과 같이 대부분의 실험영상에서 상위항목의 크기를 8x8 크기보다 크게하여도 비트를 감축의 성능에 큰 효율이 없고, 주소 분석값이 1일 때 가장 높은 비트를 감축 성능을 나타내었으며, 주소 분석값의 변화에 따른 비트의 변동은 미미하였다. 그러므로 본 논문에서는 상위 분석의 크기를 8x8로 재한하고 주소 분석값을 1로 제한하여 실험하였다. 복원된 영상의 화질은 백터 양자화 주소를 무손실 부호화로 전송하므로 복원영상의 화질은 CVQ 방법의 화질과 동일하다. 실험에서 사용한 상위분석의 크기를 8x8로 하고, 주소 분석값을 1로 하였을 경우에 하나의 상위분석에 부가되는 비트 수는 다음 다섯 가지 경우에 대해 각각 다른 비트 수가 부가된다.

CASE 1: 내 개의 입력블록들이 재정렬된 부호화에 대해서 모든 주소 분석값을 초과한 주소로 사상한 경우(1 + 4 + 0x1 + 4x7 = 33 비트)

CASE 2: 한 개의 입력블록들에 의하여 부호화가 이루어질때 주소가 재정렬된 부호화에 대해서 주소 분석값이 하단 주소로 사상 가능한 경우(1 + 4 + 1x1 + 3x7 = 27 비트)

CASE 3: 두 개의 입력블록들에 의하여 부호화가 이루어질때 주소가 재정렬된 부호화에 대해서 주소 분석값이 하단 주소로 사상 가능한 경우(1 + 4 + 2x2 + 1 + 2x7 = 21 비트)

CASE 4: 세 개의 입력블록들에 의하여 부호화가 이루어질때 주소가 재정렬된 부호화에 대해서 주소 분석값이 하단 주소로 사상 가능한 경우(1 + 4 + 3x2 + 1 + 1x7 = 15 비트)

CASE 5: 네 개의 입력블록들에 의하여 부호화가 이루어질때 주소가 재정렬된 부호화에 대해서 모든 주소 분석값이 하단 주소로 사상 가능한 경우(1 + 1x4 + 0x7 = 5 비트)

CASE 1에서 CASE 4까지인 경우에는 네 개의 입력블록들이 하나의 상위블록으로 묶여지지 않는다는 것을 나타내는 1 비트와 네 개의 입력블록당 재정렬의 유무를 판단하는 4 비트와 재정렬된 부호화에서의 변환 주소 부호화할 경우에는 그 입력블록에 변환된 부호화된 주소를 부호화하는데 필요한 1 비트를 할당하고, 그렇지 않은 경우에는 일반 백터 양자화 방법과 동일하게 재정렬되지 않은 부호화의 부호화 주소를 부호화하는데 필요한 7 비트를 할당한다. CASE 5인 경우에는 내 개의 입력블록들이 하나의 상위블록으로 묶여지지 않는 것을 나타내는 1 비트와 각각의 입력블록에 변환된 부호화된 주소를 부호화하는데 필요한 1 비트를 할당함으로써 하나의 상위블록에 총 5 비트가 전송된다. 그러므로 입력 영상에 대하여 비트수는 (7)로 나타난다.

\[
\text{bit rate} = \frac{N_i \times (1 + K \times 4 + N_i \times (1 + 4 + K \times C + M \times (4 - C))}{\text{total pixel number}}
\]

(7)
표 1) 8x8 상위블록에서의 여러 가지 주소 분석값에 대한 비트율

<table>
<thead>
<tr>
<th></th>
<th>LENA</th>
<th>BOAT</th>
<th>PEPPER</th>
<th>LADY</th>
</tr>
</thead>
<tbody>
<tr>
<td>주소</td>
<td>비트율 (bpp)</td>
<td>주소</td>
<td>비트율 (bpp)</td>
<td>주소</td>
</tr>
<tr>
<td>문덕값</td>
<td>1</td>
<td>0.239</td>
<td>1</td>
<td>0.224</td>
</tr>
<tr>
<td>3</td>
<td>0.242</td>
<td>3</td>
<td>0.247</td>
<td>3</td>
</tr>
<tr>
<td>7</td>
<td>0.258</td>
<td>7</td>
<td>0.263</td>
<td>7</td>
</tr>
<tr>
<td>15</td>
<td>0.289</td>
<td>15</td>
<td>0.294</td>
<td>15</td>
</tr>
</tbody>
</table>

표 2) 16x16 상위블록에서의 여러 가지 주소 분석값에 대한 비트율

<table>
<thead>
<tr>
<th></th>
<th>LENA</th>
<th>BOAT</th>
<th>PEPPER</th>
<th>LADY</th>
</tr>
</thead>
<tbody>
<tr>
<td>주소</td>
<td>비트율 (bpp)</td>
<td>주소</td>
<td>비트율 (bpp)</td>
<td>주소</td>
</tr>
<tr>
<td>문덕값</td>
<td>1</td>
<td>0.237</td>
<td>1</td>
<td>0.220</td>
</tr>
<tr>
<td>3</td>
<td>0.239</td>
<td>3</td>
<td>0.244</td>
<td>3</td>
</tr>
<tr>
<td>7</td>
<td>0.254</td>
<td>7</td>
<td>0.260</td>
<td>7</td>
</tr>
<tr>
<td>15</td>
<td>0.282</td>
<td>15</td>
<td>0.289</td>
<td>15</td>
</tr>
</tbody>
</table>

표 3) 8x8 상위블록에서 주소 분석값이 1일 때, 입력 영상에 따른 각 경우에 대한 백분율

<table>
<thead>
<tr>
<th></th>
<th>cases</th>
<th>case 1</th>
<th>case 2</th>
<th>case 3</th>
<th>case 4</th>
<th>case 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>LENA</td>
<td>10.815%</td>
<td>14.282%</td>
<td>15.698%</td>
<td>16.088%</td>
<td>43.117%</td>
<td></td>
</tr>
<tr>
<td>BOAT</td>
<td>10.962%</td>
<td>15.014%</td>
<td>16.187%</td>
<td>14.941%</td>
<td>42.896%</td>
<td></td>
</tr>
<tr>
<td>PEPPER</td>
<td>6.616%</td>
<td>13.354%</td>
<td>15.674%</td>
<td>20.361%</td>
<td>43.993%</td>
<td></td>
</tr>
<tr>
<td>LADY</td>
<td>9.204%</td>
<td>12.744%</td>
<td>12.598%</td>
<td>14.990%</td>
<td>50.464%</td>
<td></td>
</tr>
</tbody>
</table>

(bpp)와 0.217 [bpp]의 비트율을 가정으로써 일반적인 백터 양자화 방법보다 비트율이 약 50% 감소를 가지며, LENA 영상과 BOAT 영상과 같이 영상내에서 급격한 변화가 많은 영상은 각각 0.239 [bpp]와 0.242 [bpp]의 비트율을 가정으 로써 일반적인 백터 양자화 방법보다 비트율이 약 45% 감소를 가지는 것을 실험을 통하여 확인하였다. (그림 4)와 (그림 5)는 각각 영상 내에서 연속성이 강한 영상과 영상 내에서 급격한 변화가 많은 영상에 대한 원영상과 재현한 방법에 의한 복원영상의 나타내었다.

양후 연구 과제로서는 후호화시와 복호화시 한계의 입력블록에 대하여 후호화의 크기 만큼 사이드 메치 오차를 구하여야 하고, 후호화를 재현할수록 재질인 영상의 감소와 후호화된 현재 블록의 보다 정확한 예측 방법의 연구가 필요하다.

(그림 4) 실험 영상의 비교
(a) PEPPER 원영상
(b) PEPPER 복원영상(0.224 [bpp], PSNR = 29.3 [dB])
(c) LADY 원영상
(d) LADY 복원영상(0.217 [bpp], PSNR = 31.0 [dB])

(Fig. 4) Comparison of Test Images.
(a) PEPPER Original Image
(b) PEPPER Reconstructed Image(0.224 [bpp], PSNR = 29.3 [dB])
(c) LADY Original Image
(d) LADY Reconstructed Image(0.217 [bpp], PSNR = 31.0 [dB])
본 논문에서는 무호화 내의 무호벡터들을 재정렬하여, 입력블록의 무호벡터 주소를 새로운 주소로 사상하는 주소 변환 함수를 이용하여 비트율을 효율적으로 감소시킨 벡터 양자화 방법을 제안한다. 제안한 방법에서는 무호화한 입력블록의 인접한 이미 무호화된 입력블록과 무호벡터와의 사이드 메치 오차가 작은 순으로 무호화를 재정렬하고, 재정렬된 무호벡터에서 무호화하 입력블록에 대한 무호벡터 주소를 주소 변환값이 이하인 낮은 주소로 사상될 확률이 높은 특성을 이용하여 비트율을 효율적으로 감소하였다. 또한 재정렬된 무호벡터에서 주소 분석값이 이하인 낮은 주소로의 사상이 불가능한 경우에는, 일반 벡터 양자화의 방법과 동일하게 재정렬되지 않은 무호벡터의 무호벡터 주소를 무호화함으로써 벡터 양자화 주소의 손실이 없으므로 복원영상의 화질은 일반 벡터 양자화 방법과 동일하다. 또한, 제안한 방법에서는 재정렬된 무호벡터에서 주소 분석값이 이하인 낮은 주소로 묶이는 입력블록들의 무호벡터 주소들을 묶어 무호화함으로써 무호화 효율을 더 높였다. 실험 결과로부터 영상 내의 연속성이 높은 영상에서는 비트율의 많은 감소를 나타내었으며 일반 벡터 양자화방법과 비교시 비트율이 약 45～50% 감소함을 확인하였다.

참고 문헌


Park Gil Hoe
1982년 2월: 경북대학교 전자공학과 공학사
1984년 2월: 한국과학기술원 전기 및 전자공학과 공학석사
1990년 2월: 한국과학기술원 전기 및 전자공학과 공학박사
1984년 3월~현재 경북대학교 전자전기공학부 부교수
관심분야: 영상신호처리, 컴퓨터 그래픽스

배 성호
1991년 2월 경북대학교 전자공학과(학사)
1993년 2월 경북대학교 전자공학과(석사)
1993년 3월~현재 경북대학교 전자공학과(박사과정)
관심분야: 영상처리, 컴퓨터 비전

이 대화
1981년 경북대학교 전자공학과 졸업(학사)
1983년 한국과학기술원 전산학과(석사)
1993년 한국과학기술원 전산학과(박사)
1981년~1995년 한국전자통신연구소 시스템소프트웨어 연구실장
관심분야: 병렬/분산 컴퓨터 구조, 병렬운영체제 및 태미디어 서버

1995년~현재 경북대학교 전자전기공학부 교수
관심분야: 병렬/분산 컴퓨터 구조, 병렬운영체제, 멀티디미디어 서버