# 잡음이 존재하는 채널에서 이용되는 분류 벡터 양자화 코드북의 인덱스 할당기법

정회원 한 종 기*, 준회원 김 진 욱*

# Optimization of CVQ codebook index for noisy channels

Jong-Ki Han*, Jin-Ook Kim* Regular Members

요 약

본 논문은 분류 벡터 양자화(CVQ)기법을 이용한 통신 시스템에서 채널 오류를 감소시키기 위한 인덱스벡터 할당 방식을 다루고 있다. 제안된 시스템은 크게 내부 인덱스 할당방식(IIA : inner index assignment)과 교차 인덱스 할당방식(CIA : cross index assignment)으로 구성된다. IIA는 부(Sub)코드북 내에서 유사한 코드벡터들에 Hamming거리가 가까운 인덱스들을 할당함으로써 채널에러에 의해 발생된 화질저하를 감소시킨다. CIA는 인덱스 벡터의 클래스 정보를 나타내는 클래스 비트에 발생하는 채널 오류의 영향을 최소화할 수 있는 방법으로서 IIA에 의해 할당된 인덱스 벡터들을 수정한다.

본 논문에서 실시된 컴퓨터 모의실험은 제안된 시스템이 채널 부호화기법을 사용하지 않고도 채널 잡음을 극복할 수 있음을 보여준다.

ABSTRACT

Abstract In this paper, an improved index assignment procedure is proposed to reduce the channel error effect in a communication system employing classified vector quantization(CVQ). The proposed algorithm consists of two parts: inner index assignment (IIA) and cross index assignment (CIA). The IIA reduces the distortion resulting from the error in order bits, presenting the identity of each code vector in a subcodebook. The CIA modifies the indexes assigned by the IIA in such a way that the effect of the channel error occurring in class bits, indicating the class information of the code vector, can be minimized.

Simulation results show that the proposed algorithms enable a reliable communication over noisy channels even without employing the channel encoding.

Index Terms Classified vector quantization, index assignment.

## 1. 서 론

벡터 양자화(VQ : Vector Quantization)기법은 낮은 비트율에서 높은 화질을 유지할 수 있는 효율적인 영상 데이터 압축 방식이다. 그렇지만 VQ 코드북으로 압축된 영상은 엣지정보가 뭉그러지는 단점을 갖고 있다. 엣지정보는 영상 데이터 인식에서 중요한 부분이기 때문에, 이를 충실히 표현할 수 있는 코드북이 필요하다. VQ에서 엣지가 뭉그러지는 것을 줄이기 위해 Ramamurthi와 Gersho[1]는 분류 벡터 양자화(CVQ : Classified VQ)를 사용하였고, Kim[2]는 이산 코사인 변환(DCT : Discret

* 세종대학교 정보통신공학과 멀티미디어 신호처리 연구실(hjk@sejong.ac.kr),
논문번호 : 020245-0520, 접수일자 : 2002년 5월 24일

e Cosine Transform)을 기반으로 엣지정보를 분류하는 방식을 제안했다.

CVQ를 사용하는 통신시스템에서 인코더를 통해 선택된 코드벡터의 이진 인덱스 벡터는 전송중에 채널 잡음의 영향을 받게 된다. 이진 인덱스벡터는 채널잡음에 의해 손상되고, 손상된 인덱스 벡터는 같은 부코드북, 또는 다른 부코드북의 코드벡터로 복호화된다. 그러므로 복원된 영상 블록은 원래 블록과 다른 여러 가지 성질(평평한곳, 엣지부분 등등)을 갖게 된다. 채널 잡음에 의한 수신영상의 화질은 CVQ 통신시스템의 성능을 평가하는 중요한 요소이다.

본 논문에서는 채널 부호화를 사용하지 않는 시스템에 대해서 논의한다. 에러보정이 없는 상태에서 코드벡터의 인덱스들이 무작위로 할당된다면 잡음이 존재하는 실제 통신 시스템에서 좋은 화질을 기대할 수 없다. 잡음에 의해 변형된 인덱스 벡터는 송신된 코드벡터와 전혀 다른 코드벡터의 인덱스 벡터로 바뀌어 수신될 수 있고, 이는 수신화질을 크게 저하시킬 수 있다. 이러한 VQ의 단점을 보완하기 위한 몇가지 방법들이 연구되었다. Farvardin[4]은 채널 최적화 VQ (COVQ : Channel Optimized VQ)방식을 제안하였고, Zeger와 Gersho[3]는 이진 교환 알고리즘(BSA : Binary Switching Algorithm)이라고 불리는 인덱스벡터 교환기법을 제안하였다. McLaughlin et al.[5]는 균등 확률분포를 갖는 VQ 코드북에 대해서 자연 이진코드가 최적임을 보여주었다. 최근 Chiang과 Potter[6]는 채널코딩에서 일반적인 기준으로 사용하던 평균제곱에러 대신 Minimax 기준을 사용하는 방법을 제안했다. Wang et al.[7]은 VQ로 압축된 영상정보를 순차 전송할 수 있는 코드북을 제안했다.

기존의 인덱스 할당 연구가 단일 코드북의 인덱스 할당에 대한 연구였던 반면, 본 논문에서는 여러 개의 부 코드북으로 구성된 CVQ 코드북의 인덱스를 할당하는 알고리즘을 제안한다. 이 연구를 통해 CVQ 코드북을 사용하는 통신시스템의 채널오류를 최소화 할 수 있는 효율적인 인덱스 할당방식을 제시할 것이다. 이 연구에서 단일 부코드북의 인덱스 벡터는 내부 인덱스 할당기법(IIA)에 의해 할당하고, 이때 할당된 인덱스들은 교차 인덱스 할당기법(CIA)으로 재배치된다. 본 논문에서 제안되는 IIA는 기존의 BSA를 복잡도와 최적화 측면에서 개선시킨 알고리즘으로, 채널 잡음을 제거하는 측면에서 상당한 개선 효과가 있으며, 이를 컴퓨터 시뮬레이션을 통해 보인다.

본 논문은 다음과 같이 구성된다. 2절은 잡음이 존재하는 채널환경에서 CVQ를 사용한 통신시스템에 대해 설명한다. 3절에서는 단일 VQ코드북을 위한 내부 인덱스 할당기법을 제안하고 이를 간단히 정리한다. CVQ 코드북에서 인덱스를 재배치하는 과정은 4절에서 기술한다. 본 논문에서 제안된 시스템의 컴퓨터 실험결과를 5절에서 서술한다. 그리고 6절에서 논문의 결론을 맺는다.

## 2. 시스템 모델

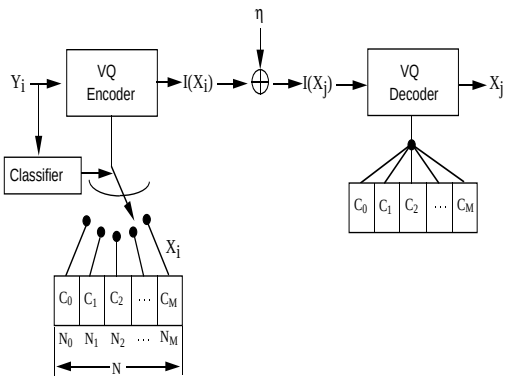


그림1 잡음이 존재하는 채널에서 CVQ기반 통신 시스템

그림1 은 잡음이 존재하는 채널에서 CVQ를 사용한 통신시스템을 나타낸다. K차원의 신호벡터 $Y_i$는 코드벡터 $X_i$로 양자화 되고 이 결과 이진인덱스 벡터 $I(X_i)$가 채널을 통해 전송된다. 코드북은 $C_0, C_1, \cdots, C_M$과 같이 M+1 개의 부(Sub)코드북으로 구성되어 있다. 전체 코드북은 $N(=2^m)$개의 코드벡터로 이루어져 있다. 코드북의 각 코드벡터들은 $m$비트의 이진 인덱스 벡터 $I(X_i)$로 표현된다. 인덱스벡터가 잡음 $\eta$이 존재하는 채널을 통해 전송되고 비트오류 확률이 $P_e$라고 가정하자. 통신채널은 memoryless이고 이진 대칭(BSC) 이다. 이 때 채널 에러에 의해 발생하는 왜곡은 다음 식과 같다.

$$D=\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} d(X_i, X_j)\ P(I(X_j)\,|\,I(X_i)) \qquad (1)$$

여기서 $X_i$와 $X_j$ 사이의 가중치가 가해진 왜곡 $d(X_i, X_j)$는

$$d(X_i, X_j) = P_{X_i} \cdot |X_i - X_j|^2 \quad (2)$$

이다. $P_{X_i}$는 $X_i$가 선택될 선험확률이고 $P(I(X_j)|I(X_i))$는 송신측에서 $I(X_i)$를 전송했을 때 수신 측에서 $I(X_j)$가 수신될 전이확률이다.

비트에러가 발생할 확률 $P_e$ 가 극히 작다고 가정하면 하나의 이진 인덱스 벡터에서 두개 이상의 비트에서 에러가 발생할 확률은 매우 작아 0에 가까운 값이 된다. 그러므로 한 인덱스 벡터에서 한 비트 오류가 발생한 경우만 고려해도 오차는 매우 작다. 이 경우 식(1)의 왜곡은

$$D \cong \sum_{i=0}^{N-1} \sum_{j \in Q_i} d(X_i, X_j)\, P(I(X_j)|I(X_i)) \quad (3)$$

으로 근사화된다. 여기서 $Q_i = \{n : H(I(X_i), I(X_n)) = 1,\ 0 \le n \le N-1\}$이 고 $H(I(X_i), I(H_j))$는 $I(X_i)$와 $I(X_j)$사이의 해밍 거리(Hamming distance)를 나타낸다.

식(3)에서 $P(I(X_j) \mid I(X_i))$가 모든 $i$와 $j \in Q_i$에 대해서 상수 값을 가지므로, 최적의 인덱스 벡터 할당문제는 다음 식을 최소화 하는 인덱스 할당 상태를 구하는 것과 같다.

$$D_a = \sum_{i=0}^{N-1} \sum_{j \in Q_i} d(X_i, X_j) \quad (4)$$

## 3. 내부 인덱스 할당(IIA ; Inner Index Assignment)

이 절에서는 단일 VQ코드북에 대한 인덱스 할당 기법에 대해 논한다. 이 방법은 인덱스 벡터의 순서(Order)비트에 발생한 채널 에러의 영향을 줄일 수 있도록, 각 부코드북 내에서 코드벡터들 간의 거리를 계산한 후, 순서비트에 에러가 나더라도 비슷한 다른 코드벡터로 복원되도록 순서(Order)비트를 할당한다. 각 부(Sub) VQ코드북이 $N_t$개의 코드벡터로 구성되어 있다면 각 코드벡터는 $m_t (= \log_2 N_t)$ 비트의 이진수로 구성된 인덱스 벡터로 표현된다. 식(4)의 $D_a$를 최소화 시키기 위해서는 가장 왜곡

내부 인덱스 할당 (IIA:Inner Index Assignment)

| | |
|---|---|
| 과정1 | 랜덤하게 인덱스 벡터를 할당한다. |
| 과정2 | $\sum_{i \in Q_r^t} X_i$, $\sum_{i \in Q_r^t} P_{X_i} X_i$, $\sum_{i \in Q_r^t} X_i^2$, $\sum_{i \in Q_r^t} P_{X_i}$ 계산. (단 $0 \le r \le N_t - 1$) |
| 과정3 | $A(\alpha, \beta)$값 중 최대값을 갖는 $\alpha^*, \beta^*$를 구한다. $(0 \le \alpha, \beta \le N_t - 1)$ |
| 과정4 | If $A(\alpha^*, \beta^*) > 0$, then 인덱스 $I(X_{\alpha^*})$와 $I(X_{\beta^*})$를 교환 else 과정6으로 이동 |
| 과정5 | If $H(I(X_r), I(X_{\alpha^*})) = k$, $H(I(X_{r)}, I(X_{\beta^*})) = k$, $k = 0, 1$이면 $\sum_{i \in Q_r^t} X_i$, $\sum_{i \in Q_r^t} P_{X_i} X_i$, $\sum_{i \in Q_r^t} X_i^2$, $\sum_{i \in Q_r^t} P_{X_i}$, $0 \le r \le N_t - 1$값을 갱신한다. 그리고 과정3으로 이동한다. |
| 과정6 | 할당과정을 끝낸다. |

그림2 본 논문에서 제안한 VQ 코드북 인덱스의 할당기법

값이 작은 인덱스 할당 상태를 찾아야 하는데, 이는 모든 인덱스 할당 경우를 고려해야 한다. 이 작업은 $N_t!$개의 인덱스 벡터 할당 경우수를 모두 고려해야 되기 때문에, 상당히 복잡도가 높은 작업을 필요로 한다. (예를 들어 $N_t = 32$라면 $N_t! = 32! \simeq 10^{35}$가지 경우를 고려해야 한다.) 따라서 다음과 같은 부최적화(Sub-optimal)과정을 거쳐야 한다. 임의의 코드벡터 $X_\alpha$와 $X_\beta$를 고려하자. 이 두 코드벡터 $X_\alpha$와 $X_\beta$ 에 할당된 인덱스 벡터들을 서로 교환했을 때 $D_a$의 감소량을 표현하면 $A(\alpha, \beta)$와 같다.

$$\begin{aligned} A(\alpha,\beta) = {} & 2\Big(\sum_{i \in Q_\beta^t} X_i - \sum_{i \in Q_\alpha^t} X_i\Big)(P_{X_\alpha} X_\alpha - P_{X_\beta} X_\beta) \\ & + 2\Big(\sum_{i \in Q_\beta^t} P_{X_i} X_i - \sum_{i \in Q_\alpha^t} P_{X_i} X_i\Big)(X_\alpha - X_\beta) \\ & + (P_{X_\alpha} - P_{X_\beta})\Big(\sum_{i \in Q_\alpha^t} X_i^2 - \sum_{i \in Q_\beta^t} X_i^2\Big) \\ & + (X_\alpha^2 - X_\beta^2)\Big(\sum_{i \in Q_\alpha^t} P_{X_i} - \sum_{i \in Q_\beta^t} P_{X_i}\Big) \\ & - B(\alpha, \beta) \end{aligned} \quad (5)$$

여기서

$Q_i^t = \{n : H(I(X_i), I(X_n)) = 1,\ 0 \le n \le N_t - 1\}$ 이고

$$B(\alpha,\beta)=\begin{cases} 2(P_{X_\alpha}+P_{X_\beta})|X_\alpha,X_\beta|^2 \\ \quad ,\ \text{if}\ (I(X_\alpha),I(X_\beta)=1 \\ 0, \quad ,\ otherwise \end{cases} \tag{6}$$

이다. 식(5)에 존재하는 다음 값들은

$$\sum_{i\in Q_r^t} X_i,\quad \sum_{i\in Q_r^t} P_{X_i}X_i,\quad \sum_{i\in Q_r^t} X_i^2,\quad \sum_{i\in Q_r^t} P_{X_i} \tag{7}$$

코드벡터 $X_r$, $r=0,1,2,\cdots,N_t-1$에 대하여 미리 계산한 후 메모리에 저장해 두면, $A(\alpha,\beta)$를 $0\le\alpha,\beta\le N_t-1$에 대해서 간단히 계산할 수 있다. 존재하는 모든 $A(\alpha,\beta)$값들 중 최대값(즉, $D_a$가 가장 많이 줄어드는 인덱스 벡터들의 교환)을 갖는 경우를 아래와 같이 찾는다.

$$A(\alpha^*,\beta^*)=\max_{0\le\alpha,\beta\le N_t-1} A(\alpha,\beta)$$

(8) 여기서 $\alpha^*,\beta^*$는 교환되는 인덱스 벡터가 할당된 코드벡터의 번호이다. 즉 $X_{\alpha^*}$와 $X_{\beta^*}$에 할당되어진 $I(X_{\alpha^*})$와 $I(X_{\beta^*})$를 교환한다. 그리고 $H(I(X_r),I(X_{\alpha^*}))=k$ 또는 $H(I(X_r),I(X_{\beta^*}))=k$, $k=0,1$ ,$0\le r\le N_t-1$을 만족하는 $r$에 대해서 식(7)의 항목들을 갱신해야 된다. 이 갱신된 값을 이용하여 새로운 ( $\hat{\alpha}^*$, $\hat{\beta}^*$ )을 구한다. 이같이 반복되는 과정을 기반으로 내부 인덱스 할당방식이 수행된다. 그림2에 이 과정을 정리해 놓았다. 이 알고리즘은 $D_a$값이 더 이상 감소하지 않을 때까지 반복된다. 그림2의 단계 3~5를 연속적으로 반복하면서 최적화된( $D_a$가 가장 작게 나올 수 있도록) 인덱스를 할당하게 된다.

내부 인덱스 할당방식의 복잡한 과정은 대부분 2~3단계에 있다. 2번 단계에서 식(7)의 항목을 계산하기 위하여 각 코드벡터 $X_r$에 대하여 $m_t$개 [즉 $O(\log N_t)$] 의 코드벡터 $\{X_i, i\in Q_r^t\}$를 고려한다. 이 계산을 모든 코드벡터 $\{X_r, r=0,\cdots,N_t-1\}$에 대해서 수행한다. 따라서 2번 단계에서는 총 $O(N_t\log N_t)$정도의 계산량이 필요하다. 3번 단계에서는 총 $(N_t-1)N_t/2$ 가지 경우 가운데 최대 $A(\alpha,\beta)$를 갖는 스위칭을 선택한다. 따라서 3번 단계에서는 $O(N_t^2)$의 연산작업이 필요하다. 인덱스 $I(X_{\alpha^*})$와 $I(X_{\beta^*})$를 교환함으로써, 식(7)의 항목들이 변하게 되는데, $N_t$개의 코드벡터들 중 단지 $\{X_r|H(I(X_r),I(X_{\alpha^*}))=k, H(I(X_r),I(X_{\beta^*}))=k$ , $k=0,1\}$를 만족하는 코드벡터들에 대해서만 변화한다. 이때, 각 코드벡터 $X_r$에 대해서 식(7)의 항목을 갱신하기 위해서는 $m_t$ (즉 $O(\log N_t)$)개의 코드벡터 $\{X_i, i\in Q_r^t\}$가 고려되어야 한다. 따라서 5번째 단계에서는 총 $O(\log N_t\cdot\log N_t)$의 계산 복잡도가 필요하다. 결국 IIA에서 필요한 총 계산 복잡도는 다음과 같다.

$$O(N_t\log N_t)+\\ \gamma\cdot\{O(N_t^2)+O(\log N_t\cdot\log N_t)\} \tag{9}$$

여기서 $\gamma$은 3~5단계의 반복횟수이다.

## 4. 교차 인덱스 할당(CIA : Cross Index Assignment)

이 절에서는 부(Sub) 코드북들을 합쳐서 하나의 큰 코드북으로 만드는 방법을 서술한다. 코드북을 합치는 이 과정은 반복적으로 수행된다. 전 단계에서 합쳐진 코드북에 다른 합쳐진 코드북을 결합시켜 새로운 더 큰 코드북으로 만든다. 이 논문에서는 간단하게 알고리즘을 설명하기 위해 CVQ 코드북이 5개의 부(Sub)코드북으로 구성되었다고 가정한다(M=4). 부코드북 $C_0$는 평평한 영역의 데이터를 위한 코드북, $C_1, C_2, C_3, C_4$는 엣지정보를 부호화하기 위한 부코드북들로써 각각 $45^\circ$ 사선, $135^\circ$ 사선, 수직, 수평 엣지를 위한 코드북이다. $N_0, N_1, N_2, N_3, N_4$는 각 부코드북 $C_0, C_1, C_2, C_3, C_4$의 크기를 나타낸다. 실제 이미지에서 평평한 부분의 발생 빈도가 가장 크기 때문에

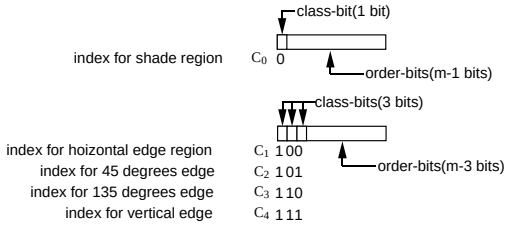


그림3 CVQ 코드북 인덱스 벡터의 구성

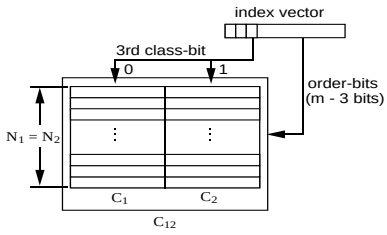


그림4 통합된 부코드북 $C_{12}$의 구조

평평한 부분의 부코드북 크기를 $N_0 = N/2$로 설정하고 나머지 부코드북의 크기는 각각 $N/8$으로 설정한다. 부코드북을 위한 인덱스 벡터의 비트 구성을 살펴보면 그림3과 같다. 각 인덱스 벡터는 분류(Class)비트와 순서(Order)비트로 구성되어 있다. 분류(Class)비트는 어느 부코드북인지에 대한 정보를, 순서(Order)비트는 각 부코드북 내에서 코드벡터의 순서위치를 표시한다. 부 코드북 $C_0$는 분류비트로 엣지의 유무를 표시하는 단 한개의 비트만을 갖고 있다. 반면 다른 부코드북은 엣지임을 표시하는 한 개의 비트와 엣지의 종류를 표시하는 2개의 비트로 구성된다. $N_1(=N/8)+N_2(=N/8)$개의 코드벡터로 이루어진 코드북 $C_{12}$는 그림4에서처럼 부코드북 $C_1$과 $C_2$를 복합해서 만든다. $C_1$과 $C_2$의 인덱스 벡터의 순서(Order)비트는 앞절에서 서술한 IIA방식을 이용해 할당되었다고 가정한다. $C_1$과 $C_2$의 차이는 분류(Class)비트 중 단 1개의 비트만 차이가 발생한다. 이 분류비트에 에러가 발생할 경우, 부코드북의 특정 코드벡터는 다른 부 코드북과 같은 순서(Order)비트를 갖는 코드벡터로 인식된다.

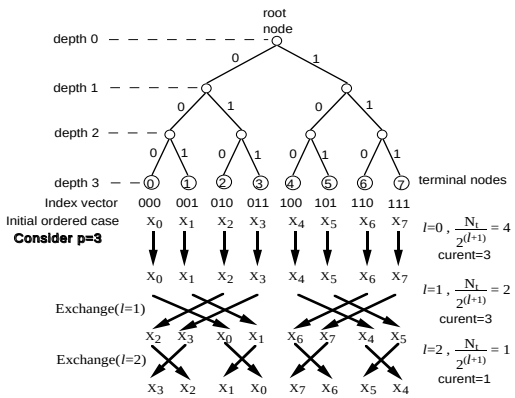


그림5 트리구조로 나타낸 인덱스벡터와 코드벡터의 관계; 정리2의 $N_t=8$ 에서 $X_3$ 의 이동과정

$X_{l0}, X_{l1}, \cdots, X_{iN_t-1}$와 $X_{j0}, X_{j1}, \cdots, X_{jN_t-1}$의 코드벡터로 이루어진 두개의 부코드북 $C_i$과 $C_j$를 고려해 보자. 이때, 분류(Class)비트 에러에 대한 가중 왜곡은 다음과 같이 표현된다.

$$D_{\text{class}}(C_i, C_j, N_t) = \sum_{k=0}^{N_t-1} |X_{ik} - X_{jk}|^2 \cdot (P_{X_{ik}} + P_{X_{jk}}) \quad (10)$$

여기서 $X_{ik}$에 할당된 인덱스 벡터의 순서(Order)비트는 $X_{jk}$의 인덱스벡터의 순서비트와 같다. CIA는 IIA를 통해 할당받은 인덱스 벡터들을 조정하여 $D_{class}(C_i C_j, N_t)$를 최소화시키는 인덱스 벡터 할당상태를 얻는다.

인덱스 벡터와 코드벡터사이의 관계를 표현하기 위해 그림 5에서처럼 깊이(depth)가 $m_t$이고 폭이 $N_t(=2^{m_t})$인 이진 트리구조를 만들었다. 이 그림에서는 $m_t$=3인 경우를 가정하였다. 그림5를 보면 각 단말노드마다 이진 인덱스 벡터가 표시되어 있다. 이는 루트(root)노드에서 각 단말노드까지 경로상에 존재하는 가지(branch)위의 이진수 모음이다. 이 이진 인덱스 벡터는 앞절에서 서술한 IIA로 최적 할당되었다고 가정하고, 인덱스 $i$를 갖는 $X_i$를 $i$번째 단말노드에 위치시킨다. 이 상태를 초기할당 상태라고 하자.

---

정의 1 : $Exchange(l)$은 깊이(depth) $l$에 있는 모든 비단말 노드들의 왼쪽 가지와 오른쪽 가지에 달려있는 모든 단말노드의 코드벡터들을 서로 교환시켜 주는 것이다.

---

정의 1에 대해 좀 더 설명하면 $Exchange(l)$은 모든 코드 벡터의 이진 인덱스벡터의 $(l+1)$번째MSB(most significant bit) 비트를 Complement시키는 것이다.

---

예 1 : 그림5에 $Exchange(l=1)$의 수행과정을 나타냈다. 모든 코드벡터의 인덱스 벡터는 $Exchange(l=1)$에 의해 두번째 $(l+1)$비트가 0에서 1로 또는 1에서 0으로 바뀌었다.

---

정리 1 : $Exchange(l)$과정은 인덱스 벡터들 간

의 해밍거리(Hemming distance)를 변화시키지 않는다.

---

식(4)에서 $D_a$의 값은 $I(X_i)$와$I(X_j)$의 해밍거리(Hermming distance)가 1일 때에만 계산한다. 따라서 $D_a$값은 $Exchange(l)$과정을 거치더라도 변하지 않는다.

---

추론 1 : $D_a$의 값은 $Exchange(l)$과정을 통해 변하지 않는다.

---

정리 2 : $p$번 단말노드에 위치한 특정 코드벡터 $X_p$는 $Exchange(l)$과정을 $W(p)$만큼 반복하면 0번째 단말노드로 옮겨질 수 있다. 여기서 $W(p)$는 $X_p$에 할당된 인덱스 벡터의 해밍중량(Hemming weight)을 뜻한다.

---

정리 2의 증명 : 정수 $I_p$를 다음과 같이 정수의 집합으로 정의하자.

$$I_p=\{k: b_{k+1}(p)=1, 0\le k\le m_t-1\} \quad (11)$$

여기서 $b_k(p)$는 이진수 $p$의 $k$번째 MSB를 의미한다. $I_p$의 원소의 개수는 $W(p)$이다. 코드벡터 $X_p$가 $p$번 단말(terminal)노드에 있다고 하고 $l_1$이 집합 $I_p$의 원소중 하나라고 하자. $Exchange(l_1)$을 실시하면 $X_p$의 새 인덱스는 $p_1(=p-2^{m_t-l_1-1})$이고 코드벡터 $X_p$의 위치는 $p_1$번 단말(terminal)노드가 될 것이다. 이때 $W(p_1)=W(p)-1$이고,

$I_{p_1}=I_p-\{l_1:2^{m_t-l_1-1}=p-p_1\}$이 된다. $I_{p1}$의 한 원소를 선택하여, 예를들면 $l_2$, 이진 코드북 트리에서 $Exchange(l_2)$를 실시해 보자. 그 결과로 $X_p$의 새 인덱스는 $p_2\ (=p_1-2^{m_t-l_2-1})$가 되고, $W(p_2)=W(p)-2$이다. 이 과정을 $W(p)$번 반복하게 되면 코드벡터 $X_p$는 0번 말단(terminal)노드로 옮겨간다. ■

---

예 2 : 정리2의 한 예로써 그림5에서 코드벡터 $X_3$이 0번 단말(terminal)노드로 옮겨지고 있다. 처음에 $I(X_3)$은 [011]이었지만 $Exchange(l=1)$과 $Exchange(l=2)$를 실시함으로써 [000]이 되었다.

---

정리 3 : $X_P$가 0번 단말노드까지 옮겨가는 과정은 $D_a$의 값을 변화시키지 않는다.

---

정리 3의 증명 : 정리 2에 근거하면 $X_P$가 0번 단말노드로 옮겨지는 과정은 $Exchange(l)$을 통해서 이루어진다. 또, 추론 1에 의하면 과정 $Exchange(l)$은 $D_a$값을 변화시키지 않는다. 따러서, $X_P$가 0번 단말노드까지 옮겨가는 과정은 $D_a$의 값을 변화시키지 않는다. ■

---

정리 4 : IIA로 최소화한 $D_a$값이 변하지 않으면서 인덱스를 할당할 수 있는 방법은 $N_t$개의 코드벡터로 이루어진 코드북의 경우, $N_t$가지 경우가 있다.

---

정리 4의 증명 : 정리 2와 3으로부터 정리 4를 유추할 수 있다. 사이즈가 $N_t$인 코드북에는 $N_t$개의

Possible Assignment Search Algorithm (PASA)

| | |
|---|---|
| 과정0 | 단말노드에 IIA로 할당된 코드벡터 $X_i$ $(0\le i\le N_t-1)$를 정렬 |
| 과정1 | $l=0$, $ref=\frac{N_t}{2^{(l+1)}}$ and $current=p$ |
| 과정2 | If $ref\le current$<br>then $Exchange(l)$실행,<br>$current=current-ref$ |
| 과정3 | $l=l+1$, $ref=\frac{N_t}{2^{(l+1)}}$<br>If $l<m_t$, then 과정2로 이동 |
| 과정4 | 코드벡터들의 배열 한 종류를 얻는다. 과정0과 같이 코드벡터 $X_i$를 재정렬시킨다.<br>$p=p+1$ |
| 과정5 | If $p<N_t$, then 과정 1로 이동,<br>else 과정을 끝냄 |

그림6. 식(4)의 $D_a$ 값이 동일한 부코드북 코드벡터의 가능한 모든 배열방법을 찾는 기법

코드벡터가 존재한다. 정리 2에 의해, 이 코드북내

의 각 코드벡터가 0번 단말노드로 이동 배치되는 경우는 총 $N_t$가지가 존재한다. 정리 3에 근거하면, 이 모든 경우는 $D_a$값을 변경시키지 않는다. ■

정의4에서 서술한 가능한 $N_t$가지 경우 각각을 가능한 할당(PA : Passible Assignment)라고 하자. 그림6은 앞의 정리들을 이용하여 가능한 모든 할당 방법을 찾는 알고리즘(PASA : Possible Assignment Search Algorithm)을 나타내고 있다. PASA에서 특정한 코드벡터 $X_p$는 $Exchange(l)$과정을 $W(p)$만큼 반복하여 단말(terminal)노드 $p$에서 0까지 이동한다. Step 0에서는 우선 $N_t$개의 코드벡터를 단말노드에 정렬시키고 $p=0$이라고 초기화시킨다. Step 1에서는 트리에서 $Exchange(l)$이 실시되는 깊이로 사용될 변수 $l$을 만들고 0으로 초기화시킨다. 변수 $current$는 현재 코드벡터 $X_p$가 위치한 단말노드의 번호를 나타낸다. 이는 $p$값으로 초기화된다. Step 2에서는 $Exchange(l)$과정을 실행할 것인지 안할 것인지 판단하기 위해서 $current$를 임계값 $ref(=N_t/2^{l+1})$와 비교한다. $ref$는 깊이(depth)

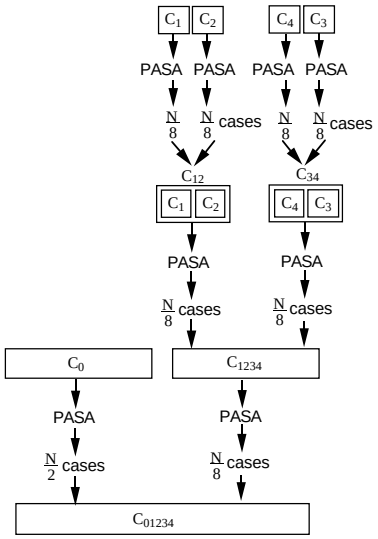


그림7 교차 인덱스 할당기법의 구성도

$l$의 한 노드의 왼쪽 또는 오른쪽 가지에 매달린 단말(terminal)노드들의 갯수이다. Step 3에서는 $l$이 1만큼 증가하고 $ref$값이 갱신된다. $0 \le l \le m_t-1$에 대해서 2~3단계를 실시하면 $X_p$는 0번 말단노드까지 이동할 것이다. Step 4에서는 $N_t$가지의 PA들 중 하나를 구하고, 단말(terminal)노드에는 Step 0에서 초기화된 코드벡터로 할당된다. 그리고 $p$값을 1만큼 증가시킨 후 $X_p$를 고려한다. 이 과정을 $p$가 $N_t-1$이 될 때까지 계속한다. PASA를 사용한 CIA를 그림7에 표시하였다. 각 부코드북 ( $C_0, C_1, C_2, C_3, C_4$)의 인덱스의 순서(Order)비트가 IIA로 할당된 후에 CIA가 적용된다. CIA는 PASA에서 발생한 모든 PA중에서 식(10)의 $D_{class}$를 최소화시키는 한 PA를 선택한다. $N_t=N/8$이고 $m_t=m-3$인 $C_1, C_2$에 PASA를 적용하면 각각 $N_1(=N/8)$, $N_2(=N/8)$개의 PA가 생성된다. $C_1$과 $C_2$를 결합시켜 $C_{12}$를 만들 때 모두 $N/8\times N/8$가지의 경우가 할당가능하다. 이 $N^2/64$경우 중 $D_{class}(C_1, C_2, N/8)$를 최소화시키는 할당 상태를 선택한다. 코드북 $C_{34}$역시 이와 같은 방법으로 $C_3$과 $C_4$을 결합시켜 만든다. 코드북 $C_{1234}$는 $C_{12}$와 $C_{34}$의 결합으로 만든다. 이 과정에서 $N_t=N/4$와 $m_t=m-2$으로 하고, $C_{12}$와 $C_{34}$에 PASA를 적용한다. 그림3과 같이 구성된 분류(Class)비트 구성을 유지시키기 위해서 $C_1$과 $C_2$ (그리고 $C_3$과 $C_4$)사이 코드벡터의 위치가 서로 교환되지 않도록 한다. 따라서 그림6의 과정5에서 $C_{12}$(또는 $C_{34}$)에 PASA가 적용될 때, 과정 5의 $P<N_t$란 조건은 PASA는 $p<N_t/2$로 바뀌어야 된다. PASA를 적용하면 $C_{12}$와$C_{34}$에 대해 각각 $N_1(=H/8)$과 $N_3(=N/8)$개의 PA가 발생한다. $C_{1234}$는 $N_1\times N_3(=N/8\times N/8)$의 가능한 경우 중 $D_{class}(C_{12}, C_{34}, N/4)$를 최소화시키는 경우를 선택한다. 마지막 코드북을 만들때에도 똑같이 위와 같은 방법으로 $C_0$와 $C_{1234}$를 결합시켜 $C_{01234}$을 만든다.

## 5. 실험 및 결과

이 논문에서 제안된 알고리즘은 이동 통신 시스템이나 인공위성을 이용하는 통신 시스템등과 같이 채널 잡음이 존재하는 환경에서 사용되는 VQ기반

저비트율 통신시스템에서 이용될 수 있다. 특히 채널 에러 보정코드를 사용하지 않는 시스템[8]에서 제안하는 알고리즘이 채널에러를 감소시키는 데 유용하게 쓰일 수 있을 것이다. 채널 대역폭이 제한되어 있을 때[9] 데이터의 중요성에 따라 에러보정 비트를 차등하게 사용할 수 있다. 이때 중요한 데이터가 전송될 때에는 채널 코딩 기술을 적용해 주고 중요성이 낮은 데이터의 경우에는 에러에 대한 보정 없이 전송해 주어야 한다. 이 경우 제안하는 알고리즘이 효율적으로 이용될 수 있다.

### A. IIA의 성능

제안 알고리즘의 반복적인 실행과정에서 $D_a$의 감소추이와 IIA의 계산 복잡도를 BSA[3]방식의 항목들과 비교해 보았다. 단일 VQ코드북을 디자인하기 위해서 19개의 서로 다른 512■512의 흑백 이미지들을 이용하였으며, LBG(Linde-Buzo_Gray) 알고리즘[10]으로 제작하였다. 인덱스 벡터들은 무작위로 코드벡터에 할당했으며, 코드북의 크기와 백터의 차원은 각각 $N_t=256$와 $K=16$으로 설정했다. 이 테스트에 사용된 컴퓨터는 DEC-3000이고, 사용된 CPU처리시간(CPU time)은 1/60초 단위로 측정하였다.

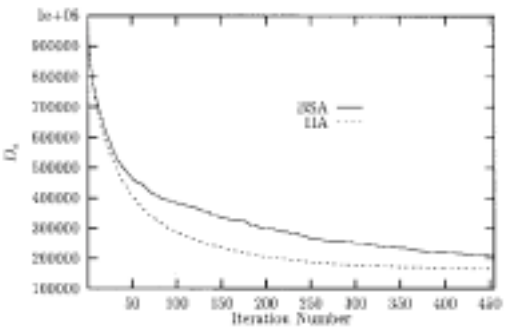


그림8 반복횟수 증가에 따른 BSA와 IIA의 최적화 비교

표1 BSA와 IIA의 CPU 사용시간 비교
[측정단위 1/60 sec]

| Codebook Size / Iteration Num | BSA | IIA |
|---|---|---|
| $N_t=32$ / $\gamma=34$ | 199 | 21 |
| $N_t=64$ / $\gamma=70$ | 1,037 | 166 |
| $N_t=128$ / $\gamma=227$ | 50,302 | 2,153 |
| $N_t=256$ / $\gamma=526$ | 2,557,337 | 24,495 |

그림8은 알고리즘의 반복횟수에 따른 $D_a$값의 변화를 나타낸다. 매 반복 단계에서 IIA의 $D_a$값이 BSA의 것보다 작은 것을 알 수 있다. IIA는 모든 가능한 스위칭의 경우들 가운데 최대의 $A(\alpha,\beta)$값을 발생시키는 경우를 선택하는 반면 BSA는 이런 과정이 없기 때문이다. CPU 사용시간을 비교해 보면 그림8에서 455번 알고리즘을 반복하는 동안 IIA가 20,182(1/60sec)동안 CPU를 사용한 반면, BSA는 1,898,635(1/60sec)동안 CPU를 사용하였다. 이로부터 IIA의 CPU 사용시간이 BSA의 1.06%밖에 안되는 값이고 BSA보다 IIA의 연산량이 훨씬 작다는 것을 알 수 있다.

BSA와 IIA에서의 CPU사용시간을 코드북 크기 $N_t$와 반복회수 $\gamma$값에 따라 측정하여 표1에 나타냈다. 표를 보면 IIA의 연산량이 BSA의 것보다 확실히 작은 것을 알 수 있다.

여러 가지 인덱스 할당 기법들의 채널 에러에 대한 강인성을 그림9에 나타냈다.

그림9는 채널 비트 에러 확률의 다양한 값에 따라 각 인덱스 할당방식의 성능을 표시한 것이다. 이 실험에서는 레나 영상을 사용하였으며, 이 레나영상이 각 방식으로 인덱스 할당된 코드북으로 압축된 후

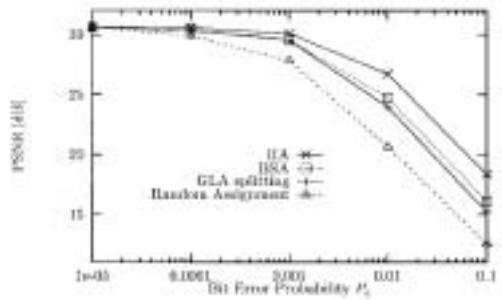


그림9 채널 비트 확률의 다양한 값에 따른 각 인덱스 할당 방식의 성능

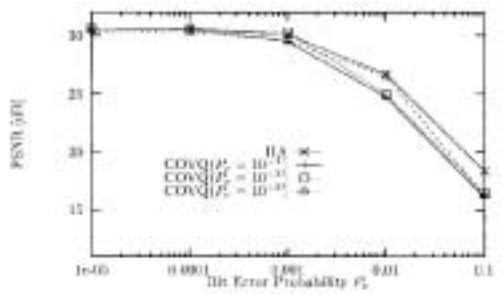


그림10 IIA와 COVQ의 성능비교

잡음 채널을 통과해 수신단에 도착한다고 가정했다. 본 실험에서 가정한 잡음 채널은 AWGN(백색 가산 잡음 : additive white Gaussian noise)이며, 비트에러확률의 범위는 $10^{-5}$~$10^{-1}$로 고려하였다. "BSA"는 IIA의 CPU처리시간과 동일한 시간동안 실행된 BSA 알고리즘의 결과이다. "GLA splitting"은 참고문헌[11]의 Splitting알고리즘을 이용한 결과이다. "Random Assignment"는 인덱스벡터를 무작위로 할당한 VQ 코드북의 성능이다. IIA는 채널비트 에러확률의 전 범위에서 다른 알고리즘보다 성능이 우수함을 보였다. 그리고 비트 에러 확률이 증가할수록 더 좋은 성능을 보였다. "Random Assignment"와 "GLA splitting"은 "IIA"나 "BSA"보다 훨씬 간단한 방식이며 비트에러 가능성이 낮을 때 ($P_e=10^{-5}$)에는 IIA와 거의 비슷한 성능을 보여주고 있다. 반면 비트에러 발생확률이 높을 때 ($P>10^{-4}$)는 IIA가 월등히 높은 성능을 나타낸다. IIA와 COVQ의 성능비교를 그림10에 나타냈다. "COVQ"는 비트 에러 확률이 $P_e=10^{-4}$,

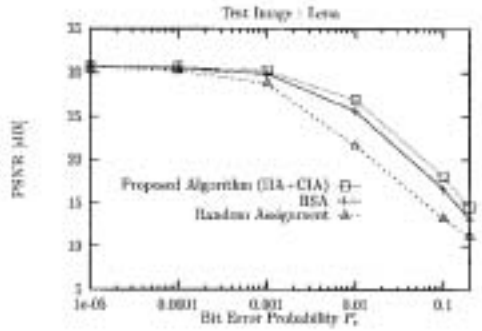


그림11 순서(Order)비트에만 에러가 날 경우의 분류 벡터 양자화 코드북의 성능비교

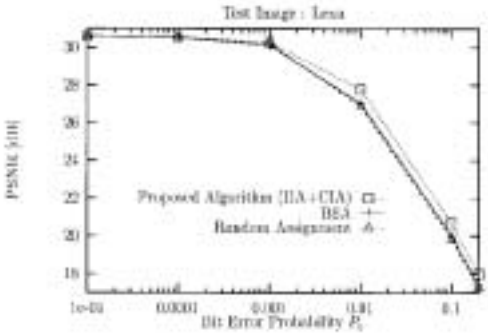


그림12 분류(Class)비트에만 에러가 발생할 경우의 분류 벡터 양자화 코드북의 성능비교

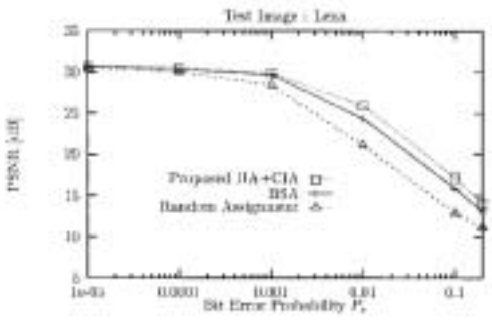


그림13 분류(Class)비트와 순서(Order)비트 모두에서 에러가 발생할 경우 CVQ 코드북의 성능 (Lena영상 실험)

$P_e=10^{-3}$, $P_e=10^{2}$인 경우를 가정하고 최적화된 COVQ 코드북을 의미한다. COVQ는 코드벡터 제작 단계에서 가정한 채널 비트 에러확률과 일치하는 채널환경에서 좋은 성능을 보인다. 그러나 처음에 가정한 확률구간 외 부분에서는 좋지 않은 성능을 나타낸다.

### B. 제안된 알고리즘(IIA와 CIA)의 성능

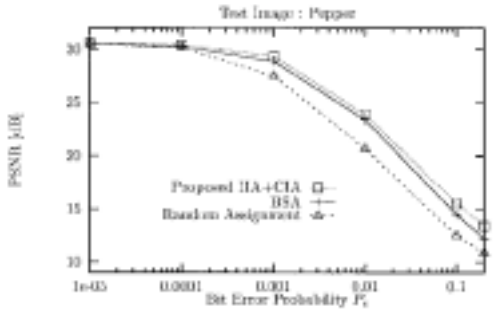


그림14 분류(Class)비트와 순서(Order)비트 모두에서 에러가 발생할 경우 CVQ 코드북의 성능 (Pepper영상 실험)

그림 26 분류벡터 양자화 코드북의 성능비교 이미지

이 절에서는 본 논문에서 제안한 IIA+CIA 알고리즘의 성능을 CVQ 코드북의 인덱스 할당 실험을 통해 분석한다. 이 실험에서 각 부(Sub)코드북은 LBG 알

고리즘을 이용해 제작되어 있다. 영상 블록은 참고문헌[2]에서처럼 DCT공간에서 분류했다. 코드북의 크기는 $N=256$, $N_0=N/2$, $N_1=N_2=N_3=N_4=N/8$으로 설정했다. 사용된 인덱스 할당 알고리즘들의 성능이 그림11~그림14 표현되었다. 이 논문에서 제안한 알고리즘은 IIA로 만들어진 부(Sub)코드북의 인덱스를 CIA로 최적화시킨 것이다. CIA기법은 CPU를 484(1/60sec)동안 사용하였다. 이는 표1에서 표시된 IIA가 사용한 값보다 훨씬 작은 값이다. "BSA"는 IIA의 CPU처리시간과 동일한 시간동안 BSA알고리즘으로 인덱스를 할당한 부코드북으로 실험했다. "Random Assignment"는 인덱스를 무작위로 할당한 CVQ 코드북을 사용했다.

그림11은 순서(Order)비트에만 채널 에러가 발생한다고 가정하고 결과를 측정했다. 이 결과로부터 제안된 알고리즘이 순서비트에서 발생한 채널 에러의 영향을 감소시킬 수 있다. 그림12는 채널 에러가 분류(Class)비트에서 발생한 경우에 대한 실험결과이다. BSA는 인덱스 할당과정에서 분류비트는 고려하지 않기 때문에 무작위로 할당한 방법과 비슷한 성능을 나타낸다. 그림13과 그림14는 분류(Class)비트와 순서(Order)비트 모두에서 채널 에러가 발생한다고 설정하고 Lena영상과 Pepper영상으로 실험한 결과이다. 제안된 알고리즘(IIA+CIA)을 이용한 코드북은 채널 비트에러 확률의 전 범위에서 다른 알고리즘보다 우수한 성능을 내고 있다.

그림 15는 잡음이 존재하는 채널을 통해 전송된 영상의 화질을 비교한 것이다. (a)는 코드북의 인덱스를 최적할당하지 않은 시스템을 이용한 결과이고, (b)는 제안하는 알고리즘을 이용하여 코드북을 할당한 시스템을 이용한 결과이다. 이 결과 영상 살펴보면, 채널 코딩을 사용하지 않고 전송되는 통신 시스템에서, 코드북의 인덱스를 최적화함으로써 채널 왜곡을 감쇄시킬 수 있음을 알 수 있다.

## 6. 결 론

이 논문은 CVQ코드북의 인덱스를 할당하기 위한 IIA와 CIA로 이루어진 알고리즘을 제안하고 있다. IIA는 채널에러의 영향을 줄일 수 있도록 인덱스벡터의 순서(Order)비트를 할당하고, CIA는 IIA에 의해 할당된 인덱스를 최적화시킨다.

컴퓨터 모의실험 결과로부터 제안한 인덱스 할당 방법이 다른 방법들에 비해 우수한 성능을 나타냄을 알 수 있다. 이는 채널 코딩기법을 사용하지 않고도 잡음이 존재하는 채널을 통해 고화질 영상통신이 가능함을 의미한다.

## 부록 $A(\alpha,\beta)$의 유도

식(5)의 $A(\alpha,\beta)$을 유도하려고 한다. 두개의 서로 다른 코드벡터 $X_\alpha$와 $X_\beta$를 고려하자 식(4)의 $D_a$는 다음과 같이 쓸 수 있다.

$$D_a = \sum_{\substack{i=0 \\ i\neq\alpha,\beta}}^{N_t-1} \sum_{\substack{j\in Q_i^t \\ j\neq\alpha,\beta}} d(X_i, X_j) + D'_{X_\alpha} + D'_{X_\beta} + D_{X_\alpha} + D_{X_\beta} \quad \text{(A1)}$$

여기서 $\alpha$와 $\beta$는 $[0, N_t-1]$사이의 정수이고 $Q_i^t=\{n: H(I(X_i), I(X_n))=1,\ 0\le n\le N_t-1\}$ 이다. $Q_i^t$에 속한 원소의 개수는 $m_t$이다.

$$D'_{X_\alpha} = \sum_{i\in Q_\alpha^t} d(X_i, X_\alpha)$$
$$D_{X_\alpha} = \sum_{i\in Q_\alpha^t} d(X_\alpha, X_i) \quad \text{(A2)}$$

$X_\alpha$와 $X_\beta$의 인덱스벡터가 서로 교환된다면, 식(4)의 왜곡은 다음과 같이 표현된다.

$$\tilde{D}_a = \sum_{\substack{i=0 \\ i\neq\alpha,\beta}}^{N_t-1} \sum_{\substack{j\in Q_i^t \\ j\neq\alpha,\beta}} d(X_i, X_j) + \tilde{D}'_{X_\alpha} + \tilde{D}'_{X_\beta} + \tilde{D}_{X_\alpha} + \tilde{D}_{X_\beta} \quad \text{(A3)}$$

각각의 인자들은 다음과 같다.

$$\tilde{D}'_{X_\alpha} = \sum_{i\in Q_\beta^t} d(X_i, X_\alpha) + R(\beta,\alpha)$$
$$\tilde{D}'_{X_\beta} = \sum_{i\in Q_\alpha^t} d(X_i, X_\beta) + R(\alpha,\beta)$$
$$\tilde{D}_{X_\alpha} = \sum_{i\in Q_\beta^t} d(X_\alpha, X_i) + R(\alpha,\beta)$$
$$\tilde{D}_{X_\beta} = \sum_{i\in Q_\alpha^t} d(X_\beta, X_i) + R(\beta,\alpha) \quad \text{(A4)}$$

(A4)식에서 $I(X_\alpha)$와 $I(X_\beta)$의 해밍거리가 1인 경우 $\alpha\in Q_\beta^t$이고 $\beta\in Q_\alpha^t$이므로

$$R(\alpha,\beta) = \begin{cases} d(X_\alpha, X_\beta) \\ \quad , \text{if } H(I(X_\alpha), I(X_\beta))=1 \\ 0 \qquad\qquad , otherwise. \end{cases} \quad \text{(A5)}$$

이다. $X_\alpha$와 $X_\beta$의 인덱스 벡터들을 서로 교환함으로써 변한 $D_a$값의 변화량을 표현하면

$$A(\alpha,\beta)=D_a-\widetilde{D}_a \tag{A6}$$

이다. (A6)식에 (A1)과 (A3)식을 대입하고 이를 간단하게 정리하면 다음과 같이 표현할 수 있다.

$$\begin{aligned}A(\alpha,\beta)=&\\ &2\Big(\sum_{i\in Q_\beta^t}X_i-\sum_{i\in Q_\alpha^t}X_i\Big)(P_{X_\alpha}X_\alpha-P_{X_\beta}X_\beta)\\ &+2\Big(\sum_{i\in Q_\beta^t}P_{X_i}X_i-\sum_{i\in Q_\alpha^t}P_{X_i}X_i\Big)(X_\alpha-X_\beta)\\ &+(P_{X_\alpha}-P_{X_\beta})\Big(\sum_{i\in Q_\alpha^t}X_i^2-\sum_{i\in Q_\beta^t}X_i^2\Big)\\ &+(X_\alpha^2-X_\beta^2)\Big(\sum_{i\in Q_\alpha^t}P_{X_i}-\sum_{i\in Q_\beta^t}P_{X_i}\Big)-B(\alpha,\beta)\end{aligned} \tag{A7}$$

*where*

$$B(\alpha,\beta)=\begin{cases}2(P_{X_\alpha}+P_{X_\beta})|X_\alpha,X_\beta|^2 & \\ \quad,\ \text{if}\ (I(X_\alpha),I(X_\beta)=1 & \\ 0, \quad\quad,\ otherwise & \end{cases} \tag{A8}$$

## 참고문헌


[1] B. Ramamurthi and A. Gersho, "Classified vector quantization of images," *IEEE Trans. Commun.*, vol. C-34, pp. 1105-1115, Nov. 1986.

[2] D. Kim and S. Lee, "Image vector quantizer based on a classification in the DCT domain," *IEEE Trans. Commun.*, vol. 39, pp. 549-556, Apr. 1991.

[3] K. Zeger and A. Gersho, "Pseudo-gray coding," *IEEE Trans. Commun.*, vol. 38, pp. 2147-2158, Dec. 1990.

[4] N. Farvardin and V. Vaishampayan, "On the performance and complexity of channel-optimized vector quantizers," *IEEE Trans. Inform. Theory*, vol. 37, pp. 155-160, Jan. 1991.

[5] S. W. McLaughlin, D. L. Neuhoff, and J. J. Ashley, "Optimal binary index assignment for a class of equiprobable scalar and vector quantizers," *IEEE Trans. Inform. Theory*, vol. 40, pp. 2031-2037, Nov. 1995.

[6] D.-M. Chiang and L. C. Potter, "Minimax nonredundant channel coding for vector quantization," in Proc. *IEEE Int. Conf. Acoustics, Speech, and Signal Processing, Minneapolis*, MN, Apr. 1993, vol. V, pp. 617-620.

[7] R.-Y. Wang, E. A. Riskin, and R. Ladner, "Codebook organization to enhance maximum a posteriori detection of progressive transmission of vector quantized images over noisy channels," *IEEE Trans. Image Processing*, vol. 5, pp. 37-48, Jan. 1996.

[8] J. H. Chen, G. Davidson, A. Gersho, and K. Zeger, "Speech coding for the mobile satellite experiment," in Proc. *IEEE Int. Conf. Commun., Seattle*, WA, June 1987, pp. 756-763.

[9] W. S. Lee, M. R. Pickering, M. R. Frater, and J. F. Arnold, "Error resilience in video and multiplexing layers for very low bit-rate video coding systems," in *IEEE J. Selected Areas Commun.*, vol. 15, pp. 1764-1774, Dec. 1997.

[10] A. Gersho and R. M. Gray, Vector Quantization and Signal Compression. Boston, MA: Kluwer Academic, 1992.

[11] K.-P. Ho and J. M. Kahn, "Transmission of analog signals using multicarrier modulation: A combined source-channel coding approach," *IEEE Trans. Commun.*, vol. 44, pp. 1432-1443, Nov. 1996.

한 종 기(Jong-Ki Han) 정회원

1992년 2월 : KAIST 전기 및 전자공학과 학사
1994년 2월 : KAIST 전기 및 전자공학과 석사
1999년 2월 : KAIST 전기 및 전자공학과 박사
1999년 3월～2001년 8월 : 삼성 전자 디지털 미디어 연구소 책임 연구원
2001년 9월～현재 : 세종대학교 정보통신공학과 조교수

<주관심분야> 음성/영상 처리 및 압축, 디지털 신호처리, 디지털 통신 시스템

김 진 욱(Jin-Ook Kim) 준회원

2003년 2월 : 세종대학교 정보통신공학과 학사
2003년 3월 ~ 현재 : 세종대학교 정보통신공학과 석사 과정

<주관심분야> 영상처리 및 압축