

# 윤곽선 분류 유한상태 벡터 양자화를 이용한 영상 시퀀스 부호화

정회원 김응성\*, 이근영\*

## An Image Sequence Coding Using Edge Classified Finite State Vector Quantization

Eung-Sung Kim\*, Keun-Young Lee\* *Regular Members*

### 요 약

본 논문에서는 기존 Side-match 유한상태 벡터 양자화기의 성능을 개선하기 위하여 윤곽선에 근거한 유한상태 벡터 양자화기를 제안하였다. 각 입력 블록들을 DCT 변환하여 윤곽선 유형에 따라 이들을 17가지 종류의 클래스로 분류하고, 각 클래스는 각기 특성에 따라 만들어진 다른 부호책을 갖게 한다. 인코더에서는 각 블록을 움직임 또는 정지 블록으로 분류하고 전송 비트 수를 줄이기 위해 윤곽선과 움직임 정보를 모두 포함하고 있는 병합 맵을 작성하여 부호화기로 전송한다. 부호화 방법에 여러 모드를 두고 전송 채널의 대역폭에 따라 선택하여 전송 비트율을 조절하였다. 제안 알고리즘으로 시뮬레이션한 결과 기존 벡터 양자화 알고리즘에 비해 좋은 복원 영상의 화질을 얻었으며, 저 전송률 비디오 코덱 표준안인 H.263, H.261과의 비교에서도 제안 알고리즘이 좋은 성능을 나타내었다.

### ABSTRACT

In this paper, we propose a new edge based finite state vector quantization method having better performance than conventional side-match finite state vector quantization. In our proposed scheme, each DCT transformed block is classified to 17 classes according to edge types. Each class has a different codebook based on its characteristics. Encoder classified each block to motion block or stationary block and constructed a merging map by using edge and motion information, and sent to decoder. We controlled amount of being bits transmitted with selecting modes according to bandwidth of transmitting channel. Compared with conventional algorithms, H.263 and H.261 at low bit rate, our proposed algorithm shows better picture quality and good performance.

### 1. 서 론

문자, 그래픽, 오디오, 영상 및 비디오 등의 총합체로서 구성되는 멀티미디어 데이터 중에서 영상 데이터는 그 의미 전달 효과가 매우 크기 때문에 멀티미디어 데이터의 핵심이라 할 수 있다. 그러나 한편으로 영상 데이터는 단위 시간당 발생 또는 재생되는 데이

\* 성균관대학교 전자공학과  
論文番號 : 97412-1107  
接受日字 : 1997年 11月 7日

터 량이 방대하여 일반적으로 압축 과정 없이는 저장, 전송이 어렵기 때문에 많은 영상 데이터의 압축 알고리즘들이 연구되고 있다[1-3]. 벡터 양자화(Vector Quantization : VQ)는 음성이나 영상 압축에 사용되는데, LBG(Linde Buzo Gray) 알고리즘[4]과 같은 실제적인 벡터 양자화기 설계 알고리즘이 개발된 후 널리 사용되어 특정 응용분야 뿐만 아니라 일반적인 벡터 양자화에 관한 여러 논문들이 발표되었다[5-6].

벡터 양자화의 관건은 벡터를 나타내는 좋은 부호책을 설계하는 것인데, 부호책의 설계가 적절하지 못하면 각 입력 벡터에 대해 부호책으로부터 탐색한 부호단어(codeword)가 입력 벡터를 대표하지 못하는 경우가 발생하여 복원 영상의 화질을 열화시킨다. 벡터 양자화기의 부호책을 설계하는 한 예로써 영상의 시각적인 특징(perceptual feature)을 유지하는 분류 벡터 양자화(Classified VQ : CVQ)가 Ramamurthi와 Gersho에 의해 제안되었고[7], 또한 C. H. Hsieh는 DCT를 이용하여 트리 구조 벡터 양자화(Tree Structure VQ : TSVQ)를 수행하는 방법을 제안하였는데[8], 부호책을 만들기 위해 2진 트리 분류기를 사용하여 각 입력 벡터에 대해 DCT를 수행한다.

동영상 시퀀스는 프레임간에 또는 장면간에 주로 변화를 일으킨다. 각 프레임을 정지 영상 부호화와 같은 방법으로 벡터 양자화 하여 전송 시에는 전송 비트 수가 많아진다. 일반적인 벡터 양자화는 시간에 대하여 영상의 화질을 일정하게 유지시킬 수 없고, 또 이웃하는 화소 사이에 높은 상관관계를 이용하지만 이웃 블럭간의 상관관계는 무시한다. 이에 대한 대안으로서 일반적인 벡터 양자화기로 가능한 것 이하로 비트율을 감소시키기 위해 블럭간 상관관계를 이용하는 유한상태 벡터 양자화기(Finite State VQ : FSVQ)를 사용한다[9-10]. 유한상태 벡터 양자화기는 일종의 기억장치가 있는 벡터 양자화기의 유형으로 생각할 수 있는데 일반적인 벡터 양자화의 유한한 집합, 즉 부호화기 상태에 따라서 자신의 부호책을 가지는 벡터 양자화기이며, 큰 벡터 양자화 부호책의 영상 화질을 유지할 수 있고, 작은 벡터 양자화 부호책의 사용으로 인한 낮은 비트율 효과를 얻을 수 있다.

Chang과 Chen은 영상을 윤곽선에 따라 분류하고 영상 데이터의 전송 비트 수를 줄이기 위한 유한상태 벡터 양자화기(Quad Tree SMVQ : QTSMVQ)를 제안하였다[11]. 이 알고리즘에서는 영상의 움직임 정보를

쿼드 트리로 작성하고 각각에 유한상태 벡터 양자화를 수행하는데, 각 블럭의 윤곽선 정보 부호화 시에는 전송 인덱스를 현재 블럭에서 결정하나 부호화 시에는 Side-match 방법[12]을 사용하여 좌측, 상위 블럭에서 윤곽선 유형을 예측한다. 그러므로 부호화기와 복호화기의 윤곽선 결정 방법 차이로 인한 현재 블럭의 윤곽선 정보에서 오류를 발생시킨다.

본 논문에서는 이웃 벡터들 사이에 상관관계가 높지 않은 경우, Side-match 유한상태 벡터 양자화기는 입력 벡터에 가깝지 않은 부호단어를 선택하고 양자화 오류를 증가시킬 수 있으므로 Side-match 유한상태 벡터 양자화기의 문제점을 개선하기 위하여 분류벡터 양자화기를 이용하고, 복원 영상의 화질을 높이기 위해 차 벡터 양자화를 수행하는 윤곽선 분류 유한상태 차 벡터 양자화기(Edge Classified Finite State Difference Vector Quantization : ECFSDVQ)를 제안한다.

본 논문의 구성은 2장에서 유한상태 벡터 양자화의 일종인 Side-match 벡터 양자화에 대하여 기술하였고, 3장에서는 본 논문의 제안 알고리즘 기법을 기술하였으며, 4장에서 제안 알고리즘의 실험결과와 기존 벡터 양자화기, H.263, H.261과 비교 분석하였고, 5장에서 결론을 기술하였다.

## II. Side-match 벡터 양자화

유한상태 벡터 양자화기(FSVQ)는 일반적인 벡터 양자화기와는 달리 영상에서 이웃 블럭간의 높은 상관관계를 이용하여 이전 블럭의 상태를 기억함으로써 부호화를 수행하여 전송 비트율을 감소시킨다[10-13].

FSVQ에서는 주 부호책(master codebook)으로부터 생성되는 상태 부호책(state codebook)을 가지고 있다. 즉, 입력 벡터에 대해 부호단어 인덱스 탐색을 주 부호책이 아닌 상태 부호책에서 하게 된다. 상태 부호책의 설계에는 Side-match 벡터 양자화(SMVQ) 알고리즘[12]을 이용하는데, SMVQ는 FSVQ의 한 종류로서 재생성된 벡터들 사이의 경계 밀도(gray level) 변화에 의해 양자화를 수행한다. 하나의 주 부호책으로부터 여러 개의 상태 부호책을 작성하는데, 각각의 상태 부호책 크기는 같다. 예를 들어, 크기가  $N_m$ 인 주 부호책을 작성했다면, 주 부호책으로부터 각각 크기가  $N_s$ 개인 상태 부호책  $N_m \times N_s$ 개를 작성하게 된다.

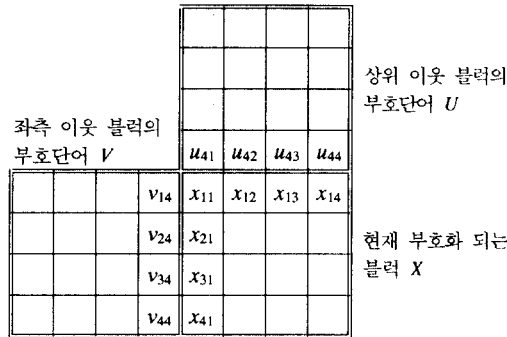


그림 1. SMVQ의 대표 상태 벡터 (블록크기 4×4)  
Fig. 1 Representative state vector of SMVQ (4×4)

상태 부호화의 설계는 SMVQ의 알고리즘에서 블록간의 경계 정보를 이용해야 하므로, 주 부호책으로부터 상태 부호책의 작성은 그와 같은 경계 정보를 이미 포함하고 있어야 한다.

각각의 상태 부호책에 대한 대표 상태 벡터는 바로 이와 같은 경계 정보, 즉 그림 1에서  $x_{11}, x_{21}, x_{31}, x_{41}$ 과  $x_{11}, x_{12}, x_{13}, x_{14}$ 의 8개 화소 값이 현재 블록의 대표 상태 벡터가 된다. 주 부호책으로부터 상태 부호책의 선택은 주 부호책의 두 부호단어  $x$ 와  $y$ 에 대하여 Side-match 왜곡(distortion)을 계산한다. 즉, 수평 방향의 Side-match 왜곡,

$$hd(x, y) = \sum_{i=1}^4 (x_{i1} - y_{i1})^2 \quad (1)$$

과 수직 방향의 Side-match 왜곡,

$$vd(x, y) = \sum_{j=1}^4 (x_{1j} - y_{1j})^2 \quad (2)$$

을 계산하고, 이에 대한 전체 Side-match 왜곡

$$smd(x, y) = hd(x, y) + vd(x, y) \quad (3)$$

을 계산하여 각  $x$ 에 대해 Side-match 왜곡이 가장 작은  $N_j$ 개의  $y$  부호단어를 선택하여 작성한다.

상태 부호책이 선택된 후에는 그 상태 부호책 내의 모든 부호단어  $Y$ 와 현재 입력 벡터  $X$ 와의 제곱 오차(식 4)를 계산하여 그 값이 가장 작은 부호단어의 인

덱스를 부호화 하여 복호화기로 전송한다.

$$d(X, Y) = \|X - Y\|^2 = \sum_{j=0}^n (X_j - Y_j)^2 \quad (4)$$

### III. 제안한 윤곽선 분류 유한상태 차 벡터 양자화기

FSVQ를 이용하여 동영상상을 부호화 하는 방법이 Chang과 Chen에 의해서 제안되었다[11]. Chang과 Chen의 동영상 부호화 알고리즘은 영상의 움직임 정보를 쿼드 트리로 작성하고 각각에 FSVQ를 수행하는 방법(QTSMVQ)으로 입력 영상의 각 블록을 윤곽선 유형에 따라 분류한다. 분류된 영상의 윤곽선에 대해서 각각 구별된 주 부호책을 가지고 있고, 그 주 부호책으로부터 상태 부호책을 설계한 다음, 각 블록에 대한 전송 인덱스를 상태 부호책에서 찾아 부호화 한다. 부호화 시에 각 블록에 대한 움직임 정보를 차 쿼드 트리 맵(difference quadtree map)에 저장하고 맵 정보를 복호화기로 전송한다. 그리고 각 블록에 SMVQ를 수행하여 상태 부호책의 인덱스를 복호화기로 전송한다.

QTSMVQ에서 쿼드 트리 맵은 한 프레임의 움직임 정보만을 가지고 있다. 이 경우 전송 비트 수는 크게 줄지만, QTSMVQ에서 사용하는 쿼드 트리 맵은 입력 영상의 가로, 세로 크기가 동일한 경우에만 사용할 수 있다. 즉, QCIF 영상(176×144)과 같이 가로, 세로의 크기가 다른 영상에서는 입력 영상의 가운데를 중심으로 정방형의 영상에 대해서는 쿼드 트리를 적용하고 그 이외 지역의 블록에 대해서는 움직임 정보를 따로 보내줘야 한다. 그리고 각 블록의 윤곽선 정보를 부호화 시에는 현재 블록에서 결정하나 복호화 시에는 Side-match 방법을 사용하여 좌측, 상위 블록에서 윤곽선 유형을 예측한다. 이는 QTSMVQ에서 사용하는 쿼드 트리가 윤곽선이 아닌 움직임 정보만을 가지고 있기 때문이다. 그러므로 그만큼 현재 블록에 대한 윤곽선 유형에서 오류가 발생할 수 있다. 또한 QTSMVQ에서는 윤곽선 블록보다는 배경 블록을 먼저 부호화 하여 전송하지만, 이 경우 먼저 부호화 되는 배경 블록들에 대해서는 복호화기에서 Side-match를 사용할 수 없어 결국 주 부호책을 사용하게 되어 전송 비트 수가 많아진다.

본 논문에서는 이와 같은 단점을 극복하고, 영상 시퀀스 부호화 시에 전송 비트 수를 줄일 수 있는 알고리즘을 제안한다.

### 3.1 윤곽선 맵 작성

본 논문에서는 인간의 시각 특성에 민감한 영상의 윤곽선 정보를 유지하기 위하여 입력 영상의 윤곽선 정보를 이용하여 각 블록을 배경 부분과 16개의 윤곽선 부분으로, 총 17가지로 분류하여 주 부호화를 각 분류 클래스에 따라 따로 작성한다. 각 프레임의 영상에 대한 윤곽선 검출은 Sobel 연산자를 이용하여 윤곽선을 검출한다. 윤곽선 검출이 끝난 후, 입력 영상을 부 블록으로 나누어 각 블록에 대해 윤곽선이 존재하는지 여부를 결정한다. 입력 영상의 윤곽선 검출 결과에 대해 윤곽선 영상을  $N \times N$  블록으로 나누어 그  $N \times N$  블록내의  $N^2$ 개 윤곽선 화소 값에 대하여 문턱치(threshold)를 주어  $N^2$ 개의 화소 값중에 하나라도 문턱치보다 큰 값을 가진 화소가 존재하면 그 블록은 윤곽선 블록으로 간주한다. 윤곽선 블록에 따라서 윤곽선 정보를 가지고 있는 윤곽선 맵(edge map)을 작성하는데, 윤곽선 맵의 하나의 위치는 입력 영상의 한 블록  $N \times N$ 에 해당한다.

### 3.2 윤곽선 블록의 16 클래스 분류

배경 블록에 비해 윤곽선 블록들은 영상의 중요한 요소인 윤곽선 정보를 가지고 있기 때문에 모든 윤곽선 블록들을 그대로 하나의 주 부호화 설계에 사용하면 여러 가지의 윤곽선 정보가 부호화에 반영되지 않는다. 그러므로 배경 블록에 대해서는 그 블록들을 모아 바로 부호화기에 입력 벡터로 들어가고 윤곽선 맵에 따른 윤곽선 블록들은 다시 16개의 부 클래스(sub class)로 세분화한다. 본 논문에서는 입력 벡터의 분류에 영상의 시각정보를 잘 유지하는 DCT(Discrete Cosine Transform)를 사용하였다.

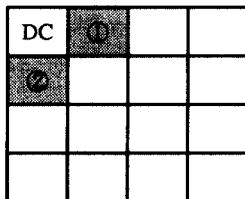


그림 2. 윤곽선의 16 클래스 분류를 위한 DCT 계수값 선택  
Fig. 2 DCT coefficients selection for 16 classes edge classification

윤곽선 맵을 검색하여 윤곽선 블록인 경우, 원 영상 화소 값에 대해 DCT 계수 값을 계산하고, 그림 2와 같이 ①, ②번의 AC 계수만을 취하여 윤곽선의 종류를 판별한다[14]. 이와 같이 두 AC 값을 그림 3과 같은  $x, y$  좌표축으로 맵핑시켜 현재 블록이 각 16개의 클래스 중에 어느 영역에 속하는지를 판단하여 윤곽선의 종류를 분류한다. 그림 3에서 좌표축의 원점, 즉 두 AC 계수 값이 모두 0인 윤곽선 블록은 배경 블록에 가깝기 때문에 윤곽선 맵의 정보를 배경 블록으로 판단한다. 이는 윤곽선 블록을 결정하는 과정에서  $4 \times 4$  블록의 한 화소만이라도 문턱치보다 크면 윤곽선 블록으로 결정하는데 있어서 Sobel 마스크 연산 결과 발생하는 문제점을 제거하기 위함이다.

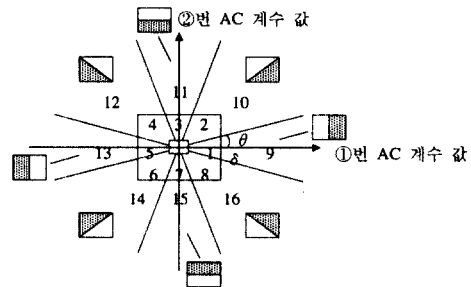


그림 3. AC 계수 값에 의한 윤곽선의 형태  
Fig. 3 Edge types of AC coefficients value

그림 3에는 16개의 영역이 존재하는데 그림 2의 두 AC 계수 값에 의해 현재 윤곽선 블록의 위치를 판단하여 16 클래스중 하나로 결정한다. 그림에서와 같이 각 영역은 윤곽선의 성질을 반영하고 있으며, 그림의 원점 근처 작은 사각형은 원래는 배경 블록이나 Sobel 연산 후에 윤곽선 블록으로 결정된 블록에 대해 두 AC 계수 값들이 0에 근접하는 성질을 이용하여 다시 배경 블록으로 수정하기 위해 설정해 놓은 것이다. 그림내의 바깥쪽 사각형을 기준으로 바깥쪽 영역은 윤곽선 블록내의 밝기 차이가 상당히 큰 부분이고 안쪽 영역은 비교적 윤곽선의 밝기 차이가 덜한 부분이다. 이와 같이 윤곽선 블록들에 대해 윤곽선 유형에 따라 16개의 클래스중 하나로 분류하고, 윤곽선 맵에 그 클래스 종류를 저장한다.

### 3.3 움직임 맵 작성

본 논문에서는 윤곽선 분류 유한상태 양자화를 이용하여 영상 시퀀스 부호화를 수행하는데, 프레임에서 각 블럭을 움직임 블럭(motion block)과 정지 블럭(stationary block)으로 분류하여 움직임 맵(motion map)을 작성하고 단지 움직임 블럭만 부호화 하여 전송한다.

영상 시퀀스의 첫 프레임은 인트라(Intra) 프레임으로서 움직임 정보를 가지고 있지 않다. 그리고 첫 프레임은 차후의 예측(Predictive) 프레임들에 대해 기준이 되는 프레임이기 때문에 부호화 시에 가능한 한 비트를 많이 할당하여 복원 화질을 좋게 한다. 부호화 단계에서는 원 영상의 참조가 가능하지만, 복호화 단계에서는 오로지 전송되어 복호화된 영상만을 참조해야 하므로, 첫 프레임의 부호화는 매우 중요하다. 그러므로 본 논문에서는 첫 프레임과 차후에 나타날 수 있는 다른 인트라 프레임에 대해서는 상태 부호책이 아닌 주 부호책으로부터 벡터 양자화를 수행한다. 이렇게 함으로써 상태 부호책을 사용하여 부호화 수행 시에 발생하는 오류를 방지할 수 있다. 영상 시퀀스 중 인트라 프레임의 차후 예측 프레임에 대해서는 각 프레임의 움직임 정보를 유지하고 있는 움직임 맵을 작성한다.

현재 프레임  $i$ 에서 블럭  $b_i$ 가 움직임 블럭인지에 대한 결정은 블럭  $b_{i-1}$ 을 이전 프레임  $i-1$ 에서  $b_i$ 와 같은 위치에 있는 블럭이라고 하면, 먼저 블럭  $b_i$ 의 윤곽선 유형을 결정하여 윤곽선 맵을 작성한다.  $b_i$ 와  $b_{i-1}$ 의 윤곽선 유형이 다르면  $b_i$ 는 움직임 블럭으로 결정하고, 유형이 다르면 식 (4)를 이용하여  $b_i$ 와  $b_{i-1}$ 의 유사 정도를 계산한다.  $b_i$ 가 윤곽선 블럭일 경우, 식 (4)의 연산 결과가  $TH_{mc}$ 보다 큰 경우 움직임 블럭으로 결정하고,  $b_i$ 가 배경 블럭일 경우, 식 (4)의 연산 결과가  $TH_{mb}$ 보다 크면 움직임 블럭으로 결정한다. 움직임 블럭 결정에 두개의 다른 문턱값을 사용하는 이유는 윤곽선 블럭이 배경 블럭보다 더 잘 부호화 되어야 하기 때문이다. 이와 같이 현재 블럭  $b_i$ 에 대한 움직임 정보를 움직임 맵에 저장한다.

### 3.4 병합 맵 작성

본 논문에서 제안한 동영상 부호화 알고리즘을 적용할 경우, 현재 프레임의 각 블럭들에 대한 윤곽선

정보와 움직임 정보를 전송해야 하는데, 이 두 가지를 모두 전송하면 전송 비트 수가 크게 증가한다. 이는 실제 각 블럭에 대해 전송하는 벡터 양자화의 인덱스 정보가 아니기 때문에 오버헤드 정보가 되어 결과적으로 고정된 비트율을 가진 채널 상으로 실제 데이터 정보를 전송하기 위한 비트 수를 감소시켜야 한다. 그러므로 본 논문에서는 움직임 맵과 윤곽선 맵을 하나로 통합한 병합 맵(merging map)을 작성하여 비트율을 줄일 수 있는 알고리즘을 제안한다.

인트라 프레임은 주 부호책으로부터 벡터 양자화를 수행하여 그 인덱스만을 전송하므로 윤곽선 맵의 정보만 복호화 단으로 전송한다. 인트라 프레임의 바로 다음 예측 프레임의 경우에는 이전 프레임인 인트라 프레임과 현재 프레임, 그 다음 '인트라+2' 프레임의 경우에는 이전 예측 프레임과 현재 예측 프레임간에 각 블럭 위치에서 현재 블럭과 이전 블럭의 움직임 맵과 윤곽선 맵으로부터 병합 맵을 작성한다. 병합 맵 작성 알고리즘을 그림 4에 나타내었다.

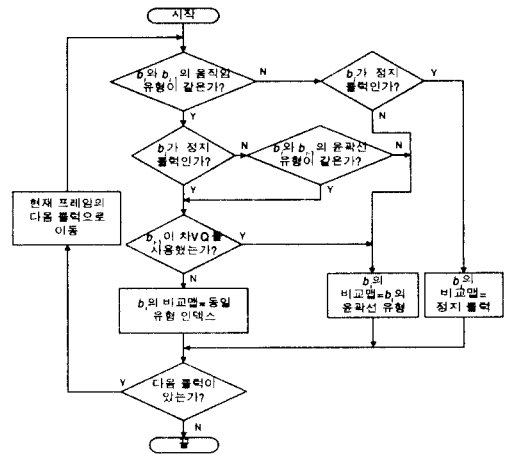


그림 4. 병합 맵 작성 알고리즘  
Fig. 4 Construction algorithm for merging map

그림 4에서 병합 맵의 정보가 “동일 유형 인덱스”인 블럭은 움직임과 윤곽선 정보가 이전 프레임의 정보와 같은 것이고, “정지 블럭”은 움직임이 없는 즉,  $b_{i-1}$ 을  $b_i$ 로 그대로 복사하는 블럭을 나타낸다. 흐름도에서 “ $b_{i-1}$ 이 차 VQ를 사용했는가?”를 판단하는

부분은 차후에 설명할 복원 영상의 화질을 높이기 위해서 사용하는 차 벡터 양자화기(VQ)에 의해 삽입된 것이다.

그림 5는 부호화기에서 전송된 병합 맵을 사용하여 움직임 맵과 윤곽선 맵을 작성하는 알고리즘이다.

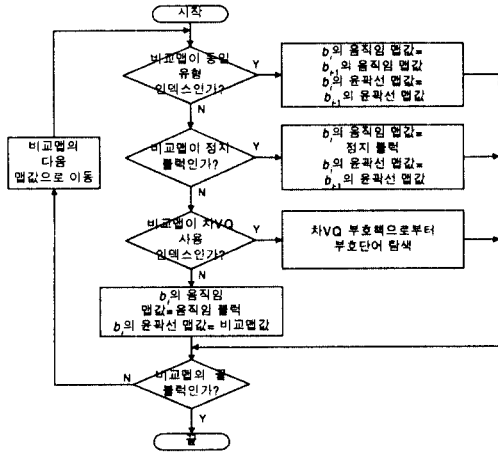


그림 5. 움직임과 윤곽선 맵을 작성하는 알고리즘  
Fig. 5 Construction algorithm of moving and edge map

### 3.5 윤곽선 분류 유한상태 양자화를 이용한 부호화

기존의 SMVQ에서 사용하는 부호화 알고리즘으로 부호화된 영상의 화질은 열화될 수 있는데, 예를 들어, 윤곽선이 현재 블럭의 오른쪽 하단 모서리를 지난다면 그 윤곽선을 올바르게 부호화할 수 없다. 이는 Side-match 유한상태 벡터 양자화기가 현재 블럭을 예측하기 위하여 단지 현재 블럭의 상위 블럭과 좌측 블럭만을 사용하기 때문이다. 그러므로 본 논문에서는 기존 SMVQ 알고리즘을 인터 프레임 부호화에 사용하고 이전 프레임  $f_{i-1}$ 과 현재 프레임  $f_i$ 에 대한 움직임 정보를 이미 가지고 있으므로, 만약 현재 블럭의 우측 블럭과 하단 블럭의 움직임 정보가 정지 블럭이라면  $f_{i-1}$ 에서 그 블럭의 화소 값들을 그대로 복사하게 되므로 현재 블럭의 부호 단어 선택에 좌측, 상위 블럭을 포함하여 최대 4개의 블럭 정보를 사용할 수 있어 그만큼 부호 단어 선택에서의 오류를 감소시킬 수 있다.

입력 프레임  $f_i$ 에 대해 전체 영상을  $N \times N$  블럭으로 나누고, 현재 블럭  $b_i$ 가 움직임 블럭이라면 복호화기로  $b_i$ 의 정보를 보내야 하므로 제한한 윤곽선 분류 유한상태 양자화기를 이용한 부호화를 실행하고,  $b_i$ 가 정지 블럭이라면 복호화 단에서 이전 프레임  $f_{i-1}$ 의 그 위치의 블럭인  $b_{i-1}$ 을 그대로 복사하므로 전송해야 하는 정보가 없다. 그러므로 정지 블럭인 경우는 부호화를 수행하지 않는다.

### 3.6 비트 할당

본 논문에서는 전송로 상으로 병합 맵 정보와 각 블럭의 인덱스를 전송하기 위한 인덱스의 부호화에 허프만 부호화(Huffman coding)[15]를 사용한다. 영상 시퀀스중의 인트라 프레임은 단지 윤곽선 정보만을 가지고 있으므로 병합 맵을 작성할 수 없다. 그러므로 인트라 프레임들은 병합 맵 정보 대신에 윤곽선 맵 정보만을 부호화기로 보낸다. 그러나 인트라 프레임 이후의 인터 프레임 부호화를 수행해야 하는 프레임들은 움직임과 윤곽선 맵 정보 모두를 가지고 있으므로 병합 맵 정보만을 전송한다.

병합 맵에서 발생할 수 있는 정보는 “동일 유형 인덱스”, “배경 블럭”, “16가지 윤곽선 유형 블럭”, “정지 블럭”, “차 VQ 사용 인덱스”등 20가지이다. 이 중에서 실제 전송되는 맵 정보로서 빈도 수가 가장 많은 것이 “동일 유형 인덱스”이다. 그러므로 “동일 유형 인덱스”에 대해서는 줄길이(run length) 부호화를 수행하여 전송 비트 수를 줄인다.

병합 맵의 전송은 크게 세 가지로 나누어지는데, 즉 래스터 스캔(raster scan) 방식으로 동일 유형 블럭으로 맵핑된 병합 맵 블럭의 갯수를 세어 줄길이에 따른 부호를 할당하고, 실제 줄길이 값의 부호 할당, 그리고 “동일 유형 인덱스”의 줄길이가 끝나는 블럭의 블럭 정보 부호를 할당한다. 즉,

$$\begin{aligned} \text{병합 맵 할당 부호} = & \text{“동일 유형 인덱스” 줄길이 레벨 부호} \\ & + \text{줄길이 값 부호} + \text{블럭 정보 부호} \end{aligned} \quad (5)$$

가 된다. 블럭 정보 부호도 허프만 부호화를 사용하여 전송한다.

### 3.7 차 벡터 양자화

본 논문에서는 SMVQ 알고리즘의 문제점을 해결하기 위해서 원 영상과 복원영상과의 차 신호에 대하여 벡터 양자화를 수행하는 차 벡터 양자화(Difference Vector Quantization)를 제안한다. 그림 6에 차 벡터 양자화의 구조를 나타내었다.

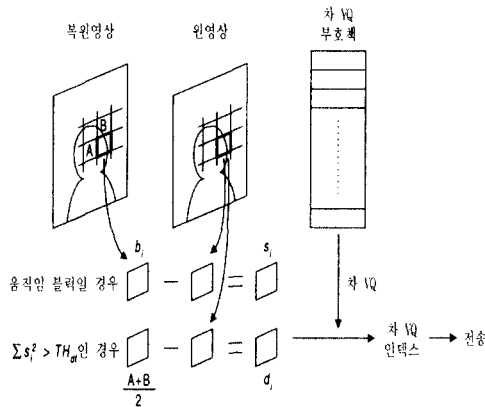


그림 6. 차 벡터 양자화 구조  
Fig. 6 Structure of difference vector quantization

움직임 블럭인  $b_i$ 에 대해서 윤곽선 분류 유한상태 양자화를 수행하고 복원된 영상에 대해 원 영상과의 왜곡( $s_i$ )을 계산하여 그 값이 문턱치  $TH_{dv}$  이상일 경우에 그림 6과 같이  $b_i$ 에 대한 좌측과 상위 블럭의 각각 대응하는 화소 값에 평균을 구하여 얻어진 블럭의 화소 값과 원 영상의 화소 값의 차 신호  $d_i$ 를 구하여 차 벡터 양자화를 수행한다. 이와 같이 차 신호를 구하는데 복원 영상 블럭에서 좌측과 상위 블럭을 사용하는 이유는 차 벡터 양자화가 수행되는 블럭에 대해 부호화기로 전송하는 정보는 오직 차 벡터 양자화 부호책의 인덱스뿐이므로, 부호화기에서 이 정보로부터 영상을 복원하려면 차 신호에 대해 더해줘야 하는 정보가 필요하기 때문이다. 훈련 영상(training image)을 선택하고 윤곽선 분류 유한상태 양자화를 수행한 후에 차 신호를 계산하여 계산된 차 신호에 대해 차 벡터 양자화를 위한 부호책을 작성한다. 그림 6에서와 같이  $d_i$ 에 대하여 차 벡터 양자화 기 부호책으로부터  $d_i$ 에 가장 가까운 부호 단어를

선택하고 그 인덱스를 부호화기로 전송한다.

$d_i$ 에 대해 차 벡터 양자화를 수행하여 차 벡터 양자화 결과인 인덱스를 부호화 단어로 전송하면 그만큼 추가 비트가 요구된다. 본 논문에서는 차 벡터 양자화를 수행함으로써 추가되는 비트 수를 최대한 감소시켜 전송 비트율을 낮게 유지하는 알고리즘을 제안한다.

차 벡터 양자화가 수행되는 블럭  $b_i$ 에 대해서는 움직임 정보나 윤곽선 정보가 전혀 필요하지 않다. 즉, 움직임 블럭인  $b_i$ 에 대해서만 차 벡터 양자화가 수행되고  $b_i$ 의 윤곽선 유형이 필요한 것은 17개(배경과 16 클래스의 윤곽선 유형)의 상태 부호책 중에 어느 것을 선택해야 하는 것인지에 대한 상황 때문이다. 그러나 차 벡터 양자화에서 사용되는 부호책은 차 벡터 양자화 부호책만을 사용하므로 윤곽선 유형 정보가 필요 없다. 많은 블럭에 차 벡터 양자화를 수행하면 복원 영상의 화질은 좋아지지만 전송 비트 수가 늘어나므로  $TH_{dv}$  값으로 적절하게 그 수를 조절한다.

### 3.8 차 영상 부호화

전송 채널의 대역폭이 커질 경우, 제안한 윤곽선 분류 유한상태 차 벡터 양자화(ECFSDVQ)를 수행한 후에도 전송 가능한 비트 수에 여유 분이 생길 수 있다. 이와 같은 경우는 H.261[16]이나 H.263[17]에서와 같이 최종 복원된 영상에 대해 차 영상 부호화(Difference Signal Coding)를 수행한다. 3.7절의 차 벡터 양자화와 이 차 영상 부호화의 차이점은 차 벡터 양자화는 부호화기로 차 벡터 양자화 부호책의 인덱스가 전송되지만, 차 영상 부호화에서는 병합 맵의 정보는 그대로 유지되고 원 영상과 차 신호에 대해 DCT를 이용한 양자화를 수행하여 각 계수 값을 부호화 하여 전송한다. 그러므로 차 영상 부호화 방법은 차 벡터 양자화보다 전송해야 하는 비트 수가 훨씬 많아진다. 그러므로 전송 채널의 대역폭이 커질수록 먼저 차 벡터 양자화를 수행하고, 전송 비트의 여유분이 있을 경우에 차 영상 부호화를 수행한다.

본 논문에서는 차 영상 부호화에 JPEG[18]에서 사용한 방법과 유사한 알고리즘을 사용한다. JPEG 알고리즘과의 차이점은 부호화의 대상이 원 영상이 아니라 복원 영상과 원 영상과의 차 신호라는 것이다.

차 신호에 대한 DCT를 계산한 후에, DC 계수는 이전 블록 DC 계수와의 DPCM을 수행하고, AC 계수 부호화는 양자화를 취하여 지그재그 시퀀스로 부호화한다. 그림 7에는 본 논문에서 사용한 4×4 양자화 배열이 나타나 있다.

0	0	2	5
0	2	2	5
2	2	5	5
5	5	5	5

그림 7. AC 계수에 대한 양자화 배열  
Fig. 7 Quantization matrix for AC coefficients

본 논문에서는 JPEG과 달리 그림 7의 양자화 배열의 각 값에 Quant 값을 곱하여 양자화 되는 값을 조절할 수 있게 하였다. 즉, 전송되어야 하는 비트 수에 여유가 있을 때는 Quant 값을 작게 하여 "0"의 줄길이를 감소시키고, 비트를 제한해야 하는 경우는 Quant 값을 크게 하여 "0"의 줄길이를 증가시켜 전송 비트 수를 감소시킨다.

### 3.9 비트율 제어

본 논문에서는 전송 채널의 대역폭에 따라 전송 비트율을 제어하기 위해서 표 1과 같은 비트율 제어 모드를 구성하였다. 표 1에서, 제한한 ECFSDVQ 방법을 기본 모드로 하였고, 만약 전송 채널 대역폭이 기본 모드시의 전송 비트 수보다 더 많은 비트를 전송할 수 있다면 증가모드를 두어 복원 영상의 화질을 높일 수 있게 하였다.

차 영상 부호화를 사용하는 모드에서 양자화 단계를 조절하는 Quant 값을 사용하여 비트율 제어를 할 수 있다. Quant 값은 양자화 배열에 곱해지는 수이기 때문에 그 값의 변화는 전송 비트 수를 크게 변화시킨다. 그러므로 기본 모드로부터 증가 모드로 전환 시에 Quant 값은 8부터 1씩 8 단계로 감소시켰으며, 감소 모드로 전환 시는 그 반대인 1씩 증가시킨다.

현재 프레임의 부호화에 표 1의 어느 모드를 사용할 것인가는 현재 프레임의 전송되는 비트 수를 계산

표 1. 비트율 제어 모드

Table 1. Bit rate control modes

모 드	프레임 부호화 방법
증가모드 6	모든 프레임이 인트라 프레임 + 차 영상 부호화
증가모드 5	모든 프레임이 인트라 프레임
증가모드 4	모든 움직임 블록에 차 벡터 양자화 + 차 영상 부호화
증가모드 3	모든 움직임 블록이 인트라 프레임 부호화 + 움직임 블록들에 차 영상 부호화
증가모드 2	모든 움직임 블록이 인트라 프레임 부호화
증가모드 1	몇몇 움직임 블록들만 인트라 프레임 부호화
기본모드	모든 움직임 블록들에 차 벡터 양자화
감소모드 1	모든 움직임 블록들에 Side-match 벡터 양자화
감소모드 2	상대 부호책의 크기를 줄여 양자화
감소모드 3	$TH_{mb}$ , $TH_{mc}$ 를 가변시켜 부호화

하여 이전 프레임들과의 전송 비트 수의 평균을 계산하여 결정한다. 그래서 그 값이 전송 대역폭보다 작으면 한 단계 증가 모드로, 크면 한 단계 아래 모드로 선택하여 부호화를 수행한다.

## IV. 실험 결과 및 고찰

본 논문의 성능 평가를 위해 기존 벡터 양자화 알고리즘인 LBG와 TSVQ, SMVQ, QTSMVQ와 결과 비교를 하였고, 전송률 측면에서 64Kbps이하 전송 대역폭에 대해서는 H.263과, 64Kbps 이상에서는 H.261과 비교하였다. 실험 영상으로 QCIF 형식(176×144)의 영상 시퀀스 Claire와 Salesman의 1에서 150 프레임을 사용하였고, 원 영상 시퀀스는 초당 30 프레임이지만 매 세 번째 프레임만을 부호화 하였다. 즉, 표본화율이 1/3이 되게 하여 알고리즘을 적용하였고 블록의 크기는 4×4로 고정하였다. 주 부호책의 크기는 기본적으로 256으로 하였으며, 주 부호책 설계에 사용한 훈련 영상(training image)은 Salesman 영상중 45 프레임을 선택하여 LBG 알고리즘으로 부호책을 작성하였다.

Sobel 연산자 사용시의 문턱치는 128을 사용하였고, DCT 계수의 윤곽선 분류 각도  $\theta$ 에 대해서 Claire 영상은 22로, Salesman 영상은 14로 하였다. 또한 DCT 계수 분류 윤곽선 영역 크기  $\delta$ 는 모두 30으로 하였다. 그리고 움직임 블록 판별 문턱치에 대해서



Claire 영상은  $TH_{mb}$ 와  $TH_{me}$  모두 500으로 하였으며, Salesman 영상은 1000으로 하였다. 그리고 차 벡터 양자화 실행 문턱치인  $TH_{df}$ 는 두 영상 모두 2500으로 하였다. 이와 같은 결과는 실험적으로 최적의 상태를 결정 한 것이다.

제안 알고리즘과 기존 벡터 양자화 알고리즘인 LBG, TSVQ, SMVQ, QTSMVQ와 부호화 결과를 비교하였는데, LBG와 TSVQ, SMVQ에 대해서는 제안 알고리즘과의 동영상 부호화 비교를 위해 각 프레임의 움직임과 정지 블록의 판별은 본 논문의 알고리즘을 그대로 사용하였다. LBG와 TSVQ의 경우 부호책의 크기는 64로 하였는데, 이는 제안 알고리즘과 비트율을 유사하게 하기 위해서였고, SMVQ의 경우 주 부호책의 크기는 256으로, 상태 부호책의 크기는 16으로 하였으며, QTSMVQ에서 윤곽선 유형의 갯수는 16으로 같게 하였다. 표 2에는 기존 벡터 양자화와 제안 알고리즘의 BPP 결과가 나타나 있다. LBG와 TSVQ, SMVQ에 비해서 제안 알고리즘이 더 적은 BPP를 나타내었고, QTSMVQ에 대해서는 제안 알고리즘이 약간 높은 BPP를 유지했지만, 그 차이는 매우 작았다.

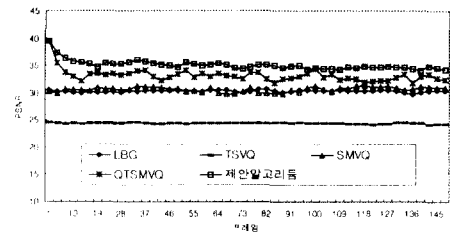
표 2. BPP 비교

Table 2. BPP comparison

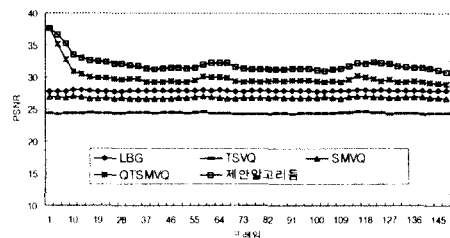
	LBG	TSVQ	SMVQ	QTSMVQ	제안알고리즘
Claire	0.1217	0.1217	0.1014	0.0662	0.0669
Salesman	0.1467	0.1467	0.1203	0.0706	0.0713

그림 8에는 각 프레임별로 실험 영상에 대한 기존 벡터 양자화와 제안 알고리즘의 PSNR 결과가 나타나 있는데, 두 영상에 대해 다른 기존 벡터 양자화 알고리즘에 비해 제안 알고리즘이 훨씬 좋은 PSNR 값을 나타냄을 볼 수 있다. QTSMVQ에 비해서 제안 알고리즘이 BPP 값은 평균적으로 약간 높았으나, 복원 영상의 화질이 좋기 때문에 BPP에 대한 PSNR 측면에서 제안 알고리즘이 우수하다는 것을 알 수 있다.

표 3에는 전송률에 따른 H.263, H.261과 제안 알고리즘과의 결과 비교가 나타나 있다. H.263은 64Kbps 이하의 저 전송률 비디오 코덱에 관한 권고안이므로 64Kbps 미만에서는 H.263과 비교하였고, 64Kbps 이상에서는 H.261과 제안 알고리즘의 결과를 비교하였다. 표 3으로부터 Claire 영상 시퀀스에 대해서는 제



(a)



(b)

그림 8. PSNR 비교. (a) Claire 영상 시퀀스 (b) Salesman 영상 시퀀스

Fig. 8 PSNR Comparison. (a) Claire Sequence (b) Salesman Sequence

표 3. 전송률에 대한 PSNR(dB) 비교 (1~150 프레임)

Table 3. PSNR(dB) comparison for bit rate (1~150 frames)

	전송률(bps)	H.263			H.261				
		24K	32K	48K	64K	128K	192K	256K	
Claire	H.263	35.73	37.68	40.20	H.261	36.91	38.44	39.04	41.42
	제안 알고리즘	35.13	37.06	40.21	제안 알고리즘	40.94	41.74	42.68	44.58
Salesman	H.263	31.73	32.72	33.88	H.261	32.53	33.65	35.32	36.52
	제안 알고리즘	32.02	34.35	35.81	제안 알고리즘	36.59	36.71	38.45	42.42

안 알고리즘이 H.263보다 PSNR 값이 다소 떨어진다. 24Kbps의 초 저 전송률 부호화에서 제안 알고리즘은 기본 모드로 동작하였다. 전송 비트 수가 대역폭보다 커지면 상태 부호책의 수를 작게 하거나,  $TH_{mb}$ 와  $TH_{me}$ 의 값을 증가시켜 전송 비트 수를 낮추었다. 24Kbps와 32Kbps에서는 결과 값이 H.263에 비해 낮은 값이 나왔다. 그러나 48Kbps에서는 다소 영상의 PSNR 값이

증가하여 H.263과 거의 같은 값을 가지고 있어 복원 영상의 화질이 비슷하게 된 것을 알 수 있다. 그러나 배경 부분이 복잡한 Salesman 영상 시퀀스에 대한 결과는 64Kbps 미만의 대역폭에서 H.263보다 0.29dB ~ 1.93dB 정도 제안 알고리즘의 성능이 더 우수한 것을 알 수 있다.

64Kbps 이상의 대역폭에서는 제안 알고리즘이 모두 증가 모드에서 동작하였다. Claire 영상에서 모든 전송률에 대해 제안 알고리즘의 평균 PSNR이 약 3~4 dB 정도 더 좋았고, Salesman 영상에 대해서는 3~6 dB 정도 더 좋은 결과를 얻을 수 있었다. 전송률이 128Kbps 이상인 경우는 전송 비트 수에 여유 분이 많아지기 때문에 비트율 제어에서 차 영상 부호화와 Quant 값을 조정하는 과정이 포함되었다.

## V. 결 론

본 논문에서는 부호책에서 최적의 부호단어를 선택하는 성능을 그대로 유지하면서 전송 비트 수를 줄이려는 기법으로 윤곽선 분류 유한상태 차 벡터 양자화를 제안하였다. 기존의 유한상태 벡터 양자화에서는 영상의 두드러진 특징인 윤곽선 정보가 옳게 전달되지 못하기 때문에 본 논문에서는 영상의 윤곽선 정보에 따라 주 부호책을 따로 설계하는 양자화 기법을 적용하였다.

기존의 유한상태 벡터 양자화를 이용한 동영상 부호화에서는 윤곽선 정보를 이용하여 현재 블럭을 부호화하지만 복호화 시에는 이전 블럭들로부터의 상관관계를 이용하여 윤곽선 정보를 예측함으로써 복원 영상에서 오류를 발생시켰다. 본 논문에서는 이점을 개선하기 위해서 현재 블럭의 윤곽선 정보도 복호화기로 전송하였다. 그러나 이 경우 전송 비트 수가 증가되기 때문에 한 프레임의 윤곽선 정보와 움직임 정보를 복호화기로 전송하기 위한 병합 맵을 설계하여 병합 맵 정보 전송을 위한 비트 할당을 수행하였다. 또한 Side-match 기법을 사용하는 유한상태 벡터 양자화기에서 현재 블럭의 부호단어를 선택하기 위해 이웃 블럭들과의 상관관계를 이용함으로써 복원 영상의 화질을 열화시키는데, 본 논문에서는 현재 블럭에 대해 유한 상태 벡터 양자화를 수행한 후 복원 영상과 원 영상과의 차 신호를 구하여 그에 대한 차 벡터 양자화를 수행함으로써 Side-match로 인한 오류를 줄

이도록 하였다. 채널의 전송 대역폭에 따라 전송 가능한 비트 수에 여유가 있다면 복원 영상과 원 영상과의 차 영상 부호화를 수행하는데, 차 벡터 양자화는 달리 DCT를 이용한 양자화를 적용함으로써 복원 영상의 화질을 더욱 좋게 하였다.

본 논문의 제안 알고리즘과 기존의 벡터 양자화 알고리즘의 성능 평가 결과, 제안 알고리즘의 복원 영상 화질이 훨씬 좋았으며, 저 전송률 비디오 압축 표준안인 H.261, H.263과의 성능 비교에서 H.261보다 제안 알고리즘이 같은 전송 비트율에서 좋은 성능을 보였다. H.263과는 Salesman과 같은 배경이 비교적 복잡한 영상 시퀀스에 대해서는 제안 알고리즘이 더 좋은 성능을 나타냈으나, Claire와 같이 배경이 단조로운 영상 시퀀스에 대해 성능 평가 전송률의 초 저속 대역폭인 24K와 32Kbps에서 제안 알고리즘의 복원 영상 화질이 다소 떨어지는 경향이 있었으나, 그 이상의 전송 대역폭에서는 H.263보다 더 좋은 성능을 나타내었다.

향후 연구 과제로서는 제안 알고리즘의 각 문턱치에 대해 본 논문에서는 실험적으로 최적치를 구하였으나 이에 대해 적용적으로 값들을 결정하는 알고리즘이 필요하다. 그리고 제안 알고리즘은 각 프레임의 블럭들을 움직임 블럭과 정지 블럭으로 나누어 부호화를 수행하므로 움직임이 상당히 많은 영상에 대해서는 전송 비트 수가 그만큼 많아진다. 그러므로 움직임이 많은 영상 시퀀스에 대한 보다 더 효과적인 부호화 알고리즘이 연구되어야 한다.

## 참 고 문 헌

1. L. Wang and M. Goldberg, "Reduced-difference Pyramid: A Data Structure for Progressive Image Transmission," *Opt. Eng.*, vol. 28, no.7, pp. 708-716, July 1989.
2. M. Kocher and M. Kunt, "A Contour-Texture Approach to Picture Coding," *ICASSP-82, Paris, France*, pp.436-440, May 1982.
3. 노근수, 김정우, 김용성, 이근영, "병합 쿼드 트리를 이용한 영상 부호화의 준최적 비트 할당," 제 7회 신호처리합동학술대회. Vol. 7, No. 1, pp. 140-143, 10월. 1994년.
4. Y. Linde, A. Buzo, and R. M. Gray, "An Algor-

ithm for Vector Quantizer Design," IEEE Trans. Commun., vol. COM-28, pp. 84-95, Jan. 1980.

5. A. Gersho and V. Cupermann, "Vector Quantization: A Pattern Matching Technique for Speech Coding," IEEE Commun. Mag., vol. 21, pp. 15-21, Dec. 1983.
6. R. M. Gray, "Vector Quantization," IEEE ASSP Mag., vol. 1, pp.4-29, April 1984.
7. B. Ramamurthi and A. Gersho, "Classified Vector Quantization of Images," IEEE Trans. Commun., vol. COM-34, no.11, pp. 1105-1115, Nov. 1986.
8. C. H. Hsieh, "DCT-Based Codebook Design for Vector Quantization of Images," IEEE Trans. on Circuits and Sys. for Video Tech., Vol. 2, No. 4, Dec. 1992.
9. J. Foster, R. M. Gray and M. O. Dunham, "Finite-state Vector Quantization for Waveform Coding," IEEE Transaction Information Theory, vol. IT-31, pp. 348-359, May 1985.
10. N. M. Nasrabadi and Y. Feng, "A Dynamic Finite-state Vector Quantization Scheme," in Proc. ICASSP, 1990, pp. 2261-2264.
11. R. F. Chang and W. M. Chen, "Interframe Difference Quadtree Edge-Based Side-match Finite-State Classified Vector Quantization for Image Sequence Coding," IEEE Trans. on Circuits and Sys. for Video Tech., Vol. 6, No. 1, Feb. 1996.
12. T. Kim, "New Finite State Vector Quantizers for Images," in Proc. ICASSP, 1988, pp. 1180-1183.
13. I. W. Nam, P. G. Park, E. S. Kim, K. Y. Lee, "A Dct Based Hierarchical Finite State Vector Quantization For Image Coding," Proceedings of The 4th International Workshop on Mobile Multimedia Communications(MoMuC '97), pp. 572-575, Oct. 1997.
14. C. H. Yim and J. K. Kim, "A Simple DCT-CVQ Based on Two DCT Coefficients," PCS 88, Picture Coding Symp., pp. 8.11-1 through 8.11-2, Torino, Italy, Sept. 1988.
15. D. A. Huffman, "A Method for The Construction of Minimun Redundancy Codes," Proc. IRE, vol. 40, no. 10, pp. 1098-1101, Sept. 1952.
16. Draft ITU-T Recommendation H.261, "Video Co-

dec for Audiovisual Services at  $p \times 64\text{kbits/s}$ ," Aug. 1990.

17. Draft ITU-T Recommendation H.263, "Video Coding for Low Bit Rate Communication," May, 1996.
18. JPEG IS, "Information Technology-Digital Compression and Coding of Continous-Tone Still Image," ISO/IEC 10918, 1993.



이근영(Keun Young Lee) 정회원  
1947년 12월 30일  
1966년 3월~1973년 2월: 전남대학교 전기공학과(공학사)  
1973년 3월~1975년 2월: 한양대학교 전자공학과(공학석사)

1975년 3월~1978년 8월: 한양대학교 전자공학과(공학박사)

1975년 3월~1977년 2월: 한양대학교 강사

1977년 3월~1981년 8월: 광운대학교 전임강사, 조교수

1979년 3월~1980년 2월: Denmark공대 객원교수

1987년 9월~1988년 8월: Loughborough대(영) 객원교수

1981년 9월~현재: 성균관대학교 조교수, 부교수, 교수

1995년 8월~1997년 7월: 성균관대학교 부설 정보통신기술연구소 소장

1996년 3월~1998년 1월: 성균관대학교 부설 정보통신대학원 원장

1998년 3월~현재: 성균관대학교 전기전자컴퓨터공학부 부장

※주관심분야: 디지털 방송, 멀티미디어, 영상 및 문서 인식, 압축 마이크로프로세서 응용, 디지털 전력제어 시스템



김응성(Eung Sung Kim) 정회원  
1967년 2월 22일생

1989년 2월: 성균관대학교 공과대학 전자공학과(공학사)

1992년 2월: 성균관대학교 대학원 전자공학과(공학석사)

1998년 2월: 성균관대학교 대학원 전자공학과(공학박사)

1995년 3월~현재: 성균관대학교 부설 과학기술연구소 연구전담요원 근무

※주관심분야: 영상통신, HDTV 신호처리, 신경회로망  
e-mail : imagecap@chollian.net