

복수 특징의 사전 검사에 의한 영상 벡터양자화의 고속 부호화 기법

정회원 류철형*, 나성웅**

A Fast Encoding Algorithm for Image Vector Quantization Based on Prior Test of Multiple Features

Chul-hyung Ryu*, Sung-woong Ra** *Regular Members*

요 약

본 논문에서는 영상 벡터 양자화를 위한 새로운 고속 부호화 기법을 제안하는데, 제안 기법은 다차원의 참조 표로 복수 특징의 부분 거리를 사용한다. 복수 특징을 사용하는 기존 기법은 탐색 순서와 연산 과정을 고려할 때 복수 특징을 단계적으로 처리한다. 반면에 제안 기법은 참조 표를 사용하여 복수 특징들을 동시에 활용한다. 본 논문에서는 가용한 수준의 메모리를 위해 테두리 효과를 고려하는 참조 표의 구성 방법과 참조 표의 부분 거리를 활용하여 현재의 탐색을 중지하는 방법을 상세하게 기술한다. 시뮬레이션 결과는 제안 기법의 효율성을 확인시켜 주는데, 부호책 크기가 256일 때 제안 기법은 OHTPDS 기법이나 M-L₂NP 기법 등과 같이 최근에 제안된 기법들이 요구하는 연산량의 70% 수준까지 연산량을 감소시킨다. 가용한 수준의 전처리와 메모리를 사용함으로써 제안 기법은 전체 탐색 기법과 동일한 화질을 유지하면서 전체 탐색 기법이 요구하는 연산량의 2.2% 이하로 연산량을 감소시킨다.

Key Words : vector quantization, full search equivalent, fast encoding, multiple features

ABSTRACT

This paper presents a new fast encoding algorithm for image vector quantization that incorporates the partial distances of multiple features with a multidimensional look-up table (LUT). Although the methods which were proposed earlier use the multiple features, they handles the multiple features step by step in terms of searching order and calculating process. On the other hand, the proposed algorithm utilizes these features simultaneously with the LUT. This paper completely describes how to build the LUT with considering the boundary effect for feasible memory cost and how to terminate the current search by utilizing partial distances of the LUT. Simulation results confirm the effectiveness of the proposed algorithm. When the codebook size is 256, the computational complexity of the proposed algorithm can be reduced by up to the 70% of the operations required by the recently proposed alternatives such as the ordered Hadamard transform partial distance search (OHTPDS), the modified L₂-norm pyramid (M-L₂NP), etc. With feasible preprocessing time and memory cost, the proposed algorithm reduces the computational complexity to below the 2.2% of those required for the exhaustive full search (EFS) algorithm while preserving the same encoding quality as that of the EFS algorithm.

I. 서 론

벡터 양자화(VQ)는 효율적인 신호 압축 기법^[1]의

하나로 입력 벡터를 부호책(codebook)의 부호 벡터들(code vectors)과 비교하여 가장 유사한 부호 벡터로 대체하는 기법이다. 벡터 양자화는 압축 효율

* 국방과학연구소 1체계-1부-2팀,

논문번호 : KICS2005-10-410, 접수일자 : 2005년 10월 12일

** 충남대학교 정보통신공학부 영상통신 연구실 (swra@cnu.ac.kr)

에 비해 복호기 구조가 간단한 장점을 가지며, 음성 압축과 영상 압축 등에 폭 넓게 적용되었고, 최근에는 음성 인식과 watermarking 분야에 활용되고 있다. 벡터 양자화와 관련된 연구는 크게 두 가지로 분류되는데, 훈련 벡터 집합(training vector set)에서 부호책을 효율적으로 구성하는 연구와 부호화 과정에서 최적의 부호 벡터를 고속으로 탐색하는 연구가 그것이다. 본 논문은 후자에 해당된다.

벡터 양자화의 부호기는 k -차원의 유클리디안 공간(Euclidean space)을 유한개 부분 집합인 부호 벡터들로 사상하는 것이며, $Q: \mathbb{R}^k \rightarrow C$ 라 표기한다. 여기서, $C = \{c_1, c_2, \dots, c_N\}$ 은 부호 벡터들의 집합인 부호책이며, N 은 부호책의 크기이다. 영상은 수평과 수직의 2차원적 상관성을 가지므로 영상 벡터 양자화를 위해 영상은 작은 크기의 블록(즉, $B_H * B_V = k$)으로 분할되며, 각각의 k -차원 입력 벡터는 부호책의 모든 부호 벡터 중에서 가장 짧은 거리에 위치하는 최근접(nearest) 부호 벡터로 양자화 된다. 일반적으로 거리 측정을 위해 제곱 유클리디안 거리(squared Euclidean distance, SED)가 사용되는데, 입력 벡터 (\mathbf{x})와 j -번째 부호 벡터($c_j \in C$) 사이의 SED 연산은 식 (1)과 같이 정의된다.

$$d_{E,j} = d(\mathbf{x}, c_j) = \sum_{i=1}^k (x_i - c_{ji})^2 \quad (1)$$

입력 벡터에 대해 최근접 부호 벡터를 찾는 과정은 수많은 SED 연산을 요구하며, 이는 벡터 양자화에서 병목 현상을 일으키는 원인이 된다. 이 문제를 해결하고자 여러 가지 탐색 기법^[2-19]이 제안되었는데, II장에서 최근 제안된 기법들을 중심으로 기존 기법에 대해 살펴본다. 본 논문은 하다마드 변환(Walsh Hadamard transform, WHT)과 다차원의 참조 표를 사용하여 2개 이상의 복수 특징을 동시에 고려함으로써 기존 기법보다 연산량이 적고 탐색 속도가 빠른 새로운 부분 부호책 탐색 기법을 제안하는데, III장에서 제안 기법에 대해 상세히 기술한다. 시뮬레이션 결과는 IV장에서 설명되며, 마지막으로 V장에서는 결론을 맺는다.

II. 기존 기법에 대한 고찰

기존 기법에 대한 객관적인 평가를 위해 가장 보편적인 평가 기준을 선택하였는데, 이 방법은 거의

모든 논문에서 해당 제안 기법의 연산량 또는 탐색 속도를 평가하기 위해 시뮬레이션 조건으로 사용되는 방법이다. 즉, 훈련 벡터 집합으로는 512x512 픽셀 크기의 Lena 영상만 사용하고, 4x4 픽셀 크기로 영상을 블록화 하여 16-차원 벡터를 구성한다. 부호책 생성에는 LBG(Linde, Buzo, and Gray) 기법^[1]을 사용하고, 벡터 양자화의 시험 영상으로는 Lena 영상을 적용한다. 각 논문이 제시하고 있는 탐색 속도 또는 연산량을 전체 탐색(exhaustive full search, EFS) 기법에 대한 비율로 표시함으로써 직관적으로 해당 기법의 성능을 평가할 수 있게 하였다. 본 장에서는 현재까지 제안된 고속 부호화 기법 중에서 EFS 기법과 동일한 화질을 가지면서 동시에 EFS 기법의 연산량 또는 탐색 속도를 100%라고 설정할 때 5% 이하 수준의 연산량 또는 탐색 속도를 갖는 기존 기법들을 살펴보고자 한다.

2.1 부분 거리를 활용하는 고속 탐색

1985년에 Bei와 Gray가 제안한 부분 거리 소거(partial distance elimination, PDE) 기법^[2]은 식 (1)의 SED 연산에서 누적 거리가 감소되지 않는 점에 착안하여 최근접 부호 벡터를 찾는 중에 SED 연산의 누적 거리가 현재까지 찾은 최소 거리($d_{E,min}$)보다 커질 경우 현재의 SED 연산을 중단한다. PDE 기법이 픽셀 영역에서 수행되면 연산량은 EFS 기법의 20% 수준으로 감소되며, 변환 영역에서 수행될 경우 13% 수준으로 감소된다. 이 기법은 본 논문의 제안 기법뿐만 아니라 대부분의 기법에서 부적합한 부호 벡터에 대한 SED 연산을 조기에 종료하기 위해 적용되고 있다.

2.2 삼각 부등식을 활용하는 고속 탐색

1989년에 Chen과 Pan이 제안한 삼각 부등식 소거(triangular inequality elimination, TIE) 기법^[3]은 공간상에 3점이 있을 때 2변 길이의 합은 다른 한 변의 길이보다 작지 않다는 점을 사용하며, 벡터 양자화 수행 전에 각 부호 벡터를 기준으로 다른 부호 벡터까지 k -차원 공간상에서의 상대 거리를 참조 표(look-up table, LUT)에 저장한다. TIE 기법은 LUT 구성을 위한 메모리를 필요로 하며, 초기 부호 벡터를 선정하는 방식에 따라 탐색 성능이 달라진다. 1998년에 Choi와 Chae는 다수의 기준 anchor들에 대한 정렬 색인 표를 활용하는 기법^[4]을 제안하여 TIE 기법의 성능을 향상하였는데, 이 기법은 연산량을 EFS 기법의 4.7% 수준으로 감소시킨다.

2.3 단일 특징을 활용하는 고속 탐색

1993년에 Ra 와 Kim이 제안한 평균 거리 정렬 부분 부호책 탐색(mean-distance-ordered partial codebook search, MPS) 기법^[5]은 입력 벡터의 평균을 특징으로 사용하며, SED 연산에 앞서 사전 검사를 수행함으로써 현재까지 얻은 최소 거리보다 큰 거리를 가질 수밖에 없는 부호 벡터를 배제한다. MPS 기법은 특징값 저장을 위한 메모리를 요구하지만, 부호책의 일부만 탐색(partial codebook search) 함으로써 픽셀 영역에서 수행될 경우 연산량이 EFS 기법의 4.5% 수준으로 감소되며, 변환 영역에서 수행되면 3.3% 수준으로 감소된다.

2.4 두개의 특징을 활용하는 고속 탐색

1994년에 Lee와 Chen이 제안한 평균 및 분산 탐색(mean or variance, MOV) 기법^[6]과 1997년에 Baek 등이 제안한 평균과 분산 탐색(mean and variance, MAV) 기법^[7]은 2개의 특징(평균과 분산)을 사용함으로써 MPS 기법에 비해 적은 탐색 시간을 갖는다. 픽셀 영역에서 MAV 기법이 수행되면 연산량은 EFS 기법의 4.2% 수준으로 감소된다. 2003년에 Lu 등은 변환 영역에서 MAV 기법을 적용한 OHTPDS(ordered Hadamard transform partial distance search) 기법^[8]을 제안하였다. 변형된 분산 ($V_x = \sqrt{\sum_{i=2}^k X_i^2}$)을 사용하는 OHTPDS 기법의 연산량은 EFS 기법의 3.5% 수준으로 MPS 기법보다 다소 증가되나, 벡터 차원과 부호책의 크기가 커질 수록 분산 효과가 증가되어 연산량이 감소된다.

2.5 세개 이상의 특징을 활용하는 고속 탐색

탐색 성능을 향상하기 위해 3개 이상의 특징을 고려하는 기법들이 제안되었는데, 2000년에 Choi와 Chae가 제안한 Extend MOS(mean-distance-ordered search) 기법^[9]은 MPS 기법에 anchor 벡터들을 접목 함으로써 연산량을 EFS 기법의 4.2% 수준으로 감소시킨다. 2003년에 Pan 등이 제안한 SIEENNS(subvector-based improved equal-average equal-variance nearest neighbor search) 기법^[10]은 부벡터들을 사용하여 연산량을 EFS 기법의 4.6% 수준으로 감소시킨다. 2005년에 Pan 등은 SIEENNS 기법을 개선하여 연산량을 EFS 기법의 3.5% 수준으로 감소시키는 ISIEENNS(improved SIEENNS) 기법^[11]을 제안하였다.

2.6 계층적 구조를 활용하는 고속 탐색

2002년에 Song과 Ra가 제안한 L₂NP(L₂-norm

pyramid) 기법^[12]은 픽셀 영역에서 계층적 구조를 형성하고, 탐색 중지를 위한 부등식을 계층별로 적용함으로써 연산량을 EFS 기법의 3.5% 수준으로 감소시킨다. 이 기법을 시작으로 계층 구조를 활용하는 많은 기법이 제안되고 있다. 2005년에 Pan 등이 제안한 M-L₂NP(modified L₂NP) 기법^[13]은 연산량을 EFS 기법의 3.3% 수준까지 감소시키는데, 이 수치는 픽셀 영역의 탐색 기법 중에서 가장 작다.

2.7 변환을 활용하는 고속 탐색

픽셀 영역에서 많은 기법이 제안되는 동안 변환 영역에서 수행되는 기법은 간헐적으로 제안되었다. 1996년에 Tai 등이 제안한 DHSS(dynamic hyperplanes shrinking search)^[14] 기법에 이산 여현 변환(discrete cosine transform)을 적용한 경우 연산량은 EFS 기법의 3.0% 수준으로 감소되지만, 탐색 시간은 EFS 기법의 4.0% 수준까지만 감소된다. 또한 2004년에 Pan 등은 변환 영역에서 수행되는 IDWT(improved discrete Walsh transform) 기법^[15]을 제안하였는데, 이 기법은 EFS 기법의 3.5% 수준으로 연산량을 감소시킨다.

III. 제안 기법

기존 기법에 대한 고찰을 통해 2개 이상의 복수 특징과 에너지를 집중하는 변환 영역을 사용하는 것이 고속 탐색에 유리함을 알 수 있었다. 또한, 기존 기법들은 복수 특징들을 동시에 활용하지 못함을 알 수 있었는데, 예를 들어 MAV 기법^[7]을 보면 SED 연산 여부 결정에는 평균과 분산을 동시에 고려하지만, 초기 부호 벡터 선정, 탐색 순서 및 탐색 중지 결정에는 MPS 기법과 동일하게 평균만 고려한다. 본 논문은 미리 준비되는 참조 표의 부분 거리를 사전 검사하여 복수 특징을 동시에 고려할 수 있는 새로운 구조의 DFPS(double feature-ordered partial codebook search) 기법을 제안하고자 한다.

3.1 복수 특징의 선택

특징이란 평균이나 분산과 같이 k-차원의 부호 벡터 및 입력 벡터의 특성을 수치로 표현한 것이다. 제안 기법에서는 WHT의 결과 계수들(coefficients) 중에서 분산이 큰 계수들을 선택하여 사용한다. 이렇게 하는 이유는 3개 이상의 복수 특징을 용이하게 선택할 수 있으며, 복수 특징들이 해당 벡터에 포함되는 에너지를 대표할수록 제안 기법의 성능이 향상되기 때문이다.

WHT는 곱셈 없이 덧셈과 뺄셈만으로 에너지를 일부 계수로 집중할 수 있다. 512x512 픽셀의 Lena 영상을 4x4 픽셀 크기로 나눈 다음 2차원 WHT를 수행하고 각각의 계수에 대해 전체 에너지에 대한 비율과 분산 크기를 계산하면 표 1과 같다. 표에서 알 수 있듯이 에너지가 일부 계수에 집중되므로 변환 영역에서 PDE 기법^[2]을 적용하면 부적격한 부호 벡터와의 SED 연산을 빨리 종료할 수 있다.

표 1. 2차원 WHT 계수의 에너지 비율과 분산 값
Table 1. Energy Ratio and Variances of 2-D WHT coefficients (in zig-zag scan order)

WHT 계수	에너지 비율[%]	분산 값
W_{00}	98.6316%	41976.314
W_{01}	0.5516%	1112.549
W_{10}	0.1112%	224.351
W_{20}	0.2289%	461.743
W_{11}	0.0987%	199.053
W_{02}	0.1179%	237.772
W_{03}	0.0414%	83.554
W_{12}	0.0236%	47.526
W_{21}	0.0317%	63.860
W_{30}	0.0382%	77.061
W_{31}	0.0223%	45.057
W_{22}	0.0188%	37.849
W_{13}	0.0105%	21.172
W_{23}	0.0532%	107.268
W_{32}	0.0117%	23.659
W_{33}	0.0087%	17.454

3.2 참조 표의 구성

부호 벡터의 탐색 과정에서 복수 특징들을 동시에 고려하기 위해 제안 기법은 특징 값들의 조합에 대한 참조 표(LUT)를 구성한다. 참조 표에는 부호 벡터의 탐색 순서와 제곱 부분 거리(squared partial distance, SPD)가 크기 순으로 저장된다. 여기서 SPD는 SED 연산 전에 수행되는 사전 검사에서 현재 입력 벡터에 대한 부호화 중지 여부를 결정하는데 활용된다.

본 논문에서는 제안 기법에 대한 빠른 이해를 위해 2개의 특징을 사용하는 경우에 대해서만 기술하고자 한다. 우선 훈련 벡터 집합인 Lena에 대한 WHT 계수 특성 분석(표 1 참조)을 통해 가장 큰 분산을 갖는 W_{00} (DC 계수)와 W_{01} (횡방향의 1차 AC 계수)를 복수 특징으로 선정한다. 또한 W_{00} 와 W_{01} 평면상에서 부호 벡터의 분포를 알고자 LBG 기법^[1]으로 생성한 256개의 부호 벡터를 WHT한 다음 W_{00} 와 W_{01} 에 해당되는 성분 값을 W_{00} 와 W_{01} 평면상에 표시하면 그림 1과 같다. 그림에서 W_{00} 성분이 W_{01} 성분보다 넓게 퍼져 있음을 알 수 있는데,

이는 표 1의 분산 값이 틀리지 않음을 뜻한다.

사용하고자 하는 참조 표의 크기(LUT size)로 W_{00} 와 W_{01} 평면을 나눈 다음, 각 참조 표의 영역 테두리로부터 모든 부호 벡터까지의 SPD를 구하고 오름차순으로 정렬한다. 이때, 참조 표의 깊이(LUT depth)는 부호책의 크기가 되는데, 그 이유는 최악의 경우라도 EFS 기법과 동일한 탐색 결과를 보장하기 위함이다.

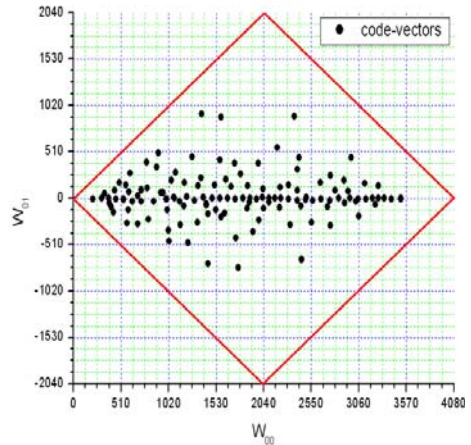


그림 1. W_{00} 와 W_{01} 평면에서 256개의 부호 벡터 분포
Fig. 1. 256 code vectors on the W_{00} and W_{01} plane

3.2.1 메모리 제약이 없는 경우

메모리가 무한정 커서 나누기(\sqrt{k})가 없는 2차원 WHT를 수행하고, W_{00} 와 W_{01} 의 평면상에 존재하는 모든 정수(integer) 점들에 대한 LUT를 구성할 때를 가정해 보자. 이런 경우 W_{00} 와 W_{01} 의 평면에 위치하는 임의의 좌표(α, β)로부터 j -번째 부호 벡터인 c_j^w 까지의 제곱 부분 거리 $SPD(d_{p,j}^w)$ 는 식(2)와 같이 정의될 수 있다.

$$d_{p,j}^w = (\alpha - c_{j1}^w)^2 + (\beta - c_{j2}^w)^2 = \sum_{i=1}^2 (x_i^w - c_{ji}^w)^2 \quad (2)$$

위 식에서 위첨자 w 는 변환 영역을 의미한다. SPD는 해당 부호 벡터의 색인과 함께 LUT[α][β]에 저장되는데, 부호 벡터의 색인과 SPD를 저장하기 위해 필요한 비트 수는 식 (3) 그리고 식 (4)와 같다.

$$L_i = \log_2(N) \quad (3)$$

$$L_d = \log_2(2 \cdot Y^2 \cdot k^2) \quad (4)$$

위 식에서 N 은 부호책 크기를, Y 는 픽셀의 표현 가능 범위를, 그리고 k 는 벡터 차원을 의미한다.

참조 표의 크기가 충분히 크다면 제안 기법의 효율은 극대화되는데, 그 이유는 사전 검사 다음에 참조 표의 SPD를 SED 연산에 그대로 사용할 수 있기 때문이다. 하지만, k 가 16, Y 가 255, N 이 256인 경우 참조 표 구성을 위해 요구되는 메모리가 $4081 \times 4081 \times 256 \times (L_i + L_d)$ 로부터 약 16GBytes가 된다. 따라서 참조 표의 크기 축소가 절대적으로 필요하다.

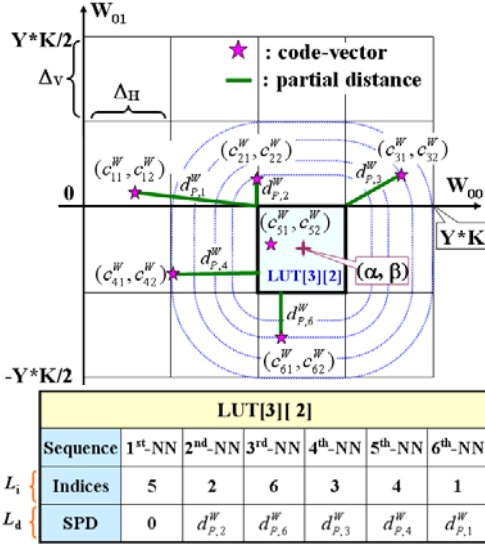


그림 2. 테두리 효과를 고려하는 참조 표 구성의 예
Fig. 2. An example of a LUT configuration according to the boundary effect

3.2.2 메모리 제약이 있는 경우

그림 2는 부호책의 크기가 6이고 참조 표의 크기가 4×4 일 때 테두리(boundary) 효과를 고려하여 LUT[3][2]를 구성하는 예이다. 그림에서 알 수 있듯이 현재 연산하고 있는 참조 표(LUT[3][2])의 테두리 내에 포함되는 5-번째 부호 벡터에 대한 SPD는 0으로 처리된다. 모서리가 둥근 사각형 형태로 표현되는 테두리 효과를 고려함에 따라 식(2)의 간단한 SPD는 식 (5)와 같이 변경된다.

$$d_{P,j}^W = d_{\alpha,j1}^W + d_{\beta,j2}^W,$$

$$\text{where } d_{\alpha,j1}^W = \left(\left| \alpha - c_{j1}^W \right| - \frac{\Delta_H}{2} \right)^2 \text{ if } \left| \alpha - c_{j1}^W \right| > \frac{\Delta_H}{2}$$

$$= 0 \text{ otherwise} \quad (5)$$

$$d_{\beta,j2}^W = \left(\left| \beta - c_{j2}^W \right| - \frac{\Delta_V}{2} \right)^2 \text{ if } \left| \beta - c_{j2}^W \right| > \frac{\Delta_V}{2}$$

$$= 0 \text{ otherwise}$$

식 (5)의 SPD는 SED 연산에 직접 활용할 수 없지만, 현재 입력 벡터에 대한 부호화 종료 여부를 결정하는 사전 검사의 기준으로는 사용될 수 있다. 주목할 점은 식 (2)와 식 (5)에서 알 수 있듯이 SPD 연산은 k -차원의 WHT 계수 중에서 특징으로 선정된 2개의 계수만 고려한다는 점과 그림 2에서 알 수 있듯이 2개의 특징을 위한 참조 표는 3차원 형태로 구성된다는 점이다.

복수 특징을 동시에 고려하는 DFPS 기법을 실행하려면 참조 표가 필요하다. 이때, 참조 표의 크기에 따른 메모리 크기는 표 2와 같다. 제안 기법은 다른 기법에 비해 많은 메모리를 요구하며, 이러한 약점을 보완하는 연구가 진행 중에 있다. 참고로 부호 책의 크기가 256일 때 참조 표 크기가 16×16 이면 TIE 기법과 유사한 메모리 크기가 된다.

표 2. 참조 표 크기에 대한 메모리 크기
Table 2. Memory cost for various LUT size

LUT size	LUT depth	Memory
16 x 16	256 (=CB_size)	256[kB]
32 x 32		1[MB]
64 x 64		4[MB]
128 x 128		16[MB]

3.3 탐색 중지 조건

제안 기법에서 입력 벡터에 대한 부호화 종료 여부를 결정하는 사전 검사 기준은 식 (6)과 같다.

$$d_{E,\min}^W < d_{P,\gamma}^W \leq d_{E,\gamma}^W \quad (6)$$

즉, 현재까지 얻은 최소 거리 $d_{E,\min}^W$ 가 참조 표의 γ -번째 SPD 보다 작으면 참조 표의 γ -번째 색인에 해당되는 부호 벡터와의 SED 연산은 무의미하다. 더욱이 참조 표를 생성할 때 테두리를 기준으로 SPD를 연산하였으므로 참조 표를 탐색하는 중에 부호 벡터가 식 (6)을 만족하면 더 이상의 연산 없이 현재 입력 벡터에 대한 부호화를 종료할 수 있다. 그 이유는 아직까지 사전 검사되지 않은 부호 벡터들은 테두리 기준에서 더 큰 SPD를 갖기 때문이다. 참고로 테두리를 기준으로 SPD를 연산하는 이유는 해당 LUT을 사용하는 입력 벡터는 테두리 안의 임의의 실수 값을 가질 수 있기 때문이다.

3.4 제안 기법의 수행 과정 및 가상 코드

제안 기법은 벡터 양자화의 준비 과정과 부호화 과정으로 구분된다. 준비 과정에서는 벡터 양자화에

필요한 참조 표를 구성하며, 부호화 과정에서는 참조 표를 활용하여 복수 특징들을 동시에 고려하면서 고속 탐색을 수행한다.

3.4.1 준비 과정

단계 ① : 모든 부호 벡터들에 대해 2차원 WHT를 수행한다. 훈련 벡터 집합의 분석을 통해 가급적 분산이 큰 계수가 먼저 SED 연산에 적용될 수 있도록 zig-zag 형태로 계수를 정렬한다.

단계 ② : 선택된 특징인 W_{00} 와 W_{01} 로 구성되는 공간을 참조 표의 크기로 나누고, 테두리 효과를 고려하면서 각 참조 표에 해당되는 영역의 중심점에서 모든 부호 벡터까지의 제곱 부분 거리(SPD)를 연산하여 크기 순으로 정렬한다.

준비 과정을 가상 코드로 기술하면 아래와 같다.

```
Do_2D_WHT_of_CB() //CB: codebook
for(a=0 ; a<(Y*Dk)/ΔH ; a++) { //Dk: dimension
for(β=0 ; β<(Y*Dk)/ΔV ; β++) {
W00=a*ΔH + ΔH/2 + [Min. of W00];
W01=β*ΔV + ΔV/2 + [Min. of W01];
for(m=0 ; m<CB_size ; m++) {
Dα=BE(CB[m].cv[0]-W00, ΔH);
Dβ=BE(CB[m].cv[1]-W01, ΔV);
LUT[α][β].dist[m]=(Dα*Dα)+(Dβ*Dβ);
LUT[α][β].index[m]=m;
} // end of m for-loop
Do_Sort_of_LUT_by_dist(a, β);
} } // end of β and α for-loop
BE(dist, delta) { // BE: boundary effect
if(dist> delta/2) return(dist - delta/2);
if(dist<-delta/2) return(dist + delta/2);
return(0);
} // end of BE function
```

3.4.2 부호화 과정

단계 ③ : 입력 벡터에 대해 단계 ①과 동일한 2차원 WHT를 수행한다. 참고로 4x4 픽셀 크기의 블록을 2차원 WHT 하려면 32번의 덧셈과 32번의 뺄셈이 필요하다.

단계 ④ : 단계 ③의 결과인 $(x_1^w, x_2^w, \dots, x_k^w)$ 에서 복수 특징 값에 해당되는 (x_1^w, x_2^w) 이 포함되는 참조 표를 찾는다. 예를 들어 식 (7)과 식 (8)이 성립되는 경우에는 LUT[u][v]가 선택된다.

$$\Delta_H \cdot (u - 1) \leq x_1^w < \Delta_H \cdot u \quad (7)$$

$$\Delta_V \cdot (v - 1) \leq x_2^w < \Delta_V \cdot v \quad (8)$$

단계 ⑤ : 입력 벡터 \mathbf{x}^w 와 단계 ④에서 선택된 참조 표(LUT[u][v])의 첫 번째 근접 부호 벡터인 \mathbf{c}_{1-NV}^w 에 대한 SED 연산을 식 (9)와 같이 수행한다. 그리고 rank pointer(γ)를 1로 설정한다.

$$d_{E,j}^w = d(\mathbf{x}^w, \mathbf{c}_j^w) = \sum_{i=1}^k (x_i^w - c_{ji}^w)^2 \quad (9)$$

단계 ⑥ : 식 (6)에 따라 현재까지 구한 최소 거리($d_{E,\min}^w$)가 LUT[u][v]의 γ -번째 SPD($d_{p,\gamma}^w$)보다 작으면 현재 입력 벡터에 대한 부호화를 종료하고 단계 ③으로 간다.

단계 ⑦ : 식 (9)를 사용하여 입력 벡터와 γ -번째 부호 벡터 간에 SED 연산을 구한다. 이때, 탐색 속도를 향상하기 위하여 PDE 기법이 적용되며, 만약 SED 연산의 누적 거리가 현재까지 구한 최소 거리보다 크면 SED 연산을 중단하고 γ 에 1을 더한 다음에 단계 ⑥으로 간다.

단계 ⑧ : 현재의 최소 거리를 갱신하고, 최근접 부호 벡터의 색인도 갱신한다. γ 가 부호책 크기(N)와 같으면 단계 ③으로 가고, 그렇지 않으면 γ 에 1을 더한 다음 단계 ⑥으로 간다.

하나의 입력 벡터에 대한 부호화 과정을 가상 코드로 기술하면 아래와 같다.

```
Do_2D_WHT_of_nth_input_vector (n);
u=(InV[n].cv[0] - [Min. of W00]) / ΔH;
v=(InV[n].cv[1] - [Min. of W01]) / ΔV;
min_I=LUT[u][v].index[0]; min_D=0;
for(k=0 ; k<Dk ; k++) //Dk: dimension
min_D+=pow(InV[n].cv[k]-CB[min_I].cv[k], 2);
for(γ=1 ; γ<CB_size ; γ++) {
ii=LUT[u][v].index[γ];
SPD=LUT[u][v].dist[γ];
if(SPD>min_D) break; //terminate encoding
// else Do_Update_minimum_SED
for(k=0, D=0 ; k<Dk ; k++) {
D+=pow((InV[n].cv[k]-CB[ii].cv[k]), 2);
if(D>min_D) break; // PDE
} // end of k for-loop
if(min_D>D) {
min_D=D; min_index=ii;
} // end of if-loop
} // end of γ for-loop
```

IV. 시뮬레이션 결과

제안 기법에 대한 시뮬레이션은 Pentium-IV PC에서 수행되었다. 기존 기법과 연산량을 비교하기 위해 앞서 II장에서 설명한 보편적인 방법으로 실험을 수행하였다. 즉, 훈련 벡터 집합으로 512x512 픽셀 크기의 Lena 영상을 사용하였고, 4x4 픽셀 크기로 영상을 나누어 16-차원의 벡터를 구성하였다. 초기 부호 벡터를 split 방법으로 선정하였고, LBG 기법^[1]을 사용하여 크기가 256인 부호책을 생성하였다. 마지막으로 Lena를 시험 영상으로 사용하였다.

표 3은 부호책 크기가 256일 때 연산량을 비교한 결과이며, 결과 수치에는 각 기법에서 요구되는 이진 탐색이나 WHT 변환을 위한 연산이 포함되었다. 기존 기법 중에서 Extend MOS^[9], SIEENNS^[10], ISIEENNS^[11], L₂NP^[12], M-L₂NP^[13], DHSS(DCT)^[14], 및 IDWT^[15] 기법은 시뮬레이션 없이 해당 논문의 제시 값을 그대로 인용하였다. 표에서 PDE-I과 MPS-I은 픽셀 영역을, PDE-II와 MPS-II는 변환 영역을 의미한다. 또한, 괄호 안의 숫자는 LUT의 크기인데, 예를 들어 (128x128)는 4081x 4081인 W₀₀와 W₀₁의 평면을 128x128로 나누는 것을 의미한다.

시뮬레이션 결과로부터 참조 표의 크기가 커질수록 제안 기법의 성능이 향상됨을 알 수 있다. EFS

기법과 비교할 때 제안 기법은 2개의 특징을 동시에 고려함으로써 화질 저하 없이 연산량을 EFS 기법의 2.2~3.3% 수준까지 감소할 수 있다. 이러한 성능은 현재까지 제안된 기존 기법과 비교할 때 최대 30%까지 연산량을 절약하는 결과이다.

V. 결론

본 논문에서는 일부 WHT 계수를 복수 특징으로 선택하고 부호 벡터의 색인과 해당 부호 벡터까지의 제곱 부분 거리(SPD)를 참조 표에 저장한 다음, 식 (6)의 탐색 중지 조건을 사용하여 고속 부호화를 수행하는 새로운 탐색 기법을 제안하였다. 그리고 시뮬레이션을 통해 복수 특징들을 동시에 고려하는 DFPS 기법이 기존 기법을 능가함을 보였다. 표 3에서 볼 수 있듯이 EFS 기법과 비교할 때 기존 기법은 연산량을 최대 3.0% 수준으로 감소시키는데 반해 DFPS 기법은 가용한 메모리 크기에 따라 연산량을 2.2% 이하로 감축할 수도 있다. 이러한 연산량은 M-L₂NP 기법 등과 같이 최근에 제안된 기법들이 요구하는 연산량의 70% 수준에 해당된다.

부호 벡터의 고속 탐색을 위해 새로운 구조를 갖는 DFPS 기법은 활용성이 높을 것으로 기대되는데, 가령 W₀₀대신 W₁₀을 사용함으로써 MPS 기법과 MAV 기법의 성능 저하가 발생하는 평균이 제거된 벡터 양자화에도 적용될 수 있다. 또한, 메모리가 충분하게 제공된다면 제안 기법은 3개 이상의 특징을 동시에 고려함으로써 탐색 성능을 더욱 향상시킬 수도 있다. 제안 기법의 유일한 약점은 사용 메모리가 다른 기법에 비해 많이 필요하다는 점이며, 현재 이에 대한 연구를 수행 중에 있다.

참 고 문 헌

- [1] Y. Linde, A. Buzo, R. Gray, "An Algorithm for Vector Quantizer Design," *IEEE Trans. Commun.*, 28, pp. 84-95, Jan. 1980.
- [2] C. Bei, R. Gray, "An Improvement of the Minimum Distortion Encoding Algorithm for Vector Quantization," *IEEE Trans. Commun.*, 33, pp. 1132-1133, Oct. 1985.
- [3] S. Chen, J. Pan, "Fast Search Algorithm for VQ-based Recognition of Isolated Words," *IEE Proceedings*, 136, pp. 391-396, Dec. 1989.
- [4] S. Choi, S. Chae, "Incremental-search Fast

표 3. 시뮬레이션 결과
Table 3. Simulation Result

기법	x	+/-	Comp	÷	SQRT	Total	
EFS	256.00	496.00	15.94	0	0	767.94	100.0%
PDE-I	38.24	60.48	53.18	0	0	151.90	19.78%
PDE-II	24.28	36.57	39.22	0	0	100.07	13.03%
MPS-I	9.63	16.81	8.02	0	0	34.46	4.49%
MPS-II	5.63	14.09	5.13	0	0	24.85	3.24%
MAV	9.32	16.69	6.29	0.06	0.06	32.30	4.21%
OHTPDS	6.64	14.71	5.21	0	0.06	26.56	3.46%
Extend MOS ^[9]	8.80	20.40	3.40	0	0	32.60	4.24%
SIEENNS ^[10]	11.01	20.21	4.32	0	0.19	35.73	4.64%
ISIEENNS ^[11]	8.45	16.04	2.75	0.06	0	27.31	3.55%
L ₂ NP ^[12]	9.40	12.20	4.99	0	0.31	26.90	3.49%
M-L ₂ NP ^[13]	8.09	14.54	2.49	0	0.31	25.43	3.30%
DHSS(DCT) ^[14]	4.22	10.70	8.53	0	0	23.45	3.05%
IDWT ^[15]	5.38	13.58	8.02	0	0.18	27.16	3.54%
DFPS (16x16)	5.54	13.67	6.01	0.13	0	25.22	3.28%
DFPS (32x32)	4.24	11.61	4.18	0.13	0	20.03	2.61%
DFPS (64x64)	3.62	10.65	3.28	0.13	0	17.55	2.29%
DFPS (128x128)	3.35	10.24	2.87	0.13	0	16.46	2.14%

- Vector Quantiser using Triangular Inequalities for Multiple Anchors,” *IEE Electron. Lett.*, 34, pp. 1192-1193, June 1998.
- [5] S. Ra, J. Kim, “A Fast Mean-distance- ordered Partial Codebook Search Algorithm for Image Vector Quantization,” *IEEE Trans. Circuits and Systems II: Analog and Digital Signal Processing.*, 40, pp. 576-579, Sept. 1993.
- [6] C. Lee, L. Chen, “Fast Closest Codeword Search Algorithm for Vector Quantization,” *IEE Proc. Vision, Image and Signal Processing*, 141, pp. 143-148, June 1994.
- [7] S. Baek, B. Jeon, K. Sung, “A Fast Encoding Algorithm for Vector Quantization,” *IEEE Signal Processing Lett.*, 4, pp. 325-327, Dec. 1997.
- [8] Z. Lu, D. Xu, S. Sun, “Fast Codeword Search Algorithm for Image Vector Quantization based on Ordered Hadamard Transform,” *IECE Trans. Inf. & Syst.*, E86-D, pp. 1318-1320, July 2003.
- [9] S. Choi, S. Chae, “Extended Mean-distance-ordered Search using Multiple L1 and L2 Inequalities for Fast Vector Quantization,” *IEEE Trans. C. & Systems II: Analog and Digital Signal Processing*, 47, pp. 349-352, Apr. 2000.
- [10] J. Pan, Z. Lu, S. Sun, “An Efficient Encoding Algorithm for Vector Quantization based on Sub-vector Technique,” *IEEE Trans. Image Processing*, 12, pp. 265-270, March 2003.
- [11] Z. Pan, K. Kotani, T. Ohmi, “Improved Fast Encoding Method for Vector Quantization based on Subvector Technique,” *IEEE International Symposium on Circuits and Systems(ISCAS)*, 2005, pp. 6332-6335, May 2005.
- [12] B. Song, J. Ra, “A Fast Search Algorithm for Vector Quantization using L2-norm Pyramid of Codewords,” *IEEE Trans. Image Process.*, 11, pp. 10-15, Jan. 2002.
- [13] Z. Pan, K. Kotani, T. Ohmi, “Fast Encoding Method for Vector Quantization using Modified L₂-Norm Pyramid,” *IEEE Signal Process. Lett.*, 12, pp. 609-612, Sept. 2005.
- [14] S. Tai, C. Lai, Y. Lin, “Two Fast Nearest Neighbor Searching Algorithms for Image Vector Quantization,” *IEEE Trans. Commun.*, 44, pp. 1623-1628, Dec. 1996.
- [15] Z. Pan, K. Kotani, T. Ohmi, “Improved Fast Search Method for Vector Quantization using Discrete Walsh Transform,” *International Conference on Image Process. (ICIP)*, 2004, 5, pp. 3177-3180, Oct. 2004.
- [16] 최지웅, 나성웅, “동영상의 차분이미지 부호화를 위한 고속 벡터양자화 알고리즘”, *한국통신학회 학술발표 논문집*, 17(3), pp.467-470, 1998.
- [17] 정일화, 최인호, 이대영, “벡터양자화의 고속 인코딩과 인덱스 무손실 압축”, *한국통신학회 논문지*, 27(9A), pp.925-934, 2002.
- [18] C. Ryu, S. Ra, “A Fast Full Search Equivalent Encoding Algorithm for Image Vector Quantization based on the Walsh-Hadamard Transform and a Look-up Table,” *IEEE Nonlinear Signal and Image Processing*, Abstract, pp.23-23, May 2005.
- [19] C. Ryu, S. Ra, “A Fast Full Search Equivalent Encoding Algorithm for Image Vector Quantization based on the WHT and a LUT,” *The 5th IEEE International Workshop on System-on-chip for Real-Time Applications*, 1, pp.405-409, July 2005.

류 철 형 (Chul-hyung Ryu)

정회원



1988년 2월 충남대학교 전자공학과 졸업
 1990년 2월 충남대학교 전자공학과 석사
 1990년 3월~현재 국방과학연구소 선임연구원
 2000년 3월~현재 충남대학교 전자공학과 박사과정

<관심분야> 실시간 영상압축/전송, 무인 시스템 원격 제어, 소프트웨어 공학

나 성 웅 (Sung-woong Ra)

정회원



1976년 2월 서울대학교 전기공학과 졸업
 1978년 2월 한국과학기술원 전기 및 전자공학과 석사
 1992년 8월 한국과학기술원 전기 및 전자공학과 박사
 1980년 7월~현재 충남대학교 공과대학 전자공학과 교수

<관심분야> 영상처리, 영상코딩, MPEG-IV, 화상 회의 시스템