



## 1. 서론

H.264는 다양한 전송 환경에서의 범용 멀티미디어 서비스를 위해 수Kbps에서 수Mbps까지의 광범위한 채널대역폭에 있어서 높은 부호화 효율을 갖도록 표준화가 진행 중이다. 따라서 특정 응용을 위해 일정 채널대역폭에 맞도록 부호화하였던 기존 표준들과 달리 광범위 채널대역폭에 대응할 수 있는 효과적인 새로운 부호화 방법이 필요하다. H.264 부호화 기술이 기존 H.263이나 MPEG-4 표준에 비해 높은 부호화 효율을 갖는 것은 다중 참조 영상 부호화, 다양한 블록 크기의 움직임 보상 및 1/4 또는 1/8 화소 단위의 움직임 보상 등의 움직임 보상에 관련된 새로운 기술들과 더불어, 간단하면서도 효율적인 엔트로피 부호화 기술이 있기 때문이다[1]. H.264의 엔트로피 부호화 기술은 크게 Exponential Golomb 부호어를 이용한 구조화된 가

변장 부호화(Structured VLC) 기법과 내용기반 적응적 2진 산술 부호화(Context-based Adaptive Binary Arithmetic Coding: CABAC) 기법으로 구성되어 있다. 여기서 구조화된 가변장 부호화 기법은 모든 심볼에 Exponential Golomb 부호어를 사용하며, 특히 변환계수값에 대해서 부호화 효율을 좀 더 높이기 위해 내용기반 적응적 가변장 부호화 (Context-Adaptive VLC: CAEVL)라는 새로운 부호화 기법을 사용한다.

기존 H.263 및 MPEG-1/2/4의 표준들의 엔트로피 부호화 방법으로 사용되어 왔던 허프만 가변장 부호화 기법은, 기본적으로 부호화되는 심볼의 redundancy를 줄이기 위해 발생빈도가 큰 심볼에는 짧은 부호어를, 발생 빈도수가 적은 심볼에는 긴 부호어를 부여하는 방식이다. 이 방법은 특히, 주어진 심볼의 통계에 기반하여 최적의 부호화 효율을 갖도록 부호어를 배정할 수 있는 개념의 간략성 및

알고리듬 구현의 상대적 용이성 때문에 일찍부터 많이 사용되어 왔다. 이러한 허프만 부호화 방식의 이론적 배경은 다음과 같다[2]. 한 심볼  $S_i$ 의 확률값을  $P_i$ 라고 가정하면,  $S_i$  각 심볼들이 갖는 정보량은 다음과 같이 나타낼 수 있다.

$$I(S_i) = \log_2 \frac{1}{P_i} = -\log_2 P_i \quad (1)$$

이때, 심볼  $S_i$ 가 갖는 평균 정보량  $H$ 는 식(2)와 같이 계산되고 이것이 심볼  $S_i$ 가 갖는 평균 부호어 길이이며, 특히, 심볼  $S_i$ 를 2진수를 이용하여 표현할 경우 이는  $S_i$ 를 표현하기 위한 최소한의 비트 수가 된다.

$$H = \sum P_i I(S_i) = -\sum P_i \log_2 (P_i) \quad (2)$$

또한, 심볼  $S$ 를 랜덤변수로 보면 랜덤변수  $S_i$ 의 각 정보량의 기대값  $E\{I\}$ 가 바로 엔트로피가 되는 것이다. 심볼을 표현하는 평균 부호어 길이가 그 심볼의 엔트로피에 가까워질수록 부호화 효율은 높아진다. 허프만 부호어의 경우에 평균 부호어의 길이와 심볼의 엔트로피는 다음의 관계식을 갖는다.

$$H(S) \leq R < H(S) + 1 \quad (3)$$

그러므로 허프만 부호어의 길이는 평균적으로 심볼의 ‘엔트로피+1’ 비트보다 항상 작기 때문에 최적의 성능을 갖게 된다. 또한 허프만 부호는 prefix 조건을 만족하기 때문에 부호어는 언제나 고유한 부호어를 가지고 있어서 복호시 다른 부호어와 겹치는 경우가 없다.

그러나, 이러한 허프만 부호화 기법은 반드시 심

볼의 확률분포를 알아야만 최적의 부호어를 설계할 수 있으므로 실제 심볼의 통계치를 분석하는 과정과 이러한 통계치에 대하여 부호어를 생성하는 두 과정이 필요하다. 이렇게 실제 심볼의 통계적 분포를 이용하는 방법은 많은 계산량을 필요로 할 뿐만 아니라 영상 부호화 시, 통계치 또는 부호어 테이블을 따로 전송해주어야 하기 때문에 엔트로피 부호화에 그대로 적용하기 어렵다. 따라서 실제 영상 부호화에 적용될 때에는 다양한 영상에 대하여 실험적으로 얻은 심볼의 통계적 특성에 기반하여 얻어진 부호어를 테이블에 저장하여 사용하는 기법을 이용한다. 이렇게 실험적으로 예측된 심볼의 통계값을 가지고 생성한 부호어를 부호기 및 복호기에 테이블로 저장하여 사용하기 때문에 당연히 실제 통계값을 이용하는 경우보다 부호화 효율이 떨어지기 마련이다. 이러한 통계적 특성의 예측에서 발생하는 부호화 효율 저하 문제를 해결하기 위한 노력으로 동적 허프만 부호화 기법이 연구되기도 하였다[3]~[6]. 그러나 이러한 방법들은 복호기에서의 계속적인 에러 전파가 일어날 수 있는 약점을 가지고 있다.

기존 H.263 영상 부호화 표준의 경우에는 56Kbps 이하 초저속 전송 채널에서의 응용을 주 목표로 하였기 때문에 가변장 테이블을 이용한 허프만 부호화 기법을 적용하더라도 엔트로피 부호화 효율이 크게 떨어지지 않았다. 그러나 H.264의 경우에는 앞에서 언급한 것처럼 보다 더 광범위한 비트율을 대상으로 하기 때문에 특정 비트율에 최적화된 부호화 기술을 사용할 수가 없다. 실제 심볼의 통계적 특성은 여러 가지 요소에 따라 달라질 수 있지만 그 가운데 중요한 변수로 작용하는 것이 바로 비트율이다. 그러므로 다양한 비트율에서 최적의 부호화 방법을 원한다면 지금까지 사용되어온 고정

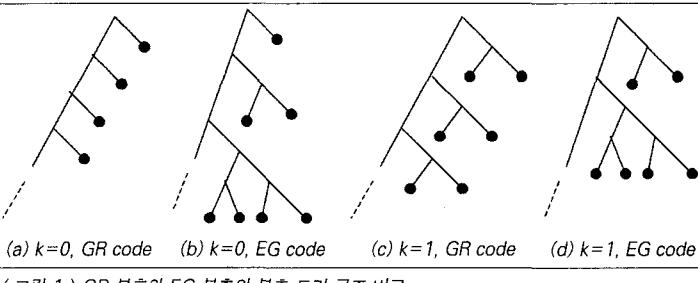
된 헤프만 테이블은 그 해답이 될 수 없다. 굳이 헤프만 테이블을 이용한 가변장부호화 기법을 적용한다면, 각 비트율에 따라 헤프만 테이블을 달리 만들어 적용할 수는 있으나 그 만큼의 많은 메모리를 요구하게 되

고 하드웨어 구현이 더 복잡해지는 단점이 생기게 된다. 그러므로 광범위한 채널 대역폭에 대해 높은 부호화 효율을 갖는 새로운 엔트로피 부호화 기법이 요구되었고 H.264는 “Exponential-Golomb” 부호화 기법과 내용기반 적응적 가변장 부호화 기법(context-adaptive VLC)을 사용하게 되었다.

## 2. Exponential Golomb 부호화 기법

“Golomb-Rice(GR)” 부호화 기법은 일반적으로 무손실 영상 압축 응용에서 예측 에러 부호화에 주로 쓰이던 방법이다[7][8]. GR 부호화는 심볼의 확률분포가 기하분포를 갖는 경우에 최적화 되도록 만들어져 있는데, 이 기법은 정수  $n$ 을 뜻  $q$ 와 나머지  $r$ 로 표현하는 방식으로  $k$ -GR 부호어라고도 한다. 이때, 특별히  $2^k$ 의  $k$ 승으로 나눈 뜻을 unary 부호어로, 나머지 값은  $k$ 비트만큼 binary 부호어로 표현한다. 예를 들어  $k=2$ 인 GR 부호어의 경우, 숫자 11은 뜻  $q=2$ (‘110’), 나머지  $r=3$ (‘11’)이 되므로 최종 부호어는 ‘11011’가 된다. GR 부호화 기법은 헤프만 부호화 기법처럼 심볼의 통계적 특성에 따라 부호어를 배정하는 것은 아니지만 심볼의  $k$ 값을

적절히 조절함으로써 기하분포의 확률모델에서 다양한 분산값의 변화에 적절히 대응할 수가 있는 장점이 있어 다양한 비트율을 대상으로 하는 H.264에 잘 맞는다. 이러한 GR 부호화 기법에서 그 길이가 동일한 부호어의 수가  $k$ 값에 따라 일정한 것과는 달리, “Exp-Golomb(EG)” 부호화 기법에서는 길이가 같은 부호어의 수가 기하급수적으로 증가한다. GR 부호와 EG 부호에 있어서 생성규칙의 차이점을 알아보기 위해 각 부호 트리를 그림 1에 나타내었다. <그림 1>에서 보는 바와 같이 부호어 길이가  $l$ 인 심볼 그룹을  $S_l$ 이라고 할 때,  $S_l$ 에 존재하는 심볼의 개수가 GR 부호에서  $2^k$ 개씩 존재하지만 EG 부호에서는  $2^{\left[\frac{l-1}{2}\right]}$ 개씩 존재하고 있다. <표 1>은 이러한 각 부호 기법으로 인해 발생되는 부호어의 예를 나타낸 것이다.



&lt;그림 1&gt; GR 부호와 EG 부호의 부호 트리 구조 비교

<표 1>  $k=0, k=1$ 인 경우의 GR 부호와 EG 부호

k = 0				k = 1			
GR		EG		GR		EG	
prefix	suffix	prefix	suffix	prefix	suffix	prefix	suffix
0	0	0		0	0	0	0
1	10		100		0	1	
2	110		101		10	0	100
3	1110		11000		10	1	100
4	11110		11001		110	0	101
5	111110		11010		110	1	101
6	1111110		11011		1110	0	11000
7	11111110		1110000		1110	1	11000
...	...	...	...	...	...	...	...

EG 부호는 입력 심볼이 기하분포의 형태를 가지면서도 확률값의 decay rate가 알려지지 않았거나 부정확하게 예측된 경우에 효과적인 부호화 기법을 위해 Teuhola에 의해 처음으로 제안되었다[9]. 이러한 EG 부호는 특히 일정한 표준편차를 갖는 기하분포 모델보다 상대적으로 높은 첨두치를 갖고 긴 꼬리(tail) 분포를 갖는 기하분포 모델에 더 적합한 것으로 나타났다. 특히, 이러한 확률모델은 실제로 양자화후 줄길이 부호화(Run-length coding)된 변환계수 심볼들의 확률분포에 잘 맞는다. 그러므로 이러한 EG 부호화 기법을 줄길이 부호나 DPCM된 여러 영상 부호에 적용할 경우 매우 간단한 구조로 높은 부호화 효율을 갖게 된다. 또한 역방향 가변장부호화(Reversible VLC)로 쉽게 변환될 수 있기 때문에 여러 내성 부호화에도 적용할 수가 있다[10].

1
0 1 $x_0$
0 0 1 $x_1 x_0$
0 0 0 1 $x_2 x_1 x_0$
0 0 0 0 1 $x_3 x_2 x_1 x_0$
.....

〈그림 2〉 H.264에서의 부호어 발생 규칙

H.264에서는 기본적으로 〈그림 2〉와 같은 발생 규칙을 갖는 EG 부호어를 모든 H.264 부호화 요소(syntax element)의 엔트로피 부호화에 적용하고 있다[1].

단, 줄길이 부호된 변환계수의 경우에는 좀 더 높은 부호화 효율을 위해 내용 기반 적응적 부호화(Context-Adaptive VLC) 기법을 이용한 변형된 EG 부호화를 사용하고 있다. 그림 2에서  $x_i$ 은 0 또는 1의 값을 갖는 2진수이며  $x_n x_{n-1} \dots x_0$ 의 값은

〈표 2〉 H.264에서의 코드 번호와 부호어와의 관계

코드 번호	부호어
0	1
1	0 1 0
2	0 1 1
3	0 0 1 0 0
4	0 0 1 0 1
5	0 0 1 1 0
6	0 0 1 1 1
7	0 0 0 1 0 0 0
8	0 0 0 1 0 0 1
9	0 0 0 1 0 1 0
10	0 0 0 1 0 1 1
.....	.....

INFO라고 한다. 이러한 발생규칙을 가지고 실제 발생되는 부호어는 〈표 2〉에 나타난 것과 같다. 〈표 2〉에서의 부호어들은  $k=0$ 인 EG 부호임을 쉽게 알 수 있다. 그러므로 모든 심볼 부호화에  $k=0$ 의 EG 부호를 사용하는 H.264에서는, 실제 발생하는 심볼의 통계적 특성이 기하분포 모델에 가까워지도록 부호화할 때 높은 부호화 효율을 얻을 수 있다. 이러한 이유로 H.264는 손실압축 알고리듬을 설계한 후 출력 심볼의 통계적 특성을 이용하여 엔트로피 부호화 방법을 설계하던 전통적인 설계순서 대신, 손실압축 알고리듬의 출력을 이미 설계된 엔트로피 부호화 방법이 가정한 입력 통계성질에 잘 맞도록 손실압축 알고리듬 자체를 변경하는 과정을 거쳐왔다.

〈그림 1〉의 부호 트리에서 왼쪽을 0, 오른쪽을 1로 정의하게 되면 〈표 1〉과 같은 부호어가 생성되고, 그 반대로 정의하면 〈표 2〉와 같이 생성된다. 이러한 EG 부호어는 다음과 같은 발생규칙이 존재하기 때문에 허프만 부호처럼 심볼과 부호어간의 관계를 나타내는 별도의 테이블이 필요 없다. 즉, 심볼의 부호길이가  $l$ 이라고 가정하면 심볼의 코드 번호  $C$ 는

$$C = 2^{\lfloor l/2 \rfloor} + \text{INFO} - 1 \quad (4)$$

가 된다. 단,  $l=1$ 인 경우에는 INFO=0으로 하여 코드 번호를 구한다. 이 식은 비트스트림을 가지고 심볼을 복호화하는 과정에도 동일하게 사용될 수 있다. 복호기에 입력된 비트스트림에서 먼저 0으로 만 이루어진 1진수 부분을 읽어낸 다음, 그 1진수 값만큼의 비트들을 추가로 읽어내면 이것이 바로 2진 INFO값이 된다. 또한 지금까지 읽어낸 비트들의 개수가  $l$ 에 해당하기 때문에 쉽게 INFO값과 부호길이  $l$ 을 복호할 수가 있다. 이렇게 구한 INFO와  $l$ 을 가지고 심볼의 코드 번호를 식(4)를 통해 계산한 후 이 코드 번호가 의미하는 각 부호화 요소를 찾으면 엔트로피 복호과정이 끝나는 것이다. 기존 허프만 부호를 이용한 가변장 복호화에 필요한 테이블 탐색이 없기 때문에 복호 과정이 매우 간단해졌음을 알 수 있다. 또한 이의 역과정인 엔트로피 부호화 과정도 간단하다. 코드 번호  $C$ 가 주어졌을 때 부호길이  $l$ 과 INFO는 각각 다음과 같다.

$$l = \lfloor 2 \log_2 \{ C + 1 \} \rfloor + 1 \quad (5)$$

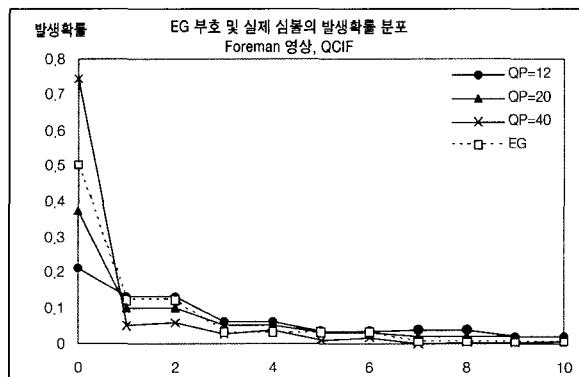
$$\text{INFO} = C - 2^{\lfloor l/2 \rfloor} + 1 \quad (6)$$

부호길이  $l$ 은 INFO값에 영향을 받지 않기 때문에 식(4)에서 INFO=0으로 하여 식(5)를 얻을 수가 있다. 또한 이렇게 얻은 부호길이  $l$ 과 코드 번호  $C$ 를 통해 INFO값을 구하면 간단히 비트스트림을 생성해 낼 수가 있다. 그러므로 이러한 관계식을 통해 몇 가지 덧셈과 쉬프트 연산으로 엔트로피 부호화 및 복호화가 가능하기 때문에 메모리나 계산량 측면에서 하드웨어 구현이 매우 유리하다고 볼 수 있다.

### 3. 내용기반 적응적 가변장 부호화 (Context-Adaptive VLC) 기법

GR부호나 EG 부호는 기하분포를 갖는 심볼에 적용될 수 있는 부호이기 때문에 실제 심볼의 통계적 특성이 기하분포 모델에서 멀어지게 되면 부호화 성능은 떨어질 수밖에 없다. 한편, 양자화되고 줄길이 부호화된 변환 계수는 일반적으로 라플라시안 분포 모델을 갖는다고 알려져 있다[11]. 라플라시안 분포 모델의 경우에는 기하분포에 근사화 될 수 있으며 특히 높은 첨두치와 긴 꼬리(tail)분포를 갖는 EG 부호의 확률모델에 잘 맞는다고 볼 수 있다. 그러나 이러한 라플라시안 분포는 실제로 다른 요소들에 의해 많은 영향을 받는데 그 가운데 가장 영향을 많이 받는 것이 바로 양자화 계수(QP)이다 [11][12]. QP는 영상의 부호화 시 비트율을 조절하는 주요 성분으로써, 각 비트율에 따른 부호화 요소의 통계적 특성을 좌우한다. <그림 3>은 각 QP에 따른 줄길이 부호의 통계적 특성을 나타낸 것이다.

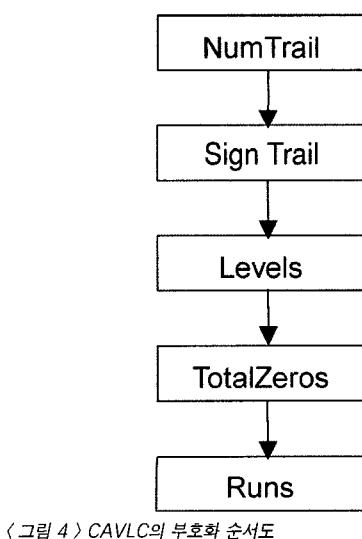
<그림 3>에서 보는 바와 같이 QP의 크기가 작아질수록 0에 가까운 코드 번호의 확률값이 작아져서 EG 부호의 확률분포 모델에서 멀어지는 것을 볼



<그림 3> QP에 따른 EG 부호 및 줄길이 부호화된 Y신호의 변환계수 확률분포 비교

수가 있다. 즉, 높은 비트율로 갈수록 기하분포 모델에서 멀어지게 되고 EG 부호의 효율도 멀어지게 되는 것을 볼 수가 있다. 따라서 변환계수의 줄길이 부호 시 높은 비트율에서의 엔트로피 부호화 성능 개선을 위한 대책이 요구된다. 이러한 문제를 해결하기 위해 H.264에서는 내용기반 적응적 가변장부호화(Context-Adaptive VLC: CAVLC) 기법을 변환계수 부호화에 적용한다[1].

CAVLC 기법은 기존 EG 부호만으로는 고비트율에서 발생하는 심볼의 통계적 특성에 적용될 수 없기 때문에 몇 가지 변형된 EG부호 테이블을 사용하여 실제 심볼의 통계적 특성에 적응시키는 방법을 사용한다. 또한 주변 블록이나 QP, 바로 이전 부호화된 심볼의 값등을 이용해 적응적 부호화를 함으로써 부호화 효율을 좀 더 향상시킬 수가 있다. 여기서 이러한 적응적 능력을 높이기 위해 기존의 2차원 (또는 3차원) 줄길이 부호화 방식에서 벗어나 줄길이(run)와 레벨(level)을 각각 분리하여 부호화한다. 이러한 특성을 갖는 CAVLC 기법의 부호화 순서도는 <그림 4>에 나타난 것과 같다.



## 1. NumTrail 및 SignTrail 부호화

먼저 지그재그방향으로 주사된 run과 level값들을 주사방향의 역순으로 부호화한다. 역순으로 부호화하는 이유는 실제 통계적 특성에 대한 적응성을 높이기 위함이다. <그림 3>에서도 알 수 있듯 줄길이 부호 시 코드 번호가 큰 부분에서는 QP에 관계없이 대부분 0에 가까운 값을 갖는 것을 볼 수 있다. 이것은 변환계수의 특성상 높은 주파수 영역으로 갈수록 level값들은 대부분 0에 가까운 값을 많이 갖기 때문이다. 변환계수는 저주파 영역으로 갈수록 QP값에 영향을 많이 받기 때문에 부호화 순서를 고주파 영역에서 저주파 영역의 순서로 부호화하면서 점차적으로 실제 통계적 특성에 적응해 나가기 위해 역방향 지그재그 순서로 부호화하는 것이다.

먼저 역방향으로 level값들을 부호화하게 되면 통계적 성질상 level값들은 대부분 ±1의 값을 갖게 되기 때문에 ±1에 대한 절대값 길이를 부호화하여 전송한다. 이러한 ±1에 대한 절대값 길이를 Trailing 1's(T1's)라고 하며, 총 level의 개수와 함께 부호화하여 전송하게 되는데 이 부호화요소를 NumTrail이라고 한다. 그리고 1의 부호(sign) 신호에 대해서 추가로 SignTrail 정보를 전송한다. T1's는 최대 3개까지로 제한하여 부호화하며 총 level 개수와 함께 NumTrail로 부호화할 때 주변 블록의 총 level 수에 따라 3가지 종류의 가변장부호화 테이블을 선택적으로 전환하여 부호화한다. 즉, 주변 블록의 총 level 수에 따라 현재 블록의 총 level 수를 어느 정도 예측할 수가 있기 때문에 주변 윗쪽과 윗쪽 블록을 이용하여 적응적으로 가변장 테이블을 바꾸면서 부호화하는 것이다. 주변 블록을 이용한 적응적 부호화 방법이 <표 3>에 나타

〈표 3〉 가변장 테이블을 선택하기 위한 조건

있음	있음	$(N_l + N_u)/2$
있음	없음	$N_u$
없음	있음	$N_l$
없음	없음	0

0 ≤ N &lt; 2 : 가변장 테이블 #1

2 ≤ N &lt; 4 : 가변장 테이블 #2

4 ≤ N &lt; 8 : 가변장 테이블 #3

 $N \geq 8$  : 6비트 고정장 부호화 xxxxyy; xxxx는 level의 총

개수-1, yy는 T1's (단, 부호어 000011은 level0이 존재하지 않는 것을 말함.)

나 있다. 〈표 3〉에서 왼쪽 및 윗쪽 블록을 통해 구한 N값을 가지고 현재 부호화에 사용할 VLC 테이블을 선택하게 되는데 만일 N이 8보다 큰 경우에는 고정장 부호화하여 6비트로 전송하게 된다.

## 2. Level값의 부호화

NumTrail 및 SignTrail 부호화가 완료된 후에는 level값들을 부호화하게 되는데 level 값 역시 〈표 4〉에 나타난 것처럼 5개의 테이블을 이용하여 상황에 따라 적응적으로 부호화하게 된다. T1's의 부호화 후 남은 level값들은 먼저 Lev-VLC0 테이블과

〈표 4〉 Level 부호화에 사용되는 테이블들

Lev-VLC0 테이블			
0	1	1	2
1	01	-1	-2
2	001	2	3
3	0001	-2	-3
..	..	..	..
13	00000000000001	-7	-8
14-29	00000000000001xxxx	$\pm 8 \text{ to } \pm 15$	$\pm 9 \text{ to } \pm 16$
30->	00000000000001xxxxxxxxxxxxxx	$\pm 16 \rightarrow$	$\pm 17 \rightarrow$

Lev-VLC1 테이블을 이용하여 부호화를 시작한다.

만일 Inter 프레임이거나  $QP \geq 21$ 인 Intra 프레임의 경우에는 처음 부호화되는 level값에 Lev-VLC0를 적용하고 그 다음 level은 Lev-VLC1을 적용한다. 여기서 바로 이전에 부호화된 level값이 3보다 크면

Lev-VLC1 테이블			
0-1	1x	$\pm 1$	$\pm 2$
2-3	01x	$\pm 2$	$\pm 3$
..	..	..	..
26-27	0000000000000001x	$\pm 14$	$\pm 15$
28-43	0000000000000001xxxx	$\pm 15 \text{ to } \pm 22$	$\pm 16 \text{ to } \pm 23$
44->	0000000000000001xxxxxxxxxxxx	$\pm 23 \rightarrow$	$\pm 24 \rightarrow$

Lev-VLC2 테이블			
0-3	1xx	$\pm 1 \text{ to } \pm 2$	
4-7	01xx	$\pm 3 \text{ to } \pm 4$	
..	..	..	..
52-55	0000000000000001xx	$\pm 27 \text{ to } \pm 28$	
56-71	0000000000000001xxxx	$\pm 29 \text{ to } \pm 36$	
72->	0000000000000001xxxxxxxxxxxx	$\pm 37 \rightarrow$	

Lev-VLC3 테이블			
0-7	1xxx	$\pm 1 \text{ to } \pm 4$	
8-16	01xxx	$\pm 5 \text{ to } \pm 8$	
..	..	..	..
104-111	00000000000001xxx	$\pm 53 \text{ to } \pm 56$	
112-127	00000000000001xxxx	$\pm 57 \text{ to } \pm 64$	
128->	00000000000001xxxxxxxxxxxx	$\pm 66 \rightarrow$	

Lev-VLC4 테이블			
0-15	1xxxx	$\pm 1 \text{ to } \pm 8$	
16-31	01xxxx	$\pm 9 \text{ to } \pm 16$	
..	..	..	..
112-127	00000000000001xxxx	$\pm 57 \text{ to } \pm 64$	
128->	00000000000001xxxxxxxxxxxx	$\pm 66 \rightarrow$	

VLC 인덱스를 하나씩 증가시키되 Lev-VLC2까지 만 증가시키도록 한다. 만일 Intra 프레임이면서 양자화 계수가 21 미만인 경우에는 총 level의 개수가 10개일 때 Lev-VLC1부터 시작하여 부호화하고 그렇지 않으면 Lev-VLC0부터 시작한다. 만일 이번 테이블이 Lev-VLC1인 경우에는 이전 부호화된 level의 절대값이 3보다 큰 경우에 한하여 Lev-VLC2로 바뀔 수 있으며, 이번 테이블이 Lev-VLC2인 경우에는 이전 부호화된 level이 5이상인 경우에 한해 테이블을 Lev-VLC4까지 VLC 인덱스를 하나씩 증가시킬 수 있다. 이러한 규칙은 색차 AC 또는 DC 신호에 대해서도 동일하게 적용된다. <표 4>를 보면 VLC 테이블이 Lev-VLC0에서 Lev-VLC4로 갈수록 고비트율에서의 통계적 특성에 맞는 가변장 부호화로 구성되어 있는 것을 볼 수가 있다. 그러므로 고주파 영역에서 시작하여 저주파영역 쪽으로 변환계수를 부호화하면서 현재까지 부호화된 변환계수의 크기에 따라 앞으로 부호화할 변환계수의 크기를 예측해 나가는 방식을 통해 저주파 영역에서의 다양한 통계적 특성에 잘 적응하도록 부호화하고 있다.

### 3. Run 부호화

run의 부호화 방법에서도 역시 먼저 총 run의 개수를 부호화하고(TotalZeros) 지그재그 주사패턴의 역순으로 run값을 부호화한다. 여기서 TotalZeros는 NumTrail에서 부호화한 총 level의 개수를 이용하여 부호화한다. TotalZeros는 “16 - 총 level의 개수”까지 존재하며 총 level의 수에 영향을 받는다. 이렇게 부호화된 TotalZeros를 이용하여 run 값을 부호화하게 되는데 총 run의 개수를 알 수 있으므로 run값들을 부호화하는데 있어서 현재

까지 남은 run의 개수를 알 수 있다. 그러므로 남은 run의 개수에 따른 통계적 성질을 이용하여 run값을 부호화함으로써 부호화 효율을 좀더 높일 수가 있다. 이렇게 남은 run값을 이용하여 가변장 부호화하는 테이블이 <표 5>에 나타나 있다.

### 4. H.264부호화 요소의 통계적 성질

각 비트율에 따른 H.264 부호화 요소들의 통계적 특성을 파악하기 위해 각 부호화 요소별 상대적 발생 비트량을 조사하였으며, 현 H.264에서 사용되는 실험 조건[13]에 따라 <표 6>에 나타난 바와 같이 JM4.0c 부호기를 사용하여 실험하였다.

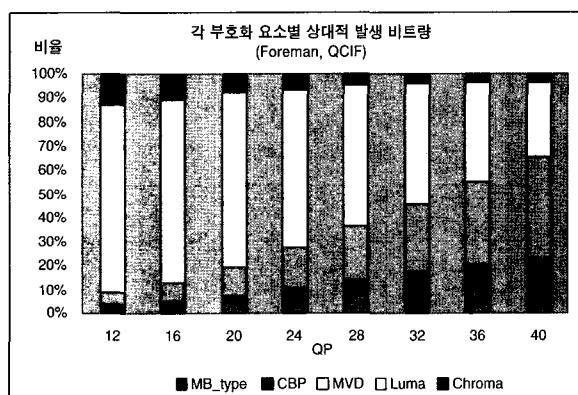
<표 5> 남은 run에 따른 run값 부호화 테이블

Runs Left Run Before	1	2	3	4	5	6	>6
0	1	1	01	01	01	01	000
1	0	01	00	00	00	00	010
2	-	00	11	11	11	101	101
3	-	-	10	101	101	100	100
4	-	-	-	100	1001	111	111
5	-	-	-	-	1000	1101	110
6	-	-	-	-	-	1100	0011
7	-	-	-	-	-	-	0010
8	-	-	-	-	-	-	00011
9	-	-	-	-	-	-	00010
10	-	-	-	-	-	-	00001
11	-	-	-	-	-	-	0000011
12	-	-	-	-	-	-	0000010
13	-	-	-	-	-	-	0000001
14	-	-	-	-	-	-	00000001

<표 6> 실험 조건

	News	Container	Foreman	Silent	Paris	Mobile	Tempete
영상크기	QCIF	QCIF	QCIF	QCIF	CIF	CIF	CIF
프레임률 (Hz)	10	10	10	15	15	30	30
총프레임수	100	100	100	150	150	300	300
QP	12, 16, 20, 24, 28, 32, 36, 40						

〈그림 5〉는 Foreman 영상에 대하여 QP별 부호화 요소의 상대적 발생 비트량을 나타낸 것이다. QP가 커질수록 총 발생 비트량은 감소하게 되며 이에 따른 각 부호화 요소별 상대적 비중도 달라지게 된다. 예를 들어 MB\_type이나 CBP와 같은 매크로블록 헤더 정보들은 QP에 관계없이 비교적 일정하게 나타나는 부호화 요소들이므로 QP 값이 커지면 총 발생 비트량이 적어짐에 따라 상대적 비중이 높아지게 된다. 반면 변환계수의 경우에는 QP 값이 커짐에 따라 발생 비트량이 줄어들기 때문에 상대적으로 차지하는 비중도 또한 감소하게 된다. 저비트율로 갈

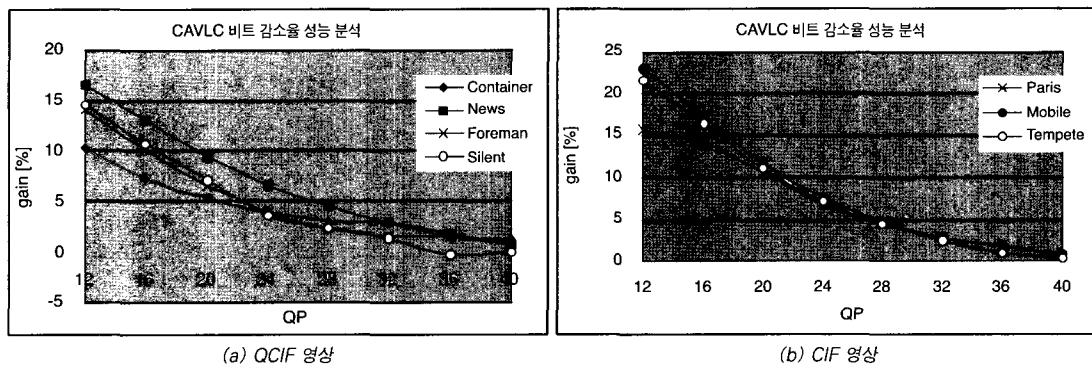


〈그림 5〉 QP에 따른 각 부호화 요소별 상대적 발생 비트량

수록 그 상대적 발생 빈도수도 작아져서 QP가 36 또는 40의 경우에는 심지어 움직임 벡터의 비트량 보다 더 적은 비율을 차지하는 것을 볼 수가 있다.

여기서 알 수 있는 것은 QP 값이 큰 저비트율에서는 변환계수보다도 다른 부호화 요소의 비중이 더 높게 나타나기 때문에 움직임 벡터 정보나 매크로블록 헤더에 해당하는 정보의 엔트로피 부호화가 그만큼 중요하게 작용한다. 그러므로 저비트율에서 주요 비중을 차지하는 부호화 요소들의 경우에는 저비트율에서의 통계에 잘 맞도록 디자인되어 현재 적용되고 있다. 반면 고비트율로 가게 되면 변환계수의 비중이 월등히 높게 나타나게 되므로 변환계수의 부호화 효율이 총 발생 비트량에 가장 큰 영향을 미치게 된다. 이를 위해 내용기반 적응적 가변장 부호화 (Context-Adaptive VLC) 기법을 사용하고 있다. 변환계수 부호화 효율을 높이기 위한 CAVLC의 성능을 〈그림 6〉에서 알 수 있다.

〈그림 6〉은 CAVLC를 사용하였을 경우의 총 비트 감소율을 나타낸 것이다. 예상대로 QP가 높은 영역에서는 이미 EG 부



〈그림 6〉 각 영상별 CAVLC의 비트 감소율 분석

〈표 7〉 CAVLC의 비트율 및 PSNR 성능

실험영상	성능			
	High bitrate	Low bitrate	BDBR (%)	BDPSNR (dB)
Container	-7.11	0.37	-2.93	0.14
News	-11.50	0.83	-2.95	0.17
Foreman	-8.86	0.59	-1.24	0.07
Silent	-9.79	0.72	-1.19	0.06
Paris	-12.54	0.81	-3.33	0.17
Mobile	-14.83	1.26	-3.32	0.14
Tempete	-13.99	1.04	-2.63	0.10
Avg	-11.23	0.80	-2.51	0.12

호의 효율이 높기 때문에 적응적 부호화를 하더라도 얻을 수 있는 비트 감소가 3%정도로 미미하지만 QP가 작아질수록 실제 통계적 특성에 적응적으로 부호화함으로써 높은 부호화 효율을 얻을 수가 있는 것이다. 또한 QP가 작아지면서 변환계수에 해당하는 상대적 발생량도 높아지기 때문에 총 비트량에 끼치는 영향도 커진다.

〈표 7〉은 각 영상별 PSNR 및 총 비트량 변화를 나타낸 것이다. 여기서 사용된 지표인 BDBR (Bjontegaard-Delta Bit Reduction) 및 BDPSNR (Bjontegaard-Delta PSNR)은 RD 곡선에서의 차이값을 평균낸 것이다[13]. 고비트율에서 CAVLC

의 BDBR은 약 -11.23%이며 저비트율에서는 약 -2.51%이다. CAVLC를 이용한 엔트로피 부호화에서 PSNR의 차이가 나타나는 것은 부호기에서 RD 최적화를 사용하였기 때문이다.

## 5. 결론

본 고에서는 현재 H.264에서 사용하고 있는 엔트로피 부호화 기법에 대하여 소개하였다. 현재 H.264에서는 엔트로피 부호화에 있어서 부호화 효율의 측면 뿐만 아니라 메모리나 연산량 등과 같은 하드웨어 구현 및 에러 내성 등과 같은 부가적인 측면도 고려하여 EG 부호를 모든 부호화 요소에 적용하고 있다. 단, 고비트율에서의 변환 계수 부호화 효율이 떨어지는 것을 막기 위해 CAVLC라는 새로운 부호화 기법을 적용하여 실제 부호화 요소의 통계에 적응적으로 부호화하도록 하고 있다. 그러나 이러한 CAVLC는 기존 EG 부호가 갖는 에러 내성 및 간단한 하드웨어 구현의 장점을 잃어버리게 되기 때문에 이러한 문제를 보완하기 위한 연구가 필요하다.

### 참고 문헌

- (1) "Text of ISO/IEC 14496-10 FCD Advanced Video Coding," ISO/IEC JTC1/SC29/WG11 N4920, Klagenfurt, July, 2002.
- (2) K. Sayood, *Introduction to Data Compression*, Morgan Kaufmann, pp.13-54, 2000.
- (3) B. Jeon, J. Park, and J. Jeong, "Huffman Coding of DCT Coefficients Using Dynamic Codeword Assignment and Adaptive Codebook Selection," *Signal Processing: Image Communications*, pp. 253-262, 1998.
- (4) W. Choi and B. Jeon, "Universal variable length coding with adaptive codeword mapping using fixed re-association table," *VLSI*, pp.66-70, 11 Oct. 2001, Greece.
- (5) 최동일, 전병우, 유국열, 선강우, "고생 자배점 테이블 기반 동적 UVLC부호화 방법", 대한전자공학회 논문지 제39권 SP 편 제2호, pp.56-63, 2002년 3월.

### • 참고문헌 •

- (6) 이진학, 김재균, "비디오 압축을 위한 영상간 차분 DCT 계산과 블록간 차분 부호화 방법," 제 1회 신호처리방법학술대회, pp.281-286, 2000.
- (7) S. W. Golomb, "Run-length Encodings," IEEE Trans. Information Theory, vol.12, pp.399-401, July, 1966.
- (8) R. F. Rice, "Some Practical Universal Noiseless Coding Techniques," Technical Report JPL-79-22, Jet Propulsion Laboratory, Mar., 1979.
- (9) J. Teuhola, "A Compression Method for Clustered Pictures," Information Processing Letters, vol.7, pp.308-311, Oct., 1978.
- (10) J. Wen and J. D. Villaseñor, "A Class of Reversible Variable Length Codes for Robust Image and Video Coding," Proceedings of International Conference Image Proc., vol.2, pp.59-62, Oct., 1997.
- (11) F. Ling, W. Li, and H. Sun, "Bitplane Coding of DCT Coefficients for Image and video Compression," SPIE Conference on Visual Communications and Image Processing, VPIE Vol. 3652, Jan., 1999.
- (12) W. Choi and B. Jeon, "Dynamic UVLC Coding with Adaptive With Fixed Pre-Association Table for H.26L," Picture Coding Symposium(PCS), Seoul, Korea, pp.169-171, April, 2001.
- (13) G. Sullivan, G. Bjontegaard, "Recommended Settings in Common Conditions for H.26L Coding Efficiency Experiments on Low-Resolution Pictures from Source Material," ITU-T Q.6/16, Doc. VCEG-N81, Sep., 2001.
- (14) G. Bjontegaard, "Calculation of Average PSNR Differences between RD-curves," ITU-T Q.6/16, Doc. VCEG-M33, Mar., 2001.

### 필자소개

#### 전병우



- 1985년 2월 : 서울대학교 전자공학과 졸업(공학사)  
- 1987년 2월 : 서울대학교 전자 공학과 졸업(공학석사)  
- 1992년 12월 : Purdue Univ. School of Elec. 졸업(공학박사)  
- 1993년~1997년 8월 삼성전자 신호처리연구소 수석연구원  
- 1997년 9월~현재 : 성균관대학교 정보통신공학부 부교수  
- 주관심분야 : 멀티미디어, 영상압축, 영상인식, 신호처리

#### 최웅일



- 2000년 2월 : 성균관대학교 정보통신공학부 졸업(공학사)  
- 2002년 2월 : 성균관대학교 정보통신공학부 졸업(공학석사)  
- 2002년 9월~현재 : 성균관대학교 정보통신공학부 박사과정  
- 주관심분야 : 영상압축, 멀티미디어 응용