

## 2진 MQ 산술부호기의 성능 개선

# Performance Improvement of Binary MQ Arithmetic Coder

고형화<sup>1\*</sup> · 서석용<sup>2</sup>

<sup>1</sup>광운대학교 전자통신공학과

<sup>2</sup>경민대학교 정보통신과

Hyung Hwa Ko<sup>1\*</sup> · Seok Yong Seo<sup>2</sup>

<sup>1</sup>Department of Electronics and Communication Engineering, Kwangwoon University, Seoul 01897, Korea

<sup>2</sup>Department of Information & Communication, Kyungmin College, Gyeonggi-do 11618, Korea

### [요 약]

2진(binary) MQ 산술부호화는 최근 들어 멀티미디어 압축 표준시스템에 기본 엔트로피 방식으로 사용되고 있다. MQ 산술부호기는 JBIG2와 JPEG2000에 적용되면서 압축성능을 인정받기 시작했다. 최근에 차세대 동영상 부호화 표준인 HEVC (high efficiency video coding)에는 산술부호화가 단일 엔트로피부호화로 채택되면서 그 중요성이 커지고 있다. 기존의 2진 MQ 산술부호기는 RANGE(구간)을 분할하는 과정에서 곱셈을 없애면서 근사화 방법을 사용하고 있다. 이 경우 MPS/LPS의 구간이 뒤바뀌는 경우가 발생하며 출력비트가 늘어날 수 있다. 본 논문에서는 이러한 문제점을 완화하기 위하여 근사식을 사용하는 대신에 룩업테이블 형태로 AQe의 값을 양자화하여 계산에 적용하는 방법을 제안하고자 한다. 제안한 방법의 압축 성능을 실험을 통해 확인한 결과, 2진영상 압축표준 방식인 JBIG2의 경우 약 4%의 압축율의 개선을 보였다. 정지영상 압축표준인 JPEG2000의 경우 약 1% 정도의 개선을 가져왔다. 룩업테이블을 사용하기 때문에 계산량이 기존방법에 비해 늘지 않는다.

### [Abstract]

Binary MQ arithmetic coding is widely used recently as a basic entropy coder in multimedia coding system. MQ coder esteems high in compression efficiency to be used in JBIG2 and JPEG2000. The importance of arithmetic coding is increasing after it is adopted as a unique entropy coder in HEVC standard. In the binary MQ coder, arithmetic approximation without multiplication is used in the process of recursive subdivision of range interval. Because of the MPS/LPS exchange activity happened in MQ coder, output byte tends to increase. This paper proposes an enhanced binary MQ arithmetic coder to make use of a lookup table for AQe using quantization skill in order to reduce the deficiency. Experimental results show that about 4% improvement of compression in case of JBIG2 for bi-level image compression standard. And also, about 1% improvement of compression ratio is obtained in case of lossless JPEG2000 coding. For the lossy JPEG2000 coding, about 1% improvement of PSNR at the same compression ratio. Additionally, computational complexity is not increasing.

**Key word** : Binary MQ arithmetic coder, Entropy coder, JBIG2, JPEG2000, Q coder.

<http://dx.doi.org/10.12673/jant.2015.19.6.614>



This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Received 19 November 2015; Revised 25 November 2015

Accepted (Publication) 7 December 2015 (30 December 2015)

\*Corresponding Author; Hyung Hwa Ko

Tel: +82-10-5276-5137

E-mail: hhkoh@kw.ac.kr

## I. 서 론

메시지를 발생확률에 근거하여 fractional 비트로 표현할 수 있다는 Shannon의 이론이 발표된 이후 엔트로피 부호화에 대한 많은 연구가 진행되었다[1],[2]. 대표적으로 허프만 부호화는 간단하면서도 성능이 좋아 최근까지 문서, 정지 영상 및 동영상 압축에서 엔트로피 부호화 방법으로 널리 쓰이고 있다.

반면에 Elias는 부호구간을 축차적으로 분할하는 산술부호화방법을 제안하였다[3]. 허프만 부호화가 미리 만들어진 테이블을 이용하는 반면에 산술부호화는 부호화를 진행하면서 확률예측 값을 적응적으로 적용하는 특징이 있어서 시간적으로 변하는 입력신호에 최적의 압축효율을 나타낸다. 산술부호화는 [0,1]의 확률 구간을 지속적으로 분할하여 소수점 이하의 값을 바이너리로 부호화하는 방법이다. 단점으로는 메시지의 길이가 길어지면 정밀도가 높아져야 하는 점이다. 이러한 단점을 개선하기 위해 [0, 1] 대신에 [0, N]의 정수범위내에서 분할하는 방법도 있다[2]. 그 후에 발표된 산술부호기들은 이러한 정밀도의 증가를 제한하기 위해 구간이 줄어들면 재정규화를 행하고 출력비트를 발생시킴으로 문제를 완화시켰다. 산술부호화 방법은 Witten[4], Lagdon[5] 등에 의해 하드웨어 및 소프트웨어로 구현되었다. 80-90년대에 산술부호기에 대해 많은 연구가 수행되었고, 적응적 방법들이 발표되었다[6]-[8].

산술부호화가 멀티미디어 부호화에 적용되기 시작한 것이 2진 산술부호기였다. 허프만부호기가 멀티심벌 부호화가 주된 응용분야인 것에 비하여, 산술부호기는 복잡성 때문에 2진 데이터에 적용하는 방법이 주로 연구되었다. 그중에 제일 관심을 끈 것은 IBM에서 제안한 Q 코더[9]로 적응적인 2진 산술부호기이다. 부호화 과정 중에 입력된 시퀀스로부터 추정된 확률에 의하여 적응적으로 부호화하는 메카니즘을 갖고 있다[9]. 16비트 길이로 확률 구간의 정밀도를 제한해서 하드웨어의 크기를 줄였다. 그러나, 제한된 레지스터 크기로 인해 여러 가지 문제가 발생할 수 있는데, 그 가운데 재정규화 과정에서의 캐리 전파 문제는 비트 스타핑으로 해결하였다. 즉, 8개의 연속적인 1이 발생하면, 0을 삽입한다. 무엇보다도 확률구간을 지속적으로 분할하는 과정에서 LPS의 발생확률과 이전 구간의 값을 [0.75, 1.5]로 유지하도록 하는 재정규화를 수행하고, 이러한 조건하에서 곱셈을 피하도록 하였다.

1990년대 들어 멀티미디어의 부호화 표준화가 활발히 진행되었다. 산술부호기는 JPEG으로 알려진 정지영상의 압축방식에 허프만부호화와 함께 엔트로피 부호화로 채택되었다[10]. 허프만 부호화는 Baseline 프로파일로, 산술부호화는 Extended 프로파일로 채택되었다. JPEG회의에서 IBM, Lucent, Mitsubish의 3개 기관에서 공동으로 QM 산술부호기를 제안하였는데 QM 산술부호기(이후 코더라고 하자)는 MPS/LPS 조건부 맞교환을 처음으로 제안하여 Q 코더의 부호화 효율을 증가시켰다. 그리고, 빠른 초기 학습이 가능하도록 초기 확률 예측을 개선시켰다. Q 코더가 하드웨어에 적합하다면 QM 코더는 소프트웨어

어로 구현하기에 적합하다. Q 코더는 캐리 전파를 복호기에서 해결하고, QM 코더는 부호기에서 해결한다. 확률 예측은 룩업 테이블 방식으로 수행하고, MPS의 확률이 LPS의 확률보다 작아지면 MPS/LPS 조건부 맞교환을 수행한다[2].

1993년 2진문서 압축을 위한 새로운 JBIG 표준이 만들어졌다[11]. JBIG 압축표준에서는 JPEG과 달리 산술부호화가 베이스라인 프로파일로 채택이 되었다. 1998년 새로운 2진압축 방식인 JBIG2 표준방식[12]이 만들어졌는데, JBIG2는 최근에 널리 사용되는 PDF 파일 압축에 사용되고 있다. JBIG2에서는 QM 코더와 유사한 MQ 코더가 채택되었다[12]-[13]. MQ 코더는 QM 코더에서 캐리 전파 방식을 변경하였다. 바이트 0xFF가 출력될 때에만 하나의 0 비트를 삽입한다. MQ 코더의 비트 스타핑은 대역폭과 수행시간에서 효율적이다. 2004년에는 새로운 정지영상 압축방식인 JPEG-2000이 표준안으로 채택되었는데[14], JPEG2000 표준에서 MQ 산술부호화가 베이스라인 프로파일로 결정되어 JPEG의 엔트로피 부호화 방식인 허프만부호화를 완전히 대체하게 되었다. 2000년초까지 산술부호기는 Q 코더로부터 변형 확장된 산술부호기가 사용되다가, 동영상 압축표준인 H.264/AVC에서는 CABAC (context adaptive binary arithmetic coding) 방식이 표준으로 채택되었다. 다른 이름으로 M 코더라고도 부른다. 이 방식은 차세대 동영상 부호화 방식인 HEVC (high efficiency video coding)에서 유일한 엔트로피부호화방식으로 채택되었다[15]-[18]. 본 논문에서는 기존의 2진 MQ 코더의 성능을 개선하는 방법을 제안하고자 한다. 2장에서는 기존의 2진 MQ 산술부호기에 관하여 기술하고, 3장에서는 개선된 2진 산술부호기를 제안하였고, 4장에는 실험 및 결과를 기술하고, 5장에서는 결론을 이끌어 내었다.

## II. 2진 산술부호기

### 2-1 2진 Elias 산술부호기

0과 1의 심벌로 구성된 입력시퀀스를 효율적으로 압축하려고 할 때, 각 심벌의 발생확률을 이용하게 된다. Shannon의 엔트로피 이론과 무손실 코딩 법칙에 의해 발생확률이 높은 심벌에 더 작은 부호어(codeword)를 할당하고, 발생확률이 낮은 심벌에 더 긴 부호어를 할당함으로써 0과 1의 발생확률이 크게 다른 2진 문서를 압축할 때 매우 효율적인 압축할 수 있다. 1960년대에 허프만부호화와 함께 높은 압축률을 갖는 산술부호기가 개발되었다. Elias에 의해 제안된 2진 산술부호기 원리를 간단히 알아보겠다.

- 현재의 구간을 [Lower, Upper) 라고 하자. 부호화를 시작할 때 [0,1) 초기화한다.

- 한 비트가 입력되면, 현재의 구간이 다음 식(1),(2)에 의해 둘로 나뉜다.

$$Lower \leftarrow Lower + (Upper - Lower) \times CDF(n-1) \quad (1)$$

$$Upper \leftarrow Lower + (Upper - Lower) \times CDF(n) \quad (2)$$

여기서, CDF(n)은 n번째 입력심벌의 cumulative distribution function(누적분포함수)이다. (Upper - Lower) 값을 RANGE(구간)이라고 부른다.

- 모든 시퀀스 입력이 들어오면 출력이 발생한다. 최후의 RANGE의 길이는 모든 심벌들의 발생확률의 곱과 같다. 매 입력비트가 들어올 때마다 구간은 지속적으로 줄어들면서 최종 입력 비트가 인가된 후에 누적된 RANGE를 표현하는 소수점 이하를 바이너리로 표현한다. 이 경우에 입력시퀀스의 규모가 커지면 구간의 정밀도가 커지는 어려움이 있다.

산술부호화의 예를 그림1에 보였다. 0의 발생확률이 0.6, 1의 발생확률이 0.4라고 가정하고, 입력시퀀스를 {0110} 이라고 하자. 0의 발생확률이 0.6이므로 CDF(-1)=0, CDF(0)=0.6, CDF(1)=1이 된다. RANGE의 초기 값은 Lower = 0, Upper = 1로 초기화된다. 최초로 0이 입력되면 Lower = 0, Upper = 0.6이 된다. 다음으로 1이 입력되면 Lower = 0.36, Upper = 0.6이 된다. 다음으로 1이 입력되면 Lower = 0.504, Upper = 0.6이 된다. 다음으로 0이 입력되면 Lower = 0.504, Upper = 0.5616이 된다.

입력전체가 4비트라고 가정하면, 최종적으로 {0.504, 0.5616}을 binary로 표현하면 {0.10000001..., 0.1000100...}이 된다. 소수점 이하의 비트들의 공통비트가 출력이 되므로, 출력은 "1000"이 나오게 된다. 만약 입력비트를 늘리게 되면 분할된 구간이 지속적으로 줄어들면서 구간을 표현하는데 급속도로 많은 비트가 필요해진다.

이렇게 산술부호기의 정밀도가 높아지는 문제를 해결하기 위해 RANGE 값이 1/2이하로 줄어들면 renormalization (재정규화) 또는 scaling 과정을 통해 RANGE 값이 2배가 되고, 재정규화 과정 중에 출력비트를 발생시킨다.

### 2-2 2진 MQ 산술부호기 개요[12]

2진 MQ 산술부호기는 그림 2와 같이 연속적인 비트입력(D)와 해당되는 컨텍스트(CX)를 입력하면, 2진 출력(CD)을 발생시킨다. 입력으로 비트 0 또는 1이 인가된다. MQ 산술부호기는 MPS(most probable symbol)와 LPS(least probable symbol)가 각각 0 또는 1인지를 먼저 정하고, 현재의 입력비트를 MPS 값과 비교하여 MPS 인지 정한다. 해당 심벌의 컨텍스트별로 확률적 모델을 이용하여 부호화를 실시한다.

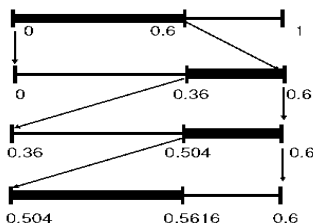


그림 1. 산술부호화의 예  
Fig. 1. Example of arithmetic coding.

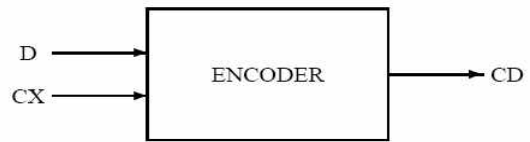


그림 2. 산술부호기 입력과 출력.  
Fig. 2. Arithmetic coder inputs and outputs.

현재의 구간 값이 입력비트 값에 의해 두 개의 구간으로 분할된다. 구간을 분할하는 과정에서 MPS의 구간은 LPS보다 위에 위치하도록 한다.

입력된 결과는 베이스(C 레지스터에 저장)값과 구간(A 레지스터에 저장) 값으로 표시된다. 부호화 결과로 출력 코드스트링은 베이스(C) 값을 가리키도록 변경된다. 부호화 초기에 베이스(C)는 0으로 초기화되고, 구간(A)은 십진수로 0.75(Hex로는 0x8000)로 초기화된다. 구간(A)은  $0.75 \leq A \leq 1.5$ 를 유지할 수 있도록 필요시 재정규화를 수행한다. 즉, A 값이 0.75보다 작으면 A 값을 2배로 하고, C 값도 A 값이 2배가 될 때 마다 함께 2배로 만들어 준다. 이러한 작업을 A 값이 0.75보다 클 때까지 반복해서 시행한다. A 레지스터는 16비트로 구성되며, C 레지스터는 32비트의 길이를 갖도록 다음과 같이 설정한다.

C-레지스터: 0000cbbb bbbbbbss xxxxxxxx xxxxxxxx

C 레지스터의 하위 16비트는 A 레지스터로부터 1 바이트씩 LSB에 시프트 입력되고, 재정규화를 수행 중에 카운트 값이 0이 되면 b위치의 1 바이트 값을 출력(BYTEOUT) 시킨다. C 레지스터에서 제거되어 출력 데이터 열 버퍼에 위치시킨다.

LPS 발생 확률 예측 값이  $Q_e$  라고 하면, 분할된 구간(A)은 다음 식 (3)에 의해 구한다.

$$\begin{aligned} \text{MPS에 대한 분할된 구간} &= A - (A \times Q_e) \\ \text{LPS에 대한 분할된 구간} &= A \times Q_e \end{aligned} \quad (3)$$

식 (3)에서 매입력 비트마다 한번의 곱셈이 필요하므로 계산량이 많다. 따라서, A 값이 1에 가깝다고 가정하고,  $A \times Q_e$ 는  $Q_e$ 로 근사화된다고 가정하였다[9]. 따라서, 식 (3)에서 곱셈을 없앴 후 식 (4)와 같이 근사화된다.

$$\begin{aligned} \text{MPS에 대한 분할된 구간} &= A - Q_e \\ \text{LPS에 대한 분할된 구간} &= Q_e \end{aligned} \quad (4)$$

MPS가 부호화될 때는, 구간은  $A - Q_e$ 로 감소하고, C 값에는  $Q_e$ 의 값이 더해진다. 반면에, LPS가 부호화될 때는, 구간은  $Q_e$ 로 감소하고, C 값은 변하지 않는다. A 값이 0.75보다 작아지면 A와 C 값은 재정규화 후에 저장된다. 그러나, 현재 구간을 분할할 때, MPS의 분할 구간보다 LPS의 분할 구간이 더 큰 경우가 생길 수 있다. 예를 들면, 만약 현재의 구간이 0.75,  $Q_e$  값이 0.5라고 하면 MPS의 분할 구간이 0.25, LPS의 분할 구간이 0.5가 된다. 이와 같은 크기의 역전을 막기 위하여, LPS 구간이 MPS 구간보다 클 때는 MPS와 LPS를 맞바꾼다. 이 MPS/LPS

조건부 맞교환은 재정규화가 필요할 때에만 발생할 수 있다.

**2-3 2진 MQ 코더의 MPS 부호기 (CodeMPS)**

그림3에 보인 것처럼 입력 비트가 MPS(CX)와 같을 때 수행된다. 구간(A) 값이 분할되어 A-Qe로 변경된다. 변경된 A 값이 0.75보다 크면 베이스(C) 값이 Qe 만큼 증가한다. 그러나, 구간 A 값이 0.75보다 작으면 A-Qe와 Qe를 비교하게 된다. 여기서, A-Qe가 Qe보다 작은 경우 LPS의 분할 구간이었던 Qe 값으로 A 값이 변경된다. 즉, 분할된 구간길이가 역전되지 않도록 MPS와 LPS 구간을 서로 바꾸어준다. 그러나 반대의 경우에는 C값에만 Qe값이 더해진다. 새로운 인덱스 I(CX)는 NMPS(I(CX)) 값으로부터 결정된다. 이어서 재정규화가 실행된다.

**2-4 2진 MQ 코더의 LPS 부호기 (CodeLPS)**

그림 4에 보인 것과 같이 입력 비트와 MPS(CX)가 서로 다를 때 수행된다. 여기서, CX는 컨텍스트를 의미하며 비트가 입력될 때 CX 값이 함께 인가된다. 컨텍스트를 저장한 인덱스 I(CX)로부터 결정된 LPS의 확률 예측 Qe(I(CX)) 값을 알아낸다. LPS 부호시에 Qe가 A-Qe 보다 더 작다는 것을 알기 위해 비교과정이 행해진다. A 레지스터에는 Qe와 A-Qe 중에서 작은 값이 입력되고, C 레지스터에는 Qe가 A-Qe 보다 작은 경우에는 원래의 C 값에 Qe 값이 더해진다. 반대로 A-Qe가 Qe 보다 작을 경우 C값은 변하지 않는다. 이어서, 인덱스 I(CX)에 대한 SWITCH 플래그가 1로 나타나면, MPS(CX) 값이 바뀌게 된다. MPS가 0이었으면 1로, 1이었으면 0으로 바뀌게 된다. 새로운 인덱스 I(CX)는 NLPS(I(CX)) 값으로부터 결정된다. 이어서 재정규화(renormalization) 과정이 실행된다.

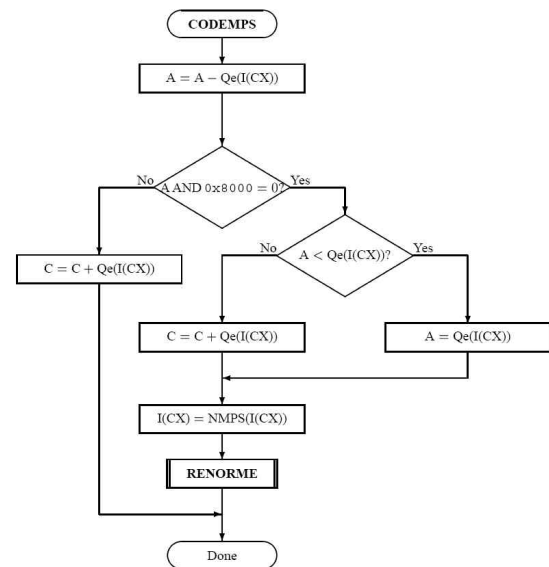


그림 3. MPS 부호화 과정  
Fig. 3. CodeMPS procedure.

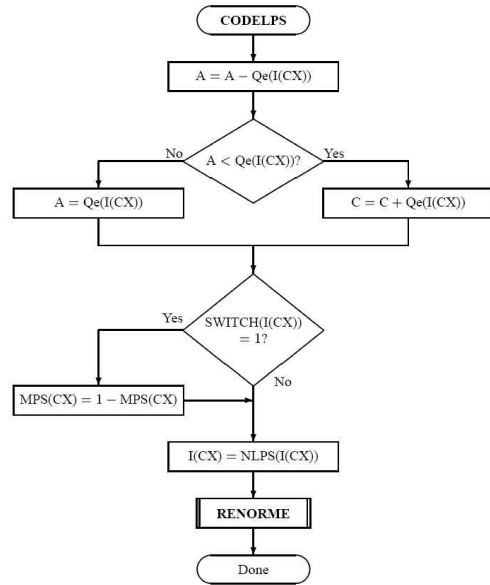


그림 4. LPS 부호화 과정  
Fig. 4. CodeLPS procedure.

**2-5 2진 MQ 코더의 재정규화 (Renormalization)**

그림 5에 보인 것처럼 재정규화 과정에서는 A와 C 값이 2배 증가하고, 카운터 값(CT)이 1 만큼 감소한다. CT 값이 0이 되면 1 바이트를 출력시킨다. 이어서 A 값이 0.75(HEX로 0x8000) 보다 작을 경우, 커질 때까지 위 과정을 반복한다.

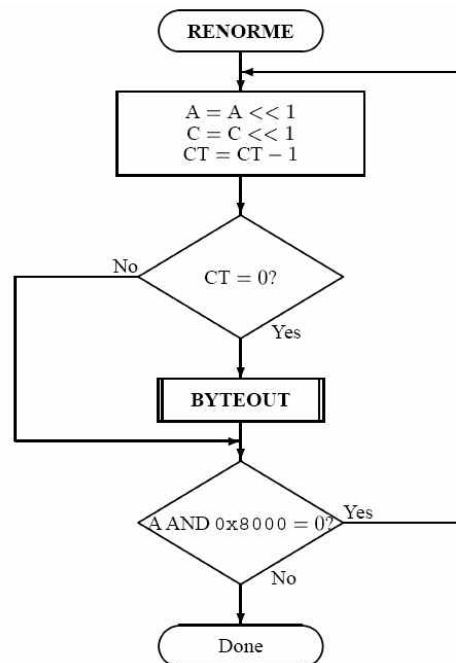


그림 5. 재정규화 과정  
Fig. 5. Renormalization procedure.

### 2-6 2진 MQ 코드의 확률 예측 방법

LPS 확률의 예측을 위한 확률예측 테이블을 표 1에 보였다. Qe 값은 hexadecimal과 decimal로 표시된다. 예측기는 유한 상태 머신으로 정의할 수 있다. 확률 상태는 재정규화가 이루어진 후에 천이된다. 즉, LPS가 입력되거나, MPS를 부호화한 후에 A 값이 0x8000보다 작아질 경우 상태가 바뀐다. LPS 재정규화 후에 NLPS는 LPS 확률의 새로운 인덱스 값을 의미한다. NMPS는 MPS의 재정규화 후에 LPS의 예측확률 값이다. 마지막에 SWITCH(I[CX]) 값은 MPS 값을 변경하도록 해준다.

MQ 산술부호기의 LPS 확률 예측 테이블은 Q 산술부호기[9]와 거의 같으나, Index 46번의 확률 값이 추가된 것만 다르다. 특징으로는 Index 0, 6, 14에 초기값(0x5601)을 갖도록 하는 3개의 확률 모델을 포함하고 있다. 연속적으로 0 또는 1이 발생할 때 LPS 발생확률이 제로에 수렴하는 속도의 차이를 나타낸다.

### III. 개선된 2진 MQ 산술부호기의 제안

#### 3-1 기존 방법의 문제점

기존의 2진 MQ 산술부호기는 계산량을 감소시키기 위해 AQe를 Qe로 근사화 시킨다고 2.2절에 밝혔다. 그러나, 2진 MQ 산술부호기가 제안할 당시에는 제반 컴퓨팅 성능이 만족스럽지 못하여 이러한 근사화 방법이 타당한 면이 있었으나, 최근에 컴퓨팅 능력이 높아지면서 이러한 근사화 방법을 수정할 필요가 있다.

기존의 2진 MQ 산술부호기에서는 A 값이  $0.75 < A < 1.5$  범위에 존재한다고 가정하고, MPS와 LPS의 분할구간을 다음 식(5)와 같이 AQe를 Qe로 근사화 시켰다.

$$\begin{aligned} \text{MPS: } A &= A - Q_e, \quad C = C + Q_e \\ \text{LPS: } A &= Q_e \end{aligned} \tag{5}$$

단점으로는 Qe 가 A/2 보다 커지면 MPS/LPS의 역전이 발생할 수 있다. 따라서, 이 경우에 MPS와 LPS를 바꿔 주게 된다. 이 과정에서 출력 비트가 늘어날 수 있다.

#### 3-2 제안하는 방법

기존의 근사화 방법과 같이 곱셈을 하지 않으면서 부호화 효율을 높이도록 하는 방법을 제안하고자 한다. 식(5) 대신에 식(3)을 적용하되 AQe를 직접 곱하는 대신에 AQe를 2레벨과 4레벨의 양자화 한 값을 구하여 47x2 혹은 47x4 크기의 look-up table 방식으로 AQe 값을 식(6)과 같이 현재 구간(A)와 베이스(C)를 구하는 방법을 제안하고자 한다.

표 1. LPS 확률 예측 테이블

Table 1. LPS probability estimation table.

Index	Qe (Hex)	Qe (decimal)	NMPS	NLPS	Switch
0	0x5601	0.503960	1	1	1
1	0x3401	0.304729	2	6	0
2	0x1801	0.140656	3	9	0
3	0x0ac1	0.063015	4	12	0
4	0x0521	0.030054	5	29	0
5	0x0221	0.012475	38	33	0
6	0x5601	0.503960	7	6	1
7	0x5401	0.492240	8	14	0
8	0x4801	0.421924	9	14	0
9	0x3801	0.328168	10	14	0
10	0x3001	0.281290	11	17	0
11	0x2401	0.210973	12	18	0
12	0x1c01	0.164095	13	20	0
13	0x1601	0.128937	29	21	0
14	0x5601	0.503960	15	14	1
15	0x5401	0.492240	16	14	0
16	0x5101	0.474661	17	15	0
17	0x4801	0.421924	18	16	0
18	0x3801	0.328168	19	17	0
19	0x3401	0.304729	20	18	0
20	0x3001	0.281290	21	19	0
21	0x2801	0.234412	22	19	0
22	0x2401	0.210973	23	20	0
23	0x2201	0.199254	24	21	0
24	0x1c01	0.164095	25	22	0
25	0x1801	0.140656	26	23	0
26	0x1601	0.128937	27	24	0
27	0x1401	0.117218	28	25	0
28	0x1201	0.105498	29	26	0
29	0x1101	0.099638	30	27	0
30	0x0ac1	0.063015	31	28	0
31	0x09c1	0.057155	32	29	0
32	0x08a1	0.050563	33	30	0
33	0x0521	0.030054	34	31	0
34	0x0441	0.024927	35	32	0
35	0x02a1	0.015405	36	33	0
36	0x0221	0.012475	37	34	0
37	0x0141	0.007348	38	35	0
38	0x0111	0.006249	39	36	0
39	0x0085	0.003044	40	37	0
40	0x0049	0.001671	41	38	0
41	0x0025	0.000847	42	39	0
42	0x0015	0.000481	43	40	0
43	0x0009	0.000206	44	41	0
44	0x0005	0.000114	45	42	0
45	0x0001	0.000023	45	43	0
46	0x5601	0.503960	46	46	0

$$\text{MPS: } A = A - LUT(AQ_e), \quad C = C + LUT(AQ_e) \tag{6}$$

$$\text{LPS: } A = LUT(AQ_e)$$

식(6)의 계산을 하기 위해서는 룩업테이블이 필요한데, Qe 값과 A 값으로부터 계산을 통해 룩업테이블 값을 구하는 방법을 다음과 같이 제안한다. 먼저 재정규화를 실시한 후에 A 값의

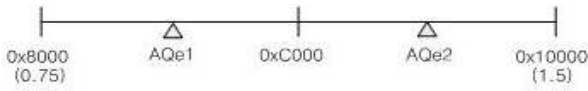


그림 6. 2레벨 양자화 방법 (N=2)  
Fig. 6. 2-level quantization method (N=2).

범위는  $0.75(=0x8000) \leq A < 1.5(0x10000)$  사이의 값을 갖는다. 예를 들어, MQ 코더의 인덱스 값이 0일 때, LPS의 발생확률  $Q_e$  값은 표 1에 보인 것과 같이  $0.50395(=0x5601)$  이므로 Aqe의 범위는  $0.75 \cdot 0.50395(\approx 0x4082) < Aqe < 1.5 \cdot 0.50395(\approx 0x8104)$  사이의 값을 갖게 된다.

첫째, 양자화 레벨을 2로 했을 때, 그림 6은 A 값을 기준으로 둘로 분할할 경우이다. 인덱스가 0일 때, Aqe 값의 범위를 둘로 분할하면, Aqe1 값은 첫 번째 구간의 중앙값인  $0x50A2$ 로 결정된다. Aqe2는 두 번째 구간의 중앙값인  $0x70E3$ 으로 결정된다. 일반적으로 식(7)과 같이 Aqe1과 Aqe2 값을 결정한다.

$$\Delta = (1.5 - 0.75) / 2 = 0.375 \text{라고 할 때,}$$

$$\text{if } (0x8000 < A < 0xC000) \text{ then}$$

$$Aqe_1[I(CX)] = (1/2)\Delta \times Q_e[I(CX)] \quad (7)$$

$$\text{else } Aqe_2[I(CX)] = (3/2)\Delta \times Q_e[I(CX)]$$

각 인덱스별로 Aqe1과 Aqe2를 구하여 룩업테이블을 작성하여 일부를 표 2에 보였다.

두 번째로 양자화레벨을 4로 했을 때, A 값을 기준으로 그림 7과 같이 4개의 구간으로 분할한다. 인덱스 값이 0 일 때, Aqe1 ~ Aqe4 는 구간의 중앙값으로 결정하는데, Aqe1 = 0x4891, Aqe2 = 0x58B2, Aqe3 = 0x68D2, Aqe4 = 0x78F3이 구해진다. 일반적으로 식(8)과 같이 Aqe1 ~ Aqe4 값을 결정한다. 각 인덱스별로 Aqe 값을 구하여 룩업테이블을 만든다. 그 일부를 표 3에 보였다.

표 2. Aqe 룩업 테이블 (N=2)  
Table 2. Aqe lookup table (N=2).

Index I(CX)	Aqe1	Aqe2	NMPS	NLPS	Switch
0	0x50A2	0x70E3	1	1	1
1	0x30C1	0x4442	2	6	0
2	0x1681	0x1F81	3	9	0
3	0x0A15	0x0E1D	4	12	0
...	.....	.....	..	..	.
43	0x0008	0x000B	44	41	0
44	0x0004	0x0006	45	42	0
45	0x0001	0x0001	45	43	0
46	0x50A2	0x70E3	46	46	0

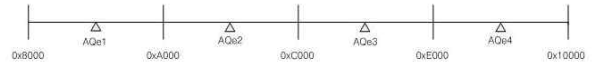


그림 7. 4레벨 양자화 방법 (N=4)  
Fig. 7. 4-level Quantization method (N = 4).

표 3. Aqe 룩업테이블 (N=4)  
Table 3. Aqe lookup table (N=4).

Index I(CX)	Aqe1	Aqe2	Aqe3	Aqe4	NMPS	NLPS	Switch
0	0x4891	0x58B2	0x68D2	0x78F3	1	1	1
1	0x2BE1	0x35A1	0x3F62	0x4922	2	6	0
2	0x1441	0x18C1	0x1D41	0x21C1	3	9	0
3	0x0912	0x0B17	0x0D1B	0x0F1F	4	12	0
..	.....	.....	.....	.....	..	..	.
43	0x0007	0x0009	0x000A	0x000C	44	41	0
44	0x0004	0x0005	0x0006	0x0007	45	42	0
45	0x0001	0x0001	0x0002	0x0004	45	43	0
46	0x4891	0x58B2	0x68D2	0x78F3	46	46	0

수정된 확률테이블을 만들어진 후, 산술부호화 할 때는 A 값의 범위에 따라 룩업테이블의 Aqe 값을 적용하여 계산함으로써 곱셈을 수행하지 않으므로 계산량이 크게 늘어나지 않는다.  $\Delta = (1.5 - 0.75) / 2 = 0.375$ 라고 할 때,

$$\text{if } (0x8000 \leq A < 0xA000) \text{ then}$$

$$Aqe_1[I(CX)] = (1/2)\Delta \times Q_e[I(CX)]$$

$$\text{else if } (0xA000 \leq A < 0xC000) \text{ then}$$

$$Aqe_2[I(CX)] = (3/2)\Delta \times Q_e[I(CX)] \quad (8)$$

$$\text{else if } (0xC000 \leq A < 0xE000) \text{ then}$$

$$Aqe_3[I(CX)] = (5/2)\Delta \times Q_e[I(CX)]$$

$$\text{else } Aqe_4[I(CX)] = (7/2)\Delta \times Q_e[I(CX)]$$

#### IV. 실험 및 결과

실험에는 2진 문서 및 표준 그레이 및 컬러영상을 사용하였다. 그림 8은 JBIG위원회에서 정한 200dpi 표준 2진문서 4개를 보였다. 그림 9는 2개의 한글문서를 각각 200, 300, 400dpi로 스케닝하여 구하였다. 2진문서 압축을 위해 이상과 같이 총 10개의 문서를 사용하였다. 컬러 및 그레이 영상은 인터넷 검색을 통해 얻은 정지영상압축 실험에 많이 사용되는 것들로 그림 10에 보였다. 제안한 방법의 압축성능을 평가하기 위해 3종류의 실험을 수행하였다.

첫 번째 실험은 표준 2진 문서를 바이너리 값만으로 압축하는 실험이다. 표 4의 실험 결과를 보면, 원래의 MQ 산술부호기보다 제안한 2레벨 양자화 방법과 4레벨 양자화 방법을 사용한 결과 0.6 ~ 4.6(%)의 압축율의 증가를 확인하였다.

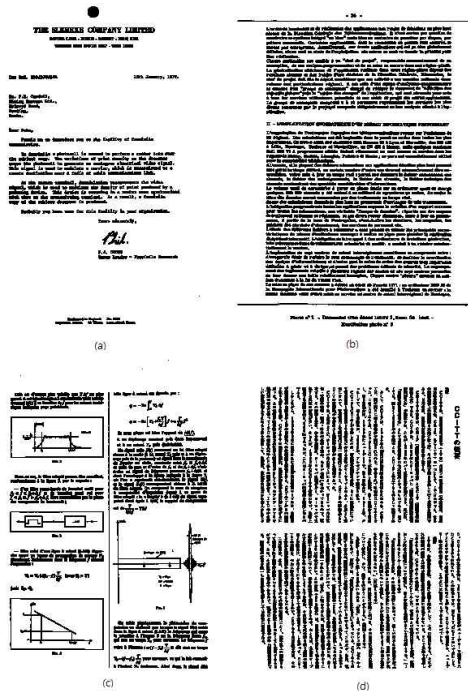


그림 8. 시뮬레이션에 사용된 2진 문서  
 Fig. 8. Binary text image used for simulation.  
 (a) CCITT1 document (b) CCITT4 document  
 (c) CCITT5 document (d) CCITT7 document.

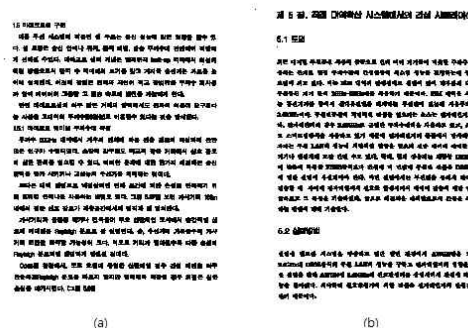


그림 9. 시뮬레이션에 사용된 한글 문서  
 Fig. 9. Binary Hangul text used for simulation.  
 (a) h1 document (b) h2 document

4레벨의 경우가 2레벨의 경우보다 압축성능의 개선이 더 높을 것으로 예상했으나, 실험에 사용한 문서별로 2레벨과 4레벨간의 우열이 엇갈리게 나타났다. 문서의 0와 1의 런(run)의 확률에 따라 압축성능의 차이를 만드는 것으로 생각된다.

두 번째 실험은 동일한 2진 표준문서를 새로운 2진압축방식인 JBIG2[12] 압축방법에 제안한 방법을 적용한 실험결과를 표 5에 보였다. 원래의 방법보다 제안한 2레벨 및 4레벨 예측테이블을 사용한 결과 3.3 ~ 7.3(%)의 압축율이 개선됨을 확인하였다. 4레벨 양자화가 2레벨 양자화보다 성능이 좋은 것으로 나타났다. JBIG2 압축은 컨텍스트 기반 확률예측 모델을 사용하므로 예상한대로 개선이 이루어지는 것으로 생각된다.

표 4. 2진 문서 압축 결과

Table 4. Compression Results for binary text image

File Size	original MQ (KB)	2-level MQ (KB)	improve (%)	4-level MQ (KB)	improve (%)
CCITT1	49.5	48.1	2.8	48.4	2.2
CCITT4	152	147	3.3	148	2.6
CCITT5	97.3	96.4	0.9	96.4	0.9
CCITT7	165	164	0.6	163	1.2
h1_200	68.3	66.4	2.8	65.4	4.2
h1_300	138	136	1.4	136	1.4
h1_400	233	229	1.7	230	1.3
h2_200	71	68.5	3.5	67.7	4.6
h2_300	146	143	2.1	144	1.4
h2_400	245	239	2.4	241	1.6

표 5. JBIG2 실험 결과

Table 5. Experimental results for JBIG2.

File Size	original MQ (KB)	2-level MQ (KB)	improve (%)	4-level MQ (KB)	improve (%)
CCITT1	16.6	15.6	6.0	15.6	6.0
CCITT4	23.2	21.7	6.5	21.6	6.9
CCITT5	34.1	32.6	4.4	32.5	4.7
CCITT7	63.2	58.7	7.1	58.6	7.3
h1_200	19.6	18.3	6.6	18.3	6.6
h1_300	27.3	25.9	5.1	25.9	5.1
h1_400	36.2	34.7	4.1	34.5	4.7
h2_200	21	19.7	6.2	19.6	6.7
h2_300	29.9	28.7	4.0	28.6	4.3
h2_400	39.5	38.2	3.3	38.1	3.5



그림 10. 시뮬레이션에 사용된 정지영상  
 Fig. 10. Still images to be used for simulation.  
 (a) Monarch (b) Lena (c) Man  
 (d) Cablecar (e) Girl (f) Zelda  
 (g) Goldhill (h) Baboon (i) Barbara

세 번째로, 새로운 정지영상 압축방법인 JPEG2000에 제안한 방법을 적용한 무손실부호화 실험결과를 표 6에 제시하였다. 대부분의 정지영상에서 1% 내외의 압축율의 개선을 얻었다. JBIG2보다 개선율은 약간 낮아졌지만 제안한 방식의 효율성을 확인할 수 있었다. 본 실험은 제안한 방법을 Jasper 프로그램[19]에 적용하여 프로그램을 수정하여 실시하였다. 표 7은 Lossy JPEG2000의 실험결과이다. 전송율을 0.05(압축율 = 1/20)로 했을 때, 복원된 정지영상의 PSNR(dB)을 비교하였다.

표 6. 무손실 JPEG2000 실험 결과

Table 6. Experimental results for JPEG2000(lossless).

File Size	Raw Image (KB)	Orig. MQ (KB)	2-level MQ (KB)	improve (%)	4-level MQ (KB)	improve (%)
Airplane	786	378	374	1.08	373	1.20
Baboon	720	569	563	1.14	562	1.30
Babara	1,244	645	638	1.04	637	1.16
Cablecar	737	363	360	1.00	359	1.18
Goldhill	1,244	673	666	1.07	665	1.20
Flowers	543	313	310	0.98	309	1.15
Lena	786	445	440	1.12	439	1.24
Monarch	1,180	442	438	0.96	436	1.12
Cats	6,292	1,988	1,967	1.06	1,963	1.22
Zelda	1,360	594	587	1.02	586	1.15
Girl	1,244	597	591	1.05	590	1.18
Man	1,049	632	625	1.10	624	1.24

표 7. 손실 JPEG2000 실험 결과 (rate = 0.05)

Table 7. Experimental results for JPEG2000 (lossy compression; rate=0.05).

PSNR	Compress. Size (Byte)	Original MQ (dB)	4-level MQ (dB)	improve (%)
		PSNR-G	PSNR-G	
Lena	39,278	36.07	36.45	1.05
Babara	62,132	36.06	36.39	0.92
Cablecar	36,659	35.78	36.09	0.87
Goldhill	62,016	35.88	36.21	0.92
Monarch	58,903	40.38	40.77	0.97
Zelda	67,588	41.00	41.42	1.02
Cats	314,482	34.99	35.35	1.03
Man	52,386	32.55	32.86	0.95

컬러영상의 경우 R,G,B 영상중 대표로 G 영상과 그레이 레벨의 영상 실험 결과 값을 보였다. 실험결과 무손실영상의 실험결과와 유사하게 1% 내외의 개선율을 보였다. R 영상과 B 영상도 비슷한 결과를 보였다.

## V. 결 론

본 논문에서는 2진 MQ 산술부호기의 확률예측 테이블을 확장하여 기존의 근사화 방법으로 인해 생기는 단점을 개선하는 방식을 제안하였다. 곱셈을 하지 않고 덧셈과 뺄셈만으로 연산함으로 인해 연산량을 줄이고자 하는 기존 방식의 단점으로 갖은 MPS/LPS 조건부 맞교환의 발생으로 부호화 효율이 떨어지는 점을 들 수 있다. AQe의 실제 곱셈 값을 사용하는 대신에 미리 계산된 2레벨 혹은 4레벨의 양자화된 확률예측 테이블을 사용함으로써 연산량은 기존의 방식에 비해 늘어나지 않으면서 부호화 효율이 개선되는 것을 실험을 통해 확인하였다. 2진문서를 컨텍스트없이 그대로 사용하여 압축하는 방법에서는 기존의 방식보다 최대 4%정도의 압축율이 개선되는 것을 확인하였다. MQ 산술부호기를 적용하여 JBIG2 압축을 하는 경우 제안한 방식은 기존의 방식에 비해 3.3 ~ 7.3 %의 압축률이 개선됨을 확인하였다. JPEG2000 정지 영상압축 방식의 경우 대부분의 영상에서 1% 내외의 압축률이 개선되는 것을 확인하였다. 제안한 방식은 현재 MQ 산술부호기를 표준 엔트로피부호기로 사용하는 주요 멀티미디어 표준에서 성능개선이 이루어짐을 보였다.

## 감사의 글

이 논문은 2014년도 광운대학교 교내학술연구비 지원에 의해 연구되었음

## 참고 문헌

- [1] J. J. Rissanen, "Generalized Kraft inequality and arithmetic coding," *IBM Journal of Research & Development*, Vol. 20, pp. 198-203. 1976.
- [2] X. Tian, T.M. Le, and Y. Lian, Review of CAVLC, arithmetic coding, and CABAC, in *Entropy Coders of the H.264/AVC Standard*, 1st ed. Berlin: Springer, ch. 2, pp. 29-39, 2011.
- [3] N. Abramson. *Information Theory and Coding*, 1st ed. New York, NY: McGraw-Hill, 1963.
- [4] I. H. Witten, R. M. Neal, and J. G. Cleary, "Arithmetic coding for data compression," *Communication of the ACM*,



Vol. 30, No. 6, pp. 520~540, June 1987.

- [5] G. G. Langdon, Jr, "An introduction to arithmetic coding," *IBM Journal of Research & Development*, Vol. 28 No. 2 pp. 135-149, March 1984
- [6] P. G. Howard and J. S. Vitter, "Arithmetic coding for data compression," *Proceedings of IEEE*, Vol. 82, No. 6, pp. 856~865, June 1994.
- [7] A. Moffat, R.M Neal, and H. Witten, "Arithmetic coding, revisited," *ACM Transaction on Information Systems*, 16(3), pp. 256~294, July 1998.
- [8] A. Said, "Introduction to arithmetic coding: theory and practice, technical report," *HPL-2004-76*, HP Laboratories, Palo Alto, USA, 2004.
- [9] W. B. Pennebaker, J. L. Mitchell, G. G. Langdon Jr., and R. B. Arps, "An overview of the basic principles of the Q-coder adaptive binary arithmetic coder," *IBM Journal of Research and Development*, Vol. 32, No. 6, pp. 717~726, 1988.
- [10] ITU-T Recommendation T.81, Information technology - digital compression and coding of continuous-tone still images - requirements and guidelines, 1993.
- [11] ITU-T Recommendation T.82, Information technology - coded representation of picture and audio information - progressive bi-level image compression, March 1993.
- [12] ITU-T Recommendation T.88, Information technology - lossy/lossless coding of bi-level images, 2000.
- [13] ITU-T Recommendation T.89, Application profiles for recommendation T.88 - lossy/lossless coding of bi-level images for facsimile, Sept. 2001.
- [14] ISO/IEC 15444-1:2004, Information technology - JPEG 2000 image coding system: Part 1: core coding system, 2004.
- [15] D. Marpe, Fast adaptive binary arithmetic coding (M-Coder) [internet] Available: <http://iphome.hhi.de/marpe/mcoder.htm>
- [16] D. Marpe and T. Wiegand, "A highly efficient multiplication-free binary arithmetic coder and its application in video coding," *Proceedings IEEE International Conference Image Processing (ICIP)*, Barcelona, Spain, September 2003.
- [17] D. Marpe, H. Schwartz, and T. Wiegand, "Context-based adaptive binary arithmetic coding in the H.264/AVC video compression standard," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 13, No. 17, pp. 620-636, July, 2003.
- [18] D. Marpe, G. Marten, and H. L. Cycon, "A fast renormalization technique for H.264/MPEG4-AVC arithmetic coding," in *Proceedings 51st Internationales Wissenschaftliches Kolloquium (IWK)*, Ilmenau University of Technology, Ilmenau: Germany, Sept. 2006
- [19] Michael Adams, JPEG2000 Jasper project, University of Victoria. [internet], Available: <http://www.ece.uvic.ca/~frodo/jasper/>



**고 형 화 (Hyung Hwa Ko)**

1989년 2월 : 서울대학교 전자공학과 (공학박사)  
 1985년 3월 ~ 현재: 광운대학교 전자통신공학과 교수  
 1998년 8월 ~ 1999년 7월: UCSD 교환교수  
 2007년 1월 ~ 2008년 1월: Princeton Univ. 교환교수  
 2008년 7월 ~ 2010년 1월: 광운대학교 전자정보공과대학장  
 ※ 관심분야 : 영상신호처리, JBIG2, JPEG2000, 산술부호기, 생체인식, 사물인터넷



**서 석 용 (Seok Yong Seo)**

1996년 2월 :관동대학교 전자통신공학과 (공학사)  
 2008년 8월:광운대학교 전자통신공학과 (공학석사)  
 2012년 2월:광운대학교 전자통신공학과 (공학박사)  
 2013년 9월~현재 : 경민대학교 정보통신과 조교수  
 ※ 관심분야 : 영상신호처리, JBIG2, MPEG