

논문 2004-41SD-4-10

고효율 엔트로피 부호화를 위한 ICMEP 알고리즘 설계

(Design of the ICMEP Algorithm for the Highly Efficient Entropy Encoding)

이 선 근*, 임 순 자*, 김 환 용*

(Seon-Keun Lee, Soon-Ja Lim, and Hwan-Yong Kim)

요 약

압축 과정 중에 빠른 처리율을 보여주는 렘펠-지브 알고리즘의 결합에 의하여 채널 전송율은 증가한다. 영상정보의 압축 과정 동안 처리율을 향상시키기 위하여 ICMEP 알고리즘을 제안하였으며 HDTV에 적용시켜 엔트로피 부호화를 설계하였으며 이를 검증하였다. ICMEP 엔트로피 압축기는 top-down 방식을 이용하여 설계하였으며 VHDL을 이용하여 회로 합성 및 모의 실험을 수행하였다. 모의실험 결과, 구현된 ICMEP 엔트로피 부호화기는 메모리 포화방지 및 압축률 증가로 인하여 전체적인 시스템 효율이 향상됨을 확인하였다.

Abstract

The channel transmission ratio is speeded up by the combination of the Huffman algorithm, the model scheme of the lossy transform having minimum average code lengths for the image information and good instantaneous decoding capability, with the Lempel-Ziv algorithm, showing the fast processing performance during the compression process. In order to increase the processing speed during the compression process, ICMEP algorithm is proposed and the entropy encoder of HDTV is designed and inspected. The ICMEP entropy encoder have been designed by choosing the top-down method and consisted of the source codes and the test benches by the behavior expression with VHDL. As a simulation results, implemented ICMEP entropy encoder confirmed that whole system efficiency by memory saturation prevention and compressibility increase improves.

Keywords : HDTV, Entropy encoder, Huffman, Lempel-Zip, ICMEP

I. 서 론

멀티미디어의 혁명은 디지털 신호처리, 광 전송, 반도체, 컴퓨터 등의 기술 발전에 힘입어 많은 발전을 이루었으며, 현대는 초고속 정보 통신망을 비롯한 여러 종류의 채널이나 매체를 통한 다양한 형태의 정보 서비스가 제공되고 새로운 수요가 창출되는 정보화 사회로 변모하고 있다.

영상정보의 비손실 압축은 정보내의 데이터 손실 없이 데이터 압축이 가능하며 손실압축과는 달리 입력정보의 특성과 상관없이 임의의 데이터에 대해 정보 압축

이 가능하고 렘펠-지브(Lempel-Ziv) 방식이 대부분의 압축방식에 응용되고 있다. 대표적인 비손실 압축방식으로는 PKZIP, GZIP, COMPRESS, 영상정보 압축의 GIF 등이 있으며 현재 대부분의 응용방식은 구현상의 복잡도 및 비용 문제로 하드웨어(hardware)가 아닌 소프트웨어(software)에 의존하고 있다.^{[1][2]}

영상압축의 일반적인 구조는 크게 세 단계로 나뉘어진다. 첫 번째 단계는 영상 화소간 상관관계를 제거하는 예측 및 변환 단계, 두 번째 단계는 상관관계를 제거한 영상 화소를 한정된 몇 개의 코드(code) 또는 심볼(symbol)로 바꾸는 양자화 단계, 그리고 양자화된 영상 화소를 최종적인 비트 형태로 표현하는 엔트로피(entropy) 부호화 단계로 나뉘어진다.^[3]

허프만 부호화는 최소의 평균 부호장을 가지며 순간적 디코딩 능력이 뛰어나기 때문에 소프트웨어 구현시

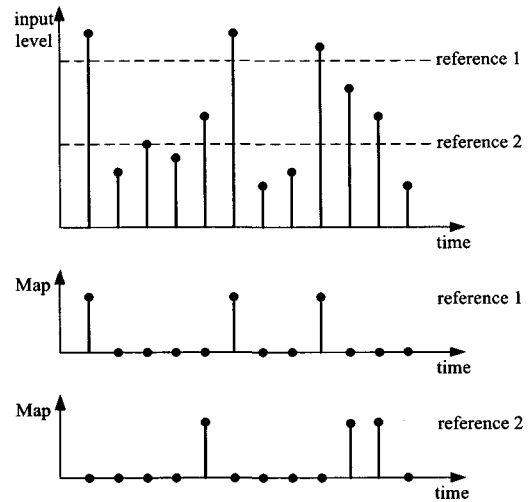
* 정회원, 원광대학교 전기전자및정보공학부
(Department of Electrical Electronic and Information Eng., Wonkwang University)
접수일자 : 2003년7월22일, 수정완료일: 2004년3월19일

보다 하드웨어 구현시 압축의 효율성이 높은 장점이 있고 램펠-지브 부호화는 데이터 단위가 아니라 데이터열 단위로 부호를 할당하기 때문에 압축과정의 수행속도를 증가시킬 수 있다. 그러므로 본 논문에서는 하드웨어 효율성과 압축효율의 극대화를 구현하고자 ICMEP (image compressing method for the entropy performance) 알고리즘을 제안하고 제안된 알고리즘을 이용하여 HDTV에 적용할 수 있는 엔트로피 부호화기를 설계하여 ICMEP 알고리즘에 대한 검증을 수행하였다. 설계된 ICMEP 엔트로피 부호화기는 영상정보의 처리효율을 높이기 위하여 데이터의 압축에 사용되는 비손실 압축방식을 적용하여 설계함으로써 영상 정보의 보상과정이 축소되어 엔트로피 부호화기의 처리효율이 증가함을 확인하였고, ICMEP 알고리즘을 적용한 엔트로피 부호화기는 부호화과정과 압축과정이 분리되지 않고 부호화를 수행하면서 압축과정이 동시에 수행되기 때문에 엔트로피 부호화기 구현시의 비용절감을 크게 증대됨을 확인하였다.^{[4][5]}

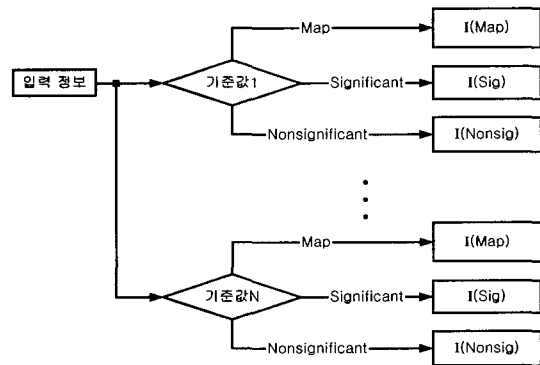
II. 제안된 ICMEP 알고리즘

대용량의 영상 데이터 처리 및 전송효율을 하드웨어적인 측면에서 극대화하기 위하여 본 논문에서는 ICMEP 알고리즘을 제안한다. 본 논문에서 제안한 ICMEP 알고리즘은 기존 허프만 부호화 방식에 대하여 부호 테이블 및 사전(dictionary)을 이용하여 영상 데이터를 처리하도록 하였다. 따라서 부호화와 압축과정에 소요되는 처리시간 및 영상 데이터를 처리하는 과정 중에 발생하는 병목현상에 의한 처리시간에 대하여 처리효율이 향상되도록 하였다. 또한 처리속도 향상 및 메모리 포화 방지를 위하여 데이터에 대한 생성과 재생을 반복 사용함으로써 압축을 수행하고 복원하는데 걸리는 시간을 줄일 수 있는 방법에 대해 고찰한다. 이러한 처리시간의 감소는 제한된 전송로 상에서 실시간 영상정보의 흐름을 원활하게 수행하여 대용량 데이터에 대한 처리한계를 극복할 수 있다는 장점이 있다.

이진 분류기(binary classification)는 부가정보의 선택여부에 따라 이진 순방향 분류기와 이진 역방향 분류기의 두 종류가 있다. 이러한 이진 분류기의 특징은 부호화와 복호화의 복잡도가 낮으며 효율적인 엔트로피 부호화를 수행할 수 있다. 그러나 이러한 이진 분류기는 정보의 분류를 두 가지 종류로써 구별하여 부호화를 수행하기 때문에 정보의 상관성이 크고 부호 복을 생성



(a) 시간 축 상에서의 MOCC



(b) MOCC 블록도

그림 1. 다치 단방향 분류 부호기
Fig. 1. Multi-value one-way classification coder.

하는 정보의 양이 방대해질 수 있다. 이러한 특징은 정보의 원활한 흐름을 방해하는 병목현상을 유발하게 되며 영상정보와 같은 데이터량이 많은 정보에 대한 처리시 더욱 두드러진다. 이런 결점을 제거하기 위하여 제안한 ICMEP 알고리즘은 단순 이진 분류가 아닌 다치 분류를 선택하였다. 다치 분류는 메모리 사용을 극대화하게 되며 데이터의 지연을 제거할 수 있기 때문에 병목현상을 없앨 수 있다.

다치 단방향 분류 부호기(multi-value one-way classification coder : MOCC)는 현재의 심볼을 대상으로 적절하게 설정된 범주로 분류하고 분류된 정보를 부가 정보로써 사용하는 방법으로써 그림 1은 MOCC의 블록도를 나타내며 그림 1 (a)는 입력 정보원을 적당한 기준값(reference value)과 비교해서 입력 심볼들을 여러 가지 범주로 분류하고 매 심볼마다 1 비트씩의 분류 부가 정보를 갖게 된다. 그림 1 (b)에서 입력 심볼은 세 가지 종류의 정보와 각각에 대하여 기준값의 개수만큼

의 정보로 나누어진다. 즉 기준값과 비교하여 일정범위보다 큰 값들인 significant symbol (*Sig*)와 기준값보다 작은 값들인 non-significant symbol (*Nonsig*), 그리고 분류 부가 정보인 binary classification map으로 분류되는 정보종류와 각각에 대하여 기준값이 달라질 경우에 해당하는 정보들로 나눌 수 있다. 입력 정보원 X 집합에 대한 정보량을 $I(X)$ 로 정의하고 큰 값으로 분류된 정보원 *Sig* 집합의 정보량을 $I(Sig)$ 로, 작은 값으로 분류된 정보원 *Nonsig* 집합의 정보량을 $I(Nonsig)$ 로, 그리고 이진 분류 부가 정보원 *Map* 집합의 정보량을 $I(Map)$ 으로 정의하면 MOCC는 그림 1과 같다.

심볼 수를 M , 입력 시퀀스의 수를 N , 문턱값을 K , 심볼의 확률을 p_i 라 할 때 매 입력 심볼마다 분류 정보를 부가하므로 *Map* 집합의 크기는 N 이 되고 *Map* 집합의 심볼인 '0'과 '1'의 각각의 확률 값은 식 (1)과 식 (2)로 구해진다. 또한 *Sig* 집합의 크기는 전체크기에서 *Sig* 집합이 되는 확률 값을 곱한 $N \times PM_1$ 이 되고 *No-sig* 집합의 크기는 전체 크기에서 *Sig* 집합이 되는 확률 값을 곱한 $N \times PM_0$ 가 된다.

$$P_{map}\{0\} = \sum_{i=0}^{k-1} p_i = PM_0 \quad (1)$$

$$P_{map}\{1\} = \sum_{i=k}^{M-1} p_i = PM_1 \quad (2)$$

Map 집합의 정보량은 식 (3)과 같이 *Map* 집합의 크기와 *Map* 집합의 엔트로피와의 곱으로 표현 할 수 있다.

$$I(Map) = Size(Map) \times H(Map) = N \times (-PM_0 \log_2 PM_0 - PM_1 \log_2 PM_1) \quad (3)$$

유사한 방법으로 *Sig* 집합의 정보량과 *Nonsig* 집합의 정보량을 계산하면 각각 식 (4)와 식 (5)와 같다.

$$I(Sig) = Size(Sig) \times H(Sig) = N \times PM_1 \times \left(- \sum_{i=k}^M \frac{p_i}{PM_1} \log_2 \frac{p_i}{PM_1} \right) \quad (4)$$

$$I(Nonsig) = Size(Nonsig) \times H(Nonsig) = N \times PM_0 \times \left(- \sum_{i=0}^{k-1} \frac{p_i}{PM_0} \log_2 \frac{p_i}{PM_0} \right) \quad (5)$$

입력 정보원은 세 가지 종류의 정보원으로 분류되고 각각의 정보원들에 대한 정보량은 식 (3)과 식 (4), 식 (5)와 같이 된다. 그러므로 전체 입력정보원은 각각의

정보원들에 대한 대수적인 합으로 표현이 가능하다. 즉, 식 (6)과 같이 표현할 수 있으며 여기에서 식 (6)을 정리하면 식 (7)이 얻어진다.

$$\begin{aligned} & N \times (-PM_0 \log_2 PM_0 - PM_1 \log_2 PM_1) \\ & + N \times PM_1 \times \left(- \sum_{i=k}^M \frac{p_i}{PM_1} \log_2 \frac{p_i}{PM_1} \right) \\ & + N \times PM_0 \times \left(- \sum_{i=0}^{k-1} \frac{p_i}{PM_0} \log_2 \frac{p_i}{PM_0} \right) \quad (6) \\ & = N \times (-PM_0 \log_2 PM_0 - PM_1 \log_2 PM_1) + \\ & N \times \left(- \sum_{i=k}^M p_i \log_2 \frac{p_i}{PM_1} \right) + N \times \left(- \sum_{i=0}^{k-1} p_i \log_2 \frac{p_i}{PM_0} \right) \\ & = N \times (-PM_0 \log_2 PM_0 - PM_1 \log_2 PM_1) + \\ & - \left\{ N \times \sum_{i=k}^M p_i \log_2 p_i - PM_1 \log_2 PM_1 + \right. \\ & \left. N \times \sum_{i=0}^{k-1} p_i \log_2 p_i - PM_0 \log_2 PM_0 \right\} \\ & = - \left(N \times \sum_{i=k}^M p_i \log_2 p_i + N \times \sum_{i=0}^{k-1} p_i \log_2 p_i \right) \\ & = - N \times \sum_{i=0}^{M-1} p_i \log_2 p_i \quad (7) \end{aligned}$$

식 (7)은 입력 신호원에 대한 전체 정보량이 되므로 분류된 세 종류의 정보원에 대한 전체 정보량은 식 (8)과 같다.

$$I(X) = I(Sig) + I(Nonsig) + I(Map) \quad (8)$$

식 (8)에서와 같이 다치 단방향 분류는 입력 신호원의 정보량에 대한 변화 없이 신호원을 단순히 분류하고 매 심볼마다 분류를 위한 부가 정보를 추가하게 되지만 분류 전 정보량과 분류 후의 전체 정보량에는 변화가 없다. 즉, 정보량의 변화가 없기 때문에 기준값을 임의의 개수로써 다치 분류를 설정하여 정보량을 분산하여도 전체에 대한 정보량에 대한 손실은 없게 된다.

허프만 부호기법은 가변길이 부호어를 저장하고 있는 허프만 테이블을 사용하여 입력 심볼을 부호화한다. 신호원의 크기가 작은 경우에는 발생 가능한 모든 심볼들에 대해 서로 다른 가변길이 부호어를 사용한다. 그러나 신호원의 크기가 매우 큰 경우에는 많은 양의 메모리를 요구하므로 발생빈도가 높은 일부 심볼에 대해서만 서로 다른 가변길이 부호어를 사용한다. 일반적으로 허프만 테이블의 크기가 커짐에 따라 평균 비트율은 감소하지만 요구되는 메모리의 양은 증가하게 된다.

제한된 메모리로 인해 사용할 수 있는 허프만 테이블의 개수가 신호원의 개수보다 적은 경우에는 몇몇 신호원들에 대해서는 하나의 공통된 허프만 테이블이 사용

되는 허프만 테이블의 공유가 필요하다. 또한 허프만 공유 테이블을 사용하는 과정 중에서 지시함수의 오판으로 인한 메모리의 미사용 영역이 존재하게 되어 한정된 메모리에 대한 비효율적인 현상이 발생하게 된다. 이상과 같은 허프만 알고리즘의 단점을 보완하기 위하여 ICMEP 알고리즘에서의 발생확률에 따른 엔트로피 부호화방법을 제안한다. 반복강화 알고리즘의 성능을 개선하기 위해서 테이블과 dictionary에 대한 재사용 및 포화방지, 초기화, 고정길이 부호화 방법이 있고 반복강화 알고리즘에서 요구되는 계산량을 줄이기 위해서는 부호어 길이의 근사방법이 있다.

모든 $m = 1, 2, \dots, M$ 에 대하여 $\mu(m) \neq n$ 를 만족하는 정수 $n \in \{1, \dots, N\}$ 를 발생시킬 수 있다. 이는 곧 허프만 테이블 H_n 가 어떤 신호원에 대해서도 적절하게 설계되지 못했음을 의미한다. 이 경우에는, 인덱스 데이터가 영을 의미하므로 확률 분포 $F'_n = \{P'_n(k) | k = 1, \dots, K\}$ 가 적절히 정의되지 않는다. 따라서 허프만 테이블 H_n 내의 부호어들의 길이는 계산될 수 없게 되어, 테이블 H_n 는 다음 단계의 과정을 수행할 수 없게 된다. 이와 같이 기존의 제한된 반복강화 알고리즘에서는 미사용 허프만 테이블이 일단 발생하면 이 테이블은 계속해서 사용될 수 없게 되어 결과적으로 주어진 메모리를 충분히 활용하지 못하는 결과를 낳게 된다. 이러한 미사용 허프만 테이블을 재사용하기 위한 방법으로 M 개의 신호원들 중 하나의 신호원 S_r 를 적절히 선택하여 $\mu(r) = u$ 로 놓는다. 이와 같은 처리는 반복계산 단계에서 H_u 가 S_r 에 최적화 되도록 하여 다음 단계에서 H_u 가 최소한 S_r 의 부호화에는 사용될 수 있도록 한다. 이때 신호원 S_r 의 선택은 처리 방법이 평균 비트율의 감소량을 미사용 테이블 처리 이득 $g(m)$ 이라 하면 식 (9)와 같이 나타낼 수 있다.

$$g(m) = \begin{cases} 0, & I_{\mu(m)} = \{m\} \text{일 때} \\ P'_{\mu(m)} \bar{\sigma}_1(F'_{\mu(m)}) \\ - (P'_{\mu(m)} - P'_m) \bar{\sigma}_1(G_{\mu(m), m}) \\ - P'_m \bar{\sigma}_1(F_m), & I_{\mu(m)} \neq \{m\} \text{일 때} \end{cases} \quad (9)$$

단, $P'_n = \sum_{i \in 0} P_i$ 이고, $G_{\mu(m), m}$ 는 $S'_{\mu(m)}$ 에서

S_m 을 제외함으로써 얻어지는 합성 신호원의 확률 분포이다. 만일 $G_{\mu(m), m} = \{q(1), \dots, q(K)\}$ 로 표시한다면 식 (10)과 같이 주어진다.

$$q(k) = \frac{P'_{\mu(m)} \cdot p'_{\mu(m)}(k) - P'_m \cdot p_m(k)}{P'_{\mu(m)} - P'_m}, \quad (10)$$

$$k = 1, 2, \dots, K$$

식 (9)에서 $\bar{\sigma}_l(F)$ 는 확률분포가 F 인 신호원을 그 확률 분포에 적합하도록 설계된 크기 l 인 허프만 테이블을 사용하여 부호화한 경우의 평균 비트율을 의미한다. 식 (9)에서와 같이 확률분포가 어떠한 신호원에 대해서도 일정한 값을 가지도록 할 경우 허프만 테이블의 미사용 부분은 사라지게 될 것이다. 또한 정보원에 대한 부호화를 수행하기 위하여 사용되어지는 허프만 테이블의 확률분포가 일정하게 되면 압축을 위해 사용되는 dictionary 또한 공유하여 사용할 수 있게 되며, 이는 허프만 테이블과 dictionary 사이에서 상호간에 미사용 테이블이 존재하는지, dictionary가 포화되는지를 자동적으로 확인하고 감시할 수 있게 된다. 이는 가변길이 부호화 방식을 사용하는 부호기의 메모리에 대한 효율을 극대화할 수 있다는 것을 의미하게 되며 이러한 알고리즘을 구현할 경우 소비되는 비용 역시 크게 감소하게 된다.

그림 2는 허프만 테이블과 dictionary와의 메모리 할당 맵을 보여주고 있다. 외부로부터의 정보원이 인가되면 dictionary 영역의 정보원과 정합여부를 판별한 후 허프만 테이블 영역, dictionary 영역, 업데이트 영역 중의 어느 한곳으로 진로가 결정되며 정보원이 인가될 때마다 스택 포인터는 메모리의 전체 영역을 전역 탐색(full search)하면서 메모리에 대한 포화 및 미사용 영역을 제어하게 된다. 또한 스택 포인터는 식 (9)에서와 같

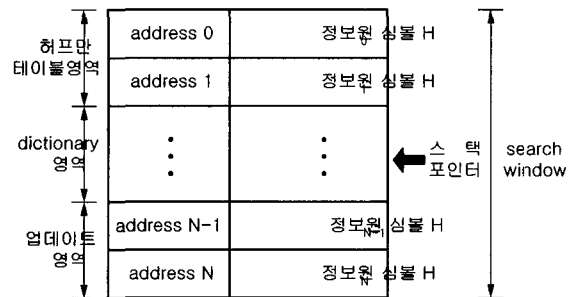


그림 2. 허프만 테이블과 dictionary 공유 메모리
Fig. 2. Concatenation memory with the Huffman table and dictionary.

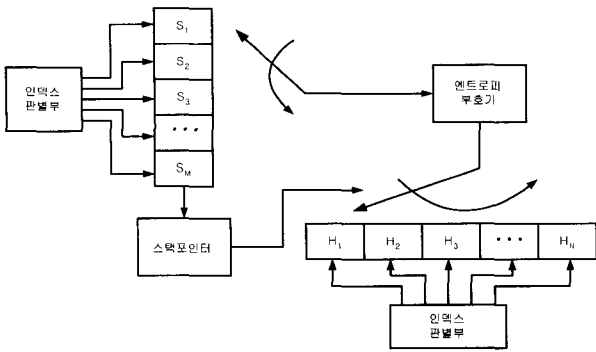


그림 3. 다차원 정보원에 대한 엔트로피 부호기
Fig. 3. An entropy encoder for multi-dimension source.

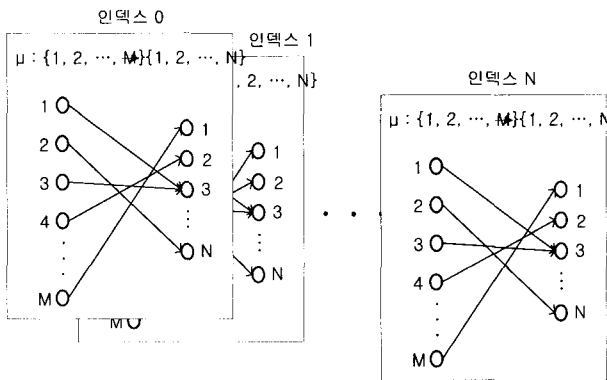


그림 4. 다차원 정보원들에 대한 지시함수의 예
Fig. 4. Example of multi-pointer function for sources.

이 메모리에 대한 효율을 증가시키기 위하여 항상 $g(m)$ 이 '0'이 되도록 address를 할당하게 된다.

사용할 수 있는 허프만 테이블의 개수가 신호원의 개수보다 매우 작은 경우 ($N \ll M$) 즉, 영상 데이터와 같이 데이터 양이 한정된 메모리에 비하여 매우 큰 경우의 신호들에 대해서는 허프만 테이블의 수를 증가시킨다 하더라도 비효율적이며 적절하지 않을 수 있다. 그래서 가변길이보다는 고정길이 부호화되는 것이 유리하다. 성능향상 정도는 N 값이 작을수록 커진다. 그러므로 고정길이 부호화를 위한 ICMEP 알고리즘의 기능은 식 (11)과 같은 표현이 가능하다.

$$\mu(m) = \arg \min_{1 \leq n \leq N} b_m^n \cdot I_{IC}, \quad (11)$$

여기서 $m = 1, 2, \dots, M$

여기에서 I_{IC} 는 ICMEP 알고리즘에서 인덱스 집합을 의미한다. 즉, 인덱스의 선택에 따라서 지시함수가 탐색해야할 메모리 공간의 영역이 변화하도록 한다. 그림 2에서 스택 포인트는 지시함수 $\mu(m)$ 을 식 (11)과 같이 변형하였을 경우 허프만 영역, dictionary 영역, 업

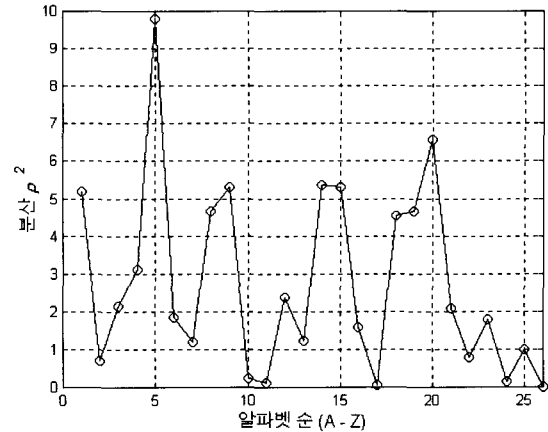


그림 5. 영문 텍스트의 분산
Fig. 5. Dispersion of alphabet text.

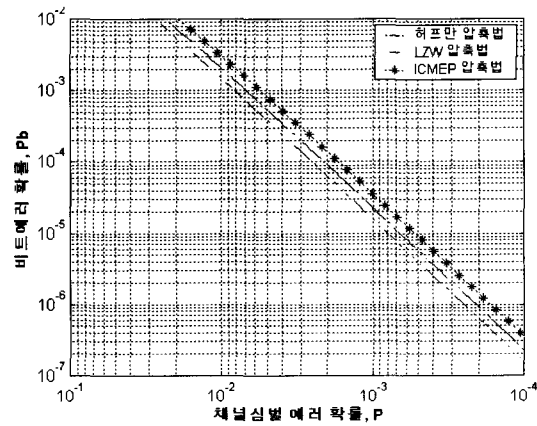


그림 6. 압축부호화방식의 에러확률 비교
Fig. 6. Error probability comparison of compression encoding method.

데이트 영역의 진행 영역을 결정짓는 지시함수로 변환다. 즉, ICMEP에서의 스택 포인트는 지시함수의 기능을 포함하게 된다.

만약, $b_m^{\mu(m)} \geq \log_2 K$ 이면 $\mu(m) = 0$ 으로 놓는다.

그림 3과 그림 4는 인덱스 집합으로 인한 지시함수의 변화를 나타낸다. 인덱스 판별부가 첨가되어 엔트로피 부호화기는 더욱 많은 정보에 대하여 선별할 수 있는 기능을 가지게 되며 이러한 지시함수의 집합인 스택 포인트는 매우 효율적인 메모리 관리를 수행할 수 있게 된다.

본 논문에서는 영문과 국문 텍스트에 대해서 발생빈도 및 압축률을 조사하고 ICMEP을 적용하였을 경우, 압축률을 조사하였다. 바바라와 부두의 영상과 텍스트에 대한 허프만, LZW, 그리고 ICMEP의 상호 성능평가를 위하여 Matlab을 이용하여 모델링을 수행하였으며 이에 대한 결과는 그림 5, 6, 7로 표현하였다.

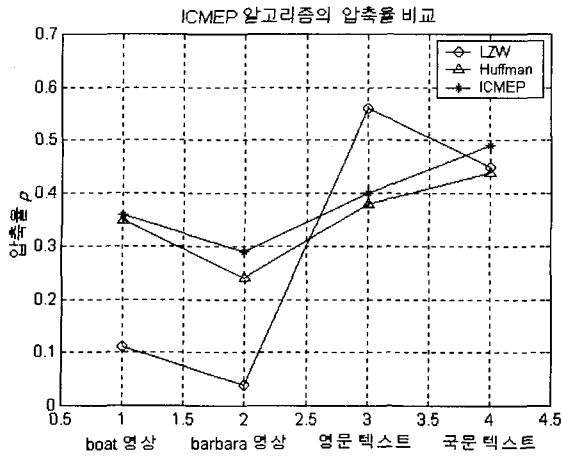


그림 7. ICMEP 알고리즘의 압축률 비교
Fig. 7. Compression ratio comparison of ICMEP algorithm.

표 1. ICMEP 알고리즘의 압축 부호화률
Table 1. Compression encoding ratio of ICMEP algorithm.

| 구분 | 원문 [Kbyte] | LZW [Kbyte] | Huffman [Kbyte] | ICMEP [Kbyte] |
|---------|------------|-------------|-----------------|---------------|
| boat | 409 | 266 | 368 | 260 |
| barbara | 409 | 311 | 392 | 291 |
| 영문 텍스트 | 665 | 290 | 411 | 400 |
| 국문 텍스트 | 728 | 406 | 410 | 375 |

표 1은 ICMEP 알고리즘의 압축 부호화에 대한 압축률을 보이고 있다. 텍스트인 경우 허프만, ICMEP, LZW 알고리즘의 순서로 압축이 이루어지며 ICMEP에 대해서 영문보다는 국문이 보다 더 압축률이 높음을 확인하였다. 영상인 경우, 손실 압축방식인 허프만 부호화 방식이 비손실 압축방식인 LZW에 비하여 압축률이 높으며 ICMEP 부호화 방식이 허프만 부호화 방식에 비하여 더욱 높은 압축률을 보임을 확인하였다.

즉, ICMEP 알고리즘을 이용하여 부호화를 수행할 경우, 영상정보와 같이 데이터 량이 많은 경우에는 손실압축방식인 허프만 부호화보다 유리하다는 것을 알 수 있었으며 대용량 데이터의 효율적인 부호화에 허프만 부호화, LZW 부호화보다는 ICMEP 알고리즘이 더욱 적합하다는 것을 확인할 수 있었다.

이상과 같이 ICMEP 알고리즘의 부호화 및 압축 부분은 다음과 같이 구성된다. 첫째로 다차 단방향 분류 기능이다. 기준값을 여러 개 설정하여 인덱스 집합을 크게 함으로써 정보량은 변하지 않으면서 정보원을 분류한다. 둘째로는 정보원의 발생확률에 따른 엔트로피

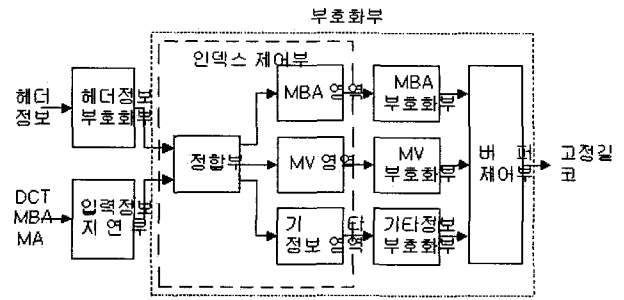


그림 8. 제안된 ICMEP 알고리즘에 의한 엔트로피 부호화기의 블록도
Fig. 8. Proposed block diagram of ICMEP entropy encoder.V

부호화를 수행한다. 즉, 허프만 테이블과 dictionary를 동시에 사용함으로써 메모리 내부에서 사용되지 않는 메모리를 없애고 가변길이 부호화를 수행하는 과정에서 발생하게 되는 메모리의 포화상태를 방지하도록 하였다. 허프만 테이블과 dictionary의 제어는 지시함수를 인덱스 집합을 이용하여 스택 포인트로 변환하여 사용한다.

III. ICMEP 알고리즘에 의한 엔트로피 부호화기 설계

본 논문에서는 영상 데이터와 같이 용량이 매우 큰 정보에 대한 처리효율을 증대시키고자 제안한 ICMEP 알고리즘을 이용하여 엔트로피 부호화기를 설계하였다.

ICMEP 알고리즘의 주요기능인 가변길이 부호화 기능과 압축 기능을 하나의 모듈로써 설계하였으며 허프만 테이블과 dictionary 공유 메모리를 첨가하여 구현하였다. 또한 설계된 ICMEP 알고리즘에 의한 엔트로피 부호화기는 허프만 테이블과 dictionary의 영역을 구별하여 사용하도록 하였으며 영역의 구별 및 제어는 스택 포인트를 사용하였다. 또한 입력 정보원과 내부에 존재하게되는 기준값들에 대하여 정합 기능을 주어 스택 포인트로써 부호화와 압축 기능을 동시에 수행하면서 메모리의 포화 및 미사용에 대한 비효율적인 요소를 배제하였다.

ICMEP 알고리즘에 의한 엔트로피 부호화기는 그림 8과 같이 입력 정보 지연부와 헤더 정보 부호화부, 부호화부로 되어있으며 부호화부는 인덱스 제어부와 하부 기능 부호화부, 비퍼 제어부로 구성되어 있다.

(1) 인덱스 제어부

그림 9는 인덱스 제어부로서 헤더정보의 부호화 과

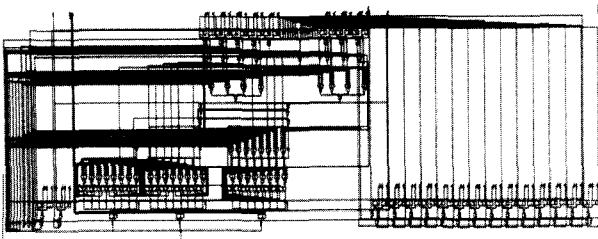


그림 9. 인덱스 제어부의 회로도
Fig. 9. Designed circuit of index control.

정 중에 발생되어지는 메모리의 비효율적인 사용으로 인한 메모리부의 메모리 효율 증가 및 병목현상을 제거하는 기능을 수행하는 부분이다. 8 비트 데이터의 입력을 받고 24 비트의 어드레스를 갖는 RAM을 이용하여 입력정보원이 dictionary에 포함된 내용인지 아닌지의 여부에 따라 압축과 부호화를 수행할 것인지, 부호화만 수행할지를 결정하게 된다. $2^{24} = 16,777,216$ 개의 어드레스는 캡처된 데이터에 대한 전역탐색을 수행하게 되는 스택 포인터로서 기능을 수행하게 된다. 입력값이 저장된 값이 아닐 경우에는 메모리에 새롭게 업데이트 되는 동시에 저장된 값과의 차이를 메모리에 별도로 저장하여 다음에 유입되어지는 데이터와의 정합여부를 판별하기 위하여 다음 데이터를 기다리게 된다. 만약 저장된 데이터와 입력 데이터가 일치할 경우 저장되어진 데이터에 해당하는 정보원을 부호화하기 위하여 메모리로부터 출력된다. 이러한 기능을 수행하게되면 메모리의 전체 어드레스를 전역 탐색하게 되고 이것은 대용량의 정보값에 대하여 효율적인 메모리 관리가 가능해진다. 즉, 미사용 메모리 및 메모리 포화 상태를 억제시킴으로써 버퍼 제어부에서의 병목현상을 제거할 수 있기 때문에 처리속도가 향상된다.

그림 10은 인덱스 제어부에 대한 모의실험 결과 파형이다. CONTROL_LOAD와 OE 신호에 따라서 내부 메모리 어드레스가 동작을 수행하게 되며 내부 메모리 내용과 유입되는 입력신호와의 정합여부를 판별하여 설정된 내부 메모리 영역에 저장되게 된다. 이때 메모리의 읽기와 쓰기 동작이 작동될 경우에는 다른 블록과의 데이터 충돌을 피하기 위하여 high 임피던스가 출력되도록 설정하였다.

(2) ICMEP 알고리즘에 의한 엔트로피 부호화부

ICMEP 알고리즘에 의한 엔트로피 부호화부의 전체 회로는 그림 11과 같다. VLC로 입력되는 외부 입력 신호들 중 pesh_val과 pesh는 pesh_clk의 상승 에지 동기를 맞추어 입력되며 caption과 caption_vld는 caption

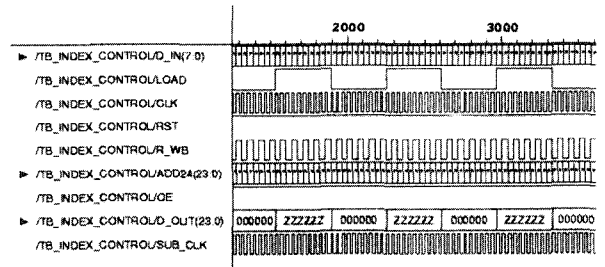


그림 10. 인덱스 제어부의 모의실험
Fig. 10. Simulation of index control.

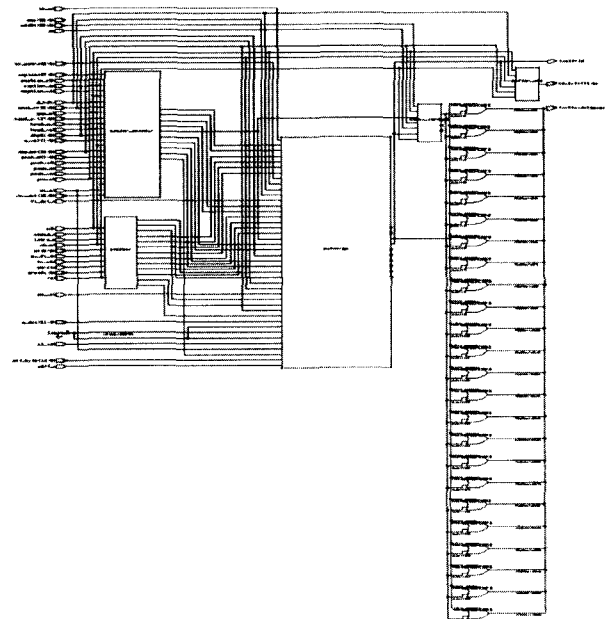


그림 11. ICMEP 엔트로피 부호화기의 회로도
Fig. 11. Circuit of ICMEP entropy encoder.

_clk의 상승 에지에 맞추어 입력된다. 나머지 입력들은 시스템 클럭인 clk의 하강 에지에 동기를 맞추게 설계하였다. 내부신호들은 head_encoder와 indelay에서 출력되는 신호들은 상승 에지일 때 출력되어 이를 입력으로 받은 부호화부(submerge) 역시 packer와 pac_buf만을 제외하고 모든 입출력이 상승 에지에서 동작하게 된다.

ICMEP 엔트로피 부호화부는 top_down 방식으로 설계하였으며 VHDL을 이용하여 소스 부호와 테스트 벤치를 구성하였다. ICMEP 엔트로피 부호화부에 대한 모의실험 결과는 그림 12에 나타났다.

회로합성에는 SYNOPSIS Ver. 1999.10 Design Analyzer를 이용하였으며 모의실험은 SYNOPSIS Ver. 1999.10 vhdldb를 이용하여 수행하였다. 모의실험 결과 기존의 엔트로피 부호화기의 전체 게이트 수는 91,768개이며 제안된 알고리즘인 ICMEP을 이용한 엔트로피 부호화기의 게이트 수는 117,463개로써 기존에 비하여 28% 증가함을 확인하였다.

