

논문 2004-41SP-1-7

# Huffman 부호에 기초한 대칭적 양방향 가변길이 부호의 설계 방법

## (Construction of Symmetrical Reversible Variable-Length Codes from the Huffman Code)

정 옥 현\*, 호 요 성\*\*

(Wook-Hyun Jeong and Yo-Sung Ho)

### 요 약

가변길이 부호는 정보원(source)의 통계적인 특성을 이용하여 부호화 효율을 높이지만, 잡음이 심한 전송 환경에서 비트 오류가 발생하면 데이터를 제대로 복원할 수 없다는 단점이 있다. 최근 가변길이 부호의 문제점을 해결하기 위해 전송 오류로 인해 손상된 비디오 비트열에서 올바른 데이터를 복구할 수 있는 양방향 가변길이 부호가 제안되었다. 이러한 양방향 가변길이 부호는 부호 구성에 따라 대칭적 양방향 가변길이 부호와 비대칭적 양방향 가변길이 부호로 나눌 수 있다. 대칭적 양방향 가변길이 부호는 비대칭적 양방향 가변길이보다 평균부호길이는 더 길지만 구현이 쉽고 메모리 사용에 장점을 가진다. 그러나, 기존의 대칭적 양방향 가변길이 부호의 생성 방법은 구현하기에 복잡하고 부호화 효율성도 다소 개선될 여지가 있다. 본 논문에서는 최적화된 Huffman 부호 테이블을 통해 대칭적 양방향 가변길이 부호를 생성하는 새로운 알고리즘을 제안한다. 제안된 알고리즘은 기존의 알고리즘보다 부호의 생성과정이 간단하며 평균부호길이가 줄어들어 향상된 성능을 보여준다.

### Abstract

Although variable-length codes (VLCs) increase coding efficiency using the statistical characteristics of the source data, they have catastrophic effects from bit errors in noisy transmission environments. In order to overcome this problem with VLCs, reversible variable-length codes (RVLCs) have recently been proposed owing to their data recovering capability. RVLCs can be divided into two categories: symmetrical and asymmetrical RVLCs. Although the symmetrical RVLC has generally more overheads than the asymmetrical RVLC, it has some advantages of simpler design and more efficient memory usage. However, existing symmetrical RVLCs still have high complexity in their code design and some room for improvement in coding efficiency. In this paper, we propose a new algorithm for constructing a symmetrical RVLC from the optimal Huffman code table. The proposed algorithm has a simpler design process and also demonstrates improved performance in terms of the average codeword length relative to the existing symmetrical RVLC algorithms.

**Keywords:** Error resilience, H.263, Huffman code, MPEG-4, Reversible variable length code (RVLC).

### I. 서 론

최근 각광을 받고 있는 인터넷 상의 대화형 비디오 서비스, 무선 네트워크에서의 개인형 영상정보 서비스, 위성 및 지상파 방송을 통한 멀티미디어 정보 서비스

시스템에서 QoS(Quality of Service)를 보장하기 위한 많은 연구가 진행되고 있다. 대부분의 네트워크에서는 간단한 오류처리 기법을 제공하고 있지만, 실시간 서비스가 요구되는 시스템이나 피드백(feedback) 채널을 사용하지 않는 시스템에서는 멀티미디어 데이터에 대한 전송오류를 복구하는 기술이 매우 필요하다. 따라서 압축된 비디오 비트열을 전송 오류에 대해 강인하도록 하기 위한 오류 내성 부호화가 많은 관심을 받고 있다<sup>[1]</sup>.

\* 학생회원 \*\*정회원, 광주과학기술원 정보통신공학과  
(Department of Information and communications, Kwangju Institute Of Science and Technology)

접수일자 : 2003년9월5일, 수정완료일 : 2003년12월26일

대부분의 정지영상 및 동영상 부호화 방식에서는 압축 효율을 높이기 위해 Huffman 부호<sup>[2]</sup>나 산술 부호<sup>[3]</sup>와 같은 가변길이 부호(variable length code, VLC)를 사용하고 있다. 가변길이부호는 소스 엔트로피에 근접한 높은 압축 효율을 제공하지만, 무선 환경과 같이 잡음이 심한 채널에서는 예측 부호화와 더불어 전송 오류에 대하여 매우 취약하다는 단점이 있다. 만약 가변길이 부호로 부호화된 비트열에 비트 오류가 발생하면, 다음 동기점 사이의 모든 데이터는 그대로 사용할 수 없기 때문에 버려진다.

이러한 문제점을 고려하여, 최근 압축된 비트열에 대한 채널 오류의 영향을 줄이기 위해 양방향 가변길이 부호(reversible variable-length code, RVLC)가 제안되었다. 전송 오류가 발생했을 때, RVLC 복호기는 VLC의 경우와는 달리 순방향과 역방향으로 비트열을 복호하여 수신된 데이터로부터 손상되지 않은 데이터를 가능한 많이 복원할 수 있다. 따라서, 보다 향상된 오류처리 능력을 요구하는 MPEG-4와 H.263+ 같은 새로운 비디오 부호화 표준에서는 양방향 가변길이 부호화를 위한 도구들이 채택되었다.

RVLC는 부호어(codeword) 구성의 모양에 따라서 대칭적(symmetrical) RVLC와 비대칭적(asymmetrical) RVLC로 분류할 수 있다. 비대칭적 RVLC는 부호 할당이 유연하여 대칭적 RVLC보다 짧은 평균부호길이를 가지지만, 두 개의 부호어 테이블을 요구한다는 단점이 있다. 반면, 대칭적 RVLC는 비대칭적 RVLC보다 일반적으로 평균부호길이가 길지만, 상대적으로 구현하기 쉽고, 순방향과 역방향의 복호 과정에 하나의 부호어 테이블을 사용하며, 단일 비트 오류에 대한 처리 능력이 뛰어나다는 장점을 가지고 있다. 비대칭적 RVLC는 MPEG-4 표준에 채택되었고, 대칭적 RVLC는 H.263+에 채택되어 사용되고 있다<sup>[4, 5]</sup>.

Fraenkel 등<sup>[6]</sup>은 RVLC의 복호에 대한 필요조건과 함께 주어진 부호길이 집합으로부터 RVLC를 생성하는 알고리즘을 발표하였다. Wen 등<sup>[7]</sup>은 Golomb-Rice 부호와 Exp-Golomb 부호와 동일한 길이 분포를 갖는 비대칭적 RVLC에 대한 새로운 설계 기법을 제시하였다. Takishima 등<sup>[8]</sup>은 주어진 Huffman 부호에 기초하여 평균부호길이가 소스 엔트로피에 가깝도록 설계된 대칭적 RVLC 생성 알고리즘을 처음으로 제안하였고, Tsai 등<sup>[9]</sup>은 이 알고리즘을 수정하여 보다 짧은 평균부호길이를 갖는 대칭적 RVLC를 생성하는 방법을 제안하였다.

하지만, 이 두 알고리즘은 몇 가지 문제점과 제한 사항을 가진다. 이 알고리즘들은 전체 이진나무(binary tree)에서 부호를 찾아야 하고, 부호 생성 이전에 주어진 Huffman 부호를 기초로 모든 레벨(level)에서 사용 가능한 대칭적 RVLC의 수를 미리 계산해야 한다. 더구나, 이를 Huffman 부호의 모든 레벨에 걸쳐 적용시켜야 하는 복잡성이 존재하고, 적용 과정의 제한 사항 때문에 사용할 수 있는 대칭적 RVLC를 놓치는 경우가 종종 발생할 수 있다.

본 논문에서는 Huffman 부호 테이블로부터 대칭적 RVLC를 설계하는 새로운 알고리즘을 제안한다. 제안된 기법은 전체 이진나무의 반쪽 영역에서 부호를 검색하고, 주어진 Huffman 부호에 대한 적용과정을 매우 단순화시켜 복잡도를 줄인다. 동시에 제안된 적용과정 자체가 평균부호길이를 줄이는 데에 기여하고, 사용 가능한 대칭적 부호어들을 놓치지 않아서 기존의 방법보다 좀더 효율적인 대칭적 RVLC를 설계할 수 있다.

## II. 기존의 RVLC 알고리즘

### 1. Takishima 알고리즘

대칭적 RVLC는 주어진 최적의 Huffman 부호에서 출발하여 그 부호들을 상위 레벨에서부터 대칭적 RVLC로 바꾸어 가며 생성된다.

가변길이부호 혹은 RVLC의 비트 길이 벡터는  $(n(1), n(2), \dots, n(L_{max}))$ 로 구성되는데, 여기서  $n(i)$ 는 비트 길이  $i$ 를 갖는 부호어 수를 나타내고,  $L_{max}$ 는 이 부호 안에서 최대 부호어 길이를 나타낸다. 그림 1과 같이 전체 이진 나무의 레벨  $L$ 에 존재하는 대칭적 부호어의 수  $mF(L)$ 은 식 (1)과 같이 표현된다.

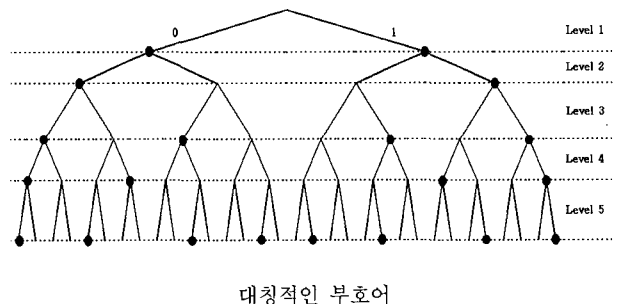


그림 1. 전체 이진 나무에서 대칭적 부호어의 분포  
Fig. 1. Distribution of symmetrical codewords on a full binary tree

$$m_F(L) = 2^{\lfloor (L+1)/2 \rfloor} \quad (1)$$

여기서  $\lfloor x \rfloor$ 은  $x$ 와 같거나 작은 수 중 가장 큰 정수를 나타낸다.

그런데, 모든 대칭적인 부호어들이 RVLC로 사용될 수 있는 것은 아니다. 상위 레벨에서 대칭적인 부호어가 선택되면 그에 따라서 하위 레벨의 몇몇 대칭적 부호어들은 접두어(prefix) 조건에 위배된다. 선택된 대칭적 부호어들을 접두어 조건에 만족시키기 위하여 하위 레벨의, 접두어 조건을 위배하는 대칭적 부호어들을 제거해야 한다. 상위  $i$ 레벨에서 대칭적 부호어가 선택되었을 때 접두어 조건을 위배하는 하위  $L$ 레벨의 대칭적 부호어의 수를  $u(i, L)$ 이라고 하면  $u(i, L)$ 은 다음과 같이 세 가지 경우로 계산된다.

- i)  $i = L/2$ 일 때,  $u(i, L) = m_F(L - 2i)$ ,
- ii)  $L/2 < i < L$ 이고,  $(2i - L)$  비트의 접미어(suffix)가 대칭적일 때  $u(i, L) = 1$ ,
- iii) 그 밖에 나머지의 경우에는  $u(i, L) = 0$ .

따라서 레벨  $L$ 에 존재하는 적합한 대칭적 부호어의 수  $m(L)$ 은 다음과 같다.

$$m(L) = m_F(L) - \sum_{i=1}^{\lfloor L/2 \rfloor} u(i, L) \cdot n_{Huff}(i) - \sum_{i=\lfloor L/2 \rfloor+1}^{L-1} x(i, L) \quad (2)$$

여기서  $n_{Huff}(i)$ 는 Huffman 부호의 비트 길이 벡터로 구성되고,  $x(i, L)$ 는 레벨  $i$ 에서  $(2i - L)$  비트의 접미어가 대칭적인 부호어의 수를 가리킨다. 따라서 i)과 ii)의 경우로부터 식 (2)의 두 번째 항과 세 번째 항은 접두어 조건을 위배하는 대칭적 부호어의 전체 수를 표현한다.

부호화 되어야 할 전체 심볼(symbol) 개수를  $S$ 로 놓으면, Takishima의 방법은 먼저  $S$ 에 대한 가능한 대칭적 부호어의 수를 미리 계산한 후에 다음과 같이 대칭적 RVLC를 생성한다.

- 1) 대칭적 RVLC의 비트 길이 벡터를 초기화 한다.

$$nrev(i) = nHuff(i) \quad (\forall i=1) \quad (3)$$

- 2)  $nrev(i) = m(i)$ 이면  $nrev(i)$  를 그대로 유지하고 그렇지 않으면 다음과 같이 한 비트가 추가된다.

$$nrev(i+1) = nrev(i+1) + nrev(i) - m(i), \quad (4)$$

$$nrev(i) = m(i) \quad (5)$$

- 3)  $S$ 에 대한 최종 대칭적인 RVLC가 할당될 때까지 과정 2)를 반복한다.

## 2. Tsai 알고리즘

Tsai의 알고리즘이 Takishima의 방법을 기초로 하고 있지만, 설계 순서의 과정 2)에서 유효한 대칭적인 부호어를 좀더 신중히 선택하여 보다 효율적인 대칭적인 RVLC를 제공한다.

$d(L)$ 을 레벨  $L$ 에서 접두어 조건의 위배로 인해 제거되는 대칭적인 부호어의 수라고 하면,  $d(L)$ 은 다음과 같이 표현된다.

$$d(L) = \sum_{i=1}^{\lfloor L/2 \rfloor} u(i, L) \cdot n_{Huff}(i) + \sum_{i=\lfloor L/2 \rfloor+1}^{L-1} x(i, L) \quad (6)$$

여기서 첫 번째 항은 모든 대칭적인 부호어에서 각 부호어의 접미어에 존재하는 대칭적인 비트의 최대 길이를 정리한 테이블을 사용하여 감소될 수 있다. 이 대칭적인 모양의 접미어가 길면 길수록 하위 레벨에 접두어 조건을 위배하는 대칭적인 부호어가 더 많아지게 된다. 예를 들어, '01110'보다 '00100'이 선택되면 7비트 길이의 대칭적인 부호어를 하나 더 제거해야 한다. 따라서, 보다 짧은 비트 길이의 대칭적인 접미어를 갖는 부호어가 선택된다면, 하위의 많은 대칭적인 부호어들이 사용될 수 있다. 이는 Takishima의 방법에 존재했던 변동(variation) 문제를 해결하고, 평균부호길이를 줄이는 데에 기여한다. 여기서 변동 문제는 최종 대칭적 RVLC와 그 평균부호길이가 처음에 선택된 대칭적인 부호어에 의하여 달라질 수 있음을 말하며, 따라서 기존의 방식에서는 선택되는 부호어에 따라서 다수의 RVLC가 존재하게 된다.

## 3. 문제점 및 제한사항

기존의 알고리즘이 가지고 있는 문제점들 중 하나는 우선 복잡한 선택 및 생성 과정이다. 전체 생성과정은 초기화 단계와 생성 단계의 두 단계로 나뉜다. 초기화 단계는 식 (2) 혹은 식 (6)과 과정 1)에서 이루어지는데,  $S$ 에 대해 전체 Huffman 이진 나무 최상단의 절점(root)에서  $L_{max}$ 레벨까지 각 레벨의 유효한 대칭적인 부호어의 수가 계산된다. 생성 단계는 식 (4)와 식 (5)와 과정 2)와 3)에서 이루어지는데 초기화 단계에서 미리 계산된 수를 기초로 전체 레벨에 걸쳐 주어진 Huffman 부호에 대한 적응 과정이 수행된다.

두 번째 문제점은 식 (2) 혹은 식 (6)의 두 번째 항과

세 번째 항의 구분이 확실하게 규정되지 않아 두 항이 겹칠 수 있다는 것이다. 두 항이 분명하게 구분되기 위해서 초기화 단계부터 접두어 조건이 절대적으로 적용되어야 한다. 더욱이, Tsai의 알고리즘이 평균부호길이를 줄이고 변동 문제를 완화시킬 수 있다고 하더라도, 각 레벨에서 사용 가능한 대칭적 RVLC의 수를 미리 결정하는 방식은 변동 문제를 명백하게 해결할 수 없고, 주어진 확률 분포에 대한 가장 적합한 해결책을 주지 못한다.

마지막으로, 적응과정이 모든 레벨에 걸쳐 진행되어 몇몇 레벨의 대칭적 RVLC의 수가 특정한 조건 아래에서 제한을 받아 사용 가능한 대칭적인 부호어들을 놓치게 된다. 과정 2)에서 주로 Huffman 부호의 비트 길이 벡터는 미리 계산된  $m(i)$  보다 크므로 비트 추가 연산이 수행된다.  $m(i)$  보다 작게 되면 대칭적 RVLC의 비트 길이 벡터는 다른 유효한 부호어들이 있음에도 불구하고 주어진 Huffman 부호의 비트 길이 벡터를 따라야 한다.

과정 2)에서  $\sum_7 n_{Huff}^{(i)} = \sum_7 m^{(i)}$ ,  $n_{rev}(i+1) = m(i+1)$ 이면,  $n_{rev}(i+1)$ 는 그대로 유지되고, 위의 제한사항이 발생한다. 만약 하위 레벨로 진행해 나가며 이 조건을 만나지 않게 되면 제한 사항은 발생하지 않는다. 하지만, 최상위 레벨에서는  $\sum_7 n_{Huff}^{(i)} = \sum_7 m^{(i)} = 0$  이므로 모든 생성과정의 시작점은 이 제한 사항을 받게 되고, 가장 발생 확률이 높은 심볼에 충분히 짧은 대칭적 부호어를 할당하는 것이 곤란할 수 있다. 결국, 이 문제점은 최종 대칭적 RVLC의 효율성을 저하시키는 데에 크게 영향을 미치게 된다.

### III. 제안된 RVLC 알고리즘

앞 장에 소개한 것처럼, 기존 알고리즘의 문제점으로는 복잡성, 변동성, 제한사항, 주어진 Huffman 부호에 대한 부적합성을 들 수 있다. 이러한 문제점들과 제한사항을 해결하기 위해서 기존의 초기화와 생성과정의 수정이 아니라 문제에 대한 새로운 접근이 요구된다.

#### 1. 대칭적 부호어의 검색

그림 2는 Huffman 부호와 이진 나무의 한 예를 보여준다. 최상단의 절점에서 각각 오른쪽과 왼쪽으로 향하

는 가지(branch)에 '0'과 '1'을 할당하고, 이진 나무의 단점(leaf)만이 부호어로 사용되어 접두어 조건을 만족한다. 하지만 오른쪽과 왼쪽 가지에 '0'과 '1'을 할당하는 방식은 정해져 있지 않다. 따라서, 그림 3과 같이 동일한 평균부호길이를 가지며 접두어 조건을 만족하는 또 하나의 Huffman 부호를 얻을 수 있다. 이처럼 '0'과 '1'의 할당 방향을 바꾸는 것은 기존의 Huffman 부호의 비트를 반전시키는 것과 같다.

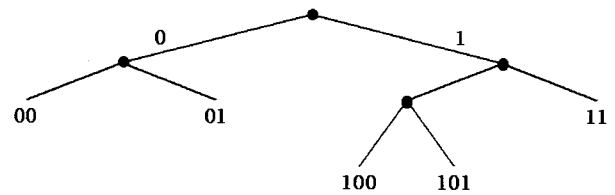


그림 2. Huffman 부호와 이진 나무

Fig. 2. Binary Huffman tree

RVLC는 순방향과 역방향의 복호를 지원하기 위해 접두어 조건뿐만 아니라 접미어 조건도 역시 만족시키는 부호어이다. 하지만 대칭적 RVLC는 부호어의 구성이 대칭적인 특수한 성질을 가지므로 접두어 조건을 만족하게 되면 접미어 조건도 역시 만족하게 된다. 이 '대칭성'은 또 하나의 장점을 가진다. 선택된 대칭적 RVLC를 비트 반전시키면 같은 부호 길이를 가지며 접두어 조건과 접미어 조건을 만족시키는 또 하나의 대칭적 RVLC를 얻게 된다. 예를 들면, 그림 2와 그림 3에서 '11'과 '101'은 '00'과 '010'을 비트 반전시켜 쉽게 구할 수 있다.

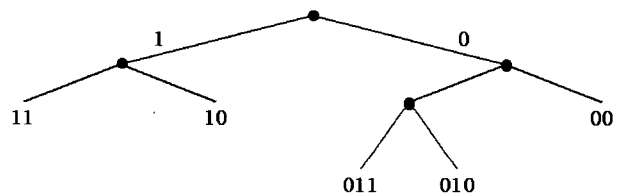


그림 3. 할당 방향이 바뀐 이진 나무의 부호

Fig. 3. Another Huffman tree

이러한 특성을 이용하여 제안된 알고리즘은 전체 이진 나무에서 대칭적인 부호어를 검색하지 않고, 특별히 '0'으로 할당된 한쪽 영역에서 대칭적인 부호어를 검색한다. 이 영역에서  $S$ 만큼이 아니라 절반에 해당하는  $\lfloor S/2 \rfloor$ 만큼 대칭적인 부호어를 선택한다. 여기서  $\lfloor \cdot \rfloor$ 는  $x$ 와 같거나 큰 수 중 가장 작은 정수를 의미한다. 최종

대칭적 RVLC는 '0'으로 할당된 한쪽 영역에서 선택된 부호어들과 그것을 비트 반전시킨 부호어로 구성된다.

'0'으로 할당된 왼쪽 영역의 레벨  $l$ 에서의 대칭적인 부호어의 수  $m_H(l)$ 은 식 (1)을 대신하여 다음과 같이 계산된다.

$$m_H(l) = 2^{\lfloor (l+1)/2 \rfloor - 1} \quad (7)$$

$C(l)$ 을 '0'으로 할당된 영역의 레벨  $l$ 에 존재하는 모든 대칭적인 부호어를 가지는 집합이라고 하면, 다음과 같이 대칭적인 부호어들은  $C(l)$ 의 원소로 표현된다.

$$C(1) = \{ '0' \} \text{ 또는 } \{ 0 \}, C(2) = \{ '00' \} \text{ 또는 } \{ 0 \} \quad (8)$$

$$C(l) = \{ l\text{-bit codeword} \mid 2 \times \forall x \in C(l-2) \text{ 또는}$$

$$\{ 2^{l-1} - 2 \} - 2 \times \forall x \in C(l-2) \} \quad (l=3) \quad (9)$$

따라서 식 (7)은 다음과 같이  $C(l)$ 의 모든 원소의 수와 같다.

$$N(C(l)) = m_H(l) \quad (10)$$

그림 4는 식 (7)~식 (10)으로부터 얻어진 '0'으로 할당된 영역에서 대칭적인 부호어들의 분포와 각 레벨에서 대칭적인 부호어에 해당하는 정수 값을 보여준다.

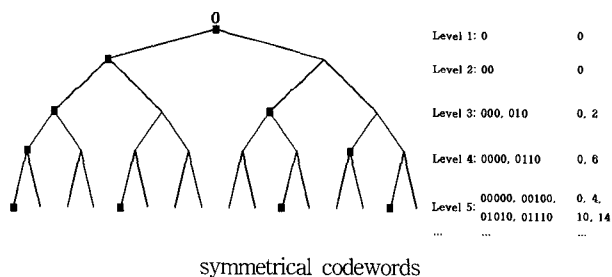


그림 4. 한쪽 영역에서의 대칭적인 부호어  
Fig. 4. Distribution of symmetrical codewords on a half binary tree

유일하게 복호가 가능한 부호를 얻기 위해서 먼저 '0'으로 할당된 한쪽 영역에서 선택된 대칭적인 부호어들은 제거과정을 통해 접두어 조건을 만족시켜야 한다. 이 부호어들은 그림 5 (a)와 같이 한쪽 영역 내에서는 유일하게 복호가 가능하다. 반대로, 비트 반전을 통해 얻은 대칭적인 부호어들은 그림 5 (b)와 같이 '1'로 할당된 영역 내에서는 역시 유일하게 복호가 가능하다.

최종 단계에서 먼저 선택되었던  $\lfloor S/2 \rfloor$  만큼의 부호어

들과 나머지 비트 반전된  $\lfloor S/2 \rfloor$ 만큼의 부호어들이 대칭적 RVLC를 구성하게 되면 각각의 접두어들은 '0'과 '1'로 다르기 때문에 그림 5 (c)처럼 유일하게 복호가 가능한 대칭적 RVLC를 얻는다.

제안된 알고리즘을 통한 대칭적 RVLC의 복호 가능성은 Kraft 부등식으로도 증명된다. Huffman 이진 나무에 대한 Kraft 부등식은 다음과 같이 주어진다.

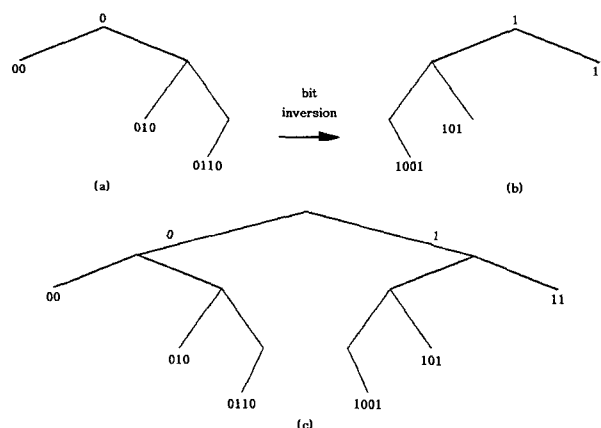


그림 5. 대칭적 RVLC의 복호 가능성  
Fig. 5. Uniquely decodable symmetrical RVLC

$$\sum_{i=1}^S 2^{-l_i} \leq 1, \quad (l_1 \leq l_2 \leq \dots \leq l_S) \quad (11)$$

여기서  $l_i$ 는 부호화 되는 비트 길이이다. 그런데, 제안된 알고리즘에서는 '0'으로 할당된 한쪽 영역에서 검색이 이루어지고, 이 영역은 Huffman 이진 나무의 첫 번째 서브 트리(sub-tree)와 같다. 따라서 Kraft 부등식은 식 (12)와 같이 고쳐진다.

$$\sum_{i=1}^{\lfloor S/2 \rfloor} 2^{-(l_i-1)} \leq 1 \quad \text{또는} \quad \sum_{i=1}^{\lfloor S/2 \rfloor} 2^{-l_i} \leq \frac{1}{2}, \quad (l_1 \leq l_2 \leq \dots \leq l_{\dots}) \quad (12)$$

식 (12)의 개정된 Kraft 부등식은 한쪽 영역에서 선택된 대칭적인 부호어들의 유일 복호 가능성에 대한 필요 충분조건을 보여준다. 제거 과정을 통해 한쪽 영역에서 선택된 부호어들이 접두어 조건을 만족하면, 그 부호어들 역시 식 (12)의 부등식을 만족한다. 비트 반전 연산을 통해 얻어진 부호어들의 비트 길이는 당연히 비트 반전 이전의 부호어들의 길이와 동일하기 때문에, 개정된 Kraft의 부등식은 식 (13)과 식 (14)로 확장된다.

$$\sum_{i=1}^{\lfloor S/2 \rfloor} (2^{-l_i} + 2^{-l_i}) = \sum_{i=1}^{\lfloor S/2 \rfloor} (2 \times 2^{-l_i}) \quad (l_1 \leq l_2 \leq \dots \leq l_{\dots}) \quad (13)$$

$$2 \times \sum_{i=1}^{\lfloor S/2 \rfloor} 2^{-l_i} \leq 2 \times \frac{1}{2}, \quad (l_1 \leq l_2 \leq \dots \leq \lfloor S/2 \rfloor) \quad (14)$$

부호화된 비트 길이인  $l_i$ 는 같거나 다를 수 있으므로,  $l_i$ 의 인덱스  $i$ 는  $S$ 로 확장된다. 결국 식 (14)로부터 다음과 같이 완전한 Kraft 부등식을 얻게 된다.

$$\therefore \sum_{i=1}^S 2^{-l_i} \leq 1, \quad (l_1 \leq l_2 \leq \dots \leq l_S) \quad (15)$$

따라서 그림 5 혹은 식 (12)~(15)를 통하여, 제안된 비트 반전을 이용한 부호 검색 방법이 유일하게 복호 가능한 대칭적 RVLC를 제공할 수 있다.

## 2. $Z_L$ 적응법

생성된 대칭적 RVLC의 효율성 증가시키려면 상위 레벨에서 선택 가능한 부호어를 놓치는 일이 없어야 하고, 심볼의 발생 확률에 적합하도록 하기 위해 주어진 Huffman 부호에 적응시켜야 한다.

대칭적인 부호어들이 '0'으로 할당된 영역에서 검색되므로 이 부호어들의 접두어 혹은 접미어에는 최소한 하나의 '0' 비트가 존재한다. 그림 4에 보인 것처럼, 각 레벨에는 모두 '0' 비트로만 구성된 대칭적인 부호어들이 존재한다. 이러한 부호어를  $Z_L$ 이라고 정의하면,  $Z_L$ 의 선택에 대하여 다음 두 가지 사항을 고려해야 한다.

- 1) 한쪽영역에서  $Z_L$ 이 선택되면 제거과정으로 인해  $Z_L$ 을 제외한 나머지 대칭적 부호어의 접두어와 접미어에 존재하는 '0' 비트의 수는  $L$ 보다 작다. 예를 들면,  $Z_3$ 인 '000'이 선택되면 나머지 부호어들의 접두어와 접미어는 '0' 혹은 '00'인 부호어들이 된다. 달리 말하면,  $Z_L$ 의 선택을 통해 각 레벨에 존재하는 RVLC의 수를 조절할 수 있다.
- 2) 최종 단계에서 비트반전 연산이 취해진다. 따라서  $Z_L$ 을 통해 마지막 단계에서  $L$ 레벨에 모두 '1'비트로만 구성된 부호어가 추가된다. 만약 적당한 상위 레벨에  $Z_L$ 이 신중하게 선택된다면 전체 대칭적 RVLC의 평균부호길이를 효과적으로 줄일 수 있다.

최적화된 Huffman 부호에서 발생확률이 가장 높은 심볼에 가장 짧은 비트가 할당되고, 이 심볼의 비트 길이인  $L_{min}$ 은 심볼의 수와 확률 분포에 영향을 받는다. 또한,  $L_{min}$ 은 주어진 발생 확률에서 최적의 부호를 설계하기 위해 필수적이다.

본 논문에서는 최적화된 Huffman 부호로부터 대칭적 RVLC를 설계하기 위해  $Z_L$  적응법을 제안한다.  $Z_L$  적응법은  $Z_L$ 의 레벨, 즉 비트 길이를 주어진 Huffman 부호의  $L_{min}$ 과 같도록 일치시키는 것이다.

따라서 주어진 Huffman 부호가 2 비트로 시작하면,  $Z_2$  즉 '00'이 선택된다.  $Z_L$  적응법은  $Z_L$ 과  $L$  개의 '1' 비트 부호를 가장 빈번한 심볼로부터 차례대로 할당하여 좀더 효율적인 대칭적 RVLC를 생성한다.  $Z_L$ 의 선택에 따라 하위 레벨에서 제거되는 대칭적인 부호어의 수는 달라지겠지만, 상위 레벨에 두 개의 부호어를 추가시키는 것이 평균부호길이를 줄이는데 좀더 큰 기여를 하게 된다.

제안된 대칭적 RVLC의 생성 방법은 다음과 같다.

- 1) Huffman 이진 나무의 한쪽 영역에서  $Z_L$ 을 결정한다.  $Z_L$ 의 비트 길이는 주어진 Huffman 부호에서 가장 짧은 부호어의 비트 길이,  $L_{min}$ 과 같다.
- 2)  $\lfloor S/2 \rfloor$ 만큼 대칭적인 부호어를 찾을 때까지 최상위 레벨부터 사용 가능한 대칭적인 부호어를 모두 검색한다. 모든 선택과정은 접두어 조건을 위배하는 부호어들의 제거 과정을 수반한다.
- 3) 한쪽 영역에서 먼저 선택된 대칭적인 부호어와 이를 비트 반전시킨 부호어로 최종적인 대칭적 RVLC를 얻는다.

따라서 한쪽 영역의 레벨  $l$ 에서 유효한 대칭적인 부호어의 수,  $m(l)$ 는 다음과 같이 표현된다.

$$m(l) = m_H(l) - \sum v_z(l) - \sum_{i=1}^{L-1} v_{ci}(l) \quad (15)$$

그리고  $F(l)$ 을 전체 이진 나무에서 모든 심볼 수  $S$ 에 대한 레벨  $l$ 까지의 누적 부호어 수라고 하면  $F(l)$ 은 다음과 같이 계산된다.

$$F(l) = \sum_l (2 \times m(l)) \quad (16)$$

여기서  $v_z(l)$ 은  $Z_L$ 의 선택에 따라 하위 레벨  $l$ 에서 접두어 조건을 위배하여 제거되어야 하는 부호어의 수이고,  $v_{ci}(l)$ 은 상위 레벨부터 선택된 나머지 대칭적 부호어의 선택에 따라 제거해야 하는 부호어의 수를 나타낸다.  $L$ 은  $Z_L$ 의 비트 길이를 의미하고,  $i$ 는 먼저 선택된 부호어의 접두어 혹은 접미어에 존재하는 '0' 비트의 수를 나타낸다.

식 (15)에서  $v_z(l)$ 는 다음과 같이 주어진다.

(i)  $l < 2L+1$ 일 때,  $v_z(l) = 1$ , (17)

(ii)  $2L+1 \leq l$ 일 때,  $v_z(l) = m_H(l-2L)$ . (18)

그리고,  $v_{\alpha}(l)$ 는 다음과 같이 계산된다.

(i)  $2L'+1-i < l < 2L'+1$ 일 때,  $v_{\alpha}(l) = 1$ , (19)

(ii)  $2L'+1 \leq l$ 일 때,  $v_{\alpha}(l) = m_H(l-2L')$  (20)

여기서  $L'$ 은 상위 레벨에서 선택된 부호어의 비트 길이를 말한다.

표 1은 식 (15)~식 (20)을 통해 얻은  $m(l)$ 과  $F(l)$ 을 보여준다. 표 1에서  $Z_0$ 열은  $Z_L$ 적용법을 적용시키지 않은 것으로,  $Z_L$  적용법과의 성능 비교를 위한 것이다.

표 1.  $Z_L$ 의 선택에 따른 각 레벨에서 RVLC의 수

Table 1. The number of RVLC from the choice of  $Z_L$

$Z_L$	$Z_0$		$Z_1$		$Z_2$		$Z_3$		$Z_4$		$Z_5$		$Z_6$		
	$m(l)$	$F(l)$	$m(l)$	$F(l)$	$m(l)$	$F(l)$	$m(l)$	$F(l)$	$m(l)$	$F(l)$	$m(l)$	$F(l)$	$m(l)$	$F(l)$	
L	1	0	0	$Z_1$	1	0	0	0	0	0	0	0	0	0	
	2	0	0	1	2	$Z_2$	2	0	0	0	0	0	0	0	
	3	1	2	1	3	1	4	$Z_3+1$	4	1	2	1	2	1	2
	4	1	4	1	4	1	6	1	6	$Z_4+1$	6	1	4	1	4
	5	2	8	1	5	1	8	2	10	2	10	$Z_5+2$	10	2	8
	6	2	12	1	6	1	10	2	14	2	14	2	14	$Z_6+2$	14
E	7	4	20	1	7	1	12	3	20	4	22	4	22	4	22
	8	3	26	1	8	1	14	2	24	3	28	3	28	3	28
V	9	7	40	1	9	1	16	4	32	6	40	7	42	7	42
	10	6	52	1	10	1	18	3	38	5	50	6	54	6	54
L	11	12	76	1	11	1	20	6	50	9	68	11	76	12	78
	12	10	96	1	12	1	22	4	58	7	82	9	94	10	98
	13	22	140	1	13	1	24	9	76	15	112	19	132	21	140
	14	20	180	1	14	1	26	8	92	14	140	17	166	19	196
	15	41	262	1	15	1	28	16	124	27	194	34	234	38	254
	16	37	336	1	16	1	30	13	150	24	222	30	294	34	322

표 2. 생성 과정 비교

Table 2. Comparison of the construction process

비 고	기존의 생성 방법	제안된 생성 방법
검색되는 심볼의 수	S	$\lceil S/2 \rceil$
검색 영역	전체 이진 나무	'0'으로 할당된 한쪽 영역
생성 과정	식 (2)와 과정 1)에서의 초기화 (i) $m(i)$ 를 미리 계산 (ii) $n_{rev}(i) = n_{Huff}(i)$	$Z_L$ 을 선택 ( $L = L_{min}$ )
	과정 2), 3)을 통해 생성: (i) Huffman 부호에 적용 (ii) 대칭적인 부호어 검색 및 그에 따른 제거 과정 수행 (S만큼)	생성: (i) 대칭적인 부호어 검색 및 그에 따른 제거 과정 수행 ( $\lceil S/2 \rceil$ 만큼) (ii) 비트 반전

#### IV. 실험 결과 및 분석

본 논문에서 제안된 알고리즘을 평가하기 위해 기존의 알고리즘에 대하여 생성 과정의 복잡도 측면과 부호의 효율성 측면을 비교한다. 또한, 여러 확률 분포를 이용한 실험을 통해 제안된  $Z_L$  적응법의 적합성을 확인한다.

##### 1. RVLC 생성 과정의 복잡도

표 2는 기존의 알고리즘과 제안된 알고리즘의 생성과정을 비교한 결과를 보여준다. 표 2와 더불어 최악의 경우를 가정하여 알고리즘의 성능을 보여주는 Big O 표기법을 통해 기존의 알고리즘과 제안된 알고리즘을 비교한다. 기존의 생성방법은 초기화 단계에서  $O(\frac{S^2}{2} + S)$ , 생성 단계에서  $O(\frac{S^2}{2} \cdot (\sqrt{2})^S)$  이 되며, 이에 비해 제안된 생성방법은 초기화 단계에서  $O(1)$  이고, 생성 단계에서  $O(\frac{S^2}{4} \cdot (\sqrt{2})^S + S)$  가 된다. 이를 정리하면 기존의 알고리즘의 복잡도 함수  $T_{conv}(N)$ 은 식 (21)과 같다.

$$T_{conv}(N) = O\left(\frac{S^2}{2} \cdot (\sqrt{2})^S + \frac{S^2}{2} + S\right) = O(N^2 \cdot (\sqrt{2})^N) \Big|_{N=S} \quad (21)$$

또한 제안된 알고리즘의 복잡도 함수  $T_{new}(N)$ 은 식 (22)와 같다.

$$T_{new}(N) = O\left(\frac{S^2}{4} \cdot (\sqrt{2})^S + S + 1\right) = O(N^2 \cdot (\sqrt{2})^N) \Big|_{N=S} \quad (22)$$

그리고, 기존의 알고리즘과 제안된 알고리즘 사이의 복잡도의 차이를 식 (23)과 같이 복잡도 함수의 비로써 표현할 수 있다.

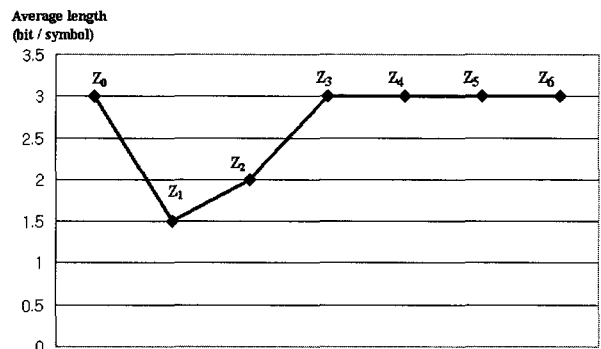
$$\frac{T_{conv}(N)}{T_{new}(N)} = O((\sqrt{2})^N) \Big|_{N=S} \quad (23)$$

따라서, 최악의 경우에 부호화되는 심볼의 수  $S$ 에 따라 약  $(1.19)^S$  정도로 기존의 알고리즘과 제안된 알고리즘이 복잡도 차이를 보이고 있으며, 제안된 알고리즘은 부호화 하는 심볼의 수와 검색 영역을 각각 50%씩 줄이고,  $Z_L$  적응법으로 주어진 확률에 대한 적응과정을 간단히 하며 많은 계산을 필요로 하지 않아서 Takishima<sup>[8]</sup>

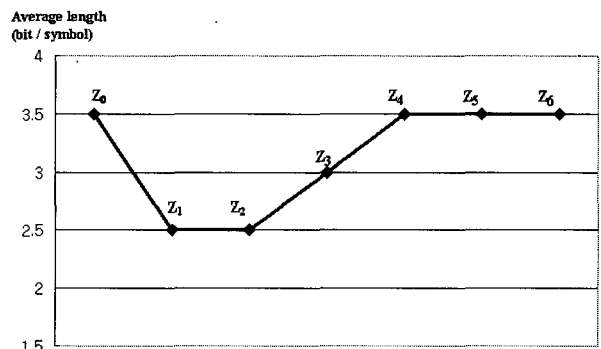
와 Tsai<sup>[9]</sup>의 알고리즘보다 훨씬 간단한 생성 과정을 가진다는 점을 알 수 있다.

##### 2. $Z_L$ 적응법의 적합성

$Z_L$  적응법의 적합성을 증명하기 위해 임의의 확률 분포와 그에 따른 Huffman 부호의  $L_{min}$ 을 변화시키고, 각 확률 분포에 대해  $Z_L$ 의 비트 길이  $L$ 을 변화시킨 결과를 평가한다. 하지만, 특정  $L_{min}$ 값을 갖는 Huffman 부호를 설계하는 일은 매우 힘든 일이다. 따라서 우선 모든 심볼이 동일한 발생 확률을 갖는 균일 분포에 대해  $Z_L$  적응법을 적용시켰다. 이러한 균일 분포에 대한 Huffman 부호는 의미가 없을 수 있지만, 특정  $L_{min}$  값을 갖는 Huffman 부호를 설계하기가 매우 쉽다는 장점이 있다. 그림 6은 5개의 균일 확률 분포를 가지고 각 확률 분포에 대해  $Z_L$ 의 비트 길이  $L$ 을 변화시키며 5개의 대칭적 RVLC를 생성하여 평균부호길이를 비교한 결과를 제시한다. 각 그림 아래의 설명은 심볼의 수, 각 심볼의 발생 확률, 그리고 Huffman 부호의 비트 길이를 순서대로 보여준다. 균일 분포이므로 Huffman 부호의  $L_{min}$ 과  $L_{max}$ 는 동일하다. 그림 6(a)~그림 6(b)에서 화살표는 제안된 방법에 따라 정확하게 적용된  $Z_L$  적응법의 위치를 가리키고, 각  $Z_0$ 는 어떠한  $Z_L$ 도 RVLC로 선택되지 않았음을 의미한다.

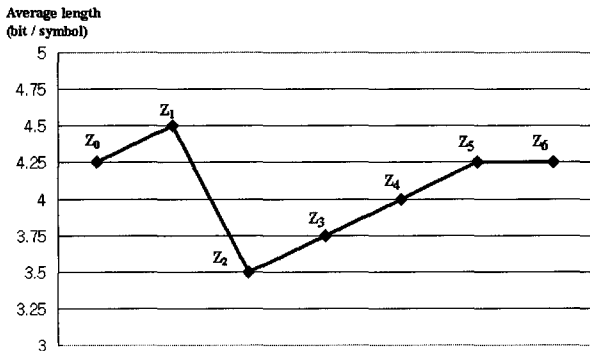


(a) (2, 0.5, 1bit)

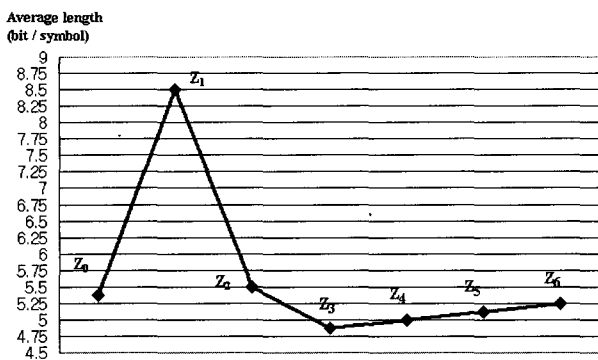


(b) (4, 0.25, 2bit)

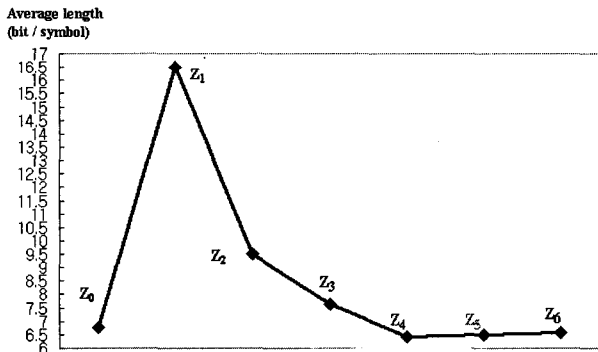




(c) (8, 0.125, 3bi)



(d) (16, 0.0625, 4bit)



(e) (32, 0.03125, 5bit)

그림 6. 균일 분포에 대한 대칭적 RVLC의 평균부호길이 (II)  
Fig. 6. Average length of symmetrical RVLC for the uniform probability distribution (II)

그림 6의 결과를 통해 적응법을 사용하지 않은 경우 ( $Z_0$ )보다 제안된  $Z_L$  적응법이 사용된 경우에 더 효율적인 대칭적 RVLC를 생성한다는 점을 알 수 있다. 그런데, 균일 분포의 경우에 특정  $L_{min}$  이상에서  $Z_L$ 보다  $Z_{L-1}$ 의 결과가 평균부호길이를 통해 더 나은 성능을 보여준다. 그러나, 앞서 말한 바와 같이, 균일 분포는 비실용적이고, 의미가 적은 분포이므로 제안된  $Z_L$  적응법을 평가하기 위해 또 하나의 실험이 필요하다.

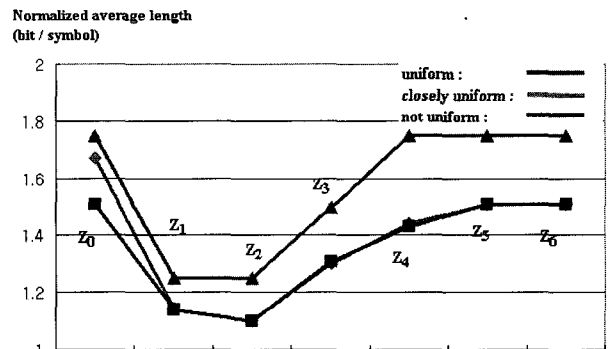
$Z_L$  적응법의  $Z_L$  적합성을 확인하기 위해서 주어진 Huffman 부호가 동일한  $L_{min}$ 을 갖는 여러 확률 분포에 대하여 대칭적 RVLC를 생성하고 평균부호길이를 비교한다. 확률분포는 균일 분포, 근사 균일 분포, 비균일 분포로 나뉘는데, 근사 균일 분포는 편의상  $n_{Huff}(L_{min})$ 이  $\lceil S/2 \rceil$ 보다 같거나 큰 확률 분포로 가정한다. 그림 7과 같이 각각의 확률 분포의 엔트로피가 틀리기 때문에 평균부호길이를 비교하는 대신에 정규화된 평균부호길이를 효율성을 비교한다.

정규화된 평균부호길이  $N(L)$ 은 다음과 같이 계산된다.

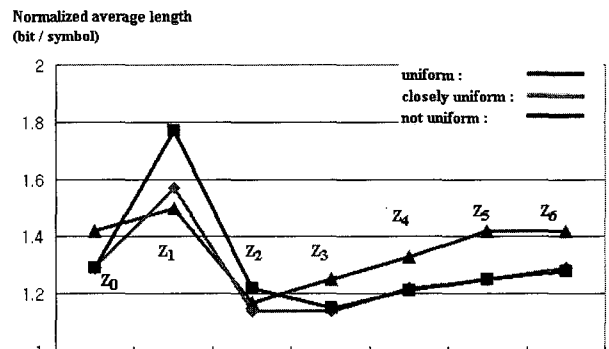
$$N(L) = \frac{\bar{L}}{H(X)} \quad (24)$$

여기서  $H(X)$ 는 확률 분포의 엔트로피이고,  $\bar{L}$ 는 대칭적 RVLC의 평균부호길이이다.

그림 7은 Huffman 부호가 2비트와 3비트에서 출발하는 여러 확률 분포에 대한 RVLC의 정규화된 평균부호길이를 보여준다. 결과를 보면 균일 분포를 제외한 나머지 경우에 제안된  $Z_L$  적응법이 더 나은 성능을 가진다.



(a)  $L_{min} = 2$



(b)  $L_{min} = 3$

그림 7. 여러 확률 분포에 대한 대칭적 RVLC의 정규화된 평균부호길이  
Fig. 7. Normalized average length of symmetrical RVLC for different probability distributions

즉, 하나의 확률 분포를 이루는 심볼들에 대하여  $Z_L$  적응법이 가장 효율적인 RVLC를 생성할 수 있어서 가장 합리적인 선택임을 확인할 수 있다.

3. 부호화 성능의 비교

앞 절에서 하나의 확률 분포에 대하여  $Z_L$  적응법이 가장 뛰어난 성능을 보여주는 것을 확인하였다. 본 절에서는 제안된 알고리즘과 기존의 알고리즘을 부호 성능을 비교한다.

표 3은 영어 알파벳에 대하여 Takishima 방법<sup>[8]</sup>과 Tsai의 방법<sup>[9]</sup>, 그리고 제안된 방법으로 대칭적 RVLC를 생성하여 평균부호길이를 비교한 결과를 보여준다. 표

3에서 C1, C2, C3은 각각 Huffman 부호, Takishima의 방법, Tsai의 방법을 통해 설계된 부호이다. Tsai는 부호를 좀더 신중히 선택하는 방식으로 평균부호길이가 1.9%정도 줄어든 RVLC를 얻었다. 그런데, C2와 C3의 결과를 보면 레벨 7에서 제한 조건을 만나 레벨 8 이하에서 사용 가능한 대칭적인 부호어들이 더 존재하는 데에도 불구하고 찾지 못하고 있다. 게다가 Huffman 부호의 시작점이며 가장 발생확률이 높은 심볼이 위치한 레벨 3에서도 제한사항이 발생하였다. 시작점에서는 제한 조건을 언제나 만나게 되어  $n_{rev}(3)$ 이  $n_{Huff}(3)$ 을 따르게 된다. 그 결과 4개의 대칭적인 부호어 중 2개만이 선택되었고 이는 부호의 효율성에 큰 영향을 미친다.

표 3. 영어 알파벳에 대한 부호화 성능 비교

Table 3. Comparison of coding performances for the English alphabet

발생 확률		Huffman 부호 C1		Takishima 방법 C2		Tsai 방법 C3		제안 방법 C4	
		L	부호어	L	부호어	L	부호어	L	부호어
E	0.14878570	3	001	3	000	3	010	3	000 ( $Z_3$ )
T	0.09354149	3	110	3	111	3	101	3	111
A	0.08833733	4	0000	4	0110	4	0110	3	010
O	0.07245796	4	0100	4	1001	4	1001	3	101
R	0.06872164	4	0101	5	00100	4	0000	4	0110
N	0.06498532	4	0110	5	11011	4	1111	4	1001
H	0.05831331	4	1000	5	01010	5	01110	5	00100
I	0.05644515	4	1001	5	10101	5	10001	5	11011
S	0.05537763	4	1010	5	01110	5	00100	5	01110
D	0.04376834	5	00010	5	10001	5	11011	5	10001
L	0.04123298	5	00011	6	001100	6	011110	6	001100
U	0.02762209	5	10110	6	110011	6	100001	6	110011
P	0.02575393	5	10111	6	010010	6	001100	6	011110
F	0.02455297	5	11100	6	101101	6	110011	6	100001
M	0.02361889	5	11110	6	011110	7	0111110	7	0010100
C	0.02081665	5	11111	6	100001	7	1000001	7	1101011
W	0.01868161	6	011100	7	0010100	7	0010100	7	0011100
G	0.01521216	6	011101	7	1101011	7	1101011	7	1100011
Y	0.01521216	6	011110	7	0011100	7	0011100	7	0111110
B	0.01267630	6	011111	7	1100011	7	1100011	7	1000001
V	0.01160928	6	111011	7	0100010	7	0001000	8	00111100
K	0.00867360	7	1110100	7	1011101	7	1110111	8	11000011
X	0.00146784	8	11101011	8	00111100	8	01111110	8	01111110
J	0.00080064	9	111010101	9	001010100	9	011111110	8	10000001
Q	0.00080064	10	1110101000	10	0010110100	10	0111111110	9	011111110
Z	0.00053376	10	1110101001	10	1101001011	10	1000000001	9	100000001
평균부호길이		4.15572392		4.69655649		4.60728507		4.46463681	

표 4. 알고리즘의 특성 비교

Table 4. Comparison of characteristics

부호의 종류	최대 부호 길이	최소 부호 길이	최상위 레벨에 존재하는 대칭적 부호의 수
Huffman 부호	$L_{max}$	$L_{min}$	$n_{Huff}(L_{min})$
기존의 대칭적 RVLC	$L_{max}$ 보다 크거나 같음	$L_{min}$ 과 같음	$n_{Huff}(L_{min})$ 보다 작거나 같음
제안된 대칭적 RVLC	$L_{max}$ 보다 작거나 같음	$L_{min}$ 보다 작거나 같음	$n_{Huff}(L_{min})$ 보다 크거나 같음

C4는 제안된 방법에 따라 설계된 대칭적 RVLC이며, C2, C3와 비교하면 각각 5.2%와 3.2% 정도 더 짧은 평균부호길이를 가진다. 제안된 알고리즘에 따라 비트 반전의 쌍으로 구성되고,  $Z_L$ 을 선택한 후에 나머지 대칭적인 부호어들은 최상위 레벨부터 차례차례 검색하여 선택되므로 기존의 알고리즘이 가지고 있던 변동 문제를 완전히 해결하였다. 또한, 주어진 Huffman 부호에 따라  $Z_3$ 와 '111'이 가장 높은 심볼에 할당되었으며, 특정 레벨에 발생하던 제한사항도 전혀 발생하지 않기 때문에 어떠한 레벨에서도 대칭적인 부호어를 놓치지 않았다. 따라서 제안된 설계 방식을 통해 더 효율적인 대칭적 RVLC를 얻을 수 있음을 알 수 있다.

표 4는 기존의 대칭적 RVLC 설계 방법과 제안된 설계 방법의 특성을 비교한 것이다. 제안된 알고리즘은 평균부호길이를 줄이는 데에 크게 영향을 미치는 최대 부호 길이, 최소 부호 길이, 그리고 최상위 레벨에 존재하는 대칭적 RVLC의 수,  $n(L_{min})$  등 세 가지 부분에서 모두 기존의 알고리즘보다 우수함을 알 수 있다.

## V. 결 론

본 논문에서는 최적의 Huffman 부호 테이블을 기초로 대칭적 RVLC를 설계하는 새로운 기법을 제안하였다. 제안된 알고리즘은  $Z_L$  적용법과 더불어 한쪽 영역에서 대칭적인 부호어를 선택한 후 비트반전 연산을 통해 최종 대칭적 RVLC를 생성한다. 비트반전 연산과  $Z_L$  적용법은 변동 문제를 해결하고, 생성 과정을 단순화하며, 평균부호길이를 줄이는 데에 크게 기여한다. 실험 결과를 통해 제안된 RVLC 기법이 기존의 대칭적 RVLC의 설계 기법보다 더 우수한 효율을 가지는

RVLC 부호를 얻을 수 있음을 보였다.

## 참 고 문 헌

- [1] M.T. Sun and A.R. Reibman, *Compressed Video over Networks*, Marcel Dekker, New York, 2001.
- [2] D. Huffman, "A method for the construction of minimum redundancy codes," *Proc. Inst. Radio Engr.*, vol. 40, pp. 1098-1101, Sept. 1952.
- [3] J.J. Rissanen and G.G. Langdon, Jr., "Arithmetic Coding," *IBM J. Res. Develop.*, 23, pp. 149-162, 1979.
- [4] ISO/IEC 14496-2, "Coding of audio-visual objects: Visual," *Final Draft Int. Std.*, Oct. 1998.
- [5] ITU-T Rec. H.263, "Video coding for low bit rate communications," Annex V, 2000.
- [6] A.S. Fraenkel and S.T. Klein, "Bidirectional Huffman coding," *Comp. J.*, vol. 33, no. 4, 1990.
- [7] J. Wen and J.D. Villasenor, "A class of reversible variable length codes for robust image and video coding," *Proc. IEEE Int. Conf. Image Processing*, vol. 2, pp. 12-49, 1997.
- [8] Y. Takishima, M. Wada and H. Murakami, "Reversible variable length codes," *IEEE Trans. Comm.*, vol. 43, pp.158-162, Feb. 1995.
- [9] C.W. Tsai and J.L. Wu, "A modified symmetrical reversible variable length code and its theoretical bounds," *IEEE Trans. Inform. Theory*, vol. 47, pp. 2543-2548, Sept. 2001.

## 저 자 소 개



정욱현(학생회원)

2002. 2 : 한국항공대학교 항공통신정보공학과 졸업(학사), 2004. 2 : 광주과학기술원 정보통신공학과 졸업(석사)  
소속기관 : 광주과학기술원 정보통신공학과 <주관심분야: 영상신호 처리

및 압축, 멀티미디어 통신, 정보 및 부호화 이론>



호요성(정회원)

1981. 2 : 서울대학교 전자공학과 졸업(학사), 1983. 2: 서울대학교 전자공학과 졸업(석사), 1983. 3~ 1995. 9 : 한국전자통신연구소 선임연구원, 1989. 12 : University of California,

Santa Barbara, Dept. of Electrical and Computer Eng. 졸업(박사), 1990. 1~1993. 5: 미국 Philips 연구소 Senior Research Member, 1995. 9~현재 : 광주과학기술원 정보통신공학과 교수, 소속기관 : 광주과학기술원 정보통신공학과 <주관심분야: 디지털 신호처리, 영상신호 처리 및 압축, 초저속 영상통신, 디지털 TV와 고선명 TV 방식, 멀티미디어 통신, 디지털 워터마킹, 3차원 실감방송 기술>