자유거리 특성을 고려한 강인한 양방향 가변길이 부호의 설계 방법

정욱현, 윤영석, 호요성

광주과학기술원 정보통신공학과

Design of Robust Reversible Variable-Length Codes Considering the Property of the Free Distance

Wook-Hyun Jeong, Young-Suk Yoon and Yo-Sung Ho
Kwangju Institute of Science and Technology (K-JIST)
E-mail: {whjeong, ysyoon, hoyo}@kjist.ac.kr

요약

가변길이 부호는 정보원(source)의 통계적인 특성을 이용하여 부호화 효율을 높이지만, 잡음이 심한 전송환경에선 비트에 대한 오류가 발생하면 심각하게 손상되는 단점을 가진다. 이러한 가변길이 부호의 문제를 해결하기 위해 양방향 가변길이 부호가 제안되었다. 양방향 가변길이 부호는 순방향과 역방향으로 복호가 가능하여 전송 오류로 인해 손상된 비디오 비트열에서 데이터를 복구할 수 있다. 부호의 자유거리(free distance)는 부호 자체가 전송 오류에 얼마나 강인한지를 보여주는 척도로 사용된다. 양방향 가변길이 부호의 자유거리(free distance)의 최소값은 1이므로 비디오 비트열 손상 자체를 방지할 수는 없다. 본 논문에서는 자유거리(free distance)의 최소값을 증가시켜 전송오류에 더욱 강인한 새로운 양방향 가변길이 부호의 설계 기법을 제안한다. 제안된 알고리즘은 주어진 Huffman 부호와 평균부호길이 함수의 특성을 이용하여 자유거리의 조건을 만족하면서 동시에 부호화 효율을 높여 기존의 알고리즘보다 향상된 성능을 보여준다.

1. 서론

동영상 데이터는 부호기에서 예측 부호화 기법과 Huffman 부호[1] 혹은 산술 부호[2]를 사용하여 압축된 후 비트열의 형태로 전송된다. 하지만 채널을 거치는 동안 전송오류는 비트열을 손상시킨다. 손상된 가변길이 부호들은 복호기에서 동기를 손실시키며, 나아가 많은 동영상 프레임(frame)의 손실을 일으킬 수 있다.

보다 강화된 오류 내성 부호화 기법에 대한 필요로 인해 H.263+[3]와 MPEG-4[4]에서는 전송된 비트스트림에 대해서 손상되지 않은 비디오 데이터를 많이 복원할 수 있는 양방향 가변길이 부호(reversible variable-length codes, RVLCs)를 채택하였다. RVLC는 어두(prefix) 조건뿐 아니라 어미(suffix) 조건을 만족하여 순방향과 역방향 모두 즉각적인(Instantaneous) 복호가 가능한 부호어이다. RVLC는 부호어 구성의 모양에 따라 대칭적(symmetrical) RVLC와 비대칭적(asymmetrical) RVLC로 나뉘어 지며 각각 H.263+와 MPEG-4에서 사용되고 있다.

효율적인 RVLC를 설계하기 위해 Takishima[5]는 Huffman 부호를 기초로 부호어 할당을 하는 알고리즘을 제안하였다. Tsai[6]는 이진 부호어의 비트 패턴(patten)을 이용하여 이 알고리즘을 개선하였다. 하지만 이 알고리즘들은 Huffman 부호의 구성을 이용할 때에 제한사항이 발생하여 사용 가능한 RVLC를 놓치게 되어 효율을 크게 증가시키지 못한다.

Tseng[7]은 Huffman 부호를 기초로 하지 않고 하나의 부호어(parent) 대신 이 부호어를 어두 혹은 어미로 갖는 하위 레벨의 부호어(child)를 선택하였을 때 효율성이 증가하면 하위 레벨의 부호어를 선택해가는 방식으로 대칭적 RVLC를 설계하였다. Lin[8]은 이 알고리즘을 비대칭적 RVLC에 적용하였다. 이 알고리즘들은 기존의 Huffman 부호 기반의 방식들보다 훨씬 개선된 효율을 보여주지만, 부호 설계에 있어서 필수적인 시작 비트 길이와 각 레벨에서 부호 선택 기법이 불명확하다는 단점이 있다.

전송 오류로 인해 손상된 가변길이 부호에 대하여 기존의 look-up 테이블 기반의 Hard 복호과정보다 상태천이 확률 기반의 Soft 복호과정이 더 많은 장점을 가진다[10, 11]. 또한 Soft 복호과정에서 자유거리의 최소값이 1인 동일한 조건에서 Huffman 부호보다 양방향 가변길이 부호가 자체의 용장성(redundancy)으로 인해 더 좋은 성능를 보인다.

Lakovic[12]은 더 나은 자유거리 특성을 갖는 비대칭적 RVLC의 설계기법을 제안하였다. 설계된 RVLC의 자유거리는 그 최소값이 2로써 RVLC 자체가 전송오류에 강인하도록 하여 오류 내성 기능을 강화하였다. 하지만, Lakovic은 RVLC를 설계할 때에 Tsai의 방법에 기반하였으며, Tsai의 알고리즘이 가지는 단점에 따라 RVLC의 효율이 제한을 받는다.

본 논문에서는 자유거리를 증가시켜 보다 전송오류에 강인한 대칭적비대칭적 RVLC의 설계 알고리즘을 제안한다. 강인한 RVLC는 어두 조건, 어미 조건뿐 아니라 자유거리 조건을 만족하므로 평균부호길이가 길어진다. 제안된 알고리즘에서는 주어진 Huffman 부호로부터 알 수 있는 중요한 정보들을 기반으로 하고 부호화 효율의 척도인 평균부호길이를 함수로 이용하여 기존의 알고리즘보다 더 효율적인 강인한 대칭적비대칭적 RVLC를 설계한다.

2. 가변길이 부호의 거리 특성

$H_d(x,y)$는 같은 비트 길이를 갖는 이진 부호어 x, y의 Hamming 거리이고, c_i는 비트 길이가 l_i인 가변길이 부호어이다. 부호어의 총 수, 즉 주어진 심볼(symbol)의 총 수는 S이다. 또한 τ를 서로 다른 비트 길이의 전체 수라고 놓으면 각각의 부호어에 대한 비트 길이는 $l_1 < l_2 < \ldots < l_{\tau-1} < l_\tau$ 인 관계를 가지게 된다. 주어진 가변길이 부호에서 가장 짧은 비트 길이 l_1을 L_{min}이라 하고 가장 긴 비트 길이 l_τ를 L_{max}라고 한다. $|m_i|$를 이진 비트열 m_i의 비트 길이를 나타내는 연산자라고 하면 집합 $G_N = \{m_i : |m_i| = N\}$은 주어진 가변길이 부호 집합을 기반으로 비트 길이가 N인 생성 가능한 모든 이진 비트열로 구성된다.

가변길이 부호의 자유거리 d_{free}는 식 (1)과 같이 서로 다른 두 개의 이진 비트열 사이의 최소 Hamming 거리를 가리킨다.

$$d_{free} = \min\{H_d(m_i, m_j) : m_i, m_j \in G_N, i \neq j\} \quad (1)$$

가변길이 부호의 d_{free}를 직접 유도하는 일은 상당히 복잡하다. Lakovic[11]은 가변길이 부호의 자유거리를 최소 블록 거리, d_{Block}에서 유도하였다. d_{Block}은 주어진 가변길이 부호 집합에서 서로 다른 두 개의 부호어 사이의 최소 Hamming 거리를 가리킨다.

$$d_{Block} = \min\{H_d(c_i, c_j) : c_i, c_j, |c_i| = |c_j|, i \neq j\} \quad (2)$$

하지만 주어진 가변길이 부호의 집합에서 l_k 비트 길이의 부호어 c_k가 하나일 수 있고, 이러한 경우 d_{Block}은 정의되지 않는다. 따라서 d_{free}는 식 (3)으로 결정된다.

$$d_{free} = \begin{cases} 1, & d_{Block}\text{이 정의되지않은경우} \\ d_{Block}, & d_{Block}\text{이 정의되는경우} \end{cases} \quad (3)$$

Huffman 부호를 비롯하여 Huffman 부호에 기반한 모든 가변길이 부호의 d_{free}의 최소값은 1이다. 대칭적비대칭적 RVLC의 경우에도 $d_{free} \geq 1$이다.

제안된 알고리즘은 주어진 알파벳(alphabet)에 대하여 자유거리의 최소값을 증가시켜 $d_{free} \geq 2$인 조건을 만족하여 전송오류에 강인하면서 동시에 보다 효율적인 대칭적비대칭적 RVLC를 설계한다.

3. Huffman 부호의 특성

기존의 알고리즘과 같이 제안된 RVLC의 설계 알고리즘은 자유거리의 조건을 유지하며 가장 발생확률이 높은 심볼부터 짧은 비트 길이를 할당하여 강인한 대칭적비대칭적 RVLC를 설계한다. 즉 발생 확률이 낮은 심볼부터 처리하는 Huffman 부호의 생성 방식과는 할당 순서가 반대가 된다. 그리고 이러한 할당 순서를 통해 생성되는 강인한 대칭적 혹은 비대칭적인 RVLC의 수는 무한하며 우리는 이중에 가장 짧은 평균부호길이를 갖는 RVLC를 검색하고 설계해야 한다.

RVLC의 평균부호길이는 RVLC에서 시작 비트 길이 레벨(level)과 각 레벨에 존재하는 부호어의 수 및 해당 부호어에 의존한다. 그리고 주어진 Huffman 부호를 기반으로 각각의 요소들을 결정하게 되면 RVLC의 효율성을 증가시킬 수 있다.

최적화 된 Huffman 부호에서 가장 빈번하게 발생하는 심볼에 가장 짧은 비트 길이 L_{min}이 할당되고, L_{min}은 전체 심볼의 수와 발생확률의 분포 등에 영향을 받는다. 또한 L_{min}은 주어진 심볼과 발생확률 분포에서 최적의 가변길이 부호를 설계하기 위한 필수적인 요소 중 하나이다.

기존의 Huffman 부호 기반의 RVLC 설계 기법[5, 6, 9]은 모두 L_{min}에서 설계를 시작하였다. 비대칭적 RVLC의 경우에 부호어 할당이 유연하여 Huffman 부호와 가까운 특성을 가지므로 L_{min}의 선택이 효율성 개선에 기여할 수 있다. 하지만 L 비트 길이의 모두 '0' 비트로만 이루어진 부호어 $\mathbf{Z}_L$을 L_{min}에 강제 할당하는 $\mathbf{Z}_L$적응법[9]을 통해 대칭적 RVLC를 설계한 방식을 살펴보면 근사적으로 균등한 발생확률 분포를 갖는 정보원(source)일 경우에 L_{min}보다는 $(L_{min}-1)$에서 출발하여 생성된 RVLC가 더 좋은 효율을 가질 수 있다. 그리고 수 개의 심볼에 높은 발생확률이 할당된 경우에는 L_{min}과 $(L_{min}+1)$에서 더 나은 부호화 성능을 보인다. 즉, 대칭적 RVLC를 설계할 경우 주어진 발생확률 분포에 따라서 L_{min}과 $(L_{min}-1)$, $(L_{min}+1)$ 이 세 가지 값을 고려한다.

주어진 Huffman 부호와 RVLC에 대한 비트 길이 벡터 $n_{Huff}(i)$와 $n_{RVLC}(i)$는 각각 비트 길이 i를 가지는 부호어의 수라고 정의한다. 만약 자유거리의 조건을 적용하지 않으면 레벨 L에서 사용 가능한 RVLC의 수 $n_{RVLC}(L)$는 최대 2^L이 된다. 하지만 제안된 알고리즘에서는 $d_{free} \geq 2$인 부호어들을 고려하기 때문에 2^{L-1}으로 제한받으며 이는 이항정리(binomial theorem)를 통해 증명이 가능하다. 레벨 L에 존재하는 2^L의 부호어들을 '1'비트의 개수가 짝수인 집합 $O_{L:even}$과 홀수인 집합 O_{Odd}로 나누어 두 집합 중 어느 하나만을 선택하면 자유거리에 대한 조건을 만족할 수 있다. 각각의 집합에 존재하는 부호어의 총 수 $N(O_i)$ (i는 $Even$ 혹은 Odd)는 식 (4)~ 식 (7)을 통해 구할 수 있다.

$$\sum_{k=0}^{L} C_k^L = 2^L \quad (4)$$

여기서 C_b^a는 a개에서 b만큼을 선택하는 조합수이다.

$$\sum_{k=0}^{L} (-1)^k \cdot C_k^L = 0 \quad (5)$$

식 (4)와 (5)를 더하게 되면,

$$2 \cdot \sum_{k=0}^{L/2} C_{2k}^L = 2^L \quad (6)$$

따라서, $N(O_i)$의 최대 수는

$$N(O_i) \leq \sum_{k=0}^{L/2} C_{2K}^L = \sum_{k=0}^{L/2} C_{2k+1}^L = 2^{L-1} \quad (7)$$

여기서 인덱스 i는 $Even$ 혹은 Odd가 된다.

제안된 알고리즘에서 L_{min}부터 L_{max}까지 각 레벨의 $n_{RVLC}(i)$는 과도(exhaustive) 검색을 통해 결정한다. 비

대칭적 RVLC의 최상위 레벨에서 $n_{RVLC}(L_{min})$ 의 결정은 $n_{Huff}(L_{min})$을 참조한다. 하위 레벨의 $n_{RVLC}(i)$는 어두 조건과 어미 조건뿐 아니라 자유거리의 조건에 의해 제한되므로 최상위 레벨에서는 주어진 Huffman 부호의 $n_{Huff}(L_{min})$ 이상의 부호어를 할당해야 RVLC의 효율성을 높일 수 있다. 즉, $n_{Huff}(L_{min}) \le n_{RVLC}(L_{min}) \le 2^{L_{min}-1}$ 이어야 한다. 반면 대칭적 RVLC의 경우에 최상위 레벨과 부호어 수의 결정은 좀더 유연하다.

$\mathbf{Z}_L$[9]의 선택 또한 효율성의 개선에 기여할 수 있다. 이 부호어는 모두 '1'비트로만 구성된 부호어와 더불어 전체 가변길이 부호 집합의 모든 레벨에 존재하는 부호어로써 Huffman 부호어에서나 RVLC에서 두 부호어 모두 반드시 한 번은 선택된다. 따라서 $\mathbf{Z}_L$을 최상위 레벨에 할당해서 L_{min}에 하나의 부호어를 보장하여 평균부호길이를 줄일 수 있도록 한다.

주어진 발생확률의 분포와 그에 따른 Huffman 부호의 구성을 보면 Laplacian 혹은 지수 분포와 같이 수개의 심볼에 발생확률이 크게 집중되어 있을때 Huffman 부호는 몇몇 비트 레벨에 부호어를 할당하지 않고, 스킵(skip)하여 하위 레벨에 보다 많은 부호어를 보장하도록 한다.

효율적인 RVLC를 설계하려면 각 비트 레벨에 할당되는 부호어의 수를 알맞게 조절해야 한다. $\lceil S/2 \rceil$개의 부호어를 포함하는 상위 레벨에서는 많은 부호어를 보다 짧은 비트에 할당하는 동시에 하위 레벨에 많은 부호어들을 보장해야 하고, 반면에 하위 레벨에서는 짧은 비트 길이에 많은 부호어들을 할당해야 평균부호길이를 줄일 수 있다. 여기서 $\lceil x \rceil$는 x보다 크거나 같은 수 중 가장 작은 정수이다. 따라서, 본 논문에서는 레벨 스킵 적응법을 제안한다. 레벨 스킵 적응법을 사용하면 주어진 Huffman 부호에서 $\lceil S/2 \rceil$개의 부호어를 포함하는 상위 레벨에서 스킵된 레벨 L_{Skip}들이 존재하면 RVLC를 설계할 때에도 L_{Skip}에는 부호어를 할당하지 않는다. 하위 레벨에서는 단순히 어두 조건, 어미 조건, 그리고 자유거리 조건을 만족하는 부호어들을 선택하게 된다.

4. 평균부호길이 함수

우리는 가변길이 부호의 부호화 성능을 측정하기 위해 평균부호길이를 사용한다. 주어진 하나의 알파벳에 대해 생성된 대칭적비대칭적 RVLC에서 특정 레벨 L에 존재하는 부호어 수가 $n_{RVLC}(L)=M$ 이고 L보다 상위 레벨에 할당된 부호어가 같은 RVLC의 집합을 $R(L;M)$이라고 정의한다. 표 1은 $R(3;2)$의 한 예를 보여준다.

표 1. 비대칭적 RVLC에서 집합 $R(L;M)$

심볼	비대칭적 RVLC에서 $R(3;2)$					
A	2	00	2	00	2	00
B	3	010	3	011	3	101
C	3	011	3	101	3	111
D	4	1001	4	1001	4	0110
E	4	1101	4	1110	4	1001
F	4	1110	4	1111	5	01010
G	4	1111	5	01010	5	01011

또한 이 집합에서 RVLC의 최소 평균부호길이값 $\min(\bar{L}(R))$ 을 대응시킨 함수 $f_{\bar{L}}(x)$를 평균부호길이 함수로 정의한다. 여기서 x값은 $n_{RVLC}(L)$이 된다. 평균부호길이 함수 $f_{\bar{L}}(x)$는 구간 내에서 최소값을 갖는 볼록(convex) 함수 이며 이 최소값을 기준으로 하여 부호어 수에 따라 단조 증가 혹은 단조 감수할 수 있다. 예를 들어, 1부터 최대 $2^{L_{min}-1}$의 범위를 갖는 $n_{RVLC}(L_{min})$에 대하여 $f_{\bar{L}}(x)$는 그림 1과 같이 이 구간에서 최소값을 가지게 된다. $f_{\bar{L}}(x)$는 볼록 함수이므로 국부적인(local) 최소값은 전체적인(global) 최소값을 대표하게 되고 보다 효율적인 RVLC는 현재 레벨 L에서 최소값을 갖는 지점에 위치한다. 현재 레벨이 L_{min}이면 최소값의 위치가 $n_{RVLC}(L_{min})$이 된다.

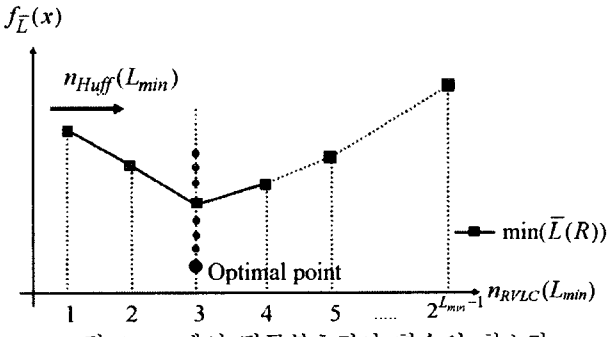

그림 1. L_{min}에서 평균부호길이 함수의 최소값

대칭적 RVLC의 경우에는 $(L_{min}-1)$ 또는 $(L_{min}+1)$에 대해서도 고려해야 하므로 $n_{RVLC}(L_{min}\pm1)$ 에 대한 $f_{\bar{L}}(x)$와 비교를 한 후 결정한다. $(L_{min}-1)$은 근사적으로 균등한 분포를 갖는 소스에 대한 것이므로 주어진 Huffman 부호의 상위 레벨에서 레벨 스킵이 발생하였다면 $(L_{min}+1)$에 대한 $f_{\bar{L}}(x)$를 고려해야 한다. 반대로 레벨 스킵이 발생하지 않았다면 $(L_{min}-1)$에 대한 $f_{\bar{L}}(x)$가 고려되어야 한다. 그리고, $(L_{min}+1)$이 L_{Skip} 중 하나와 같더라도 해당 레벨에 부호어를 할당하도록 한다.

$n_{RVLC}(L_{min})$를 결정한 후 $\lceil S/2 \rceil$개의 부호어를 포함하는 상위 레벨에서 각 비트 레벨의 부호어 수 $n_{RVLC}(i)$, $(i>L_{min})$를 결정하는 방식은 다음과 같다.

1) 이전 비트 레벨 ($L_{Curr}-1$)까지 선택된 RVLC에 의해 현재 레벨 L_{Curr}에서 사용 가능한 부호어가 결정된다. 이 때 부호어들의 자유거리는 아직 $d_{free} \ge 1$이다.
2) L_{Curr} 의 부호어들을 O_{Even} 과 O_{Odd} 로 나눈다. $N(O_i)$ (i는 $Even$ 혹은 Odd)를 비교하여 큰 집합의 부호어를 선택한다. 또한 $N(O_i)$ 가 작은 집합에서 먼저 선택된 부호어들과 자유거리를 비교하여 $d_{free} \ge 2$ 를 만족하는 부호어들을 모두 선택한다. 선택된 부호어의 총 수를 $n_{RVLC}(L_{Curr})$로 놓는다.
3) ($n_{RVLC}(L_{Curr})-1$)에서 $f_{\bar{L}}(x)$ 를 구하고 최소값이 존재하면 $n_{RVLC}(L_{Curr})$ 를 다음과 같이 대치한다. 1의

범위는 $0 \le l < n_{RVLC}(L_{Curr})$ 이다.

$$n_{RVLC}(L_{Curr}) = n_{RVLC}(L_{Curr}) - l \quad (6)$$

l은 0부터 '1'씩 증가된다. 하나의 l에 대한 $N(R)$은 $\binom{n_{RVLC}(L_{Curr})}{n_{RVLC}(L_{Curr})-l}$이 되고, 여기서 $\binom{a}{b}$는 C_b^a이다.

4) 현재의 $f_{\bar{L}}(x)$ 값이 이전의 값보다 크다면 현재 레벨에서 선택되는 부호어와 그 수는 그림 2와 같이 이전의 $n_{RVLC}(L_{Curr})$와 그에 따른 부호어이다.

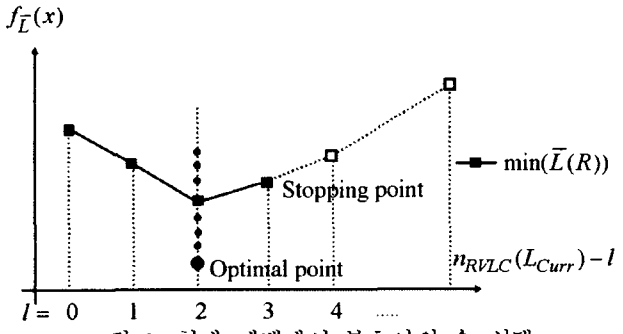

그림 2. 현재 레벨에서 부호어의 수 선택

본 논문에서 제안한 강인한 RVLC의 설계 방법을 정리하면 다음과 같다.

1) 그림 2와 같이 $n_{Huff}(L_{min})$에서 출발하여 $f_{\bar{L}}(x)$의 최소값이 위치하는 지점을 $n_{RVLC}(L_{min})$으로 결정한다. RVLC를 생성할 때 레벨 스킵 적응법을 적용한다. L_{min}에서 하나의 부호어는 반드시 $\mathbf{Z}_L$이 사용된다. 대칭적 RVLC이며 주어진 Huffman 부호의 상위 레벨에서 스킵된 레벨이 없으면 $(L_{min}-1)$에 대해서 고려해야 한다. 이 때에는 $\mathbf{Z}_{L-1}$이 사용된다. 또한 스킵된 레벨이 있으면 $(L_{min}+1)$에 대하여 $\mathbf{Z}_{L+1}$이 적용된다.
2) $\lceil S/2 \rceil$개의 심볼을 포함하는 상위 레벨에서 각 비트 레벨에 사용 가능한 부호어를 결정한다.
3) 하위 레벨에서 S에 대한 모든 부호어가 할당될 때까지 어두 조건, 어미 조건, 그리고 자유거리 조건을 만족하는 부호어를 검색하면 오류에 강인한 RVLC를 얻을 수 있다.

5. 실험 결과

표 2와 표 3은 기존의 방법들과 제안된 방법을 가지고 압축 알고리즘에 대한 성능 평가를 위해 사용되는 Canterbury Corpus 파일(http://corpus.canterbury.ac.nz)에 대하여 강인한 대칭적비대칭적 RVLC를 생성하여 평균부호길이를 비교한 결과를 보여준다. 파일 F1부터 F11까지 기존의 Lakovic[12]의 방법으로 생성된 강인한 RVLC의 평균부호길이보다 부호화 효율이 현저하게 개선되었음을 알 수 있다. 표 2와 표 3에서 음영으로 표시된 F2, F4, F8, F9, F10, F11에 대한 RVLC의 평균부호길이는 레벨 스킵 적응법이 사용된 것으로 제안된 레벨 스킵 적응법이 RVLC의 효율성에 크게 기여하는 사실을 알 수 있다. 더욱이 비대칭적 RVLC에서 F2, F5, F6을 제외한 나머지 파일들에서는 $d_{free} \ge 1$인 Lin[8]의 RVLC보다 제안된 $d_{free} \ge 2$인 강인한 RVLC의 부호화 성능이 더 뛰어나다.

표 4와 표 5는 기존의 방법과 제안된 방법을 가지고 영어 알파벳에 대하여 강인한 대칭적비대칭적 RVLC를 생성하여 그에 따른 부호어 할당과 평균부호길이를 보여준다. 영어 알파벳의 경우 레벨 스킵 적응법이 사용되지 않았고 따라서 대칭적 RVLC에서 $\mathbf{Z}_2$가 사용되었다. 영어 알파벳에 대하여 기존의 Tseng의 방법보다 제안된 대칭적 RVLC는 1.2%정도 그리고, Lin의 방법보다 제안된 비대칭적 RVLC는 2.5%정도의 부호화 성능 개선이 있었다.

6. 결론

본 논문에서 우리는 부호어의 자유거리를 증가시켜 전송오류에 강인하도록 설계된 대칭적비대칭적 RVLC를 생성하는 새로운 알고리즘을 제안하였다. 제안된 RVLC는 어두 조건과 어미 조건뿐 아니라 $d_{free} \ge 2$인 자유거리 조건도 만족하기 때문에 부호화 효율이 저하될 가능성이 매우 높다. 이를 해결하기 위하여 본 논문에서는 Huffman 부호를 기반으로 레벨 스킵 적응법과 평균부호길이 함수의 특성을 이용하였다. 레벨 스킵 적응법은 주어진 Huffman 부호의 부호어가 할당된 상태를 RVLC의 설계에 적용하여 제안된 RVLC의 평균부호길이가 Huffman 부호의 평균부호길이에 접근할 수 있도록 한다. 평균부호길이 함수는 RVLC의 각 비트 레벨에서 부호어들을 조절하여 부호화 성능을 개선시킨다. 실험 결과를 통해 제안된 기법으로 기존의 RVLC보다 더 효율적이면서 전송오류에 강인한 RVLC를 얻을 수 있음을 알 수 있다.

감사의 글

본 연구는 광주과학기술원(K-JIST)과 광주과학기술원 실감방송연구센터를 통한 대학IT연구센터(ITRC), 그리고 교육부 두뇌한국21(BK21) 정보기술사업단의 지원에 의한 것입니다.

참고문헌

[1] D. Huffman, "A method for the construction of minimum redundancy codes," *Proc. Inst. Radio. Engr.*, vol. 40, pp. 1098-1101, Sept. 1952.
[2] J.J. Rissanen and G.G. Langdon. Jr., "Arithmetic coding," *IBM J. Res. Develop.*, 23, pp. 149-162, 1979.
[3] ISO/IEC 14496-2, "Information technology – coding of audio/video objects," *Final Draft Int. Std.*, Part 2 : Visual, Oct. 1998.
[4] ITU-T Rec. H.263, "Video coding for low bit communications," Annex V, 2000.
[5] Y. Takishima, M. Wada and H. Murakami, "Reversible variable length codes," *IEEE Trans. Comm.*, vol. 43, pp. 158-162, Feb. 1995.
[6] C.W. Tsai and J.L. Wu, "On constructing the

Huffman-code-based reversible variable-length codes," *IEEE Trans. Comm.*, vol 49, pp. 1506-1509, Sept. 2001.

[7] H.W. Tseng and C.C. Chang, "Construction of symmetrical reversible variable length codes using backtracking," *Comp. J.*, vol. 46, no. 1, Jan. 2003.

[8] C.W. Lin, Y.J. Chuang, and J.L. Wu, "Generic construction algorithms for symmetric and asymmetric RVLCs," *Proc. IEEE Int. Conf. Communication Systems*, vol. 2, pp. 968-972, Nov. 2002.

[9] W.H. Jeong and Y.S. Ho, "Design of symmetrical reversible variable-length codes from the Huffman code," *Picture Coding Symposium*, pp. 135-138, April 2003.

[10] S. Kaiser and M. Bystrom, "Soft decoding of variable length codes," *Proc. IEEE Int. Conf. Commun.*, pp. 1203-1207, 2000.

[11] R. Bauer and J. Hagenauer, "On variable length codes for iterative source/channel-decoding," *Proc. IEEE Data Compression Conf.*, pp. 273-282, June 2000.

[12] L. Lakovic and J. Vallasenor, "On design of error-correcting reversible variable length codes," *IEEE Communs. Letters*, vol. 6, pp. 337-339, Aug. 2002.

표 2. 다양한 소스 파일에 대하여 기존의 알고리즘과 제안된 알고리즘을 사용하여 생성된 강인한 대칭적 RVLC의 평균부호길이 비교 (▭ : 레벨 스킵 적응법 적용)

File		부호어 수 (심볼수)	Huffman 부호	대칭적 RVLC			
				Tsai의 방법	Tseng의 방법	Lakovic의 방법 ($d_{free} \geq 2$)	제안된 방법 ($d_{free} \geq 2$)
			평균부호길이	평균부호길이	평균부호길이	평균부호길이	평균부호길이
F1	asyoulik.txt	68	**4.84465**	5.27886	5.21025	5.41179	**5.31476**
F2	Alice29.txt	74	**4.61244**	5.01398	4.93155	5.21075	**5.02562**
F3	Xargs.1	74	**4.92382**	5.39863	5.33996	5.52851	**5.44342**
F4	grammar.lsp	76	**4.66434**	5.04757	5.01774	5.24805	**5.13455**
F5	Plabn12.txt	81	**4.57534**	4.94473	4.89527	4.98592	**4.98433**
F6	lcet10.txt	84	**4.69712**	5.12232	5.01682	5.22562	**5.10563**
F7	cp.html	86	**5.26716**	5.85839	5.81173	6.09414	**5.98710**
F8	Fields.c	90	**5.04090**	5.47596	5.46332	5.69381	**5.22125**
F9	Ptt5	159	**1.66091**	1.77735	1.75992	1.80165	**1.79499**
F10	Sum	255	**5.36504**	6.10683	6.03917	6.32367	**6.27025**
F11	kennedy.xls	256	**3.59337**	4.25681	4.27209	4.37120	**4.32269**

표 3. 다양한 소스 파일에 대하여 기존의 알고리즘과 제안된 알고리즘을 사용하여 생성된 강인한 비대칭적 RVLC의 평균부호길이 비교 (▭ : 레벨 스킵 적응법 적용)

File		부호어 수 (심볼수)	Huffman 부호	비대칭적 RVLC			
				Tsai의 방법	Lin의 방법	Lakovic의 방법 ($d_{free} \geq 2$)	제안된 방법 ($d_{free} \geq 2$)
			평균부호길이	평균부호길이	평균부호길이	평균부호길이	평균부호길이
F1	asyoulik.txt	68	**4.84465**	5.01142	5.00954	5.20114	**4.99592**
F2	alice29.txt	74	**4.61244**	4.80326	4.68871	5.01147	**4.73161**
F3	xargs.1	74	**4.92382**	5.07334	5.16087	5.34651	**5.08761**
F4	grammar.lsp	76	**4.66434**	4.85461	4.78581	5.04473	**4.76816**
F5	plabn12.txt	81	**4.57534**	4.80659	4.64910	4.83863	**4.69002**
F6	lcet10.txt	84	**4.69712**	4.87868	4.74177	4.95115	**4.81642**
F7	cp.html	86	**5.26716**	5.37113	5.77080	5.42775	**5.28917**
F8	fields.c	90	**5.04090**	5.26987	5.20278	5.39520	**5.17480**
F9	Ptt5	159	**1.66091**	1.71814	1.70401	1.79911	**1.67945**
F10	Sum	255	**5.36504**	5.49767	6.01870	6.30885	**5.49070**
F11	kennedy.xls	256	**3.59337**	3.89401	3.85384	3.96247	**3.82626**

표 4. 영어 알파벳에 대한 강인한 대칭적 RVLC의 부호어 할당과 부호화 성능 비교

발생확률		Huffman 부호		Tsai의 방법		Tseng의 방법		Lakovic의 방법 ($d_{free} \geq 2$)		제안된 방법 ($d_{free} \geq 2$)	
		L	부호어	L	부호어	L	부호어	L	부호어	L	부호어
E	0.14878570	3	001	3	010	3	000	3	010	2	00 (Z_2)
T	0.09354149	3	110	3	101	3	111	3	101	3	010
A	0.08833733	4	0000	4	0110	3	010	4	0110	3	101
O	0.07245796	4	0100	4	1001	3	101	4	1001	4	1111
R	0.06872164	4	0101	4	0000	4	0110	4	0000	4	0110
N	0.06498532	4	0110	4	1111	4	1001	4	1111	4	1001
H	0.05831331	4	1000	5	01110	5	00100	5	01110	5	11011
I	0.05644515	4	1001	5	10001	5	11011	5	10001	5	01110
S	0.05537763	4	1010	5	00100	5	01110	5	00100	5	10001
D	0.04376834	5	00010	5	11011	5	10001	5	11011	6	110011
L	0.04123298	5	00011	6	011110	6	001100	6	011110	6	011110
U	0.02762209	5	10110	6	100001	6	110011	6	100001	6	100001
P	0.02575393	5	10111	6	001100	6	011110	6	001100	7	1110111
F	0.02455297	5	11100	6	110011	6	100001	6	110011	7	1100011
M	0.02361889	5	11110	7	0111110	7	0010100	7	0111110	7	0111110
C	0.02081665	5	11111	7	1000001	7	1101011	7	1000001	7	1000001
W	0.01868161	6	011100	7	0010100	7	0011100	7	0011100	8	11100111
G	0.01521216	6	011101	7	1101011	7	1100011	7	1100011	8	11000011
Y	0.01521216	6	011110	7	0011100	7	0111110	7	0001000	8	01111110
B	0.01267680	6	011111	7	1100011	7	1000001	7	1110111	8	10000001
V	0.01160928	6	111011	7	0001000	8	00111100	8	01111110	9	111000111
K	0.00867360	7	1110100	7	1110111	8	11000011	8	10000001	9	110000011
X	0.00146784	8	11101011	8	01111110	8	01111110	8	00111100	9	110101011
J	0.00080064	9	111010101	9	011111110	8	10000001	9	011111110	9	011111110
Q	0.00080064	10	1110101000	10	0111111110	9	011111110	10	0111111110	9	100000001
Z	0.00053376	10	1110101001	10	1000000001	9	100000001	10	10000000001	10	1110000111
평균부호길이		4.15572392		4.60728507		4.46463681		4.6272572		**4.567250**	

표 5. 영어 알파벳에 대한 강인한 비대칭적 RVLC의 부호어 할당과 부호화 성능 비교

발생확률		Huffman 부호		Tsai의 방법		Lin의 방법		Lakovic의 방법 ($d_{free} \geq 2$)		제안된 방법 ($d_{free} \geq 2$)	
		L	부호어	L	부호어	L	부호어	L	부호어	L	부호어
E	0.14878570	3	001	3	000	3	000	3	000	3	000 (Z_3)
T	0.09354149	3	110	3	111	3	100	3	111	3	011
A	0.08833733	4	0000	4	0101	3	101	4	0101	3	101
O	0.07245796	4	0100	4	1010	4	0010	4	1010	3	110
R	0.06872164	4	0101	4	0010	4	0011	4	0110	4	0010
N	0.06498532	4	0110	4	1101	4	0110	4	1001	4	0100
H	0.05831331	4	1000	4	0100	4	0111	4	0011	4	1001
I	0.05644515	4	1001	4	1011	4	1110	4	1100	4	1111
S	0.05537763	4	1010	4	0110	4	1111	5	00100	5	00111
D	0.04376834	5	00010	5	11001	5	01001	5	11011	5	01010
L	0.04123298	5	00011	5	10011	5	01010	5	01110	5	10001
U	0.02762209	5	10110	5	01110	5	01011	5	10001	5	11100
P	0.02575393	5	10111	5	10001	5	11001	6	010010	6	001100
F	0.02455297	5	11100	6	001100	5	11010	6	101101	6	010111
M	0.02361889	5	11110	6	011110	5	11011	6	100001	6	100001
C	0.02081665	5	11111	6	100001	6	010001	6	011110	6	111010
W	0.01868161	6	011100	7	1001001	6	110001	6	001011	7	0011010
G	0.01521216	6	011101	7	0011100	7	0100001	6	1101100	7	0101100
Y	0.01521216	6	011110	7	1100011	7	1100001	7	0100010	7	1000001
B	0.01267680	6	011111	7	0111110	8	01000001	7	1011101	7	1110111
V	0.01160928	6	111011	7	1000001	8	11000001	7	0010100	8	00110111
K	0.00867360	7	1110100	8	00111100	9	010000001	7	1101011	8	01011010
X	0.00146784	8	11101011	8	11000011	9	110000001	8	10111101	8	10000001
J	0.00080064	9	111010101	9	100101001	10	0100000001	9	010000010	8	11101100
Q	0.00080064	10	1110101000	10	0011101001	10	1100000001	10	0100000010	9	001101100
Z	0.00053376	10	1110101001	10	1001011100	11	01000000001	10	1011111101	9	010110111
평균부호길이		4.15572392		4.30677804		4.18734808		4.34534		**4.236589**	