

Huffman 부호에 기초한 대칭적 가역 가변길이 부호의 생성 방법

정 옥 현, 호 요 성
광주과학기술원 정보 통신 공학과
전화 : 062-970-2263 / 핸드폰 : 019-9183-3542

A Construction Algorithm for Symmetrical Reversible Variable Length Codes Based on the Huffman Code

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

Abstract

In this paper, we propose a new efficient algorithm to construct a symmetrical reversible variable length code (RVLC) from a given Huffman code table. The symmetrical RVLC has the advantage of memory usage, while the asymmetrical RVLC has less overheads. The proposed algorithm has a simpler construction process and also demonstrates a better performance in terms of the average codeword length than those of other conventional RVLC algorithms.

I. 서 론

대부분의 정지영상 및 동영상 부호화 방식에서는 엔트로피 부호화 단계에서 허프만 부호[1]와 같은 가변길이 부호를 사용하고 있다. 가변길이 부호는 엔트로피에 근접한 높은 압축 효율을 제공하지만, 무선 환경과 같이 잡음이 심한 채널에서는, 예측 부호화와 더불어, 전송 오류에 대해서 매우 취약하다는 단점이 있다. 따라서, 단일 비트에 대한 비트 반전, 삽입, 제거 등의 오류가 발생하더라도 심각한 문제를 발생시킬 수 있다.

최근 MPEG-4 표준화 작업에서는 이러한 영향을 줄이기 위한 오류 내성 부호화 기법의 하나로써 가역 가변길이 부호(reversible variable length code, RVLC)

가 제안되었다. RVLC는 순방향과 역방향으로 모두 유일한 복호가 가능한 부호로써, 오류가 발생했을 때 수신된 데이터로부터 손상되지 않은 데이터를 가능한 한 많이 복원할 수 있다는 장점을 가지고 있다. RVLC는 부호어 구성의 모양에 따라서 대칭적(symmetrical) RVLC와 비대칭적(asymmetrical) RVLC로 나뉜다[2]. 대칭적 RVLC는 하나의 부호어(codeword) 테이블만 사용한다는 장점이 있고, 비대칭적 RVLC는 보다 적은 평균 부호 길이를 제공한다는 장점이 있어, 각각 H.263+와 MPEG-4에 채택되어 사용되고 있다[4, 5].

Takishima *et al.*[2]은 주어진 허프만 부호에 기초하여 평균 부호 길이가 엔트로피에 가깝도록 설계된 대칭적 RVLC 생성 알고리즘을 제안하였고, Tsai *et al.*[3]은 이 알고리즘을 수정하여 보다 적은 평균 부호 길이를 갖는 대칭적 RVLC 알고리즘을 제안하였다. 하지만, 이 알고리즘은 전체 이진 나무(binary tree)에서 부호를 찾아야 하고, 부호 생성 이전에 주어진 허프만 부호를 기초로 모든 레벨에서 사용 가능한 대칭적 RVLC의 수를 미리 계산해야 한다. 그리고, 이를 허프만 부호의 모든 레벨에 걸쳐 적용시켜야 하는 복잡성이 존재하고, 계산 및 적용 과정에 제한 사항이 있어 사용 가능한 대칭적 RVLC를 놓치는 경우가 발생한다.

본 논문에서는 전체 이진 나무의 반쪽 영역에서 부호를 검색하고, 주어진 허프만 부호에 대한 적용과정을 매우 단순화시켜 복잡도를 줄이면서, 동시에 적용과정 자체가 평균 부호 길이를 줄이는데 기여하고, 사용 가능한 대칭적 부호어들을 놓치지 않아 평균 부

호 길이가 역시 줄일 수 있는 효과적인 대칭적 RVLC 생성 알고리즘을 제안한다.

II. 비트 반전을 이용한 영역 검색

그림 1은 허프만 부호화에서 각각의 기호에 대응하는 화를 선으로 연결하여 구성된 이진 나무의 한 예를 보여준다. 최상단의 절점(root)에서 각각 오른쪽과 왼쪽으로 향하는 가지(branch)에 0과 1을 할당하고, 이진 나무에서의 단점(leaf)만이 부호어가 될 수 있다. 이와 같이 모든 단점만이 부호어가 되기 때문에 허프만 부호는 어두(prefix) 조건을 만족할 수 있다.

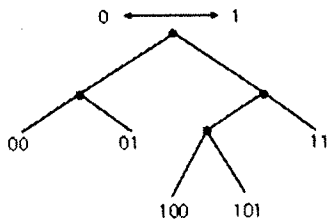


그림 1. 허프만 부호와 이진 나무

이진 나무의 최상단의 절점에서 오른쪽과 왼쪽으로 나온 가지에 각각 0과 1을 할당하는 방식은 자유이다. 따라서, 우리는 그림 2와 같이 동일한 평균 부호 길이를 가지며, 어두 조건을 만족하는 허프만 부호어를 얻을 수 있다. 그리고, 0과 1의 할당 방향을 바꾸는 것은 모든 부호어의 비트를 반전시키는 것과 같음을 알 수 있다.

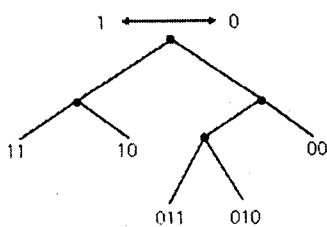


그림 2. 할당 방향이 바뀐 이진 나무의 부호

RVLC는 순방향과 역방향의 복호를 지원하기 위하여 어두 조건 뿐 아니라, 어미(suffix) 조건도 만족시켜야 한다. 하지만 대칭적 RVLC는 '대칭성'이란 특수한 성질 때문에 어두 조건을 만족하게 되면 어미 조건 역시 만족하게 된다. 그리고, 이 '대칭성'은 또 하나의 특징을 가진다. 선택된 대칭적 RVLC를 비트 반전시키면 같은 부호 길이를 가지며 어두 조건과 어미 조건을 만족시키는 또 하나의 대칭적 RVLC를 얻을 수 있다.

예를 들어, 그림 1과 그림 2의 이진 나무에서 '101'이 대칭적 RVLC로 선택되었다면, '010' 역시 같은 부호 길이를 가지며 어두 조건과 어미 조건을 만족시키는 대칭적 RVLC로 선택될 수 있다.

이러한 특성을 이용하여, 제안된 알고리즘에서 대칭적 RVLC를 찾는 과정은 전체 이진 나무에서 부호어를 찾지 않고 최상단의 절점에서 0으로 할당된 한쪽 영역에서만 이루어진다. 0의 영역에서는 주어진 기호의 절반 $\lfloor S/2 \rfloor$ 에 해당하는 수의 부호어를 찾는다. 전체 기호 S에 대한 최종 대칭적 RVLC는 이미 선택된 부호어와 그것을 비트 반전시킨 부호어로 구성된다. 선택된 부호어들과 비트 반전된 부호어들은 각 영역에서 유일하게 복호될 수 있다. 그리고, 이 두 부호어들은 각각 '0'과 '1'로 시작하기 때문에 어두 조건을 만족하므로, 결국 최종의 대칭적 RVLC는 유일하게 복호될 수 있다. 이때 한쪽 영역에서 각 레벨 l에 존재하는 모든 대칭적인 부호어의 수는 식 (1)과 같다.

$$m_0(l) = 2^{\lfloor (l+1)/2 \rfloor - 1} \quad (1)$$

III. 대칭적 RVLC 부호 생성

생성된 대칭적 RVLC의 평균 부호 길이를 줄이기 위해서는 상위 레벨에서 선택 가능한 부호어를 놓치는 일이 없어야 하고, 기호의 발생 확률에 적합하도록 하기 위해 주어진 허프만 부호에 적용을 시켜야 한다.

1. 부호어의 선택

상위 레벨에서 하나의 대칭적인 부호어가 선택되면, 하위 레벨의 대칭적인 부호어 중에는 어두 조건에 위배(violation)되는 일이 발생한다. 상위 레벨에서 선택된 대칭적인 부호어가 복호 가능한 가변길이 부호어가 되기 위해서는 하위 레벨에서 어두 조건에 위배되는 대칭적 부호어를 제거해야 한다.

0으로 할당된 한쪽 영역에서 부호어를 찾기 때문에, 이 영역에서 선택된 부호어는 어두와 어미에 최소한 하나의 '0' 비트를 가진다. 그리고, 각 레벨에는 모두 '0' 비트로만 이루어진 부호어, Z_L 이 존재한다. 이 Z_L 에 대하여 다음 두 가지 사항을 고려해야 한다.

첫째, L 레벨의 Z_L 이 선택되었다면, 선택된 부호어들은 Z_L 과 더불어 어두와 어미에 각각 L 보다 작은 수의 '0' 비트를 가지는 부호어들로 구성된다.

둘째, 전체 기호에 대한 부호어는 비트 반전된 부호어를 포함하므로, L 레벨의 모두 '1' 비트로만 이루어진 부호어도 얻을 수 있다. 따라서 주어진 기호의

발생 확률이 맞도록, 적당한 상위 레벨에서 Z_L 이 선택 된다면 L 레벨에 두 개의 부호어를 더 추가시키므로 평균 부호 길이를 줄이는 데에 기여할 수 있다.

2. Z_L 적용법

허프만 부호화에서 가장 작은 비트 길이의 부호어는 가장 높은 발생 확률을 가진다. 그런데, 허프만 부호화가 최적의 부호를 구성하는 과정을 보면, 각 기호의 발생 확률, 분포와 기호의 개수에 의해 가장 높은 발생 확률을 가지는 부호어의 비트 길이는 달라지게 된다. 하지만, 이 비트 길이는 주어진 조건 안에서 최적의 부호를 위해 필수적이다.

본 논문에서는 주어진 허프만 부호에 적용시켜 평균 부호 길이를 줄이기 위해서 Z_L 적용법을 제안한다. Z_L 적용법은 Z_L 의 레벨, 즉 비트 길이를 주어진 허프만 부호에서 가장 작은 비트 길이의 부호어의 비트 길이와 같도록 일치시키는 것이다.

제안된 대칭적 RVLC의 생성 방법은 다음과 같다.

- 1) 0으로 할당된 한쪽 영역에서, 주어진 허프만 부호어에 따라서 Z_L 을 선택한다. L 은 가장 작은 허프만 부호어의 비트 길이와 같다.
- 2) $\lceil S/2 \rceil$ 만큼 대칭적 부호어를 찾을 때까지, 상위 레벨부터 사용 가능한 대칭적 부호어들을 모두 선택한다. 모든 선택 과정은 반드시 하위 레벨에서 어두 조건에 위배되는 부호어의 제거가 뒤따른다.
- 3) 선택된 대칭적 부호어와 이 부호어들을 비트 반전시킨 부호어으로써 최종적인 대칭적 RVLC를 얻는다.

따라서, 각 레벨에서 사용 가능한 대칭적 부호어의 수 $m(l)$ 은 식 (2)와 같다.

$$m(l) = m_0(l) - \sum v_z(l) - \sum_{i=1}^{L-1} v_{ci}(l) \quad (2)$$

식 (2)에서 두 번째 항의 $v_z(l)$ 은 Z_L 에 따라서 위배되는 부호어의 수이고, 세 번째 항의 $v_{ci}(l)$ 은 Z_L 을 제외한 이전 레벨에서 선택된 부호어에 따라서 위배되는 부호어의 수이다. L 은 Z_L 의 비트 길이이고, 첨자 i 는 부호어에서 어두와 어미에 붙은 '0' 비트의 수이다. 여기서 $v_z(l)$ 의 값은 다음과 같다.

(i) $l < 2L+1$ 일 때, $v_z(l) = 1$,

(ii) $2L+1 \leq l$ 일 때, $v_z(l) = m_0(l-2L)$

$v_{ci}(l)$ 의 값은 다음과 같다. L' 은 이전에 선택된 대칭적 부호어의 비트수이다.

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

(ii) $2L'+1 \leq l$ 일 때, $v_{ci}(l) = m_0(l-2L')$

표 1은 식 (2)에 의하여 계산된 $m(l)$ 값을 보여준다.

표 1. Z_L 선택에 따른 각 레벨에서의 RVLC 수

Z_L	Z_1		Z_2		Z_3		Z_4		Z_5	
	H	F	H	F	H	F	H	F	H	F
L	1	Z_1	1	0	0	0	0	0	0	0
	2	1	2	Z_2	2	0	0	0	0	0
	3	1	3	1	4	Z_3+1	4	1	2	1
	4	1	4	1	6	1	6	Z_4+1	6	1
E	5	1	5	1	8	2	10	2	10	Z_5+2
	6	1	6	1	10	2	14	2	14	2
V	7	1	7	1	12	3	20	4	22	4
	8	1	8	1	14	2	24	3	28	3
E	9	1	9	1	16	4	32	6	40	7
	10	1	10	1	18	3	38	5	50	6
L	11	1	11	1	20	6	50	9	68	11
	12	1	12	1	22	4	58	7	82	9
L	13	1	13	1	24	9	76	15	112	19
	14	1	14	1	26	8	92	14	140	17
	15	1	15	1	28	16	124	27	194	34
	16	1	16	1	30	13	150	24	222	30

H : 반 영역에서의 선택된 부호어 수, $m(l)$

F : 전체 이진 나무에서 누적 부호어 수, $\sum (2 \times m(l))$

표 1에서 Z_L 의 경우에는 0으로 할당된 영역에서 '0'만을 선택하므로, 하위 레벨에서는 더 이상 대칭적 부호어를 선택할 수 없다. 따라서, 나머지 부호어들은 1로 할당된 영역에서 선택된 부호어들이다.

IV. 실험 결과

Takishima의 알고리즘과 Tsai에 의해 수정된 알고리즘에서는 전체 이진 나무에서 부호어를 찾으며, 각 레벨에서 대칭적 RVLC의 수를 미리 계산한다. 그리고, 생성 과정에서 모든 레벨에 걸쳐 주어진 허프만 부호어의 수에 적용시키며 수정해야 하므로 상당히 복잡하다. 이에 비해, 제안된 알고리즘에서는 비트 반전을 이용하여 한쪽 영역에서 검색을 수행하고, Z_L 적용법을 사용하여 단지 가장 작은 비트 길이만을 확인하며, Z_L 의 레벨을 그에 맞춰 선택하고 나머지 대칭적 부호어를 선택하므로 매우 간편하다.

또한, 제안된 방법에서는 Z_L 적용법을 사용하여 보다 효율적인 부호의 생성 방법을 제시하였다. Z_L 적용법은 주어진 환경에서 가장 높은 발생 확률을 갖는 레벨에 두 개의 부호어를 할당하여 평균 부호 길이를 줄이는 효과를 가져올 수 있다. 그리고, 기존의 알고리즘에서 주어진 허프만 부호어의 모든 레벨에 걸쳐 적용시키는 방법은 결국 특정 레벨에서 부호어수에 제한 사항을 발생시킨다. 이와 달리, 제안된 방법에서는 Z_L 을 선택한 후 가장 작은 비트의 대칭적 부호어부터 차례로 선택하므로 특정 레벨에서 부호어의 수가 제한되지 않는다. 따라서, 사용 가능한 대칭적 부호어를 놓치지

는 경우가 발생하지 않는다.

본 논문에서 제안한 대칭적 RVLC의 생성 알고리즘의 성능을 평가하기 위해 표 2와 같이 Takishima가 선택했던 영어 알파벳에 대하여 기존의 알고리즘과 제안된 알고리즘으로 대칭적 RVLC를 생성하였다.

표 2. 영어 알파벳에 대한 부호화 성능 비교

발생 확률	허프만 부호 C1		Takishima 알고리즘 C2		Tsai 알고리즘 C3		제안된 알고리즘 C4	
	L	부호어	L	부호어	L	부호어	L	부호어
E	0.14878570	3 001	3 000	3 010	3 010	3 000 (Z ₁)		
T	0.09354149	3 110	3 111	3 101	3 101	3 111		
A	0.08833733	4 0000	4 0110	4 0110	4 0110	3 010		
O	0.07245796	4 0100	4 1001	4 1001	4 1001	3 101		
R	0.06872164	4 0101	5 00100	4 0000	4 0000	4 0110		
N	0.06498532	4 0110	5 11011	4 1111	4 1111	4 1001		
H	0.05831331	4 1000	5 01010	5 01110	5 01110	5 00100		
I	0.05644515	4 1001	5 10101	5 10001	5 10001	5 11011		
S	0.05537763	4 1010	5 01110	5 00100	5 00100	5 01110		
D	0.04376834	5 00010	5 10001	5 11011	5 11011	5 10001		
L	0.04123298	5 00011	6 001100	6 011110	6 011110	6 001100		
U	0.02762209	5 10110	6 110011	6 100001	6 100001	6 110011		
P	0.02575393	5 10111	6 010010	6 001100	6 001100	6 011110		
F	0.02455297	5 11100	6 101101	6 110011	6 110011	6 100001		
M	0.02361889	5 11110	6 011110	7 0111110	7 0111110	7 0010100		
C	0.02081665	5 11111	6 100001	7 1000001	7 1000001	7 1101011		
W	0.01868161	6 011100	7 0010100	7 0010100	7 0010100	7 0011100		
G	0.01521216	6 011101	7 1101011	7 1101011	7 1101011	7 1100011		
Y	0.01521216	6 011110	7 0011100	7 0011100	7 0011100	7 0111110		
B	0.01267680	6 011111	7 1100011	7 1100011	7 1100011	7 1000001		
V	0.01160928	6 111011	7 0100010	7 0001000	7 0001000	8 00111100		
K	0.00867360	7 1110100	7 1011101	7 1110111	7 1110111	8 11000011		
X	0.00146784	8 11101011	8 00111100	8 01111110	8 01111110	8 01111110		
J	0.00080064	9 111010101	9 001010100	9 011111110	9 011111110	8 10000001		
Q	0.00080064	10 1110101000	10 0010110100	10 0111111110	9 011111110	9 011111110		
Z	0.00053376	10 1110101001	10 1101001011	10 1000000001	9 100000001	9 100000001		
평균부호길이	4.15572392		4.69655649	4.60728507		4.46463681		

위에 보인 표 2에서, C1은 허프만 부호이고, C2는 Takishima의 알고리즘으로 생성된 부호이다. Tsai는 부호를 좀더 신중히 선택하는 방식으로 C3와 같이 평균 부호 길이가 1.9%정도 줄어든 부호를 얻었다. C2와 C3의 결과를 비교해 보면, 레벨 8과 레벨 9에서 사용 가능한 대칭적 RVLC가 더 존재하는 데에도 불구하고, 하나밖에 찾지 못했음을 알 수 있다. 이러한 문제점은 처음에 각 레벨에서의 사용할 수 있는 대칭적 부호어의 수를 계산할 때 계산 과정이 부정확하고, 모든 레벨이 주어진 허프만 부호어에 제한을 받아 발생한다.

표 2의 C4는 제안된 알고리즘으로 생성된 부호로써, C2나 C3에 비해 각각 5.2%, 3.2% 정도 평균 부호

길이가 짧아졌다. 이 결과는 C4가 제안된 알고리즘에 따라 비트 반전의 쌍으로 구성되고, 주어진 허프만 부호어에 따라 Z₃가 적용되어 높은 발생 확률을 갖는 기호에 보다 더 적은 비트의 부호어가 할당되었으며, 어떠한 레벨에서도 대칭적 부호어들을 놓치지 않기 때문에, 제안된 알고리즘을 통해 보다 적은 평균 부호 길이를 갖는 대칭적 RVLC를 얻을 수 있음을 보여준다.

V. 결론

본 논문에서는 허프만 부호에 기초한 대칭적 RVLC를 생성하는 새로운 방법을 제안하였다. 제안된 알고리즘은 기존의 대칭적 RVLC 생성 방법이 가지고 있는 문제점을 비트 반전을 이용한 영역 검색과 Z_L 적용법을 통해 해결하였다. 비트 반전을 이용한 영역 검색은 검색 영역을 절반으로 줄여 복잡도를 줄이고, 대칭적 부호어 검색 및 선택이 쉽도록 한다. Z_L 적용법 역시 매우 단순하여 계산 복잡도를 줄이며, 기호의 확률 분포에 대한 적용 과정이 상당히 우수하여, 평균 부호 길이를 줄이는 데에 크게 기여한다. 이러한 제안된 알고리즘의 우수함을 여러 발생 확률 분포를 갖는 기호에 대한 실험을 통하여 확인할 수 있었다.

감사의 글

본 연구는 광주과학기술원 (K-JIST) 초고속광네트워크 연구센터 (UFON)를 통한 한국과학재단 우수연구센터 (ERC)와 교육부 두뇌한국21 (BK21) 정보기술사업단의 지원에 의한 것입니다.

참고 문헌

- [1] D. Huffman, "A method for the construction of minimum redundancy codes," in Proc. Inst. Radio Eng., vol. 40, pp. 1098-1101, Sept. 1952.
- [2] Y. Takishima, M. Wada and H. Murakami, "Reversible variable length codes," IEEE Trans. Commun., vol. 43, pp.158-162, Feb. 1995.
- [3] 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.
- [4] ITU-T Rec. H.263, "Video coding for low bit rate communications," Annex V, 2000.
- [5] ISO/IEC 14496-2, "Coding of audio-visual objects: Visual," Final Draft Int. Std, Oct. 1998.