SISO 시스템을 위한 오류에 강인한 가변길이 부호의 효율적 생성 방법

윤 영 석°, 정 욱 현, 호 요 성
광주과학기술원 정보통신공학과

Design of Robust Variable-Length Codes for a Soft-Input Soft-Output System

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

요 약

가변길이 부호는 정보원(source)의 통계적 특성을 이용하여 부호화 효율을 높일 수 있으나, 이동통신과 같이 잡음이 심한 전송환경에서 비트 오류가 발생할 경우 데이터를 제대로 복호할 수 없는 단점이 있다. 본 논문에서는 가변길이 부호에 SISO(soft-input soft-output) 시스템, 자유거리, Huffman 부호, 평균부호길이 함수의 특성을 고려하여 비트 오류에 강인하고 부호화 효율을 증가시킨 새로운 가변길이 부호의 설계 방법을 제안한다. 제안된 방법은 각 부호어에 대해 최소 오류거리를 갖는 경로를 찾아 복호하는 Soft 복호를 사용하여 시스템 자체의 오류 정정(error correction) 능력을 향상시키고, 자유거리를 증가시켜 전송오류에 강인한 가변길이 부호를 생성한다. 또한, Huffman 부호를 기반으로 하는 특성들을 적용하고 가변길이 부호의 성능을 비교하는 평균부호길이 함수를 이용하여 최적의 가변길이 부호를 생성한다.

1. 서 론

최근 각종 유무선 네크워크 상에서 주문형 비디오(VOD)나 스트리밍과 같은 멀티미디어 서비스를 위해 다양한 동영상 부호화 표준들이 사용되고 있다. 이런 표준에서는 전송 효율을 높이기 위해 Huffman 부호나 산술 부호와 같은 가변길이 부호(variable-length code, VLC)를 사용하고 있지만, 부호화된 비트열에 비트 오류가 발생한 경우 재동기를 잃어버려 데이터를 제대로 복호할 수 없다.

VLC로 부호화된 비트열은 전송 중 잡음에 의해 손상된다. Hard 복호는 최대 유사도(Maximum Likelihood) 기법 등을 이용해 잡음에 의해 손상된 비트열의 각 비트를 결정하고, 이 비트열을 look-up 테이블을 통해 복호하는 방법이다. 반면에, Soft 복호[1]는 상태천이 확률을 기반으로 최소 오류거리를 가지는 경로를 검색하여 복호하는 방법이다. 따라서, Soft 복호는 Hard 복호보다 비트 오류에 상대적으로 강인하며, Soft 복호를 이용하는 SISO 시스템은 기존의 시스템보다 향상된 시스템 자체의 오류 정정 능력을 제공한다.

자유거리(free distance)는 전송 오류에 대하여 이진 부호의 강인성을 나타내는 척도로 사용된다. 그러므로, 통신 시스템에서 사용되는 부호의 자유거리를 증가시킬수록 동일한 환경에서 자유거리에 따른 부호 자체의 용장성(redundancy)으로 인해 오류에 강인한 특성을 나타낸다.

하지만, 즉각적인(instantaneous) 복호를 위한 어두(prefix) 조건을 적용시키는 동시에 오류에 강인하도록 자유거리를 증가시키는 것은 강인한 VLC의 부호화 성능을 저하시킨다. 그러므로, 본 논문에서는 강인한 VLC 설계의 효율성을 증가시키기 위해서 평균부호길이 함수와 더불어 Huffman 부호의 특성을 최대한 이용한다. 즉, 주어진 Huffman 부호를 통해서 최소 비트 길이를 이용하는 Z_L 적응법과 Huffman 부호의 부호어 할당 상태를 강인한 VLC 설계에 반영하는 레벨 스킵 적응법[2]을 적용한다.

본 논문에서는 강인한 VLC의 설계를 위해서 SISO 시스템을 기반으로 자유거리 조건을 이용하고 효율적인 VLC 설계를 위해서 Huffman 부호로부터 얻을 수 있는 많은 정보들과 평균부호길이 함수를 이용한다.

2. SISO 시스템

VLC를 사용하는 대부분의 응용 시스템에서는 최대 유사도 기법 등으로 결정된 비트열의 각 비트만을 고려하는 Hard 복호기를 사용한다. 그렇지만, 전송

© JCCI 2004

SISO 시스템을 위한 오류에 강인한 가변길이 부호의 효율적 생성 방법

윤 영 석°, 정 욱 현, 호 요 성

광주과학기술원 정보통신공학과

Design of Robust Variable-Length Codes for a Soft-Input Soft-Output System

Young-Suk Yoon°, Wook-Hyun Jeong, Yo-Sung Ho

Kwangju Institute of Science and Technology (K-JIST)

E-mail : {ysyoon, whjeong, hoyo}@kjist.ac.kr

요 약

가변길이 부호는 정보원(source)의 통계적 특성을 이용하여 부호화 효율을 높일 수 있으나, 이동통신과 같이 잡음이 심한 전송환경에서 비트 오류가 발생할 경우 데이터를 제대로 복호할 수 없는 단점이 있다. 본 논문에서는 가변길이 부호에 SISO(soft-input soft-output) 시스템, 자유거리, Huffman 부호, 평균부호길이 함수의 특성을 고려하여 비트 오류에 강인하고 부호화 효율을 증가시킨 새로운 가변길이 부호의 설계 방법을 제안한다. 제안된 방법은 각 부호어에 대해 최소 오류거리를 갖는 경로를 찾아 복호하는 Soft 복호를 사용하여 시스템 자체의 오류 정정(error correction) 능력을 향상시키고, 자유거리를 증가시켜 전송오류에 강인한 가변길이 부호를 생성한다. 또한, Huffman 부호를 기반으로 하는 특성들을 적용하고 가변길이 부호의 성능을 비교하는 평균부호길이 함수를 이용하여 최적의 가변길이 부호를 생성한다.

1. 서 론

최근 각종 유무선 네크워크 상에서 주문형 비디오(VOD)나 스트리밍과 같은 멀티미디어 서비스를 위해 다양한 동영상 부호화 표준들이 사용되고 있다. 이런 표준에서는 전송 효율을 높이기 위해 Huffman 부호나 산술 부호와 같은 가변길이 부호(variable-length code, VLC)를 사용하고 있지만, 부호화된 비트열에 비트 오류가 발생한 경우 재동기를 잃어버려 데이터를 제대로 복호할 수 없다.

VLC로 부호화된 비트열은 전송 중 잡음에 의해 손상된다. Hard 복호는 최대 유사도(Maximum Likelihood) 기법 등을 이용해 잡음에 의해 손상된 비트열의 각 비트를 결정하고, 이 비트열을 look-up 테이블을 통해 복호하는 방법이다. 반면에, Soft 복호[1]는 상태천이 확률을 기반으로 최소 오류거리를 가지는 경로를 검색하여 복호하는 방법이다. 따라서, Soft 복호는 Hard 복호보다 비트 오류에 상대적으로 강인하며, Soft 복호를 이용하는 SISO 시스템은 기존의 시스템보다 향상된 시스템 자체의 오류 정정 능력을 제공한다.

자유거리(free distance)는 전송 오류에 대하여 이진 부호의 강인성을 나타내는 척도로 사용된다. 그러므로, 통신 시스템에서 사용되는 부호의 자유거리를 증가시킬수록 동일한 환경에서 자유거리에 따른 부호 자체의 용장성(redundancy)으로 인해 오류에 강인한 특성을 나타낸다.

하지만, 즉각적인(instantaneous) 복호를 위한 어두(prefix) 조건을 적용시키는 동시에 오류에 강인하도록 자유거리를 증가시키는 것은 강인한 VLC의 부호화 성능을 저하시킨다. 그러므로, 본 논문에서는 강인한 VLC 설계의 효율성을 증가시키기 위해서 평균부호길이 함수와 더불어 Huffman 부호의 특성을 최대한 이용한다. 즉, 주어진 Huffman 부호를 통해서 최소 비트 길이를 이용하는 Z_L 적응법과 Huffman 부호의 부호어 할당 상태를 강인한 VLC 설계에 반영하는 레벨 스킵 적응법[2]을 적용한다.

본 논문에서는 강인한 VLC의 설계를 위해서 SISO 시스템을 기반으로 자유거리 조건을 이용하고 효율적인 VLC 설계를 위해서 Huffman 부호로부터 얻을 수 있는 많은 정보들과 평균부호길이 함수를 이용한다.

2. SISO 시스템

VLC를 사용하는 대부분의 응용 시스템에서는 최대 유사도 기법 등으로 결정된 비트열의 각 비트만을 고려하는 Hard 복호기를 사용한다. 그렇지만, 전송

잡음이 발생한 경우 결정되지 않은 각 비트의 손상된 신호(soft information)를 이용한 Soft 복호를 하면 비트 오류를 줄일 수 있다. Soft 복호는 Hard 복호와 달리 상태 천이 확률을 기반으로 복호된 부호어의 수와 심볼의 수, 수신된 비트 수와 복호된 비트 수가 각각 같으며 복호가 가능한 모든 경로에 대해 최소 오류거리를 갖는 경로로 복호하는 방법이다.

Fossorier은 soft-output Viterbi algorithm(SOVA)와 같은 SISO 복호 알고리즘을 제안하였다. 그리고, Wen은 손상된 신호를 이용하여 Soft 복호를 하는 SISO 시스템을 제안하였다. Wen이 제안한 SISO 시스템은 기존의 VLC 부호기와 SISO VLC 복호기로 구성된다. 주어진 심볼들은 기존의 VLC 부호기로 부호화된 후 전송되며 도중에 잡음이 더해진다. 그리고, 수신된 비트열은 SISO VLC 복호기에 의해 Soft 복호된다. 기존의 시스템보다 SISO 시스템은 오류 정정 능력 측면에서 뛰어난 성능을 보이므로, 부호어에 대해 오류에 강인한 특성을 부여한다. 또한, Buttigieg[3]는 Soft 복호는 많은 시간 및 자원을 요구하므로 순차적 복호(Sequential Decoding)를 사용하여 실시간 처리가 가능하도록 했다. 본 논문에서는 시스템 자체의 오류 강인성과 더불어 자유거리를 증가시켜 부호어의 오류 강인성도 고려한다.

3. VLC 의 자유거리

가변길이 부호어 c_i 의 비트 길이를 $|c_i| = l_i$ 로 나타내면, $H_d(x,y)$ 는 $|x| = |y|$ 인 이진 부호어 x, y의 Hamming 거리이다. 또한, 집합 $G_N = \{c_i : |c_i| = N\}$ 은 주어진 VLC 집합을 기반으로 비트 길이가 N 인 생성 가능한 모든 이진 비트열로 구성된다.

VLC의 자유거리 d_{free} 는 식 (1)과 같이 서로 다른 두 개의 이진 비트열 사이의 최소 Hamming 거리로 정의한다.

$$d_{free} = \min\left\{H_d(c_i,c_j) : c_i, c_j \in G_N, i \neq j\right\} \quad (1)$$

Laković은 VLC의 d_{free} 를 직접 유도하는 것은 어렵기 때문에, VLC의 자유거리를 최소 블록거리, d_{Block} 를 이용해 표현했다. d_{Block} 은 주어진 VLC 집합에서 동일한 길이를 갖는 서로 다른 두 부호어간의 최소 Hamming 거리를 나타낸다.

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

하지만, 주어진 VLC의 집합에서 비트 길이가 l_k 인 부호어 c_k 가 하나일 경우 d_{Block} 은 정의되지 않으므로 d_{free} 는 식 (3)으로 정의한다.

$$d_{free} = \begin{cases} 2, & d_{Block}\text{이 정의되지 않은 경우} \\ d_{Block}, & d_{Block}\text{이 정의되는 경우} \end{cases} \quad (3)$$

Huffman 부호를 포함한 Huffman 부호에 기반한 모든 VLC의 자유거리 d_{free} 는 1이다. 제안된 알고리즘에서는 오류에 강인하도록 하기 위해서 자유거리 조건인 VLC의 d_{free} 를 2로 증가시킨다.

4. Huffman 부호의 특징

Huffman 부호는 발생확률이 가장 낮은 심볼에 먼저 부호어를 할당하는 상향 방식으로 최소 평균부호길이를 갖는 최적의 부호로 생성된다. 반면에, 제안된 강인한 VLC 생성 방식은 하향 방식으로 진행된다. 즉, 가장 발생확률이 높은 심볼부터 짧은 길이의 부호어를 할당하며, 전체 심볼의 개수 S 까지 부호어를 생성하는 방식이다. 그러므로, 강인한 VLC를 효율적으로 생성하기 위해서는 주어진 정보원에 대해 모든 발생 가능한 경우의 VLC를 생성하고 이 중에서 최소 평균부호길이를 갖는 VLC를 선택하면 된다. 하지만, 발생할 수 있는 경우의 수는 무한하므로 최적의 효율성을 갖는 강인한 VLC를 설계하는 것은 쉽지 않다.

강인한 VLC 생성의 효율성을 증대시키는 방식은 VLC의 최소 비트 길이 레벨, 각 비트 길이에 할당된 부호어와 그 수의 선택에 좌우된다. 따라서, Huffman 부호의 이러한 특성을 이용하면 최적의 부호를 효율적으로 생성하는데 크게 기여한다.

최적화된 Huffman 부호로부터 발생확률이 가장 높은 심볼에 최소 비트 길이 L_{min} 이 할당되고, 이것은 심볼의 개수와 심볼의 확률분포 등에 영향을 받는다. 강인한 VLC는 Huffman 부호와 유사하기 때문에 Huffman 부호의 L_{min} 을 이용한다. 따라서, L_{min} 의 선택은 주어진 심볼에 대해 강인한 VLC 생성의 효율성을 증가시킨다.

Z_L 은 모두 '0' 비트만으로 구성된 비트 길이가 L 인 부호어로 정의된다. Z_L 은 전체 VLC에서 반드시 한 번은 할당되는 부호어 이므로 최소 비트 길이 레벨인 L_{min} 에 할당하는 Z_L 적응법을 사용하여 효율성을 개선하도록 한다.

비트 길이 벡터 $n(i)$ 는 비트 길이 i 를 갖는 부호어의 수라고 정의한다. 그렇다면, Huffman 부호와 강인한 VLC에 비트 길이 L 을 갖는 부호어의 수는 각각 $n_{Huff}(L)$, $n_{SVLC}(L)$ 으로 나타낼 수 있다. 상대적으로 하위 레벨의 부호어 수, $n_{SVLC}(L)$ 은 어두 조건과 자유거리 조건에 제약을 받아 $n_{Huff}(L)$ 보다 작으므로 L_{min} 에서는 하한(lower-bound)이 $n_{Huff}(L_{min})$ 이어야만 부호어 할당에 효율성을 높일 수 있다. 그리고, 자유거리 조건 때문에 이항 정리를 이용하여 정리하면 상한(upper-bound)은 $2^{L_{min}-1}$ 이 된다.[2] 따라서, 강인한 VLC의 최소 비트 길이 L_{min} 에서 선택되는 부호어의 수, $n_{SVLC}(L_{min})$ 은 식 (4)와 같은 범위를 갖는다.

$$n_{Huff}(L_{min}) \le n_{SVLC}(L_{min}) \le 2^{L_{min}-1} \quad (4)$$

그런데, 주어진 심볼들의 발생확률분포에 따른 Huffman 부호의 구성 방식을 보면 확률분포가 Laplacian과 같은 편향(skewed) 분포일 때, Huffman 부호에서 특정 비트 길이가 스킵되어 하위 레벨에 많은 부호어를 생성하도록 보장함을 알 수 있다.

효율적인 부호어를 설계하기 위해서 짧은 비트 길이인 상위 레벨에서는 Huffman 부호보다 많은 부호어를 할당하는 동시에 긴 비트 길이인 하위 레벨에서는 많은 부호어가 할당되도록 보장해야 하므로 레벨 스킵 적응법을 사용한다. 여기서, 상위 레벨이란 $\lceil S/2 \rceil$의 부호어가 할당되어 있는 비트 길이 레벨로, 하위 레벨은 이를 제외한 나머지 레벨로 정의한다. 또한, $\lceil i \rceil$는 i보다 크거나 같은 가장 작은 정수로 정의한다.

레벨 스킵 적응법[2]은 Huffman 부호에서 스킵된 레벨 L_{Skip}이 존재할 경우 상위 레벨에서는 강인한 VLC 생성시 동일하게 레벨을 스킵한다. 그리고, 하위 레벨에서는 어두 조건과 자유거리 조건만을 만족하는 부호어를 선택한다.

5. 평균부호길이 함수

VLC의 성능을 비교하기 위해 평균부호길이 함수[2]를 이용한다. 특정 비트 길이 L에 존재하는 부호어 수가 $n_{SVLC}(L) = M$ 이고 L 보다 짧은 비트 길이에 할당된 부호어가 같은 강인한 VLC의 집합을 $S(L;M)$으로 정의한다. 그러면, 평균부호길이 함수 $f_{\bar{L}}(x)$는 식 (5)와 같이 $n_{SVLC}(L)$인 M을 정의역, 최소 평균부호길이인 $\min(\bar{L}(S))$을 치역으로 하는 함수이다.

$$f_{\bar{L}} : n_{SVLC}(L) \rightarrow \min(\bar{L}(S)) \qquad (5)$$

$n_{SVLC}(L_{min})$를 결정한 후 $L_{min} < L_i$인 경우 레벨 스킵이 일어나지 않는 각 비트 길이의 부호어 수 $n_{SVLC}(L_i)$를 선택하는 알고리즘은 다음과 같다.

1) 이전 비트 길이 $L_{Curr}-1$까지 선택된 강인한 VLC에 의해 현재 L_{Curr}에서 사용 가능한 부호어가 결정된다. 아직 $d_{free} \geq 1$이다.
2) 집합 $M_L(D)$는 비트 길이 L을 갖는 부호어 중에서 자유거리 D를 갖는 부호어들의 집합인 원소 m^i를 갖는 집합으로 정의한다. 예를 들어 $M_2(2) = \{\{00,11\},\{01,10\}\} = \{m^1, m^2\}$ 으로 나타낼 수 있다. 물론 이전에 짧은 길이의 부호어가 선택된 경우 존재하지 않는 원소들은 제거된다. $N(S)$를 집합 S의 원소의 수로 정의한다. $N(m^i)$ $(1 \leq i \leq N(M_L(D)))$를 비교하여 가장 큰 집합의 부호어를 선택한다. 선택된 부호어의 집합을 m^s라 하면 선택되지 않은 다른 집합에서 자유거리 D 이상인 부호어들을 모두 선택한다. 선택된 부호어의 수를 $n_{SVLC}(L_{Curr})$로 한다. 제안된 알고리즘에서는 $D=2$인 경우를 고려한다.
3) $0 \leq l < n_{SVLC}(L_{Curr})$ 일 때, $n_{SVLC}(L_{Curr}) - l$ 에서 $f_{\bar{L}}(x)$를 구하고 최소값이 존재하면 $n_{SVLC}(L_{Curr})$를 $n_{SVLC}(L_{Curr}) = n_{SVLC}(L_{Curr}) - l$으로 갱신한다. l은 0에서 하나씩 증가된다. 하나의 l에 대한 $N(R)$은 $\begin{pmatrix} n_{SVLC}(L_{Curr}) \\ n_{SVLC}(L_{Curr}) - l \end{pmatrix}$이 된다.
4) 현재의 $f_{\bar{L}}(x)$ 값이 이전의 값보다 크다면 L_{Curr}에서 선택되는 부호어와 그 수는 이전에 선택된 부호어의 집합과 $n_{SVLC}(L_{Curr})$이다.

본 논문에서 제안된 방법을 정리하면 그림 1처럼 나타낼 수 있다.

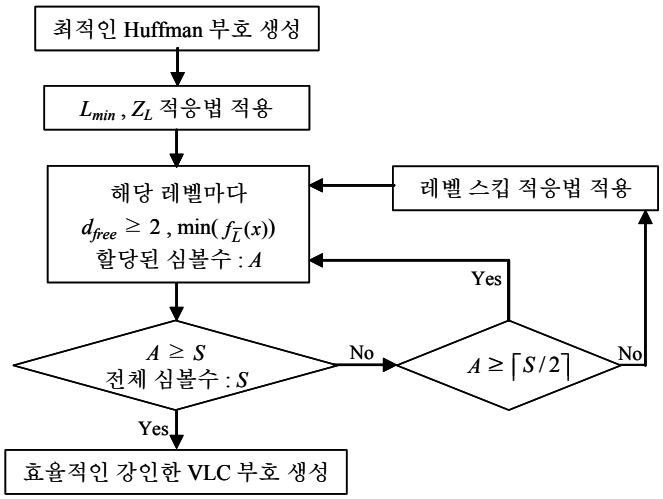

그림 1. 강인한 VLC 의 효율적 생성 알고리즘

6. 실험 결과

표 1은 Huffman 부호와 제안된 강인한 VLC 부호를 각각 압축 알고리즘에 대한 성능 평가를 위해 사용되는 Canterbury Corpus 파일(http://corpus.canterbury.ac.nz)들에 대하여 생성된 부호와 평균부호길이를 보여준다. 표 1에서 음영으로 표시된 F2, F4, F8, F9, F10, F11에서는 Huffman 부호 생성에서 레벨 스킵이 일어났으므로 레벨 스킵 적응법이 사용되었다. 강인한 VLC 생성시 레벨 스킵 적응법이 효율성을 증대시키는 효과가 있음을 알 수 있다.

표 2는 Huffman 부호와 제안된 강인한 VLC 부호를 영어 알파벳에 대하여 생성된 부호와 평균부호길이를 보여준다. 영어 알파벳의 경우에는 레벨 스킵 적응법이 사용되지 않았고 Huffman 부호의 최소 비트 길이 레벨인 L_{min}에 의해서 Z_3이 사용되었다. 또한, $n_{Huff}(L_{min}) < n_{SVLC}(L_{min})$을 만족하는 것은 상위 레벨에 보다 많은 부호어를 할당하는 동시에 하위 레벨에도 많은 부호어가 할당되도록 해야 부호 생성의 효율성을 증가시킴을 보여준다.

표 1. 다양한 정보원 파일에 대한 Huffman 부호와 제안된 방법으로 생성된 강인한 VLC의 평균부호길이 비교 (: 레벨 스킵 적응법 적용)

Canterbury Corpus 파일		부호어 (심볼) 수	Huffman 부호	제안된 방법 ($d_{free} \geq 2$)
			평균 부호길이	평균 부호길이
F1	asyoulik.txt	68	4.84465	**4.850708**
F2	alice29.txt	74	4.61244	**4.618620**
F3	xargs.1	74	4.92382	**4.926554**
F4	grammar.lsp	76	4.66434	**4.672788**
F5	plabn12.txt	81	4.57534	**4.584056**
F6	lcet10.txt	84	4.69712	**4.710620**
F7	cp.html	86	5.26716	**5.278077**
F8	fields.c	90	5.04090	**5.065212**
F9	Ptt5	159	1.66091	**1.661944**
F10	Sum	255	5.36504	**5.374732**
F11	kennedy.xls	256	3.59337	**3.630420**

표 2. 영어 알파벳에 대해 Huffman부호와 강인한 VLC의 부호어 할당과 평균부호길이 비교

발생확률		Huffman 부호		제안된 방법 ($d_{free} \geq 2$)	
		L	부호어	L	부호어
E	0.14878570	3	001	3	000 (Z_3)
T	0.09354149	3	110	3	011
A	0.08833733	4	0000	3	101
O	0.07245796	4	0100	4	0010
R	0.06872164	4	0101	4	0100
N	0.06498532	4	0110	4	1000
H	0.05831331	4	1000	4	1101
I	0.05644515	4	1001	4	1110
S	0.05537763	4	1010	5	00110
D	0.04376834	5	00010	5	01010
L	0.04123298	5	00011	5	10010
U	0.02762209	5	10110	5	11000
P	0.02575393	5	10111	5	11110
F	0.02455297	5	11100	6	001110
M	0.02361889	5	11110	6	010110
C	0.02081665	5	11111	6	100110
W	0.01868161	6	011100	6	110010
G	0.01521216	6	011101	6	111110
Y	0.01521216	6	011110	7	0011110
B	0.01267680	6	011111	7	0101110
V	0.01160928	6	111011	7	1001110
K	0.00867360	7	1110100	7	1100110
X	0.00146784	8	11101011	7	1111110
J	0.00080064	9	111010101	8	00111110
Q	0.00080064	10	1110101000	8	01011110
Z	0.00053376	10	1110101001	8	10011110
평균부호길이		4.15572392		**4.226314**	

그림 2는 각 경우에서 발생된 비트 오류율에 대한 문자 오류율을 통해 제안된 VLC가 얼마나 오류에 강인한지를 보여준다. 임의의 문자열을 입력으로 사용했고, 복호는 순차적 복호와 재동기점을 이용하여 복잡도를 최소화했다. 그림 2로부터 강인한 VLC

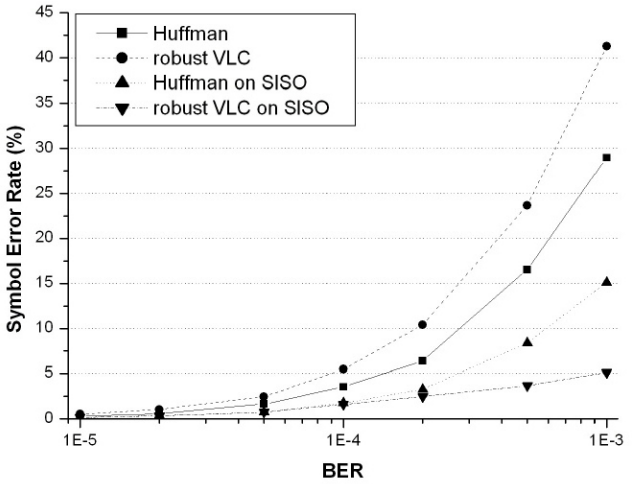

그림 2. 비트 오류율에 대한 문자 오류율의 비교

가 문자당 오류율이 최소로 나타남을 알 수 있다.

7. 결 론

본 논문에서는 자유거리를 증가시켜 강인한 VLC를 효율적으로 생성하는 알고리즘을 제안하였다. 강인한 VLC는 어두 조건과 자유거리 조건을 만족하므로 부호화 효율이 저하될 가능성이 매우 높다. 하지만, 효율성을 보장하기 위해서 평균부호길이 함수와 Huffman 부호를 기반으로 레벨 스킵 적응법과 Z_L 적응법을 이용하였다. 평균부호길이 함수는 강인한 VLC의 각 비트 레벨에서 부호의 수를 조절하여 부호화 성능을 최적화시킨다. 레벨 스킵 적응법은 Huffman 부호의 부호어 할당 상태를 부호 생성에 반영하고 Z_L 적응법은 Huffman 부호의 최소 비트길이 L_{min} 에 Z_L 을 할당해서 Huffman 부호와 유사한 최적의 부호를 생성하도록 한다. 실험 결과를 통해서 제안된 방법으로 강인한 VLC를 효율적으로 생성할 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.

참고 문헌

[1] J.Wen and J.Villasenor, "Utilizing soft information in decoding variable length codes," in *Proc. IEEE Data Compression Conf.*, pp. 131–139, March 1999.

[2] 정욱현, 윤영석, 호요성, "자유거리 특성을 고려한 강인한 양방향 가변길이 부호의 생성방법," *방송공학회 학술대회 논문집*, pp. 133-138, 2003. 11.

[3] V. Buttigieg and P. G. Farrell, "Sequential Decoding of Variable-Length Error-Correcting Codes," *Proc. Eurocode 94*, pp. 93-98, Oct. 1994