

논문 2006-43SP-3-2

# 무선 통신상 임의 에러에 대한 에러내성 엔트로피 부호화에 기반한 정지영상의 화질 개선

( An Improvement of Still Image Quality Based on Error Resilient  
Entropy Coding for Random Error over Wireless Communications )

김 정 식\*, 이 근 영\*\*

( Jeong-Sig Kim and Keun-Young Lee )

## 요 약

많은 영상과 비디오 압축 알고리즘들은 영상을 블록으로 나누어 처리하여 각 블록에서 가변길이 부호비트를 생성한다. 만일 에러 검출기법을 사용하지 않고 가변길이 부호데이터를 에러 발생채널에 전송한다면 수신측 복호화는 압축된 스트림(Stream)을 적절히 복호할 수 없다. 따라서 표준 영상 및 비디오 압축 알고리즘에서는 채널 에러로부터 데이터 스트림을 보호하기 위해 추가적인 정보들을 삽입한다. 그런 추가 정보 중의 하나가 재동기 마커(resynchronization marker)이다. 이 방법은 전송 에러 발생시 복호화를 다시 시작하기 위한 위치를 복호화기에게 알려줄 수 있지만 주파수 대역폭의 낭비가 심한 단점이 있다. 에러 내성 엔트로피 부호화(EREC)는 어떤 추가 정보 없이 재동기 시작점을 찾을 수 있는 방법으로 잘 알려져 있다. 이 방법은 대부분의 영상 압축 기법에서 사용되는 접두 코드(prefix code)에 적용될 수 있으므로 본 논문에서는 FEREC(Fast Error-Resilient Entropy Coding)의 성능을 개선한 EREREC(Efficient and Robust EREC) 기법을 제안하였다. 첫째로 연속 블록들의 부호화비트 길이를 이용하여 초기 탐색 위치를 계산한다. 둘째, 초기 오프셋은 가변 길이 부호들에서 길고 짧은 블록들의 확률 분포를 이용하여 결정되고, 결정된 초기 오프셋 값은 제안 방법에서 사용되는 모든 오프셋 시퀀스 값들을 보장하기 위해 조정된다. 제안된 EREREC 알고리즘은 슬롯 구성에 있어 EREC보다 빠르며, 전송 에러 발생시 복호화된 영상의 화질이 개선된다. 실험 결과는, 임의 에러 발생 채널에서 기존의 EREC 및 FEREC와 복원영상의 화질을 비교하였을 때 약 0.3 ~ 3.5 dB의 화질이 개선됨을 보여준다.

## Abstract

Many image and video compression algorithms work by splitting the image into blocks and producing variable-length code bits for each block data. If variable-length code data are transmitted consecutively over error-prone channel without any error protection technique, the receiving decoder cannot decode the stream properly. So the standard image and video compression algorithms insert some redundant information into the stream to provide some protection against channel errors. One of redundancies is resynchronization marker, which enables the decoder to restart the decoding process from a known state in the event of transmission errors, but its usage should be restricted not to consume bandwidth too much. The Error Resilient Entropy Code(EREC) is well known method which can regain synchronization without any redundant information. It can work with the overall prefix codes, which many image compression methods use. This paper proposes EREREC method to improve FEREC(Fast Error-Resilient Entropy Coding). It first calculates initial searching position according to bit lengths of consecutive blocks. Second, initial offset is decided using statistical distribution of long and short blocks, and initial offset can be adjusted to insure all offset sequence values can be used. The proposed EREREC algorithm can speed up the construction of FEREC slots, and can improve the compressed image quality in the event of transmission errors. The simulation result shows that the quality of transmitted image is enhanced about 0.3 ~ 3.5 dB compared with the existing FEREC when random channel error happens.

**Keywords :** EREC(Error Resilient Entropy Coding), JPEG, Channel Error, Video Compression, VLC(Variable-Length Coding)

\* 정회원, \*\* 평생회원, 성균관대학교 전기전자 및 컴퓨터공학부  
(School of Electrical and Computer Engineering, SungKyunKwan University)  
접수일자: 2005년6월30일, 수정완료일: 2006년4월17일

## I. 서 론

최근 몇 년 동안 멀티미디어 통신 서비스를 위한 요구가 크게 증가 되었다. 하지만 멀티미디어 서비스의 실행에 있어서 두 가지 중요한 문제점이 존재한다. 첫째는 무선 및 이동 채널의 제한된 용량이다. 이는 가변길이 부호화(VLC : Variable-Length Coding)를 이용하여 신호의 필요 대역폭을 줄여 해결할 수 있다. 둘째는 전송환경에 의한 신호의 페이딩(fading)이나 다중 채널로 인한 간섭(interference)으로 인하여 전송 에러의 발생이다. 대부분의 소스 부호화 기법들은 에러가 없는 채널 환경을 대상으로 설계되었으므로 압축된 데이터에 에러가 발생하면 복호화에 문제가 발생한다. 따라서 무선, 이동 통신 및 인터넷 통신과 같은 에러 발생이 빈번한 채널들을 통한 압축 영상과 비디오 전송에 대한 연구가 활발하게 진행되고 있다.

JPEG, MPEG 시리즈와 H.261, H.263 등의 많은 영상 및 비디오 압축 기법들은 입력 영상을 고정 블록(8×8 블록) 데이터로 나누어 처리한다. 이들 각 블록들은 가변 길이 부호를 생성하기 위해 DCT(Discrete Cosine Transform), 양자화, 지그재그 스캔, 줄-길이 및 허프만 부호를 사용하여 부호화된다. 부호화된 비트열에 발생하는 에러는 부호화기와 복호화기의 동작에 불일치를 발생시키므로 복호화된 영상의 화질에 커다란 영향을 준다. 이를 해결하기 위하여 LC(Layered Coding)<sup>[1]</sup>, FEC(Forward Error Correction)<sup>[2]</sup>, ARQ(Automatic Repeat Request), 에러 검출, 에러 은닉<sup>[3]</sup>, 에러 내성 부호<sup>[4, 5, 6]</sup> 등의 많은 전송에러 제어 기법들이 제안 되었다.

하지만 FEC 등의 기존의 에러 검출 및 정정 방법들은 재동기 부호를 삽입함으로써 높은 압축률을 필요로 하는 디지털 비디오의 특성에 적합하지 않다. ARQ는 에러가 발생하거나 수신측에 도달하지 못한 데이터를 수신측으로 재전송하는 기법이지만, 수신측이 송신측으로 에러 발생 신호를 전송하기 위한 백-채널을 가지지 않는 브로드캐스트 비디오 전송에는 적합하지 못하다. 또한 재전송 데이터가 수신측에 도달하기까지의 지연 시간으로 인한 영상의 디스플레이 지연이 발생한다. 에러 내성 부호화(ERC: Error Resilient Coding)는 채널 부호화에 의한 데이터의 용장성을 감소시킬 수 있으며 영상 및 비디오 전송에 발생하는 에러의 영향을 줄이기 위한 기법으로 제안되었다.<sup>[4, 5, 6]</sup> 논문 [4]에서는 활성 블록들의 위치와 값들을 부호화하였으며, 종료 비트를

이용하여 각 활성 블록들의 끝을 표시하였다. 따라서 전송되는 각 이미지 블록들의 위치를 부호화하기 위한 추가적인 오버헤드를 요구하는 단점이 있다. 이를 해결하기 위하여 제안된 EREC(Error Resilient Entropy Coding) 알고리즘<sup>[5]</sup>은 비트 재구성 기법을 사용하였다.

EREC는 어떤 오버헤드 정보 없이 전송에러의 공간적 전파를 감소시킬 수 있는 특징을 가지고 있다. 블록의 가변길이 비트들을 일정 크기의 고정길이 슬롯에 할당하고, 미리 정의된 오프셋 값(비어있는 슬롯의 탐색 패턴)들을 이용하여 각 블록에 남아있는 가변길이 비트들을 비어 있는 고정슬롯으로 할당한다. 비어있는 슬롯을 탐색하는 과정은 남아있는 가변길이 비트들이 고정 크기 슬롯에 전부 채워질 때까지 반복된다. 결과적으로 남아있는 가변길이 블록 데이터를 고정길이 슬롯에 채우기 위한 탐색 반복 횟수는 비어 있는 고정길이 슬롯을 찾는 속도에 의존하게 된다. EREC 기법은 0의 오프셋 값에서 시작하여 연속적으로 일정하게 증가하도록 미리 결정되어 있는 오프셋 값을 이용하여 가변길이 비트들을 고정길이 슬롯에 채우게 된다. 이런 형태의 오프셋 값들을 사용할 경우 복호화기가 오프셋 값을 유추할 수 있으므로 오프셋 값이 따로 전송될 필요는 없다. 그러나 만일 오프셋 값이 0부터 시작하지 않을 경우 초기 오프셋 값을 따로 복호화기에 알려 주어야 한다. EREC 처리 기법은 JPEG 압축 데이터의 통계적 특성을 조사한 새로운 채널 적응 양자화와 빠른 에러 내성 엔트로피 부호화(FEREC)<sup>[6]</sup>에 의해 향상되었다.

무선이나 이동 통신상의 전송 에러는 임의 에러(random error) 또는 버스트(burst) 에러이다. 버스트 에러에서는 어떤 일정 시간 구간에서 연속적으로 에러가 발생한다. EREC 알고리즘이 사용될 경우, 고정길이 슬롯에 발생하는 임의 에러는 그 이후의 비트가 속한 원래의 가변길이 블록을 찾지 못하게 방해하므로 임의 에러와 버스트 에러는 같은 형태로 다루어질 수 있다.

EREC에서는 에러의 존재 유무와는 상관없이 고정길이 슬롯들은 순차적으로 복호화 될 수 없다. 왜냐하면 모든 가변길이 블록은 복호화하기 전에 몇 개의 고정길이 슬롯에 흩어져 있는 자신의 비트들을 모아야 하기 때문이다. 따라서 부호화기의 슬롯 탐색 횟수가 증가되면 복호화기의 처리 성능은 낮아진다. 결국 탐색 횟수가 부호화기와 복호화기의 처리 성능 및 에러 발생 시의 복원 영상 화질에 영향을 미친다. 이는 하나의 슬롯에 임의 에러가 발생했을 경우에 대한 복호화기의 동작을 살펴보면 명백해진다. 고정길이 슬롯에 임의에러가

발생할 경우 이 슬롯은 에러 슬롯으로 취급되어 에러비트의 뒤에 존재하는 다른 가변길이 블록의 데이터들도 복호화 될 수 없다. 이는 임의의 에러가 들어있는 비트열을 허프만 트리로 복호화하면 블록의 계수가 64를 넘을 수도 있기 때문이다. 따라서 다른 슬롯으로 이동되는 블록 데이터의 수가 감소할수록 에러의 영향을 적게 받으므로 복원 영상의 화질이 개선될 수 있다. 이를 위하여 부호화기에서는 우선 부호화된 영상 블록의 비트들을 초기 슬롯에 할당한 후, 비어 있는 공간이 큰 슬롯을 찾아 남은 비트가 많은 블록의 비트를 채움으로써 복호화 시 해당 슬롯에 에러가 발생하더라도 에러 블록으로 표시되는 부호화된 영상 데이터 블록의 수를 감소시킬 수 있다.

따라서 본 논문에서는 영상 내의 이웃 블록들의 상관관계를 이용하여 FEREC의 성능을 향상시킨 에러내성 엔트로피 부호화 EREREC (Efficient and Robust Error Resilient Entropy Coding) 기법을 제안하였다. 우선 이동할 초기 위치는 연속적으로 가장 긴 블록들과 짧은 블록들의 비트수를 이용하여 계산한다. 그 다음으로 초기 오프셋 값은 연속된 긴 블록들과 짧은 블록들의 평균 개수를 이용하여 선택하고, 선택된 초기 오프셋 값이 모든 가능한 오프셋 값들에 사용될 수 있는지를 결정하여 만약 사용될 수 없다면 사용 가능한 값으로 수정한다. 실험결과 제안된 기법은 블록 재동기 마커와 같은 추가적인 오버헤드 없이 부호화와 복호화의 슬롯 탐색 횟수를 줄일 수 있었으며 복원 영상의 화질도 0.3 ~ 3.5 dB 정도 향상시켰다.

본 논문의 구성은 다음과 같다. II장에서는 EREC 기법<sup>[5,6]</sup>을 간단히 소개한다. III장에서는 본 논문의 제안된 기법을 설명하고, 기존의 EREC 및 FEREC와 제안된 기법의 비교 결과를 IV장에 나타내었다. 마지막 V장에서는 결론을 나타내었다.

## II. 여러 가지 에러 내성 엔트로피 부호화

### 1. 기본 에러 내성 엔트로피 부호화(EREC)

EREC 기법은 JPEG과 같이 입력영상을 블록으로 나누어 처리하여 가변길이 블록 데이터비트를 생성하는 블록코딩 알고리즘에 적용될 수 있다. 각 블록의 가변 길이 블록 데이터 비트들은, 채널 에러가 없을 경우 복호화기가 어떤 정보나 이전 블록의 참조 없이 블록을 복호화 할 수 있는 접두 코드(prefix code)이어야 한다.

EREC 기법 또한 각 가변 길이 블록에 포함된 채널 에러는 현재 데이터와 다음 데이터에만 영향을 주는 것으로 가정한다. 이런 가정은 허프만 부호화나 산술 부호화와 같은 대부분의 엔트로피 기반 소스 부호화에 부합된다.

EREC 기법은 원 블록들의 가변 길이 데이터 비트를 미리 정의된 고정된 길이의 슬롯들 내부에 적절하게 재배치함으로써 재동기 마커와 같은 추가적인 오버헤드 없이 각 블록의 시작부분에서 재동기를 획득하는 기법이다.

EREC 부호화기는 길이  $b_i$ 를 가지는  $M$ 개의 가변 길이 블록 데이터를 고정 길이  $S$ 를 갖는  $M$ 개의 슬롯에 재배치한다. 복호화기가 슬롯의 길이를 안다면 각 블록의 시작 위치를 찾을 수 있으며 그 위치로부터 독립적으로 복호화를 할 수 있다. 부호화기는 수식 (1)을 이용하여 모든 가변 길이 블록 데이터들을 부호화하는데 사용되는 전체 데이터 비트들의 길이  $L_{all}$  값을 계산한다. 오버헤드에 해당하는 이  $L_{all}$  값은 복호화기의 동작에 커다란 영향을 미치므로 전송 시에 에러가 발생하지 않도록 특별히 보호되어야 한다.

$$L_{all} = \sum_{i=1}^M S_i \geq \sum_{i=1}^M b_i \quad (1)$$

$b_i$  : 각 부호화된 블록들의 가변 길이 비트

$M$ 개 블록들에 할당된 가변 길이 비트들의 합은 전체 데이터 비트 길이  $L_{all}$ 을 초과할 수 없다. 또한 슬롯 길이  $S$ 는 수식 (2)와 같이 영상으로부터 부호화된  $M$ 개 블록들의 평균 길이로써 결정된다.

$$S = \left\lceil \frac{1}{M} \left( \sum_{i=1}^M b_i \right) \right\rceil \quad (2)$$

여기서  $\lceil \cdot \rceil$ 는 정수형 값을 만들기 위한 상한 함수(ceiling function)이다.

부호화기는 가변 길이 블록 데이터들을 전체 길이  $L_{all}$ 로부터 근사적으로 계산된 고정길이  $S$ 를 가지는  $M$ 개의 슬롯들로 재배치하는데,  $N_{th}$ -단계 반복 알고리즘을 이용하여 현재 할당되어야 하는 고정 길이 슬롯 번호(즉, 현재 블록번호+오프셋)를 계산하게 된다. 임의의  $N_k$ -단계에서는  $i$ 번째 슬롯에 채워지고 남은 원 블록의 데이터 전체 또는 일부는 현재 탐색되고 있는 빈 공간이 남아있는 슬롯  $j = i + \Phi_k \pmod{M}$ 에 할당된다. 그림 1은 가변 길이 블록 데이터 비트 길이 11, 8, 6, 12,

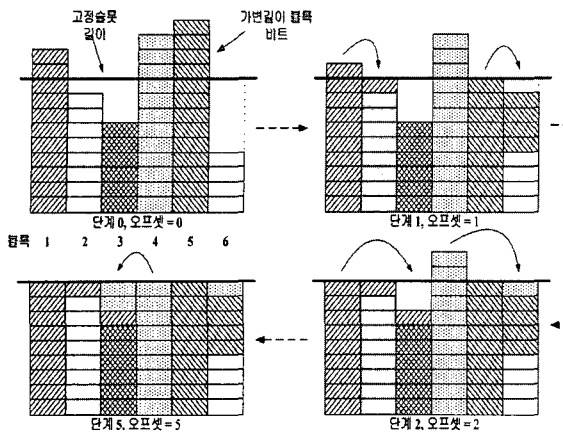


그림 1. EREC 알고리즘의 예  
Fig. 1. The example of EREC algorithm.

13, 4를 가지는 6개의 블록에 대한 알고리즘의 적용 예를 보여준다. 고정 길이 슬롯은 수식 (1)과 (2)에 의해 9가 사용되었고, 연속적으로 사용된 오프셋 시퀀스의 값은 0, 1, 2, 3, 4, 5 이다.

단계 0은 모든 가변 길이 블록들을 자신의 위치에 해당하는 고정 길이 슬롯에 할당하는 초기 단계이다. 오프셋 값 1을 가지는 단계 1에서 2, 3, 6번의 블록들은 자신의 슬롯인 2, 3, 6에 완전하게 할당되고 슬롯은 윗부분에 여분의 공간을 가지고 있다. 따라서 오프셋 값이 1이므로 남은 데이터 비트를 가지고 슬롯에 초기 할당된 블록들은 자신의 오른쪽에 비어 있는 슬롯이 있는지를 찾고, 비어 있으면 자신의 슬롯을 채우고 남은 가변 길이 비트들을 그곳에 채울 수 있는 만큼의 해당 가변 길이 비트를 할당한다. 즉, 블록 1번의 남은 데이터 비트는 슬롯 2번의 남은 공간에 해당 데이터 비트를 부분적으로 할당하고, 블록 5번의 남은 데이터 비트 전체는 슬롯 6번으로 이동되어 완전하게 부호화된다. 오프셋 값이 2인 단계 2에서는 블록 1의 부분적으로 남은 데이터를 다시 슬롯 3에 할당하게 되고, 블록 4의 남은 일부분 데이터 비트는 슬롯 6으로 채워져서 부호화된다. 결국 이런  $N_{th}$ -단계의 반복 알고리즘 과정을 거쳐서 초기 슬롯으로부터 남은 모든 가변 길이 블록 데이터 비트들은 빈 공간이 있는 다른 슬롯으로 이동되어 부호화를 하게 된다. 부호화기는 각 해당 가변 길이 블록의 데이터들이 완전하게 부호화될 때까지 부호화의 역 과정을 반복하게 된다.

2. 빠른 에러 내성 엔트로피 부호화 (FEREC)

2.1절에서 언급된 기본 EREC 방법은 오프셋 값들을 이용하여 슬롯에 영상 비트들을 할당한다. 따라서

EREC의 복잡도는 오프셋 값들의 선택에 크게 의존하게 된다. 또한 EREC의 처리 성능도 비어 있는 슬롯을 찾기 위한 오프셋 시퀀스 값의 순서에 크게 의존한다. 따라서 FEREC<sup>[6]</sup>는 부호화된 가변 길이 블록 비트들의 통계적 특성을 이용하여 EREC의 처리 성능을 개선하기 위해 제안되었다.

각 S의 길이를 가지는 M개의 슬롯에 할당될 i번째 블록 데이터를  $b_i$ ,  $i=1, 2, 3, \dots, M$  이라고 하자. 여기서 M은 허프만 부호로부터 생성되는 영상의 출력 블록들의 수이다. 그리고 슬롯에 할당될 블록 데이터  $b_i$ 의 비트 수를  $\chi(b_i)$ 라 하고, 알고리즘의 n 단계에서 슬롯  $m_i$  내의 비트 수를  $l(m_i^n)$ 이라고 하자. 또한 지시 함수 I (indicator function)는 연속된 슬롯의 구간이다. 이 구간 내 모든 슬롯은 S보다 크거나(Large), 작을(Small) 값을 갖는다. 이 구간 I 내의 슬롯들로 구성되는 집합을 아래와 같이 정의한다.

정의 1 : 만일  $I\{l(m_j) \geq S\} = Large, j=i, i+1, \dots, k$  이고  $I\{l(m_{i-1}) < S\} = Small, I\{l(m_{k+1}) < S\} = Small$  이면, 집합  $F = \{m_i, m_{i+1}, \dots, m_k\}$ 를 풀 클러스터(full cluster)라고 부른다.

정의 2 : 만일  $I\{l(m_j) < S\} = Small, j=i, i+1, \dots, k$  이고  $I\{l(m_{i-1}) \geq S\} = Large, I\{l(m_{k+1}) \geq S\} = Large$  이면, 집합  $P = \{m_i, m_{i+1}, \dots, m_k\}$ 를 부분 풀 클러스터(partially full cluster)라고 부른다.

풀 클러스터 블록  $m_i$ 가 부분 풀 클러스터 슬롯  $m_j$ 를 찾아서 자신의 남은 비트를 채울 수 있는 확률은 풀 클러스터들의 집합 개수를 건너뛴으로써 높아질 수 있다. 왜냐하면 슬롯의 고정 길이 S를 초과하는 블록들이나 초과하지 않은 블록들은 영상에서 이웃 블록들 간의 높은 상관관계로 인하여 연속적으로 모여 있을 확률이 높기 때문이다. 따라서 풀 클러스터 블록은 풀 클러스터들의 연속적인 슬롯들을 건너뛰고 난 후에 존재하는 부분 풀 클러스터에 자신의 남은 비트를 할당할 수 있는 확률이 높아진다.

만일  $F_1, F_2, \dots, F_r$  을 r개의 풀 클러스터들 집합이라고 하고,  $P_1, P_2, \dots, P_s$  를 s개의 부분 풀 클러스터들 집합이라고 각각 정의하면, 각 클러스터 집합들의 평균 길이는 다음과 같이 주어진다.

$$L_f = \left[ \frac{1}{r} [C(F_1) + C(F_2) + \dots + C(F_r)] \right] \quad (3)$$

$$L_p = \left\lceil \frac{1}{s} [C(P_1) + C(P_2) + \dots + C(P_s)] \right\rceil \quad (4)$$

여기서  $C(\cdot)$ 는 각 클러스터 집합 내에 존재하는 슬롯의 개수로 정의한다. 따라서 평균적으로 비어 있는 슬롯은  $\lceil (L_f + L_p)/2 \rceil$  만큼의 슬롯을 건너뛰면 찾을 수 있다. 따라서 초기 오프셋 값( $\Phi_1$ )은  $\lceil (L_f + L_p)/2 \rceil$  값으로 선택하고, 양방향(bidirection) 탐색을 사용한다. 이후의 단계에서 사용될 오프셋 시퀀스 값들은 초기 오프셋 값을 이용하여 각 단계마다 다음과 같이 계산된다.

$$\Phi_{k+1} = \begin{cases} -\Phi_1, & \text{if } n = 2 \\ \Phi_1 + (2k-1)(\text{mod } N_1), & \text{if } n = 2k+1 \\ \Phi_1 - (2k-1)(\text{mod } N_1), & \text{if } n = 2k+2 \end{cases}$$

### III. 제안된 알고리즘 EREREC

대부분의 블록 기반 영상 코딩 기법들은 입력 원본영상을 8x8 블록으로 나누어 처리한다. 블록내의 모든 픽셀들이 동일한 정보량을 가지므로 DCT를 이용하여 저주파영역으로 정보를 모아 처리하게 된다. 또한 인간 시각 시스템(HVS : Human Visual System)의 특성이 고주파의 변화에 둔감하고 저주파의 변화에는 민감하다는 특성이 있다. 인간 시각 시스템의 특성을 이용하여 고주파 계수보다 저주파 계수를 더 정확히 재생하기 위한 비선형 양자화를 사용한다. 이러한 양자화 과정을 통해 영상 신호는 왜곡되지만 영상의 압축이 가능해진다. 복원 영상은 원본 영상과는 차이가 있으나, 위에서 언급한 바와 같이 인간 시각 시스템이 고주파의 왜곡에 둔감하므로 원본과의 특별한 차이를 느끼지 못한다.

양자화 된 계수들은 지그재그 스캔되고 줄-길이 부호화와 더불어 대부분 압축 표준의 가변 길이 부호화로써 널리 사용되는 허프만 부호를 이용하여 부호화된다. 그 결과로 생성된 가변 길이 블록 비트열들은 EREC 알고리즘의 처리를 통해 일정 길이의 슬롯 구조를 갖는 비트열로 재구성 후에 전송된다. 대부분의 영상 부호화의 경우 변환된 블록의 부호 길이가 크면 이웃 블록의 부호 길이도 클 확률이 높고, 반대의 경우도 마찬가지이다.

따라서 본 논문에서는 연속적으로 큰 블록들과 작은 블록들의 상관관계와 VLC 블록의 특성에 의한 초기 탐색 위치 선택, 오프셋 값의 조절 및 슬롯 탐색 패턴을

사용하여 기존 FEREC의 처리 속도 성능과 압축 영상의 화질을 개선한 EREREC 기법을 제안한다.

제안된 기법을 설명하기 위해 사용된 정의는 기존의 EREC와 동일하다.  $M$ 개의 슬롯의 전체 길이( $L_{all}$ )와  $M$ 개 블록들의 가변 길이 비트의 평균인 고정 슬롯 길이( $S$ )는 2.1절에서 소개된 수식 (1)과 (2)를 각각 사용하여 계산한다. 그리고 2.2절에서 사용된 수식 (3)과 (4)를 이용하여 풀 클러스터와 부분 풀 클러스터의 평균 길이를 계산한다. 본 논문에서는 FEREC의 장점을 그대로 사용하고 효율성을 개선하기 위하여 다음의 알고리즘이 사용된다.

부분 풀 클러스터를 찾아서 풀 클러스터의 남은 비트들을 할당하기 위한 탐색 패턴 ( $SP$  : searching step pattern)은 다음의 수식 (5)와 같다.

$$SP = \left\lceil \frac{L_f + L_p}{2} \right\rceil \quad (5)$$

따라서 공간적인 이웃 블록들의 상관관계 특성을 이용하여 슬롯 탐색 시간을 감소시킬 수 있다.

풀 클러스터와 부분 풀 클러스터들의 집합들 중에서 평균 슬롯 길이( $S$ )보다 연속적으로 더 긴 블록들( $L_l$ )과 더 짧은 블록들( $L_s$ )의 최대 갯수를 각각 계산한다. 그리고 연속적인  $L_l$  블록들과  $L_s$  블록들의 시작 위치인  $P_{max}$ 와  $P_{min}$ 을 수식 (6)과 (7)을 사용하여 찾는다.

$$P_{max} = \arg \left( \max_p \left( \sum_{i=p}^{p+L_l} l(d_i) \right) \right) \quad (6)$$

$$P_{min} = \arg \left( \min_p \left( \sum_{i=p}^{p+L_s} l(d_i) \right) \right) \quad (7)$$

여기서  $p$ 는 최대 크기의 풀 클러스터와 최소 크기의 부분 풀 클러스터들의 시작 위치 값들이다. 이 값들을 이용하여 초기 위치( $P_{initial}$ )를 수식 (8)과 같이 구한다.

$$P_{initial} = (P_{min} - P_{max} + M) \text{mod } M \quad (8)$$

여기서  $M$ 은 전체 블록들의 수이다.  $P_{initial}$ 은 최대 블록 비트들의 합을 가지는 연속적인 큰 슬롯들과 최소 블록 비트들의 합을 가지는 연속적인 작은 블록들 간의 공간적 거리를 의미한다. 따라서 처리해야 할 비트가 많이 남은 블록들은 이 값을 최대한 활용하여 남은 공간이 많은 슬롯들을 찾을 수 있다.

본 논문에서는 초기 오프셋 값( $\Phi_1$ )으로 수식 (8)에서 구해진 초기 위치 값( $P_{initial}$ )을 사용한다. 초기 위치 값

인 초기 오프셋 값은 또한 이후의 단계에서 사용될 오프셋 시퀀스 값을 계산하는데 사용한다.

수식 (5)에 의해 결정된  $SP$ 와 위에서 선택된 초기 오프셋 값을 이용하여 수식 (9)와 같이 오프셋 시퀀스 값들을 계산한다.

$$\Phi_{k+1} = \begin{cases} (\Phi_1 + k \cdot SP) \bmod M, & k = \text{odd} \\ (\Phi_1 - k \cdot SP) \bmod M, & k = \text{even} \end{cases} \quad (9)$$

여기서  $k=1, 2, 3, \dots$  이며, '+'와 '-' 심볼은 각각 전방향(forward direction)과 후방향(backward direction)으로 비어 있는 슬롯의 탐색을 의미하고,  $\Phi_{k+1}$ 은 반복 탐색을 하는 단계의 오프셋 값을 나타낸다.

위에서 제시된 수식 (9)를 보면, 오프셋 값들은 수식 (5)에서 구해진 탐색 스텝 크기( $SP$ )의 곱으로 증가한다. 따라서 만약 임의의  $SP$ 가 사용되면, 오프셋 값들이 어느 정도의 반복 단계 후에 이전 오프셋 값들과 같은 값을 가질 수 있는데 이는 이전에 검색했던 슬롯을 다시 검색하는 것과 같다. 다시 말하면, 모든 슬롯의 탐색은 불가능하게 되고 남은 데이터를 가지는 슬롯은 더 이상 빈 슬롯을 찾지 못하는 결과를 낳게 된다.

본 논문에서는 이러한 오프셋 값들의 반복을 피하기 위한 필수적인 조건을 찾아서 적용하였다.  $SP$ 의 인수들  $\{1, x_1, x_2, \dots, x_m\}$  이라 하고 전체 슬롯의 인수들  $\{1, y_1, y_2, \dots, y_n\}$  이라고 할 때, 오프셋 값들의 반복은 아래와 같이 결정함으로써 해결할 수 있다.

만약  $x_i$  ( $i=1, 2, \dots, m$ )의 어떤 값들이  $y_j$  ( $j=1, 2, \dots, n$ )의 값들 중에 같은 값을 가지지 않으면, 각 반복 단계에서 사용될 오프셋 시퀀스 값들은 수식 (9)에 의해 즉시 선택 가능하다. 하지만, 만일 위의 조건이 어긋나는 같은 값을 가질 경우에는 수식 (9)에 의한 오프셋 시퀀스 값들을 바로 선택할 수 없다. 이런 경우에 본 논문에서는  $x_i$  값과  $y_j$  값이 같은 값을 가지지 않도록 하기 위해  $SP$  값을 조정한다.  $SP$  값 조정에 대한 동작은 아래의 요약에 나타내었다.

요약 1.  $SP$ 의 인수들과 전체 슬롯 값의 인수들 중에서 1을 제외한 동일한 인수 값이 존재하면, 해당  $SP$ 를 적용한 수식 (9)에서 얻어진 오프셋 시퀀스 값들을 사용할 수 없다. 왜냐하면 비어있는 슬롯을 찾기 위한 단계에서 반복 탐색 단계가 발생하기 때문이다.

요약 2.  $SP$ 의 인수들과 전체 슬롯 값의 인수들 중에

동일한 인수 값을 갖지 않는 필수 조건을 찾기 위해  $SP$ 의 값을 1씩 증가시켜 비교한다. 그리고 나서 수식 (9)에 얻어진 새로운 오프셋 값을 적용하여 새로운 오프셋 시퀀스 값들을 결정한다.

위의 설명 중에서  $SP$ 의 값은 증가를 시키거나 감소시킬 수 있다. 그러나 이웃 블록들과의 공간적 상관관계에 의하여 큰 오프셋 값을 가지는 경우가 비어 있는 슬롯을 찾을 확률이 더 높기 때문에 본 논문에서는  $SP$ 의 값을 1씩 증가시키면서 새로운  $SP$ 의 값을 찾았다.

제안된 EREREC 기법의 첫 단계는 각 영상 블록의 가변 길이 비트들을 대응하는 각각의 슬롯에 할당을 한다. 다음 단계에서는 가장 많이 남은 비트들을 가지는 슬롯과 가장 많이 비어 있는 슬롯들을 우선적으로 찾고, 초기 오프셋 값을 이용하여 남은 비트들을 해당 비어 있는 슬롯에 할당하기를 시도한다. 비어 있는 슬롯을 탐색하고 할당하는 작업은 오프셋 시퀀스 값들을 이용하여 남은 모든 슬롯 비트들이 비어 있는 슬롯들에 할당되어 채워질 때까지 반복된다.

복호화는 부호화의 역으로 동작한다. 부호화기가 오버헤더로 삽입한 슬롯길이, 초기 오프셋 값을 에러없이 수신하는 경우 슬롯에 에러가 발생하기 전까지는 부호화기가 검색한 슬롯을 동일한 순서로 검색할 수 있게 된다.

제안된 EREREC의 초기 오프셋 값 선택 방법은 기본 EREC나 FEREC의 초기 오프셋 값의 선택보다 더 적절하게 변화할 수 있다. 기본 EREC의 오프셋 값은 고정적이었지만, 제안된 기법의 오프셋 값들은 영상 블록들의 특성에 의존적이다. 이런 이유로 제안된 EREREC 기법은 영상 화질 전송에 있어서 무선과 이동 통신과 같은 에러 발생 환경 채널에서의 임의 에러 발생에 대해 효과적이다.

#### IV. 실험 결과

제안된 기법의 효과성을 증명하기 위하여 본 논문에서는 JPEG 부호화 기법에 대해 제안된 EREREC와 기존 EREC 및 FEREC 방법들을 적용하였다. JPEG 부호화에서 사용되는  $8 \times 8$  DCT, 선형 양자화, 지그재그 스캔과 허프만 부호를 이용한 가변 길이 부호화를 사용하였다. 그러나 DCT 계수들 중 DC 값은 DPCM 대신 8-비트 부호화를 사용하였다. 왜냐하면 DC 값은 영상



(a) Boat 영상 (a) Boat image  
(b) Lenna 영상 (b) Lenna image

그림 2. 원본 영상(256×256 크기 Boat와 Lenna 영상)  
Fig. 2. Original image(Boat and Lenna, 256×256 size).



(a) FERECE (a) FERECE  
(b) 제안 EREREC (b) Proposed EREREC

그림 3. 영상 화질 비교  
Fig. 3. The comparison of image quality.

정보에서 중요한 값이고 에러 전파에 대해 강건해야 하기 때문이다. 그런 후 생성된 가변 길이 부호(VLC) 비트열들은 EREC 슬롯 구조에 채워지고, 슬롯에서 남은 비트들은 비어 있는 슬롯으로 재할당 된다.

그림 2는 실험에 사용된 8-비트 그레이 레벨의 256×256 크기를 가지는 원본 영상을 보여준다.

그림 3 (a)와 (b)는 Boat 영상에 대한 1.0112bpp(bits per pixel)에서 0.03% 비트 에러율(BER : Bit Error Rate)을 가지는 기존의 FERECE와 제안된 EREREC 기법의 영상 화질 비교를 보여준다. 그림 3 (a)의 PSNR은 19.9609dB 이고, (b)의 PSNR은 22.4358dB로 약 2.47dB 정도의 높은 화질을 보여준다. 따라서 제안된 EREREC 기법이 영상 화질의 성능 면에 있어서 주관적으로나 객관적으로 우수한 성능을 보여준다.

그림 4는 Lenna 영상에 대한 0.9176bpp에서의 BER에 대한 PSNR의 비교 그래프를 보여준다. 그림 4의 결과에서 보여 지는 바와 같이 제안된 EREREC 알고리즘이 임의의 에러에 대해 기존의 EREC나 FERECE 알고리즘

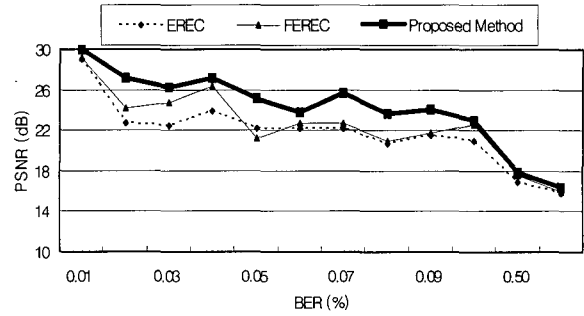


그림 4. 0.9176bpp에서의 BER / PSNR  
Fig. 4. BER / PSNR at 0.9176bpp.

보다 약 0.3 ~ 3.5dB 이상 영상 화질이 향상됨을 알 수 있다.

그림 3과 4의 결과와 같이 복원 영상의 화질이 개선되는 이유는 알고리즘의 반복횟수가 작을수록 원 블록의 데이터가 슬롯으로 분산되는 횟수가 줄어들기 때문이다. 슬롯에 에러가 발생하게 되면 복호화기가 잘못된 동작을 수행하므로 그 슬롯은 사용이 중지된다. 만약 어떤 블록이 그 슬롯에 자신의 데이터가 있는지 검색하게 되면 에러가 발생한 슬롯이므로 자신의 데이터가 있는지 판단할 수 없으므로 그 블록도 복호화를 중지해야만 한다. 따라서 알고리즘의 반복횟수가 복호화기에서의 에러발생 블록 수에 영향을 미치므로 복원영상의 화질에도 영향을 미친다.

제안된 방법은 임의의 에러 발생시 기본 EREC나 FERECE보다 블록의 분산량을 감소시켜 에러 블록으로 간주될 수 있는 상태수를 줄임으로써 복원 영상의 화질을 향상시킬 수 있었다.

표 1은 제안 EREREC와 기본 EREC 및 FERECE 기법들의 첫 단계에서 남은 비트를 가지는 슬롯들을 비어 있는 슬롯에 할당하는 블록들의 개수를 비교한 것이다. 앞에서 설명한 바와 같이 첫 단계에서 이동 블록의 수가 많을수록 전체적인 탐색 단계의 횟수를 줄일 수 있고, 탐색 단계 반복 횟수는 복원 영상의 화질에 영향을 준다. 제안 기법이 기존의 방법들 보다 더 많이 비어 있는 슬롯을 찾고, 그 곳에 슬롯에서의 남은 비트들을 할당할 수 있다.

표 2는 여러 종류 영상들의 bpp에 대한 제안 알고리즘과 기존 방법들의 전체 탐색 반복 회수를 비교하였다. 제안 EREREC 기법이 기존의 방법들 보다 전체 탐색 반복 횟수가 현저히 감소하였다. 이는 제안 기법이 기본 EREC나 FERECE 보다 슬롯 처리 성능 면에서도

표 1. 첫 단계에서의 이동하는 블록들의 수 비교  
Table 1. Comparison of the number of moving blocks at 1st stage.

Methods Image	EREC	FEREC	Proposed EREREC
Boat	128	183	306
Lenna	134	211	266
Baboon	74	236	277

표 2. EREC 처리 속도를 위한 방법들의 탐색단계 반복횟수 비교  
Table 2. Iteration number of searching steps for EREC processing speed.

Image	bpp	EREC	FEREC	EREREC
Boat	1.0112	960	532	298
Lenna	0.9176	960	557	288
Zelda	0.7715	946	466	127
Peppers	0.8220	966	432	220
Couple	0.7607	968	524	326
Fruit	0.8220	966	432	220
Church	0.9008	974	499	317
Kids	0.9202	1001	530	362
House	0.6152	975	502	292
Face	0.8553	953	509	356
Collie	0.8519	954	512	180
Einstein	0.7487	978	599	225
Cronkite	0.5790	928	252	101
Cbcar	1.0957	952	487	336
Baboon	1.6607	1005	480	235

약 2배 이상 빠르고 효과적임을 보여 준다.

### V. 결 론

본 논문에서는 기존의 FEREC보다 성능이 개선된 EREREC 기법(Efficient and Robust EREC)을 제안하였다. 먼저, 연속적인 블록들의 비트 길이에 의한 초기 탐색 위치를 결정한다. 다음으로 초기 오프셋 값은 슬롯의 길이보다 긴 블록들과 짧은 블록들의 통계적 분포를 사용하여 결정하고, 선택된 초기 오프셋이 모든 오프셋 시퀀스 값에 사용될 수 있는지를 확인하고 조정하였다. 실험 결과에서 보여진 바와 같이, 제안된 EREREC 알고리즘은 기존의 EREC 기법들보다 처리 속도가 개선되었으며, 무선이나 이동 통신상에서 발생할 수 있는 임의 에러 발생에 대해 전송되는 압축 영상의 화질 또한 향상시킬 수 있었다.

### 참 고 문 헌

- [1] Mohammad Ghanbari and Vassilis Deferidis, "Cell-loss concealment in ATM Video Codecs," *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 3, no. 3, pp. 238-247, June 1993.
- [2] Hiroshi Ohta and Tokuhiko Kitami, "A cell loss recovery method using FEC in ATM network," *IEEE Journal on Selected Area in Communi.*, vol. 9, no. 9, pp. 1471-1483, Dec. 1991.
- [3] Yao Wang and Qin-Fan Zhu, "Error control and concealment for video communication: A Review," *Proc. of the IEEE*, vol. 86, no. 5, May 1998.
- [4] N. T. Cheng and N. G. Kingsbury, "The ERPC: An Efficient Error-Resilient Technique for Encoding Positional Information or Sparse Data," *IEEE Trans. on Communi.*, vol. 40, pp.140-148, Jan. 1992.
- [5] D. W. Redamill and N. G. Kingsbury, "The EREC: An Error Resilient Technique for Coding Variable-Length Blocks of Data," *IEEE Trans. on Image Processing*, vol. 5, pp. 565-574, Apr. 1996.
- [6] R. Chandramouli, N. Ranganathan and Shivaraman J. Ramadoss, "Adaptive Quantization and Fast Error-Resilient Entropy Coding for Image Transmission," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 8, no. 4, pp. 411-421, August 1998.

### 저 자 소 개



김 정 식(정회원)  
제 38권 SP편 제 5호 참조



이 근 영(평생회원)  
제 35권 S편 제 1호 참조