

# 에러 환경에 강한 항등비트율 동영상 부호화기에 관한 연구

정희원 한 동 원\*, 송 진 규\*\*, 김 용 구\*\*, 최 윤 식\*\*

## A Study on An Error-Resilient Constant Bit Rate Video Codec

Dong-Won Han\*, Jin-Gyu Song\*\*, Yong-Goo Kim\*\*, Yoonsik Choe\*\* *Regular Members*

### 요 약

본 논문에서는 이동 통신망과 같은 열악한 에러 환경을 고려한 부호화 알고리즘을 제안한다. 제안된 부호화기의 알고리즘은 인트라 프레임 부호화에는 벡터 양자화 기법을 적용하였으며 Classified VQ기법을 통해 유사 벡터 탐색에 필요한 부하를 절감하였다. 인터 프레임 부호화에는 압축 효율의 증대를 위하여 움직임의 변화가 없는 블록은 전송하지 않고, 전송하는 블록의 위치와 개수에 대한 정보를 두 프레임에 걸쳐 이중적으로 부가하는 이중 벡터 위치부호기법을 제안하였다. 또한 부호화된 비트열의 길이가 항등하도록 함으로써 발생된 에러의 전파현상을 방지하였으며 복호화 과정에서 제안된 이중 벡터 위치부호를 이용함으로써 주요 헤더 부분이 급격한 에러에 의해서 손상되더라도 올바른 복호화가 가능하도록 하였다. 부호화된 비트열에 실제 에러를 발생시킨 후에 복호화를 한 실험 결과 기존의 부호화 기법인 H.263과 비교하였을 때 에러에 훨씬 강한 특성이 나타남을 알 수 있다.

### ABSTRACT

In this thesis, an error resilient video coding algorithm, under the error-prone environment such as wireless communication, is suggested. The suggested algorithm adapts the Classified VQ method for intra images that reduces some load by searching similar vectors. The Duplicate Vector Position Code is proposed for the higher compression efficiency and the robust decoding in error environment. As a result, the bitstream encoded by the proposed method is in a CBR(Constant Bit Rate) preventing from error propagation. The experiment that adds practical error to the encoded bitstream shows the error-robustness superior to H.263.

### I. 서론

최근 들어 광통신, 이동 통신 등의 통신망이 확충됨에 따라 점차적으로 기존의 음성, 문자 등의 정보 뿐만 아니라, 동영상을 포함하는 멀티미디어 정보의 처리 및 전송기술의 요구가 증대하고 있다. 그러나, 이동 통신망을 이용하여 동영상과 같은 대용량의 데이터를 전송하기 위해서는 기본적으로 기존의 압축 부호화 기법에 비해 높은 압축율을 실현해야 하며, 그 외에 다음의 두 가지 제약 조건을 해결해야 한다<sup>[1]</sup>.

첫째는, 가용 대역폭이 크게 제한되어, 동영상 및 멀티미디어 데이터를 전송하기에는 매우 열악한 조건이다.

둘째로는, 이동 통신망은 지상 통신망과는 달리, 주위 반사체의 영향에 매우 민감하며, 지표의 구조나 기후 등의 변화에도 영향을 받는 등, 매우 불안정한 전송 매체이다. 즉, 이동 통신망에서는 반사체의 구조나 이동, 움직임으로 인한 Fading효과로 인해 시간과 장소에 따라 Bit Error Rate나 SNR의 변화가 매우 심하고, 발생하는 에러의 특성이 급격

\* LG 정보통신

\*\* 연세대학교 공과대학 기계전자공학부(yschoe@bubble.yonsei.ac.kr)

논문번호 : 99164-0424, 접수일자 : 1999년 4월 24일

※ 본 연구는 한국과학재단 핵심전문연구비(961-0920-103-2) 지원으로 수행되었음.

하고 순간적이어서 에러 발생 시 데이터의 손실이 상당히 크다.

따라서 부/복호화기 설계 시, 하드웨어뿐만 아니라, 소프트웨어적으로도 이러한 제약점들을 고려하여야 한다.

본 논문에서는 동영상 부호화 알고리즘을 개발함에 있어, 이동통신망과 같은 열악한 에러 환경을 고려한 부/복호화 기법에 중점을 두어 그 가능성을 검토한다. 한편, 동영상 정보를 부호화함에 있어서 일반적으로 프레임간의 시간적 중복성을 이용하여 움직임 예측을 통한 부호화방법을 채택하는데, 기존의 DCT와 움직임 보상기법을 기반으로 하는 부호화 기법들에서는 높은 압축 효율을 위해서 변환계수 및 움직임 벡터 등의 파라미터를 엔트로피 부호화방법<sup>[2]</sup>에 의해 부호화하고, 움직임 벡터들은 DPCM방식으로 부호화하는 방식을 채택한다. 이러한 결과 각 부호화된 파라미터들에 효율적으로 비트를 부여할 수 있게되어 압축 효율이 극대화되었지만, 각 비트들의 길이가 가변적이어서 통신 채널의 상태가 불안정하여 에러가 발생할 경우, 각 비트의 정확한 복호화가 불가능해 화질의 열화가 심한 문제점이 있다.

기본 부호화 알고리즘으로서는 인트라 프레임 부호화에는 벡터 분류기법에 의한 벡터 양자화를 사용하며, 인터 프레임 부호화에는 일반적인 움직임 예측 기법을 채택하되, 에러전과 현상을 방지하게 위해 DPCM<sup>[3][4]</sup>과 엔트로피 부호화 기법<sup>[3][4]</sup>을 사용하지 않는다. 따라서 부호화된 비트열의 길이가 항등한 구조를 지니게 되는데, 이에 따른 압축 효율의 저하문제를 해결하고 아울러 에러발생 시 올바른 복호화를 가능하게 하는 방법으로서 제로 벡터 생략과 이중 벡터 위치정보에 의한 부호화기법을 제안한다.

그리고, 기존의 동영상 부호화 표준인 H.263<sup>[5]</sup>과 비교하여, 제안되어진 부호화 방법이 이동 통신망과 같은 열악한 에러환경에서의 부호화방법으로서 효율적인 대안이 될 수 있음을 제시한다.

본 논문에서는 인트라 프레임의 부/복호화 방식으로 쓰이는 벡터 양자화 기법을 2장에서 설명하고, 제안된 부호화 기법은 3장에 설명이 나온다. 4장에는 실험과 그 결과에 대해서 기술하고, 5장에서 결론을 맺는다.

## II. 인트라 프레임 부/복호화

본 장에서는 제안된 벡터 양자화를 이용한 항등

비트열 부호화기를 구성하기 위해서 그 기본이 되는 벡터 양자화기법에 대하여 설명하고, 아울러 부호화기의 계산상의 부하를 줄이기 위한 벡터 classify기법, 부호화 테이블 생성기법 등의 이론적 배경을 요약한다.

### 2.1 벡터 양자화

벡터 양자화(Vector Quantization)는 80년대 초반부터 음성 및 영상 신호 처리 분야에서 활발히 연구되어져 오고 있는 압축 부호화 기법의 하나이다<sup>[6][7]</sup>. 예측 부호화나, DCT같은 변환 부호화 기법들은 각 개별 입력 값들에 대해 이를 적용한 후, 이를 양자화 하는 방법을 사용하고 있다. 그러나, 만약, 처리할 신호들의 특성이 영상과 같이 일정 영역 안에서 서로 연관되어있거나 의존적인 관계라면, 개별 스칼라의 양자화보다는 연관되어진 스칼라들을 동시에 양자화 하는 벡터 양자화가 보다 효율적이다.

벡터 양자화의 부호화 과정을 살펴보면, 여러 가지 영상의 특성을 표현할 수 있는 다수의 실험 벡터들로부터 부호화 테이블(code book)을 구성하고, 이로부터 입력 영상의 단위 블록과 유사한 벡터를 찾아냄으로서 부호화를 수행한다. 따라서 유사 벡터를 찾는 과정에서 탐색으로 인한 연산량은 비교적 크나 복호화기의 구조는 다른 기법들에 비해 상당히 간소하다.

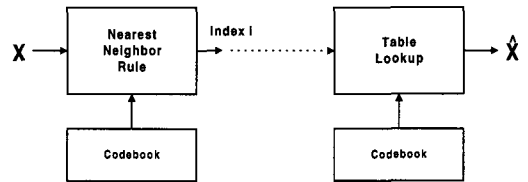


그림 1. 벡터 양자화에 의한 부/복호화 과정

그림에서 볼 수 있듯이, 복호화 단에서는 부호화 단과 동일한 테이블로부터 인덱스의 벡터를 찾아 영상  $\hat{x}$  를 복원한다.

본 논문에서는 부호화기의 입력 영상을 인물 중심의 영상으로 한정하고, 그 특성을 고려하여, 부호화 테이블 구성을 위한 실험 영상들을 인물 영상위주로 구성하여 제한된 부호화 테이블의 크기에서의 부호화 성능을 극대화하도록 하였다. 또한 탐색에 소요되는 시간 및 계산상의 부하를 줄이기 위해서, 주어진 영상을 경사도와 마스크 효과를 이용하여 분류하여 탐색을 제한하는 Classified VQ<sup>[8]</sup> 기법을 사용하였다.



문에서 제안하는 부호화 기법은 항등 비트율의 부호화기를 구성함에 있어 다음과 같은 점에 중점을 두도록 한다.

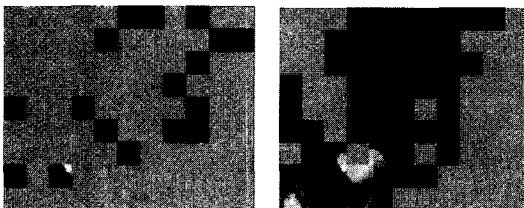
- ① 에러 전파의 방지
- ② 항등 비트율 부호화기의 압축 효율
- ③ 에러에 대한 저항성

본 논문에서는 복호화에 불필요한 정보를 전송하지 않는 방법과 필수적인 정보를 이중적으로 부가하는 방법을 제안하여, 위의 세 가지 중점 사항을 동시에 고려할 수 있는 부호화기를 구성한다. 기본이 되는 부호화 기법은 벡터 양자화기법과 H.263에서 사용한 것과 동일한 움직임 예측기법 등이다.

### 3.1 부호화기의 구성

본 절에서는 본 논문에서 제안한 항등비트율 부호화기의 구조에 대해서 설명하기로 한다. 부호화기는 일반적인 부호화기법들과 마찬가지로, 부호화 기법에 따라 크게 두 부분으로 나누어진다. 하나는 2장에서 설명하였던 벡터 양자화기법에 의한 인트라 프레임 부호화 부분이고, 다른 하나는 인터 프레임 부호화 부분으로서 H.263에서 사용했던 것과 동일한 움직임 예측 알고리즘을 사용한다. 이와 함께, 항등비트율 부호화기의 문제점의 하나인 압축효율을 개선하기 위해서 다음 절에서 이중 벡터 위치정보를 이용한 부호화 기법을 제안한다.

벡터 양자화의 경우는 본 논문에서는 인트라 프레임의 부호화에 사용한다. 이를 위해 부호화기는 입력영상을 4×4 크기의 블록으로 나눈다. 따라서, 복호화에 필요한 헤더가 불필요하며, 같은 길이를 가지는 인덱스 비트의 열로만으로 비트열이 구성되어 전송시 어느 한 비트가 손상된다 하더라도 해당 블록의 복원에만 영향을 주게 되어 에러의 전파가 발생하지 않는다. 부호화 테이블은 11비트 크기를 사용한다. 그 외에 부호화 과정에 드는 계산상의 부하를 줄이기 위해서 2장에서 언급한 Classified VQ<sup>[8]</sup>방법을 채택하였다.



(가) 프레임4:제로블록82개 (나) 프레임7:제로블록44개  
그림 3. 움직임 예측결과 변화가 있는 블록

움직임 벡터를 찾기 위한 탐색의 범위는 해당 매크로 블록을 중심으로 x, y 방향으로 각각 -15~+15 화소 범위로 제한하였으며, H.263에서 사용한 반화소 예측<sup>[5]</sup>은 사용하지 않는다. 또한 찾아진 움직임 벡터들은 x, y 각각 5비트씩을 이용하여 같은 길이로 부호화 되며, 에러 전파의 원인에 될 수 있는 DPCM방식은 사용하지 않는다.

그렇게 되면 압축 효율에 있어서 문제가 생기게 되는데, 그 해결책은 다음과 같다. 일반적인 화상전 화같은 용도로 부호화 되어지는 영상들은 그 특성 상 시간의 변화에 따른 움직임의 변화가 크지 않은 것이 보통이다. 이는 해당하는 블록을 복원할 때에 그 전의 복원영상에서 같은 위치의 블록을 그대로 사용하는 것을 의미하며, 만약 이러한 여분의 정보들을 전송하지 않아도 된다면 압축율 면에서 상당한 개선 효과가 있을 것이다. 그림 3의 (가), (나)는 위에서 언급한 움직임 예측 과정을 수행한 후, 그에 따라 복원한 영상으로 회색으로 나타나는 블록들은 전 프레임과 비교하여 움직임의 변화가 없는 블록들이다. 두 영상은 각기 3프레임 간격으로 부호화 된 것으로 제로 블록의 개수가 각각 82개와 44개였다. 두 그림에서 볼 수 있듯이 프레임간에 움직임이 있는 블록은 주로 인물의 얼굴과 손 부분에 한정되어 있고 배경에 해당하는 대다수의 블록들은 변화가 없음을 알 수 있다. 그러나 여기서 고려되어야 할 점은 전송되어지는 움직임 벡터들의 위치이다. 이 점은 다음에 오는 이중 벡터 위치정보 부호화방법에 의해서 해결한다.

### 3.2 이중 벡터 위치 정보 부호화

제로 블록의 정보를 전송 안 할 경우에는 전송할 나머지 움직임 벡터의 개수가 일정치 않기 때문에 복호화 단에서 이들 움직임 벡터들의 위치와 개수에 대한 정보가 필요하다. 본 논문에서는 이를 위해서 이중 벡터 위치정보 부호화기법을 제안한다. 제안방법을 적용함으로써, 복호화 단에서 일정치 않은 움직임 벡터에 대한 정보를 알 수 있으므로 움직임이 없는 매크로 블록의 제로 벡터들을 전송하지 않는 것이 가능해져 앞에서 말한 문제점이 해결될 뿐만 아니라 그 구조면에서 항등 비트율을 유지하기 때문에 발생 에러에 대한 저항성을 제공할 수 있는 이점이 있다. 제안된 이중 벡터 위치정보에 의한 부호화 기법이 그림 4와 그림 5에 표현되어 있다.

제안된 방법은 그림 4와 그림 5에서 볼 수 있는 것과 같이 부호화되어질 프레임  $Frame F(t)$ 의

비트열에 그 전에 부호화되어진  $Frame F(t-1)$ 의 움직임 벡터에 대한 위치정보를 동시에 가지도록 부호화하는 방법이다. 그림 4의 각 Frame들이 입력되어진 영상의 시퀀스라고 가정할 때, 이들은 각기 9개의 매크로 블록씩을 가지며 채색되어진 블록들은 그 전 프레임과 비교하여 움직임의 변화가 발생한 블록, 즉, 움직임 벡터들이 제로가 아닌 블록들을 나타낸다.

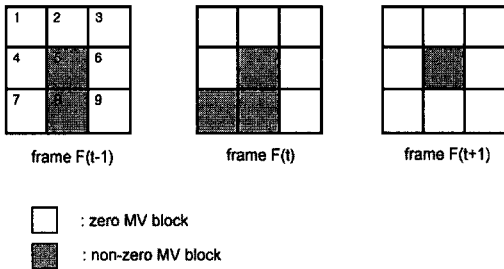


그림 4. 이중 벡터 위치부호 기법의 적용예

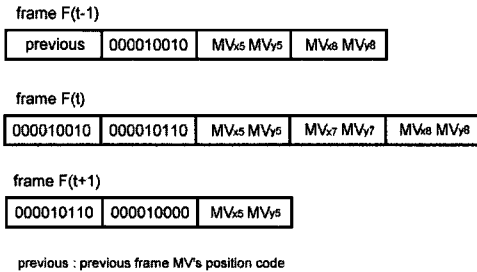
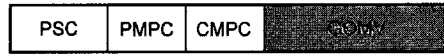


그림 5. 이중 벡터 위치정보에 의한 부호화방법

각 움직임 벡터들은 그림 4에 나와있는 순서대로 배정된다. 그림 5에서 보는 것과 같이 각 프레임의 비트열의 처음에 그 전 프레임의 벡터 위치부호 9비트와 부호화되어질 해당 프레임의 위치부호 9비트가 덧붙여진다. 이는 비트가 0이면 해당 위치가 제로 벡터임을 뜻하고, 1이면 제로 벡터가 아님을 뜻한다. 이렇게 이전 프레임의 움직임 정보를 현재 부호화되어질 프레임의 비트열에 이중적으로 덧붙이는 이유는 에러에 대한 저항성을 제공하기 위함이며, 이는 에러가 발생하더라도 복원할 영상의 전 프레임에 해당하는 비트부터 현재까지에 걸쳐 전체적으로 영향을 미치는 에러가 발생하는 확률이 희박하다는 가정에 기초한다. 따라서 에러 발생으로 인한 비트의 손실이 있을지라도 여분의 위치정보를 통해서 복호화가 가능할 것이다. 이에 대한 자세한

설명은 다음 3.3절에서 하기로 한다.



PSC : Picture Start Code (12bit)  
 PMPC : Previous frame MV's Position Code (99bit)  
 CMPC : Current frame MV's Position Code (99bit)  
 GOMV : Group Of Motion Vector

그림 6. 이중 벡터 위치부호 기법에 의해 부호화된 비트열의 구조

### 3.2.1 비트열의 구조

위 절에서 설명한 부호화 방법에 의한 부호화 비트열이 복호화단에서 완전하게 복호화가 가능하도록 하려면 몇가지 고려하여야 할 점들이 있다. 다음 그림 6은 부호화되어진 비트열의 구조에 대한 그림이다.

#### ① PSC(Picture Start Code)

그림 6에서 보는 바와같이 앞에서 한 프레임의 비트열의 처음에 위치하는 비트로, 전 프레임의 움직임 벡터 비트들과 다음 프레임의 시작을 구별해주는 기능을 한다. 이 비트는 총 12비트로 이루어져 있으며, 값은 모두 1들로 구성된다. 따라서 다른 비트들과 합쳐져 전체적인 발생 비트량이 움직임 벡터  $MV_x$ ,  $MV_y$ 의 비트합인 10비트(= 5비트 + 5비트)의 배수가 되도록 한다. 이렇게 함으로써, 프레임 간의 동기화를 가능하게 해주고 또한, 에러의 전파를 방지해줄 수가 있다.

#### ② 이중 벡터 위치부호

앞 절에서 설명한 것과 같이 그림 6의 PMPC와 CMPC에 해당하는 것으로 각각, 이전 프레임과 현재 프레임의 벡터 위치정보이다. QCIF의 영상인 경우, 매크로 블록의 개수가 총 99개이므로 이 두 부호들은 각각 99비트로 이루어져 있다. 결과적으로 한 프레임을 부호화하는 데 필요한 비트는 다음과 같다.

$$PSC(12 \text{ bit}) + PMPC(99 \text{ bit}) + CMPC(99 \text{ bit}) + \text{Number of Non-Zero block} \times 10 \text{ bit} \quad (1)$$

3.2.2절에서 언급했듯이 평균 제로 블록의 개수를 약 65개로 가정하여 계산하면, 총 비트 발생량은 대략 평균 550비트 정도이다. 만약, 움직임 벡터 모두를 같은 길이의 비트로 전송한다면 총 비트 발생량

은 영상안의 총 매크로 블록의 수와 움직임 벡터의 비트와의 곱인  $99 \times 10 = 990$  비트가 되어 제안된 방법에 의해 부호화를 했을 경우와 약 1.8 : 1의 비트율 감소 효과가 있었다.

3.3 여러 환경에서의 저항성

본 절에서는 복호화에 대해서 설명하고자 한다. 그 중에서도 특히 제안된 이중 벡터 위치 부호화기 법에 의해서 부호화되어진 비트열의 복호화에 중점을 두기로 한다.

3.3.1 이중 벡터 위치 부호의 복호화

벡터 위치 부호를 이중으로 전송을 하는 이유는, 움직임 벡터 자체는 같은 길이로 부호화되었기 때문에 에러에 의한 손상이 오더라도 해당 부분의 복호화에만 영향을 미치지, 압축효율을 위하여 움직임 정보가 없는 블록의 벡터는 전송하지 않았기 때문에 이에 대한 위치와 개수를 정확히 알지 못하면 복호화가 불가능하기 때문이다. 부호화 되어진 비트열의 구조는 그림 6 와 같으며, 이와 같이 부호화 되어진 비트열을 복호화할 경우, 기본적으로 해당 프레임의 CMPC를 읽은 후, 다음에 검출되는 PSC 까지의 비트 길이를 계산하여 에러 발생여부를 확인한다. 만약 해당 프레임의 CMPC와 검출된 PSC 다음에 오는 PMPC중 하나에 에러가 발생한 것이 확인되면, 에러에 영향을 받지 않은 나머지를 이용하여 복호화를 수행한다. 단, 앞에서 언급한 것과 같이 에러가 한 프레임 이상에 걸쳐서 생기지 않는다는 가정이 있어야 한다.

3.3.2 에러의 검출과 복호화

복호화과정에서의 발생된 에러에 대한 검출과 그에 따른 복호화 방법은 다음과 같다. 복호화과정은 그림 7 에서 보는 바와 같이 해당 프레임의 CMPC를 읽는 것으로부터 시작한다.

CMPC(t)는 현재 프레임의 움직임 벡터의 총 비트길이를 뜻하고, PMPC(t+)는 그 다음 프레임의 코드에 들어있는 현재 프레임의 움직임 벡터의 총 비트길이를 뜻한다.

즉

$$CMPC(t) = x, (x = \text{Number of } '1's \times 10) \text{bits}$$

$$PMPC(t+) = y, (y = \text{Number of } '1's \times 10) \text{bits(3-2)}$$

이 된다.

복호화시 CMPC를 읽어 x 비트후에 PSC가 검출된 경우(x = 1)에는, 해당 비트열에 에러가 발생하

지 않은 것으로 간주한다. 그러나 이 경우에도 에러 발생의 가능성이 있는데, 이는 x = 1 이고, 또한 x = y 인 경우에도 불구하고 CMPC에 의한 움직임 벡터의 위치정보와 PMPC에 의한 정보가 서로 같지 않은 경우이다. 그 원인은 두 헤더부분 중 한 곳에 발생한 에러로 인하여 비트가 손상되어 위치에 대한 정보는 바뀌었지만, 값 x 자체는 변하지 않은 경우이다. 이 경우에는 해당 프레임의 움직임 벡터를 모두 0으로 하여 복호화를 수행한다.

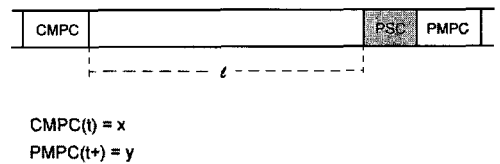


그림 7. 에러의 검출 및 복호화

에러가 발생하였다고 생각할 수 있는 경우는 x 비트후에 PSC가 검출되지 않은 경우인데, 이는 다음과 같이 세 가지로 생각할 수가 있다.

- ① 다음에 오는 프레임의 PSC비트가 손상되었을 경우
- ② 현재 읽은 CMPC가 손상되었을 경우
- ③ PSC, CMPC모두 손상된 경우

③과 같은 경우는 앞에서 언급한 바와 같이 두 프레임에 해당하는 구간에 걸쳐 에러가 발생한 경우에 해당하는 것으로, 본 논문에서는 그와 같은 열악한 환경은 고려하지 않기로 한다.

기본적으로 부호화시의 기본 단위인 10비트씩 읽는 과정을 통해, 다음 PSC가 검출되었을 때에는 다음 프레임의 PMPC값과 현재 프레임의 CMPC값과의 비교를 수행한다. 이 과정을 자세히 살펴보면 다음과 같다.

- y = 1 인 경우,
  - ②의 경우에 해당되며 복원시 PMPC를 이용한다.
- y ≠ 1 인 경우,
  - ①의 경우에 해당되며 복원시 CMPC를 이용한다. 이때 검출된 PSC는 2프레임 후의 영상을 뜻한다.

위에서 본 바와 같이, 제안되어진 방법에 의해서 부호화되어진 비트열은 임의의 한 부분이 급격한 에러 발생으로 인하여 손상되었다 하더라도, 에러 발생으로 인한 전체적인 복원 영상의 화질 저하를

방지할 수 있고, 비트열의 구조가 동일한 길이단위로 구성되는 항등 비트열이므로 그 특성상 발생 에러의 전파 현상이 일어나지 않는다. 또한 압축률도 제안된 제로 벡터 생략 기법을 통해서 상당한 개선이 이루어졌다.

#### IV. 실험 및 결과

본 논문에서 제안한 부호화기의 성능을 확인하기 위해서, 본문에서 언급한 대표적인 가변 비트율 부호화기인 H.263과의 비교를 행한다. 제안되어진 부호화기가 에러의 전파 및 에러에 대한 저항성에 중점을 둔 것이므로, 본 논문에서는 발생된 에러에 대한 복원 영상의 질을 비교하겠다. 제안된 부호화기가 압축율을 고려하여 인트라 프레임에서는 비교적 작은 부호화 테이블을 사용하였고, 인터 프레임에서는 H.263을 비롯한, 일반적인 동영상 부호화기에서 사용하는 오차 영상 부호화 기법을 채택하지 않았으므로, 에러가 발생하지 않은 상태에서의 정상적인 복호화 영상의 화질은 H.263에 비해 다소 떨어짐을 밝혀둔다.

##### 4.1 실험 방법

본 논문에서 수행한 실험은 임의의 주어진 에러 환경 하에서의 복호화 성능을 비교하는 것인데, 크게 두 가지로 나뉘 볼 수 있다.

하나는 인트라 프레임의 경우이고, 다른 하나는 인터 프레임의 경우이다. 비교 대상으로는 H.263 부호화 기법을 사용하였다. 동일한 환경에서 두 기법의 복호화 성능의 차이를 비교하여서 제안된 기법의 우수함을 보인다.

H.263과 제안된 기법의 경우에서 인트라 프레임과 인터 프레임을 부호화할 때에 각각 다른 부호화 기법을 사용하게 된다. 그렇기 때문에 이 두 종류의 프레임에 대해서 각각 실험을 해 볼 필요가 있다. H.263의 인트라 프레임은 DCT와 양자화를 통해서 부호화를 하게 되는데, 여기에서 양자화 계수를 Run-Length coding을 사용하여 부호화하기 때문에 에러에 대한 저항성이 VQ를 사용한 제안 기법의 저항성과 비교할 필요가 있다. 그리고 인터 프레임에서는 H.263이 움직임 보상을 위해서 사용하는 움직임 벡터를 VLC를 사용하여서 부호화하고, 제안 기법은 항등한 비트 수로 부호화하기 때문에 또한 비교할 필요가 있다.

또한, 제안되어진 방법에 대해서는 이중 벡터 위

치정보에 의한 에러에 대한 저항성을 확인하기 위해서, 비트열 중 움직임 벡터의 위치에 대한 정보를 담고 있는 부분에 집중적으로 에러를 발생시켜 그 결과를 제시한다. 이는, 제안된 방법이 전송되어지는 움직임 벡터의 위치 및 개수에 대한 정보를 연속되어진 프레임에 걸쳐 이중적으로 부가함으로써, 둘 중 하나의 손상으로 인한 전체적인 복호화의 불능 상태를 방지하도록 하는 것이었으므로, 비트열의 MPC(MV's Position Code)에 발생 에러의 대부분을 집중시킴으로서, 그 타당성을 알아보고자 함이다. H.263 기법과 제안 기법 모두, 부호화시 부호화 간격을 3프레임으로 하였으며, 5프레임만을 부호화했다. 부호화 영상은 QCIF(176×144) 형식의 susie 시퀀스를 사용하였다.

생성한 에러는 일반적인 이동통신 시스템의 설계 목표치인 BER  $10^{-3}$ ,  $10^{-4}$ 이며, 실험에 사용한 비트열이 5프레임에 해당하고, 자기 특정 프레임의 해당 비트열의 일부에 집중적으로 발생시켰으므로, 실험에서 가정한 에러 환경은 실제보다 상당히 열악할 것이라고 생각된다.

##### 4.2 실험 결과 및 고찰

###### 4.2.1 복호화 결과

표 1은 실험에 사용한 다섯 프레임에 대한 부호화 및 복원 결과이다. 이 표를 보면, 인터 프레임의 경우에 움직임 예측 결과, 움직임이 없는 제로 블록이 프레임 7의 경우, 전체 99개의 블록 중 거의 대부분임을 알 수 있다. 결과적으로 인터 프레임의 경우, 인터 프레임만의 평균 압축율이 354.2 : 1(평균 572.5비트)이다. 그러나 전체적인 평균 압축율의 경우는 벡터 양자화에 의한 인트라 프레임이 비트 발생량이 많았기 때문에, 인트라 프레임을 포함한 다섯 프레임의 평균 압축율은 51.4 : 1(평균 3,942비트)로 많이 떨어진다. 따라서 장면 전환이 적어서 인터 프레임의 수가 늘어나게 되면 압축율이 더 높아지게 됨을 알 수 있다.

표 1. 제안 기법에 의한 부호화 결과

frame	zero block	PSNR	bits	comp.ratio
1(intra)	.	31.83	17,424	11.6
4(inter)	44	31.94	760	266.8
7(inter)	82	31.95	380	533.6
10(inter)	76	31.70	440	460.8
13(inter)	49	31.09	710	285.6

4.2.2 에러 발생 실험 결과

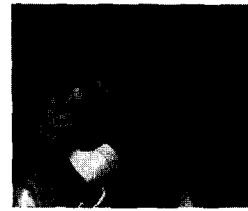
실험 1. H.263 기법에 의한 비트열에 에러가 발생한 경우(인트라 프레임)

그림 8 (나)는 H.263에 의해서 부호화되어진 비트열에 BER  $10^{-3}$ ,  $10^{-4}$ 에 해당하는 에러를 발생시킨 후, 복호화하여 복원한 전형적인 결과 영상이다. 발생 에러는 각각 3비트와 19비트에 해당하며, 에러 발생구간은 각 프레임 비트열의 30~60 % 영역이었다. 실험 결과, H.263의 부호화된 비트열의 구조가 항등 비트인 여러 가지 헤더들과 DCT등의 가변 비트인 파라미터들로 복잡하게 구성되어 있기 때문에 에러의 발생 위치에 따라 다양한 패턴을 가짐을 알 수 있었다. 그러나, 가변 비트인 파라미터들의 약점을 보완하기 위해서 삽입된 각 부호화 단위의 시작을 알리는 헤더 및 동기화 부호, 또한 복호화에 있어서 중요한 역할을 하는 DCT 계수중 DC값에 해당되는 비트들이 손상되면, 그림 8 의 (나)에서 볼 수 있는 것과 같이, 그 이후의 복호화가 불가능해짐을 확인하였다. 또한 블록 하나의 크기가  $8 \times 8$ 로서  $176 \times 144$ 의 QCIF 영상에 있어서는 비교적 큰 크기이기 때문에 에러의 전파를 고려하지 않더라도 블록 하나만의 에러만으로도 인간의 눈으로 인식하기에 영상의 질이 많이 떨어짐을 쉽게 알 수 있다.

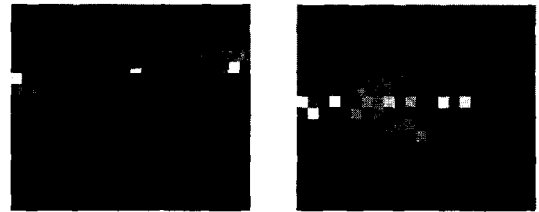
실험 2. H.263 기법에 의한 비트열에 에러가 발생한 경우(인터 프레임)

H.263에서의 움직임 예측 기법에 의한 인터 프레임의 해당 비트열에 대한 실험으로 실험 조건은 실험 1 과 같다. 인터 프레임의 경우는 인트라 프레임의 경우보다 압축율이 매우 높는데, 이는 움직임 벡터를 부호화할 때 허프만 부호화 기법<sup>[2]</sup>과 DPCM을 이용하여 효율을 극대화하였기 때문이다. 그러나 역시 그림 8 (다)에서 볼 수 있는 것과 같이, 발생되어진 에러에 의한 영향은 영상의 절반 정도를 복원할 수 없는 정도이며, DPCM의 특성으로 인하여 에러 전파 현상이 심하였다.

MPEG-1/2과 같은 H.263보다 높은 전송율을 위한 부호화기에서는 이러한 에러의 전파현상을 방지하기 위해서 각 슬라이스 단위마다 동기화 부호를 넣는 방법을 쓰기도 하는데, 이에 따른 오버헤드의 문제 때문에 화상 전화 같은 저 전송률의 응용 분야에는 적합하지 않다고 생각된다. 또한 슬라이스 헤더에서 에러가 발생했을 경우에는 그러한 복호화도 수월하지 않게 된다.



(가) 원 영상 frame 1



BER :  $10^{-3}$

BER :  $10^{-4}$

(나) H.263의 Intra Frame



BER :  $10^{-3}$

BER :  $10^{-4}$

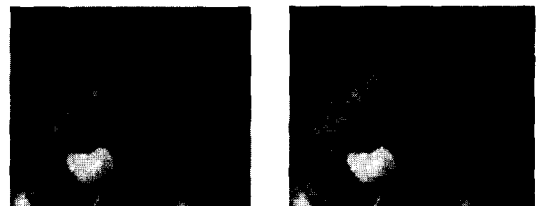
(다) H.263의 Inter Frame



BER :  $10^{-3}$

BER :  $10^{-4}$

(라) 제안 기법의 Intra Frame



BER :  $10^{-3}$

BER :  $10^{-4}$

(마) 제안 기법의 Inter Frame

그림 8. 원 영상과 H.263과 제안 기법에 에러를 발생 시켰을 때의 복호화 영상



실험 1, 2에서 볼 수 있듯이 H.263은 전송 비트율이 매우 낮은 상황에 적합하게 하기 위해서 여러 가지 압축 방법을 사용하였다. 그 결과 압축율은 높일 수 있었으나 에러에 매우 약한 비트열이 만들어지게 된다. 프레임의 종류에 관계 없이 비트열의 구조가 항등하지 않은 경우에는 에러의 전파에 대응하기가 힘들기 때문이다. 특히 DPCM의 맨 첫 번째 값에서 에러가 나면 그 한 프레임의 적절한 복호화는 보장할 수 없게 된다. 그렇기 때문에 에러의 발생 위치가 복호화된 시퀀스의 질에 많은 영향을 미치게 된다.

실험 3. 제안 기법에 의한 비트열 중에서 인트라 프레임에 에러가 발생한 경우

실험 3 은 제안된 부호화기에서 인트라 프레임의 에러에 대한 저항성을 확인하는 실험이다. 그림 8의 (라)는 벡터 양자화를 이용한 인트라 부호화에 의한 비트열에 BER이  $10^{-3}$ ,  $10^{-4}$ 인 에러를 발생시킨 후 복원한 영상이다. 벡터 양자화의 특성상 비트열이 항등한 구조이기 때문에 발생한 에러에 의해 손상되어진 인덱스에 해당하는  $4 \times 4$  블록부분만이 영향을 받을 뿐, 에러의 전파 현상은 일어나지 않음을 알 수 있다. 또한 그 인덱스의 단위도  $4 \times 4$  블록으로 비교적 작기 때문에 에러에 의한 영상의 질의 저하는 큰 문제가 되지 않는다. 그 이유는 인간의 시각 능력상 크기가 작은 노이즈에 둔감하기 때문이다. 따라서 그림 (나)와 비교하면, 영상의 질적인 면에서 좋은 성능을 보여줌을 알 수 있다. 실험 1에서 볼 수 있듯이, H.263의 경우에 블록의 단위가 커서 블록 내의 에러일지라도 한 프레임 전체에 큰 영향을 미치는 결과를 보여준다.

실험 3의 결과는 항등 비트율을 가지는 부호화기의 장점을 확인한 것으로, 부호화되어진 비트열의 구조가 항등하기 때문에 에러 전파 현상이 발생하지 않고, 따라서 이동 통신망과 같은 열악한 에러 환경에서 유리한 것을 알 수 있었다.

실험 4. 제안 기법에 의한 비트열 중에서 인트라 프레임에 에러가 발생한 경우

(이중 벡터 위치부호에 위치)

실험 4 에서는 본 논문에서 제안한 이중 벡터 위치 부호화 기법의 에러에 대한 저항성을 알아보기 위해서, 이중 벡터 위치부호가 들어 있는 비트열 부분에 집중적으로 에러를 발생시켜 보았다. 그 결과, 그림 8의 (마)에서 볼 수 있는 것과 같이 (다)보다 화질이 상당히 좋음을 확인 할 수 있었다. 이는, 3장에서 설명한 것과 같이, 전송하는 움직임 벡터의

위치와 개수에 대한 중요한 정보를 가지는 부호 비트를 연속된 프레임에 걸쳐 동일한 정보를 가질 수 있도록 했기 때문이다. 따라서 이중으로 전송된 정보의 어떤 한 부분이 에러에 의해서 손상되었다 하더라도, 손상되지 않은 다른 부호를 이용하여 복호화 과정을 수행할 수 있기 때문이다. 물론 움직임 벡터에 에러가 생기면 아무리 전송된 블록의 위치를 알더라도 잘못된 영역으로부터 움직임 보상을 하기 때문에 그 블록에서는 문제가 생길 수 있다. 그러나 DPCM과 VLC를 사용하지 않았기 때문에 바로 다음 블록부터는 이전의 블록에 전혀 영향을 받지 않고 정상적인 복호화가 가능하다.

이와 같은 결과에서 확인한 바, 일반적인 항등 비트율을 가지는 부호화기는 그 특성상 에러의 전파 문제를 해결할 수 있지만, 압축율이 현저히 떨어지는 문제가 있으나, 본 논문에서 제안된 이중 벡터 위치 부호화 기법에 의해서, 압축률 문제를 상당 부분 개선할 수 있었으며, 이와 함께, 발생한 에러 자체에 대한 저항성도 가지는 것으로 확인 하였다.

## V. 결론

본 논문에서는 벡터 양자화기법과 제안되어진 이중 벡터 위치정보 기법을 이용하여 이동 통신망과 같은 열악한 에러 환경을 위한 동영상 부호화 알고리즘의 가능성을 제시하였다.

부호화기에서는 일반적인 압축 부호화 기법들에서 적용하고 있는 DCT 기법을 대체하여 부호화와 복호화시 구조가 간단한 벡터 양자화 방식을 채택하였고, 움직임 벡터의 부호화시 에러 환경에 약한 DPCM방식을 피하고 개별 부호화 함과 동시에 압축효율을 위해 불필요한 정보를 전송하지 않음과 동시에 제안된 이중 벡터 위치정보를 이용하였다. 이렇게 함으로써 전체적으로 부호화기가 항등 비트율을 가지도록 설계하였다.

그 결과, 본 논문에서 고려되어진 이동 통신망이라는 열악한 에러 환경을 지니는 통신 채널을 통하여 동영상 데이터를 전송할 경우, 기존의 MPEG-1/2, H.263등과 같은 가변 비트율의 동영상 부호화 표준안들에서 적용한 부호화 기법들보다 에러 환경에 강한 특성을 지님을 알 수 있었다. 이와 함께, 부호화 과정의 단순함으로 인하여, 이를 통한 하드웨어 구현시, 이동 통신용 단말기로서 적합한 방법이 될 수있다고 생각되어진다.

앞으로 개선할 점은 본 논문에서 제안한 기법이

