

데이터 숨김과 오류 내성 기법을 이용한 빠른 비디오 오류 은닉

김진옥[†]

요약

오류 은닉은 데이터 전송시 발생한 오류를 처리하는 데 중요한 역할을 하는 기술로 우수한 데이터 품질을 보이는 다양한 오류 은닉 방법들은 대개 복잡도가 높다. 하지만 복잡한 알고리즘은 실시간 응용 분야에 적용하기 어렵다. 본 연구에서는 오류 내성 기술과 데이터 숨김 기법을 이용하여 디코더의 오류 은닉 부담을 줄이는 방법을 제안한다. 이를 위해 공간적 오류 내성 인코딩 방법으로써 손실 블록의 확산을 막는 블록 인터리빙을 적용하며, 시간적 오류 내성 방법으로는 움직임 벡터의 손실을 확인할 수 있는 패리티 비트를 데이터 숨김 방법을 이용하여 디코더로 전송하는 구조를 적용한다. 또한 전송 비디오 블록의 경계선 특징을 미리 추출한 후 이 데이터를 데이터숨김을 통해 디코더로 전달하여 비디오 데이터가 전송시 손상되면 전달된 특징을 이용하여 은닉 처리함으로써 디코더에서 오류 은닉시 손실 정보를 주변 블록으로부터 예측해야 하는 과정을 줄여 계산 복잡도를 낮춘다. 본 연구에서 제안한 움직임 벡터 확인 패리티 비트와 블록 경계선 특징 데이터를 전송 블록에 데이터 숨김방법으로 전송하는 것은 표준 인코더의 복잡도에 큰 영향을 미치지 않는다. 제안 오류 은닉 방법이 인터넷과 같이 버스트 오류가 많은 채널에서도 디코더에서 전송 오류를 효과적으로 빠르게 처리 함을 실험 결과를 통해 알 수 있다.

A Fast Error Concealment Using a Data Hiding Technique and a Robust Error Resilience for Video

Jin-Ok Kim[†]

ABSTRACT

Error concealment plays an important role in combating transmission errors. Methods of error concealment which produce better quality are generally of higher complexity, thus making some of the more sophisticated algorithms is not suitable for real-time applications. In this paper, we develop temporal and spatial error resilient video encoding and data hiding approach to facilitate the error concealment at the decoder. Block interleaving scheme is introduced to isolate erroneous blocks caused by packet losses for spatial area of error resilience. For temporal area of error resilience, data hiding is applied to the transmission of parity bits to protect motion vectors. To do error concealment quickly, a set of edge features extracted from a block is embedded imperceptibly using data hiding into the host media and transmitted to decoder. If some part of the media data is damaged during transmission, the embedded features are used for concealment of lost data at decoder. This method decreases a complexity of error concealment by reducing the estimation process of lost data from neighbor blocks. The proposed data hiding method of parity bits and block features is not influence much to the complexity of standard encoder. Experimental results show that proposed method conceals properly and effectively burst errors occurred on transmission channel like Internet.

키워드 : 오류 은닉(Error Concealment), 오류 내성(Error Resilience), 블록 인터리빙(Block Interleaving), 데이터 숨김(Data Hiding)

1. 서론

인터넷의 발달로 멀티미디어 데이터를 네트워크를 이용, 전송하는 일은 보편화되었다. 압축된 미디어 데이터는 불안정한 네트워크 채널에 민감하기 때문에 사소한 비트 오류라도 디코더에서 영상 왜곡을 일으킨다. 특히 인터넷을 통해 비디오 데이터를 전송할 때 패킷 손실이 발생하는데 이때의 손실은 버스트 오류[1, 2]형태를 띤다. 이러한 오류를

처리하기 위해 은닉 기술의 적용이 필요하다.

비디오 데이터에 대한 오류 은닉은 공간적 영역에서의 오류 은닉 방법과 시간적 영역의 오류 은닉 방법으로 나뉜다. 공간적 영역에서의 오류 은닉방법[3, 4]은 움직임이 많은 영역에서 프레임 내 부호화 블록에 효과적인 방법으로 정상적인 주변 블록 정보를 이용하여 오류 블록을 보간함으로써 복잡도가 높지만 시각적인 품질 개선 효과는 더 높다. 시간적 영역에서의 오류 은닉 방법[5]은 복원된 주변 블록의 정보를 이용해서 손실 블록의 움직임 벡터를 추정 후 이전 프레임으로부터 손실된 블록을 보상한다.

[†] 정 회 원 : 세종대학교 컴퓨터공학과 교수
논문접수 : 2002년 10월 10일, 심사완료 : 2003년 3월 31일

본 연구에서는 인터넷과 같이 불안정한 채널을 통해 전송된 비디오의 품질을 유지하기 위해 디코더에서 오류 은닉을 처리하는데 은닉 효과를 개선하는 방법으로 오류 내성 방법과 데이터 숨김을 제안한다.

오류 내성은 디코더에서의 오류 은닉을 효과적으로 처리하기 위해 인코더에서 데이터 손실에 의한 오류의 여파를 최소화하는 오류 방지 기법이다[6, 7].

본 연구에서는 블록 인터리빙 구조와 움직임벡터 확인 패리티비트를 이용한 오류 내성 방법을 제안하여 인코딩시 블록 인터리빙 구조를 적용해서 프레임 내 오류 블록의 확산을 방지하도록 한다. 프레임간 움직임 벡터를 확인하는 패리티 비트를 생성한 후 이 패리티 비트를 데이터 숨김 방법으로 디코더로 전송 후 움직임 벡터 프레임에 오류가 발생했는지를 빨리 확인하도록 한다.

또한 본 연구에서는 디코더에서 오류 은닉 시 쌍선형 보간[3, 4]을 수행할 때 오류가 발생한 블록을 처리하기 위해 주변 정보를 이용하는 대신 인코더에서 블록의 특징 정보를 미리 추출하여 데이터 숨김을 이용, 디코더로 전송하면 디코더는 인코더가 추출한 특징을 이용하여 손실 정보의 보간을 빠르게 수행하는 방법을 제안한다. 인코더에서 특징 추출을 수행하면 인코더의 복잡도는 증가하지만 인코더는 더 많은 계산 자원을 가지고 있으며 오프라인에서도 수행할 수 있기 때문에 효과적이다. 인코더에서 추출한 특징은 미디어 데이터에 숨겨 별도 정보로 디코더로 보내는데 이 방법은 비트율을 높이거나 쓸모 없는 데이터를 생성하지 않는다.

2. 관련 연구

2.1 오류 제어

네트워크상에 비디오 데이터 전송시 발생하는 오류 제어 방법으로 FEC(Forward Error Correction)과 ARQ(Automatic Repeat Request)와 같은 소스 코딩 방법을 가장 많이 이용한다. 하지만 FEC는 오류 복구에 필요한 잉여 데이터로 인해 제어하기 어려운 비트 스트림을 만들기 때문에 인코더의 복잡도가 증가하면서 대용량 네트워크 대역이 필요하다. 또한 ARQ는 지연 시간이 엄격한 실시간 응용 분야에 적용하기 어렵다. 그래서 이 방법들은 인터넷과 같이 버스트 오류가 집중적으로 발생하는 통신 채널의 패킷 분실에 대해서는 효과적이지 않다. 이에 반해 오류 내성을 이용한 오류 은닉 방법은 이미지와 비디오의 공간적, 시간적 연관 관계를 이용해 잉여 데이터 생성이나 전송 지연 없이 오류를 처리하는 방법으로 인정받고 있다. 오류 은닉 방법으로 계산 복잡도와 품질간의 상충 관계를 다룬 다양한 연구가 제안되었다. Wang[7]은 손실된 DCT 계수를 예측하기 위해 공간적, 시간적 은닉 방법을 제안했다. 패킷 손실을 최소화하기 위해 블록을 인터리빙하여 손실 블록을 격리시

킴으로써 오류 복구를 쉽게 한다. 선형 보간법을 기반으로 한 오류 은닉 알고리즘은 계산이 단순하지만 블록 단위로 연산을 처리하기 때문에 인접 블록과 경계선 성분이 비연속적일 때는 블록 부산물을 다량 만들어낸다[8]. Zeng[3]과 Sun[4]이 제안한 알고리즘은 손상된 블록을 은닉하기 위해 주변 픽셀과 연결된 경계선 정보를 이용한다. 이 경우 데이터 부산물은 경감시키지만 계산이 복잡해서 실시간 응용에는 부적당하다. Lee[9]는 블록기반의 이미지 코딩 기술로 퍼지 논리 기반의 오류 은닉 방법을 제안했다. 대부분의 오류 은닉 방법들은 반복적인 계산이 필요하고 이 계산 비용이 높다는 단점이 있다.

오류 내성 방법으로는 MPEG-4에서 재동기화 마커, 데이터 분리, 가역 VLC 코딩과 같은 방법을 제안하고 있다. 이 방법은 랜덤 비트 오류에는 효과적이거나 인터넷의 오류 발생 특성인 버스트 오류에는 효과적이지 않다[10, 11].

따라서 본 연구에서는 인터넷과 같이 불안정한 패킷 채널을 통해 전송된 비디오의 버스트 오류를 방지하기 위해 블록 인터리빙과 움직임 벡터 패리티비트로 오류 내성 처리하는 동시에 디코더의 부담을 줄이면서 낮은 계산 비용으로 오류 은닉을 빠르게 수행하는 방법을 제안한다.

2.2 데이터 숨김

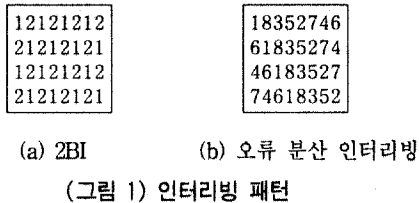
사용 권한 확인과 데이터 변조를 감지하는 목적으로 사용되고 있는 데이터 숨김은 일반적으로 비밀 전송, 워터마킹 등과 같이 원천 데이터에 다른 목적으로 사용하는 정보를 숨겨서 전달하는 방법이다. 데이터 숨김은 대상 비디오의 품질을 사람이 인지할 정도로 낮추면서 별도의 정보를 같이 전달한다. 별도 정보는 패킷의 이용자 데이터 필드나 헤더 필드와 같은 곳에 별도로 부가해서 전송한다. 이 방법은 소스 코딩이나 채널 코딩에서 잉여 데이터를 사용하는 것과 유사하다. 그러나 데이터 숨김의 이점은 MPEG의 표준 형태를 유지하면서 효과적으로 정보를 전달한다는 점이다. 또한 데이터 숨김은 전체 비트율을 유지하면서도 사용자 필드에 별도 정보를 덧붙여 비디오를 더 낮은 비율로 변형 코딩한다. 변형 코딩시 비트율과 시각적 품질간의 상충 관계를 조율하는 데는 복잡한 비율 제어가 필요하기 때문에 이 분야에 많은 알고리즘이 제안되고 있다[14]. 데이터 숨김은 적용 방법에 따라 분류되는데 그 중 공간적 영역 방법[15]은 몇 개 픽셀을 변형하여 송신자와 정해진 수신자만 그 변형된 위치를 감지하는 방법이다. 주파수 영역 방법[16]은 DCT 상관 계수의 일부 또는 전체를 바꾸는 방법으로 이미지를 변형하여 에너지가 전체적으로 확산하게 하는 방법이다.

본 연구에서는 시간적 오류 내성 방법인 움직임 벡터의 패리티 비트를 I 프레임에 숨겨 전송하고 비디오 블록의 경계선 정보를 디코더로 전송하기 위해서는 주파수 영역 방법의 데이터 숨김을 이용한다.

3. 공간적 오류 은닉을 위한 제안 오류 내성 방법

인터넷을 통해 전송된 패킷은 분실될 경우 그 분실 위치가 노출되는 버스트 오류이다. 버스트 오류는 코딩 블록을 심하게 손상시켜 오류 은닉이 불가능하게 만들기도 한다. 따라서 버스트 오류를 처리하는 동시에 프레임 내에서 손상된 블록의 오류가 다른 블록에 확산되지 않도록 비디오 데이터의 인코딩시 데이터 치환 방법인 블록 인터리빙을 적용한다. 오류 블록이 분리되지 않으면 비디오 전체에 영향을 끼쳐 디코더에서 오류 은닉을 처리하기 어렵다. 블록 인터리빙의 다른 잇점은 오류를 분산시킨다는 점이다. 집중적으로 발생한 오류는 움직임 보상을 통해 다음 프레임에 더 잘 파악되기 때문에 블록을 인터리빙 함으로써 오류 파악 효과를 낮춰 오류 발생을 줄인다. 랜덤 블록 인터리빙 [16]은 오류를 격리하는데 오류가 발생한 블록이 4% 이하여야 효과적이기 때문에 인터넷 응용 분야에 적용하기 어렵다. 랜덤 블록 인터리빙에서 제안된 홀-짝수 블록 인터리빙 또는 양방향 블록 인터리빙(2BI) 방법 [17]은 (그림 1)(a)과 같다. 모든 블록은 '1' 또는 '2'로 정한다. 각 줄의 모든 '1' 블록을 차례로 인코딩하고 난 후 다음 줄의 '1' 블록들을 인코딩 한다. 다음 '2' 블록들을 같은 방법으로 인코딩하고 나면 각 '1' 블록은 4개의 '2' 블록들로 싸이게 되고 각 '2' 블록들은 4개의 '1' 블록들로 싸인다. 이 경우 '1' 또는 '2'로 정한 블록이 분실되면 분실율이 50%에 달하게 되어 오류 복구가 어려워진다.

본 연구에서 제안하는 인터리빙 패턴은 오류를 이미지의 전체 영역으로 분산시켜 중요한 연결 블록들이 동시에 분실될 수 있는 가능성을 낮춘다. 이 인터리빙 패턴을 '오류 분산 인터리빙'이라 부른다. 오류 분산 인터리빙은 오류를 효과적으로 분산하고 압축할 때는 잉여 데이터를 줄인다. 오류 분산 인터리빙은 송신된 패킷에서 블록의 모든 열과 행은 분실 가능성이 동일하다는 점을 고려했다.



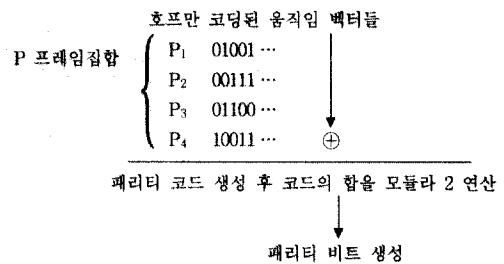
(그림 1)(b)는 제안 인터리빙 패턴을 설명하고 있다. $S = N \times M$ 크기의 이미지가 8×8 크기의 블록으로 DCT 변환 되었을 때 이 패턴은 다음과 같은 방법으로 생성한다. N 개의 열과 M 개의 행으로 구성된 블록에서 $M > N$ 이라고 가정한다. 최상 좌측 모서리부터 시작하여 (i, i) 블록을 '1'로 레이블을 붙인다. N 블록을 '1'로 표시한 다음 $(1, N+1)$ 블록부터 시작하여 모든 $(i, i+N)$ 블록을 '2'로 표시한다. 이미지의 우측 끝까지 표시한 후 $(1, j+1)$ 블록부터는 ' $k+1$ '로 표시한

다. 만약 $(1, j+1)$ 이 이미 표시되었다면 ' $k+1$ '을 $(1, j')$ 블록에 할당한다. j' 은 $j+1$ 과 가장 큰 값으로 표시한 인접 행의 인덱스 중간 값이다. 만약 이 역시 표시되어 있다면 ' $k+1$ '을 $(1, j'')$ 블록에 할당한다. j'' 은 $j+2$ 와 가장 큰 값으로 표시한 인접 행의 인덱스의 중간 값이다. 이 인터리빙 방법을 적용하면 버스트 오류가 발생한 블록이 격리된다.

2BI와 같은 고정된 인터리빙 패턴은 긴 비디오 스트림에서 쓸모없는 부산물 데이터를 만들 수 있다. 그래서 제안 방법과 같이 방향이 다른 인터리빙 패턴을 혼합하여 적용하면 부산물 데이터를 줄인다.

4. 시간적 오류 은닉을 위한 제안 오류 내성 방법

움직임 벡터는 시간적 영역에서의 오류 은닉을 위해 필수적이다 그래서 많은 오류은닉 방법이 손상된 움직임 벡터를 복원하는데 제안되었다. 그러나 손상된 영역이 그 주변과 연관 관계가 거의 없을 경우 움직임 벡터를 복원하는 것은 매우 어려운 일이다. 본 연구에서는 손상된 움직임 벡터를 빨리 복원할 수 있도록 인코더에서 움직임 벡터 확인 패리티 비트를 생성한 후 데이터 숨김을 통해 디코더로 전송하는 오류 내성 방법을 제안한다. Song [18]은 매크로 블록 그룹(GOB) 전체에 패리티 비트를 삽입하는 방법을 제안했는데 이 방법은 손실된 움직임 벡터가 프레임에서 한 개 GOB에만 한정되기 때문에 연속적인 GOB들이 한 개 프레임에서 손실될 가능성이 높은 인터넷에는 적절하지 않다. 따라서 본 연구에서는 미리 정해진 프레임집합의 움직임 벡터에 대해 패리티 비트를 생성하는 방법과 생성한 패리티 비트를 후속되는 프레임에 삽입하는 방법을 제안한다. 이 때 데이터 숨김을 이용함으로써 데이터 코딩 비율을 변하게 않게 유지한다. 이 구조는 버스트 오류 형태로 손실된 움직임 벡터를 잘 복원한다. 프레임간에는 P 프레임만 앵커 프레임으로 사용되고 오류 확산에 민감하기 때문에 P 프레임에만 움직임 벡터 보호방법을 적용한다. 움직임 벡터 보호를 위한 패리티 비트의 생성은 (그림 2)와 같다.



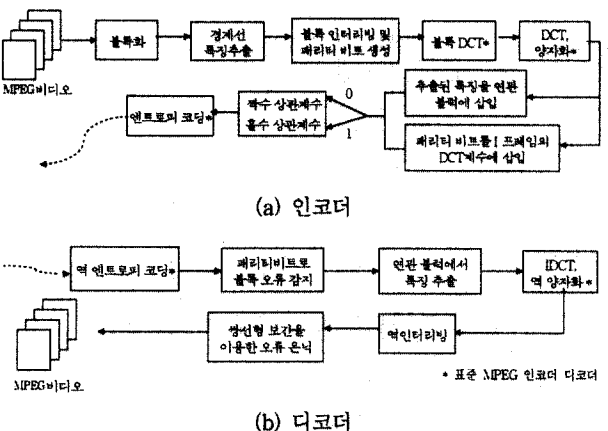
MPEG-1에서 움직임 벡터는 호프만 코딩된다. 각 P 프레임의 움직임 벡터의 코딩 비트를 열 단위로 만든다. 만약 매크로 블록에 움직임 벡터가 없다면 '0'을 할당한다. P 프

레이의 매크로 블록이 프레임 내 코딩인지 프레임 간 코딩인지 판별하게 위해 한 비트를 잉여 데이터 처리한다. (그림 2)와 같이 코딩된 움직임 벡터의 각 라인의 합을 모듈라 2 연산하여 패리티 비트를 생성한다. 각 열들의 길이가 동일하지 않기 때문에 긴 열과 짧은 열을 맞추기 위해 짧은 열의 빈 부분에는 '0'비트를 삽입한다. 이 구조는 각 비트 평면에 발생하는 오류를 확인하기 위해 1개의 패리티 비트를 사용하는 것이다. 생성된 패리티 비트를 그 다음 P 프레임의 양자화된 DCT 계수에 삽입한다. 두드러져 보이는 부산물 데이터를 덜 만들기 위해 DC와 낮은 AC 상관 계수는 바꾸지 않는다. 프레임간에 패리티 비트를 적용하는 것은 최소의 패리티 비트를 이용하여 같은 프레임에서 연속적으로 발생하는 오류를 보호하는 장점이 있다. 각 블록의 서로 다른 데이터 삽입 용량을 균일하게 하기 위해 인터리빙은 데이터 삽입 전에 적용한다. 만약 비디오의 삽입 용량이 적으면 움직임 벡터를 선택적으로 삽입하거나 움직임 벡터의 평균이나 합만을 삽입한다. 한 개 패리티 비트 집합을 삽입하기 위해 한 개 이상의 프레임을 이용할 수도 있다. 디코더에서는 전송된 패리티 비트를 이전 P 프레임 집합의 움직임 벡터와 비교하여 그 결과가 상이한 경우 오류가 발생한 경우로 판단하여 오류 은닉한다.

5. 데이터 숨김을 이용한 빠른 오류 은닉

5.1 제안 구조

(그림 3)은 MPEG 비디오의 오류 은닉을 지원하는 오류 내성 방법과 블록 특징 벡터의 데이터 숨김 및 처리 과정이다. 인코더에서 MPEG 비디오는 블록으로 분리되어 블록의 특징이 추출된 후 인터리빙 된다. DCT와 양자화가 이루어진 다음 연속 I 프레임에 오류 내성을 위한 패리티 비트가 삽입된다. 블록의 특징은 주변 블록에 삽입된다. 엔트로피 코딩 후 새롭게 생성된 MPEG 스트림이 패킷으로 만들어져 전송된다. 디코더에서 내용 블록 특징의 추출은 삽입과정을 반대로 하면 된다.



(그림 3) 인코더의 오류 내성 및 은닉 처리 흐름도

디코더에서 수신한 비트 스트림은 먼저 역 엔트로피 코딩 된다. 만약 역 엔트로피된 프레임이 I 프레임이고 패리티 비트의 추출을 통해 이전 P 프레임에 움직임 벡터 오류가 있음을 감지하면 정확하게 수신된 다음 I 프레임 또는 P 프레임에서 해당 슬라이스를 확인하여 복원한다.

역 양자화 과정과 역 DCT 과정 후 블록들은 그 다음 원래 순서대로 역 인터리빙되고 손실된 패킷의 I 프레임은 쌍선형 보간법을 이용하여 오류 은닉 처리된다.

5.2 블록 특징 추출, 특징 삽입 및 쌍선형 보간

본 연구에서는 디코더에서 오류 은닉을 실시간으로 빠르게 처리하기 위해 손실 블록의 공간 정보를 인코더에서 추출하여 전송하게 한다. 이 방법은 인코딩시 8x8로 나눈 블록 'A'에서 블록의 경계선 특징을 뽑아 연관 블록의 스트림 내 사용자 데이터로 삽입하여 디코더에 전송한다. 데이터의 삽입 방법은 특징 블록과 연관 블록이 동시에 손상되지 않도록 순환 삽입 방법[19]을 이용하여 블록 'A'의 특징은 'B' 블록에, 'B' 블록의 특징은 'C'에, 'C'의 특징은 'A'블록에 삽입한다. 디코더에서는 전송된 비디오 블록이 손상되었을 경우 협력 블록에 삽입된 블록 'A'의 특징을 이용하여 오류를 은닉한다. 이 때 이미 인코더에서 블록 'A'의 특징이 추출되어 왔기 때문에 손실 블록 'A'의 오류 은닉 계산 부담은 경감한다.

본 연구에서는 오류 은닉 처리에 주변 픽셀에서 손실되거나 손상된 데이터의 지역적 기하 정보를 뽑아내어 오류 블록을 보간하는 쌍선형 보간법[3,4]을 적용한다.

쌍선형 보간법은 두 단계로 이루어진다. 먼저 주변의 정확하게 수신된 블록으로부터 손상된 블록의 경계선 방향과 같은 지역적 기하정보를 예측한다. 그리고 주변의 정확하게 수신된 블록으로 경계선을 따라 손상된 블록을 보간한다.

내용블록의 경계선은 작은 블록에서 직선과 유사하다. 경계선 방향을 결정하기 위해서 로버츠 기울기 연산자를 적용하여 수평, 수직 기울기인 g_x 와 g_y 를 찾아낸다 좌표 (i, j) 에서 기울기 벡터의 크기와 방향은 식 (1)과 식 (2)이다.

$$g(i, j) = \sqrt{g_x^2 + g_y^2} \tag{1}$$

$$\theta(i, j) = \tan^{-1} \left(\frac{g_y}{g_x} \right) \tag{2}$$

픽셀위치 (i, j) 는 만약 $g(i, j)$ 가 임계값 t 를 초과하면 경계선 점이 된다. 경계선 점이 없으면 내용 블록은 경계선이 없는 블록이다. 만약 경계선 블록이면 이 경계선 블록에 대한 경계선 방향은 $0^\circ \sim 180^\circ$ 범위에서 동일한 간격으로 배치된 m 방향이다.

$$q_{md}^0(i, j) = \left\lfloor \frac{\theta(i, j) + \pi / (2m)}{\pi / m} \right\rfloor \tag{3}$$

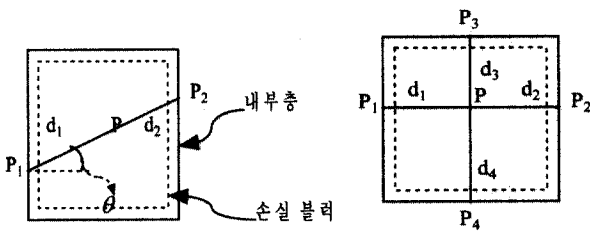
여기서 $q_{ind}^{\theta}(i, j) \in \{0, \dots, m-1\}$ 이다.

데이터의 한정된 삽입 용량 때문에 내용 블록이 한 개 이상의 경계선을 가졌다 해도 한 개의 경계선 값만을 별도 정보로 전송된다. 식 (4)의 다수결 투표 방법을 이용하여 한 개의 경계선을 선택한다.

$$q_{ind}^{\theta} = \{k | \sum_{q_{ind}^{\theta}(p, q) = k} g(p, q) \geq \sum_{q_{ind}^{\theta}(p, q) = l} g(p, q), \quad l = 0, \dots, m-1, l \neq k\} \quad (4)$$

$$q^{\theta} = q_{ind}^{\theta} \times \pi / m + \pi / 2m \quad (5)$$

(p, q) 는 내용 블록에서 경계선 점의 좌표이고 q^{θ} 는 경계선의 각도이며 q_{ind}^{θ} 는 동위 인덱스이다. 내용 블록이 경계선 블록인지 아닌지 1비트로 표시하고 에지 방향 인덱스 q_{ind}^{θ} 는 $b = \lceil \log_2 m \rceil$ 비트로 표시한다. 그래서 한 비트를 에지가 없는 블록에, $1 + b$ 비트를 에지 블록에 할당하여 추출한 내용 블록의 특징을 연관 블록의 DCT AC 계수에 홀짝 삽입법[20]으로 데이터를 숨긴다. 홀짝 삽입법은 표 검색 삽입법의 한 가지로 삽입할 데이터인 '0', '1'과 AC 계수와 같은 미디어 성분 값을 사상하는 검색표를 이용한 데이터 숨김 방법이다. 고주파의 DCT 계수를 변형하면 눈에 띄는 데이터 부산물을 덜 만들기 때문에 추출한 특징 정보를 연관 블록의 고주파 성분에 삽입한다. 만약 디코더에서 손실 블록이 감지되면 이 블록 타입과 경계선 특징은 연관 블록으로부터 추출되고 (그림 4)와 같이 분실 블록을 재현하기 위해 쌍선형 보간을 수행한다. 만약 분실 블록이 경계선이 없는 블록이면 (그림 4)(a)와 같이 픽셀 P의 쌍선형 보간을 수행한다.



(a) 경계선이 없는 블록 (b) 경계선 블록
(그림 4) 쌍선형 보간

$$P = \frac{\left[\begin{array}{l} \frac{d_2}{d_1+d_2} P_1 + \frac{d_1}{d_1+d_2} P_2 \\ + \frac{d_4}{d_3+d_4} P_3 + \frac{d_3}{d_3+d_4} P_4 \end{array} \right]}{2} \quad (6)$$

P_1, P_2, P_3, P_4 는 주변 블록의 내부 계측에서 이웃하는 픽셀이다. 만약 손실 블록이 경계선 블록이면 경계선 방향 θ 에 따라 P의 쌍선형 보간은 (그림 4)(b)와 같이 이루어진다.

$$P = \frac{d_2}{d_1+d_2} P_1 + \frac{d_1}{d_1+d_2} P_2 \quad (7)$$

여기서 P_1, P_2 는 주변 블록의 내부층에서 2개의 최근접 픽셀로부터 선형적으로 보간된다. $d_i = |P - P_i|$ 이다.

6. 실험 및 고찰

6.1 채널 모델

인터넷에서 패킷 손실을 시뮬레이션 하기 위한 방법으로 2개 상태 마코프 모델을 이용한다. 채널은 평균 패킷 손실 확률 (P_i)과 평균 버스트 길이 (L_b)를 설정한다. 비디오는 변동 길이의 패킷으로 만든다. 각 패킷은 재동기 마커가 있는 한 개 매크로 블록 라인으로 구성한다.

인터넷은 2개의 상태로 구성된 마코프 체인으로 모델링 가능하다[21]. 이 모델에서 채널은 두개 상태 중 하나가 되는데 좋은 상태는 "0"으로 나쁜 상태는 "1"로 모델링한다. 상태 "0"에서 패킷은 정확하게 수신되고 상태 "1"에서는 패킷이 손실된다. 모델은 천이 매트릭스 P로 묘사된다.

만약 채널이 n^{th} 시간 단위에서 좋은 상태라면 $S_n = "0"$ 이라 하고 반대의 경우에는 $S_n = "1"$ 이 된다. S_n 은 천이 매트릭스 형태의 이진 마코프 프로세스이다

$$P = \begin{pmatrix} P[S_n=0|S_{n-1}=0] & P[S_n=0|S_{n-1}=1] \\ P[S_n=1|S_{n-1}=0] & P[S_n=1|S_{n-1}=1] \end{pmatrix} = \begin{pmatrix} p_{00} & p_{01} \\ p_{10} & p_{11} \end{pmatrix} \quad (8)$$

마코프 체인은 정적임을 가정한다. 평균 버스트 손실 확률을 식 (9)로 표현하고

$$P_i = \frac{p_{0i}}{p_{10} + p_{0i}} \quad (9)$$

평균 버스트 길이는 식 (10)로 채널 조건을 만든다.

$$L_b = \frac{1}{p_{10}} \quad (10)$$

6.2 실험

'football'과 'flower garden' 비디오로 실험 결과를 나타낸다. 비디오는 초당 10프레임으로 QSIF(176×112) 크기의 MPEG-1으로 인코딩했다. GOP 프레임은 IPPP이고 오류 확산에 영향이 없는 2B 프레임은 건너뛰었다. 움직임벡터는 ±15 픽셀의 크기 내이다. 비트 전송율은 1Mbps이다.

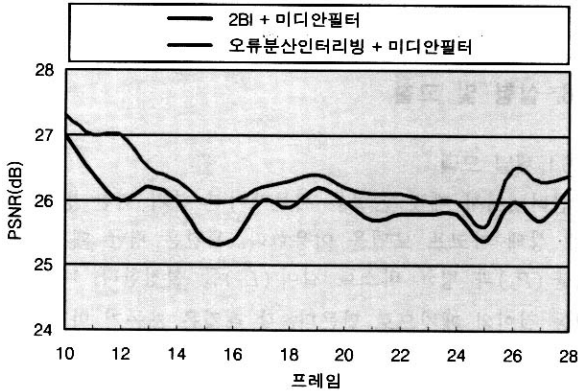
이미지 품질은 식 (11)에서 정의한 PSNR로 나타낸다.

$$PSNR = 10 \log_{10} \frac{\text{이미지 크기} \times 255^2}{\sum_{i=1}^{rowscale} \sum_{j=1}^{colscale} (t(i, j) - f(i, j))^2} \quad (11)$$

$t(i, j)$ 와 $f(i, j)$ 는 인코딩 전과 디코딩 후의 픽셀의 휘

도를 나타낸다.

(그림 5)는 공간적 오류 내성을 위해 본 연구에서 제안한 오류 분산 인터리빙과 2BI 인터리빙 방법을 비교한 결과이다.

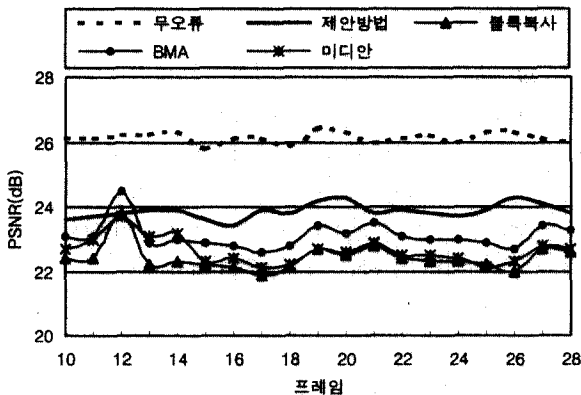


(그림 5) 오류분산 인터리빙과 2BI인터리빙 비교

'football' 비디오 이미지에 오류 내성 방법은 달리하면서 동일한 오류 은닉 기법인 미디안 필터를 이용하여 발생 오류를 은닉한 결과 오류 분산 인터리빙을 적용한 방법의 PSNR이 더 높음을 알 수 있다.

2BI 인터리빙 알고리즘의 복잡도는 $O(n)$ 이고 제안 오류 분산 인터리빙의 복잡도는 $O(n^2)$ 이지만 오류 분산 인터리빙 적용에 의해 이미지 블록의 오류가 덜 발생하여 실제 비디오 품질이 향상되며 인터리빙은 제안 오류은닉 방법의 일부이기 때문에 전체 시스템의 복잡도에는 큰 영향을 미치지 않는다.

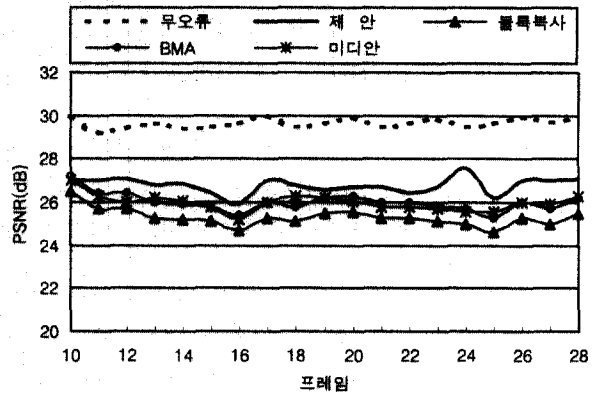
인코더에서 시간적, 공간적인 오류를 보정 처리하고 데이터 숨김을 이용, 블록의 특징을 디코더로 전달하여 쌍선형 보간을 통해 오류 은닉을 처리한 본 연구의 제안 방법을 실험 비디오에 적용하여 오류가 발생하지 않은 무오류상태 및 BMA(Boundary Matching Algorithm), 블록 복사, 미디안 필터 등의 기존 오류 은닉 방법과 비교하였다.



(그림 6) 'flower garden'의 오류 은닉 품질

(그림 6)은 'flower garden' 비디오에 발생한 오류를 은닉

한 결과이고 (그림 7)은 'football' 비디오에 적용한 결과이다. 제안 방법의 PSNR 결과가 더 우수함을 알 수 있다.



(그림 7) 'football'의 오류 은닉 품질

<표 1>과 <표 2>는 실험 비디오에 적용한 각 오류 은닉 방법의 평균 PSNR이다. 제안 방법이 타 오류 은닉 방법보다 높은 PSNR을 보인다. <표 1>과 <표 2>의 무오류 은닉은 오류가 발생했지만 오류 은닉을 하지 않은 결과이다.

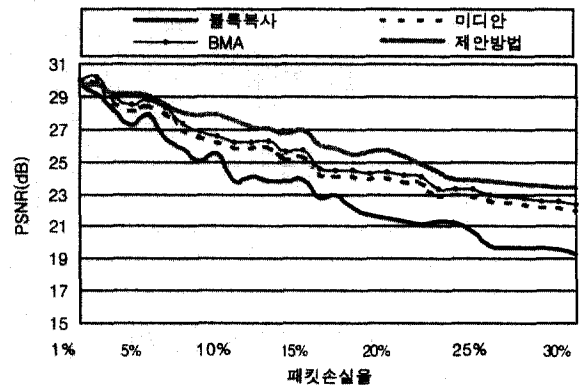
<표 1> 오류 은닉한 'flower garden' 이미지의 평균 PSNR

네트워크	무오류	제안 방법	BMA	미디안 필터	블록 복사	무오류 은닉
1Mbps	26.011	23.949	22.794	22.692	22.374	12.213

<표 2> 오류 은닉한 'football' 이미지의 평균 PSNR

네트워크	무오류	제안 방법	BMA	블록 복사	미디안 필터	무오류 은닉
1Mbps	29.575	26.394	25.742	25.372	25.121	13.817

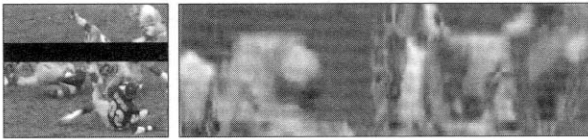
(그림 8)은 'football' 비디오에 대하여 인터넷에서 발생하는 버스트 오류 길이가 2일때 패킷 손실율에 따른 오류 은닉 방법을 비교했다. 실험결과에서 제안 방법이 블록 복사에 대해 평균 3.37dB, 미디안에 대해 평균 1.63dB, BMA에 대해 1.23dB의 높은 PSNR을 보인다.



(그림 8) 패킷 손실율에 따른 PSNR

오류 은닉이 수행된 각 영상 차이는 그림 9에서 확인할 수 있다. 대상 영상은 'football' 비디오의 I 프레임 중 3번째 프레임이 손실되었을 때 임의로 생성한 손실 블록의 오류 패턴이다.

(그림 9)(a)는 오류 패턴에 블록 복사를 적용한 결과이다. (그림 9)(b)는 BMA를 적용했으며 (그림 9)(c)는 제안 방법을 적용했다.



(a) 오류 패턴과 블록 복사를 적용한 후의 내용 확대

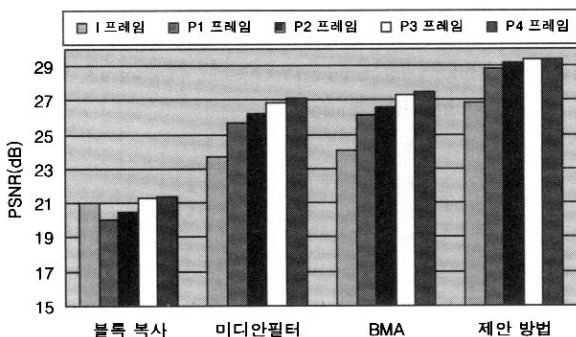


(b) 오류 패턴과 미디안필터를 적용한 후의 내용 확대



(c) 오류 패턴과 제안 방법을 적용한 후의 내용 확대
(그림 9) football 비디오의 오류와 은닉처리

(그림 10)은 재현된 I 프레임의 PSNR과 P 프레임에서 아무런 오류가 발생하지 않았다는 가정 아래 I 프레임에 종속된 4개 P 프레임의 평균 PSNR이다. 제안 방법의 공간적 오류 내성 결과로 I 프레임에서 흩어진 오류는 오류 은닉 품질을 개선할 뿐만 아니라 다른 종속 프레임에 대한 오류 파급 효과를 줄인다. 시간적 오류 내성 방법인 움직임 벡터의 확인 패리티 비트를 이용함으로써 빠른 오류 감지가 가능하고 오류 블록을 보정하기 때문에 PSNR이 높아진다. 연속적 I 프레임에 데이터 삽입 여력이 있다면 더 강력한 오류 수정 코드를 적용할 수 있다.



(그림 10) 오류 내성 처리한 I 프레임과 연속 P 프레임의 PSNR

제안 오류 은닉 방법은 오류 보정과 데이터 숨김을 적용했기 때문에 전체 데이터 비트율은 제안 방법이 1.02%로 증가했으나 이 증가량은 계산량에는 큰 영향을 미치지 않으므로 무시할 수 있다. 제안 방법의 디코더에서의 계산량은 이미 인코더에서 데이터 숨김으로 전송되어 온 블록의 특징 정보로 인해 디코더에서 주변 블록의 정보를 이용하여 오류 은닉 계산을 하는 BMA보다 약 20% 줄었다. 이에 따라 오류 내성 방법과 데이터 숨김을 적용하는데 따르는 계산 복잡도는 디코더에서의 쌍선형 보간 처리시 계산량이 줄어든 효과로 상쇄되고 동시에 비디오 데이터의 품질은 오류의 확산, 빠른 오류 감지등을 통해 훨씬 높아지는 효과를 누리게 된다.

7. 결 론

본 연구에서는 인터넷을 통해 비디오 데이터 전송시 발생 하는 버스트 오류에 대한 오류 은닉 효과를 높이기 위해 인코딩 시 오류 내성 기법을 적용하고 블록의 특징 정보를 데이터 숨김을 통해 디코더에 전송하는 방법을 제안했다.

제안 오류 분산 인터리빙을 적용하면 더 우수한 PSNR 효과를 누릴 뿐 아니라 I 프레임에서 발생한 오류가 연속 P 프레임에 확산되는 문제를 방지함을 실험 결과를 통해 확인했다. 또한 시간 오류 내성 기법인 움직임 벡터를 확인하는 패리티 비트를 이용하여 오류가 발생한 P 프레임을 빠르게 감지했다.

데이터 숨김을 통해 인코더에서 디코더로 전송된 블록의 경계선 특징을 이용하여 오류 은닉을 처리하면 디코더의 계산 부담을 줄이기 때문에 실시간으로 처리되는 오류 은닉의 복잡도가 미디안 필터 및 BMA과 비교하여 약 20% 감소했다.

움직임 벡터의 패리티 정보와 인코더에서 추출한 데이터 블록의 특징은 데이터 숨김 방법을 통해 전송하였기 때문에 전체 데이터 양의 증가에는 큰 영향을 미치지 않으면서도 디코더의 부담을 크게 줄인 것이다. 따라서 제안 오류 보정 방법과 데이터 숨김을 이용하여 오류 은닉하면 오류의 확산을 줄이는 동시에 빠른 오류 감지와 은닉으로 높은 비디오 품질을 보장하게 된다.

실험 결과는 제안 방법을 적용하면 비디오의 패킷 손실율이 25~30%가 되어도 높은 비디오 품질로 복원되며 인터넷의 버스트 오류에 대해서도 강건함을 보여준다. 또한 MPEG-1 표준 인코더/디코더 구조를 이용함으로써 제안 방법이 표준 적합성을 있음을 알 수 있다.

향후 연구에서는 다양한 MPEG 비디오 데이터에 적용하여 신뢰성을 높일 계획이며 인코더에서 블록의 특징 정보를 추출하여 디코더로 전송하면서도 데이터 품질은 유지하는 데이터 숨김 방법에 대한 연구를 계속할 것이다.

참 고 문 헌

- [1] Jill M. Boyce and Robert D. Gaglianella, "Packet Loss Effects on MPEG Video Sent Over the Public Internet," ACM Multimedia '98, 1998.
- [2] U. Horn, K. Stuhlmuller, M. Link and B. Girod, "Robust Internet Video Transmission based on Scalable Coding and Unequal Error Protection," Image Communication, Sept., 1999.
- [3] Wenjun Zeng and Bede Liu, "Geometric Structure-Based Error Concealment with Novel Application in Block-Based Low Bit Rate Coding," IEEE Transaction on CSVT, Vol.9, No.4, pp.648-665, June, 1999.
- [4] H. Sun and W. Kwok, "Concealment of Damaged Block Transform Coded Images Using Projections Onto Convex Sets," IEEE Transactions on Image Processing, Vol.4, No. 4, pp.470-477, April, 1995.
- [5] W. Lam, A. Reibman and B. Liu, "Recovery of Lost or Erroneously Received Motion Vectors," ICASSP '93, pp.304-315, 1993.
- [6] 김진옥, 황대준, "인터넷상에서 비디오데이터 전송시 오류내성에 관한 연구", 2002 추계 한국정보과학회학술집, 제29권 제2호, pp.481-483, 2002.
- [7] Yao Wang and Qin Fan Zhu, "Error Control and Concealment for Video Communication : an Overview," Proceedings of IEEE, Vol.86, No.5, pp.974-997, May, 1998.
- [8] S. Aign and K. Fazel, "Temporal and Spatial Error Concealment Techniques for Hierarchical MPEG-2 Video Codec," Globecom '95, Vol.3, pp.1778-1783, 1995.
- [9] X. Lee, Y. Zhang and A. Leon-Garcia, "Information Loss Recovery for Block-Based Image Coding Techniques-a Fuzzy Logic Approach," IEEE Trans. on Image Processing, Vol.4, No.3, pp.259-273, March, 1995.
- [10] ISO/IEC 14496-2, "Information Technology-Coding of Audio/Visual Objects," Part 2 : Visual, 1999.
- [11] Corinne Lebuhan, "Software-Embedded Data Retrieval and Error Concealment Schemes for MPEG-2 Video Sequences," in Digital Video Compression : Algorithms and Technologies, Proceedings of the SPIE, Vol.2668, Jan., 1996.
- [12] W. Bender, D. Gruhl, N. Morimoto and A. Lu, "Techniques for Data Hiding," IBM Systems Journal, Vol.35, No.3/4, pp.313-336, 1996.
- [13] I. Cox, J. Kilian, T. Leighton, T. Shamoan, "Secure Spread Spectrum Watermarking for Multimedia," IEEE Trans. On Image Processing, Vol.6, No.12, 1997.
- [14] Gerhard Langelaar, Iwan Setawan and Reginald Lagendijk, "Watermarking Digital Image and Video Data, IEEE Signal Processing," Vol.17, No.5, pp.20-46, Sept., 2000.
- [15] S. Gringeri, R. Egorov, K. Shuaib, A. Lewis and B. Basch, "Robust Compression and Transmission of MPEG-4 Video," in ACM MM2000 Electronic Proceedings, June, 2000.
- [16] King Ip Chan, Jianhua Lu and Justin C. I. Chung, "Block Shuffling and Adaptive Interleaving for Still Image Transmission over Rayless Fading Channels," ITVT, Vol.48, No.3, May, 1999.
- [17] Qin Fan Zhu, Yao Wang and Loenard Shaw, "Coding and Cell-Loss Recovery in DCT-Based Packet Video," IEEE Transactions on CSVT, Vol.3, No.3, June, 1993.
- [18] Jie Song and K. J. R. Liu, "A Data Embedding Scheme for H.263 Compatible Video Coding," ISCAS '99, Vol.4, 1999.
- [19] J. W. Park, J. W. Kim and S. U. Lee, "DCT Coefficients Recovery-Based Technique and its Application to the MPEG-2 Bit Stream Error," IEEE Transaction on CSVT, Vol.7, No.6, pp.845-854, 1997.
- [20] D. Mukherjee, J. J. Chae, J. K. Mitr and B. S. Manjunath, "A Source and Channel Coding Framework for Vector-Based Data Hiding in Video," IEEE Transactions on CSVT, Vol.1, No.4, pp.630-645, June, 2000.
- [21] L. N. Kanal and A. R. K. Sastry, "Models for Channels with Memory and Their Applications to Error Control," Proceedings of the IEEE, Vol.66, No.7, pp.724-244, 1978.

김진옥



e-mail : jokim@sejong.ac.kr
 1989년 성균관대학교(문학사)
 1998년 성균관대학교 정보통신공학과 (공학석사)
 2002년 성균관대학교 전기전자 및 컴퓨터 공학과(공학박사)

1992년~1994년 (주)현대전자산업 정보통신사업본부
 1994년~1999년 (주)현대정보기술 인터넷사업본부 과장
 1999년~2000년 (주)은세통신 온라인사업 팀장
 2000년~2001년 (주)유로코넷 기술담당 이사
 2003년~현재 세종대학교 컴퓨터공학과 초빙교수
 관심분야 : Multimedia, Image Processing, Biometrics, Data mining, Recognition