

MPEG-2의 GOP 구조를 이용한 I 프레임의 시공간적 오류 은닉

정회원 강민정*, 류철**

Spatio-Temporal Error Concealment of I-frame using GOP structure of MPEG-2

Min-Jung Kang*, Chul Ryu** *Regular Members*

요 약

본 논문은 MPEG-2 인트라 (I) 프레임의 더욱 강화된 오류 은닉 기술을 제안한다. MPEG-2 소스 부호화 알고리즘은 가변장 부호화를 사용하기 때문에 채널 오류에 매우 민감하다. 채널에서 발생된 전송 오류는 오류 정정 기술을 사용하여 보정되지만 제대로 보정되지 않은 오류는 디코더에서 오류 은닉 기술을 사용함으로써 시각적 왜곡이 최소화될 수 있다. 또한 GOP의 시작 프레임인 I 프레임의 오류는 움직임 보상 예측 부호화 방식으로 인하여 다른 인터 프레임에 전달되어 더욱 심각한 화질 저하를 가져온다. 본 논문에서 제안하는 오류 은닉 방식은 I 프레임에 발생된 연속적인 슬라이스 오류를 인접한 인터 프레임의 시공간적 정보를 이용하여 기존 방식보다 효과적으로 처리하는 방식이다. 이 방식은 기존의 방식들이 갖는 시간적 그리고 공간적에서 발생하는 단점을 효율적으로 개선할 수 있으며, 전송 오류가 심한 망에서 발생하는 극심한 슬라이스 오류에서도 좋은 시각적 효과를 얻을 수 있다. 알고리즘은 MPEG-2 비디오 코덱 범위에서 수행되었으며 모의실험을 통하여 제안된 방식이 다른 방식들보다 높은 PSNR과 개선된 시각적 효과를 확인할 수 있었다.

Key Words : Error Concealment, Error Control, Error Detection, MPEG-2

ABSTRACT

This paper proposes more robust error concealment techniques (ECTs) for MPEG-2 intra coded frame. MPEG-2 source coding algorithm is very sensitive to transmission errors due to the use of variable-length coding. The transmission errors are corrected by error correction scheme, however, they cannot be revised properly. Error concealment (EC) is used to conceal the errors which are not corrected by error correction and to provide minimum visual distortion at the decoder. If errors are generated in intra coded frame, that is the starting frame of GOP, they are propagated to other inter coded frames due to the nature of motion compensated prediction coding. Such propagation of error may cause severe visual distortion. The proposed algorithm in this paper utilizes the spatio-temporal information of neighboring inter coded frames to conceal the successive slices errors occurred in I-frame. The proposed method also overcomes the problems that previous ECTs reside. The proposed algorithm generates consistent performance even in network where the violent transmission errors frequently occur. Algorithm is performed in MPEG-2 video codec and we can confirm that the proposed algorithm provides less visible distortion and higher PSNR than other approaches through simulations.

* 동국대학교 정보통신공학과 (minjong@dgu.ac.kr)
논문번호 : 030255-0613, 접수일자 : 2003년 6월 13일

** 동국대학교 정보통신공학과 (cryu@dgu.ac.kr)

I. 서론

최근 정보화 사회로의 급속한 발전에 따른 여러 가지 비디오에 관련된 멀티미디어 응용의 증가로 통신망을 통하여 전송되는 멀티미디어 데이터들이 급격히 증가하고 있다. 특히 몇몇 디지털화된 비디오 데이터는 매우 높은 데이터율을 요구하므로 실제 사용이 가능한 채널의 대역폭을 통하여 전송할 수 있는 수준으로 데이터를 압축하는 것이 매우 중요하며, MPEG-2는 이러한 비디오 압축 알고리즘의 대표적인 예이다^[1]. 가변장 부호화로 압축된 MPEG-2 비디오 데이터는 전송 채널에 매우 민감하여 채널의 잡음이나 물리적인 채널의 손상과 같은 전송 오류로 인하여 데이터가 손실 또는 손상될 수 있는 위험성을 안고 있다. 물리적 채널의 결함에 의하여 발생하는 랜덤 비트 오류는 고정 길이 부호화의 경우 하나의 코드워드에만 영향을 주지만 가변장 부호화의 경우 각 코드워드의 길이가 다르기 때문에 정보의 동기를 잃는 원인이 되기도 한다. 패킷 망에서의 패킷 손실이나 물리적 결함에 의한 저장 미디어에서의 버스트 오류 또는 짧은 시간 동안의 시스템 결함에 의해 발생하는 삭제 오류는 비트의 연속적인 세그먼트 손실로 인하여 랜덤 비트 오류보다 훨씬 더 심각한 데이터의 손상을 유발시킨다^[2].

비디오 통신에서 발생이 가능한 다양한 전송 오류에 의하여 손상된 데이터는 다음과 같은 이유로 반드시 오류가 제어되어야 한다. 첫째, MPEG-2로 압축된 비디오 데이터는 가변장 부호화를 사용하기 때문에 전송 오류에 매우 민감하며, 예측 부호화에 의한 시공간적 예측 오류로 인하여 잘못된 데이터를 복원하게 된다. 이렇게 복원된 데이터는 현재 프레임뿐만 아니라 다음 프레임에도 영향을 미치게 된다. 둘째, 비디오 신호와 채널 상태는 시간적으로 계속 변하므로 통계적 모델을 기반으로 한 최적의 해결책을 찾는 것은 매우 어렵거나 불가능하다. 셋째, 매우 높은 데이터율을 요구하는 비디오 데이터를 실시간으로 처리하기 위해서는 인코더와 디코더가 너무 복잡해서는 안된다. 그리고 비디오 통신의 전송 오류는 디코더에서 시각적인 왜곡으로 표현되기 때문에 전송 오류의 제어는 더욱 필수적이다^[3].

전송 오류를 처리하기 위한 가장 간단한 방법은 손실된 데이터를 재전송 하는 것이다. 재전송 기술은 일반적으로 데이터의 무손실 전송이 가능하지만 몇 가지의 제약이 따른다^[4]. 만약 전송 채널의 혼잡

에 의하여 데이터 손실이 발생된 경우 재전송 기술은 더욱 심각한 채널의 혼잡을 일으키게 되고 더 많은 데이터 손실을 발생시킬 수 있다. 예를 들어, 이동 통신이나 무선 통신 채널에서 송수신측간의 연결이 약한 경우 손실된 데이터의 재전송은 채널의 부하를 증가시키며 이로 인한 데이터의 충돌 가능성과 전송 지연이 가중됨으로써 더 많은 데이터 손실을 유발시킬 수 있다. 또한 송신측에 손실된 데이터의 재전송을 요구하고 그에 따른 데이터를 재전송하기 까지 전송 지연이 발생되며 실시간 데이터 처리를 요구하는 실시간 방송의 경우 허용할 수 없는 지연을 발생시킬 수 있다^[5-7]. 그러므로 비디오 통신에서의 전송 오류를 보다 효율적으로 처리하기 위한 오류 제어 방식들이 필요하다.

전송 오류를 처리하기 위한 방식은 인코더에 의한 오류 반발 부호화, 디코더에 의한 오류 은닉 그리고 인코더와 디코더의 상호작용적인 오류 제어 등 3가지로 나누어 볼 수 있다^[8]. 본 논문에서는 디코더에서 전송 오류를 처리하는 여러 가지 오류 은닉 방식의 동작 원리와 장단점을 살펴보고, 시공간적 정보를 이용한 새로운 방식을 제안한다. 본 논문의 구성은 다음과 같다. 제 II장에서 전송 오류를 검출하는 방식들을 살펴본 후 제 III장에서 현재 사용되고 있는 오류 은닉 방식들을 소개한다. 제 IV장은 기존의 오류 은닉 방식의 단점을 보완하기 위한 새로운 시공간적 정보를 이용한 오류 은닉 방식을 제안하고, 제 V장과 VI장에서 실험 결과를 통한 제안된 방식의 성능 분석 및 장단점이 설명된다.

II. 전송 오류 검출 방식

오류 은닉 방식을 적용하기 위하여 먼저 채널에서 발생한 전송 오류가 비디오 데이터의 어느 부분에 발생하였는지에 대한 위치를 파악해야 한다. 전송 오류를 검출하는 방식에는 소스 복호화에 의한 오류 검출 방식과 채널 복호화에 의한 오류 검출 방식이 있다^[6].

1. 소스 복호화에 의한 오류 검출 방식

가변장 부호화는 데이터의 발생 빈도에 따라 부호화 길이를 다르게 하는 방식으로 가변장 부호화 표에 따라 각각의 코드워드를 생성한다. 그러나 가변장 부호화된 비디오 데이터가 채널 오류에 의하여 손실된 경우 몇몇 코드워드는 가변장 부호화의 표와 일치하지 않아 복호화할 수 없게 된다. 또한 부호화

길이 일정하지 않아 잃어버린 동기를 다시 찾을 수 없게 됨으로써 다음 코드워드들을 복호화할 수 없다. 이러한 경우, 소스 복호화에 의한 오류 검출 방식은 코드워드를 오류가 발생한 것으로 판단하여 오류를 검출한다. 그러나 오류가 발생한 코드워드가 가변장 부호화 표의 다른 코드워드와 일치하는 경우 그 코드워드는 오류가 발생하지 않은 것으로 간주되어 다음 동기화 지점까지 오류가 전달되는 결과를 초래할 수 있다.

2. 채널 복호화에 의한 오류 검출 방식

채널 복호화에 의한 오류 검출 방식은 패킷의 헤더 정보를 이용한다. 패킷 기반 비디오 통신에서 각각의 패킷은 자신의 헤더 정보를 포함하며 헤더에는 패킷이 전송된 순서를 나타내는 고유 시퀀스 번호가 포함되어 있다. 즉, 복호기에서 수신된 패킷을 처리할 때 이 시퀀스 번호를 사용하여 몇 번째 패킷이 손실되었는지를 정확히 파악할 수 있다. 또한 패킷의 헤더에 손실된 패킷의 정보를 표시하는 방식이 있다. 이 방식은 오류가 발생한 패킷의 정보가 헤더에 포함됨으로써 보다 정확한 오류의 위치를 검출할 수 있다. 일반적으로 채널 복호화에 의한 오류 검출 방식은 추가적인 헤더 정보로 인하여 더 많은 데이터의 전송이 필요하지만 소스 복호화에 의한 오류 검출 방식에 비하여 높은 신뢰성과 효율성을 갖는다.

본 논문에서는 오류 검출 방식은 고려하지 않고 오류 은닉 방식에 초점을 맞추었다. 따라서 임의로 특정 매크로블록에 오류를 삽입함으로써 디코더에서 오류가 발생한 위치를 알고 있고 이미 오류 검출이 되었다는 가정 하에 오류 은닉 방식을 적용한다.

III. 오류 은닉 방식

오류 은닉 방식은 채널을 통하여 비디오 데이터가 전송되는 동안 발생된 오류를 오류 정정 기술로 보정할 수 없는 경우 영상의 공간적 중복성과 시간적 중복성을 이용하여 영상의 화질 저하를 최소화하기 위하여 적용되는 방식이다. 프레임내 공간적 중복성을 이용하는 공간적 오류 은닉 방식은 움직임 정보가 없는 I 프레임에 주로 사용된다. 반면 프레임간 시간적 중복성을 이용하는 시간적 오류 은닉 방식은 움직임 정보를 갖는 P 또는 B 프레임에 주로 사용되며 제안된 알고리즘에 따라 I 프레임에도 적용이 가능하다. 또한 공간적 중복성과 시간적 중복성을

모두 이용하는 혼합 오류 은닉 방식은 프레임과 매크로블록의 부호화 모드에 따라 시간적 오류 은닉 방식과 공간적 오류 은닉 방식을 선택적으로 사용하며 모든 프레임에 적용이 가능하다.

1. 시간적 오류 은닉 방식

시간적 오류 은닉 방식은 영상에 존재하는 시간적 중복성을 이용하는 방식으로 단순 오류 은닉 방식, 움직임 보상 오류 은닉 방식 그리고 로그 검색 기반의 순방향-역방향 블록 정합 오류 은닉 방식 등으로 나누어 볼 수 있다.

단순 오류 은닉 방식은 프레임간의 움직임 정보를 이용하지 않고 오류가 발생한 매크로블록을 이전 앵커 프레임에서 이에 상응하는 매크로블록으로 대체하는 방식으로 P 또는 B 프레임 외에 움직임 정보를 갖지 않는 I 프레임에도 적용될 수 있다^[7]. 이 방식은 구현이 간단하지만 현재 프레임과 이전 앵커 프레임간의 움직임 차이로 인한 왜곡 현상이 발생할 수 있다. 단순 오류 은닉 방식의 움직임 왜곡 현상을 보완한 움직임 보상 오류 은닉 방식은 오류가 발생한 매크로블록의 주변 매크로블록이 갖는 움직임 벡터를 사용하여 이전 앵커 프레임에서 움직임 벡터에 준하는 매크로블록으로 대체하는 방식이다^[8]. 이 방식은 움직임 보상으로 인하여 단순 오류 은닉 방식보다 효과적이거나 주변 매크로블록의 움직임 벡터가 손상되거나 부정확한 경우 정확하지 않은 움직임 예측으로 인한 왜곡이 발생할 수 있다. 또한 움직임 정보를 이용해야 하므로 I 프레임에 적용될 수 없다.

로그 검색 기반의 순방향-역방향 블록 정합 오류 은닉 방식은 현재 프레임에서 오류가 발생한 매크로블록의 위 또는 아래쪽에 위치한 매크로블록의 움직임 벡터를 이용한다^[9]. 이 때 사용되는 움직임 벡터는 로그 검색 기반의 블록 정합 알고리즘에 의하여 얻어진 것으로 검색 영역에서 최소 절대 평균 편차를 갖는 참조 매크로블록과 현재 매크로블록간의 움직임 차이값이다. 만약 위쪽 매크로블록의 움직임 벡터가 존재할 경우 이 매크로블록을 움직임 벡터만큼 이동시킨 매크로블록의 바로 아래쪽 16×16 픽셀을 현재 프레임의 오류가 발생한 매크로블록으로 복사하여 오류를 은닉한다. 또한 아래쪽 매크로블록의 움직임 벡터가 존재할 경우에는 움직임 벡터만큼 이동시킨 매크로블록의 바로 위쪽 16×16 픽셀을 복사하여 오류를 은닉할 수 있으며, 위와 아래쪽 2개 매크로블록의 움직임 벡터가 존재할 경우에는 그

움직임 벡터를 혼합적으로 사용할 수 있다. 이 방식은 비디오 압축 알고리즘에 의하여 움직임 정보를 갖는 P 또는 B 프레임에 사용이 가능하며 I 프레임에도 적용 가능하다. I 프레임은 비디오 압축 알고리즘에서 제공하는 기본적인 정보 외에 별도의 블록 정합 알고리즘을 이용하여 이미 복호된 P 프레임으로부터의 움직임 벡터를 구함으로써 오류를 은닉할 수 있다.

2. 공간적 오류 은닉 방식

공간적 오류 은닉 방식은 주로 움직임 정보가 없는 I 프레임에 사용되며 오류가 발생한 매크로블록을 서브 블록 단위로 은닉하는 방식과 매크로블록 단위로 은닉하는 방식이 있다¹⁰⁾. 서브 블록 단위의 오류 은닉 방식은 오류가 발생한 매크로블록을 구성하는 네 개의 서브 블록과 인접한 블록들을 이용하는 방식이다. 이 방식은 인접한 블록의 사용이 가능할 때 좋은 결과를 얻을 수 있지만 주변 블록이 손상되었을 경우에는 오류를 제대로 은닉할 수 없다. 예를 들어 연속적인 매크로블록 오류가 발생한 경우 매크로블록과 인접한 주변 4개의 매크로블록을 구성하는 서브 블록이 이용되는 것이 아니라 위 또는 아래쪽에 위치한 2개 매크로블록의 서브 블록이 오류 은닉에 사용되기 때문에 오류를 제대로 은닉할 수 없다. 매크로블록 단위의 오류 은닉 방식은 비디오 전송 단위인 슬라이스에 오류가 발생한 경우 많이 사용되며, 오류가 발생한 매크로블록의 주변 네 개 매크로블록을 이용한다. 만약 슬라이스 오류로 인하여 여러 개의 연속적인 매크로블록이 손상되었을 경우 오류가 발생한 매크로블록의 위와 아래쪽에 위치한 매크로블록을 사용하여 오류를 은닉하므로 서브 블록 단위의 오류 은닉 방식보다 좋은 결과를 얻을 수 있지만 가중 평균으로 인한 번짐 현상이 발생하게 된다. 또한 하나의 슬라이스가 아닌 연속적인 슬라이스 오류가 발생한 경우에는 오류 은닉에 사용되는 매크로블록의 수가 줄어들어 서브 블록 단위의 오류 은닉 방식과 마찬가지로 오류를 제대로 은닉할 수 없다.

3. 혼합 오류 은닉 방식

혼합 오류 은닉 방식은 프레임과 매크로블록의 부호화 모드에 따라 시간적 오류 은닉 방식과 공간적 오류 은닉 방식을 선택적으로 적용하는 방식이다¹¹⁾. I 프레임에 오류가 발생된 경우 프레임 내의 모든 매크로블록은 인트라 모드로 부호화되기 때문에 오

류가 발생한 매크로블록은 공간적 오류 은닉 방식으로 처리된다. 그러나 P 또는 B 프레임 내의 오류가 발생한 매크로블록이 인터 모드로 부호화된 경우에는 시간적 오류 은닉 방식이 사용되며 인트라 모드로 부호화된 경우에는 움직임 벡터가 존재하지 않기 때문에 공간적 오류 은닉 방식이 적용된다.

IV. 제안한 시공간적 정보를 이용한 오류 은닉 방식

비디오 신호가 채널을 통하여 전송될 때 여러 가지 원인에 의하여 오류가 발생하게 된다. 특히 심한 슬라이스 오류가 발생한 경우 GOP의 시작 프레임인 I 프레임의 오류는 움직임 보상 예측 부호화 방식에 의하여 부호화되는 다른 프레임에 그 오류가 전달되어 GOP 전체의 화질에 영향을 미치므로 I 프레임에서 발생된 오류의 은닉은 매우 중요하다. 제 III장에서 설명한 여러 가지 오류 은닉 방식 중 서브 블록 단위와 매크로블록 단위의 공간적 오류 은닉 방식, 단순 시간적 오류 은닉 방식, 로그 검색 기반의 순방향-역방향 블록 정합 오류 은닉 방식 그리고 혼합 오류 은닉 방식 등이 I 프레임에 적용 가능하지만 효율적으로 오류를 은닉하지는 못한다. 예를 들면, 공간적 오류 은닉 방식을 이용하여 I 프레임의 슬라이스 오류를 처리할 경우 번짐 현상이 발생할 수 있으며, 단순 시간적 오류 은닉 방식의 경우 움직임 왜곡 현상이 발생하게 된다. 혼합 오류 은닉 방식의 경우 I 프레임의 모든 매크로블록은 인트라 모드로 부호화됨에 따라 공간적 오류 은닉 방식과 동일한 결과를 나타낸다. 또한 로그 검색 기반의 순방향-역방향 블록 정합 오류 은닉 방식을 이용할 경우 I 프레임에 사용되는 움직임 벡터를 MPEG-2 비디오 압축 알고리즘 외에 별도의 로그 검색 기반의 블록 정합 알고리즘을 이용하여 이전에 복호된 P 프레임으로부터 구해야 하며 이에 따라 기본 비디오 압축 정보 외에 추가적인 정보를 전송해야 한다.

본 논문은 I 프레임의 연속적인 슬라이스 오류를 추가 정보없이 이전 P와 B 프레임의 시공간적 정보만을 이용하여 은닉하는 기법을 제안한다. 제안하는 방식을 성공적으로 수행하기 위하여 I 프레임 외의 다른 프레임의 정보 손실은 없다고 가정한다. 움직임 정보를 갖는 P 또는 B 프레임은 블록 정합 알고리즘을 기반으로 한 움직임 보상 예측 부호화 방식

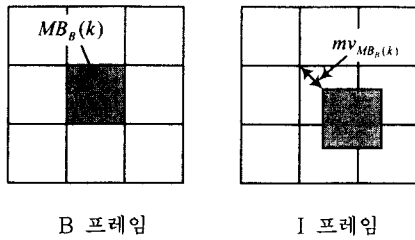


그림 1-1. $MB_B(k)$ 가 부호화될 때 사용되는 $mv_{MB_B(k)}$ 와 이에 상응하는 I 프레임에서의 매크로블록

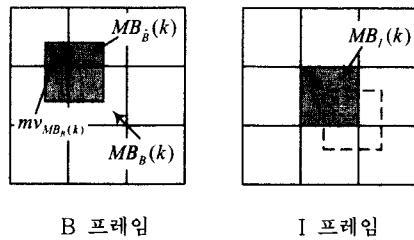


그림 1-2. 블록 정합 알고리즘의 선형성을 이용하여 I 프레임의 매크로블록 위치를 B 프레임에서 파악

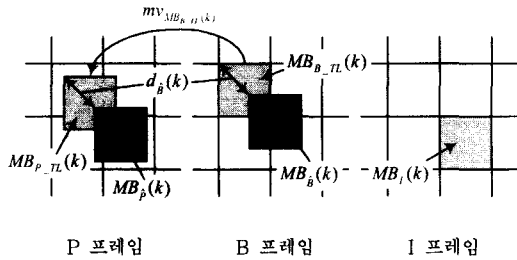
으로 부호화된다. P 프레임은 순방향 움직임 벡터를 사용하여 이전 I 또는 P 프레임으로부터 복원이 가능하고, B 프레임은 순방향 움직임 벡터와 역방향 움직임 벡터를 사용하여 이전 I 또는 P 프레임 그리고 다음 I 또는 P 프레임으로부터 복원이 가능하다. 이러한 움직임 보상 예측 부호화 방식은 GOP의 시작 프레임인 I 프레임의 오류가 인터 프레임에 영향을 미치게 되어 결과적으로 GOP 전체에 오류가 발생하게 된다. 따라서 I 프레임에서 발생한 오류는 전체 출력 데이터의 화질에 중요한 영향을 미치므로 I 프레임에서 발생한 오류를 은닉하는 방법이 시각적 효과를 높이는데 필요한 중요한 기법 중 하나임에 틀림없다. I 프레임에 오류가 발생한 k 번째 매크로블록을 $MB_I(k)$ 라 하고 이전 GOP의 마지막 B 프레임에서 $MB_I(k)$ 와 동일한 좌표에 위치한 매크로블록을 $MB_B(k)$ 라 하자. B 프레임의 각 매크로블록은 블록 정합 알고리즘에 의한 역방향 움직임 벡터를 사용하여 부호화되며 그림 1은 블록 정합 알고리즘에 의하여 얻어진 $MB_B(k)$ 의 역방향 움직임 벡터 $mv_{MB_B(k)}$ 의 위치를 나타낸다. 일반적으로 블록 정합 알고리즘은 프레임간의 움직임이 선형적이라는 가정 하에 이루어지며 이러한 특성을 이용하여 B 프레임의 부호화 정보인 양방향 움직임 벡터 중 I 프레임으로부터 얻어진 역방향 움직임 벡터를 역으로 이용하여 I 프레임의 매크로블록 위치를 이전 B 프레임에서 파악할 수 있다. 즉, 그림 1-2와 같이 오류가 발생한 $MB_I(k)$ 를 그와 동일한 위치에 있는 매크로블록 $MB_B(k)$ 의 역방향 움직임 벡터만큼 이동함으로써 식 (1)과 같이 $MB_I(k)$ 의 위치를 B 프레임에서 예측할 수 있다.

$$MB_{\sim B}(k) = MB_B(k) - mv_{MB_B(k)} \quad (1)$$

비디오 압축 알고리즘에서 B 프레임은 인접한 I와 P 프레임에 의하여 부호화되어 압축률이 I 프레임이나 P 프레임에 비하여 높으며 예측 오류 또한 P 프레임보다 높다. 따라서 B 프레임의 공간적 정보인 $MB_{\sim B}(k)$ 를 이용하여 I 프레임에서 발생한 $MB_I(k)$ 의 오류를 은닉하는 방법보다는 P 프레임의 공간적 정보를 이용하여 오류를 은닉하는 방법이 보다 효과적이다. 그러므로 앵커 프레임인 P 프레임의 정보를 사용하기 위하여 B 프레임의 순방향 움직임 벡터를 이용하여 $MB_{\sim B}(k)$ 에 상응하는 위치를 P 프레임에서 찾아야 한다. 그러나 $MB_{\sim B}(k)$ 는 순방향 움직임 벡터가 존재하지 않으므로 $MB_{\sim B}(k)$ 와 인접한 $MB_{B,TL}(k)$ 의 순방향 움직임 벡터를 이용하여 $MB_{\sim B}(k)$ 에 상응하는 위치를 파악할 수 있다 (그림 2). $MB_{B,TL}(k)$ 의 위치는 $MB_{\sim B}(k)$ 의 왼쪽 상단에 위치한다. 그 다음 $MB_{B,TL}(k)$ 와 $MB_{\sim B}(k)$ 의 거리 $d_{\sim B}(k)$ 를 구한다. $d_{\sim B}(k)$ 는 P 프레임에서 $MB_{\sim B}(k)$ 의 위치를 보정하기 위한 것으로, 두 점간의 수학적 거리를 의미하는 것이 아니고 $MB_{B,TL}(k)$ 가 $MB_{\sim B}(k)$ 에 위치하기 위하여 x, y 축으로 얼마만큼 이동해야 하는지를 의미하여 식 (2)와 같이 나타낼 수 있다.

$$d_{\sim B}(k) = MB_{\sim B}(k) - MB_{B,TL}(k) \quad (2)$$

B 프레임은 이전 P 프레임으로부터 순방향 움직임 예측으로 얻어진 움직임 벡터와 예측 오류가 같이 부호화되기 때문에 순방향 움직임 벡터를 이용하여 이전 P 프레임에서 B 프레임의 위치를 예측할



$$\begin{aligned}
 MB_I(k) &= [MB_{B.TL}(k) + mv_{MB_{B.TL}(k)}] + d_B(k) \quad (3) \\
 &= MB_{P.TL}(k) + d_B(k) \\
 &= MB_P(k)
 \end{aligned}$$

V. 실험 결과

그림 2. 역방향 움직임 벡터만큼 이동된 매크로블록 $MB_{P.TL}(k)$ 의 참조 지점을 포함하고 있는 $MB_{B.TL}(k)$ 의 순방향 움직임 벡터를 이용하여 이전 앵커 프레임인 P 프레임에서의 위치를 찾아내어 I 프레임으로 복사

실험은 비디오 데이터가 이상적인 채널을 통하여 전송된다는 가정 하에 이루어졌다. 즉, 실험에서 사용된 비디오 데이터의 블록 오류는 디코더에서 임의로 삽입된 것으로 본 실험에서 오류 검출 방식은 사용되지 않았다. 입력 데이터는 휘도 성분으로만 구성된 총 61장의 MPEG-2 Flower Garden과 Mobile Calendar를 사용하여 GOP 내의 프레임 수가 12인 총 5개의 GOP를 구성하였다. 앵커 프레임의 간격(M)은 3과 5 두 가지의 경우로 실험하였고 비트율과 프레임율은 각각 5 Mbps 와 25 fps로 그리고 인위적인 블록 손실률은 10%와 20%로 지정하였다. 10%의 블록 손실률이란 연속적인 매크로블록 오류가 발생한 경우를 나타내며 20%인 경우는 매크로블록 오류가 세로로도 반복된 심각한 슬라이스 오류를 나타낸다. 본 논문에서 제안하는

수 있다. 즉, 그림 2에서와 같이 $MB_{B.TL}(k)$ 의 순방향 움직임 벡터 $mv_{MB_{B.TL}(k)}$ 를 이용하여 $MB_{B.TL}(k)$ 에 상응하는 $MB_{P.TL}(k)$ 를 P 프레임에서 찾아낼 수 있다. 그리고 매크로블록 $MB_{P.TL}(k)$ 를 거리 $d_B(k)$ 만큼 이동시킨 $MB_P(k)$ 가 $MB_I(k)$ 에 상응하는 매크로블록이므로 $MB_P(k)$ 를 I 프레임의 오류가 발생한 매크로블록에 복사함으로써 오류를 은닉할 수 있다.

표 1. M=3일 때 Flower Garden 시퀀스에 포함된 I 프레임의 PSNR

I_n : n번째 GOP의 I 프레임 L_1, L_2 : 10%와 20%의 블록 손실율

Flower	Error Free	Temporal		Spatial_SB		Spatial_MB		F-B BM		ST_B1		ST_B2	
		L_1	L_2	L_1	L_2	L_1	L_2	L_1	L_2	L_1	L_2	L_1	L_2
I_1	33.49	24.08	20.97	18.36	12.95	23.70	15.23	25.19	21.77	28.06	25.28	28.27	25.83
I_2	32.92	23.80	20.73	18.31	13.12	22.99	15.37	24.79	21.78	28.89	26.12	29.16	26.96
I_3	33.22	23.40	20.18	18.35	13.18	22.80	15.41	24.48	21.62	28.84	26.30	29.93	27.72
I_4	32.78	23.18	20.09	18.24	13.00	22.46	15.16	23.71	20.85	28.05	25.36	29.44	26.99

표 2. M=3일 때 Mobile Calendar 시퀀스에 포함된 I 프레임의 PSNR

I_n : n번째 GOP의 I 프레임 L_1, L_2 : 10%와 20%의 블록 손실율

Mobile	Error Free	Temporal		Spatial_SB		Spatial_MB		F-B BM		ST_B1		ST_B2	
		L_1	L_2	L_1	L_2	L_1	L_2	L_1	L_2	L_1	L_2	L_1	L_2
I_1	39.09	23.97	20.77	19.42	14.29	23.07	16.28	24.42	21.16	27.52	25.18	27.64	24.68
I_2	34.98	23.53	20.46	19.47	14.27	22.73	16.24	24.20	21.16	26.80	24.74	26.79	24.27
I_3	35.94	23.64	20.62	19.50	14.45	22.63	16.34	24.38	21.56	27.14	25.00	27.27	24.63
I_4	37.46	24.00	20.98	19.06	14.41	22.58	16.28	25.83	22.91	27.99	25.49	27.80	25.21

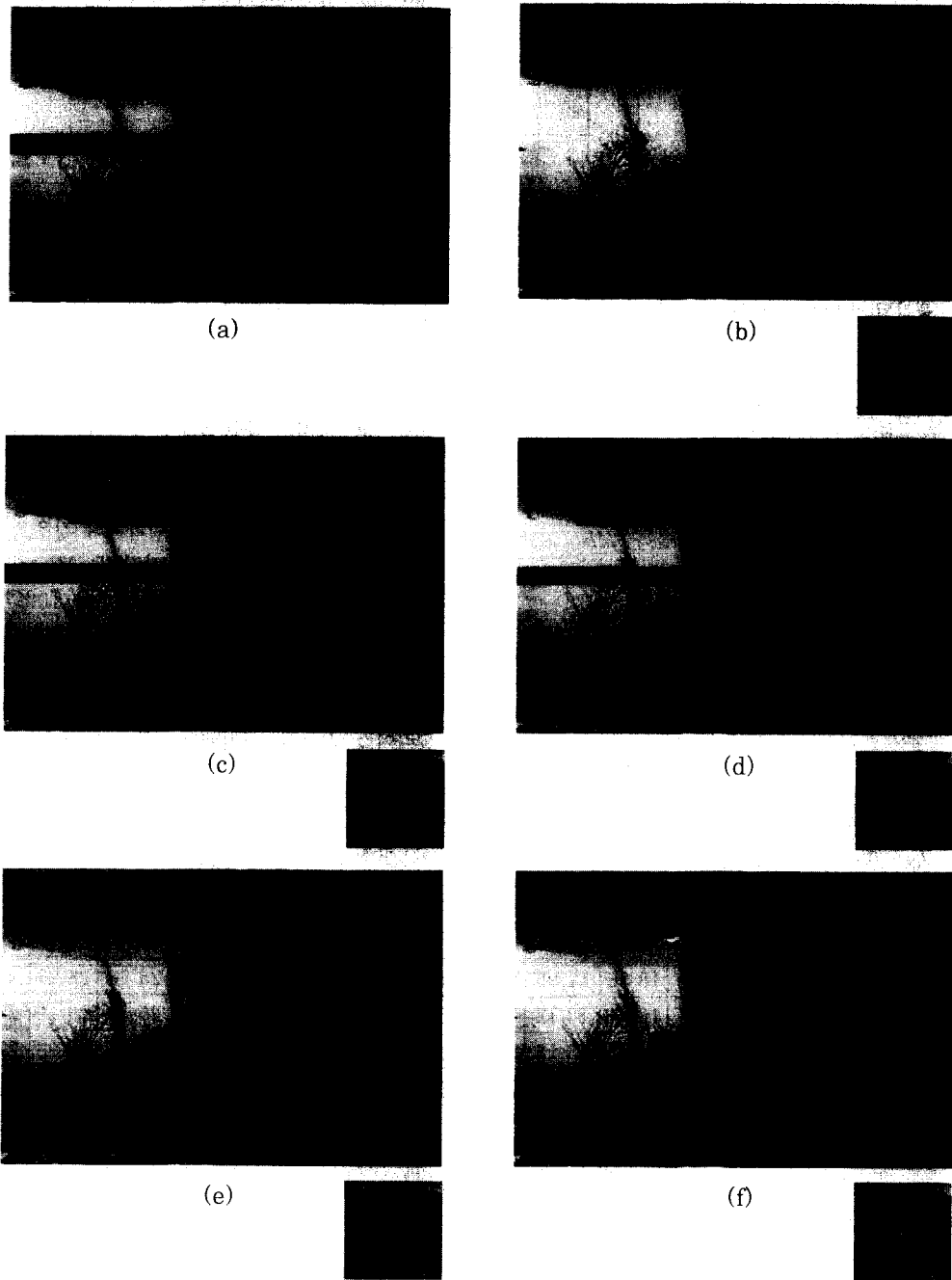


그림 3. 블록 손실률이 20%인 연속적인 슬라이스 오류가 발생한 경우 여러 가지 오류 은닉 방식

- (a) 오류가 발생한 경우
- (b) 단순 시간적 오류 은닉 (Temporal)
- (c) 서브 블록 단위의 공간적 오류 은닉 (Spatial_SB)
- (d) MB 단위의 공간적 오류 은닉 (Spatial_MB)
- (e) 순방향-역방향 블록 정합 오류 은닉 (F-B BM)
- (f) B2 프레임을 참조한 시공간적 오류 은닉 (ST_B2)

알고리즘의 성능 평가를 위하여 이미 제안된 여러 가지 오류 은닉 방식 중 I 프레임의 오류 은닉에 적용이 가능한 단순 시간적 오류 은닉 방식, 서브 블록 단위의 공간적 오류 은닉 방식, 매크로블록 단위의 공간적 오류 은닉 방식 그리고 로그 검색 기반의 순방향-역방향 블록 정합 오류 은닉 방식 등과 성능을 비교하였다 (그림 3). 혼합 오류 은닉 방식의 경우 I 프레임의 모든 매크로블록은 인트라 모드로 부호화되기 때문에 공간적 오류 은닉 방식과 같은 결과를 나타내므로 비교시 제외되었다.

시간적 오류 은닉 방식(Temporal)은 오류가 발생된 부분을 이전의 앵커 프레임에서 움직임 보정 없이 사용하므로 I 프레임과 앵커 프레임간의 움직임 차이로 인한 왜곡 현상이 발생된 것을 볼 수 있다 (그림 3-(b)). 그림 3-(c)의 서브 블록 단위의 공간적 오류 은닉 방식(Spatial_SB)은 연속적인 매크로블록 오류의 경우 주변 블록의 손상으로 인하여 오류를 효과적으로 은닉할 수 없다. 매크로블록 단위의 공간적 오류 은닉 방식(Spatial_MB)은 연속적인 매크로블록 오류가 발생한 경우 상하에 위치한 매크로블록을 이용하여 오류를 은닉할 수 있으나 가중 평균으로 인한 번짐 현상이 동반된다 (그림 3-(d)). 또한 연속적인 슬라이스 오류가 발생하여 좌우에 위치한 매크로블록 뿐만 아니라 상하에 위치한 매크로블록 역시 오류 은닉에 사용될 수 없으며

로 오류를 제대로 은닉할 수 없다. 로그 검색 기반의 순방향-역방향 블록 정합 오류 은닉 방식(F-B BM)의 경우 그림 3-(e)에서 보는 바와 같이 움직임이 적은 부분의 오류는 제대로 은닉되는 반면 나무 기둥과 같이 움직임이 큰 부분은 로그 검색의 부정확한 움직임 예측으로 인한 움직임 왜곡 현상이 발생된다. 또한 I 프레임 내의 오류를 은닉하기 위하여 I 프레임에 필요한 추가적인 움직임 벡터를 구해야 한다는 단점을 갖는다. 본 논문에서 제안한 알고리즘(ST_B1, ST_B2, ST_B3, ST_B4, ST_B5)은 이전 GOP의 마지막 B 프레임의 순방향, 역방향 움직임 벡터와 P 프레임의 공간적 정보를 이용하는 방식으로 다른 기존의 방식들에 비하여 시각적으로 많이 개선되었음을 확인할 수 있다 (그림 3-(f)). 또한 다음의 표에서 알 수 있듯이 제안한 알고리즘을 이용하여 오류를 은닉하는 방법이 다른 알고리즘에 비하여 높은 PSNR을 얻을 수 있었다. 특히 표 1, 표 2와 같이 M이 3인 경우 P 프레임과 I 프레임 사이의 두 개의 B 프레임 중 첫 번째 B 프레임(B1)과 두 번째 B 프레임(B2)을 참조 프레임으로 사용한 오류 은닉 방식을 비교해보면 B2 프레임의 움직임 벡터를 이용하여 오류를 은닉한 방식이 B1 프레임을 이용한 방식보다 더 높은 PSNR을 얻을 수 있었다. 그 이유는 B2 프레임이 B1 프레임보다 오류가 발생한 I 프레임과 시간적으

표 3. (a) M=5일 때 Flower Garden 시퀀스에 포함된 I 프레임의 PSNR
 I_n : n번째 GOP의 I 프레임 L_1, L_2 : 10%와 20%의 블록 손실율

Flower	Error Free	Temporal		Spatial_SB		Spatial_MB		F-B BM	
		L_1	L_2	L_1	L_2	L_1	L_2	L_1	L_2
I_2	34.51	23.64	20.27	18.49	12.99	24.20	15.30	24.01	20.70
I_3	33.22	23.16	19.80	18.46	13.17	23.48	15.46	23.88	20.41
I_4	35.48	22.12	19.06	18.50	13.23	23.35	15.49	22.66	19.49
I_5	35.10	22.71	19.36	18.39	13.05	22.89	15.24	23.08	20.06

표 3. (b) M=5일 때 Flower Garden 시퀀스에 포함된 I 프레임의 PSNR
 I_n : n번째 GOP의 I 프레임 L_1, L_2 : 10%와 20%의 블록 손실율

Flower	ST_B1		ST_B2		ST_B3		ST_B4		ST_B5	
	L_1	L_2	L_1	L_2	L_1	L_2	L_1	L_2	L_1	L_2
I_2	25.08	21.54	25.24	21.67	25.03	21.53	24.89	21.37	25.57	22.00
I_3	24.87	21.18	25.18	21.72	25.12	21.35	25.46	21.59	25.79	22.05
I_4	24.36	20.97	24.97	21.37	24.32	20.75	24.48	20.90	24.83	21.21
I_5	24.60	21.33	25.31	21.82	24.58	21.21	24.89	21.43	25.08	21.55

표 4. (a) M=5일 때 Mobile Calendar 시퀀스에 포함된 I 프레임의 PSNR
 I_n : n번째 GOP의 I 프레임 L_1, L_2 : 10%와 20%의 블록 손실율

Mobile	Error Free	Temporal		Spatial_SB		Spatial_MB		F-B BM	
		L_1	L_2	L_1	L_2	L_1	L_2	L_1	L_2
I_2	38.60	23.06	19.77	19.42	14.29	23.08	16.29	24.49	21.83
I_3	36.42	22.85	19.66	19.49	14.27	22.80	16.25	24.45	21.47
I_4	36.72	23.03	19.75	19.51	14.45	22.63	16.34	24.27	21.44
I_5	37.20	22.69	19.81	19.05	14.41	22.56	16.28	23.50	20.73

표 4. (b) M=5일 때 Mobile Calendar 시퀀스에 포함된 I 프레임의 PSNR
 I_n : n번째 GOP의 I 프레임 L_1, L_2 : 10%와 20%의 블록 손실율

Mobile	ST_B1		ST_B2		ST_B3		ST_B4		ST_B5	
	L_1	L_2	L_1	L_2	L_1	L_2	L_1	L_2	L_1	L_2
I_2	26.76	23.83	26.87	23.83	27.52	24.56	27.16	23.99	26.73	23.80
I_3	26.42	23.91	26.70	24.00	26.78	24.23	26.48	23.77	26.27	23.49
I_4	25.94	23.81	26.29	23.90	26.47	24.08	26.63	24.14	26.45	23.65
I_5	25.16	22.26	26.08	23.39	26.67	23.93	25.93	23.11	26.83	23.82

로 가까이 위치하여 오류 은닉에 사용된 움직임 벡터의 오류가 적기 때문이다. 또한 B 프레임이 많이 포함된 M이 5인 경우 역시 제안한 MPEG-2의 GOP 구조를 이용한 I 프레임의 시공간적 오류 은닉 방식이 더 좋은 결과를 얻었음을 표 3과 표 4를 통하여 확인할 수 있다. 마지막으로 GOP의 시작 프레임인 I 프레임에 의하여 영향을 받은 다른 인터 프레임들의 PSNR 변화를 그림 4와 그림 5에서 살펴 보았다. 그림과 같이 I 프레임의 오류가 다른 인터 프레임에 전달됨으로써 같은 GOP를 구성하는 모든 프레임의 PSNR이 저하됨을 확인할 수 있다. 또한 연속적인 매크로블록 오류가 세로로도 반복되는 심각한 슬라이스 오류의 경우 그림 5와 같이 두 가지 공간적 오류 은닉 방식이 가장 낮은 PSNR을 얻었으며 제안한 알고리즘에 의한 방식 중 B2 프레임을 참조하여 오류를 은닉한 방식이 가장 높은 PSNR을 얻은 것을 볼 수 있다. 만약 B 프레임이 없는 영상의 경우에는 제안하는 방식에서 오류 은닉에 필요한 정보를 제공할 B 프레임이 없으므로 I 프레임의 오류를 은닉하기 위하여 제안하는 알고리즘을 적용할 수 없다. 따라서 이러한 경우에는 I 프레임의 오류를 은닉하는 방법 중 P 프레임의 정보를 이용한 로그 검색 기반의 순방향-역방향 블록 정합 오류 은닉 방식이나 I 프레임의 정보만을 이용

하는 공간적 오류 은닉 방식을 적용할 수 있다.

VI. 결 론

본 논문에서 제안한 시공간적 정보를 이용한 오류 은닉 방식은 MPEG-2 비디오 시퀀스에서 GOP의 시작 프레임인 I 프레임의 오류가 다른 인터 프레임에 전달됨으로써 발생하는 화질 저하를 최소화하기 위한 방식이다. 공간적 오류 은닉 방식과는 달리 이전 GOP의 마지막 B 프레임의 양방향 움직임 벡터를 이용하여 이전 P 프레임의 공간적 정보를 사용하기 때문에 공간적 오류 은닉 방식에서 발생하는 번짐 현상과 주변 매크로블록의 손상으로 오류를 제대로 은닉할 수 없다는 단점을 효과적으로 극복할 수 있다. 또한 로그 검색 기반의 순방향-역방향 블록 정합 오류 은닉 방식은 I 프레임의 움직임 정보를 위하여 추가적인 블록 정합 알고리즘을 이용하는 것에 비하여 제안하는 알고리즘은 추가적인 정보 없이 MPEG-2 비디오 압축 알고리즘에 의한 기본적인 B 프레임의 움직임 벡터만을 이용한다. 그러나 프레임의 가장자리 부분이나 프레임간의 움직임이 매우 큰 경우 또는 확대, 축소가 된 매크로블록은 움직임 보상 예측 부호화 방식을 사용할 때 부정확한 움직임 벡터를 얻을 수 있기 때문에 몇몇 매크로

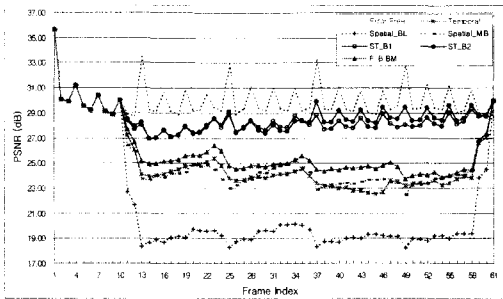


그림 4. M=3이고 비트율과 프레임율이 각각 5 Mbps 와 25 fps로 부호화된 Flower Garden 시퀀스의 블록 손실율이 10%일 때 PSNR 변화

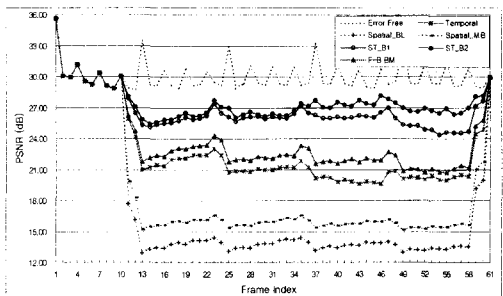


그림 5. M=3이고 비트율과 프레임율이 각각 5 Mbps 와 25 fps로 부호화된 Flower Garden 시퀀스의 블록 손실율이 20%일 때 PSNR 변화

블록에서 움직임 왜곡 현상이 발생할 수도 있다. 이는 블록 정합 알고리즘에서 가정하는 움직임의 선형성에 위배되는 블록들을 사용할 때 발생하는 피할 수 없는 왜곡이다. 실험 결과를 통하여 제안한 알고리즘이 기존의 I 프레임에 적용이 가능한 방식에 비하여 더 높은 PSNR을 제공하며 시각적으로 개선된 오류 은닉된 이미지를 보여준다.

참 고 문 헌

[1] MPEG-2 video and systems international standard, ISO/IEC/JTC1/WG11 MPEG 94, Nov. 1994

[2] Sheila S. Hemami, "Robust image communication over wireless channels," *IEEE Magazine*, vol. 39, pp. 120-124, Nov. 2001

[3] Y. Wang, S. Wenger, Jiangtao Wen and Aggelos K. Katsaggelos, "Error resilient video coding techniques," *IEEE Magazine*,

vol. 17, pp. 61-82, July 2000

[4] Victor Debrunner, Linda Decrunner, Longji Wang and Sridhar Radhakrishnan, "Error control and concealment for image transmission," *IEEE Communication Survey & Tutorials*, vol. 3, pp. 2-9, 2000

[5] Y. Wang and Q. Zhu, "Error control and concealment for video communications: a review," *Proc. IEEE*, vol. 86, pp. 974-997, May 1998

[6] S. Aign and K. Fazel, "Error detection & concealment measures in MPEG-2 video decoder," *Proc. of the International Workshop on HDTV'94*, Oct. 1994

[7] H. Sun, K. Challapali and J. Zdepski, "Error concealment in digital simulcast AD-HDTV decoder," *IEEE Trans. Consumer Electro.*, vol. 38, pp. 108-118, June 1992

[8] J. W. Suh and Y. S. Ho, "Error concealment based on directional interpolation," *IEEE Trans. Consumer Electro.*, vol. 43, pp. 295-302, Aug. 1997

[9] S. Tsekeridou and I. Pitas, "MPEG-2 error concealment based on block-matching principles," *IEEE Trans. Circuits and Systems*, vol. 10, pp. 646-657, June 2000

[10] S. Aign and K. Fazel, "Temporal & spatial error concealment techniques for hierarchical MPEG-2 video codec," *Proc. Globecom'95*, pp. 1778-1783

[11] M. H. Jo and W. J. Song, "Error concealment for MPEG-2 video decoders with enhanced coding mode estimation," *IEEE Trans. Consumer Electro.*, vol. 46, pp. 962-969, Nov. 2000

강민정 (Min-Jung Kang)

정회원

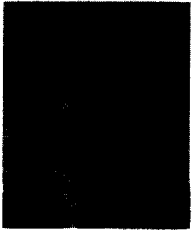


2001년 2월 : 동국대학교
정보통신공학과 (공학사)
2003년 2월 : 동국대학교
정보통신공학과 (공학석사)
2003년 3월 ~ 현재 : 동국대
학교 정보통신공학과 박사과정

〈주관심분야〉 Multimedia and Visual
communications, error control,
data compression

류철 (Chul Ryu)

정회원



1989년 : Florida Institute
of Technology 컴퓨터
공학과 (공학사)
1991년 : Polytechnic
University 전기공학과
(공학석사)
1997년 : Polytechnic
University 전기공학과 (공학박사)

1998년 ~1999년 : LG 정보통신 연구소
선임 연구원
1999년 ~ 현재 : 동국대학교 정보통신공학과
조교수

〈주관심분야〉 Visual communications, error
control, data compression