

# 동영상 트랜스코더를 위한 적응적 인트라 매크로블록 업데이트에 대한 연구

## Adaptive Intra Macroblock Update for Video Transcoder

서 동 완\*, 최 윤 식\*

Dong-Wan Seo\*, Yoon-sik Choe\*

### 요 약

일반적으로 동영상 트랜스코딩 기술에서는 시스템의 복잡도를 줄이기 위해, 입력 비트열로부터 매크로블록 종류와 움직임 벡터, DCT 계수 등의 중간 데이터를 추출하여 사용한다. 본 논문에서는 프레임 내에서 매크로블록의 중요도를 이용한 새로운 인트라 매크로블록 업데이트 방법을 제안한다. 매크로블록의 중요도는 입력 비트열을 복호화하는 과정에서 얻어지는 움직임벡터에 의해 계산한다. 참조 프레임 내의 매크로블록의 참조율은 현재 복호화하는 프레임의 움직임 벡터에 의해서 계산이 가능하며, 참조율이 높은 매크로블록은 다음 프레임과 그 이후 프레임에 영향력이 높다. 제안한 기법을 통해, 전송 오류에 대해 강인하고 더 우수한 화질을 나타내는 동영상 비트열을 얻을 수 있다.

### Abstract

To reduce the complexity of video transcoder, transcoding techniques generally use intermediate data such as macroblock types, motion vectors, DCT coefficients, which are extracted from incoming video bitstream. In this paper, we propose a new adaptive INTRA update scheme utilizing the importance of each macroblock in a given frame. The importance measure is calculated from the motion vectors which are extracted from the decoding procedure of a incoming video data. The ratio of the macroblock to be referred by the next frame is calculated by the motion vectors of the next frame and the macroblocks to be referred higher ratio has more effect on the next frame and the followed frames. The proposed scheme can obtain transcoded bitstreams which has better image quality and more improved resilience against transmission errors.

**Keywords** : Error Resilience, Intra Macroblock Update, Intermediate Data, Video Transcoder

### I. 서 론

입력된 동영상 비트열을 복호화하여, 다른 형태의 동영상 비트열로 부호화하는 기술을 트랜스코딩 기술이라 한다. 트랜스코더는 입력 동영상을 출력 동영상으로 부호화하기 위해 복호기와 부호기를 결합한 형태를 띄고 있다. 트랜스코딩 기술에 관한 연구에서는 복호화와 부호화 과정 내의 엄청난 계산량을 감소시키기 위해 입력된 동영상 비트열에서 부호화 과정에서 필요한 정보(중간 데이터)를 이용하여 부호화 과정의 연산량을 감소시킨다 [1-4]. 추출한 중간 데이터의 이용 방법에 따라 트랜스코더의

구조와 효율성은 결정되며, 트랜스코더의 목적에 따라 트랜스코더의 종류가 결정된다.

트랜스코더는 입력 동영상의 압축 방법과 출력 동영상의 압축 방법의 동일성에 따라 동종의 트랜스코더와 이종의 트랜스코더로 나뉘며 [1],[2],[5], 트랜스코딩 기술은 가장 많은 연산량을 요구하는 부호화 과정내의 움직임 예측의 계산량 감소를 위한 중간 데이터 활용 기술 [4][6][7], 시간·공간 해상도 감소에 따른 중간 데이터 활용 기술 [4][8], 공간 영역까지 복호화하지 않고 트랜스폼 영역에서 행하는 트랜스코딩 기술 및 중간 데이터 활용 기술 [5][9]에 관한 주제에 초점이 맞추어 연구되어 왔다. 즉, 실시간 트랜스코딩을 위한 중간 데이터 활용 기술과 DCT 영역에서의 트랜스코딩 기술에 그 초점을 두고 진행해 왔다. 트랜스코딩 기술은 일반적으로 서버나 네트워

\*연세대학교 전기전자공학부

논문 번호 : 2005-4-3

심사 일자 : 2005. 12. 26

접수 일자 : 2005. 10. 14

수정 완료 : 2006. 1. 27

크의 게이트 단에서 서로 다른 네트워크 또는 시스템 환경의 미디어 서비스를 위해 필요한 기술이다. 따라서, 일반적인 동영상의 부·복호화와는 다르게 영상 통신의 측면을 고려한 기술이어야 한다. 유선망과 무선망을 연결하는 동영상 서비스인 경우, 동영상 트랜스코딩은 무선망의 특수적인 상황에서 발생하는 비트 에러 또는 패킷 로스 등을 고려한 기술이 필요하다.

무선망에서 일어나는 에러에 대처하기 위해 동영상 부호화 기술은 에러에 강인한 부호화 방법(Error robust video coding), 에러 복원 부호화 방법(Error resilient video coding) 등의 주제에 관한 연구를 진행해 왔다 [10]. 에러 복원 부호화 방법 중에 하나가 INTRA 업데이트를 이용한 방법으로 패킷 로스가 일어난 경우 빠르게 잃어버린 비디오 데이터에 대한 보상이 이루어지게 한다. 본 논문에서는 동영상 트랜스코더 내에서 패킷 로스에 효율적으로 대응할 수 있는 INTRA 업데이트 기술을 제안한다. 기존의 INTRA 업데이트 기술에서는 에러 확률 모델에 근간하여 에러 발생 가능한 매크로블록을 선별하여 업데이트하는 방법을 택하고 있다 [11][12]. 비트 에러의 경우, 에러 확률 모델을 통한 접근이 가능하나 패킷 로스의 경우 매크로블록 별로 에러 확률을 계산하는 것은 불가능하다. 일반적으로 패킷 로스의 경우, 하나의 프레임 전체 또는 그 상당 부분의 비디오 데이터를 잃어버린다. H.263+의 TMN8[16]에서 채택한 인트라 업데이트 기법에서는 블록의 중요도를 고려하는 접근 방법을 택하고 있지만 실제로는 블록의 중요도 보다는 전체 시퀀스에서 각 해당 위치의 매크로블록이 동일한 확률로 업데이트되는 알고리즘이다. 따라서 본 논문은 전체 시퀀스에 영향력이 강한 매크로블록을 우선적으로 INTRA 업데이트하는 방법을 제안한다. 참조 프레임 내의 매크로블록의 참조율은 현재 복호화하는 프레임의 움직임 벡터에 의해서 계산이 가능하며, 참조율이 높은 매크로블록은 다음 프레임과 그 이후 프레임에 영향력이 높다. 제안 알고리즘에서는 매크로블록의 중요도를 움직임 보상에 사용되는 참조율로 설정한다.

본 논문의 구성은 다음과 같다. II장에서는 기존의 INTRA 업데이트 기술에 대한 소개와 이의 문제점을 제시하고, III장에서 매크로블록의 중요도를 트랜스코더 내에서 추출 가능한 중간 데이터인 움직임 벡터에 의해 계산하는 방법을 설명하고, 제안 알고리즘의 INTRA 업데이트 기술을 설명한다. IV장에서는 제안 INTRA 업데이트 방법과 TMN8에서 채택한 적응적 인트라 업데이트 기법을 MPEG-2 [13]에서 H.263+ [14]로의 트랜스코더를 통해 성능을 검증하고, 이에 이어 V장에서 본 논문의 결론을 제시한다.

## II. INTRA 업데이트 기법

인트라 매크로블록 업데이트는 H.263+ 나 MPEG-4

[15]와 같이 주로 저비트율을 목표로 하는 동영상에서 오류의 시간적·공간적 전파(Temporal-Spatial error propagation)를 줄이기 위해 사용하는 기법이다. MPEG-2와 같은 높은 비트율을 가지는 동영상에서는 주기적으로 I 픽처를 삽입하여 오류의 전파를 막지만, 저비트율 영상에서는 프레임당 몇 개의 인트라 매크로블록(Inter macroblock)을 선택하여 인트라 매크로블록(Intra macroblock)으로 변환하는 것이 효과적이다. 다음 그림 1은 동영상에서 흔히 나타나는 오류의 시간적·공간적 전파 현상을 보여주고 있다. 오류의 전파를 막기 위해 사용되는 인트라 매크로블록 업데이트는 부호화에 사용할 비트량을 한정시킨 경우, 화질에 오히려 부정적인 영향을 준다. 다음의 표 1은 실제 실험 결과로서, susie 영상에 대해 에러가 없는 상황에서 프레임별로 인트라 업데이트 되는 매크로블록의 수를 다르게 하였을 때의 100 프레임 평균 PSNR을 나타낸 것이다.



그림 1. 오류의 시간적·공간적 전파

Fig. 1. Spatial and temporal propagation of errors

표 1. 인트라 업데이트 매크로블록 수와 PSNR의 관계  
Table 1. Relation of the number of intra updated macroblocks to PSNR

프레임당 업데이트되는 매크로블록 수	5	4	3	...	0
PSNR(dB)	32.92	33.04	33.14	...	33.54

프레임당 인트라 업데이트 되는 매크로블록의 수가 늘어날수록 각 프레임의 화질은 떨어지게 된다. 인트라 업데이트는 한 프레임의 매크로블록 몇 개를 인트라로 전환함으로써 이후 프레임에서의 오류 전파를 막지만, 인트라 매크로블록은 비트율-왜곡 관점에서 비트 효율을 감소시켜 업데이트가 행해진 해당 프레임의 화질에는 좋지 않은 영향을 줄 수 있다.

인트라 매크로블록 업데이트 기법은 영상의 특성 고려 여부에 따라 비적응적 업데이트(Nonadaptive Update)와 적응적 업데이트(Adaptive Update) [11][12]로 구분된다. 비적응적 업데이트는 영상의 특성을 전혀 고려하지 않는 방법으로서, 구현이 쉽고 간단하다는 장점이 있지만, 많은 문제점을 가지고 있어서 실제로 응용되는 경우는 드

물다. 대표적으로 래스터-스캔 인트라 매크로블록 업데이트(Raster-scan Intra Macroblock Update)와 랜덤 인트라 매크로블록 업데이트(Random Intra Macroblock Update)가 있다. 적응적 업데이트는 영상의 특성에 따라 인트라 업데이트를 달리 하는 기법으로, 많은 연구 결과를 통해 비적응적 업데이트보다 우수한 성능의 다양한 적응적 업데이트 기법이 제안되고 있다. 대표적인 기법으로 H.263+ TMN8에서 사용되는 업데이트 기법이 있으며, MPEG-4에서는 비디오 객체 평면(Video Object Plane) 단위로 적용되는 별도의 적응적 업데이트 기법이 사용된다.

H.263+의 TMN8에서 사용되고 있는 인트라 업데이트 기법은, 영상의 특성을 고려하는 적응적 업데이트에 해당되며, 비적응적 업데이트에 비해 상당히 복잡한 절차로 이루어진다. 이 방법은 각 프레임에서 업데이트 할 매크로블록을 선정할 때, 상당 부분 불규칙적인 요인에 의해 영향을 받는다는 특징이 있는데, 대략적으로 다음과 같은 절차에 의해 이루어진다.

**TMN8 INTRA 업데이트 과정**

**과정 1.** 인트라 매크로블록 업데이트율(Intra Macroblock Update Rate)을 정한다. 이는 매크로블록을 업데이트 하는 빈도와 관련이 있다.

**과정 2.** 한 프레임을 구성하는 각 위치의 매크로블록마다 0부터 업데이트율까지의 범위에서 발생시킨 난수를 할당한다. 이 수는 정수값을 가져야 하며, 매크로블록마다 다른 수를 할당해 줄 필요는 없다. 각 매크로블록에 할당한 숫자는 프레임 번호가 증가하면서, 1씩 증가하거나, 그대로 유지된다. 즉, 프레임  $t$ 의 첫 번째 매크로블록에  $a$ 가 할당되면 프레임  $(t+1)$ 의 같은 위치에 있는 매크로블록이 가지는 숫자는  $(a+1)$ 이 되거나,  $a$ 로 유지되는 것이다. 이 때, 생략된(Skipped) 매크로블록의 숫자는 증가되지 않고 그대로 유지된다.

**과정 3.** 매크로블록이 가지고 있는 숫자가 처음에 지정한 업데이트율보다 크거나 같으면, 해당 매크로블록이 인트라로 업데이트 되며, 다음 프레임의 같은 위치의 매크로블록이 갖는 숫자는 0으로 바뀐다.

그림 2는 기존의 인트라 업데이트 기법을 적용한 예이다. 인트라 업데이트율이 8일 때를 가정하였다. 프레임  $t$ 와 프레임  $(t+1)$ 을 비교해보면, 모든 매크로블록이 가지고 있는 숫자가 1씩 증가하였음을 알 수 있다. 지정한 인트라 업데이트율과 같은 숫자를 가지고 있는 프레임  $(t+1)$ 의 매크로블록  $b$ 는 인트라 매크로블록으로 변환되며, 프레임  $(t+2)$ 에 있는 같은 위치의 매크로블록  $b$ 의 숫자는 0이 되었음을 확인할 수 있다. 매크로블록  $b$ 가 다시 인트라 업데이트 되기 위해서는 프레임  $(t+2)$ 로부터

최소한 8개의 프레임이 진행되어야 한다. 프레임  $(t+1)$ 과 프레임  $(t+2)$ 를 비교해보면, 매크로블록  $c$ 가 가지고 있는 숫자가 6으로 유지되고 있음을 확인할 수 있다. 이것은 매크로블록  $c$ 가 생략된 매크로블록이기 때문에 발생한 현상이다. 같은 위치에 생략된 매크로블록이 많이 나타날수록 해당 위치의 매크로블록이 인트라 업데이트 될 확률은 줄어들게 된다.

TMN8의 인트라 업데이트 방법은 매크로블록마다 중요도를 달리 하여 업데이트 한다. 매크로블록 중요도를 판별할 때에 중요하게 다루어지는 요인은 부호화시 사용되는 비트량이다. 적은 비트로 부호화되는 매크로블록의 인트라 변환 가능성을 줄여, 인트라 업데이트로 인한 비트의 사용량 증가를 최소화하겠다는 것이다. 특정 위치의 매크로블록이 거의 업데이트 되지 않거나, 필요 이상으로 빈번하게 업데이트 되는 경우가 생기는 단점이 존재한다. 특히, SKIP 매크로블록이 문제가 될 수 있다. 일반적으로, SKIP 매크로블록은 연속된 프레임에서 비슷한 위치에 집중되어 나타나는 경우가 많다. TMN8의 방법을 사용하면, SKIP 매크로블록의 업데이트 확률이 줄어들게 되며, 이런 매크로블록에 오류가 생기는 경우, 회복에 많은

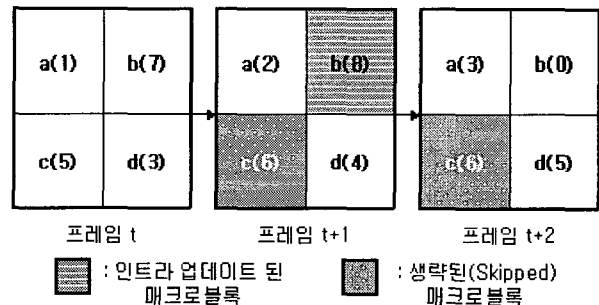


그림 2. TMN8의 인트라 매크로블록 업데이트  
Fig. 2. Intra Macroblock Update of TMN8

시간이 소비될 뿐만 아니라, 오류의 공간적 전파 현상으로 인해 주변 매크로블록까지 오류의 영향을 받을 수 있다. 주변 매크로블록 또한 SKIP 매크로블록이라면 문제가 더욱 심각해진다. 필요 이상으로 인트라 업데이트가 빈번하게 이루어지는 매크로블록이 발생하는 현상은 비트 활용의 측면에서 고려해볼 때 문제가 될 수 있다. 5번의 업데이트로도 충분한 매크로블록을 10번 업데이트 하는 경우가 생길 수 있다는 것이다. 필요 이상으로 업데이트가 이루어짐으로써 다른 부분에서 사용될 비트의 양이 줄어들게 되고 그에 따른 화질 저하가 발생할 수 있다.

TMN8 업데이트 방법의 또 다른 문제점은 일정한 성능을 보이지 못한다는 것이다. 과정 1에서 지정해주는 인트라 매크로블록 업데이트율의 역할은, 각 매크로블록의 최소 업데이트 간격을 정해주는 것에 불과하다. 실제로 각 매크로블록이 몇 프레임마다 업데이트 되는지는 업데이트

이트율보다는 영상 고유의 특성으로부터 큰 영향을 받는다. 과정 2에서 매크로블록마다 할당되는 난수는 0부터 업데이트율까지의 정수인데, 부호화할 때마다 다른 결과를 얻는다. 즉, 다른 영상뿐만 아니라, 같은 영상에 대해서도 부호화할 때마다 다른 결과를 얻을 수 있다는 것이다. 이 숫자들에 의해 각 매크로블록이 몇 번째 프레임에서 업데이트 될 것인지 정해지며, 각 프레임별로 업데이트 되는 매크로블록의 수가 결정된다. 프레임당 업데이트 매크로블록 수가 다르다는 것은 부호화된 동영상의 전송 시, 인트라 매크로블록이 많은 프레임에 오류가 발생한 경우, 대처 능력이 떨어질 수 있음을 뜻하며, 어떤 목표 비트율을 설정하고 이에 따라 율 조정(Rate Control)을 해야 하는 경우, 각 프레임의 화질을 일정 수준으로 유지하는 것이 어려워짐을 의미한다. TMN8의 인트라 업데이트 기법은 매크로블록 중요도 결정 과정에 위와 같은 단점이 존재한다.

### III. 중간데이터를 이용한 INTRA 업데이트 기법

적응적 인트라 업데이트에서 가장 중요하게 다루어져야 할 점은 매크로블록의 중요도 결정 기준이다. 인트라 매크로블록 업데이트가 현재 프레임뿐만 아니라 그 이후의 프레임에서의 오류의 시간적, 공간적 전파를 막기 위한 것이

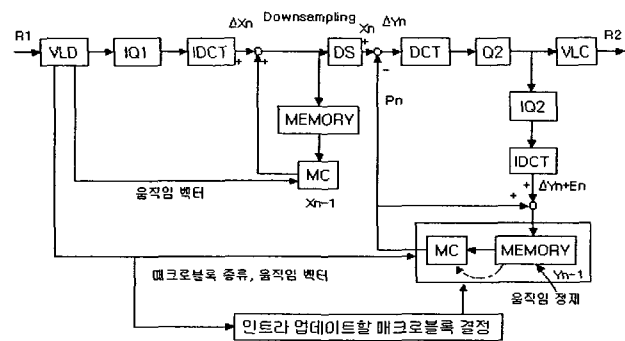


그림 3. 제안한 인트라 업데이트 기법이 적용된 트랜스코더

Fig. 3. Transcoder using proposed intra update algorithm

므로, 매크로블록의 중요도를 판별할 때에 현재 프레임의 정보보다는 다음 프레임들에 영향을 미치는 정보를 중요하게 판단해야 한다. 일반적으로, 한 프레임을 이루는 각각의 매크로블록은 다음 프레임을 구성할 때에 주는 영향이 다르다. 다음 프레임에서 주변의 여러 매크로블록의 움직임 예측, 보상에 영향을 주는 매크로블록일수록 중요도가 높다고 판단할 수 있으며, 이런 매크로블록을 우선적으로 인트라 업데이트 하면 좀 더 전송 오류에 효과적으로 대처할 수 있을 것이다. 즉, 매크로블록의 중요도는 다음 프레임의 각 매크로블록이 가지는 움직임 벡터를 통해 파악할 수 있

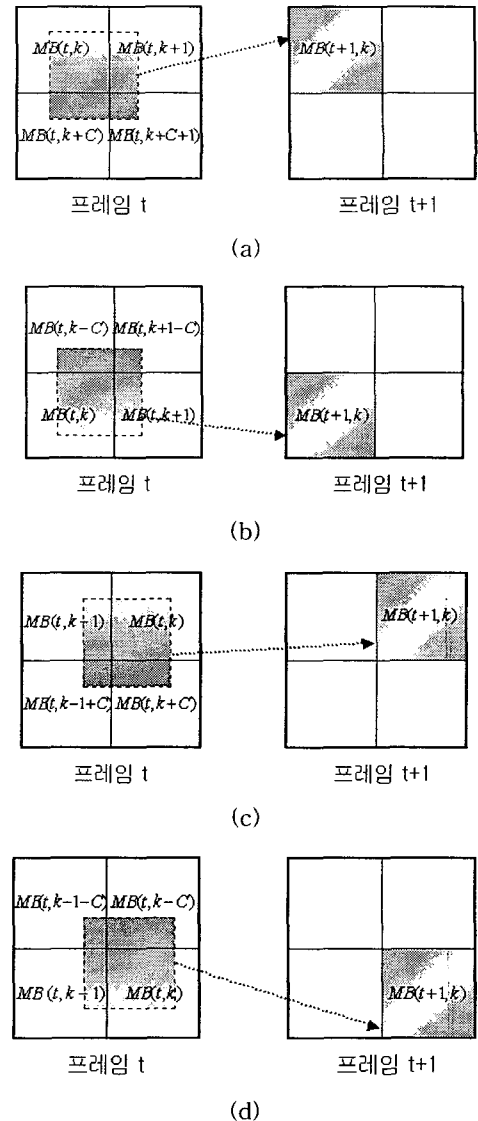


그림 4 움직임 벡터에 따른 매크로블록의 중요도 계산  
Fig. 4. Calculation of the importance for a given macroblock according to the motion vector

다. 문제는 일반적인 동영상 부호기에서는 현재 프레임의 매크로블록 중요도를 알기 위해 다음 프레임의 정보를 이용하는 일이 쉽지 않다는 점이다. 비인과성(noncausality)의 문제가 발생하게 된다[17]. 중간 데이터를 이용하는 트랜스코더에서는 이 점이 문제가 되지 않는다. 추출된 데이터로부터 이전 프레임에 의해 영향을 받는 다음 프레임의 정보들을 알 수 있기 때문이다. 본 논문에서는 트랜스코더 내에서 중간데이터(움직임 벡터)를 활용하여 새로운 적응적 인트라 업데이트 기법을 제안한다.

제안한 적응적 인트라 매크로블록 업데이트 과정은 크게 네 부분으로 나누어진다. 먼저 중간 데이터를 추출하여 움직임 벡터를 결정한 후, 이를 이용하여 각 매크로블

록의 중요도를 구하고, 중요도가 높은 매크로블록을 우선적으로 인트라 변환한다. 업데이트 횟수가 필요보다 적은 매크로블록의 발생 가능성을 줄이기 위해 랜덤 인트라 업데이트 기법을 부분 적용한다. 제안한 업데이트 기법이 적용된 트랜스코더의 구조는 다음 그림 3과 같다. 현재 프레임의 매크로블록 중요도를 판별하기 위해서는 다음 프레임의 움직임 벡터를 알아야 한다. 우선, 중간 데이터 추출기로부터 얻은 후보 움직임 벡터를 이용하여 최종적으로 부호기에 전송할 움직임 벡터 성분을 구하는 과정이 필요하다. 일반적으로 트랜스코더 입·출력 동영상의 시간, 공간 해상도, 압축에 이용하는 동영상 표준의 차이 등으로 인해, 프레임과 매크로블록의 종류가 다른 경우가 생길 수 있기 때문에 이를 적절히 고려하여 움직임 벡터를 결정하여야 한다. 후보 움직임 벡터를 이용하여 움직임 벡터를 근사화하여 구하는 방법은 [2]에 제시되어 있는데, 여기에는 프레임 사이의 움직임이 유니폼(uniform)하다는 가정이 내포되어 있다. 트랜스코더 입력 동영상에 B 픽처가 존재하는 경우에는, [2]의 방법을 수정한 [3]의 방법을 이용하면 프레임 재정렬 지연(Frame Reordering Delay)의 영향을 덜 받게 되므로 본 논문의 출력 움직임 벡터의 추출은 [3]에 의해서 하고, 추출된 움직임 벡터에 의해 매크로블록 별 참조율, 즉, 중요도를 산출한다.

다음 프레임의 움직임을 이용하여 현재 프레임의 매크로블록 중요도를 파악한다. 현재 프레임의 매크로블록 중요도는 다음 프레임의 움직임 보상 과정에 사용되는 픽셀의 수로 결정한다. 이 때, 같은 위치의 매크로블록 움직임 보상에 이용되는 픽셀은 제외한다. 각 매크로블록의 중요도는 다음 프레임을 구성하는 매크로블록의 움직임 벡터를 이용하여 아래의 방법으로 구한다. 중요도 계산에 사용된 움직임 벡터는 부호기로 전달되어 시스템의 복잡도를 줄이기 위한 목적으로 재사용된다.

$i(t, k)^{MB(t+1, k)}$ 는 프레임  $(t+1)$ 의 매크로블록  $k$ 에 의해 결정되는 프레임  $t$ 의  $k$  번째 매크로블록의 중요도를 나타내며,  $I(t, k)$ 는 최종적으로 계산되는 프레임  $t$ 의  $k$  번째 매크로블록의 중요도이다.  $MV_x(t, k)$ 와  $MV_y(t, k)$ 는 각각 프레임  $t$ 에 있는  $k$  번째 매크로블록의 움직임 벡터의  $x$  성분과  $y$  성분이며,  $N$ 은 한 프레임당 매크로블록 수이고,  $C$ 는 프레임의 가로 길이를 매크로블록 크기(16)로 나눈 값이다.  $MB(t, k)$ 는 프레임  $t$ 의  $k$  번째 매크로블록을 나타낸다.  $MV_x(t, k)$ 와  $MV_y(t, k)$ 의 부호에 따라 그림 4의 경우와 같이 네 가지 경우로 나누어진다. 그림 4의 네 경우에 대하여 매크로블록의 중요도를 구하는 방법은 식 (1)과 같이 정의될 수 있다. TMN8 방법에 비하여 이 매크로블록의 중요도를 구하는 방법에 의해 연산량이 증가한다.

여기서,  $sgn(x)$ 는 식 (2)와 같이 정의 될 수 있다.

각 매크로블록의 중요도를 구하는 방법은 최종적으로 식 (3)과 같이 표현된다. 식 (3)에서  $S(t, k, t+1)$ 는

$$\begin{aligned}
 & i(t, k + sgn(MV_x(t+1, k)))^{MB(t+1, k)} \\
 & = |MV_x(t+1, k)| \{16 - |MV_y(t+1, k)|\} \\
 & i(t, k + C \cdot sgn(MV_x(t+1, k)))^{MB(t+1, k)} \\
 & = \{16 - |MV_x(t+1, k)|\} |MV_y(t+1, k)| \\
 & i(t, k + sgn(MV_x(t+1, k))) \\
 & + C \cdot sgn(MV_y(t+1, k))^{MB(t+1, k)} \\
 & = |MV_x(t+1, k)| |MV_y(t+1, k)|
 \end{aligned} \tag{1}$$

$$sgn(x) = \begin{cases} -1, & x < 0 \\ 0, & x = 0 \\ 1, & x > 0 \end{cases} \tag{2}$$

프레임  $(t+1)$ 의 매크로블록 중, 움직임 예측, 보상 과정에서 프레임  $t$ 의 매크로블록  $k$ 에 의한 영향을 받는 매크로블록의 집합이다. 이 때 프레임  $(t+1)$ 의  $k$ 번째 매크로블록은  $S(t, k, t+1)$ 에 포함되지 않는다.

$$I(t, k) = \sum_{MB(t+1, k) \in S(t, k, t+1)} i(t, k)^{MB(t+1, k)} \tag{3}$$

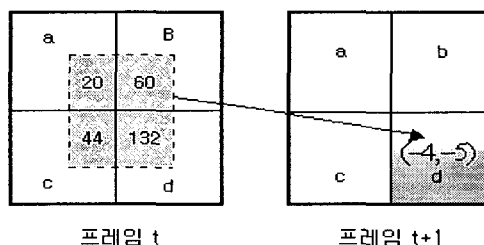


그림 5. 매크로블록 중요도의 예

Fig. 5. Example of the importance for a given macroblock

그림 5는 매크로블록의 중요도를 구하는 방법을 구체적인 예를 통해 나타낸 것이다. 한 프레임은 4개의 매크로블록으로 이루어져 있다고 가정한다. 프레임  $(t+1)$ 의 매크로블록  $d$ 의 움직임 벡터가  $(-4, 5)$ 이고, 나머지 매크로블록의 움직임 벡터가  $(0, 0)$ 이라면, 프레임  $t$ 의 각 매크로블록의 중요도는 매크로블록  $d$ 의 움직임 벡터에만 영향을 받게 된다. 프레임  $t$ 의 매크로블록  $a$ 의 중요도는 20,  $b$ 는 60,  $c$ 는 44가 되며,  $d$ 의 중요도는 0이 된다. 프레임  $t$ 의 매크로블록  $d$ 에서 픽셀 132개가 프레임  $(t+1)$ 의 매크로블록  $d$ 의 움직임 예측, 보상에 영향을 주고 있지만, 같은 위치의 매크로블록이므로, 이 픽셀 수는 중요도 계산 과정에서는 제외된다.

INTRAs 업데이트 과정은 아래에서와 같이 간단히 적용할 수 있다. 먼저 각 프레임별로 몇 개의 매크로블록을

인트라로 변환할 것인지 정한 후, 앞의 과정을 통해 구한 각 매크로블록의 중요도를 비교하여 중요도가 높은 매크로블록을 우선적으로 인트라 업데이트 한다. 이 때, 앞 프레임에서 업데이트 된 위치의 매크로블록은 업데이트 과정에서 배제한다. 이는 특정 위치의 매크로블록이 과도하게 연속적으로 업데이트 되는 것을 막아 비트 활용의 효율성을 높이기 위함이다. 중요도가 같은 매크로블록이 다수 발생하여 이 중 몇 개를 선택하여 업데이트 해야 하는 경우는 임의로 선정하여 업데이트 한다.

기존 기법에서 그랬던 것처럼 이 기법에서도 업데이트가 거의 일어나지 않는 매크로블록들이 발생할 수 있다. 이런 현상을 막는 방법은 중요도 고려 방법을 적용하기에 앞서 랜덤 인트라 업데이트 기법을 도입하여 중요도와 무관하게 프레임당 한두 개 정도의 매크로블록을 인트라 업데이트 하는 것이다. 먼저 각 위치의 매크로블록에 1부터  $N$ 까지의 수를 랜덤하게 할당한다. 여기서  $N$ 은 한 프레임을 구성하는 매크로블록의 수이며, 다른 위치에 있는 매크로블록에 같은 값이 배정되지 않도록 한다. 할당된 숫자는 업데이트 하는 순서를 나타낸다. 즉,  $k$ 번째 프레임에서는  $k$ 를 할당받은 매크로블록이 인트라 업데이트 된다. 이 방법에 의해 업데이트 된 매크로블록은 위의 중요도를 이용한 업데이트 과정에 포함시키지 않는다.

#### IV. 실험 결과

본 장에서는 TMN8에서 채택하고 있는 기존의 인트라 업데이트 방법과 본 논문을 통해 제안한 적응적 인트라 업데이트 방법의 성능을 구체적인 실험을 통해 비교·분석한다. 실험은 비트 에러가 발생한 경우와 패킷 로스가 발생한 경우로 나누어 행하였다.

##### 4.1 실험 방법

실험에는 MPEG-2로 압축된 4Mbps의 CIF 동영상을 64kbps, 96kbps, 128kbps의 H.263 QCIF 동영상으로 변환하는 공간 해상도 감소 트랜스코더를 사용하였으며, 실험 영상으로는 Carphone, Foreman, Susie 영상을 사용하였다. 화면 구성이 비교적 간단하고 움직임이 많은 영역이 중앙에 집중적으로 나타나는 Susie 영상은 각 프레임에서 SKIP 매크로블록이 많이 나타나는 특징을 가지는 반면, Carphone 영상과 Foreman 영상은 화면 구성이 복잡하고, 움직임이 많은 영역이 다양하게 나타나며, SKIP 매크로블록의 수도 적다. 실험에 사용된 모든 동영상의 프레임 임율은 30fps이다. 출력된 H.263+ 동영상은 첫 번째 I 프레임을 제외한 나머지 프레임이 모두 P 프레임으로 구성되어 있다. 비트율 조정 방법은 TMN8의 방법을 사용하였다.

제안한 방법을 이용한 인트라 업데이트에서는 프레임당 4개의 매크로블록을 업데이트 하였다. 4개의 매크로블

록 중 1개에 대해 중요도에 무관한 랜덤 업데이트 방법을 적용하여 인트라 변환한 후, 중요도 고려 방법을 이용하여 나머지 3개의 매크로블록을 인트라 업데이트 하였다.

기존의 TMN8 인트라 업데이트 기법의 경우, 각 프레임별로 평균 4개의 매크로블록이 인트라로 변환되도록 인트라 매크로블록 업데이트율(Intra Macroblock Update Rate)을 조정하였다. 앞장에서 언급한 바와 같이, 이 방법은 일정한 성능을 나타내지 않기 때문에 모든 결과를 비교 대상으로 채택할 수 없으므로, 조건에 맞는 결과를 비교 대상으로 선정하는 과정이 필요하다. 각 프레임마다 인트라 업데이트 된 매크로블록의 수를 파악한 후, 이를 합하여 프레임 개수로 나눔으로써 프레임당 평균 인트라 업데이트 매크로블록의 수를 계산한다. 프레임당 업데이트 된 매크로블록의 수가  $[4 - 0.05, 4 + 0.05]$ 를 벗어나는 경우는 실험 결과로 채택하지 않았다. 채택되지 않은 결과 중에는  $4 + 0.05$ 보다 많은 수의 매크로블록이 업데이트 된 경우보다  $4 - 0.05$ 보다 적은 수의 매크로블록이 업데이트 된 경우가 많았으며, 이 경우 프레임별 PSNR 비교시 미세한 우세를 보이나, 오류로부터의 회복도가 많이 떨어지는 특징을 가진다.

##### 4.2 전송 오류가 발생하지 않은 경우의 결과

다음의 표 2는 두 업데이트 기법을 적용한 트랜스코더 출력 동영상에 대해 각 프레임마다 PSNR을 계산하여 100 프레임의 평균값을 구한 것이다. 입력 동영상 비트율을 복호화한 후, 해상도 감소시킨 영상과 트랜스코딩 된 후, 복호화한 영상 사이의 PSNR 값을 이용하였다. 다음 절의 전송 오류가 발생한 경우는 전송 오류가 발생하지 않았을 때 복호화된 영상과 전송 오류가 발생하였을 때 복호화된 영상 사이의 PSNR로 비교하였다.

표 2. 전송오류가 발생하지 않은 경우의 평균 PSNR 결과

Table 2. Average PSNR results without transmission errors

	비트율	64kbps	96kbps	128kbps
	업데이트기법			
Carphon	Proposed	28.31	30.31	31.56
	TMN8	28.21	30.20	31.50
Susie	Proposed	32.93	34.76	35.88
	TMN8	32.81	34.66	35.79
Foreman	Proposed	38.30	39.74	40.77
	TMN8	37.97	39.56	40.52

PSNR 비교를 통해 제안한 기법이 TMN8 업데이트 기법에 비해 같은 비트율에서 다소 양호한 화질을 보임을

알 수 있다. 이는 기존의 TMN8 기법에 비해 제안한 기법이 비트의 활용면에서 우수함을 보여준다. TMN8 기법이 제안한 기법에 비해 PSNR이 좋지 않은 이유는 특정 위치의 매크로블록에 인트라 업데이트가 과도하게 집중되면서, 업데이트가 한 번도 일어나지 않는 매크로블록이 다수 발생하기 때문이다. 이런 매크로블록들로 인해 PSNR 값이 낮아지게 된다. 그 한 예로, 30번 인트라 업데이트를 수행하면서 96kbps로 Carphone 영상에 TMN8 기법을 적용한 경우, 100 프레임에서 인트라 업데이트가 1번도 일어나지 않은 매크로블록의 위치 수는 평균 4.11개였고, 96kbps로 적용한 Susie 영상의 경우는 이보다 많은 12.5개였다. 제안한 기법에서는 랜덤 업데이트 기법을 부분적으로 적용하였기 때문에, 업데이트가 전혀 일어나지 않는 매크로블록 위치는 생기지 않는다.

그림 6.(a)는 각각의 업데이트 기법을 적용하여 64kbps로 트랜스코딩한 Susie 영상의 6번째 프레임부터 20번째 프레임까지의 PSNR 값을 그래프로 나타낸 것이다. TMN8 방법을 이용했을 때보다 제안한 업데이트 기법을 이용했을 때 PSNR 값의 분포가 고르게 나타남을 확인할 수 있다. 이는 제안한 기법이 TMN8 방법보다 고른 화질을 보장함을 뜻한다. 그림 6.(a)에 나타난 15 프레임의 PSNR 평균값은 제안한 업데이트 기법이 33.04dB, TMN8 방법이 32.90dB이다. TMN8 업데이트 기법을 적용했을 때, 프레임별 PSNR의 변화가 큰 이유는 각 프레임마다 인트라 업데이트 되는 매크로블록의 수가 다르기 때문이다. 그림 6.(b)은 그림 6.(a)의 각 프레임에서 몇 개의 매크로블록이 인트라로 변환되었는지 보여주는 결과이다. 제안한 방법은 프레임당 4개의 매크로블록이 일정하게 인

그림 6. 제안 기법과 TMN8의 프레임별 비교

(Susie, 64kbps, frame 6~20)

(a) PSNR, (b) INTRA 업데이트 매크로블록 수

Fig. 6. Comparison between the proposed scheme and

TMN8 (Susie, 64kbps, frame 6~20)

(a) PSNR, (b) The number of INTRA updated macroblock

트라 변환되고 있지만, TMN8의 방법은 업데이트 되는 매크로블록의 수가 매우 불규칙하다. 전체 프레임에서 인트라 업데이트 된 매크로블록 수는 제안한 방법이 60개, TMN8 업데이트가 61개로 거의 비슷하다.

### 4.3 비트 에러가 발생한 경우

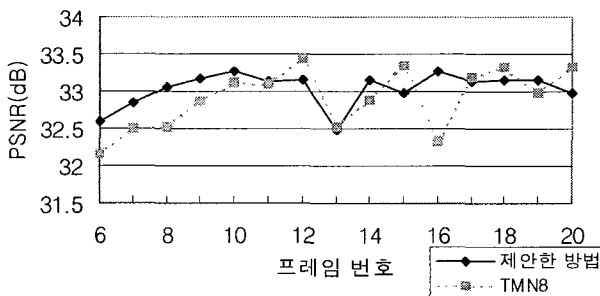
본 절의 실험에서는 트랜스코더 출력 동영상에 무선망을 통해 전송될 때 비트 오류가 발생하는 경우를 가정하였다. 전송 채널은 AWGN(Additive White Gaussian Noise)으로 가정하였으며, 비트 에러는 P-프레임의 매크로블록 레이어에만 발생시켰다. 64kbps로 트랜스코딩된 Carphone, Foreman, Susie의 네 가지 영상에 대해  $1.0 \times 10^{-3}$ ,  $1.0 \times 10^{-4}$ ,  $1.0 \times 10^{-5}$ 의 세 가지 BER에 대해 실험을 진행하였다. 같은 영상에 대해서도 수행할 때마다 다른 결과를 얻게 되는 인트라 매크로블록 업데이트 기법의 특성을 감안하여, 각 인트라 업데이트 기법별로, 4.1에 언급된 조건을 만족시키는 30개의 결과 영상을 실험에 사용하였다. 각각의 결과 영상들은 포맷이나 비트율, 전체 프레임에서 업데이트 된 매크로블록 수 등에는 큰 차이가 없지만, 같은 프레임에서 인트라 업데이트 된 매크로블록의 위치가 모두 다르다. 100 프레임으로 구성되어 있는 각 결과 영상들마다 일정 BER의 비트 에러를 발생시킨 후, 각 프레임별로 PSNR을 구하고, 이를 이용하여 프레임 평균 PSNR을 계산하였다. 이렇게 하여 얻은 각각 30개의 프레임 평균 PSNR 값들을 모두 더한 후 30으로 나누어 표 3에 나타내었다.

표 3. 비트 에러가 발생한 경우 평균 PSNR 결과

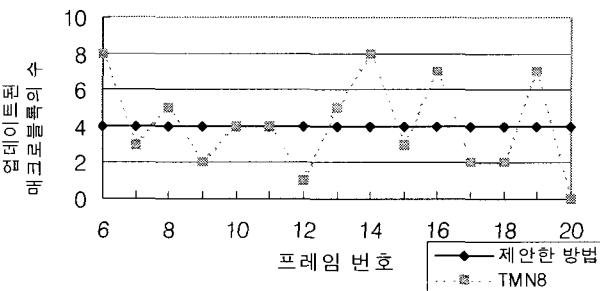
Table 2. Average PSNR results with bit errors

	업데이트기법	BER		
		$10^{-3}$	$10^{-4}$	$10^{-5}$
Carphon	Proposed	15.34	24.52	29.81
	TMN8	13.62	23.34	29.42
Susie	Proposed	17.98	26.12	32.75
	TMN8	13.20	24.87	32.11
Foreman	Proposed	20.01	30.20	38.09
	TMN8	16.15	28.22	36.91

같은 BER에서 제안한 기법이 TMN8 업데이트에 비해 다소 우수한 성능을 나타냄을 알 수 있다. Carphone,



(a)



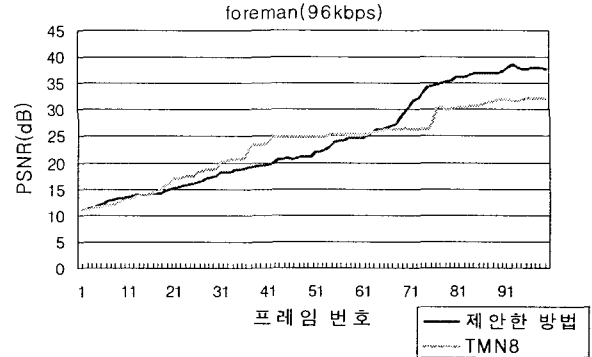
(b)

Foreman 영상보다 Susie 영상의 경우가 차이가 크데, 이것은 TMN8의 업데이트 기법을 적용했을 때 Susie 영상에 인트라 업데이트가 일어나지 않는 매크로블록이 다수 발생하기 때문이다.

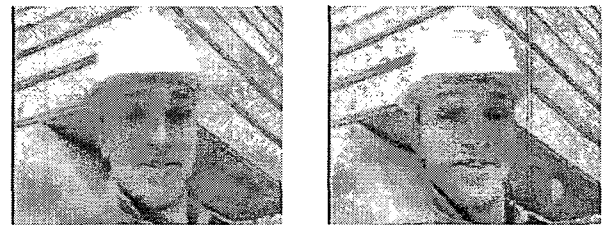
4.4 패킷 에러가 발생한 경우

트랜스코더의 출력 동영상 비트열에 패킷 손실에 의한 전송 오류가 발생한 경우를 가정한 실험을 위하여 출력 동영상의 첫 번째 I 프레임을 블랭크 영상(blank image)으로 교체한 후, 이로부터 회복되는 과정을 비교하였다. 동영상 전송에서는 비트 에러가 발생하는 경우보다 패킷 손실로 인해 다량의 비트를 한꺼번에 잃어버리는 경우가 많이 발생한다. 몇 개의 매크로블록 정보를 한꺼번에 잃을 수도 있고, 심한 경우 한 프레임 전체의 정보를 잃을 수도 있다. 본 절의 실험에서는 패킷 손실로 인해 한 프레임 전체에 오류가 생기는 경우를 가정하였다. 특정 프레임을 블랭크 영상으로 교체하는 것은 해당 프레임 전체에 오류를 발생시키는 것과 같은 효과를 가져다준다. 첫 번째 프레임 이후의 모든 프레임이 P 프레임이기 때문에, 대부분의 매크로블록의 오류로부터의 회복은 인트라 매크로블록 업데이트에 의해서만 가능하다. 물론, P 프레임을 구성하는 매크로블록 중에는 트랜스코더의 입력 비트열로부터 추출된 데이터에 의해 인트라 매크로블록 업데이트와 관계없이 인트라로 부호화되는 것들이 있지만, 그 비율은 극히 적은 수준이다. 프레임별 PSNR 비교에서는 30번 수행한 결과 영상에 대해 평균값을 취했으며, 100 번째 프레임의 비교에서는 형

Fig. 7. Carphone(64kbps). (a) PSNR comparison of each frame, 100<sup>th</sup> frame: (b) the proposed scheme, (c) TMN8



(a)

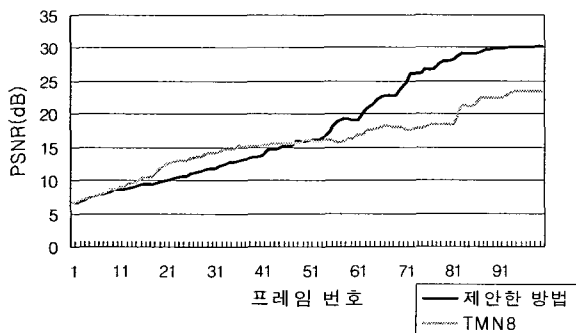


(b)

(c)

그림 8. Foreman(96kbps). (a) 프레임별 PSNR 비교, 100<sup>th</sup> 프레임: (b) 제안 알고리즘, (c) TMN8

Fig. 8. Foreman(96kbps). (a) PSNR comparison of each frame, 100<sup>th</sup> frame: (b) the proposed scheme, (c) TMN8



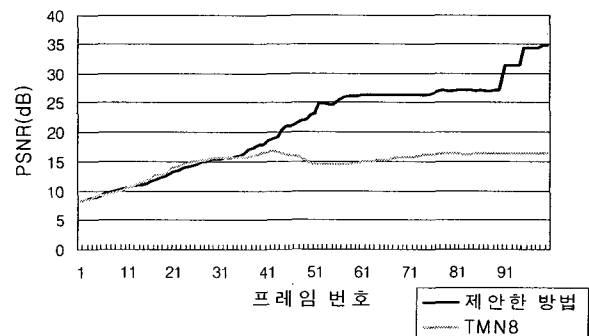
(a)



(b)

(c)

그림 7. Carphone(64kbps). (a) 프레임별 PSNR 비교, 100<sup>th</sup> 프레임: (b) 제안 알고리즘, (c) TMN8



(a)



(b)

(c)

그림 9. Susie(128kbps). (a) 프레임별 PSNR 비교,



100<sup>th</sup> 프레임: (b) 제안 알고리즘, (c) TMN8

Fig. 9. Susie(128kbps) (a) PSNR comparison of each frame, 100<sup>th</sup> frame: (b) the proposed scheme, (c) TMN8

평성을 고려하여 각 업데이트 기법별로 PSNR이 가장 좋지 않은 영상들을 택하여 비교하였다. 거의 모든 비트율에서 동일한 성능을 보이므로 본 논문에서는 영상에 따라 특정 비트율의 그림으로 표현하겠다.

그림 7, 8, 9를 통해, 기존의 TMN8 기법 적용 시, 인트라 업데이트 되지 않는 매크로블록이 다수 발생하여, 이런 매크로블록에 오류가 발생하는 경우 회복이 쉽지 않음을 알 수 있다. 두 기법의 성능 차이는, 구성이 복잡하고 움직임이 많은 영역이 프레임별로 다양한 분포를 보이는 Carphone이나 Foreman 영상에 비해, 움직임이 많은 영역이 중앙에 집중되어 있고, 생략된 매크로블록이 많이 나타나는 Susie 영상에서 더 크게 나타난다. TMN8의 경우 II장의 설명에서와 같이 생략된 매크로블록에 대한 업데이트 확률이 떨어지므로, 생략된 매크로블록이 다수 존재하는 Susie 영상의 경우 TMN8 방법에서는 PSNR이 프레임이 진행되더라도 거의 호전되지 않는다. Susie 영상의 경우, 세 가지 목표 비트율 모두에서 100 번째 프레임이 20dB 정도의 차이를 보이는 반면, Carphone 영상의 100 번째 프레임은 64kbps에서 약 7dB 정도의 차이를 보이고, 96kbps와 128kbps에서는 큰 차이를 보이지 않는다. Foreman 영상의 경우는 64kbps와 96kbps에서 제안한 기법이 TMN8 업데이트에 비해 약 6dB 정도 우수한 성능을 보인다.

그림 7, 8, 9의 100 번째 프레임 비교 결과에서 알 수 있듯이, 제안한 적응적 업데이트 기법이, TMN8 업데이트 기법에 비해, 오류를 분산시키는 효과가 우수하다. 즉, 오류를 분산시키기 때문에 오류가 남아 있더라도 어느 부분에 남아 있는지 육안으로 구분하기 어렵게 한다는 것이다. TMN8의 기법을 적용한 그림 7, 9의 (c)를 보면 첫 번째 프레임에서 발생한 오류가 그대로 남아 있다는 것을 확인할 수 있다. 이 결과는 앞 절의 전송 오류가 없는 경우의 실험 결과와 같은 맥락에서 이해할 수 있다. 제안한 기법에서의 업데이트가 TMN8 기법의 경우보다 고르게 행해지는 것으로 인한 결과이다.

## V. 결론

본 논문에서는 H.263+를 출력으로 하는 트랜스코더를 위한 새로운 적응적 인트라 업데이트 기법을 제안하였다. 트랜스코더의 복잡도를 줄이기 위한 목적으로 사용되는 중간 데이터를 재활용하여 매크로블록의 중요도를 구하고, 높은 중요도를 가지는 매크로블록들을 우선적으로 인트라 매크로블록 업데이트 함으로써 출력 동영상 비트율을 오류에 강인하게 할 수 있다. 이는 추가 정보의 추출 없이 기존 데이터의 활용도를 높이는 것만으로, 향상된

성능의 트랜스코더를 구현할 수 있음을 의미한다. 제안한 기법의 비교 대상으로는 H.263+ TMN 부호기에서 사용되는 업데이트 기법을 택하였으며, 두 업데이트 방법의 성능 비교 실험은, 트랜스코더의 출력 비트열에 비트 에러와 패킷 에러의 전송 오류가 발생한 경우와 그렇지 않은 경우로 나누어 이루어졌다.

전송 오류가 발생하지 않은 경우, 제안한 업데이트 기법이 TMN8 업데이트 기법에 비해 같은 비트율에서 다소 우수한 화질을 보여준다. 비록 0.1dB 이하의 미세한 차이 이기는 하지만, 이 결과는 기존의 TMN8 기법보다 제안한 기법에서 비트의 활용이 효율적으로 이루어지고 있음을 보여준다. 제안한 업데이트 기법이 TMN8 기법에 비해 각 프레임마다 고른 화질을 보장해준다는 사실도 이 점을 뒷받침한다. TMN8 기법이 제안한 기법에 비해 영상 압축의 측면을 중요하게 고려하고 있음에도 불구하고 PSNR이 떨어지는 이유는, 특정 위치의 매크로블록에 인트라 업데이트가 과도하게 집중되면서, 업데이트가 한 번도 일어나지 않는 매크로블록이 다수 발생하기 때문이다.

트랜스코더의 출력 동영상의 전송 중 비트 에러에 의해 영향을 받는 경우와 패킷 손실로 인해 오류가 발생하는 경우 모두 제안한 업데이트 기법이 기존의 기법에 비해 우수한 오류 회복도를 나타낸다. 특히, 패킷 손실로 인한 오류에 대한 실험에서는, 두 기법의 성능이, PSNR 값으로만 확인할 수 있는 미세한 차이가 아닌, 육안으로 쉽게 확인할 수 있을 정도의 큰 차이를 보인다. 제안한 기법은 전송 오류를 효과적으로 제거하여, 오류로부터의 회복도에서 우수한 성능을 보일 뿐만 아니라, 제거되지 않은 오류를 분산시키는 효과도 기존 TMN8의 방법에 비해 우수하다.

## 참고 문헌

- [1] S. Huifang, W. Kwok and J. W. Zdepski, "Architectures for MPEG compressed bitstream scaling", *IEEE Trans. on Circuits and Syst. for Video Tech.*, vol. 6, no 2, pp. 191-199, Apr. 1996.
- [2] T. Shanableh and M. Ghanbari, "Heterogeneous video transcoding to lower spatio-temporal resolutions and different encoding formats", *IEEE Trans. on Multimedia*, vol. 2, no. 2, pp. 101-110, Jun. 2000.
- [3] 윤규섭, "동영상 부호화 형식 변환을 위한 움직임 벡터 재추출 기법에 관한 연구", *연세대학교 석사 학위 논문*, 2001.
- [4] J. Nakajima, H. Tsuji, Y. Yashima, and N. Kobayashi, "Motion vector re-estimation for fast video transcoding from MPEG-2 to MPEG-4", *2001 Proc. of Workshop and Exhibition on MPEG-4*, pp. 87-90, Jun. 2001.

- [5] P. A. A. Assuncao, M. Ghanbari, "Transcoding of MPEG-2 video in the frequency domain", *IEEE Int. Conf. on Acoustics, Speech, and Signal Process.*, vol. 4, pp. 2633-2636, Apr. 1997.
- [6] L.-P. Chau, Y. Liang and Y.-P. Tan, "Motion vector re-estimation for fractional-scale video transcoding", *IEEE Int. Conf. on Information Tech. Coding and Computing*, pp. 212-215, Apr. 2001.
- [7] J. Youn, M.-T. Sun, "Motion Estimation For High Performance Transcoding", *IEEE Int. Conf. on Consumer Electronics*, pp. 136-137, Jun. 1998.
- [8] S. Guobin, Z. Bing, Y.-Q. Zhang and M. L. Liou, "Transcoder with arbitrarily resizing capability", *IEEE Int. Symp. on Circuits and Syst. for Video Tech.*, vol. 5, pp. 25-28, May 2001.
- [9] P. A. A. Assuncao and M. Ghanbari, "A frequency-domain video transcoder for dynamic bit-rate reduction of MPEG-2 bit streams", *IEEE Trans. on Circuits and Syst. for Video Tech.*, vol. 8, no. 8, pp. 953-967, Dec. 1998.
- [10] Y. Wang and Q.-F. Zhu, "Error control and concealment for video communication: A review," *Proc. of the IEEE*, vol. 86, no. 5, pp. 974-997, May 1998.
- [11] J. Y. Liao and J. D. Villasenor, "Adaptive intra update for video coding over noisy channels," *Int. Conf. on Image Process.*, vol. 3, pp. 763-766, Sept. 1996.
- [12] J. Y. Liao and J. D. Villasenor, "Adaptive intra block update for robust transmission of H.263", *IEEE Trans. on Circuits and Syst. for Video Tech.*, vol. 10, no. 1, pp. 30-35, Feb. 2000.
- [13] ITU-T, IT(Information technology-Generic coding of Moving pictures and associated audio information: VIDEO), ITU-T Recommendation H.262, July 1995.
- [14] ITU-T, IT(Video coding for low bit rate communication), Draft, ITU-T Recommendation H.263 Version 2, Sept. 1997.
- [15] ISO/IEC, IT(Information technology-Coding of audio-visual objects-Part 2: Visual), ISO/IEC 14496-2, Dec. 1999.
- [16] ITU-T Study Group 16, Video Codec Test Model, Near-Term, Version 8 (TMN8), 24-27 June 1997.
- [17] M. H. Willebeek-LeMair, Z.-Y. Shae and Y.-C. Chang, "Robust H.263 video coding for transmission over the Internet", *INFOCOM '98. the 17-th Annual Joint Conf. of the IEEE Computer and Commun. Societies*, vol. 1, pp. 225-232, Apr. 1998.



서 동 완(Dong-Wan Seo)

1999년 2월 연세대 전기공학과(공학사)  
 2001년 2월 연세대 전기전자공학과(공학 석사)  
 2001년 2월~현재 연세대 전기전자공학과 (박사과정 재학중)

관심분야 : 영상신호처리, 동영상 압축, 비트율 조정 기법, 비트율-왜곡(R-D) 최적화



최 윤 식(Yoon-sik Choe)

1979년 연세대학교 전기공학과(공학사)  
 1984년 Case Western Reserve Univ. 시스템공학과 (공학사)  
 1987년 Pennsylvania State Univ. 전기공학과 (공학 석사)

1990년 Purdue Univ. 전기공학부(공학 박사)  
 1990년~1993년 (주)현대전자 산업전자 연구소 책임 연구원  
 1993년~현재 연세대학교 전기전자공학부 교수  
 관심분야 : 영상신호처리, HDTV, 동영상압축