

# MPEG-2 to MPEG-4 트랜스코딩

김진경\*, 김규식\*, 김은태\*  
\*(주)오픈웹정보통신

e-mail:jkkim@openwebit.com

## MPEG-2 to MPEG-4 Transcoding

Jin-Kyoung Kim\*, Kyu-Sik Kim\*, Eun-Tae Kim\*  
\*OpenWeb Co., Ltd.

### 요 약

트랜스코딩은 통신 및 방송 등과 융합되어 활발히 진행되어질 분야이다. 본 논문에서는 MPEG-2에서 MPEG-4로 비디오 트랜스코딩을 할 때 MPEG-2로 디코딩하고 MPEG-4로 재인코딩하지 않고 MPEG-2로 디코딩할 때 움직임 벡터에 대한 정보와 매크로블록에 대한 타입 등의 파라미터를 추출하여 이를 수정하여 MPEG-4로 인코딩할 때 이를 재 사용함으로써 인코딩하는 시간을 줄이는 방법이다. 실험 결과 별다른 화질의 열화가 없었고 처리 시간의 향상을 보였다.

### 1. 서론

동영상 압축 표준은 지금까지 MPEG-1, MPEG-2, MPEG-4를 비롯하여 H.261, H.263, H.264 등이 제정되었거나 제정되고 있으며 제 각각의 응용 분야를 가지고 있다. 이 중에서 MPEG는 CD-ROM과 같은 디지털 저장 매체를 주 용도로 하는 MPEG-1에 이어서 MPEG-2는 고품질의 DVD와 같은 저장 매체뿐만 아니라 HDTV와 같은 디지털 방송에 적용되면서 디지털 영상 시대를 이끌어 오고 있다. 그리고 MPEG-4는 최신 규격으로 인터넷, 이동 통신을 비롯한 다양한 응용 분야에서 표준으로 사용되고 있으며 점차로 그 응용 분야가 확산되고 있는 실정이다[1].

정보 통신 기술의 발전이 가속화되면서 통신과 방송이 융합되고 인터넷과 같은 통신망의 고속화로 동영상의 이동, 보급, 전송이 늘어나고 있는 추세이다. 앞에서 언급한 것처럼 동영상 압축 규격은 지금까지 각각의 응용 분야 별로 사용되어 왔기 때문에 다양한 통신망 및 방송망이 통합되어 가는 추세에서 보면 호환성 문제를 내포하고 있는 실정이다.

MPEG-4 활용 분야는 확대되어 갈 전망이다. 따라서 기존에 다른 규격으로 제작 및 저장되어 있는 디지털 동영상을 MPEG-4 규격으로 다시 제작하거나 저장하는 것은 비경제적이라 할 수 있다. 또한 인터넷과 무선 통신망을 비롯한 다양하고 이질적인 네트워크 상호 연결되어 멀티미디어 데이터를 상호 교환하고자 하는 추세이므로 네트워크의 특성에 따라 비트율이나 품질을 조정할 수 있는 기능이 필요하다.

본 논문은 이와 같은 환경에서 기존의 대표적 동영상 압축 규격인 MPEG-2 비디오 데이터를 MPEG-4 비디오로 형식 변환하고 해상도와 프레임율을 변환할 수 있는 트랜스코딩 기술을 연구하였다.

### 2. 트랜스코딩

비디오 트랜스코딩은 그림 1처럼 인터넷 통신이 가능하도록 입력 비디오 비트 스트림의 비트율을 변환함으로써 대역폭 조건이 다른 통신 채널을 통하여 비디오 데이터의 전송을 가능하거나 특정 선택스

(syntax)로 압축된 비디오 스트림을 다른 신택스의 압축 비디오 스트림으로 변환하여 전송이 가능하도록 하는 것이다.

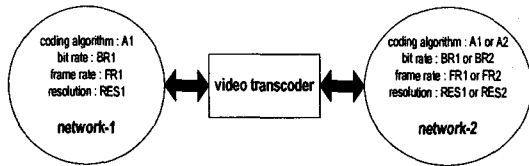


그림 1. 비디오 트랜스코딩

비디오 트랜스코딩 알고리즘은 동기종 (homogeneous) 비디오 트랜스코딩 및 이기종 (heterogeneous) 비디오 트랜스코딩의 2가지로 크게 분류할 수 있다. 일반적으로 언급되는 동기종 비디오 트랜스코딩은 전송 환경이 다른 네트워크 사이에서 동일 코딩 규격 또는 신택스를 유지하며 비트율, 프레임 율 및 해상도를 변환하는 하는 것이다. 그리고 이기종 비디오 트랜스코딩은 각 네트워크에서 사용하는 코딩 규격이 다른 환경에서 코딩 규격 또는 신택스를 변환하는 것이다[2-4].

비디오 트랜스코딩에서 가장 단순한 방법은 입력 비트스트림을 완전 디코딩한 후 원하는 새로운 규격으로 재 인코딩하는 방법이다. 그러나 완전 디코딩/재 인코딩 방법은 성능이나 복잡도 면에서 좋지 않은 방법이다[5]. 동기종 트랜스코딩 경우는 인코더와 디코더의 루프를 결합한 단순화된 트랜스코더를 유도하여 사용한다. 그러나 이기종 트랜스코딩 경우 입력 및 출력 비트스트림의 다른 인코딩 구조로 인하여 이와 같은 루프가 동일하지 않기 때문에 인코더와 디코더의 루프를 결합한 단순화된 트랜스코더를 유도할 수 없다.

### 3. MPEG-2 to MPEG-4 트랜스코딩 알고리즘

MPEG-2에서 MPEG-4로 변환하는 것과 같은 이기종 트랜스코딩은 신택스 변환 외에 픽처 타입, 픽처 해상도, 모션 벡터의 방향, 픽처율이 변화되는 특징이 있으므로 동기종 트랜스코딩에 비하여 복잡도가 크다. 이는 트랜스코더의 동작에 많은 처리 부하를 유발하기 때문에 이와 같은 병목 현상을 분석하고 이 문제를 개선해야 한다.

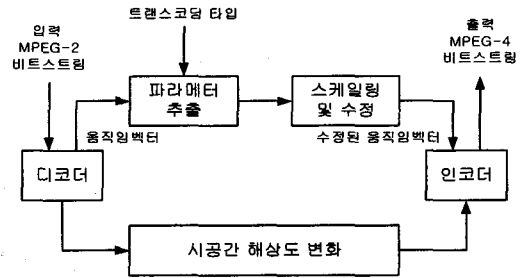
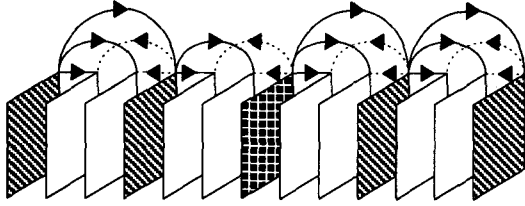


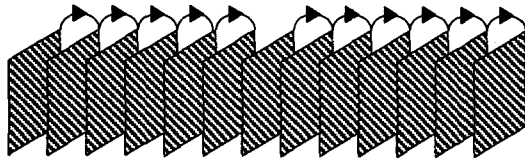
그림 2 MPEG-2 to MPEG-4 비디오 트랜스코더 기본 구성도

그림 2는 본 논문에서 제안한 MPEG-2 to MPEG-4 트랜스코더의 기본 구성을 나타낸다. 그림 2에서 입력되는 MPEG-2 비트스트림은 디코더에서 완전 복호되고 이 디코딩 과정에서 추출된 움직임 벡터는 원하는 출력 인코딩 구조인 MPEG-4에 따라 후처리되고 필요한 경우 원하는 해상도에 맞도록 스케일링된다. 후처리가 충분하지 않은 경우 추출된 모션 벡터는 인코딩 효율을 개선하도록 수정(refine)한다. 이에 따라 복호된 픽처는 공간적 또는 시간적으로 다운 샘플링되고 다운 샘플링된 이미지는 새로운 움직임 벡터로 부호화된다. 두 개의 완전한 디코더와 인코더가 필요한 것처럼 보이지만 입력 움직임 벡터를 재사용하고 있으며 매크로블록 타입과 같은 인코딩에서 필요한 판단이 입력 비트스트림에서 추출되기 때문에 위의 디코더/인코더는 완전한 디코더/인코더에 비해 단순하다. 왜냐하면 움직임 추정 및 매크로블록 판단은 일반적으로 인코딩 복잡도에서 약 70% 이상을 차지하고 있기 때문이다. 그러므로 매크로블록 판단 및 움직임 추정의 생략은 트랜스코딩 과정을 약 3배 이상 빠르게 할 수 있다.

본 논문에서 제안한 MPEG-2 to MPEG-4 트랜스코딩은 입력되는 MPEG-2 비디오 코딩은 B-픽처를 포함하고 있고 MPEG-4 비디오 코딩은 Simple Profile을 타겟으로 하기 때문에 B-픽처를 사용하지 않고 있으며 객체 기반이 아닌 프레임 기반이다. 따라서, 그림 3과 같이 비트스트림의 GOP(Group of pictures)가 12픽처 길이(N=12)이고 서브그룹의 길이가 3픽처(M=3)인 경우와 출력 픽처 시퀀스 형식이 N=12고 M=1인 경우에서 움직임 벡터를 수정하는 알고리즘은 다음과 같다.



(a) 입력 픽처, N=12, M=3



(b) 출력 픽처, N=12, M=1

그림 3 입력 및 출력 비트스트림의 픽처 타입

그림 3에서 픽처는 디스플레이 순서대로 정렬되어 있지만 순서 번호는 인코딩 순서대로 붙여져 있다. MPEG-2 to MPEG-4 트랜스코딩의 주요 특징은 입력 비트스트림의 움직임 벡터를 출력 비트스트림으로 적용하는 것이므로 추출된 움직임 벡터는 출력 비트스트림의 인코딩 속성과 호환되어야 한다. 그림 3(a)에서 각 픽처 타입에 대한 움직임 벡터를 보면 I-픽처는 모션 벡터를 이용하지 않고 P-픽처의 움직임 벡터는 세 번째 이전 픽처인 I-픽처나 P-픽처를 참조하지만 B-픽처의 움직임 벡터는 그 앵커 I-픽처나 P-픽처로부터의 전방향 및 역방향 예측을 이용한다. 반면에 그림 3(b)처럼 N=12, M=1 인 출력 비트스트림에서 P-픽처의 움직임 벡터는 이전 픽처만을 참조한다. N=12, M=3에서 N=12, M=1 인 경우로 픽처 타입의 인코딩 형식 변환의 기본 개념은 픽처 타입의 차이뿐만 아니라 입력 픽처의 모든 매크로블록이 출력 비트스트림에서 직접 사용할 수 있는 해당 전방향 움직임 벡터를 가지고 있지 않다. 그러나 이같은 경우 픽처 간의 움직임이 균일하다고 가정하면 전방향 및 역방향 움직임 벡터가 서로 대칭이라거나 또는 프레임간 움직임 벡터가 더 큰 픽처 간격의 움직임 벡터를 스케일한 값과 같다는 등으로 움직임 벡터를 근사화 할 수 있다. 여기서 고려되어야 할 픽처의 움직임 벡터는 첫 번째 B-픽처와 두 번째 B-픽처 그리고 다음에 나오는 P-픽처의

움직임 벡터에 대한 변환이다. 그림 3(a)에서  $B_{n+8}$  과 같이 입력 비트스트림의 서브그룹에서 첫 번째 B-픽처의 경우 이 픽처는 그림 3(b)와 같이  $P_{n+6}$  에서 예측되는 출력 P-픽처  $P_{n+7}$ 로 변환되어야 한다. 새로운 출력 움직임 벡터  $MV_{n+6 \rightarrow n+7}^{out}$ 은 입력 움직임과 다음과 관계가 있다.

$$MV_{n+4 \rightarrow n+8}^{fwd} \quad (1)$$

$$MV_{n+4 \rightarrow n+7}^{fwd} + MV_{n+7 \rightarrow n+8}^{bwd} \quad (2)$$

$$(1/3) MV_{n+4 \rightarrow n+7}^{fwd} \quad (3)$$

$$-(1/2) MV_{n+7 \rightarrow n+8}^{bwd} \quad (4)$$

$$(1/2) MV_{n+4 \rightarrow n+9}^{fwd} \quad (5)$$

식(1) 및 식(2)는 전방향 인터프레임 움직임 벡터의 값과 일치한다. 식(3), (4)와 (5)의 경우 출력 움직임 벡터는 움직임이 균일하다고 가정하여 입력 모션 벡터를 스케일한 값이다. 그리고 이와 유사하게 서브 그룹에서 두 번째 입력 B-픽처( $B_{n+9}$ )에 해당하는 출력 P-픽처( $P_{n+9}$ )의 움직임 벡터는 다음과 관계가 있다.

$$MV_{n+4 \rightarrow n+9}^{fwd} - MV_{n+4 \rightarrow n+8}^{fwd} \quad (6)$$

$$-(MV_{n+7 \rightarrow n+8}^{bwd} - MV_{n+7 \rightarrow n+9}^{bwd}) \quad (7)$$

$$-MV_{n+7 \rightarrow n+9}^{bwd} \quad (8)$$

$$(1/2)(MV_{n+4 \rightarrow n+7}^{fwd} - MV_{n+7 \rightarrow n+9}^{bwd}) \quad (9)$$

$$(1/2) MV_{n+4 \rightarrow n+9}^{fwd} \quad (10)$$

$$-(1/2) MV_{n+7 \rightarrow n+8}^{bwd} \quad (11)$$

$$(1/3) MV_{n+4 \rightarrow n+7}^{fwd} \quad (12)$$

그림 3(a)에서의 입력 P-픽처( $P_{n+7}$ )에 해당하는 출력 P-픽처( $P_{n+9}$ )의 움직임 벡터의 관계는 식(12)와 같다. 이렇게 얻은 움직임 벡터를 중 최적의 벡터를 선택하기 위하여 복호된 픽처는 각각의 모든 추정 움직임 벡터를 비교하여 SAD(Sum of absolute difference) 관점에서 코딩 에러가 가장 적은 것을 선택한다. 모션 벡터를 얻을 수 없는 경우 (0, 0) 모션 벡터 또는 인트라 코딩을 검사한다. 따라서 입력 움직임 중 두 번째 B-픽처에 대한 경우 비교할 경우는 9가지이므로 이는 움직임 추정에서의  $\pm 1$  픽셀 탐색과 유사하다. 따라서 움직임 벡터를 추정하는 시간을 대폭 줄인다.

본 논문에서 해상도는 CIF와 QCIF만을 고려하였다. 해상도를 CIF에서 QCIF로 변환할 때 공간적인 영역은 다운 샘플링을 하고 다운 샘플링된 각 매크로블록의 움직임 벡터는 위에서 추출한 4 개의 움직임 벡터의 산술적 평균을 구해서 새로운 움직임 벡터를 구한다. 새로운 움직임 벡터는 식(13)으로 구할 수 있다.

$$MV_{avg} = \frac{1}{4} \sum_{i=1}^4 MV_i \quad (13)$$

#### 4. 실험 결과

실험은 본 논문에서 제안한 방법과 MPEG-2로 디코딩하고 MPEG-4로 재인코딩한 방법을 비교 실험하였고 해상도 변환은 제외하였다. 실험 영상은 cactus이고 실험에 따른 각 방법에 대한 영상의 PSNR을 그림 4에 나타내었다.

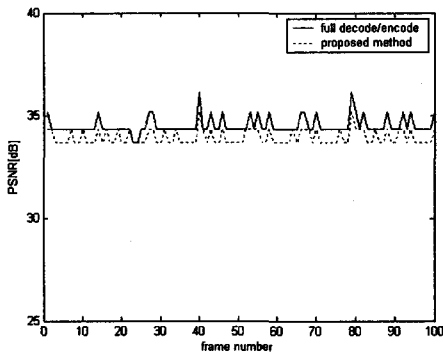


그림 4. PSNR 비교

그림 4를 보면 알 수 있듯이 각 방법에 대한 PSNR의 차이는 I-프레임인 경우에는 그 차이가 없고 다른 타입의 프레임인 경우는 그 차이가 최대 0.97 dB의 차이를 보여 별다른 화질 열화 현상은 보이지 않았다. 그리고 본 논문이 제안한 방법은 복잡도 및 계산량을 줄여 디코딩/재디코딩하는 방법에 비해 처리시간이 약 2.5배의 향상을 보였다.

#### 5. 결론

본 논문은 MPEG-2에서 MPEG-4로 비디오 트랜스코딩을 할 때 MPEG-2로 디코딩하고 MPEG-4로

재인코딩하지 않고 MPEG-2로 디코딩할 때 움직임 벡터에 대한 정보와 매크로블록에 대한 타입 등의 파라미터를 추출하여 이를 수정하여 MPEG-4로 인코딩할 때 이를 재 사용함으로써 인코딩하는 시간을 줄이는 방법에 대하여 제안하였다. 실험 결과 디코딩/재인코딩한 방법에 비해 약간의 화질 열화는 있었지만 고려될만한 정도는 아니었고 처리시간에 대해서는 약 2.5배의 향상을 보여 제안된 알고리즘 기법이 우수함을 확인할 수 있었다.

#### 참고문헌

- [1] ISO/IEC JTC1/SC29/WG11 N2802: Information technology - Generic coding of audio-visual objects - Part2: visual, Vancouver meeting, July 1999.
- [2] M. Ghanbari, "Two-layer coding of video signals for VBR networks", IEEE Journal on Selected Areas in Communications, No.5, pp.771-781, June 1989.
- [3] N. Bjojk, C. Christopoulos, "Transcoder architectures for video coding", IEEE Transactions on Consumer Electronics, No. 1, pp.88-98, Feb. 1998.
- [4] K. S. Kan, K. C. Fan, "Video transcoding architecture with minimum buffer requirement for compressed MPEG-2 bit stream", Signal Processing, No.2, pp.223-225, June 1998.
- [5] G. Keesman, R. Hellinghuizen, F. Hoeksema, G. Heigeman, "Transcoding of MPEG bit streams", Signal Processing: Image Communication, No.6, pp.481-500, Sept. 1996.
- [6] M. T. Sun, T. D. Wn, J. N. Hwang, "Dynamic bit allocation in video combining for multipoint conferencing", IEEE Transactions on Circuits and System-II: Analog and Digital Signal Processing, No.5, pp.644-648, May 1998.
- [7] Tamer Shanableh, Mohammed Ghanbari, "Heterogeneous Video Transcoding to Lower Spatio-Temporal Resolutions and Different Encoding Formats", IEEE Transactions on Multimedia, Vol.2, No.2, June 2000.