
움직임벡터 군집화를 이용한 H.264/AVC에서 MPEG-2로의 비디오 트랜스코딩

신윤정* · 손남례** · Nguyen Dinh Toan*** · 이귀상****

H.264/AVC to MPEG-2 Video Transcoding by using Motion Vector Clustering

Yoon-Jeong Shin*, Nam-Rye Son**, Nguyen Dinh Toan***, Guee-Sang Lee****

요 약

H.264/AVC는 기존 비디오 코덱보다 성능이 우수하여 최근 IPTV, DMB등의 압축표준으로 사용되고 있다. 따라서 H.264/AVC로 압축된 데이터를 이전 코덱을 사용하는 장비에서 이용하기 위하여 트랜스코딩 연구가 이루어지고 있다. 본 연구에서는 움직임벡터의 군집화(Clustering)를 이용한 H.264/AVC에서 MPEG-2로의 트랜스코딩 방법을 제시한다. H.264/AVC 비트스트림을 MPEG-2의 인코더로 보낼 때 H.264/AVC 가변블록의 움직임벡터들의 거리와 방향성을 고려한 클러스터링을 수행하여 후보벡터를 선택한 후 최소의 왜곡치를 갖는 1개의 움직임벡터로 최종 결정한다. 이렇게 선정된 최종 움직임벡터는 MPEG-2 인코더에서 ± 2 pixel 만큼 전역탐색으로 보정 한 후 재사용하는 방법으로 트랜스코딩 시간을 최소화하고자 한다. 실험을 통하여 계산시간과 비디오 화질을 비교한 결과 기존연구보다 PSNR값이 최대 6.7% 향상되었으며 부호화 시간은 최대 64% 개선되었다.

ABSTRACT

The H.264/AVC is increasingly used in broadcast video applications such as Internet Protocol television (IPTV), digital multimedia broadcasting (DMB) because of high compression performance. But the H.264/AVC coded video can be delivered to the widespread end-user equipment for MPEG-2 after transcoding between this video standards. This paper suggests a new transcoding algorithm for H.264/AVC to MPEG-2 transcoder that uses motion vector clustering in order to reduce the complexity without loss of video quality. The proposed method is exploiting the motion information gathered during h.264 decoding stage. To reduce the search space for the MPEG-2 motion estimation, the predictive motion vector is selected with a least distortion of the candidated motion vectors. These candidate motion vectors are considering the correlation of direction and distance of motion vectors of variable blocks in H.264/AVC. And then the best predictive motion vector is refined with full-search in ± 2 pixel search area. Compared with a cascaded decoder-encoder, the proposed transcoder achieves computational complexity savings up to 64% with a similar PSNR at the constant bitrate(CBR).

키워드

H.264/AVC, MPEG-2, transcoding, motion vector, clustering

* 교신저자, 광주대학교 컴퓨터공학과 부교수(syj@gwangju.ac.kr) ** 목포대학교 중점연구소(nrson72@gmail.com)

*** 전남대학교 전자정보통신컴퓨터공학부(toan_mulmi@hotmail.com, gslee@chonnam.ac.kr)

접수일자 : 2010. 1. 3

심사완료일자 : 2010. 2. 1

1. 서 론

MPEG-2 표준은 비디오 CD 매체용으로 제한된 MPEG-1 표준을 기반으로 4Mbps 이상의 속도에서 TV화질을 유지하는 것을 목표로 하였다. 그러나 MPEG-2 표준화 작업은 HDTV 표준을 위한 MPEG-3 표준을 흡수하여 HDTV 품질까지 표준화 대상으로하기로 결정하고 1994년 11월 국제 표준으로 승인되었다.[1] MPEG-2 표준은 DMB (Digital Multimedia Broadcasting), HDTV (High Definition Digital Television), DVD (Digital Video Disc), 스트리밍 비디오, 전자 영화관, VOD (Video on Demand) 등 현재 많은 분야에서 표준으로 채택되어 멀티미디어 서비스의 주요 압축표준이 되어 왔다.

그러나 가장 최신의 동영상 압축 표준인 H.264/AVC는 다양한 네트워크 환경에 쉽게 적용할 수 있는 유연성과 동영상의 부호화 효율성 측면에서 MPEG-2, MPEG-4, H.263 등의 성능을 능가한다. 현존하는 비디오 표준들 중 가장 높은 부호화 효율을 가지는 H.264/AVC는 위성 DMB서비스의 동영상 표준으로 채용되었으며 전송 대역폭이 낮은 이동통신에서의 활용이 높을 것으로 예상된다[2].

그러나 기존에 널리 사용되던 HDTV, DVD 플레이어 등의 기존장비로 H.264/AVC 콘텐츠를 전송하기 위해서는 H.264 비트스트림을 MPEG-2 비트스트림으로 변환해 주는 과정이 필요하다. 이를 동영상 변환 또는 비디오 트랜스코딩이라고 부른다.

비디오 트랜스코딩의 종류는 변환 대상에 따라 비트율, 프레임율, 해상도 등을 조절하는 동종간 트랜스코딩과 이질적인 네트워크 환경과 다양한 사용자 단말간에 비디오 서비스가 가능하도록 특정 표준으로 압축된 비디오 스트림에서 원하는 다른 압축된 비디오 스트림으로 변환하는 이종간 트랜스코딩이 있다. 이종간 트랜스코딩에는 변환 방법에 따라 직렬화소 영역기반 트랜스코더와 DCT기반 트랜스코더가 있다. 본 논문에서 다루고 있는 픽셀 기반 트랜스코더분야에서는 H.264/AVC 표준이 지원하는 가변블록 움직임벡터들의 정보를 활용하여 MPEG-2에서 움직임추정에 이용하는 다양한 연구가 이루어지고 있다[3-9].

본 논문에는 화질의 열화를 줄이기 위해서 직렬화소 영역기반 트랜스코더 방식을 사용하고, 이 방식의 단점

으로 지적되는 계산량을 줄이기 위하여 H.264/AVC 복호기에서 전송되는 매크로블록 모드와 움직임벡터 정보를 분석 활용하여 MPEG-2 인코더에서 움직임벡터를 재사용하는 방법을 제안한다.

이 논문의 구성은 다음과 같다. 2장에서는 MPEG-2와 H.264/AVC 표준의 특징 및 차이점과 트랜스코딩 구조를 설명하고 기존 트랜스코딩 방법을 소개한다. 3장에서는 움직임추정 계산량을 최소화하고 화질을 유지할 수 있는 트랜스코딩 설계를 위하여 H.264/AVC의 가변블록 움직임벡터들을 클러스터링하여 후보 움직임벡터를 생성하고 최소 왜곡값을 갖는 움직임벡터를 선택한 후 보정하여 MPEG-2 인코더에서 재사용하는 방법을 제안한다. 5장에서는 본 논문에서 제안한 방법과 기존연구의 성능을 실험을 통해 평가하고 결론을 맺는다.

II. MPEG-2에서 H.264/AVC로의 트랜스코딩

2.1 MPEG-2와 H.264/AVC의 주요 차이점

MPEG-2 표준의 부호화과정은 인트라프레임, 인터프레임이 공통으로 DCT, 양자화, VLC를 수행하고 인터프레임의 경우 압축효율을 높이기 위해 중복성을 제거하는 프레임 간 움직임 예측과 움직임 보상 과정이 추가된다. 움직임 예측은 16×16 매크로 블록 단위로 복호화된 참조 프레임을 대상으로 최소의 왜곡값을 갖는 위치를 움직임벡터로 표현한다. 움직임 보상은 움직임 예측을 수행한 후 참조프레임과 차이값을 구하고 이를 부호화하여 정보의 양을 줄인다. H.264/AVC와 MPEG-2는 서로 다른 블록 크기로 움직임 예측 기법이 수행되므로 트랜스코딩 시 문제점이 발생한다. 먼저 H.264/AVC 표준에서는 인터프레임의 경우 시간 중복성을 제거하기 위해 이전 프레임 참조하는데 프레임의 개수를 최대 16 프레임으로 하여 압축 효율을 높이고 인터프레임의 움직임 예측 및 보상에서 사용되는 블록 단위를 영상의 복잡도에 따라 16×16, 16×8, 8×16, 8×8, 8×4, 4×8, 4×4로 세분화 하였다. 또한 움직임 예측 범위를 1/4 픽셀 단위로 하여 MPEG-2의 1/2 픽셀 단위보다 좀 더 정확성을 높일 수 있다. 표 1은 두 표준의 주요 차이점을 나타낸 것이다.

표 1. H.264/AVC와 MPEG-2의 주요 차이점
Table 1. Main Differences H.264/AVC and MPEG-2

특징	MPEG-2	H.264
움직임예측 정밀도	1/2 pel	1/4 pel
움직임벡터	프레임 경계 제한	경계 제한 없음
참조프레임	P 픽처 1개 B 픽처 2개까지	최대 16개
매크로블록 분할	16×16 16×8 (interlace)	16×16, 16×8, 8×16 8×8, 8×4, 4×8, 4×4
인트라 예측	없음	있음 (Spatial Domain)
전송율	2-15Mbps	64Kbps-150Mbps
엔트로피 코딩	VLC	CAVLC, CABAC
프로파일	5 profiles	7 profiles
이전 표준과의 호환성	있음	없음

2.2 트랜스코딩 구조

H.264/AVC 비트스트림을 MPEG-2 비트스트림으로 변환하는 가장 간단하고 강력한 트랜스코딩 방법은 그림 1과 같이 입력으로 들어오는 H.264 비트스트림을 모두 복원하여 MPEG-2에서 입력영상으로 받아들인 후 다시 부호화를 거쳐 압축하는 직렬 트랜스코딩 방법이다. 이 구조는 다른 트랜스코더 구조나 알고리즘과 비교하기 위하여 사용된다. 그러나 이 방법은 MPEG-2 부호화과정 소요 시간 중 60~70% 이상의 시간을 차지하는 움직임 추정 과정을 거쳐야 하므로 실시간 응용분야에서는 적합하지 않다. 따라서 MPEG-2 부호화 시 계산시간을 줄이기 위해 H.264/AVC 비트스트림에 포함된 움직임벡터 정보를 이용하려는 연구가 진행되어왔다[3-9].

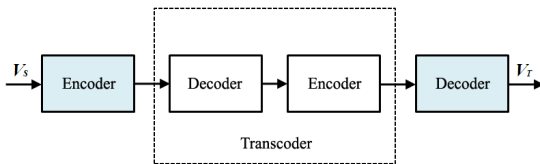


그림 1. 직렬 트랜스코더 구조
Fig. 1 Cascade Transcoder Architecture

그림 2는 개선된 트랜스코딩 구조를 보여준다[5]. 이 방법에서는 변수 변환 모듈(parameter converter

module)을 통하여 MB 모드, 픽처 타입, 움직임벡터와 같은 H.264/AVC 정보를 추출하여 MPEG-2에 이용 가능한 구조로 변환시킨다. MPEG-2 인코더는 움직임 추정 과정 없이 입력되는 H.264/AVC 비트스트림의 움직임 정보를 재사용하기 때문에 트랜스코딩 속도를 향상시킬 수 있다.

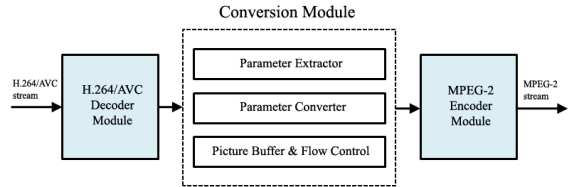


그림 2. 개선된 트랜스코더 구조
Fig. 2 Modified Transcoder Architecture

H.264/AVC 표준에서는 그림 3과 같이 움직임벡터가 최소 1개에서 최대 16개까지 가지고 있지만 MPEG-2 인코더에서는 16×16 블록에 대한 움직임벡터 1개를 사용한다. 그러므로 H.264/AVC 비트스트림의 매크로블록 움직임벡터들은 MPEG-2에서 부호화가 가능하도록 1개의 움직임벡터로 변환되어 해야 한다. 이 과정에서 계산량 감소를 위해 검색영역을 재조정하거나 정확도가 높은 예측 움직임벡터를 생성하기 위한 연구들이 진행되고 있다.

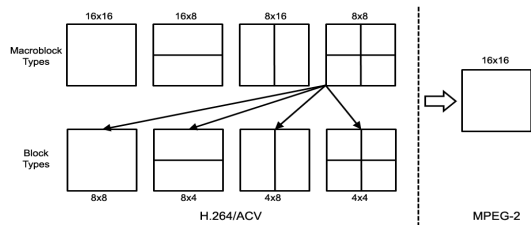


그림 3. 매크로블럭 분할 타입
Fig. 3 Macroblock Partition Type

2.3 가중치 평균법 트랜스코딩 [4]

이 방법은 인터프레임 트랜스코딩시 H.264의 가변블록 움직임벡터를 평균값으로 구하여 MPEG-2의 움직임벡터 정보로 전송하는 방법으로 평균값을 구할 때 블록 크기에 따라 가중치를 두어 평균값을 구한다. 트랜스코딩 과정은 다음 4단계로 진행된다. 첫째, H.264 디코딩과정에서 움직임벡터 추출하고 둘째, 블록크기 정보

를 이용하여 움직임벡터를 리샘플링하며 셋째, 앞 단계에서 계산된 움직임벡터를 정교화시키고 마지막으로 최종 결정된 움직임벡터를 MPEG-2 인코딩에서 재사용한다.

움직임벡터 리샘플링은 MB 타입에 따라 가중치를 적용한 움직임벡터의 평균값을 구하는 것으로 각 MB타입에 따른 계산식은 표 2와 같다. $H.264mv_x$ 와 $H.264mv_y$ 는 H.264 디코더에서 사용하는 16x16매크로블록의 x와 y의 움직임벡터이고, $MPEG2mv_x$ 와 $MPEG2mv_y$ 는 MPEG-2 인코더에서 16x16매크로블록단위로 사용할 x와 y의 예측된 움직임벡터이다. 예를 들어 매크로블록크기가 16x8인 경우 수평방향 움직임벡터인 $H.264mv_{ax}$ 과 $H.264mv_{bx}$ 는 더하여 구한다. 8x16인 경우 수평과 수직방향 반대로 계산하여 평균 움직임벡터를 구하고, 8x8 블록인 경우 H.264의 움직임벡터 4개를 합하여 평균한 움직임벡터를 각각 움직임벡터로 예측한다. 마지막으로 매크로블록 타입이 3 또는 4(8x4,4x8,4x4)인 경우는 매크로블록안의 서브매크로블록을 크기방향에 따라 16x16매크로블록에서 사용한 것과 동일하게 적용하여 계산한다.

세 번째 단계인 움직임벡터 정교화과정에서는 HAVS(horizontal and vertical search)를 이용한다. 이 방법은 앞 단계에서 계산된 움직임벡터를 수평으로 검색하여 최소 SAD값을 가진 위치를 정하고 다시 그 위치의 수직 방향에서 최소 SAD값 갖는 위치를 최종 움직임 벡터로 결정하는 것이다. 이 방법은 계산시간은 빠르나 영상 화질은 직렬 트랜스코더보다 상당히 떨어지는 단점이 있다.

표 2. 움직임벡터 재조정
Table 2. Motion Vector Resampling

가변블록 크기	식
16x16	$MPEG2mv_x = H.264mv_x$ $MPEG2mv_y = H.264mv_y$
16x8	$MPEG2mv_x = (H.264mv_{ax} + H.264mv_{bx})/2$ $MPEG2mv_y = (H.264mv_{ay} + H.264mv_{by})/2$
8x16	$MPEG2mv_x = (H.264mv_{ax} + H.264mv_{bx})/2$ $MPEG2mv_y = (H.264mv_{ay} + H.264mv_{by})/2$
8x8	$MPEG2mv_x = (H.264mv_{x1} + H.264mv_{x2} + H.264mv_{x3} + H.264mv_{x4})/4$ $MPEG2mv_y = (H.264mv_{y1} + H.264mv_{y2} + H.264mv_{y3} + H.264mv_{y4})/4$

III. 제안 방법

본 연구에서는 H.264/AVC에서 MPEG-2로 트랜스 코딩을 수행할 때 H.264/AVC 비트스트림에 존재하는 파라미터 중 가변블록 움직임벡터를 재사용하는 방법을 제안한다. 여기서 MPEG-2 인코더는 별도의 움직임 추정 없이 부호화할 수 있기 때문에 트랜스코딩 속도가 개선된다. 제안된 트랜스코딩 방법에서 H.264/AVC 가변블록 움직임벡터들은 거리와 방향성을 이용하여 클러스터링되며 각 클러스터의 대표값을 후보예측움직임벡터로 생성한다. 따라서 후보예측움직임벡터의 수가 줄어들기 때문에 계산 속도를 더욱 향상시킬 수 있다. 다음 단계에서 최소 왜곡값을 갖는 후보예측움직임벡터를 MPEG-2의 예측움직임벡터로 결정하고 정교화를 수행한다.

3.1 후보 움직임벡터 선정

그림 4와 같이 H.264/AVC 디코더의 가변 매크로블록 안에 서로 다른 방향의 움직임 벡터들이 존재한다. 본 논문에서는 수정된 평균연결알고리즘으로 거리와 방향성을 고려하여 클러스터링을 하여 후보예측 움직임벡터 수를 줄일 수 있다. 이 과정은 다음과 같다. 먼저 H.264/AVC 디코더의 매크로블록안에 존재하는 움직임벡터들을 검출한 다음, 식(1)을 이용하여 움직임 벡터간의 거리를 구하고 최소가 되는 두 블록간의 평균값을 계산하는 과정을 반복함으로써 평균연결알고리즘을 수행한다. 식(1)에서 움직임 벡터는 (x_n, y_n) 와 (x_{n-1}, y_{n-1}) 이고, D 는 움직임 벡터간의 거리이다.

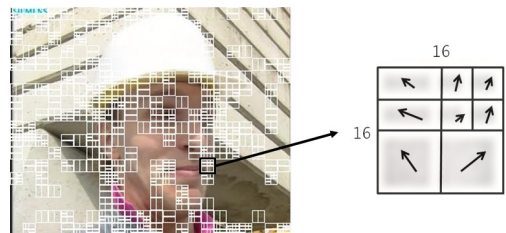


그림 4. 가변매크로블록 움직임 벡터
Fig. 4 Motion Vectors in Variable Macroblock

$$D = \sqrt{|x_{n-1} - x_n|^2 + |y_{n-1} - y_n|^2}, (n \geq 2) \quad (1)$$

그림 5(a)는 매크로블록의 다양한 방향을 가진 움직임 벡터들에 대하여 평균연결알고리즘을 사용하여 두 개의 클러스터(G1, G2)를 생성한 결과이다. 즉, 평균연결알고리즘은 항상 최종적으로 두 개의 클러스터를 생성하므로 클러스터링된 움직임 벡터들은 거리가 떨어져 있어도 같은 그룹을 생성하는 단점이 있다. 따라서 본 논문에서는 그림 5(b)와 같이 임계값을 수정하여 사용하였는데 움직임 벡터간의 거리가 임계값(=3) 이상일 경우 클러스터 G1은 여러 개의 클러스터 G1, G3, G4로 나누어진다. 이렇게 생성된 각 클러스터의 중앙값(median value)을 후보예측움직임벡터로 결정한다.

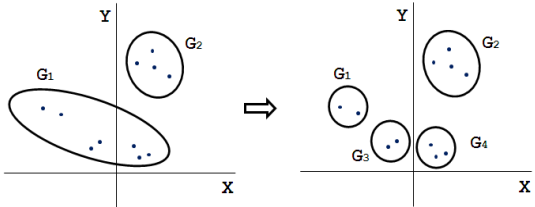
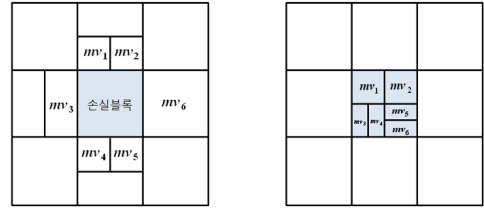


그림 5. ALA를 이용한 움직임벡터 클러스터링
Fig. 5 Motion Vector Clustering by using ALA

클러스터링 임계값이 참고논문[10]에 비해 더 적은 값인 이유는 [10]의 경우 그림 6(a)에서처럼 여러 은닉 시에는 주변의 움직임 벡터를 사용하지만 본 논문에서는 그림 6(b)와 같이 같은 블록 내에서의 움직임 벡터들을 클러스터링 하기 때문이다. 따라서 움직임 벡터들의 관련성이 그림 6(a)에 비해 더 크기 때문에 움직임 벡터들의 거리 값이 더 적다. 이 경우 클러스터링을 위한 임계값으로 큰 값을 사용하게 되면 대부분 동일한 클러스터가 될 수 있어서 정확한 후보 예측 움직임 벡터가 선택되기 어렵고 후보 예측 움직임 벡터 수가 적어 계산량은 줄어들 수 있지만 화질이 저하될 수 있다. 제안한 알고리즘에서는 제안논문[10]의 임계값보다 적은 임계값 3으로 클러스터를 생성하였다. 영상의 복잡도가 높아 가변블록 움직임 벡터들이 여러 개 존재하는 경우 MPEG-2 인코더에서 모든 움직임 벡터들에 대한 움직임 추정을 수행하기 않아도 되기 때문에 트랜스코딩 속도를 개선하는데 도움이 된다.



(a) 주변블록 후보 움직임벡터들 (b) 현재블록 내 후보 움직임벡터들

그림 6. 후보움직임벡터 상관관계
Fig. 6 Correlation Of Candidate Motion Vectors

3.2 최종 움직임벡터 선정

앞 단계에서 선정된 후보예측후보벡터들 중에서 MPEG-2의 최적 움직임 벡터를 선정하기 위하여 식(2)의 비용함수를 사용한다. 그림 7과 같이 현재 매크로블록내의 움직임 벡터들마다 매크로블록 내부의 경계면 영역과 참조영상과의 MAD를 통해 식(2)의 오차값을 계산한다. diff는 상(dT)하(dB)좌(dL)우(dR) 오차값의 합인데, 이 값이 가장 작은 후보 예측움직임 벡터들 중에서 MPEG-2인코더의 예측움직임 벡터로 결정한다.

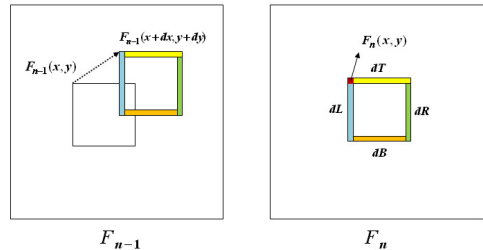


그림 7. 제안한 비용함수
Fig. 7 Distortion Function

$$\begin{aligned}
 dL &= \sum_{i=0}^{15} |f_n(x, y+i) - f_{n-1}(x+dx, y+dy+i)| \\
 dT &= \sum_{i=1}^{15} |f_n(x+i, y) - f_{n-1}(x+dx+i, y+dy)| \\
 dR &= \sum_{i=1}^{15} |f_n(x+15, y+i) - f_{n-1}(x+dx+i, y+dy+i)| \\
 dB &= \sum_{i=0}^{15} |f_n(x+i, y+15) - f_{n-1}(x+dx+i, y+15)| \\
 diff &= dL + dT + dR + dB \quad (2)
 \end{aligned}$$

3.3 움직임벡터 정교화

본 절에서는 앞 절에서 결정된 예측움직임벡터를 그대로 사용하지 않고 화질 개선을 위해 움직임벡터 정교화 과정을 수행한다. 최적의 예측움직임벡터를 결정하기 위하여 ± 2 픽셀 내 반화소 검색을 수행하고 최소 왜곡 값을 갖는 예측움직임벡터를 최종 움직임벡터로 결정한다. 이때 탐색영역을 ± 2 픽셀 내에서 수행하는 이유는 다음 장의 실험결과에서 제시하듯 H.264/AVC 비트스트림에서 한 매크로블록 안에 존재하는 움직임벡터들이 MPEG-2 인코더에서 같은 위치의 매크로블록에 대하여 전역 탐색 움직임추정 후 생성되는 움직임벡터 값의 ± 2 픽셀 범위 내에 존재할 확률이 평균 93%이기 때문이다.

IV. 실험 및 결과

본 논문에서 본 실험은 H.264/AVC JM 10.2 디코더를 이용하여 베이스 프로파일인 H.264/AVC JM 10.2 인코더를 사용한 것을 디코딩하였으며 실험영상들은 B프레임을 제외한 15 프레임의 GOP 사이즈로 1Mbps/sec의 비트율로 부호화하였다. 또한 양자화 파라미터는 24, 움직임벡터 검색범위는 ± 15 , 참조프레임은 1개, CABAC, RD optimization를 사용하였다. 여기서 생성된 H.264/AVC 비트스트림을 MPEG-2 표준 인코더의 입력으로 사용하였다. 실험 PC 환경은 AMD Dual Core 4600+ CPU, 1G RAM이며 테스트 영상은 CIF(352x288)와 SIF(352x240)크기의 영상을 가지고 각 30프레임씩 테스트하였다.

표 4은 각 실험영상에서 H.264/AVC 매크로블록 움직임벡터들이 같은 위치의 MPEG-2 매크로블록 움직임벡터의 ± 2 픽셀 내에 존재할 확률을 나타낸다. 제안한 알고리즘에서 움직임벡터 정교화 시 반화소 검색 범위를 ± 2 픽셀로 사용하는 근거이다.

표 4. 움직임벡터 일치율
Table 4. Consistent Rate for Motion Vector

영상 방법	BUS	SKI	COST GUARD	FORE MAN	STE FAN
± 2 이내	93.33%	90.23%	96.41%	93.85%	87.03%

표 5는 제안한 방법과 직렬 트랜스코더, 가중치평균법을 이용한 트랜스코더의 PSNR값을 비교하여 보여준다. 제안한 방법의 PSNR값은 '가중치 평균법'과 비교하여 각각 평균 0.92dB정도 증가하였고 직렬 트랜스코더와 비슷한 PSNR값을 가진다.

표 5. PSNR 결과 비교
Table 5. PSNR Results for Sequences

실험영상	직렬 트랜스코더	가중치 평균법	제안방법
foreman	36.62	35.93	36.58
bus	30.71	28.96	30.62
coastguard	34.07	33.94	34.06
ski	42.79	42.70	42.99
stefan	30.35	28.39	30.28

표 6는 '전역탐색'과 각 방법 간의 계산속도를 비교한 것이다. 제안한방법은 직렬 트랜스코더에 비해 각 계산 속도를 57~64% 정도 개선하였다.

표 6. 트랜스코딩 시간 (단위:ms)
Table 6. Tanscoding Time

실험영상	직렬 트랜스코더	가중치 평균법	제안방법
foreman	57.50	22.03	24.15
bus	32.95	11.32	11.81
coastguard	56.62	22.09	24.22
ski	59.24	21.11	20.94
stefan	65.11	21.69	24.06

그림 8은 'stefan' 영상에 대하여 움직임벡터 정교화를 수행하지 않은 경우와 정교화를 위한 탐색영역 범위에 따른 PSNR값을 보여준다. 움직임벡터 정교화 없이 트랜스코딩을 한 경우 ± 2 픽셀을 사용한 경우와 비교하여 화질이 저하되는 것을 볼 수 있다. 그리고 움직임벡터 정교화를 위한 전역탐색 범위로 ± 2 픽셀을 사용한 경우의 PSNR값과 ± 15 픽셀을 사용한 경우의 PSNR값은 거의 차이가 나지 않는다. 그러나 ± 15 픽셀 범위로 전역탐색을 한 경우에는 움직임 보정 과정에 많은 계산량을 요구한다. 따라서 본 논문에서는 움직임벡터 보정 범위로 계산량을 줄이면서 화질 저하가 거의 없는 ± 2 픽셀을 사용하였다.



그림 8. 움직임벡터 보정 범위에 따른 PSNR 비교
Fig. 8 PSNR by Motion Vector Refinement Search Range

그림 9는 제안한 방법과 직렬 트랜스코더에서 stefan 영상을 트랜스코딩 하였을 경우 프레임별 소요되는 트랜스코딩 시간을 나타낸다. 제안한 방법은 직렬 트랜스코더 대비 평균 60%의 계산속도를 향상 시킬 수 있으며 모든 프레임에서 영상 내 움직임의 크기와 상관없이 항상 일정한 트랜스코딩 속도를 유지할 수 있다.

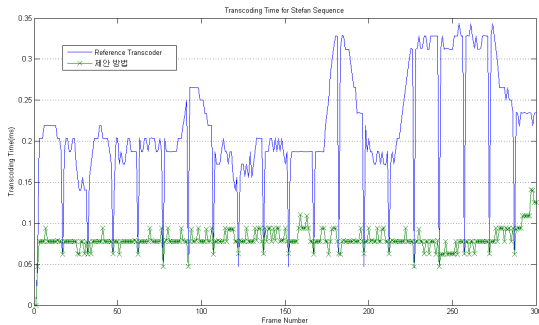


그림 9. 트랜스코딩 시간 비교
Fig. 9 Comparison of Transcoding Time

V. 결 론

본 논문에서는 H.264/AVC 비디오스트림에 존재하는 움직임 정보를 MPEG-2 인코더에서 사용 가능한 형식으로 변환 후 재사용 하여 움직임 추정 없이 MPEG-2 부호화가 가능하도록 하여 좋은 화질을 유지하면서도 계산 속도를 개선한 트랜스코딩 방법을 제안하였다. 실험결과 직렬 트랜스코더와 비교할 경

우 계산 복잡도를 평균 57~64% 정도 줄이면서도 PSNR값은 비슷하게 유지하였고 기존 트랜스코딩 방법보다 0.05~0.92dB 높은 PSNR값을 가진다. 또한 어떤 프레임에서도 일정한 트랜스코딩 속도를 유지하기 때문에 제한한 트랜스코딩 방법은 실시간 멀티미디어 서비스에도 적용될 수 있을 것이다.

감사의 글

본 연구는 지식경제부 및 정보통신산업진흥원의 대학 IT연구센터 지원사업의 연구결과로 수행되었음 (NIPA-2009-(C1090-0903-0008))

참고 문헌

- [1] ISO/IEC 13818-2, "Information technology - Generic coding of moving pictures and associated audio information-Part 2: Video", 1995.
- [2] TTA Standard, "Standard for Satellite Digital Multimedia Broadcasting Transmitter / Receiver Interface", TTAS.KO-07.0027 Sep. 2004.
- [3] Yang, L., Song, X., Hou, C., Dai, J., "H.264/MPEG-2 transcoding based on personal video recorder platform", Proc. of the Ninth International Symposium on Consumer Electronics, P.438-440, 2005.
- [4] S. Sharmat, "Transcoding of H.264 BITSTREAM to MPEG-2 BITSTREAM", Master's thesis, 2007.
- [5] Moiron, S., S.M.M. Faria, P.A. Assuncao, V. Silva, A. Navarro, "H.264/AVC to MPEG-2 Video Transcoding Architecture", Proc Conf. on Telecommunications - ConfTele, Peniche, Portugal, Vol. 1, pp. 449 - 452, May, 2007.
- [6] Kunzelmann, P., Kalva, H., "Reduced complexity H.264 to MPEG-2 transcoder", ICCE International Conference on Consumer Electronics, 2007.
- [7] Moiron, S., Faria, S., Assuncao, P., Silva, V., Navarro, A., "Fast interframe transcoding from H.264/AVC to MPEG-2", ICIP, 2007.
- [8] Moiron, S., Faria, S., Assuncao, P., Silva, V., Navarro A., "Mode Conversion for H.264 to MPEG-2 Video Transcoder", Proc Picture Coding Symposium, 2007.

- [9] Moiron, S., Faria, S., Navarro, A., Silva V., Assuncao, P., "Video transcoding from H.264/AVC to MPEG-2 with reduced computational complexity" *Signal Processing : Image Communication* 24, p637-650, 2009.
- [10] 손남례, 이귀상, "움직임벡터의 거리와 방향성을 고려한 H.264 에러은닉방법", *한국통신학회 논문지*, 34권, 1호, pp34-47, 2009.

저자 소개



신윤정(Yoon-Jeong Shin)

1994년 2월 목포대학교 컴퓨터공학과 학사
 1998년 2월 경희대학교 전산교육 전공 석사

2001년 8월 전남대학교 전산학과 박사수료
 1999년 3월~현재 광주대학교 컴퓨터공학과 부교수
 ※ 관심분야 : 비디오 코딩, 멀티미디어 콘텐츠, 이러닝



손남례(Nam-Rye Son)

1995년 2월 호남대학교 컴퓨터공학과 학사
 1999년 8월 전남대학교 전산학과 석사

2005년 2월 전남대학교 전산학과 박사
 2007년 3월~2009년 2월 호남대학교 인터넷소프트웨어학과 전임강사
 2009년 12월~현재 목포대학교 중점연구소 연구교수
 ※ 관심분야 : 비디오 코딩, e-learning, 방송통신융합 기술, 영상통신



NGUYEN DINH TOAN

September, 2005: Received the B.S. degree in Information Technology from Vietnam National University - Ho Chi Minh City, University of Natural Sciences.

February, 2008: Received the M.S. degree in Computer Science from Chonnam National University, Korea. Currently pursuing Ph.D. degree in Computer Science at Chonnam National University, Korea

※ 관심분야 : H.264 video standard, text image processing, tensor voting



이귀상(Guee-Sang Lee)

1980년 2월 서울대학교 전기공학과 학사
 1982년 2월 서울대학교 전기공학과 석사

1991년 8월 Pennsylvania 주립 대학교 전산학박사
 1984년 3월~현재 전남대학교 전자컴퓨터공학부 교수
 ※ 관심분야 : 멀티미디어통신, 영상처리 및 컴퓨터 비전, 임베디드 시스템