

가변블럭의 경계정합방법을 이용한 H.264 to MPEG-2 Video Transcoding 연구

정회원 손남례*, 정민아**, 이성로***, 이귀상****

A Study using Variable Blocks of Boundary Matching Method for H.264 to MPEG-2 Video Transcoding

Nam-rye Son*, Min-A Jung**, Sung-ro Lee***, Guee-sang Lee**** *Regular Members*

요 약

성능이 우수한 H.264 동영상압축표준이 발표된 이후, 기존 동영상압축표준인 MPEG-2에서 H.264표준으로 대체되었고, 이로 인하여 기존에 MPEG-2표준으로 압축된 콘텐츠들은 H.264표준을 따르는 디바이스에 적합한 트랜스코딩 방법들이 연구되어왔다. 또한 H.264표준으로 압축된 동영상은 현재 IPTV 및 DMB 등 서비스를 제공하고 있으나 MPEG-2 표준 디바이스를 사용하고 있는 사용자들은 시청할 수 없게 되었다. 따라서 본 논문에서는 MPEG-2 표준 디바이스를 사용하고 있는 사용자들은 H.264표준을 이용할 수 있는 H.264 to MPEG-2 트랜스코딩을 제안한다. 제안한 알고리즘은 H.264인코더의 특징인 가변블럭의 움직임벡터를 MPEG-2의 매크로블럭에 적합한 한 개의 움직임벡터를 예측하여 MPEG-2인코더의 계산량을 향상시키는 방법이다. 이때 물체의 경계와 배경의 특성을 파악하여 수정된 경계정합알고리즘을 사용하여 최적의 움직임벡터를 예측한다. 실험 결과, 제안한 방법은 MPEG-2 인코더의 “전역탐색”보다 PSNR은 거의 비슷하면서 부호화 계산 속도는 평균적으로 약 65% 감소하였다.

Key Words : H.264/AVC, MPEG-2, Transcoding, Motion Estimation, Motion Vector, BMA(Boundary Matching Algorithm)

ABSTRACT

After the efficiency of H.264 video compression has been announced, it replaced MPEG-2 standard in several applications. So transcoding methods of MPEG-2 to H.264 have been studying because there are variety devices and contents followed by MPEG-2. Although H.264 supported various service such as IPTV, DMB, digital broadcasting etc, but users using MPEG-2 devices cannot accessible to them. This paper propose H.264 to MPEG-2 transcoding for users of MPEG-2 devices without displacement H.264. The proposed method predicted a motion vector for MPEG-2 encoder after it extracted from motion vectors of variable blocks in H.264 to improve processing time. Also it predicted a optimal motion vector using modified boundary matching algorithm after grasped a special character for boundary and background of object. The experimental results from proposed method show a considerable reduction in processing time, as much as 65% averagely, with a small objective quality reduction in PSNR.

※ 이 논문은 2009년도 정부(교육과학기술부)의 재원으로 한국연구재단의 대학중점 연구소 지원 사업으로 수행된 연구임(2009-0093828)

※ 본 연구는 지식경제부 및 정보통신산업진흥원의 대학IT연구센터지원사업의 연구결과로 수행되었음(NIPA-2009-C1020-0902-0010)

* 목포대학교 중점연구소(nrson72@gmail.com), **목포대학교 컴퓨터공학과(majung@mokpo.ac.kr), ***목포대학교 정보전자공학전공(srlee@mokpo.ac.kr), ****전남대학교 컴퓨터정보학부(gslee@chonnam.ac.kr)

논문번호 : KICS2009-08-345, 접수일자 : 2009년 8월 10일, 2009년 11월 6일

1. 서 론

MPEG-2 표준은 1994년에 ISO 표준으로 승인된 이후 멀티미디어의 압축 기술로서 많은 분야인 디지털 위성방송, 디지털 유선방송(HDTV: High Digital Television)을 위한 오디오와 비디오 정보전송, 그리고 DVD 표준 등에서 활발하게 이용되어 왔다¹⁾. ITU-T Q.6/SG16과 MPEG 위원회(ISO/IEC/SG29/WG11)에서는 JVT(Joint Video Team)란 팀을 구성하고 공동으로 표준화를 진행하여 2003년 H.264/AVC 표준안을 발표하였다²⁾. H.264/AVC 표준은 고품질, 고압축을 목적으로 설계되었으며 낮은 비트율에서도 다른 영상 포맷들에 비해 우수한 화질을 보장하는 것이 특징이다. 현재 IPTV와 DMB, 인터넷 화상회의, triple-play service 등의 표준으로 지정되었으며, 각 분야에서 활발하게 이용되고 있다.

최근에는 MPEG-2 표준으로 부호화된 멀티미디어 콘텐츠를 H.264 표준에 맞게 사용하기 위하여 비디오 트랜스코더를 개발하고 있다^{3)~6)}. 하지만 시장 및 가정 내에서 사용하는 개인 비디오 레코더(PVR: personal video recorder) 및 HDTV는 MPEG-2 표준을 따르고 있으므로 H.264 표준을 제공하지 않는 상황이다. 따라서 기존 MPEG-2 표준을 지원했던 디바이스들은 H.264 표준으로 부호화된 비디오를 재생할 수 없으므로 디바이스를 새로 구입해야 하는 경우가 발생한다. 이 비용을 줄이기 위해서는 H.264 표준을 MPEG-2 표준으로 변환하는 트랜스코더가 필요하다^{7)~10), 11)}.

현재 트랜스코더의 연구 분야는 크게 두 가지로 나누어 진행되고 있다. 첫째, 동종간 트랜스코더로서 서버에 저장된 영상스트림을 비트율이나 화면의 크기 등을 조절하여 성능이 낮은 클라이언트로 전송할 때 사용되어지는 분야이다. 둘째, 이종간 트랜스코더로서 서버의 규격이 다른 디바이스로 영상스트림을 전송하기 위해 비디오 표준을 다른 표준으로 변환할 때 사용한다. 이종간 트랜스코더로는 직렬화소영역기반 트랜스코더(CPDT: Cascaded Pixel Domain Transcoder)와 주파수영역기반 트랜스코더(DDT: DCT Domain Transcoder)로 나누어진다. 직렬화소영역기반 트랜스코더는 복호기와 부호기가 직렬로 연결되어 있어 입력영상을 YUV 형태로 완전히 복호화 후 다시 부호화 하는 형태이다. 트랜스코더로서 가장 간단한 구조이면서 화질이 우수하다는 장점을 가지고 있다. 하지만 움직임 보상 연산을 두 번 수행함으로써 연산량을 증가하는 단점을 가지고

있다⁹⁾. 주파수영역기반 트랜스코더는 움직임 예측 연산이 DCT 영역에서 진행된다. 즉, 복호기의 IDCT 부터 DCT 까지의 과정이 생략되므로 직렬화소영역기반 트랜스코더보다 더 빠른 수행 성능을 보이지만 IDCT와DCT 과정을 수행하지 않으므로 참조프레임과의 오차가 발생하므로 오차가 누적되어 화질의 열화를 초래한다¹¹⁾.

또한 H.264 표준으로 복호화된 영상을 MPEG-2 표준으로 트랜스코딩할 때 기존 방법들은 인트라프레임과 인터프레임에 따라 방법들이 다르다. 첫째, 인트라프레임의 transcoding은 디코딩된 H.264 비트스트림을 MPEG-2 표준에 맞게 부호화하는 brute force(억지, 抑止) 방법이다. MPEG-2 표준은 H.264 표준의 Intra prediction 과정이 없으므로 매크로블록 모드 결정이 없고, H.264로 복호화된 비트스트림을 MPEG-2 표준에 맞게 매크로블록단위를 4개의 8x8 블록으로 나누어 2D 8x8DCT로 부호화한다. 둘째, 인터프레임의 transcoding은 H.264 표준으로 부호화된 가변블럭의 움직임벡터들을 MPEG-2 표준으로 부호화할 때 사용할 수 있는 최적의 움직임벡터를 resampling 하는 방법이다. 이때 H.264의 움직임벡터를 재사용하지 않고 MPEG-2 인코더에서 자체적으로 움직임추정을 하여 최적의 움직임벡터를 찾는 방법이 있다. 하지만 이 방법은 움직임추정 시간이 인코더의 60~80%를 차지하므로 과부하를 초래하므로 대부분 기존 방법들은 재사용 방법이 있다. 재사용 방법으로는 가변블럭의 움직임벡터들을 평균하여 움직임벡터를 예측하고 예측된 움직임벡터를 가지고 다시 최적의 움직임벡터를 정교화 한다. 정교화할때는 인코더의 계산시간을 최소화하기 위하여 ±2 정확소 검색 및 HAVS (horizontal and vertical search) 방법이 있다¹²⁾.

따라서 본 논문에는 화질의 열화를 줄이기 위해서 직렬화소영역기반 트랜스코더 방식을 사용하고, 직렬화소영역기반 트랜스코더 방식의 단점인 고 계산 수행속도를 개선하기 위하여 H.264 복호기에서 전송되는 매크로블록 모드와 움직임벡터의 특성을 파악하여 MPEG-2 인코더에서 움직임벡터를 재사용 하는 방법을 제안한다.

이 논문의 구성은 다음과 같다. 2장에서는 MPEG-2 및 H.264 표준의 특징과 차이점을 설명하고, 3장에서는 H.264 to MPEG-2의 트랜스코더 기법에 관한 기존 연구 중 가변블럭의 움직임벡터들을 MPEG-2 표준에서 사용하는 한 개의 움직임벡터를 추출하는 방법에 대하여 기술한다. 4장에서는

MPEG-2표준에서 움직임추정방법을 최소화 할 수 있는 최적의 움직임벡터를 추출하는 방법을 제안한다. 5장에서는 4장에서 제안한 방법의 타당성을 검증하기 위하여 기존방법들과 성능을 비교한다. 마지막으로 6장에서는 결론 및 향후연구를 제시한다.

II. MPEG-2 및 H.264 표준 특징 비교

H.264 표준의 목표는 비디오 압축을 보다 효율적인 방법으로 수행하여, 시장에 보급되고 있는 방송, 저장, 스트리밍 같은 분야를 지원하는 것이었다. 그래서 표 1에 나와 있는 H.264의 각 특징들을 살펴 보면, MPEG-2에 비해 공간적으로나 시간적으로 중복되는 부분들을 좀 더 효율적으로 제거하기 위한 방법들이 적용되었다.

또한 MPEG-2 표준에서는 오직 16x16 픽셀단위의 매크로블록만을 이용하였으나, H.264에서는 16x16부터 4x4까지 7가지의 블록 타입을 이용하였고, 화소 정밀도를 1/4픽셀까지 증가시킴으로서 보다 정밀한 예측이 가능하다.

마지막으로 MPEG-2 표준 에서는 한 개의 매크로 블록만을 이용하므로 오직 한 개의 움직임벡터

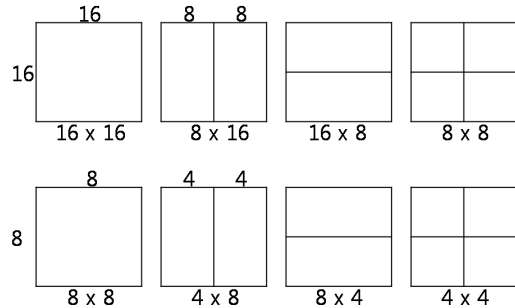


그림 1. H.264 매크로블록 파티션

만을 갖는다. 그러나 H.264에서는 1개에서 최대 16개의 움직임벡터가 존재할 수 있게 된다. 따라서 이 움직임벡터들을 MPEG-2에 맞게 한 개의 움직임벡터로 만들어야 한다. 따라서 본 논문에서는 여러 개로 구성되어 있는 H.264의 움직임벡터들 중에서 한 개를 선택하여 예측 움직임벡터로 결정한다. 그리고 예측된 움직임벡터를 중심으로 ± 2 반화소(half pel) 검색을 수행한다.

III. 기존연구방법

기존 H.264 디코더에서 MPEG-2 인코더로 트랜스코딩 구조도를 보면 그림 2와 같다^[13]. 트랜스코딩 구조도에서 보는 것과 같이 인코더에서 움직임추정이 60~80%를 차지하므로 움직임벡터 예측이 중요한 역할을 한다. 기존 동영상표준의 움직임벡터 예측방법으로는 시간적 예측(temporal prediction) 및 공간적 예측 등을 사용하여 인코더의 계산시간을 상당히 절약하였다^[14].

따라서 H.264 표준으로 복호화된 최소 1개에서 최대 16개까지 가지고 있는 움직임벡터들을 MPEG-2 인코더에서 사용할 수 있는 한 개의 움직임벡터를

표 1. MPEG-2와 H.264표준 비교

비교기준	MPEG-2	H.264
매크로블록 사이즈	16x16(frame mode), 16x8(field mode)	16x16
블록 사이즈	8x8	16x16,8x16,16x8,8x 8.4x8,8x4,4x4
Transform	8x8DCT	8x8,4x4,IntegerDCT 4x4, 2x2 Hadamard
화소 정밀도	1/2 pel	1/4 pel
양자화	상수비율로 증가하는 step size의 Scalar Quantization	12.5%비율로 증가하는 step size의 Scalar Quantization
영상 타입	I,P,B	I,P,B,SI,SP
전송율	2-15Mbps	64Kbps-150Mbps
엔트로피 코딩	VLC	CAVLC, CABAC
참조프레임	1개	1~5개
방향성 예측 모드	forward/backward	forward/backward, forward/forward, backward/backward
프로파일	5 profiles	7 profiles
인코더 복잡도	중간	높음
인트라 예측	없음	있음(Spatial Domain)
이전표준과의 호환성	있음	없음

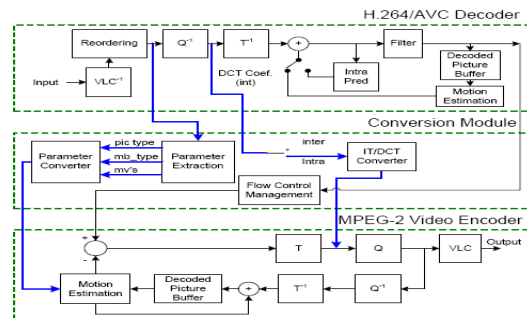


그림 2. 트랜스코딩 구조도

예측하고 예측된 움직임벡터를 가지고 재사용하는 가중치 평균법 및 동적인검색영역(dynamic search range) 방법을 소개한다^{9,15)}.

3.1 가중치 평균법¹⁵⁾

기존 평균법¹⁸⁾은 매크로블록 크기와 관계없이 매크로블록안에 존재하는 움직임벡터들을 평균하여 움직임벡터를 예측하는 방법으로 기존 MPEG-2 baseline보다 화질(PSNR)이 1~3dB을 초래하였다. 따라서 가중치 평균법은 가변블록의 크기에 따라 움직임벡터를 평균하는 방법과 다르게 H.264 디코딩과정에서 움직임벡터 추출, 움직임벡터 resampling, 움직임벡터 정교화, 움직임벡터 재사용으로 총 4단계로 구성되어있다.

3.1.1 H.264 디코딩과정에서 움직임벡터 추출

이 단계에서는 H.264 디코더에서 움직임벡터를 추출하여 매크로블록 크기 및 sub-macroblock size을 기반으로 하여 16x16매크로블록에 맞게 하나의 움직임벡터를 구하기 위하여 움직임벡터들을 총합한다.

3.1.2 움직임벡터 Resampling

가변블록 크기에 따라서 16x16매크로블록의 움직임벡터는 다양하게 아래 표와 같이 resampling 된다.

$H.264MV_x$ 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. $H.264MV_{ax}$ 와 $H.264MV_{bx}$ 는 수평으로 구성된 16x8 및 8x16블럭의 움직임벡터이고, $H.264MV_{ay}$ 와 $H.264MV_{by}$ 는 수직으로 구성된 16x8 및 8x16블럭의 움직임벡터이다. 16x8블럭에서 수평으로 구성된 움직임벡터만 평균한 이유는 수직(8)보

표 2. 움직임벡터 Resampling

가변블럭	식
16x16	$MPEG2MV_x = H.264MV_x$ $MPEG2MV_y = H.264MV_y$
16x8	$MPEG2MV_x = (H.264MV_{ax} + H.264MV_{bx})/2$ $MPEG2MV_y = (H.264MV_{ay} + H.264MV_{by})$
8x16	$MPEG2MV_x = (H.264MV_{ax} + H.264MV_{bx})/2$ $MPEG2MV_y = (H.264MV_{ay} + H.264MV_{by})/2$
8x8	$MPEG2MV_x = (H.264MV_{x1} + H.264MV_{x2} + H.264MV_{x3} + H.264MV_{x4})/4$ $MPEG2MV_y = (H.264MV_{y1} + H.264MV_{y2} + H.264MV_{y3} + H.264MV_{y4})/4$

다는 수평(16)블럭 크기가 크므로 움직임을 일정한 유지를 위해서 평균한다. 또한 8x8블럭인 경우 H.264의 움직임벡터 4개($H.264MV_{x1}, H.264MV_{x2}, H.264MV_{x3}, H.264MV_{x4}$)를 평균한 움직임벡터를 각각 움직임벡터로 예측한다. 그리고 매크로블록 타입이 3 또는 4(8x4,4x8,4x4)인 경우는 매크로블록안에 sub-macroblock이 존재하므로 앞 식에서 평균하였듯이 동일한 방법을 사용한다.

3.1.3 움직임벡터 정교화

앞 단계에서 예측된 움직임벡터를 가지고 움직임을 추정을 한다면 높은 계산 시간을 초래하므로 움직임벡터를 재사용하는데 있어서 참고문헌¹²⁾을 수정하여 사용한다. 즉 예측된 움직임벡터의 주변을 ±2하여 움직임벡터를 추정한다. 이때 작은 검색 범위를 추정하는게 아니라 그림 3과 같이 수평으로 검색하여 최소의 SAD값을 가진 위치에서 다시 수직 방향으로 최소의 SAD 값을 가진 움직임벡터를 최종적으로 결정한다.

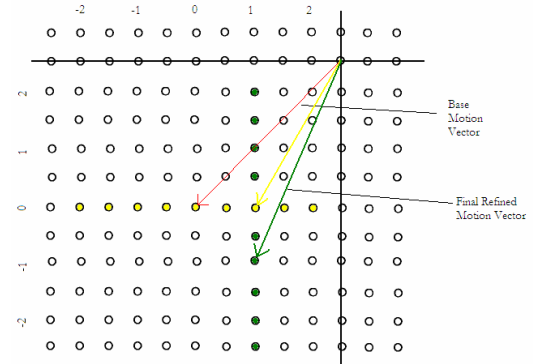


그림 3. 움직임벡터 정교화 방법

3.1.4 움직임벡터 재사용

최종적으로 결정된 움직임벡터를 MPEG-2 인코더에서 재사용한다.

3.2 동적인검색영역(dynamic search range)

이 방법은 MPEG-2 인코더에서 계산시간을 줄이기 위해서 H.264로 부호화된 MB mode(Skip, Inter, Intra)에 따라서 MPEG-2 인코더에 MB mode를 변경 없이 매핑한다. 또한 아래 식 (1)과 같이 H.264 디코더에 존재하는 움직임벡터 중에서 절대값이 가장 큰 값을 가진 움직임벡터를 예측된 움직임벡터를 사용하고 예측된 움직임벡터 만큼 동적인 검색 범위를 결정한다.

$$\begin{aligned}
 MPEG2MV_x &= MAX(ABS(H.264MV_{x1}, \\
 &\quad H.264MV_{x2}, \dots, H.264MV_{x16})) \\
 MPEG2MV_y &= MAX(ABS(H.264MV_{y1}, \\
 &\quad H.264MV_{y2}, \dots, H.264MV_{y16}))
 \end{aligned}
 \quad (1)$$

이 방법은 구현이 간단하나 동적인 검색 영역을 사용하므로 동작이 큰 영상인 경우, 움직임벡터가 커지므로 큰 영상에서는 수행속도가 저하된다는 단점이 있다.

IV. 제안한방법

기존연구방법에서 예측된 움직임벡터를 결정하는데 있어서 단지 H.264디코더의 움직임벡터들의 특성과 관계없이 평균법 및 가중치 평균법, 동적인검색영역 등을 사용하였으나 본 논문에서는 예측된 움직임벡터를 찾는데 있어서 H.264디코더에 존재하는 움직임벡터들의 상관관계를 이용하여 움직임벡터를 예측한다. 또한 본 논문에서는 H.264 디코더에서 MPEG-2 인코더로 전송되어지는 SKIP 모드와 INTRA 모드는 MPEG-2 인코더에서 그대로 사용하고, 단지 INTER 모드 안에 존재하는 가변블록의 움직임벡터들을 이용한다.

4.1 움직임벡터 비교

이 절에서는 H.264디코더에 존재하는 매크로블록 모드에 따른 움직임벡터들과 MPEG-2인코더에서 전역탐색(Full Search)할 때 움직임벡터, 그리고 기존방법인 가중치평균법을 적용할 때 움직임벡터를 비교한다. 다음 그림 4는 H.264로 복호화된 Football 영상의 일부분이다. 이때 매크로블록의 타입이 16x16 이거나 Skip된 매크로 블록을 제외한 나머지 블록



그림 4. H.264 매크로블록모드('Football' 영상)

들의 모드를 흰색으로 표시하였다.

또한 Football 영상에서 프레임 안에 존재하는 매크로블록의 움직임벡터들을 표 3에 제시하였다. 표 3과 같이 각 H.264 디코더의 움직임벡터는 대부분 MPEG-2인코더(baseline)의 전역탐색으로 찾은 움직임벡터들과 거의 비슷한 값을 가지고 있다. 또한 H.264 디코더의 매크로블록이 61, 62, 67번인 경우, 블록들이 서로 유사한 값을 가지고 있으므로 가중치평균법의 움직임벡터들과 거의 유사하다. 그러나 H.264 디코더의 매크로블록이 63, 73, 74번인 경우, 가중치평균법은 움직임벡터의 값을 비교하지 않고 단지 움직임벡터의 개수에 따라 평균을 계산하였기 때문에 전혀 다른 움직임벡터를 예측한 결과를 볼 수 있다. 가중치평균법과 같이 움직임벡터를 잘못 예측함으로써 인코더의 계산시간이 증가되고 또한 비트량이 증가됨으로써 화질의 열화가 발생한다. 따라서 본 논문에서는 H.264디코더의 다양한 가변블록의 움직임벡터들 중 최적의 움직임벡터를 선별하여 예측하는 방법을 제안한다.

모든 비디오인코더표준에서 움직임벡터를 추정할 때 탐색범위를 ±15 전역탐색을 수행함으로써 계산 시간 60~80%를 차지한다. 따라서 참고문헌 [12]에서 언급하였듯이 대부분 영상들의 움직임벡터들은 ±2이내에 존재하므로 본 논문에서도 표 4와 같이 실험하여 ±2 반화소 검색을 수행함으로써 인코더의

표 3. 움직임 벡터 비교

H.264디코더의 움직임벡터				가중치 평균법		MPEG-2 인코더(Full)	
번호	모드	X	Y	X	Y	X	Y
61	8x8	11	-2	11	-2	9	0
61	8x8	11	-2				
61	8x8	10	-2				
62	8x8	11	-1	11	-2	10	-2
62	8x8	11	-1				
62	8x8	10	-2				
62	8x8	10	-2				
63	16x8	0	0	5	-2	9	-1
63	16x8	10	-2				
67	16x8	0	0	0	0	0	0
67	16x8	0	0				
67	16x8	0	0				
73	8x8	0	0	3	-2	0	0
73	8x8	10	-8				
73	8x4	0	0				
73	8x4	0	0				
74	16x8	11	-8	6	-8	10	-8
74	16x8	0	0				

표 4. 움직임임베터의 일치율 비교(%)

비교방법	실험영상	BUS	SKI	COST-GUARD	FORE-MAN	STEFAN	FOOT-BALL
Inter MB		52,705	88,093	109,183	105,201	100,723	30,676
MV값 일치		75.37	76.04	81.94	78.11	76.20	85.60
MV값의 ±2 이내		93.33	90.23	96.41	93.85	87.03	97.99
평균값의 ±2 이내		86.57	89.28	98.44	93.45	84.81	94.29

계산 시간을 감소할 수 있다.

표 4는 움직임임베터의 일치율을 평가하기 위하여 H.264디코더의 가변블록의 움직임임베터들과 MPEG-2인코더(baseline)의 움직임임베터를 4가지(“Inter MB”, “MV값이 일치”, “MV값이 ±2이내” “평균값이 ±2이내”)로 나누어서 실험하였다. “Inter MB”는 앞에서 언급한 MPEG-2인코더와 H.264인코더가 차이를 가지고 있으므로 SKIP, INTRA모드를 제외한 INTER 모드인 경우에 해당하는 매크로블록 개수를 표현한 것이다. “MV값이 일치”는 H.264디코더의 가변블록에 존재하는 움직임임베터들 중 MPEG-2인코더의 움직임임베터와 일치하는 매크로블록의 비율이다. “MV값이 ±2 이내”는 H.264디코더의 가변블록에 존재하는 움직임임베터들이 MPEG-2인코더의 움직임임베터 ±2 이내에 존재하는 매크로블록의 비율이다. “평균값이 ±2이내”는 H.264디코더의 가변블록에 존재하는 움직임임베터들을 평균한 움직임임베터가 MPEG-2인코더의 움직임임베터 ±2 이내에 존재하는 매크로블록의 비율이다. 이때 실험 영상들의 특징을 보면 움직임이 일정한 BUS, FOREMAN, COSTGUARD 영상은 수평방향으로 일정하게 움직이므로 움직임임베터 값이 ±2이내에 90%이상 존재한다. 하지만 STEFAN 영상과 같이 움직임이 큰 영상은 움직임임베터 값이 ±2이내에 존재하지 않는다.

4.2 제안한 방법

그림 5에서 H.264 디코더의 매크로블록이 63, 74 번인 경우, 움직이고 있는 물체와 배경을 가리키는 블록들로 구분할 수 있다. 즉, 움직이고 있는 물체를 가리키는 블록은 배경과 구분되는 선명한 경계면을 가지고 있고 이 경계면을 가지고 있는 블록의 움직임임베터는 움직이고 있는 물체의 이동 값을 나타낸다. 그러므로 본 논문에서는 후보 블록들 중 경계면을 포함한 블록을 찾기 위해 아래와 같은 방법을 사용하였다.

Step 1: MBk(M,N)의 개수 결정

이 단계에서는 MBk(M,N)의 SKIP 및 INTRA

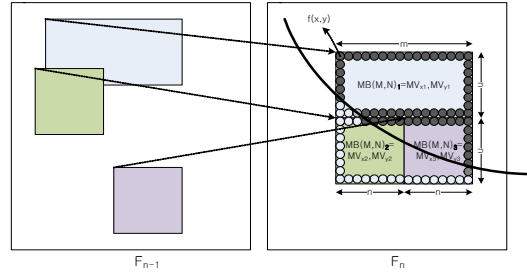


그림 5. 비용함수 및 예측움직임임베터 결정

모드를 제외한 INTER 모드 안에 존재하는 가변블록에 대한 움직임임베터 개수(k)를 결정한다. 단, M, N은 매크로블록의 위치이고, k는 가변블록의 번호이므로 최소 1개에서 최대 16개 존재한다.

Step 2: 비용함수 및 예측움직임임베터(MVp) 결정

MBk(M,N)에 존재하는 가변블록의 개수에 대하여 아래 (식 2)와 (식 3)과 같이 비용함수를 사용하여 왜곡을 추정한다.

$$\begin{aligned}
 dis_T(k) &= \sum_{i=0}^m |f(x,y) - f(x+1,y)| \\
 dis_B(k) &= \sum_{i=0}^m |f(x,y+n) - f(x+1,y+n)| \\
 dis_L(k) &= \sum_{i=0}^n |f(x,y) - f(x,y+1)|
 \end{aligned} \tag{2}$$

$$\begin{aligned}
 dis_R(k) &= \sum_{i=0}^n |f(x+m,y) - f(x+m,y+1)| \\
 cost_{fun}(k) &= (dis_T(k) + dis_B(k) + dis_L(k) + dis_R(k)) / (m+n)*k
 \end{aligned} \tag{3}$$

(식 2)에서 $dis_T(k), dis_B(k), dis_L(k), dis_R(k)$ 는 k번째 가변블록 내에 존재하는 경계의 위, 아래, 왼쪽, 오른쪽에 대한 왜곡 값이다. m, n은 가변블록의 가로 및 세로 길이이고, $f(x,y)$ 는 현재 프레임의 픽셀의 좌표이다. 그리고 비용함수인 $cost_{fun}(k)$ 는 왜곡 값의 총합을 평균한 것이다. 이때 블록의 전체 SAD 계산은 많은 계산시간을 요구하고 블록내부의 영상이 복잡할 때 SAD 영향을 미치지 않는다. 따라서 블록내의 경계를 선택하여 왜곡 값을 계산한다. 마지막으로 각 가변블록 중 비용함수 최대값을 가진 가변블록의 움직임임베터를 예측움직임임베터(MVp)로 한다. 비용함수 값이 최대값을 선택한 이유는 움직임이 블록의 경계에 걸쳐 있을 경우 픽셀간의 비연속성이 발생하므로 픽셀 차가 확연히 커지므로

블록의 경계에 영상이 있는 것으로 판단하여 최대 값을 선정한다.

Step 3: 최종 움직임벡터 결정

앞 단계에서 비용함수 값이 최대값을 가진 가변 블록의 움직임벡터를 MVp로 결정하고 ±2 변화소 검색을 수행하여 최종적으로 움직임벡터를 결정한다.

V. 실험 및 결과

CIF(352x288)와 SIF(352x240)크기의 영상을 가지고 각 300프레임씩 테스트하였다. 트랜스코더의 구성은 H.264 JM10.2 디코더와 MPEG Software Simulation Group의 MPEG-2 인코더를 사용하였다. 이때 H.264 디코더는 베이스 프로파일인 H.264 JM 10.2 인코더를 사용한 것을 디코딩한 것이다. 실험 영상들은 B프레임을 제외한 15 프레임의 GOP 사이즈로 1Mbits/sec의 비트율로 부호화하였다. 또한 양자화 파라미터는 24, 움직임벡터 검색범위는 ±15, 참조프레임은 1개, CABAC, RD optimization를 사용하였다. 그리고 MPEG-2 인코더는 H.264 부호화 환경과 동일하다.

다음 (표 5)은 H.264의 움직임벡터들 중 제안한 방법을 이용하여 선택한 움직임벡터와 가중치평균법, 그리고 MPEG-2 인코더의 '전역탐색'로 찾아낸 움직임벡터를 비교한 결과이다. 음영 처리한 매크로블록번호 14, 63, 73, 74, 81, 169는 본 논문에서 제안한 방법이 MPEG-2 인코더의 '전역탐색'에

표 5. MPEG-2의 전역탐색, 가중치평균법, 제안한방법의 움직임벡터 비교

방법 MB	MPEG-2인코더 (전역탐색)		가중치평균법		제안한 방법	
	MVx	MVy	MVx	MVy	MVx	MVy
14	10	0	3	0	10	-1
16	12	0	16	-1	15	0
61	9	0	11	-2	10	-2
62	10	-2	11	-2	11	-1
63	9	-1	5	-2	10	-2
73	0	0	3	-2	0	0
74	10	-8	6	-8	11	-8
81	-3	3	4	3	-3	3
129	5	-1	5	-1	5	-2
168	3	-1	2	-1	3	-1
169	0	1	4	-3	1	1

서 찾아낸 움직임벡터와 거의 비슷하였다.

다음 표 6 및 표 7은 MPEG-2 인코더의 '전역탐색', '가중치평균법', '동적인윈도우영역', 그리고 제안한 방법에 대한 PSNR 및 부호화 시간 비교이고, MPEG-2 인코더의 '전역탐색'와 비교한 결과이다. PSNR측면에서 '가중치평균법(W-AVERAGE)', '동적인윈도우영역(Dynamic)', '제안한방법' 은 'MPEG-2 인코더의 '전역탐색'에 비해 평균적으로 각각 1.7dB, 0.07dB, 0.08dB 정도 감소하였다. 그리고 부호화 시간 측면에서는 '전역탐색'에 비해 평균적으로 각각 66.4%, 64.4%, 64.9% 정도 감소하였다.

표 6. MPEG-2 인코더, 가중치평균법, 동적인윈도우영역, 제안한 방법의 PSNR비교

영상 비교방법	MPEG-2인코더 (전역탐색:A)	가중치평균법 (B-A)		동적인 윈도우 영역(C-A)		제안한 방법 (D-A)	
FOREMAN	36.62	35.05	-1.57	36.51	-0.11	36.48	-0.14
STEFAN	30.35	27.47	-2.88	30.22	-0.13	30.10	-0.25
COASTGUARD	34.06	32.26	-1.8	34.01	-0.05	34.00	-0.06
M&D	43.86	43.42	-0.44	43.87	0.01	43.85	-0.01
CONTAINER	38.06	38.06	-1.8	38.07	-0.05	38.06	-0.06

표 7. MPEG-2 인코더, 가중치평균법, 동적인 윈도우영역, 제안한 방법의 부호화 시간 비교

영상 비교방법	MPEG-2인코더 (전역탐색: A)	가중치평균법 (B-A)		동적인 윈도우 영역(C-A)		제안한 방법 (D-A)	
FOREMAN	81.57	28.89	64.58%	35.23	56.81%	27.97	65.71%
STEFAN	152.37	33.79	77.82%	33.94	77.73%	30.28	80.13%
COASTGUARD	81.13	28.39	65.01%	27.15	66.54%	34.37	57.64%
M&D	63.12	24.29	61.52%	25.86	59.03%	25.70	59.28%
CONTAINER	67.82	24.94	63.23%	28.89	57.40%	25.80	61.96%

기존 방법 중 성능이 우수한 ‘동적인원도우영역’은 MPEG-2 인코더의 ‘전역탐색’과 거의 비슷하나, 움직임이 큰 영상 즉, 움직임벡터가 큰 값을 가진 매크로블록은 탐색 범위가 커지므로 부호화 계산량이 증가한다. 움직임이 거의 없는 ‘CONTAINER’ 영상은 그림 6과 같이 부호화 시간이 거의 일정하고, 그림 7과 같이 PSNR도 거의 비슷하다. 그러나 움직임이 큰 영상인 ‘FOREMAN’ 및 ‘STEFAN’ 영상은 그림 9와 같이 PSNR 성능은 유사하지만, 그림 8의 부호화계산량을 비교해보면 ‘동적인원도우영역’

은 프레임 181번~221번에서 계산량이 일정하지 못하지만 ‘제한한 방법’은 움직임에 상관없이 일정하게 부호화를 수행한다.

본 논문에서 제안한 방법과 ‘동적인원도우영역’과 비교했을 때, PSNR은 거의 차이가 없으나 부호화 시간은 약 5%정도 감소되었다. 그리고 ‘가중치평균법’과 비교했을 때 부호화시간은 제안한 방법과 거의 비슷하나 PSNR는 영상의 움직임에 따라 1~3dB 감소하였다. 따라서 제안한 방법이 부호화 시간 대비 PSNR는 우수한 성능을 보였다.

VI. 결론

H.264 to MPEG-2 트랜스코딩은 새로운 타입의 서비스를 제공하면서 MPEG-2 인프라구조를 활용할 수 있는 계기가 될 것이다. 하지만 H.264 표준과 MPEG-2 표준이 서로 상이하므로 H.264 to MPEG-2 트랜스코딩이 필요하다. 본 논문에서는 트랜스코딩의 효율을 위하여 MPEG-2 인코더에서 계산량을 가장 많이 차지하는 움직임추정을 효율적으로 진행하기 위하여 제안한다. 실험결과, 제안한 알고리즘은 기존 방법에 비하여 PSNR은 거의 비슷하지만 움직임추정은 약 65% 감소하여 우수한 성능을 보였다. 향후연구는 가변블럭안에 존재하는 다양한 움직임벡터의 방향성을 고려하여 최적의 움직임벡터를 예측하고 예측움직임벡터를 이용하여 MPEG-2 인코더에서 저 계산할 수 있는 알고리즘 개발 연구이다.

참고 문헌

- [1] ISO/IEC 13818, Information Technology(MPEG-2): Generic Coding of Moving Pictures and Associated Audio Information, 1995
- [2] Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG (ISO/IEC JTC1/SC29 /WG11 and ITU-T SG16 Q.6), November 2002.
- [3] Kalva H. and Petljanski B, “Exploiting the directional features in MPEG-2 for H.264 intra transcoding”, IEEE Transactions on Consumer Electronics, Volume 52, Issue 2, p.706-711, 2006
- [4] Li Wang, Qi Wang, Yu Liu and Wei Lu, “A fast Intra Mode Decision Algorithm for MPEG-2 to H.264 Video Transcoding”, IEEE

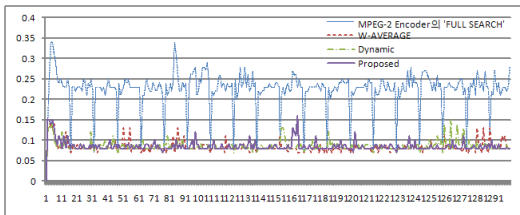


그림 6. Container영상의 부호화시간 비교

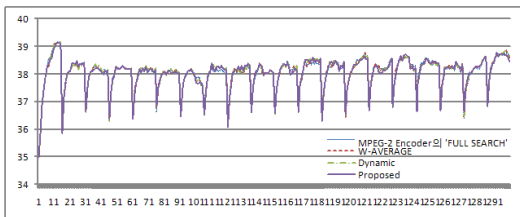


그림 7. Container영상의 PSNR비교

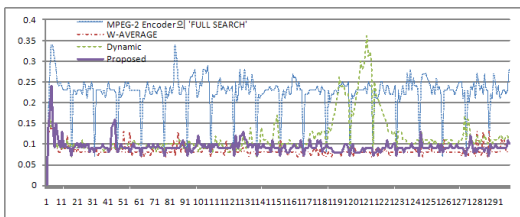


그림 8. FOREMAN 영상의 부호화시간 비교

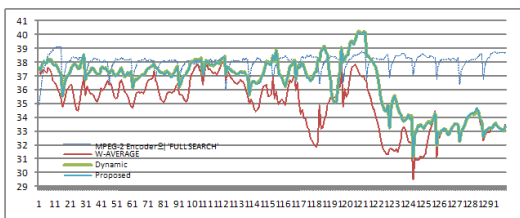


그림 9. FOREMAN영상의 PSNR 비교

10th International Symposium on Consumer Electronics (ISCE) p. 1-5, 2006

[5] Gao Chen, Shouxun Lin and Yongdong Zhang, "A Fast Coefficients Conversion Method for the Transform Domain MPEG-2 to H.264 Transcoding", International Conference on Digital Telecommunications (ICDT) p.17-17, 2006

[6] Xiaolan Lu, Tourapis, A.M., Peng Yin and Boyce, J., "Fast mode decision and motion estimation for H.264 with a focus on MPEG-2/H.264 transcoding", IEEE International Symposium on Circuits and Systems (ISCAS), p.1246-1249 Vol. 2, 2005

[7] Kalva, H.: Issues in H.264/MPEG-2 video transcoding. Computer Science and Engineering, 2004

[8] Yang, L., Song, X., Hou, C., Dai, J.: H.264 MPEG-2 transcoding based on personal video recorder platform. In: Proc. of the Ninth International Symposium on Consumer Electronics, P.438-440, 2005

[9] Kunzelmann, P., Kalva, H.: Reduced complexity H.264 to MPEG-2 transcoder. ICCE International Conference on Consumer Electronics, 2007

[10] Moiron, S., Faria, S., Assuncao, P., Silva, V., Navarro, A.: Fast interframe transcoding from H.264/AVC to MPEG-2. ICIP, 2007

[11] 신익경, "비디오 트랜스코더에서 움직임벡터를 이용한 효율적 매크로블록 모드 결정 기법 연구", 석사학위논문, 2003

[12] J. Youn and Ming-Ting Sun, "Motion Vector Refinement for high-performance transcoding", in IEEE Int. Conf. Consumer Electronics, Los Angeles, C.A., Vol. 1, Issue 1, pp 30-40, March 1999.

[13] Moiron. S, et. al, "H.264/AVC to MPEG-2 Video Transcoding Architecture", Proc. conf. on telecommunications, Vol.1, pp.449-452, 2007

[14] Ishfaq Ahmad; Wei Xiaohui; Yu Sun; Ya-Qin Zhang, "Video transcoding: an overview of various techniques and research issues", IEEE Transaction on Multimedia, vol.7, p. 793-803, 2005.

[15] S. Sharmat, "Transcoding of H.264 BITSTREAM to MPEG-2 BITSTREAM", Master's thesis, 2007

[16] Moiron, S., Faria, S., Assuncao, P., Silva, V., Navarro, "Mode Conversion for H.264 to MPEG-2 Video Transcoder", Proc Picture Coding Symposium, 2007

손 남 레 (Nam-rye Son)

정회원



1996년 2월 호남대학교 컴퓨터 공학과 이학사
2000년 2월 전남대학교 전산학과 이학석사
2005년 2월 전남대학교 전산학과 이학박사
2007년 3월~2009년 2월 호남대학교 인터넷소프트웨어학과 전임강사

2009년 11월~현재 목포대학교 중점연구소 연구교수
<관심분야> 비디오 코딩, IPTV, e-learning, 방송통신융합기술, 임베디드시스템

정 민 아 (Min-A Jung)

정회원



1992년 2월 전남대학교 전산통계학과 졸업
1994년 2월 전남대학교 전산통계학과 석사
2002년 2월 전남대학교 전산통계학과 박사
2005년 3월~현재 목포대학교 컴퓨터교육과 조교수

<관심분야> 데이터베이스/데이터마이닝, 생체인식시스템, 무선통신응용분야(RFID, USN, 텔레메틱스), 임베디드시스템

이 성 로 (Seong-ro Lee)

정회원



1987년 2월 고려대학교 전자
공학과 졸업
1990년 2월 한국과학기술원 전
기및전자공학과 석사
1996년 8월 한국과학기술원 전
기및전자공학과 박사
2005년 3월~현재 목포대학교 정
보공학부 정보공학전공 부교수

<관심분야> 디지털통신시스템, 이동 및 위성통신시
스템, USN/텔레미틱스응용분야, 임베디드시스템,
생체인식시스템

이 귀 상 (Guee-sang Lee)

정회원



1980년 2월 서울대학교 전기공
학과 학사
1982년 2월 서울대학교 전자계
산기공학과 석사
1983년~1984년 금성통신연구
소 연구원
1984년~현재 전남대학교 컴퓨
터정보학부 교수

<관심분야> 멀티미디어통신, 영상처리 및 컴퓨터비
전, 임베디드 시스템