

움직임벡터의 거리를 고려한 H.264 to MPEG-2 Transcoding

정희원 손남례*, 정민아**, 이성로***

H.264 to MPEG-2 Transcoding considering Distance of Motion Vectors

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

요약

H.264표준으로 압축된 동영상들은 IPTV, DMB, 디지털콘텐츠서비스 등 다양한 응용분야에 사용되고 있다. 기존 MPEG-2표준으로 압축된 동영상들은 H.264표준을 따르는 디바이스에 적합한 트랜스코딩 방법들이 연구되어왔다. 반대로 H.264표준으로 압축된 동영상은 MPEG-2표준 디바이스를 사용하고 있는 사용자들은 시청할 수 없게 되었다. 따라서 본 논문에서는 MPEG-2표준 디바이스를 사용하고 있는 사용자들은 H.264표준을 이용할 수 있는 H.264 to MPEG-2 트랜스코딩을 제안한다. 제안한 알고리즘은 H.264부호기의 특징인 가변블록의 움직임벡터를 MPEG-2의 매크로블록에 적합한 한 개의 움직임벡터를 예측하여 MPEG-2부호기의 계산량을 향상시키는 방법을 제안한다. 이때 H.264표준의 매크로블록안에 존재하는 다양한 움직임벡터들 간의 거리에 따라 MPEG-2부호기에서 최적의 움직임벡터를 예측한다. 실험결과, 제안한 방법의 PSNR은 MPEG-2 부호기의 “전역탐색”과 거의 비슷하면서, 부호화 계산 시간은 평균적으로 약 68% 감소하였다.

Key Words : H.264/AVC, MPEG-2, transcoding, 움직임추정, 움직임벡터

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 68% averagely, with a small objective quality reduction in PSNR.

※ 이 논문은 2009년도 정부(교육과학기술부)의 재원으로 한국연구재단의 대학중점연구소지원사업으로 수행된 연구임(2009-0093828)

※ 이 논문은 2008년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (No. D00280)

※ 본 연구는 2008년도 목포대학교 학술연구비에 의하여 연구 되었음

* 목포대학교 정보산업연구소(nrson72@gmail.com), **목포대학교 컴퓨터공학과 (majung@mokpo.ac.kr),

*** 목포대학교 정보전자공학전공 (srlee@mokpo.ac.kr)

논문번호 : KICS2009-09-397, 접수일자 : 2009년 9월 10일, 최종논문접수일자 : 2010년 4월 28일

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 고화질, 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 있다³⁻⁶⁾. 하지만 시장 및 가정 내에서 사용하는 개인 비디오 레코더(PVR: personal video recorder) 및 HDTV는 MPEG-2 표준을 따르고 있으므로 H.264 표준은 제공하지 않는 상황이다. 따라서 기존 MPEG-2 표준을 지원했던 디바이스들은 H.264 표준으로 부호화된 비디오를 재생할 수 없으므로 디바이스를 새로 구입해야 하는 경우가 발생한다. 이러한 구입 비용을 줄이기 위해서 H.264 표준을 MPEG-2 표준으로 변환하는 트랜스코더가 필요하다^{7-10,16)}.

현재 트랜스코더의 연구 분야는 크게 두 가지로 나누어 진행되고 있다. 첫째, 동종간 트랜스코더로서 서버에 저장된 영상 스트림을 비트율이나 화면의 크기 등을 조절하여 성능이 낮은 클라이언트로 전송할 때 사용되어지는 분야이다. 둘째, 이종간 트랜스코더로서 서버의 규격이 다른 디바이스로 영상 스트림을 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 부호화하는 형태이다. 트랜스코더로서 가장 간단한 구조이면서 화질이 우수하다는 장점을 가지고 있다. 하지만 움직임 보상 연산을 두 번 수행함으로써 연산량을 증가하는 단점을 가지고 있다⁹⁾. 주파수영역기반 트랜스코더

는 움직임 예측 연산이 DCT 영역에서 진행된다. 즉, 복호기의 IDCT 부터 DCT 까지의 과정이 생략되므로 직렬화소영역기반 트랜스코더보다 더 빠른 수행 성능을 보이지만 IDCT와 DCT 과정을 수행하지 않으므로 참조프레임과의 오차가 발생하여 오차 누적으로 화질의 열화를 초래한다¹¹⁾.

또한 H.264 표준으로 부호화된 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 부호화된 비트스트림을 MPEG-2 표준에 맞게 매크로블록단위를 4개의 8x8 블록으로 나누어 2D 8x8DCT로 부호화한다. 둘째, 인터프레임의 transcoding은 H.264 표준으로 부호화된 가변블록의 움직임벡터들을 MPEG-2 표준으로 부호화할 때 사용할 수 있는 최적의 움직임벡터를 재사용(re-use) 하는 방법이다. 이때 H.264의 움직임벡터를 재사용하지 않고 MPEG-2 부호기에서 자체적으로 움직임추정을 하여 최적의 움직임벡터를 찾는 방법이 있다. 하지만 이 방법은 움직임추정 시간이 부호기의 60~80%를 차지하므로 과부하를 초래하므로 대부분 기존 방법들은 재사용 방법이 있다. 재사용 방법으로는 가변블록의 움직임벡터들을 평균하여 움직임벡터를 예측하고 예측된 움직임벡터를 가지고 다시 최적의 움직임벡터를 정교화 한다. 정교화(refinement)할 때는 부호기의 계산시간을 최소화하기 위하여 ± 2 정화소 검색(integer pixel search) 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 움직임벡터들 간의 거리 특성을 파악하여 MPEG-2 부호기에서 움직임벡터를 재사용하는 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 H.264 복호기의 가변블록내에 존재하는 움직임벡터들을 MPEG-2 부호기에서 사용할 수 있는 한 개의 움직임벡터를 추출하는 방법에 대하여 기술한다. 4장에서는 H.264 표준의 매크로블록 안에

존재하는 다수개의 움직임벡터들 간의 거리를 고려하여 최적의 움직임벡터를 재사용하는 방법을 제안한다. 5장에서는 4장에서 제안한 방법의 타당성을 검증하기 위하여 기존방법들과 성능을 비교한다. 마지막으로 6장에서는 결론 및 향후연구를 제시하고 맺음한다.

II. MPEG-2 및 H.264표준 특징 비교

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

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

마지막으로 MPEG-2 부호기에서는 한 개의 매크로블록만을 이용하므로 오직 한 개의 움직임벡터만을 갖는다. 그러나 H.264에서는 1개에서 최대 16개의 움직임벡터가 존재하므로 MPEG-2 부호기에 맞게 최적

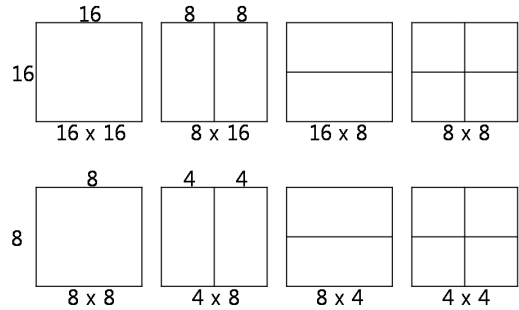


그림 1. H.264 매크로블록 파티션

의 움직임벡터를 결정해야 한다. 따라서 본 논문에서는 매크로블록 안에 존재하는 움직임벡터들의 거리를 고려하여 MPEG-2 부호기에서 최적의 예측 움직임벡터(PMV: Predictive Motion Vector)로 결정한다. 그리고 예측된 움직임벡터를 중심으로 ± 2 window size로 반화소(half pixel) 검색을 수행한다.

III. 기존연구방법

기존 H.264 복호기에서 MPEG-2 부호기로 트랜스코딩 구조도를 보면 그림 2와 같다^[3]. 트랜스코딩 구조도에서 MPEG-2 부호기의 움직임추정이 60~80%를 차지하므로 움직임벡터 예측이 중요하다.

현재 H.264 to MPEG-2 transcoding에 관련된 기존 연구는 움직임벡터 간의 상관성을 고려하지 않는 동적인 검색영역(dynamic search range)^[9] 및 가중치 평균법^[15] 방법과 움직임벡터의 기울기를 고려한 방법^[17] 그리고 방향성을 고려한 방법^[18]을 소개한다.

표 1. MPEG-2와 H.264표준 비교

비교기준	MPEG-2	H.264
매크로블록 사이즈	16x16(frame mode), 16x8(filed mode)	16x16
블록 사이즈	8x8	16x16, 8x16, 16x8, 8x8, 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
인트라 예측	없음	있음(Spatial Domain)
이전표준과의 호환성	있음	없음

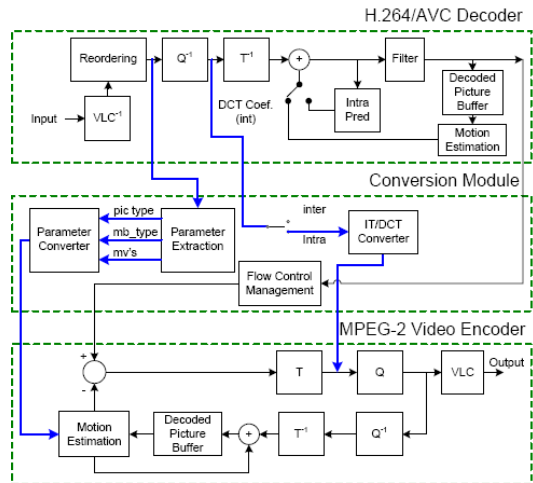


그림 2. 트랜스코딩 구조도

3.1 동적인 검색영역^[9]

이 방법은 MPEG-2 부호기에서 계산시간을 줄이기 위해서 H.264로 부호화된 MB mode(Skip, Inter, Intra)에 따라서 MPEG-2 부호기에 MB mode를 변경 없이 매핑한다. 또한 아래 식 (1)과 같이 H.264복호기에 존재하는 움직임벡터 중에서 절대값이 가장 큰 값을 가진 움직임벡터를 예측된 움직임벡터로 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, 움직임 벡터가 커지므로 큰 영상에서는 수행속도가 저하된다는 단점이 있다.

3.2 가중치 평균법^[15]

기존 평균법^[8]은 매크로블록 사이즈와 관계없이 매크로블록안에 존재하는 움직임벡터들을 평균하여 움직임벡터를 예측하는 방법으로 기존 MPEG-2 baseline보다 화질(PSNR)이 1~3dB을 초래하였다. 따라서 가중치 평균법은 가변블럭의 사이즈에 따라 움직임벡터를 평균하는 방법을 달리하는 방법으로 H.264 디코딩과정에서 움직임벡터 추출, 움직임벡터 resampling, 움직임벡터 정교화, 움직임벡터 재사용으로 총 4단계로 구성되어있다.

3.2.1 H.264 디코딩과정에서 움직임벡터 추출

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

3.2.2 움직임벡터 Resampling

가변블럭사이즈에 따라서 16x16매크로블럭의 움직임벡터는 다양하게 아래 표 2와 같이 resampling 된다.

$H.264MV_x$ 와 $H.264MV_y$ 는 H.264 복호기에서 사용하는 16x16매크로블럭의 x와 y의 움직임벡터이고, $MPEG2MV_x$ 와 $MPEG2MV_y$ 는 MPEG-2 부호기에서 16x16매크로블럭단위로 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 8x16블럭

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

의 움직임벡터이다. 16x8블럭에서 수평으로 구성된 움직임벡터만 평균한 이유는 수직(8)보다는 수평(16)블럭 사이즈가 크므로 움직임을 일정하게 유지하기 위해서 평균한다. 또한 8x8블럭인 경우 H.264의 움직임벡터 4개($H.264MV_{x1}, H.264MV_{x2}, H.264MV_{x3}, H.264MV_{x4}$)를 합하여 평균한 움직임벡터를 각각 움직임벡터로 예측한다. 그리고 매크로블록 타입이 3 또는 4(8x4,4x8,4x4)인 경우는 매크로블록안에 sub-macroblock이 존재하므로 16x8,8x16,8x8블럭을 계산하는 식과 같은 방법을 사용한다.

3.2.3 움직임벡터 정교화

앞 단계에서 예측된 움직임 벡터를 가지고 움직임 추정을 한다면 높은 계산 시간을 초래하므로 움직임 벡터를 재사용하는데 있어서 참고문헌[12]을 수정하여 사용한다. 즉, 최종 움직임벡터는(C)는 그림 3과 같이 예측된 움직임벡터(A) 위치 중심에서 수평으로 최소의 SAD값을 가진 위치(B)에서 다시 수직방향으로 최소의 SAD 값을 가진 움직임벡터를 최종적으로 결정한다.

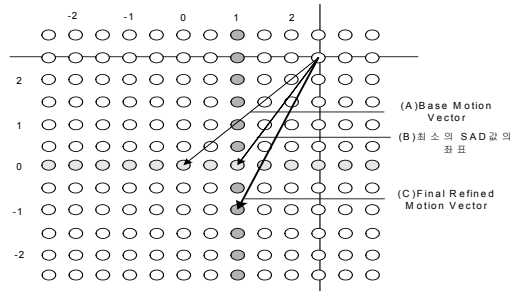


그림 3. 움직임벡터 정교화 방법

3.2.4 움직임벡터 재사용

가중치평균법은 최종적으로 결정된 움직임벡터를 MPEG-2 부호기에서 재사용한다.

3.3 움직임벡터의 기울기를 고려한 방법^[17]

움직임벡터의 기울기를 고려한 방법은 기존 어려운 닉방법^[14] 중 가장 많이 사용하는 방법이다. 이 방법은 H.264 표준의 매크로블록안에 존재하는 움직임벡터들을 그림 4와 같이 기울기에 따라 8개의 방향으로 그룹핑 한다^[17]. 그리고 그룹핑을 토대로 그룹안에 존재하는 움직임벡터들을 평균하여 MPEG-2 부호기에서 사용되어질 후보 예측움직임벡터를 선정한다. 그리고 후보 예측움직임벡터들의 위치에서 각각 BMA (boundary matching algorithm)을 수행하여 왜곡 값이 최소인 후보예측움직임벡터를 최종적인 예측움직임벡터로 결정한다.

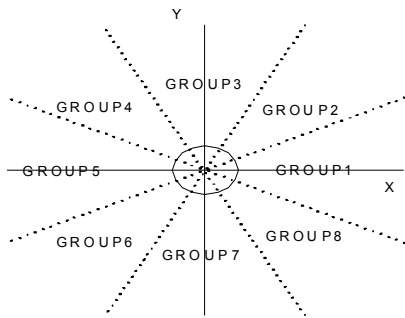


그림 4. 움직임벡터의 기울기에 따라 8가지 그룹 생성

3.4 움직임벡터의 방향을 고려한 방법^[18]

참고문헌^[18]은 기존 동적인검색영역방법^[9]과 평균법^[15]을 확장한 내용으로 식 2와 같이 H.264의 움직임벡터들 중 다른 값들과 가장 차이가 나지 않는 움직임벡터를 찾아내는 방법이다.

$$MV = \min \|MV_{selected} - MV_n\|, n = 1..N \quad (2)$$

식 2에서 결정된 움직임 벡터를 중심으로 식 3~식5와 같이 움직임벡터들의 기울기 편차를 이용해서 식 6~식 7과 같이 x축과 y축 탐색영역을 결정하는 동적인 재정교화 윈도우(dynamic refinement window)를 수행한다.

$$\theta = \arctan2(MV_y, MV_x) \quad (3)$$

$$\sigma_{\theta}^2 = \frac{\sum_{n=1}^N (\theta_n - \theta_{selected})^2}{N} \quad (4)$$

$$P = \sqrt{MV_x^2 + MV_y^2} \quad (5)$$

$$X_{range} = \|\cos(\theta - \sigma_{\theta}^2) \cdot P\| - MV_x \quad (6)$$

$$Y_{range} = \|\sin(\theta - \sigma_{\theta}^2) \cdot P\| - MV_y \quad (7)$$

IV. 제안한 방법

기존관련연구는 H.264복호기에서 전송되어지는 움직임벡터들의 특성을 고려하지 않고 단지 블록의 사이즈에 따라 평균^[15]이나 움직임벡터가 크면 움직임벡터가 커질거라는 예측만으로 검색윈도우 사이즈를 확장하였다^[9]. 그리고 움직임벡터의 기울기^[17] 및 방향^[18]을 고려한 방법은 움직임벡터의 특성을 고려하였지만 ‘참고문헌^[17]’은 그림 5(a)의 G4를 보면 움직임 벡터의 기울기는 같지만 거리가 떨어져 있어서 같은 그룹으로 보기가 힘들다. 또한 G1과 G2 결과에서 정해진 기울기 각도에 따라 그룹을 결정하므로 바로 인접한 움직임 벡터가 존재하여도 다른 그룹으로 결정한다는 단점이 있다. ‘참고문헌^[18]’은 ‘동적인검색영역방법’과 같이 움직임이 커지면 계산량이 증가하는 단점을 가지고 있다.

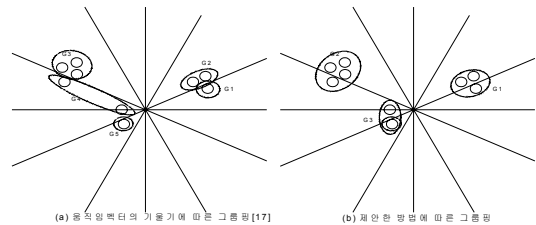


그림 5. 참고문헌^[17]과 제안한 그룹핑 방법 비교

따라서 본 논문에서는 그림 5(b)와 같이 움직임 벡터들을 정해진 기울기의 각도로 그룹화 하지 않고 움직임 벡터들 간의 거리에 따라 그룹핑하는 방법을 제안하고 알고리즘은 다음과 같다. 첫째, H.264복호기로 전송되어지는 매크로블록 안에 존재하는 움직임벡터들을 검출한다. 둘째, 검출된 움직임 벡터들의 거리를 계산하기 위하여 클러스터링의 대표적인 평균연결알고리즘(average linkage algorithm)을 수정하여 그룹핑 집합을 생성한다. 셋째, 그룹핑 집합들 중에서 움직임벡터 수를 최대로 가지고 있는 그룹핑 집합을 선택하고, 선택된 그룹핑 집합내에서 중앙값을 취하여 예측움직임벡터로 결정한다. 마지막으로 예측된 움직임벡터 중심으로 ±2 반화소 검색을 실시한다.

4.1 움직임벡터 검출

H.264 복호기에서 MPEG-2 부호기로 전송되어지는 SKIP 모드, INTRA 모드는 MPEG-2 부호기에서 그대로 사용하고, 단지 INTER 모드 안에 존재하는 가변블록의 움직임벡터들을 이용한다. 이때 식(8)과 같이 가변블록의 움직임벡터는 모드별로 최소 1개에서 최대 16개까지 움직임벡터를 검출할 수 있다.

$$MVset = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}, n \leq 16 \quad (8)$$

그림 6은 본 논문에서 제안한 아이디어로 매크로블록이 전경 또는 배경에 속한 경우에는 가변블록의 움직임벡터가 거의 일치하지만 매크로블록이 전경 및 배경에 걸친 경우에는 그림(6)의 'A' 부분을 확대한바와 같이 가변블록 안의 움직임벡터는 다양한 방향을 가지고 있다.

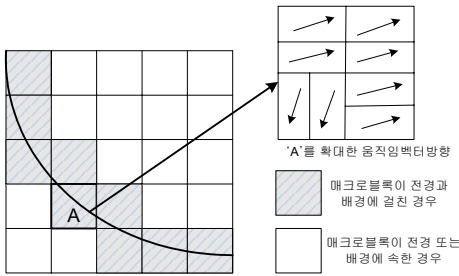


그림 6. 제안한 아이디어

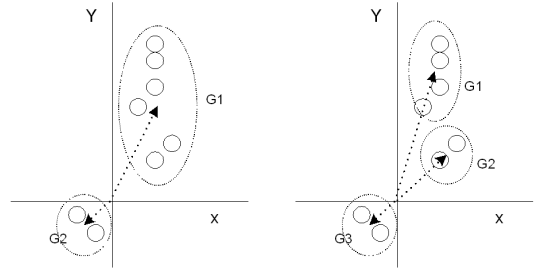
4.2 그룹핑 집합 생성

앞 단계에서 움직임 벡터를 검출했다면 식(9)와 같이 움직임벡터들 간의 거리를 구한다. 식(9)에서 움직임 벡터는 (x_n, y_n) 와 (x_{n-1}, y_{n-1}) 이고, D 는 움직임 벡터간의 거리이다.

$$D = \sqrt{|x_{n-1} - x_n|^2 + |y_{n-1} - y_n|^2}, (n \geq 2) \quad (9)$$

식 9를 이용하여 움직임 벡터간의 거리를 구하고 최소가 되는 두 블록간의 평균값을 계산하는 과정을 반복함으로써 평균연결알고리즘을 수행한다. 그림 7은 그림 6의 'A' 매크로블록의 다양한 방향을 가진 움직임벡터들을 평균연결알고리즘을 사용하여 그룹핑한 결과이다.

그림 7(a) 결과에서 보듯이 그룹핑 집합은 G1, G2 개 생성되었다. 즉, 평균연결알고리즘은 최종적으로



(a) 평균연결알고리즘을 적용한 결과 (b) 임계값을 적용한 결과(제한 방법)
그림 7. 기존 방법 및 제안 방법 비교

두 개의 그룹을 생성하므로 그룹된 움직임 벡터들은 거리가 떨어져 있어도 같은 그룹을 생성하는 단점이 있다. 따라서 본 논문에서는 그림 7(b)와 같이 움직임 벡터간의 거리가 임계값(=5) 이상일 경우 같은 그룹(G1)으로 하지 않고 다른 그룹(G2)을 생성하여 총 3개이다.

4.3 예측움직임벡터 결정

이 단계에서는 그룹핑 집합들 중에서 움직임벡터 수를 최대를 가지고 있는 그룹핑 집합(G1)을 선택하고 선택된 그룹핑 집합내에서 중앙값을 취하여 예측 움직임벡터로 결정한다. 마지막으로 예측된 움직임벡터 중심으로 ± 2 범위내에서 반화소 검색을 실시한다.

V. 실험 및 결과

실험환경은 Pentium IV CPU 3.06GHz의 PC에서 VC++ 2005를 사용하였고, 테스트 영상은 CIF (352x288)와 SIF(352x240)크기의 영상을 가지고 각 300프레임씩 테스트하였다. 트랜스코더의 구성은 H.264 JM10.2 복호기와 MPEG Software Simulation Group의 MPEG-2 부호기를 사용하였다. 이때 H.264 복호기는 베이스 프로파일인 H.264 JM 10.2 부호기를 사용한 것을 디코딩한 것이다. H.264 부호기는 B 프레임을 제외한 15 프레임의 GOP(IPPP)사이즈, 양자화 파라미터는 24, 움직임벡터 검색범위는 ± 15 , 참조프레임은 1개, CAVLC, RDO(rate-distortion optimization) 옵션을 사용하여 1Mbps/sec 비트율(CBR: constant bit rate)로 부호화하였다. 그리고 MPEG-2 부호기는 H.264 부호화 환경과 동일하다.

그림 8은 'Foreman'영상을 1Mbps/sec로 트랜스코딩 후, 부호화된 비트율 성능평가이다. 이 결과에서 보여지듯이 H.264부호기는 비트율제어(JVT-E069)¹⁾를 제공하기 때문에 1Mbps/sec로 옵션을 초기화하면

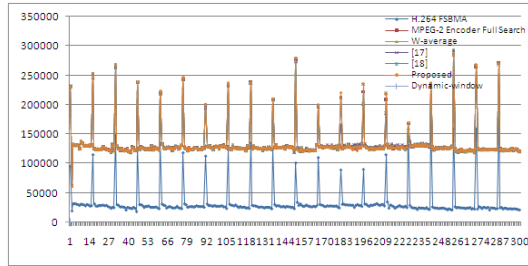


그림 8. 'Foreman' 영상의 부호화된 비트량 비교(1Mbits/sec)

자체적으로 비트율제어가 수행된다. 또한 MPEG-2 부호기에서는 TM5를 제공하기 때문에 1Mbits/sec 부호화하면서 비트율을 제어하였다. 결론적으로 H.264 부호기의 성능은 가장 우수지만, 제안한 방법이나 기존 방법의 부호화된 비트량은 거의 유사하였다.

다음 표 3 및 표 4는 MPEG-2 부호기의 '전역탐색', '가중치평균법(W-average)', '동적인원도우영역(Dynamic-window)', '참고문헌[17]과 [18]', 그리고 '제한한 방법(Proposed)'에 대한 PSNR 및 트랜스코딩 계산시간 비교이다.

PSNR 측면에서 '가중치평균법', '동적인원도우영역', '참고문헌[17]과 [18]', '제한한방법'는 'MPEG-2 부호기의 전역탐색'에 비해 평균적으로 각각 1.34dB, 0.06dB, 1.33dB, 0.14dB, 0.18dB 정도 감소하였다. '가중치평균법'과 '참고문헌[17]'의 PSNR은 최소 0.01dB에서 최대 2.88dB까지 화질이 떨어지지

만, '동적인원도우영역', '참고문헌[18]'과 '제한한방법'의 PSNR은 MPEG-2 부호기의 전역탐색과 거의 유사하였다.

트랜스코딩에 소요되는 계산시간은 '가중치평균법', '동적인원도우영역', '참고문헌[17]과 [18]', '제한한 방법'는 'MPEG-2 부호기의 전역탐색'에 비해 평균적으로 각각 68.5%, 66.1%, 70.7%, 54.1%, 68% 정도 감소하였다. 앞 절에서 언급하였듯이 부호화 시간 중 움직임추정이 최대 60~80%를 차지하므로 움직임추정 시간을 최대로 감소하는 방법이 중요하다. 하지만 실험결과에서 보여지듯이 움직임이 큰 영상과 작은 영상에 따라 다소 차이점을 보이고 있다. 즉, 그림 9~그림 10과 같이 움직임이 작은 영상인 'CONTAINER' 영상은 기존방법이나 제안한 방법의 트랜스코딩 계산시간이나 PSNR은 다소 차이가 없다.

하지만 'FOREMAN', 'STEFAN', 'COASTGUARD' 영상과 같이 움직임이 큰 영상은 탐색 범위가 커지므로 트랜스코딩 계산시간이 증가한다. 그림 11~그림 12과 같이 'FOREMAN' 영상의 '동적인원도우영역'과 '참고문헌[18]'은 트랜스코딩 계산시간은 증가하지만 PSNR 성능은 우수하였다. 반대로 '가중치평균법'과 '참고문헌[17]'은 트랜스코딩 계산시간은 감소하지만, PSNR 성능은 확연히 우수하지 않았다. 즉, 참고문헌[17]의 PSNR 성능 저하 원인은 매크로블록 안에 존재하는 움직임벡터들을 일정한 기울기 간격으로 나누어 그룹핑하였기 때문이다. 하지만 본 논문

표 3. MPEG-2 인코더,가중치평균법,동적인원도우영역,[17],[18],제한한 방법의 PSNR비교(dB)

영상	비교방법	MPEG-2인코더 (전역탐색)	가중치 평균법	동적인 원도우영역	[17]	[18]	제한한 방법
FOREMAN		36.62	35.05	36.51	35.05	36.35	36.29
STEFAN		30.35	27.47	30.22	27.47	30.09	29.98
COASTGUARD		34.06	32.26	34.01	32.27	33.94	33.91
M&D		43.86	43.42	43.87	43.42	43.83	43.82
CONTAINER		38.06	38.06	38.07	38.07	38.05	38.06
평균		36.59	35.25	36.53	35.29	36.45	36.41

표 4. MPEG-2 인코더,가중치평균법,동적인원도우영역,[17],[18],제한한 방법의 트랜스코딩 계산시간 비교(sec)

영상	비교방법	MPEG-2인코더 (전역탐색)	가중치 평균법	동적인 원도우영역	[17]	[18]	제한한 방법
FOREMAN		81.57	28.89	35.23	25.98	29.98	29.55
STEFAN		152.37	33.79	33.94	27.86	48.43	29.84
COASTGUARD		81.13	28.39	27.15	27.27	76.29	30.33
M&D		63.12	24.29	25.86	24.48	25.31	27.98
CONTAINER		67.82	24.94	28.89	25.2	24.85	25.09
평균		89.20	28.06	30.21	26.15	40.97	28.55

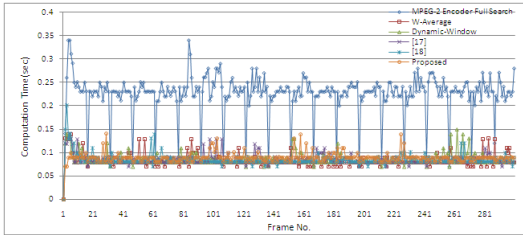


그림 9. Container 영상의 트랜스코딩 시간 비교

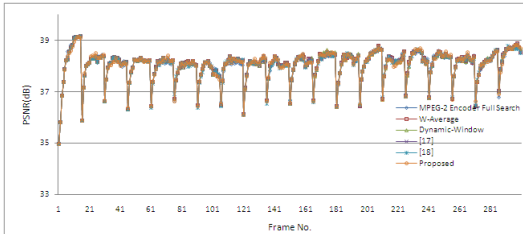


그림 10. Container 영상의 PSNR 비교

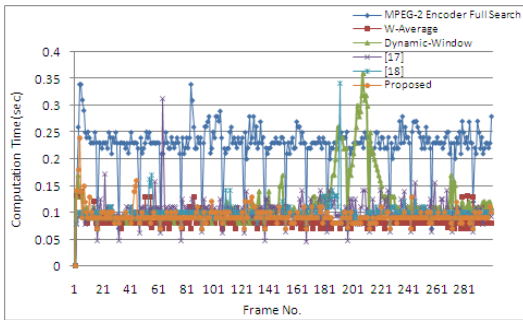


그림 11. Foreman 영상의 트랜스코딩 시간 비교

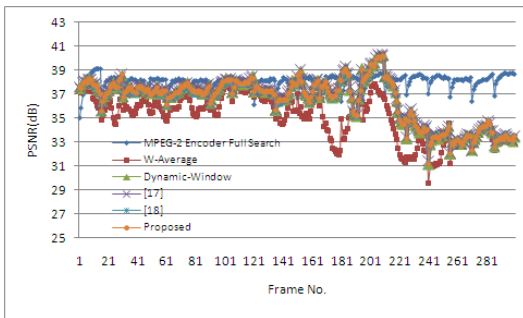


그림 12. Foreman 영상의 PSNR 비교

서 제안한 방법은 트랜스코딩 계산시간과 관계없이 PSNR 성능도 우수하였다. 즉, 제안한 방법은 움직임 벡터의 거리를 고려하면서 일정한 거리를 초과하면

새롭게 그룹핑을 하므로 움직임벡터를 효율적으로 예측할 수 있었다.

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 부호기에서 계산량을 가장 많이 차지하는 움직임추정을 효율적으로 진행하기 위하여 제안한다. 이때 가변블럭안에 존재하는 다양한 움직임벡터의 방향성을 고려하여 최적의 움직임벡터를 예측하고 예측된 움직임벡터를 이용하여 MPEG-2 부호기에서 계산할 수 있는 알고리즘 개발이다.

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, Vol.52, Issue 2, pp.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 10th International Symposium on Consumer Electronics (ISCE) pp.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) pp.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), pp.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, pp.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”, 석사학위 논문, 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] S. Ghanbari and M. Z. Bober, “A cluster based method for the recovery of the lost motion vectors in video coding,” IEEE Mobile and wireless communications network, 2002.

[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

[17] 김기홍, 손남례, 신운정, 이귀상, “H.264 to MPEG-2 Transcoder에서 P-frame을 위한 움직임

벡터 비교 연구”, 한국컴퓨터정보학회학계학술대회, July, 2009.

[18] Hari Kalva and Phil Kunzelmann, “Dynamic Motion Estimation for Transcoding P frames in H.264 to MPEG-2 Transcoders”, IEEE Trans. on Consumer Electronics, Vol.54, No.2, May, 2008

[19] Siwei Ma, Wen Gao, P. Gao, and Y. Lu, “Rate control for advanced video coding standard,” IEEE International Symposium on Circuits and Systems, Vol.2, pp.892-895, May 2003.

손 남 례 (Nam-rye Son)

정회원



1996년 2월 호남대학교 컴퓨터공학과 이학사

2000년 2월 전남대학교대학원 전산학과 이학석사

2005년 2월 전남대학교대학원 전산학과 이학박사

2007년 3월~2009년 2월 호남대학교 인터넷소프트웨어학과 전임강사

2009년 11월~현재 목포대학교 정보산업연구소 연구교수

<관심분야> 비디오 코딩, IPTV, e-learning, 방송통신융합기술, 임베디드시스템

정 민 아 (Min-A Jung)

정회원



1992년 2월 전남대학교 전산통계학과

1994년 2월 전남대학교 전산통계학과 석사

2002년 2월 전남대학교 전산통계학과 박사

2005년 3월~현재 목포대학교 컴퓨터공학과 조교수

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

이 성 로 (Seong-ro Lee)

정회원



1987년 2월 고려대학교 전자
공학과

1990년 2월 한국과학기술원 전
기 및 전자공학과 석사

1996년 8월 한국과학기술원 전
기 및 전자공학과 박사

2005년 3월~현재 목포대학교

정보공학부 정보전자공학 교수

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