

논문 2011-48SP-1-11

선박-육지간 통신을 위한 실시간 H.264 to MPEG-2 트랜스코딩

(A Real-time H.264 to MPEG-2 Transcoding for Ship to Shore
Communication)

손 남 례*, 정 민 아**, 이 성 로***

(Nam Rye Son, Min A Jeong, and Seong Ro Lee)

요 약

현재 위성방송을 통해 신호를 전송 또는 재전송하는 무선통신서비스를 이용하는 시청자 층은 다양하다. 하지만 방송국은 H.264표준으로 압축된 동영상들을 위성통신을 통하여 전송하기 때문에 H.264표준 디바이스를 갖추지 않는 선박은 실시간으로 데이터를 전송받지 못하는 단점을 가지고 있다. 따라서 본 논문에서는 기존 MPEG-2 표준 디바이스를 사용하고 있는 선박을 위하여 H.264 to MPEG-2 트랜스코딩 방법을 제안한다. 제안한 방법은 H.264 표준의 매크로블록모드의 특성을 분석하여 H.264 to MPEG-2 transcoding의 계산시간 및 화질을 개선한다. 첫째, H.264와 MPEG-2표준의 INTRA 모드 방법이 상이하므로 새로운 방법을 제안한다. 둘째, 매크로블록모드가 INTER 모드인 경우에는 H.264표준의 가변블록 안에 존재하는 움직임벡터의 방향성을 고려하여 새로운 예측움직임벡터(PMV: predictor motion vector)를 제안한다. 이때 최종움직임벡터는 예측움직임벡터를 그대로 사용하거나, H.264표준의 매크로블록내에 존재하는 가변블록들의 움직임벡터들과 MPEG-2부호기(baseline)의 움직임벡터의 일치율을 비교하여 최종적으로 움직임벡터를 예측할 범위(window size)를 결정한다. 실험결과, 제안한 트랜스코딩방법의 PSNR은 MPEG-2 FSBMA와 거의 일치하고, 트랜스코딩에 필요한 계산시간은 평균적으로 각각 70% 또는 67% 감소하였다.

Abstract

Recently, the grade of users using wireless communication services which transmits and re-transmits to the signal via the broadcasting satellite have a variety. However the ships not preparing of H.264 standard devices should not received the realtime data because the broadcasting stations have transmitted the compressed video data through the satellite communication. Therefore this paper proposes H.264 to MPEG-2 transcoding for the ships using MPEG-2 devices. Proposed method improves a speed and object quality in H.264 to MPEG-2 transcoding by analysis features of macroblock modes in H.264. In the Intra mode of P-frame, it proposes new method by computing coincidence proportion after comparing of Intra mode methods of H.264 and MPEG-2. In the Inter mode, it proposes a PMV(predictive motion vector) considering movement of motion vectors in H.264 decoder. we reuses a PMV directly as like the final MV in MPEG-2 encoder and refinements the MV after coincidence ratio comparing of variable motion vectors of H.264 and these of MPEG-2. The experimental results from proposed method show a considerable reduction in processing time, as much as 70% and 67% respectively, with a small objective quality reduction in PSNR.

Keywords: transcoding, motion estimation, H.264/AVC, MPEG-2, refinement

* 정희원, 목포대학교 정보산업연구소

(Institute of Information Science and Engineering Research, Mokpo National University)

** 정희원-교신저자, 목포대학교 컴퓨터공학과

(Dept. of Computer Engineering, Mokpo National University)

*** 정희원, 목포대학교 전자정보공학과

(Dept. of Information & Electronics, Mokpo National University)

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

※ 본 과제는 교육과학기술부의 재원으로 한국연구재단의 지원을 받아 수행된 광역경제권 선도산업 인재양성사업의 연구결과입니다.

접수일자: 2010년4월12일, 수정완료일: 2010년8월27일

I. 서 론

국제해사위성기구(INMARSAT: INternational MA-Ritime SATellite organization)는 해상통신을 개선하기 위하여 필요한 위성통신시설을 제공하고, 이를 통하여 해상에서의 조난, 인명의 안전에 필요한 통신, 선박과 육지간, 선박과 선박간 통신의 질을 높이고자 1979년에 설립된 국제기구이다^[1]. 국제해사위성기구는 원래 선박통신과 선박이나 조난당했을 때 비상통신을 위해 설립된 기구로서 이동물체를 서비스 대상으로 삼기 때문에 가입자가 제한되어 있다. 그러나 1985년 10월 제4차 총회에서 항공통신을, 1989년 1월 제8차 총회에서는 육상이동통신 및 휴대용 단말기기를 이용대상에 각각 포함시킴으로써 새로운 서비스의 개발, 단말기의 소형화·경량화 및 통신제공 범위 확대 등을 적극 추진하고 있다. 하지만 현재 선박과 육지간의 서비스는 위성통신을 통하여 간단한 영상이나 문자를 전송하기 때문에 선박은 육지에서 전송중인 다양한 콘텐츠를 제공받지 못한 상태이다. 따라서 육지에서 유통되고 있는 다양한 비디오 콘텐츠를 위성통신을 통하여 선박에서도 서비스 되는 방법이 필요하다^[2].

현재 비디오 콘텐츠를 압축하는 방법은 저장매체 및 통신방법에 따라 국제전기통신연합 (ITU-T)에 정한 H.26x계열과 ISO 산하에 있는 MPEG(Moving Picture Experts Group)계열이 있다. MPEG계열 중 MPEG-2는 오디오와 비디오 인코딩(부호화)에 관한 일련의 표준을 말하며, ISO 표준 13818(13818-1은 시스템, 13818-2는 비디오 부호화, 13818-3은 오디오)로 공표되었다^[3]. MPEG-2는 일반적으로 디지털 위성방송, 디지털 유선 방송 등의 디지털 방송을 위한 오디오와 비디오 정보 전송을 위해 쓰이고 있다. H.264/AVC는 매우 높은 데이터 압축률을 가지는 디지털 비디오 코덱 표준으로 MPEG-4 파트 10 또는 AVC(Advanced Video Coding)라 부르기도 한다. 이 표준은 2003년 5월에 ITU-T의 VCEG(Video Coding Experts Group)과 ISO/IEC의 MPEG이 공동으로 JVT(Joint Video Team)을 구성하고 표준화를 진행한 결과물로 나온 것이다^[4]. H.264/AVC는 기존의 표준(MPEG-2, H.263, MPEG-4 Part 2)과 압축 성능을 비교하면, 저-비트율에서는 45%~65%와 고-비트율에서는 25~45% 정도 감소한다^[5]. 현재 H.264/AVC는 IPTV, DMB, 디지털 케이블, DVT/TiVO, 디지털 콘텐츠 서비스 등 다양한 응용분야

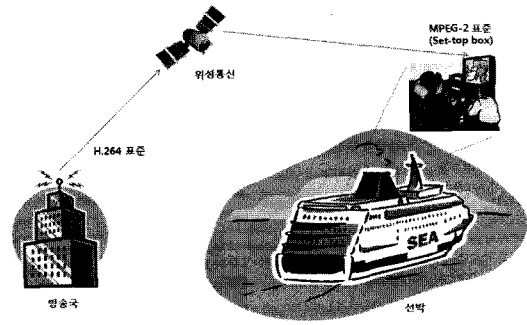


그림 1. 선박과 육지간 통신
Fig. 1. Communication between ship and shore.

에 사용되고 있다. 비록 H.264표준이 뛰어난 성능을 가지고 있지만 기존 MPEG-2 표준을 사용하고 있는 선박 내의 디바이스 시장은 전 세계적으로 유통되고 있다. 따라서 그림(1)과 같이 다양한 콘텐츠를 제공하는 방송국 및 콘텐츠 제공자는 위성통신을 통하여 H.264표준으로 압축한 동영상상을 실시간으로 선박에 전송한다면 H.264표준 디바이스를 사용하는 선박은 무리없이 비디오 콘텐츠를 전송받을 수 있지만, MPEG-2표준 디바이스를 사용 중인 대부분의 선박은 원활한 서비스를 제공받기 어렵기 때문에 H.264 to MPEG-2 트랜스코딩 방법이 필요하다.

현재 트랜스코딩 방법은 크게 두 종류로 나눌 수 있다. 첫째는 동종 간 트랜스코딩으로 서버에 저장된 영상 스트림을 비트율이나 화면의 크기 등을 조절하여 성능이 낮은 클라이언트로 전송할 때 쓰인다. 둘째, 이종 간 트랜스코딩으로 서버에서 규격이 다른 디바이스로 영상 스트림을 전송하기 위해 비디오 표준을 다른 표준으로 변환할 때 사용한다. 이때 이종 간 트랜스코딩은 다시 두 가지 방법으로 나눌 수 있다. 첫째, 그림(2)과 같이 주파수영역기반 트랜스코더(DDT: DCT Domain Transcoder)방식은 움직임 추정 연산이 DCT영역에서 수행된다. 즉, 복호기의 IDCT부분부터 부호기의 DCT 부분까지의 과정을 생략했기 때문에 CPDT방식에 비해

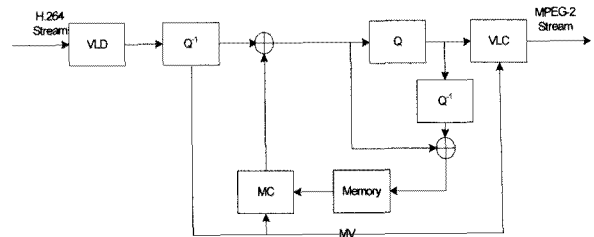


그림 2. DCT 도메인 트랜스코딩
Fig. 2. DCT Domain transcoding.

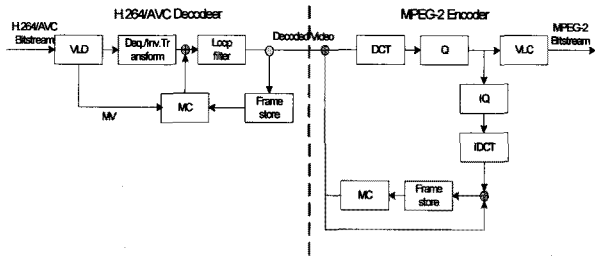


그림 3. 직렬화소영역기반 트랜스코딩
Fig. 3. Cascaded Pixel Domain Transcoding.

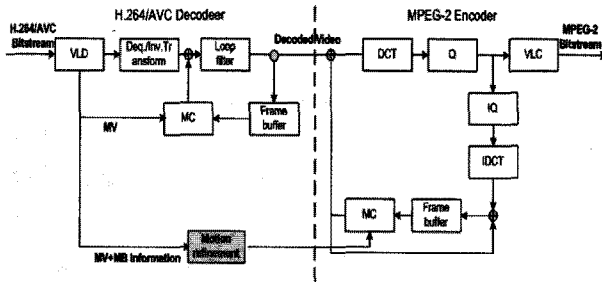


그림 4. 제안한 방법의 구조도
Fig. 4. Architecture of proposed method.

더 빠른 계산시간이 보인다. 그러나 IDCT와 DCT과정을 거치지 않으므로 참조프레임과의 오차가 발생하고, 이 오차가 누적되면서 화질의 열화가 발생된다^[6~7].

둘째, 직렬화소영역기반트랜스코더(CPDT: Cascaded Pixel Domain Transcoder)는 그림(3)와 같이 복호기와 부호기가 직렬로 연결되어 있으며 입력영상을 YUV형태로 완전히 복호한 뒤 다시 부호화하는 형태이다. 이 방법은 가장 간단한 구조이면서 트랜스코딩 중 가장 화질이 좋다는 장점이 있다. 그러나 움직임 보상 연산을 두 번 수행하기 때문에 연산량이 증가하는 단점이 있다^[8~10].

따라서 본 논문에서는 화질의 열화를 감소하기 위하여 CPDT방식을 사용하고, CPDT의 연산시간을 효율적으로 감소하기 위하여 그림(4)과 같이 H.264복호기의 H.264표준의 움직임벡터 및 매크로블록모드의 특성을 분석하여 화질을 방지하면서 H.264 to MPEG-2 transcoding의 계산시간을 개선한다. 첫째, H.264와 MPEG-2표준의 INTRA 모드 방법이 상이하므로 새로운 방법을 제안한다. 둘째, 매크로블록모드가 INTER 모드인 경우에는 H.264표준의 가변블록 안에 존재하는 움직임벡터의 방향성을 고려하여 새로운 예측움직임벡터(PMV: Predictor Motion Vector)를 제안한다. 그리고 H.264표준의 매크로블록내에 존재하는 가변블록들의 움직임벡터들과 MPEG-2부호기의 움직임벡터의 일치

율을 비교하여 최종적인 움직임벡터를 예측할 범위(window size)를 결정한다.

이 논문의 구성은 다음과 같다. II장에서는 MPEG-2 및 H.264 표준의 특징과 차이점을 설명하고, III장에서는 H.264 to MPEG-2 transcoding에 관한 기존 연구 중 움직임벡터 정교화 방법을 기술한다. IV장에서는 H.264표준의 매크로블록모드 특성을 분석하여 최적의 움직임벡터 정교화 방법을 제안한다. V장에서는 IV장에서 제안한 방법의 타당성을 검증하기 위하여 기존방법들과 성능을 비교한다. 마지막으로 VI장에서는 결론 및 향후 연구를 제시하고 맺음한다.

II. MPEG-2 및 H.264 표준 비교

H.264표준의 목표는 비디오 압축을 보다 효율적인 방법으로 수행하여, 콘텐츠 시장에 널리 보급되고 있는 방송, 저장, 스트리밍 같은 다양한 콘텐츠 서비스를 지원하는 것이다. H.264표준은 MPEG-2표준에 비해 공간적 및 시간적으로 중복되는 부분들을 좀 더 효율적으로 제거하기 위한 방법들을 사용하고 있다. 즉, 공간영역에서 수행되는 인트라 예측과정, 루프필터, 4x4정수변환, 다양한 크기의 화면간 예측 모드(inter prediction mode), 그리고 화면내 예측 모드(intra prediction mode)는 MPEG-2표준과 비교되는 H.264표준의 부호화 특징들이다^[11]. 또한 H.264표준은 그림(5)과 같이 매크로블록을 8가지로 나누어서 움직임벡터를 추정하므로 한 가지 모드(16x16)를 가지고 있는 MPEG-2표준에 비해 상대적으로 부호기의 계산 복잡도가 높다.

특히, H.264표준에서 화면간, 화면내 예측 모드를 위한 매크로블록 모드 선택은 많은 계산 과정이 필요하다^[12]. 즉, 화면내 예측 모드에서 휘도(luminance) 블록들에 대한 예측 부호화는 4x4 서브 블록이나 16x16 매크

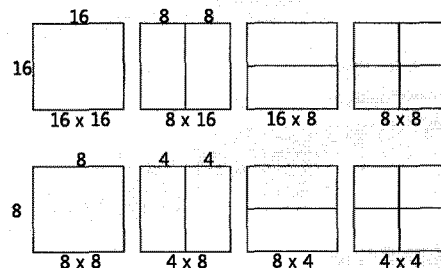


그림 5. H.264 매크로블록의 서브블록
Fig. 5. Sub-blocks of H.264 macroblock.

로블록 단위로 수행되고, 방향성에 따라 각각 9가지, 4가지의 예측모드가 존재한다. 또한 4:2:0 색차 포맷의 영상에 대해 하나의 매크로블록은 두 개의 8×8 크기의 색차(chrominance)블록이 존재하는데 이에 대해서도 4가지 방향성에 따른 예측 부호화를 수행한다. 단, 색차블록에 대해서는 두 블록에 대해 같은 예측 모드를 적용한다. 화면간 예측 모드에서는 휘도성분에 대해서 영상의 국부특성을 효율적으로 반영하기 위하여 16×16부터 4×4까지의 다양한 블록 크기의 움직임 추정 및 보상을 지원한다. 각 매크로블록(16×16화소)의 휘도성분은 16×16, 16×8, 8×16, 8×8블록의 4가지 형태로 나누어진다. 8×8블록모드가 선택되면, 매크로블록 내의 4개의 8×8블록을 8×8, 8×4, 4×8, 4×4의 4가지 형태로 다시 나눈다. P 프레임 내의 매크로블록 모드들 중에서 인터예측을 위한 모드인 SKIP, 16×16, 16×8, 8×16, 8×8 모드를 각각 SKIP, Mode1, Mode2, Mode3, Mode8로 정의하고, Mode8 내의 선택 가능한 부-매크로블록 모드인 8×8, 8×4, 4×8, 4×4 모드를 각각 Mode4, Mode5, Mode6, Mode7 이라고 정의한다. H.264표준은 이러한 다양한 예측 모드 중 화질과 비트율 측면에서 가장 적합한 모드를 선택하기 위해서 RDO(Rate Distortion Optimization)기법을 사용한다.

III. 기존 움직임 추정 방법

1. MPEG-2 및 H.264 표준 부호기의 FSBMA비교

MPEG-2표준 부호기의 움직임 추정 기법은 그림(6)과 같이 블록정합알고리즘(BMA: Block Matching Algorithm)이 주로 사용된다. BMA는 이전 프레임으로부터 블록 단위로 움직임 추정을 하는 기법으로, 현재 프레임을 16x16 크기 등의 블록으로 나누어 현재 프레

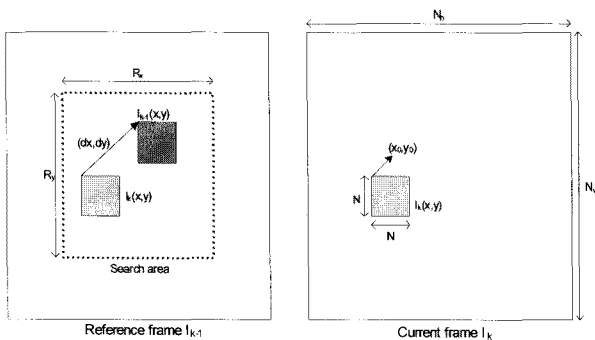


그림 6. MPEG-2 표준의 FSBMA
Fig. 6. FSBMA of MPEG-2.

임의 블록을 그에 해당하는 이전 프레임의 탐색 영역 내의 블록들과 비교하여 가장 유사한 블록을 찾아내는 것이다. 이때, 탐색시 사용하는 정합 기준은 여러 가지가 있으나 식(1)과 같이 SAD(Sum Absolute Difference)을 주로 사용하고, 검색영역범위가 x축으로 $3 \times N_x$ 와 y축으로 $3 \times N_y$ 인 경우에 전 영역 블록 경계 알고리즘(FSBMA: Full Search BMA)이라 한다. 이때 N 은 블록사이즈이다.

$$SAD(dx, dy) = \sum_{m=x}^{x+N_x-1} \sum_{n=y}^{y+N_y-1} |I_k(m, n) - I_{k-1}(m+dx, n+dy)|$$

$$\vec{MV} = (MV_x, MV_y) = \min_{(dx, dy) \in R^2} SAD(dx, dy) \quad (1)$$

이 식(1)에서 $I_k(x, y)$ 및 $I_{k-1}(x, y)$ 는 각각 k 번째 및 $k-1$ 번째 프레임에 존재하는 (x, y) 의 픽셀좌표, R_x 및 R_y 는 각각 x축 및 y축의 검색영역, $I_{k-1}(m+dx, n+dy)$ 는 이전프레임의 탐색 영역 내의 후보 블록이며, (dx, dy) 은 변위 벡터이다. $\vec{MV} = (MV_x, MV_y)$ 는 움직임 벡터로 이전프레임의 탐색 영역 내에서 현재 블록과의 SAD값이 최소인 위치를 가진 좌표이다. 이때 MPEG-2 표준 부호기는 움직임 추정시 1/2화소 위치까지 움직임을 추정한다. 그리고 H.264 표준 부호기는 가변블록이 존재하므로 블록사이즈만 달리하여 MPEG-2표준에서 사용하는 블록정합방법과 같다. 단, 움직임 추정시 그림(7)과 같이 결정된 정화소를 중심으로 8개의 1/2화소 위치에서 움직임을 추정하고 이후 결정된 위치를 중심으로 8개의 1/4화소 위치에서 움직임을 추정한다.

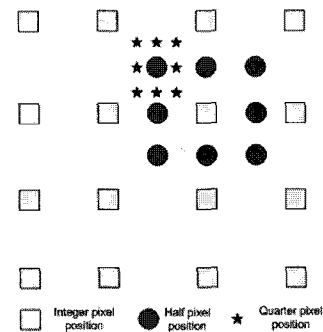


그림 7. H.264표준 부호기의 1/4화소 추정 방법
Fig. 7. Quarter-pel motion estimation of H.264.

2. H.264 에서 MPEG-2 transcoding의 움직임벡터 비교

현재 H.264 to MPEG-2 transcoding에 관한 연구^[7]

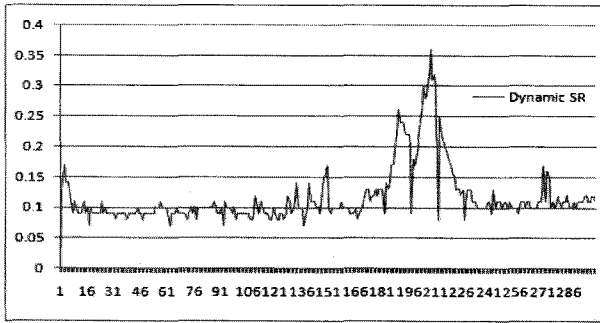


그림 8. Foreman영상에 대한 [16] 결과
Fig. 8. Results of computation time for Foreman sequence[16].

는 움직임 추정 및 보상, 인트라 픽처 예측, 적응적인 필드 및 프레임 코딩, 트랜스폼 코딩, 슬라이스 스트럭처, 더블러킹 필터, NAL(Network Adaptation Layer) 등 트랜스코딩의 계산 복잡도를 줄이기 위한 방법이 제시되고 있다. 특히, 최근에는 H.264에서 MPEG-2로의 트랜스코딩 방법 중 계산시간이 60~80%를 차지하고 있는 움직임 추정 방법이 연구되고 있다. 기존의 H.264에서 MPEG-2로의 트랜스코딩에서 제안하고 있는 움직임 추정 방법의 문제점은 다음과 같다. [14]와 [15]는 H.264 복호기에서 전송되어지는 움직임벡터들의 특성을 고려하지 않고 단지 블록 사이즈에 따라 평균^[14] 및 가중치 평균^[15]을 계산하여 MPEG-2 부호기에서 움직임벡터를 재사용한다. 이 방법은 MPEG-2 FSBMA의 계산시간에 비하여 68%감소하나, 화질은 2~5dB정도 감소하여 화질을 저하시키는 단점이 있다.

또한 [16]은 H.264표준의 가변블록에 존재하는 다양한 움직임벡터 중 가장 큰 움직임 벡터를 검색윈도우사이즈로 재결정하는 방법으로 움직임벡터가 크면 움직임벡터가 커질거라는 예측만으로 검색윈도우사이즈를 확장하였다. 이 방법은 그림(8)과 같이 움직임이 복잡한 영상인 경우 계산시간이 증가하는 단점을 가지고 있다.

마지막으로 [17]은 움직임벡터 방향을 고려하여 예측 움직임벡터를 결정하고, 움직임벡터 크기에 따른 검색윈도우사이즈를 재조절하는 방법이다. 영상의 움직임이 작은 경우, 이 방법은 화질 및 계산시간은 우수하나, 영상의 움직임이 커지면 'Dynamic SR'와 같이 계산량이 증가하는 단점을 가지고 있다.

IV. 제안한 방법

본 논문에서는 그림(9)와 같이 H.264표준으로 압축된

표 1. H.264부호기의 매크로블록모드
Table 1. Macroblock mode of H.264.

종류	내용
SLICEmode	I (H264_SLICE_INTRA), P (H264_SLICE_INTER)
MBmode	SKIP Static(H264_MBmode_SKIP_STATIC), Constant(H264_MBmode_SKIP_CONSTANT)
	INTRA H264_MBmode_INTRA16x16, H264_MBmode_INTRA4x4
	INTER H264_MBmode_INTER16x16, H264_MBmode_INTER16x8, H264_MBmode_INTER8x16, H264_MBmode_INTER8x8, H264_MBmode_INTER8x4, H264_MBmode_INTER4x8 H264_MBmode_INTER4x4

비트스트림을 MPEG-2부호기로 변환하는 과정에서 P slice(frame)의 매크로블록모드의 특성을 분석하여 움직임벡터를 정교화하여 계산시간 감소 및 화질 증가시키는 방법을 제안한다.

표(1)는 그림(9)의 내용을 요약한 것으로 SLICEmode는 I Slice 또는 P Slice이다. 이때 H.264의 I Slice는 MPEG-2부호기에서 인트라로 부호화하고, H.264의 P Slice는 제안한 방법과 같이 부호화한다. H.264의 MBmode는 SKIP, INTRA, INTER로 총 3가지로 이루어졌다. MBmode의 세부적인 내용은 1, 2, 3절에서 언급한다.

1. SKIP Mode

MPEG-2 부호기에서 SKIP Mode는 현재 매크로블록과 참조 매크로블록의 화소 값이 같을 경우로 현재 매크로블록의 정보를 부호화하지 않고 움직임 벡터를 '0'으로 하여 참조 매크로블록을 그대로 사용하는 방법이다. 그러나 H.264부호기는 MPEG-2부호기와 달리 SKIP Mode는 고정(static) SKIP과 일정한(constant) SKIP Mode를 사용한다^[5]. 첫 번째 방법은 MPEG-2부호기의 SKIP Mode와 같이 움직임벡터를 '0'으로 하여 현재 매크로블록을 참조매크로블록을 그대로 사용하는 방법이다. 두 번째 방법은 주변의 움직임 벡터를 적용하여 그 위치의 화소를 그대로 사용하는 방법이다. 이 방법을 함께 사용하면 고정 SKIP Mode만을 사용했을 때보다 더 나은 효율을 갖게 된다^[5]. 본 논문에서는 연산량을 줄이기 위해 H.264의 매크로블록이 고정 SKIP Mode인 경우에는 MPEG-2부호기에서 움직임벡터를 '0'

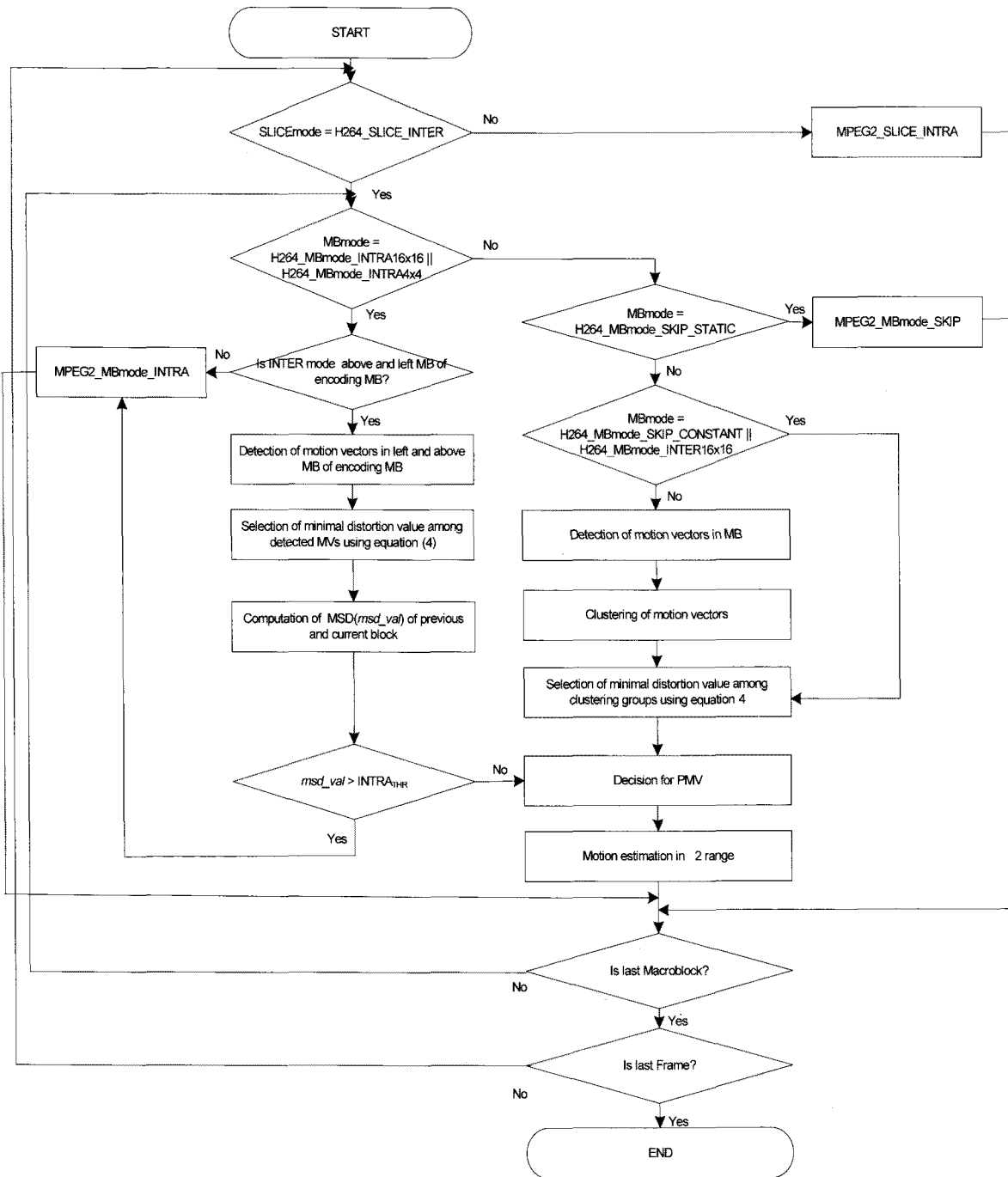


그림 9. 제안한 방법의 순서도
 Fig. 9. Flowchart of proposed method.

으로 부호화한다. 그리고 일정한 SKIP Mode인 경우에는 MPEG-2부호기에서 16x16 매크로블록으로 취급하여 움직임 벡터를 그대로 적용한다.

2. INTRA Mode.

H.264부호기의 인트라 모드는 2절에서 언급하였듯이

Intra Prediction을 사용하고, MPEG-2부호기의 인트라 모드는 이전 및 현재블록간의 예측오차값이 블록간의 차보다 크면 매크로블록을 인트라 모드로 전환하여 부호화하므로 H.264부호기의 인트라 모드 부호화방법이 상이하하다. 또한 표(2)와 같이 동일한 영상에 대하여 각각 H.264와 MPEG-2의 Inter slice을 부호화 한 후, 동

표 2. H.264와 MPEG-2부호기의 Intra Mode 일치율 (%)

Table 2. Match ratio of Intra mode for H.264 and MPEG-2

	Foreman	Soccer	Coastguard	Stefan	Bus
Intra Mode(%) 일치율	38.97	57.11	58.17	24.12	10.48

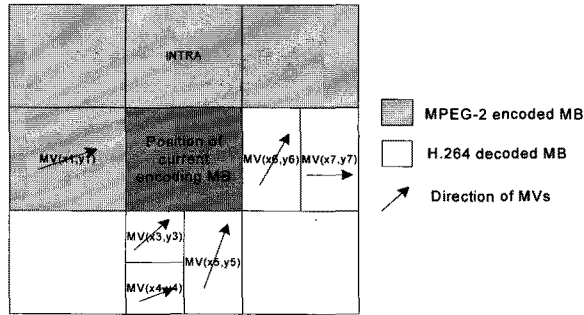


그림 10. 인트라모드 결정방법
Fig. 10. Decision of INTRA mode.

일한 위치의 매크로블록에 대하여 인트라 모드의 일치율을 비교한다. 영상의 크기에 따라 차이는 있지만 각각 부호기의 일치율은 평균적으로 37.77%이다. 따라서 H.264와 MPEG-2부호기의 인트라모드 차이로 인하여 H.264부호기의 인트라모드를 MPEG-2 부호기에서 그대로 사용할 수 없다.

본 논문에서는 현재 부호화할 매크로블록은 주변 매크로블록의 움직임벡터와 상관관계가 있으므로 H.264 부호기의 매크로블록 모드가 H264_MBmode_INTRA 4x4 또는 H264_MBmode_INTRA16x16이면, 그림(10)과 같이 MPEG-2부호기에서 부호화할 매크로블록의 상, 하, 좌, 우 블록의 움직임 벡터를 검출한다. 검출된 움직임 벡터는 식(4)을 이용하여 왜곡 값을 계산한다. 최소 왜곡 값을 가진 움직임벡터는 이전프레임과 현재프레임간의 SAD를 계산한다. 그리고 SAD값이 $INTRA_thr$ ($=2,304$)보다 크면 인트라 모드로 결정하고, 작은 경우에는 예측움직임벡터로 결정한다. 이때 $INTRA_thr$ 는 MPEG-2부호기에서 인트라 및 인터모드를 결정할 때 사용하는 값이다^[20].

3. INTER Mode

본 논문에서는 H.264부호기의 가변블록안에 존재하는 움직임벡터(1~16개)들을 MPEG-2부호기에서 재사용하기 위하여 표(3)과 같이 움직임벡터의 일치율을 비

표 3. H.264 및 MPEG-2부호기의 움직임벡터 일치율 (%)

Table 3. Match ratio of MV for H.264 and MPEG-2.

	Bus	SKI	Coastguard	Foreman	Stefan	Football
MV값 일치	75.37	76.04	81.94	78.11	76.20	85.64
MV값의 ±2이내	93.33	90.23	96.41	93.85	87.03	97.99

교하였다. 실험방법은 'MV값 일치'와 'MV값의 ±2 이내'를 실험하였다. 즉, 'MV값 일치'는 H.264부호기의 가변블록 안에 존재하는 움직임벡터 값과 MPEG-2부호기에서 전역탐색방법으로 추정된 움직임벡터 값의 일치이다. 'MV값이 ±2 이내'는 H.264부호기의 가변블록에 존재하는 움직임벡터 값이 MPEG-2부호기의 전역탐색방법으로 추정된 움직임벡터 값의 ±2 이내에 존재하는 것을 말한다. 실험결과, 일치율은 영상의 움직임에 따라 차이가 있지만 평균적으로 'MV값 일치'보다 'MV값의 ±2 이내'에 존재하는 영상들이 많다.

따라서 본 논문은 H.264부호기의 가변블록에 존재하는 움직임벡터를 MPEG-2부호기에서 재사용한다면 MPEG-2부호기의 움직임추정 시간을 효율적으로 감소할 수 있을 것이다. 또한 H.264부호기의 인터 모드는 16x16블록과 16x16블록을 제외한 나머지 블록으로 나누어서 움직임벡터를 결정한다.

가. 16x16블록.

16x16블록은 H.264 및 MPEG-2부호기에서 모두 사용하고 있으며 연산과정도 비슷하다. 이로 인해 H.264 표준의 16x16블록의 움직임 벡터는 MPEG-2의 움직임 벡터와 유사한 경우가 상당한 많다. 따라서 본 논문에서는 H.264의 매크로블록이 16x16인 경우 별다른 연산을 하지 않고 재사용한다.

나. 16x16블록을 제외한 나머지 블록

16x16블록을 제외한 나머지 블록(Mode2~Mode8)의 움직임벡터는 최대 16개를 가질 수 있으므로 이 중에서 최적의 움직임벡터를 예측해야한다. 그림(11)은 본 논문에서 제안한 아이디어로 매크로블록이 전경 또는 배경에 속한 경우에는 가변블록의 움직임벡터가 거의 일치하지만, 매크로블록이 전경 및 배경에 걸친 경우에는 그림(11)의 'A'부분을 확대한 바와 같이 가변블록 안의 움직임벡터는 다양한 방향을 가지고 있다.

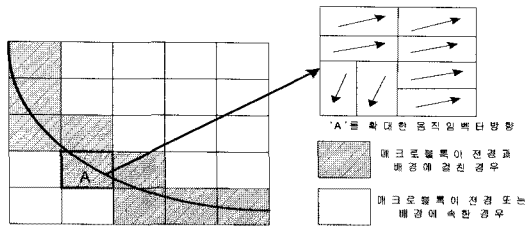


그림 11. 가변블럭내의 다양한 방향을 가진 움직임벡터
Fig. 11. MVs having multi-directions in variable blocks

따라서 본 논문에서는 다양한 방향을 가지고 있는 움직임벡터들 간에 거리를 이용하여 클러스터링하여 동일한 방향으로 움직인다는 것을 가정하고 움직임벡터를 다음과 같이 예측한다. 첫째, H.264복호기의 가변블럭안에 존재하는 움직임벡터들을 검출한다. 둘째, 검출된 움직임 벡터들의 거리를 계산하기 위하여 클러스터링의 대표적인 평균연결알고리즘(average linkage algorithm)을 수정하여 클러스터링 집합을 생성한다. 셋째, 클러스터링 집합들 중에서 중앙값을 취하여 후보예측움직임벡터로 결정하고, 결정된 후보예측움직임벡터에서 최소의 왜곡값을 예측움직임벡터로 결정한다. 마지막으로 예측움직임벡터 중심으로 ±2 반화소 검색을 실시하거나, 검색없이 바로 움직임벡터로 결정하여 부호화한다.

Step 1. 움직임벡터 검출

H.264복호기에서 MPEG-2부호기로 전송되어지는 인터 모드 안에 존재하는 가변블록의 움직임벡터들을 이용한다. 이때 식(2)과 같이 가변블록의 움직임벡터는 모드별로 최소 1개에서 최대 16개까지 움직임벡터를 검출할 수 있다.

$$MV_{set} = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}, 1 \leq n \leq 16 \quad (2)$$

Step 2. 클러스터링 집합 생성

앞 단계에서 움직임 벡터를 검출했다면 식(3)과 같이 움직임벡터들 간의 거리를 구한다. 식(3)에서 움직임 벡터는 (x_{n-1}, y_{n-1}) 와 (x_n, y_n) 이고, D는 움직임 벡터간의 거리이다.

$$D = \sqrt{|x_{n-1} - x_n|^2 + |y_{n-1} - y_n|^2}, (n \geq 2) \quad (3)$$

식(3)을 이용하여 움직임 벡터간의 거리를 구하고, 최소가 되는 두 블록간의 평균값을 계산하는 과정을 반복함으로써 평균연결알고리즘을 수행한다. 그림 12(a)

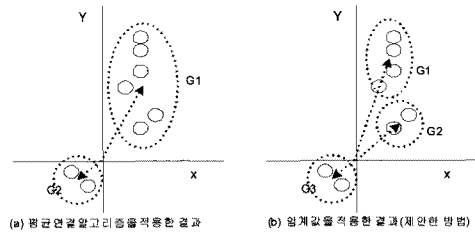


그림 12. 기존 ALA방법과 제안한 클러스터링 비교
Fig. 12. Comparison of ALA and proposed method.

는 그림(11)의 'A' 매크로블록의 다양한 방향을 가진 움직임벡터들을 평균연결알고리즘을 사용한 클러스터링 결과이다.

그림 12(a) 결과에서 보듯이 클러스터링 집합은 G1, G2가 생성되었다. 즉, 평균연결알고리즘은 최종적으로 두 개의 클러스터링을 생성하므로 클러스터링된 움직임 벡터들은 거리가 떨어져 있어도 같은 클러스터링으로 만드는 단점을 가지고 있다. 따라서 본 논문에서는 그림12(b)와 같이 움직임 벡터간의 거리가 임계값(=3) 이상인 경우, 같은 클러스터링(G1)으로 분류하지 않고 다른 클러스터링(G2)으로 분류하여 총 3개를 구성한다.

Step 3. 예측움직임벡터 결정

이 단계에서는 클러스터링 집합내에서 중앙값을 후보예측움직임벡터로 결정한다. 그리고 예측움직임벡터를 결정하기 위하여 식(4)과 같이 비용함수를 사용하여 왜곡을 측정한다. 또한 그림(13)은 식(4)를 도식화한 것이다.

$$\begin{aligned} dis_T(k) &= \sum_{i=0}^m |f(x, y) - f(x+1, y)| \\ dis_B(k) &= \sum_{i=0}^m |f(x, y+n) - f(x+1, y+n)| \\ dis_L(k) &= \sum_{i=0}^n |f(x, y) - f(x, y+1)| \\ dis_R(k) &= \sum_{i=0}^n |f(x+m, y) - f(x+m, y+1)| \\ Cost_{fun}(k) &= (dis_T(k) + dis_B(k) + dis_L(k) + dis_R(k)) / (m+n) \times k \end{aligned} \quad (4)$$

식(4)에서 $dis_T(k), dis_B(k), dis_L(k), dis_R(k)$ 는 k번째 가변블록 내에 존재하는 경계의 위(T), 아래(B), 왼쪽(L), 오른쪽(R)에 대한 왜곡 값(dis)이다. m, n은 가변블록의 가로 및 세로 길이이고, $f(x, y)$ 는 현재 프레임의 픽셀의 좌표이다. 그리고 비용함수인 $cost_{fun}(k)$ 는 왜곡값의 총합을 평균한 것이다. 이때 블록의 전체 SAD 계산은 많은 계산시간을 요구하고, 블록내부의 영상이 복

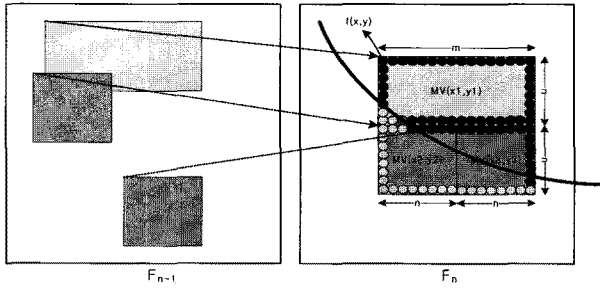


그림 13. 비용함수계산
Fig. 13. Computation of cost function.

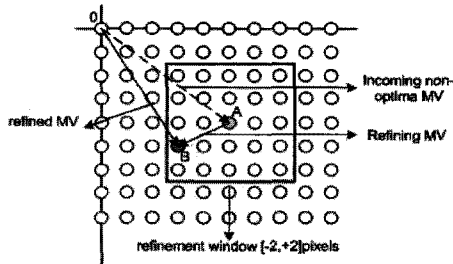


그림 14. 움직임벡터 정교화
Fig. 14. Refinement of MV.

잡할 때 SAD 값은 일반적으로 의미가 없다. 따라서 블록내의 경계를 선택하여 왜곡 값을 계산한다. 마지막으로 각 가변블록 중 비용함수의 최대값을 가진 가변블록의 움직임벡터를 예측움직임벡터로 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 최대값을 선정한다.

그리고 마지막으로 최종움직임벡터결정은 2가지 방법으로 실험하였다. 첫째, Proposed(R=0)는 예측움직임벡터를 바로 최종움직임벡터로 결정하여 검색원도우사이즈(R)를 '0'으로 설정하여 실험하였다. 둘째, Proposed(R=2)는 그림(14)과 같이 예측움직임벡터 중심으로 ± 2 범위내(R=2)에서 반화소 검색을 실시하여 최소인 SAD값을 가진 움직임벡터를 최종적으로 부호화한다.

V. 실험 및 결과

실험환경은 Pentium IV CPU 3.06GHz의 PC에서 VC++ 2005를 사용하였고, 트랜스코더의 구성은 H.264 JM10.2 복호기와 MPEG Software Simulation Group의 MPEG-2 부호기를 사용하였다. 테스트 영상은 CIF(352x288)영상(FOREMAN,CONTAINER,MOBILE,

SKI, STEFAN, SOCCER)을 각각 300프레임씩 30fps으로 1Mbits/sec의 비트율로 부호화하였다. 또한 초기 양자화 파라미터는 28, GOP 사이즈는 15로 B프레임을 제외하고 I와 P프레임만 사용하고, 움직임벡터 검색범위는 ± 15 , 참조프레임은 1개, CAVLC, RDO를 사용하였다.

다음 표(4)~(6)는 각 영상에 대하여 1Mbit/sec로 트랜스코딩 하였을 때 'MPEG-2 FSBMA', 'Weighted Average', 'Dynamic SR', 'Dynamic ME', 'Proposed (R=0)', 'Proposed(R=2)에 대한 PSNR', 트랜스코딩계산 시간, 움직임추정계산시간이다.

PSNR측면에서 'Weighted Average'는 'MPEG-2 FSBMA'보다 0~2dB 정도 화질이 떨어졌고, 'Dynamic SR', 'Dynamic ME', 'Proposed(R=0)', 'Proposed(R=2)'은 'MPEG-2 FSBMA'와 거의 비슷하였다.

트랜스코딩계산시간측면에서 'Weighted Average', 'Dynamic SR', 'Dynamic ME', 'Proposed(R=0)', 'Proposed(R=2)'은 'MPEG-2 FSBMA'에 비하여 평균적으로 각각 70.08%, 66.79%, 68.80%, 70.36%, 67.64%로 감소하였다. 이때 'Proposed(R=0)'은 예측움직임벡터를 바로 재사용하였으므로 기존 방법에 비하여 0.2%~3.57%감소하였다. 하지만 'Proposed(R=2)'은 예측움직임벡터 중심으로 2이내에서 반화소검색을 하였으므로 기존방법과 거의 비슷하였다.

움직임추정계산시간측면에서 'MPEG-2 FSBMA'는 움직임추정시간이 평균적으로 78.87%을 차지하였고, 'Weighted Average', 'Dynamic SR', 'Dynamic ME', 'Proposed(R=0)', 'Proposed(R=2)'은 각각 평균적으로 30.17%, 36.39%, 29.71%, 28.06%, 34.26%만큼 계산시간이 줄었다. 하지만 실험결과에서 보여 지듯이 움직임이 큰 영상과 작은 영상에 따라 다소 차이점을 보이고 있다. 즉 그림(15)~(16)과 같이 움직임이 작은 영상인 'CONTAINER' 영상을 살펴보면 기존방법이나 제안한 방법의 트랜스코딩계산시간 및 PSNR은 다소 차이가 없다. 하지만 'FOREMAN', 'STEFAN', 'COASTGUARD', 'SKI', 'SOCCER'영상과 같이 움직임이 큰 영상은 검색원도우사이즈가 증가되므로 그림(17) 및 (19)과 같이 'Dynamic-SR' 및 'Dynamic-ME'의 트랜스코딩계산시간은 증가한다. 즉, 그림(17)은 'FOREMAN'영상의 실험결과로 'Dynamic-SR'의 트랜스코딩계산시간으로 움직임이 많아지는 180th~230th프레임에서 계산시간이 증가하는 모습을 보이고 있다. 또한 그림 (19)은 'SKI'

표 4. 테스트영상에 대한 PSNR 성능비교(dB)
Table 4. Comparison of PSNR in test sequences.

	MPEG-2 FSBMA	W-average [15]	Dynamic-SR [16]	Dynamic-ME [17]	Proposed (R=0)	Proposed (R=2)
CONTAINER	38.06	38.06	38.06	38.06	38.06	38.07
FOREMAN	36.62	35.92	36.59	36.48	36.34	36.57
SOCCER	36.62	35.65	37.07	36.57	36.51	36.64
SKI	42.78	42.70	43.01	42.98	42.98	42.99
STEFAN	30.35	28.38	30.76	30.20	30.06	30.27
MOBILE	26.49	24.85	26.17	26.24	26.06	26.52

표 5. 테스트영상에 대한 트랜스코딩 계산시간 성능비교(sec)
Table 5. Comparison of transcoding computation time in test sequences

	MPEG-2 FSBMA	W-average [15]	Dynamic-SR [16]	Dynamic-ME [17]	Proposed (R=0)	Proposed (R=2)
CONTAINER	36.03	13.16(63.47%)	13.70(61.98%)	13.60(62.25%)	13.31(63.06%)	13.98(61.20%)
FOREMAN	44.80	14.77(67.03%)	17.15(61.72%)	15.07(66.36%)	14.75(67.08%)	16.43(63.33%)
SOCCER	57.45	14.81(74.22%)	18.64(67.55%)	15.84(72.43%)	15.56(72.92%)	16.02(72.11%)
SKI	46.32	13.38(71.11%)	14.77(68.11%)	14.89(67.85%)	12.63(72.73%)	13.73(70.36%)
STEFAN	52.68	14.38(72.70%)	16.00(69.63%)	14.76(71.98%)	14.62(72.25%)	16.21(69.23%)
MOBILE	54.90	15.37(72.00%)	15.51(71.75%)	15.39(71.97%)	14.20(74.13%)	16.68(69.62%)

표 6. 테스트영상에 대한 움직임추정시간 성능비교(sec)
Table 6. Comparison of motion estimation time in test sequences

	MPEG-2 FSBMA	W-average [15]	Dynamic-SR [16]	Dynamic-ME [17]	Proposed (R=0)	Proposed (R=2)
CONTAINER	26.01(72.19%)	3.31(25.15%)	3.41(24.89%)	3.60(26.47%)	3.32(24.94%)	4.03(28.83%)
FOREMAN	34.92(77.95%)	4.68(31.69%)	7.01(40.87%)	4.57(30.33%)	4.76(32.27%)	6.58(40.05%)
SOCCER	47.59(82.84%)	4.76(32.14%)	8.61(46.19%)	5.15(32.51%)	4.55(29.24%)	6.03(37.64%)
SKI	36.49(78.78%)	3.59(26.83%)	4.94(33.45%)	3.73(25.05%)	2.73(21.62%)	3.90(28.40%)
STEFAN	42.43(80.54%)	4.47(31.08%)	6.12(38.25%)	4.55(30.83%)	4.76(32.56%)	6.10(37.63%)
MOBILE	44.76(81.53%)	5.25(34.16%)	5.38(34.69%)	5.09(33.07%)	3.94(27.75%)	5.51(33.03%)

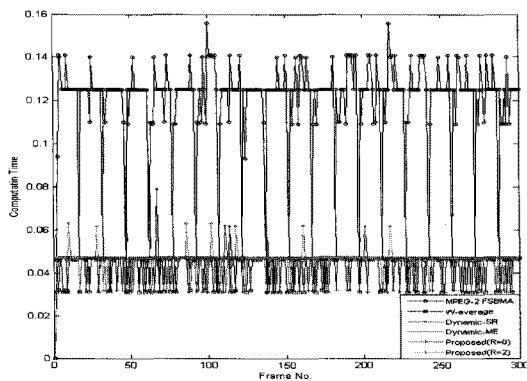


그림 15. CONTAINER영상의 트랜스코딩계산시간비교 (sec)

Fig. 15. Comparison of transcoding computation time in CONTAINER sequence.

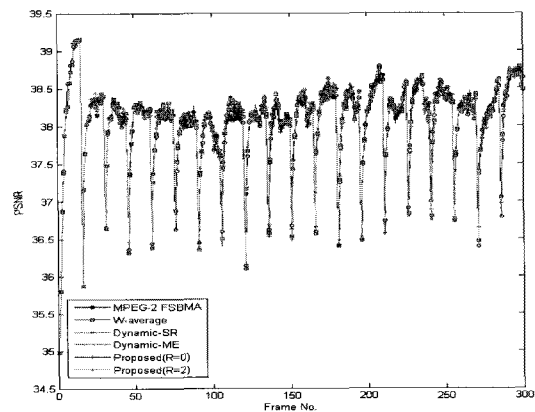


그림 16. CONTAINER 영상의 PSNR 성능 비교(dB)

Fig. 16. Comparison of PSNR in CONTAINER sequence.

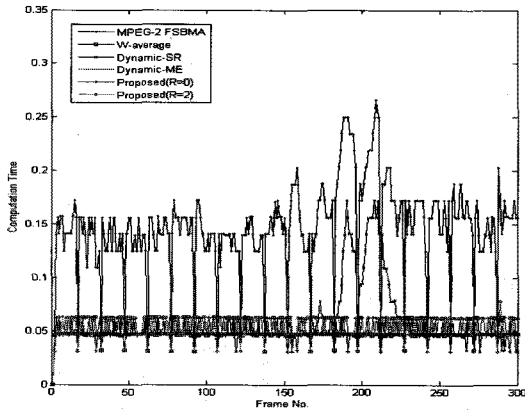


그림 17. FOREMAN 영상의 트랜스코딩계산시간비교 (sec)

Fig. 17. Comparison of transcoding computation time in FOREMAN sequence.

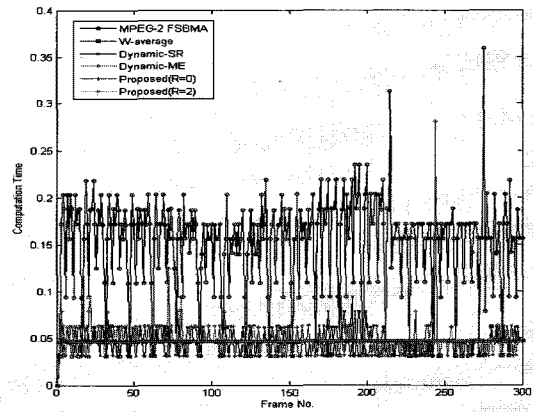


그림 19. SKI영상의 트랜스코딩 계산시간 비교(sec)

Fig. 19. Comparison of transcoding computation time in SKI sequence.

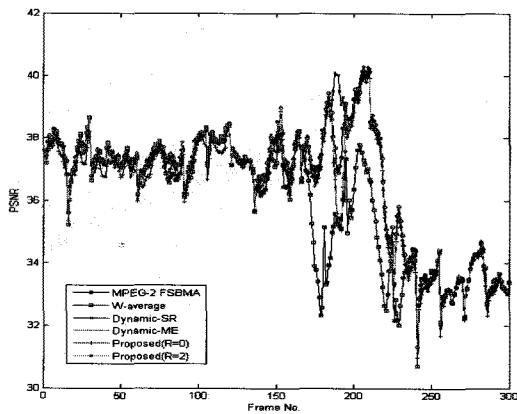


그림 18. FOREMAN 영상의 PSNR 성능 비교(dB)

Fig. 18. Comparison of PNSR in FOREMAN sequence.

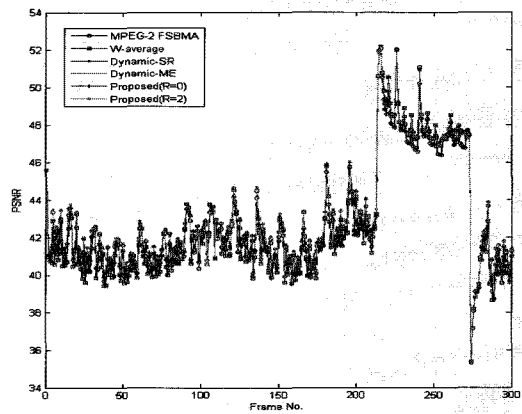


그림 20. SKI영상의 PSNR성능 비교(dB)

Fig. 20. Comparison of PNSR SKI sequence.

영상의 트랜스코딩계산시간으로 움직임이 많아지는 프레임 부분에서 'Dynamic SR'와 'Dynamic ME'의 트랜스코딩계산시간은 증가하였다. 하지만 'Proposed(R=0)'의 트랜스코딩계산시간은 움직임이 많은 영상과 관계없이 일정하게 유지하는 모습을 보이고 있다. 그리고 그림(18) 및 그림(20)와 같이 움직임이 많은 영상의 PSNR은 'MPEG-2 FSBMA', 'Dynamic-SR', 'Dynamic-ME', 'Proposed(R=0)', 'Proposed(R=2)'은 거의 유사하였지만, 'Weighted Average'의 PSNR은 움직임이 많은 영상에서 PSNR이 감소하였다.

마지막으로 본 논문에서 제안한 'Proposed(R=0)'과 'Proposed(R=2)'를 비교해보면, 'Proposed(R=0)'은 검색 윈도우 사이즈 없이 바로 예측움직임벡터를 최종움직임 벡터로 결정하기 때문에 'Proposed(R=2)'보다 트랜스코

딩 계산시간은 감소하면서 PSNR의 화질은 거의 비슷 하였다. 따라서 H.264복호기의 매크로블록에 존재하는 다양한 움직임벡터의 방향성을 고려하여 클러스터링한 후, 최적의 움직임벡터를 추출하여 MPEG-2부호기에서 효율적으로 재사용 할 수 있었다.

VI. 결론 및 향후 연구

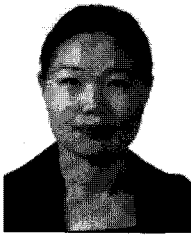
현재 선박과 육지간의 서비스는 위성통신을 통하여 간단한 영상이나 문자를 전송하기 때문에 선박은 육지에서 전송중인 다양한 콘텐츠를 제공받지 못한 상태이다. 따라서 육지에서 유통되고 있는 다양한 비디오 콘텐츠를 위성통신을 통하여 선박에서도 서비스되는 방법이 필요하다. H.264 to MPEG-2 트랜스코딩이 선박과 육지 간에 새로운 타입의 서비스를 제공된다면 기존 선

박에 구축된 MPEG-2 인프라 장비를 활용할 수 있는 계기가 될 것이다. 본 논문에서는 H.264에서 MPEG-2로 트랜스코딩의 효율을 위하여 MPEG-2부호기에서 계산량을 가장 많이 차지하는 움직임추정을 효율적으로 감소하기 위하여 가변블럭안에 존재하는 다양한 움직임 벡터의 방향성을 고려하여 최적의 움직임벡터를 예측하는 방법을 제안하였다. 향후 연구는 본 논문에서 제안한 방법을 B 프레임에 적용과 움직임벡터를 효율적으로 추정하기 위하여 적응적인 검색윈도우사이즈 알고리즘 개발이다.

참 고 문 헌

- [1] INMARSAT, "국제해사위성기구에 관한 협약 및 국제해사위성기구에 관한 운용협정," 1979.
- [2] 이상면, "이동통신의 발달을 위한 국제해사위성기구(INMARSAT)의 역할," 서울대학교 법학, 제34권 2호, 1993.
- [3] ISO/IEC 13818-2: Information Technology Generic Coding of Moving Pictures and Associated Audio Information. Part2: Video
- [4] Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification(ITU-T Rec. H.264/ISO/IEC 14496-10 AVC) JVT-G050, Geneva, Switzerland, 23-27 May, 2003.
- [5] Thomas Wiegand, Heiko Scharz, Anthony Joch, Faouzi Kossentini, Gary J. Sullivan, "Rate-Constrained Coder Control and Comparison of Video Coding Standards", *IEEE Trans. On Circuits and Systems for Video Tech.*, Vol. 13, No.7, July 2003.
- [6] P. Assuncao and M. Ghanbari, "A frequency domain video transcoder for dynamic bit-rate reduction of MPEG-2 bitstreams," *IEEE Trans Circ Syst Video Technol* 8, p.953 - 967, 1998.
- [7] C.W. Lin and Y.R. Lee, "Fast algorithms for DCT-domain video transcoding," in *Proceedings of the IEEE International Conference on Image Processing*, Thessaloniki, Greece, Vol. I, p. 421 - 424, 2001.
- [8] P. Yin, M. Wu, and B. Lui, "Video transcoding by reducing spatial resolution," in *Proceedings of the IEEE International Conference on Image Processing*, Vancouver, BC, Canada, Vol. I, pp. 972 - 975, 2000.
- [9] P. Yin, M. Wu, B. Lui, and H. Sun, "Drift compensation for reduced spatial resolution transcoding," *IEEE Trans Circ. Syst. Video Technol.* 12, P.1009 - 1020, 2002.
- [10] Keesman, g. et al., "Transcoding of MPEG Bitstreams," *Signal Processing: Image Commun.* 8, p. 481-500, 1996.
- [11] H.264, Draft ITU-T Recommendation and Final Draft International Standard, Pattaya, Thailand, 2003.
- [12] H. S. Malvar, A. Hallapuro, and M. Karzewicz, "Low-Complexity Transform and Quantization in H.264/AVC," *IEEE Trans. Circuits Syst.Video Technol.*, vol. 13, no. 7, pp.598 - 603, Jul. 2003.
- [13] Kalva, H.: Issues in H.264/MPEG-2 video transcoding. *Computer Science and Engineering*, 2004.
- [14] Yang, L., Song, X., Hou, C., Dai, J.: H.264 to MPEG-2 transcoding based on personal video recorder platform. In *Proc. of the Ninth International Symposium on Consumer Electronics*, P.438-440, 2005.
- [15] S. Sharmat, "Transcoding of H.264 BITSTREAM to MPEG-2 BITSTREAM", Master's thesis, 2007.
- [16] Kunzelmann, P., Kalva, H., "Reduced complexity H.264 to MPEG-2 transcoder," *ICCE International Conference on Consumer Electronics*, 2007.
- [17] 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, 2008.
- [18] Abdul H. Sadka, "Compressed Video Communications", John Wiley & Sons, LTD, pp.232~234, 2002.
- [19] <http://iphome.hhi.de/suehring/tml/>
- [20] <http://www.downloadatoz.com/codecs/mpeg2-video-codecs.php>

— 저 자 소 개 —



손 남 례(정회원)
 1996년 호남대학교 컴퓨터
 공학과 학사 졸업
 2000년 전남대학교 전산학과
 석사 졸업
 2005년 전남대학교 전산학과
 박사 졸업

2007년~2009년 2월 호남대학교 인터넷소프트
 웨어학과 전임강사

2009년 12월~현재 목포대학교 중점연구소
 연구교수

<관심분야> 영상통신, IPTV, 3D 비디오 코딩,
 방송통신융합기술, 임베디드시스템



정 민 아(정회원)
 1992년 전남대학교 전산통계 학사
 졸업
 1994년 전남대학교 전산통계 석사
 졸업
 2002년 전남대학교 전산통계 박사

2005년 3월~현재 목포대학교 컴퓨터공학과
 조교수

<관심분야> 데이터베이스/데이터마이닝, 생체인
 식시스템, 무선통신응용분야(RFID, USN, 텔레메
 틱스), 임베디드시스템



이 성 로(정회원)
 1987년 고려대학교 전자공학
 학사 졸업.
 1990년 한국과학기술원 전기및
 전자공학 석사 졸업.
 1990년 한국과학기술원 전기및
 전자공학 박사 졸업.

2009년~현재 목포대학교 정보전자공학과 교수.

<주관심분야 : 디지털통신, 위성통신, 해양텔레메
 틱스, 인공지능, USN>