압축 비디오에서 중간정보를 이용한 트랜스 부호화의 움직임 추정

구성조*, *김강욱, **김종훈, *황찬식
*경북대학교 전자·전기공학부
**동양대학교 정보통신 공학부

Motion Estimation for Transcoding Using Intermediate data on the Compressed Video

°Sung-Jo Koo*, *Kang-Wook Kim, **Jong-Hoon Kim, *Chan-Sik Hwang
*School of Electronic and Electrical Engineering, Kyungpook National University
**School of Information and Communication Engineering, Dongyang University

요 약

높은 비트율로 부호화 되어있는 비디오 스트림을 낮은 비트율로 다시 부호화할 때 입력 비트 스트림으로부터 뽑아낸 움직임 벡터를 그대로 재 사용한다면 양자화 에러에 의해 심각한 화질의 열화가 발생하게 된다. 따라서 본 논문에서는 이러한 트랜스 부호화 시에 계산량을 줄이면서 화질의 열화를 방지하는 새로운 알고리즘을 제안한다. 제안한 방법은 입력 비트 스트림으로부터 뽑아낼 수 있는 중간 정보들을 이용해서 움직임 벡터를 정재해서 사용할 것인지 아니면 그대로 재 사용할 것인지를 판단한다. 제안한 방법은 영상의 움직임 특성에 따른 임계값을 결정할 수 있어서 움직임 벡터를 단순히 재 사용하는 방법보다 계산량을 줄이면서 화질의 열화를 방지할 수 있다.

ABSTRACT

In transcoding, simply reusing the motion vectors extracted from an incoming video bit stream may not result in the best quality because the incoming motion vectors become non-optimal due to the reconstruction errors. To achieve the best video quality possible, a new motion estimation should be performed in the transcoder. An adaptive motion vector refinement is proposed that refines the base motion vector according to the activity of macroblock using intermediate data extracted from an incoming video bit stream. Experimental results shows that the proposed method can improve the video quality to the level achieved by using the full-scale motion estimation with minimal computational complexity.

Ⅰ. 서 론

최근에 화상 회의, 주문형 비디오, 원격 강의 같은 네트워크 상의 멀티미디어 서비스들이 크게 증가하고 있다. 현재 제공되는 대부분의 비디오 서비스와 멀티미디어 응용에서는 미리 부호화 되어있는 비디오를 사용한다. 이러한 응용 서비스를 다양한 네트워크에 접속된 사용자가 이용하기 위해서는 부호화 된 비디오 비트 스트림의 비트율을 다양한 채널의 대역폭에 맞추는 것이 필요하다. 만약 10 Mbits/s로 부호화 되어있는 비디오를 대역폭이 좁은 채널로 그대로 전송한다면 심각한 화질의 열화가 발생하게 된다. 그러므로 높은 비트율로 부호화 되어있는 비디오를 제한된 대역폭을 가지는 채널로 보낼 시에는 적절한 비트율로 바꾼 후 전송해 주어야 한다[1]. 이러한 과정을 트랜스 부호화(transcoding)라고 한다. 일반적으로 트랜스 부호화기(transcoder)는 부호화기와 복호화기를 직렬로 연결한 구조를 사용한다. 직렬 연결된 트랜스 부호화기는 복잡한 구조를 가지는 것이 단점이지만 부호화 과정 중에 입력 비디오 스트림으로부터 여러 정보를 추출해 낼 수 있으므로 전체적인 트랜스 부호화과정에서 본다면 많은 장점들을 가질 수 있다[2-4].

특정 비트율로 부호화 되어있는 비디오를 원하는 특정 비트율로 다시 변환하기 위해서는 복호화해서 다시 부호화 과정을 수행해야 되기 때문에 계산량의 증가와 더불어 전송시간에 문제가 발생하게 된다. 일반적으로 비디오를 부호화 하는 과정 중에서 움직임 추정 부분이 계산량이 가장 많으므로 이 부분에서 계산량을 줄인다면 트랜스 부호화를 수행하는데 소모되는 시간을 크게 줄일 수 있다. 이를 위해서 입력 비트 스트림으로부터 추출해낸 움직임 벡터인 기저 움직임 벡터(base motion vector)를 그대로 이용하는 재 사용하는 방법이 있다. 하지만 단순히 기저 움직임 벡터를 그대로 사용하게 되면 양자화 에러에 의해 영상의 열화가 발생하게 된다. 그래서 이러한 영상의 열화를 막기 위해서 전 탐색 움직임 추정을 수행해서 최적의 움직임 벡터를 찾아낼 수는 있으나 계산량이 아주 많아지게 되고 전송시간에 문제가 발생하게 된다. 그러므로 기저 움직임 벡터를 그대로 사용하지 않고 정재(refinement)과정을 거치고 난 후에 사용한다면 계산량을 상당히 줄이면서 영상의 화질 열화를 방지할 수 있다[5]. 그러나 매크로블록의 특성을 고려하지 않고 모든 매크로블록에 대해서 정재 과정을 수행한다면 비효율적이다. 왜냐하면 움직임이 크지 않은 매크로블록은 정재 과정 없이 기저 움직임 벡터를 재 사용해도 화질의 열화가 거의 없는 경우가 많이 존재할 수 있기 때문이다.

본 논문에서는 압축된 비디오를 낮은 비트율로 트랜스 부호화 할 때 입력 비디오 스트림으로부터 추출해낸 중간 정보(intermediate data)들를 이용해서 기저 움직임 벡터를 적응적으로 정재해서 사용하는 방법을 제안한다. 제안한 방법은 복호화 과정에서 뽑아낸 매크로블록의 AC 계수들의 에너지와 움직임 벡터의 거리를 사용해서 매크로블록의 활동도를 계산해서 정재 과정을 적응적으로 수행한다.

II. 트랜스 부호화

미리 압축되어 있는 비디오 스트림을 트랜스 부호화를 이용해서 낮은 비트율로 변환하기 위해서는 다양한 채널의 환경에 맞게 비디오 스트림의 비트율을 적절히 바꾸어 주는 것이 중요하다. 트랜스 부호화를 위한 가장 간단한 구조는 DCT 계수를 변형함으로써 입력 비트율을 낮추는 개방형(open-loop) 트랜스 부호화기이다[6]. 개방형 구조에서는 DCT 계수들을 자르거나 재 양자화 하거나 부분적으로 생략을 해서 원하는 비트율을 얻을 수 있다. 개방형 트랜스 부호화는 완전 복호화 및 재 부호화가 없이 부호화 된 영역에서 직접 수행되므로 간략하고 빠른 트랜스 부호화기에 사용될 수 있다. 그러나 개방형 트랜스 부호화는 서로 다른 복원 영상에 의해 화질의 열화가 누적되어 전파되는 단점이 있다. 이러한 단점을 해결하기 위해서 그림 1과 같이 복화기와 부호화기를 직접 연결한 직렬 트랜스 부호화기를 사용한다.

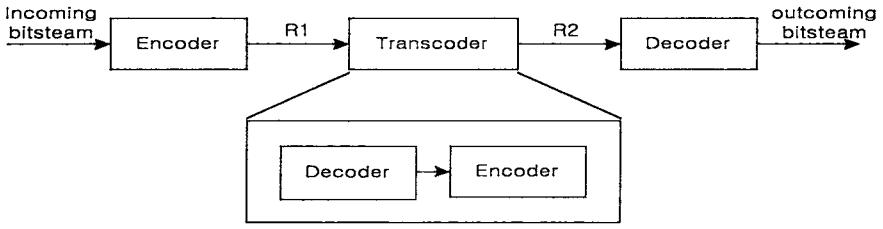


그림 1. 직렬 트랜스 부호화기.

Fig. 1. Cascaded transcoder.

R_1(bits/s) 비트율로 부호화된 비디오 스트림을 R_2(bits/s) 비트율로 변환하기 위해서는 입력 비디오 스트림을 복호화한 후 다시 한번 더 부호화과정을 수행해야한다. 그런데 그림 1에서처럼 직렬 연결된 트랜스 부호화기는 수행과정이 개방형 구조보다 복잡하지만 입력 비트 스트림에서 뽑아낸 여러 가지 정보들을 이용해서 부호화기의 복잡성을 많이 제거할 수 있으므로 계산량은 큰 문제가 되지 않는다.

본 논문에서는 트랜스 부호화의 기본 구조로 복호화기와 부호화기를 직렬로 연결한 직렬 트랜스 부호화기를 사용한다. 이러한 트랜스 부호화기는 시간적, 공간적 해상도 변환같은 트랜스 부호화 형태에 아주 유연하고 쉬게 확장할 수 있다.

1. 트랜스 부호화에서 움직임 추정

트랜스 부호화 하는 과정 중에서 가장 계산량이 많고 시간이 오래 걸리는 부분이 움직임 추정부분이다. 기본적으로 MPEG에서 움직임 추정하는 방법은 전 탐색(fullsearch)이다.

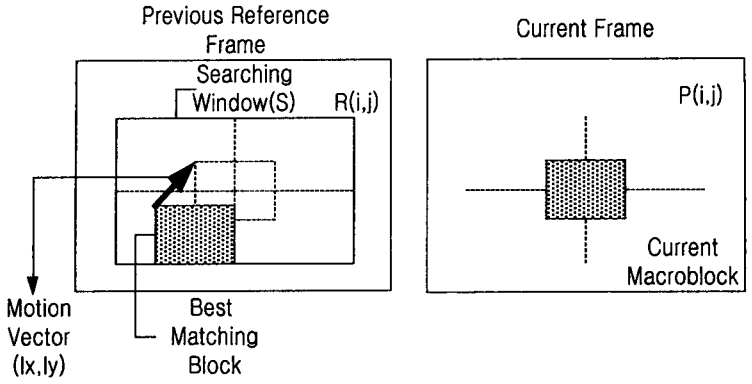


그림 2. 전 탐색 움직임 추정.

Fig. 2. Fullsearch motion estimation.

현재 매크로블록(macroblock)의 움직임 벡터를 구하기 위해서는 식 (1)과 (2)에서처럼 움직임 벡터를 구하고자하는 현재 프레임의 매크로블록과 이전 프레임의 탐색 영역 내의 매크로블록과의 픽셀 값의 차를 구한 후 절대값의 합(SAD)을 구해서 그 값이 가장 작은 매크로블록을 찾은 후 두 블록간의 움직임 거리를 움직임 벡터로 정한다.

$$(I_x, I_y) = \arg\min_{(m,n)\in S} SAD(m,n) \tag{1}$$

$$SAD(m,n) = \sum_{i}^{i+16}\sum_{j}^{j+16} |P(i,j) - R(i+m, j+n)| \tag{2}$$

이러한 방법으로 움직임 벡터를 구한다면 정확하게 움직임 벡터를 구할 수는 있으나 탐색 영역이 (-15, 15) 만큼 되어서 계산량이 많아진다. 그러나 기저 움직임 벡터와 위와 같은 전 탐색 추정으로 구한 움직임 벡터는 비트율에 따른 차이는 있지만 유사한 것이 많이 존재할 것이다. 그러므로 기저 움직임 벡터를 적절히 잘 사용한다면 계산량도 줄이면서 최적의 움직임 벡터를 찾아낼 수 있을 것이다.

2. 움직임 벡터의 정제

최적의 움직임 벡터는 부호화기에서의 움직임 추정에 의해 얻을 수 있으나 이는 많은 계산량을 요구하기 때문에 바람직하지 않다. 그래서 기저 움직임 벡터를 이용해서 움직임 벡터를 구하는 방법이 사용된다. 그런데 비트율 변화에 의한 양자화 에러의 영향 때문에 기저 움직임 벡터를 그대로 사용할 수 없고 이 벡터를 이용해서 움직임 벡터를 찾아 내야한다. 식 (3), (4)와 같이 기저 움직임 벡터 (B_x, B_y)를 기준으로 해서 움직임 벡터를 구하게 되면 작은 탐색 영역에서 최적에 가까운 움직임 벡터를 전 탐색 추정 없이 얻을 수 있다.

$$(I_x, I_y) = \arg\min_{(m,n)\in small\ S} SAD(m,n) \tag{3}$$

$$SAD(m,n) = \sum_{i}^{i+16}\sum_{j}^{j+16} |P(i,j) - R(i+B_x+m, j+B_x+n)| \tag{4}$$

정재 과정 시 (-1,1)보다 큰 탐색 영역에서는 비슷한 PSNR을 가지므로 탐색 영역을 (-1,1)로 정한다[7].

그림 3은 1.152 Mbits/s로 부호화 되어있는 비디오 스트림을 576 kbits/s로 트랜스 부호화한 결과이다. 움직임 추정 방법으로 전 탐색 움직임 추정, 기저 움직임 벡터를 그대로 재 사용하는 방법, 그리고 추출한 움직임 벡터를 탐색 영역 (-1,1)에서 정재 과정을 수행하는 방법을 사용하였다. 실험 결과 기저 움직임 벡터를 정재 한다면 전 탐색 움직임 추정보다 탐색 영역이 작더라도 그와 비슷한 PSNR을 얻을 수 있음을 알 수 있다.

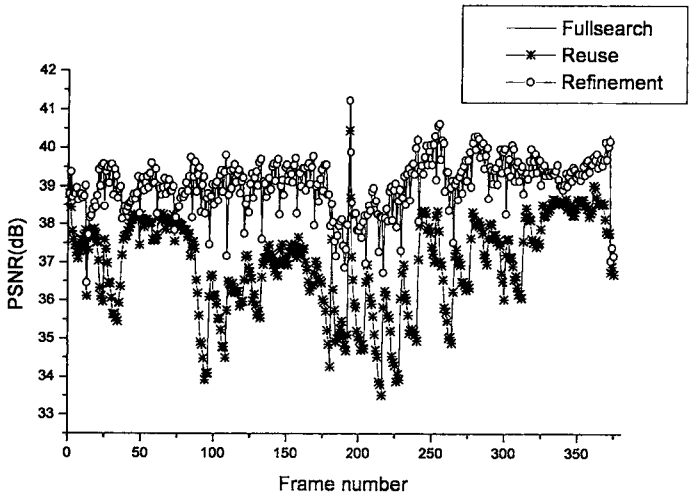


그림 3. 움직임 벡터 정재의 결과(CIF 포맷의 "Susie" 영상, 375 프레임).

Fig. 3. Performance of motion vector refinement("Susie" of CIF format, 375 frames).

III. 제안한 중간 정보를 이용한 적응적 움직임 벡터 정재

기저 움직임 벡터를 정재 하면 전 탐색 추정을 통해서 구한 최적의 움직임 벡터와 거의 비슷한 벡터를 찾아낼 수 있다. 그러나 모든 매크로블록을 정재 한다면 이는 비효율적이다. 왜냐하면 매크로블록의 특성을 고려하지 않고 정재 과정을 수행했기 때문이다. 움직임이 크지 않은 매크로블록은 정재 과정 없이 기저 움직임 벡터를 재 사용해도 화질의 열화가 거의

없는 경우가 많이 존재할 수 있기 때문이다. 본 논문에서는 그림 4와 같이 트랜스 부호의 움직임 추정시 중간 정보를 이용해서 매크로블록의 활동도(activity)를 계산해서 적응적으로 움직임 벡터를 정재하는 방법을 제안한다.

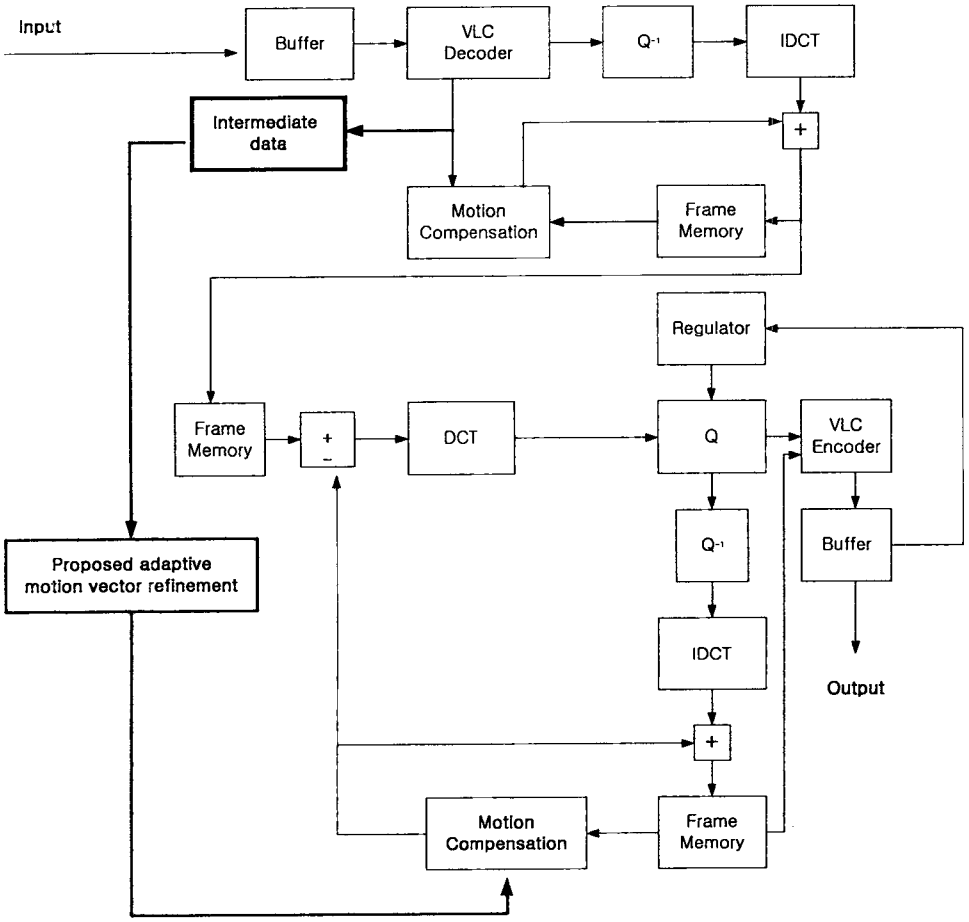


그림 4. 제안한 중간 정보를 이용한 트랜스 부호화기.

Fig. 4. Proposed transcoder using intermediate data.

트랜스 부호화에서 사용할 수 있는 중간 정보로는 움직임 벡터, 매크로블록 타입, DCT 계수, 양자화 파라미터 등이 있다. 이러한 것들 중에서 매크로블록의 활동도를 나타내는 요인으로는 움직임 벡터와 DCT 계수 등이 있다. 블록의 활동도는 공간 영역에서 블록내의 기울기나 에너지를 계산해서 특성화할 수 있다. 본 논문에서는 매크로블록의 DCT 계수 중에서 DC값을 뺀 나머지 AC계수들의 에너지와 기저 움직임 벡터의 거리를 사용해서 블록의 활동도를 계산한다. 이 방법은 역 DCT과정이 필요없으므로 효율적으로 블록의 활동도를 판단할

수 있다. 매크로블록의 AC 계수들의 에너지가 적정 수준보다 작은 값을 가진다면 그 매크로블록은 활동성이 적은 것이라고 가정하고 이러한 매크로블록들은 기저 움직임 벡터와 트랜스 부호화기에서 움직임 추정 후에 찾아낸 움직임 벡터는 거의 비슷하게 된다. 그러므로 움직임이 거의 없는 매크로블록들은 정재하지 않고 기저 움직임 벡터를 그대로 사용해도 영상의 화질에는 영향을 미치지 않게 된다.

제안한 방법에서는 트랜스 부호화 할 때 프레임 당 매크로블록의 AC 계수들의 평균 에너지와 기저움직임 벡터의 움직인 거리를 구하고 그 값의 1/4값을 판단 $threshold_1$로 정하고기저 움직임 벡터의 움직임 거리를 $threshold_2$로 정한다. 현재 움직임 벡터를 구하고자하는 매크로블록의 AC 계수들의 에너지가 $threshold_1$보다 작고 움직임 벡터의 거리가 $threshold_2$보다 작을 경우에는 기저 움직임 벡터를 그대로 사용한다. 여기까지는 중간 정보를 이용한 것이고 계산량을 더 줄이기 위해서 추가적인 조건으로 트랜스 부호화기에서 움직임 벡터를 구할 때 기저 움직임 벡터를 시작점으로 해서 블록간의 값의 차의 절대값의 합이 $threshold_3$보다 작다면 움직임 추정을 할 필요없이 기저 움직임 벡터를 그대로 사용하도록 한다. 그리고 위의 조건들을 만족하지 않을 경우에는 기저 움직임 벡터를 중심으로 해서 (-1,1)의 탐색 영역 내에서 움직임 벡터를 정재한다. 그림 5은 제안한 방법의 순서도를 나타낸다.

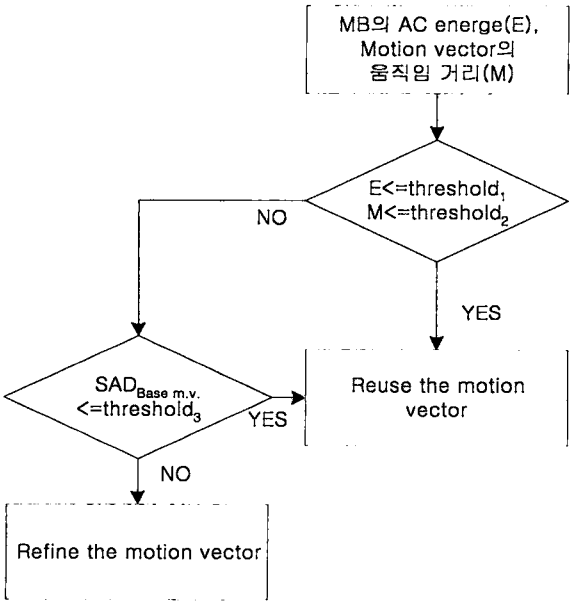


그림 5. 제안한 방법의 순서도.

Fig. 5. Flow chart of proposed method.

Ⅳ. 실험 결과 및 고찰

성능 평가를 위해서 제안된 방법과 전 탐색 움직임 추정 방법, 기저 움직임 벡터를 단순히 재 사용하는 방법을 비교하였다. 트랜스 부호화 시 부호화과정에서 매크로블록의 움직임 벡터를 구할 때 기저 움직임 벡터를 그대로 사용할 것인지 아니면 움직임 벡터 정재 과정을 수행할 것인지를 판단하는 임계값은 다음과 같이 정한다.

$$threshold_1 = \frac{\text{각 프레임의 } AC\ coefficients\ energy\text{의 평균}}{4},\ threshold_2 = 2,$$

$$threshold_3 = 300$$

그리고 정재 과정 시 탐색 영역은 (-1,1)으로 정한다.

실험에 사용된 시퀀스는 움직임이 거의 없는 영상①, 움직임이 조금있는 영상②, 움직임이 큰 영상③, 3 가지를 사용했다. 각 비디오 시퀀스의 비트율은 1.152 Mbits/sec.이고 총 프레임 수는 375 프레임이다. 각 시퀀스는 MPEG-1으로 부호화 되어있으며 GOP내에는 I 프레임과 P 프레임만 존재한다. 표 1은 각각의 실험에 사용된 비디오 스트림에 대한 조건을 나타내었다.

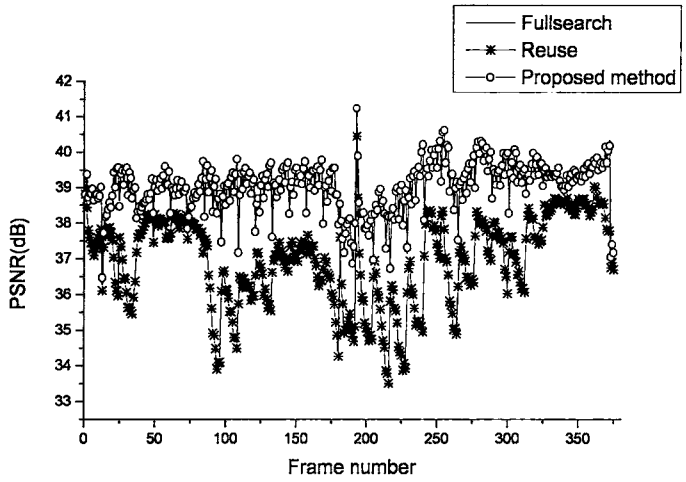
기본적으로 MPEG에서 움직임 추정시 전 탐색 움직임 추정을 수행하므로 제안된 방법과 전 탐색 움직임 추정 방법을 실험하고, 움직임 벡터 정재 과정의 영향을 알아보기 위해 기저 움직임 벡터를 재 사용하는 경우도 실험해서 그 결과를 서로 비교하여 제시하고자 한다. 실험 영상 ①, ②, ③에 대해 실험한 결과를 표 1에 나타내었다.

표 1. 제안된 방법에 의한 결과

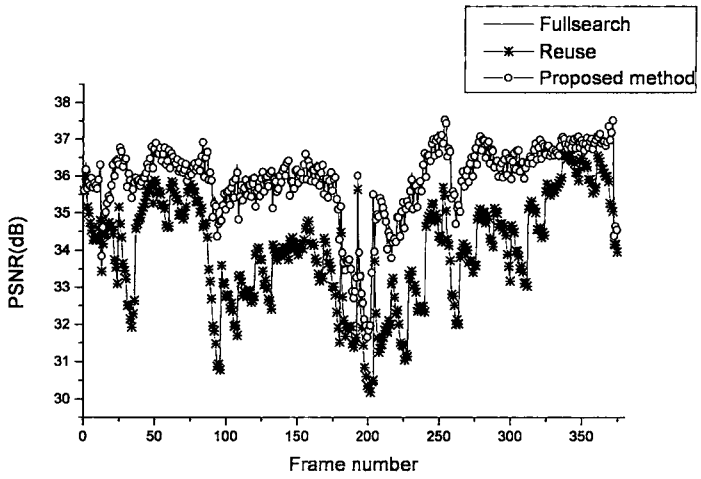
Table 1. Results of proposed method.

Test Sequence	Motion Estimation Method	number of skipped MB (총 MB 수 : 136224)	Outgoing bit-rate	
			576 kbps	288 kbps
Flower ①	Full-ME	x	26.17	24.95
	reuse	136224	25.03	24.05
	Proposed Method	64766	**26.08**	**24.90**
Susie ②	Full-ME	x	39.06	35.95
	reuse	136224	36.95	34.00
	Proposed Method	64045	**38.99**	**35.88**
Tennis ③	Full-ME	x	28.95	26.98
	reuse	136224	27.83	26.21
	Proposed Method	62537	**28.89**	**26.93**

그림 6의 (a)는 실험 비디오 시퀀스 중 Susie를 576 kbits/s로 트랜스 부호화한 결과이고 (b)는 288 kbits/s로 트랜스 부호화한 결과이다.



(a)



(b)

그림 6. 제안한 적응적 움직임 벡터 정재의 결과

(a) 576 kbps/s로 트랜스 부호화, (b) 288 kbps/s로 트랜스 부호화.

Fig. 6. Performance of adaptive motion vector refinement

(a) transcoding at 576 kbps/s, (b) transcoding at 288 kbps/s.

표 1에서 살펴보면 기저 움직임 벡터를 재 사용하는 경우가 총 매크로블록의 거의 절반 이상이어도 PSNR은 거의 떨어지지 않는 것을 볼 수 있다. 기저 움직임 벡터를 재 사용하는

것으로 인한 계산량의 감소 정도를 살펴보기 위해 프레임 당 계산량을 표 2에 나타내었다. N은 하나의 프레임에서 정재 과정을 거치는 매크로블록의 갯수이다.

표 2. 프레임 당 계산량의 복잡성.

Table 2. Computational complexity per frame.

Function	Complexity		
	Mults.	Adds	Total
Full-search ME(±15 pels, 738048a per 16×16 block)		738048×22×18	
Motion compensation(256a per 16×16 block)		101376	
DCT+Quant.(144m,464a per 8×8 block)	228096	734976	
total	228096	293103360	
Total count(Add = 1, Shift = 1, Mult. = 3)			293787640
proposed method(±1 pels, 6912a per 16×16 block)		6912×N	
Motion compensation(256a per 16×16 block)		101376	
DCT+Quant.(144m,464a per 8×8 block)	228096	734976	
total	228096	6912×N+836342	
Total count(Add = 1, Shift = 1, Mult. = 3)			6912×N+1520630

전 탐색 움직임 추정 방법과 제안된 방법의 전체 계산량은 식 (5)로 비교해 볼 수 있을 것이다.

전체 계산량(%)

= 전체 프레임에서 *refinement*하는 *MB*의갯수×탐색 영역에서 소요되는 계산량 (5)

각각의 실험 비디오들에 대해 제안된 방법에 의한 전체 계산량을 구해보면 아래와 표 3과 같다.

표 3. 실험 비디오 시퀀스에 대한 전체 계산량.

Table 3. Total computational complexity of test video sequence.

	비디오 시퀀스에 대한 전체 계산량		
	Flower	Susie	Tennis
Full-scale motion estimation searching range : (−15、15)	110,539,850,752	110,539,850,752	110,539,850,752
Proposed adaptive motion vector refinement searching range : (−1、1)	493,917,696	498,901,248	509,324,544

위에서 살펴본 결과를 종합해 보면 제안된 방법이 전 탐색 움직임 추정과 비교했을 때 0.05dB 정도의 영상의 열화를 가지면서 계산량에서 월등히 개선된 것을 알 수 있다.

V. 결 론

본 논문에서는 트랜스 부호화 시 움직임 벡터를 찾아낼 때 계산량을 줄이면서 정확하게 찾아낼 수 있는 방법을 제안하고 있다. 가장 기본적인 방법은 기저 움직임 벡터를 그대로 사용하는 것이지만 이 방법은 영상의 열화가 발생하기 때문에 바람직하지 못하다. 그래서 기저 움직임 벡터를 그대로 사용하지 않고 전 탐색 움직임 추정보다 탐색 영역이 훨씬 작은 범위에서 정재 과정을 수행하게 되면 전 탐색 움직임 추정과 비슷한 결과를 얻을 수 있다. 그런데 어떤 매크로블록은 정재 과정을 하지 않고 기저 움직임 벡터를 그대로 사용해도 화질의 열화가 거의 없는 경우가 분명히 존재할 것이다. 그러면 기저 움직임 벡터를 그대로 사용해도 되는 경우를 적절히 잘 판단할 수 있다면 계산량을 상당히 줄일 수 있다. 그 판단 기준으로 본 논문에서는 복호화 과정에서 뽑아낼 수 있는 중간 정보 중 매크로블록의 AC 계수들의 에너지와 움직임 벡터의 움직임 거리를 사용하도록 한다. 매크로블록의 AC 에너지가 작다는 것은 활동성이 적다는 것을 의미하므로 이런 매크로블록은 움직임 추정 시 기저 움직임 벡터를 그냥 사용해도 무방하다. 실험에서는 정재 과정을 수행할 것인지 하지 않을 것인지를 판단하기 위한 임계값을 정해서 매크로블록의 AC 에너지가 임계값을 넘지 않으면 정재 과정을 수행하지 않고 기저 움직임 벡터를 재 사용했다. 실험결과를 살펴보면 기저 움직임 벡터를 적응적으로 정재하여도 전 탐색 움직임 추정과 유사한 성능을 나타내는 것을 알 수 있다.

그리고 본 논문에서는 영상마다 똑같은 임계값을 적용하는 것이 아니라 각 프레임의 AC 계수들의 에너지의 평균을 이용하기 때문에 비디오 시퀀스의 특성을 고려한 적응적 움직임 벡터 정재 과정을 수행할 수 있다. 즉 움직임이 많은 비디오 시퀀스와 움직임이 적은 비디오 시퀀스를 비교했을 때 기저 움직임 벡터를 그대로 사용해도 되는 경우는 후자 쪽에 더 많이 나타나게 된다. 그러므로 AC 에너지를 임계로 선택하게 되면 움직임이 적은 비디오 시퀀스는 움직임 추정을 하지 않는 매크로블록을 많이 선택하게 되어도 영상의 열화는 작을 것이고, 움직임이 많은 비디오 시퀀스의 경우에는 더 적게 선택해서 최소한의 영상의 열화를 가지게 만들 수 있다. 비디오 시퀀스의 각 프레임마다 다른 임계값을 적용할 수 있으므로 좀더 효율적으로 기저 움직임 벡터를 적응적으로 정재할 수 있는 장점이 있다.

참고문헌

[1] M. Ghabari, "Two-layer coding of video signals for VBR networks", *IEEE j. Select. Areas Commun.*, vol. 7, pp. 771-781, June 1989.

[2] G. Keesman, R. Hellinghuizen, F. Hoeksema, G. Heidema, "Transcoding of MPEG bitstreams", *Signal Processing : Image Communication* 8, 1996, pp. 481-500

[3] M. Ghanbari, "Transcoding of single-layer MPEG video into lower rates", *IEE Proc.-Vis. Image Signal Process*, Vol. 144, No. 6, 1997, pp.377-383

[4]P. Assuncao and M. Ghabari, "Optimal transcoding of compressed video", *IEEE international Conference on Image Processing, ICIP' 97*, vol. 1, pp. 739-742, October, 1997.

[5] Jeongnam Youn, Ming-ting Sun, Chia-Wen Lin, "Motion Vector Refinement for High Performance Transcoding", *IEEE Transactions on Multimedia*, Vol. 1, No. 1, 1999, pp. 30-40

[6] N. Bjork and C. Christopoulos, "Transcoder Architecture for Video Coding", *IEEE Transaction Consumer Electron.*, Vol. 44, 1998, pp. 88-98

[7] Tamer Shanableh, M. Ghabari, "Heterogeneous Video Transcoding to Lower spatio-Temporal Reolutions and Different Encoding Formats", *IEEE Transactions on Multimedia*, Vol. 2, No. 2, June, 2000, pp. 101-110