

순방향과 역방향 움직임 추정을 이용한 효율적인 움직임 보상 외삽 기법

준회원 권혜경*, 종신회원 이창우*^o

Efficient Motion Compensated Extrapolation Technique Using Forward and Backward Motion Estimation

Hye-gyung Kwon* Associate Member, Chang-woo Lee*^o Lifelong Member

요약

움직임 보상 외삽 기법은 움직임 보상 보간 기법에 비해서 과거의 프레임만 사용하기 때문에 성능은 다소 떨어지지만, 전송 중에 손상된 프레임의 복원 및 프레임율 증가 뿐 아니라 분산 동영상 부호화 시스템의 부가 정보 생성에도 활용될 수 있다. 본 논문에서는 다양한 움직임 보상 외삽 기법의 성능을 평가하고 순방향과 역방향 움직임 추정을 함께 이용한 효율적인 움직임 보상 외삽 기법을 제안한다. 제안하는 기법은 순방향과 역방향 움직임 추정을 통해 두 개의 프레임을 생성하여 두 프레임의 화소값의 평균을 현재 프레임으로 한다. 모의 실험 결과 제안하는 기법이 기존의 기법에 비해서 블럭 현상이 감소하고 우수한 PSNR 성능을 보임을 확인하였다.

Key Words : Motion Compensated Extrapolation, Motion Compensated Interpolation, Motion Estimation, Frame Rate, Distributed Video Coding

ABSTRACT

Motion compensated extrapolation (MCE) techniques show inferior performance compared to motion compensated interpolation techniques, since only past frames are used in MCE. MCE techniques are used for the reconstruction of corrupted frames, the up-conversion of frame rates and the generation of side information in the distributed video coding system. In this paper, the performance of various MCE techniques are evaluated and an efficient MCE technique using the forward and backward motion estimation is proposed. In the proposed technique, the present frame is extrapolated by averaging two frames which are generated by forward and backward motion estimation respectively. It is shown that the proposed method produces better PSNR results and less blocking phenomena than conventional methods.

I. 서론

동영상 신호의 전송 중에 손상된 프레임의 복원 및 동영상 신호의 프레임율 증가(frame rate up conversion)에 움직임 보상 보간(MCI : motion compen-

sated interpolation) 혹은 움직임 보상 외삽(MCE : motion compensated extrapolation) 기법이 많이 사용된다^[2-10]. MCI 기법은 MCE 기법에 비해서 성능이 우수한 편이지만 복원하려는 프레임보다 미래의 프레임을 사용하기 때문에 지연(delay)이 발생하고

※ 본 연구는 2009년도 가톨릭대학교 교비 연구비 지원 및 2010년도 정부의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임 (No.2010-0028178)

* 가톨릭대학교 정보통신전자공학과 (khg1207@catholic.ac.kr, changwoo@catholic.ac.kr) (°:교신저자)
논문번호 : KISC2010-08-410, 접수일자 : 2010년 8월 23일, 최종논문접수일자 : 2011년 2월 28일

장면 전환 등의 상황에 취약한 단점이 있다. MCE 기법은 미래의 프레임을 사용하지 않는 장점이 있으나 MCI 기법에 비해서 성능이 떨어지는 편이므로 MCE 기법의 성능 향상을 위한 연구가 필요하다. 특히 최근 연구가 활발히 진행되고 있는 분산 동영상 부호화(DVC : distributed video coding) 시스템의 복호기에서 부가 정보(SI : side information) 생성을 위해서 MCI 혹은 MCE 기법이 사용될 수 있다¹⁾.

본 논문에서는 기존의 MCE 기법들의 성능을 분석하고 효율적인 MCE 기법을 제안한다. 비교적 간단한 MCE 기법인 이전 프레임의 같은 위치의 움직임 벡터를 그대로 사용하는 방법, 선형적으로 움직임을 추정하는 방법 및 화소별 가중치를 계산하여 움직임 벡터를 추정하는 방법의 성능을 분석한다. 가장 기본적인 방법인 이전 프레임의 같은 위치의 움직임 벡터를 그대로 사용하는 기법은 움직임을 고려하지 않기 때문에 움직임이 큰 동영상에서는 정확한 움직임 벡터를 결정하지 못하는 단점이 있으며, 선형적으로 움직임 벡터를 추정하는 기법은 움직임이 적은 영상에 대한 추정이 어려운 단점이 있다. 이러한 단점을 보완하고 MCE 기법의 성능을 향상시키기 위해서 순방향(forward)과 역방향(backward) 추정을 이용한 효율적인 MCE 기법을 제안한다. 제안하는 MCE 기법은 순방향과 역방향 추정을 통해 두 개의 프레임을 생성하여 두 프레임의 휘도값의 평균을 현재 프레임으로 추정하는 기법으로 기존의 단방향 추정 기법에 비해서 움직임을 더 정교하게 추정할 수 있다. 모의 실험을 통해서 제안하는 기법이 기존의 기법에 비해서 우수한 성능을 보임을 입증한다. 본 논문의 2장에서는 기존에 제안된 다양한 MCE 기법에 대해 설명하고, 3장에서 효율적인 MCE 기법을 제안한다. 4장에서는 각 MCE 기법의 성능을 분석하고 5장에서는 결론을 맺는다.

II. 기존의 MCE 기법

기존의 MCE 기법에서는 이전 프레임의 같은 위치의 움직임 벡터를 그대로 사용하거나 선형적으로 움직임 벡터를 추정하는 방법을 많이 사용하였다. 이전 프레임의 움직임 벡터를 그대로 사용하는 방식은 그림 1에 도시한 것과 같이 과거 두 프레임에서 추정된 움직임 벡터 \vec{v} 를 그대로 사용하여 MCE를 수행한다. 이 방식은 비교적 간단하지만 특

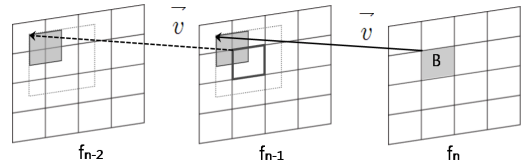


그림 1. 이전 프레임의 움직임 벡터를 그대로 사용한 MCE 기법

정 위치의 움직임이 다음 영상의 같은 위치에 그대로 적용되므로 비교적 움직임이 큰 동영상에 대해서는 잘 동작하지 않는 단점이 있다. 그림 2는 물체의 동선에 따라 선형적으로 추정한 방식으로 현재 프레임에서 현재 블록을 중심으로 이전 프레임은 \vec{v} 로 더 이전 프레임은 $2\vec{v}$ 의 움직임 벡터를 이용하여 이전 프레임의 블록간의 SAD(sum of absolute difference)가 적은 움직임 벡터를 사용한다. 이 방식은 빠르게 움직이는 물체를 추정하는데 용이하다. 그러나 이전 프레임들 간의 SAD를 비교할 때 과거 프레임(f_{n-2})에서 두 화소씩 이동하여 추정하므로 움직임이 적은 영상의 추정이 어렵다.

기존의 두 가지 MCE 방식 이외에 Yan 등이 제안한 HMVE(hybrid motion vector estimation) 방식에서는 과거 두 프레임의 모든 블록에 대해 움직임 벡터를 추정한 후 그림 3과 같이 각 블록을 반대방향의 움직임 벡터를 이용하여 현재 프레임에 투영(projection)하면 블록 B에 hole과 overlap이 발생한다⁴⁾. 이때 추정하고자 하는 블록 B는 overlap된 화소 수에 따라 가중치를 계산하여 움직임 벡터를 결정한다. 그림 4와 같이 4x4 블록에서 2개 이상 overlap된 집합인 part A는 {1, 2, 3, 5}, overlap된 집합 중 나머지 part B는 {4, 6, 7, 9, 13}, 그리고 hole 부분의 화소들의 집합인 part C는 {8, 10, 11, 12, 14, 15, 16}이라 가정한다. 식 (1)을 이용하여 overlap된 정도에 따라 가중치를 계산하고, 식 (2)와 (3)을 이용하여 움직임 벡터 MV_m 과 MV_a 를 결정한다.

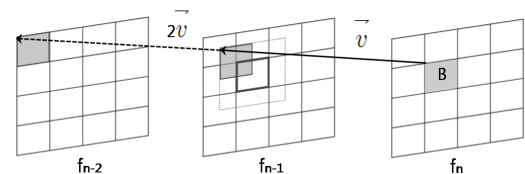


그림 2. 선형적으로 움직임을 추정한 MCE 기법

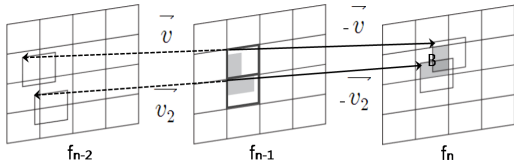


그림 3. HMVE를 위한 MCE 기법

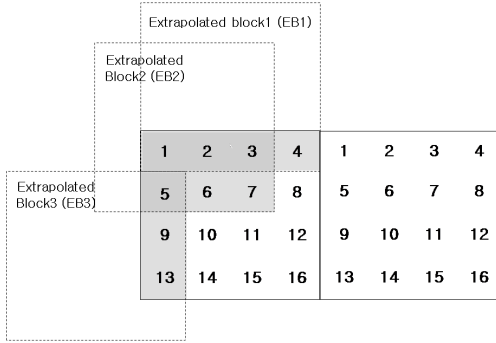


그림 4. 4×4블록에 대한 현재 블록 overlap의 보기

$$w^{i,j} = \sum_{p \in B^i} f_j(p) \quad , \quad i, j = 1, 2, \dots, M \quad (1)$$

$$f_j(p) = \begin{cases} 1, & p \in EB^j \\ 0, & p \notin EB^j \end{cases}$$

$$MV_m(B^i) = MV(EB^{j^*}) \quad (2)$$

where $j^* = \operatorname{argmax}\{w^{i,j}\}, j = 1, 2, \dots, M$

$$MV_a(B^i) = \frac{\sum_{j=1}^M MV(EB^j) w^{i,j}}{\sum_{j=1}^M w^{i,j}} \quad (3)$$

이때 EB^j 는 이전 프레임(f_{n-1})의 j 번째 블록에서 현재 프레임으로 투영한 블록이고, B^i 는 현재 프레임의 i 번째 블록이며, $MV(EB^j)$ 는 EB^j 의 움직임 벡터를 의미한다. M 은 한 프레임내의 전체 블록의 개수이고, $f_j(p)$ 는 이전 프레임으로부터 현재 프레임에 투영된 EB^j 가 현재 프레임 B^i 의 화소 p 에 overlap되면 1의 값을 갖는다. $w^{i,j}$ 는 $f_j(p)$ 의 합으로 j 번째 블록 EB^j 를 이용한 현재 프레임의 i 번째 블록의 가중치이다. MV_m 은 가중치값이 큰 j^* 번째 블록의 움직임 벡터인 $MV(EB^{j^*})$ 가 되고, MV_a 는 가중치를 이용하여 구한 $MV(EB^j)$ 의 평균값이 된다. 1개 이상의 MV 가 존재하는 Part A와 part B는 MV_m, MV_a 를 포함한 후보 움직임

벡터 중에서 중간값의 움직임 벡터를 선택하고, part C는 MV_m 과 MV_a 의 평균 벡터를 선택한다. 만일 블록 전체가 hole인 경우는 이전 프레임의 같은 위치의 블록의 움직임 벡터를 그대로 이용한다. 이 방식은 다수의 가중치 계산과 움직임 벡터 계산으로 인하여 계산량이 증가하는 단점이 있다.

III. 제안하는 MCE 기법

본 논문에서는 순방향과 역방향 움직임 추정을 이용한 효율적인 MCE 기법을 제안한다. 즉, 순방향과 역방향 추정을 통해 두 개의 프레임(f_n)을 생성하여 두 프레임의 휘도값의 평균을 현재 프레임(f_n)으로 생성한다. 두 개의 프레임을 생성하여 평균을 구하기 때문에 기존의 단방향 기법보다 계산량이 늘어나지만 움직임을 좀 더 정교하게 추정할 수 있고 블록 현상이 줄어드는 효과를 얻을 수 있다. 특히 순방향 및 역방향 추정에 모두 1/2 화소 단위로 움직임 추정을 하면 단방향 추정에 1/2 화소 단위로 움직임 추정을 하는 것 보다 좀 더 향상된 성능을 얻을 수 있다. 이때 순방향 추정은 기존 MCE 기법 중 이전 프레임의 움직임 벡터를 그대로 사용하는 방식을 이용하고, 역방향 추정에서는 두 가지 추정 기법중 하나를 이용하는데, 그림 6에 두 가지 역방향 추정 기법을 도시하였다.

첫 번째 역방향 추정 기법은 특정 위치의 움직임 벡터를 다음 프레임의 해당 위치의 움직임 벡터로 이용하는 방식(알고리즘 1)이고, 두 번째는 이전 프레임에서 구한 움직임 벡터를 선형적으로 연장하여 두 배의 움직임 벡터로 현재 프레임을 생성하는 방식(알고리즘 2)이다. 그림 5에서 알 수 있는 것과 같이 역방향 추정 기법의 두 가지 알고리즘은 이전 프레임들 간에서 추정한 움직임 벡터는 같지만 결과적으로는 서로 다른 현재 프레임(f_n)을 생성하므로 동영상 특성에 따라 PSNR 성능의 차이가 있다. 알고리즘 1은 움직이는 물체의 동선을 고려하지 않고 특정 위치의 움직임 벡터를 그대로 사용하는 비교적 간단한 방법으로 움직임이 적은 동영상 추정에 용이하고, 알고리즘 2는 움직이는 물체의 동선을 고려하여 선형적으로 추정한 방법으로 움직임이 큰 동영상 추정에 용이하다. 이와 같이 생성된 역방향 추정 알고리즘의 현재 프레임(f_n)은 각각의 화소마다 두 개 이상의 블록이 겹치는 overlap이나 어떠한 블록도 겹치지 않는 hole이 발생한다. overlap이 발

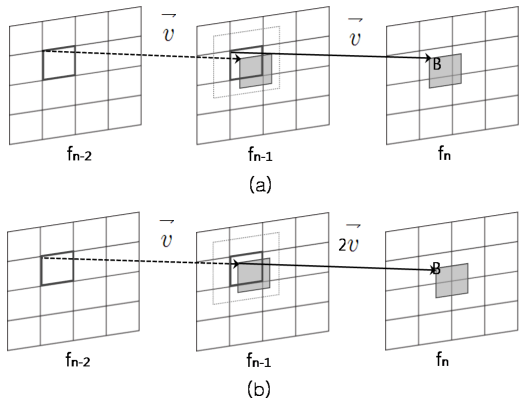


그림 5. 역방향 추정에 사용되는 두 가지 기법 (a)이전 프레임의 움직임 벡터를 그대로 사용하여 추정하는 기법(알고리즘 1), (b)선형적으로 추정하는 기법(알고리즘 2)

생한 화소는 휘도값의 평균으로 구한다. 다음 단계로 그림 6과 같이 순방향 추정에서 구한 프레임과 역방향 추정에서 구한 프레임의 휘도값의 평균을 구하는데, 역방향 추정에서 생긴 hole은 순방향 추정에서 구한 값이 채워지게 된다.

기존의 MCE 기법은 현재 프레임을 중심으로 이전 프레임으로 향하는 순방향 추정만 이루어지는데, 제안하는 양방향 추정 기법에서는 역방향 추정도 수행하여 여러 가지 움직임 벡터를 이용함으로써, 움직임이 크거나 작은 영상에 대해 각각 취약한 기존의 MCE 기법의 단점을 보완할 수 있고 두 프레임의 평균값이 현재 프레임이 되므로 블록현상이 줄어들드는 장점이 있다.

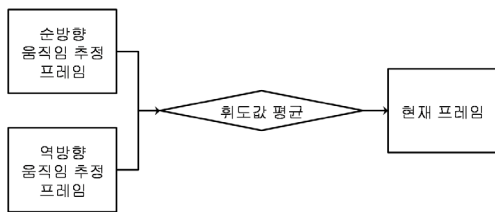


그림 6. 제안하는 순방향 및 역방향 추정을 이용한 현재 프레임 생성 기법

IV. 모의 실험

다양한 MCE 기법에 대한 성능 평가를 위해서 다음 식으로 계산되는 원영상과 추정되는 영상간의 PSNR 값을 측정하였다.

$$PSNR = 10 \cdot \log(255^2 / (\sum_{m=0}^{M-1} \sum_{n=0}^{N-1} (u(m,n) - u'(m,n))^2) / MN) (dB) \quad (4)$$

식 (4)의 $u(m,n)$ 과 $u'(m,n)$ 은 원 영상과 추정된 영상의 화소값을 나타낸다. 각 기법에 대한 성능 평가를 위하여 그림 7에서 제시하는 4가지 종류의 원본 영상과 영상 부호화가 성능에 미치는 영향을 평가하기 위하여 영상마다 H.264 intra frame 부호화 기법에서 양자화 파라미터(QP)를 3가지로 변화시키면서 영상을 압축 후 복원한 영상을 사용하였다. 이중 Foreman 영상은 표준 동영상으로 가장 많이 사용되며, Hall monitor 영상은 비교적 단순로운 영상이고, Stefan 영상과 Mobile 영상은 움직임이 크고 복잡한 부분을 많이 포함하는 영상이다. 그림 8, 9에 Foreman과 Mobile의 원본 영상을 이용하여 4 가지 MCE 기법의 성능을 비교한 그래프와 그림 10, 11에 Foreman과 Mobile 영상을 H.264 intra frame 부호화 기법에서 QP 값이 25일 때 영상 압축 후 복원된 영상을 이용하여 성능을 비교한 그래프를 제시하였다. 이때 복원 영상의 PSNR 결과는 과거의 복원된 영상으로 현재 프레임을 생성하여 원본 영상과의 성능을 평가한 것이며, 그래프에 도시한 제안하는 기법은 역방향 추정 시 알고리즘 2를 적용한 것이다.

제시하는 그래프에서 알 수 있는 것과 같이 기존의 세 가지 방식 중 가장 낮은 성능을 보이는 MCE 기법은 선형적으로 추정하는 방식이다. 이는 더 이전 프레임(f_{n-2})은 현재 프레임(f_n)을 중심으로 움직임 벡터 $2\vec{v}$ 를 이용하여 움직임을 추정하기 때문에, 두 화소씩 이동하며 추정하므로 잘못된 움직임 벡터가 선택되어 오류가 발생한 것이다. 나머지 기존 추정 기법인 이전 프레임의 움직임 벡터를 사용하는 MCE 기법과 Yan 등이 제안한 화소별로

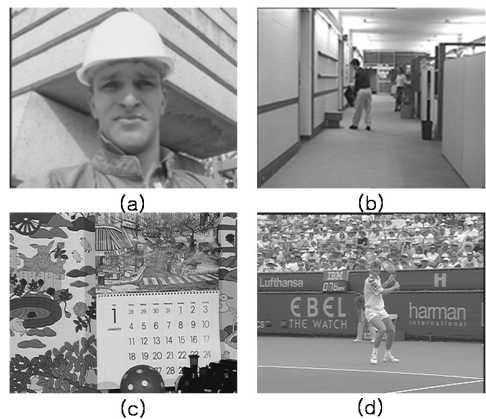


그림 7. 성능 평가에 사용된 영상 (a)Foreman(QCIF), (b)Hall monitor(QCIF), (c)Mobile(CIF), (d)Stefan(CIF)

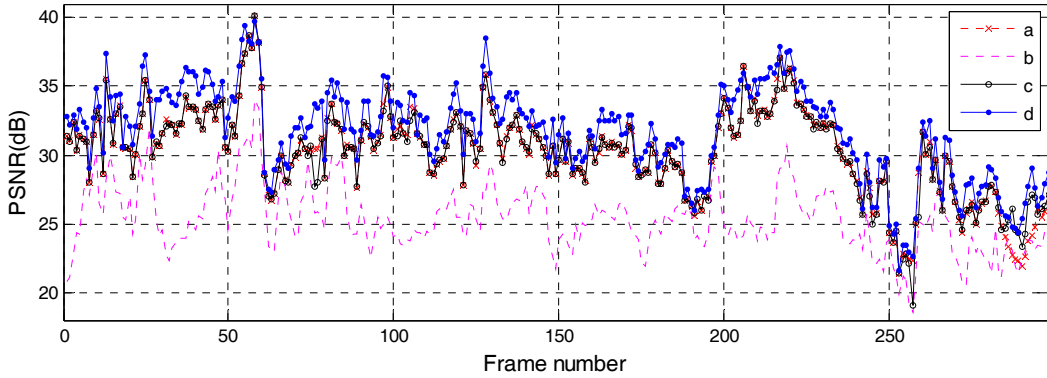


그림 8. Foreman 원본영상에 대한 각 MCE 기법의 PSNR성능 (a 이전 프레임의 움직임 벡터를 그대로 사용하여 추정, b 선형적으로 움직임을 추정, c Yan이 제안한 화소별로 추정, d 본 논문에서 제안한 순방향과 역방향 추정 알고리즘2)

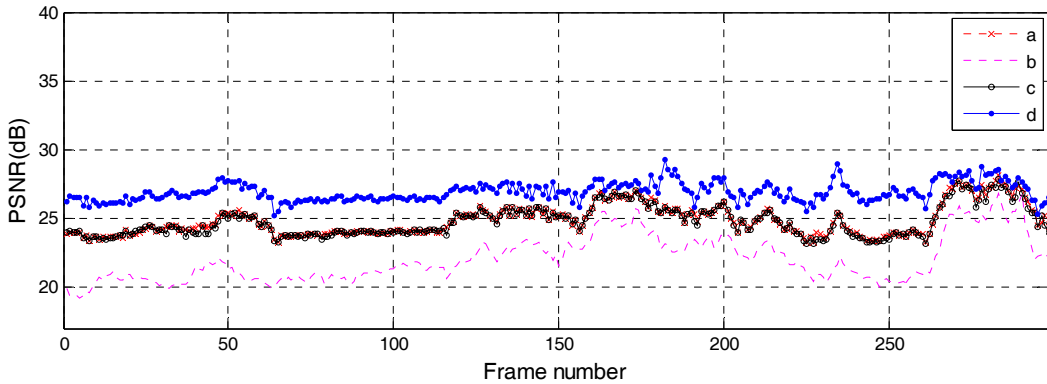


그림 9. Mobile 원본영상에 대한 각 MCE 기법의 PSNR성능 (a 이전 프레임의 움직임 벡터를 그대로 사용하여 추정, b 선형적으로 움직임을 추정, c Yan이 제안한 화소별로 추정, d 본 논문에서 제안한 순방향과 역방향 추정 알고리즘2)

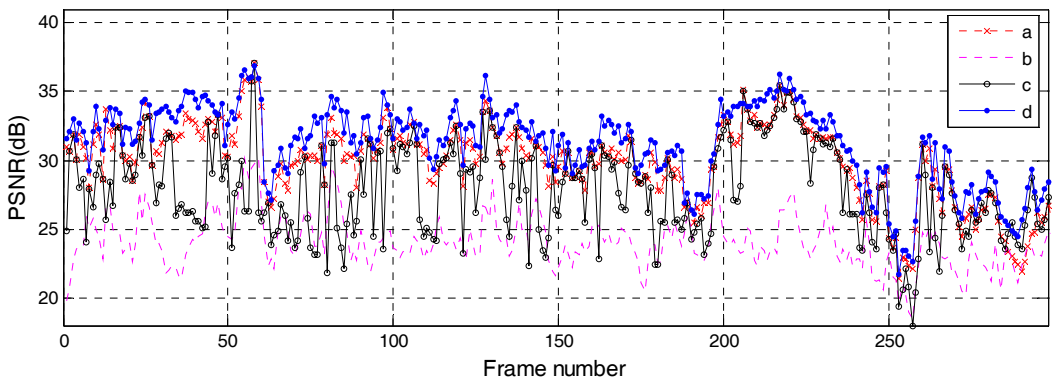


그림 10. Foreman 영상을 H.264 intra frame 압축 기법을 이용하여 압축 후(QP 25) 복원한 영상에 대한 각 MCE 기법의 PSNR성능 (a 이전 프레임의 움직임 벡터를 그대로 사용하여 추정, b 선형적으로 움직임을 추정, c Yan이 제안한 화소별로 추정, d 본 논문에서 제안한 순방향과 역방향 추정 알고리즘2)

추정하는 HMVE는 서로 비슷한 성능을 보인다.

이 두 기법은 성능이 비슷하지만 화소별로 가장

치를 계산하여 움직임 벡터를 선택하는 Yan 등이

제안한 HMVE 기법이 복잡도가 훨씬 더 높은 단점

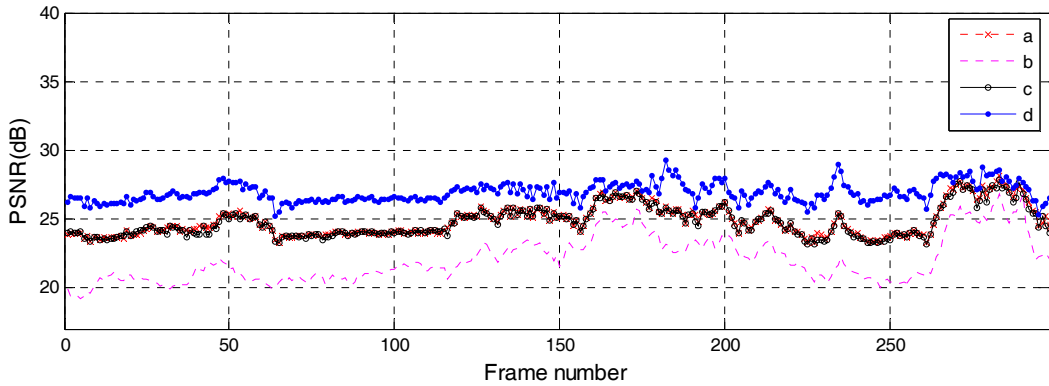


그림 11. Mobile 영상을 H.264 intra frame 압축 기법을 이용하여 압축 후(QP 25) 복원한 영상에 대한 각 MCE 기법의 PSNR 성능 (a 이전 프레임의 움직임 벡터를 그대로 사용하여 추정, b 선형적으로 움직임을 추정, c Yan이 제안한 화소별로 추정, d 본 논문에서 제안한 순방향과 역방향 추정 알고리즘2)

이 있다. 기존의 MCE 기법 중 가장 복잡도가 높은 HMVE 기법은 블록의 크기가 b 이고, 검색범위가 a 인 경우, 블록 당 기본 탐색 연산량인 $b \times b \times (a/2 + 1) \times (a/2 + 1)$ 에 대략 (9×3) 배의 연산을 수행한다. 여기서 9배는 현재블록이 중앙에 위치하여 자신을 포함한 9개의 주변 블록으로부터 overlap이 발생하는지 판별하는데 필요한 연산량이고, 3배는 part A, B, C의 존재 여부에 따라 움직임 벡터를 결정하기 위해서 필요한 연산량이다. 나머지 기존의 두 MCE 기법은 블록 당 기본 연산량인 $b \times b \times (a/2 + 1) \times (a/2 + 1)$

을 수행하고, 제안하는 기법인 순방향과 역방향 움직임 추정 기법은 기본 연산량의 2 배의 연산을 수행한다. 제안하는 기법이 양방향 추정 기법이므로 기본 연산을 수행하는 기존 MCE 기법의 2 배의 연산을 수행하지만 그림 8~11에서 알 수 있듯이 기존의 기법들과 비교하여 전체 프레임에 대한 PSNR 측정 결과에서 좋은 성능을 보인다.

그림 12~14에 원본영상을 이용하여 각 MCE 기법으로 추정한 영상을 도시하였다. 기존의 기법들을 이용한 결과 영상을 비교하면 선형적으로 추정한

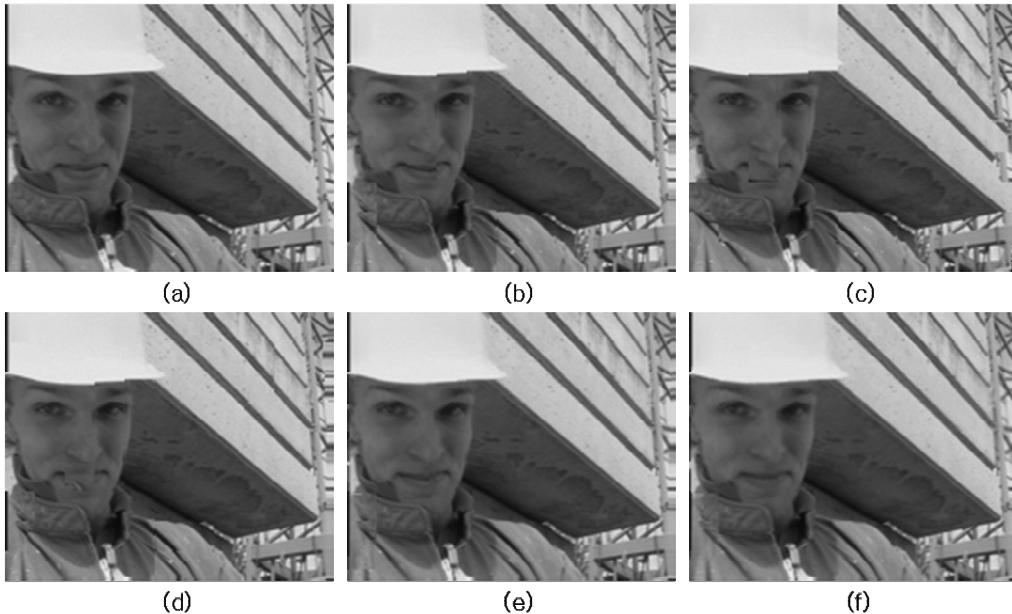


그림 12. 각 MCE 기법에 대한 결과 영상 비교(Foreman 277번째 영상) (a)원 영상 (b)이전 프레임의 움직임 벡터를 그대로 사용하여 추정한 영상 (c)선형적으로 움직임을 추정한 영상 (d)HMVE 기법으로 추정한 영상 (e)제안하는 양방향 추정 영상(역방향 알고리즘 1을 사용) (f)제안하는 양방향 추정 영상(역방향 알고리즘 2를 사용)

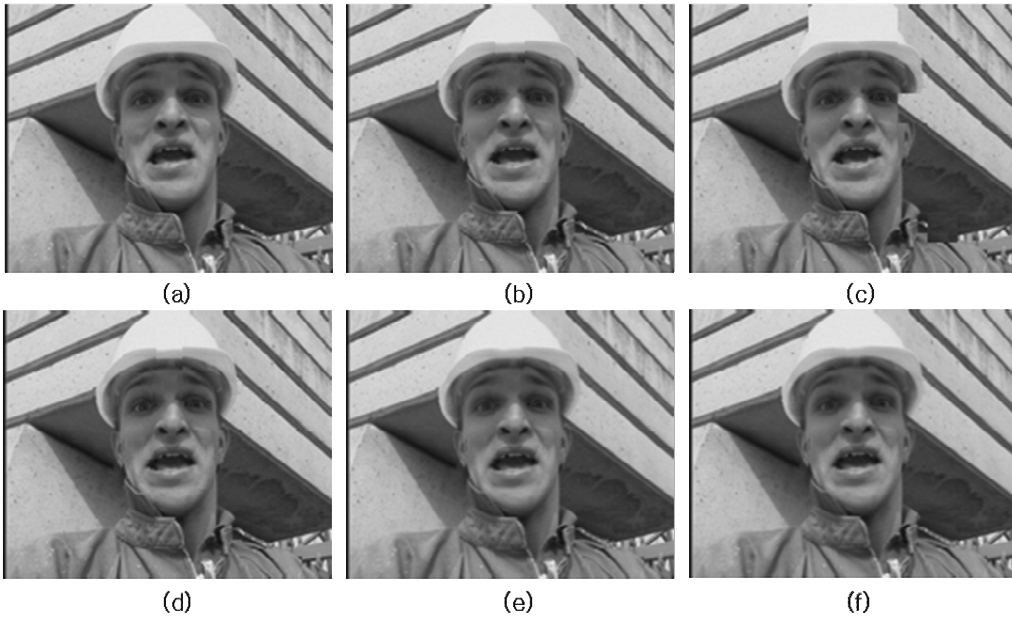


그림 13. 각 MCE 기법에 대한 결과 영상 비교(Foreman 67번째 영상) (a)원 영상 (b)이전 프레임의 움직임 벡터를 그대로 사용하여 추정한 영상 (c)선형적으로 움직임을 추정한 영상 (d)HMVE 기법으로 추정한 영상 (e)제한하는 양방향 추정 영상(역방향 알고리즘 1을 사용) (f)제한하는 양방향 추정 영상(역방향 알고리즘 2를 사용)

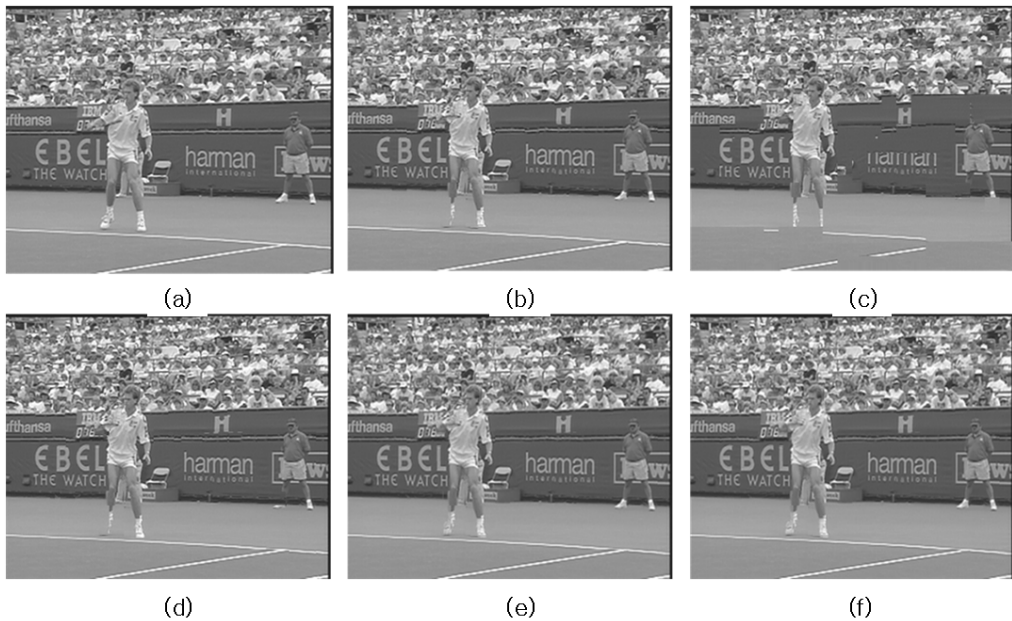


그림 14. 각 MCE 기법에 대한 결과 영상 비교(Stefan 38번째 영상) (a)원 영상 (b)이전 프레임의 움직임 벡터를 그대로 사용하여 추정한 영상 (c)선형적으로 움직임을 추정한 영상 (d)HMVE 기법으로 추정한 영상 (e)제한하는 양방향 추정 영상(역방향 알고리즘 1을 사용) (f)제한하는 양방향 추정 영상(역방향 알고리즘 2를 사용)

결과 영상인 그림 12의 (c)를 보면 사람의 입, 모자 그리고 왼쪽 옷깃에 심한 왜곡이 나타나고, 기존 기법들 중에서 가장 좋은 결과를 보이는 그림 12(b)

의 이전 프레임의 움직임 벡터를 그대로 사용하여 추정한 경우가 가장 원 영상을 잘 추정해낸다. 그림 12 (e), (f)의 사람의 입과 모자, 옷깃의 블록현상이

거의 줄어든 두 영상은 제안하는 기법을 사용할 결과 영상인데, 그림 12(b)와 비교하면 기존의 기법보다 좋은 성능을 보임을 알 수 있다. 그림 13의 기존 MCE 기법의 결과 영상을 보면 모자 가운데 부분에 블록 현상이 나타나지만 제안하는 기법의 결과 영상인 그림 13 (e), (f)에서는 나타나지 않는다. 그림 14는 움직임이 가장 큰 Stefan 영상에 대한 결과인데 역시 제안하는 기법의 결과 영상에서 알 수 있는 것과 같이 테니스 코트의 선과 선수의 왼쪽 발을 잘 추정해낸다. 표 1에 원본영상에 대한 PSNR 결과 평균을 제시하였는데 제안하는 기법은 PSNR 수치 비교에서도 기존의 방식에 비해서 좋은 성능을 보인다. 기존의 추정기법 중 비교적 높은 성능을 보이는 이전 프레임의 움직임 벡터를 그대로 사용하여 추정한 영상과 제안하는 기법에서 역방향으로 선형적으로 추정한 기법을 사용한 경우(알고리즘 2)의 결과 영상을 비교할 때, QCIF급 Foreman 영상에서 0.41dB, CIF급 Mobile 영상에서 0.31dB의 성능향상을 보였다. 그리고 기존 기법과 제안하는 기법 모두에 1/2 화소 단위로 움직임 추정을 수행한 경우 Foreman 영상에서 0.58dB, Mobile 영상에서 0.46dB의 성능향상을 보였고, 제안하는 기법에만 1/2 화소 단위로 움직임 추정을 한 경우

Foreman 영상에서 1.43dB, Mobile 영상에서 2.17dB의 성능향상을 보였다. 이때 표 1에 제시한 것과 같이 양방향 추정에서 사용한 두 가지 역방향 추정 알고리즘 중 선형적으로 움직임을 추정하는 방식(알고리즘 2)이 근소한 차이로 더 좋은 성능을 보인다. 표 2~4에 제시한 복원영상에 대한 PSNR 결과에서도 제안하는 기법이 기존의 기법보다 좋은 성능을 보인다. QP 값은 각각 25, 32, 39이며, 값이 커질수록 압축 후 복원 과정에서 양자화 손실이 많이 발생하여 복원 영상의 PSNR 결과가 원본 영상의 결과보다 낮다. 복원영상에 대해서 기존의 추정 기법 중 높은 성능을 보이는 기법과 제안하는 기법(알고리즘 2)의 결과 영상을 비교할 때, QP값이 25인 Foreman 영상에서 0.39dB, CIF급 Mobile 영상에서 0.21dB의 성능향상을 보였다. 그리고 기존 기법과 제안하는 기법 모두에 1/2 화소 단위로 움직임 추정을 수행한 경우 QP값이 25인 Foreman 영상에서 0.43dB, Mobile 영상에서 0.34dB의 성능향상을 보였고, 제안하는 기법에만 1/2 화소 단위로 움직임 추정을 한 경우 QP값이 25인 Foreman 영상에서 1.27dB, Mobile 영상에서 2.1dB의 성능향상을 보였다.

표 1. 원본영상에 대한 각 MCE 기법의 평균 PSNR 결과 (dB)

| 영상 (frame수) | 이전 프레임의 움직임 벡터를 그대로 이용 | 이전 프레임의 움직임 벡터를 그대로 이용 (1/2 화소 보간) | 선형 추정 | HMVE | 제안하는 알고리즘1 | 제안하는 알고리즘1 (1/2 화소 보간) | 제안하는 알고리즘2 | 제안하는 알고리즘2 (1/2 화소 보간) |
|-------------------|------------------------|------------------------------------|-------|-------|------------|------------------------|------------|------------------------|
| Foreman(400) | 30.52 | 31.37 | 26.11 | 29.35 | 30.92 | 31.91 | 30.93 | 31.95 |
| Hall monitor(300) | 36.81 | 36.92 | 35.45 | 36.79 | 36.87 | 37.14 | 36.88 | 37.18 |
| Mobile(300) | 24.86 | 26.57 | 22.05 | 24.77 | 25.15 | 26.89 | 25.17 | 27.03 |
| Stefan(90) | 24.46 | 25.25 | 22.79 | 24.5 | 24.81 | 25.66 | 24.87 | 25.89 |

표 2. H.264 intra frame 압축 기법을 이용하여 압축 후(QP 25) 복원한 영상에 대한 각 MCE 기법의 평균 PSNR 결과 (dB)

| 영상 (frame수) | 이전 프레임의 움직임 벡터를 그대로 이용 | 이전 프레임의 움직임 벡터를 그대로 이용 (1/2 화소 보간) | 선형 추정 | HMVE | 제안하는 알고리즘1 | 제안하는 알고리즘1 (1/2 화소 보간) | 제안하는 알고리즘2 | 제안하는 알고리즘2 (1/2 화소 보간) |
|--------------------|------------------------|------------------------------------|-------|-------|------------|------------------------|------------|------------------------|
| Foreman(400) | 30.07 | 30.91 | 24.73 | 28.57 | 30.46 | 31.32 | 30.46 | 31.34 |
| Hall monitor (300) | 35.63 | 35.86 | 30.81 | 35.39 | 35.69 | 35.92 | 35.7 | 35.95 |
| Mobile(300) | 24.74 | 26.5 | 21.47 | 24.63 | 24.94 | 26.74 | 24.96 | 26.84 |
| Stefan(90) | 24.32 | 25.29 | 22.23 | 24.35 | 24.74 | 25.58 | 24.79 | 25.8 |

표 3. H.264 intra frame 압축 기법을 이용하여 압축 후(QP 32) 복원한 영상에 대한 각 MCE 기법의 평균 PSNR 결과 (dB)

| 영상 (frame수) | 이전 프레임의 움직임 벡터를 그대로 이용 | 이전 프레임의 움직임 벡터를 그대로 이용 (1/2 화소 보간) | 선형 추정 | HMVE | 제안하는 알고리즘1 | 제안하는 알고리즘1 (1/2 화소 보간) | 제안하는 알고리즘2 | 제안하는 알고리즘2 (1/2 화소 보간) |
|-------------------|------------------------|------------------------------------|-------|-------|------------|------------------------|------------|------------------------|
| Foreman(400) | 28.98 | 29.37 | 23.04 | 26.40 | 29.34 | 29.89 | 29.31 | 29.83 |
| Hall monitor(300) | 33.20 | 33.22 | 28.08 | 33.07 | 33.26 | 33.41 | 33.25 | 33.40 |
| Mobile(300) | 24.31 | 25.70 | 20.65 | 24.11 | 24.51 | 26.10 | 24.52 | 26.15 |
| Stefan(90) | 24.09 | 24.85 | 21.80 | 24.03 | 24.49 | 25.26 | 24.53 | 25.45 |

표 4. H.264 intra frame 압축 기법을 이용하여 압축 후(QP 39) 복원한 영상에 대한 각 MCE 기법의 평균 PSNR 결과 (dB)

| 영상 (frame수) | 이전 프레임의 움직임 벡터를 그대로 이용 | 이전 프레임의 움직임 벡터를 그대로 이용 (1/2 화소 보간) | 선형 추정 | HMVE | 제안하는 알고리즘1 | 제안하는 알고리즘1 (1/2 화소 보간) | 제안하는 알고리즘2 | 제안하는 알고리즘2 (1/2 화소 보간) |
|-------------------|------------------------|------------------------------------|-------|-------|------------|------------------------|------------|------------------------|
| Foreman(400) | 26.82 | 26.95 | 21.55 | 23.33 | 27.10 | 27.30 | 27.05 | 27.23 |
| Hall monitor(300) | 29.02 | 29.02 | 25.43 | 28.28 | 29.06 | 29.10 | 29.04 | 29.08 |
| Mobile(300) | 22.91 | 23.80 | 19.06 | 22.61 | 23.09 | 24.07 | 23.10 | 24.09 |
| Stefan(90) | 23.27 | 23.89 | 20.99 | 23.07 | 23.61 | 24.17 | 23.63 | 24.28 |

V. 결론

본 논문에서는 손상된 프레임 복원 및 프레임율 증가 뿐만 아니라 분산 동영상 부호화 시스템의 부가 정보 생성에도 활용될 수 있는 효율적인 MCE 기법을 제안하였다. 제안하는 기법은 순방향 및 역방향 움직임 추정을 이용하여 생성된 두 프레임의 평균을 구하여 MCE 기법의 성능을 향상시키는 방법으로 순방향과 역방향 두 번의 추정을 수행하여 기존의 방식보다 복잡도가 늘어나지만 움직임을 정밀하게 추정하며 블록 현상을 줄일 수 있다. 모의 실험을 통하여 기존의 MCE 기법과 제안하는 MCE 기법의 성능을 분석하였는데 다양한 특색을 가진 4개의 원본동영상과 복원영상에 대하여 모의 실험을 수행한 결과, 제안하는 기법은 기존의 기법과 비교하여 주관적 화질 및 PSNR 수치에서 성능 향상을 보였다, 특히 제안하는 방법에서 1/2 화소 단위로 움직임 벡터를 찾은 경우 움직임 벡터를 양방향으로 좀 더 세밀하게 찾는 장점이 있고, 기존의 기법과 비교하여 PSNR 성능 향상 및 주관적 화질 향상이 있음을 확인하였다.

참고 문헌

- [1] B. Girod, A.M. Aaron, S. Rane and D. Rebollo-Monedero, "Distributed video coding", *Proc. IEEE*, Vol.93, pp.71-83, Jan. 2005.
- [2] B. Yan and H. Gharavi, "Efficient error concealment for the whole frame loss based on H.264/AVC", *Proc. IEEE Int. Conf. on Image Process.*, pp.3064-3067, Oct., 2008.
- [3] J. Zhai, K. Yu, J. Li and S. Li, "A low complexity motion compensated frame interpolation method," *IEEE Int. Symposium on Circuits and Systems*, pp.4927-4930, May 2005.
- [4] B. Yan and H. Gharavi, "A Hybrid frame concealment algorithm for H.264/AVC," *IEEE Trans. on Image Process.*, Vol.19, pp.98-107, Jan. 2010.
- [5] Pang, L. Yu, M. Yi, W. Jiang, G. Liu and W. Jiang. "Relativity analysis-based error concealment algorithm for entire frame loss of stereo video," *The 8th IEEE int. Conf. on Signal Process.*, Apr. 2007.
- [6] C. Brites, J. Ascenso and F. Pereira, "Improving

- transform domain Wyner-Ziv video coding performance,” *IEEE Int. Conf. on Speech Signal Process.*, pp.525-528, May 2006.
- [7] R. Martins, C. Brites, J. Ascenso and F. Pereira, “Refining side information for improved transform domain Wyner-Ziv video coding,” *IEEE Trans. on Circuits and Systems for Video Technology*, Vol.19, pp.1327-1341, Sept., 2009.
- [8] T. N. Dinh, G. S. Lee, J. Y. Chang, H. J. Cho, “Side information generation using extra information in distributed video coding,” *IEEE Int. Symposium on Signal Processing and Information Technology 2007*, pp.138-143, Dec. 2007.
- [9] B. D Choi, J. W Han, C. S Kim, and S. J Ko. “Motion-compensated frame interpolation using bilateral motion estimation and adaptive overlapped block motion compensation,” *IEEE Trans. on Circuits and Systems for Video Technology*, Vol.17, pp.407-416, Apr. 2007.
- [10] Deng Hui-Ping, Yu Li, Xiong Wei, He Qing-Di and Li Rong, “Adaptive interpolation/extrapolation and motion vector processing method for frame rate up conversion,” *ICIG '09 Fifth Int. Conf. on Image and Graphics*, pp.18-22, Sept. 2009.
- [11] L. Natrio, C. Brites, J. Ascenso and F. Pereira, “Extrapolating side information for low-delay pixel-domain distributed video coding,” *Int. Workshop on Very Low Bitrate Video Coding*, Sept. 2005.
- [12] Mei-Juan Chen, Yi-Yen Chiang, Hung-Ju Li and Ming-Chieh Chi, “Efficient multi-frame motion estimation algorithms for MPEG-4 AVC/JVT/H.264,” *Proc. Int. Symposium on Circuits and Systems*, May. 2004.
- [13] J. Ascenso, C. Brites and F. Pereira, “Content adaptive Wyner-ZIV video coding driven by motion activity,” *IEEE Int. Conf. on Image Process.*, pp.605-608, Oct., 2006.
- [14] Mei-Juan Chen, Ming-Chung Chu and Chih-Wei Pan, “Efficient motion-estimation algorithm for reduced frame-rate video transcoder,” *IEEE Trans. on Circuits and Systems for Video Technology*, pp.269-272, Apr. 2002.
- [15] S. Borchert, R.P. Westerlaken, R. Klein Gunnewiek and R.L. Lagendijk, “Improving motion compensated extrapolation for distributed video coding,” *Proceedings of the 13th annual onference of the Advanced School for Computing and Imaging*, Jun. 2007.

권혜경 (Hye-gyung Kwon)

준회원



2010년 8월 가톨릭대학교 정보통신전자공학부 학사
2011년 3월~현재 가톨릭대학교 정보통신전자공학과 석사과정
<관심분야> 영상통신, 영상처리

이창우 (Chang-woo Lee)

총신회원

현재 가톨릭대학교 정보통신전자공학과 교수
<관심분야> 영상통신, 영상처리