

# MPEG Bit Stream에서의 Reverse Trick Mode 알고리즘

신성욱, 이동호

한양대학교 전자전기제어계측공학과

## Reverse Trick Mode Algorithms of MPEG Bit Stream

Sung-wook Shin, Dong-ho Lee

Department of Electronic, Electrical, Control and Instrumentation Engineering of Hanyang University

E-mail : swshin74@image.hanyang.ac.kr, dhlee@image.hanyang.ac.kr

### 요약

대용량 hard disk를 내장하고 있는 DTV용 PVR(personal video recorder)에서는 단순히 수신되는 방송 stream을 녹화하고 재생하는 기능 뿐만 아니라 다른 여러 가지의 부가적인 기능들을 지원하는 것이 필요하다. 그 중의 하나가 기존의 아날로그 VCR의 사용자들에게 친숙한 fast-forward play, reverse play, pause 등과 같은 trick mode play기능을 지원하는 것이다. 그러나 MPEG video는 화면간의 상관관계를 이용하여 압축하는 방식을 채택하고 있으므로 재생하고자 하는 frame이 intra frame이 아닌 한 독립적으로 재생할 수가 없어서 trick mode play 기능을 구현하기가 용이하지 않다. 특히, reverse trick mode의 경우에는 original stream에서의 마지막 frame이 먼저 display되어야 하는데 이를 위해 하나의 GOP가 모두 decoding되어야 하므로 더욱 그 구현이 어렵다. 본 논문에서는 reverse trick mode를 구현하기 위한 여러 알고리즘을 소개하고 이에 대한 system 복잡도, 메모리 사용량, 성능 등을 분석하고자 한다.

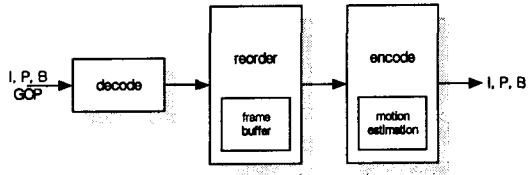
### 1. 서론

Bit stream의 reverse play란 video frame을 역순으로 play한다는 것이다. 즉, normal play에서 frame이 [I B1 B2 P1 B3 B4 P2 B5 B6 P3 B7 B8]의 순서로 display 되어야 한다면 reverse play는 [B8 B7 P3 B6 B5 P2 B4 B3 P1 B2 B1 I]의 순서로 display되어야 한다. 그러나 단순히 frame을 display 순서로 [B8 B7 P3 B6 B5 P2 B4 B3 P1 B2 B1 I]와 같이 순서

만 역순으로 하여 처리된 bit stream을 decoding 했을 때 생기는 두 가지의 문제가 있다. 첫번째 문제는 순방향 예측을 해야 decoding될 수 있는 P3가 예측의 기준으로 삼을 frame이 없어서 decoding 될 수 없으므로 한 개의 GOP에서 마지막의 I frame을 제외하고는 모두 decoding 될 수 없게 된다. 또 다른 문제는 motion vector가 forward 방향의 normal play를 기준으로 하여 만들어져 있으므로 reverse trick play에서 이 MV를 그대로 사용할 수 없다는 것이다. 이러한 문제들을 해결하면서 reverse trick play를 수행할 수 있게 하기 위한 가장 명확한 방법은 하나의 GOP를 decoding해서 모두 저장하고 있다가 역순으로 encoding하여 출력하게 하는 것이다. 그러나 이렇게 하기 위해서는 완전히 decoding된 GOP의 모든 frame들을 모두 저장하고 있을 정도의 큰 메모리가 요구된다. 또, GOP를 full로 decoding하였다가 다시 reverse play의 순서에 맞게 encoding하여야 하므로 DCT, VLC, quantization, 그리고 많은 연산량이 필요한 motion estimation 등의 과정이 필요하게 되고 이로 인해 system의 복잡도가 매우 높아지게 된다는 문제가 있어서 실제로 적용하기 어렵다.

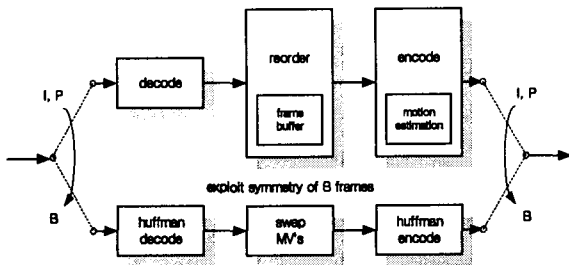
본 논문에서는 reverse trick mode play를 위한 구조와 이를 이용한 알고리즘들을 소개하고 이에 대한 system의 복잡도, 메모리 사용량, 성능 등을 분석하고 향후 연구 방향을 제시하고자 한다.

## 2. Reverse Trick Mode를 수행하기 위한 구조



<그림 2.1> 구조 1

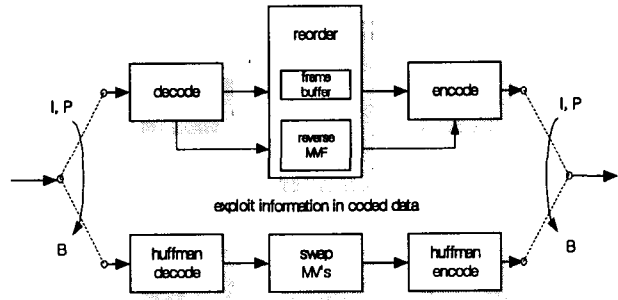
<그림 2.1>에 도시한 (구조 1)은 GOP 단위의 입력 bit stream을 모두 decoding한 후에 frame buffer에 모두 저장하고 있다가 reverse order로 re-encoding하여 출력 bit stream을 만들게 된다. 그러나, 15 frame GOP를 가정해보면, decoding된 영상 15frame을 모두 저장하고 있어야 하고 모든 P, B frame에 대해서 ME가 수행되어야 한다. 그러므로 서론에서도 언급했던 것처럼 이와 같은 구조는 매우 큰 frame buffer가 필요하고 re-encoding 과정에서 ME를 위해 많은 연산량이 요구되며, 이로 인해 system의 복잡도도 증가하게 된다. 그러므로 실제 field에 적용하기에는 문제가 있다고 할 수 있다.



<그림 2.2> 구조 2

<그림 2.2>에 도시한 (구조 2)에서는 GOP 단위의 입력으로부터 anchor frame과 non-anchor frame을 분리하여 처리한다. 먼저 non-anchor frame인 B frame의 경우 I-B-P가 P-B-I의 역순이 되어도 B frame의 symmetric한 성질 때문에 B frame의 forward MV와 backward MV를 단순히 서로 바꾸어 준 후에 압축된 bit stream 상태로 저장하고 있다가 최종 출력 시에 reverse order에 맞도록 출력하기만 하면 된다. Anchor frame인 I와 P frame의 경우에는 그림 1의 구조 1과 마찬가지로 frame을 완전히 decoding을 한 후에 frame buffer에 저

장하고 있다가 P frame에 대한 ME를 수행하여 encoding 한 후에 압축된 bit stream을 저장하고 있다가 reverse order로 출력하게 된다. 일반적인 15 frame, M이 3인 GOP를 가정해보면, 다섯장의 anchor frame만 저장하고 있으면 되므로 (구조 2)는 (구조 1)보다 frame memory의 용량이 덜 요구되고 ME도 P frame에 대해서만 행해지므로 연산량도 (구조 1)보다 덜 요구된다.

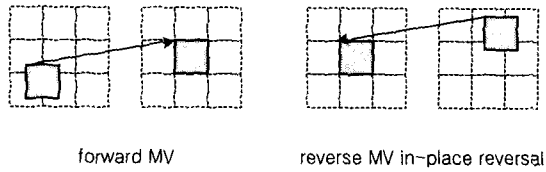


<그림 2.3> 구조 3

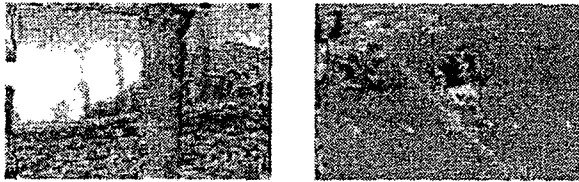
<그림 2.3>의 (구조 3)은 (구조 2)와 마찬가지로 anchor frame과 non-anchor frame을 분리하여 처리한다. B frame에 대해서는 (구조 2)와 같은 과정을 수행한다. 일반적인 15 frame, M이 3인 GOP를 가정해보면 I, P frame에 대해서 [I P1 P2 P3 P4]를 [I(P4) P1(P3) P2(P2) P3(P1) I(I)]와 같이 바꾸어주기 위한 frame memory가 요구되는데, 즉, decoding된 I를 저장하고 있다가 P1을 decoding하는데 이용하고 P1을 모두 decoding하여 저장하고 있다가 P2를 decoding하는데 사용하는 방식으로 마지막 P4까지 decoding하여 P4를 I frame으로 encoding 한다. 따라서 frame memory는 두 장 분량의 frame을 저장하고 있을 정도의 크기를 요구한다. 마지막 I frame은 original stream의 첫번째 I를 bypass시킨 frame이다. 나머지 P frame은 각각의 P frame이 가지고 있는 압축된 상태에서의 MV만을 이용하여 reverse bit stream에 맞도록 추정하여 수정할 수 있다. 그러므로, ME에 의한 연산량이 전혀 없게 된다. 이와 같이 (구조 3)은 (구조 1)이나 (구조 2)에 비하여 월등히 system 복잡도를 줄일 수 있으나 정확한 MV를 추정한다는 것이 매우 어려워 다소 화질의 열화가 있으므로 MV를 얼마나 잘 추정해내느냐가 관건이라고 할 수 있다. 이하에서 (구조 3)을 이용하여 MV를 추정해 내는 알고리즘과 변형된 (구조 2)를 이용하는 알고리즘을 소개하겠다.

### 3. Reverse Trick Mode 알고리즘

#### 3.1 In-place reversal 알고리즘

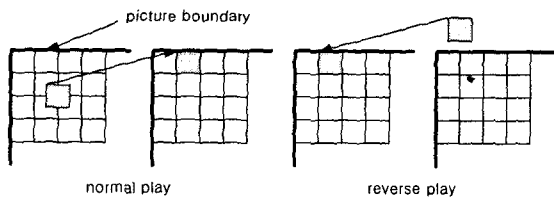


<그림 3.1> In-place reversal

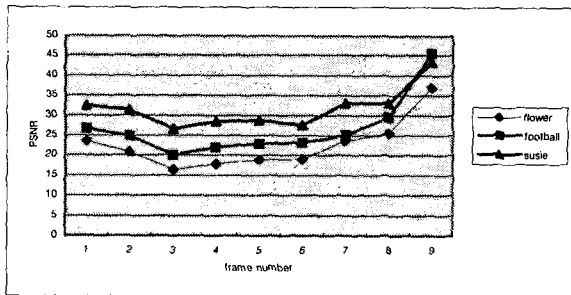


<그림 3.2> In-place reversal의 P1

(구조 3)을 이용하는 가장 간단한 In-place reversal은 P frame의 수평, 수직 MV element의 부호를 바꾸어서 이렇게 수정된 MV를 다시 입력 bit stream에 넣어주는 알고리즘이다. <그림 3.1>에 이 알고리즘을 간단히 도시하였다.



<그림 3.3> boundary problem



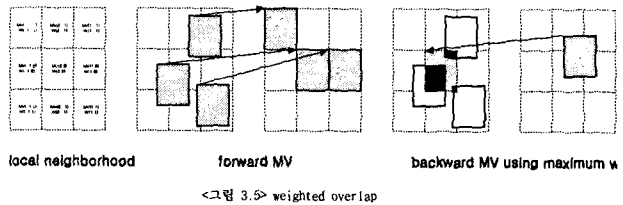
<그림 3.4> In-place reversal PSNR

<그림 3.2>는 In-place reversal을 적용한 flower garden과 football 영상의 P1(original bit stream order) frame이다. 꽃밭과 경기장 바닥에 대해서는 문제가 없지만 나무의 경계와 움직임이 불규칙한 사람 부분에서 심한 왜곡을 볼 수 있

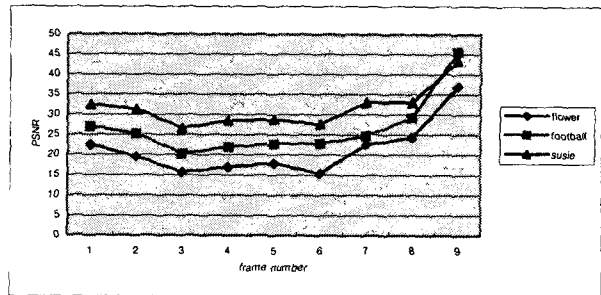
다. 그리고 화면의 경계에서 boundary problem이 발생하는 데 이는 <그림 3.3>과 같이 새롭게 만들어진 reverse MV가 화면 경계 밖으로부터 MC를 하도록 하게 해서 생기는 문제이다. Boundary problem이 발생하지 않도록 보정하면 PSNR을 더욱 높일 수 있다.

<그림 3.4>에서 flower, football 그리고 Susie영상의 한 개 GOP에 대한 PSNR(원 영상의 순서로)을 그래프로 나타내었다. 대체로 화질 열화가 심하며 움직임이 많을수록 더욱 좋지 않은 PSNR을 나타내는 것을 알 수 있다.

#### 3.2 maximum overlap 알고리즘



<그림 3.5> weighted overlap



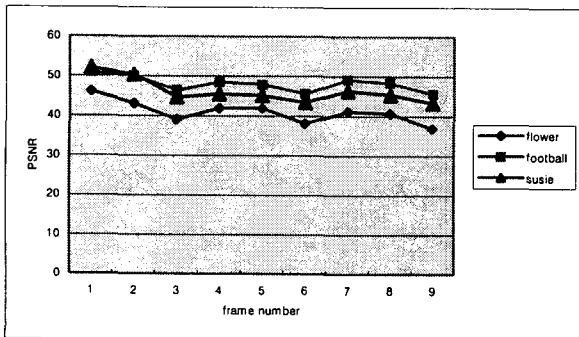
<그림 3.6> maximum overlap PSNR

(구조 3)을 이용하는 이 알고리즘은 In-place reversal이 무조건 MV를 반대로만 해주는 방법과는 달리 reverse MV를 만들어주려는 MB에 가장 overlap이 큰 16X16 pixel을 MC에 사용하는 MB의 MV를 이용하는 방법이다. Buffer 사용량 및 계산량을 줄이기 위하여 자신을 포함한 주위 9개 MB(local neighborhood)의 MV에 대해서만 overlap을 계산한다. In-place reversal에 비해 maximum weighting 계산 및 local neighborhood의 MV를 저장하기 위한 buffer가 추가되어 복잡도가 다소 증가된다. 이 알고리즘을 이용하면 Susie, football 영상과 같이 motion이 별로 없는 영상에 대해서는

In-place reversal과 별 차이가 없는 PSNR을 나타내지만 움직임이 큰 부분에서는 더 나은 결과를 보인다. Flower 영상에 대해서 <그림 3.6>과 같이 In-place reversal보다 결과가 약간 더 나쁘게 나온 것은 boundary problem에 더 크게 작용했기 때문이며 이를 해결하면 더 나은 PSNR을 보일 것이다.

### 3.3 IP conversion 알고리즘

변형된 (구조 2)를 이용하는 IP conversion은 원래 입력 bit stream이 [I B B P B B P B B P]일 때, 이 stream을 [I B B I B B I B B I]가 되도록 P frame을 I frame으로 변환시키는 알고리즘이다. 이 알고리즘을 이용하는 두 가지 방법이 있는데, 먼저 일반적으로 decoder가 decoding하는 순서에 따라 P frame을 복호하여 frame memory에 저장된 P frame을 I frame으로 encoding하여 HD에 저장하였다가 reverse play 요청이 있으면 I, B stream을 HD로부터 읽어서 display 하는 알고리즘이다. 추가적인 frame memory 및 ME와 같은 많은 연산이 요구되지 않는다는 장점이 있지만 HD의 용량이 약 20% 이상 더 필요하게 된다는 단점이 있다. 두 번째 방법은 HD에는 [I B B P B B P B B P]를 저장하고 reverse play 요청이 있으면 HD로부터 sequence를 decoder에 입력하기 전에 IP conversion을 수행하는 방법이다. 첫째 방법에 비해 추가적인 HD가 요구되지 않는 장점이 있지만, decoder와는 별도로 I, P frame을 decoding하기 위한 보조 decoder가 필요하고, 복원된 frame을 저장하고 있어야 할 메모리도 필요하다. 그러나, 새롭게 MV를 만들지는 않기 때문에 ME는 필요 없다. 이와 같이 IP conversion을 하면 sequence에 P frame이 없기 때문에 상당히 높은 PSNR 결과를 얻을 수 있다.



<그림3.7> IP conversion PSNR

### 3.4 기타 다른 알고리즘

기타 다른 알고리즘은 대부분 (구조 3)을 이용하여 3.2에서 설명한 maximum overlap 알고리즘을 응용한 알고리즘들인데, 최대 overlap이 아닌 overlap된 부분의 평균이 되는 16X16 pixel을 MC에 사용하도록 MV를 정해주는 방법, overlap된 부분에 weighting을 주어서 이를 가지고 weighting 평균을 취하여 MV를 정해주는 방법이 있다. 또 CBP는 MC를 위하여 취하는 16X16 pixel에 대해서 수정해줘야 할 정도를 나타내는 값이므로 이 값의 크기에 따라 weighting을 달리 해주는 방법도 있다.

## 4. 결론 및 이후 과제

3.2와 3.4에서 소개한 알고리즘은 영상의 종류에 따라 화질 열화의 정도가 다르다. 따라서 다양한 test 영상을 분석하여 각 알고리즘을 적응적으로 적용할 수 있는 방법을 찾아야 하며 각 boundary problem과 같은 알고리즘 자체의 문제도 수정해야 한다. 3.3에서 소개된 알고리즘은 높은 시스템의 복잡도와 추가적인 메모리 또는 HD가 요구된다. 따라서 video decoding을 위해 필요한 최소한의 system만으로 reverse trick play를 구현하기 위해 기존의 decoder와 중복되는 부분을 공유하는 방법 및 frame 메모리를 공유할 수 있는 방법을 연구해야 하며 IP conversion시에 증가하는 bit rate를 control할 수 있는 방법 또한 연구 대상이다. 그 이후에는 이러한 알고리즘을 fast reverse mode, fast forward mode에도 확장할 수 있어야 한다.

## 5. 참고 문헌

- [1] Susie J Wee, "Reverse Motion Vector Fields", IEEE International Conference on Image Processing, Oct.1988
- [2] S. Wee and B.Vasudev, "Compressed-domain reverse play of MPEG video streams", in SPIE voice, video, and data communications conference, (Boston MA), Nov, 1998.
- [3] M.S.Chen and D.D Kandlur, "Downloading and stream conversion: Supporting interactive playout of videos in a client station," in proc. 2<sup>nd</sup> Int. IEEE conference Multimedia Computing and systems, 1995.