

화상통신에서의 오류전파 제어를 위한 보조 모션벡터 압축 기법

이주경⁰ 최태욱 정기동
부산대학교 전자계산학과
{jkleee⁰, tuchoi, kdchung}@melon.cs.pusan.ac.kr

Encoding Scheme for Error Propagation Control Using Motion Vector in Video Communication

Joo-Kyong Lee⁰ Tae-Uk Choi Ki-Dong Chung
Dept. of Computer Science, Pusan National University

요 약

표준 비디오 압축 기법은 프레임 간의 시간적, 공간적 중복성을 이용하여 데이터를 압축한다. 압축된 프레임이 인터넷을 통하여 전송되는 중에 네트워크의 상태에 따라 패킷의 손실이 발생할 수 있다. 이러한 오류가 발생하면 손실된 블록을 포함하는 프레임의 화질 뿐 아니라, 이 프레임을 참조하여 압축된 이후의 프레임에도 영향을 미친다. 본 논문에서는 전송오류가 발생한 경우, 디코더에서의 오류전파를 최소화하기 위해 보조 모션벡터를 이용한 압축 기법을 제안한다. 즉, 여러 개의 후보 프레임 중 현재 압축하려는 매크로블록과 가장 가까운 값을 가진 두 개의 매크로 블록을 선택하여 그 위치를 기본 모션벡터와 보조 모션벡터로 지정한다. 이때, 두 모션벡터는 서로 다른 프레임에 속한다. 디코더에서 압축해제를 위해 참조하는 기본 모션벡터의 프레임이 손실된 경우, 보조모션 벡터를 이용하여 원래의 화질에 가깝게 디코딩 할 수 있다. 이 기법은 오류제어를 위한 피드백 채널이 필요하지 않으며, 네트워크상의 전송오류 발생시 표준 H.263 압축보다 높은 성능을 보인다.

1. 서 론

인터넷이 널리 보급되고 발전함에 따라 인터넷을 통한 화상 통신이 일반화되고 있다. 그러나, 화상통신은 협소한 대역폭, 대용량 데이터 등의 문제점으로 만족할 수준의 서비스를 제공하지 못하는 실정이다.

일반적으로 화상통신의 데이터는 압축되어 전송된다. 음성 데이터의 대표적인 압축 표준으로 G.723.1이 널리 이용되고 있다. 화상 압축 표준으로는 일반적인 동영상 압축을 위한 MPEG과 대화식 화상 통신을 위한 H.263이 대표적이다. 화상 압축은 DCT, 움직임 예측, 움직임 보상 과정을 거치면서 압축 효율을 높인다. 그러나, 특정 프레임의 오류는 압축 기법으로 연관된 다른 프레임에도 영향을 미치게 되어 오류가 시간적, 공간적으로 전파된다. 특히, 인터넷은 화상 통신의 필수조건인 실시간성, QoS 관리 기능의 미약으로 오류는 더욱 확대된다.

지금까지 전송오류에 대응하여 화상통신에서의 화질을 향상시키기 위한 많은 기법이 연구되어 왔다. 특성에 따라 분류해보면 오류에 강한 압축기법[1][2][3], 디코더에서의 오류는닉 기법[4][5], 코더와 디코더의 피드백 채널을 이용한 에러 제어 기법[6][7] 등이 있다. 실시간 전송 또는 멀티포인트 응용에서의 오류전파를 제어하기 위해 피드백 채널을 사용할 경우 네트워크의 부하 증가, 피드백 정보 적용 전 기간 동안의 오류 전파가 발생하는 문제점이 있다. 본 논문에서는 전송오류가 발생한 경우 디코더에서의 오류전파를 제어하기 위한 압축 기법인 AMV (Auxiliary Motion Vector) 기법을 제안한다. 이 기법은 엔코더와 디코더 사이에 피드백 채널이 필요하지 않으며, 모션벡터라는 추가적인 정보를 이용하여 오류전파를 제어한다. 이 기법은 엔코더에서 매크로블록을 압축할 때, 참조되는 여러 개의 후보 프레임 중에서 움직임 예측, 보상을 통해 2개의 모션벡터를 선

택하여 기본벡터와 보조벡터로 지정한다. 이때, 기본 벡터는 매크로블록에 가장 가까운 값을 가지는 블록을 지정하고, 보조 벡터는 두 번째로 가까운 블록을 지정한다. 또한 두 벡터는 서로 다른 프레임에 존재하도록 한다. 실제 매크로블록의 압축은 기본벡터의 정보를 이용하여 수행하며, 보조벡터를 부가적으로 전송한다. 전송 도중 기본벡터의 프레임에 오류가 발생한 경우 보조벡터를 이용하여 압축해제를 수행한다. 데이터에 가장 가까운 블록을 가리키므로 보조벡터를 이용한 경우 완벽한 복구는 불가능하지만 원래 값에 근접하게 복구 가능하다.

제안하는 기법의 성능평가를 위하여 인터넷 트레이스를 기반으로 하는 모의 실험환경을 구축하였다. 비디오 압축 모듈은 H.263 TMN10을 제안하는 기법에 맞게 수정하였으며, NS-2(Network Simulation Version 2)를 이용하여 인터넷 트레이스를 측정했다.

논문의 구성은 다음과 같다. 2장에서는 관련된 오류제어를 위한 기존 압축 기법을 살펴보고, 3장에서는 AMV기법에 대해서 설명한다. 4장에서는 모의 실험을 통한 성능평가를 보이며, 5장에서 결론과 향후 연구 방향에 대해 설명한다.

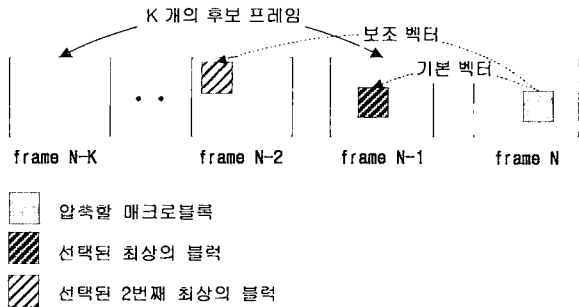
2. 관련 연구

엔코더와 디코더 사이에 피드백 정보를 사용하지 않고 오류 전파를 제어하는 기법으로는 Intra-Mode Refresh[8] 기법이 있다. 이 기법은 채널의 상태와 비디오의 특성에 따라 Intra Mode로 압축하는 비율을 조정하는 기법이다. Intra Mode로 압축되면 비디오 코딩의 효율이 떨어지므로 Intra Mode로 압축되어야 할 시점을 잘 선택해야 한다. 다른 기법으로는 M. Budagavi가 제안한 MF-BMC[9] 기법이 있다. 이 기법은 프레임을 블록으로 나누고, 여러 개의 후보 프레임 중에서 그 블록에 가장 잘 맞는 블록을 선택한다. MF-BMC (MultiFrame-BMC) 기법은 각 블록이 서로 다른 프레임을 참조할 수 있으

므로 전통적인 압축기법에 비해 오류전파가 발생할 확률이 적어진다. 그러나, 참조 블록에 오류가 발생한 경우, 오류전파를 막을 수 없다. 그 외 기법으로는 Y. Yu가 제안한 DMRP (Dynamic Multi-Reference Prediction)[10] 기법이 있다. DMRP는 MF-BMC와 유사하게 여러 개의 후보 프레임 중에서 움직임 예측을 통해 현재 매크로블록에 가장 잘 맞는 K개의 블록을 구하고, 각각의 블록에 대하여 움직임 보상을 수행한다. 이때, K개의 블록은 서로 다른 프레임에 존재하며, 각 블록에 대하여 수행된 움직임 보상의 결과 값을 평균하여 압축을 수행한다. 그러므로 한 개의 참조 블록에 오류가 발생하더라도 다른 참조 블록이 정상적이면 오류는 희석되어 전파된다. 즉, 전파되는 오류의 양은 MF-BMC에 비하여 적어진다. 그러나, 한 개의 블록을 압축하기 위해 참조되는 프레임 수가 증가하므로 특정 프레임에 오류가 발생하면 그 프레임을 참조하는 여러 블록에서 오류가 전파될 확률은 높아지는 단점이 있다.

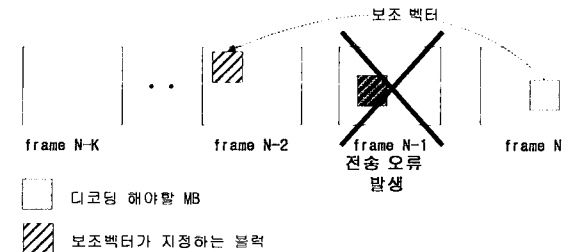
3. 보조 모션벡터를 이용한 오류 제어

본 장에서는 모션벡터를 이용한 오류 기법에 대해 설명한다. 2장에서 살펴본 바와 같이 MF-BMC는 참조하는 블록에 오류가 발생하면 INTRA Mode로 코딩될 때까지 디코더에서 오류는 계속 전파된다. 또한, DMRP 기법은 참조하는 블록 중 한 개만 오류가 발생하더라도 오류는 희석되어 전파된다.



[그림 1] 매크로블록 압축을 위한 모션벡터 찾기

제안하는 AMV기법은 이러한 점에 착안하여 참조블록에 오류가 발생한 경우에만 보조벡터를 사용하여 압축해제를 수행한다. 코더에서의 압축과정은 다음과 같다. N번째 프레임을 압축한다고 가정하자. 첫째, N번째 프레임이 참조할 K개의 후보 프레임을 지정한다. 현재 프레임 이전에 압축된 후, 압축 해제된 K개의 프레임이 후보 프레임이 된다(그림 1). 둘째, K개의 프레임 중 압축할 매크로블록에 가장 가까운 2개의 블록을 선택한다. 셋째, 선택한 블록 중 최상위 블록의 위치 정보를 기본벡



[그림 2] 보조벡터를 이용한 압축해제

```

for ( i = 0; i < Total_MBs ; i++) { //각 MB에 대하여
//K개 중 첫 번째 프레임
pmv = do_motion_estimation(i,1); //(sad, mv)
amv = (-1,0); //(sad, mv)
for ( j = 2; j <= K ; j++) { //K개의 후보 프레임
cmv=do_motion_estimation(i,j); //current_mv 구하기
if( cmv.sad < pmv.sad ) {
amv = pmv; //primary mv값을 auxiliary mv로 복사
pmv = cmv; //primary mv를 갱신
} 기본벡터 및 보조벡터 정보 갱신
else if( cmv.sad > pmv.sad &&& (cmv.sad < amv.sad
|| amv.sad = -1)){
amv = cmv;
} // 보조벡터 정보 갱신
}
diff = do_motion_compensation(pmv,mv);
store_MB(pmv,amv,diff) //프레임 압축 알고리즘
}
    
```

터로 지정하고, 두 번째 블록의 위치를 보조벡터로 지정한다. 넷째, 기본벡터를 이용하여 매크로블록을 압축하고, 보조벡터를 부가적으로 전송한다.

디코더에서의 압축 해제과정은 다음과 같이 세 가지 경우로 나눌 수 있다. 첫째, 기본벡터가 지정하는 프레임의 블록에 오류가 없다면, 기본벡터를 이용하여 압축해제를 수행한다. 둘째, 기본벡터가 참조하는 프레임에 전송오류가 발생한 경우, 디코더는 보조벡터가 지정한 프레임의 블록정보를 사용하여 압축을 해제한다. 셋째, 보조벡터가 지정한 프레임만 오류가 발생한 경우, 디코더는 보조벡터의 손실에 영향을 받지 않고 압축해제를 수행한다. AMV 기법에서는 오류전파에 영향을 미치는 요소가 기본벡터 뿐이므로 DMRP과 비교하여 오류전파 발생 확률이 낮아진다. 또한, 기본벡터가 참조하는 블록에 손실이 발생하여 보조벡터를 사용하더라도 K개의 블록 중 2번째로 현재 매크로블록에 가까운 값이므로 MF-BMC에 비하여 전파되는 오류는 적게된다.

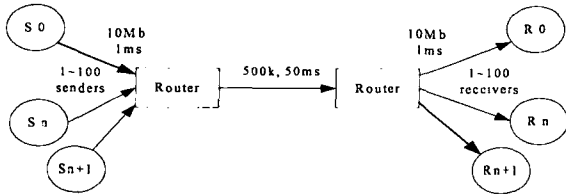
```

for ( i = 0; i < Total_MBs ; i++) {
pmv = get_pmotionvector(i);
amv = get_amotionvector(i)
if ( is_exist(pmv))
block= get_referred_block(pmv);//pmv block 구하기
else if( is_exist(amv)){
block= get_referred_block(amv);//amv block 구하기
else
block= get_previous_frame_block(i); // 현재 블록과
// 가장 가까이 존재하는 프레임 블록 구하기
// 현재 MB와 같은 번호의 MB를 구함
do_decoding(block);
}
}
    
```

[그림 4] 프레임 압축해제 알고리즘

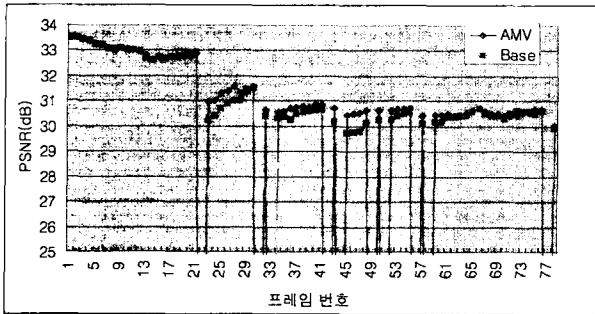
4. 성능평가

제안된 기법의 성능 평가를 위해 H.263 TMN10을 수정하였으며, 인터넷 트레이스 수집을 위하여 NS-2 사용했다. 인터넷 트레이스를 구하기 위한 모의 실험환경은 (그림 5)와 같이 토폴로지를 구성하고 TCP 노드의 수를 조절하여 다양한 손실율을 가진 인터넷 트레이스를 수집하였다. 토폴로지의 모든 단말은 병목 현상이 없는 링크를 통해 동일한 라우터에 연결되어 있고, 10Mb의 대역폭과 1ms의 전파지연을 가진다. 라우터는 500Kb의 대역폭과 50ms의 전파지연을 갖는 병목 링크로 연결된다. S 노드는 데이터를 전송하고, R 노드는 데이터를 수신한다. S_{0-n}와 R_{0-n} 노드는 TCP노드로 한 번에 1KB의 패킷을 송수신한다. n+1번째의 노드는 UDP노드로 인터넷 화상통신 응용을 수행한다. 이 노드는 초당 10프레임, 800byte를 전송한다.



[그림 5] 인터넷 트레이스 수집을 위한 NS-2 토폴로지

사용된 화상은 QCIF 포맷으로 초당 10 프레임 비율로 압축하였다. (그림 6)는 인터넷에서 전송되는 패킷의 평균 손실율이 13%인 경우의 PSNR 값을 측정한 것이다. 불연속적인 점으로부터 오류가 발생하였으며, 그 직후의 PSNR값을 비교하면 AMV가 H.263 표준에 비하여 효과적임을 알 수 있다.



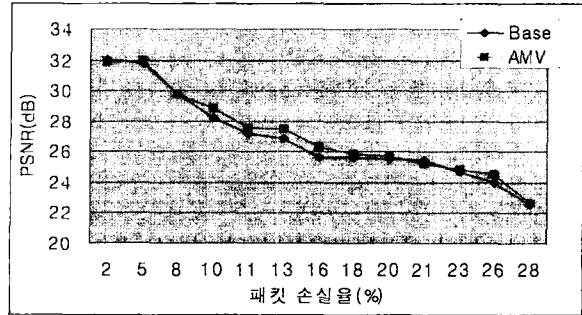
[그림 6] 패킷 손실율 13%에서의 PSNR 비교

(그림 7)은 패킷의 평균 손실율을 달리하여 평균 PSNR을 구한 것이다. 패킷의 손실율이 적은 경우에는 H.263 표준과의 차이가 거의 없으나, 10%대의 손실율에서 AMV가 우수함을 알 수 있다. 그러나, 참조하는 프레임의 범위를 넘는 연속적인 패킷 오류가 발생한 경우, 즉, 20%대에서는 H.263 표준과의 차이가 다시 줄어든다. 이를 통하여 AMV는 10%대의 전송오류에 강함을 알 수 있다.

5. 결론 및 향후과제

본 논문에서는 인터넷 통신에서 발생하는 전송오류를 피드백 채널을 사용하지 않고, 디코더에서 제어할 수 있는 AMV (Auxiliary Motion Vector) 압축 기법을 제안하였다. AMV 기법은 전송오류의 확률을 줄이고, 최적의 참조 블록을 찾기 위해 여러 프레임에서 움직임 보상을 수행한다. 또한 기본벡터에

전송오류가 발생하면 디코더에서는 정상적으로 도착한 보조벡터를 사용함으로써 오류전파를 최소한으로 줄일 수 있다. 향후 연구과제로 AMV를 찾기 위한 시간을 줄이는 방법과 무선네트워크 상에서의 적용방안 연구, 현재 매크로블록 단위로 수행하는 움직임 보상을 GOB 또는 블록 단위로 확장하여 AMV의 장단점을 연구할 것이다.



[그림 7] 패킷 손실율에 따른 PSNR 비교

6. 참고 문헌

- [1] Y. Takashima, M. Wada, and H. Murakami, "Reversible variable length codes", IEEE Trans. Communications., vol. 43, pp. 158-162, Feb./Mar./Apr. 1995
- [2] D. W. Redmill and N. G. Kingsbury, "The EREC: An error resilient technique for coding variable-length blocks of data," IEEE Trans. Image Processing, vol. 5, pp. 565-574, Apr. 1996.
- [3] S. Wenger, G.Knorr, J. Ott, F. Kossentini: "Error resilience support in H.263+", IEEE Trans. on circuit and System for Video Technology, vol. 8, no. 6 pp. 867-877, Nov. 1998.
- [4] S. Aign, "Error concealment for MPEG-2 video", Signal Recovery Techniques for Image and Video Compression and Transmission, A. K. Katsaggelos and N. P. Galatsanos, editors, ch. 8, pp 235-268, Kluwer Academic Publishers, 1998.
- [5] H. Sun and W. Kwok, "Concealment of damaged block transform coded images using projections onto convex sets", IEEE Trans. Image Proc., vol. 4, no. 4, pp. 470-477, Apr. 1995.
- [6] T. Nakai, and Y. Tomita: "Core Experiments on Feedback channel Operation for H.263+", ITU-T SG15 contribution LBC 96-308, Nov. 1996.
- [7] Wada M. Selective Recovery of Video Packet Loss Using Error Concealment. IEEE J. Select. Areas in Commun. 7(5):807-814, June 1989.
- [8] Judy Y. Liao and John Villasenor "Adaptive Intra Block Update for Robust Transmission of H.263", IEEE Transactions on Circuits and Systems for Video.
- [9] Budagavi, M., Gibson, J.D, "Multiframe Video Coding for Improved Performance over Wireless channels." IEEE Trans. Image Processing. Vol. 10, pp. 252-265, Feb. 2001
- [10] Yang Yu, Xuelong Zhu: "Dynamic Multi-reference Prediction in Video Coding for Improved Error Resilience over Internet", IEEE Pacific Rim Conference on Multimedia 2001, pp. 102-109, Oct. 2001,