

H.261 비디오 스트림상의 부가영상 삽입 알고리즘 개발

이성우 오하령 성영락

국 민 대 학 교 전 자 공 학 과

Development of an additive video-stream insertion algorithm on the H.261 video stream

Sung-Woo Lee Ha-Ryoung Oh Yeong-Rak Seong
Dept. of Electronic Engineering, Kookmin University

요 약

화상회의 시스템의 영상 압축표준 중 하나인 H.261은 화상전화기에서 주로 사용되고 있다. 본 논문에서는 기존의 H.261 영상에 부가영상을 삽입하는 방법을 제안한다. H.261 영상에 단순히 부가영상을 삽입하면 움직임보상 데이터 처리를 하지 않기 때문에 전달된 H.261 영상으로부터 원래의 영상을 복원하기 어렵다. 이를 해결하는 방법으로 원시 H.261 영상 전체를 복호화 한 후 부가영상을 삽입하고 다시 부호화하는 방법이 있으나 이 경우 처리해 주어야 할 데이터가 너무 많아 수행속도의 저하를 가져온다. 본 논문에서는 영상의 전체를 부호화하는 대신 실제 필요한 부분만 부호화하는 방법을 제안한다. 제안한 방법은 움직임 보상 정보가 영상에 아무 영향을 미치지 않을 경우에는 허프만 복/부호화만을 사용하여 단순 삽입을 하고, 움직임 보상 정보가 부가영상과 겹치게 되어 부가 영상이 포함된 영상을 전달 받는 측에서 문제가 될 경우만 미리 복호화해 두었던 영상데이터를 보낸다. 간단한 실험을 통하여 제안된 알고리즘의 성능을 분석한 결과 전체를 복/부호화하는 방법에 비하여 대략 3배의 속도의 향상을 보였다.

1. 서론

최근 인터넷과 고속통신망, 그리고 가까운 미래에 등장할 IMT-2000 등은 영상전화, 화상회의 및 영상을 이용한 여러 가지 다양한 서비스들을 요구하고 있다. 사용자에게 보여지는 영상에 광고 및 기타 영상 정보를 포함시켜 보낸다면 통화 영상에 좀더 많은 부가 가치를 담을 수 있을 것이다. 그리고 영상전화, 화상회의, 여러 가지 영상 서비스 및 관련 기반 시설들의 보급에 크게 기여할 수 있을 것이다. 그러나 기존의 H.261 영상에 부가 영상을 단순히 삽입하는 것은 원 영상을 수신측에서 손상 없이 복원해내지 못하는 문제점이 있다. 이것은 송신측에서 중간에 부가정보가 삽입된 사실을 모르고 실제와는 다른 예측 값으로 움직임 보상 데이터를 보내기 때문이다.[1]

부가정보를 원 영상의 손실 없이 수신측에서 복호화 할 수 있기 위해서는 H.261 영상으로 전달된 데이터를 부호화 한 후 부가 영상을 복호영상 위에 덮어 씌우고, 다시 그 복호영상을 H.261 영상으로 부호화해 주는 방법을 생각해볼 수 있다. 그러나 이 경우 프레임 전체를 부호화해야 하기 때문에 시스템에 많은 부하가 걸리는 문제가 있다. 본 논문에서 제안한 방법은 우선 원시 H.261 영상을 복호화 한다. 그리고 실제로 움직임벡터로 인해 수신측의 복호영상의 손상이 예상될 경우에만 손상되지 않도록 해당 부분을 인트라 블록으로 부호화하여 전달한다. 손상이 가지 않는 경우에는 H.261 영상에 부가 H.261 영상을 단순 삽입한다. 본 논문은 이를 위해서 2장에서

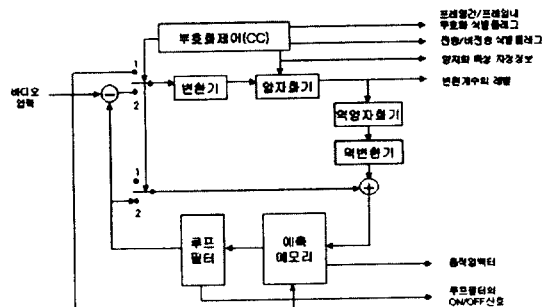
H.261표준에 대하여 설명하고, 3장에서는 제안한 알고리즘에 대해서 설명한다. 4장에서는 알고리즘의 구현 및 결과에 대해서 고찰한다. 5장은 결론 및 향후과제이다.

2. H.261

2.1 H.261 소개

H.261은 ISDN 화상전화의 표준인 H.320 중 영상 데이터의 압축을 담당하는 권고안으로 1990년 ITU-T의 승인을 얻었다. 동영상 데이터전송은 다른 데이터에 비해서 동영상 데이터의 크기가 매우 크므로 높은 수준의 압축율을 필요로 한다.[2]

따라서 H.261은 효과적인 동영상 압축을 위하여 <그림 1>에서 보는 바와 같이 변환기에서 DCT 변환, 런-랭스 부호화를 수행하고, 이후에 양자화, 움직임보상 DPCM, 허프만 부호화 등의 여러 가지 손실/무손실 압축 기술들을 결합하여 데이터의 압축을 수행한다.



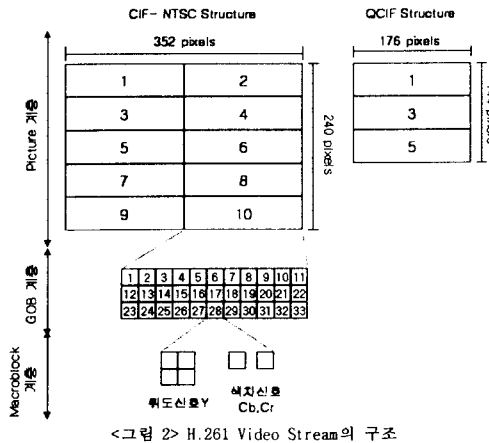
<그림 1> H.261 Video Stream의 소스 부호기

움직임 보상데이터 처리를 위하여 H.261은 움직임 보상 정보를 포함하지 않은 블록을 인트라 블록, 움직임 보상 정보를 포함한 블록을 인터 블록으로 나누어서 서로 다른 방법으로 부호화한다.

<그림1>은 H.261 영상의 소스 부호기이다. 부호화 제어부에서는 해당 블록을 인트라/인터 블록 중 하나로 결정하는 역할을 한다. 인트라 블록이 선택될 경우에 변환기에서 DCT 변환이 수행되고 결과를 양자화한다. 양자화한 데이터는 결과로 출력하고, 다음 프레임을 위해 결과 데이터를 역양자화와 역부호화하여 원시영상을 예측메모리에 저장한다. 인터 블록일 경우에는 DCT 변환을 하지 않고 예측메모리의 영상과 실제영상의 차이를 움직임벡터 데이터로 출력한다. 루프필터는 예측영상에 누적되는 에러를 보정해주는 역할을 한다. 소스 부호기에서 만들어진 데이터는 다시 한번 허프만 부호화로 다시 한번 압축한다. [3]

2.2 H.261의 구조

앞에서 설명한 방법으로 소스데이터가 부호화되면 H.261은 소스를 <그림2>와 같은 구조로 계층화한 후 수신측으로 전달한다. [1]



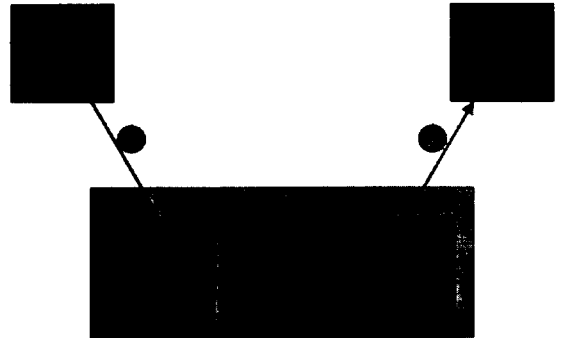
<그림 2> H.261 Video Stream의 구조

전달할 H.261 영상의 구조는 픽처계층, GOB 계층, 매크로블럭 계층으로 이루어진다. 픽처계층은 CIF와 QCIF 두 가지 종류가 있다. CIF의 경우 화면의 크기가 352x288 pixel, NTSC 타입의 CIF는 352x240이다. QCIF의 경우에는 CIF의 1/4인 176x144 pixel이다. CIF의 경우에 12개의 GOB(Group of Blocks)으로 이루어져 있고, NTSC타입의 경우에는 10개, QCIF의 경우에는 3개의 GOB로 이루어져 있다. <그림 2>에서 보는 바와 같이 각각의 GOB의 번호는 픽처 계층에서 화면의 위치를 나타낸다. 각각의 GOB는 33개의 매크로블럭을 가지고 있다. 만약 해당 매크로블럭이 인트라 블록이라면 DCT 변환을 수행한 데이터를 보내는 대신 예측영상과의 차이값인 움직임 벡터값만을 보내준다. 매크로블럭은 16x16의 위도정보 블럭 Y와 8x8의 색차정보 Cb,Cr로 이루어진다. 색차정보가 적은 이유는 인간의 시각이 위도에는 민감한데 비하여 색차정보에는 둔감하기 때문이다. H.261은 이렇게 계층화된 구조로 영상데이터를 전달한다.

3. 부가영상 삽입 알고리즘

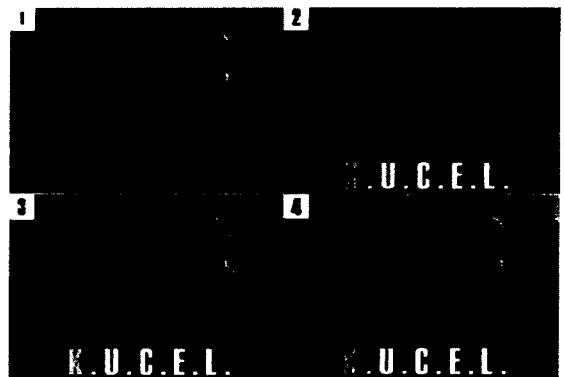
3.1. 기본개념

본 논문에서는 NTSC 타입의 H.261 영상 위에 하단 9GOB와 10 GOB를 원 영상이 아닌 부가영상으로 대체하는 알고리즘을 제안한다. <그림 3>은 기본 개념도이다. 송신측에서 H.261 영상을 전송하면 영상 합성기는 부가영상을 9,10GOB에 원 영상과 합성하여 수신측에 전달한다. 여기서 문제가 되는 것은 하단의 2 GOB만을 단순히 대체하게 되면 수신측에서 원 영상을 손상 없이 복구할 수 없다는 것이다.



<그림 3> 기본 개념도

아래의 <그림4>에서 보면 원 영상은 <그림4-1>이고, 부가영상은 <그림4-2>이다. <그림4-3>은 허프만 복호만을 수행하여 GOB와 매크로블럭을 찾아내고 9,10GOB에 부가영상을 단순히 대체할 경우이다. 이 경우에 아래쪽의 부가영상이 위쪽의 원 영상으로 번져서 원 영상부분이 제대로 복호화되지 않는 것을 확인할 수 있다. 이런 현상은 수신측에서 복호화할 때, 실제로는 원 H.261 영상의 일부만이 부가영상으로 대체되었는데도 불구하고 부가영상으로 대체되기 전의 영상부분을 참조하는 움직임벡터 정보를 이용하여 화면을 복호하기 때문이다. 거꾸로 말하면 송신측에는 중간에 부가영상이 삽입되었는지 알 수 있는 방법이 없기 때문에 송신측이 부가영상이 삽입된 부분을 삽입된 지 모르고 부가영상으로 대체된 부분을 참조하는 H.261 영상을 만들어 내고 있기 때문이다.



<그림 4> H.261 영상

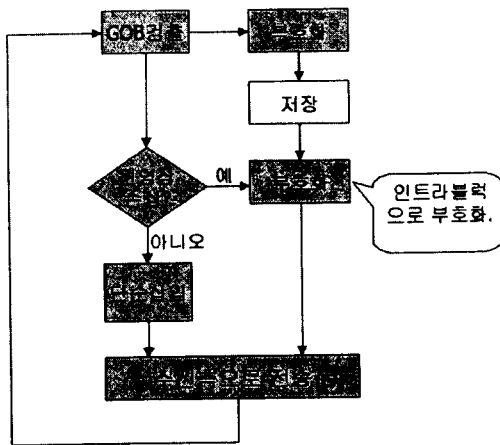
이와 같은 문제를 해결하기 위해서는 단순히 부가의 H.261 영상을 삽입하는 것이 아니라, 우선 송신측으로부터

전송 받은 H.261 영상의 한 프레임 전체를 복호화 해야 한다. 그리고 그 복호화된 영상 위에 복호된 상태의 부가영상을 삽입한다. 그리고 합성된 영상 전체를 다시 부호화하여 다시 수신측에게 전달해야 한다. 이렇게 하는 이유는 움직임 벡터가 복호화의 마지막 단계인 역DCT변환이 끝난 상태의 영상을 대상으로 만들어 졌기 때문이다. 따라서 시간이 많이 걸리는 DCT 변환을 하지 않고 움직임벡터 정보만으로는 손상된 부분을 복구하기 어렵다. 하지만 이 방법은 한 프레임 전체를 복호화와 부호화하기 때문에 중간에서 영상을 삽입하는 측의 컴퓨팅과파워가 많이 소모되는 문제가 있다.

3.2. 제안한 삽입방법

전체를 복호/부호화 할 경우에 엄청난 양의 계산량이 필요하다. 따라서 본 논문에서는 전체를 복호화 하되 부호화는 실제로 원 영상의 움직임 벡터가 부가영상 부분을 참조하여 원영상의 복호에 손상이 예상될 때에만 미리 복호해 두었던 영상을 부호화하여 전달하는 방법을 제안한다. 여기서 부호화 할 때에는 인트라 블록으로 부호화하여야 한다. 인트라 블록으로 부호화하지 않을 경우에는 수신측에서 잘못된 예측영상을 가지고 있을 것이기 때문에 부호화의 의미가 사라지게 된다.

<그림 5>와 같이 처음에는 원 영상으로부터 받은 H.261 영상에서 GOB 정보와 움직임 벡터 등의 정보를 검출해낸 다음 한쪽으로는 전달 받은 H.261 영상 전체를 복호화하고 다른 한편으로는 해당부분의 움직임 벡터정보를 판단한다. 부가영상 영역을 참조하지 않는 경우에는 DCT 변환을 하기전의 데이터를 가지고 단순 삽입을 한다. 만약 원 H.261 영상의 움직임 벡터가 부가영상 부분을 참조할 경우에는 전체 복호화해두었던 영상을 인트라 블록으로 부호화하여 해당 되는 부분의 H.261 영상 대신 전송한다.



<그림 5> 제안한 알고리즘

4. 알고리즘 구현 및 결과

본 논문에서는 H.261의 기본 코덱으로 Stanford 대학의 PVRG-P64 CODEC 1.1을 사용하였고 H.261 파일을 읽어서 부가영상을 삽입하고 결과를 파일로 만드는 방법으로 테스트하였다. [4] 실제로 제안한 알고리즘이 사용되는 환경은 ISDN이나 LAN 상이다.

H.261은 이 경우에 전송되는 상황에 따라서 비트율을 조정하여 송신측과 수신측의 전송데이터가 달라진다. 하지만 본 논문의 테스트에서는 파일로 테스트를 하기 때문에 비트율 조정에 관련된 부분은 고려하지 않았다.

테스트는 펜티엄II 400MHz CPU, 128MHz 메모리, 윈도우 98 환경에서 수행되었다. 테스트로 사용된 영상은 길이가 150 Frame인 영상이 사용되었다.

제안한 알고리즘과 전체를 복/부호화하는 방법에 대해서 실험하였다. 제안한 방법으로 실험한 결과 <그림 4-4>의 경우처럼 깨끗하게 부가영상이 삽입이 되는 것을 확인할 수 있었다. 실험결과는 아래의 <표 1>과 같다. 제안된 방법을 사용하였을 경우에 평균 3.7배의 수행시간이 단축되었음을 알 수 있다.

<표 1> 실험 결과

파일명	전체 복/부호화		제안 알고리즘	
	수행 시간 (초)	FPS	수행시간 (초)	FPS
Bike	43.5	3.4	9.7	15.4
Sflows	53.2	2.8	18.9	7.9
Stennis	44.0	3.4	11.1	13.5

5. 결론 및 향후 연구방향

본 논문에서는 H.261 영상에 부가영상을 삽입하는 방안을 제안하였다. 부가영상을 삽입하기 위하여 우선 원시 H.261 영상을 복호화하여 저장하고, 부가 H.261 영상을 삽입하다가 통화영상의 손상이 올 경우에만 저장해두었던 복호영상을 이용하여 해당 매크로 블록을 부호화한다. 실험결과 전체를 복/부호화 하는 방법보다 향상된 성능을 나타냈다.

향후 연구과제로는 현재 파일로 구현한 시스템을 LAN 및 ISDN으로 확장하여 화상 통화 시스템을 구축하는 것이다. 또한, 본 논문을 H.263으로 확장하여, H.263이 IMT-2000의 영상으로 사용될 경우 IMT-2000상에서 부가영상 삽입 및 영상 부가서비스 개발의 가능성을 타진해 보는 것이다.

6. 참고 문헌

- [1] ITU-T Rec. H.261, Video Codec For Audio-visual Services At px64 kbits.
- [2] 정제창역, 그림으로 보는 최신 MPEG, 교보문고, 1995
- [3] K.R.Rao and J.J.Hwang, *Techniques & Standards For Image, Video & Audio Coding*, Prentice Hall, 1996
- [4] Andy C. Hung, "PVRG-P64 Codec 1.1 Document", Stanford University, 1993
- [5] Video-conferencing FAQ, <http://www.bitscout.com/faqtoc.htm>
- [6] Dan Kegel's ISDN Page, <http://alumni.caltech.edu/~dank/isdn/>
- [7] William B. Pennebaker and Joan L. Mitchell, *JPEG Still Image Data Compression Standard*, Chapman & Hall, 1992
- [8] Jerry D.Gibson, *Digital Compression for Multimedia : Principles and Standards*, Morgan Kaufmann Publishers, 1998
- [9] 정제창역, 그림으로 보는 응용 MPEG, 교보문고, 1997