

압축상태에서 MPEG2 P frame 을 H.263 P frame 으로 변환

양혁, 강의선, 임성민, 임영환
 송실대학교 컴퓨터학과
 e-mail : tip95@media.ssu.ac.kr

Transcoding of MPEG2 P frame to H.263 P frame in a Compressed Domain

Hyuck Yang, Eui-Sun Kang, Sung-Min Um, Young-Hwan Lim
 Dept. of Computer Science, Soongsil University

요 약

본 논문은 이 기종간의 표준 압축형식을 변환하는데 있어 실시간 변환이 가능하도록 하며 프레임의 특성을 이용하여 압축률을 좀 더 높여 보는데 그 목적을 두고 있다. 이 기종간의 표준 압축형식 변환으로 MPEG2 에서 H.263 으로의 변환을 시도하였다. MPEG2 에서 H.263 으로 변환 시 압축률을 높이기 위한 분석 결과 MPEG2 의 P 프레임내 인트라 매크로블록이 H.263 의 P 프레임의 인트라 매크로블록 수보다 많다는 것을 알 수 있었다. 따라서 본 논문에서는 P 프레임의 압축률을 높이기 위하여 P 프레임내의 인트라 매크로블록 변환 과정에 움직임 예측과정을 삽입하고 움직임 예측 과정의 복잡도를 줄이기 위하여 Guided 탐색 방법을 사용하여 MPEG2 P 프레임에 생성된 인트라 매크로블록의 개수를 줄일 수 있다는 것을 제안하였다.

1. 서론

최근에는 컴퓨터와 네트워크의 발달로 송신측에서 수신측으로 대용량 멀티미디어 데이터를 전송할 수 있게 되었다. 이를 위해 멀티미디어 표준 압축 형식들이 소개되고 있으며 송·수신측의 환경을 고려하기 위하여 QoS(Quality Of Service)를 받아들여 각 환경에 맞게 변환하는 트랜스코딩(Transcoding) 개념이 도입되고 있다.

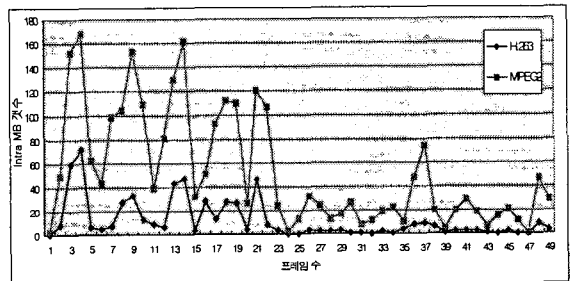
본 논문은 여러 트랜스코딩 기법 중 이 기종간의 멀티미디어 표준 압축 형식을 변환하기 위하여 MPEG2 에서 H.263 으로 변환을 시도해 보았다.

시간에 따라 변화하는 일련의 동영상상을 효율적으로 압축하기 위한 방법으로는 크게 시간적인 압축과 공간적인 압축으로 구분된다. 공간적인 압축은 DCT 와 양자화를 이용하여 시간적인 압축은 움직임 예측 과정을 이용한다. 움직임 예측은 움직임 벡터 정보와 그에 따른 매크로블록 정보를 갖는다. 이 때 블록 매칭 알고리즘을 사용하여 최적의 움직임 벡터 값을 구한 후 최적의 매크로 블록과 Threshold 를 비교하여 현 매크로 블록에 대한 움직임 벡터를 사용할지를 결정한다. 움직임 벡터를 사용할 경우에 현 매크로 블록은 시간적인 압축 내에서 인터 매크로블록(Inter MacroBlock)으로 결정되고 그렇지 않을 경우에는 공간적인 압축과 같은 인트라 매크로블록(Intra MacroBlock)으로 결정된다. 인트라 매크로블록으로 구분 짓는 이유는 움직임 벡터를 사용하여 더욱 많은 비트가 발생하면 오히려 데이터가 증가하기 때문에 전체 압축률을 낮게 하는 원인이 된다.

본 논문은 H.263 으로 변환과정에서 압축률을 좀 더 높이기 위하여 MPEG2 와 H.263 의 시간적 압축 내에서

의 인트라 매크로 블록의 수를 비교하여 보았다.

[그림 1]은 MPEG2 와 H.263 의 P 프레임 내의 인트라 매크로블록 수를 비교하기 위하여 전체 50 개의 프레임 중 첫 프레임을 제외한 나머지 49 개의 P 프레임을 생성하여 보인다.



[그림 1] H.263 과 MPEG2 의 프레임별 인트라 매크로블록 개수

[그림 1]에서 시간적 압축에 있어서 인트라 매크로블록과 인터 매크로블록을 결정하는 Threshold 값이 다르므로 각 P 프레임마다 MPEG2 인트라 매크로블록이 H.263 보다 많으므로 H.263 의 압축률이 높음을 보인다. 본 논문은 MPEG2 에서 H.263 으로 변환 시 압축률을 좀 더 높이고 실시간 전송이 가능한 방법을 제시할 것이다.

본 논문은 다음과 같은 순서로 진행된다. 제 2 장에서는 압축된 영역에서 MPEG2 P 프레임을 H.263 P 프레임으로 변환하는 방법을 제안하고 제 3 장에서는 가이드 탐색 알고리즘을 소개한다. 그리고 4 장에서는 3 장의 가이드 탐색 알고리즘들에 대하여 평가하며 마치

막으로 5장에서 결론을 짓는다.

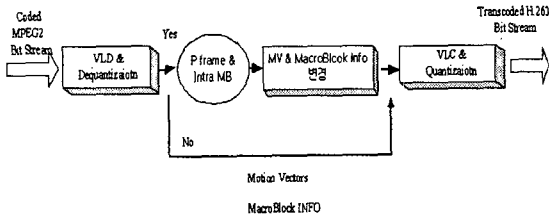
2. 압축된 영역에서의 매크로블록 형식 변환

2.1 압축된 영역에서 매크로블록 형식 변환 방법

MPEG2 P 프레임내의 인트라 매크로블록을 H.263 P 프레임의 인트라 매크로블록으로 변환하는 가장 간단한 방법은 입력으로 들어오는 압축된 MPEG2 P 프레임을 모두 복원하고 H.263의 움직임 예측과정을 수행하여 H.263의 Threshold 의해 모든 매크로블록 형식을 재 설정하는 것이다. 하지만 이 방법은 프레임에 대한 움직임 예측 과정이 모두 수행되기 때문에 전체 변환 시간에 많은 비용 소모를 가진다.

이러한 문제점을 해결하기 위하여 압축된 상태에서 MPEG2 P 프레임내의 인트라 매크로블록을 H.263 P 프레임내의 인트라 매크로블록으로 변환해 보았다.

아래 [그림 2]는 압축 상태에서의 MPEG2 P 프레임을 H.263 프레임으로 변환 과정을 그린 그림이다.



[그림 2] 압축 상태에서 P 프레임 변환 과정

복원된 비트 스트림의 프레임 형식이 I 프레임이거나 P 프레임이면서 인트라 매크로블록인 경우에는 움직임 벡터와 매크로블록 정보를 재 사용하여 H.263의 양자화를 수행하고 VLC 과정을 수행하여 H.263으로 변환된 비트 스트림을 생성한다. 그리고 복원된 비트 스트림의 프레임 형식이 P 프레임이면서 인트라 매크로블록인 경우에는 인트라 매크로블록의 형식과 새로운 움직임 벡터를 재 설정한다.

이웃하는 움직임 벡터를 이용하는 방법으로는 다음과 같은 방법들이 있다.

이웃하는 움직임 벡터의 평균 방향성 또는 움직임 벡터 크기에 따른 가중치 적용 이웃하는 움직임 벡터 중 최대, 최소 혹은 중간값을 이용하는 방법

[표 1] 새로운 움직임 벡터를 유추하기 위해 이웃하는 움직임 벡터를 이용하는 방법

2.2 문제점

압축 영역에서의 변환방법은 복원시간과 압축시간에 소요되는 움직임 예측 과정이 없으므로 전체적으로 처리 시간이 많이 줄어든다. 하지만 움직임 벡터는 이전 프레임에서 가장 유사한 블록을 찾지 못하기 때문에 움직임 벡터를 정확히 찾지 못할 경우에는 화질의 손상을 우려하지 않을 수 없다

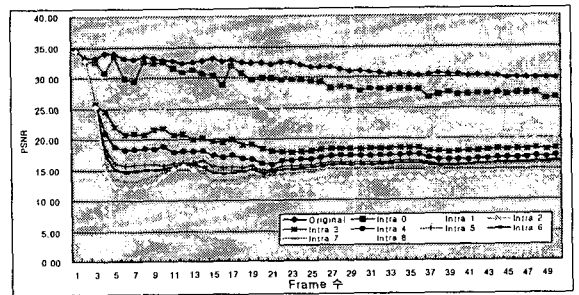
[그림 3]은 변환하기 이전의 이미지이고 그림 4는

[표 1]에서 제시한 방법 중 이웃하는 움직임 벡터들의 평균을 이용하여 변환하였을 때의 이미지를 보여 준다. 그리고 [그림 5]는 [표 1]에서 제시하는 방법들 중 이웃하는 움직임 벡터의 평균을 이용하여 인트라 매크로블록을 인트라 매크로블록으로 변환하였을 때의 PSNR을 보여주는 그림이다.



[그림 3] 변환 전 이미지

[그림 4] 변환 후 이미지

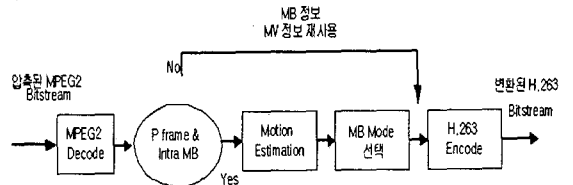


[그림 5] 이웃하는 움직임 벡터의 평균을 이용한 PSNR

위 [그림 4]에서 볼 수 있듯이 블록 깨짐 현상이 보이는 이유는 새롭게 계산된 움직임 벡터가 이웃하는 움직임 벡터의 방향성 혹은 움직임의 정도와 는 다르게 다른 특정한 곳으로부터의 움직임 벡터 값을 가지며 그에 따른 양자화 계수 값들 또한 조절해 줘야 하는데 양자화 계수 값들은 이웃하는 매크로블록의 양자화 계수 값들과 독립적이기 때문이다. 그리고 복원시 처음 복원된 프레임에 블록 깨짐 현상이 있는 경우 복원한 프레임 수가 증가할수록 사람이 인식 할수 없을 정도의 블록 깨짐 현상이 뚜렷하게 나타난다.

3. 이웃하는 움직임 벡터를 이용한 가이드 탐색 알고리즘

이 장에서는 압축영역에서의 매크로블록변환 시 수행되는 시간적인 복잡도는 다소 높지만 움직임 예측과정을 삽입하여 화질문제를 해결해 보고자 한다.



[그림 6] 움직임 예측과정을 이용한 P 프레임 변환 과정

위의 [그림 6]은 움직임 예측과정을 이용한 인트라 매크로블록을 인트라 매크로블록으로 변환과정을 나타낸다. 여기서 볼 수 있듯이 한 프레임에 대한 모든 매크로

·블록에 대하여 움직임 예측 과정을 수행하는 것이 아니라 MPEG2 P 프레임내의 인트라 매크로블록에 대해서만 움직임 예측과정을 수행함으로써 한 프레임에 대한 움직임 예측과정에 비하여 전체 변환 시간을 줄일 수 있다.

또한 본 논문은 움직임 예측에서 변환 시간을 줄이기 위한 방법으로 여러 탐색 알고리즘 중 3 단계 탐색 알고리즘을 개선하여 변환을 시도하였고 개선 방법으로 3 단계 탐색 알고리즘의 1 단계 과정을 줄임으로 전체 처리 시간을 줄이고자 한다.

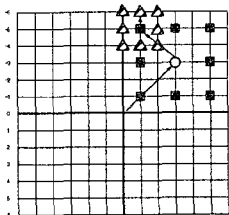
3.1.2 단계 탐색 알고리즘(Two Step Search Algorithm)

3 단계 탐색 알고리즘은 1 단계에서 움직임 방향이 결정된다. 따라서 움직임의 방향이 결정된다면 1 단계를 수행하지 않아도 된다. 2 단계 탐색 알고리즘은 움직임의 방향을 결정하기 위하여 변환하고자 하는 인트라 매크로블록을 중심으로 이웃하는 8 개의 움직임 벡터를 이용하였다.

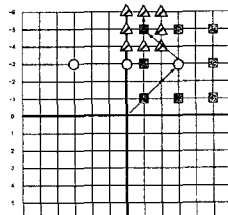
[그림 7]에서 보듯이 X, Y 모두 양의 방향으로 움직였을 경우 X, Y 가 모두 양수인 부분으로 이동하고 현 위치를 중심으로 3 단계 탐색 알고리즘의 2 번째 단계를 수행한다. 만약 이웃하는 움직임 벡터의 음과 양의 방향의 개수가 동일할 경우 즉 결과 값이 0 일 경우에는 중앙을 중심으로 탐색하며 3 단계 탐색 알고리즘의 1 단계부터 수행한다.

[알고리즘 1] 2 단계 탐색 알고리즘

- Step 1.** 이웃하는 움직임 벡터를 이용하여 움직임 방향을 예측한다. 만약 움직임 방향이 0 일 경우에는 3 단계 탐색 알고리즘의 Step 1 을 수행한다.
- Step 2.** 예측된 방향을 중심으로 전체 탐색 영역의 $1/2^2$ 인 주변 8 개의 위치를 설정하고 예측된 방향을 포함한 전체 9 개의 위치에 대하여 Cost Function 이 가장 작은 값을 갖는 위치를 찾는다.
- Step 3 .** Step 2에서 찾은 Cost Function 이 가장 작은 위치를 중심으로 주변 8 개의 매크로블록에서 Cost Function 이 가장 작은 값을 갖는 위치를 찾는다.



[그림 7] 2 단계 탐색 알고리즘 경로



[그림 8] 축 분할 탐색 알고리즘의 경로

3.2 축 분할 탐색 알고리즘

2 단계 탐색 알고리즘은 3 단계 탐색 알고리즘에 비해 가장 유사한 매크로블록을 찾는 정확성이 떨어지기 때문에 압축률을 크게 향상시킬 수 없다

[그림 8]은 축 분할 탐색 알고리즘의 경로로 2 단계 탐색 알고리즘에 좀더 정확성을 부여하기 위한 것으로서

1 단계에 비교되는 횟수의 범위를 넓힌 알고리즘이다.. 다음은 축 분할 탐색 알고리즘의 전체적인 과정을 각 단계별로 기술한 것이다.

[알고리즘 1] 축 분할 알고리즘

- Step 1.** 이웃하는 움직임 벡터를 이용하여 움직임 방향을 예측하고 전체 탐색 영역의 1/2 의 위치에 있는 수평 또는 수직방향으로 위치를 3 개 설정한다. 그리고 이 3 개의 위치 중 Cost Function 이 가장 작은 위치를 설정한 후 다음 단계를 수행한다. 만약 예측된 움직임 방향이 0 일 경우에는 3 단계 탐색 알고리즘의 Step 1 단계를 수행한다.
- Step 2.** Step 1 에서 결정된 위치를 중심으로 전체 탐색 영역의 $1/2^2$ 인 주변 8 개의 위치를 설정하고 각 위치에 대하여 Cost Function 이 가장 작은 값을 갖는 위치를 찾는다.
- Step 3.** Step 2 에서 찾은 Cost Function 이 가장 작은 위치를 중심으로 주변 8 개의 위치에서 Cost Function 이 가장 작은 값을 갖는 위치를 찾는다.

4. 실험결과

4.1 실험환경

CPU Intel Pentium III와 RAM 이 256Mbyte, 운영체제가 Microsoft Windows 2000 Server 인 환경상에서 움직임이 있는 화면을 CIF 크기로 50 프레임 Capture 하여 첫 번째 프레임은 I 프레임으로 하고 나머지 프레임을 P 프레임으로 하여 MPEG2 압축과정을 통하여 압축하고 제안한 알고리즘을 사용하여 H.263 으로 변환 하였다.

4.2 성능평가

여기서는 2 단계 탐색 알고리즘과 축 분할 탐색 알고리즘을 3 단계 탐색 알고리즘과 비교함으로써 제안한 알고리즘들의 성능을 평가 하고자 한다.

성능평가의 방법으로는 전체 프레임에 대한 비교횟수와 전체 압축률, 마지막으로 화질을 비교하고자 한다 그리고 이웃하는 매크로블록의 정보가 모두 인트라 매크로블록일 경우뿐만 아니라 인트라 매크로블록이 1~ 8 개 존재할 경우를 포함하여 비교하였다. 제안한 가이드 알고리즘들은 압축된 MPEG2 P 프레임 내에서 인트라 매크로블록일 경우에만 적용하였다.

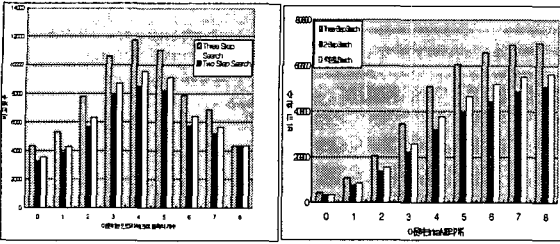
4.2.1 비교 횟수

비교횟수는 각 단계별 최적의 Cost Function 을 찾기 위해 비교되는 횟수를 나타낸다.

[그림 9]는 MPEG2 P 프레임내의 인트라 매크로블록을 H.263 의 인트라 매크로블록으로 변환하고자 할 때 3 단계 탐색 알고리즘, 2 단계 탐색 알고리즘, 축 분할 탐색 알고리즘 방법을 사용하였을 때의 전체 프레임에 대한 비교 횟수를 나타낸다.

[그림 10]은 탐색 알고리즘에 따른 매크로블록별 비교한 그림이다. 전체적으로 3 단계 탐색 알고리즘일 경우 비교 횟수가 가장 많고 다음으로 축 분할 탐색 알

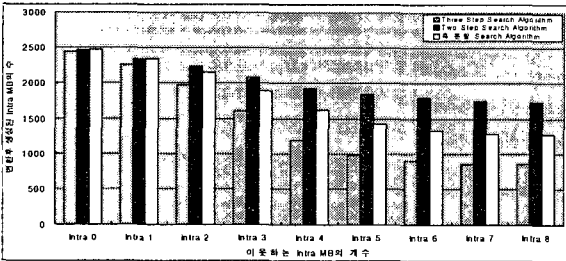
고리즘, 그리고 2 단계 탐색 알고리즘일 경우가 가장 적은 것을 볼 수 있다.



[그림 9] 탐색 알고리즘에 따른 이동하는 [그림 10] 탐색 알고리즘에 따른 비교 횟수 인트라 매크로블록개수 별 비교횟수

4.2.2 압축률

[그림 11]은 제안한 가이드 탐색 알고리즘과 3 단계 탐색 알고리즘을 사용하였을 때 전체 프레임에 대한 압축률을 그림으로 표현한 것이다.



[그림 11] Guided 탐색 알고리즘을 적용 후 생성된 인트라 매크로블록의 수

제안한 가이드 탐색 알고리즘은 인트라 매크로블록일 경우에만 적용되므로 압축률이 높지 않다. 2 단계 탐색 알고리즘인 경우에는 1 단계에서 이미 탐색 방향이 설정되었기 때문에 H.263의 Threshold를 비교했을 때 3 단계 탐색 알고리즘에 비하여 인트라 매크로블록의 개수가 많다. 그리고 축분할 탐색 알고리즘은 2 단계 탐색 알고리즘에 비하여 3 번의 비교를 더하기 때문에 2 단계 탐색 알고리즘보다 인트라 매크로블록의 개수가 적지만 3 단계 탐색 알고리즘에 비하여 인트라 매크로블록의 개수가 많다.

4.2.2 화질평가

본 논문은 화질을 평가하기 위하여 PSNR을 사용하여 MPEG2로 압축되기 이전의 이미지와 가이드 탐색 알고리즘을 사용하여 변환한 후 복원했을 때의 이미지를 비교하였다

	3 단계 탐색	2 단계 탐색	축분할 탐색
인트라 0	31.60	31.61	31.61
인트라 1	31.53	31.55	31.56
인트라 2	31.43	31.53	31.51
인트라 3	31.30	31.48	31.44
인트라 4	31.19	31.44	31.37
인트라 5	31.10	31.42	31.31
인트라 6	31.07	31.41	31.29
인트라 7	31.06	31.40	31.27
인트라 8	31.06	31.40	31.27

[표 3] 각 탐색 알고리즘 별 PSNR

위의 [표 3]에서 볼 수 있듯 인트라 매크로블록의 개수에 영향을 받기 때문에 전체적으로 3 개의 탐색 알고리즘이 큰 차이를 보이지 않지만 2 단계 탐색 알고리즘일 경우 화질이 가장 좋은 것을 확인할 수 있다.

5. 결론

본 논문은 MPEG2 P 프레임을 H.263 P 프레임으로 변환하는 방법으로서 먼저 압축영역에서의 변환방법을 시도해 보았으나 심각한 화질문제가 발생하여 변환 시 움직임 예측과정을 삽입하였고 움직임 예측 시 소요되는 복잡도를 줄이기 위하여 변환하고자 하는 매크로블록의 이웃 하는 움직임 벡터를 이용한 가이드 탐색 알고리즘(2 단계 탐색 알고리즘 & 축분할 탐색 알고리즘)을 제안하였다. 그리고 실험결과에서 가이드 탐색 알고리즘이 기존 알고리즘에 비해 적은 비교횟수를 가지는 것을 확인함으로써 MPEG2 P 프레임을 H.263 P 프레임으로 변환하는 데 있어 전체 처리시간을 줄일 수 있었다.

참고문헌

- [1] Guy Cote, Berna Erol, Michael Gallant, and Faouzi Kossentini, "H.263+: VIDEO CODING AT LOW BIT RATES," IEEE Transactions on Circuits and Systems for Video Technology, Vol. 8, No. 7, pp.848-866, November 1998.
- [2] R.J. Clarke, *Digital Compression of Still Image and Video*, ACADEMIC PRESS, 1995.
- [3] K.R.Rao, J.J. Hwang, *Techniques & Standards for Image Video & Audio Coding*, Prentice Hall, 1996.
- [4] Borko Furht and Joshua Greenberg, *Motion Estimation Algorithms for Video Compression*, Kluwer Academic Publishers, 1997.
- [5] Martyn J.Riley and Iain E.G.Richardson, *Digital Video Communications*, Artech House, 1997.
- [6] ITU-T Standardization Sector of ITU, Video Coding for Low Bitrate Communication, Draft ITU-T Recommendation H.263 Version 2, September 1997.
- [7] Information Technology-Generics Coding of Moving Picture and Associated Audio, ISO/IEC 13818-2 Committee Draft(MPEG-2)
- [8] Soam Acharya, Brain C. Smith, "Compressed Domain Transcoding of MPEG," ICMCS 1998: 295-304.
- [9] G. Keesman, R. Hellinghuizen, F. Hoeksema, and G. Heideman, "Transcoding MPEG bitstream," Signal Processing: Image Communication, vol. 8, September 1996.
- [10] T. Shanableh and M. Ghanbari, "Heterogeneous video transcoding MPEG:1,2 to H.263," in International Packet Video Workshop, (New York City, NY), April 1999.
- [11] N. Feamster and S.J. Wee, "An MPEG-2 to H.263 Transcoder," SPIE Voice, Video, and Data Communications Conference, Boston, MA, September 1999.
- [12] Oliver Werner, "Requantization for Transcoding of MPEG-2 Intraframes," IEEE transaction on image processing, Vol. 8, No. 2, February 1999.
- [13] N. Bjork and C. Christopoulos, "Transcoder architecture for video coding," IEEE Trans. Consumer Electron., vol. 44, pp. 88-98, Feb. 1998
- [14] B. Shen, I.K. Sethi and V. Bhaskaran, "Adaptive motion Vector resampling for compressed video down-scaling," Proc. Of IEEE Int. Conf. On Image Processing, Oct. 1997