

압축상태에서 MPEG2 P 프레임을 H.263 P 프레임으로 변환하기 위한 가이드 탐색 방법 연구

엄 성 민[†] · 강 의 선[†] · 임 영 환^{††} · 황 재 각^{†††}

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 변환으로는 압축률이 낮은 MPEG2에서 압축률이 높은 H.263으로의 변환을 시도하였다. MPEG2에서 H.263으로 변환한 후 좀 더 압축률을 높이기 위하여 분석한 결과 MPEG2의 P 프레임내의 인트라 매크로블록 수가 H.263의 P 프레임의 인트라 매크로블록보다 많다는 것을 알 수 있었다. 인트라 매크로블록은 P 프레임을 생성하는 과정에서 움직임 예측 과정의 결과치가 임계치에 미치지 못했을 경우에 생성되는 것으로써 공간적인 압축 과정을 통하여 압축되기 때문에 P 프레임의 압축률에 큰 영향을 준다. 따라서 본 논문에서는 P 프레임의 압축률을 높이기 위하여 MPEG2 P 프레임에 생성된 인트라 매크로블록의 개수를 줄여보았다. 이를 위한 해결방법으로써 P 프레임내의 인트라 매크로블록 변환 과정에 움직임 예측 과정을 삽입하여 변환하였다. 그리고 움직임 예측 과정의 복잡도를 줄이기 위하여 복원과정에서 알 수 있는 이웃하는 매크로블록의 정보와 움직임 벡터 정보를 이용하여 가이드 탐색 방법을 제안하였다. 실험 결과 MPEG2에서 H.263으로 변환이 실시간으로 이루어짐을 확인하였다.

A Study on the Guided Search Method for Transcoding MPEG2 P frame to H.263 P frame in a Compressed Domain

Sungmin Um[†] · Euseon Kang[†] · Younghwan Lim^{††} · Jaegak Hwang^{†††}

ABSTRACT

The purpose of the paper is to enable a format transcoding between a heterogeneous compression format in a real time, and to enhance the compression ratio using characteristics of the compressed frame. In this paper, for the heterogeneous format transcoding, we tried to transcode from MPEG2 having a lower compression ratio to H.263 having a higher compression ratio. After analyzing MPEG 2 bit stream and H.263 bit stream of the same original video, we found that the number of intra coded macro blocks in MPEG 2 data is much higher than the number of the intra coded macro blocks in H.263 data. In the process of P frame generation, a intra coded macro block is generated when a motion estimation value representing the similarity between the previous frame and current frame does not meet a threshold. Especially the intra coded macro block has a great impact on the compression ratio. Hence the paper, we tried to minimize the number of intra coded macro blocks in H.263 data stream which is transcoded from MPEG 2 in a compressed domain. For the purpose, we propose a guided search method for transcoding the INTRA coded block into INTER coded block using the information about motion vectors surrounding the intra macro block in order to minimize the complexity of the motion estimation process. The experimental results show that the transcoding of MPEG 2 into H.263 can be done in a real time successfully.

키워드 : 포맷 트랜스코딩(Format Transcoding), 가이드 탐색 알고리즘(Guided search Algorithms)

1. 서 론

최근에는 컴퓨터와 네트워크의 발달로 송신측에서 수신측으로 다양한 데이터를 전송할 수 있게 되었다. 이중 대용량의 멀티미디어 데이터를 네트워크를 통하여 전송하기 위하여 멀티미디어 표준 압축 형식들이 소개되고 있으며 송신측과 수신측의 환경을 고려하기 위하여 송신측과 수신측의 QoS(Quality Of Service)를 받아들여 각 환경에 맞게 변환하는

트랜스코딩(Transcoding)이란 개념이 도입되고 있다. 트랜스코딩의 종류로는 이 기종간의 멀티미디어 표준 압축 형식을 변환하는 방법[11], 이미지의 크기를 줄이는 방법[13, 14], 네트워크의 지연현상을 고려한 비트율을 줄이는 방법[13] 등이 있다. 본 논문은 이들 트랜스코딩 중 이기종간의 멀티미디어 표준 압축 형식을 변환하기 위하여 MPEG2에서 H.263으로 변환을 시도해 보았다.

MPEG2는 그 응용분야가 디지털 매체 저장등에 사용되기 때문에 화상회의를 위한 H.263에 비하면 동일 프레임 수라 할지라도 압축률은 MPEG2가 낮다. 압축률을 높이는 방법으로는 양자화를 이용하여 비트율을 조절하는 방법[9,

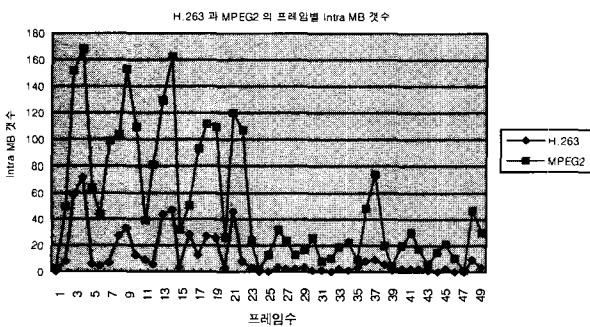
† 준 회 원 : 송실대 대학원 컴퓨터공학
†† 종신회원 : 송실대학교 미디어학부 교수
††† 정 회 원 : 한국전자통신연구원 책임기술원
논문접수 : 2002년 2월 26일, 심사완료 : 2002년 10월 9일

13], 입력되는 일련의 영상의 크기를 줄이는 방법[8, 13], 프레임 레이트(Rate)를 조절하는 방법[11], 그리고 프레임의 특성을 이용한 방법[10, 11] 등이 있다.

시간에 따라 변화하는 일련의 동영상을 효율적으로 압축하기 위한 방법으로는 크게 시간적인 압축과 공간적인 압축으로 구분된다. 공간적인 압축은 DCT와 양자화를 이용하며 시간적인 압축은 움직임 예측 과정을 이용한다. 시간적인 압축에서 움직임 예측이란 블록 매칭 알고리즘을 이용하여 이전 영상과 현재 영상사이에서 가장 비슷한 블록을 찾는 작업으로써 이전 영상에서 움직임의 정도를 나타내는 움직임 벡터정보와 그에 따른 매크로블록 정보를 갖는다. 이 때 블록 매칭 알고리즘을 사용하여 최적의 움직임 벡터값을 구한 후 최적의 매크로블록과 Threshold를 비교한다. 즉 현 매크로블록에 대하여 움직임 벡터를 사용할지 사용하지 않을지를 결정한다. 여기서 움직임 벡터를 사용할 경우에는 현 매크로블록은 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 때문이다. 따라서 시간적 압축 내에서 인트라 매크로블록은 전체 압축률을 낮게 하는 원인이 된다.

본 논문은 H.263으로 변환과정에서 H.263의 압축률을 좀 더 높이기 위하여 MPEG2와 H.263의 시간적 압축 내에서의 인트라 매크로블록의 수를 비교하여 보았다.

(그림 1)은 MPEG2와 H.263의 P 프레임 내의 인트라 매크로블록에 대한 수를 비교하기 위하여 전체 50개의 프레임 중 첫 프레임을 제외한 나머지 49개의 P 프레임을 생성하여 인트라 매크로블록의 수를 비교한 그림이다.



(그림 1) H.263과 MPEG2의 프레임별 인트라 매크로블록 개수

위 그림에서 볼 수 있듯이 각 P 프레임마다 MPEG2의 인트라 매크로블록이 H.263 보다 많음을 볼 수 있다. 그 이유는 MPEG2와 H.263의 시간적 압축에 있어서 인트라 매크로블록과 인터 매크로블록을 결정하는 Threshold 값이 다르기 때문이다. 따라서 MPEG2와 H.263의 시간적 압축과정에 의해

생성된 프레임에 비교해 보면 H.263이 MPEG2에 비해 압축률이 높은 것을 확인할 수 있다.

본 논문은 MPEG2에서 H.263으로 변환시 압축률을 좀 더 높이고 실시간 전송이 가능하도록 하고자 한다. 이 문제를 해결하기 위한 방법으로써 P 프레임의 특성을 이용하여 입력으로 들어오는 MPEG2 P 프레임내의 인트라 매크로블록을 H.263의 인터 매크로블록으로 변환함으로써 실시간 전송에 적합하도록 MPEG2의 인트라 매크로블록의 개수를 줄이고 압축률을 높이며 변환 후 가급적 화질의 손상이 없도록 하고자 한다. MPEG2 인트라 매크로블록을 H.263 인터 매크로블록으로 변환하는 방법으로는 압축영역에서 변환하는 방법과 움직임 예측을 적용한 방법을 이용한다.

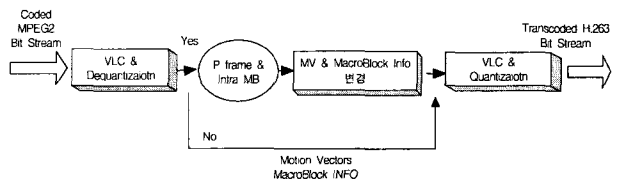
본 논문은 다음과 같은 순서로 진행된다. 제 2장에서는 압축된 영역에서 MPEG2 P 프레임을 H.263 P 프레임으로 변환하는 방법을 제안하고 제 3장에는 변환과정에 있어 움직임 예측과정에 포함되는 탐색 알고리즘을 개선한 가이드 탐색 알고리즘들을 소개한다. 그리고 제 4장에서는 3장에서 제안한 가이드 탐색 알고리즘들에 대하여 평가하며 마지막으로 제 5장에서 결론을 짓는다.

2. 압축된 영역에서의 매크로블록 형식 변환

2.1 압축된 영역에서 매크로블록 형식 변환 방법

MPEG2의 P 프레임내의 인트라 매크로블록을 H.263 P 프레임의 인터 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 비용을 소모할 뿐만 아니라 실시간 전송문제를 포함한다.

이러한 문제점을 해결하기 위하여 움직임 예측 과정을 수행하지 않으면서 실시간 전송에 적합하도록 하기 위하여 압축된 상태에서 MPEG2 P 프레임내의 인트라 매크로블록을 H.263 P 프레임내의 인터 매크로블록으로 변환해 보았다. 다음 그림은 압축 상태에서의 MPEG2 P 프레임을 H.263 P 프레임으로 변환하는 과정을 간략하게 그린 그림이다.



(그림 2) 압축 상태에서 P 프레임 변환 과정

MPEG2로 압축된 비트 스트림을 입력받아 역 양자화와 VLD를 수행하는 과정에서 현재 복원되고 있는 프레임 형

식과 매크로블록의 정보를 알 수 있다. 복원된 비트 스트림의 프레임 형식이 I 프레임이거나 P 프레임이면서 인터 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 매크로블록을 인터 매크로블록으로 변환하고 그에 따라 매크로블록의 형식과 새로운 움직임 벡터를 재 설정한다.

인트라 매크로블록을 인터 매크로블록으로 변환하기 위해 중점적으로 다뤄야 하는 부분이 움직임 벡터이다. 움직임 벡터는 프레임 상에서 가장 유사한 블록을 찾아 움직임의 정도를 나타낸 것으로서 이전에 복원된 프레임을 참조하여 움직임 예측 과정을 통하여 구할 수 있다. 하지만 움직임 예측 과정을 포함하지 않고 압축된 상태에서 새로운 움직임 벡터를 알아내기 위해서는 변환하고자 하는 인트라 매크로블록 주변의 이웃하는 8개의 매크로블록 정보와 움직임 벡터를 이용할 수 있다. 이웃하는 움직임 벡터를 이용하는 방법으로는 다음과 같은 방법들이 있다.

〈표 1〉 새로운 움직임 벡터를 유추하기 위해 이웃하는 움직임 벡터를 이용하는 방법

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

이웃하는 움직임 벡터의 평균을 이용한 방법은 가장 간단한 방법으로서 이웃하는 움직임 벡터의 방향성이나 움직임의 정도를 고려하지 않고 움직임 벡터의 평균으로써 새로운 움직임 벡터를 구하는 방법이다. 움직임 벡터는 방향성과 움직임의 정보를 나타내 주고 있다. 이웃하는 움직임 벡터들이 동일한 방향으로 움직인 경우가 많거나 움직임이 큰 경우 변환하고자 하는 매크로블록 또한 유사하게 움직일 것이라고 예측 할 수 있다. 따라서 동일한 방향으로 움직인 경우와 움직임이 큰 경우에 가중치를 많이 부여하고 그렇지 않은 경우 가중치를 적게 부여할 수 있다. 또한 이웃하는 움직임 벡터 중 최대, 최소값 혹은 중간값만을 이용할 수 있다.

위 방법들을 적용하는데 있어 주변의 매크로블록이 모두 움직임 벡터를 가지는 인터 매크로블록인 경우와 그렇지 않은 경우 적용해 볼 수 있다.

2.2 문제점

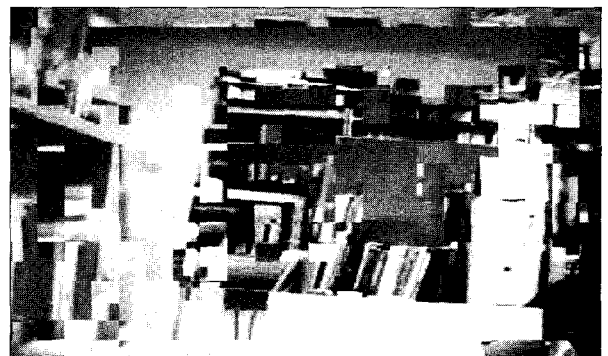
압축 영역에서의 변환방법은 복원시간과 압축시간에 소요되는 움직임 예측 과정이 없으므로 전체적으로 처리 시간이 많이 줄어든다. 하지만 움직임 벡터는 이전 프레임에서 가장 유사한 블록을 찾기 때문에 움직임 벡터를 정확히

찾지 못할 경우에는 화질의 손상을 우려하지 않을 수 없다.

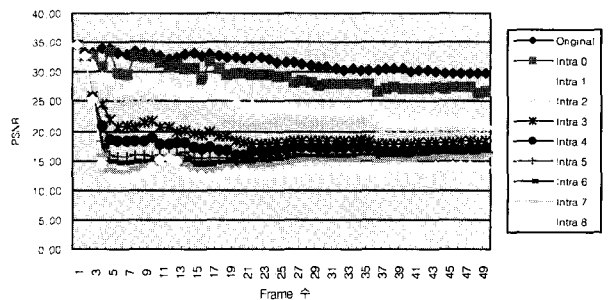
(그림 3)은 변환하기 이전의 이미지이고 (그림 4)는 <표 1>에서 제시한 방법 중 이웃하는 움직임 벡터들의 평균을 이용하여 변환하였을 때의 이미지를 보여준다. 그리고 (그림 5)는 <표 1>에서 제시하는 방법들을 중 이웃하는 움직임 벡터의 평균을 이용하여 인트라 매크로블록을 인터 매크로블록으로 변환하였을 때의 PSNR을 보여주는 그림이다. 여기서 인트라 0~인트라 8까지는 이웃하는 매크로블록이 인트라 매크로블록인 경우를 나타내주는 것으로써 예를 들어 "인트라 4"란 이웃하는 매크로블록 중 인트라 매크로블록 이 4개 미만인 경우를 의미한다.



(그림 3) 변환 전 이미지



(그림 4) 변환 후 이미지



(그림 5) 이웃하는 움직임 벡터의 평균을 이용한 PSNR

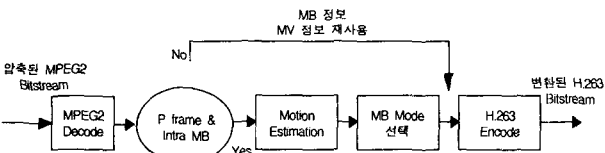
위 (그림 5)에서 이웃하는 매크로블록의 개수가 많을수록 화질 저하 현상이 나타나는 것을 PSNR을 통하여 볼 수 있

다. 이는 다른 방법들을 사용하였을 경우에도 확인 할 수 있었다. 이를 육안으로 확인하면 (그림 4)에서 볼 수 있듯이 블록 깨짐 현상이 뚜렷이 보이는 것을 볼 수 있었다. 그 이유는 새롭게 계산된 움직임 벡터가 이웃하는 움직임 벡터의 방향성 혹은 움직임의 정도와는 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 복원 시 처음 복원된 프레임에 블록 깨짐 현상이 있는 경우 다음 복원할 프레임은 블록 경계선이 있는 이전 프레임을 참조하여 복원하기 때문에 복원한 프레임 수가 증가할수록 사람이 인식 할 수 없을 정도의 블록 깨짐 현상이 더욱 뚜렷하게 나타난다.

3. 이웃하는 움직임 벡터를 이용한 가이드 탐색 알고리즘

압축 영역에서 변환을 할 경우 새롭게 계산된 움직임 벡터가 정확한 부분을 찾지 못하거나 양자화 계수를 고려하지 않을 경우 화질 문제가 발생하는 것을 제 2장에서 확인할 수 있었다. 따라서 이 장에서는 압축영역에서의 매크로블록 변환 시 수행되는 시간적인 복잡도는 다소 높지만 움직임 예측 과정을 삽입하여 화질문제를 해결해 보고자 한다.

다음 그림은 움직임 예측 과정을 이용하여 인트라 매크로블록을 인터 매크로블록으로 변환하는 과정을 간략하게 나타낸 그림이다.



(그림 6) 움직임 예측 과정을 이용한 P 프레임 변환 과정

MPEG2에 의해 압축된 비트 스트림은 인트라 매크로블록을 인터 매크로블록으로 변환하기 위하여 움직임 예측과정을 수행하여 복원하고 현재 프레임 형식과 매크로블록의 정보를 알아낸다. 만약 복호화한 프레임의 정보가 I 프레임이거나 P 프레임 내의 인터 매크로블록인 경우에는 매크로블록 정보와 움직임 벡터 정보를 재사용하고 H.263의 양자화와 VLC 과정을 수행하여 H.263 비트 스트림을 생성한다. 만약 복원된 프레임이 P 프레임이고 P 프레임내의 인트라 매크로블록인 경우에는 움직임 예측과정을 통하여 움직임 벡터와 매크로블록의 정보를 수정하고 H.263의 압축 과정을 수행한다.

여기서 볼 수 있듯이 한 프레임에 대한 모든 매크로블록에 대하여 움직임 예측 과정을 수행하는 것이 아니라 MPEG2 P 프레임내의 인트라 매크로블록에 대해서만 움직임 예측 과정

을 수행함으로써 한 프레임에 대한 움직임 예측 과정에 비하여 전체 변환 시간을 줄일 수 있다. 또한 본 논문은 움직임 예측과정에서 수행되는 탐색 알고리즘을 개선한 가이드 탐색 알고리즘을 제안하여 움직임 예측과정에 소요되는 복잡도를 좀 더 줄이고자 한다.

지금까지 움직임 예측과정에서 많은 시간적 비용을 최소화하기 위한 많은 탐색 알고리즘[4]들이 소개되어 왔다.

[4]에서 보여주는 많은 탐색 알고리즘 들 중 본 논문은 3단계 탐색 알고리즘(Three Step Search Algorithm)을 개선하여 MPEG2 P 프레임을 H.263 P 프레임으로 변환하는데 좀 더 시간적인 비용을 줄여 보고자 한다. 그 이유는 3단계 탐색 알고리즘이 Full 탐색 알고리즘에 비해 처리 시간이 빠르며 중앙 편중적인 특성(Center-Biased characteristics)을 가지고 있기 때문이다.

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

3.1 2단계 탐색 알고리즘(Two Step Search Algorithm)

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

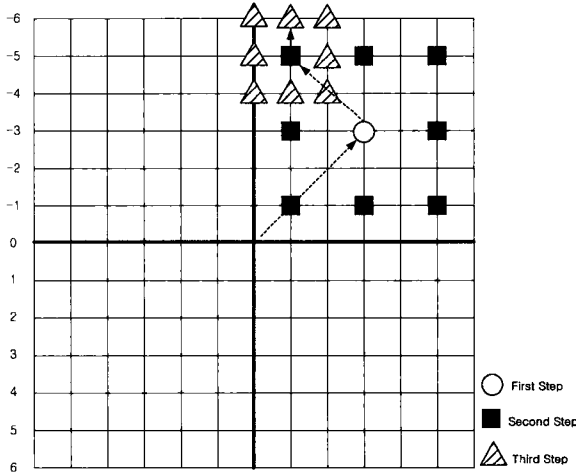
<표 2> 이웃하는 움직임 벡터를 사용하여 움직임의 방향을 결정하는 예

15, 7 (①)	15, -1 (②)	15, 1 (③)
-15, 5 (⑧)	인트라 매크로블록	15, 13 (④)
15, 9 (⑦)	15, 9 (⑥)	15, -3 (⑤)

<표 2>는 인트라 매크로블록을 중심으로 이웃하는 매크로블록이 모두 인터 매크로블록 이며 그에 따른 움직임 벡터를 (X, Y)로 표현한 것이다. 위 표의 X방향으로 볼 때 양의 방향으로 움직인 X(1, 2, 3, 4, 5, 6, 7)는 7개이며 음의 방향으로 움직인 X(8)는 1개이다. 그리고 양의 방향으로 움직인 Y(1, 3, 4, 6, 7, 8)는 모두 6개이며 음의 방향으로 움직인 Y(2, 5)는 2개이다. 즉 이웃하는 움직임 벡터를 볼 때 전체적으로 X, Y 모두 양의 방향으로 움직였다고 볼 수 있다.

이점을 이용하여 3단계 탐색 알고리즘의 3단계 중 첫 번째 단계를 줄임으로써 첫 번째 단계에서 수행되는 비교 횟수를 줄일 수 있다.

아래 그림에서 보듯이 X, Y 모두 양의 방향으로 움직였을 경우 X, Y가 모두 양수인 부분(●)으로 이동하고 현 위치를 중심으로 3단계 탐색 알고리즘의 2번째 단계를 수행한다. 만약 이웃하는 움직임 벡터의 음과 양의 방향의 개수가 동일할 경우 즉 결과 값이 0일 경우에는 중앙을 중심으로 탐색하며 3단계 탐색 알고리즘의 1단계부터 수행한다.



(그림 7) 2단계 탐색 알고리즘 경로

다음은 제한하는 2단계 탐색 알고리즘의 수행순서를 각 단계별로 설명한 것이다.

Step 1.
이웃하는 움직임 벡터를 이용하여 움직임 방향을 예측한다. 만약 움직임 방향이 0일 경우에는 3단계 탐색 알고리즘의 Step 1을 수행한다. 즉 전체 탐색 영역의 중앙을 설정한 다음 중앙을 중심으로 탐색 영역의 1/2에 해당되는 주변 8개의 위치를 선택한다. 그리고 중앙을 포함한 9개 위치에 대하여 Cost Function이 가장 작은 값을 갖는 위치를 찾고 Step 2 과정을 수행한다.

Step 2.
예측된 방향을 중심으로 전체 탐색 영역의 1/2²인 주변 8개의 위치를 설정하고 예측된 방향을 포함한 전체 9개의 위치에 대하여 Cost Function이 가장 작은 값을 갖는 위치를 찾는다.

Step 3.
Step 2에서 찾은 Cost Function이 가장 작은 위치를 중심으로 주변 8개의 매크로블록에서 Cost Function이 가장 작은 값을 갖는 위치를 찾는다.

(알고리즘 1) 2단계 탐색 알고리즘

이 과정에서 볼 수 있듯이 첫 단계에서 이웃하는 움직임 벡터를 이용하여 3단계 탐색 알고리즘의 1단계에서 수행되는 방향성을 설정하기 때문에 가장 유사한 위치를 찾기 위한 8번의 비교를 하지 않아도 된다.

3.2 축 분할 탐색 알고리즘

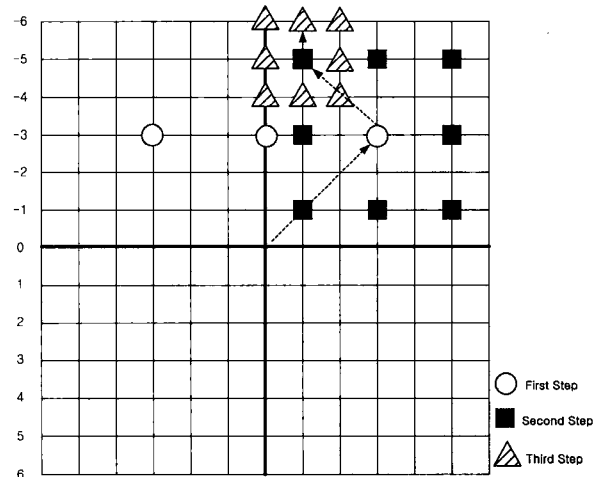
2단계 탐색 알고리즘은 3단계 탐색 알고리즘에 비해 1단계에서 수행되는 9번의 비교를 1번으로 줄이기 때문에 전

체 비교횟수를 많이 줄일 수 있다. 하지만 3단계 탐색 알고리즘에 비해 가장 유사한 매크로블록을 찾는 정확성이 떨어지기 때문에 압축률을 크게 향상시킬 수 없다.

축 분할 탐색 알고리즘은 2단계 탐색 알고리즘에 좀 더 정확성을 부여하기 위한 것으로서 1단계에 비교되는 횟수의 범위를 좀 더 넓은 알고리즘이다. 3단계 탐색 알고리즘에서 1단계의 비교 횟수를 줄이는 방법을 설명하기 위한 앞 절에서 예로써 제시된 <표 2>를 이용하고자 한다.

<표 2>에서 이웃하는 움직임 벡터를 볼 때 변환하고자 하는 인트라 매크로블록은 전체적으로 X, Y 모두 양의 방향으로 움직였다고 볼 수 있었다.

여기서 3단계 탐색 알고리즘의 1단계의 비교횟수를 줄이기 위해서는 2단계 탐색 알고리즘처럼 예측된 방향으로 움직이는 것이 아니라 이웃하는 움직임 벡터에 의해 구해진 방향을 기준으로 X축 혹은 Y축을 기준으로 수평, 수직으로 3개의 위치를 설정한다. 이때 X축과 Y축의 수평, 수직의 위치 결정은 X축과 Y축의 방향별 개수를 비교하여 많은 축을 중심으로 한다. 예를 들어 <표 2>에서 양의 방향 X는 7개이고 양의 방향 Y는 6개이다. 여기서 양의 방향 X가 양의 방향 Y 보다 개수가 많기 때문에 X 축을 중심으로 3개를 설정한다.



(그림 8) 축 분할 탐색 알고리즘의 경로

다음은 축 분할 탐색 알고리즘의 전체적인 과정을 각 단계별로 기술한 것이다.

Step 1.
이웃하는 움직임 벡터를 이용하여 움직임 방향을 예측하고 전체 탐색 영역의 1/2의 위치에 있는 수평 또는 수직방향으로 위치를 3개 설정한다. 그리고 이 3개의 위치 중 Cost Function이 가장 작은 위치를 설정한 후 다음 단계를 수행한다. 만약 예측된 움직임 방향이 0일 경우에는 3단계 탐색 알고리즘의 Step 1 단계를 수행한다.

Step 2.
Step 1에서 결정된 위치를 중심으로 전체 탐색 영역의 1/2²인 주변 8개의 위치를 설정하고 각 위치에 대하여 Cost Function이 가장 작은 값을 갖는 위치를 찾는다.

Step 3.

Step 2에서 찾은 Cost Function이 가장 작은 위치를 중심으로 주변 8개의 위치에서 Cost Function이 가장 작은 값을 갖는 위치를 찾는다.

(알고리즘 2) 축 분할 탐색 알고리즘

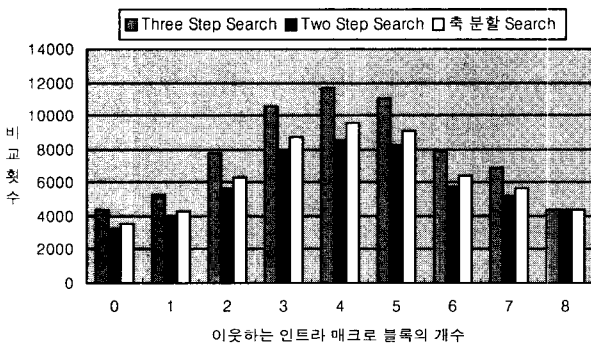
축 분할 탐색 알고리즘은 1단계에서 예측된 움직임 방향을 중심으로 X, Y축을 수평으로 3개 비교한다. 즉 2단계 탐색 알고리즘에 비해 1단계에서 수행되는 비교횟수는 많지만 좀 더 정확성을 부여할 수 있고 압축률 또한 높일 수 있다.

4. 실험 결과

본 논문은 성능 평가를 위하여 움직임이 있는 화면을 CIF 크기로 50 프레임 캡처하여 첫 번째 프레임을 I 프레임으로 하고 나머지 49개의 프레임을 P 프레임으로 하여 MPEG2 압축 과정을 통하여 압축하고 제안하는 알고리즘을 사용하여 H.263으로 변환하였다. 여기서는 3장에서 제안한 2단계 탐색 알고리즘과 축 분할 탐색 알고리즘을 3단계 탐색 알고리즘과 비교함으로써 제안한 알고리즘들의 성능을 평가하고자 한다. 성능을 평가하는 방법으로는 전체 프레임에 대한 비교횟수, 전체 압축률, 마지막으로 화질을 비교하고자 한다. 그리고 이웃하는 매크로블록의 정보가 모두 인트라 매크로블록일 경우뿐만 아니라 인트라 매크로블록이 1~8개 존재할 경우를 모두 포함하여 비교하였다. 제안한 가이드 알고리즘들은 압축된 MPEG2 비트 스트림 중 P 프레임 내에서 인트라 매크로블록일 경우에만 적용한 것으로서 I 프레임과 P 프레임내의 인트라 매크로블록일 경우에는 역양자화와 VLD 과정을 거쳐 복원한 후 H.263의 양자화와 VLC 과정을 수행하여 H.263 비트 스트림을 생성하였다. 여기서 P 프레임내의 인트라 매크로블록일 경우에는 매크로블록정보와 움직임 벡터 정보를 재 사용하였다.

4.1 비교 횟수

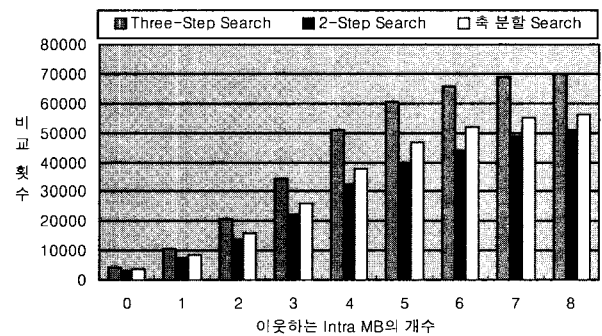
비교횟수는 각 단계별 최적의 Cost Function을 찾기 위해 비교되는 횟수를 나타낸다.



(그림 9) 탐색 알고리즘에 따른 이웃하는 인트라 매크로블록 개수 별 비교횟수

앞의 그림은 MPEG2의 P 프레임내의 인트라 매크로블록을 H.263의 인트라 매크로블록으로 변환하고자 할 때 3단계 탐색 알고리즘, 2단계 탐색 알고리즘, 축 분할 탐색 알고리즘 방법을 사용하였을 때의 전체 프레임에 대한 비교 횟수를 나타낸 그림이다. 여기서 X축인 이웃하는 인트라 매크로블록의 개수는 변환하고자 하는 매크로블록의 이웃하는 매크로블록의 형식이 인트라 일 경우에 따른 경우의 수로써 예를 들어 4는 이웃하는 매크로블록 중 인트라 매크로블록이 4개일 경우를 의미한다.

다음 그림은 탐색 알고리즘에 따른 인트라 매크로블록별 비교 횟수를 비교한 그림으로써 위 그림과는 다르게 이전의 경우를 모두 포함한 경우이다. 즉 4는 이웃하는 인트라 매크로블록의 개수가 4 미만일 경우를 의미한다.



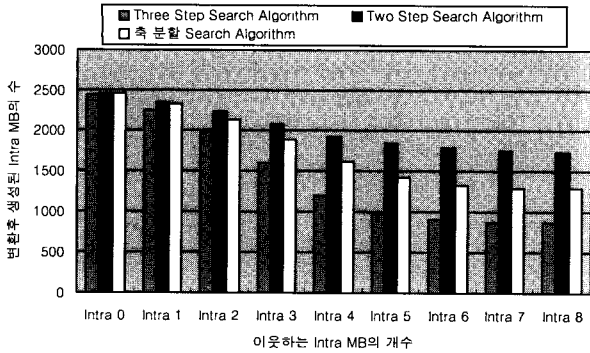
(그림 10) 탐색 알고리즘에 따른 비교 횟수

전체적으로 3단계 탐색 알고리즘일 경우 비교 횟수가 가장 많고 다음으로 축 분할 탐색 알고리즘, 그리고 2단계 탐색 알고리즘일 경우가 가장 적은 것을 볼 수 있다. 그 이유는 3단계 탐색 알고리즘일 경우에는 1단계에서 먼저 9번의 비교를 수행하기 때문에 가장 많고 축 분할 탐색 알고리즘일 경우에는 1단계에서 3번을 비교하며 2단계 탐색 알고리즘은 1단계를 수행하지 않기 때문에 비교 횟수가 가장 적다. 그림에서 이웃하는 인트라 매크로블록이 8 미만일 경우를 볼 때 3단계 탐색 알고리즘과 비교하여 Two 탐색 알고리즘은 27%, 축 분할 탐색 알고리즘은 25% 감소한 것을 볼 수 있다.

4.2 압축률

다음 그림은 제안한 가이드 탐색 알고리즘과 3단계 탐색 알고리즘을 사용하였을 때 전체 프레임에 대한 압축률을 그림으로 표현한 것이다.

다음 그림에서 제안한 탐색 알고리즘은 3단계 탐색 알고리즘에 비하여 압축률을 많이 높이지 못하였다. 그 이유는 제안한 가이드 탐색 알고리즘들은 인트라 매크로블록일 경우에만 적용된다. 2단계 탐색 알고리즘인 경우에는 1단계에서 이미 탐색 방향이 설정되었기 때문에 H.263의 Threshold를 비교했을 때 3단계 탐색 알고리즘에 비하여 인트라 매크로블록의 개수가 많다. 그리고 축 분할 탐색 알고리즘은



(그림 11) 가이드 탐색 알고리즘을 적용 후 생성된 인트라 매크로블록의 수

2단계 탐색 알고리즘에 비하여 3번의 비교를 더 하기 때문에 2단계 탐색 알고리즘보다 인트라 매크로블록의 개수가 적지만 3단계 탐색 알고리즘에 비하여 인트라 매크로블록의 개수가 많다.

4.3 화질 평가

PSNR(Peak Signal to Noise Ratio)은 원본 이미지와 상대적으로 얼마나 손상되어있는지를 측정하는 방식으로 객관적인 자료를 바탕으로 이미지의 화질을 평가하는 방법이다. 본 논문은 화질을 평가하기 위하여 PSNR을 사용하여 MPEG2로 압축되기 이전의 이미지와 가이드 탐색 알고리즘을 사용하여 변환한 후 복원했을 때의 이미지를 비교하였다. 다음 표는 각 탐색 알고리즘에 대하여 이웃하는 인트라 매크로블록의 개수에 따른 PSNR을 수치로 표현한 것이다.

<표 3> 각 탐색 알고리즘 별 PSNR

	3 단계 탐색	2 단계 탐색	축 분할 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

위 표에서 볼 수 있듯이 전체적으로 3개의 탐색 알고리즘이 큰 차이를 보이지는 않지만 2단계 탐색 알고리즘일 경우 화질이 가장 좋은 것을 확인할 수 있다. 이는 인트라 매크로블록의 개수에 영향을 받기 때문이다.

5. 결론

멀티미디어의 압축 기술로 인하여 대용량의 멀티미디어

비디오 데이터들이 정보로서의 이용가치가 날로 증가하고 있는 추세에 있다. 특히 요즘은 통신 기술의 발달로 언제, 어디서나 정보를 확인할 수 있게 되었다. 하지만 통신 기술이 발달하였다 하더라도 송신측과 수신측의 환경을 고려하지 않는다면 원활하게 정보를 확인할 수 없게 된다.

본 논문은 수신측의 환경을 고려하기 위하여 먼저 수신측의 QoS 정보 중 비디오 압축 형식을 받아들여 그에 맞게 변환하였고 네트워크의 환경을 고려하기 위하여 압축률을 좀 더 높이는데 있어 MPEG2의 P 프레임 내의 인트라 매크로블록의 개수를 최소화하는 방법을 고려하였다. MPEG2의 P 프레임을 H.263 P 프레임으로 변환하는 방법으로서 먼저 압축영역에서의 변환방법을 시도해 보았으나 심각한 화질문제가 발생하여 변환 시 움직임 예측과정을 삽입하였고 움직임 예측 시 소모되는 복잡도를 줄이기 위하여 변환하고자 하는 매크로블록의 이웃하는 움직임 벡터를 이용한 가이드 탐색 알고리즘(Two 단계 탐색 알고리즘, 축 분할 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.

참고 문헌

- [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 Birate Communication, Draft ITU-T Recommendation H.263 Version 2, September, 1997.
- [7] Information Technology- Generic 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, pp.295-304, 1998.
- [9] G. Keesman, R. Hellinghuizen, F. Hoeksema, and G. Heide-man, "Transcoding MPEG bitstream," Signal Processing : Image Communication, Vol.8, September, 1996.

- [10] T. Shanableh and M. Ghanbari, "Heterogeneous videc 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.



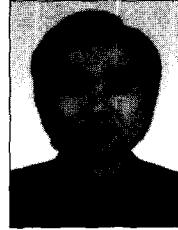
엄 성 민

e-mail : smcami@media.ssu.ac.kr
 1998년 숭실대 전산학과 졸업(학사)
 2000년 숭실대 대학원 컴퓨터공학 졸업
 (석사)
 현재 숭실대 대학원 컴퓨터공학 박사과정



강 의 선

e-mail : kanges@media.ssu.ac.kr
 1999년 제주대 컴퓨터학과 졸업(학사)
 2002년 숭실대 대학원 컴퓨터공학 졸업
 (석사)



임 영 환

e-mail : yhlim@computing.ssu.ac.kr
 1977년 경북대학교 수학과 졸업(학사)
 1979년 한국과학원 전산학과 졸업(석사)
 1985년 Northwestern University 전산학과
 (박사)
 1979년~1996년 한국전자통신연구소 책임
 연구원
 1996년~현재 숭실대학교 미디어학부 부교수
 관심분야 : 멀티미디어



황 재 각

e-mail : jghwang@etri.re.kr
 1983년 서울산업대 전자계산학과 졸업
 (학사)
 현재 한국전자통신연구원 책임기술원