

효율적인 미디어 데이터 전송을 위한 움직임분할 알고리즘에 관한 연구

김현정, 유관중
충남대학교 컴퓨터과학과
e-mail: {hjkim, kijoo}@cs.cnu.ac.kr

A Study on Motion Scaling Algorithm for Efficient Multimedia Transmission

Hyun-Jeong Kim, Kwan-Jong Yoo
Dept of Computer Science, Chungnam National University

요약

인터넷을 통한 정보전송이 급증하고 있는 오늘날, 멀티미디어 데이터 전송 또한 상당부분을 차지하고 있다. 그러나 한정된 네트워크 대역폭 상에서 대용량의 멀티미디어 데이터를 서비스하는 것은 여러가지 문제점을 안고 있어 계층적 코딩기법들을 통해 해결방안을 찾는 시도가 계속되고 있다. 본 논문에서는 이러한 계층적 코딩기법의 하나로 MPEG 비디오 데이터의 움직임 벡터를 이용한 분할기법을 제안한다. 이 기법은 데이터 전송량을 줄이면서도 고화질의 영상전송을 가능하게 한다.

1. 서론 1)

한정된 통신망 자원의 제약아래 대용량의 멀티미디어 데이터를 효율적으로 전송하기 위해 사용자의 컴퓨터환경과 통신망 환경을 고려한 계층적 코딩기법에 관한 연구가 시도되었다.

실제로 국제 동영상 압축 표준인 MPEG(Moving Picture Experts Group)-2 비디오에서는 시간해상도(temporal), 공간해상도(spatial), SNR(Signal to Noise Ratio), Data Partitioning의 4가지의 계층적 코딩방법을 표준으로 제시하였다. 그러나 이러한 기법 중 어느 한가지만을 선택하여 스트림을 전송할 경우 화질 또는 움직임 등 특정 부분의 절대적인 열화를 피할 수 없었다. 이에 복수개의 표준 계층화 기법을 조합하거나, 기존 방식을 응용하여 새로운 계층화 방식을 접목하는 연구가 진행되고 있다.

본 논문에서는 기존 계층화 연구에 대해 살펴보고 각각의 문제점을 파악한 후, 이를 개선할 수 있는 새로운 계층화 기법인 움직임 분할기법을 제안하고자 한다. 움직임 분할기법은 MPEG 비디오 데이터

의 움직임 벡터를 이용한 계층화 기법으로서, 네트워크 상태 악화시 작은 움직임 정보를 생략함으로써 전송량을 조절한다.

본 논문의 구성은 다음과 같다. 2장에서는 MPEG 비디오 계층적 코딩에 대한 기존의 연구동향 및 문제점을 분석하고, 3장에서는 MPEG-2 비디오에서의 움직임 처리 방법에 대해 알아본다. 4장에서 제안하는 움직임 분할기법에 대해 설명한다. 이 기법은 기존의 MPEG-2의 움직임 처리기법을 따르면서, 전송량 조절을 위한 새로운 계층화 방식을 시도하였다. 5장에서는 움직임 분할기법의 효율성을 알아볼 수 있도록 몇가지 실험결과를 제시하며, 이에 대한 분석 및 결론을 맺도록 하겠다.

2. 기존 연구동향 및 문제점

계층적 코딩방식에 대한 최근의 연구는 MPEG-2 국제압축 표준에서 제시한 4가지 계층적 코딩방식을 혼합하거나, 기존 방식에 새로운 계층화 방식을 덧붙이는 방향으로 진행되어 왔다.

실제로 MPEG-2 표준에서 제시한 방식 이외에 지금까지 제안된 계층화 방식에는 혼합계층화 방식, STS(Spatio-Temporal Scaling) 방식, TFS

1) 이 연구는 BK21충남대학교 정보통신인력양성사업단의 지원을 받았음.

(Temporal-Fidelity Scaling) 방식 등이 있다.

그러나 제안된 기존의 방식들은 하나의 스트림을 각기 다른 재생률을 갖는 복수개의 스트림으로 부호화하여 중복 저장하는 Simulcast 방식으로서 저장공간의 낭비문제를 피할 수 없었다.

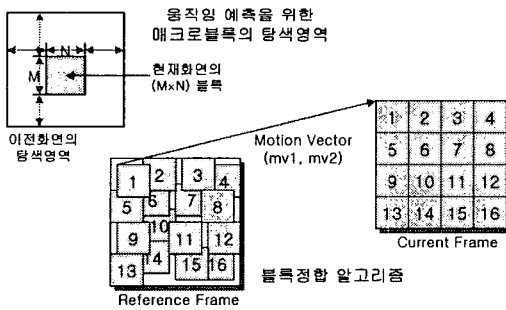
아울러 STS, TFS 분할방식 역시 한정된 표준에서만 사용가능한 분할방법인 점과 시스템 과부하 문제를 발생에 대한 해결방안은 고려하지 않은 한계를 가지고 있다.

3. MPEG 비디오 움직임 처리방법

3.1 움직임 예측(Motion Estimation)

본 논문에서 제안한 움직임 분할방식을 설명하기 앞서, MPEG 표준에서의 움직임 정보 처리방식에 대해 먼저 설명하겠다.

MPEG 표준에서는 연속된 화면 사이의 중복된 정보를 줄이기 위해 움직임 예측과 움직임 보상을 수행한다. 즉, 시간적으로 인접한 화면간에 중복되는 정보를 줄이기 위해 화면 사이의 가장 비슷한 블록을 찾아, 변하지 않는 부분이나 비슷한 부분을 바로 전 화면의 일부분으로 채운다.



[그림1] 블록정합 알고리즘을 이용한 움직임 예측

움직임 예측방법으로는 일반적으로 블록정합 알고리즘이 쓰이며, 이 기법은 [그림1]과 같이 현재 화면에 있는 매크로블록과 가장 유사한 매크로블록을 이전화면의 현재 매크로블록 위치를 중심으로한 탐색영역 안에서 찾고, 위치의 변이 정도를 움직임 벡터(Motion Vector)로 표현하게 된다.

3.2 움직임 벡터

앞절에서 살펴본 바와 같이 현재 움직임 벡터값은 부호화된 이전 움직임 벡터와의 차이만을 부호화하여 표현되므로 비트수를 줄인다.

즉, 실제로 부호기에서 코딩하게 되는 값은 현재 벡터값과 이전 벡터값의 차분인 $vector[r][s][t] - PMV[r][s][t]$ 이며, 반대로 복호기에서는 이전 벡터값에 차분값을 더한 $PMV[r][s][t] + differential_vector[r][s][t]$ 를 벡터값으로 추출하게 된다. 여기서 r은 필드, s는 순/역방향, t는 수평/수직방향을 나타내는 변수이다.

그러나 움직임 벡터는 가변장 부호화 하기에 상당하며 다양하므로, 이들 움직임 벡터들을 f 변수로 나누어 발생하는 몫(motion_code)만을 가변장부호화하고, 나머지(motion_residual)는 고정길이 부호화한다.

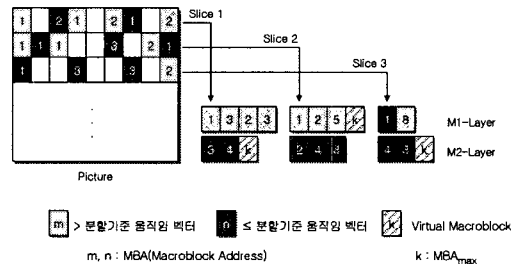
4. 움직임 분할(Motion Scaling)

4.1 움직임 분할과정

MPEG 비디오의 슬라이스 층은 복수개의 매크로블록들로 구성되어 있으며, 이들의 위치는 MBA (Macroblock Address) 증가분 값을 통해 결정된다. 각 매크로블록은 움직임 벡터값을 갖게 되는데, 작은 벡터값은 미세한 움직임을, 큰 벡터값은 많은 움직임을 의미한다. 움직임 분할은 이러한 움직임 벡터값들을 분할하여 네트워크 대역폭 약화시 미세한 움직임 정보들을 인위적으로 누락시켜 전송량을 줄이는 기법이다.

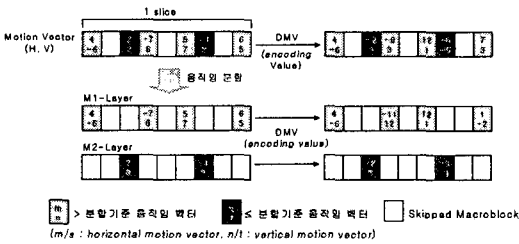
움직임 분할을 위해 원 스트림의 움직임 벡터 정보를 모두 추출하여, 지정한 분할기준 벡터값 (MV_filter) 보다 클 경우에는 M1-Layer, 작을 경우에는 M2-Layer에 매크로블록정보를 저장한다. 이때 각 매크로블록이 가진 수평, 수직방향의 두 벡터값과 MV_filter의 크기 비교를 위해 Pythagorean theorem을 이용하여 대각선 방향의 움직임 벡터값을 계산하였고 이 값을 해당 매크로블록이 갖는 움직임 정보라고 가정하였다.

[그림2]는 슬라이스 층에 존재하는 실제 매크로블록들을 움직임 분할하여 저장하는 과정이다.



[그림2] 매크로블록 분할을 통한 움직임 분할

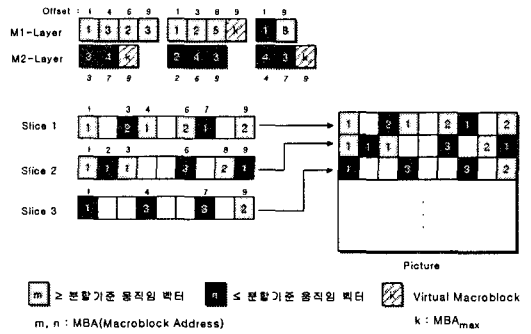
이러한 움직임 분할시에는 나누어진 두 스트림의 정확한 병합을 위해 몇가지 사항들을 고려하였다. 먼저 슬라이스 내 마지막 매크로블록 정보 제공을 위해 마지막 매크로블록이 저장되지 않는 계층에는 최대 MBA 값을 가지는 Virtual Macroblock을 삽입하였으며, 분할된 각 계층에 맞는 MBA 값으로 수정하였다. 이때 M2-layer의 첫 매크로블록은 절대 주소값을 갖도록 하였고, Virtual Macroblock은 MBA 값을 수정하지 않았다. 또한, 각 계층에 분할된 매크로블록들이 해당 계층에 맞는 움직임 벡터 차분값을 갖도록 수정하였다. [그림3]은 MV_filter가 5인 움직임 분할시의 DMV값 수정결과다.



[그림3] 움직임 분할에 따른 DMV값 수정

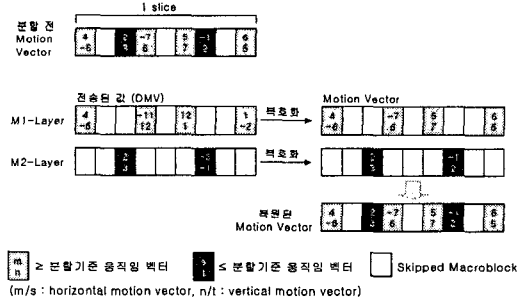
4.2 움직임 병합과정

나누어진 움직임 정보들이 사용자측에 도달하면, M1, M2-layer로 나누어진 정보들을 하나로 병합해야 한다. M1-layer만 전송되었을 경우에는 앞 매크로블록의 MBA값에 현재 매크로블록의 MBA값을 더하여 절대주소를 구한다. 반면 M1, M2-layer가 모두 전송되었을 경우에는 두 layer의 각 매크로블록이 갖는 절대 MBA를 계산하여, 작은 절대 주소값을 갖는 매크로블록부터 병합하게 된다. 분할시 슬라이스의 첫 번째 매크로블록은 M1-layer에 저장했으므로, 한 슬라이스를 만들 때 M1-layer의 첫 번째 매크로블록부터 병합을 시작한다. 나누어진 MBA값 비교를 반복하며 병합하는 과정에서 최대 MBA를 갖는 Virtual Macroblock이 나오면, 해당 layer에서 한 슬라이스의 끝을 의미하므로 MBA값 비교없이 상대계층의 남은 정보들을 병합한다. [그림4]는 이러한 과정을 통해 두 계층을 병합하는 과정을 나타내고 있다.



[그림4] M1, M2-layer 병합과정

이러한 병합과정과 병행되는 움직임 벡터 복원과정은 [그림5]와 같다. 움직임 분할시 해당계층 내에서의 벡터 차분값으로 부호화하였으므로, 복호시 두 계층의 벡터값을 독립적으로 계산하여 복호한다.



[그림5] 움직임 벡터 복원과정

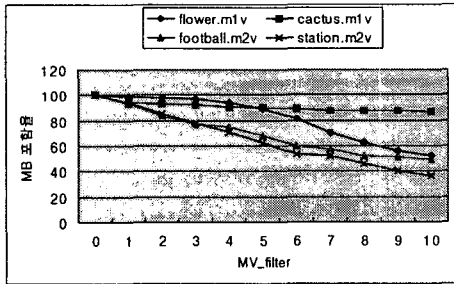
5. 실험 및 결과

본 논문에서 제안한 움직임 분할 기법을 통한 전송량 조절 가능성을 알아보기 위해 분할 후의 M1-layer 크기를 측정하는 실험을 하였다. 실험에 쓰인 4가지 MPEG 비디오 파일의 종류 및 특징은 [표1]과 같다.

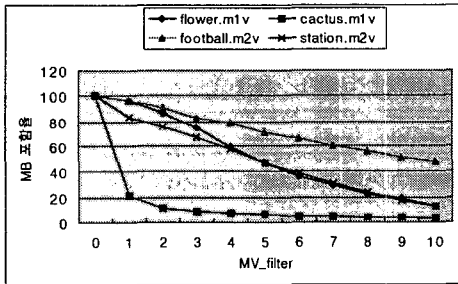
[표1] 실험데이터 특징

특성	스트림	flower.m1v	cactus.m1v	football.m2v	station.m2v
		파일크기 (KB)	704	1,469	2,042
	분할 후	935	1,736	2,417	2,316
평균 움직임 벡터값		6.5152	1.3022	8.5841	2.9408

4가지 실험 파일의 움직임 정보들을 일정 벡터값을 기준으로 계층화하였을 때, 기본계층에 포함된 매크로블록의 개수를 측정하였다. 이때 일정 벡터값을 움직임 벡터(MV_filter)라 하였으며, 이 값을 1에서부터 10까지 변경했을 때 기본계층인 M1-layer의 포함율을 계산하였다. [그림5]는 MV_filter값 변경시 P픽처의 M1-layer 포함율 변화를 나타내며, [그림6]은 M2-layer의 포함율 변화를 나타낸다.



[그림5] P픽처 분할결과



[그림6] B픽처 분할결과

실험결과 화면 내 움직임이 큰 station.m2v의 경우 MV_filter 값을 9 이상으로 할 때, 두 분할계층에 포함된 매크로블록의 수가 비슷한 결과를 보이는 반면, 배경의 움직임이 거의 없으며 평균 움직임 벡터값도 작은 cactus.m1v의 경우, MV_filter 1에서도 B픽처의 양을 21%로 줄일 수 있었다.

MPEG 비디오 파일은 움직임 정도에 따라 움직임 벡터값의 분포가 상당히 다양하므로 분할후의 특징을 규정짓는 것은 불가능하다. 아울러 제안한 움직임 분할기법은 오프라인상에서 움직임 벡터값을 분석해 두어야 하는 단점이 있다. 그러나 이 기법을 통해 계층화 한 후 네트워크 상태에 따라 전송량을 조절할 수 있으며, 매크로블록층에서 계층적 코딩을 적용하므로 블록층을 분할하는 기존의 TFS 기법에

비해 시스템부하가 적다. 또한 네트워크 상황이 좋지 않을 경우 미세한 움직임을 누락시킴으로서 전체적인 움직임이 부자유스러운 단점이 있는 반면, 화면 전체를 부호화하므로 상대적으로 좋은 화질의 영상 제공이 가능하다.

향후에는 움직임 정보의 누락과 화질 보장과의 상관관계에 대해 연구하고자 한다. PSNR(Peak Signal to Noise Ratio) 값을 통해 움직임 분할로 인한 전송량 조절시 화질열화 정도를 측정하여, 양질의 화질을 보장하는 전송량 감소 한계점을 연구해보고자 한다.

참고문헌

- [1] Banerjee, A., "Heterogeneous Networking," IEEE Multimedia, vol. 4, issue2, pp.84-87, 1997
- [2] Hoshin Son, TFS Technique for Scalable Video-on-Demand, Chungnam Nat'l Univ., 2000
- [3] Fu Ruijin, Lee Bu Sung, and Ani Gupta, "Scalable Layered MPEG-2 Video Multicast Architecture," IEEE Transactions on Consumer Electronics, vol. 47, issue 1, Feb. 2001
- [4] Dapeng Wu, Yiwei Thomas Hou, Ya-Qin Zhang, "Transporting Real-Time Video over the Internet: Challenges and Approaches," vol. 88, No. 12 Dec. 2000
- [5] 대우전자 영상연구소, MPEG 비디오, 연암출판사, 1995
- [6] ETRI 정보화기술 연구본부, 영상정보 압축의 표준, ETRI, 2000
- [7] MPEG Committee, MPEG Software Simulation Group(MSSG) Video Codec, <http://mpeg.org/MPEG/MSSG>