

# Temporal-Fidelity Scaling 기법의 MPEG-4 Visual Component로의 확장

김현정\* 이흥기\* 김상형\* 이성인\* 유관중\* 김두현\*\*

\*충남대학교 컴퓨터학과, \*\*한국전자통신연구원

(hjkim, helius, shkim, silee, kjoyoo)@cs.cnu.ac.kr, doohyun@etri.re.kr

## An Extended Temporal-Fidelity Scaling Technique for MPEG-4 Visual Component

Hyunjeong Kim\*, Heungki Lee\*, Sanghyoung Kim\*, Sungin Lee\*, Kwanjong Yoo\*, Doohyun Kim\*\*

\* Dept. of Computer Science, Chungnam National University

\*\* Electronics and Telecommunications Research Institute

### 요 약

대역폭이 보장되지 않는 인터넷 상에서 대용량의 멀티미디어 데이터를 실시간으로 서비스하는 것은 여러가지 문제점을 안고 있다. 이에 미디어 데이터의 전송량을 줄이기 위한 연구로서 layered coding을 통한 스케일러블 전송에 대한 연구가 시도되었다.

본 논문에서는 MPEG 비디오 스트림의 스케일러블 전송을 가능케 하기 위하여 Temporal-Fidelity Scaling 기법을 제안하며, MPEG-4 Visual Component에서의 적용 방법에 대해 제안하고자 한다.

### 1. 서 론

인터넷이 대중화되어가고 있지만 멀티미디어를 전송하는 streaming service에 대한 사용자들의 불만은 날로 증가하고 있다. text나 image 파일들과는 달리 멀티미디어 파일은 전송량이 커서 네트워크 상황에 많은 영향을 받기 때문이다. [1][2]

이에 네트워크 상태에 따라 데이터 전송량을 조절하는 계층적 코딩(layered coding) 기법이 도입되어 현재까지 많은 연구가 진행되고 있다. 계층적 코딩기법이란 원래의 멀티미디어 데이터를 기본계층(base layer)과 고위계층(enhancement layer)으로 나누어, 변화하는 네트워크 상황에 따라 전송량을 달리하는 기법을 말한다. 실제로 이러한 계층적 코딩기법을 사용할 경우 전송량의 조절이 가능하므로 효율적인 네트워크 사용이 가능해진다.

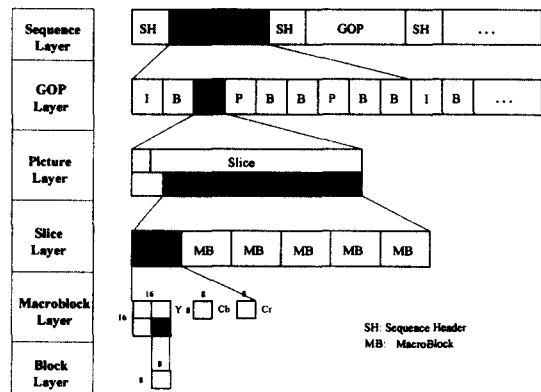
본 논문에서는 기존의 layered coding 방식과는 다른 MPEG-1, MPEG-2, MPEG-4 video stream에 적용 가능한 Temporal-Fidelity Scaling 기법에 대해서 제안한다.

본 논문의 순서는 다음과 같다. 2장에서는 MPEG 비디오의 구조에 대해서 간략히 기술하고, 3장에서는 제안한 Temporal-Fidelity Scaling 기법을 설명한다. 4장에서는 Temporal-Fidelity Scaling 기법 사용시 전송량의 변동에 관한 결과와 분석을, 마지막 5장에서는 결론을 맺는다.

### 2. MPEG 비디오 데이터 구조

#### 1.1 MPEG 비디오 데이터 구조

MPEG 표준의 영상 데이터는 [그림 1]과 같이 시퀀스층, GOP층, 픽처층, 슬라이스층, 매크로블록층, 블록층의 6개 계층으로 이루어져 있다.[3]



[그림 1] MPEG 비디오 구조

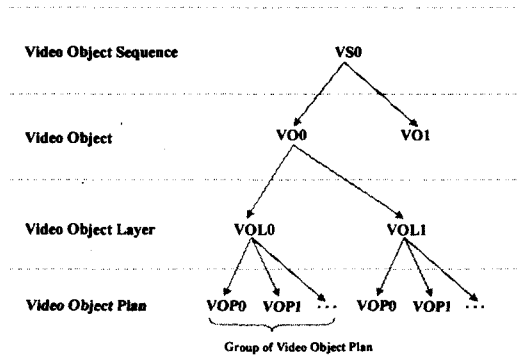
최상위의 시퀀스층은 화면형식에 관한 정보를 포함하고 있으며, 다음의 GOP(Group-of-Picture)층은 영상 시퀀스의 임의 위치로 액세스하기 위한 목적으로 만들어진 계층이다. 픽처층은 크게 I픽처, P픽처, B픽처로 이루어져 있으며, VCR의 빨리감기, 되감기 등의 기능을 위한 D픽처도 존재한다. 매크로블록층은 6개의 8×8 단위

DCT 블록으로 구성되며, 이중 4개는 휘도신호를 나타내고, 나머지 2개는 색차신호를 나타낸다. 하단의 블록 계층은 8×8 화소로 구성되어 있다.

1.2 MPEG-4 비디오 신택스 구조

기존의 MPEG-1, MPEG-2 영상에 포함되어 있는 내용과는 무관하게 영상의 화소값만을 이용하여 데이터 압축을 수행하였다. MPEG-4에서는 이러한 한계점을 인지하여 압축효율이 높은 차세대 영상 부호화 방식을 채택하고 있으며, 내용기반 부호화(Content-based Coding) 방식에 중점을 두었다.

내용기반 부호화 방식은 영상 내용을 객체(object) 단위로 나누어 처리하여 전송하는데, 이 객체 개념의 도입으로 MPEG-4 비디오 데이터는 [그림2]와 같은 신택스 구조를 가지게 되었다.[7][8]



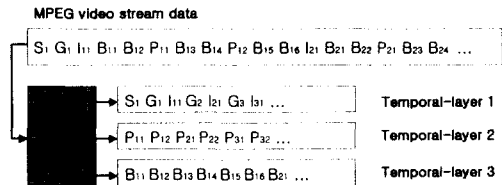
[그림2] MPEG-4 비디오 Syntax 구조

동영상은 복수의 비주얼 객체(VO)로 구성되어 있고, VO는 다시 복수의 비디오 객체 레이어(VOL)로 나뉘어지며, VOL은 복수의 VOP로 이루어져 있다.

3. Temporal-Fidelity Scaling 기법

3.1 Temporal-Fidelity Scaling

Temporal-Fidelity Scaling 기법이란 MPEG 비디오 데이터의 픽처층과 블록층을 이용한 layered coding을 통해 스케일러빌리티를 제공하는 기술이다.[9]



[그림3] Temporal Layering

먼저 위의 [그림3]과 같이 MPEG 비디오의 픽처층을

이용하여 Temporal Scaling 기법을 적용하면 다음의 3개 레이어로 분할된다.

- Temporal-layer 1: I픽처, 픽처층 상위헤더
- Temporal-layer 2: P픽처
- Temporal-layer 3: B픽처

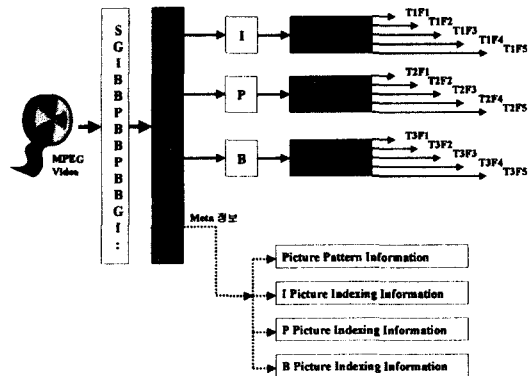
Temporal Layering 기법에 의해 3개의 레이어로 분할 후, 각각의 레이어에 대해 다시 Fidelity Scaling 기법을 적용한다. Fidelity Scaling 기법은 [그림4]와 같이 픽처의 DCT 블록 계수를 5부분으로 나누는 기법이다.

- Fidelity-layer 1: 헤더정보, DC
- Fidelity-layer 2: AC 1-2
- Fidelity-layer 3: AC 3-5
- Fidelity-layer 4: AC 6-9
- Fidelity-layer 5: AC 10-63



[그림4] Fidelity Layering

Temporal-Fidelity Scaling 기법을 두가지 분할 기법을 통합한 것으로 통합한 분할기의 전체적인 구조도는 [그림5]와 같다. 먼저 Temporal Layering을 통하여 3개의 layer로 분할하고, 이 분할된 layer 각각에 다시 Fidelity Layering 기법을 적용하게 된다.



[그림5] 통합된 분할기 구조

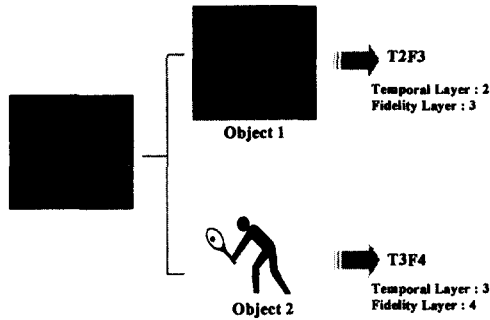
나누어진 15개 layered stream과는 별도로 이러한 다수의 스트림에 direct 접근을 가능하게 하는 픽처패턴과 픽처별 GOP 위치에 대한 인덱싱 정보를 저장한 Meta 정보파일이 별도로 생성된다.

3.2 MPEG-4에서의 Scalability 적용

내용기반 부호화(Content-based Coding)에 중점을 둔 MPEG-4에서는 영상 내용을 각각의 객체(Object) 단위

로 나누어 처리한다. MPEG-4 비디오 Syntax 구조에서 VOP는 이러한 객체 개념을 바탕으로 제안된 구조로 IVOP, PVOP, BVOP, SVOP로 구성되어져 있다.

제안한 Temporal-Fidelity Scaling 기법을 MPEG-4에 적용할 경우 MPEG-1, MPEG-2 보다 활용의 범위가 크다. 이는 [그림6]과 같이 MPEG-4에서는 객체별로 Layering 정도를 달리 할 수 있기 때문이다. 네트워크 상황이 좋지 않을 경우 상대적으로 움직임의 변화가 작은 배경화면은 낮은 Scaling layer로 전송하며, 움직임의 변화가 큰 객체는 높은 Scaling layer로 전송이 가능하다.



[그림6] MPEG-4에서의 Scalability 적용

그러나 MPEG-4에 Scalability 적용시 고려해야할 부분으로 MPEG-4에서 전경과 배경을 분리한다는 관점에서 제안된 스프라이트 부호화가 있다. 스프라이트 부호화에는 basic 스프라이트 모드와 low-latency 스프라이트 모드가 존재하며, 이중 low-latency 스프라이트 모드는 매크로블록 참조 유무에 따라 다시 piece 모드와 update 모드로 구분된다. 후자인 update 모드는 스프라이트 버퍼에 채워진 정보에 변화가 있을 경우 추가적인 정보인 update-piece를 전송하는 모드로서, 본 논문에서 제안한 알고리즘을 적용시 약간의 문제가 발생한다.

이미 스프라이트 버퍼에 채워진 배경정보에 변화가 있을 경우 update-piece가 추가적으로 전송되는데, 이 모드는 기존의 PVOP와 같은 역할을 한다. 만약 이때, 네트워크 상황의 악화로 Temporal Layer-1으로 전송량을 낮추어야 할 경우, 이 update-piece는 전송이 불가능해지며 디코더에서는 이를 hole로서 간주하게 되어 정상적인 디코딩이 불가능해진다.

이러한 문제점을 해결하기 위해 네트워크 대역폭의 변화로 layered stream의 level을 변경해야 할 경우, 스프라이트에서는 예외적으로 Temporal level을 낮추는 대신 Fidelity level을 낮춤으로서 Temporal level이 2 이하로 떨어지는 것을 막는다. 즉, T3F5로 전송하다가 네트워크 상황이 악화될 경우 T2Fm( $1 \leq m \leq 5$ )으로 낮추는 대신 T3Fn ( $1 \leq n \leq 4$ )으로 level을 변경한다.

4. 실험결과 및 분석

Temporal-Fidelity Scaling을 이용하여 MPEG 비

디오 15개의 layer로 분할한 후, 1GOP 당 TmFn( $1 \leq m \leq 3, 1 \leq n \leq 5$ )의 사이즈를 측정한 결과 평균적으로 [표1]과 같은 수치를 얻을 수 있었다.

station.m2v (size: 1,996 KB)					[단위: KB (%)]
88 (1)	156 (8)	220 (11)	275 (14)	473 (24)	
303 (15)	385 (19)	481 (24)	582 (29)	1,200 (60)	
724 (36)	808 (24)	582 (46)	1052 (53)	1998 (100)	

[표1] 1GOP 당 TmFn Layer 평균크기

다수의 실험결과에 의해 Temporal-Fidelity Scaling을 적용할 경우, 원 영상보다 화질이 다소 열화되지만 약 40%의 정보만으로도 서비스가 가능함을 알 수 있었다.

5. 결론

본 연구에서는 Temporal-Fidelity Scaling을 이용하여 stream layering하는 기법에 대해 제안하였다.

Scalable한 전송을 위하여 기존의 MPEG 비디오 스트림 데이터를 픽처 정보를 이용하여 Temporal Scaling한 후, 다시 block 정보를 통해 Fidelity Scaling을 시도하였다. 이 결과 원래의 MPEG 비디오 스트림은 15개 layered stream으로 나누어졌다.

실제 네트워크 상에서 전송시 MPEG video data 전송시 네트워크 QoS 변화에 따라 layered stream의 level을 조정함으로써 변화에 적절히 적용할 수 있으며, 아울러 효율적인 통신자원의 사용을 가능하게 할 수 있다.

참고문헌

- [1] Harrick Vin, Heterogeneous Networking, IEEE multimedia, pp84-87, 1995
- [2] S.R. McCanne, Scalable Compression and Transmission of Internet Multicast Video, Report No. UCB/CSD-96-928, 1996
- [3] 정제창, 그림으로 보는 최신 MPEG, 교보문고, 1995
- [4] International Standard ISO/IEC 13818, 1996
- [5] International Standard ISO/IEC 11172-1, 1996
- [6] International Standard ISO/IEC 14496-2, 1999
- [7] 미키 스케이치, MPEG-4의 세계, 영풍문고, 1999
- [8] Atul Puri, Tsuhan Chen, Multimedia Systems Standards and Networks, 2000
- [9] Hyungchul Kim, Yoojae Won, Changun Jeong, Scalable Media Object Framework for Heterogeneous Network Environment, IEEE/IEE ICT '99, vol.2, pp427-431, June, 1999