

## 초저속 고압축 비디오의 미디어내 동기화를 위한 멀티 스트림 생성 기법

강 경 원, 류 권 열\*, 권 기 룡\*\*, 문 광 석, 김 문 수  
부경대학교 전자컴퓨터정보통신공학부, \*위덕대학교 멀티미디어공학과,  
\*\*부산외국어대학교 전자컴퓨터공학부

### Multi-stream Generation Method for Intra-synchronization of Very Low Bit Rate Video

Kyung-Won Kang, \*Kwon-Yeol Ryu, \*\*Ki-Ryong Kwon, Kwang-Seok Moon, Moon-Su Kim  
Division of Electronic, Computer and Telecommunication Engineering, Pukyong National University,  
\*Department of Multimedia Engineering, Uiduk University,  
\*\*School of Electronic and Computer Engineering, Pusan University of Foreign Studies  
E-mail : kangkw@mail1.pknu.ac.kr

#### 요 약

초저속 고압축 비디오는 고압축 부호화를 위해 주로 P-픽처 부호화 방법을 사용한다. P-픽처는 현재 프레임과 이전 프레임간의 데이터 의존성이 높기 때문에 전송시 패킷 손실이 발생할 경우 복원 영상에 심각한 화질 열화를 야기할 수가 있다. 따라서 본 논문에서 초저속 고압축 비디오 데이터에 대한 미디어내 동기화를 위해 TCP 기반 멀티스트림 생성 기법을 제안한다. 제안한 기법은 TCP 기반으로 신뢰성 있는 전송이 가능하며, 멀티스트림을 통해 병렬 전송이 가능하므로, TCP 기반에서 네트워크 트래픽에 의해 발생하는 지터에 민감하지 않아 최선의 서비스를 제공할 수 있다.

#### 1. 서 론

최근 압축기술, 교환기술 그리고 전송기술등의 급속한 발전에 따라 사용자들은 멀티미디어 서비스에 대한 욕구가 증가하고 있다. 그러나 LAN이나 MAN같은 비봉기적인 패킷망을 사용하여 멀티미디어 데이터 전송할 경우, 생성된 미디어들의 원래 시간 정보는 망과 시스템에서의 지터(jitter)등에 의해 상실된다. 따라서 원래의 시간 관계를 복구해 줌으로써 수신측에서의 연속적인 재

생(playout)을 보장하는 동기화 기법이 필수적으로 요구된다.

동기화는 미디어 내에서 패킷과 패킷 사이의 시간 관계를 유지시키기 위한 미디어내 동기화(intra-media synchronization)와 미디어들 간의 시간 관계를 유지시키기 위한 미디어간 동기화(inter-media synchronization)로 구분할 수 있다[1,2].

기존의 동기화에 대한 연구는 다수개의 가상 채널을 하나의 채널에 실어서 미디어의 시간적인 연관성을 유지하려는 Leung[3]의 MVC방법과 미디어 정보를 데이터 내 또는 밖에 실어 보내는 Sheperd[4]의 SM 또는 SC 방법이 있다. 응용과 사용자의 입장에서 멀티미디어의 형태를 시간적인 연관성을 갖도록 구성한 Little[5]의 방법과 망에서 시간적인 차이를 보상하기 위한 Escobar[6]의 흐름 동기화 방법도 제안되었다. 현재까지 기존의 방법들은 멀티미디어 동기화를 완벽히 만족 시켜주지 못하는 것이 사실이다.

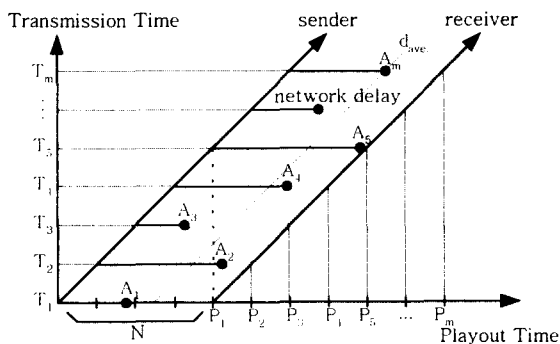
본 논문에서 현재의 인터넷 환경하에서 H.263[7]과 같은 초저속 고압축 비디오 데이터의 미디어내 동기화를 수행하기 위해 신뢰성이 있으면서 효율적인 전송과 클라이언트가 가진 대역폭과 성능을 최대한 활용할 수 있는 멀티스트림 생성 기법을 제안한다.

II. 미디어내 동기화

미디어내 동기화는 흐름동기(stream synchronization)를 의미하는 것으로 단일 미디어의 연속적인 표현을 위한 동기화다. 송신단에서 전송된 미디어 데이터는 송수신단간의 지연과 지연의 변화인 지터의 영향을 받는다. 따라서 연속적으로 보낸 송신단의 미디어 데이터는 수신단에서 불연속성을 갖게 된다. 이러한 불연속성을 없애고 송신단에서와 같은 순서의 연속성을 갖도록 하는 것이 미디어내 동기화다. 그림 1은 미디어가 UDP(user datagram protocol) 상에서 전송될 때 미디어내 동기화를 수행하는 과정을 나타낸 것이다. 그림 1에서 보는 바와 같이 전송시 지터가 존재하기 때문에 연속적으로 미디어 데이터가 재생되기 위해서는 처음  $N$ 개의 프레임에 대한 버퍼링(buffering)이 선행되어야 한다. 식 (1)은 초기 버퍼링에 필요한 프레임 수  $N$ 을 구하는 식이다[8].

$$N \geq 1 + \frac{2\delta_{max}}{I} \quad (1)$$

여기서, 미리 버퍼링 되어야 할 프레임의 수( $N$ )는 프레임간의 간격이  $I$ 와, 최대 지연 편차가  $\delta_{max}$ 가 주어지면 구할 수 있다. 그러나 UDP는 전송할 경우 네트워크가 혼잡하거나 라우팅이 복잡할 경우 네트워크상의 패킷이 손실될 수 있어 신뢰성 있는 전송은 아니다. H.263과 같이 고압축을 위해 주로 P-픽처로 부호화된 경우 현재 프레임과 이전 프레임의 데이터 의존성이 높기 때문에 전송시 패킷의 손실로 인한 영향이 복원에 영상에 심각한 화질 열화를 야기할 수가 있다. 따라서 보다 신뢰성있는 전송이 요구된다.



- $T_m$  : Transmission time of  $m$ th frame
- $A_m$  : Arrival time of  $m$ th frame
- $P_m$  : Playout time of  $m$ th frame
- $d_{ave}$  : Average delay from server to client
- $N$  : number of the buffering frames

그림 1. UDP에서의 미디어내 동기화

III. 제안한 멀티 스트림 생성 기법

초저속 고압축 비디오 데이터인 경우 주로 P-픽처로 부호화 되기 때문에 하나의 프레임의 크기는 작을 뿐만 아니라 현재 프레임과 이전 프레임간의 데이터 의존성은 매우 높다. 따라서 전송시 하나의 패킷의 손실은 연속적인 에러 전파를 야기 시킨다. 만약 UDP로 전송할 경우 패킷 손실로 인한 영향이 복원에 치명적일 수 있다. 따라서 신뢰성 있는 전송을 위해 TCP (Transmission Control Protocol) 기반 데이터 전송이 필요하다. 그러나 TCP는 패킷의 손실이나 전송 에러를 처리하여 신뢰성 있는 전송은 가능하나 스트리밍에서 필요한 데이터를 실시간에 전달할 수는 없어 미디어내 동기화가 어렵다는 단점이 있다. 따라서 본 논문에서는 신뢰성 있는 전송과 효율적인 데이터 전달을 위해 TCP 기반 멀티스트림 생성 기법을 제안한다.

그림 2는 멀티스트림의 효능을 설명하기 위해 기존의 인터넷 상에서 단일 스트림과 멀티 스트림으로 MTU (maximum transmission unit)를 전송하는 과정을 설명한 것이다. 멀티스트림으로 전송할 경우 각각의 스트림마다 개별적인 MTU 단위로 전송하기 때문에 총 패킷의 크기가 전송 대역폭보다 작을 경우 모든 MTU를 병렬적으로 전송할 수 있다.

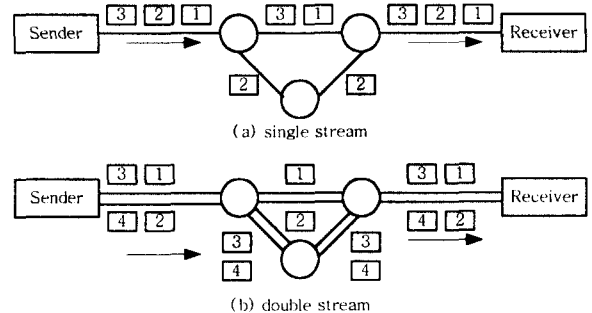


그림 2. 단일스트림과 이중스트림의 전송 예

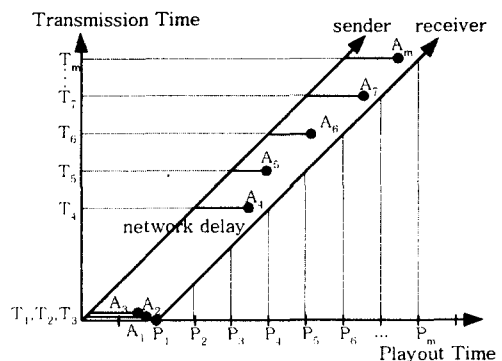


그림 3. 3개 스트림의 미디어내 동기화

그림 3은 본 논문에서 제안한 3개의 스트림을 이용하여 미디어내 동기화 과정을 나타낸 것이다. 송신단은 각각의 스트림을 통해 병렬적으로 계속 패킷을 전송할 수 있지만 수신단은 여러개의 스트림을 하나의 스트림으로 재조합 해야 하기 때문에 송신단과 수신단 사이에 동기가 필요하다.

최적의 멀티스트림 수  $N$ 을 찾기 위해  $N$ 개의 스트림을 갖는 시스템이 있다고 가정하며, 처음 접속시  $N$ 개의 스트림을 통해 병렬적으로  $N$ 개가 전송되기 때문에 식 (2)와 같은 시간적인 관계를 갖게됨을 알 수 있다. 여기서, 수신단에서는 스트림의 재조합을 위해서 송신단과 동기되어야 하므로, 송신단의 프레임이 전송 시작 시간 ( $T_m$ )은 수신단에서 이전의 프레임을 처리 한 직후 ( $P_{m-N+1}$ )가 되도록 설정하고,  $d_{ave}(N)$ 은  $N$ 에 따른 평균 지연시간을 의미한다.

$$\begin{aligned}
 A_1 &= T_1 + d_{ave} + \delta_1 \\
 A_2 &= T_2 + d_{ave} + \delta_2 \\
 &\vdots \\
 A_N &= T_N + d_{ave} + \delta_N \\
 P_1 &= T_1 + d_{ave} + \text{Max}(\delta_1, \dots, \delta_N) \\
 T_m &= P_{m-N+1} \quad (\text{단, } m > N) \\
 A_m &= P_{m-N+1} + d_{ave}(N) + \delta_m \\
 &= P_1 + (m-N)I + d_{ave}(N) + \delta_m \\
 P_m &= P_1 + (m-1)I
 \end{aligned} \tag{2}$$

제대로 재생되기 위해서는 적어도 프레임의 도착 시간이 재생시간 보다 빨라야 한다. 즉,  $A_m \leq P_m$ 이 되어야 연속적인 재생이 가능하다. 최악의 경우에도 재생이 가능하기 위해서는  $A_m^{\max} \leq P_m^{\min}$ 이 되어야 하므로 식 (2)로부터 식 (3), (4)을 유도할 수 있다.

$$\begin{aligned}
 A_m^{\max} &= P_{m-N+1} + d_{ave}(N) - \delta_{\max} \\
 &= P_1 + (m-N)I + d_{ave}(N) - \delta_{\max} \\
 P_m^{\min} &= P_1 + (m-1)I \\
 \delta_{\max} + I &\leq NI - d_{ave}(N)
 \end{aligned} \tag{3}$$

$$\delta_{\max} + I \leq NI - d_{ave}(N) \tag{4}$$

식 (4)는 수신단에서 미디어내 동기화를 구현하기 위해 필요한 스트림의 수( $N$ )를 구하는 과정을 식을 나타

낸 것이다. 즉, 미디어내 동기화를 통한 연속 재생을 위해 필요한 스트림의 수  $N$ 는 최대 지연 편차  $\delta_{\max}$ 와 프레임간의 간격  $I$ , 그리고 스트림 수에 의한 평균 지연시간  $d_{ave}(N)$ 에 의해 구할 수 있다.

#### IV. 실험 및 고찰

수신측에서 재구성된 멀티미디어 데이터의 표현 서비스 품질은 표현이 평활하게 나타나야 한다. 그리고 일정한 재생 시간 간격을 유지하여 연속성을 보장해야 한다. 따라서 본 논문에서는 이를 위한 척도로서 지터가 없는 상태에서의 기준 처리 시간( $t_{ref}$ )에 대한 지터가 있는 상태에서의 현재 처리 시간( $t_c$ )과 지터가 없는 상태에서의 기준 처리시간( $t_{ref}$ )의 차의 비를 사용하였으며, 식 (5)와 같다.

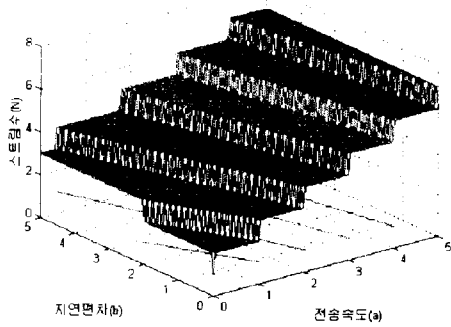
$$\text{DR(Delay Rate)} = \begin{cases} \frac{t_c - t_{ref}}{t_{ref}} & \text{if } t_c > t_{ref} \\ 0 & \text{if } t_c \leq t_{ref} \end{cases} \tag{5}$$

제안한 미디어내 동기화를 통한 연속 재생을 실험하기 위해 프레임간의 간격  $I$ 를 50ms, 최대 지연 편차  $\delta_{\max}$ 를 50ms, 평균 지연시간  $d_{ave}$ 를 100ms로 설정하였다. 이 경우 식 (4)에 의해 미디어내 동기화를 위한 최소의 스트림의 수는 4 ( $N \geq 4$ )이다. 따라서 4개의 스트림을 통해 실험한 결과 DR=0이 됨을 알 수 있었다.

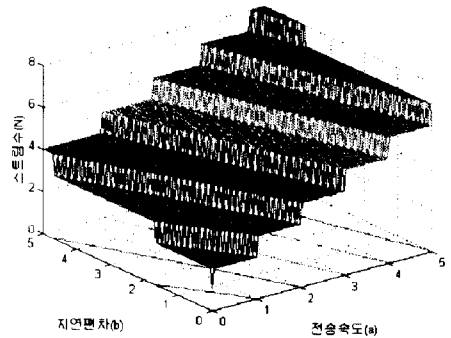
그림 4는 최대 지연 편차  $\delta_{\max}$ 의 변화와 평균 지연시간  $d_{ave}(N)$ 의 변화에 따른 연속 재생을 위해 필요한 스트림의 수  $N$ 을 구한 것이다. 식 (4)를 간단히 하기 위해,  $\delta_{\max} = bI$ ,  $d_{ave}(N) = \frac{aI}{k(N)}$ 로 하였다. 여기서  $a, b$ 는 임의의 실수를 의미한다.

일반적으로  $d_{ave}(N)$ 은  $N$ 의 함수이므로, 그림 4의 (a)는  $d_{ave}(N)$ 이 멀티 스트림의 수의 증가에 따라 전송속도가 정비례하여 전송 지연이 감소하는 이상적인 경우에 대해  $a, b$ 의 변화에 따른  $N$ 의 값을 나타낸 것이다. 그림 4의 (b)는  $d_{ave}(N)$ 이 그림 5를 이용하여 멀티 스트림의 수의 증가에 따라 전송속도가 식 (4)와 같이 근사화 한 후  $a, b$ 의 변화에 따른  $N$ 의 값을 나타낸 것이다.

$$k(N) = \begin{cases} \frac{2N+1}{3} & \text{if } 0 < N \leq 4 \\ \frac{N+8}{4} & \text{if } 4 < N \leq 8 \end{cases} \tag{4}$$



(a) 이상적인 경우



(b) 실험치에 의한 경우

그림 4. 지연편차와 전송속도에 따른 스트림의 수

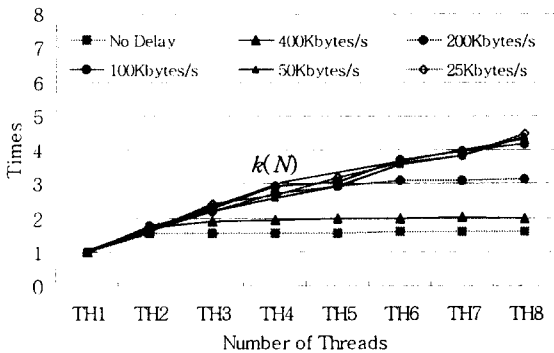


그림 5. 스트림 수에 따른 성능비교

여기서 그림 5는 LAN을 통해 동일한 크기의 파일을 전송 받을 경우 스트림의 수에 따른 수신 시간을 나타낸 것이다. 전송속도가 LAN의 최대 전송속도에 접근 할 수록 스트림에 대한 효과는 줄어들지만 저속인 경우에는 전송 속도가 스트림의 수에 따라 증가함을 알 수 있다.

### V. 결론

본 논문에서 초저속 고압축 비디오 데이터에 대한 미디어내 동기화를 위해 멀티스트림을 생성 기법을 제안 하였다. 제안한 기법은 UDP로 전송 할 경우 패킷 손실

에 인한 영향이 복원 영상에 심각한 화질 열화를 야기시킬 수 있기 때문에 신뢰성 있는 전송을 위해 TCP 기반 상에서 병렬전송을 통해 네트워크 트래픽에 의해 발생하는 지터에 민감하지 않는 멀티 스트림 생성 기법을 제안하였다. 그리고 여러 가지 최대 지연 편차  $\delta_{max}$ 와 프레임간의 간격  $I$ , 그리고 스트림 수에 의한 평균 지연 시간  $d_{ave}(M)$ 에 따라 미디어내 동기화를 가능하게 하는 최소의 스트림의 수를 제시하였다. 따라서 네트워크의 상황에 따라 적절한 스트림의 생성할 경우 TCP 상에서 초저속 고압축 비디오에 대한 미디어내 동기화를 구현할 수 있다.

향후 전송 속도에 따른 비디오 스트림에 대한 스킵과 에러 은닉에 관한 연구가 된다면 보다 효율적인 비디오 스트림 전송이 가능하리라 사료된다.

### 참고문헌

- [1] G. Blackowski and R. Steinmetz, "A media synchronization survey: Reference model, specification, and case Studies," IEEE Journal of selected areas in communications, vol. 14, no. 1, pp. 5-35, Jan. 1996.
- [2] E. Biersack, W. Geyer, C. Bernhardt, "Intra- and inter-stream synchronization for stored multimedia streams," IEEE proceeding of Multimedai '96, pp. 372-381, 1996.
- [3] W. Leng, et al., "A software architecture for worksataions supporting multimedia conferencing in packet switching networks, IEEE JSAC, vol. 8, no. 3, pp. 380-390, Apr. 1990.
- [4] D. Shepherd, and M. Salmony, "Extending OSI to support synchronization required by multimedia applications," Computer communication, vol. 15, no. 10, pp. 399-406, Dec. 1992.
- [5] T. Little and A. Ghafoor, "Multimedia synchronization protocols for broadband integrated service," IEEE JSAC, vol. 9, no. 9, Dec. 1991.
- [6] J. Escobar, D. Deutch and C. Partridge, "Flow Synchronizatin protocol, IEEE proceeding of globecom, vol. 3, pp. 1381-1387, 1992.
- [7] Draft ITU-T Recommendation H.263, "Video coding for low bitrate communication," Dec. 1995.
- [8] 김승천, 박기현, 이현태, 박재성, 이재용, 이상백, "수신측 버퍼 레벨을 이용한 멀티미디어 동기화 기법," 한국통신학회논문지, vol. 22, no. 6, pp. 1334-1342, 1997.