

게임 동영상의 효율적인 전송 계획에 관한 연구

이면재⁰, 조성현

남서울대학교 멀티미디어 학과⁰, 홍익대학교 게임학부

mjlee@nsu.ac.kr⁰, scho@hongik.ac.kr

A Study on The efficient Transmission Plan of Game Moving Picture

Myoun-Jae Lee⁰, Sung Hyun Cho

Dept. of Multimedia, Namseoul University⁰, Dept. of Game S/W Hongik University

요약

스무딩(Smoothing)은 가변 비트율로 저장된 비디오 데이터를 고정 비트율로 변환하는 전송 계획이다. 이러한 스무딩 알고리즘들에는 CBA, MCBA, MVBA 알고리즘 등이 있다. 그러나 이 알고리즘들에서는 서버에서 제공할 수 있는 여분의 대역폭의 크기를 고려하지 않아서 서버에서 비디오 데이터를 보다 많은 클라이언트들에게 제공하는 것을 어렵게 할 수 있다.

본 논문에서는 제한된 네트워크 대역폭을 갖는 동영상 서버에서 보다 많은 클라이언트들에게 비디오 데이터를 제공할 수 있게 하기 위하여 전송률 감소량(증가량)이 최대(최소)로 하는 스무딩 알고리즘을 제안한다. 제안 알고리즘의 성능을 평가하기 위하여 여러 가지 평가 요소를 사용하였으며, 제안 알고리즘은 기존의 다른 알고리즘들과 비교하여 최대 클라이언트 서비스 개수, 평균 클라이언트 서비스 개수 등과 같은 평가 요소에서 우수함을 보였다.

Abstract

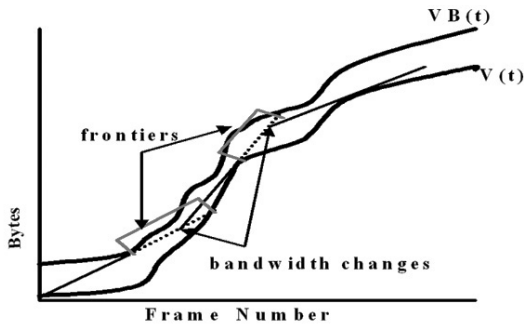
Smoothing is a transmission plan where variable rate video data is converted to a constant bit rate stream. Among them are CBA, MCBA, MVBA and others. However, these smoothing algorithms produce a transmission plan where extra bandwidth in server is not considered. This may cause difficulty in providing videos to many clients in a server.

In this paper, we propose the smoothing algorithm with monotonic transmission rate increase and abrupt transmission rate decrease, in order to provide videos to as many clients in server with limited network bandwidth as possible. In order to show the performance of our proposed algorithm, various evaluation factors were used such as the maximum number of clients, the average number of clients, and so on. Experiments demonstrated that the proposed algorithm outperformed other algorithms in evaluation factors such as the maximum number of clients and the average number of clients.

Keywords: Smoothing, Variable Bit Rate

1. 서론

비디오 데이터는 저장 공간과 효율적인 네트워크 전송을 위하여 압축되어 저장된다. 이러한 압축 방식에는 프레임 을 구성하는 비트 수가 동일한 고정 비트율(Constant Bit Rate) 방법과 동일하지 않는 가변 비트율(Variable Bit Rate) 방법이 있다[1,11]. 특히, 가변 비트율 방법으로 저장된 비디오 데이터는 프레임을 구성하는 비트 수의 차이가 심하므로 이것을 그대로 전송한다면 QoS를 보장하기 위해서 전송률을 급격히 증가시켜야 되는 버스트(Burst) 현상이 발생될 수 있다[1,2,3]. 스무딩 기법은 이러한 버스트 현상을 막기 위해 가변 비트율 방법으로 저장된 비디오 데이터를 일련의 고정 비트율로 전송할 수 있도록 비디오 데이터에 대한 전송 계획을 세우는 것이다[1,2,3].



[그림 1] 스무딩 기법의 원리

$$V(t) = \sum_{i=0}^t f_i \dots\dots\dots(1)$$

$$VB(t) = b + \sum_{i=0}^t f_i \dots(2)$$

[그림 1]은 스무딩 기법의 원리를 나타내는데[4], X축은 시간 즉 프레임 번호이며 Y축은 바이트 수이다. 식(1)의 V(t)는 언더플로우 경계선으로 프레임 0부터 프레임 t까지의 누적된 바이트 수이며, f_i 는 프레임 i의 바이트 수이다. 서버가 이 경계선보다 낮은 전송률로 프레임을 전송하면 클라이언트에서 언더플로우가 발생되어 QoS를 보장할 수 없다. VB(t)는 오버플로우 경계선으로 식(2)와 같이 언더플로우 경계선에 클라이언트 버퍼 크기 b를 더한 값으로 표현된다. 서버가 오버플로우 경계선 보다 큰 전송률로 프레임을 전송하면 클라이언트에서 오버플로우가 발생되어 QoS를 보장할 수 없다. 따라서, 서버에서 계산된 전송률이 QoS를

만족하려면 언더플로우 경계선과 오버플로우 경계선 영역 내에 있어야 하며, 이 경우에 스무딩 알고리즘은 언더플로우 경계선과 오버플로우 경계선 사이의 영역에서 동일한 전송률로 전송할 수 있는 연속적인 프레임들을 검색해야 한다. 이때 이 전송률에 의해 언더플로우 경계선(오버플로우 경계선)을 만나는 경우에 이 지점부터 오버플로우(언더플로우)를 발생시키는 프레임까지의 구간을 연장 구간(frontiers)이라 하고, 연장 구간에 속한 프레임들 중에서 스무딩 알고리즘의 목적에 적합한 프레임을 검색하여 이 프레임에서 전송률을 변화시킨다. 이 경우에 동일한 전송률로 전송하는 프레임들의 그룹을 런(Run)이라 한다.

이러한 스무딩을 위한 알고리즘에는 전송률 증가 횟수의 최소화가 목표인 CBA(Critical Bandwidth Allocation)[5], 전송률 변화 횟수의 최소화가 목표인 MCBA(Minimum Changes Bandwidth Allocation)[2], 전송률 변화량의 최소화가 목표인 MVBA(Minimum Variability Bandwidth Allocation)[6], 일정한 구간 크기에서 요구되는 버퍼 크기를 최소화하는 것을 목표로 하는 PCRTT (Piecewise Constant Rate Transmission and Transport)[7], 버퍼 크기가 커질 수 있는 PCRTT 알고리즘을 개선한 e-PCRTT(enhanced-PCRTT) 알고리즘[8] 등이 있다. 그러나 이 알고리즘들에서는 새로운 구간에서 전송률 감소가 요구되는 경우에 해제되는 전송률의 크기가 작아질 수 있고, 새로운 구간의 전송률이 증가되는 경우에 서버에서 추가로 할당해야 하는 전송률의 크기가 커질 수 있다. 그래서 서버에서 제공할 수 있는 여분의 대역폭의 크기가 작아져서 서버에서 보다 많은 클라이언트들에게 비디오 데이터를 제공하는 것이 어려워질 수 있다.

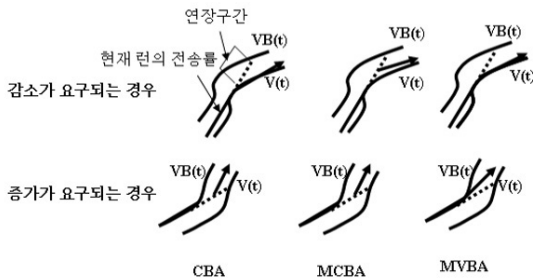
따라서, 본 논문에서는 연장 구간에서 새로운 구간의 전송률의 감소(증가)가 요구되는 경우에는 감소량(증가량)을 최대(최소)로 하는 프레임을 검색하여 검색된 프레임까지를 런으로 설정하는 스무딩 알고리즘을 제안한다. 제안 알고리즘의 성능평가를 위해 E.T.100 비디오 데이터를 사용하여 첨두 전송률, 전송률 변화 횟수, 전송률 변화량, 평균 전송률, 최소 전송률, 최대 클라이언트 개수 등을 다른 스무딩 알고리즘들과 비교한다. 본 연구는 영화에서와 같이 기획자의 의도대로 웃음과 울음 등의 다양한 감정을 온라인 게임에서도 느낄 수 있도록 게임 내에서 동영상을 제공하는 시네마틱 RPG(Cinematic Role Playing Game)[9]와 게임 소

개 동영상, 그리고 게임 플레이 도중에 동영상을 자주 제공하는 게임 동영상 서버의 전송 계획으로 적용될 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구를 기술하고, 3장에서는 제안 알고리즘을 기술한다. 4장에서는 실험 결과를 비교 분석하고, 5장에서는 결론 및 추후 연구 방향을 기술한다.

2. 관련 연구

스무딩 기법은 가변 비트율로 저장된 비디오 스트림을 전송할 때 버스트를 줄이기 위해 일련의 고정 비트율로 전송하는 방법으로, 연장 구간에 속하는 프레임 중에서 다음 런의 시작 프레임을 선택하는 방법은 알고리즘의 목적에 따라 다르다. [그림 2]는 CBA, MCBA, MVBA 알고리즘의 전송률 조절 방법을 나타낸다[4].



[그림 2] CBA, MCBA, MVBA 스무딩 알고리즘의 전송률 조절 방법

CBA 알고리즘[5]에서는 현재 전송률에 의해 오버플로우가 발생된다면 연장 구간의 첫 프레임을 다음 런의 시작 프레임으로 설정하며, 현재 전송률에 의해 언더플로우가 발생하는 경우에는 연장 구간에 속한 프레임들 중에서 새로운 전송률로 언더플로우나 오버플로우가 발생되지 않고 가장 많은 프레임을 전송할 수 있는 프레임을 검색하여 이 프레임을 다음 런의 시작 프레임으로 설정하여 전송률 증가 횟수를 최소화한다. MCBA 알고리즘[2]의 목적은 전송률 변화 횟수의 최소화인데, 이를 위해 다음 런의 전송률을 감소 또는 증가시켜야 할 때에 연장 구간에 있는 프레임 중에서 새로운 전송률에 의해 가장 많은 프레임을 전송할 수 있는 프레임을 검색하여 이 프레임을 다음 런의 시작 프레임으로 설정하지만, 알고리즘의 계산 시간이 크다[4]. MVBA 알

고리즘[3,6]은 전송률 변화량을 최소화하는 것이 목적이며, 다음 런의 전송률을 증가 또는 감소시켜야 할 때에 연장 구간의 첫 프레임을 다음 런의 시작 프레임으로 설정하는데 전송률 변화 횟수가 상대적으로 많다[4]. 전송률 변화 횟수가 주어지는 PCRTT 알고리즘[7]은 일정한 런의 크기를 갖으면서 요구되는 클라이언트 버퍼 크기를 최소화하는 것이 목적인데, 모든 구간에 동일한 오프셋 값을 더하므로 요구되는 버퍼 크기가 커질 수 있다[8]. 이를 개선한 e-PCRTT 알고리즘[8]은 각 구간의 언더플로우가 발생되지 않을 최소 전송률과 오버플로우가 발생되지 않을 최대 전송률의 평균을 해당 구간의 전송률로 설정한다. ECVBA(Efficient Changes Variability Bandwidth Allocation) 알고리즘[12]은 서버가 보다 많은 비디오 스트림을 서비스 할 수 있도록 다음 런의 전송률을 감소시켜야 할 경우에는 시작 프레임을 MCBA 알고리즘에서와 같이 설정하여 현재 전송률 보다 낮은 전송률로 보낼 수 있는 구간의 크기를 증가시키고, 다음 런의 전송률을 증가시켜야 할 경우에는 시작 프레임을 MVBA 알고리즘에서와 같이 설정하여 전송률 증가량을 최소화한다. Slice-and-Patch 알고리즘[13]은 비디오 프레임들을 첨두 전송률에 영향을 주는 프레임들과 그렇지 않는 프레임들로 분할하여 첨두 전송률에 영향을 주는 프레임들은 초기 지연 시간 동안에 전송하여 첨두 전송률을 최소화하지만 초기 지연 시간이 클 수 있고 프레임들의 재조합에 대한 오버헤드가 발생한다. 그러나, 이 알고리즘들에서는 해제되는 전송률의 크기가 작아서 서버에서 제공할 수 있는 여분의 대역폭의 크기가 적어질 수 있다. 그래서, 제한된 네트워크 대역폭을 갖는 동영상 서버에서 보다 많은 비디오 데이터를 제공하는 것이 어려울 수 있다.

3. 제안 알고리즘

기존 스무딩 알고리즘들에서는 새로운 구간에서 전송률 감소가 요구되는 경우에 해제되는 전송률의 양이 작아지거나 새로운 구간에서 전송률 증가가 요구되는 경우에 추가로 할당 받아야 할 전송률의 양이 커질 수 있다. 그래서, 네트워크 대역폭이 제한된 서버의 경우에 보다 많은 비디오 데이터를 클라이언트들에게 제공하는 것이 어려울 수 있다.

본 논문에서는, 이러한 문제점을 개선하기 위하여 전송률

의 감소가 요구되는 경우에 연장 구간에 속한 프레임들 중에 가장 크게 전송률이 감소될 수 있는 프레임을 검색하여 이 프레임을 새로운 런의 시작 프레임으로 설정하고 전송률의 증가가 요구되는 경우에 연장 구간의 첫 프레임을 새로운 구간의 시작 프레임으로 설정하여 전송률 증가량을 최소로 하는 스무딩 알고리즘을 제안한다. 그래서, 네트워크 대역폭이 제한적인 동영상 서버에서 여분의 대역폭을 많이 보유 하여 보다 많은 비디오 데이터를 클라이언트들에게 제공 할 수 있게 한다.

```

proposed algorithm(){
(1) ts=0, q=0
(2) t=ts+1
(3) compute Cmax=b, Cmin=0
(4) n=number of last frame
(5) do {
(6) t=t+1
(7) oc=(VB[t]-(V[ts]+q))/(t-ts)
(8) uc=(V[t]-(V[ts]+q))/(t-ts)
(9) if (Cmax < uc) {
(10) target=find a starting frame of
frontier
(11) output(ts-target, Cmax)
(12) ts=target
(13) }
(14) else if (Cmin>oc){
(15) target=find a frame the largest decreasing
transmission rate
(16) output(ts-target, Cmin)
(17) ts=target
(18) }else {}
(19) t=ts
(20) compute Cmin,Cmax,q
(21) }
(22) while(t<=n)
}

```

[표 1] 제안 알고리즘

이러한 원리를 사용한 알고리즘이 [표 1]인데, q 는 버퍼에 채워져 있는 바이트 수이고 ts 는 새로운 런의 시작 프레임으로 단계 (1)에서 각각 0으로 초기화 된다. 단계 (2)에서 t 는 검색되는 프레임 번호를 의미하고 단계 (3)의 C_{max} 와 C_{min} 은 각각 QoS를 보장하는 전송률 중에서 최대 전송률과 최소 전송률이다. 이 경우에 C_{max} 는 버퍼 크기로 초기화되고 C_{min} 은 0으로 초기화된다. 단계 (4)의 n 은 비디오 스트림을 구성하는 프레임 개수이다. 단계 (7)의 oc 는 런의 시작 프레임부터 프레임 t 의 오버플로우 경계점까지의 평균 전송률이고, 단계 (8)의 uc 는 런의 시작 프레임부터 프레임 t 의 언더플로우 경계점까지의 평균 전송률이다. 단계 (9)에서 단계 (13)까지는 언더플로우가 발생되어 새로운 런에서

전송률 증가가 요구되는 경우에 연장 구간의 첫 프레임을 새로운 런의 시작 프레임으로 설정하는 과정이다. 단계 (14)에서 단계 (17)까지에서는 오버플로우가 발생되어 새로운 런에서 전송률의 감소가 요구되는 경우에 연장 구간에 있는 프레임들 중에서 현재 런의 전송률에 비해 새로운 런의 전송률이 가장 크게 감소될 수 있는 프레임을 검색하여 이 프레임을 새로운 런의 시작 프레임으로 설정하는 과정이다.

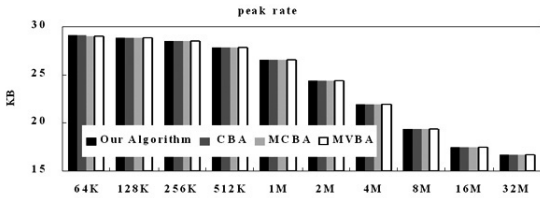
4. 실험 결과

C 언어로 스무딩 알고리즘을 구현하여 실험하였으며, [표 2]는 실험에 사용된 E.T.100 비디오 소스에 대한 파라미터 [4,10]이다. Length는 비디오 재생 시간, Ave Frame Size는 각 프레임들의 평균 바이트 수, Max Frame Size와 Min Frame Size는 프레임들의 바이트 수들 중에서 가장 큰 값과 작은 값이고, Std Dev는 프레임들의 바이트 수에 대한 표준 편차로써 이 값이 클수록 프레임들 사이에 바이트 수의 변화가 심하다. 이 경우에 사용된 단위는 KB이다. 제안 알고리즘의 성능 평가를 위해 첨두 전송률[4], 전송률 변화 횟수[4], 평균 전송률, 최소 전송률, 최대 클라이언트 개수, 최소 클라이언트 개수, 평균 클라이언트 개수를 CBA, MCBA, MVBA 스무딩 알고리즘과 비교하였다. 평균 전송률은 런의 전송률들의 평균이고 최소 전송률은 런의 전송률들 중에서 최소 전송률이다. 최대 클라이언트 개수, 평균 클라이언트 개수, 최소 클라이언트 개수는 서버에서 제공할 수 있는 대역폭을 각각 최소 전송률, 평균 전송률, 첨두 전송률로 나눈 몫이다. 본 논문에서는 서버의 대역폭을 10MB로 제한하였다.

Video Clip Name	Length (min)	Ave Frame Size	Max Frame Size	Min Frame Size	Std Dev
E.T.100	110	15.28	29.836	6.666	15.71

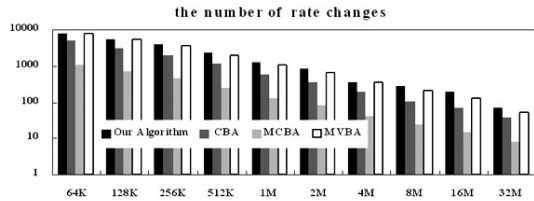
[표 2] E. T. 100 비디오 소스들의 파라미터들

[그림 3]은 첨두 전송률을 비교한 결과이다. 이 값이 낮을수록 서버와 클라이언트 간의 네트워크 자원들이 다수의 비디오 스트림을 서비스 할 수 있다[4]. 실험에 사용된 모든 알고리즘의 첨두 전송률이 거의 동일한데, 이는 실험에 사용된 모든 알고리즘들이 첨두 전송률을 최소화하는 것을 기본적인 목적으로 전송 계획을 세우기 때문이다.



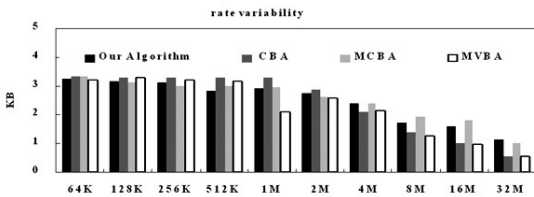
[그림 3] 첨두 전송률 비교

[그림 4]는 전송률 변화 횟수를 비교한 결과이다. 전송률 변화 횟수를 최소화하는 MCBA 알고리즘이 가장 적다. 제안 알고리즘의 전송률 변화 횟수가 MVBA 알고리즘보다 크거나 같은데, 이는 제안 알고리즘의 경우 최대로 전송률을 감소시키지만 최소로 전송률을 증가시켜서 MVBA 알고리즘에 비해 상대적으로 전송률 증가 횟수가 훨씬 많아질 수 있기 때문이다.



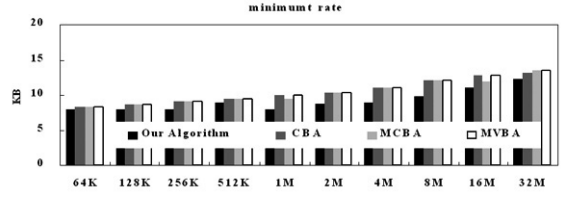
[그림 4] 전송률 변화 횟수 비교

[그림 5]는 전송률 변화량을 비교한 결과이다. 제안 알고리즘의 전송률 변화량이 큰 편에 속하는데, 이는 새로운 런의 전송률이 증가되는 경우에는 전송률 증가량을 적게 하고 새로운 런의 전송률이 감소되는 경우에는 감소량을 크게 하기 때문이다. 전송률 변화량을 최소화하는 MVBA 알고리즘의 전송률 변화량이 가장 적다.



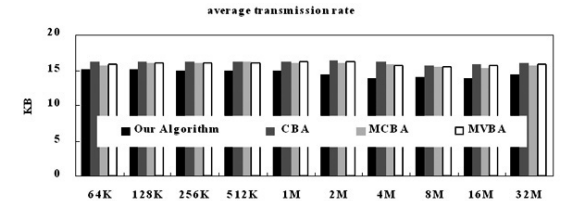
[그림 5] 전송률 변화량 비교

[그림 6]은 최소 전송률을 비교한 결과이다. 이 값이 적을수록 서버에서 제공할 수 있는 여분의 대역폭이 커질 수 있다. 제안 알고리즘의 최소 전송률이 가장 작는데, 이는 전송률 감소가 요구되는 경우에 연장 구간에서 가장 작은 전송률을 갖는 프레임을 검색하기 때문이다.



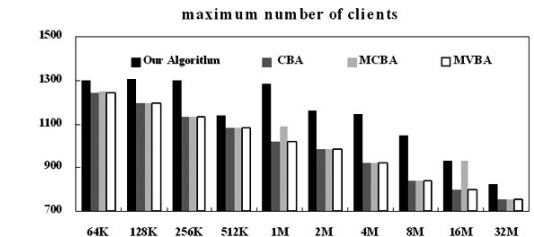
[그림 6] 최소 전송률 비교

[그림 7]은 평균 전송률을 비교한 결과인데, 이 값이 적을수록 해당 비디오 스트림을 전송하기 위해 요구되는 전송률이 적게 요구되어서 서버에서 제공할 수 있는 여분의 대역폭이 커질 수 있다. 제안 알고리즘의 평균 전송률이 가장 낮는데, 이는 제안 알고리즘에서 전송률 증가량이 적고 전송률 감소량이 큰 전송 계획을 세워서 가능한 적은 전송률로 비디오 스트림을 전송하기 때문이다.



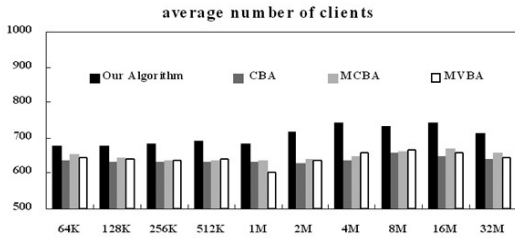
[그림 7] 평균 전송률 비교

[그림 8]은 최대 클라이언트 서비스 개수를 비교한 결과이다. 제안 알고리즘의 최대 클라이언트 서비스 개수는 [그림 6]의 최소 전송률이 가장 작아서 가장 많다.



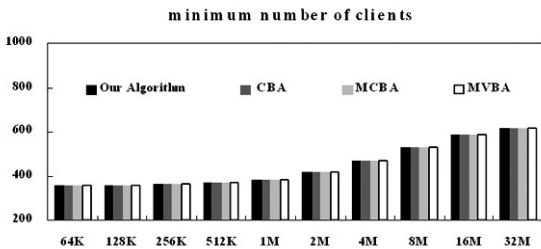
[그림 8] 최대 클라이언트 서비스 개수

[그림 9]는 평균 클라이언트 서비스 개수를 비교한 결과이다. [그림 7]에서 이미 보는 바와 같이 제안 알고리즘의 평균 전송률이 가장 낮기 때문에 제안 알고리즘의 평균 클라이언트 서비스 개수가 많아진다.



[그림 9] 평균 클라이언트 서비스 개수

[그림 10]은 최소 클라이언트 서비스 개수를 비교한 결과이다. 실험에 사용된 모든 알고리즘들의 첨두 전송률이 [그림 3]에서와 같이 거의 동일하여서 이 값도 거의 동일한 결과를 보인다.



[그림 10] 최소 클라이언트 서비스 개

5. 결론 및 추후 연구 방향

기존 스무딩 알고리즘들에서는 전송률의 감소가 요구되는 경우에 감소량을 최대로 하는 것을 고려하지 않아서 해제되는 전송률의 크기가 작아질 수 있고 전송률의 증가가 요구되는 경우에 서버에서 추가적으로 할당해야 되는 전송률의 크기가 커질 수 있다. 그래서, 서버에서 제공할 수 있는 여분의 대역폭이 작아져서 네트워크 대역폭이 제한적인 동영상 서버의 경우에 보다 많은 비디오 데이터를 클라이언트들에게 제공하는 것이 어려워 질 수 있다.

본 논문에서는 제한된 네트워크 대역폭을 갖고 있는 동영상 서버에서 보다 많은 비디오 데이터를 서비스하기 위해 새로운 구간에서 전송률의 감소가 요구되는 경우에는 연장 구간에서 전송률 감소량이 가장 큰 프레임을 새로운 구간의 시작 프레임으로 설정하고 새로운 구간에서 전송률의 증가가 요구되는 경우에는 연장 구간의 첫 프레임을 새로운 구간의 시작 프레임으로 설정하는 스무딩 알고리즘을 제안하였다. 그리고, 제안 알고리즘의 성능 평가를 위해 다

른 스무딩 알고리즘들과 전송률 변화 횟수, 첨두 전송률, 전송률 변화량, 최소 전송률, 평균 전송률, 최대 클라이언트 서비스 개수, 최소 클라이언트 서비스 개수 등을 비교하였다. 그 결과, 제안 알고리즘은 다른 스무딩 알고리즘들과 첨두 전송률은 동일하고 평균 전송률, 최소 전송률, 최대 클라이언트 서비스 개수, 평균 클라이언트 서비스 개수 비교에서는 우수한 결과를 보이고 전송률 변화량과 전송률 변화 횟수 비교에서는 다른 알고리즘들에 비해 큰 결과를 보였다. 이는 제안 알고리즘에서 전송률 증가량이 적고 전송률 감소량이 큰 전송 계획을 세우기 때문이다. 따라서, 제안 알고리즘은 전송률 변화 횟수와 전송률 변화량에 관한 오버헤드가 적고 제한된 대역폭을 갖고 있는 동영상 서버에서 보다 많은 클라이언트들에게 비디오 데이터를 제공하려는 경우에 적합하다.

추후에는 다양한 비디오 데이터와 평가 요소를 이용하여 제안 알고리즘을 비교 분석할 예정이다.

참고 문헌

- [1] D. Le Gall, "MPEG: A video compression standard for multimedia applications", Communications of the ACM, vol. 34, pp. 47-58, April 1991.
- [2] W. Feng, F. Jahanian, and S. Sechrest, "An Optimal Bandwidth Allocation Strategy for the Delivery of Compressed Prerecorded Video", ACM/ Springer Multimedia Systems, vol. 5, no. 5, pp. 297-309, Sept 1997.
- [3] J. D. Salehi, et. al. "Supporting stored video: Reducing rate variability and end-to-end resource requirements through optimal smoothing", in Proc. of ACM SIGMETRICS, pp. 222-231, May 1996.
- [4] W. Feng and J. Rexford, "Performance evaluation of smoothing algorithms for transmitting prerecorded VBR video," IEEE Trans. on Multimedia, September 1999.
- [5] W. Feng, and S. Sechrest, "Critical Bandwidth Allocation for the Delivery of Compressed Prerecorded Video", Computer Communications, vol. 18, vo. 10,

pp. 709-717, Oct. 1995.

[6] J. D. Salehi, "Scheduling network processing on multimedia and multiprocessor servers", Ph.D. dissertation, Univ. Massachusetts, Amherst, Sept. 1996.

[7] J. M. McManus et. al., "Video-on-Demand Over ATM: Constant-Rate Transmission and Transport", in IEEE Journal on selected areas in comm., Vol. 14 No. 6, August 1996.

[8] Ofer Hadar Reuven Cohen, "PCRTT Enhancement for Off-Line Video Smoothing", The Journal of Real Time Imaging, Vol. 7, No. 3, pp. 301-314, June 2001.

[9] 김경식 외, 컴퓨터 게임 개론, 글누림출판사, 2005.12

[10] <http://web.cecs.pdx.edu/~wuchi/Video/>

[11] K.R. Rao, Zoran S. Bojkovic, "Multimedia Communication Systems Technique, Standards and Networks", Prentice Hall, 2002.

[12] Han-Chel Cha, C.L Hung and T.G Tsuel, "ECVBA Traffic Smoothing Scheme for VBR Media streams", International Journal of Network Management, pp:179-185, 2002.

[13] Kong Chun Wai and Jack Y. B. Lee, "Slice-and-Patch- An Algorithm to Support VBR Video Streaming in a Multicast-based Video-On-Demand System", Journal of Information Science and Engineering, vol.19(3), May 2003.



이 면 재 (Myoun-Jae Lee)

1992년 2월 홍익대학교 전자계산학과(이학사)
 1994년 2월 홍익대학교 전자계산학과(이학 석사)
 2006년 2월 홍익대학교 전자계산학과(이학 박사)
 2006년 9월~현재 남서울대학교 멀티미디어학과 교수

관심분야: 게임 프로그래밍, 멀티미디어 통신



조 성 현 (Sung Hyun Cho)

1978년 2월 서울대학교 계산통계학과(이학사)
 1980년 2월 서울대학교 계산통계학과(이학석사)
 1995년 8월 UCLA 컴퓨터과학과(이학 박사)
 1996년 9월~현재 홍익대학교 게임학부 교수

관심분야: 게임프로그래밍, 게임그래픽스, 게임물리, 분산시스템