

GOP를 구성하는 프레임들의 순서 변경을 이용한 효율적인 스무딩 알고리즘

이면재
홍익대학교 전자계산학과
leemj@cs.hongik.ac.kr

An Efficient Smoothing Algorithm Using the Change of Frame Sequence in GOP

Myoun-Jae Lee
Dept. of Computer Science, Hongik Univ

요약

스무딩은 가변 비트율로 저장된 비디오 데이터를 클라이언트로 전송할 때 일련의 고정 비트율로 전송할 수 있도록 전송 계획을 세우는 것으로 이러한 스무딩 알고리즘에는 CBA, MCBA, MVBA, PCRTT 등이 있다. 그러나 이 알고리즘들에서는 GOP내에서 프레임들간의 바이트 수의 차이가 심한 경우에도 이 순서대로 전송 계획을 세우기 때문에 불필요하게 전송률이 변화되거나 급격하게 높은 전송률이 요구될 수 있다. 이는 네트워크 자원의 효율적인 사용을 어렵게 할 수 있다. 이를 개선하기 위하여 본 논문에서는 빠른 시간안에 최적의 해를 찾을 수 있는 백트래킹 방법을 이용하여 GOP 내에서 가장 완만하게 프레임의 바이트 수가 변화되는 프레임들의 순서를 검색하여 이 순서대로 전송 계획을 세우는 스무딩 알고리즘과 이를 위한 구조를 제안한다. 제안 알고리즘의 성능은 다양한 비디오 소스를 가지고 MVBA 알고리즘과 전송률 변화 횟수, 첨두 전송률, 전송률 변화량을 비교 분석하여 평가한다.

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, PCRTT and others. But, in these algorithm, a transmission plan is made in according to stored frame sequence in these algorithms. In case that the number of bytes in frames in GOP differs greatly each other, this may cause unnecessary transmission rate changes and may require high transmission rates abruptly when frame's byte is large. In result, it is difficult to use efficient network resource. In this paper, we proposed a smoothing algorithm that find the optimal frame sequence in short time by using backtracking method and smoothing's structure for the proposed smoothing algorithm. This algorithm decides the sequence of frames which requires the lowest variance of frame's bytes in GOP and make a transmission plan. In order to show the performance, we compared with MVBA algorithm by various evaluation factors such as the number of rate changes, peak rate, rate variability.

Keyword: Variable Bit Rate, Smoothing

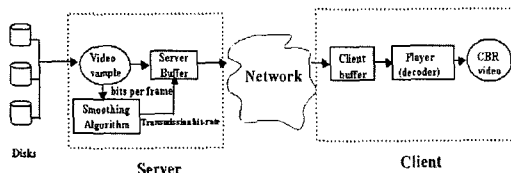
1. 서론

MPEG(Moving Picture Expert Group) 비디오 데이터는

저장 공간과 효율적인 네트워크 전송을 위하여 압축되어 저장된다. 이러한 압축 방식에는 프레임을 구성하는 비트 수가 동일한 가변 비트율(Variable Bit Rate) 방법과 동일하

지 않는 고정 비트율(Constant Bit Rate) 방법이 있다[1]. 특히, 가변 비트율 방법으로 저장된 비디오 데이터는 프레임을 구성하는 비트 수의 차이가 심하므로 이것을 그대로 전송한다면 QoS(Quality of Service)를 보장하기 위해서 전송률을 급격히 증가시켜야 되는 버스트(Burst) 현상이 발생할 수 있다[1, 2, 3]. 스무딩 기법은 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]은 스무딩(Smoothing) 기법을 위한 구조이다[4]. 클라이언트가 서버에게 비디오 스트림을 요청하면, 서버는 클라이언트에서 요청한 비디오 스트림을 검색하여 비디오 스트림을 구성하는 모든 프레임들의 정보, 즉 프레임 당 비트 수를 스무딩 알고리즘에게 전달한다. 그러면 스무딩 알고리즘은 프레임 당 비트 수와 클라이언트 버퍼 크기를 고려하여 클라이언트 버퍼에서 언더플로우와 오버플로우가 발생되지 않을 전송률, 즉 QoS가 보장되는 전송률을 설정하여 해당 비디오 스트림을 클라이언트에게 전송한다.



[그림 1] 스무딩 기법의 구조

이러한 스무딩을 위한 알고리즘에는 CBA(Critical Bandwidth Allocation)[5], MCBA(Minimum Changes Bandwidth Allocation)[5], MVBA(Minimum Variability Bandwidth Allocation)[6], PCRTT (Piecewise Constant Rate Transmission and Transport)[7, 8], e-PCRTT(enhanced-PCRTT)[8], MVBAG(Minimum Variability Bandwidth Allocation with Group of Picture)[9]등이 있다. 그러나 이 알고리즘들에서는 GOP(Group of Picture)를 구성하는 프레임 간의 크기가 심한 경우에도 저장된 순서대로 전송 계획을 세우기 때문에 불필요하게 전송률을 변화시켜야 하거나 급격하게 높은 전송률이 요구될 수 있다. 이를 개선하기 위하여 본 논문에서는 빠른 시간 안에 최적의 해를 찾을 수 있는 백트래킹(Backtracking) 방법[10]을 이용하여 GOP를 구성하는 프레임들의 비트 수가 가장 완만하게 변화될 수 있

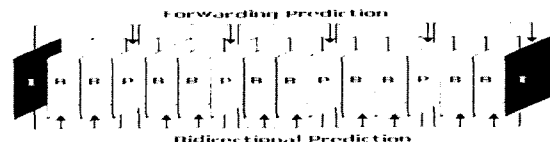
는 프레임들의 순서를 검색하여 검색된 순서대로 전송 계획을 세우는 스무딩 알고리즘과 이를 위한 구조를 제안한다. 제안 알고리즘의 성능 평가를 위해 다양한 비디오 데이터를 사용하여 전송률 변화 횟수, 침투 전송률, 전송률 변화량을 비교하였다. 본 논문의 구성은 다음과 같다. 2장에서는 관련 연구를 기술하고, 3장에서는 제안 알고리즘과 이를 위한 스무딩 구조를 기술한다. 4장에서는 실험 결과를 비교 분석하고, 5장에서는 결론 및 추후 연구 방향을 기술한다.

2. 관련 연구

본장에서는 MPEG과 스무딩 기법, 그리고 백트래킹 방법을 살펴본다.

2.1 MPEG

MPEG은 프레임과 프레임간의 연관성을 이용한 동영상 압축 방법[11]으로 I(Intra picture) 프레임, B(Bidirectionally predicted picture) 프레임, 그리고 P(Unidirectionally predicted picture) 프레임으로 구성된다[2]. I 프레임은 공간적 중복성의 원리만을 이용하여 압축된 프레임으로 독립적으로 부호화 될 수 있어서 독립적인 재생이 가능한 특징을 갖고 있다. P 프레임은 공간적 중복성 뿐만 아니라 시간적 중복성의 원리를 이용하여 압축된 프레임으로 부호화와 복호화를 할 때 이전의 I 프레임과 이전의 P 프레임의 정보를 참조하는 특징을 갖고 있다. B 프레임은 이전 또는 이후의 I 프레임 또는 P 프레임의 차이 값을 갖는 프레임이다. GOP는 하나의 I 프레임에서 다음 I 프레임 직전 프레임까지의 프레임들의 집합으로 구성된다. 하나의 GOP 내에서 B 프레임의 수는 융통성 있게 결정할 수 있으며 GOP는 임의로 접근 가능한 최소의 독립적인 단위이다. [그림 2]는 MPEG의 기본적인 GOP 패턴 예를 보여준다[1]. 이 예에서 GOP를 구성하는 프레임 개수는 15이다.

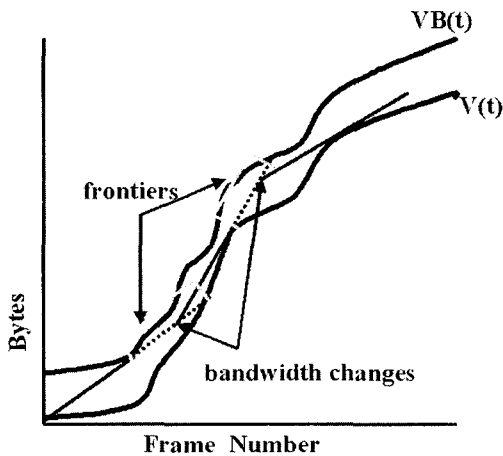


[그림 2] MPEG의 GOP 패턴 예

2.2 스무딩 기법

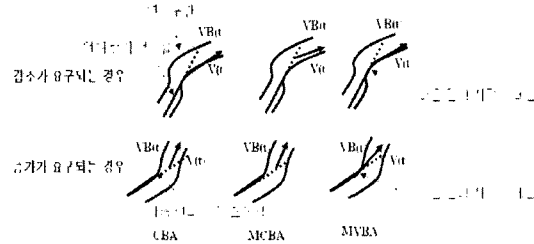
스무딩 기법은 가변 비트율로 저장된 비디오 스트림을 전송할 때 버스트를 줄이기 위해 일련의 고정 비트율로 전송하는 방법이다.

그림 3은 스무딩 알고리즘의 원리이다. $V(t)$ 는 프레임들의 누적된 바이트 수를 나타내는 언더플로우 경계선으로, 서버가 이 경계선보다 낮은 전송률로 프레임을 전송하면 클라이언트에서 언더플로우가 발생되어 QoS를 보장할 수 없다. $VB(t)$ 는 오버플로우 경계선으로 언더플로우 경계선에 버퍼 크기 b 를 더한 값으로 표현된다. 서버가 오버플로우 경계선 보다 큰 전송률로 프레임을 전송하면 클라이언트에서 오버플로우가 발생되어 QoS를 보장할 수 없다. 스무딩 알고리즘은 동일한 전송률로 전송할 수 있는 연속적인 프레임들을 검색해야 하며, 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 변화시키며, 동일한 전송률로 전송하는 구간을 런(Run)이라 한다.



[그림 3] 스무딩 기법의 원리

이러한 원리를 바탕으로 침투 전송률(Peak rate)을 최소화하면서 전송률 변화 횟수, 전송률, 버퍼 이용률 등의 특정 요소를 최적화하는 목적으로 다양한 스무딩 알고리즘이 연구되어 왔다. [그림 4]는 CBA, MCBA, MVBA 스무딩 알고리즘의 전송률 조절 방법을 나타낸다[4].



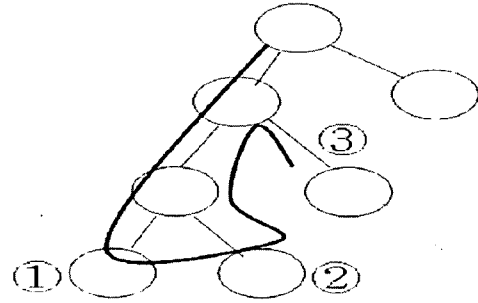
[그림 4] CBA, MCBA, MVBA 스무딩 알고리즘의 전송률 조절 방법

CBA 알고리즘[5]에서는 현재 런의 전송률에 의해 오버플로우가 발생되어 다음 런에서 전송률의 감소가 요구되는 경우에는 연장 구간의 첫 프레임을 다음 런의 시작 프레임으로 설정하며, 현재 전송률에 의해 언더플로우가 발생되어 다음 런에서 전송률의 증가가 요구되는 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[4]. MCBA 알고리즘[5]의 목적은 전송률 변화 횟수의 최소화인데, 이는 서버와 네트워크 사이의 전송률 조절에 관련된 통신 오버헤드를 감소시키고 서버에 저장된 비디오 데이터에 대한 접근 스케줄링을 용이하게 한다[4]. 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 알고리즘[6]은 전송률 변화량을 최소화하는 것이 목적인데, 공유되는 네트워크의 자원들에게 이전 전송률보다 증가량이 큰 전송률을 요구하면 요구된 전송률을 예약하지 못할 수도 있다. 그래서 이전 전송률보다 전송률의 증감이 완만한 전송 계획을 세우는 것이 좋다[4]. 이를 위해 MVBA 알고리즘은 다음 런의 전송률을 증가 또는 감소시켜야 할 때에 연장 구간의 첫 프레임을 다음 런의 시작 프레임으로 설정하는데 전송률 변화 횟수가 상대적으로 많다[4]. 특히, 전송률 변화 횟수가 주어지는 PCRTT 알고리즘은 전체 비디오 스트림을 구성하는 프레임 개수를 전송률 변화 횟수로 나눈 몫을 1개의 구간 크기로 설정한다. 이 구간 크기로 비디오 스트림을 분할하고 각 구간의 시작 프레임의 언더플로우 경계점과 구간의 끝 프레임의 언더플로우

경계점을 연결하는데, 이 연결선이 해당 구간의 임시 전송률이다. 이와같이 모든 구간의 임시 전송률을 계산하고 언더플로우가 가장 크게 발생된 프레임이 속한 구간의 프레임들이 언더플로우를 피할 수 있는 최소의 오프셋 값을 구하고 이 오프셋 값을 모든 임시 전송률에 더해 각 구간의 전송률을 설정한다. PCRIT 알고리즘은 특정 구간만 언더플로우가 심하게 발생되었을 때에도 모든 구간에 오프셋 값을 더하므로 버퍼 이용률과 비디오 데이터의 전송에 요구되는 버퍼 크기가 커질 수 있다. 이 문제점을 개선하기 위하여 e-PCRIT 알고리즘[8]은 각 구간의 언더플로우가 발생되지 않을 최소 전송률과 오버플로우가 발생되지 않을 최대 전송률의 평균을 해당 구간의 전송률로 설정한다. 그러나, e-PCRIT 알고리즘에서는 버퍼 크기가 작은 경우에 QoS를 만족시키지 못하는 구간이 발생될 수 있으며 강제적으로 구간마다 전송률을 변화시켜야 되므로 불필요하게 전송률이 변화 될 수 있다. MVBA 알고리즘[9]에서는 GOP 단위로 전송 계획을 세워서 계산 시간, 전송률 변화 횟수가 감소되지만 QoS를 보장하지 못하는 단점을 갖고 있다. 이러한 스무딩 알고리즘들에서는 GOP를 구성하는 프레임간의 바이트 수의 차이가 심한 경우에도 저장된 순서대로 전송 계획을 세우기 때문에 불필요하게 전송률이 변화되거나 요구되는 전송률이 커질 수 있다.

2.3 백트래킹 방법

백트래킹 방법은 일반적으로 깊이 우선 탐색을 이용하여 최적의 해를 찾는 경우에 많이 사용된다[10]. 이때 최적의 해를 찾기 위해 소요되는 시간은 트리 구성 방법과 불필요한 탐색을 줄이는데 사용되는 가지치기 함수(Pruning Function)에 따라 좌우된다. 백트래킹 방법은 주어진 문제를 일반적으로 트리 형태로 구성하고 깊이 우선 탐색을 사용하여 모든 노드들에 대한 검색을 시도하여 원하는 해를 찾는다. 원하는 해를 검색하지 못한 경우에는 이전 분기 노드로 이동하여 검색한다. 이와 같은 과정을 반복하여 트리의 모든 노드를 검사한 결과 원하는 해를 검색하지 못할 경우에는 주어진 문제의 해는 없다고 판단한다. [그림 5]는 백트래킹 방법의 검색 과정을 보여준다. 리프(leaf) 노드 ①를 검색하고 상위 노드의 오른쪽 노드인 리프 노드 ②를 검색한 후에 노드 ③을 검색한다. 이와 같은 과정을 반복하여 최적의 해를 구한다.



[그림 5] 백트래킹 방법

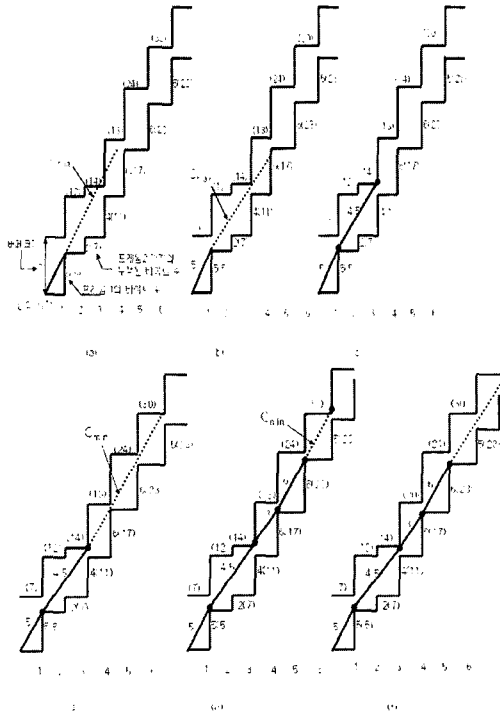
3. 제안 알고리즘

본장에서는 백트래킹 방법을 이용하여 GOP내의 프레임들의 순서를 변경하여 전송 계획을 세우는 제안 알고리즘과 이를 위한 구조를 기술한다.

3.1 제안 알고리즘

기존 스무딩 알고리즘들에서는 런의 시작 부분 또는 특정 프레임에서 바이트 수가 큰 경우에도 저장된 순서대로 전송 계획을 세우므로 비디오 프레임들을 전송하기 위해 요구되는 전송률, 전송률 변화 횟수, 그리고 전송률 변화량이 불필요하게 커질 수 있다. [그림 6]은 저장된 순서대로 전송 계획을 세우는 기존 스무딩 알고리즘들의 문제점을 보여주기 위한 예로써 사용된 알고리즘은 MVBA 알고리즘이며 버퍼 크기는 7이라고 가정한다. C_{min} , C_{max} 는 각각 런의 첫번째 프레임부터 검색되는 프레임까지 QoS를 보장하는 전송률 중에서 최소 전송률과 최대 전송률이다. [그림 6] (a)에서는 C_{min} 의 전송률로 프레임 1부터 프레임 3까지를 전송하려는 경우에 오버플로우가 발생되어서 전송률이 조절되어야 함을 보여준다. 그래서, [그림 6] (b)에서는 런의 시작 프레임부터 프레임 1까지를 첫번째 런으로 완성하고 이 런의 전송률을 5(5/1)로 설정한다. [그림 6] (c)에서는 프레임 2부터 프레임 4까지를 C_{max} 의 전송률로 전송하려는 경우에 언더플로우가 발생되어 프레임 2부터 프레임 3까지를 새로운 런으로 완성하고 전송률을 4.5((14-5)/2)로 설정한다. [그림 6] (d)부터 [그림 6] (f)까지에서는 이와 같은 과정을 반복하여 생성된 런과 런의 전송률을 보여주고 있다.

따라서, 프레임 1부터 프레임 6까지에 속한 프레임들을 전송하기 위해 요구되는 첨두 전송률은 6이 되고 전송률 변화 횟수는 4이다.



[그림 6] 기존 스무딩 알고리즘의 문제점 예

이러한 기존 스무딩 알고리즘의 문제점을 개선하기 위하여, 본 논문에서는 프레임들 구성하는 바이트 수가 가장 완만하게 변화되는 순서로 GOP를 구성하는 프레임들의 순서를 변경하여 이 순서를 바탕으로 전송 계획을 세우는 스무딩 알고리즘을 제안한다. 이는 GOP를 구성하는 프레임들 간의 버스트를 감소시켜서 불필요하게 전송률이 변화되거나 요구되는 전송률이 커지는 것을 막는데 도움을 줄 수 있다. 이를 위해 제안 알고리즘은 GOP를 구성하는 프레임들의 순서를 설정하는 과정과 전송 계획을 세우는 과정으로 구성된다. 첫 번째 과정은 GOP를 구성하는 프레임들에서 완만하게 바이트 수가 변화되는 프레임들의 순서를 얻는 과정으로 이를 위하여 빠른 시간 안에 최적의 해를 찾을 수 있는 백트래킹 방법을 이용한다.

[그림 6]은 [그림 6]의 예에 대해서 백트래킹 방법을 이용하여 프레임들의 순서를 설정하는 과정을 보여준다. [그림 6] (a)는 검색 트리로서 트리의 노드는 GOP를 구성하는 프

레이프, 깊이는 GOP를 구성하는 프레임 개수, 단계는 해당 프레임의 위치이다. 예를 들어 [그림 6] (a)의 단계 1에 프레임 1이 있는데 이는 임시적으로 GOP를 구성하는 첫 번째 프레임으로 설정된다는 것을 나타낸다. GOP를 구성하는 프레임 개수는 6이라고 가정한다. 백트래킹 방법에서 사용된 가지치기 함수는 식 (1)과 식 (2)이다.

$$Cur(i)_{rate} \leq \min_{rate} \dots \dots \dots \text{식 (1)}$$

$$Cur(i)_{diff} \leq \min_{diff} \dots \dots \dots \text{식 (2)}$$

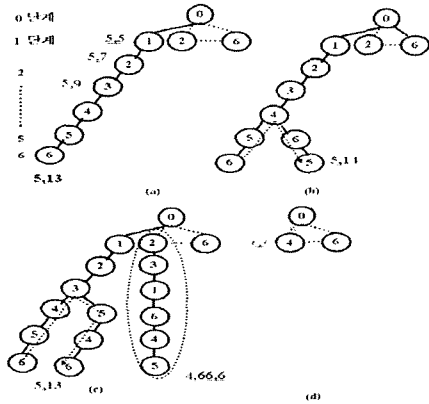
$Cur(i)_{rate}$ 와 $Cur(i)_{diff}$ 는 각각 현재 검색되는 순서로 프레임들의 순서가 변경되는 경우에 검색 트리의 단계 1에 있는 프레임부터 단계 i 의 프레임까지를 전송하기 위해 요구되는 임시 전송률과 프레임간의 바이트 수의 차이의 합이다. 식 (3)은 i 번째 프레임에서의 임시 전송률인 $Cur(i)_{rate}$ 을 보여준다. $Cur(i)_{rate}$ 는 첫 번째 단계에서부터 i 번째 단계까지의 평균 바이트 수들 중에서 가장 큰 값으로 이전 단계에서의 임시 전송률인 $Cur(i-1)_{rate}$ 와 i 번째 단계에서의 평균 바이트 수인 $acc(i)/i$ 중에서 큰 값으로 설정된다. $acc(i)$ 는 GOP를 구성하는 첫 번째 프레임부터 i 번째 프레임까지의 누적된 바이트 수이다.

$$Cur(i)_{rate} = \max(Cur(i-1)_{rate}, acc(i)/i) \dots \dots \text{식 (3)}$$

예를 들어 [그림 6] (a)의 예에서 프레임 1, 프레임 2, 프레임 3, 프레임 4, 프레임 5, 프레임 6으로 프레임들의 순서가 설정되는 경우에 $Cur(1)_{rate}$ 는 5(5/1)이고 $Cur(2)_{rate}$ 는 $Cur(1)_{rate}$ 값 5와 $3.5((5+2)/2)$ 중에서 큰 값인 5로 설정된다. \min_{rate} 와 \min_{diff} 는 각각 현재까지 검색된 순서들 중에서 가장 적은 임시 전송률과 가장 적은 프레임 바이트 수의 차이이다. 이들의 초기 값은 무한대로 설정된다. 식 (1)과 식 (2)의 조건을 만족하는 경우에는 자식 노드에 대한 검색을 진행하고 그렇지 않을 경우에는 검색을 중단하고 이전 분기 노드에서 검색되지 않는 자식 노드들에 대한 검색을 진행한다.

[그림 6] (a)에서 프레임 1, 프레임 2, 프레임 3, 프레임 4, 프레임 5, 프레임 6으로 프레임들의 임시의적인 순서가 설정되는 경우 즉 리프 노드를 검색하는 경우에는 $Cur(6)_{rate}$ 와 \min_{rate} 값과 $Cur(6)_{diff}$ 와 \min_{diff} 를 비교하여 $Cur(6)_{rate}$ 과 $Cur(6)_{diff}$ 이 작다면 현재 검색된 프레임들의 순서를 가장 좋은 프레임들의 순서로 저장하고 $Cur(6)_{rate}$ 와 $Cur(6)_{diff}$ 를

각각 min_{rate} 와 min_{diff} 로 변경한다. [그림 7] (a) 에서는 min_{rate} 와 min_{diff} 값이 각각 5과 13으로 변경된다. 이렇게 리프 노드를 검색한 이후에는 분기 노드인 프레임 4로 이동하여 [그림 7] (b)에서와 같이 프레임 5와 프레임 6의 순서가 설정되고 이러한 경우에 $Cur(6)_{rate}$ 와 $Cur(6)_{diff}$ 를 계산하여 min_{rate} 와 min_{diff} 를 결정한다. 이와 같은 과정을 반복하여 [그림 7] (c)에서와 같이 프레임 2, 프레임 3, 프레임 1, 프레임 6, 프레임 4, 프레임 5로 설정되는 경우에 min_{rate} 는 4.66 이고 min_{diff} 는 6을 갖는 프레임들의 순서를 얻는다. [그림 7] (d)에서는 GOP의 첫 번째 위치로 프레임 4가 검색되는 경우에 $Cur(1)_{rate}$ 가 6(6/1)이 되어 min_{rate} 보다 크므로 더 이상 자식 노드들을 생성하지 않고 검색을 중단한다. 이와 같은 과정을 반복하여 min_{rate} 와 min_{diff} 가 가장 작은 값을 갖는 프레임들의 순서 즉 프레임 2, 프레임 3, 프레임 1, 프레임 6, 프레임 4, 프레임 5를 얻는다.



[그림 7] 백트랙킹 방법을 이용한 [그림 6]의 프레임 순서 설정 예

[그림 8]은 제안 스무딩 알고리즘이다. 단계 1부터 3까지는 백트랙킹 방법을 이용하여 GOP내의 프레임들의 순서를 설정하는 과정이다. 단계 1은 GOP 버퍼 크기를 고려하여 이 크기에 만족되는 범위의 프레임들에 대한 검색 트리를 만드는 과정이다. 이때 GOP 버퍼 크기는 해당 비디오 스트림을 구성하는 GOP를 구성하는 프레임들의 바이트 수의 최대 값으로 설정된다. 단계 4부터 단계 19까지는 MVBA 스무딩 알고리즘[3]의 과정이다. ts 는 런의 시작 프레임이고 te 는 검색되는 프레임에 의미하고 q 는 버퍼에 채워져 있는 바이트 수이다. tB 와 tD 는 런의 시작 프레임부터 검색되는 프레임 te 까지의 프레임들 중에서 QoS를 만족하는 전송률 중

에서 각각 최대 전송률과 최소 전송률을 설정한 프레임이다. 단계 8에서 단계 11까지에서는 언더플로우가 발생하는 경우에 런의 첫 프레임부터 tB 프레임까지를 현재 런으로 설정하는 과정이며 단계 12부터 단계 15까지에서는 오버플로우가 발생하는 경우에 런의 첫 프레임부터 tD 프레임까지를 현재 런으로 설정하는 과정이다.

[그림 9]는 [그림 8]의 제안 알고리즘에 의한 [그림 6]의 예이다. [그림 7]에서와 같은 백트랙킹 과정을 수행하여 검색된 프레임들의 순서는 프레임 2, 프레임 3, 프레임 1, 프레임 6, 프레임 4, 프레임 5이다. 이 순서를 바탕으로 단계 4부터 단계 19까지를 수행하여 전송 계획을 얻게 되는데 첫 번째 프레임부터 두 번째 프레임까지에서 요구되는 전송률은 3이 되고 세 번째 프레임부터 4번째 프레임까지에서 요구되는 전송률은 5.5가 되어 첨두 전송률은 5.5가 되고 전송률 변화 횟수는 2가 되어 [그림 6]에서보다 전송률 변화 횟수, 첨두 전송률, 그리고 전송률 변화량이 적다.

```

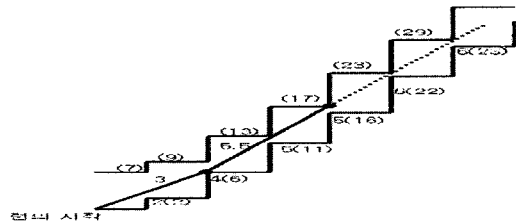
1 make a search tree considering GOP buffer size
2.  $min_{rate} = \infty, min_{var} = \infty$ 
3. checknode(node v) {
    3.1 If  $(Cur(GOP)_{rate}$  smaller than  $min_{rate}$ )
         $min_{rate} = Cur(GOP)_{rate}$ 
    3.2 If  $(Cur(GOP)_{diff}$  smaller than  $min_{diff}$ )
         $min_{diff} = Cur(GOP)_{diff}$ 
    3.3 If (promising(v))
        for (each child u of v)
            checknode(u)
}
4.  $ts=0, te=1, q=0$ 
5.  $Cmax=b, tB=1, Cmin=VB(1)$ 
6. REPEAT
7.   set  $te=te+1$ 
8.   IF  $Cmax < \frac{V(te) - (D(ts) + q)}{te - ts}$ 
9.     output segment( $tB-ts, Cmax$ )
10.    start segment at  $tB$ 
11.     $ts=tB, te=tB+1, q=VB(tB)-V(tB)$ 
12.  ELSE IF  $Cmin > \frac{VB(te) - (D(ts) + q)}{te - ts}$ 
13.    output segment( $tD-ts, Cmax$ )
14.    start segment at  $tD$ 
15.     $ts=tD, te=tD+1, q=0$ 
16.  ELSE ;
17.  END IF
18.  Compute  $Cmax, tB, Cmin$ , and  $tD$  over[ $ts, te$ ]
19.  UNTIL  $te=N$ 
    
```

[그림 8] 제안 알고리즘

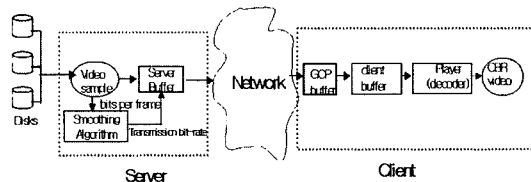
3.2 제안 알고리즘을 위한 스무딩 기법 구조

제안 알고리즘에서는 GOP를 구성하는 프레임들의 순서를 변경하여 전송 계획을 세우기 때문에 이 순서로 전송하는 경우에는 클라이언트측에서 정상적인 재생이 어려울 수 있다. 즉 원래 재생되어야 하는 프레임이 도착할 때까지 기다려야 되므로 화면의 연속성이 저하될 수 있다.

따라서, 기존 스무딩 기법의 구조의 변경이 필요하다. [그림 10]은 제안 알고리즘을 위한 스무딩 기법의 구조를 보여 준다. [그림 1]의 기존 스무딩 기법의 구조에 비교하여 GOP 버퍼가 추가되었는데, 이는 프레임들을 GOP 단위로 모아서 원래 저장된 순서대로 클라이언트 버퍼에 전달하는 역할을 수행한다. 클라이언트가 서버에게 비디오 스트림을 요청하면, 서버는 클라이언트에서 요청한 비디오 스트림을 검색하여 비디오 스트림을 구성하는 모든 프레임들의 정보를 바탕으로 전송 계획을 세운다. 그러면, 클라이언트측에서는 전송된 프레임들을 GOP 버퍼에 저장한 후에 원래 저장된 프레임의 순서대로 클라이언트 버퍼에 보내어서 비디오 스트림을 정상적으로 재생한다.



[그림 9] 제안 알고리즘에 의한 [그림 6]의 예



[그림 10] 제안 알고리즘을 위한 스무딩 기법의 구조

4. 실험 결과

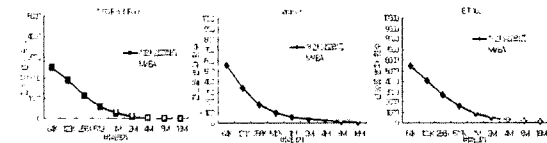
C 언어로 스무딩 알고리즘을 구현하여 실험하였으며, [표 1]은 사용된 비디오 소스에 대한 파라미터들(5, 9)이다. Length는 비디오 재생 시간, Ave Size는 각 프레임들의 평균

바이트 수, Max Size와 Min 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. 즉 1993 Final Four는 바이트 수의 변화가 가장 심한 소스이고, Seminar는 바이트 수의 변화가 아주 적은 소스이다. GOP pattern은 해당 비디오 소스의 GOP 패턴을 나타낸다. 제안 알고리즘의 성능 평가를 위해 첨두 전송률, 전송률 변화량, 전송률 변화 횟수를 MVBA 알고리즘과 비교하였다.

Video Clip Name	Length (min)	Ave	Max Size	Min Size	Std Dev	GOP pattern
1993 Final Four	41	16.07	28.872	2.504	16.57	IBBPBB
Seminar	63	8.4	10.791	7.012	8.422	
E.T100	110	15.38	29.836	6.66	3.22	

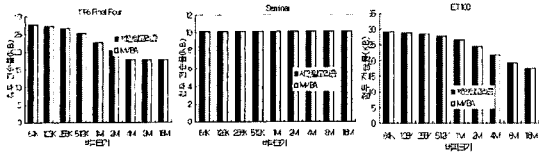
[표 1] MPEG 비디오 소스들의 파라미터들

[그림 11]은 전송률 변화 횟수 비교로써, 이 값이 낮을수록 서버와 클라이언트 간의 전송률 예약에 관한 통신 오버헤드가 적게 소요된다. 제안 알고리즘의 전송률 변화 횟수가 적다. 이것은 제안 알고리즘에서 GOP내의 프레임들의 바이트 수가 완만하게 변화되게 하였기 때문이다. 두 알고리즘 모두에서 버퍼 크기가 커질수록 전송률 변화 횟수는 적어지는데 이는 버퍼 크기가 커질수록 1개의 전송률로 보낼 수 있는 런의 크기가 커질 수 있기 때문이다.



[그림 11] 첨두 전송률 비교

[그림 12]는 첨두 전송률 비교이다. 이 값이 낮을수록 서버와 클라이언트 간의 네트워크 자원들이 다수의 비디오 스트림을 서비스 할 수 있다. 제안 알고리즘의 첨두 전송률이 E.T100 비디오 데이터에서 버퍼 크기가 256 KB인 경우에 낮는데, 이는 급격하게 높은 전송률이 요구되는 프레임의 위치가 변경되었기 때문이다. 이를 제외하면 MVBA 알고리즘과 동일하다. 이렇게 제안 알고리즘의 첨두 전송률이 크게 감소되지 않는 것은 전송되어야 될 프레임들의 개수와 바이트 수가 동일하고 MVBA 알고리즘이 첨두 전송률을 최소화하면서 전송률 변화량을 최소화하는 전송 계획을 세우기 때문이다.

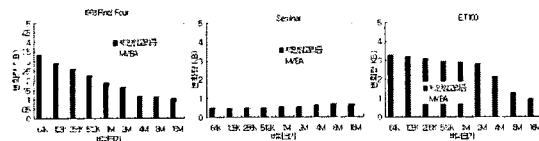


[그림 12] 침두 전송률 비교

[그림 13]은 전송률 변화량 비교이다. 전송률 변화량은 식 (4)에 의해 계산되며, n 은 전송률 변화 횟수, C_i 는 i 번째 룰의 전송률, 그리고 $stddev$ 는 표준 편차이다.

$$\frac{stddev\{c_0, c_1, \dots, c_{n-1}\}}{\frac{1}{n} \times \sum_{i=0}^{n-1} c_i} \dots \dots \dots \text{식 (4)}$$

이 값이 적을수록 네트워크 자원들에서 자원을 예약받기 쉽다. 제안 알고리즘의 전송률 변화량이 1993 Final Four에서 대부분 버퍼 크기에서 높는데 이는 인접되는 GOP간의 바이트 수의 차이를 고려하지 않기 때문이다. Seminar 비디오 데이터의 경우 버퍼 크기가 커질수록 전송률 변화량이 커지는데, 이는 프레임 바이트 수의 변화가 적은 비디오 데이터의 경우 버퍼가 커질수록 1개의 전송률로 보낼 수 있는 구간의 크기가 커지면서 전송률의 크기도 크게 변화되었기 때문이다.



[그림 13] 전송률 변화량 비교

5. 결론 및 추후 연구 방향

CBA, MCBA, MVBA 등의 기존 스무딩 알고리즘들에서는 GOP를 구성하는 프레임간의 바이트 수의 차이가 심한 경우에도 저장된 순서대로 전송 계획을 세우기 때문에 특정 프레임의 바이트 수가 심하게 큰 경우에는 이 프레임 때문에 전송률을 변화시켜야 되므로 불필요하게 전송률 변화 횟수가 증가되거나 침두 전송률, 그리고 전송률 변화량이 커질 수 있다.

따라서, 본 논문에서는 이러한 기존 스무딩 알고리즘들의

문제점을 개선하기 위하여 백트래킹 방법을 이용하여 GOP를 구성하는 프레임들의 바이트 수가 가장 완만하게 변화될 수 있는 프레임들의 순서를 검색하여 이 순서에 의해 전송 계획을 세우는 스무딩 알고리즘과 이를 위한 구조를 제안하고 다양한 비디오 소스를 사용하여 MVBA 알고리즘과 성능을 평가하였다. 그 결과, 제안 알고리즘은 MVBA 알고리즘에서 보다 전송률 변화 횟수는 적고 침두 전송률은 동일하거나 적고 전송률 변화량은 특정 버퍼 크기를 제외하고 우수한 결과를 보였다. 그러나, 제안 알고리즘에서는 프레임의 순서를 재조합해야 하는 오버헤드가 요구된다. 따라서, 제안 알고리즘은 제한된 네트워크 대역폭을 갖고 있는 서버에서 스무딩 알고리즘의 목적을 만족시키면서 침두 전송률과 서버와 네트워크 간의 전송률 변화로 인한 통신 오버헤드를 줄이려는 경우에 유용하다. 특히, GOP를 구성하는 프레임들의 바이트 수의 변화가 큰 경우에 보다 유용한 방법이 될 것이다.

추후에는 백트래킹 방법에서 효율적인 가치치기 함수에 관한 연구와 제안 알고리즘과 다른 스무딩 알고리즘들과의 성능을 다양한 평가 요소를 사용하여 평가할 예정이다.

참고문헌

- [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, S. Sechrest, "An Optimal Bandwidth Allocation Strategy for the Delivery of Compressed Preencoded Video", ACM/Springer-Verlag 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, S. Sechrest, "Critical Bandwidth Allocation for the Delivery of Compressed Pre-recorded Video," Computer Communications, Vol 18, No. 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]곽준원, 이면재, 송하윤, 박도순, MPEG 동영상 전송을 위한 GOP 단위의 최소 변경 대역폭 할당 기법, 한국정보처리학회논문지 C 2002.10.
- [10] ELLIS HOROWITZ, SARTAJ SAHNI, "Fundamental of computer algorithm", computer science press, 1978.
- [11] C. Lin, et. al., "MPEG Video Streaming with VCR Functionality," in IEEE Trans. on C&S for Video Tech., Vol.11, No.3 p.415-425, Mar 2001.
- [12] <http://www.cis.ohio-state.edu/~wuchi>.



이 면 재

1994년 2월 22일 홍익대학교 전자계산학과 석사 과정 졸업
 2005년 2월 22일 홍익대학교 전자 계산학과 이학박사.
 1994년 1월 1일 ~ 1999년 6월 30일 정원엔 시스템 연구소 근무
 2000년 5월 ~ 2001년 코리아 홈넷 수석 연구원
 2004년 9월 1일 ~ 현재 인터메이저 선임 연구원
 관심분야: 게임 프로그래밍, 게임 디자인, 멀티미디어, 멀티미디어
 통신

논문투고일 - 2006년 5월 11일
 심사완료일 - 2006년 6월 22일