다중 시각 객체 전송을 위한 최적화 적응 프레임워크

(An Optimal Adaptation Framework for Transmission of Multiple Visual Objects)

임 정 연 † 김 문 철 ††
(Jeongyeon Lim) (Munchurl Kim)

요 약 인터넷의 발달과 함께 멀티미디어 스트리밍은 비디오 콘텐츠를 전달하는데 중요한 수단이 되었으며 스트리밍 되는 데이타의 양도 점점 증가하고 있다. 하지만 스트리밍의 양이 증가함에 따라 가변적 네트워크 대역폭을 갖는 IP 네트워크 환경 상에서 실시간 QoS를 보장하기 어려워진다. 본 논문에서는 한정된 가용 대역폭과 트랜스코딩 코스트를 고려하면서 동시에 다중 앵글 콘텐츠의 인지적 화질을 최대화하는 최적화 적응 프레임워크를 제안한다. 제안된 다중 앵글 서비스 프레임워크에서 사용자는 같은 이벤트에 대해 다른 뷰 앵글로 캡쳐된 다중 비디오 스트림들 중에 사용자가 원하는 뷰를 선택할 수 있다. 이러한 향상된 서비스는 시간에 따른 네트워크 대역폭 변화와 같은 네트워크의 스트리밍 문제를 수반한다. 이를 위해 본 논문에서는 제안된 비디오 적응 프레임워크에서 다중 앵글 비디오 콘텐츠는 주어진 대역폭에 맞게 전달하기 위해 다른 비트율로 인코딩되며 적절한 비디오 스트림이 선택되거나 트랜스코딩될 수 있다고 가정한다. 각 사용자의 선호도에 따른 다양한 위치에서 캡쳐된 콘텐츠의 선택적 소비를 위해, 각 앵글에 제공되는 비트스트림은 다양한 비트율로 인코딩될 수 있고, 제약된 대역폭 환경에서 실시간으로 변하는 비트율에 최적화된 콘텐츠를 전송하기 위해 콘텐츠를 트랜스코딩되거나 서브 비트스트림이 선택되도록 한다. 이러한 가정을 기반으로 제안된 최적화 적응 프레임워크에서 트랜스코딩하는데 걸리는 시간을 트랜스코딩 코스트로 정의하고 주어진 트랜스코딩 코스트와 제한된 네트워크 대역폭을 만족하며 동시에 인지적 화질을 최대화하는 방법을 수식화 한다. 또한 제안된 방법의 효율성을 증명하기 위한 다양한 실험 결과를 보인다.

키워드 : 다중 앵글 비디오 적응, 최적화, 분리계획법

Abstract With the growth of the Internet, multimedia streaming becomes an important means to deliver video contents over the Internet and the amount of the streaming multimedia contents is also getting increased. However, it becomes difficult to guarantee the quality of service in real-time over the IP network environment with instantaneously varying bandwidth. In this paper, we propose an optimal adaptation framework for streaming contents over the Internet in the sense that the perceptual quality of the multi-angle content with multiple visual objects is maximized given the constraints such as available bandwidth and transcoding cost. In the multi-angle video service framework, the user can select his/her preferred alternate views among the given multiple video streams captured at different view angles for a same event. This enhanced experience often entails streaming problems in real-time over the network, such as instantaneous bandwidth changes in the Internet. In order to cope with this problem, we assume that multi-angle video contents are encoded at different bitrates and the appropriate video streams are then selected or transcoded for delivery to meet such bandwidth

† 정 회 원 : SK 텔레콤 Network기술원 매니저
jeongyeon@sktelecom.com
†† 정 회 원 : 한국정보통신대학교 공학부 교수
mkim@icu.ac.kr
논문접수 : 2005년 12월 1일
심사완료 : 2008년 2월 18일

Copyright@2008 한국정보과학회 : 개인 목적이나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.
정보과학회논문지: 소프트웨어 및 응용 제35권 제4호(2008.4)

constraints. For the user selective consumption of the various bitstreams in the multi-angle video service, the bitstream in each angle can be encoded in various bitrate, and the user can select a sub-bitrstream in the given bitrstreams or transcode the corresponding content in order to deliver the optimally adapted video contents to the instantaneously changing network condition. Therefore, we *define the transcoding cost which means the time taken for transcoding* the video stream and formulate a unified optimization framework which maximizes the perceptual quality of the multiple video objects in the given constraints such as the transcoding cost and the network bandwidth. Finally, we present plenty of the experimental results to show the effectiveness of the proposed method.

Key words : Multi-angle video adaptation, Optimization, Separable Programming

1. 서 론

인터넷을 통한 멀티미디어 스트리밍 방법은 콘텐츠를 전달하는 기본적인 방법으로서 주요한 인터넷 어플리케이션이 되었다. 또한 네트워크의 속도와 대역폭의 향상에 따라 여러 비디오 스트림을 함께 전송함으로써 다양한 뷰(view)를 선택할 수 있도록 하는 다중 앵글 비디오 서비스(multi-angle video service)가 가능하게 되었다. 다중 앵글 비디오 서비스는 여러 대의 카메라로부터 획득된 영상을 바탕으로 사용자에게 다양한 뷰를 제공함으로써 사용자가 원하는 뷰를 선택해서 시청할 수 있도록 한다. 다중 앵글 비디오 서비스는 IPTV어플리케이션[1] 중의 하나로 사용자가 원하는 위치를 골라 선택하여 볼 수 있도록 하며, 방송 기술 표준인 TV Anytime에서도 Multi-parallel stream linear AV program이라는 킬러 어플리케이션으로 소개되고 있다[2]. 그림 1은 다중 앵글 비디오 서비스의 예로 그림에서 보이는 바와 같이 축구와 같은 스포츠 프로그램에서 각각의 카메라에 의해 다양한 각도의 뷰가 보여질 수 있다. 예를 들어 골이 터졌을 경우, 골문 앞, 뒤, 좌, 우 혹은 특정 선수 등의 다양한 장면을 보일 수 있다. 다양하고 풍부한 장면의 뷰를 제공하는 다중 앵글 비디오 서비스를 위해 해당 사용자 터미널에 렌더링 가능하도록 가능한 모든 앵글의 비디오 스트림이 제공되어야 한다.

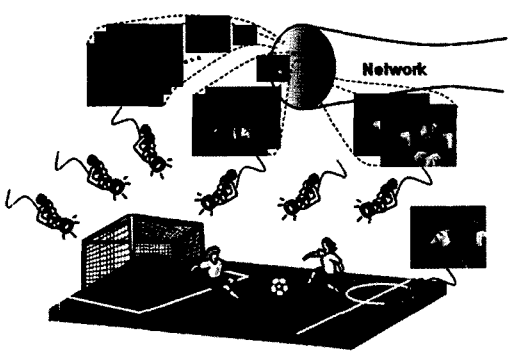


그림 1 다중 앵글 비디오 서비스의 예

하지만 실시간의 콘텐츠 전달은 네트워크 대역폭의 변화에 민감하며, 인터넷을 이용한 스트리밍 서비스의 증가로 각 미디어 스트림이 요구하는 전송률을 보장하기 어렵다. 또한 사용자 환경에 따른 적절한 대응도 쉽지 않다. 따라서 가변적인 대역폭 특성을 갖는 인터넷 상에 미리 인코딩된 비디오 압축 비트스트림을 적응적으로 스트리밍하기 위한 많은 스케일러빌리티 방식이 연구되었다.

이전 연구에서는 주로 MPEG-4 시스템 장면 정보에서 제공되는 다양한 비디오 객체들 가운데 사용자 환경을 고려해 적절한 객체를 고르는 다양한 방법이 제안되었다[3,4]. 이러한 연구들에서는 각 비디오 객체가 다양한 압축 비트율로 표현된 여러 레벨의 비트스트림을 가진다고 가정한다. 이 경우, 스트리밍 되기 전에 각 객체를 다양한 비트율로 압축한 여러 비트스트림이 각 레벨로 미리 준비되어야 한다. 하지만 시간에 따라 변하는 네트워크 대역폭에 맞게 비트율을 조절하기 위한 다양한 레벨의 비디오 스트림을 준비하기 어렵다.

본 논문에서는 다양한 객체를 제공하는 멀티미디어 어플리케이션 환경을 기반으로 주어진 네트워크 대역폭 제한에서 다중 객체 스트림을 전송할 수 있도록 하는 다중 비디오 콘텐츠의 적응 방식을 제안한다. 주어진 비디오 객체는 다양한 비트율을 갖는 하나 이상의 압축된 여러 비트스트림 혹은 버전을 갖는다고 가정한다. 주어진 네트워크 대역폭 상에서 최고의 화질을 갖는 객체의 버전 집합을 선택하기 위해 각 객체는 다양한 비트율을 갖는 비트스트림들, 즉, 여러 버전들을 가질 수 있고, 여러 객체 중 더 중요한 객체에 보다 더 나은 서비스가 되도록 비디오 객체 우선 순위(priority)를 부여할 수 있다. 또한 제한된 네트워크 대역폭에 따라 객체는 실시간으로 GOP(group of picture) 단위로 트랜스코딩 (transcoding) 되어 전송될 수 있다고 가정하고, 각 객체를 트랜스코딩하는데 걸리는 시간을 트랜스코딩 코스트로 정의한다. 본 논문에서는 주어진 제한식(constraints)을 만족하는 동시에 인지적 화질(perceptual quality)을 최대로 하며 선택(selection)과 트랜스코딩(transcoding)을

함께 고려하는 최적화 방법을 제안하고 모델링한다. 하지만 이는 각 객체의 적절한 버전 선택과 트랜스코딩의 여부를 함께 고려하기 어렵고 비트율과 PSNR(peak signal to noise rate) 값 간의 관계가 비선형이므로 비선형 모델로 표현된다. 따라서 다른 비트율과 PSNR 값을 갖는 버전을 R-D(rate-distortion) 커브의 불연속 점으로 보고 R-D 커브를 선형으로 근사화하여 비선형 계획법(nonlinear programming, NLP)의 한 방법인 분리 계획법(separable programming)으로 표현하여 해결한다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구를 기술하고 제안된 아이디어와 비교하며, 3장에서는 제안된 최적화 적응 프래임워크를 표현하기 위한 최적화 방법을 이용한 간략한 모델을 소개한다. 4장에서는 비선형 계획법 중의 하나인 분리 계획법에 대해 설명하고 제안된 적응 프래임워크에 분리 계획법에 적용시킬 수 있는 방법에 대해 기술한다. 5장에서는 제안된 모델 및 풀이 방법을 바탕으로 선택된 객체의 집합에 대한 실험 및 결과를 분석하며 마지막으로 6장에서 결론을 맺는다.

2. 관련 연구

지금까지 시간에 따라 변하는 대역폭을 갖는 인터넷에 압축 비디오 비트스트림을 적응적으로 스트리밍하기 위한 다양한 스케일러블 방식이 연구되어 왔다. 배낭 문제(knapsack problem)을 이용한 객체 기반 스케일러빌리티 방식 중의 하나로, Batra[3]는 저작 및 멀티플렉싱 과정에서 정수 계획법(integer programming)에 라그랑지 풀이(Lagrange relaxation) 방식을 이용하여 여러 콘텐츠를 보내는 경우 전송될 객체의 적합한 셋을 고르는 방식을 제안하였으며, Cha[4]는 MPEG-4 시스템 기반의 저작 툴에서 사용자 환경에 따른 장면 정보의 트랜스코딩 및 탐욕 알고리즘(greedy algorithm)을 이용하여 MPEG-4와 같은 다중 객체 환경에 사용자에게 제한된 환경과 사용자의 비디오 객체 우선 순위에 따라 제공될 객체를 선택하는 방법에 대해 고려하였다. 여기서 MPEG-4 콘텐츠 객체는 오디오, 비디오, 이미지, 그래픽 및 텍스트 데이타 등 다양한 여러 개의 객체로 구성될 수 있다. 각 객체는 여러 버전으로 서로 다른 비트율로 압축되어 표현될 수 있고 각 버전의 비트율은 하나의 스케일러빌리티 레벨로 표현된다. 이러한 기존의 방식들은 선택된 객체의 총 규모가 제한된 대역폭보다 작아야 하며 스트리밍 되기 전에 각 객체에 대해 여러 레벨(버전) 중 하나를 선택해야 한다. 이러한 방법은 콘텐츠를 저작(authoring) 혹은 멀티플랙싱(multiplexing) 단계에서 적절한 객체의 집합을 고르므로 각 비디오 객체에서는 다양한 버전이 미리 준비되어야 하고 시간에 따라 변하는 네트워크 대역폭의 변화에 적절하게 대응하기 어려우며 미리 제공되어야 하는 버전도 매우 다양해야 하는 단점이 있다. 즉, 각 멀티미디어 콘텐츠 객체에 주어진 버전을 선택하는 기능만을 제공하고 있다.

또한, 실시간으로 변하는 인터넷의 대역폭에 맞춰 원하는 화질의 콘텐츠를 실시간으로 제공하기 위해 인코딩된 비디오에서 확장 혹은 트랜스코딩된 버전을 추출하는 다양한 트랜스코딩 방법이 연구되었다[5]. 트랜스코딩 방법으로 크게 DCT 계수의 재양자화나 DCT 계수의 선택적 제거와 같은 공간축 상에서 트랜스코딩(spatial domain transcoding), 프래임 제거(frame dropping)과 같은 시간축 상에서 트랜스코딩(temporal domain transcoding), 해상도의 축소방법 및 객체에 우선순위를 부여하는 객체 기반의 트랜스코딩 및 여러 트랜스코딩을 함께 고려한 스케일러빌리티 방법 등이 있다[6-9]. 또한 MPEG-4 FGS(fine granular scalability)와 같은 하나의 기본 레이어와 여러 개의 강화된 레이어를 이용하여 비트스트림을 조절하는 방식도 제안되었다[10]. 이러한 다양한 방식을 이용함과 동시에 IO 코스트와 같은 파라미터와 CPU 코스트와 같은 트랜스코딩 오버헤드를 함께 고려하는 이슈들과 트랜스코딩 코스트를 결정하고 측정하기 위한 프래임워크들이 연구되었다[11-13]. 이러한 연구들은 UMA(universal multimedia access)의 환경을 지원할 수 있도록 하는 다양한 이종의 단말에 주로 단일의 비디오 콘텐츠를 적응하는 방법에서 주로 논의 되고 있거나 특정 코덱에 제한적인 방법만을 제안하고 있다.

본 논문에서는 한 화면에 여러 객체를 지원하는 다중 앵글 비디오 서비스에서 주어진 네트워크 대역폭과 트랜스코딩하는데 걸리는 시간을 함께 고려하여 최고의 화질을 가지며, 주어진 객체의 다양한 버전 중 시간에 따라 가변하는 네트워크 대역폭에 가장 잘 맞도록 각 객체의 버전을 선택하거나 현재의 가변적 환경에 잘 맞도록 트랜스코딩하기에 적합한 객체를 선택하는 방법을 제안한다.

3. 다중 선택 기반의 선택 및 트랜스코딩 프래임워크

가변적 네트워크 대역폭에 적합한 스트림을 전송하기 위해 서버는 네트워크 환경을 모니터링하며, 현재의 네트워크 상황에 맞는 스트림을 보내기 위해 선택된 콘텐츠를 트랜스코딩할 수 있다고 가정한다. 가변적인 네트워크 환경에서 콘텐츠 내에 객체들은 그것의 중요도 혹은 유용성에 따라 특정 객체에 트랜스코딩 연산을 부여할 수 있다. 예를 들어, 네트워크의 가능한 대역폭이 현재 스트리밍되는 콘텐츠의 양보다 작은 경우, 현재 선택된 객체들 중에서 트랜스코딩을 수행하거나 객체의 선

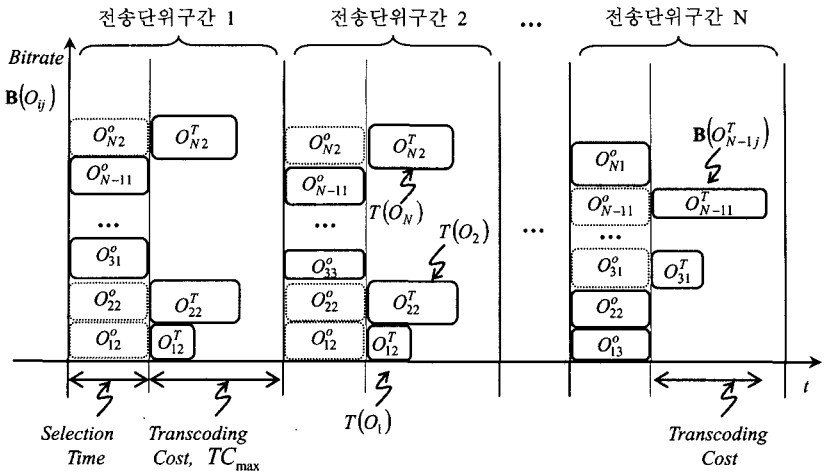


그림 2 스트리밍될 객체 선택 및 트랜스코딩

택을 다시 수행할 수 있고, 반대로 네트워크의 가능한 대역폭이 현재 스트리밍되는 콘텐츠의 양보다 큰 경우, 현재 선택된 객체들 중 특정 객체를 골라 더 나은 화질을 제공하도록 할 수 있다. 그림 2는 시간에 따라 변하는 네트워크 대역폭에 따라 트랜스코딩이 적용되는 과정을 설명한다.

그림 2는 스트리밍 될 객체의 선택 및 트랜스코딩을 위해 선택된 객체들의 시간 단위 당 객체 집합을 보인다. 각 객체는 비트율에 따른 여러 버전과 해당 PSNR 값을 가지며 한 객체를 트랜스코딩하는데 걸리는 시간인 트랜스코딩 시간 정보를 미리 갖고 있다고 가정한다. 시간 축과 전송 단위 구간(Transmission Time Interval)당 전송될 콘텐츠의 비트율 축으로 구성되어 있다. 전송 단위 시간은 콘텐츠를 스트리밍할 때 콘텐츠에 접근하고 전송되기까지의 총 시간을 의미하며, 각 객체의 알맞은 버전을 선택하기 위한 선택 시간(Selection Time)과 선택된 객체에 좀 나은 화질을 얻기 위한 트랜스코딩 과정에 걸리는 트랜스코딩 시간(Transcoding Time)으로 할당할 수 있다. 여기서 트랜스코딩이라 함은 여러 방법이 쓰일 수 있으며, 본 논문에서 쓰일 트랜스코딩 방법은 보내질 비트율을 조정하기 위해 각 객체의 최고 화질(비트율)을 갖는 버전에 대한 압축된 콘텐츠를 GOP 단위로 디코딩하고 목표 비트율로 재압축하는 과정을 의미한다. 즉, 각 객체의 트랜스코딩 시간을 각 객체의 트랜스코딩 오버헤드를 측정하기 위한 트랜스코딩 코스트라고 부른다. O^o_{ij}는 선택된 i 번째 객체의 j 번째 원래의 버전을 나타내며 $j = 1, 2, ..., n_i$ 이고 그것의 비트율은 $B(O^o_{ij})$로 표현된다. 트랜스코딩 시간(Transcoding Time) 상에 O^T_{ij}는 i 번째 객체의 j 번째 버전에 대한 트랜스코딩으로 목표 비트율이 감소된 객체를 의미한다. 첫번째와 두 번째 전송 단위 구간에서 첫 번째와 두 번째 그리고 N 번째 객체의 트랜스코딩 코스트는 $T(O_1)$, $T(O_2)$와 $T(O_N)$으로 표현된다. 첫번째와 두번째 전송 단위 구간에서 객체 N개를 선택하고 더 나은 화질을 위해 N개의 객체들 중 첫 번째(O_1)와 두 번째 객체(O_2) 그리고 N 번째 객체(O_N)를 트랜스코딩한다. 마지막 전송 시간 구간에서는 네트워크 대역폭의 감소로 두 개의 트랜스코딩된 객체, 즉, N-1번 째(O_{N-1})와 세 번째 객체(O_3)의 트랜스코딩을 허용하며 첫 번째 객체는 두 번째 버전(O_{12})에서 세 번째 버전(O_{13})으로 바꿨다.

그림 2에서와 같이 콘텐츠의 트랜스코딩이 이루어질 경우, 두 가지 제약 조건인 최대 비트율과 최대 트랜스코딩 코스트에 의해 최고의 효용성(utility)을 갖는 콘텐츠를 전송하게 된다. 선택 알고리즘을 위한 모델은 최적화 방법으로 표현된다. 본 논문에서는 직관적인 화질을 표현하기 위해 PSNR을 이용하며 제한된 최대 허용 비트율과 트랜스코딩 코스트를 이용하여 제한식을 만든다. 이는 제한된 환경을 표현하기 위한 제한 식과 콘텐츠 인지적 화질을 최대화하는 목적 함수로 다음과 같이 구성된다.

$$Max\left(\underbrace{\sum_i\sum_j P_{SNR}(O^o_{ij})\cdot P_{rior}(O_i)\cdot Y(O^o_{ij})}_{\text{각 객체에 존재하는 버전이 선택된 경우}}+\underbrace{\sum_i\sum_j P_{SNR}(O^T_{ij})\cdot P_{rior}(O_i)\cdot Y(O^T_{ij})}_{\text{트랜스코딩된 버전이 선택된 경우}}\right) \quad (1)$$

제한식:

$$\underbrace{\sum_i\sum_j B(O^o_{ij})\cdot Y(O^o_{ij})}_{\text{존재하는 버전의 비트율}}+\underbrace{\sum_i\sum_j B(O^T_{ij})\cdot Y(O^T_{ij})}_{\text{트랜스코딩된 버전의 비트율}}\le B_{\max} \quad (2)$$

$$\sum_j T(O_{ij}^T)\cdot Y(O_{ij}^T) \le TC_i \tag{3}$$

$$\sum_j \left(Y(O_{ij}^o)+Y(O_{ij}^T)\right)=1 \tag{4}$$

주어진 식 (1)에서 최고의 효율을 얻기 위해 각 객체 O_{ij}의 PSNR 값을 $P_{SNR}(O_{ij})$로 하여 효용성으로 표현하였으며, 각 객체는 순위는 $P_{rior}(O_i)$로 표현되어 높은 순위를 갖는 객체가 선택될 수 있도록 PSNR과의 곱으로 표현된다. $Y(O_{ij}^o)$와 $Y(O_{ij}^T)$는 선택의 여부를 나타내는 0 또는 1의 값을 가지며, 원래의 객체로 선택될 경우 $Y(O_{ij}^o)$로, 트랜스코딩을 하게 될 경우 $Y(O_{ij}^T)$로 표기된다. 최고의 PSNR을 갖는 객체들 중 네트워크의 최대 대역폭 B_{max}보다 작은 비트율을 가져야 하고, 트랜스코딩을 수행할 경우 제한된 시간 내에서 수행되어야 하므로 각 객체의 최대 트랜스코딩 코스트인 TC_i보다 작아야 한다. TC_i는 콘텐츠를 제공받는 각 디코더의 프래임율에 따라 결정될 수 있다.

위에서 제안한 바와 같이, 제안된 선택 알고리즘은 각 객체의 모든 버전 정보가 주어져 있고 객체 또는 버전의 선택만 고려한다면 이진 정수 계획법(binary integer programming, BIP)으로 쉽게 풀 수 있다[14]. 그러나 위의 제안된 식 (1) - (4)에서 시간에 따라 가능한 대역폭의 변화에 따라 현재의 대역폭에 거의 비슷하게 맞추기 위해 어떤 객체를 얼마나 트랜스코딩할 것인지 결정하기 어렵다. 또한 트랜스코딩을 얼마나 수행할 지와 관련하여 PSNR과 비트율의 관련식이 요구된다. 하지만 PSNR 값과 비트율의 관련식을 구하기 어렵고, PSNR 값과 비트율의 커브는 비선형(non-linear)이므로 선형 계획법(linear programming)이나 이진 정수 계획법 방식으로는 풀이가 불가능하다. Batra[3]의 경우, 비선형 R-D 커브의 관계를 제공된 정보에 따라 불연속의 형태로 나타내며 이를 이진 정수 계획법으로 계산한다. 또한 많은 수의 객체 가운데 적절한 객체의 집합을 고르기 위한 시간의 복잡도를 줄이기 위해 라그랑지 풀이 방식을 이용하여 분기 한정법(branch and bound) 방식의 가지를 줄여나가게 된다[3,14]. 이 경우, 실시간의 트랜스코딩을 고려하지 않으며 제공된 버전의 선택 및 많은 수의 객체 가운데 적절한 객체의 집합을 고르고 그것의 복잡도를 줄이는 방법에 대해 고려하므로 이진 정수 계획법의 적용이 타당하다. 그러나 본 논문에서는 객체의 선택 및 트랜스코딩을 함께 고려하므로 PSNR 값과 비트율의 비선형 관계를 근사화로 표현한 분리 계획법(separable programming)[15]을 적용하고 다음 장에서 이를 자세히 설명한다.

4. 분리 계획법의 적용

4.1 분리 계획법 소개

분리 계획법은 비선형 계획법의 하나로 비선형 모델을 선형의 형태로 풀 수 있는 유형이며, 일반적으로 분리 계획법은 아래와 같이 나타낼 수 있다.

$$Maximize \sum_{j=1}^{n} f_j(x_j) \tag{5}$$

제한식:

$$\sum_{j=1}^{n} g_{ij}(x_j) \le 0 \qquad (i=1,2,\cdots,m) \tag{6}$$

목적 함수(Objective function)가 각 함수 f_j와 제한식에 각 함수 g_j에 결정 변수(decision variable)이 각각 표현되기 때문에 이러한 경우 분리 가능하다고 표현한다. 식 (5)의 목적함수는 $f(x) = f_1(x_1) + f_2(x_2) + f_3(x_3)$로 표현될 수 있고, 식 $f_1(x_1)$, $f_2(x_2)$, $f_3(x_3)$이 각각의 식을 가지고 있다면 이를 분리 가능하다고 말한다. 또한 선형의 제한식도 분리 가능하다. 목적 함수의 최대 값을 찾는 경우 목적함수의 커브는 오목(concave)해야 하고 이를 직접 푸는 대신, 선형 계획 방식으로 풀 수 있도록 다음과 같이 어림 값(approximation)을 이용한다.

그림 3과 같이 일반적으로 비선형인 함수 $f(x)$를 가정한다. r개의 세그먼트를 이용하여 불연속 선형 근사값을 형성하기 위해 $r+1$개의 불연속 점이 요구된다.

그림 3에서 실선은 일반적인 비선형 함수 $f(x)$를 나타내고 직선의 점선은 비선형 함수의 불연속 선형 근사값 $\hat{f}(x)$을 나타낸다. 만약 어떤 두 점($\bar{x}_{k-1}, \bar{x}_k$) 사이에 해 x가 존재한다면, 다음과 같이 표현할 수 있다.

$$x = \lambda_{k-1}\cdot\bar{x}_{k-1} + \lambda_k\cdot\bar{x}_k, \quad \lambda_{k-1}+\lambda_k = 1 \tag{7}$$

$$\hat{f}(x) = f(\bar{x}_{k-1})\cdot\lambda_{k-1} + f(\bar{x}_k)\cdot\lambda_k = f(\bar{x}_{k-1})\cdot\lambda_{k-1} + f(\bar{x}_k)\cdot(1-\lambda_{k-1}) \tag{8}$$

$$0 \le \lambda_i \le 1$$

여기서 음수가 아닌 λ_{k-1}, λ_k는 선택된 $\bar{x}_{k-1}$의 값 $f(\bar{x}_{k-1})$와 $\bar{x}_k$의 값 $f(\bar{x}_k)$에 가중치로 새로운 해 x를 식 (8)과 같이 표현한다.

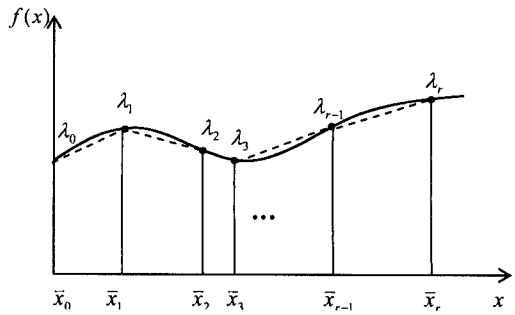


그림 3 비선형 함수와 불연속 선형 근사값

위 식에 대한 전체의 근사화된 식 $\hat{f}(x)$는 다음과 같이 표현될 수 있다.

$$\hat{f}(x) = f(\bar{x}_0)\cdot\lambda_0 + f(\bar{x}_1)\cdot\lambda_1 + \cdots + f(\bar{x}_{r-1})\cdot\lambda_{r-1} + f(\bar{x}_r)\cdot\lambda_r = \sum_{i=0}^{r} f(\bar{x}_i)\cdot\lambda_i \tag{9}$$

$$x = \lambda_1\cdot\bar{x}_1 + \lambda_2\cdot\bar{x}_2 + \cdots + \lambda_{r-1}\cdot\bar{x}_{r-1} + \lambda_r\cdot\bar{x}_r = \sum_{i=0}^{r}\lambda_i\cdot\bar{x}_i \tag{10}$$

$$\lambda_1 + \lambda_2 + \cdots \lambda_{r-1} + \lambda_r = \sum_{i=0}^{r}\lambda_i = 1 \tag{11}$$

비트율과 PSNR의 실제 식을 사용하지 않고 그들 간에 산출된 불연속의 값을 이용하여 근사화된 값을 사용하므로 불연속의 값이 많을수록 실제 커브와 비슷해지며, 커브가 오목한 가운데 두 불연속 점간의 어림 값을 이용하므로 얻게 되는 최대 목적함수 값은 커브 안쪽 혹은 커브 상에 존재하게 된다. 비트율과 PSNR의 관계식은 오목하게 나타나므로 연속된 불연속 점 간에 값이 존재하는 인접 조건(adjacency condition)을 만족한다. 더 정확한 값을 위해서는 더 자세한 비트율 및 PSNR과의 관계가 요구된다[14,15].

4.2 분리 계획법의 적용

한 객체는 여러 버전을 가질 수 있으며 네트워크의 상황 및 조건에 따라 적절한 객체를 골라 비트율에 따라 트랜스코딩할 수 있다. 또한 트랜스코딩이 항상 최고의 화질을 갖는 버전에서 이루어지므로 트랜스코딩이 수행된 객체 $Y(O_i^T)$는 인덱스 j가 제거될 수 있다. 처음 제안된 식에서 트랜스코딩이 항상 최적의 화질, 즉, 원본에서 이루어진다면 식 (1)~(4)는 다음과 같이 표현될 수 있다.

$$Max\Bigg(\underbrace{\sum_i\sum_j P_{SNR}(O_{ij}^o)\cdot P_{rior}(O_i)\cdot Y(O_{ij}^o)}_{\text{각 객체에 존재하는 버전이 선택된 경우}} + \underbrace{\sum_i P_{SNR}(O_i^T)\cdot P_{rior}(O_i)\cdot Y(O_i^T)}_{\text{트랜스코딩된 버전이 선택된 경우}}\Bigg) \tag{12}$$

제한식:

$$\sum_i\sum_j \underbrace{B(O_{ij}^o)\cdot Y(O_{ij}^o)}_{\text{존재하는 버전의 비트율}} + \sum_i \underbrace{B(O_i^T)\cdot Y(O_i^T)}_{\text{트랜스코딩된 버전의 비트율}} \le B_{\max} \tag{13}$$

$$T(O_i)\cdot Y(O_i^T) \le TC_i \tag{14}$$

$$\sum_j Y(O_{ij}^o) + Y(O_i^T) = 1 \tag{15}$$

비트율에 따라 트랜스코딩되면 객체의 PSNR 값이 결정된다. 위의 식 (12)와 (13)은 아래와 같이 정리될 수 있다.

$$Max\Bigg(\sum_i P_{rior}(O_i)\cdot\underbrace{\Bigg(\sum_j P_{SNR}(O_{ij}^o)\cdot Y(O_{ij}^o) + P_{SNR}(O_i^T)\cdot Y(O_i^T)\Bigg)}_{f_{PSNR,i}(Y(O_i))\text{의 함수로 표현함}}\Bigg) \tag{16}$$

$$\sum_i\underbrace{\Bigg(\sum_j B(O_{ij}^o)\cdot Y(O_{ij}^o) + B(O_i^T)\cdot Y(O_i^T)\Bigg)}_{g_{Bitrate,i}(Y(O_i))\text{의 함수로 표현함}} \le B_{\max} \tag{17}$$

위의 식 (16)의 두 번째 괄호와 (17)에 첫번째 괄호의 값은 한 객체당 오직 하나의 버전만을 표현한다. 즉, 각 객체 O_i에 n_i개의 버전 중 한 개의 버전이 선택되거나 각 객체에 존재하는 버전이 현재 네트워크 대역폭에 적응하기에 충분하지 않다면, 원래의 버전, 즉, 최적의 화질을 갖는 콘텐츠가 트랜스코딩될 수 있다. 식 (16)에 PSNR과 식 (17)에 비트율에 대한 식은 모든 가능한 PSNR 값과 비트율을 표현할 수 있다. 여기서 $Y(O_i)$를 객체 선택의 정도를 나타내는 값으로 정의하였다. 위의 식은 변수 $Y(O_i)$에 의해 표현될 수 있으며 식 (16)의 두 번째 괄호와 식 (17)의 괄호는 식 (18), (19)와 같이 각각 PSNR과 비트율의 함수로 표현될 수 있다.

$$Max\Bigg(\sum_i P_{rior}(O_i)\cdot f_{PSNR,i}(Y(O_i))\Bigg) \tag{18}$$

제한식:

$$\sum_i g_{Bitrate,i}(Y(O_i)) \le B_{\max} \tag{19}$$

$$T(O_i)\cdot Y(O_i^T) \le TC_i^{\max} \tag{20}$$

$$0 < Y(O_i) \le 1. \tag{21}$$

본 논문에서 객체 선택의 정도 $Y(O_i)$는 선택의 유무를 가리키는 식 (15)의 0 또는 1의 $Y(O_{ij}^o)$, $Y(O_{ij}^T)$와는 달리 (0,1]의 값을 가질 수 있다고 가정한다. 이 범위에서 0은 객체의 어떤 버전도 선택되지 않았음을 의미하고 1은 최고의 화질을 갖는 버전이 선택되었음을 의미한다. 0과 1 사이의 값은 객체가 트랜스코딩 되었거나 세부의 버전이 선택되었음을 의미한다. PSNR의 함수와 식 (18)에 목적함수는 모든 객체에 분리 가능한 함수의 합으로 표현될 수 있다.

$$f_{PSNR}(Y(O_i)) = f_{PSNR,1}(Y(O_1)) + f_{PSNR,2}(Y(O_2)) + \cdots + f_{PSNR,n}(Y(O_n)) \tag{22}$$

$$\sum_i P_{rior}(O_i)\cdot f_{PSNR}(Y(O_i)) = P_{rior}(O_1)\cdot f_{PSNR,1}(Y(O_1)) + P_{rior}(O_2)\cdot f_{PSNR,2}(Y(O_2)) + \cdots + P_{rior}(O_n)\cdot f_{PSNR,n}(Y(O_n)) \quad (23)$$

PSNR의 함수 $f_{PSNR,1}(Y(O_1)), f_{PSNR,2}(Y(O_2)), \cdots f_{PSNR,n}(Y(O_n))$는 비트율에 따라 오목하게 나타나며 이는 비선형으로 보여진다. 또한 선택의 정도에 따른 비트율도 함수로 표현되며 각 객체를 가리키는 인덱스 i에 따라 분리가능 해진다. 위의 PSNR 함수는 선택된 비트율에 따라 변하며 커브의 그림이 오목하므로 주어진 버전(인덱스 j)의 값을 이용하여 불연속의 선형 커브(linear curve)로 접근시킨다. 즉, 한 객체에 대해 여러 가지의 버전이 존재하므로 이 값을 이용하여 각 객체의 PSNR 함수를 근사값으로 표현할 수 있다. 즉, f_{PSNR}을 위한 근사 함수 f^a_{PSNR}은 불연속의 값에 가중치(weight)를 적용함으로써 표현될 수 있다. 따라서 PSNR의 근사함수는 다음과 같이 표현될 수 있다.

$$\hat{f}_{PSNR,i}(Y(O_i)) = P_{SNR}(O_{i1})\cdot\lambda_{i1} + P_{SNR}(O_{i2})\cdot\lambda_{i2} + \cdots + P_{SNR}(O_{in_i})\cdot\lambda_{in_i} = \sum_j^{n_i} P_{SNR}(O_{ij})\cdot\lambda_{ij} \quad (24)$$

또한 비트율의 함수도 다음과 같이 근사화시킬 수 있다.

$$\hat{g}_{Bitrate,i}(Y(O_i)) = B(O_{i1})\cdot\lambda_{i1} + B(O_{i2})\cdot\lambda_{i2} + \cdots + B(O_{in_i})\cdot\lambda_{in_i} = \sum_i^{n_i} B(O_{ij})\cdot\lambda_{ij} \quad (25)$$

위의 근사화된 식 (24), (25)에서 근사식을 구성하는 PSNR 값 $P_{SNR}(O_{ij})$과 비트율 $B(O_{ij})$는 실제 존재하는 각 객체의 버전과 존재하지 않으나 더 실제 곡선과 가깝게 근접하기 위한 가상의 버전으로 표현될 수 있다. 가상의 버전이 선택될 경우 트랜스코딩하여 얻을 수 있으며 트랜스코딩이 허용되지 않는 객체는 실제 존재하는 객체의 버전만을 이용하여 표현한다.

변수 λ_{ij}는 가중치를 나타내는 비음수(nonnegative) 값이며, $\lambda_{i1}+\lambda_{i2}+\cdots+\lambda_{in_i}=1$이다. 버전의 수가 많을수록 원래의 곡선에 가깝게 근접한다. 그림 4는 비트율을 조절함에 따라 변하는 PSNR의 곡선을 보이며 트랜스코딩의 정도와 비트율이 비례하므로 트랜스코딩과 PSNR의 관계도 함께 나타낼 수 있다.

트랜스코딩 코스트의 경우, 불연속점을 이용하여 어림된 트랜스코딩 코스트를 구한 후 선택된 트랜스코딩 코스트의 객체가 $Y(O_i)=Y(O_{ij})$를 만족하고 $Y(O_{ij})$가 존재하는 버전이라면 해당 객체의 트랜스코딩 코스트는 0이 된다. 예를 들어, 한 객체는 1,024 kbps와 512 kbps로 각각 인코딩된 두 버전을 갖고 있다고 하자. 1,024 kbps의 버전은 최고의 화질을 가지고 이를 원 버전이라고

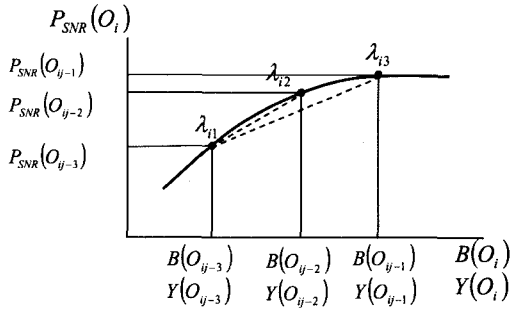


그림 4 트랜스코딩 정도에 따른 PSNR과 비트율의 관계

부른다. 만약 1,024 kbps로 인코딩된 버전이 선택된다면, 객체 선택의 정도인 $Y(O_i)$는 1이다. 만약 512 kbps로 인코딩된 버전이 선택된다면 객체 선택의 정도는 비트율과 비례한다고 가정했으므로 객체 선택의 정도인 $Y(O_i)$는 0.5가 된다. 하지만 512 kbps로 인코딩된 버전을 의미하고 객체 선택의 정도가 0.5인 버전이 이미 존재한다면, 객체를 트랜스코딩할 필요는 없다. 하지만 현재의 네트워크 상태가 256 kbps의 버전을 요구한다면 객체 선택의 정도인 $Y(O_i)$는 0.25가 되고 현재 가지고 있는 버전 중에 맞는 것이 존재하지 않으므로 트랜스코딩이 필요하다. 요구되는 비트율이 512 kbps인 경우, 트랜스코딩할 필요가 없으므로 0이 된다. 어떤 객체를 얼마나 트랜스코딩해야 할지 결정되지 않았으므로, 트랜스코딩 코스트도 존재하는 불연속점으로 아래와 같이 근사화가 가능하다.

$$\hat{t}_{TC,i}(Y(O_i)) = T(O_{i1})\cdot\{\lambda_{i1} - floor(\lambda_{i1})\} + T(O_{i2})\cdot\{\lambda_{i2} - floor(\lambda_{i2})\} + \cdots + T(O_{in})\cdot\{\lambda_{in} - floor(\lambda_{in})\} = \sum_j^{n_i} T(O_{ij})\cdot\{\lambda_{ij} - c\cdot floor(\lambda_{ij})\} \quad (26)$$

여기서 $c=1$는 O_{ij}의 해당 비트스트림이 존재할 경우를 가리키며 $c=0$는 해당 비트스트림이 존재하지 않음을 의미한다.

위의 식을 정리하면 아래와 같이 표현될 수 있다.

$$Max\left(\sum_i P_{rior}(O_i)\cdot\hat{f}_{PSNR,i}(Y(O_i))\right) \quad (27)$$

$$\sum_i \hat{g}_{Bitrate,i}(Y(O_i)) \le B_{\max} \quad (28)$$

$$\hat{t}_{TC,i}(Y(O_i)) \le TC_{\max} \quad (29)$$

$$0 < Y(O_i) \le 1$$

트랜스코딩 코스트의 제한 조건인 각 객체의 최대 트랜스코딩 코스트 $TC_{\max}$는 디코더에서 요구되는 각 객체의 프레임율 보다 작으며 각 객체의 트랜스코딩 코스트

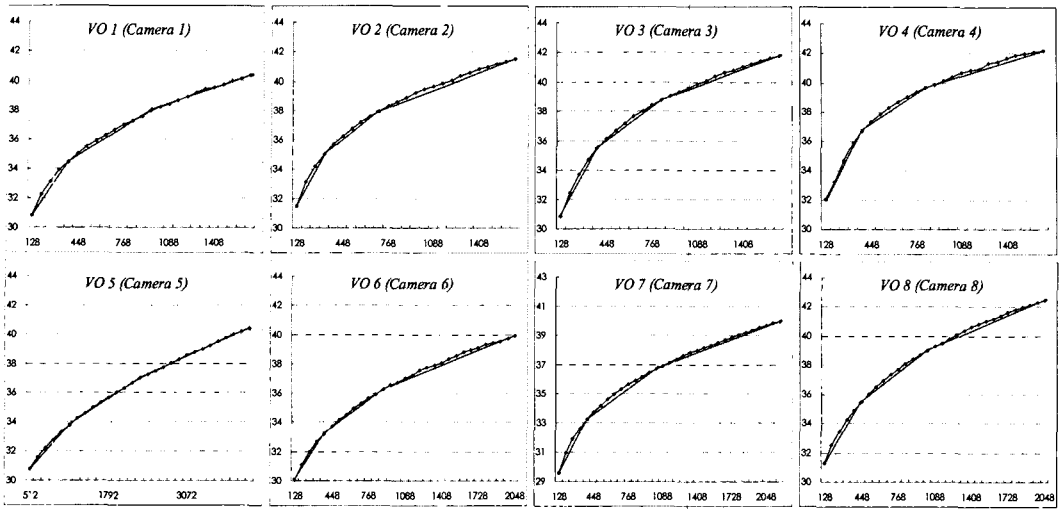


그림 5 비디오 객체의 R-D 커브와 근사화된 불연속 선형 커브

는 압축 방법, 해상도, 프레임율 등에 따라 달라질 수 있다. 예를 들어, 트랜스코딩 코스트가 큰 객체를 미리 제한함으로써 계산량을 줄일 수 있다.

5. 실 험

제안된 최적화 적응 프래임워크의 효율성을 보이기 위해 여덟 개의 비디오 객체를 이용하여 실시간으로 변하는 네트워크 대역폭의 변화에 따라 선택되거나 트랜스코딩되는 객체의 총 비트율 변화와 우선 순위가 부여된 비디오 객체의 인지적 화질을 비교한다. 그림 5는 해당 카메라로부터 캡쳐된 비디오 스트림의 R-D 커브와 근사화되어 표현된 불연속 선형 커브를 보인다.

여덟 개의 비디오 객체는 각각 3에서 5개의 버전을 가지고 있으며 각각의 비디오 객체는 해당 우선순위가 매겨져 있다. 이 비디오 객체들은 352×288의 해상도를 가지며 FFmpeg [16]을 이용하여 30 frame/s, 1 GOP = 16 frames인 MPEG-4 Visual (ISO/IEC 14496-2) Simple Profile로 인코딩되어 있다. 따라서 각 비디오 객체의 버전에서 트랜스코딩 코스트와 선택하는데 걸리는 시간의 합이 0.53(=16/30)초가 넘는다면, 해당 비디오 객체는 트랜스코딩 기능을 사용할 수 없다. 해당 비디오 스트림에 버전의 세부적 PSNR 정보와 비트율, 트랜스코딩 코스트 그리고 객체의 우선 순위 정보는 표 1과 같다.

실험에서 콘텐츠를 제공하는 데 있어 한 콘텐츠 당 여러 버전을 모두 가질 수는 없으므로, 각 객체에 회색으로 표기된 버전만 갖고 있고 나머지는 해당 비트율로 트랜스코딩 시 얻을 수 있는 PSNR과 트랜스코딩시 요구되는 트랜스코딩 코스트를 의미한다. 다양한 객체의 버전을 선택하는데 걸리는 시간은 모델의 구조와 밀집도 등을 표현하는 변수의 개수에 비례한다. 본 실험에서는 Lingo 9.0 [17]를 이용하였으며 표 1의 비디오 객체를 이용하여 정수 비선형 계획법을 풀 경우 3.0 GHz, 1 GRAM의 펜티엄 PC에서 대략 0.001~0.1초의 시간이 걸렸다. 따라서 최대 트랜스코딩 코스트 TC_{max}는 0.4초로 계산하였다.

그림 6은 시간에 따라 변하는 최대 네트워크 대역폭 B_{max}에 따라 트랜스코딩을 허용한 객체 선택의 방식과 트랜스코딩을 허용하지 않은 객체 선택 방식에 따른 비트율 변화를 보인다.

그림 6의 1번 점선은 시간에 따라 변하는 네트워크 대역폭을 나타내며 2번 선은 제안된 프래임워크인 트랜스코딩을 허용한 분리 계획법의 결과이고, 3번 선은 트랜스코딩을 허용하지 않은 객체 버전 알고리즘의 선택 방법이다. 트랜스코딩을 수행하는 경우 현재 주어진 대역폭에 맞도록 트랜스코딩할 수 있으므로 트랜스코딩이 가능한 제안된 프래임워크를 적용한 비트율의 결과는

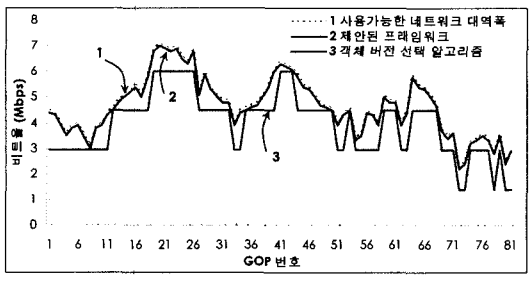


그림 6 시간에 따라 변하는 네트워크 대역폭에 따른 비트율의 변화

표 1 비디오 객체의 버전 및 버전의 비트율, PSNR, 트랜스코딩 코스트(TC)

VO 1 ($P_{rior}(O_1)$ = 0.4)			*VO 2* ($P_{rior}(O_2)$ = 0.4)		
비트율 $B(Y(O_1))$	PSNR $P(Y(O_1))$	TC $T(Y(O_1))$	비트율 $B(Y(O_2))$	PSNR $P(Y(O_2))$	TC $T(Y(O_2))$
128	30.9	0.046	128	31.5	0.047
384	34.5	0.054	320	35.0	0.052
960	38.0	0.059	704	38.0	0.060
1664	40.4	0.065	1664	41.5	0.068
VO 3 ($P_{rior}(O_3)$ = 0.3)			*VO 4* ($P_{rior}(O_4)$ = 0.3)		
비트율 $B(Y(O_3))$	PSNR $P(Y(O_3))$	TC $T(Y(O_3))$	비트율 $B(Y(O_4))$	PSNR $P(Y(O_4))$	TC $T(Y(O_4))$
128	30.9	0.044	128	32.0	0.053
384	35.5	0.053	384	36.7	0.059
832	38.7	0.058	832	39.7	0.066
1664	41.8	0.064	1664	42.4	0.072
VO 5 ($P_{rior}(O_5)$ = 0.2)			*VO 6* ($P_{rior}(O_6)$ = 0.2)		
비트율 $B(Y(O_5))$	PSNR $P(Y(O_5))$	TC $T(Y(O_5))$	비트율 $B(Y(O_6))$	PSNR $P(Y(O_6))$	TC $T(Y(O_6))$
512	30.8	0.056	128	30.0	0.045
1152	33.7	0.063	384	33.2	0.050
2304	37.0	0.069	896	36.2	0.056
4096	40.4	0.076	2048	39.9	0.063
VO 7 ($P_{rior}(O_7)$ = 0.1)			*VO 8* ($P_{rior}(O_8)$ = 0.1)		
비트율 $B(Y(O_7))$	PSNR $P(Y(O_7))$	TC $T(Y(O_7))$	비트율 $B(Y(O_8))$	PSNR $P(Y(O_8))$	TC $T(Y(O_8))$
128	29.6	0.046	128	31.3	0.045
384	33.2	0.056	448	35.5	0.052
1024	36.7	0.065	1024	39.0	0.059
2176	40.0	0.072	2048	42.5	0.063

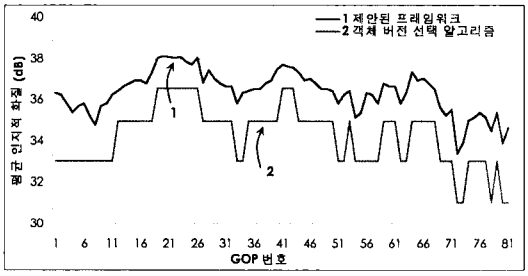


그림 7 평균 인지적 화질 비교

가능한 네트워크 대역폭과 같다. 객체 버전 선택 알고리즘을 이용한 방법도 네트워크 대역폭의 변화에 맞춰 가용 네트워크 대역폭을 활용할 수 있으나 선택 가능한 버전에 한계가 있으므로 선택된 비디오 객체들의 총 비트율의 차이는 그림 6에서와 같이 큰 폭을 보인다.

그림 7은 위의 변화하는 네트워크 대역폭에 따른 두 방법의 평균 인지적 화질 값이다. 1번 선은 트랜스코딩이 가능한 제안된 프래임워크의 결과를 나타내고 2번 선은 객체 버전 선택 알고리즘의 결과를 보인다. 그림 6에서는 제안된 프래임워크에서 사용하는 비트율은 객체 버전 선택 알고리즘의 결과와 비슷하거나 높았지만 그림 7에서 보이는 바와 같이 평균 인지적 화질의 결과는 전체적으로 높음을 알 수 있다. 제안된 프래임워크에서는 비디오 객체의 우선 순위와 PSNR를 고려해 좀 더 좋은 화질을 갖도록 객체의 버전을 트랜스코딩하거나 버전을 선택할 수 있다. 하지만 객체 버전 선택 알고리즘의 경우 선택할 수 있는 버전에 제한이 있으며 우선 순위가 높은 비디오 객체가 우선적으로 선택되어 많은 비트율을 할당받게 되므로 우선 순위가 낮은 비디오 객체의 화질이 고려되기 어렵다.

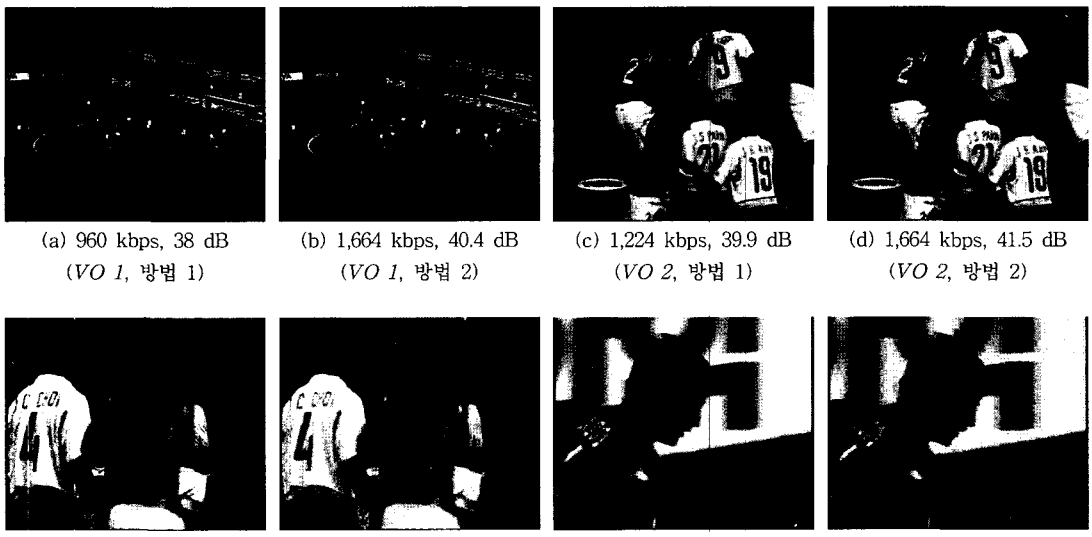
표 2는 위의 실험 조건에서 최대 전송 비트율에 따라 얻어지는 비디오 객체의 화질을 비교 하기 위해 각 방법에 따라 결정되는 각 객체의 비트율과 PSNR 값을 보인다. 최대 전송 비트율은 2~5 Mbps로 주어지며 우선순위를 함께 고려하였다. 표의 음영부분은 제안된 프래임워크 (방법 1)과 객체 버전 선택만을 고려한 알고리즘 (방법 2)를 적용하였을 때 결과의 차이를 보이는 구간을 나타내며 옅은 회색의 셀은 제안된 프래임워크의 결과가 더 높은 비트율을 가질 때이고 짙은 회색의 셀은 객체 버전 선택 알고리즘을 이용한 결과에 더 높은 비트율이 할당되었을 때를 표시한다. 최대 네트워크 대역폭이 2 Mbps의 경우 제안된 프래임워크 (방법 1)이 전반적으로 더 높은 비트율을 갖거나 같은 비트율을 가짐을 알 수 있다. 하지만 3~4 Mbps의 경우, 전체적으로 제안된 프래임워크 (방법 1)의 결과가 좋으나 객체 2의 경우 객체 버전 선택만을 고려한 알고리즘 (방법 2)

표 2 방법 1과 2에 의한 비디오 객체의 적응된 비트율과 PSNR 결과 비교

VOs		2Mbps		3Mbps		4Mbps		5Mbps	
		방법 1	방법 2	방법 1	방법 2	방법 1	방법 2	방법 1	방법 2
-	비트율(kbps)	384	128	384	128	960	128	960	1664
	PSNR (dB)	34.5	30.9	34.5	30.9	38	30.9	38	40.4
2	비트율(kbps)	320	128	704	1664	704	1664	1224	1664
	PSNR (dB)	35	31.5	38	41.5	38	41.5	39.9	41.5
3	비트율(kbps)	128	128	384	128	800	128	832	128
	PSNR (dB)	30.9	30.9	35.5	30.9	38.5	30.9	38.7	30.9
4	비트율(kbps)	272	128	384	128	384	128	832	128
	PSNR (dB)	34.6	32	36.7	32	36.7	32	39.7	32
5	비트율(kbps)	512	512	512	512	512	512	512	512
	PSNR (dB)	30.8	30.8	30.8	30.8	30.8	30.8	30.8	30.8
6	비트율(kbps)	128	128	376	128	384	128	384	128
	PSNR (dB)	30	30	33.6	30	33.2	30	33.2	30
7	비트율(kbps)	128	128	128	128	128	128	128	128
	PSNR (dB)	29.6	29.6	29.6	29.6	29.6	29.6	29.6	29.6
8	비트율(kbps)	128	128	128	128	128	128	128	128
	PSNR (dB)	31.3	31.3	31.3	31.3	31.3	31.3	31.3	31.3

의 비트율이 더 높으며, 5 Mbps에서 비디오 객체 1과 2의 결과가 더 좋음을 알 수 있다. 이는 비디오 객체 1과 2가 높은 우선 순위를 가지므로 다른 객체에 비해 좀 더 많은 비트율을 할당 받게 되지만 비트율과 PSNR의 상관관계가 특정 구간에서 포화되므로 높은 우선 순위를 갖는 비디오 객체에 비트율을 할당하기 보다는 그 다음의 우선 순위를 갖는 비디오 객체에 비트율을 할당하게 된다. 즉, 3Mbps~5Mbps 구간에 제안된 프래임워크에 의한 비디오 객체 2의 비트율은 객체 버전 선택 알고리즘에 의한 비디오 객체 2의 비트율 보다 낮으나 PSNR의 값은 충분히 높음을 알 수 있다.

그림 8은 최대 네트워크 대역폭 5Mbps에서 재구성된 비디오 객체 1, 2, 3, 4의 캡쳐된 이미지를 보인다. 비디오 객체 1, 2번의 경우 제안된 프래임워크 (방법 1)의 화질이 크게 차이 나지 않음을 알 수 있다. 하지만 비디오 객체 3, 4번의 경우 각각 7.8 dB와 7.7dB의 차이로 화질이 차이 남을 알 수 있다.



(a) 960 kbps, 38 dB (*VO 1*, 방법 1) (b) 1,664 kbps, 40.4 dB (*VO 1*, 방법 2) (c) 1,224 kbps, 39.9 dB (*VO 2*, 방법 1) (d) 1,664 kbps, 41.5 dB (*VO 2*, 방법 2)

(e) 832 kbps, 38.7 dB (*VO 3*, 방법 1) (f) 128 kbps, 30.9 dB (*VO 3*, 방법 2) (g) 832 kbps, 39.7 dB (*VO 4*, 방법 1) (h) 128 kbps, 32 dB (*VO 4*, 방법 2)

그림 8 방법 1과 방법 2에 의해 재구성된 프래임의 화질 비교

6. 결 론

인터넷의 광대역화와 함께 IPTV와 같은 비디오 스트리밍 서비스가 가능하게 되었으며 이를 기반으로 다중 앵글 비디오 서비스와 같은 진화된 비디오 스트리밍 서비스도 제안되고 있다. 하지만 다양한 스트리밍 서비스의 제공이 가능해짐에 따라 스트리밍의 양 또한 증가하게 되며 가변적 네트워크 대역폭을 갖는 IP 네트워크 환경 상에서 실시간 QoS를 보장하기 어려워진다. 본 논문에서는 여러 비디오 스트림을 함께 전송함으로써 다양한 뷰를 선택할 수 있도록 하는 다중 앵글 비디오 서비스(multi-angle video service) 시나리오를 기반으로 주어진 네트워크 대역폭과 트랜스코딩 코스트의 두 가지 제한 환경에 따라 다중 비디오 스트림을 전송하는데 있어 최고의 화질을 갖는 비디오 객체의 집합을 결정하는 최적화 적응 프래임워크를 제안하였다. 다중 앵글 비디오 서비스는 여러 대의 카메라로부터 획득된 영상을 바탕으로 사용자에게 다양한 뷰를 제공함으로써 사용자가 원하는 뷰를 선택해서 시청할 수 있도록 한다. 하지만 네트워크 대역폭의 제한이 있는 경우, 모든 비디오 스트림을 고화질의 형태로 보내기 어려우며 사용자의 선호도 및 사용자의 환경에 따라 적응된 형태의 다중 비스트림 선택이 요구된다. 이를 위해 트랜스코딩하는데 걸리는 시간인 트랜스코딩 코스트와 PSNR과 비디오 객체의 우선순위를 고려한 인지적 화질의 인덱스를 정의하고 주어진 네트워크 대역 폭 및 트랜스코딩 코스트를 만족하면서 최고의 인지적 화질의 값을 갖는 다중 비디오 스트림 집합을 결정하도록 하는 최적화 적응 프래임워크를 제안하고 이를 수식화하여 선형 모델을 적용하였다. 주어진 환경 하에서 최적의 화질을 갖는 다중 비디오 스트림 집합을 결정하도록 하기 위해 본 연구에서는 첫째로 여러 객체의 버전 선택 및 트랜스코딩의 문제를 한꺼번에 고려할 수 있도록 하는 적응 프래임워크를 제안하였고, 둘째로 이를 위해 트랜스코딩 코스트 정의 및 PSNR, 비트율, 우선순위, 트랜스코딩 코스트로 구성된 비선형 최적화 모델을 표현하였으며 셋째로 제안된 프래임워크를 분리계획법을 이용하여 선형의 형태로 변환하였다. 또한 제안된 방법은 특정 비디오 코딩 방식이나 트랜스코딩 방법에 국한되지 않으며, 마지막으로 실험을 통해 제안된 최적화 적응 프래임워크를 객체 버전만을 선택하는 기존의 방법과 비교하여 제한된 환경에서 더 좋은 인지적 화질의 증가를 보였다.

참 고 문 헌

[1] ITU-T, "Working document: IPTV service scenarios," ITU-T 2nd FG IPTV, FG IPTV-DOC-0055, Busan, October, 2006.

[2] TV Anytime Forum, "Requirements series: R-1 (Phase 2) on business models," RQ001.v2.1, Sept. 2004.

[3] P. Batra, "Modeling and efficient optimization for object-based scalability and some related problems," *IEEE Transactions on Image Processing*, Vol.9, No.10, pp. 1677-1692, Oct. 2000.

[4] K.-A. Cha, "A scalability for adaptive MPEG-4 contents," Ph. D Thesis, Kyungpook National University, Korea, June, 2003.

[5] A. Vetro, C. Christopoulos, and H. Sun, "Video transcoding architectures and techniques: An overview," *IEEE Signal Processing Magazine*, Vol.20, No.2, pp. 18-29, Mar. 2003.

[6] T. Wiegand, H. Schwarz, A. Joch, F. Kossentini, and G. J. Sullivan, "Rate-constrained coder control and comparison of video coding standards," *IEEE Transactions on Circuit and Systems for Video Technology*, Vol.11, No.3, pp. 1-19, July 2003.

[7] J.-G. Kim, Y. Wang, S.-F. Chang, and H.-M. Kim, "An optimal framework of video adaptation and its application to rate adaptation transcoding," *ETRI Journal*, Vol.27, No.4, pp. 341-354, Aug. 2005.

[8] A. Vetro, and C. W. Chen, "Rate-reduction transcoding design for wireless video streaming," *Proceedings of International Conference on Image Processing*, Vol.1, pp. 29-32, 2002.

[9] P. Yin, M. Wu, and B. Liu, "Video transcoding by reducing spatial resolution," *Proceedings of the International Conference on Image Processing*, Vol.1, pp. 972-975, Sept. 2000.

[10] W. Li, "Overview of fine granularity scalability in MPEG-4 video standard," *IEEE Transactions on Circuit and Systems for Video technology*, Vol.11, No.3, pp. 301-317, March 2001.

[11] K. Li and H. Shen, "Proxy placement problem for coordinated en-route transcoding proxy caching," *International Journal of Computer System Science & Engineering*, Vol.19, No.5, pp. 95-103, Sept. 2004.

[12] R. Han, P. Bhagwat, R. Lamaire, T. Mummert, V. Perret, and J. Rubas, "Dynamic Adaptation in an image transcoding proxy for mobile web browsing," *IEEE Personal Communications Magazine*, Vol.5, No.6, pp. 8-17, Dec. 1998.

[13] A. Raghuveer, N. O. Kang, and D. H. C. Du, "Techniques for efficient streaming of layered video in heterogeneous client environments," *Proceedings of the IEEE Global Telecommunications Conference*, St. Louise, MO, USA, pp. 1-6, Nov. 2005.

[14] H. Lieberman, *Introduction to operations research*, McGraw Hill, 2005.

[15] P. A. Jensen and J. F. Bard, *Operations research models and methods*, Wiley, 2002.

[16] FFmpeg, Available: http://ffmpeg.sourceforge.net.

[17] Lingo 9.0, Available: http://www.lindo.com.

임 정 연

1999년 2월 충남대학교, 학사. 2001년 2월 충남대학교, 석사. 2007년 2월 한국정보통신대학교, 박사. 2007년 6월~현재 SK 텔레콤 Network기술원, 매니저. 관심분야는 영상통화정보서비스, 멀티미디어 시스템, MPEG-4/7/21/A, 비주얼 정보처리

김 문 철

1989년 2월 경북대학교 전자공학과, 공학사. 1992년 12월 University of Florida, Electrical and Computer Engineering, 석사. 1996년 8월 University of Florida, Electrical and Computer Engineering, 박사. 1997년 1월~2001년 2월 한국전자통신연구원 방송미디어연구부, 선임연구원. 2001년 2월~현재 한국정보통신대학교 공학부, 부교수. 2001년 2월~현재 SC29-Korea 전문위원, MPEG 포럼 운영위원. 관심분야는 비디오 코딩, 영상 통신, 비주얼 정보처리, 멀티미디어 시스템