

다중 객체 기반 전송을 위한 최적화 적응 프레임워크

임정연¹, 김문철²,
한국정보통신대학교 (ICU)^{1 2}
{jylim¹,mkim²}@icu.ac.kr

An Optimal Adaptation Framework for Transmission of multiple Visual Objects

Jeongyeon Lim¹, Munchurl Kim²
Information and Communications University (ICU)^{1 2}

요약

인터넷의 성장과 함께 인터넷 상의 비디오 스트리밍은 콘텐츠 전달에 중요한 수단이 되었다. 기존의 비디오 전송 방식은 사용자 터미널에 하나의 압축된 비디오를 스트리밍하는 것이었으나 최근 다중 뷰 비디오 코딩은 다른 시각 각도에 비디오 스트리밍 서비스를 지원하는 새로운 어플리케이션이 되었다. 따라서 단일 비디오는 다른 초점을 갖는 여러 독립된 비디오 스트림으로 구성되며 이는 더 많은 대역폭을 요구한다. 하지만 시간에 따라 네트워크의 대역폭이 변하는 경우, 계획된 비트율에 따라 콘텐츠의 비트량을 조절하여 양질의 콘텐츠 제공을 보장하기 어렵다. 따라서 트랜스코딩은 가변적인 네트워크의 대역폭에 적응하기 위한 필수적인 처리 단계가 되었다. 본 논문에서는 동시에 다중 비디오의 선택 및 트랜스코딩을 함께 고려하는 최적화 방법을 기반으로 한 스트리밍 다중 뷰 비디오의 최적 적응 프레임워크를 제안한다. 최적화된 비디오 스트림의 집합은 네트워크 대역폭, 트랜스코딩 코스트와 같은 제한을 만족하고 최적의 인지적 화질을 갖도록 선택되거나 트랜스코딩될 수 있으며 이는 MCKP 로 모델링될 수 있다. 또한 트랜스코딩시 R-D 커브 상에 최적의 점을 얻기 위해 분리계획법이 적용하며 결과를 비교 분석한다.

Keyword : 콘텐츠 적응, 분리계획법

1. 서론

초고속 네트워크 서비스는 일상 생활에 보편화되고 있으며 멀티미디어 스트리밍은 이러한 환경에 인터넷 킬러 어플리케이션 중 하나가 되었다. 대개의 비디오 스트리밍 어플리케이션은 단일의 비디오 스트리밍을 다룬다. 하지만 더 실제적인 비디오 콘텐츠를 제공하기 위해 다른 각도의 여러 카메라로부터 얻어진 다중의 비디오로 저작되기도 한다. 같은 콘텐츠로부터 사용자 터미널에서 다른 각도로 렌더링될 수 있다. 이러한 어플리케이션의 예로 축구, 농구, 미식 축구와 같은 스포츠 경기를 들 수 있다. 골 장면의 경우 골대를 중심으로 여러 각도로 표현될 수 있다 [1]. 이러한 단일의 비디오 콘텐츠 상에서 다중의 비디오 스트림 서비스를 제공하기 위해, 하나의 사용자 터미널 안에 모든 비디오 스트림의 렌더링이 가능해야 한다. 이러한 다중 비디오 스트림을 가변적인 네트워크를 통해 다양한 멀티미디어 터미널 상에 서비스하기 위해, 각각의 비디오 객체는 다른 비트율을 갖는 하나 이상의 비디오 스트림으로 구성되어야 하고 이들 중 가변적인 네트워크 대역폭에 맞게 적절한 객체의 버전을 선택하거나 트랜스코딩하여 적절한 비디오 스트림의 집합을 전송하게 된다. 본 논문에서는 가변적 네트워크 환경에서 다중 비디오

전송을 위한 최적의 적응 프레임워크를 제안한다. 제안된 적응 프레임워크는 각각의 우선 순위와 다양한 비트율을 갖는 하나 이상의 스케일러블 레벨을 지원하는 비디오의 선택 및 비디오 객체의 트랜스코딩을 함께 고려한다. 트랜스코딩 코스트는 최적의 인지적 화질을 얻도록 허락된 최대 비트율을 갖도록 최고의 화질을 갖는 버전에서 트랜스코딩 시 걸리는 시간을 의미한다. 비디오 객체는 매 GOP 마다 트랜스코딩될 수 있다. 제안된 적응 프레임워크는 두 제한점인 네트워크 대역폭과 트랜스코딩 코스트를 만족하며 인지적 화질을 최대화하는 최적화 문제를 고려한다. 인지적 화질은 선택되거나 트랜스코딩되는 다중 객체의 PSNR 과 우선순위의 곱으로 표현된다. 제안된 최적화 문제는 Multiple Choice Knapsack Problem(MCKP)로 표현되며 객체의 버전을 선택하거나 트랜스코딩하는데 있어 비디오 객체의 R-D 커브에 최적의 값을 갖도록 제안된 모델에 분리계획법을 적용한다.

2. 다중 객체 기반 최적화 적응 프레임워크

MPEG-4 와 같은 콘텐츠는 다양한 객체로 구성될 수 있으며, 각 객체는 다른 비트율을 갖는 다양한 버전을 가질 수 있다. 다중 객체 기반 최적 적응

프레임워크에서 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 트랜스코딩을 부여할 수 있다. 그림 1은 시간에 따라 변하는 네트워크 대역폭에 따라 트랜스코딩이 적용되는 과정을 설명한다.

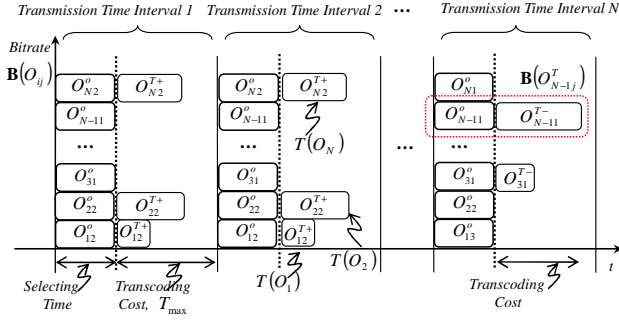


그림 1 스트리밍 될 객체 선택 및 트랜스코딩

그림 1은 스트리밍 될 객체의 선택 및 트랜스코딩된 객체들의 시간 단위 당 객체 집합을 보인다. 각 객체는 비트율에 따른 여러 버전과 해당 PSNR 값을 가지며 한 객체를 트랜스코딩하는데 걸리는 시간인 트랜스코딩 코스트를 주어져 있다고 가정한다. 전송 시간 간격(Transmission Time Interval)은 콘텐츠를 스트리밍할 때 콘텐츠의 한 GOP에 접근하고 전송되기까지의 총 시간을 의미하며, 각 객체의 알맞은 버전을 선택하기 위한 선택 시간(Selecting Time)과 트랜스코딩 과정에 걸리는 트랜스코딩 시간(Transcoding Time)으로 할당할 수 있다. 본 논문에서 트랜스코딩 방법은 각 객체의 최고 화질을 갖는 버전을 GOP 단위로 디코딩하고 목표 비트율로 재압축하는 과정을 의미한다. 각 객체의 트랜스코딩 시간을 각 객체의 트랜스코딩 오버헤드를 측정하기 위한 트랜스코딩 코스트라고 부른다. O_{ij}^o 는 선택된 i 번째 객체의 j 번째 버전을 나타내며 $j=1, 2, \dots, n_i$ 이고 그것의 비트율은 $B(O_{ij}^o)$ 로 표현된다. 트랜스코딩 코스트 상에 O_{ij}^{T+} 는 i 번째 객체의 j 번째 버전에 대한 트랜스코딩으로 목표 비트율이 증가된 객체를 의미하며, O_{ij}^{T-} 는 트랜스코딩으로 감소한 객체를 의미한다. 두 번째 전송 시간 단위에서 첫 번째와 두 번째 그리고 N 번째 객체의 트랜스코딩 코스트는 $T(O_1)$, $T(O_2)$ 와 $T(O_N)$ 으로 표현된다. 첫 번째와 두 번째 전송 시간 구간에서 더 나은 화질을 위해 N 개의 객체들 중 첫 번째 O_1 와 두 번째 객체 O_2 그리고 N 번째 객체 O_N 를 트랜스코딩한다. 반면에 마지막 전송 시간 구간에서는 네트워크 대역폭의 감소로 $N-1$ 번째 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. 주어진 조건에서 적절한 버전을 선택 및 트랜스코딩 하기 위한 모델은 배낭문제(MCKP) 문제로 표현된다. 인지적 화질을 표현하기 위해 PSNR을 이용하여 제한된 최대 허용 비트율과 트랜스코딩 코스트를 이용하여 다음과 같은 목적함수와 제한식으로 표현될 수 있다.

$$\text{Max} \left(\underbrace{\sum_i \sum_j P_{SNR}(O_{ij}^o) \cdot P_{rior}(O_i) \cdot Y(O_{ij}^o)}_{\text{각 객체의 버전이 선택된 경우}} + \underbrace{\sum_i \sum_j P_{SNR}(O_{ij}^T) \cdot P_{rior}(O_i) \cdot Y(O_{ij}^T)}_{\text{트랜스코딩 버전이 선택된 경우}} \right) \quad (1)$$

$$\text{제한식} \quad \sum_i \sum_j B(O_{ij}^o) \cdot Y(O_{ij}^o) + \sum_i \sum_j B(O_{ij}^T) \cdot Y(O_{ij}^T) \leq B_{\max} \quad (2)$$

선택된 버전의 비트율 트랜스코딩된 버전의 비트율

$$T(O_{ij}^T) \cdot Y(O_{ij}^T) \leq TC_i \quad (3)$$

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

주어진 식 (1)에서 최고의 효율을 얻기 위해 각 객체 O_{ij} 의 PSNR 값을 $P_{SNR}(O_{ij})$ 로 하여 효용성(utility)으로 표현하였으며, 우선순위는 $P_{rior}(O_i)$ 로 나타내어 더 높은 순위를 갖는 객체가 선택될 수 있도록 곱으로 표현한다. $Y(O_{ij}^o)$ 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_i 보다 작아야 한다. TC_i 는 콘텐츠를 제공받는 각 디코더의 프레임율에 따라 결정될 수 있다. 제안된 선택 알고리즘은 각 객체의 모든 버전 정보가 주어지고 객체 또는 버전의 선택만 고려한다면 이진 정수 계획법 (binary integer programming)으로 풀 수 있다[2]. 그러나 어떤 객체를 얼마나 트랜스코딩할 것인지 결정하기 어렵다. 또한 트랜스코딩을 얼마나 수행할 지와 관련하여 PSNR과 비트율의 관련식이 요구된다. 하지만 PSNR값과 비트율의 관련식을 구하기 어렵고, PSNR값과 비트율의 커브는 비선형이다. 본 논문에서는 객체의 선택 및 트랜스코딩을 함께 고려하기 위해 PSNR값과 비트율의 비선형 관계를 근사화로 표현한 분리 계획법을 적용한다[3].

3. 분리 계획법의 적용

3-1 분리 계획법 (separable programming) 소개

분리 계획법은 비선형 계획법의 하나로 비선형 모델을 선형의 형태로 표현할 수 있으며 일반적으로 아래와 같이 나타낼 수 있다.

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

제한식:

$$\sum_{j=1}^n g_{ij}(x_j) \leq 0 \quad (i=1, 2, \dots, m) \quad (6)$$

목적 함수에 각 함수 f_j 와 제한식에 각 함수 g_j 에 결정 변수가 각각 표현되기 때문에 이러한 경우 분리 가능하다고 표현한다. 식 (5)의 목적함수는 $f(x) = f_1(x_1) + \dots + f_n(x_n)$ 로 표현될 수 있고, 식 $f_1(x_1), \dots, f_n(x_n)$ 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, 선형계획법으로 풀 수 있도록 근사값을 이용한다

그림 2 와 같이 일반적으로 비선형인 함수 $f(x)$ 를 가정한다. r 개의 세그먼트를 이용하여 불연속 선형 근사값을 형성하기 위해 $r+1$ 개의 불연속 점이 요구된다.

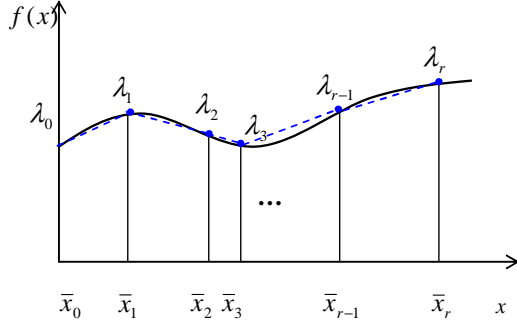


그림 2 비선형 함수와 불연속 선형 근사값

그림 2 에서 실선은 일반적인 비선형 함수 $f(x)$ 를 나타내고 직선의 점선은 비선형 함수의 불연속 선형 근사값 $f^a(x)$ 을 나타낸다. 만약 어떤 두 점 $(\bar{x}_{k-1}, \bar{x}_k)$ 사이에 해 x 가 존재한다면, 다음과 같이 표현할 수 있다.

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

$$\lambda_{k-1} + \lambda_k = 1, \quad 0 \leq \lambda_i \leq 1$$

$$\begin{aligned} f^a(x) &= f(\bar{x}_{k-1}) \cdot \lambda_{k-1} + f(\bar{x}_k) \cdot \lambda_k \\ &= f(\bar{x}_{k-1}) \cdot \lambda_{k-1} + f(\bar{x}_k) \cdot (1 - \lambda_{k-1}) \end{aligned} \quad (8)$$

여기서 음수가 아닌 λ_{k-1}, λ_k 는 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 x 를 식 (8)와 같이 표현한다. 전체의 근사화된 식 $f^a(x)$ 는 다음과 같이 표현될 수 있다.

$$\begin{aligned} f^a(x) &= f(\bar{x}_0) \cdot \lambda_0 + \dots + f(\bar{x}_{r-1}) \cdot \lambda_{r-1} + f(\bar{x}_r) \cdot \lambda_r \\ &= \sum_{i=0}^r f(\bar{x}_i) \cdot \lambda_i \end{aligned} \quad (9)$$

$$x = \lambda_1 \cdot \bar{x}_1 + \dots + \lambda_{r-1} \cdot \bar{x}_{r-1} + \lambda_r \cdot \bar{x}_r = \sum_{i=0}^r \lambda_i \cdot \bar{x}_i \quad (10)$$

$$\lambda_1 + \lambda_2 + \dots + \lambda_{r-1} + \lambda_r = \sum_{i=0}^r \lambda_i = 1 \quad (11)$$

비트율과 PSNR 의 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 어둡 값을 이용하므로 얻게 되는 최대 목적함수 값은 커브 안쪽 혹은 커브 상에 존재하게 된다. 비트율과 PSNR 의 관계 식은 오목하게 나타나므로 연속된 불연속 점 간에 값이 존재하는 인접 조건(adjacency condition)을 만족한다[3].

3-2 분리 계획법 적용

객체는 다중의 버전을 갖거나, 이 버전들 가운데 현재의 조건에 적합한 버전이 존재하지 않는다면

트랜스코딩될 수 있다. 식 (1)에 PSNR 과 식 (2)에 비트율에 대한 식은 모든 가능한 PSNR 값과 비트율을 표현할 수 있다. 만약 $Y(O_i)$ 가 객체 선택의 정도를 나타낸다면, 위의 식 (1)와 (2)은 아래와 같이 PSNR 의 함수 $f_{PSNR,i}$ 와 비트율의 함수 $g_{Bitrate,i}$ 로 정리될 수 있다.

$$\text{Max} \left(\sum_i P_{rior}(O_i) \cdot \left(\sum_j P_{SNR}(O_{ij}^o) \cdot Y(O_{ij}^o) + P_{SNR}(O_i^T) \cdot Y(O_i^T) \right) \right) \quad (11)$$

$$= \text{Max} \left(\sum_i P_{rior}(O_i) \cdot f_{PSNR}(Y(O_i)) \right)$$

$$\begin{aligned} &\sum_i \left(\sum_j B(O_{ij}^o) \cdot Y(O_{ij}^o) + B(O_i^T) \cdot Y(O_i^T) \right) \\ &= \sum_i g_{Bitrate}(Y(O_i)) \leq B_{\max} \end{aligned} \quad (12)$$

본 논문에서 객체 선택의 정도 $Y(O_i)$ 는 $(0,1]$ 의 값을 갖는다고 가정한다. 0 은 객체의 어떤 버전도 선택되지 않았음을 의미하고 1 은 최고의 화질을 갖는 버전이 선택되었음을 의미한다. 0 과 1 사이의 값은 객체가 트랜스코딩 되었거나 세부의 버전이 선택되었음을 의미한다. PSNR 의 함수와 식 (18)에 목적함수는 모든 객체에 분리 가능한 함수의 합으로 표현될 수 있다.

$$\sum_i P_{rior}(O_i) \cdot f_{PSNR}(Y(O_i)) \quad (13)$$

$$= P_{rior}(O_1) \cdot f_{PSNR}(Y(O_1)) + \dots + P_{rior}(O_n) \cdot f_{PSNR}(Y(O_n))$$

PSNR 의 함수 $f_{PSNR}(Y(O_1)), \dots, f_{PSNR}(Y(O_n))$ 는 비트율에 따라 오목(concave)하게 나타나며 이는 비선형으로 보여진다. 또한 선택의 정도에 따라 각 객체의 비트율도 함수로 표현되며 각 객체를 가리키는 인덱스 i 에 따라 분리가 가능 해진다. 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_{PSNR}^a 은 불연속의 값에 가중치를 적용함으로써 표현될 수 있다. PSNR 와 비트율의 근사함수는 다음과 같이 표현될 수 있다.

$$f_{PSNR}^a(Y(O_i)) = P_{SNR}(O_{i0}) \cdot \lambda_{i0} + \dots + P_{SNR}(O_{in_i}) \cdot \lambda_{in_i} \quad (14)$$

$$= \sum_j^{n_i} P_{SNR}(O_{ij}) \cdot \lambda_{ij}$$

$$g_{Bitrate}^a(Y(O_i)) = B(O_{i0}) \cdot \lambda_{i0} + \dots + B(O_{in_i}) \cdot \lambda_{in_i} \quad (15)$$

$$= \sum_i^{n_i} B(O_{ij}) \cdot \lambda_{ij}$$

변수 λ_{ij} 는 가중치를 나타내는 nonnegative 값이며, $\lambda_{i0} + \lambda_{i1} + \dots + \lambda_{in_i} = 1$ 이다. 버전의 수가 많을수록 원래의 곡선에 가깝게 근접한다. 위의 근사화된 식 (14), (15)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. 트랜스코딩 코스트도 존재하는 불연속점으로 어렵이 가능하다.

하지만 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 0 이 된다. 예를 들어, 한 객체는 1,024 kbps 와 512 kbps 로 각각 인코딩된 두 버전을 갖고 있다고 하자. 1,024 kbps 의 버전은 최고의 화질을 가지고 이를 원 버전이라고 부른다. 만약 1,024 kbps 로 인코딩된 버전이 선택된다면, 객체 선택의 정도인 $Y(O_i)$ 는 1 이다. 만약 512 kbps 로 인코딩된 버전이 선택된다면 객체 선택의 정도는 비트율과 비례한다고 가정했으므로 객체 선택의 정도인 $Y(O_i)$ 는 0.5 가 된다. 하지만 512 kbps 로 인코딩된 버전을 의미하고 객체 선택의 정도가 0.5 인 버전이 이미 존재한다면, 객체를 트랜스코딩할 필요는 없다. 하지만 현재의 네트워크 상태가 256 kbps 의 버전을 요구한다면 객체 선택의 정도인 $Y(O_i)$ 는 0.25 가 되고 현재 가지고 있는 버전 중에 맞는 것이 존재하지 않으므로 트랜스코딩이 필요하다. 요구되는 비트율이 512 kbps 인 경우, 트랜스코딩할 필요가 없으므로 0 이 된다. 즉, 이미 존재하는 버전의 $Y(O_{ij})$ 에 트랜스코딩 코스트 $T(O_{ij})$ 는 트랜스코딩 코스트의 커브를 근사화할 때만 쓰인다.

또한 트랜스코딩 코스트는 각 객체의 최대 트랜스코딩 코스트를 초과하지 않는 점만을 만족시키면 되므로 제한된 트랜스코딩 코스트를 초과하는 객체에 대해서는 존재하는 버전에 대해 이진 값만 갖도록 미리 제한식을 추가할 수 있다. 즉, 트랜스코딩이 불가능한 객체에 대해서는 존재하는 버전만을 가지고 이진정수계획법을 적용한다. 따라서 제안된 적응 프레임워크는 다음과 같이 정리될 수 있다.

$$\text{Max} \left(\sum_i P_{rior}(O_i) \cdot f_{PSNR}^a(Y(O_i)) \right) \quad (16)$$

$$\sum_i g_{Bitrate}^a(Y(O_i)) \leq B_{\max} \quad (17)$$

$$\lambda_{ij} = \begin{cases} 0 \text{ or } 1, & \text{if } Y(O_{ij}) \text{ is the existing version} \\ [0..1], & \text{otherwise} \end{cases}$$

트랜스코딩 코스트의 제한 조건인 각 객체의 최대 트랜스코딩 코스트 TC_i^{\max} 는 디코더에서 요구되는 각 객체의 프레임율 보다 작으며 각 객체의 트랜스코딩 코스트는 압축 방법, 해상도, 프레임율 등에 따라 달라질 수 있다. 트랜스코딩 코스트가 큰 객체를 미리 제한함으로써 계산량을 줄일 수 있다.

4. 실험

분리 계획법에 의한 풀이를 확인하기 위해 다음과 같은 조건의 비트율에 따라 PSNR 값과 트랜스코딩 코스트를 갖고 모두 트랜스코딩이 가능한 4 가지 객체를 이용한다. 다음의 객체들은 352x288 의 해상도를 가지며, 각각의 객체는 FFmpeg[4]과 x264[5]를 이용하여 MPEG-1 video, MPEG-4 video SP(Simple Profile), H.263 video, AVC|H.264 BL(baseline)으로 각각 인코딩 되었다. 트랜스코딩 수행 시 디코딩하고 재인코딩하며 GOP (Group of Picture) 단위로 각각에 맞게 트랜스코딩하여 전송한다. 모든 객체가 트랜스코딩이 가능하다고 가정한다. 표 1 은 다양한 버전의 비트율과 해당 PSNR 값을 보인다.

표 1 비트율에 따른 각 객체별 PSNR

	객체 1 $P_{rior}(O_1)=0.1$	객체 2 $P_{rior}(O_2)=0.2$	객체 3 $P_{rior}(O_3)=0.3$	객체 4 $P_{rior}(O_4)=0.4$
256	38.5	39.25	38.33	38.47
512	41.4	42.91	41.3	41.47
1024	45.09	44.78	44.28	44.23
2048				46.82

표 1 의 각 객체에 PSNR 값이 너무 낮은 값에 대해서는 고려하지 않았으며 각 객체에 분리계획법을 적용하면 다음과 같다.

객체 1 의 경우,

$$f_{PSNR}^a(Y(O_1)) = 38.5 \cdot \lambda_{11} + 41.4 \cdot \lambda_{12} + 45.09 \cdot \lambda_{13}$$

$$g_{Bitrate}^a(Y(O_1)) = 256 \cdot \lambda_{11} + 512 \cdot \lambda_{12} + 1024 \cdot \lambda_{13}$$

$$\lambda_{11} + \lambda_{12} + \lambda_{13} = 1$$

객체 2 의 경우,

$$f_{PSNR}^a(Y(O_2)) = 39.25 \cdot \lambda_{21} + 42.91 \cdot \lambda_{22} + 44.78 \cdot \lambda_{23}$$

$$g_{Bitrate}^a(Y(O_2)) = 256 \cdot \lambda_{21} + 512 \cdot \lambda_{22} + 1024 \cdot \lambda_{23}$$

$$\lambda_{21} + \lambda_{22} + \lambda_{23} = 1$$

객체 3 의 경우,

$$f_{PSNR}^a(Y(O_3)) = 38.33 \cdot \lambda_{31} + 41.3 \cdot \lambda_{32} + 44.28 \cdot \lambda_{33}$$

$$g_{Bitrate}^a(Y(O_3)) = 256 \cdot \lambda_{31} + 512 \cdot \lambda_{32} + 1024 \cdot \lambda_{33}$$

$$\lambda_{31} + \lambda_{32} + \lambda_{33} = 1$$

객체 4 의 경우,

$$f_{PSNR}^a(Y(O_4)) = 38.47 \cdot \lambda_{41} + 41.47 \cdot \lambda_{42} + 44.23 \cdot \lambda_{43} + 44.82 \cdot \lambda_{44}$$

$$g_{Bitrate}^a(Y(O_4)) = 256 \cdot \lambda_{41} + 512 \cdot \lambda_{42} + 1024 \cdot \lambda_{43} + 2048 \cdot \lambda_{44}$$

$$\lambda_{41} + \lambda_{42} + \lambda_{43} + \lambda_{44} = 1$$

각 객체의 버전 수에 따라 변수, λ 의 수가 증가된다. 분리계획법에 의해 표현된 각각의 PSNR 값과 비트율의 식은 simplex method [2]를 이용하여 풀 수 있다. 최대의 비트율 B_{\max} 를 4Mbps 로 결정하면 $\lambda_{12} = 0.1875$, $\lambda_{13} = 0.8125$, $\lambda_{23} = 1$, $\lambda_{33} = 1$, $\lambda_{43} = 1$ 와 같은 결과가 산출되며 나머지 λ_{ij} 는 0 이 된다. 여기서 $\lambda_{12} = 0.1875$, $\lambda_{13} = 0.8125$ 는 첫 번째 객체가 두 번째 버전과 3 번째 버전의 사이에 해당되는 새로운 버전으로 트랜스코딩 됐음을 의미하며 두 번째, 세 번째, 네 번째 객체 모두 3 번째 버전이 선택되었음을 알 수 있다. 이 값을 각각의 근사함에 대입하면 다음과 같은 결정된 PSNR 값과 비트율을 알 수 있다.

$$f_{PSNR}^a(Y(O_1)) = 41.4 \cdot 0.1875 + 45.09 \cdot 0.8125 = 44.4 \text{ dB}$$

$$g_{Bitrate}^a(Y(O_1)) = 512 \cdot 0.1875 + 1024 \cdot 0.8125 = 928 \text{ kbps}$$

$$f_{PSNR}^a(Y(O_2)) = 44.78 \text{ dB}, \quad g_{Bitrate}^a(Y(O_2)) = 1,024 \text{ kbps}$$

$$f_{PSNR}^a(Y(O_3)) = 44.28 \text{ dB}, \quad g_{Bitrate}^a(Y(O_3)) = 1,024 \text{ kbps}$$

$$f_{PSNR}^a(Y(O_4)) = 44.23 \text{ dB}, \quad g_{Bitrate}^a(Y(O_4)) = 1,024 \text{ kbps}$$

다음은 실시간으로 변하는 네트워크 대역폭의 변화에 따른 선택되는 객체의 총 비트율 변화와 우선순위가 부여된 객체의 최대 목적함수 값을 보인다. 제공된 객체의 구성은 아래와 같다. 실험해서 각각의 객체는 30 frame/s 로 인코딩되어 있고 한 GOP 는 12 프레임으로 되어 있다. 따라서 각 객체의 버전에 트랜스코딩 코스트와 선택하는데 걸리는 시간의 합이 0.4 이상 되는 경우 트랜스코딩하는데 제한을 둔다. 즉, 객체 6 의 경우는 존재하는 버전으로 선택한다

가능하도록 제한하였다.

표 2 객체의 PSNR 및 트랜스코딩 코스트

	객체 1		객체 2		객체 3		객체 4	
	$P_{rior}(Y(Q_1))=0.1$	$P_{rior}(Y(Q_2))=0.1$	$P_{rior}(Y(Q_2))=0.1$	$P_{rior}(Y(Q_2))=0.1$	$P_{rior}(Y(Q_3))=0.2$	$P_{rior}(Y(Q_3))=0.2$	$P_{rior}(Y(Q_4))=0.2$	$P_{rior}(Y(Q_4))=0.2$
	PSNR	TC(s)	PSNR	TC(s)	PSNR	TC(s)	PSNR	TC(s)
64	33.81	0.077	35.56	0.041	34.62	0.081		
128	35.89	0.077	37.95	0.041	36.87	0.081	39.4	0.042
256	38.5	0.08	42.26	0.041	39.25	0.081	43.23	0.042
512	41.4	0.085	44.32	0.041	42.91	0.088	43.69	0.042
1024	45.09	0.091			44.78	0.088		
	객체 5		객체 6		객체 7		객체 8	
	$P_{rior}(Y(Q_5))=0.3$	$P_{rior}(Y(Q_6))=0.3$	$P_{rior}(Y(Q_7))=0.4$	$P_{rior}(Y(Q_8))=0.4$	$P_{rior}(Y(Q_7))=0.4$	$P_{rior}(Y(Q_8))=0.4$	$P_{rior}(Y(Q_8))=0.4$	$P_{rior}(Y(Q_8))=0.4$
	PSNR	TC(s)	PSNR	TC(s)	PSNR	TC(s)	PSNR	TC(s)
64	33.09	0.029						
128	35.75	0.031	23.72	0.197				
256	38.33	0.031	26.57	0.266				
512	41.3	0.037	29.92	0.352	30.59	0.098	30.24	0.100
1024			33.17	0.434	33.82	0.107	33.12	0.105
2048			36.46	0.508	37.48	0.112	36.95	0.112
3072			38.55	0.546			39.94	0.117
4096			40.2	0.574				

표 2 에서와 같이 8 개의 객체는 352x288 과 176x144 와 같은 다양한 크기 및 다양한 인코딩 방법(MPEG-1 video, MPEG-4 video SP, H.263 video, AVC(H.264) 그리고 느린 변화를 갖는 비디오와 빠른 움직임을 갖는 비디오로 구성되어 있다. 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 PSNR 과 트랜스코딩시 요구되는 트랜스코딩 코스트를 의미한다. 다양한 객체의 버전을 선택하는데 걸리는 시간은 모델의 구조와 밀집도 등을 표현하는 변수의 개수에 비례한다. 본 실험에서는 Lingo 9.0 [6]를 이용하여 변수가 300 개인 경우 0.015 초의 값이 계산되었다. 표 2 에 변수는 총 35 개이며 0.015 초로 계산하였다. 그림 3 은 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.

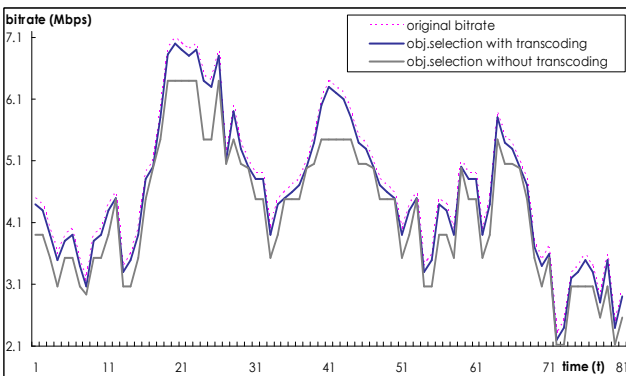


그림 3 시간에 따라 변하는 비트율의 변화

그림에서 분홍색 점선은 시간에 따라 변하는 네트워크 대역폭을 나타내며 파란색선은 분리계획법의 결과이고, 회색선은 이진정수 계획법의 결과이다. 트랜스코딩을 수행하는 경우 현재 주어진 대역폭에 맞도록 트랜스코딩할 수 있으므로 가능한 대역폭과 분리계획법을 적용한 비트율의 결과는 같다. 비트율이 작은 경우 두 알고리즘의 결과는 큰 차이를 보이지

않으나 비트율이 커짐에 따라 선택가능한 버전에 한계가 있으므로 선택된 총 비트율의 차이는 그림에서와 같이 큰 폭을 보인다. 또한 비트율이 작아도 가능한 대역폭을 채우는데 있어서 충분한 버전이 존재하지 않으므로 두 방법 간에 비트율의 차이를 볼 수 있다. 그림 4 는 위의 변화하는 대역폭에 따라 구해지는 각 방법의 평균 최대 목적함수 값이다. 파란색 선은 표 2 에서 제공된 모든 버전으로 분리 계획법을 이용하여 객체를 선택할 경우 계산되는 최대 목적함수 값이고 분홍색 선은 표 2 에서 회색으로 표기된 두 가지 버전으로 분리 계획법을 이용하여 객체를 선택할 경우 계산되는 최대 목적 함수 값이다. 두 실험의 경우 실제 서버가 가지고 있는 버전은 회색으로 표기된 두 버전이지만 트랜스코딩이 가능하므로 가지고 있는 정보만으로 비트율과 PSNR 의 관계를 더 잘 표현할 수 있다. 따라서 더 나은 값을 가지게 된다. 그림의 회색 선은 트랜스코딩이 제공되지 않은 결과로서 이진 정수 계획법(BIP)의 결과와 같다.

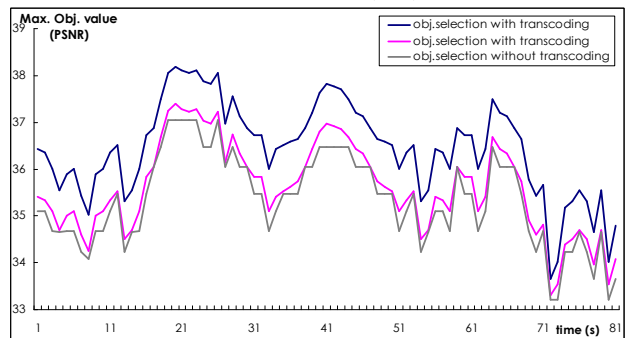


그림 4 시간에 따른 평균 목적 함수 값의 변화

그림 4 에서 분리계획법과 이진정수계획법에서 요구되는 비트율은 몇 구간을 제외하고 큰 차이를 보이지 않는다. 하지만 비슷한 양을 전송함과 달리 화질을 나타내는 최대 목적 함수 값의 결과는 그림 5, 6에서와 같이 큰 차이를 보임을 알 수 있다.

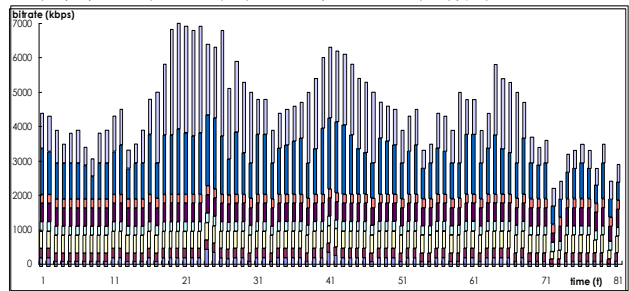


그림 5 분리계획법으로 선택된 객체 집합

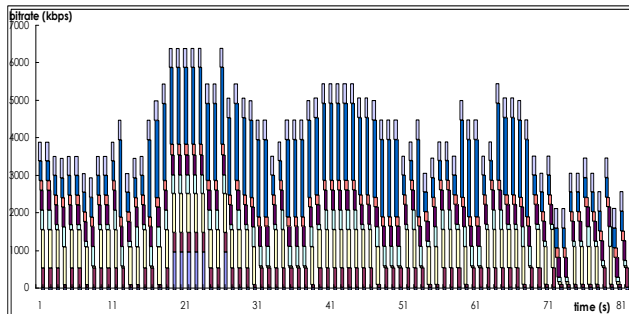


그림 6 이진정수계획법으로 선택된 객체 집합

그림 5, 6 은 각 주어진 비트율에 따라 분리계획법과 이진정수계획법을 적용할 때의 선택된 객체의 집합을 보인다. 하나의 막대에는 여덟 객체의 비트율로 구성되어 있으며 아래쪽이 객체 1 이고 가장 윗쪽의

객체 비트율이 객체 8에 해당된다. 그림 5의 경우 주로 높은 우선 순위를 갖는 객체에 더 많은 비트율이 할당된 것을 알 수 있다. 반면에 그림 6의 경우 남은 비트율 중 높은 우선 순위를 갖는 객체 4과 7이 할당되어 다른 객체에 비해 많은 비트 할당을 갖는 것을 알 수 있다. 트랜스코딩이 허용되는 그림 5에서는 우선 순위가 높은 객체의 비트율이 그림 6에 비해 더 많이 포함되어 있으며, 그림 7의 경우 포함시킬 수 있는 객체의 비트율에 제한이 있으므로 허용된 대역폭에 충분히 들어가는 객체 중 우선 순위가 높은 객체가 선택되어 그림 4에서와 같이 전체의 화질은 분리계획법을 적용한 결과에 비해 떨어지게 된다. 즉, 트랜스코딩을 허용하지 않는 경우 우선순위가 존재해도 객체의 선택에 큰 영향을 미치지 않는다.

다음은 위의 실험 조건에서 최대 전송 비트율에 따라 얻어지는 비디오 객체의 화질을 보인다. 표 2의 실험조건에 따라 트랜스코딩을 허용하는 분리계획법을 이용한 방법과 각 객체에 존재하는 버전만을 이용하여 정수 계획법으로 산출된 결과를 비교한다. 최대 전송 비트율은 2, 3, 4, 5, 6 Mbps로 주어지며 우선순위를 함께 고려하였다. 표 3은 최대 전송 비트율에 따라 각 방법에 결정되는 각 객체의 비트율과 PSNR 값을 보인다. 표의 음영부분은 분리계획법과 정수계획법을 적용하였을 때 결과의 차이를 보이는 구간을 나타내며 노란색의 셀은 분리계획법의 결과가 더 높은 비트율을 가질 때이고 회색의 셀은 이진정수계획법의 결과에 더 높은 비트가 할당되었을 때를 표시한다.

표 3 분리계획법과 정수계획법을 이용한 각 객체의 비트율과 PSNR

	2Mbps		3Mbps		4Mbps		5Mbps		6Mbps	
	SP	BIP	SP	BIP	SP	BIP	SP	BIP	SP	BIP
1	64	64	128	64	128	64	136	64	256	64
	33.8	33.8	35.9	33.8	35.9	33.8	36.1	33.8	38.5	33.8
2	64	64	256	512	256	512	256	64	256	512
	35.6	35.6	42.3	44.3	42.3	44.3	42.3	35.6	42.3	44.3
3	128	64	312	64	288	1024	512	1024	512	1024
	36.9	34.6	40.1	34.6	39.7	44.8	42.9	44.8	42.9	44.8
4	256	128	256	512	256	512	256	512	256	512
	43.2	39.4	43.2	43.7	43.2	43.7	43.2	43.7	43.2	43.7
5	208	64	512	512	512	512	512	512	512	512
	37.4	33.1	41.3	41.3	41.3	41.3	41.3	41.3	41.3	41.3
6	256	256	512	256	512	256	256	256	256	256
	26.6	26.6	29.9	26.6	29.9	26.6	26.6	26.6	26.6	26.6
7	512	512	512	512	1024	512	1024	2048	1904	2048
	30.6	30.6	30.6	30.6	30.6	30.6	33.8	37.5	37.0	37.5
8	512	512	512	512	1024	512	2048	512	2048	512
	30.2	30.2	30.2	30.2	33.1	30.2	37.0	30.2	37.0	30.2

표 3에서 보이는 것과 같이 각 최대 전송 비트율이 주어졌을 때의 결과는 우선순위에 영향을 받았음을 알 수 있다. 객체에 걸리는 트랜스코딩 코스트는 그림 7과 같다. 객체 6의 경우 트랜스코딩 코스트가 크므로 트랜스코딩될 객체의 선택에서는 제외되며 네트워크 대역폭의 증가함에 따라 우선 순위가 높은 객체 5, 7, 8의 트랜스코딩 선택의 변화가 큼을 알 수 있다. 반대로 객체 1, 2, 3, 4는 네트워크 대역폭의 변화에 크게 영향을 받지 않는다.

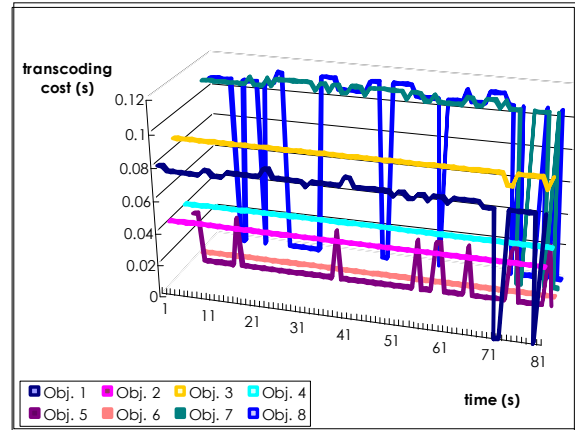


그림 7 요구되는 각 객체의 트랜스코딩 코스트

5. 결론

본 논문에서는 여러 객체를 지원하는 MPEG-4 콘텐츠에서 각 객체에 우선 순위와 요구되는 비트율, 트랜스코딩 코스트를 고려하여 제한된 대역폭 및 트랜스코딩 코스트 상에 최고의 인지적 화질을 갖는 최적화 적응 프레임워크에 대해 제안하였다. 이를 위해 트랜스코딩 코스트를 정의하고 각 객체에서 제공될 수 있는 버전 및 트랜스코딩 시 요구되는 비트율 및 PSNR을 가지고 배낭문제(MCKP)를 이용한 모델을 제안하였으며, 객체의 버전 선택 및 트랜스코딩을 함께 고려하기 위해 분리 계획법을 적용하여 각 객체의 R-D 커브를 선형적으로 근사화하였다. 마지막으로 이진정수계획법의 결과와 비교, 분석하였다. 실험을 통해 다음과 같은 결론을 내린다. 각 객체의 비트율 및 PSNR은 오목한(concave)한 관계를 가지며 이는 분리계획법을 이용하여 표현하기에 적절하다. 실험을 통해 첫째, 비트율 및 PSNR 간의 관계 상에 존재하는 불연속 점이 더 많이 존재함에 따라 두 값 간의 관계는 더욱 정확해지며 더 높은 화질의 객체를 선택할 수 있음을 알 수 있다. 둘째, 여러 버전이 존재하는 경우 현재 가능한 대역폭에 맞춰 보내도록 객체를 선택하는 방법은 현재 대역폭에 비슷하게 맞춰 보낼 수 있으나 트랜스코딩을 허용하는 경우 최대 목적 함수 값은 큰 차이를 보이게 된다. 마지막으로 우선 순위가 높을수록 네트워크 대역폭에 민감하며, 객체를 표현하는데 있어서 더 많은 버전 정보를 갖는 분리계획법이, 버전을 표현하는데 있어서 제한을 갖는 이진 정수 계획법에 비해 객체의 우선순위 혹은 중요도 적용이 수월하다.

6. 참조

- [1] A.Smolcic, and P. Kauff, "Interactive 3D video representation and coding technology," Proceeding of the IEEE Special Issue on Advanced in Video Coding and Delivery, vol. 93, no. 1, Jan. 2005.
- [2] H. Lieberman, Introduction to operations research, McGraw Hill, 2005.
- [3] Paul A. Jensen and Jonathan F. Bard, Operations research models and methods, Willey, 2002.
- [4] FFmpeg, <http://ffmpeg.sourceforge.net>
- [5] x264, <http://developers.videolan.org/x264.html>
- [6] Lingo 9.0, <http://www.lindo.com>.