

■ 2004년 정보과학회 논문경진대회 수상작

# 동영상 스트림 크기 및 품질 예측에 기반한 동적 동영상 적응변환 방법

(A Dynamic Video Adaptation Scheme based on Size and Quality Predictions)

김 증 향 <sup>†</sup>      낭 증 호 <sup>\*\*</sup>  
(Jonghang Kim)      (Jongho Nang)

**요 약** 본 논문에서는 반복적인 인코딩/디코딩 작업 없이 모바일 단말기나 현재 네트워크 상황에 적합한 동영상 스트림을 생성하는 새로운 동적 동영상 적응변환 방법을 제안한다. 제안한 방법에서는 MPEG-1/-2/-4와 같은 비디오 코덱의 특성을 부호화 된 동영상 스트림 크기와 품질에 초점을 맞추어 미리 분석하고, 이를 코덱 의존적인 특성 테이블로 프록시에 저장한다. 이런 특성 테이블 내용과 단말기가 요청한 동영상에 대한 최고 품질 스트림의 크기 및 품질 정보를 이용하여 요청한 모바일 단말기에 적합한 동영상 스트림의 크기 및 품질을 동적으로 예측할 수 있다. 제안한 방법에서는 이런 예측을 바탕으로 동영상의 최대 품질을 유지하며 모바일 단말기의 공간 제약을 만족하는 적응화 된 동영상 스트림 버전을 반복적인 인코딩/디코딩 작업 없이 생성한다. 실험 결과 제안한 방법은 5% 미만의 오차율로 매우 빠르게 동적 동영상 적응변환을 수행함을 알 수 있었다. 제안한 방법은 다양한 비디오 코덱으로 부호화 된 인터넷상의 동영상을 빠르게 변환하는 모바일 단말기를 위한 프록시 서버에 사용될 수 있을 것이다.

**키워드** : 동영상 변환, 무선 멀티미디어, 동영상 화질 예측, 이동 단말기용 동영상

**Abstract** This paper proposes a new dynamic video adaptation scheme that could generate an adapted video stream customized to the requesting mobile device and current network status without repeated decode-encode cycles. In the proposed adaptation scheme, the characteristics of the video codec such as MPEG-1/-2/-4 are analyzed in advance focused on the relationships between the size and quality of the encoded video stream, and they are stored in the proxy as a codec-dependent characteristic table. When a mobile device requests a video stream, it is dynamically decoded-encoded in the proxy with the highest quality to extract the contents-dependent attributes of the requested video stream. By comparing these attributes with codec-dependent characteristic table, the size and quality of the requested video stream when being adapted to the target mobile device could be predicted. With this prediction, a version of adapted video stream, that meets the size constraints of mobile device while keeping the quality of encoded video stream as high as possible, could be selected without repeated decode-encode cycles. Experimental results show that the errors in our proposed scheme are less than 5% and produce an appropriate adapted video stream very quickly. It could be used to build a proxy server for mobile devices that could quickly transcode the video streams widely spread in Internet which are encoded with various video codecs.

**Key words** : Video Adaptation, Wireless Multimedia, Prediction of Video Quality, Video for Mobile Devices

## 1. 서론

이동통신망 시스템은 아날로그 방식의 제1세대와 디지털 방식의 제2세대를 거쳐 전자 상거래와 멀티미디어 서비스를 위한 제3세대에 이르렀다. 이로 인해 지원하는 네트워크의 대역폭은 2.5세대의 144Kbps에 비해 2Mbps로 증가하였으며[1], 국내에서도 제3세대 방식 중 하나

· 본 연구는 한국과학재단 목적기초연구(R01-2002-000-00141-0) 지원으로 수행되었음

† 비회원 : LG전자 홈넷사업팀  
kayin@mlneptune.sogang.ac.kr

\*\* 종신회원 : 서강대학교 컴퓨터학과 교수  
jhnang@ccs.sogang.ac.kr

논문접수 : 2004년 5월 12일

심사완료 : 2004년 10월 14일

인 CDMA2000 1x EV-DO가 상용화 서비스를 시작함에 따라 모바일 환경에서 동영상 콘텐츠에 대한 서비스가 가능해졌다. 하지만 동영상 재생에 있어 아직 모바일 환경은 데스크톱 환경에 비해서는 해상도나 처리 속도 면에서 한계가 있으므로, 동영상 콘텐츠는 모바일 클라이언트에서 재생될 수 있도록 변형 되어야만 한다. 데스크톱 환경에서 자유로이 웹 상의 동영상을 제공 받듯이 모바일 환경에서도 이러한 서비스를 받기 위해서는 프록시 상에서의 동적 동영상 적응변환이 필요하다. 프록시(proxy) 상의 동적 동영상 적응변환은 변화하는 클라이언트 및 네트워크 환경까지 고려해 콘텐츠를 변형시켜준다는 장점이 있지만, 기존의 방법은 서버에서 많은 시간을 들여 콘텐츠를 분석한 뒤 분석한 정보를 프록시에 제공해야 하는 서버 의존적인 제약성이 존재한다. 서버에서 콘텐츠 분석 정보를 제공하지 않을 경우 프록시에서 반복적인 복호화, 부호화 작업을 거쳐 콘텐츠를 분석해야 하는데 이 경우 재생 시간이 지연되는 문제점이 발생한다.

본 논문에서는 클라이언트에서 지원하는 코덱의 특성을 미리 분석하여, 단 한번의 복호화, 부호화 작업만으로 클라이언트 환경에 적합한 동영상 스트림 형태를 찾는 방법을 제안한다. 제안한 방법에서는 우선 코덱의 특성을 파악하기 위해 코덱 별로 적응변환 기법에 의한 스트림 크기 및 품질의 변화율을 미리 테이블로 만들어 저장한다. 클라이언트가 콘텐츠를 요구하면 한번의 복호화 및 부호화 작업만으로 해당 콘텐츠의 공간 복잡도와 시간 복잡도가 반영된 값들을 동적으로 추출한 뒤에 앞서 만들어둔 테이블과 조합해 콘텐츠를 직접 만들지 않고 스트림 크기와 품질을 예측한다. 또한 다양한 형태의 콘텐츠 버전 중에서 요청한 클라이언트의 남은 저장 공간, 일정 수준 이상의 품질, 압축률의 효율성, 사용자 선호도 등을 고려하여 가장 적합한 버전을 결정하는 선택 알고리즘을 제안한다. 기존의 방법과 제안한 방법을 비교 실험한 결과, 스트림 크기와 품질의 예측 오차율은 낮으면서, 적응변환 시간을 크게 줄일 수 있었다. 또한 본 논문에서 제안한 방법을 이용하면 콘텐츠를 공급하는 웹 서버에 구애 받지 않으므로, 모바일 환경에서 웹 상의 동영상 콘텐츠를 최고의 품질로 자유로이 다운로드 받아보는 것이 가능하다.

본 논문은 다음과 같이 구성된다. 2장에서는 적응변환을 제약하는 요소들을 살펴보고, 기존 연구들의 문제점에 대해 분석한다. 3장에서는 스트림 크기 및 품질의 예측 방법을 제안하고 4장에서는 제안 방법을 이용한 시스템의 구조에 대해 살펴본다. 5장에서는 기존 방법과 제안한 방법을 비교 실험을 통해 평가를 내리고, 6장에서 결론을 맺도록 한다.

## 2. 연구 배경

적응변환은 적응변환 기법(Adaptation techniques)을 사용해서 콘텐츠를 자원상황에 맞도록 변환하는 작업이다. 2.1절에서는 적응변환을 제약하는 요소들에 대해 살펴 보고, 2.2절에서는 적응변환을 수행 시기 및 장소에 따라 분류하고, 동영상 콘텐츠들을 적응변환 함에 있어 관련된 기존 연구들과 각각의 문제점에 대해 분석하도록 한다.

### 2.1 적응변환을 제약하는 요소들

일반적으로 모바일 단말기를 위한 적응 변환에 영향을 미치는 요소 및 고려 사항들을 분석하면 다음과 같다.

- 디바이스 프로파일 : 디바이스 프로파일은 클라이언트의 재생 환경에 대한 총체적인 정보로서 화면 크기, 색상 수, 재생 가능한 비디오 및 오디오의 디코더 종류, 메모리 크기 및 저장 가능한 공간 등을 나타낸다. 디바이스 프로파일은 단말기가 시스템에 최초로 접속할 때 User-Agent 변수의 값을 분석하여 알아낸다.
- 네트워크 : 네트워크 상황에 관련된 정보는 네트워크의 종류와 현재 사용중인 대역폭, 패킷의 손실 비율 등을 예로 들 수 있다. 네트워크 정보는 디바이스 프로파일 제공 받을 때 시스템에 함께 전송되며, 실시간 스트리밍 서비스를 제공하는 경우, 끊김 없는 동영상 콘텐츠를 클라이언트에 제공하려면 네트워크의 현재 대역폭을 감안해 서버나 프록시에서 동적으로 콘텐츠의 비트율(Bit-Rate)을 조절해야 한다.
- 스트림 크기 : 원본 콘텐츠를 변형하여, 클라이언트에서 재생 가능한 형태로 부호화할 경우 스트림 크기는 스트리밍 형태로 서비스 될 경우에는 끊김없는 동영상의 재생을 위해서 스트림의 비트율을 조절해야 하며, Store-and-Forward 형태로 서비스 될 경우에는 콘텐츠의 모든 내용을 클라이언트에 전달하기 위해 클라이언트의 남은 저장 공간(Storage Size)보다 스트림 크기가 작아야 한다.
- 동영상의 품질 : 동영상의 품질은 공간적인 품질(Spatial Quality)과 시간적인 품질(Temporal Quality)로 분류가 가능하다. 동영상의 품질 측정 방법은 여러 가지가 있지만 본 논문에서는 제안할 시스템에서 사용하는 품질 측정 방법에 대해서만 기술하기로 한다.

### - 신호 대 잡음 비를 이용한 방법

PSNR(Peak Signal to Noise Ratio)은 영상의 신호 양에 비례한 잡음(noise)의 양을 측정하여 공간적인 품질을 측정한다[6].

$$PSNR = 10 \log_{10} \left[ \frac{255^2}{MSE} \right] [dB] \quad (1)$$

$$MSE = \frac{1}{MN} \sum_{x=1}^M \sum_{y=1}^N [f(x, y) - \hat{f}(x, y)]^2 \quad (2)$$

식 (1)은 PSNR을 구하는 수식이고, MSE(Mean Square Error)는 원래의 영상과 복원된 영상의 화소 값 간의 평균 자승 오차로 식 (2)를 이용해 산출한다. 식 (2)의  $M$ 은 이미지의 가로 사이즈,  $N$ 은 이미지의 세로 사이즈며  $x, y$ 는 비교하는 픽셀의 좌표 값을 나타낸다.  $f(x, y)$ 는 원본 이미지의 해당 좌표의 픽셀 값,  $\hat{f}(x, y)$ 는 비교 이미지의 해당 좌표의 픽셀 값을 나타낸다.

**- 프레임간의 차이(Frame Difference)를 이용한 방법**

시간적인 품질을 측정하기 위해서는 프레임간의 차이를 이용하는 방법이 있다[7]. 식 (3)과 같이 원본 콘텐츠  $O$ 의  $n$ 개의 프레임을 그레이 칼라로 변형한 뒤 현재 프레임과 다음 프레임간의 대응되는 픽셀의 차이를 구해, 이를 이차원 배열  $D$ 라 한다. 또 적응변환 된 콘텐츠  $A$ 에 대해서도 같은 작업을 행하여 이차원 배열  $d$ 를 생성한다. 이차원 배열  $D$ 와  $d$ 의 대응되는 좌표의 차에 절대값을 낸 이차원 배열을  $R$ 이라 한다.  $R$ 값의 합이 높으면 높을수록 프레임간의 차이는 증가하며, 이는 부드럽지 않은(Un-Smoothness) 동영상을 의미한다.

$$R_n = |D_n - d_n| = |(O_n - O_{n+1}) - (A_n - A_{n+1})| \quad (3)$$

**2.2 동영상 적응변환에 관한 기존 연구들의 분석**

동영상 적응변환은 어느 시기에 적응변환 작업을 하는지에 따라 두 개의 부류로 나눌 수 있다. 하나는 정적인 적응 변환이고 다른 하나는 동적인 적응 변환이다[2].

**2.1.1 정적인 적응변환**

정적인 적응변환은 서버에서 콘텐츠 마다 서로 다른 여러 가지 버전을 미리 만들어 보관하고 있다가 클라이언트의 요청이 들어오면 적합한 버전을 전송해주는 방식이다. 이 방법의 장점은 콘텐츠 제공 시 별도의 부호화 작업 없이 빠르게 콘텐츠를 전송 가능하다는 점이다. 반면 여러 가지 버전을 모두 서비스하기 위해서는 서버에 많은 저장공간이 필요하며, 서비스되는 콘텐츠의 형태는 제한이 따를 수 밖에 없다. 또한, 유동적인 네트워크 상황을 고려하지 못하므로 네트워크의 대역폭이 떨어질 경우 동영상의 재생이 끊기거나 지연되는 문제점이 발생한다.

**2.1.2 동적인 적응변환**

동적인 적응변환은 클라이언트의 요청이 있을 때마다 원본 콘텐츠를 적절한 형태로 변환 시키는 방식으로 수행 장소에 따라 다음과 같이 나뉘어진다[3]. 클라이언트 기반은 휴대폰, PDA 등의 클라이언트 내에서 다운로드

받은 콘텐츠를 직접 클라이언트 환경에 맞게 변형하는 방식으로 네트워크 및 단말기 자원 낭비 등의 문제로 잘 사용하지 않는 방법이다. 서버 기반의 적응변환은 콘텐츠를 서비스 하기에 앞서 클라이언트 및 네트워크 대역폭을 비롯한 자원상황을 토대로 콘텐츠의 분석 과정을 거쳐 각각의 상황에 적합한 콘텐츠 버전을 결정한다. 이는 클라이언트의 요청에 신속하고 정확히 대응한다는 장점이 있지만 항상 콘텐츠의 분석을 미리 해주어야 하므로 웹 상의 동영상들에 대해 클라이언트가 접근하기 위해서는 반드시 웹 서버가 콘텐츠의 분석과정을 지원해야만 한다는 제약이 따른다. 끝으로 프록시 기반의 적응변환은 적응변환 기능을 서버와 분리시켜, 적응변환에 이용되는 계산량과 서버에 집중되는 네트워크 트래픽의 부하를 줄일 수 있지만 기존 프록시 기반의 적응변환 방법들은 서버에서 콘텐츠를 분석한 내용을 제공 받아야만 콘텐츠를 적응변환 할 수 있으므로, 이를 지원하지 않는 서버의 콘텐츠는 적응변환을 할 수 없거나, 직접 콘텐츠의 분석을 행해야 하는데 이 때 분석 시간이 다운로드 타임을 크게 증가시킨다는 단점이 있다.

**2.1.3 프록시 상의 적응변환에 대한 기존 연구의 문제점**

프록시 상에서 동영상 적응변환에 관한 기존 연구로서는 유틸리티 함수(Utility Function)에 기반하여 가로축의 비트율과 세로축의 유틸리티(Utility, 동영상의 전체적인 품질)를 고려해 최적의 콘텐츠 버전을 찾는 방법[4]이 있다. 이 방법은 프레임 드랍과 DCT 계수의 드랍을 이용하되 전자를 고정시키고, 후자의 계수 드랍의 양을 조절하며 콘텐츠 버전을 생성한 뒤에, 이들 버전의 품질과 비트율을 측정하여 그림 1과 같이 이를 선 그래프로 연결한다. 이후 콘텐츠의 비트율이 변함에 따라 각각의 선 그래프와의 해당 비트율의 교점 중에서 가장 품질이 높은 점을 선택해 그 위치에 해당하는 형태로 콘텐츠를 변형시킨다. 이는 비트율의 변화에 빠르게 대응할 수 있다는 장점이 있지만 가장 적합한 콘텐츠의 형태를 찾기 위해서는 콘텐츠 버전 별로 미리 측정된 품질과 비트율 정보를 가지고 있어야 하므로 서버 기반의 동적 적응변환과 큰 차이가 없게 된다.

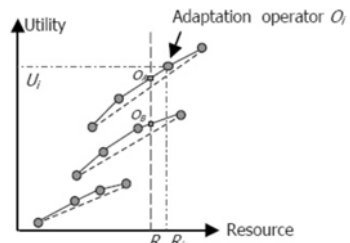


그림 1 Frame Drop과 Coefficient Drop을 고려한 유틸리티 함수

위 방법을 불특정 콘텐츠를 대상으로 실시간 품질 예측에 이용하기 위해서는 콘텐츠 특징(Content Feature)과 훈련 풀(Training Pool)을 이용하는 방법[5]이 최근 제안되었다. 이 방법은 많은 샘플을 대상으로 유틸리티 함수와 모션 벡터, 모션의 세기, DCT 계수 등을 기반으로 한 공간적, 시간적인 특징들을 추출한 뒤에, 이를 이용하여 훈련 풀을 생성하고, 동영상의 요청이 들어오면 동적으로 해당 콘텐츠의 특징을 추출한 뒤에 추출 값을 훈련 풀에 대응하여 적합한 적응화 형태를 제시하는 방법이다.

### 3. 동영상 스트림 크기 및 품질 예측 방법

본 절에서는 본 논문에서 제안하는 동영상 스트림의 크기 및 품질을 예측하는 방법을 설명하도록 한다.

#### 3.1 동영상 스트림 크기의 예측 방법

##### 3.1.1 Quality factor에 의한 압축률의 변화

MPEG-1/-2/-4, Motion JPEG, H.261, H.263과 같은 동영상 압축 기술의 표준안을 이용한 코덱들은 부호화 과정에서 Quality factor 또는 Quality level이란 이름으로 양자화(Quantization) 기법을 이용한 영상 부호의 발생량 제어 방법을 지원하고 있다. Quality factor의 값을 낮출수록 양자화 스텝의 사이즈는 늘어나게 되며, 이를 통해 영상 부호를 구성하는 계수간의 미세한 차이를 없애 이미지의 품질은 떨어지게 되지만 부호의 중복성을 높이게 된다. 중복성이 높아진 부호에 대해 Run-Length Coding과 같은 부호화 압축 기법을 이용하면 이미지를 구성하는 비트의 수를 좀 더 줄일 수 있게 된다.

일반적으로 콘텐츠마다 이미지 구성의 공간적인 복잡도가 다르기 때문에 같은 값의 Quality Factor로 부호화 하더라도 단순한 이미지는 압축률이 높고, 복잡한 이미지는 압축률이 낮게 나타난다. 그러므로 스트림 크기의 예측을 위해 우선은 콘텐츠를 Quality Factor의 최대값으로 부호화 한 뒤 스트림의 크기를 측정한다. 이때의 부호화 된 스트림의 크기는 이미지의 공간적 구성의 복잡도가 감안되어 복잡도가 높은 이미지일수록 스트림의 크기는 증가하게 된다. 이후 Quality Factor를 낮춤에 따라 스트림의 크기는 줄어들게 된다. 그림 2는 MPEG-4 기반의 WMV9(Windows Media Video 9) 코덱, XviD 코덱, 그리고 MPEG-1 기반의 Indeo@ Video 5.2 코덱을 이용할 경우 Quality Factor에 따른 각각의 스트림 크기 압축률의 상관관계를 나타낸 그래프이다. 이 그림을 통하여 Quality Factor를 낮출수록 스트림의 크기는 Quality Factor가 최대일 때의 크기에 비례해 일정한 비율로 떨어진다는 것을 알 수 있다. 그러나 코덱 내부에서 사용하는 양자화 테이블이나

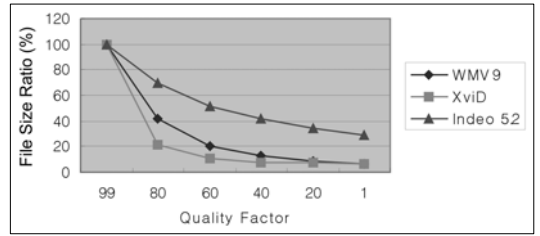


그림 2 코덱 별 Quality Factor에 따른 스트림 크기의 변화율

Quality Factor에 따라 양자화 스텝을 늘리는 정도가 서로 다르기 때문에 떨어지는 비율은 서로 다르다. 만약 Quality factor에 대한 스트림 크기 변화율을 테이블 형태로 만들어 둔 뒤에, 콘텐츠마다 Quality Factor가 최대치일 때의 스트림 크기를 측정하면, Quality Factor를 변경해 부호화 하더라도 결과물의 스트림 크기를 예측하는 것이 가능할 것이다.

##### 3.1.2 해상도에 의한 압축률의 변화

리사이징(Resizing)은 원본 콘텐츠를 복호화 한 뒤 클라이언트가 지원하는 코덱에 맞게 부호화하기에 앞서 처리 된다. 원본 이미지를 클라이언트에서 지원하는 최대 해상도에 맞춰 리사이징 한 이미지를  $C^0$ 라 하자.  $C^0$ 는 부호화 이전에는 전혀 압축이 되지 않은 DIB형태의 Raw Data이므로 지원하는 컬러수만 같다면 콘텐츠에 상관없이 스트림 크기는 해상도에 정확히 비례한다. 하지만 영상 압축 방법을 이용하여 부호화를 하면, 공간 복잡도는 유사하되, 스트림 크기의 비율은 부호화 이전과는 달라지게 된다. 그러므로 리사이징에 의해 변화되는 스트림 크기를 예측하려면 인코더의 특성이 반영된 스트림 크기를 이용해 예측을 해야 한다. 우선은 콘텐츠를 클라이언트에서 요청하는 최대의 해상도로 부호화 한 뒤 이보다 더 작은 해상도로 리사이징과 부호화를 할수록 콘텐츠의 스트림 크기의 비율은 점차 줄어들을 수 있다. 그림 3은 코덱 별로 해상도를 줄여 나감에 따라 부호화 된 스트림 크기가 어떻게 변화하는지를 나타내고 있다.

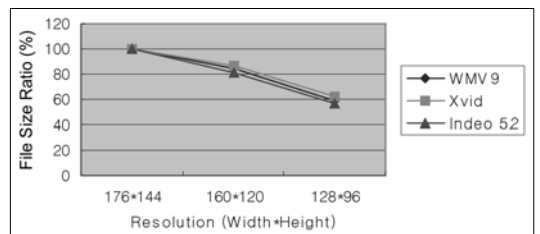


그림 3 코덱 별 해상도에 따른 스트림 크기의 변화율

3.1.3 프레임율에 의한 압축률의 변화

프레임율(Frame Rate)이란 초당 몇 장의 연속적인 이미지를 재생하는지를 의미한다. 동영상 콘텐츠의 프레임율을 떨어트리면 콘텐츠를 구성하는 프레임 중에서 일부를 제거하고 남은 프레임의 재생 시간을 연장시킨다. 이로 인해 제거되는 프레임을 구성하고 있던 데이터의 양만큼이 줄어들므로 스트림의 크기는 작아지며 프레임율을 많이 떨어트릴수록 스트림의 크기도 점차 줄어든다. 그림 4는 코덱 별로 프레임 제거율을 높여나감에 따라 부호화된 스트림 크기가 어떻게 변화하는지를 나타낸 것이다.

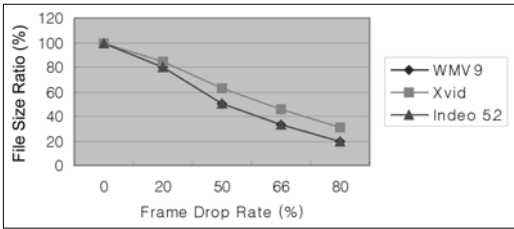


그림 4 코덱 별 프레임 제거에 따른 스트림 크기의 변화율

3.1.4 스트림 크기를 예측하는 수식

제안한 방법에서는 위의 분석을 바탕으로 적응변환된 콘텐츠의 스트림 크기를 예측하기 위해서는 식 (4)를 이용한다.

$$C'_{filesize} = C^0_{filesize} * Q_{codec}(C'_{QF}) * R_{codec}(C'_{resolution}) * F_{codec}(C'_{fps}) \quad (4)$$

$C'_{filesize}$ 는 적응변환된 콘텐츠  $C'$ 의 예측하려는 스트림 크기를 의미한다. 예측을 위해서는 우선 각각의 적응변환 기법 별 최대 품질로 부호화 한 콘텐츠,  $C^0$ 의 스트림 크기( $= C^0_{filesize}$ )를 측정한다.  $C^0$ 의 스트림 크기는 공간 복잡도 및 시간 복잡도가 반영된 수치로 여기에 적응변환 기법 별 제어 값에 따른 스트림 크기의 변화율을 곱하면 콘텐츠  $C'$ 의 스트림 크기가 예측 가능하다. 본 논문에는 앞서 언급한 세가지 적응변환 기법을 이용해 콘텐츠를 적응변환하며,  $Q_{codec}(x)$ ,  $R_{codec}(x)$ 와  $F_{codec}(x)$ 는 각각의 적응변환 기법에 따른 콘텐츠  $x$ 의 코덱 별 스트림 크기 비율을 의미한다. 각각의 함수에는 콘텐츠  $x$ 를 만들 때 이용한 제어 값을 넣어주는데  $C'_{QF}$ 는  $C'$ 의 Quality Factor를 의미하고,  $C'_{resolution}$ 은  $C'$ 의 해상도,  $C'_{fps}$ 는  $C'$ 의 프레임율을 의미한다.

3.2 동영상 품질의 예측 방법

3.2.1 Quality factor에 의한 공간적 품질의 변화  
부호화 과정에서 압축에 의해 변형된 이미지의 선행

도를 측정하는 방법은 원본과 변형본 사이의 잡음(noise)을 측정하는 PSNR을 주로 이용한다. 원본 동영상을 부호화 할 때 Quality Factor를 낮출수록 정보의 손실이 많아지므로 PSNR은 점차 낮아지지만 이 특성만 가지고는 PSNR을 예측할 수 없다. 같은 Quality Factor로 부호화를 하더라도 각 프레임마다 공간적 복잡도와 이용된 비트율이 다르기 때문이다. 그러므로 PSNR의 예측을 위해서는 우선 Quality Factor의 최대값, 99로 부호화 한 콘텐츠 버전의 프레임 별 PSNR값을 측정한다. 이 때의 PSNR값을 기준으로 이후 Quality Factor를 차츰 낮출수록 양자화 스텝의 사이즈가 증가하며, PSNR값도 줄어들게 된다. 그림 5는 MPEG-4 기반의 WMV9 코덱을 이용 시 Quality Factor의 최대값, 99로 부호화 한 콘텐츠  $C^0$ 의 프레임 별 PSNR값을 측정한 뒤에 이를 x축으로 하여 Quality Factor를 80, 60, 40, 20, 1(Quality Factor의 최소값)과 같이 다섯 단계로 낮추며 부호화함에 따라 프레임 별 PSNR값이 유사한 형태로 낮아짐을 확인할 수 있다. 이는 XviD 코덱, Indeo® Video 5.2 코덱을 이용하여도 유사한 결과가 나타났다.

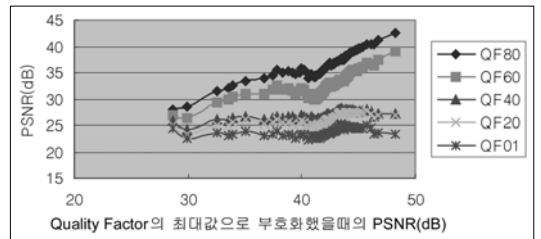


그림 5 WMV9 코덱을 이용 시 Quality Factor에 따른 PSNR 값의 변화

식 (5)는 그림 (5)의 Quality Factor에 의한 PSNR 변화율을 이용해 PSNR값을 예측하는 수식이다.

$$C'_{PSNR} = S_{codec}(C^0_{PSNR}) = \frac{1}{N - \sum_{n=1}^N D(n)} * S_{codec} \left( \sum_{n=1}^N (D(n) - 1) * PSNR(C^0_{frame(n)}) \right) \quad (5)$$

$$D(n) \begin{cases} 1, & n \text{ frame이 Drop할 경우} \\ 0, & \text{그 이외의 경우} \end{cases}$$

$C'_{PSNR}$ 은 적응변환 할 콘텐츠  $C'$ 의 PSNR 예측 값을 의미한다.  $C'_{PSNR}$ 은 Quality Factor의 최대값으로 부호화 된 콘텐츠  $C^0$ 의 프레임 별 PSNR값이 콘텐츠  $C'$ 의 Quality Factor에 의해 변화하는 값을 함수  $S_{codec}(x)$ 를 통해 예측한 뒤, 이를 평균 내어 구한다.

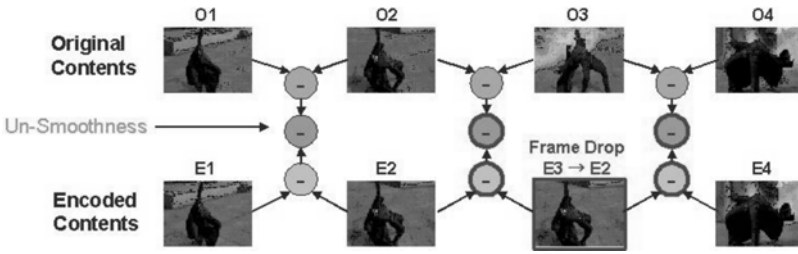


그림 6 프레임 제거에 따른 Un-Smoothness의 변화

3.2.2 프레임유에 의한 시간적 품질의 변화

동영상의 부드러운 재생 정도를 측정하기 위해서는 부호화 기법에 상관없이 적용할 수 있는 프레임간 차이를 비교하는 방법을 이용한다.

그림 6과 같이 부호화 된 콘텐츠의 세 번째 프레임, E3가 제거된 상황을 설정해보자. 프레임 간 Un-Smoothness를 측정하는 방법은 식 (3)에 기술하였듯이 원본 콘텐츠의 프레임간 차와 부호화 된 콘텐츠의 프레임간 차에 대한 차의 절대값으로 구한다. 이 때 프레임 제거와 관계없는 첫 프레임과 두 번째 프레임 사이의 Un-Smoothness 값은 변화가 없다. 두 번째 프레임과 세 번째 프레임 사이의 Un-Smoothness는 E3의 제거로 E3의 위치에 E2가 존재하므로 식 (6)과 같이 원본 콘텐츠의 프레임 O2와 O3의 차의 절대값으로 정리된다.

$$\begin{aligned} & |(O2 - O3) - (E2 - E3)| = |(O2 - O3) - (E2 - E2)| \\ & = |(O2 - O3)| \end{aligned} \quad (6)$$

세 번째 프레임과 네 번째 프레임 사이의 Un-Smoothness는 E3의 제거로 E3의 위치에 E2가 존재하므로 E2와 E4의 프레임간 차이를 구해야 하는데 부호화 후 프레임의 손실이 거의 없다는 가정하에 식 (7)과 같이 O3과 O2의 차의 절대값으로 정리한다.

$$\begin{aligned} & |(O3 - O4) - (E3 - E4)| = |(O3 - O4) - (E2 - E4)| \\ & = |(O3 - O4) - (O2 - O4)| = |(O3 - O2)| \end{aligned} \quad (7)$$

실제로 부호화 된 콘텐츠는 어느 정도의 손실이 발생하지만, 손실의 정도는 프레임간 차이에 비하면 적은 양이므로 원본 콘텐츠와 부호화 한 콘텐츠의 프레임 간 Un-Smoothness 값과 원본 콘텐츠의 프레임 간 차이만을 측정해 보관한다면 이후 원본 콘텐츠의 특정 프레임을 제거하더라도 이 둘을 이용해 Un-Smoothness를 측정하는 계산량을 줄일 수 있다.

3.2.3 유틸리티를 산출하는 수식

동영상의 품질은 공간적 품질(Spatial Quality)과 시간적 품질(Temporal Quality)로 구분 가능하며 이 둘을 조합하여 전체적인 품질을 나타내는 방법은 다음 두 가지가 있다[7].

$$Quality = \alpha * Quality_{spatial} + (1 - \alpha) * Quality_{temporal} \quad (8)$$

$$Quality = Quality_{spatial} * Quality_{temporal} \quad (9)$$

식 (8)은 공간적 품질과 시간적 품질에 가중치를 두고 더하는 방법이며, 식 (9)는 이 둘을 곱하는 방법이다. 두 영역의 품질 중 어느 한쪽의 값이 높더라도 다른 한쪽이 치명적으로 낮아 사용자가 도저히 알아보기 힘들 경우, 전체적인 품질 값이 낮아야 하는데 전자의 방법을 이용하면 이를 반영하기가 힘들다. 그러므로 어느 한쪽의 값이 낮을 경우 전체적인 품질 값을 떨어트리기 위해서는 본 논문에서는 후자의 방법을 사용한다.

기존의 이미지간 품질을 측정하는 방법들은 원본과 변형본의 해상도가 같음을 전제하므로, 본 논문에서는 적응변환 된 콘텐츠의 해상도가 클라이언트의 해상도에 얼마만큼 적합한지를 Suitability<sub>resolution</sub>으로 정의하고, 공간적인 품질, 시간적인 품질과 곱해 식 (10)과 같이 동영상 콘텐츠의 전체적인 품질로 유틸리티[4]란 이름을 빌어 표현한다. Suitability<sub>resolution</sub>은 식 (11)과 같이 적응변환 한 콘텐츠의 전체 픽셀의 개수를 클라이언트가 최대 해상도일 때의 픽셀수로 나누어 구한다.

$$\begin{aligned} Utility &= Suitability_{resolution} * Quality_{spatial} * Quality_{temporal} \\ &= Suitability_{resolution} * Normalized PSNR * (1 - Normalized UnSmoothness) \end{aligned} \quad (10)$$

$$\begin{aligned} Suitability_{resolution} &= \frac{resolution\ of\ adpated\ content}{screen\ size\ of\ client} = \frac{C'_{width} * C'_{height}}{Client_{width} * Client_{height}} \\ &(C'_{width} \leq Client_{width}, C'_{height} \leq Client_{height}) \end{aligned} \quad (11)$$

4. 동적 동영상 적응변환 시스템의 구조

4장에서는 3장의 예측 방법을 실제 시스템에 적용하는 방법 및 시스템 구조에 대하여 설명하도록 한다.

4.1 동적 동영상 적응변환 시나리오

그림 7은 본 논문에서 제안한 동적 동영상 적응변환 방법의 전체 프로시유어를 플로우차트 형식으로 나타낸 그림이다. 이 그림에 나타난 것과 같이 제안한 방법에서는 서비스 이전에 코덱의 특성을 분석하고, 클라이언트

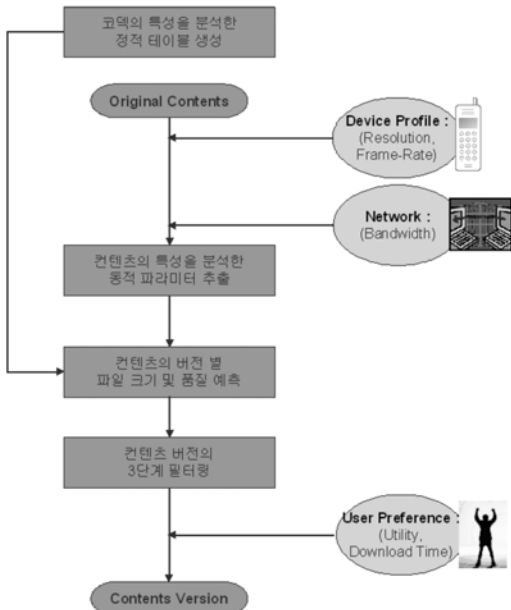


그림 7 동적 동영상 적응변환 시나리오

가 콘텐츠를 요청하면 디바이스 프로필과 네트워크 상황을 고려해 콘텐츠의 특성을 분석한 동적 파라미터를 추출한다. 이후 다양한 버전들의 스트림 크기 및 품질을 예측한 뒤 3단계의 필터링을 거치고 사용자 선호도를 반영하여 최종적으로 클라이언트에 전송할 콘텐츠의 버전을 결정짓는다.

4.2 코덱의 특성을 분석한 테이블 생성

스트림 크기와 품질을 예측하기 위해서는 우선 코덱 별로 적응변환 기법에 따른 스트림 크기 및 품질의 변화율 테이블이 필요하다. 이를 위해 프록시에서 서비스를 시작하기 이전에 많은 콘텐츠를 대상으로 적응변환 기법의 수치를 조절해나가며 변화율을 측정한 뒤 그 평균 값을 테이블로 생성한다. 생성하는 테이블의 종류는 표 1과 같다. 테이블의 생성 방법은 적응변환 기법 별로 콘텐츠의 손실을 최소화한 콘텐츠  $C^0$ 를 생성하고,  $C^0$ 에 대해 각각의 적응변환 기법에 대해 제어 값을 조절해 나가며 변화하는 스트림 크기를 측정하여 기록한다.

표 1 코덱 별로 생성하는 적응변환 기법에 따른 변화율의 테이블

| Table Name   | X축 단위           | Y축 단위      |
|--|-----------------|------------|
| Quality Factor-File Size Ratio                       | Quality Factor  | Percentage |
| Resolution-File Size Ratio                           | Width*Height    | Percentage |
| Frame Rate-File Size Ratio                           | Frame Drop Rate | Percentage |
| PSNR at Quality Factor 99 - PSNR at Quality Factor x | dB              | dB         |

PSNR 예측 테이블은  $C^0$ 의 프레임 별 PSNR 값을 측정하고, Quality Factor를 낮춰가며 프레임 별 PSNR 값을 측정하여 생성한다. 테이블 생성 시간은 적응변환 시간에 영향을 주지 않으므로 보다 많은 샘플들의 통계 값을 이용해 신뢰도를 높이며, 적응변환 기법의 제어 값을 세밀히 조절하면 보다 정확한 예측 값을 얻어낼 수 있다.

4.3 콘텐츠의 특성을 분석한 파라미터 추출

해상도와 프레임 길이가 서로 같은 콘텐츠라 할지라도 콘텐츠의 공간 및 시간적인 복잡도가 서로 다를 경우 부호화 된 콘텐츠의 스트림 크기나 품질은 서로 다르다. 또한 같은 Quality Factor로 부호화 하더라도 콘텐츠의 특성에 의해 스트림 크기나 품질은 서로 달라진다. 그러므로 부호화 된 콘텐츠의 스트림 크기나 품질을 예측하기 위해서는 원본 콘텐츠의 특성이 반영된 값이 필요하며, 이를 위해 적응변환 기법 별로 손실을 최소화한 최대 품질의 콘텐츠  $C^0$ 를 생성하고, 원본 콘텐츠와  $C^0$ 를 이용해 표 2의 파라미터를 추출한다. 스트림 크기를 예측하기 위해서는 콘텐츠의 공간 및 시간 복잡도가 반영된  $C^0$ 의 스트림 크기를 측정하며, 품질을 예측하기 위해서는  $C^0$ 의 PSNR,  $C^0$ 의 Un-Smoothness와 원본 콘텐츠의 프레임간 차이를 측정한다. 파라미터의 추출 작업은 클라이언트가 콘텐츠를 요청을 하면 수행하는 동적인 작업이므로 추출에 걸리는 시간은 적응변환 시 콘텐츠의 전송 시간을 지연시키는 요인이 된다. 이를 단축시키는 방법에 대해서는 5.2절에서 언급하기로 한다.

표 2 콘텐츠 별로 생성하는 최고 품질을 가진 콘텐츠 버전의 파라미터

| Parameter Name                        | 단위                 |
|---------------------------------------|--------------------|
| Encoded File Size of $C^0$            | Bytes              |
| PSNR of $C^0$                         | dB                 |
| Un-Smoothness of $C^0$                | Normalized Quality |
| Frame Difference of Original Contents | Normalized Quality |

4.4 콘텐츠 버전의 선택 알고리즘

스트림 크기와 품질을 예측한 콘텐츠 버전들을 대상으로 세가지 필터링 작업을 통해 부적합한 콘텐츠 버전들을 고려 대상에서 제거하고, 남은 콘텐츠 버전들에 대해 사용자 선호도(User Preference)를 반영하여 전송할 콘텐츠 버전을 결정한다.

첫 번째 필터링 작업으로 클라이언트의 남은 저장 공간을 초과하는 크기의 버전들을 고려 대상에서 제외 시

킨다. 유틸리티가 아무리 좋더라도 클라이언트의 저장 공간을 초과하는 콘텐츠는 그 내용을 모두 전달할 수 없고, 일부 동영상 압축 포맷에 따라서는 동영상 스트림 전체가 도달하지 않으면 재생이 불가능하므로 이러한 경우 스트림의 일부만 클라이언트에 전송하는 것은 무의미한 작업이 된다. 두 번째 필터링 작업으로 유틸리티의 임계 값(Threshold)을 두고 임계 값 미만의 유틸리티를 지닌 콘텐츠 버전들을 고려 대상에서 제외한다. 아무리 전송시간이 빠르거나 효율적인 압축률을 지닌 버전이라 할지라도 유틸리티가 너무 떨어져서 사용자가 알아보기가 힘들다면 콘텐츠로서 의미가 없기 때문이다. 단 첫 번째 필터링을 거치고 남아있는 콘텐츠 버전들이 모두 임계 값 미만일 경우에는 그 중에서 가장 유틸리티가 높은 버전을 남겨 놓는다. 세 번째 필터링 작업으로는 콘텐츠 버전들을 유틸리티의 내림차순으로 정렬한 후에 배열의 앞쪽부터 순차적으로 기준 버전으로 지정한다. 이후 기준 버전의 뒤 쪽에 위치한 버전들 중에서 기준 버전보다 스트림 크기가 큰 버전을 찾아 고려 대상에서 제거한다. 이는 유틸리티는 떨어지는데 오히려 스트림 크기는 큰 비효율적인 콘텐츠 버전들을 제거하기 위한 방법이다.

필터링을 거친 콘텐츠 버전들을 대상으로 사용자 선호도를 반영시켜 최종적으로 전송할 버전을 결정한다. 사용자 선호도는 유틸리티와 다운로드 시간 중 어느 쪽에 비중을 둔 콘텐츠를 원하는지를 슬라이드 바의 형태로 전달 받아 남은 콘텐츠 버전들을 품질의 내림차순으로 정렬한 뒤 슬라이드 바의 위치에 비례하여 배열 내의 콘텐츠 버전을 선택, 전송한다.

### 5. 실험 및 분석

#### 5.1 실험 결과 비교 및 분석

본 실험에서는 콘텐츠 버전 별로 스트림 크기 및 품질을 예측 한 값과, 실제로 콘텐츠를 만들어 스트림 크기와 품질을 측정한 값을 비교하고 예측의 정확도를 평가한다. 적응변환 기법 별 제어 값은 Quality Factor는 99~1 사이의 범위를 6단계로 나누었으며, 해상도는 H.263에서 제시한 QCIF(176\*144), SQCIF(128\*96)와 MPEG-1에서 제시한 SIF(160\*120), 이렇게 세 단계를 대상으로 한다. 프레임 제거율은 0%, 20%, 50%, 66%, 80%로 늘려가며 실험을 하였다. 이 경우 조합 가능한 콘텐츠의 버전 개수는 총 90가지가 된다. 본 실험은 15초~30초 길이의 20개의 샘플 동영상에 대해 각각 90개의 버전을 실제로 만들어 측정한 값과 한 개의 버전만을 생성하고, 나머지 버전은 예측한 값을 서로 비교해 보았다.

##### 5.1.1 스트림 크기의 예측 실험

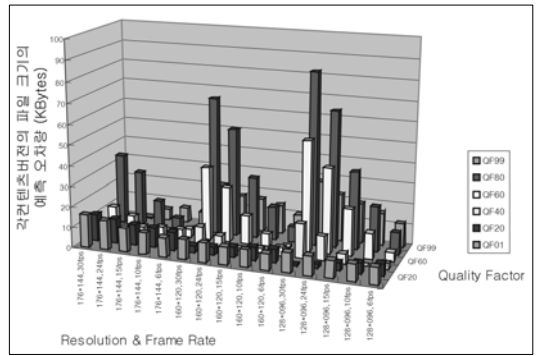


그림 8 콘텐츠 버전 별 스트림 크기의 평균 예측 오차량

그림 8은 20개의 샘플에 대해 콘텐츠 버전 별로 실험과 예측 스트림 크기 차의 절대값을 평균 낸 그래프이다. 실험 스트림 크기가 90개 버전에 대해 40KBytes~6MBytes, 평균 852KByte임을 감안하면 예측으로 인해 발생으로 발생하는 평균 오차량은 14KByte 정도로 매우 적은 양임을 알 수 있다. 평균 오차량이 가장 큰 버전 (Quality Factor 80, Resolution 128\*96, Frame Rate 30fps)에서도 실제 스트림 크기와 비교하면 4.5%의 정도의 오차율을 보여주었으며, 가장 오차율이 큰 버전 (Quality Factor 1, Resolution 128\*96, Frame Rate 6fps)에서는 22% 정도의 오차율이 발생했지만 이 경우에는 콘텐츠의 스트림 크기가 매우 적으므로(평균 40Kbytes) 예측 오차의 정도가 큰 문제가 되지 않는다.

##### 5.1.2 유틸리티의 예측 실험

콘텐츠 버전 별로 실험과 예측을 통해 얻은 유틸리티의 절대값의 차를 낸 뒤에 20개의 샘플에 대해 평균값을 내어 비교를 해보았다. 그 결과 그림 9와 같이 유틸리티의 평균 예측 오차량은 최대 0.026이 나타났으며, 이는 유틸리티의 범위가 0~1 사이인 점을 고려하면 무시해도 좋은 수준의 예측 오차량이다.

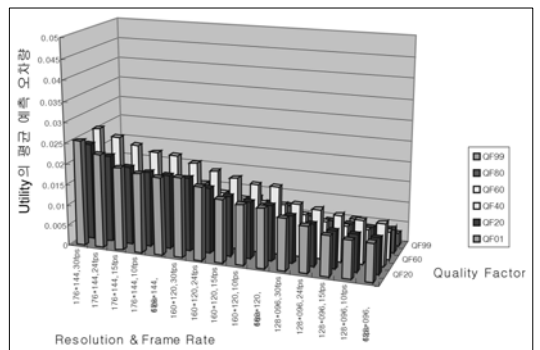


그림 9 콘텐츠 버전 별 유틸리티의 평균 예측 오차량



**5.2 예측 속도 향상을 위한 방법**

동적 적응변환에서는 예측의 정확성과 함께 예측 소요 시간도 시스템의 평가에 있어 중요한 요소가 된다. 품질 예측에 소요되는 시간을 줄이기 위해서는 우선 품질 측정에 이용되는 픽셀 수를 줄이는 방법이 있다. PSNR과 Un-Smoothness는 모두 프레임간 픽셀 차이 값을 이용하는 방법으로 비교하는 픽셀의 수를 줄여나 가면 품질 측정의 정확도는 떨어지지만 계산 횟수는 줄어들게 된다. 또 다른 방법으로 PSNR 측정에 이용되는 프레임의 수를 줄이는 방법이 있다. PSNR측정 시 프레임 수를 줄이면 픽셀 수를 줄이는 것과 마찬가지로 계산량의 이득을 얻게 된다. 하지만 Un-Smoothness의 경우에는 모든 프레임 간의 차이를 미리 측정해야 특정 프레임이 제거되더라도 그때의 Un-Smoothness값을 예측할 수 있으므로 프레임 수를 줄이는 방법을 적용하지는 않는다.

위의 속도 향상 방법을 조합하여 품질 측정의 소요 시간 및 예측 오차율이 얼마나 차이가 있을 지를 실험을 통해 알아보았다. 실험에 사용한 샘플은 해상도 640\*304에 전체 프레임 720개의 동영상 스트림이며, 다음 세 가지 방법으로 측정해 보았다.

- (1) 속도 향상 방법을 적용하지 않은 결과
- (2) 품질 측정에 1/4픽셀과 1/30프레임만을 사용한 결과
- (3) 품질 측정에 1/256픽셀과 1/120프레임만을 사용한 결과

실험 결과(표 3 참조) 보다 적은 픽셀과 프레임을 품질 측정에 이용할수록 소요 시간이 줄어들지만 반면에 예측의 오차율은 점차 증가하는 결과를 보여주었다. 하지만 품질의 기준으로 이용된 유틸리티 값의 범위가 0~1 사이임을 고려하면 증가한 오차율은 그다지 무리가 없는 수준으로 나타났다. 본 실험 결과 동영상의 품질 측정에 이용되는 픽셀 수와 프레임 수는 전체 소요 시간을 결정하는 중요한 제약 요소이며, 샘플의 해상도나 전체 프레임의 길이가 길수록 품질 예측을 위해 소요되는 시간 또한 증가하게 됨을 확인하였다. 또한 품질 측정을 위한 계산량은 프로시에 부담을 주게 되므로, 동적 적응변환을 요청하는 클라이언트의 수가 많을 경우나 콘텐츠의 크기가 큰 경우에는 품질 예측의 오차 범위 내에서 보다 많은 픽셀과 프레임을 드랍하여 품질

예측 시간과 계산량을 줄이는 것을 권장한다.

**5.3 기존 연구 방법과의 성능 비교**

본 실험에서는 S. F. Chang이 제안한 방법[4]과 본 논문에서 제안한 방법의 비교 실험을 통해 각각의 특징을 분석하고, 본 논문의 제안방법에 대한 타당성을 검증토록 한다. S. F. Chang이 제안한 방법[4]은 콘텐츠의 해상도의 조절은 고려하지 않았으며, 프레임 제거율을 몇 단계로 나누고, 각각의 단계에 Quality Factor을 달리한 몇 개의 버전을 부호화를 거쳐 실제로 만든 뒤 스트림 크기 및 품질을 측정하고 그 외의 버전에 대해서는 보간법을 이용해 예측하였다. 실험 조건을 같도록 하기 위해 본 실험에서는 표 4(a)에 명시한 바와 같이 해상도는 176\*144로 고정시켰으며, 프레임 제거율을 다섯 단계로 나누고, Quality Factor을 6단계로 나누어 총 30개의 콘텐츠 버전을 비교 대상으로 정하였으며, 실험 방법은 표 4(b)에 언급한 바와 같이 첫 번째 방법은 S. F. Chang이 제안한 방법을 적용해 다섯 단계의 프레임 제거율과 세 단계의 Quality Factor에 대해 각각 콘텐츠 버전을 생성한 뒤 스트림 크기 및 품질을 측정하였고, 나머지는 보간법을 이용해 예측하였다. 두 번째 방법은 본 논문에서 제안한 방법을 적용해 단 한 개의 콘텐츠 버전에 대해서만 스트림 크기 및 품질을 측정하고 나머지 버전들은 적응변환 기법에 의한 변화율 테이블을 이

표 4 실험에 이용된 콘텐츠 버전의 종류 및 제어 조건의 비교

(a) 적응변환 기법 별 구분 단계 및 각 단계별 제어 값

| 적응변환 기법        | 구분 단계 | 각 단계별 제어 값               |
|----------------|-------|--------------------------|
| Quality Factor | 6단계   | 99, 80, 60, 40, 20, 1    |
| 해상도            | 1단계   | 176*144                  |
| 프레임 제거율        | 5단계   | 0, 20, 50, 66, 80 (단위 %) |

(b) 스트림 크기 및 품질 측정에 이용한 제어 조건의 비교

| 방법  | 프레임 제거율                  | Quality Factor  | 실제 생성한 버전의 개수 | 예측 방법             |
|-----|--------------------------|-----------------|---------------|-------------------|
| (1) | 5단계 (0, 20, 50, 66, 80%) | 3단계 (99, 60, 1) | 15            | 보간법 (S. F. Chang) |
| (2) | 1단계 (0%)                 | 1단계 (99)        | 1             | 테이블 이용 (제안한 방법)   |

표 3 실험 별 콘텐츠 버전의 및 품질 예측 소요 시간과 평균 오차량

| 실험 방법 | 부호화에 걸리는 시간 | PSNR 측정 시간 | Un-Smoothness 측정시간 | 전체 소요 시간 | 품질의 평균 예측 오차 |
|-------|-------------|------------|--------------------|----------|--------------|
| (1)   | 5초          | 14초        | 17초                | 36초      | 0.0121       |
| (2)   | 5초          | 4초         | 13초                | 22초      | 0.0205       |
| (3)   | 5초          | 2초         | 11초                | 18초      | 0.0407       |

(c) 실험에 이용한 샘플

| 샘플  | 원본 해상도  | 전체 프레임 수 | 분류 특징                 |
|-----|---------|----------|-----------------------|
| 샘플1 | 640*352 | 578      | 변화율 테이블과 편차가 가장 큰 샘플  |
| 샘플2 | 672*352 | 500      | 변화율 테이블과 편차가 가장 작은 샘플 |

표 5 실험 방법 및 샘플에 대한 소요시간과 예측의 오차 정도

(a) 실험 방법 및 샘플에 대한 크기와 품질 측정에 소요되는 시간 (단위: 초)

| 실험 방법       | 실험 샘플 | 부호화에 걸리는 시간 | 품질 측정 시간 | 전체 소요 시간 |
|-------------|-------|-------------|----------|----------|
| S. F. Chang | 샘플1   | 109         | 115      | 224      |
|             | 샘플2   | 98          | 102      | 200      |
| 제안한 방법      | 샘플1   | 8           | 32       | 40       |
|             | 샘플2   | 7           | 30       | 37       |

(b) 실험 방법 및 샘플에 대한 샘플 별 스트림 크기와 품질의 평균 예측 오차

| 실험 방법       | 실험 샘플 | 스트림 크기의 평균 예측 오차율 | 품질의 예측 오차량 |
|-------------|-------|-------------------|------------|
| S. F. Chang | 샘플1   | 14.19%            | 0.022      |
|             | 샘플2   | 16.1%             | 0.027      |
| 제안한 방법      | 샘플1   | 16.15%            | 0.028      |
|             | 샘플2   | 6.21%             | 0.0034     |

용해 스트림 크기 및 품질을 예측하였다. 실험에 이용한 샘플은 표 4(c)에 언급한 두 가지 샘플을 이용하였다.

실험 결과 S. F. Chang이 제안한 방법은 많은 콘텐츠 버전을 직접 만들어 보관법에 덜 의존할수록 예측률이 높아지므로 표 5(b)와 같이 예측 오차는 제안한 방법과 유사하면서 표 5(a)와 같이 부호화 및 품질 측정에 걸리는 시간은 5.5배 걸렸다. 반면에 본 실험에서 제안한 방법을 이용하면 테이블과 샘플의 변화를 편차에 따라 어느 정도의 오차율은 발생하지만 S. F. Chang이 제안한 방법에 비해 속도를 대폭 줄일 수 있으며 5.2절에서 제안한 속도 향상 방법을 이용하면 여기에 50% 이상의 속도 향상 효과를 얻을 수 있으므로, 콘텐츠의 분석 정보를 지원하지 않는 일반적인 웹 서버의 동영상에 대해 유용한 적응변환 방법임을 확인했다.

6. 결론

프록시 상의 동적 동영상 적응변환은 변화하는 클라이언트 및 네트워크 환경을 고려해 콘텐츠를 이에 알맞도록 변형시켜준다는 장점이 있지만, 기존의 방법은 서버에서 많은 시간을 들여 콘텐츠를 분석한 뒤 분석 정보를 프록시에 제공해야 한다는 제약이 따른다. 그러므

로 콘텐츠 분석을 지원하지 않는 일반적인 웹 서버의 동영상에 접근하기 위해서는 프록시에서 반복적인 복호화, 부호화 작업을 거쳐 콘텐츠를 분석해야 하는데 이로 인해 전송시간이 지연되는 문제점이 발생한다.

본 논문에서는 기존의 서버 의존적인 동적 적응변환 방법과 이로 인해 발생하는 전송 시간의 지연 문제를 해결하기 위해 클라이언트에서 지원하는 코덱의 특성을 미리 분석해, 단 한번의 복호화, 부호화 작업만으로 클라이언트 환경에 적합한 콘텐츠의 형태를 찾는 방법을 제안하였다. 제안한 방법은 우선 클라이언트에서 지원하는 코덱 별로 부호화 특성을 파악하기 위해 적응변환에 사용되는 기법에 의한 스트림 크기 및 품질의 변화율을 서비스 이전에 미리 정적인 테이블로 만들어 놓으며, 이후 클라이언트가 콘텐츠를 요구할 때마다 한번의 복호화 및 부호화 작업만으로 해당 콘텐츠의 공간 복잡도와 시간 복잡도가 반영된 스트림 크기 및 PSNR과 Un-Smoothness, Frame-Difference를 동적으로 측정하여 앞서 만들어둔 테이블과 조합해 콘텐츠를 어떤 형태로 변형하더라도 콘텐츠를 직접 만들 필요 없이 스트림 크기와 품질을 예측하는 방법을 고안하였다. 또한 콘텐츠의 다양한 형태 중에서 클라이언트의 남은 저장 공간과 일정 수준 이상의 품질, 압축률의 효율성, 사용자 선호도 등을 고려하여 가장 적합한 버전을 결정하는 선택 알고리즘을 제안하였으며, 예측 오차율을 일정 수준으로 유지하는 선에서 50% 이상의 속도 향상을 얻는 방법을 제시하였다.

기존의 방법과 제안한 방법을 비교 실험한 결과, 적응변환 시간을 최대 91%까지 줄일 수 있었으며 이에 반해 스트림 크기와 품질의 예측 오차율은 각각 평균 10%와 4% 수준으로 크게 문제삼지 않아도 될 수치로 나타났다. 제안한 방법을 이용하면 모바일 클라이언트 환경에서 웹 상에 존재하는 동영상 콘텐츠에 대해 자유로이 접근하는 것이 가능하며, 제안한 시스템을 통해 프록시 상의 동적 동영상 적응변환에 적합한 시스템을 구현하였다. 본 논문의 예측 방법을 확장하면 동영상의 다운로드 서비스뿐만 아니라 실시간 스트리밍 서비스에 적용하는 것도 가능하며, 프록시 상의 QoS에 대한 조절 방안이나, 비디오뿐만 아니라 오디오에 관한 적응변환 방법도 차후 연구할 계획이다.

참고 문헌

[1] Ingo Elsen, Frank Hartung, Uwe Horn, Markus Kampmann and Liliane Peters, "Streaming Technology in 3G Mobile Communication Systems," *IEEE Computer Magazine*, Vol. 34 Issue: 9, Sep. 2001 pp. 46-52.

[2] Zhijun Lei and Nicolas D. Georganas, "Context-

- based Media Adaptation in Pervasive Computing," *Proceedings of Canadian Conference on Electrical and Computer Engineering (CCECE 2001)*, Toronto, May 2001.
- [3] Margaritis Margaritidis and George C. Polyzos, "Adaptation Techniques for Ubiquitous Internet Multimedia," *Wireless Communications and Mobile Computing, Vol. 1, no. 2*, April-June, 2001, pp. 141-63.
- [4] Jae-Gon Kim, Yong Wang, Shih-Fu Chang "Content-Adaptive Utility-Based Video Adaptation," *Proceedings of IEEE ICME'2003, Vol. 3, Baltimore, MD, USA*, July, 2003, pp. 281-284.
- [5] Yong Wang, Jae-Gon Kim, Shih-Fu Chang, "Content-Based Utility Function Prediction for Real-Time MPEG-4 Video Transcoding," *Proceedings of ICIP'2003, Barcelona, Spain*, Sep., 2003, pp. 14-17.
- [6] Thrasyvoulos N. Pappas, "Perceptual Criteria for Image Quality Evaluation," in *Handbook of Image and Video Processing, Academic Press, San Diego*, 2000, pp. 669-684.
- [7] Christoph Kuhmunch, Gerald Kuhne, Claudia Schremmer and Thomas Haenselmann, "Video-Scaling Algorithm Based on Human Perception for Spatio-Temporal Stimuli," *Proceedings of SPIE, Multimedia Computing and Networking, vol. 4312, San Jose, California, USA*, Jan., 2001, pp. 13-24.



김 종 항

2002년 2월 서강대학교 컴퓨터학과(공학사). 2004년 2월 서강대학교 컴퓨터학과(공학석사). 2004년 5월~LG전자 홈넷사업팀(연구원). 관심분야는 멀티미디어 시스템, 동영상 분석, 동영상 변환



남 종 호

1986년 2월 서강대학교 전자계산학과 졸업(학사). 1988년 2월 한국과학기술원 전산과 졸업(석사). 1992년 2월 한국과학기술원 전산과 졸업(박사). 1992년 3월~1992년 8월 한국과학기술원 정보전자 연구소(연구원). 1992년 9월~1993년 8월 일본 Fujitsu 연구소(방문연구원). 1993년 9월~현재 서강대학교 컴퓨터학과 교수. 관심분야는 멀티미디어 시스템, 동영상 검색, 동영상 분석, 병렬/분산 처리, 인터넷 컴퓨팅