

특집논문-03-08-4-01

UMA를 위한 유틸리티 기반 MPEG-21 비디오 적응

김 재 곤*, 김 형 명**, 강 경 옥*, 김 진 용*

Utility-Based Video Adaptation in MPEG-21 for Universal Multimedia Access

Jae-Gon Kim*, Hyung-Myung Kim**, Kyeongok Kang* and Jinwoong Kim*

요 약

가변의 자원제한(resource constraint) 및 사용자 선호(preference)를 만족하기 위한 비디오 적응(adaptation)은 다양한 사용자 단말과 이종망을 통한 범용적멀티미디어접근(Universal Multimedia Access: UMA)을 위한 주요 요소기술이다. 기준의 많은 비디오 적응 기법이 존재하지만 주어진 자원제한을 만족하는 최적의 적응을 제공하기 위한 체계적인 기법은 제시되지 못하였다. 본 논문에서는 체계적인 최적의 적응을 제공하기 위하여, 적응단위를 정의하는 비디오 엔티티(entity), 주어진 자원제한을 나타내는 리소스(resource), 적용 가능한 적응동작을 규정하는 적응(adaptation), 각 적응결과 비디오 품질(quality)을 나타내는 유틸리티/utility)의 관계를 모형화하는 개념적인 적응 프레임워크를 제시하고자 한다. 본 프레임워크는 다양한 형태의 적응의 문제를 제한된 자원에서 유틸리티를 최대화하는 문제(resource-constrained utility maximization)로 정형화(formulation)할 수 있도록 한다. 본 논문에서는 이 프레임워크를 프레임 제거(frame dropping)과 DCT 계수 제거(coefficient dropping)을 이용한 MPEG-4 압축 비디오의 비트율 적용의 실제적인 예에 적용한다. 또한 상호연동 가능한(interoperable) 형태로 터미널 및 네트워크 QoS(Quality of Service)를 제공하기 위한 블록 MPEG-21 Digital Item Adaptation(DIA)에 채택된 기술자(descriptor)를 제시한다. 이 기술자는 본 프레임워크의 적응-리소스-유틸리티의 관계를 유틸리티 함수(utility function)를 이용하여 기술(description)한다. 실험을 통하여 본 논문의 표준 기술자를 사용하는 적응 프레임워크의 타당성을 보인다.

Abstract

Video adaptation in response to dynamic resource conditions and user preferences is required as a key technology to enable universal multimedia access (UMA) through heterogeneous networks by a multitude of devices in a seamless way. Although many adaptation techniques exist, selections of appropriate adaptations among multiple choices that would satisfy given constraints are often ad hoc. To provide a systematic solution, we present a general conceptual framework to model video entity, adaptation, resource, utility, and relations among them. It allows for formulation of various adaptation problems as resource-constrained utility maximization. We apply the framework to a practical case of dynamic bit rate adaptation of MPEG-4 video streams by employing combination of frame dropping and DCT coefficient dropping. Furthermore, we present a descriptor, which has been accepted as a part of MPEG-21 Digital Item Adaptation (DIA), for supporting terminal and network quality of service (QoS) in an interoperable manner. Experiments are presented to demonstrate the feasibility of the presented framework using the descriptor.

* 한국전자통신연구원 디지털방송연구단 방송미디어연구그룹
Broad Media Research Group, Digital Broad Research Division, ETRI
** 한국과학기술원 전자 전산학과 전기 및 전자공학 전공
Dept. of Electrical Engineering and Computer Science, KAIST
※ 본 연구는 정보통신부 과제의 일부로서 수행되었습니다.

I. 서 론
최근 이종망과 다양한 사용자 단말(TV, PC, PDA, cellular phone 등)을 통하여 언제 어디서나 끊김없이(seamless) 멀

티미디어 컨텐츠를 접근/소비할 수 있는 UMA 멀티미디어 응용 서비스가 대두되고 있다^{[1][2][3]}. 이러한 멀티미디어 환경에서 UMA 서비스를 위해서는 네트워크 상태, 사용자 단말 특성 및 사용자 선호도 등의 사용자 환경에 대응하기 위한 컨텐츠 적응이 핵심요소 기술로 요구된다.

비디오 적응은 공간해상도 적응, 공간화질 적응, 프레임 울적응, 비디오 재생길이를 조정하는 비디오 요약(summarization) 등 다양한 형태를 포함한다. 특히, 다양한 단말과의 접속을 위한 가변대역 접속망이나 가변의 망 상태에 적응하기 위한 비트율 적응은 UMA 환경에서 전형적으로 요구되는 적응이다. 본 논문에서는 미리 부호화된 비디오의 비트율을 다양한 울로 변환함으로써 동일한 비디오를 여러가지 다른 울의 비디오로 저장하지 않고도 요구되는 가변 가용대역에 대응할 수 있는 울적응변환(rate adaptation transcoding)을 다룬다.

기존의 울적응변환^[4]은 재양자화(requantization)^{[5][6][7]}, CD(DCT Coefficient Dropping)^[8] 등의 공간영역 적응, 프레임 울을 조정하는 FD(Frame Dropping) 등의 시간영역 적응^{[9][10][11]}, 객체기반 적응^[12] 등이 있다. 한편 scalable 비디오의 경우 bit-plane 수를 조절함으로써 비트율을 조정하는 FGS(Fine Granularity Scalability)가 가변 대역에 따른 울적응을 위한 scalable 부호화 툴로 MPEG-4에 채택되었다^{[13][14]}.

그러나 기존의 적응 기법들은 주로 주어진 제한 리소스를 만족하면서 적응 자체의 성능을 개선하는데 집중되었으며, 동적 제약을 만족시키는 최적의 적응기(adaptation operator)를 선택하기 위한 체계적인 기법을 제시하지 못하였다. 특히 UMA 환경에서는 기존의 성능을 우선시하여 서버에서 비실시간으로 수행하는 적응과 달리, 넓은 범위의 가변 리소스에 동적으로 대응하기 위한 네트워크 노드(proxy, gateway, router 등)에서의 실시간 적응이 요구된다. 또한 주어진 제한을 만족하는 다수의 적응기로부터 최적의 적응기를 선택할 수 있는 적응선택규칙(adaptation strategy)이 요구된다.

이러한 UMA 환경에서 요구되는 울적응변환을 위하여 본 논문에서는 선행연구^{[15][16]}에서 그 개념이 소개된 유틸리티 기반의 적응 프레임워크를 도입한다. 적응 문제를 결정짓는 3가지 주요 요소인 적응, 리소스, 유틸리티의 분포와 그 상호관계를 모형화하고, 이를 유틸리티 함수를 이용하여 기술(description) 함으로써 체계적인 최적의 적응을 가능하게 한다. 이와 같은 접근 방법은 기존의 울-왜곡

(rate-distortion: R-D) 모형에서 다양한 종류의 리소스와 유틸리티를 수용, 그 개념을 확장한 것으로 리소스-유틸리티는 울-왜곡과 유사하지만 훨씬 넓은 개념을 갖는다. 적응 방법으로는 계산의 효율성을 고려하여 완전복호화하지 않고 압축영역에서 특정부분을 제거함으로써 실시간 적응에 적합한 FD와 CD를 결합한 FD-CD 방법을 사용한다.

유틸리티 기반의 프레임워크는 적응, 리소스, 유tility를 적절히 정의하면 다양한 형태의 적응에 적용할 수 있는 일반적인 접근방법이다. 본 논문의 FD-CD를 이용한 MPEG-4 비디오 울적응은 무선 모발 비디오 응용에서의 일반적인 적응으로 유틸리티 기반 프레임워크이 적응 문제에 어떻게 계산적인 해를 제공하는지를 보여줄 수 있는 적절한 적용 예라고 할 수 있다.

유tility 기반 프레임워크에서 주요 이슈는 유tility 함수의 생성이다. On-demand 응용과 같이 저장된 비디오의 경우는 계산량이 크게 문제되지 않는 서버에서 off-line의 계산으로 미리 구할 수 있다. 반면 라이브 비디오의 경우 실시간 생성이 요구되며, 이를 위하여 비디오의 내용특징(content feature)과 유tility 함수의 상관관계를 활용하여 내용특징 추출과 통계적 방법을 도입한 내용기반 예측 기법을 제시하였다^[17]. 본 논문에서는 계산에 의한 유tility 함수의 생성만을 고려한다.

한편 UMA와 같이 다양한 망과 사용자 및 단말로 구성된 멀티미디어 프레임워크에서 멀티미디어 응용 서비스를 위해서는 상호연동 가능한(interoperable) 단-대-단 적응 기술이 요구된다. MPEG에서는 이러한 멀티미디어 응용 서비스를 위한 기술로 “MPEG-21: 멀티미디어 프레임워크”^[3]의 표준화를 진행하고 있다. 이 중 MPEG-21 DIA에서는 네트워크 특성, 단말 성능, 사용자 특성 및 자연환경을 포함하는 사용환경(Usage Environment), 그리고 미디어 리소스 및 메타데이터 적응을 위한 정보를 기술하기 위한 표준 기술자를 규정하고 있다. 유tility 기반 접근방법이 임의의 적응엔진에서 적용되기 위해서는 유tility 함수가 상호연동 가능한 형태로 기술되어야 한다. 본 논문에서는 MPEG-21 DIA의 터미널 및 네트워크 QoS를 위한 기술자로 제안하여 표준 툴의 일부로 채택된 유tility 함수 기술자에 대해서 기술한다.

이 논문의 구성은 다음과 같다. II 장에서는 기존의 울적응 기법과 본 논문의 접근방법에 대한 배경에 대해서 기술한다. III 장에서는 유tility 기반 적응 프레임워크에 대해 소개하고, IV 장에서는 본 프레임워크의 한 구현 예로

FD-CD를 이용한 유틸리티 기반의 윤적응변환을 제시한다. V 장에서는 유틸리티 함수를 기술하기 위한 기술자와 그 사용법을 예시하고, VI 장에서는 유틸리티 기반의 윤적응변환 접근방법에 대한 타당성을 보인다. 마지막으로 VII 장에서 결론을 맺는다.

II. 기존의 윤적응 변환기법 및 배경

윤변환의 가장 직접적인 방법은 복호화기와 부호화기를 직렬연결한 re-encoding으로 입력되는 암축 비트열을 화소 영역으로 복호화한 후 새로운 비트율로 재 부호화를 하는 방법이다. 이 방법은 가장 좋은 화질의 성능을 얻을 수 있지만 계산이 복잡하다는 단점이 있다. 기존의 윤적응변환 기법들은 re-encoding을 기준으로 최대한 화질을 유지하면서 변환구조를 간략화하여 계산량을 줄이는, 즉 화질과 복잡도의 타협(trade-off)에 집중되었다고 볼 수 있다. 원 암축 비디오에 포함된 정보(움직임 벡터 등)를 재사용하거나 복호화기-부호화기의 병렬구조를 간략화함으로써 어느 정도의 화질은 유지하면서 계산량을 상당히 줄일 수 있는 기법들이 제안되었다.

재양자화는 암축 비디오를 부분복호화하여 DCT 계수를 얻고 이를 계수를 더 큰 양자화 레벨로 양자화하여 비트율을 줄이는 것으로 re-encoding에 비해서 상당량의 계산량을 절감할 수 있다. 하지만 부호화기에서 예측부호화에 사용한 기준 프레임과 재양자화로 인한 기준 프레임의 열화로 복호화기에서 움직임 보상에 사용한 기준 프레임의 불일치(즉, prediction memory mismatch)로 인하여 연속적으로 예측 부호화된 프레임의 화질열화, 즉 drift error를 야기한다. 이러한 drift error는 종종 심한 화질열화를 야기할 수 있으며 이를 개선하기 위한 기법들이 연구되었다^{[5][6][7]}.

재양자화와 약간 다른 형태의 공간화질 적응 기법으로 DCT 계수를 재양자화하지 않고 단순히 각 매크로블록마다 고주파 계수에 대항하는 부호(code word)를 제거함으로써 율 감소를 얻는 DRS(dynamic rate shaping)^[8]이 제시되었다. 본 논문에서는 CD로 명명한 DRS는 DCT 계수를 구하기 위한 양자화와 재양자화의 과정이 없으므로 재양자화보다 더 간단한 방법이다.

FD는 원 암축 비디오에 포함된 프레임을 생략(skip)하여 시간축의 해상도 감소로 율 감소(rate reduction)를 얻는 시간영역의 적응 방법이다. 예측 부호화의 기준 프레임이 제

거된 경우 제거되지 않은 바로 이전 프레임을 새로운 기준 프레임으로 하여 잔여신호를 다시 구하여야 하는, 즉 re-encoding을 해야하는 문제가 발생한다. 원 암축 비디오에 포함된 움직임 벡터를 보정하여 재사용하거나^{[9][10]}, 새로운 기준 프레임에 대한 잔여신호를 DCT 영역에서 구함으로써^[11] 계산량을 줄이는 기법들이 보고 되었다.

그러나 대부분의 기준의 적응 기법들은 주로 제한된 리소스를 만족하는 임의의 적응에 대해서 적응 자체의 성능을 개선하는데 집중되었으며, 동적 제약을 만족시키는 최적의 적응기를 선택하기 위한 체계적인 기법을 제시하지 못하였다. 즉, 주요 목적은 어떻게 re-encoding에서 얻을 수 있는 화질을 최대한 유지하면서 re-encoding의 구조를 간략화하여 계산량을 줄이는 가에 있었고, 최적의 적응을 선택하기 위한 적응선택규칙에 대한 연구는 미미하다.

적응선택규칙에 대한 연구로는 FD의 경우 입력 암축 비디오의 움직임 벡터와 변환에 의한 오차를 기준으로 제거되는 프레임 수를 조절하여 변환된 비디오가 가능한 자연스런 움직임을 갖도록 하는 다소 임시방편적인(ad-hoc) 방법이 제시되었다^[11]. 다른 한편으로는 비디오 부호화 단계에서 R-D 특성을 이용하여 최적의 프레임 율과 공간화질을 위한 율제어(rate control)에 대한 기법이 제시되었다^{[18][19]}. 이러한 해석적인 접근방법은 그 효과성에도 불구하고, 복잡하고 비선행적인 적응을 모형화해야 하는 일반적인 적응문제에 적용하기에는 어려워 보인다. 또한 다양한 형태의 리소스와 적응을 수용할 수 있도록 확장이 용이하지 않은 면이 있다.

UMA 환경에서의 윤적응은 기존의 서버에서 수행하는 최적 성능을 목표로한 비실시간 적응과는 달리, 다음과 같은 측면에서 적응 자체의 성능 보다는 넓은 범위에 걸쳐서 가변하는 리소스에 어떻게 효율적으로 최적의 적응으로 대응할 수 있는가 하는 것이 관건이라고 할 수 있다. 즉, 일반적으로 다 수의 적응기가 주어진 제한을 만족할 수 있으며, 이 때 최적의 적응기를 선택하는 체계적인 적응선택규칙이 필요하다. 또한 이종망 또는 단말 접속망과의 네트워크 노드에서 넓은 범위의 가변 가용대역폭(available bandwidth)에 대응할 수 있는 실시간 적응이 요구되므로 적응 방법의 복잡도는 최소화되어야 한다. 특히 하나의 비디오가 다른 접속 대역의 다수의 단말로 동시에 접속될 때 그 복잡도의 정도는 한층 더 결정적인 요인이 된다. 이러한 UMA에서 요구되는 요구조건들을 수용하기 위한 접근방법으로 본 논문에서는 유틸리티 기반의 적응 프레임워크를 도입하고자 한다.

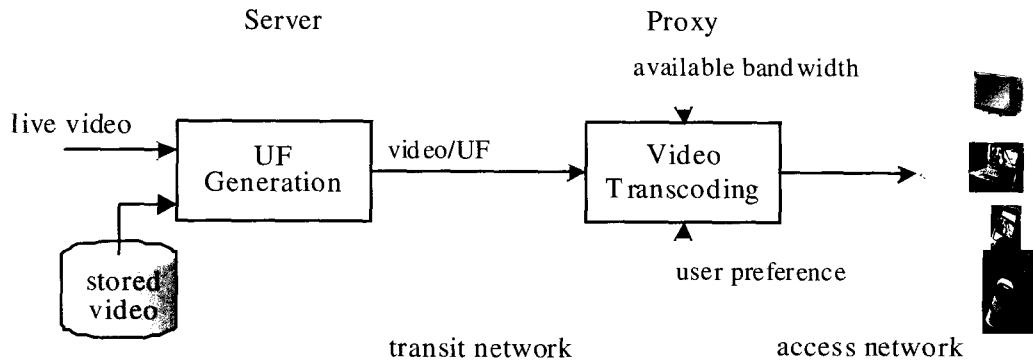


그림 1. 유용성 기반의 3 계층 비디오 적응 구조

Fig. 1. A three-tier adaptation architecture using the utility-based framework

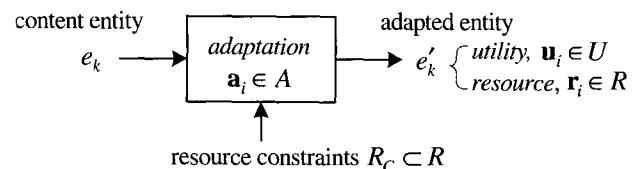
III. 유용성 기반 적응 프레임워크

이 장에서는 유용성 기반의 적응 프레임워크를 소개한다. 또한 resource-constraint utility maximization 또는 utility-constraint resource minimization의 비디오 적응 문제의 정형화를 기술한다. 그림 1은 UMA 환경에서의 다양한 리소스 제약을 만족하기 위한 서버-프록시(proxy)-단말의 3 계층 적응 구조이다. 적응 엔진을 proxy에 둠으로써 이종 망과 단말에서의 동적 리소스 제한에 적응을 통하여 실시간으로 대응할 수 있게 된다.

서버에서는 전송하고자 하는 압축 비디오에 대한 유용성 함수를 미리 생성하고 표준 기술자를 이용하여 기술한다. 이때 유용성 함수는 컨텐츠에 의존적인 것으로 유사한 컨텐츠 특성을 갖는 비디오 세그먼트(segment) 단위로 유용성 함수를 생성 제공한다. 유용성 함수 기술 데이터는 해당 압축 비디오와 함께 적응 엔진으로 전달되어 주어진 리소스에서 최적의 적응기를 선택할 수 있도록 한다.

1. 적응, 리소스, 유용성의 정의 및 매핑관계

비디오 적응의 문제는 그림 2과 같이 적응의 단위인 비디오 엔티티와 적응 공간(space) A , 리소스 공간 R , 유용성 공간 U 의 정의로 구체화된다. 여기서 공간은 각 요소들이 다 차원화될 수 있음을 나타낸 것이다. 예를 들면 주어진 k 번째 엔티티 e_k 에 대해서 리소스 공간에서 정의된 리소스 제한을 만족시키기 위한 다양한 적응방법에서 정의되는 다수의 적응기 $A = \{a_i\}$ 가 적용될 수 있고, 다양한 방

그림 2. 유용성 기반 프레임워크에서의 적응, 리소스, 유용성의 정의
Fig. 2. Definition of adaptation, resource, and utility spaces involved in video adaptation problems in the utility-based framework

법으로 측정된 그 적응 결과는 유용성 공간에서 정의된다

전술한 바와 같이 유사한 컨텐츠 특성을 유지하는 세그먼트마다 유용성 함수를 구하고 이를 바탕으로 각 세그먼트 단위로 적응을 수행함으로써 내용기반(content-based)의 적응을 가능하게 한다. 본 논문에서는 이러한 세그먼트의 분할은 미리 수행되어 하나의 비디오 시퀀스는 $V = \{e_k\}_{k=1}^K$ 로 주어진다고 가정한다.

적응 공간은 모든 가능한 적응을 나타내는 개념적 공간으로 $A = \{a_i\}_{i=1}^O$ 로 정의되고 여기서 $a_i = (a_{i1}, a_{i2}, \dots, a_{io})$ 이다. 즉, a_i 는 o 가지의 적응방법으로 구성되는 o -차원의 적응 공간에서 정의되는 적응기로 각 차원에서 정의되는 적응을 결합한 적응을 나타낸다. 본 논문에서 다루고자 하는 FD-CD를 예로들면, 적응 공간은 FD와 CD의 적응방법의 2차원으로 구성되고, 각 차원은 FD 또는 CD의 해당 적응방법에서 정의되는 몇 가지의 적응기들로 정의된다. 따라서 2차원 적응 공간에서 정의된 한 점은 FD와 CD의 적응을 결합한 한 FD-CD의 적응기를 나타내게 된다.

리소스는 대역폭, 전력, 화면크기 등 네트워크와 터미널

의 제한된 자원을 나타낸다. 일반적으로 리소스는 한 가지 이상의 종류가 제한 조건으로 주어질 수 있다. 즉, PDA와 같은 모발 장치에 비디오 스트리밍 서비스를 위한 적응 예에서 가용 대역폭뿐만 아니라 화면크기 및 전력 등의 제한 조건도 동시에 고려되어야 한다. 이러한 적응에서 고려하여야 할 p 가지의 리소스는 리소스 공간($R = \{\mathbf{r}_i\}_{i=1}^P$ 여기서 $\mathbf{r}_i = (r_{i1}, r_{i2}, \dots, r_{ip})$)에서 표현될 수 있다. 리소스 공간은 일반성을 잃지 않으면서 P 개의 이산 포인트를 포함하는 것으로 가정한다.

유틸리티는 적응된 비디오가 사용자에게 재생될 때의 품질을 나타내는 것으로 주관적인 방법(예, MOS(mean opinion scale)) 또는 객관적인 방법(예, PSNR, MSE(mean square error)) 등 다수의 측정방법으로 측정될 수 있다. 일반적으로 q 가지의 품질 측정방법에 의한 유틸리티는 q 차원의 유tility 공간($U = \{u_i\}_{i=1}^Q$, 여기서 $u_i = (u_{i1}, u_{i2}, \dots, u_{iq})$)에서 표현된다.

이제 이들 세 공간간의 적응에 따른 매핑 관계를 기술한다. 주어진 비디오 엔티티 e_k 와 정의된 적응 공간 $A = \{\mathbf{a}_i\}$ 에 대해서, 각 적응기는 $e'_k = \mathbf{a}_i(e_k)$ 와 같이 원 엔티티와 적응된 엔티니간의 매핑을 나타내는 함수가 된다. 각 적응된 엔티티는 적응 결과에 다른 리소스 값과 유tility 값은 갖고 이를 값은 각각 리소스 공간과 유tility 공간에 매핑된다. 이들 매핑관계는 식 (1)과 식 (2)로 표현된다.

$$h_j : \mathbf{a}_i(e_k) \rightarrow r_{ij}^k \text{ 또는 } r_{ij}^k = h_j(\mathbf{a}_i(e_k)), j = 1, \dots, p \quad (1)$$

$$g_j : \mathbf{a}_i(e_k) \rightarrow u_{ij}^k \text{ 또는 } u_{ij}^k = g_j(\mathbf{a}_i(e_k)), j = 1, \dots, q, \quad (2)$$

여기서 $h(\cdot)$ 과 $g(\cdot)$ 은 각각 리소스 측정기와 유tility 측정기를 나타낸다. 여기서 주목할 것은 이들 세 공간간의 매핑은 종종 multiple-to-one이 된다. 즉, 서로 다른 적응기가 동일한 리소스 값 또는 유tility 값을 줄 수 있다. 따라서 동일한 리소스 값을 주는 적응기들로부터 적응기 선택 문제가 관심의 대상이 된다.

이러한 매핑관계는 적응의 시나리오를 고려하여 더 구체화될 수 있다. 즉, 리소스 제한 $R_c \subset R$ 이 주어질 때 R 과

A , 그리고 A 와 U 간의 매핑관계는 다음과 같이 주어진다.

$$A_S = \{\mathbf{a}_i \mid h(\mathbf{a}_i(e_k)) \in R_C\} \quad (3)$$

$$U_S = \{\mathbf{u}_i \mid \mathbf{u}_i = g(\mathbf{a}_i(e_k)), \mathbf{a}_i \in A_S\} \quad (4)$$

식 (3)과 식 (4)를 이용하면 임의의 e_k 에 대해서 주어진 리소스 제한을 만족하는 적응기들의 집합 A_S 와 이에 대응하는 유tility 공간의 부공간(sub-space) U_S 가 결정되어 진다.

또 다른 형태의 적응 시나리오는 유tility 제한 $U_C \subset U$ 이 주어지는 경우(예, 화질이 30dB 이상이 되도록 적응을 하여야 한다)도 마찬가지로 각 공간간의 매핑관계는 식 (5)와 식 (6)과 같이 표현되고 U_C 를 만족하는 A_S 가 결정된다.

$$A_S = \{\mathbf{a}_i \mid g(\mathbf{a}_i(e_k)) \in U_C\} \quad (5)$$

$$R_S = \{\mathbf{r}_i \mid \mathbf{r}_i = h(\mathbf{a}_i(e_k)), \mathbf{a}_i \in A_S\} \quad (6)$$

2. 적응 문제의 정형화

유tility 기반 프레임워크에서 주어진 리소스를 만족시키면서 품질을 최대화하고자 하는 resource-constrained 시나리오의 비디오 적응문제는 다음과 같이 정형화 된다.

Formulation 1: Resource-Constrained Utility Maximization Problem
임의의 비디오 엔티티 e_k 에 대해서 주어진 리소스 제한 $R_C \subset R$ 을 만족하는 최적의 적응기 a_{opt} 는 다음 식 (7)의 해로 주어진다.

$$\begin{aligned} & \arg \max_{\mathbf{a}_i} \|\mathbf{u}_i\| = \|g(\mathbf{a}_i(e_k))\| \\ & \text{subject to } \mathbf{r}_i = h(\mathbf{a}_i(e_k)) \in R_C \end{aligned} \quad (7)$$

여기서 $\|\cdot\|$ 주어진 응용에서 정의된 다른 형태의 성분의 결합을 타나낸다.

위의 정형화는 유트리티 공간에 제한이 주어지고 리소스를 최소화하는 경우의 적응 문제로 식 (8)과 같이 쉽게 확장된다.

Formulation 2: Utility-Constrained Resource Minimization

임의의 비디오 엔터티 e_k 에 대해서 주어진 유트리티 제한 $U_C \subset U$ 을 만족하는 최적의 적응기 a_{opt} 는 다음의 식 (8)의 해로 주어진다.

$$\begin{aligned} & \arg \min_{\mathbf{a}_i} \|\mathbf{r}_i\| = \|h(\mathbf{a}_i(e_k))\| \\ & \text{subject to } \mathbf{u}_i = g(\mathbf{a}_i(e_k)) \in U_C \end{aligned} \quad (8)$$

위의 두 가지 형태로 정형화되는 적응问题是 다음의 해당 명제(proposition)로부터 해를 구할 수 있다. 이들 명제의 증명은 식 (3)과 식 (4), 식 (5)와 식(6)으로부터 바로 주어진다.

Proposition 1: 주어진 e_k 그리고 $R_C \subset R$ 에 대해서, $A_S = \{\mathbf{a}_i \mid h(\mathbf{a}_i(e_k)) \in R_C\}$ 이 공집합이 아닌 경우, 적응 엔터티 e_k 의 유트리티를 최대화 하는 최적 적응기 a_{opt} 가 A 에 존재한다.

Proposition 2: 주어진 e_k 그리고 $U_C \subset U$ 에 대해서, $A_S = \{\mathbf{a}_i \mid g(\mathbf{a}_i(e_k)) \in U_C\}$ 이 공집합이 아닌 경우, 적응 엔터티 e_k 의 리소스를 최소화하는 최적 적응기 a_{opt} 가 A 에 존재한다.

3. 유트리티 함수 표현

유트리티 기반의 접근방법에서는 적응-리소스-유트리티 공간의 분포와 그 매핑관계 정보를 적응엔진으로 전달하기 위하여 유트리티 함수를 이용하여 그 정보를 기술한다. 유

트리티 함수는 일반적으로 리소스의 함수로 미디어의 품질에 대한 사용자의 만족도를 나타내는 것으로 유트리티 기반의 네트워크 차원 할당^[20] 및 비디오 스트리밍 응용^[21]에 성공적으로 적용되었다.

주어진 e_k 와 정의된 A 에 대해서, 유트리티 함수는 식 (9)와 같은 리소스-유트리티 쌍의 집합을 리소스-유트리티 평면에서 이산 포인트로 표현한다.

$$\{(r_i, u_i) \mid r_i = h(\mathbf{a}_i), u_i = g(\mathbf{a}_i), \forall \mathbf{a}_i \in A\} \quad (9)$$

식 (9)는 리소스와 유트리티가 각각 한 종류인 경우의 식 (1)과 식 (2)의 매핑관계를 나타낸 것으로, 리소스 축을 다차원으로 확장하여 리소스의 종류가 두 가지 이상인 경우도 쉽게 수용할 수 있다. 유트리티의 경우는 유트리티 종류마다 유트리티 함수를 두어 다 차원의 유트리티를 기술하도록 확장한다.

이와 같은 방법으로 유트리티 함수는 적응-리소스-유트리티 세 공간의 분포와 각 공간 간의 매핑관계를 간결한 형태로 기술할 수 있도록 한다. 또한 리소스-유트리티의 관계를 연관된 적응기와 함께 표현함으로써 식 (7)과 식 (8)로 정형화된 적응의 문제를 해결하기 위한 탐색과 선택을 효율적으로 할 수 있도록 한다. 유트리티 함수는 어떤 특정한 경우, 즉 적응 공간이 양자화 레벨의 크기, 리소스 공간이 부호화 비트율을, 유트리티가 왜곡(MSE)을 나타내는 일차원일 경우 유트리티 함수는 R-D 모형으로 간략화 된다.

IV. FD-CD 율적응변환

이 장에서는 유트리티 기반 프레임워크를 이용한 FD와 CD 결합한 율적응변환을 기술한다. 유트리티 기반의 접근방법의 한 예로서 FD-CD를 이용한 율적응변환 문제의 정형화와 적응, 리소스, 유트리티 공간의 구체적 정의, 그리고 이에 따른 유트리티 함수의 생성을 보이고자 한다.

1. 적응, 리소스, 유트리티의 정의

비디오 엔터티 세그먼트는 편의상 그 경계가 GOP의 경계와 일치한다고 가정한다. 따라서 임의의 FD-CD 적응기는 그림 3과 같이 GOP 단위로 적용된다. 임의의 엔터티는

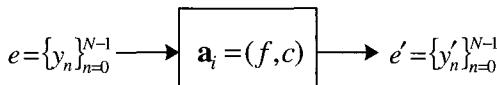


그림 3. FD-CD 적용기를 이용한 율적응변환

Fig. 3. Illustration of rate adaptation transcoding with an adaptation operator of FD-CD

$e = \{y_n\}_{n=0}^{N-1}$ 로 표현되고 여기서 y_n 은 크기 N 인 해당 GOP의 $(n+1)$ 번째 프레임이다. FD와 CD에 의해서 구성되는 2 차원의 적응 공간은 $A = \{a_i = (f, c)\}$ 로 정의 된다. FD와 CD는 각각 고정 수의 적응기로 구성되는데, FD의 특정 적응기는 식 (10)과 같은 frame-dropping factor f 로 표현한다.

$$f = \frac{F_{org}}{F'} \quad (10)$$

여기서 F_{org} 와 F' 는 각각 입력 압축 비디오의 프레임을과 적응 후의 프레임율을 나타낸다. CD 적응동작을 표현하기 위해서 식 (11)의 DCT 계수 제거의 결과로 얻어지는 비트율 감소 비율(rate-reduction ratio) c 를 사용한다.

$$c = 1 - \frac{R'}{R_{org}} \quad (11)$$

여기서 R_{org} 와 R' 는 각각 적응 전의 입력 비디오의 비트율과 적응 후의 비트율을 나타낸다. 결과적으로 $a_i = (f, c)$ 는 그림 4와 같이 먼저 GOP의 첫번째 프레임에 c 로 규정된 CD를 적용하고, 연속된 $(f-1)$ 의 프레임을 제거하고, 이러한 기본 적응패턴을 GOP의 마지막 프레임에 적용하는 것이다.

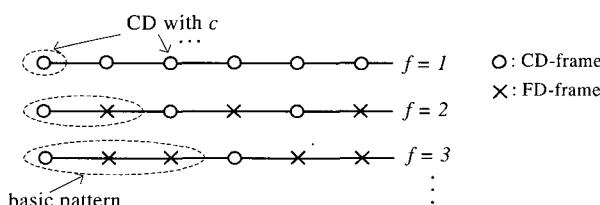
그림 4. $a_i = (f, c)$ 의 율적응변환

Fig. 4. Illustration of an adaptation operation of $a_i = (f, c)$

임까지 수행하는 적응동작을 규정한다. 이러한 FD-CD의 정의는 UMA 환경에서의 실시간 적응을 위하여 다음과 같은 간략화된 적응으로 제한한다. 즉, CD는 한 적응단위인 GOP 내에서 프레임마다 동일한 c 값을 적용하는(uniform CD) 것으로 가정하여 정의된 FD와 CD의 적용 순서에 상관없이 동일한 결과를 얻도록 하고, FD는 예측 부호화의 기준 프레임이 아닌 프레임(예, B-프레임)만을 제거하는 규칙적인 프레임 제거를 가정한다.

FD-CD 적용은 실시간 구현에 적합하도록 처리가 간단할 뿐만 아니라 UMA 환경에 적합한 다음의 장점을 갖는다. FD는 비트율을 조절할 수 있는 최소의 단위가 한 프레임의 비트량으로 정확하게 주어진 목표비트율을 맞출 수 없다. 반면 CD는 제거되는 계수를 조절함으로써 훨씬 세밀한 수준에서 비트율을 감소할 수 있다. 또한, 이종망의 UMA 환경에서는 대역폭이 넓은 범위에서 가변적이며 따라서 FD 또는 CD의 한 가지 적용방법만으로는 필요한 유품질을 얻을 수 없는 경우가 있다. 따라서 두 적용방법을 결합한 FD-CD는 더 넓은 범위의 대역폭 변화까지 수용할 수 있다. 동시에 시간영역과 공간영역의 적응을 결합함으로써 유트리티 함수를 기반으로 체계적인 시·공간 화질의 최적 trade-off를 얻을 수 있다.

율적응변환에서 리소스 공간은 비디오 전송을 위한 사용 대역폭이 된다. 주어진 e 와 a_i 에 대해서, 율측정기 $h(\cdot)$ 는 $a_i(e)$ 를 리소스 공간 R 의 한 점 $r_i = h(a_i(e))$ 로 매핑한다. 유트리티 공간은 $e' = a_i(e)$ 의 품질 측정으로 PSNR을 사용한다. 이는 $a_i(e)$ 와 U 의 한 점 u_i 와의 매핑을 규정한다.

2. 율적응 문제

율적응 문제는 Formulation 1의 특정한 경우로 식 (12)와 같이 정형화된다.

$$\begin{aligned} & \arg \max_{a_i} PSNR(a_i(e)) = \arg \min D(f, c) \\ & \text{subject to } r_i = h(a_i) \leq R_T \end{aligned} \quad (12)$$

여기서 $D(f, c)$ 는 $a_i = (f, c)$ 의 적응에 의하여 야기되는 전체 왜곡이고 R_T 는 목표 비트율이다. $[0, N-1]$ 프레임 구간의 GOP에 걸친 전체 왜곡은 다음과 같이 표현된다.

$$D_{[0,N-1]}(f,c) = \sum_{n=0}^{N-1} \|y_n - y'_n\| = \sum_{n=0}^{N/f-1} \left[D_{cd}(nf, c) + \sum_{m=1}^f D_{fd}(nf+m) \right] \quad (13)$$

여기서 $\|\cdot\|$ 는 MSE를 나타낸다. 전체 왜곡은 식 (13)과 같이 프레임 제거에 의한 FD 왜곡 D_{fd} 와 DCT 계수 제거에 의한 CD 왜곡 D_{cd} 의 합이다. FD 왜곡을 모형화하기 위해 서 제거된 프레임은 복호화기에서 바로 이전 프레임이 재생되는 zero-order hold 보간을 가정한다. 이러한 가정하에 $(nf+m)$ 번째 프레임의 제거에 의한 왜곡은 다음과 같이 주어진다.

$$D_{fd}(nf+m) = \|y_{nf+m} - y'_{nf}\|. \quad (14)$$

nf 번째 프레임의 c 에 의한 CD 왜곡은 다음과 같다.

$$D_{cd}(nf, c) = \|y_{nf} - y'_{nf}\|. \quad (15)$$

식 (12)의 최적화 문제는 다음의 절차에 따라서 해를 구한다.

- ▶ Step 1: 한 특정 FD 적응 $f \in \{f_i\}$ 에 대하여 전체 FD 왜곡과 주어진 R_T 를 만족하기 위한 식 (16)의 c 를 구한다.

$$c = 1 - \frac{R'}{R_{org}} = 1 - \frac{R_T 1/F_{org} N/f}{\sum_{n=0}^{N/f-1} B_{nf}} \quad (16)$$

여기서 B_{nf} 는 nf 번째 프레임의 비트량이다.

- ▶ Step 2: CD가 적용되는 각 프레임에 대해서 최소 CD 왜곡을 계산한다. 최소 CD 왜곡은 식 (17)과 같이 한 프레임내의 각 블록마다 제거되는 DCT 계수를 주어진 c 를 만족하면서 최적으로 분포시킴으로써 얻을 수 있다.

$$\begin{aligned} & \min \|y - y'\| \\ & \text{subject to } B' \leq B_T = (1 - c)B \end{aligned} \quad (17)$$

여기서 B_T 는 해당 프레임이 만족시켜야 할 비트량이고 B

는 c 로 적용 후의 비트량을 각각 나타낸다. 본 논문에서 가정한 uniform CD에서의 최소 전 CD 왜곡은 각 프레임의 최소 왜곡의 합이 된다. 선행연구^[16]에서 Langrange 기법을 도입한 한 프레임내에서 최적 CD 기법을 제시하였다.

- ▶ Step 3: 정의된 모든 FD 적응 $f \in \{f_i\}$ 에 대하여 위의 단계 1과 2를 반복하고, 최종적으로 최소의 왜곡을 갖는 적응기를 최적 적응기 $\mathbf{a}_{op} = (f_{op}, c_{op})$ 로 선택한다.

위의 절차가 주어진 리소스에서의 최적 적응기를 구하는 데 사용될 수 있지만 proxy에서의 실시간 응용에는 적용할 수 없다.

유틸리티 기반의 접근방법에서는 위의 절차에 따라서 하나의 리소스 값에서의 최적 적응기를 구하기보다 서버에서 예상되는 넓은 범위의 리소스에 대한 적응기와 그 유틸리티를 미리 계산하여 유틸리티 함수로 기술하고, 이를 proxy로 전달한다.

3. 유틸리티 함수 생성

유tility 함수는 비트율의 전 범위에서 몇 가지의 표본화된 비트율 값에 대하여 III-2에서 기술한 방법으로 구할 수 있다. 또는 미리 정의된 적응 공간 $A = \{\mathbf{a}_i | \mathbf{a}_i = (f, c)\}$ 에 대한 식 (19)과 같은 유tility-리소스 쌍의 계산을 통하여 구할 수도 있다.

$$\{(r_i, u_i) | r_i = PSNR(\mathbf{a}_i(e)), u_i = rate(\mathbf{a}_i(e)), \forall \mathbf{a}_i \in A\} \quad (19)$$

그림 5는 한 예로 MPEG-4로 부호화된 “Coastguard”的 임의의 한 GOP에 대해서 생성한 유tility 함수를 보인 것이다. 리소스와 유tility를 비트율과 PSNR로 정의함에 따라 R-D 커브와 유사한 형태의 커브들로 구성된다. PSNR 계산에서는 제거된 프레임은 이전 프레임이 재생된다는 가정을 하였다.

이 예제에서는 $f \in \{1, 3, 1, 5\}$ 의 FD와 $c \in \{0.0, 0.1, 0.2, 0.3, 0.4, 0.5\}$ 의 CD를 결합한 18 개의 FD-CD 적응기를 정의한다. 따라서 리소스-유tility 평면에 각 적응기에 해당하는 18개의 점이 존재한다. 주목할 것은 동일한 FD 적응을 갖는 이웃한 적응기 사이를 선형 균사화를 할 수 있다는 것이다. 이는 정의된 CD 적응 사이에 FD와는 달리 보간이 가능하기 때문이다. 따라서 동일한 f 와

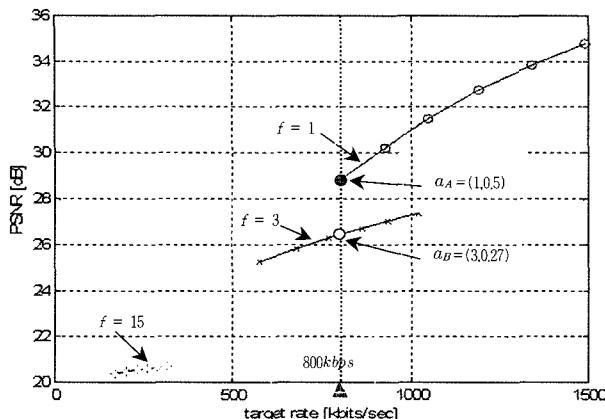


그림 5. 유틸리티 함수의 예(GOP 크기 N=15, sub-group 크기 M=3)로 1.5Mbps로 부호화된 MPEG-4 시퀀스 Coastguard의 FD-CD를 적용, 여기서 최적 CD 기법으로 블록 수준에서 memoryless 기법^[8]을 사용하였음)
Fig. 5. An example of utility function using FD-CD adaptations on the Coastguard coded at 1.5Mbps with GOP (N=15, M=3). Here the memoryless optimization on block level has been applied as an optimized CD

$c \in [0, 0.5]$ 의 다른 CD 적응을 갖는 적응기들은 하나의 R-D 커브 형태를 갖는다. 유틸리티 생성은 실제로 제한된 수의 리소스 값만 고려하지만 넓은 범위의 리소스에 걸쳐 세밀한 수준에서 그 값을 만족하는 적응기와 해당 유틸리티 값을 표현할 수 있다.

이 예제에서 미리 정의된 $A=\{a_i\}$ 는 200kbps까지의 울감소를 수용하고 있음을 알 수 있다. 주어진 리소스 값을 만족시키는 모든 가능한 적응기들이 기술되어 있고, 각 적응 엔진의 적응선택규칙에 따라서 적절한 적응기가 선택되어 진다. 예로 최적 품질을 갖는 적응기를 선택할 경우, 목표 비트율 800kbps에 대해서 적용 가능한 적응기 $a_A = (1, 0.5)$ 와 $a_B = (3, 0.27)$ 중에서 a_A 가 PSNR 기준으로 최적 적응기로 선택된다.

V. 유틸리티 함수 기술(Description)

이종망과 다양한 사용자 단말을 이용한 미디어(media) 접근을 고려하는 측면에서 MPEG-21과 UMA의 목적은 서로 잘 부합되고, 실제로 MPEG-21 멀티미디어 프레임워크의 대표적인 응용 서비스가 UMA라고 할 수 있다. 특히, MPEG-21 DIA는 다양한 형태의 망과 사용자 단말에 구애 받지 않고 투명하게 미디어를 접근하기 위한 사용환경 기술자 및 미디어 적용을 위한 미디어 기술자들의 표준 툴을 정의하고 있다. 이 장에서는 유tility 함수로 표현되는 정보를 어떻게 제안 채택된 표준 툴을 사용하여 XML 형태로 기술하는지를 보인다.

1. 유tility 함수 기술

제안된 툴은 *UtilityFunctionType*을 중심으로 MPEG-21 DIA의 터미널과 네트워크 QoS를 위한 툴, 즉 네트워크와 터미널의 제한 조건 하에서 QoS 지원을 위한 위한 적응의 문제를 다룬 *AdaptationQoSType*의 일부로 채택되었으며, 그림 6는 이들 XML 스키마(schema)를 보여준 것이다. 그림 5에 예시된 유tility 함수는 *UtilityFunctionType*에 정의된 기술자들에 의해 다음과 같이 기술된다. *Constraint*는 리소스 공간의 관심 영역에서의 표본들을 기술하고 *AdaptationOperator*는 표본화된 각 리소스 값을 만족시킬 수 있는 하나 또는 그 이상의 적응기들을 기술하고 *Utility*는 각 해당 적응기가 적용된 후의 미디어 품질을 기술한다. 여기서 한 가지 이상의 리소스 및 유tility를 수용할 수 있도록 *Constraint* 및 *Utility*의 occurrence를 복수로 규정하였다. 이와 같이하여, *UtilityFunctionType*은 리소스의 고정 수의 이산 표본값을 인덱스로 이용하여 허

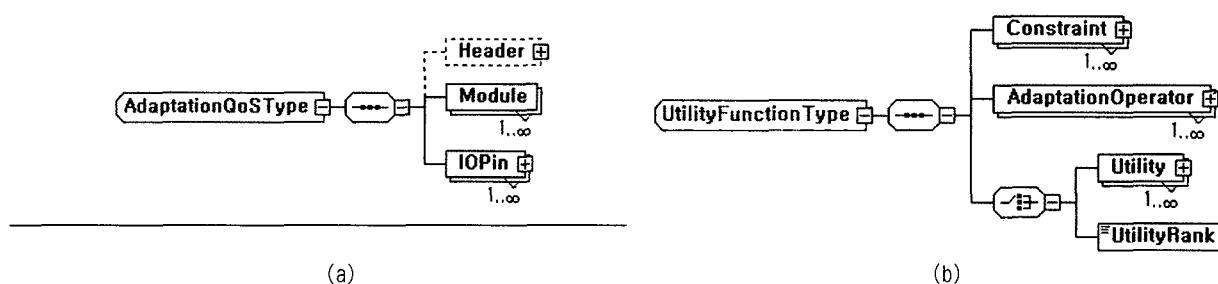


그림 6. XML 스키마 (a) *AdaptationQoSType* (b) *UtilityFunctionType*
Fig. 6. Schema diagrams of the descriptors of (a) *AdaptationQoSType* and (b) *UtilityFunctionType*

용되는 적응기. 이에 의한 결과 유트리티를 나열식으로 기술한다.

전술한 AdaptationOperator, Constraint, Utility의 정확한 semantics는 미리 정의된 Classification Scheme을 IOPin을 이용하여 참조함으로써 정의된다. 그리고 Header는 참조의 편의를 위하여 Classification Scheme의 alias를 제공한다. 현재 표준에는 최대한 많은 적응의 경우를 수용하기 위하여 본 논문의 접근방법에 기반하여 제안한 다양한 종류의 adaptationoperator, utility, constraint를 Classification Scheme에 포함하고 있다. 특히, Wavelet reduction, FGS 적응, 공간해상도 적응 등의 적응방법을 수용할 수 있는 각

적용방법에 대한 적응기들을 제시하였다. Adaptation-QoSType은 그림 6에서와 같이 하나 또는 다수의 *Module*을 포함할 수 있으며, 각 *Module*은 기술하고자 하는 정보의 특성에 따라서 나열식, 매트릭스 또는 함수 형태로 선택적으로 조합하거나 단독으로 사용할 수 있다

다음 기술 예는 UtilityFunctionType을 이용하여 그림 5의 유트리티 함수를 기술한 AdaptationQoS의 생성 예이다. 여기서 FD-CD가 주어진 bandwidth의 constraint를 만족하기 위한 위한 비트율 적응의 적응기로 사용되었고, PSNR이 유트리티 측정기로 사용되었다. 그리고 이들의 semantics는 IOPins으로 정의된다.

```

<Description xsi:type = "AdaptationQoSType">
  <Header>
    <ClassificationAlias alias = "AQoS" href = "urn:mpeg:mpeg21:2003:01-DIA-AdaptationQoS-NS"/>
  </Header>
  <Module xsi:type = "UtilityFunctionType">
    <Constraint iOPinRef = "BANDWIDTH">
      <Values xsi:type = "IntegerVectorType">
        <Vector>1510 1359 1200 1200 1071 1071 1071 941 814 814
          814 1296 1000 1000 842 744 909 712 600 396
          359 331 293 255 217</Vector>
      </Values>
    </Constraint>
    <AdaptationOperator iOPinRef = "B_FRAMES">
      <Values xsi:type = "IntegerVectorType">
        <Vector>0 0 1 0 1 2 0 0 1 2 1 0 1 2 1
          1 2 2 2 2 2 2 2 2</Vector>
      </Values>
    </AdaptationOperator>
    <AdaptationOperator iOPinRef = "P_FRAMES">
      <Values xsi:type = "IntegerVectorType">
        <Vector>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
          0 0 0 4 4 4 4 4</Vector>
      </Values>
    </AdaptationOperator>
    <AdaptationOperator iOPinRef = "COEFF_DROPPING">
      <Values xsi:type = "FloatVectorType">
        <Vector>0.0 0.1 0.21 0.09 0.3 0.2 0.0 0.4 0.5
          0.44 0.31 0.0 0.35 0.270.08 0.4 0.5
          0.2 0.4 0.5 0.0 0.1 0.2 0.3 0.4 0.5</Vector>
      </Values>
    </AdaptationOperator>
    <Utility iOPinRef = "PSNR">
      <Values xsi:type = "FloatVectorType">
        <Vector>34.47 33.56 32.48 31.40 31.58 30.82 28.62
          30.27 29.10 28.57 27.53 31.94 30.69 30.15
          28.33 29.04 28.11 28.01 27.03 26.49 23.44
          23.36 23.29 23.18 23.02 22.87</Vector>
      </Values>
    </Utility>
  </Module>
  <IOPin semantics = "AQoS:1.1.1" id = "BANDWIDTH" input = "true" output = "false"/>
  <IOPin semantics = "AQoS:2.1" id = "PSNR" input = "false" output = "true"/>
  <IOPin semantics = "AQoS:3.1.1" id = "B_FRAMES" input = "false" output = "true"/>
  <IOPin semantics = "AQoS:3.1.2" id = "P_FRAMES" input = "false" output = "true"/>
  <IOPin semantics = "AQoS:3.1.3" id = "COEFF_DROPPING" input = "false" output = "true"/>
</Description>
</DIA>
```

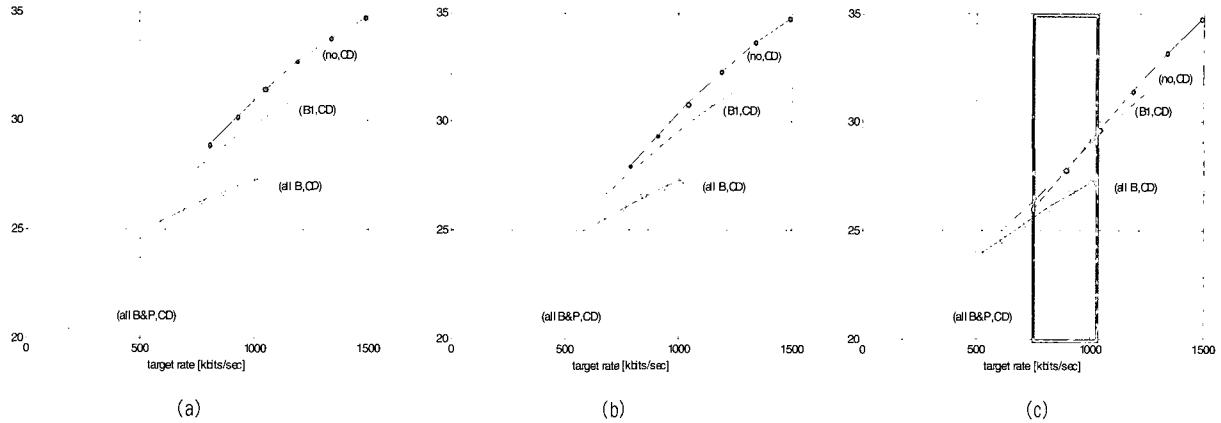


그림 7. 다른 CD 구현방법을 포함하는 유저리티 함수 (a) block 수준의 memoryless 최적 CD (b) MB 수준의 memoryless 최적 CD (b) rate-based CD
Fig. 7. Utility functions resulting from different implementations of CD: (a) memoryless optimization on block level (b) memoryless optimization on MB level (b) rate-based

Constraint-AdaptationOperator-Utility 쌍이 서로 연관지어지도록 순서대로 기술한다. 즉, 주어진 Constraint 값과 이를 만족하는 AdaptationOperator, 적용 결과 Utility 가기술 순서대로 하나의 쌍이된다. 이를 위해서 동일한 Constraint 값에 대한 다수의 AdaptationOperator가 존재할 경우 그 적용기 수 만큼 Constraint 값을 반복하여 기술함으로써 이러한 연관 관계가 구성되도록 한다.

2. 적응기 정의의 모호성

일부 적응방법에서 정의된 적응기는 구현방법에 따라서 서로 다른 리소스 및 유저리티 결과를 줄 수 있다. 이는 적응기의 정의(specification)의 모호성(ambiguity)에 기인한다. 예를들면, CD를 rate-reduction ratio c 로 규정하는 경우, 예로 $c = 0.2$ 는 DCT 계수 제거에 의해서 주어진 프레임의 비트량을 20% 줄이는 것을 의미하지만, 제거되어야 할 계수를 명확히 규정하지는 않는다. 따라서 이 적응기는 적응엔진마다 구현방법에 따라서 유저리티 함수에서 기술한 값과 조금씩 다른 유저리티 값의 결과를 줄 수 있다. 한편 JPEG-2000나 MPEG-4 FGS 등의 scalable 부호화의 경우 모호성없이 비트열의 일부를 제거할 수 있음을 알 수 있다.

이러한 모호성의 문제를 다루기 위해서, 본 논문의 기술자에 유저리티 순위(utility ranking)의 개념을 도입하였다. 이는 임의의 리소스를 만족하는 적응기들 사이에서 구현에 따라서 유저리티 값 자체 보다는 유저리티 순위가 더 일관

성을 유지할 가능성이 크므로 모호성이 존재하는 적응방법의 경우 적응기의 유저리티 순위를 기술할 수 있도록 한 것이다. 더구나 실제 응용에서 다수의 사용 적응기들로부터 적응기를 선택함에 있어서 유저리티 값 보다는 그 순위가 더 유용한 정보가 될 수 있다. 따라서 그림 6에서처럼 UtilityFunctionType은 Utility 또는 UtilityRank로 유저리티를 기술할 수 있도록 한다.

그림 7은 UtilityRank의 유용성을 확인하기 위한 것으로 동일한 비디오에 대한 3 가지 다른 CD 구현방법^[16]을 사용한 유저리티 함수를 보여준다. 실험 조건은 그림 5의 유저리티 함수 생성과 동일하다. CD 구현방법에 따른 유저리티 함수의 변화를 확인할 수 있다. 하지만 대부분의 비트율 범위에서(그림 7(c)에 표시된 박스 부분 제외) 주어진 비트율에서 유저리티 순위는 동일하게 유지됨을 알 수 있다. 순위가 변하는 경우에서도 가장 낮은 PSNR 값을 갖는 적응기를 제외하고는 적응기의 순위가 유지되고 있다. 따라서 각 적응기의 유저리티 순위를 기술하여 구현에 따른 모호성이 존재하는 경우에도 유저리티 기반의 적응기 선택을 지원할 수 있도록 한다.

VI. 프로토타입 적응률 구현

본 논문에서는 유저리티 기반 프레임워크를 FD-CD를 이용한 MPEG-4 비디오의 율적응변환에 적용하고 이 적응률의

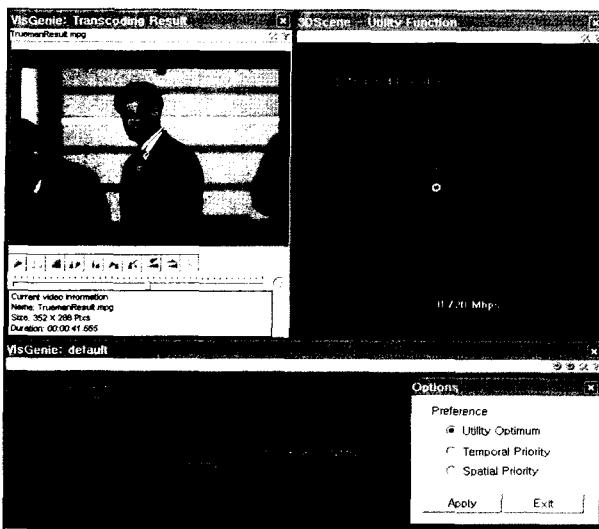


그림 8. 유틸리티 기반의 윤적응변환 툴의 GUI

Fig. 8. User interface of the implemented adaptation tool

프로토타입 구현을 통하여 그 타당성과 효율성을 검증하였다. 본 적응툴은 파서(parser), 적응엔진, 그리고 사용자인터페이스 (GUI)로 구성된다. 파서는 전송되는 압축 비디오의 적응 엔티티에 해당하는 AdaptationQoS 기술자를 파싱하고 유틸리티 함수를 재구성한다. 적응엔진은 적응결정 (Adaptation Decision Taking : ADT) 모듈과 미디어적응 모듈로 구성되는데, ADT 모듈은 재구성된 유tility 함수 정보를 기반으로 적응선택규칙에 따라서 주어진 목표율을 만족하는 적응기를 선택해서 미디어적응 모듈로 전달한다. 적응선택규칙은 기본적으로 유tility의 품질 기준뿐만 아니라 시공간 품질에 대한 사용자 선호도를 입력하여 적응기 결정에 반영한다. 미디어적응 모듈은 입력 MPEG-4 비트열을 연속된 엔티티 단위로 선택된 FD-CD 적응기를 적용하여 실시간 윤적응을 수행한다. 그림 8의 GUI는 적응과정을 확인하기 위한 것으로, 왼쪽 상단의 비디오 창에서는 적응된 비디오의 화질을 확인하고, 상단 오른쪽 창은 재구성된 유tility 함수와 하단의 시변 대역폭에 적응하기 위한 최적 적응기의 선택과정을 도시한다. 하단 오른쪽의 사용자선호도 입력창은 사용자가 PSNR에 의한 최적 적응기의 자동 선택 대신에 공간 또는 시간 품질의 사용자 선호도에 따라서 적응기를 선택할 수 있도록 한다.

본 적응 툴을 이용한 실험에서 다수의 MPEG-4 비디오 시퀀스에서 priori probability에 의한 기존의 방법(목표비트율에서 유tility 함수를 사용하지 않고 주어진 다양

한 컨텐츠에 대해서 평균적으로 가장 많이 선택된 최적의 적응기를 사용)과 주어진 유tility 함수에서 제시되는 최적 적응기를 사용한 경우의 비교에서 후자의 유tility 기반의 적응이 객관적 및 주관적 화질에서 우수함을 보였다 (0.83 cB 개선과 시공간 화질 trade-off에 의한 주관적 화질 개선).

VII. 결 론

본 논문에서는 UMA 환경에서 다양한 자원 조건을 만족하기 위한 비디오 적응에 대한 일반적인 방법론을 제시하였다. 본 방법론의 유tility 기반 프레임워크는 적응 과정에 관련된 주요 개념인 적응, 리소스, 유tility와 이들의 관계를 유tility 함수를 이용하여 명시적으로 모형화 한다. 유tility 함수는 R-D 모형과 유사하지만 다양한 리소스와 유tility 및 적응의 결합을 수용할 수 있는 유연성을 제공한다. 유tility 기반의 접근방법을 이용하여, 본 논문에서는 UMA 환경에서의 가변 대역폭에 실시간으로 대응할 수 있는 윤적응변환으로 DCT 계수 제거와 프레임 제거를 결합한 시공간 적응을 제시하였다.

또한 본 논문의 유tility 기반의 적응 프레임워크를 상호연동 가능한 형태로 적용하기 위하여 기술하기 위하여 MPEG-21 DIA에 제안 채택된 유tility 함수 기술자를 소개하고 주요 설계 배경을 기술하였다.

유tility 기반의 적응 프레임워크의 한 실제적인 예로서 MPEG-4 비디오의 비트율 적응을 위한 FD-CD 윤적응변환 툴의 구현을 통하여 그 타당성과 효율성을 보였다. 본 논문의 적응 프레임워크 및 기술자는 다양한 비디오 적응 방법도 수용할 수 있는 일반적인 것으로, scalable 비디오에도 효과적으로 적용할 수 있을 것이다.

참 고 문 헌

- [1] R. Mohan, J. R. Smith, and C.-S. Liu, "Adapting multimedia Internet content for universal access," *IEEE Trans. Multimedia*, vol. 1., pp. 104-114, Mar. 1999.
- [2] P. van Beek, J. R. Smith, T. Ebrahimi, T. Suzuki, and J. Askelof, "Metadata-driven multimedia access", *IEEE Signal Processing Mag.*, vol. 20, pp. 40-52, Mar. 2003.
- [3] J. Bormans, J. Gelissen, and A. Perkis, "MPEG-21: The 21st

- century multimedia framework." *IEEE Signal Processing Mag.*, vol. 20, pp. 53-62, Mar. 2003.
- [4] A. Vetro, C. Christopoulos, and H. Sun, "Video transcoding architectures and techniques: An overview", *IEEE Signal Processing Mag.*, vol. 20, pp. 18-29, Mar. 2003.
- [5] Y. Nakajima, H. Hori, and T. Kanoh, "Rate conversion of MPEG coded video by requantization process," in *Proc. IEEE Int. Conf. Image Processing*, Washington, DC, 1995, pp. 408-411.
- [6] H. Sun, W. Kwok, and J. Zdepski, "Architecture for MPEG compressed bitstream scaling", *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, pp. 191-199, Apr. 1996.
- [7] P. Assuncano and M. Ghanbari, "Post-processing of MPEG2 coded video for transmission at lower bit rates," in *Proc. IEEE Int. Conf. On Acoustic, Speech, and Signal Processing'96*, vol. 4. Atlanta, GA, May 1996, pp. 1998-2001.
- [8] A. Elefteriadis, "Dynamic rate shaping of compressed digital video," Ph. D. dissertation, Dept., Elec. Eng., Columbia Univ., New York, June 1995.
- [9] J. Youn, M. T. Sun, and C. W. Lin, "Motion vector refinement for high performance transcoding," *IEEE Multimedia*, vol. 1, pp. 30-40, Mar. 1999.
- [10] T. Shanableh and M. Ghanbari, "Heterogeneous video transcoding to lower spatio-temporal resolutions and different encoding formats," *IEEE Trans. Multimedia*, vol. 2, pp. 101-110, June 2000.
- [11] K. T. Fung, Y. L. Chan, and W. C. Siu, "New architecture for dynamic frame-skipping transcoder." *IEEE Trans. Image Processing*, vol. 11, pp. 886-900, Aug. 2002.
- [12] A. Vetro, H. Sun, and Y. Wang, "Object-based transcoding for adaptive video content delivery," *IEEE Trans. Circuit Syst. Video Technol.*, vol. 11, pp. 387-401, Mar., 2001.
- [13] W. Li, "Overview of fine granularity scalability in MPEG-4 video standard," *IEEE Trans. Circuit Syst. Video Technol.*, vol. 11, pp. 301-317, Mar. 2001.
- [14] M. van der Schaar and H. Radha, "A hybrid temporal-SNR fine granular scalability for internet video," *IEEE Trans. Circuit Syst. Video Technol.*, vol. 11, pp. 318-331, Mar. 2001.
- [15] S.-F. Chang, "Optimal video adaptation and skimming using a utility-based framework," in *Proc. IWDC-2002*, Capri Island, Italy, Sep. 2002.
- [16] J.-G. Kim, Y. Wang, and S.-F. Chang, "Content-adaptive utility-based video adaptation", In *Proc. ICME-2003*, Baltimore, July 2003.
- [17] Y. Wang, J.-G. Kim, and S.-F. Chang, "Content-based utility function prediction for real-time MPEG-4 transcoding", In *Proc. ICIP-2003*, Barcelona, Sep. 2003.
- [18] A. Vetro, Y. Wang, and H. Sun, "Rate-distortion optimized video coding considering frameskip," in *Proc. IEEE Int. Conf. Image Processing*, Thessaloniki, Greece, Oct. 2001, pp. 534-537.
- [19] E.C. Reed and J.S. Lim, "Optimal multidimensional bit-rate control for video communications," *IEEE Trans Image Processing*, vol. 11, no. 8, pp. 873-885, Aug. 2002.
- [20] P. Bocheck, A. Campbell, S.-F. Chang and R. R.-F. Liao, "Utility-based network adaptation for MPEG-4 systems", in *Proc. NOSSDAV '99*, Basking Ridge, New Jersey, June 1999.
- [21] C. E. Luna, L. P. Kondi, and A. K. Katsaggelos, "Maximizing user utility in video streaming applications", *IEEE Tran. Circuits and Syst. Video Technol.*, vol 13, no 2, pp. 141-148, Feb. 2003.
- [22] J.-G. Kim, Y. Wang, S.-F. Chang, K. Kang, and J. Kim, "Description of utility function based optimum transcoding," ISO/IEC JTC1/SC29/WG11/M8319, Fairfax May 2002.
- [23] ISO/IEC 21000-7 FDIS Part 7: *Digital Item Adaptation*, ISO/IEC JTC1/SC29/WG11/N6168, Hawaii, USA, Dec. 2003.

저자소개

김재곤

- 1990년 2월: 경북대학교 전자공학과(학사)
- 1992년 2월: 한국과학기술원 전기 및 전자공학과(석사)
- 1992년 3월 ~ 현재: 한국전자통신연구원 방송미디어연구그룹(선임연구원)
- 주관심분야: 비디오신호처리, 멀티미디어 프레임워크, 디지털방송,

김형명

- 1974년 2월 서울대학교 공과대학(학사)
- 1982년 4월 미국 Pittsburgh 대학 전기공학과(석사)
- 1985년 12월 미국 Pittsburgh 대학 전기공학과(박사)
- 1986년 9월 현재 한국과학기술원 전기 및 전자공학과 교수
- 주관심분야: 이동통신 기술, 디지털신호와 영상처리, 다차원 시스템 이론, 다중사용자 검파기

강경옥

- 1985년 2월: 부산대학교 물리학과(학사)
- 1988년 2월: 부산대학교 물리학과(석사)
- 2003년 2월 한국항공대학교 전자공학과(박사)
- 1993년 3월 ~ 현재: 한국전자통신연구원(선임연구원/방송미디어연구그룹 팀장)
- 주관심분야: 디지털방송, 메타데이터신호처리, 음향신호처리, 객체기반3차원 오디오 신호처리

김진웅

- 1981년 2월: 서울대학교 공과대학 전자공학과(학사)
- 1983년 2월: 서울대학교 대학원 전자공학과(석사)
- 1993년 8월: Texas A&M Univ. 전기전자공학과(박사)
- 1983년 3월 ~ 현재: 한국전자통신연구원(책임연구원/방송미디어연구그룹장)
- 주관심분야: 영상통신, 멀티미디어 방송