An Optimal Framework of Video Adaptation and Its Application to Rate Adaptation Transcoding

Jae-Gon Kim, Yong Wang, Shih-Fu Chang, and Hyung-Myung Kim

The adaptation of video according to the heterogeneous and dynamic resource constraints on networks and devices, as well as on user preferences, is a promising approach for universal access and consumption of video content. For optimal adaptation that satisfies the constraints while maximizing the utility that results from the adapted video, it is necessary to devise a systematic way of selecting an appropriate adaptation operation among multiple feasible choices. This paper presents a general conceptual framework that allows the formulation of various adaptations as constrained optimization problems by modeling the relations among feasible adaptation operations, constraints, and utilities. In particular, we present the feasibility of the framework by applying it to a use case of rate adaptation of MPEG-4 video with an explicit modeling of adaptation employing a combination of frame dropping and discrete cosine transform coefficient dropping, constraint, utility, and their mapping relations. Furthermore, we provide a description tool that describes the adaptation-constraint-utility relations as a functional form referred to as a utility function, which has been accepted as a part of the terminal and network quality of service tool in MPEG-21 Digital Item Adaptation (DIA).

Keywords: Video rate adaptation, universal multimedia access (UMA), MPEG-21 DIA, utility function.

I. Introduction

Universal multimedia access (UMA) [1]-[3], which refers to the access and consumption of multimedia content over heterogeneous networks by using diverse terminals (TVs, PCs, PDAs, cellular phones, and so on) in a seamless and transparent way, has been recognized as an essential application in a convergent environment. In order to support UMA, video adaptation that transforms video content into an appropriate form is a promising approach for satisfying the constraints imposed by the usage environments of heterogeneous networks and/or diverse terminals, as well as those imposed by user preferences.

Media content adaptation is performed in many different ways: adaptation of spatial resolution, spatial quality, temporal frame rate, sequence duration, and so on. In particular, bit-rate adaptation to network bandwidth variations due to heterogeneous access technologies or due to dynamic changes in network conditions is a pretty common case required in UMA environments. The rate adaptation transcoding on which we focus in this paper converts a pre-encoded video stream into a different one with a different bit-rate. This allows for avoiding the storage of videos with the same content in several bit-rates.

Many methods of rate adaptation exist [4], which can be categorized as follows: i) spatial domain transcoding (for example, requantization of discrete cosine transform (DCT) coefficients [5]-[7] and DCT coefficients dropping [8]), ii) temporal domain transcoding (for example, frame dropping) [9]-[11], and iii) object-based transcoding [12] (for example, video object prioritization and dropping). On the other hand, in the case of a scalable stream, the fine granular scalability (FGS) and some of its variant forms that have been adopted as new

---

Manuscript received Jan. 31, 2005; revised June 11, 2005.

Jae-Gon Kim (phone: +82 42 860 4980, email: jgkim@etri.re.kr) is with Digital Broadcasting Division, ETRI, Daejeon, Korea.

Yong Wang (email: ywang@ee.columbia.edu) and Shih-Fu Chang (email: sfchang@ee.columbia.edu) are with Department of Electrical Engineering, Columbia University, New York, USA.

Hyung-Myung Kim (email: hmkim@csplab.kaist.ac.kr) is with Department of Electrical Engineering & Computer Science, KAIST, Daejeon, Korea.

scalable coding tools in MPEG-4 also enable the dynamic rate adaptation to time-varying bandwidth by simply truncating a certain number of bit-planes [13], [14].

In this section, we briefly review the prior work on bit-rate adaptation transcoding, where the focus has been centered on the trade-off between complexity and quality. We assume readers are familiar with the MPEG compression algorithm, which is the most popular encoding method [15]. The most straightforward way is to use the cascade of a decoder and an encoder, commonly known as re-encoding. The incoming video stream is decoded in the pixel domain and the decoded video frame is re-encoded at the desired bit-rate matching the bandwidth constraint. This involves a high complexity, while it can achieve the best performance in terms of quality.

Requantization partially decodes the incoming video stream to form the DCT coefficients and then requantizes the coefficients with coarser step sizes in an open-loop architecture. It significantly reduces the processing complexity by simplifying the architecture of re-encoding. However, the problem with this approach is error drift caused by a reference frame mismatch, which often results in unacceptable quality. Thus, several techniques for eliminating drift degradation have been proposed [5]-[7].

An alternative to requantization, which is called DCT coefficient dropping (CD) or dynamic rate shaping (DRS), is to directly cut high-frequency coefficients from each macroblock (MB) without requantization-like processing [8]. Frame dropping (FD) is a temporal domain technique to reduce the bit-rate by reducing temporal resolution with frame skipping. When a reference frame has been dropped, it is necessary to re-compute the residual signal with a previous nonskipped frame as a new reference frame. Some solutions to this problem with less computation have been proposed recently. For instance, [9] and [10] made use of motion vector refinement schemes to reuse the existing motion vectors in the incoming stream, and [11] proposed a way to compute this new residual in the DCT domain.

However, most existing works concentrate on optimization of a pre-selected adaptation operation, rather than on systematic solutions in choosing the optimal adaptation operation. In other words, the main focus has been on how to reduce the complexity by simplifying the architecture of re-encoding, while preserving the quality as much as possible by reducing drift degradation. There is no sufficient work on a transcoding strategy for finding the optimal adaptation operation among multiple options in spatial, temporal domains, or their combinations.

With respect to transcoding strategy, there exists an ad-hoc control scheme for frame dropping [11] that dynamically adjusts the number of skipped frames according to the incoming motion vectors and re-encoding errors due to transcoding, such that the decoded sequence can have smooth motion as well as improved transcoded picture quality. On the other hand, information about the rate-distortion (R-D) characteristics was needed in order to find optimal combinations of the frame rate and spatial quality in the rate control for source coding [16]-[18]. Despite its potential effectiveness, such an analytic approach does not seem applicable to general spatio-temporal adaptation and complex video content characteristics. Besides the complexity aspect, this approach cannot be readily extensible to different types of resources (other than the bit rate) and utility (quality measures).

It is likely that several transcoding operations satisfying the given constraints are feasible. Thus, we need a systematic and effective strategy for choosing the best one that meets the given constraints, which may even change rapidly. The complexity of the transcoder should be as low as possible because adaptation is often performed in an intermediate network device (proxy, gateway, router) between different networks and client devices to react to temporal bandwidth variations in real-time. In particular, the efficiency becomes crucial in the case when a video stream has to be delivered at different bit-rates to more than one client at the same time.

To facilitate a rate adaptation transcoding scheme in the face of such requirements, we propose a utility-based framework, which has been introduced in our previous work [19]-[21] at the conceptual level. In this paper, we describe the concrete formulation of the framework and its instantiation in realistic scenarios. The distribution and relations of three key parameters involved in adaptation problems—*adaptation*, *resource*, and *utility*—are modeled and represented using a utility function to guide the systematic solutions. This utility-based approach extends the conventional R-D model by allowing flexible consideration of diverse types of these parameters. The resource-utility relationships represent analogous but broader concepts than the R-D relationships. To illustrate the adaptation methods, we exemplify the implementation by employing the combination of FD and CD (referred to as FD-CD) on MPEG-4 (simple profile) compressed streams. Both methods can be efficiently implemented in the compressed domain by truncating parts of the compressed streams without full decoding. The case study on the MPEG-4 videos with FD-CD is appropriate for demonstrating computational solutions to this approach since the case is common in practical wireless mobile applications.

In the utility-based framework, one key issue is how to generate a utility function in real-time to accommodate live videos. For stored videos in on-demand applications, a utility function can be generated by exhaustive off-line simulation resulting in extensive computation since the computation time may not be a serious issue. Some works attempted to develop analytic models [16]-[18] that approximate the relationship between bit-rate and distortion. In another part of this work [22], we propose an alternative approach that exploits the strong correlations between content characteristics (scene complexity, motion activity, and so on) and utility function shapes (namely, resource-utility behaviors) for any given adaptation operation. We present a content-based statistical approach that combines content feature extraction, statistical clustering, and regression for predicting utility functions of live videos in real-time.

In addition, a utility function representing the relations among adaptation, resource, and utility should be described in an interoperable way so that adaptation engines from different parties can be used. In this paper, we present a description tool [23] that has been accepted as a part of MPEG-21 DIA [24] to address this issue.

This paper is organized as follows. In the next section, we introduce the utility-based adaptation framework with its overall architecture. In section III, a utility-based rate adaptation transcoding using FD-CD over MPEG-4 is used as an example to illustrate the usage of the framework. We present the proposed utility function description tool for utility-based adaptation in section IV. Experimental results illustrating the performance of the utility-based rate adaptation approach will be presented in section V. Section VI concludes the work of this paper.

II. Utility-Based Adaptation Framework

A three-tier server-proxy-client architecture, depicted in Fig. 1, is considered as a promising solution to meet diverse resource constraints in the UMA. Deploying an adaptation engine in the proxy allows adapting videos to dynamic resource constraints that were not known a priori from heterogeneous networks and terminals in real-time.

One possible scenario is that the server generates utility functions (UF) and sends them to the adaptation engine located in the proxy. The interoperable format for describing a utility function is presented in more detail in section IV. The description is used in the adaptation engine to guide the selection of the best adaptation operation. For live videos, UFs could be generated by a prediction-based approach in real-time. Such real-time prediction methods can also be implemented at the proxy if the proxy has enough computing power.

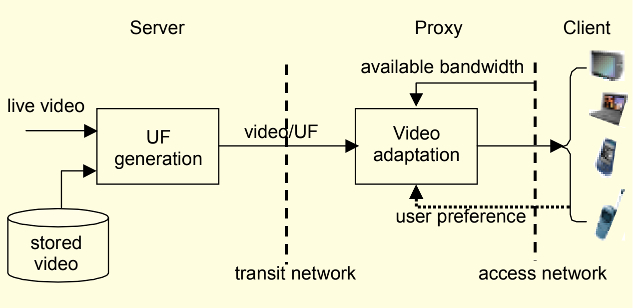


Fig. 1. A three-tier adaptation architecture using the utility-based framework.

1. Definition of Adaptation, Resource, Utility and Their Relations

In formulating video adaptation problems, we need to determine the video content entity *e*, adaptation space *A*, resource space *R*, and utility space *U*, as shown in Fig. 2. We use the term “space” in a loose sense here to indicate the multiple dimensionalities involved. For example, given the *k*-th entity e_k (as an example, a compressed video segment), there exists a set of adaptation operations $A=\{\mathbf{a}_i\}$ that can be applied to meet the diverse types of resource constraints defined in *R*. The quality of the resulting adapted video is measured using dimensions in utility space *U*.

In general, a video segment that is a sequence of consecutive image frames, which share some consistent content characteristics, could be a video entity. We assume certain content consistency so that unique content features can be extracted from each entity and used to predict the associated UF. In some cases, a video entity may correspond to a shot if the content characteristics do not vary within the shot. Otherwise, a shot may be further decomposed into several sub-segments, each of which is considered as an entity undergoing adaptation. Detection of entity boundaries can be easily achieved by computing the content characteristics (for example, camera motion) in the compressed video stream [25]. In practice, there is not a single correct decomposition of entities in a video sequence. Each entity should not be too long in order to ensure consistency within each entity, yet should not be too short so that the overhead of switching between different adaptation operations for different entities can be controlled. Based on the above definition, a video sequence *V* is composed of a number of segments as $V = \{e_k\}_{k=1}^{K}$.

The adaptation space is a conceptual space representing all

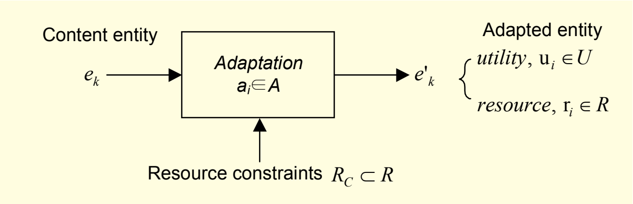


Fig. 2. Definition of adaptation, resource, and utility spaces involved in video adaptation problems in the utility-based framework.

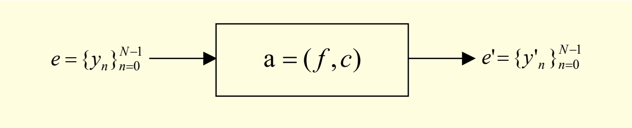


Fig. 3. Illustration of rate adaptation transcoding with an adaptation operator of FD-CD.

possible adaptations as $A=\{\mathbf{a}_i\}_{i=1}^{Q}$, where $\mathbf{a}_i=(a_{i1}, a_{i2},\ldots,a_{iq})$. Each point $\mathbf{a}_i$ in the q-dimensional space represents an adaptation operator that combines q adaptation operations (that is, $a_{i1}, a_{i2},\ldots,a_{iq}$), each of which is defined in each constituent dimension representing a corresponding adaptation method. For instance, if we consider the adaptation methods of FD and CD only, there are two dimensions: each dimension represents specific adaptation operations of FD and CD, respectively. For example, the FD dimension can be indexed based on a finite set of feasible frame dropping operations. The CD dimension can be indexed by the percentage of transform coefficients dropped. The origin of the space represents the null operation, namely, the original entity is kept intact.

Resources are constraints from terminals or networks; they include network bandwidth, power consumption, display size, and so on. In some applications, the resource constraints may include several types of resources. For example, in order to provide video streaming service to handheld devices, on top of the bandwidth, some other factors such as spatial resolution or power consumption should also be taken into account, simultaneously. In general, all types of resources to be met in an underlying application are represented by m-dimensional resource space as $R=\{\mathbf{r}_i\}_{i=1}^{M}$, where $\mathbf{r}_i=(r_{i1}, r_{i2},\ldots, r_{im})$. Each dimension represents a particular type of resource. It is assumed that there exist M discrete points in R for simplicity without the loss of generality. For the case of continuous resource space, the search and/or optimization in the decision of an optimal adaptation operator could be more complex [26].

Utility is the quality of a content entity when it's rendered on a user device after undergoing adaptation. When a content entity is adapted, its utility value is usually changed accordingly. Utility space may include attributes in l dimensions as $U=\{\mathbf{u}_i\}_{i=1}^{L}$, where $\mathbf{u}_i=(u_{i1}, u_{i2},\ldots, u_{il})$, since it could be measured in an objective or subjective manner with different measures. For instance, in addition to the peak signal-to-noise ratio (PSNR), a subjective preference such as a mean opinion scale or temporal smoothness may be included in other dimensions.

Let us consider the mapping relations among these three spaces with respect to video adaptation. Given e_k and $A=\{\mathbf{a}_i\}$, each point in the adaptation space, for example $\mathbf{a}_i$, represents a function that maps the input entity to its corresponding adapted entity as $e'_{ki}=\mathbf{a}_i(e_k)$. Each adapted entity has a corresponding point in R and a corresponding point in U; both corresponding points are associated with the $\mathbf{a}_i$ that has been applied. Such mapping relations are

$$h_j:\mathbf{a}_i(e_k)\rightarrow r_{ij}^k\text{, or simply } r_{ij}^k=h_j(\mathbf{a}_i(e_k)),\ j=1,\ldots,m \qquad (1)$$

and

$$g_j:\mathbf{a}_i(e_k)\rightarrow u_{ij}^k\text{, or simply } u_{ij}^k=g_j(\mathbf{a}_i(e_k)),\ j=1,\ldots,l, \qquad (2)$$

where $h(\bullet)$ and $g(\bullet)$ denote resource measurement operators and utility measurement operators, respectively. Note that the mapping among the points in the three spaces is often multiple-to-one since different adaptation operations may give the same resource values or induce the same utility value. The interesting point of the adaptation problem lies in choosing an optimal adaptation operator among multiple choices.

The mapping relations will be further specified when we consider two typical types of adaptation scenarios as below. First, let us assume an adaptation scenario in which the resource constraint is given as $R_C\subset R$. The mapping relationship between R and A, and between A and U, are given by

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

and

$$U_R=\{\mathbf{u}_i \mid \mathbf{u}_i=g(\mathbf{a}_i(e_k)), \mathbf{a}_i\in A_R\}. \qquad (4)$$

Using (3) and (4) for a given e_k and a set of constraints R_C, a set of adaptation operators A_R that satisfies the resource constraint and a corresponding set of points U_R is specified.

Second, similarly to the above formulation, we can also find the mapping relationships between U and A, and A and R, under the utility constraint, $U_C\subset U$, as

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

and

$$R_U=\{\mathbf{r}_i \mid \mathbf{r}_i=h(\mathbf{a}_i(e_k)), \mathbf{a}_i\in A_U\}. \qquad (6)$$

Similarly, for a given U_C, A_U satisfying U_C and a corresponding set of points R_U are specified using (5) and (6).

2. Adaptation Problem Formulations

Using the utility-based framework, many problems of resource-constrained scenarios in video adaptation, in which the main goal of adaptation is to maximize the quality of video in the environment of limited resources, can be formulated as follows.

Formulation 1. Resource-Constrained Utility Maximization Problem

Given a content entity e_k, and resource constraints $R_C \subset R$, find the optimal adaptation operation, $\mathbf{a}_{opt}$ that solves

$$\arg\max_{\mathbf{a}_i} |\mathbf{u}_i| = |g(\mathbf{a}_i(e_k))|,$$
$$\text{subject to } \mathbf{r}_i = h(\mathbf{a}_i(e_k)) \in R_C, \tag{7}$$

where $|\cdot|$ represents a measure combining different types of components to be defined under a given application.

Formulation 1 can be easily extended to one that imposes constraints in the utility space and aims at an overall resource minimization problem:

Formulation 2. Utility-Constrained Resource Minimization Problem

Given a content entity e_k, and utility constraints $U_C \subset U$, find the optimal adaptation operation, $\mathbf{a}_{opt}$, that solves

$$\arg\min_{\mathbf{a}_i} |\mathbf{r}_i| = |h(\mathbf{a}_i(e_k))|$$
$$\text{subject to } \mathbf{u}_i = g(\mathbf{a}_i(e_k)) \in U_C\ . \tag{8}$$

Adaptation problems formulated in either way are solved by a following corresponding proposition, of which the proof is quite simple from the relations of (3) and (4), or (5) and (6).

Proposition 1. For a given e_k and $R_C \subset R$, there exists an $\mathbf{a}_{opt}$ in A, which maximizes the utility of adapted entity, e'_k, if $A_S = \{\mathbf{a}_i \mid h(\mathbf{a}_i(e_k)) \in R_C\}$ is not empty.

Proposition 2. For a given e_k and $U_C \subset U$, there exists an $\mathbf{a}_{opt}$ in A, which minimizes the resource of adapted entity, e'_k, if $A_S = \{\mathbf{a}_i \mid g(\mathbf{a}_i(e_k)) \in U_C\}$ is not empty.

3. Utility Function Representation

In order to represent distributions of the three spaces and mapping relations among them, which should be delivered to an adaptation engine, a utility function is employed in our utility-based approach. A utility function is in general defined as a media quality metric representing a user satisfaction index as a function of resources. Utility functions are successfully applied for utility-based network adaptation [27] and video streaming applications [28].

For a given e_k and the defined A, a utility function represents a set of resource-utility pairs given by (9) as a discrete point in a resource-utility plan.

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

Note that (9) represents mapping relations of (1) and (2) in the case of involving one type of resource and utility. A utility function can be easily extended to represent the resource space with multiple types by extending the resource-axis into multidimensional space. Extension to the utility space is made in the same way.

In this manner, a utility function is able to encompass whole distributions over the three spaces and their mapping relations in a compact form. Moreover, it allows efficient searching and selection in solving the adaptation problems formulated as (7) or (8) by directly representing the resource-utility relations along with their associated adaptation operators.

A utility function can be simplified into classical forms in certain fields. For example, if the adaptation space is a one-dimensional space specifying only the step size for transform coefficient quantization, the resource space includes only the bit rate, and if the utility space includes only the SNR measure of the compressed video, the utility function representing the relationship between bit rate and utility can be reduced to the familiar R-D curve that is often used for video coding optimization.

III. FD-CD Transcoding

In this section, we present a realistic example that instantiates the above utility-based framework in UMA applications. We present a rate adaptation transcoding of MPEG-4 videos employing the combination of FD and CD operations. Let us illustrate how this specific adaptation problem is formulated and how corresponding utility functions are generated with the definitions of the three spaces as a case study of the utility-based approach.

1. Definitions of Adaptation, Resource, and Utility

For simplicity, we assume the entity undergoing adaptation is a group of pictures (GOP) in the MPEG-4 sequence. This means a particular adaptation operation of FD-CD is applied in the entire segment of a GOP. We denote the GOP entity as $e = \{y_n\}_{n=0}^{N-1}$, where y_n is the $(n+1)$th frame in the GOP of size N.

A two-dimensional adaptation space constituted by FD and CD, each of which is indexed by a finite set of adaptation operations, is defined as $A = \{\mathbf{a}_i = (f, c)\}$. A particular adaptation operation of FD is specified by frame-dropping factor f, which is given by

$$f = \frac{F_{org}}{F'}, \tag{10}$$

where F_{org} and F' represent the frame rate of the incoming video and the adapted video, respectively. To express a CD operation, the ratio of rate-reduction c is used,

$$c = 1 - \frac{R'}{R_{org}}, \tag{11}$$

where R_{org} and R' denote the bit rate of the incoming video and the adapted video, respectively. As a result, as illustrated in Fig. 4, $\mathbf{a} = (f, c)$ specifies an adaptation operation that repeats the basic pattern of FD-CD, which reduces the coefficients by percentage c on one frame and then drops the next (f–1) frames, until the end of the GOP. In this manner, the same value of c is applied among frames in the GOP. In other words, the uniform CD, where the amount of rate reduction to be achieved by CD is uniformly allocated with the same percentage among different frames, is assumed. In addition, to exploit the predictive coding dependence structure among frames in MPEG sequences, we further limit the operations to drop non-reference frames only, that is, B frames.

It's worthwhile to note some additional benefits of FD-CD in the context of UMA besides its simplicity in allowing real-time implementation. FD can meet only a coarse level of the target rate since the smallest unit of data that can be adjusted is an entire frame. On the other hand, CD is able to meet the target rate with a finer granularity by adjusting the amount of dropped coefficients. Moreover, it is often impossible to achieve the required target rate by performing FD or CD alone. In particular, in UMA applications, a wide range of bandwidth variation may exist due to heterogeneous network conditions. Therefore, the FD-CD enables us to accommodate a wide range of bandwidth variations by expanding the dynamic range of achievable bit rate while meeting the target rate at a finer level. At the same time, the FD-CD adaptation provides freedom in balancing the trade-offs between spatial and temporal quality, which is guided to be optimal by utility functions in a systematic way.

In this case of rate adaptation transcoding, the resource space includes only the network bandwidth available. For a given e and $\mathbf{a}$, the rate measurement operator $h(\bullet)$ maps $\mathbf{a}(e)$ into a point r in R, which is expressed as $r = h(\mathbf{a}(e))$. For the utility space, we use the PSNR to measure the quality of the adapted entity, $e' = \mathbf{a}(e)$, and specify the mapping between $\mathbf{a}(e)$ and u in U.

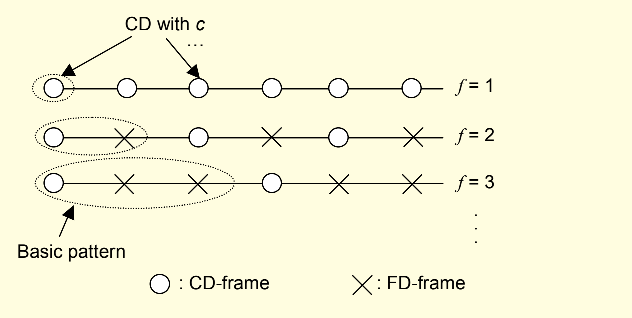


Fig. 4. Illustration of an adaptation operation of $\mathbf{a}=(f, c)$. Each row represents an adaptation operation $\mathbf{a}$. Each circle/cross represents a frame.

2. Rate Adaptation Problem

The problem of rate adaptation to dynamic bandwidth is formulated as (12), which is a specific case of Formulation 1.

$$\arg\max_{\mathbf{a}_i} PSNR(\mathbf{a}_i(e)) = \arg\min_{\mathbf{a}_i} D(f, c)$$
$$\text{subject to } \; r_i = h(\mathbf{a}_i) \le R_T, \tag{12}$$

where $D(f, c)$ denotes the overall distortion caused by the adaptation $\mathbf{a}=(f, c)$, and R_T represents the target rate to be met.

The overall distortion over a GOP having a frame interval $[0, N–1]$ is rewritten by

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

where the metric $\|\cdot\|$ denotes the mean square error. As shown in (13), the overall distortion is the sum of the FD distortion due to dropped frames and the CD distortion due to dropped coefficients, denoted by D_{fd} and D_{cd}, respectively. To model the FD distortion, we assume that zero-order hold interpolation is used to reconstruct the dropped frame at the receiver. So, the last non-dropped frame to which CD has been applied is repeated for the reconstruction of the dropped frame, that is, $y_{nf+m} = y'_{nf}$. Under this assumption, the FD distortion due to the $(nf+m)$th dropped frame is given by

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

The CD distortion at the nf-th frame with c is given by

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

Let us present the procedure to solve the optimization problem formulated in (12).

Step 1. For a particular case of FD, $f \in \{f_i\}$, determine the overall FD distortion and rate-reduction ratio c satisfying the rate constraint, R_T. Ratio c is given by

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

where B_{nf} is the bit count of the nf-th frame of the incoming stream.

Step 2. For each frame, to which CD is to be applied, determine the minimum CD distortion by optimally allocating the dropped coefficients among blocks within the frame with the determined c in the previous step by solving the following:

$$\min \|y - y'\|$$

$$\text{subject to} \quad B' \le B_T = (1-c)B \,, \tag{17}$$

where B_T is the target bit count of the frame that should be met and B' is the bit count of the adapted frame by CD with c. The minimum overall CD distortion is obtained by summing the minimum distortion of each frame in the uniform CD. The details of this step of optimized CD are presented in the following subsection III.3.

Step 3. Repeat steps 1 and 2 for all other cases of FD, $f \in \{f_i\}$ and compute the overall distortion (13). Finally, determine an optimal adaptation operator $\mathbf{a}_{opt}= (f_{opt}, c_{opt})$ that minimizes the overall distortion.

Even though the above procedure can be applied to find out $\mathbf{a}_{opt}$ for a given rate constraint at the proxy, it is not applicable to real-time applications. In the utility-based approach, however, we use the above procedure in generating a utility function at the server in advance to cope with the wide variation of rate constraint in a dynamic way rather than dealing with a given specific rate constraint. The details will be given in subsection III.4.

3. Optimized CD

For a given c, CD within a frame can be implemented in many different ways resulting in different qualities and complexities. This means that there is ambiguity in the specification of CD with c. In other words, different allocations of dropped coefficients among blocks or MBs even with the same c are possible according to different implementations of CD. With respect to the description of utility function, this ambiguity issue will be presented in detail in subsection IV.2.

We implement our CD using the Lagrange optimization algorithm based on the work in [8] with some modifications. Specifically, instead of using MB-level dropping (that is, each MB has a unique dropping point shared by all four luminance and two chrominance blocks), we use a block-level search to find individual dropping points for each block. The implemented CD solves the problem of (17) which is rewritten by

$$\min_{\{b_m\}_{m=1}^{M}} \left\{ \sum_{m=1}^{M} D_m(b_m) \right\}$$

$$\text{subject to} \quad \textstyle\sum_{m=1}^{M} B'_m(b_m) \le B_T \,, \tag{18}$$

where $D_m(b_m)$ and $B'_m(b_m)$ denote the resulting distortion and rate for block m when the coefficients of which the scan order are greater than b_m have been truncated, respectively, and M is the total number of blocks in a frame.

In the computation of $D_m(b_m)$, we not only have to take into account the error accumulated from past pictures, but also the error that will be propagated to future pictures, in addition to the current error when we engage in coefficient dropping. However, the causally optimal algorithm that ignores the propagation error can be a practical approach to avoid complexity and delay due to “look-ahead” optimization. For further simplification to be deployed at proxies, we finally adopt the memoryless algorithm [8] that ignores the accumulated error and treats each picture as an intra one in the generation of utility function. It was reported that ignoring the accumulated errors does not much affect the quality and allows achieving essentially optimal (with a difference of about 0.2 dB) performance in a typical size of GOP [8]. A very simple alternative approach that can serve as a lower bound in terms of quality is a purely rate-based approach. Here, it does not take into account the distortion at all (either current or accumulated), and only meets the prescribed rate constraints with a uniform amount of bit reduction among each block.

Figure 5 shows the video quality improvement (average PSNR) by using the block-level optimization versus the MB-level optimization when the memoryless algorithm is applied. As a comparison baseline, the result from the rate-based approach is also shown. Our experiment on standard sequences of “Coastguard” with a range of different rates shows 0.5 dB to 2 dB improvement by performing block-wise optimization that exploits a smaller search region for dropping points

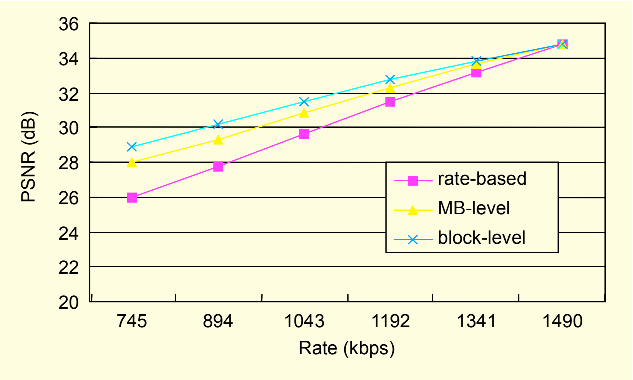


Fig. 5. PSNR comparison for CD with different implementations: the rate-based, memoryless optimization on block level, and memoryless optimization on MB level. The sequence of “Coastguard” is originally coded by MPEG-4 at 1.5 Mbps with GOP (GOP size=15, sub-GOP size=3) and adapted up to the rate value of 745 kbps.

resulting in more even distortion. The detail of this implementation is beyond the discussion of this paper and the users are recommended to refer to report [29]. We also see that the memoryless optimization is quite useful in terms of improved quality over the rate-based approach with an allowable increase of computation complexity even though it does not account for the accumulated errors.

4. Utility Function Generation

A utility function is generated by applying the procedure presented in the previous subsection III.2, where an optimal operator and its corresponding quality are obtained for a given rate constraint, to a set of rates sampled over a whole dynamic range of rate reduction. We can also generate a utility function in a slightly different way, where distortion-rate pairs given by (19) are computed and collected for the predefined $A = \{\mathbf{a}_i \mid \mathbf{a}_i = (f, c)\}$.

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

Figure 6 shows an example of a utility function computed for an entity (here, a GOP) of the "Coastguard" sequence compressed by MPEG-4. We consider only a single measure of resource (for example, bit rate) and utility (for example, PSNR) in this case study of FD-CD, so the similar scheme to the conventional R-D curve can be used. The PSNR for all simulations are computed with respect to the original uncompressed image sequence under the assumption of a zero-order hold interpolation.

In this example, we define 18 adaptation operators, $\mathbf{a}_i = (f, c)$ of FD-CD that combines FD with $f \in \{1, 3, 15\}$ and CD with $c \in \{0.0, 0.1, 0.2, 0.3, 0.4, 0.5\}$. As a result, 18 corresponding points exist in the resource–utility plan. Based on the given GOP structure (GOP size=15, sub-GOP size=3), each of $f \in \{1, 3, 15\}$ corresponds to "no dropping", "all B frames dropping", and "all B and P frames dropping" (resulting in an I-frame only sequence), respectively. Note that we can use linear approximation to represent other adaptation operators by connecting two adjacent points sharing the same FD operation because interpolation on CD operation is possible between two such points. Consequently, $\mathbf{a}_i$ is grouped in the same curve having the same value of f with different values of $c \in [0, 0.5]$. In this way, we can represent adaptation operators and their corresponding utilities meeting the rate constraint over a wide range at a finer level while assuming a few rate constraints in the generation of the utility function.

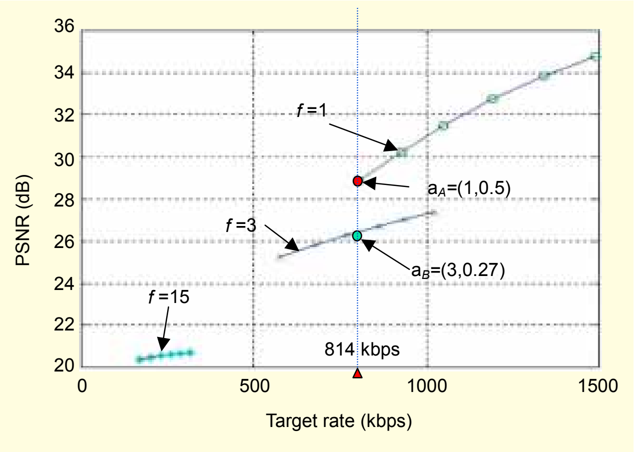


Fig. 6. An example of utility function using FD-CD adaptations on the "Coastguard" coded at 1.5 Mbps with GOP (GOP size=15, sub-GOP size=3). Here the memoryless optimization on block level has been applied as an optimized CD.

We observe that the predefined $A=\{\mathbf{a}_i\}$ covers a range of bandwidth variation up to about 200 kbps as shown in Fig. 6. For a given rate constraint, all of the feasible adaptation operators will be known by the utility function. Then, according to a rule of decision-making employed in the adaptation engine, which is typically the utility criterion, one out of these operators will be chosen. For instance, there are available $\mathbf{a}_A = (1, 0.5)$ and $\mathbf{a}_B = (3, 0.27)$ for the target rate of 814 kbps, then $\mathbf{a}_{opt} = \mathbf{a}_A$ in terms of PSNR quality.

Generation of the utility functions can be done using exhaustive computations in which all possible adaptation operations are performed and the resulting utility value and required resource are computed. Alternatively, utility functions can be predicted using the content features automatically extracted from the incoming video in real-time. In another part of this paper [22] we present our work in such solutions, called a content-based utility function prediction.

IV. Utility Function Description

In the utility-based approach, to explore a systematic solution of adaptation, such utility function representing the relations among adaptation, utility, and resource should be described in an interoperable way in order to be delivered to adaptation engines. In order to address this issue, we proposed a description tool to MPEG [23], which has been adopted into the current draft of MPEG-21 DIA [24].

Note that there is a good match between the goals of UMA and MPEG-21 as far as access of media using heterogeneous terminals and networks is concerned. In particular, MPEG-21 DIA [24], [30] defines a set of descriptors of user, natural environment, terminal, and network characteristics and other resource (namely media) descriptions for interoperable adaptation allowing transparent access to multimedia across a wide range of networks and devises. In this section, we present how to describe the information represented by utility functions in an XML description using the proposed descriptor along with some related issues.

1. Utility Function Description

Figure 7 depicts an XML schema diagram of the proposed descriptor of *UtilityFunctionType* under *AdaptationQoSType*, which is a tool defined in the DIA part of MPG-21 for terminal and network quality of service (QoS), addressing the problem of media adaptation to constraints imposed by terminals and/or networks. The description of a utility function, whose example is shown in Fig. 6, is done by several descriptors defined in UtilityFunctionType as follows: Constraint describes a set of sample points of the region of interest in the resource space, and AdaptationOperator describes one or more permissible adaptation operators meeting the resource constraints corresponding to each constraint point; *Utility* indicates the value of video utility after the adaptation specified by corresponding *AdaptationOperator* is applied. Note that the descriptor accommodates the cases where multiple instances of constraint and/or utility are required by allowing multiple occurrences of *Constraint* and *Utility*, respectively. In this manner, *UtilityFunctionType* describes feasible adaptation operators and their resulting utilities in a list format using a discrete set of sampled constraint points as indexes.

The exact semantics of the aforementioned elements are specified by the elements of *IOPin* under *AdaptationQoSType* by referring to the predefined classification schemes [24], of which aliases are optionally listed by *DescriptionMetadata*. In the current standard, various kinds of adaptation operators, utility, and constraint are defined to accommodate other feasible cases of adaptation as much as possible. For instance, the proposed descriptor also supports other types of adaptations such as wavelet reduction, adaptation of FGS streams, spatial size reduction, and so on by instantiating corresponding adaptation operators defined in the classification scheme. The *AdaptationQoSType* contains one or more *Module*, each of which is derived by the most appropriate representation format among the list, matrix or function.

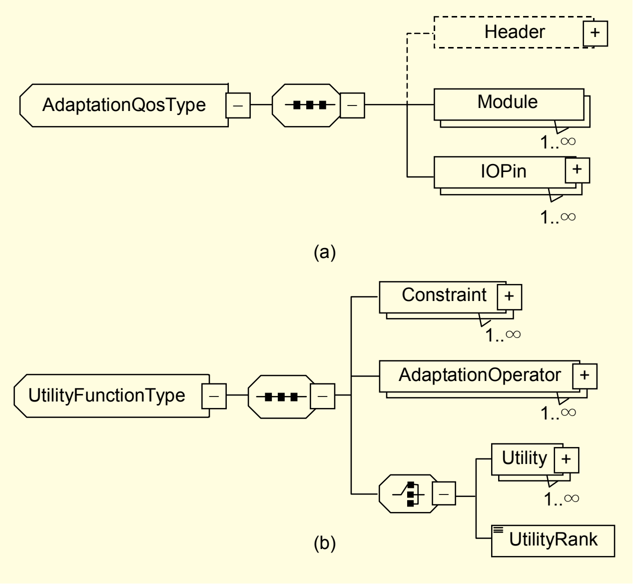


Fig. 7. Schema diagrams of the descriptors of (a) *AdaptationQoSType* and (b) *UtilityFunctionType*.

The following example of Fig. 8 shows an instance of the *AdaptationQoS* describing the utility function shown in Fig. 6 using *UtilityFunctionType*. The FD-CD is used for bit-rate adaptation to the given constraint of bandwidth, and a PSNR-measured utility is described. Specific semantics of instantiated adaptation operator, constraint, and utility are specified by using *IOPin*s.

The values of *Constraint* are ordered so that a triplet of a constraint point, a set of *AdaptationOperators* representing a certain adaptation operation, and a corresponding value of *Utility* can be associated. Therefore, if there exist multiple adaptation operations satisfying the same constraint value, the value of that constraint point is repeated for the association. For instance, there exist two adaptation operations satisfying the given bandwidth of 814 kbps: one is CD with the value of (COEFF_DROPPING = 0.5), and the other is a combination of FD with (B_FRAMES = 2, P_FRAMES=0) and CD with (COEFF_DROPPING = 0.31); the two adaptation operations have PSNR values of 29.10 and 27.53, respectively. Both descriptions correspond to $\mathbf{a}_A = (1,0.5)$ and $\mathbf{a}_B = (3,0.27)$ shown in Fig. 6.

2. Ambiguity in Specifying Adaptation Operations

Adaptation operations may not be unambiguously defined in some adaptation methods. For instance, there is ambiguity in the specification of the operation of CD with c since, for example, $c = 0.2$ does not specify the exact set of coefficients to be dropped as described in subsection III.3. Therefore, different implementations of CD between the server and adaptation engine may choose different sets of dropped coefficients and result in slightly different utility values. In other words, the described utility values at the server may be inconsistent depending on the implementation of the associated adaptation operator at the adaptation engine. On the other hand, for scalable compression formats such as JPEG-2000 and MPEG-4 FGS, parts of the scalable stream can be truncated in a consistent way without such ambiguity.

```
<DIA>
  <DescriptionMetadata>
    <ClassificationAlias alias="AQoS" href="urn:mpeg:mpeg21:2003:01-DIA-AdaptationQoSCS-NS"/>
  </DescriptionMetadata>
  <Description xsi:type="AdaptationQoSType">
    <Module xsi:type="UtilityFunctionType">
      <Constraint iOPinRef="BANDWIDTH">
        <Values xsi:type="IntegerVectorType">
          <Vector>1510 1359 1200 1071 1071 941 814 814 1000 1000 909 712 600 396 359 331 293 255 217</Vector>
        </Values>
      </Constraint>
      <AdaptationOperator iOPinRef="B_FRAMES">
        <Values xsi:type="IntegerVectorType">
          <Vector>0 0 0 0 2 0 0 2 0 2 2 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 4 4 4 4 4 4</Vector>
        </Values>
      </AdaptationOperator>
      <AdaptationOperator iOPinRef="COEFF_DROPPING">
          <Values xsi:type="FloatVectorType">
            <Vector>0.0 0.1 0.21 0.3 0.0 0.4 0.5 0.31 0.35 0.08 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 31.58 28.62 30.27 29.10 27.53 30.69 28.33 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>
```

Fig. 8. XML instance of AdaptationQoS using utility function shown in Fig. 6.

To address the ambiguity issue, we introduce the notion of utility ranking in our descriptor [23]. The likelihood of getting consistent utility rankings from different implementations is higher than getting consistent absolute utility values. Furthermore, we assume that, in some applications, the relative rankings of utility values among different adaptation operations meeting the same constraint are more useful than the absolute values of utility. In this consideration, the *UtilityFunctionType* allows instantiation of either *Utility* or *UtilityRank* as shown in Fig. 7.

V. Performance Evaluation

In this section, we present experimental results, which have been carried out to validate the feasibility of the proposed utility-based approach in terms of FD-CD rate adaptation transcoding as well as the descriptor addressing the ambiguity issue. With respect to the evaluation of the overall efficiency, we developed a software tool of the adaptation engine employing our utility-based FD-CD to adapt MPEG-4 videos in response to dynamic bandwidth constraints. Given the resource-utility descriptions available in an interoperable form, the adaptation tools and software can be deployed in the third-party proxy, server, or client to perform real-time selection of optimal adaptation.

1. Implementation of an Adaptation Tool

We have implemented a software prototype to demonstrate the feasibility and efficiency of the utility-based transcoding solutions. The prototype consists of a parser, an adaptation engine, and a user interface as shown in Fig. 9. The parser

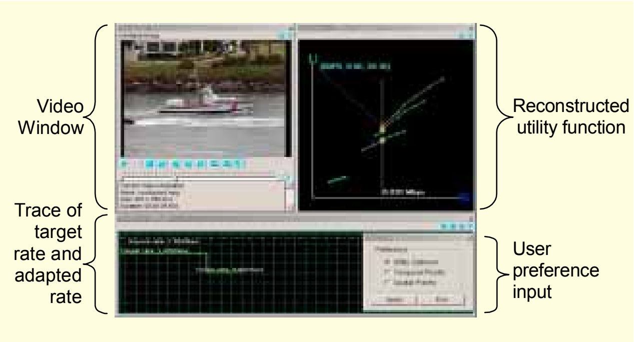


Fig. 9. User interface of the implemented adaptation tool.

extracts the utility function information by parsing the delivered XML description (described in section IV.1). The adaptation engine selects an optimal adaptation operator based on the extracted utility ranking at the given target rate, and carries out adaptation of an MPEG-4 video with the selected FD-CD operation in real-time.

In the experiment, an adaptation engine can efficiently select an optimal adaptation operator by an exhaustive search since it is a simple case of a resource-constrained utility maximization problem. In general, however, an adaptation engine needs to solve one or more constrained optimization problems to make an appropriate adaptation decision. The details of search and optimization strategies in an adaptation decision, which is out of scope of this paper, are addressed in [26].

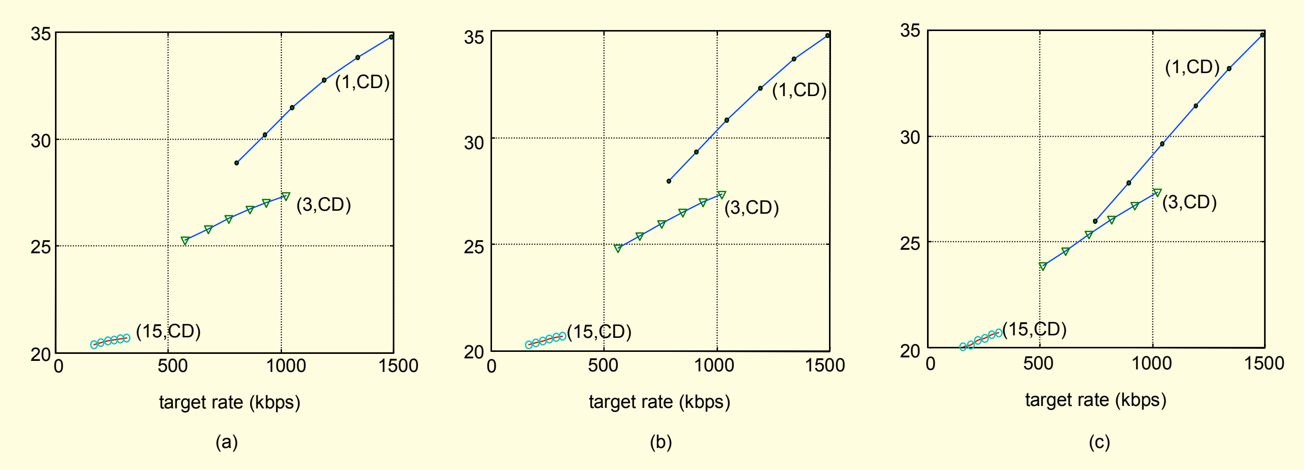
We have implemented an FD-CD transcoder using the Microsoft MPEG-4 Verification Model reference codes. The operation of FD-CD has been shown to be fast enough for real-time operation on a common PC system (Intel Pentium IV) with test sequences of CIF (352×288 pixels) format and 30 FPS, coded at 1.5 Mbps with a typical GOP size (15 frames) structure. In contrast to full decoding by the reference decoder, the FD-CD (with optimized CD routine) works around twice as fast.

The user interface allows for monitoring of the adaptation process by showing the adapted output video, the utility functions from which the optimal operator is selected, and the trace of the adapted rate following the time-varying target rate. The interfaces are updated in real time when consecutive video segments are adapted one by one, under the dynamically changing network resource condition.

In terms of the overall performance, the utility-based adaptation achieves promising gain (0.83 dB in PSNR and noticeable improvement in subjective quality) [22] over the conventional system, which simply uses the most frequent adaptation operation for each given target rate. The most frequent operation is the one that produces the best quality at the given bit rate for most video clips in the training set. However, such an approach does not take into account the content characteristics of the current video segment in question and thus does not benefit from using the knowledge contained in the utility function that characterizes the unique behavior of the current video segment.

2. Utility Rank Description

In order to show the usefulness of *UtilityRank* in addressing the ambiguity issue, we generated utility functions for three different implementations of CD with the same sequence used in Fig. 6. In this experiment, we adopt the adaptation space of FD-CD described in subsection III.4, in which there exist 18 adaptation points of $\mathbf{a}_i = (f, c)$ that constitute four curves in the utility function. Figure 10 shows the variations of utility functions resulting from different implementations. We observe that there are indeed noticeable variations of utility values among



(a)

(b)

(c)

Fig. 10. Utility functions resulting from different implementations of CD: (a) memoryless optimization on block level, (b) memoryless optimization on MB level, and (c) rate-based CD.

different implementations. However, the utility rankings among competing adaptation operations at a given bit rate are consistent among different transcoder implementations. Although it is possible that in some cases even the utility ranking may vary among different transcoder implementations, we found it seems to be rare based on our empirical experimentations. Therefore, the ranking description proves to be useful in addressing the ambiguity of utility values caused by different transcoder implementations.

VI. Conclusions

A general framework and systematic methodology for video adaptation are described to meet diverse resource conditions and user preferences in UMA environments. The framework explicitly models the major concepts involved in adaptation processes—adaptation, resource, and utility—using a utility function. It resembles the conventional R-D model, but allows for greater flexibility in combining diverse types of adaptations, resources, and utility measures. To demonstrate such a utility-based approach, we describe effective rate adaptation transcoding based on the combination of frame dropping and DCT coefficient dropping in meeting dynamic bandwidth variations in real-time in the context of UMA.

In addition, we present description schemes of utility functions, which have been accepted as a part of MPEG-21 DIA. Such a standardized description scheme is valuable for achieving an interoperable solution across multiple service or content providers. In addition, to address the sensitivity of the utility function to codec implementation variations, we introduce the concept of utility ranking to resolve the ambiguity in choosing the optimal adaptation operation.

We demonstrate the realization of the utility-based approach in a practical case of MPEG-4 video bit rate adaptation transcoding employing FD-CD. Our experiments show promising performance gain as well as computational efficiency by using the utility-based adaptation. The method developed in this paper can also be applied to other scenarios of adaptation, for instance, adaptation of scalable video streams and optimal adaptation based on a subjective quality measurement.

References

[1] R. Mohan, J.R. Smith, and C.-S. Li, “Adapting Multimedia Internet Content for Universal Access,” *IEEE Trans. Multimedia*, vol. 1., Mar. 1999, pp. 104-114.

[2] P. van Beek, J.R. Smith, T. Ebrahimi, T. Suzuki, and J. Askelof, “Metadata-Driven Multimedia Access,” *IEEE Signal Processing Mag.*, vol. 20, Mar. 2003, pp. 40-52.

[3] J. Bormans, J. Gelissen, and A. Perkis, “MPEG-21: The 21st Century Multimedia Framework,” *IEEE Signal Processing Mag.*, vol. 20, Mar. 2003, pp. 53-62.

[4] A. Vetro, C. Christopoulos, and H. Sun, “Video Transcoding Architectures and Techniques: An Overview,” *IEEE Signal Processing Mag.*, vol. 20, Mar. 2003, pp. 18-29.

[5] Y. Nakajima, H. Hori, and T. Kanoh, “Rate Conversion of MPEG Coded Video by Requantization Process,” *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, Apr. 1996, pp. 191-199.

[7] P. Assuncano and M. Ghanbari, “Post-Processing of MPEG2 Coded Video for Transmission at Lower Bit Rates,” *Proc. IEEE Int. Conf. On Acoustic, Speech, and Signal Processing'96*, vol. 4, Atlanta, GA, May 1996, pp. 1998-2001.

[8] A. Eleftheriadis, “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, Mar. 1999, pp. 30-40.

[10] T. Shanableh and M. Ghanbari, “Heterogeneous Video Transcoding to Lower Spatio-Temporal Resolutions and Different Encoding Formats,” *IEEE Trans. Multimedia*, vol. 2, June 2000, pp. 101-110.

[11] K.T. Fung, Y.L. Chan, and W.C. Siu, “New Architecture for Dynamic Frame-Skipping Transcoder,” *IEEE Trans. Image Processing*, vol. 11, Aug. 2002, pp. 886-900.

[12] A. Vetro, H. Sun, and Y. Wang, “Object-Based Transcoding for Adaptive Video Content Delivery,” *IEEE Trans. Circuit Syst. Video Technol.*, vol. 11, Mar. 2001, pp. 387-401.

[13] W. Li, “Overview of Fine Granularity Scalability in MPEG-4 Video Standard,” *IEEE Trans. Circuit Syst. Video Technol.*, vol. 11, Mar. 2001, pp. 301-317.

[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, Mar. 2001, pp. 318-331.

[15] MPEG Available at: http://www.chiariglione.org/mpeg/

[16] A. Vetro, Y. Wang, and H. Sun, “Rate-Distortion Optimized Video Coding Considering Frameskip,” *Proc. IEEE Int. Conf. Image Processing*, Thessaloniki, Greece, Oct. 2001, pp. 534-537.

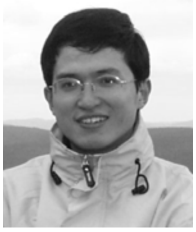
[17] J.-J. Chen and H.-M. Hang, “Source Model of Transform Video Coder and its Application— Part II: Variable Frame Rate Coding,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, Apr. 1997, pp. 187-298.

[18] E.C. Reed and J.S. Lim, “Optimal Multidimensional Bit-Rate Control for Video Communications,” *IEEE Trans Image Processing*, vol. 11, no. 8, Aug. 2002, pp. 873-885.

[19] S.-F. Chang, “Optimal Video Adaptation and Skimming Using a Utility-Based Framework,” *Proc. IWDC-2002*, Capri Island, Italy,

Sept. 2002.
[20] S.-F. Chang and A. Vetro, "Video Adaptation: Concepts, Technologies, and Open Issues," *Proc. IEEE*, vol. 93, no. 1, Jan. 2005, pp. 148-158.
[21] J.-G. Kim, Y. Wang, and S.-F. Chang, "Content-Adaptive Utility-Based Video Adaptation," *Proc. ICME-2003*, Baltimore, Maryland, July 2003.
[22] Y. Wang, J.-G. Kim, S.-F. Chang, and Hyung-Myung Kim, "Utility-Based Video Adaptation for UMA and Content-Based Utility Function Prediction for Real Time Video Transcoding," revision for publication in *IEEE Trans. Multimedia*.
[23] 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.
[24] ISO/IEC 21000-7:2004, *Information Technology-Multimedia Framework-Part 7: Digital Item Adaptation*.
[25] J-G Kim, H.S. Chang, J. Kim, and H.-M. Kim, "Threshold-Based Camera Motion Characterization of MPEG Video," *ETRI J.*, vol. 26, no. 3, June 2004, pp.269-272.
[26] D. Mukherjee, E. Delfosse, J.-G. Kim, and Y. Wang, "Optimal Adaptation Decision-Taking and Network Quality-of-Service," *IEEE Trans. Multimedia,* vol. 7, no. 3, June 2005, pp. 454-462.
[27] P. Bocheck, A. Campbell, S.-F. Chang, and R.R.-F. Liao, "Utility-Based Network Adaptation for MPEG-4 Systems," *Proc. NOSSDAV '99*, Basking Ridge, New Jersey, June 1999.
[28] C.E. Luna, L.P. Kondi, and A.K. Katsaggelos, "Maximizing User Utility in Video Streaming Applications," *IEEE Trans. Circuits and Syst. Video Technol.*, vol. 13, no. 2, Feb. 2003, pp. 141-148.
[29] Y. Wang, J.-G. Kim, and S.-F. Chang, "MPEG-4 Real Time FD-CD Transcoding," Columbia University, ADVENT Technical Report #11122003, 2003.
[30] A. Vetro and C. Timmerer, "Digital Item Adaptation: Overview of Standardization and Research Activities," *IEEE Trans. Multimedia*, vol. 7, no. 3, June 2005, pp. 418-426.

Jae-Gon Kim received the BS degree in electronics engineering from Kyungpook National University, Korea, in 1990, the MS and PhD degrees in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Korea, in 1992 and 2005, respectively. Since 1992, he has been a Senior Member of Research Staff, in the Broadcasting Media Research Group of Electronics and Telecommunications Research Institute (ETRI), Korea. He is currently the Team Leader of the Convergence Media Research Team. From 2001 to 2002, he was a Staff Associate at the Department of Electrical Engineering, Columbia University, New York. His research interests include video processing, networked video, multimedia applications, MPEG-7 and MPEG-21.



Yong Wang received his BS and MS degrees from Tsinghua University, Beijing, China in 1999 and 2001 respectively, both in electrical engineering. He is currently a PhD candidate supervised by Professor Shih-Fu Chang in the Department of Electrical Engineering at Columbia University, New York, NY. From 2001, he has been as a Research Assistant in the digital video and multimedia (DVMM) group at Columbia University. During the summer of 2003, he was an intern working at HP labs, Palo Alto, CA. From 1999 to 2001, he was a visiting student at Microsoft Research Asia, Beijing, China. His research interests include video coding, adaptation, communication and content-based analysis.

Shih-Fu Chang is a Professor in the Department of Electrical Engineering of Columbia University. He leads Columbia University's Digital Video and Multimedia Lab (http://www.ee.columbia.edu/dvmm), conducting research in multimedia content analysis, video retrieval, multimedia authentication, and video adaptation. Systems developed by his group have been widely used, including VisualSEEk, VideoQ, WebSEEk for image/video searching, WebClip for networked video editing, and Sari for online image authentication. He has initiated major projects in several domains, including a digital video library in echocardiogram, a content-adaptive streaming system for sports, and a topic tracking system for multi-source broadcast news video. Chang's group has received best paper or student paper awards from the IEEE, ACM, and SPIE. He is Editor in Chief of IEEE Signal Processing Magazine (2006-7); a Distinguished Lecturer of the IEEE Circuits and Systems Society, 2001-2002; a recipient of a Navy ONR Young Investigator Award, IBM Faculty Development Award, and NSF CAREER Award; and a Fellow of IEEE since 2004. He helped as a General Co-Chair for ACM Multimedia Conference 2000 and IEEE ICME 2004. His group has also made significant contributions to the development of MPEG-7 multimedia description schemes.

Hyung-Myung Kim received the BS degree in electronics engineering form Seoul National University, Seoul, Korea, in 1974, and the MS and PhD degrees in electrical engineering form the University of Pittsburgh, Pittsburgh, PA, in 1982 and 1985, respectively. Since 1986, he has been with the Department of Electrical Engineering and Computer Science, KAIST, Daejeon, Korea, and is currently a Professor there. During the summer of 1997, he was on sabbatical leave as a Visiting Researcher at the Department of Electrical Engineering, Pennsylvania State University, University Park. His research interests include digital signal/image processing, digital transmission of voice and communications data, and image and multidimensional system theory. Dr. Kim was the Treasurer of the IEEE Daejeon Section in 1992. He has been an Editorial Board Member of Multidimensional Systems and Signal Processing since 1990.