Available online at www.sciencedirect.com
SCIENCE DIRECT®
Signal Processing: *Image Communication* 20 (2005) 413–434
SIGNAL PROCESSING: IMAGE COMMUNICATION
www.elsevier.com/locate/image

Effective adaptation of multimedia documents with modality conversion

Truong Cong Thang, Yong Ju Jung, Yong Man Ro*

Multimedia Group, Information and Communications University (ICU), Yuseong, Daejon, PO Box. 77, 305-732, Republic of Korea

Received 18 October 2004; accepted 8 March 2005

Abstract

Besides content scaling, modality conversion is an important aspect of content adaptation. In this paper, we study modality conversion of a multimedia document under the constraints of available resource and human factor (user preference). We first formulate the content adaptation process of a multimedia document as a general constrained optimization problem and then extend it to effectively support modality conversion. To represent conversion boundaries between different modalities, we propose the overlapped content value (OCV) model that relates the content value of different modalities with resources. Also, the human factor is specified in the form of modality conversion preference, and then integrated into the framework by modifying the OCV model. We apply the Viterbi algorithm of dynamic programming and its fast approximation to the optimization problem with a practical resource constraint, namely the total datasize. Experiments demonstrate that modality conversion (in combination with content scaling) brings a wider range of adaptation for QoS support. Moreover, the proposed approach is shown to be effective to apply in practice.

Keywords: Content adaptation; MPEG-21 DIA; Modality conversion; User preference; Multimedia processing

1. Introduction

Content adaptation is a key solution to support the quality of service (QoS) for multimedia services over heterogeneous environments [9,7]. More specifically, multimedia documents are adapted to meet various constraints from terminals, network connections, and user preference, while providing the best possible presentation to the user. Content adaptation has two major aspects: one is *modality conversion* (also called transmoding), which converts contents from one modality to a different modality (e.g. from video to image or text); the other is *content scaling*, which changes the amount of resource (e.g. datasize) and so the quality of the contents without converting their modalities. The term *resource* here means a kind of computing support needed for delivering contents

*Corresponding author. Tel.: +82 42 866 6279; fax: +82 42 866 6245.
E-mail address: yro@icu.ac.kr (Y.M. Ro).

0923-5965/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.image.2005.03.004

to the user. Typical types of resources include datasize, bitrate, complexity, power supply, etc. It should be noted that, in the literature, the term *transcoding* is often interchangeable with content adaptation. In this paper, transcoding is used as a common name for both content scaling and modality conversion.

Most research on content adaptation for multimedia documents has so far focused on content scaling (e.g. [9,7]). However, content scaling sometimes may not be able to support a wide range of resource constraint variations, and a possible solution to this problem is to convert the contents into other modalities. For example, let us consider the adaptation of a multimedia document containing multiple content objects (e.g., videos, images), under the datasize constraint. This type of resource constraint can be obtained from the available memory at the target terminal, or the product of connection bitrate and download time. When the datasize constraint is too low, some objects may be scaled with very low qualities or even discarded. In this case, it can be expected that a sequence of "important" images may be more appropriate than a scaled video of low quality, and some textual description could be more useful than simply discarding an image. These are some typical examples of modality conversion, i.e., video-to-image and image-to-text. So, modality conversion, in combination with content scaling, may bring a wider adaptation range for QoS support. For each content object, two important questions corresponding to the two aspects of content adaptation are:

1. "What is the output modality of the object?"
2. "What is the quality (content value) of the object?"

Generally, there are four main factors that may affect the decision on modality conversion [34]. The first factor is the *modality capability*, which is the support for the user's consumption of certain modalities. This factor can be determined from the characteristics of a terminal (e.g., text-only pager), or surrounding environments (e.g., a too noisy place). The second factor is the *user preference* (i.e., the human factor) that shows the user's interest in different modalities. The third factor includes the *resource* constraints, e.g., at some point the allocated amount of memory is not enough to play a video content. The fourth factor is the *semantics* of the content itself. For instance, between an interview video and a ballet video, the provider would be more willing to convert the former to a stream of text.

Many previous studies tackle specific cases of modality conversion under the constraint of modality capability. For instance, text-to-speech (TTS) technologies have long been used for many applications, such as receiving emails through telephones [40], or reading textual contents on PC screens for blind users [3]. And, video-to-image conversion can be employed to send an image sequence (i.e., a video summary) to a terminal that does not support video playback (e.g., [16]). Some recent studies use metadata to help automate the processing of conversion between specific modalities [19,2]. Meanwhile, there has so far been a little research on modality conversion under the resource constraint. In [23], modality is considered as one variable of the adaptation of a multimedia document to resource constraint, yet this approach does not show the relationship between different modalities. The work in [24] uses the Lagrangian method for selecting content versions of different modalities. However, as shown later in Section 6, if the quality is modeled by a non-concave function of the resource, this method is not suitable to find the output modality (as well as the quality). The research in [6] presents a cross-media adaptation scheme, but no QoS framework has been established. Recently, the use of modality conversion for QoS management in streaming service has been investigated through representing the qualities of different modalities with respect to the amount of resource [32]. In these previous studies, the user preference on modality conversion is not considered. Moreover, the usefulness of modality conversion in adapting multimedia documents under the low resource constraint has not been quantitatively shown.

In this paper, we propose a systematic approach that can convert (and scale) the content objects of a multimedia document subject to the resource constraint and the user preference. We formulate the content adaptation process of a multimedia

document as a general constrained optimization problem and then extend it to support modality conversion. The overlapped content value (OCV) model, first mentioned in [33], is employed to describe the relationship between the resources and the content values of different modalities. Based on the OCV model, we then propose a novel way for incorporating the user preference on modality conversion into the adaptation process. In terms of the constrained optimization problem, we propose two algorithms based on the Viterbi algorithm of dynamic programming to find accurate solutions, thus helping answer the two above questions for each content object. Through experiments, we verify that when modality conversion is applied, the user will receive higher overall quality. The proposed approach is also shown to be effective in (1) supporting the user preference on modality conversion and (2) selecting the destination modalities of objects under different conditions of the resource constraint.

This paper is organized as follows. Section 2 discusses some related work. Section 3 presents an overview of content adaptation, including modality conversion. In Section 4, the modeling of modality conversion is shown by the OCV model, which relates the content qualities of different modalities. The user preference on modality conversion and its use in the adaptation process are presented in Section 5. Section 6 addresses dynamic programming based methods to determine the modality and quality of contents in the adaptation. In Section 7, we present experiments that show the usefulness of the proposed approach. Finally, Section 8 presents our conclusions and future work.

2. Related work

The problem of content adaptation for QoS management has long been studied. As described in [20], the problem may have single or multiple resources, and single or multiple qualities. Our work in this paper focuses on the problem with a single quality (called content value) and a single resource (namely datasize). Actually, a quality measure consistent with human perception and an efficient resource allocation method are the two most important issues in content adaptation for QoS management [8]. However, a good quality measure is still an open research topic because the quality is subjective and may vary between different users. In order to customize the quality for different users, the user preference may be employed [23,8]. As to the resource allocation issue, a lot of studies model this problem as a variation of the rate-distortion framework [9,24,27] which is solved by either the Lagrangian method, the dynamic programming method, or various related approximation methods [27,28]. While formulations based on the rate-distortion framework often consider the case of a single resource, they can be straightforwardly extended to handle multiple resources (e.g., bitrate and complexity in [22]). Besides, some authors employ the formulation of the multiple choice knapsack problem (MCKP) for the resource allocation issue, which results in various interesting approximation algorithms for both single resource case [5] and multiple resource case [20]. Recently, a good overall picture for video adaptation with multiple resources and multiple utilities (qualities) has been described in [8].

Modality conversion is also an interesting topic in Human Computer Interfaces (HCI). Traditionally, HCI studies the use of multiple modalities for the interface between human and computers/terminals. As terminals become more and more heterogeneous, some HCI-related research considers the changes of modalities in the interface according to different terminal characteristics (e.g. [1,11,4]). In [1], the authors mention the idea of a data model which is independent of modalities, so as to give the user the same experience across different interfaces. In [11], users' choices of presentation modalities for different terminals are subjectively investigated, resulting in general guidelines of converting modalities in certain contexts. It should be noted that, in HCI, not only contents modalities but also interaction modalities (e.g., voice or keyboard) are subject to conversion [4]. In addition, modality conversion could be particularly useful for disabled users to access multimedia contents [3,35]; however, this research area still requires more study.

Some recent standardization activities target content adaptation. Regarding the modality conversion aspect of content adaptation, MPEG-7 [13] describes a wide variety of modalities by different classification schemes (e.g., ContentCS, GraphicsCodingCS, etc.). The modality capability of a terminal can be determined from Usage Environment descriptions of MPEG-21 Digital Item Adaptation (DIA) [14]. MPEG-21 DIA also provides the Conversion Preference tool to specify the user preference on modality conversion, and the Universal Constraints description tool to define the (resource) constraints of the adaptation [14]. Meanwhile, the W3C creates the Composite Capability/Preference Profiles (CC/PP) protocol to exchange the characteristics of users and terminals [39]. In addition, description tools which suggest how to convert modalities will also be standardized by MPEG [31].

3. Overview of content adaptation

This section presents an overview of content adaptation. While modality conversion is the main point of this paper, content scaling is also described in detail due to the close relationship of these two operations.

3.1. Adaptation engine

First, let us define some basic terms used in this paper. From the highest level, a *multimedia document* is a container of one or more *content objects* (or *object* for short). An object is an entity conveying some information, e.g., a football match. Each object may have many *content versions* of different modalities and qualities. A content version is a physical instance of the object, e.g., a video or audio file reporting a football match. The modality of a content version may be single (e.g., video only or text only) or combined (e.g., "audio + video" modality combining both video and audio).

In a multimedia system, content adaptation is carried out by an adaptation engine whose architecture is depicted in Fig. 1. The adaptation engine consists of three main parts: a decision engine, a modality converter, and a content scaler.

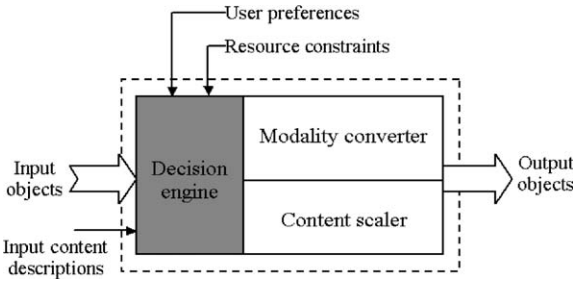

Fig. 1. Architecture of an adaptation engine. The decision engine analyzes the content descriptions, resource constraints, and user preferences to make optimal decisions on content scaling and modality conversion.

The decision engine has three important metadata inputs. The first input is the content description, which provides a format-agonistic adaptation engine with information about multimedia contents to be adapted (e.g., genres, bitrates, formats, etc.). The second input consists of the resource constraints that show the characteristics of networks and terminals (e.g., bitrate, storage). The third input is the user preference. In this paper, the user preference implicitly means the user preference on modality conversion (or modality conversion preference). The decision engine analyzes these inputs and then makes optimal decisions on modality conversion and content scaling.

The modality converter and the content scaler include specific (modality) conversion and (content) scaling operations in order to adapt the content objects according to instructions from the decision engine. Some objects may be passed directly to the content scaler, without converting their modality; while others are first passed to the modality converter, and then possibly to the content scaler.

The modality converter and the content scaler can be *either offline or online*. In the offline case, content objects are transcoded in advance into various versions of different modalities and qualities; then under certain constraints, appropriate versions are selected and sent to the user [34,23,24]. In the online case, the needed versions are created on the fly. The decision engine, in all cases, must find the appropriate modality and the

level of scaling for each object. That is, the functionality of the decision engine is essentially the same for both online and offline transcodings. Besides that, the current trend is to ease online transcoding by various techniques, such as employing scalable coding [26] and bitstream syntax metadata [29]. So, the decision engine needs to be examined in the first place. The mechanism of a decision engine that can make accurate decisions on modality conversion, as well as content scaling, is the very focus of this paper.

3.2. General problem formulation

To adapt a multimedia document to resource constraints, the decision engine must answer the two questions raised in Section 1 for every object. Otherwise, we cannot apply appropriate conversion and scaling operations to the objects. To this end, the decision-making process is first presented as a traditional constrained optimization problem. Denote N as the number of objects in the multimedia document. For each object i in the document, we have the following definitions:

- V_i is the content value of the object.
- $\boldsymbol{R}_i = \{R_{i1}, R_{i2}, \ldots, R_{iL}\}$ is the set of resources of the object where R_{il} is the amount of resource l that object i consumes; and L is the number of resources.
- $\boldsymbol{P}_i$ is the user preference for the object.
- $\boldsymbol{M} = \{m_1, m_2, \ldots, m_Z\}$ is the *modality capability* that lists the indexes of supported modalities; Z is the number of supported modalities.

The normal trend is that V_i is a non-decreasing function with respect to each R_{il}. V_i also depends on the personal evaluation of the user. In addition, when some modalities are not supported at the target terminal, the content versions of those modalities will be useless. So, V_i can be represented as a function of $\boldsymbol{R}_i$, $\boldsymbol{P}_i$, and $\boldsymbol{M}$:

$$V_i = f_i(\boldsymbol{R}_i, \boldsymbol{P}_i, \boldsymbol{M}). \tag{1}$$

Given a set of resource constraints $\boldsymbol{R}^c = \{R_1^c, R_2^c, \ldots R_L^c\}$, we have the problem statement as follows:

Find $\boldsymbol{R}_i$ for each object, so as to

$$\text{maximize} \sum_{i=1}^{N} w_i \cdot V_i \quad \text{while} \sum_{i=1}^{N} R_{il} \leqslant R_l^c \quad \text{for all } l = 1...L, \tag{2}$$

where w_i is the importance value of object i, $0 \leqslant w_i \leqslant 1$.

After solving the amounts of resources from problem (2), one can find the modality and the quality for each object using (1). This is actually the conventional problem of resource allocation [9,24,27,28]. Our goal is to extend this framework to effectively support modality conversion. Here, two main challenging issues are: (1) the quantification of the content value, and (2) the method to accurately decide the amounts of resources (and so the modality and the content value) for each object given that quantification. The first issue will be tackled by Sections 4 and 5, while the second issue will be the topic of Section 6.

4. Overlapped content value model

As the first step to quantify the content value in the context of modality conversion, this section deals with the modeling of content value w.r.t. resources and modalities. For simplicity, in this and the next sections, we present chiefly the case of a single resource, where the set of resources $\boldsymbol{R}_i$ is replaced by resource R_i. The involvement with multiple resources can be treated in a similar manner [20].

The process of content scaling, either online or offline, can be represented by some "rate-quality" curve, which shows the quality of a scaled content according to the bitrate (or any resource in general). A recent trend is to use the rate-quality curve as metadata to automate content scaling [38,18,37]. Usually, the curve is obtained for a particular modality because each modality has its own scaling characteristics. Extending the concept of the rate-quality curve, we propose the overlapped content value model to conceptually represent both the content scaling and modality conversion of each content object.

The OCV model of an object consists of multiple rate-quality curves representing the content value of different modalities versus the amount of resource. The number of curves in the model is the number of modalities the object would have. Fig. 2a illustrates the OCV model of an object that is originally of video modality. Here, the rate-quality curve of each modality (called *modality curve*) can be assigned manually or automatically. For example, the provider may manually set the rate-quality curve to be a log function [24]; and in [18], utility functions (i.e., rate-quality curves) are estimated in an automatic way using a machine learning approach. Normally, a modality curve saturates when the amount of resource is high. Each point on a modality curve corresponds to a content version of that modality. The intersection points of the modality curves represent the conversion boundaries between the modalities. Though not really mandatory, each modality curve should cut another one, at most, at one point. Similar to rate-quality curves in image/video coding, the OCV model can be either operational (i.e., empirically measured) or parametric [28].

The content value function is the convex hull of the modality curves in the OCV model (Fig. 2b). We call a point on the content value function a *selection*. Denote $VM_{ij}(R_i)$ as the modality curve of modality j of object i, $j = 1 \ldots J_i$, where J_i is the number of modalities of object i and $j = 1$ is the index of the original modality; $VM_{ij}(R_i) \geqslant 0$ for all $j = 1 \ldots J_i$; and $\rho_{ij} \geqslant R_i$ where ρ_{ij} is the maximum amount of resource for the modality j of object i. The content value function of object i can be represented as follows:

$$V_i = \max\{VM_{ij}(R_i) | j = 1 \ldots J_i\}. \quad (3)$$

If we know the amount of resource allocated for an object, we can find its appropriate modality and content value by mapping the amount of resource to the content value function. As the amount of resource decreases from a maximum to zero, the object's modality will be converted in an ordered manner. For instance, in Fig. 2, the original video modality will be converted to image, to audio, and then to text. We say, the order of video-to-video conversion is "first", the order of video-to-image is "second", and so on. Obviously, the order of video-to-video ("first") is higher than the order of video-to-image ("second"). These orders of conversions are called *existing orders*, as opposed to the user's orders described later in Section 5.

Currently, to measure the content value, we decompose it into *perceptual quality* and *semantic quality*. The former refers to the user's satisfaction in perceiving an object, regardless of what information the object contains; the latter, which is crucial for modality conversion, refers to the amount of conveyed information, regardless of how the object is presented. The content value can be defined as the average of these two qualities. The perceptual and semantic qualities can be obtained from subjective tests, where for each adapted version subjects give scores of the two qualities compared to the original version. The detailed procedure and result analysis of these subjective tests can be found in [32,36].

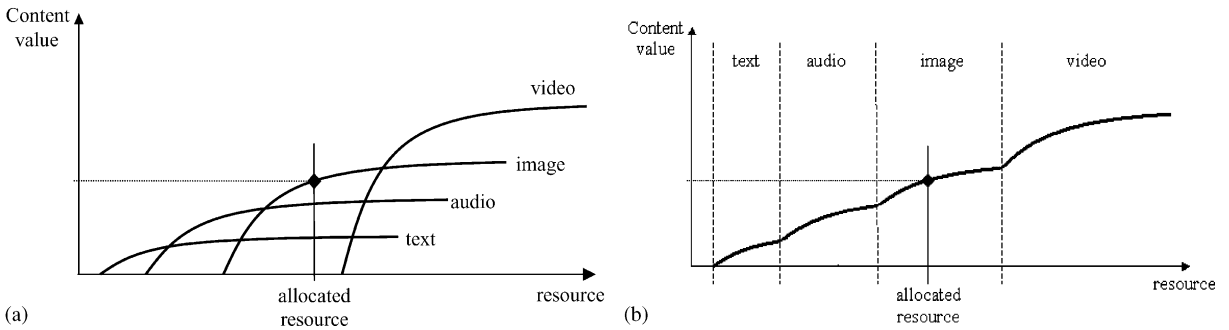

Fig. 2. Overlapped content value model of a content object (a). The model consists of modality curves; each curve relates the content value of a modality to the amount of resource. The final content value function is the convex hull of the modality curves (b).

Fig. 3a depicts an OCV model obtained from the subjective test of an audiovisual object. The original object has both audio and video channels; its combined modality is called "audio + video", denoted as AV. This OCV model shows that, to give the best quality at a certain amount of resource, the original AV modality may be converted to "audio + image" modality (denoted as AI, where the video channel is converted to an image sequence), to audio modality (where the video channel is removed), and to text modality (which is the transcription of the audio channel). The text modality does not include the description of the visual scene because we intentionally want the user to receive similar information from the audio and text modalities. Here, the scaling operation for the AV modality is increasing the quantization parameter of the video channel [37]; and the scaling operation for the AI modality is reducing the number of key-images extracted from the video channel [21] (the audio channel is unchanged). The audio modality is scaled by reducing the audio spectrum [10], and the text modality is scaled by summarizing the textual content [25]. More information about content formats and scaling/conversion methods used in the subjective test can be found in [32,36].

Another example OCV model for an image object is shown in Fig. 3b. This model includes the curves of image, audio, and text modalities. However we can see that the audio modality curve is below the convex hull, so this modality is not selected in the adaptation except when only audio modality is supported at the terminal. The scaling operation for the image modality of this object is reducing the spatial resolution.

The orders of conversions of the AV and image objects are given in the second and third rows of

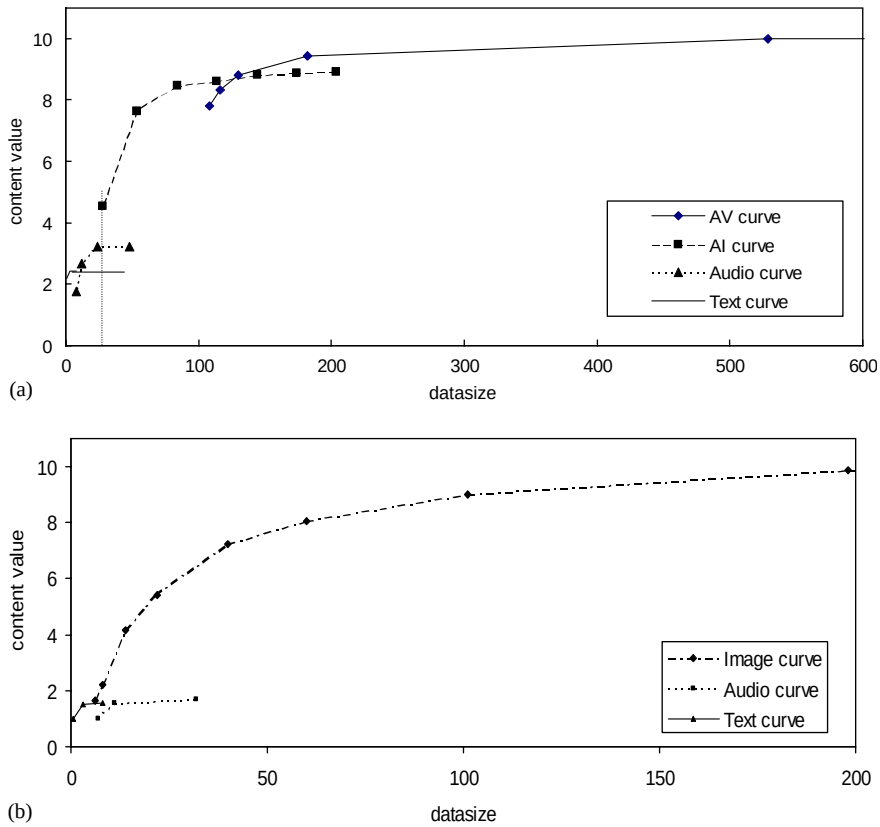

Fig. 3. OCV models obtained from subjective tests for an AV object (a) and an image object (b).

Table 1
Example orders of conversions. Each cell shows the order of a conversion from one modality to one modality

From\To	AV	AI	Image	Audio	Text
AV	First	Second	*Unsupported*	Third	Fourth
Image	*Unsupported*	*Unsupported*	First	Third	Second
Text	*Unsupported*	*Unsupported*	*Unsupported*	Second	First

Note: some conversions are not supported.

Table 1. Note that the maximum content value of each object is 10. These two operational OCV models can be used directly in content adaptation. Besides, parametric OCV models can be obtained by curve-fitting to the empirical data of each modality curve. When the object is involved with multiple resources, the modality curves become "modality surfaces" in a multidimensional space, and the above discussion is still valid.

In one sense, the OCV model is somewhat similar to the InfoPyramid framework [24], which describes the pre-transcoded versions of an object (i.e., in the offline case). Yet, the key advantages of the OCV model include: (1) it clarifies the relationship of modality conversion and content scaling for both offline and online transcodings; and especially (2) it is the *underlying basis* to handle the modality conversion preference as presented in the following.

5. Modality conversion preference in content adaptation

This section addresses the dependence of the content value on the user preference, which is the second step to quantify the content value in our approach. We first present an efficient specification of modality conversion preference and then propose a method to integrate the user preference into the adaptation process, specifically to modify the OCV model. This user preference (officially named Conversion Preference) [33] was developed as a description tool for MPEG-21 Digital Item Adaptation (DIA) [14].

5.1. Modality conversion preference

During the adaptation process, there are many possibilities of modality conversion, but the user may prefer some modalities to others. The role of the modality conversion preference is to enable the user to specify his/her choices on modality conversion for different objects.

5.1.1. Preference on modality-to-modality conversions

We see that this user preference should not allow for the fixed choices of modality conversion only. For example, the user requests that all videos be converted to audios, yet if the terminal cannot support audio modality, the objects will be discarded. Furthermore, the user preference should not allow only for the choices on the destination modalities regardless of the original modalities. For example, in some cases, audio may be the best alternative to video, whereas text may be the best alternative to image. Thus, we contend that, to flexibly support the various conditions of the terminal/network, the user preference should support choices on modality-to-modality conversions. From now on, a *conversion* means a translation from an original modality to a destination modality.

5.1.2. Two levels of preference

To help answer the two questions pointed out in Section 1, the user preference on a conversion is divided into two levels: a qualitative *order* and a quantitative *weight*.

Given an object, the *order* of a conversion from the object's original modality indicates the order by which the corresponding destination modality will be selected as the amount of resource is reduced. As described in Section 4, with a given OCV model, there are already *existing orders* of conversions, which can be considered as the provider's orders and are used as the *default orders* for the user. Using the modality conversion preference, the user may specify new orders of conversions that are different from the existing orders. In that case, the OCV model will be modified according to the user's orders of conversions.

The *weight* of a conversion indicates the importance, in the user's view, of the corresponding destination modality of an object. The *weights* help the decision engine to determine at which content value of the current modality, modality

conversion should be made. This is based on the fact that the conversion boundaries between modalities are determined by the user's quality evaluation of different modalities, which is very subjective. So, the user's weights can be used to scale the content values of different modalities, resulting in changes of conversion boundaries. Each weight has a default value of 1, and the higher the weight is, the more important the modality. The detailed specification of the user preference can be found in [14]. In the following, we focus on the use of this user preference in quantifying the content value.

5.2. Modifying the OCV model according to the user preference and the modality capability

In this subsection, we present methods to effectively modify OCV models according to the user preference. For completeness, modality capability is also considered. It should be noted that the procedures described here are independent of the number of resources. The modality curve $VM_{ij}(R_i)$ is now denoted as VM_{ij} for simplicity.

5.2.1. Modifying according to the modality capability

When a modality is not supported, the versions of that modality have zero content value at the terminal. Thus, the curves of unsupported modalities should be removed. Various characteristics of terminals, users, and surrounding environments, which may affect the set of supported modalities, have been standardized as the description tools in MPEG-21 DIA [14]. Denote $\Im_i$ as the set of supported modalities of object i, $\Im_i = \boldsymbol{M} \cap \{1, \ldots, J_i\}$. The content value is now represented by:

$$V_i = \max\{VM_{ij} | j \in \Im_i\}. \quad (4)$$

5.2.2. Modifying according to the modality conversion preference

(*a*) *According to the orders of conversions*

The orders of conversions represent the qualitative preference of the user. The user may just want to specify this qualitative preference level, ignoring the weights of conversions which are quantitative. As mentioned above, the OCV model of a given object has the existing (or default) orders of conversions. If the user's orders of conversions are different from the existing orders, the OCV model will be changed. Denote $ORD_e(j)$ and $ORD_u(j)$ respectively as the existing order and user's order of the conversion from the object's original modality to modality j. The algorithm to modify the OCV model of object i according to the user's orders of conversions can be carried out as follows:

Algorithm 1:

Step 0: $j = 1$

Step 1: $j^* = 1$

Step 2: If $j^* \neq j$ and $\{j^*, j\} \subset \Im_i$, go to Step 3, otherwise go to Step 5.

Step 3: If $\mathrm{ORD}_e(j)$ is higher than $\mathrm{ORD}_e(j^*)$, go to Step 4, otherwise go to Step 5.

Step 4: If $\mathrm{ORD}_u(j)$ is lower than $\mathrm{ORD}_u(j^*)$, remove VM_{ij} and go to Step 6, otherwise go to Step 5.

Step 5: $j^* = j^* + 1$. If $j^* > J_i$, go to Step 6, otherwise go back to Step 2.

Step 6: $j = j + 1$. If $j > J_i$, stop, otherwise go back to Step 1.

Essentially, this algorithm considers to remove modality curves whose orders are not compatible with the user preference. Normally, if $\mathrm{ORD}_e(j)$ is higher than $\mathrm{ORD}_e(j^*)$, modality j will be selected before modality j^*. However, when considering the user preference, if $\mathrm{ORD}_u(j)$ is lower than $\mathrm{ORD}_u(j^*)$, VM_{ij} should be removed so that modality j will not be selected before modality j^*. This comparison process (by Steps 3 and 4) is repeated for every modality in the OCV model. Because the number of modalities in an OCV model is usually not many so the complexity of this algorithm is negligible.

As an example, given the OCV model in Fig. 2, we have the existing orders of conversions as follows: $\mathrm{ORD}_e(1)$ of video-to-video is "first", $\mathrm{ORD}_e(2)$ of video-to-image is "second", $\mathrm{ORD}_e(3)$ of video-to-audio is "third", and $\mathrm{ORD}_e(4)$ of video-to-text is "fourth". With these orders, the audio modality will be selected before the text modality. Now, suppose the user's orders of

conversions are specified as follows: $ORD_u(1)$ is "first", $ORD_u(2)$ is "second", $ORD_u(3)$ is "fourth", and $ORD_u(4)$ is "third". That means the user prefer the text modality to the audio modality. The above algorithm first takes the video modality to check against other modalities as in Steps 3 and 4; and the decision is not to remove the video modality curve. Next, the decision is similar for the image modality. However, in order not to select the audio modality before the text modality, the audio modality curve is removed by Step 4. The text modality curve is also unaffected. Finally, the resultant modified OCV model is depicted in Fig. 4.

(*b*) *According to the weights of conversions*

After specifying the orders of conversions, the user can further give the weights of conversions. In this case, the weights of conversions can be used to adjust the conversion boundaries between the modalities while the orders of conversions are maintained. Denote u_{ij} as the user's weight of conversion j of object i. The weights are now used to scale the distances d_{ij}'s between the maximum content values of different modalities as depicted in Fig. 5a. Note that the sum of d_{ij}'s is fixed and equal to the maximum content value of object i. First, the scaled distances d_{ij}^s can be computed by

$$d_{ij}^s = \frac{u_{ij}\sum_{r\in\Im_i} d_{ir}}{\sum_{r\in\Im_i} u_{ir}d_{ir}}\, d_{ij}. \tag{5}$$

We see that d_{ij}^s's reflect the user preference and $\sum_j d_{ij} = \sum_j d_{ij}^s$. Then, the scale factor for modality j is calculated as follows:

$$s_{ij} = \frac{\sum_{r\geqslant j, r\in\Im_i} d_{ir}^s}{\sum_{r\geqslant j, r\in\Im_i} d_{ir}}. \tag{6}$$

And the modified content value function is rewritten as:

$$V_i = \max\{s_{ij}\cdot VM_{ij} | j\in\Im_i\}. \tag{7}$$

By this method, the modality curves are scaled according to the user's weights, while the maximum content value and the orders of conversions remain unchanged. An example result of this modification is shown in Fig. 5b. The result of modifying the OCV model according to the weights of conversions is obviously changes in the boundaries between the modalities. If the weight of a conversion increases, the operating range of the corresponding destination modality (delimited by the boundaries) will be broadened.

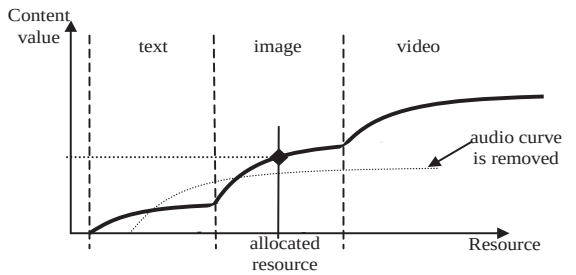

Fig. 4. Modifying the overlapped content value model according to the user's orders of conversions. In this example, the audio curve is removed.

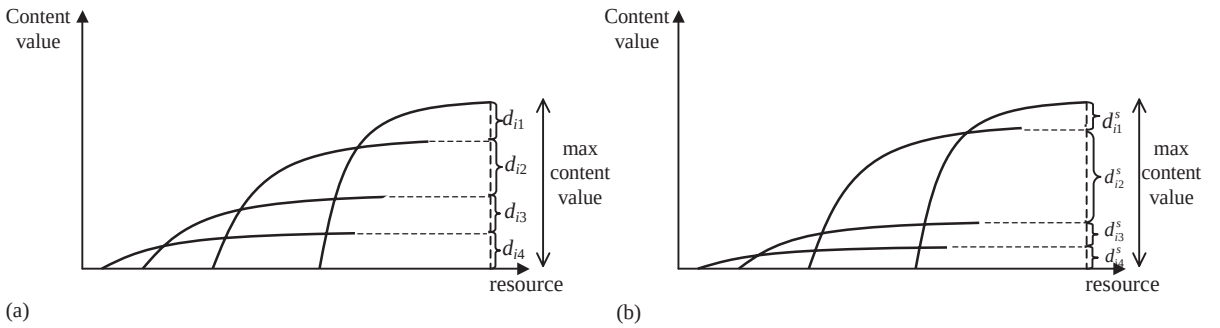

Fig. 5. Modifying the overlapped content value model according to the user's weights of conversions. The weights are used to scale the distances d_{ij}'s between the modality curves (a). The scaled curves reflect the user preference, while the maximum content value remains unchanged (b).

6. Resource allocation algorithm

6.1. Formulation of the single resource problem

The previous sections have quantified the content value based on the OCV model, the modality capability, and the user preference. Now given that we have the final content value functions of the objects, we need to find some efficient method to solve problem (2). In this paper, our main concern is modality conversion, which may exist no matter what types of resources are involved. For simplicity, in this section we consider only one practical resource—the datasize of each object. Correspondingly, the only resource constraint is the datasize constraint, i.e., the total datasize available for a multimedia document. The problem with multiple resource constraints is reserved for our future research.

Denote D_i as the datasize of object i, and D^c as the datasize constraint. The content value function can be represented by

$$V_i = f_i(D_i, \boldsymbol{P}_i, \boldsymbol{M}). \tag{8}$$

Problem (2) is now converted into an optimization with a single constraint as follows:

Find D_i for each object, so as to

$$\text{maximize} \sum_{i=1}^{N} w_i V_i \quad \text{while} \sum_{i=1}^{N} D_i \leqslant D^c. \tag{9}$$

6.2. Optimal solution by the Viterbi algorithm

A content value function can be continuous or discrete. If it is continuous, we may discretize it because practical transcoding is done in the unit of bits or bytes. In the following, we implicitly suppose it is discrete, either originally or after discretization. Then, a content value function will have a finite number of *selections*. Meanwhile, function (8), which is constituted from multiple modality curves, is inherently non-concave. Thus the above optimization can be solved optimally by the Viterbi algorithm of dynamic programming [27,12].

Apart from channel coding, the Viterbi algorithm has also been used in lossy compression [16], which is somewhat similar to the application of content adaptation. The principle of the Viterbi algorithm lies in building a trellis to represent all viable allocations at each instant, given all the predefined constraints. The basic terms used in the algorithm are defined as follows (Fig. 6):

- *Trellis*: The trellis is made of all surviving paths that link the initial node to the nodes in the final stage.
- *Stage*: Each stage corresponds to an object to be adapted.
- *Node*: In our problem, each node is represented by a pair (i,a_i), where $i = 0 \ldots N$ is the stage number, and a_i is the accumulated datasize of all objects until this stage.
- *Branch*: If selection k at stage i has the value-datasize pair (V_{ik}, D_{ik}), then node $(i-1, a_{i-1})$ will be linked by a branch of value V_{ik} to node (i, a_i) with:

$$a_i = a_{i-1} + D_{ik}, \tag{10}$$

satisfying (if not, the branch will not be linked):

$$a_i \leqslant D^c. \tag{11}$$

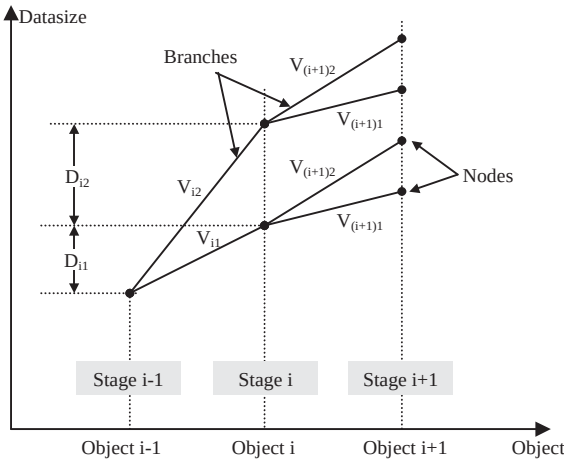

Fig. 6. Trellis diagram grown by the Viterbi algorithm. Each stage corresponds to an object of the document, and each branch corresponds to a *selection* for a given object. Each branch has an associated pair of content value and datasize, for example selection 1 at stage i has content value V_{i1} and datasize D_{i1}. A path corresponds to a set of selections for all objects in the document.

- *Path*: A path is a concatenation of branches. A path from the first stage to the final stage corresponds to a set of possible selections for all objects.

From the above, we can immediately see that the optimal path, corresponding to the optimal set of selections, is the one having the highest weighted sum $\sum_{i=1}^{N} w_i V_i$ (called the total content value of the adapted document). We now apply the Viterbi algorithm to generate the trellis and to find the optimal path as follows [27,12]:

Algorithm 2:

Step 0: $i = 0$. Start from the initial node (0, 0)

Step 1: At each stage i, add possible branches to the end nodes of the surviving paths. At each node, a branch is grown for each of the available selections; the branch must satisfy condition (11).

Step 2: Among all paths arriving at a node in stage $i+1$, the one having the highest accumulated sum of $\sum_{t=1}^{i+1} w_t V_t$ is chosen, and the rest are pruned.

Step 3: $i = i + 1$. If $i \leqslant N$ go back to step 1, otherwise go to step 4.

Step 4: At the final stage, compare all surviving paths then select the path having the highest value of $\sum_{i=1}^{N} w_i V_i$. That path corresponds to the optimal set of selections for all objects.

The complexity of this algorithm can be estimated as follows. Suppose H_i is the maximum datasize of object *i,* measured in a given transcoding unit (e.g., 1 KBs). That is, there are at most $H_i + 1$ selections (including the selection of zero datasize) on the content value function of object i. Also, the resource constraint is D^c units. At stage i, the number of nodes to be considered is $\min(H_1 + H_2 + \cdots + H_i, D^c)$; this excludes the *zero* node (with accumulated datasize of zero) because it has only one associated branch. The best branch at a node on stage i is chosen among at most $H_i + 1$ branches arriving at that node. Moreover, the content value of each incoming branch is calculated as the sum of the content value accumulated in the path up to the previous stage and the content value of the current branch. So, for each node on stage i, the complexity is linear with $H_i + 1$. The complexity at the stage i will be $O[(H_i + 1)\min(H_1 + H_2 + \cdots + H_i, D^c)]$. The complexity of the first stage ($i = 1$) is not taken into account because the operation at this stage is just linking $H_1 + 1$ branches. Also, backtracking to get the optimal path can be neglected in estimating the order of complexity [27]. Then the total complexity C of the algorithm is

$$C = O\left\{\sum_{i=2}^{N}\left[(H_i + 1)\min\left(\sum_{m=1}^{i} H_m, D^c\right)\right]\right\}. \quad (12)$$

In practice, the problem of resource allocation represented as a constrained optimization is often solved by two basic methods: the Lagrangian method and dynamic programming method [28]. In [24], the Lagrangian method is adopted to find the allocated amounts of resource. However, the Lagrangian method is not really suitable with non-concave content value functions. The drawback of the Lagrangian method is illustrated in Fig. 7, in which we have an AV object that may be converted to audio and text. The concave hull of the model is the curve (ABCD), where (BC) is a linear segment. With the Lagrangian method, the selected content version corresponds to the contact point of the concave hull with a tangent line having a slope λ, called the Lagrangian multiplier. Because segment (BC) is linear, the points on that segment, except B and C, are never selected. This leads to an unexpected consequence: the points below the concave hull (e.g., points around the conversion boundaries) are not selected. In particular, if a modality curve lies below the concave hull (as the audio curve in Fig. 7), that modality is

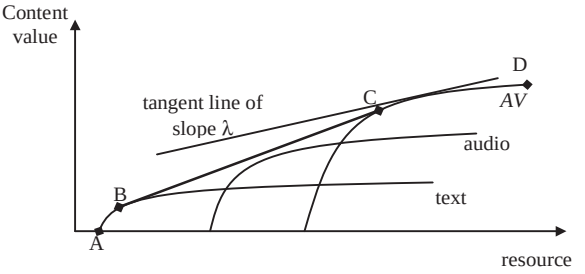

Fig. 7. Drawback of Lagrangian method. The solution ranges include only segments (AB) and (CD).

never reached. The actual range of selections is also limited, e.g. (AB) and (CD) in Fig. 7. In this case, we see that the allocated datasize may be either much larger or much smaller than the amount that is actually needed. A similar example of this issue can be found in [28].

The advantage of dynamic programming is that it can work with non-concave content value functions, while its disadvantage is high complexity. However, in Section 7, through practical considerations, we will show that the Viterbi algorithm of dynamic programming has potential for real-time computation of the decision engine.

6.3. Fast approximation

Given a number of objects and a value of resource constraint, we see that the complexity of algorithm 2 will decrease if the number of selections of each object is reduced. For this purpose, one can intuitively omit or merge some neighboring selections of high amounts of resource because the resource changes between those selections are small, compared to their absolute amounts of resource.

In this subsection, we propose an efficient approximation that can be applied to any range of resource amount. We note that in the searching process at each stage, if the current selection has a negligible content value change compared to the last un-omitted selection (called the last *considered selection*), this selection can be omitted. The "omittable" selections are often either very close to other selections or lie in the saturate range of the content value function. So, if we set a minimum *threshold* for the changes of the content value, we can reduce the number of selections considered for each object by omitting the selections having content value changes within the threshold. This threshold technique is illustrated in Fig. 8. Here, selection k is already considered, and selection $k+1$ is omitted because the content value difference is smaller than the threshold. However, selection $k+2$ is considered because the content value difference (w.r.t. selection k) is higher than the threshold.

Denote k^* as the last considered selection, we have the fast approximation algorithm as follows:

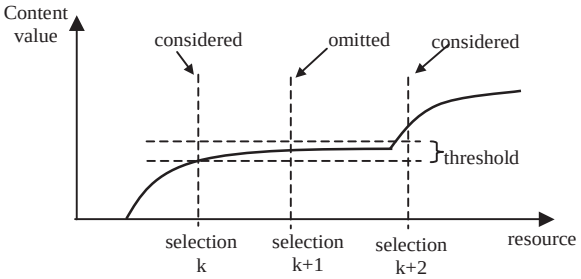

Fig. 8. Illustration of the threshold technique. Selection k is the last considered, selection $k+1$ is omitted because the content value difference is within the threshold. Selection $k+2$ is considered because the content value difference with selection k is higher than the threshold.

Algorithm 3:

Step 0: $i = 0$. Start from the initial node (0, 0)

Step 1: At each stage i, add possible branches to the end nodes of surviving paths. At each node, do the following:

Step 1.1: Add branch for selection $k = 0$

Step 1.2: Check the selection k ($k > 0$): if $|V_{(i+1)k} - V_{(i+1)k*}| >$ threshold and if condition (11) is satisfied, add branch for selection k and let $k^* = k$.

Step 1.3: $k = k + 1$. If $k \leqslant H_i$, go back to Step 1.2, otherwise go to Step 2.

Step 2: Among all paths arriving at a node in stage $i+1$, the one having the highest accumulated sum of $\sum_{t=1}^{i+1} w_t V_t$ is chosen, and the rest are pruned.

Step 3: $I = i + 1$. If $i \leqslant N$ go to Step 1, otherwise go to Step 4.

Step 4: At the final stage, compare all surviving paths then select the path having the highest value of $\sum_{i=1}^{N} w_i V_i$. That path corresponds to the optimal set of selections for all objects.

The difference between algorithms 2 and 3 lies in Step 1. This modified step constantly checks the content value changes and tries to connect a branch only if the content value difference between the current selection and the last considered one is greater than the threshold. This technique is applicable to any range of resource amount because the omittable selections can exist anywhere, especially when the content value function

has several saturate intervals. Let G_i be the number of considered selections for object i, the complexity C' of this fast algorithm is computed similarly to that of algorithm 2, which is given by

$$C' = O\left\{\sum_{i=2}^{N}\left[G_i \min\left(\sum_{m=1}^{i} H_m, D^c\right)\right]\right\}. \quad (13)$$

The value of G_i depends on the threshold. In the experiments part, we will show how good values of the threshold can be empirically obtained.

7. Experiments

We have developed a test bed for providing adaptive multimedia services in heterogeneous environments [15]. The system is compatible with Part 7 (Digital Item Adaptation) of the MPEG-21 standard [14]. Multimedia documents and various metadata (user preference, terminal/network characteristics, etc.) are created in the form of *Digital Items* (DIs) of MPEG-21 and validated against the standardized schemas. The presentation of an adapted document at a terminal is controlled by an XSL (eXtensible Stylesheet Language) stylesheet [17]. As well, modality curves are stored as *utility* functions of MPEG-21 DIA *AdaptationQoS* [14]. Our adaptation engine is integrated onto a Windows 2000 server, whose configuration is a Pentium IV 2.6 GHz with 1 GB RAM.

For our experiments, we employ a multimedia document of six independent objects. Originally, object 1 is a short AV clip; objects 2, 3, 4 and 5 are JPEG images; and object 6 is a text paragraph. The AV object (object 1) has a maximum datasize of 1500 KBs and a length of 8 s. The maximum datasizes of the four image objects are 731, 834, 773, and 813 KBs. And the maximum datasize of the text object is 8 KBs. Upon the user's request, the multimedia document is adapted to meet the datasize constraint and then downloaded to the terminal as a normal Web page. In our system, content transcoding is simply done offline.

The OCV model of each object is obtained by curve-fitting to the empirical data provided by subjective tests. Fig. 9 shows the six OCV models of the six objects. To illustrate clearly the conversion boundaries, we show only the beginning parts of the OCV models. The OCV models in Fig. 9a and b are actually obtained from the operational models depicted in Fig. 3a and b. Note that in the OCV model of Fig. 9a, the audio curve is below the concave hull, so audio modality will be ignored if the Lagrangian method is used. To apply algorithm 2, the content value functions are discretized with the transcoding unit of 1 KB. Actually, the functions can be discretized more sparsely, especially at a high amount of resource, to reduce the number of selections. The default orders of conversions for the objects are described in Table 1; and the default weights of conversions of the objects are 1. The importance values w_i's of the six objects are set to be 1, 0.55, 0.5, 0.6, 0.55, and 0.15. These values are subjectively decided based on the relative importance of the objects in the document.

7.1. Experiments on the adaptation to the datasize constraint and the user preference

To check the response of the adaptation system, we first use the default user preference and vary the datasize constraint D^c. Each row of Table 2 shows one document version adapted to a value of D^c. In this table, the first column is D^c; each object has two columns—one for the datasize and the other for the modality; and the last column is the total content value of the adapted document. We can see that as D^c decreases, the datasizes of the objects are reduced to satisfy the datasize constraint of the whole document. Also, at certain points, the modalities of the objects are converted to meet the constraint and to give the highest possible total content value. The conversions of the objects follow the default orders of conversion. For example, as D^c decreases, object 1 has modalities of AV, AI, audio, and text in succession.

We also carry out an experiment without modality conversion, in which the OCV model of each object contains only the curve of the original modality. The result is shown in Table 3. We see that when modality conversion is not applied, some objects may be discarded as the resource constraint is reduced. In contrast, in Table 2, the

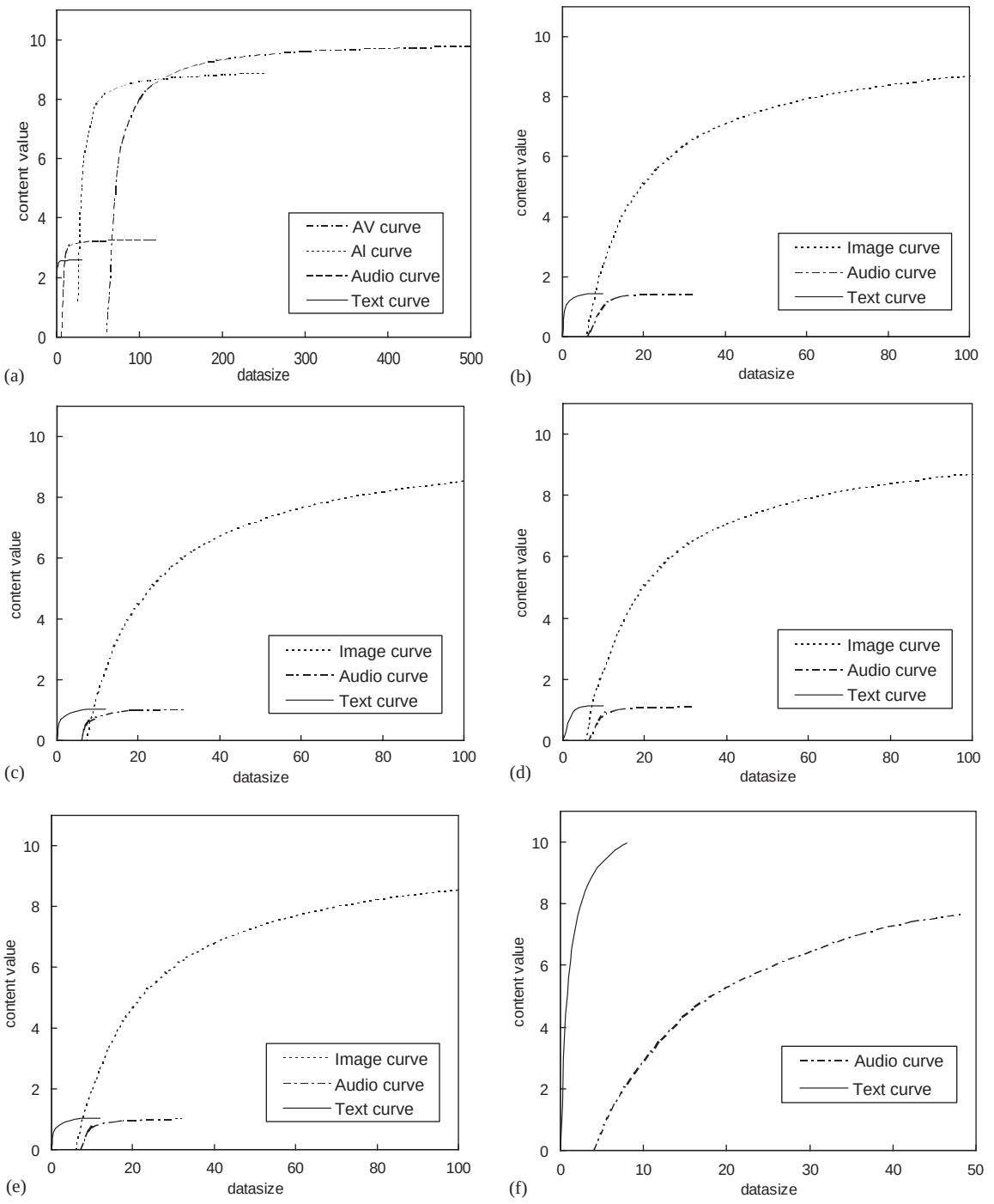

Fig. 9. The OCV models for the six objects of the experimental document: AV object (a); image objects (b), (c), (d), and (f); text object (f).

Table 2
Results of adapted documents with modality conversion and with default user preference

D^c (KBs)	Object 1		Object 2		Object 3		Object 4		Object 5		Object 6		Content value
	D_1 (KBs)	Mod	D_2 (KBs)	Mod	D_3 (KBs)	Mod	D_4 (KBs)	Mod	D_5 (KBs)	Mod	D_6 (KBs)	Mod	
3000	701	AV	585	I	562	I	588	I	556	I	8	T	33.37
1000	264	AV	186	I	178	I	187	I	177	I	8	T	31.93
600	176	AV	107	I	102	I	106	I	101	I	8	T	30.44
500	85	AI	104	I	100	I	104	I	99	I	8	T	29.76
100	41	AI	22	I	15	I	16	I	2	T	4	T	18.42
70	39	AI	21	I	2	T	2	T	2	T	4	T	15.12
40	12	A	18	I	2	T	2	T	2	T	4	T	11.77
10	2	T	2	T	1	T	1	T	1	T	3	T	7.76

Each row shows one version of document with a corresponding value of datasize constraint D^c in the first column. Here, *Mod* means modality and *AV*, *AI*, *I*, *A*, *T* mean "audio + video", "audio + image", image, audio, and text modalities respectively.

Table 3
Results of adapted documents without modality conversion.

D^c (KBs)	Object 1		Object 2		Object 3		Object 4		Object 5		Object 6		Content value
	D_1 (KBs)	Mod	D_2 (KBs)	Mod	D_3 (KBs)	Mod	D_4 (KBs)	Mod	D_5 (KBs)	Mod	D_6 (KBs)	Mod	
3000	701	AV	585	I	562	I	588	I	556	I	8	T	33.37
1000	264	AV	186	I	178	I	187	I	177	I	8	T	31.93
600	176	AV	107	I	102	I	106	I	101	I	8	T	30.44
500	154	AV	87	I	82	I	87	I	82	I	8	T	29.68
100	0		26	I	22	I	24	I	24	I	4	T	14.10
70	0		18	I	16	I	17	I	16	I	3	T	11.19
40	0		13	I	12	I	13	I	0		2	T	7.23
10	0		0		0		8	I	0		2	T	2.31

As the datasize constraint is reduced, some objects are allocated with zero amount of resource, i.e. the objects are removed.

objects are converted to other modalities to maintain some meaning for the user. And the consequence is that when modality conversion is applied, the total content value of the adapted document is always higher than, or equal to, the case where only content scaling is employed. For example, when $D^c = 70$ KBs, the total content value of the adapted document in Table 1 is 15.12, compared to the corresponding value of 11.19 in Table 3. In Fig. 10, this behavior is depicted over a large interval of constraint value. It can be seen that the advantage of modality conversion becomes clear as the datasize constraint is reduced to below 500 KBs.

From Table 2, we note that as D^c is reduced, object 1 is converted to AI, to audio, and finally to text modality as a result of default orders of conversions. Now, the user may prefer that, after the AI modality, object 1 should be converted to text modality. Using the modality conversion preference, the user changes the order of AV-to-audio to "fourth", and the order of AV-to-text to "third". Fig. 11 shows the modified OCV model of object 1, where audio curve is removed (as in the example in Section 4.2.2). The corresponding adapted documents are described in Table 4, in which the user's need is obviously satisfied. Now, as object 1 is converted, it will be converted to AI and then to text modality.

In addition, we see in Table 2 that object 1 is converted to AI modality when D^c is reduced to 500 KBs. Suppose that the user still wishes to watch the video modality beyond this constraint value, then the user increases the weight of the

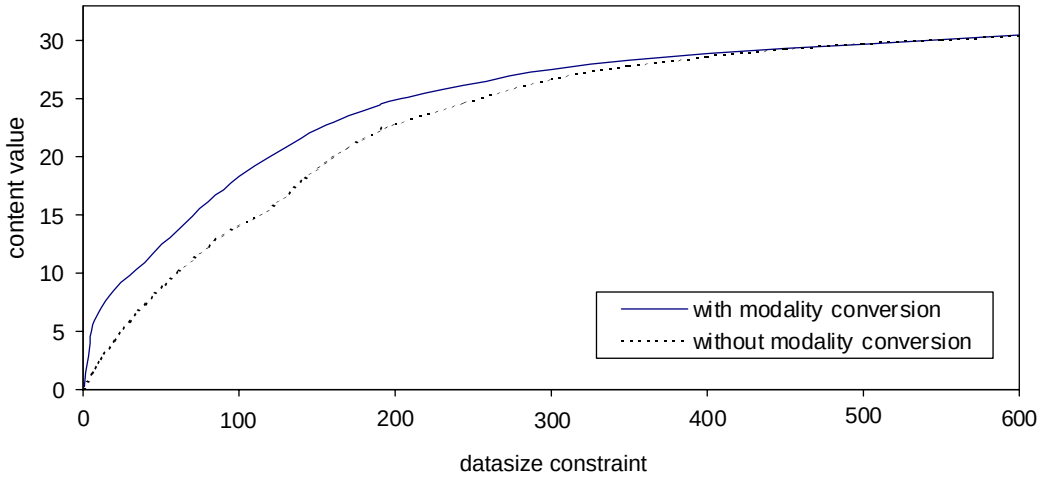

Fig. 10. The total content value of the adapted document w.r.t. the datasize constraint. In the low-value range of constraint, the adaptation with modality conversion provides higher content value than the adaptation without modality conversion.

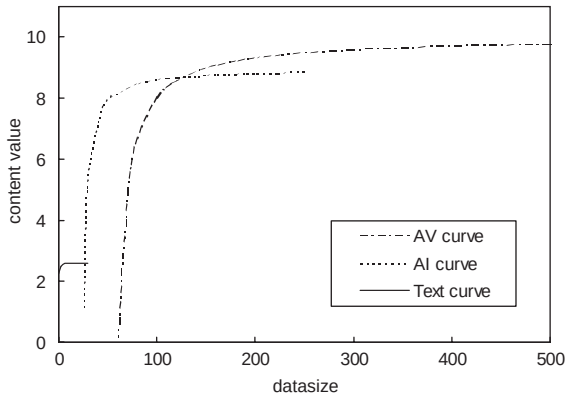

Fig. 11. The modified OCV model of object 1 (AV object) when the order of AV-to-text conversion is "third" and the order of AV-to-audio conversion is "fourth". According to algorithm 1, the audio curve is now removed.

AV-to-AV conversion to 2 for example. Fig. 12 shows the corresponding modified OCV model of object 1, where the conversion boundary of AV-to-AI is now at 97 KBs, compared to 125 KBs in Fig. 9a. Table 5 shows the adaptations subject to this user preference. We see that now the AV modality, obviously with a lower quality (datasize of 111 KBs), is retained even at $D^c = 300$ KBs. However, this tradeoff between modality and quality is acceptable to the user.

7.2. Experiments on the performance of the decision engine

The above experiments show that the system adapts effectively to different conditions of the datasize constraint as well as the user preference. Now, using the default user preference, we check the performance, including the processing time and the optimality, of the decision engine. First we employ algorithm 2 for the decision engine. The continuous lines in Fig. 13a and b show respectively the total content value of the adapted document and the processing time (to find the optimal solution) versus different values of D^c. We see that, using algorithm 2 (threshold = 0), the processing time of the decision engine is smaller than 0.3 s, which is good for the real-time requirement. This result is actually due to the fact that there are only six objects in the document.

Next, to reduce the processing time, we try to apply algorithm 3. Before that, we need to estimate a "good threshold" by considering the content value and the processing time versus different thresholds, given some fixed values of the datasize constraint. These relationships are depicted in Fig. 14 for three cases, $D^c = 4500$, 2500 KBs, and 700 KBs. We see that the total content value is reduced just a little even when the threshold is

Table 4
Results of adapted documents when the user's order of AV-to-text conversion is "third" and the user's order of AV-to-audio conversion is "fourth".

D^c (KBs)	Object 1		Object 2		Object 3		Object 4		Object 5		Object 6		Content value
	D_1 (KBs)	Mod	D_2 (KBs)	Mod	D_3 (KBs)	Mod	D_4 (KBs)	Mod	D_5 (KBs)	Mod	D_6 (KBs)	Mod	
600	176	AV	107	I	102	I	106	I	101	I	8	T	30.44
500	85	AI	104	I	100	I	104	I	99	I	8	T	29.76
100	41	AI	22	I	15	I	16	I	2	T	4	T	18.42
70	37	AI	14	I	1	T	13	I	1	T	4	T	15.11
40	2	AI	16	I	1	T	16	I	1	T	4	T	11.75
10	2	T	2	T	1	T	1	T	1	T	3	T	7.76

Object 1 is converted to AI modality and then to text modality as D^c is reduced. Note that in Table 2, object 1 is converted to audio modality before being converted to text modality.

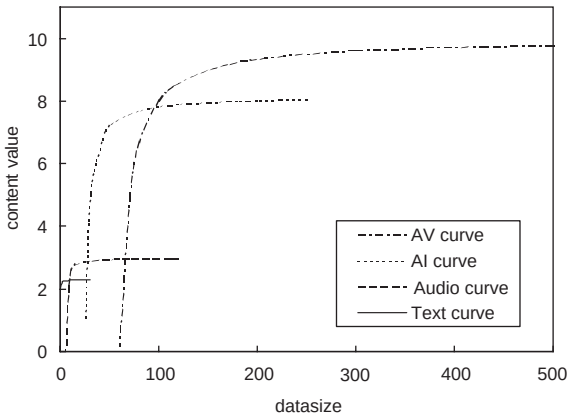

Fig. 12. The modified OCV model of object 1 (AV object) when the weight of AV-to-AV conversion is 2. Now, the conversion boundary of AV-to-AI is at 97 KBs, compared to 125 KBs in Fig. 9a.

increased up to 0.2. As well, the threshold seems to have more effect when D^c is higher. This is because with the small values of D^c (e.g. $D^c = 700$ KBs), the datasizes of adapted objects lie in small-value ranges. In these ranges, the slope of a content value function is often high, or the content value differences of adjacent selections are often larger than the thresholds. This suggests that when the resource constraint is low, the threshold technique may be unnecessary. From Fig. 14 we can guess that, when the threshold is around 0.02, the complexity may be reduced to one tenth while the total content value remains nearly the same.

The dashed lines in Fig. 13a and b show the performance of the decision engine using algorithm 3 with threshold = 0.02. Now we can see that the processing time is much reduced, i.e. below 0.03 s compared to 0.3 s of algorithm 2; meanwhile, the total content value decreases very little.

In previous experiments, there were only six objects in the document. Now we add some more image objects, which are practically of the most popular modality, to see how the processing time depends on the number of objects. The maximum datasize of each added image object is 900 KBs. The datasize constraint is now set to be rather high, at $D^c = 5000$ KBs. Fig. 15 shows the relationship of the processing time versus the number of objects for various cases. When the transcoding unit is 1KBs, we have two cases: threshold = 0 (algorithm 2) and threshold = 0.02 (algorithm 3). We see that when the number of objects is more than 20, the processing time of algorithm 2 is a little high (over 2 s), but the result of the fast approximation algorithm is very interesting. With threshold = 0.02, the processing time for the document of as many as 30 objects is still below 0.5 s. Fig. 15 also illustrates the case where the threshold is 0 but the transcoding unit is doubled (2 KBs). In this case, the processing time is also significantly reduced (just about 1 s). This shows the potential of low processing time when the content value functions are more sparsely discretized.

Table 5
Results of the adapted documents when the weight of AV-to-AV conversion is 2.

D^c (KBs)	Object 1		Object 2		Object 3		Object 4		Object 5		Object 6		Content value
	D_1 (KBs)	Mod	D_2 (KBs)	Mod	D_3 (KBs)	Mod	D_4 (KBs)	Mod	D_5 (KBs)	Mod	D_6 (KBs)	Mod	
600	176	AV	107	I	102	I	106	I	101	I	8	T	30.44
500	154	AV	87	I	82	I	87	I	82	I	8	T	29.68
300	111	AV	47	I	44	I	47	I	44	I	7	T	26.50
100	36	AI	21	I	19	I	20	I	1	T	3	T	17.21
70	12	A	18	I	16	I	20	I	1	T	3	T	13.96
40	1	T	17	I	1	T	17	I	1	T	3	T	10.74
10	1	T	1	T	1	T	1	T	1	T	5	T	7.33

Now, object 1 retains its AV modality even at $D^c = 300$ KBs. Note that in Table 2 the object 1 is converted to AI modality at $D^c = 500$ KBs.

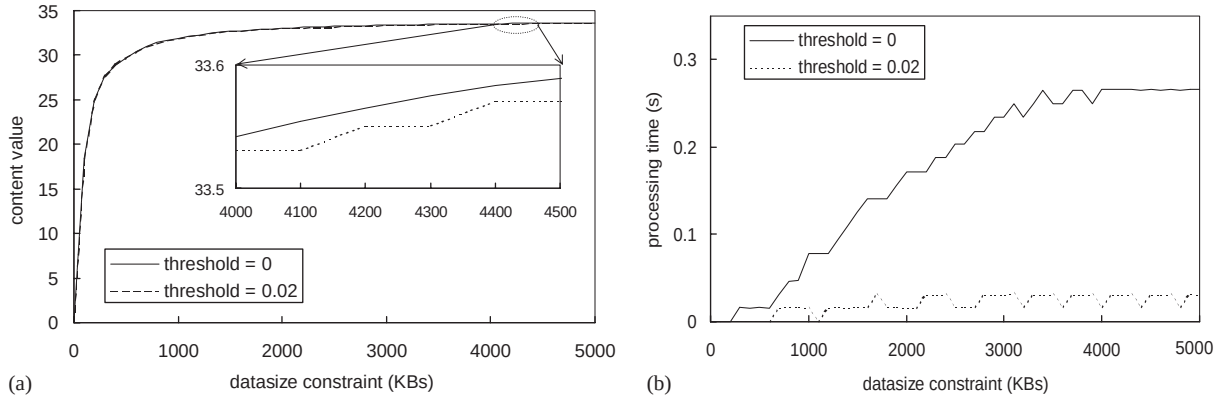

Fig. 13. Performance comparison of the decision engines using algorithm 2 (i.e., threshold = 0) and algorithm 3 (threshold = 0.02). (a) shows the total content value of the adapted document and (b) shows the processing time with respect to different values of datasize constraint. We see that when threshold = 0.02, the processing time is much reduced while the total content value is nearly the same.

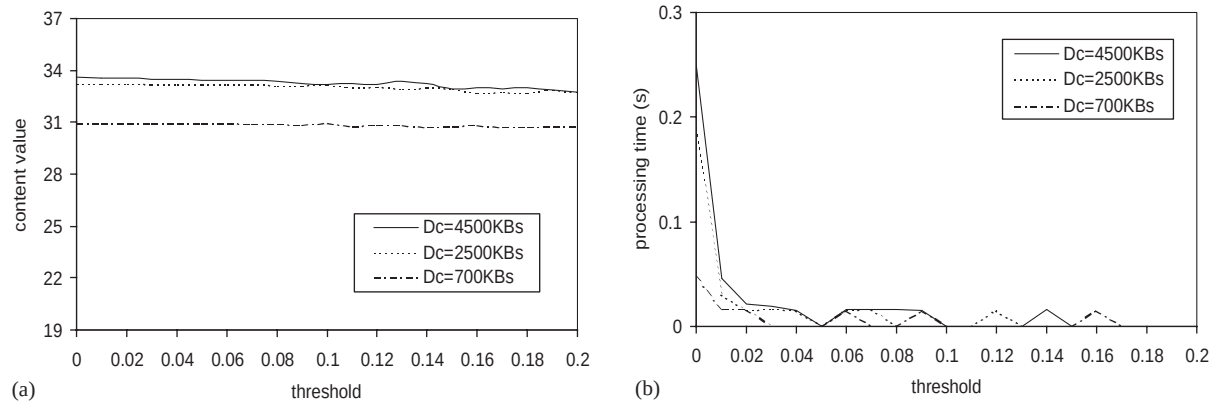

Fig. 14. The total content value of the adapted document (a) and the processing time (b) vs. threshold. We can see that when threshold is 0.02, the processing time can be reduced to one tenth while the total content value changes just a little.

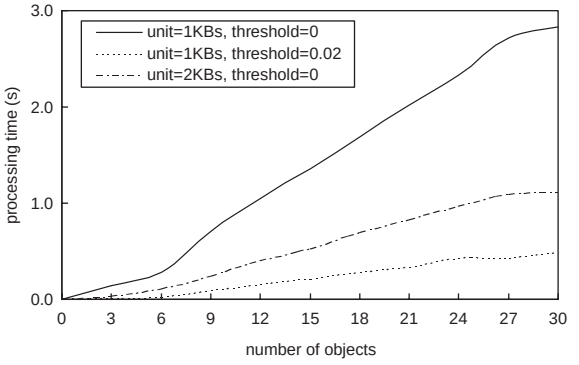

Fig. 15. Dependence of the processing time vs. the number of objects. With transcoding unit = 1 KBs and threshold = 0.02, the processing time of a 30 object document is still below 0.5 s; while with trancoding unit = 2 KBs and threshold = 0, the processing time is just about 1 s.

7.3. Discussions

The experiments quantitatively verify that, to deliver multimedia documents under the low resource constraint, using modality conversion in combination with content scaling would bring more benefit to the user than using content scaling only. However, building the OCV models in a cost-effective way is necessary for the preparation of content descriptions. Besides the subjective test, the perceptual quality and semantic quality of certain modalities (e.g., video, image) can be estimated using computational methods. For example, the perceptual quality of image and video can be reliably estimated using Human Visual System based methods [41]. The content-based framework [18] is also very promising to estimate the multidimensional quality of multimedia contents. Currently, we are studying model-based methods to obtain the semantic quality for domain-specific applications.

As to the modality conversion preference, the *order* of conversion is rather straightforward for the end user to specify; while the *weight* of conversion is not so easy. However, if a provider in the content delivery chain (considered as a "User" in the context of MPEG-21 multimedia framework [14]) has the OCV models in advance, the provider can make use of this parameter to modify the OCV models according to his/her QoS policy. Besides that, it is possible to learn end user behavior and then estimate the user preference. This is an interesting topic for our future work.

The threshold technique in essence aims at reducing the number of considered selections by omitting the "negligible" ones. In practice, the thresholds can be empirically estimated in advance and one or more thresholds can be stored as parameters of content description of the document. The number of selections can be further reduced by various techniques. For example, the threshold may be adaptive to the slope of content value function, so as to remove more selections when possible. In fact, the selections on a content value function may be very sparsely spaced. For example, the number of scaling levels using requantization for video and image often varies from 1 to 31. Furthermore, a human normally is able to differentiate just a small number of quality levels, regardless of the number of resources. As such, the number of practical selections for each object is not really high (about several hundreds or dozens from our experience), which is not burdensome for the decision engine even in the multiple resource case.

Moreover, a special point is that the number of objects to be transcoded in a multimedia document is not many, normally not more than several dozens. This is actually a major reason for the potential real-time support of the decision engine. In addition, there are some simple techniques to reduce the number of objects in a document. For example, by analyzing the document or using the content description, we can discard the semantically less important objects [30], or we can heuristically select the large objects to be transcoded, leaving the small objects intact. As well, some similar objects in the document, e.g. all images of the sports genre, can be grouped into a combined object. Taking advantage of these practical techniques, the Viterbi algorithm can be applied to the decision engine to solve in real-time the complex problem of content adaptation.

8. Conclusion

In this paper, we have proposed a systematic approach to tackle modality conversion, an important aspect in adapting multimedia documents. We stressed that the decision engine is the most

important part of an adaptation engine, for both online and offline transcodings. The mechanism of the decision engine was then formulated by an extended framework supporting a wide context of content adaptation, including the content descriptions across different modalities, the user preference, and the modality capability. We proposed the overlapped content value model to harmonize different modalities of a content object. The modality conversion preference was used to give users more freedom in customizing the modalities of adapted contents. The Viterbi algorithm of dynamic programming was employed to optimally allocate resource to different content objects. Our experiments demonstrated that modality conversion helps increase the overall content value of adapted multimedia documents under limited resource availability. The combination of the overlapped content value model, the modality conversion preference, and the dynamic programming method, was a crucial point in making efficient decisions on modality conversion as well as content scaling, so as to meet various conditions of the resource constraint and the user preference.

Our future work will focus on the automation of online transcoding, in terms of both content scaling and modality conversion. In addition, when the content objects are adapted, their content descriptions should also be modified. This modification, called "metadata adaptation", will be explored and deployed in our system in the future.

References

[1] D. Archambault, D. Burger, From Multimodality to multimodalities: the need for independent models, in: Proceedings of the Universal Access in Human-Computer Interaction, 2001, pp. 227–231.

[2] M.K. Asadi, J.-C. Dufourd, Multimedia adaptation by transmoding in MPEG-21, in: Proceedings of the International Workshop on Image Analysis for Multimedia Interactive Services, 2004.

[3] V. Balasubramanian, N. Venkatasubramanian, Server transcoding of multimedia information for cross disability access, in: Proceedings of the ACM/SPIE Conference on Multimedia Computing and Networking, 2003.

[4] R. Bandelloni, S. Berti, F. Paternò, Mixed-initiative, transmodal interface migration, in: Proceedings of the Mobile HCI 2004 (MHCI04), Lecture notes in Computer Science, vol. 160. Springer, pp. 216–227.

[5] P. Batra, Modeling and efficient optimization for object-based scalability and some related problems, IEEE Trans. Image Processing 9 (Ocotber 2000) 1677–1692.

[6] S. Boll, W. Klas, J. Wandel, A cross-media adaptation strategy for multimedia presentations, in: Proceedings of the ACM Multimedia'99, 1999.

[7] K.-A. Cha, S. Kim, Adaptive scheme for streaming MPEG-4 contents to various devices, IEEE Trans. Consumer Electronics 49 (November 2003) 1061–1066.

[8] S.-F. Chang, A. Vetro, Video adaptation: concepts technologies and open issues, Proceedings of the IEEE 93 (January 2005) 148–158.

[9] J. Chen, Y. Yang, H. Zhang, An adaptive Web content delivery system, in: Proceedings of the International Conference on Adaptive Hypermedia and Adaptive Web-based Systems, 2000, pp. 284–288.

[10] L. Christianson, K. Brown, Rate adaptation for improved audio quality in wireless networks, in: Proceedings of the Sixth IEEE International Workshop on Mobile Multimedia Communications, 1999, pp. 15–17.

[11] C. Elting, J. Zwickel, R. Malaka, Device-dependent modality selection for user-interfaces—an empirical study, in: Proceedings of the International Conference on Intelligent User-Interfaces, 2002, pp. 55–62.

[12] G.D. Forney, The Viterbi algorithm, Proc. IEEE 61 (March 1973) 268–278.

[13] ISO/IEC IS 15938-5:2001, 'Information Technology—Multimedia Content Description Interface—Multimedia Description Schemes', 2003.

[14] ISO/IEC IS 21000-7, 'Information Technology—Multimedia Framework—Part 7: DIA', 2004.

[15] Y.J. Jung, T.C. Thang, J. Lee, Y.M. Ro, Visual media adaptation system for active media, in: Proceedings of the 2003 International Conference on Imaging Science Systems and Technology 2003.

[16] A. Kaup, Video analysis for universal multimedia messaging, in: Proceedings of the Fifth IEEE Southwest Symposium on Image Analysis and Interpretation, 2002, 211–215.

[17] C. Kerer, E. Kirda, M. Jazayeri, R. Kurmanowytsch, Building and managing XML/XSL-powered Web sites: an experience report, in: Proceedings of the International Computer Software and Applications Conference 2001, pp. 547–554.

[18] J.-G. Kim, Y. Wang, S.-F. Chang, Content-adaptive utility based video adaptation, in: Proceedings of the International Conference on Multimedia & Expo (ICME), 2003.

[19] M.B. Kim, J. Nam, W. Baek, J. Son, J. Hong, The adaptation of 3D stereoscopic video in MPEG-21 DIA, Signal Process.: Image Commun. 18 (2003) 685–697.

[20] C. Lee, J. Lehoczky, D. Siewiorek, R. Rajkumar, J. Hansen, A scalable solution to the multi-resource QoS problem, in: Proceedings of the 20th IEEE Real-Time Systems Symposium, 1999.

[21] H.-C. Lee, S.-D. Kim, Iterative key frame selection in the rate-constraint environment, Signal Process.: Image Commun. 18 (2003) 1–15.

[22] K. Lengwehasatit, A. Ortega, Rate-complexity-distortion optimization for quadtree-based DCT coding, in: Proceedings and Conference on Image Processing, 2000, pp. 821–824.
[23] W.Y. Lum, F.C.M. Lau, A QoS-sensitive content adaptation system for mobile computing, in: Proceedings of the Computer Software and Applications Conference 2002, pp. 680–685.
[24] R. Mohan, J.R. Smith, C.-S. Li, Adapting multimedia internet content for universal access, IEEE Trans. Multimedia 1 (March 1999) 104–114.
[25] K. Nagao, Y. Shirai, K. Squire, Semantic annotation and transcoding: making Web content more accessible, IEEE Multimedia 8 (2001) 69–81.
[26] J.-R. Ohm, Advances in scalable video coding, Proceedings of the IEEE 93 (2005) 42–56.
[27] A. Ortega, K. Ramchandran, M. Vetterli, Optimal trellis-based buffered compression and fast approximations, IEEE Trans. Image Processing 3 (January 1994) 26–40.
[28] A. Ortega, K. Ramchandran, Rate-distortion methods for image and video compression, IEEE Signal Processing Magazine (November 1998) 23–50.
[29] G. Panis, et al., Bitstream Syntax Description: a tool for multimedia resource adaptation within MPEG-21, Signal Process.: Image Commun. 18 (2003) 721–747.
[30] J.R. Smith, R. Mohan, C.-S. Li, Content-based transcoding of images in the Internet, in: Proceedings of the IEEE ICIP'98, 1998, pp. 7–11.
[31] Text of ISO/IEC 21000-7 P/DAM-1, ISO/IEC JTC 1/SC 29/WG 11/N6776, 2004.
[32] T.C. Thang, Y.J. Jung, Y.M. Ro, Modality conversion for QoS management in Universal Multimedia Access, IEE Proceedings—Vision, Image and Signal Processing, in press.
[33] T.C. Thang, Y.J. Jung, Y.M. Ro, Modality conversion in content adaptation for Universal Multimedia Access, in: Proceedings of the International Conference on Imaging Science, Systems, and Technology, 2003, pp. 434–440.
[34] T.C. Thang, Y.J. Jung, J.W. Lee, Y.M. Ro, Modality conversion for Universal Multimedia Services, in: Proceedings of the International Workshop on Image Analysis for Multimedia Interactive Services, 2004.
[35] T.C. Thang, Y.M. Ro, Visual content adaptation for low vision users in MPEG-21 framework, in: Proceedings of the IEEE ICIP2004, 2004.
[36] T.C. Thang, Y.M. Ro, "Multimedia quality evaluation across different modalities", in: Proceedings of the SPIE Electronic Imaging, vol. 5668, 2005, pp. 270–279.
[37] A. Vetro, C. Christopoulos, H. Sun, An Overview of Video Transcoding Architectures and Techniques, IEEE Signal Process. Mag. (March 2003) 18–29.
[38] A. Vetro, A. Divakaran, H. Sun, Adaptable compressed bitstream transcoder, US Patent No. 6,542,546, 2003.
[39] W3C: Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies 1.0 W3C Recommendation, 2004, http://www.w3.org/TR/2004/REC-CCPP-struct-vocab-20040115/.
[40] C.-H. Wu, J.-H. Chen, Speech activated telephony email reader (SATER) based on speaker verification and text-to-speech conversion, IEEE Trans. Consumer Electronics 43 (1997) 707–716.
[41] Z. Yu, H.R. Wu, S. Winkler, T. Chen, Vision-model-based impairment metric to evaluate blocking artifacts in digital video, Proc. IEEE 90 (2002) 154–169.

Truong Cong Thang received the BS and MS degrees in Electrical Engineering from Hanoi University of Technology, Vietnam, in 1997 and 2000 respectively. From 1997 to 1999, he was a satellite engineer of Vietnam Telecom International. From 1999 to 2000, he worked as a project engineer in Vietnam Datacommunications Company. In 2001, he entered Information and Communication University, Korea, where he is currently a Ph.D. candidate in Image and Video System Lab. His research interests include image/video processing, video streaming, content adaptation, MPEG-21, video abstraction.

Yong Ju Jung received the B.S. degree from Hongik University, Seoul, Korea in 1999, and the M.S. degree in August 2000 from the Information and Communications University (ICU), Daejeon, Korea, where he is currently a Ph.D. candidate in Image Video System Lab. In 1999, he was a research intern at DACOM, and from 2000 to 2001 he was a researcher at IVSystem. His research interests include image/video processing, content adaptation, scalable video coding, MPEG-21, and data hiding. He received the Best Paper Award from the Joint Conference on Multimedia 2000, Korea in 2000.

Yong Man Ro received the BS from Yonsei University, Seoul, Korea, in 1981 and the MS and Ph.D. degrees from the Korea Advanced Institute in Science and Technology (KAIST), in 1987 and 1992, respectively. In 1987, he was a staff associate at Columbia University, and from 1992 to 1995, he was a visiting researcher in University of California at Irvine and KAIST. In 1996, he was a research fellow at Department of EECS in University of California at Berkeley. In 1997 he joined Information and Communication University, Korea where he is currently associate professor and director of Image Video System Lab. His research interests include image/video processing, MPEG-7, MPEG-21, feature recognition, image/video indexing, and spectral analysis of image signal. He received the Young Investigator Finalist Award in ISMRM in 1992. He is a Senior member of IEEE and member of SPIE and ISMRM.