Scalable Multimedia Delivery for Pervasive Computing

John R. Smith, Rakesh Mohan and Chung-Sheng Li

IBM T.J. Watson Research Center
30 Saw Mill River Road
Hawthorne, NY 10532
{jrsmith,rakesh,csli}@watson.ibm.com

Abstract

Growing numbers of pervasive devices are gaining access to the Internet and other information sources. However, much of the rich multimedia content cannot be easily handled by the client devices with limited communication, processing, storage and display capabilities. In order to improve access, we are developing a system for scalable delivery of multimedia. The system uses an InfoPyramid for managing and manipulating multimedia content composed of video, images, audio and text. The InfoPyramid manages the different variations of media objects with different fidelities and modalities and generates and selects among the alternatives in order to adapt the delivery to different client devices. We describe a system for scalable multimedia delivery for a variety of client devices, including PDAs, HHCs, smart phones, TV browsers and color PCs.

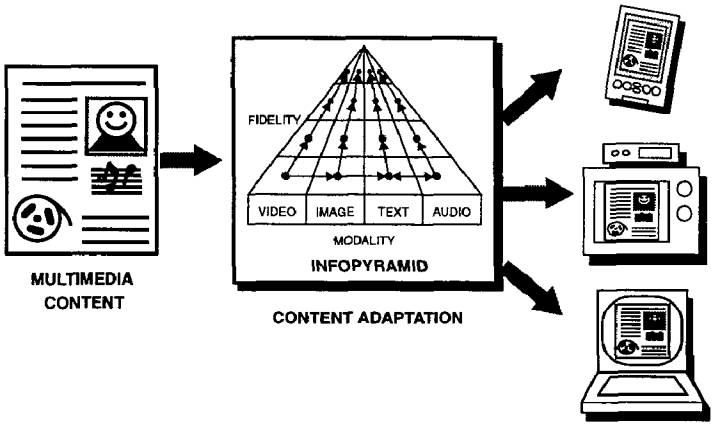

Figure 1: Scalable multimedia content delivery to pervasive computing devices.

1 INTRODUCTION

An increasing amount of electronic information takes the form of multimedia – an integration of images, video, graphics, audio and text. Many technologies are being developed for compression, indexing, searching and filtering in order to better manage multimedia data. However, as depicted in Figure 1, enabling effective multimedia content access for a wide diversity of client devices is becoming one of the important emerging problems in the generation of pervasive computing.

1.1 Pervasive computing

New classes of pervasive computing devices such as personal digital assistants (PDAs), hand-held computers (HHC), smart phones, automotive computing devices, and wearable computers allow users more ubiquitous access to information than ever. Many of the devices have capabilities of serving as calendar tools, address books, pagers, global positioning devices, travel and mapping tools, email clients, and Web browsers.

As users are beginning to rely more heavily on pervasive computing devices, there is a growing need for applications to bring multimedia information to the devices. However, due to limited device capabilities – in terms of the display size, storage, processing power, and network access – there are new challenges for designing the applications that allow these devices to effectively access, store and process multimedia information. Concurrent with the developments in pervasive computing, advances in storage, networking and authoring tools are driving the production of large amounts of rich multimedia content. The result is a growing *mismatch* between the available rich content and the capabilities of the client devices to access and process it. The mismatch between content requirements and client capabilities impacts a number of applications including Internet access and Web browsing, multimedia presentation and digital libraries.

1.2 Internet access

Several recent projects have focussed on improving Internet access for pervasive devices (for example, see [1, 2, 3, 4]). Fox, et al., developed a system for compressing Internet content at a proxy in order to deal with client variability and improve end-to-end performance [5]. Ortega, et al., developed an image caching method for Internet proxies, which reduces the resolution of infrequently accessed images in order to conserve storage space and bandwidth [6]. Previously, we investigated methods for on-line transcoding images [7] and general Internet content [8]. Several recent commercial systems have appeared, such as Intel's Quick Web [9]

and Spyglass' Prism [10], that compress Internet content to speed-up download time.

In the area of content authoring, special markup languages, such as WML ([11]), SMIL ([12]), and multimedia formats, such as MPEG-4 ([13]), are being developed with pervasive devices or content adaptation in mind. WML is suited for developing text-only applications for devices such as pagers and cell phones. SMIL provides a language for creating synchronized multimedia presentations. SMIL supports client adaptation by providing switch statements, such as for high and low bandwidth. By representing video streams by collections of individual and independent media objects, MPEG-4 allows for better adaptation of the delivery of video to different bandwidth conditions.

1.3 Multimedia presentation

In many multimedia presentations, maintaining synchronization among the individual media objects is important. Synchronization presents a problem when the media objects are distributed and/or need to be transcoded. Candan, et al [14], and Vogel, et al [15], investigated synchronized presentation issues for distributed multimedia documents. Candan included modeling parameters for maintaining synchronization under transcoding, such as image conversion between JPEG and GIF formats. Previously, we investigated methods for adapting multimedia presentations to client devices by optimizing media object selection for the device constraints [16]. Weitzman, et al, investigated grammars for dynamic presentation and layout of multimedia documents under different presentation conditions [17].

1.4 Universal access

Universal access is an important functionality of digital libraries [18, 19]. Since digital libraries are designed to serve large numbers of diverse users, they need to accommodate the special needs of the users, constraints of the client devices, and bandwidth conditions. In some cases, the users, such as those that are hearing or sight impaired, need speech-only or text-only access to the digital library. To provide universal access, the digital libraries need to either store and deliver different variations of the content, or perform media conversion before content delivery. Recently, proposals have been made to MPEG-7 to establish standard meta-data to assist in media conversion for universal access [20, 21, 22].

1.5 InfoPyramid

In order to improve the accessibility of multimedia content by pervasive computing devices, we are developing a scalable multimedia delivery system, as shown in Figure 1. We represent the media objects that constitute the multimedia documents using an InfoPyramid data model, as shown in Figure 2. The InfoPyramid manages the different variations of the media objects with different modalities and fidelities [23]. The scalable multimedia delivery is then enabled by either

1. storing, managing, selecting, and delivering different variations of the media objects in the InfoPyramid in order to adapt the multimedia documents to the client devices [16], or
2. manipulating the media objects on-the-fly, such as by using methods in the InfoPyramid for text-to-speech translation, image transcoding and summarization [8].

This allows the multimedia content delivery to adapt to the wide diversity of client device capabilities for communication, processing, storage, and display.

1.6 Outline

In this paper, we describe the scalable system for delivering multimedia content, as follows: in section 2, we present the InfoPyramid multimedia data model and describe the processes for media object management and manipulation. In section 3, we present a system for adaptive delivery of multimedia content by optimizing the selection of the different variations of the media objects. In section 4, we present a system for dynamically transcoding media objects by automatically routing through the content manipulation methods. Finally, in section 5, we describe the deployment of a scalable multimedia delivery transcoding proxy system to enable Internet access for pervasive computing devices.

2 INFOPYRAMID FRAMEWORK

The InfoPyramid provides a general framework for managing and manipulating media objects. As depicted in Figure 2, the InfoPyramid manages different variations of media objects with different *modalities* (video, image, text, and audio) and *fidelities* (summarized, compressed, and scaled variations) [23]. The InfoPyramid also provides and manages the *translation* and *summarization* methods that generate the different variations of the media objects.

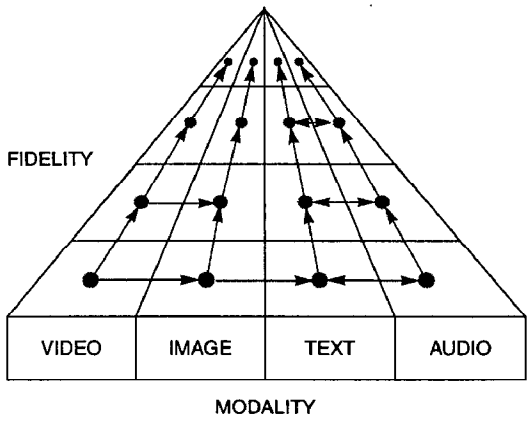

Figure 2: The InfoPyramid provides a framework for managing multiple variations of media objects and for manipulating them using methods for translation and summarization.

Each media object is represented by a cell in the InfoPyramid. For example, in Figure 2, the cell in the lower-left corner of the InfoPyramid corresponds to a high-resolution video. The cells above this one in the video column correspond to the lower-resolution or compressed alternatives (lower fidelity) of the video. The cells to the right correspond to the different image, text and audio alternatives (different modalities) of the video sequence. Here, the notion of fidelity refers to a subjective score and not necessarily a quantitative measure such as signal-to-noise ratio (SNR).

On the other hand, the cell in the bottom of the text column corresponds to a full-detailed body of text. The cells above it in the text column correspond to the summarized and compressed alternatives (lower fidelity) of the text body.

The cells in the audio column correspond to different variations of the text rendered as audio (different modality), such as by text-to-speech conversion.

2.1 Translation and summarization

Manipulation operations for media objects in the InfoPyramid alter their modality and fidelity. The *translation* methods convert the media objects to different modalities, such as text to audio, or video to images. The *summarization* methods generate different variations within the same modality, but with different fidelity. For example, the summarization methods compress the images, summarize text, and generate video abstractions. The translation and summarization methods can be cascaded to change both the modality and fidelity of the media objects, as shown in Figure 2.

2.2 Other variations

In general, other variations of the multimedia objects can serve as substitutions for the purpose of scalable delivery. This notion of variations has been proposed to MPEG-7 for applications relating to universal multimedia access. In [22], the variations refer to relationships such as translation, summarization, extraction, scaling, visualization and substitution. Each variation relationship is assigned a fidelity score which indicates how well it substitutes for the original.

2.3 InfoPyramid data model

The InfoPyramid provides a complete data model for managing and manipulating the media objects. The InfoPyramid data model consists of classes for the different modalities and fidelities (data) and transcoders (methods). The data model distinguishes between the two types of transcoders – translators and summarizers. The data model is extensible in that, initially, we define four modalities – video, image, text and audio.

Other modalities can be added, such as 3-D graphic models and text languages. We initially define several transcoders such as text-to-speech conversion, image transcoding, video transcoding, video-to-image key-frame extraction and text summarization. The data model can also be extended by adding new transcoders by deriving from the translator or summarizer classes.

2.4 Multimedia document layout and synchronization

Since the InfoPyramid provides only a data model for managing the constituent media objects in a multimedia document, it does not itself address the issues of the multimedia document layout and synchronization. Several languages are available for specifying the multimedia document layout, such as HTML or SMIL [12]. The multimedia presentation specifies the spatial and temporal location, size and duration of each media object in the multimedia document.

Since the InfoPyramid provides a way to substitute and transcode the media objects, the document layout may need to be modified to accommodate the different combinations of media objects. However, in this paper we assume that each media object has an initial relative location and duration in space and time, as determined by the document layout. When we substitute different variations of the media objects in the multimedia presentation, we do not alter the relative location and size of the media object.

3 ADAPTIVE DELIVERY

We consider the two forms of scalable multimedia delivery to pervasive devices – adaptive delivery and on-line transcoding. In adaptive delivery, the content server uses the InfoPyramid to manage and select the different variations of the media objects. When a client device requests a multimedia document, the server selects and delivers the most appropriate variation of each of the media objects. The selection can be made on the basis of the capabilities of the client devices, such as display size, display color depth, network bandwidth, client storage and so forth, as we describe next. We describe the InfoPyramid on-line transcoder later in Section 4.

3.1 Content selection

We provide a procedure for optimizing the selection of the different variations of media objects within a multimedia document that maximizes the total *content value* given the constraints of the client devices. We define a multimedia presentation $\mathcal{M} = [\mathcal{D}, \mathcal{L}]$ as a tuple consisting of a multimedia document $\mathcal{D}$ and a document layout $\mathcal{L}$. We define the multimedia document $\mathcal{D}$ as the set of media objects O_{ij}, as follows:

$$\mathcal{D} = \{(O_{ij})_n\} \quad \text{(multimedia document)},$$

where $(O_{ij})_n$ gives the n^{th} media object, which has modality i and fidelity j. The document layout $\mathcal{L}$ gives the relative spatial and temporal location and size of each media object. We define an InfoPyramid $\mathcal{IP}$ of a media object as a collection of the different variations of the media object O_{ij}, as follows:

$$\mathcal{IP} = \{O_{ij}\} \quad \text{(InfoPyramid)}.$$

We then define an InfoPyramid document $\mathcal{IPD}$ as a set of InfoPyramid objects $\{O_{ij}\}$, as follows:

$$\mathcal{IPD} = \{\mathcal{IP}_n\} = \{\{O_{ij}\}_n\} \quad \text{(InfoPyramid document)}.$$

3.1.1 Content value scores

In order to optimize the selection, the InfoPyramid uses content value scores $V((O_{ij}))$ for each of the media objects O_{ij}, as shown in Figure 3. The content value scores can be based on automatic measures, such as entropy, or loss in fidelity that results from translating or summarizing the content. For example, the content value scores can be linked to the distortion introduced from compressing the images or audio. Otherwise, the content value scores can be tied directly to the methods that manipulate the content, or be assigned manually.

Figure 3 illustrates examples of the relative reciprocal content value scores of different variations of a video object. In this example, the original video (lower-left) has the highest content value. The manipulation of the video along the dimension of fidelity or modality reduces the content value. For example, converting the video to a sequence of images results in a small reduction in content value. Converting the video to a highly compressed audio track produces a higher reduction in the content value.

3.2 Media object variation selection

Given a multimedia document with N media objects, let $\{O_{ij}\}_n$ give the InfoPyramid of the n^{th} media object. Let

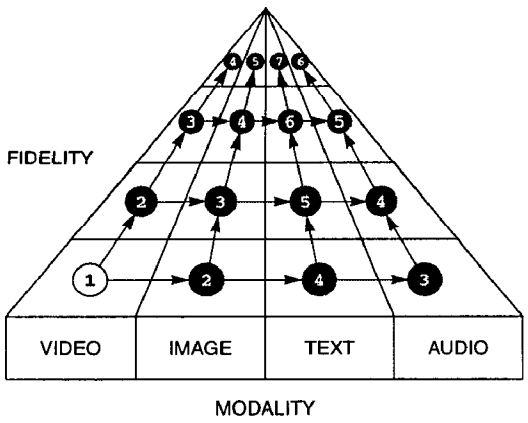

Figure 3: Example of the reciprocal content value scores assigned for different media objects variations of a video in the InfoPyramid.

$V((O_{ij})_n)$ give the relative content value score of the variation of the n^{th} media object with modality i and fidelity j, and let $D((O_{ij})_n)$ give its data size.

Let D_T give the total maximum data size allocated for the multimedia document by the client device. The total maximum data size may, in practice, be derived from the user's specified maximum load-time and the network conditions, or from the device constraints in storage or processing.

3.2.1 Maximum content value

The content selection process selects media object variation O^*_{ij} from each InfoPyramid in order to maximize the total content value for a target data size D_T as follows:

$$\begin{aligned} \sum_n V((O^*_{ij})_n) &= \max(\sum_n V((O_{ij})_n)), \text{ and} \\ \sum_n D((O^*_{ij})_n) &\leq D_T, \end{aligned} \quad (1)$$

where $(O^*_{ij})_n$ gives for each n, the optimal variation of the media object, which has fidelity i and modality j.

3.2.2 Minimum load time

Alternatively, given a minimum acceptable total content value V_T, the content select process selects media object variations O^*_{ij} from each InfoPyramid in order to minimum the total data size as follows:

$$\begin{aligned} \sum_n D((O^*_{ij})_n) &= \min(\sum_n D((O_{ij})_n)), \text{ and} \\ \sum_n V((O^*_{ij})_n) &\geq V_T, \end{aligned} \quad (2)$$

where, as above, $(O^*_{ij})_n$ gives for each n, the optimal variation of the media object, which has fidelity i and modality j.

3.2.3 Device constraints and preferences

By extending the selection process, we can consider other constraints of the client devices. For example, the content selection system can incorporate device screen size S_T, as follows: let $S((O_{ij})_n)$ give the spatial size of the variation of the n^{th} media object with modality i and fidelity j. Then, we add the constraint

$$\sum_n S((O^*_{ij})_n) \leq S_T, \quad (3)$$

to the optimization process. In the same way, we can include additional device constraints such as color depth, streaming bandwidth, and processing power.

3.3 Selection optimization

Using the InfoPyramid for managing the variations of the media objects, the number of different variations of each multimedia document is combinatorial in the number of media objects (N) and number of variations of each media object (M), and is given by M^N. In order to solve the optimization problems of Eq. 1 and Eq. 2, we convert the constrained optimization problems into the equivalent Lagrangian unconstrained problems, as described in [16].

The optimization solution is based on the resource allocation technique proposed in [24] for arbitrary discrete functions. We illustrate this by converting the problem in Eq. 1 to the following unconstrained problem:

$$\min\{\sum_n D((O_{ij})_n) - \lambda(V_T - V((O_{ij})_n))\}. \quad (4)$$

We can see that the optimal solution gives that for all n, the selected variation of the media objects $(O^*_{ij})_n$ operate at the same constant trade-off λ in content-value $V((O_{ij})_n)$ *vs* data size $D((O_{ij})_n)$. In order to solve the optimization problem, we need only to search over values of λ.

3.4 Example content selection

We illustrate the content selection in an example multimedia document, as shown in Figure 4. The multimedia document has two media objects: a video object $= (O_{00})_0$ and a text object $= (O_{20})_1$. For each media object $(O_{ij})_n$, where $n \in \{0, 1\}$, we construct an InfoPyramid $\{O_{ij}\}_n$, which gives the different variations of the media object. The selection process selects the variations $(O^*_{ij})_0$ and $(O^*_{ij})_1$, respectively, in order to maximize the total content value.

$(O_{ij})_n$	$V((O_{ij})_n)$	$D((O_{ij})_n)$	Modality
$(O_{00})_0$	1.0	1.0	video
$(O_{01})_0$	0.75	0.25	video
$(O_{10})_0$	0.5	0.10	image
$(O_{11})_0$	0.25	0.05	image
$(O_{20})_1$	1.0	0.5	text
$(O_{21})_1$	0.5	0.10	text
$(O_{32})_1$	0.75	0.25	audio
$(O_{33})_1$	0.25	0.05	audio

Table 1: Summary of different variations of two media objects, $(O_{ij})_0$ and $(O_{ij})_1$.

We consider four variations of each media object with content values and data sizes given in Table 1. By iterating over values for the trade-off λ in content value and data size, we obtain the content selection table of Table 2, which shows the media object variations that maximize the total content value $\max(\sum_n V((O_{ij})_n))$ for different total maximum data sizes D_T, as given in Eq 1:

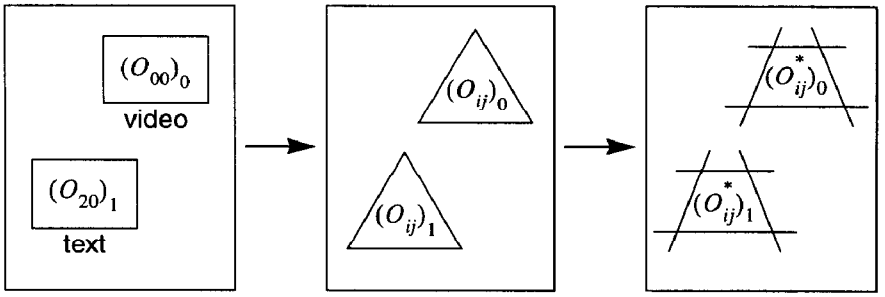

Figure 4: Example content selection for a multimedia document $\mathcal{D}$ consisting of two media objects: (a) $(O_{00})_0$ (video) and $(O_{20})_1$ (text), (b) InfoPyramid document with $(O_{ij})_0$ and $(O_{ij})_1$, and (c) selected variation of the media objects $(O^*_{ij})_0$ and $(O^*_{ij})_1$.

D_T	$(O^*_{ij})_0$	$(O^*_{ij})_1$	$\sum_n V((O^*_{ij})_n)$	$\sum_n D((O^*_{ij})_n)$
1.5	$(O_{00})_0$	$(O_{20})_1$	2.0	1.5
1.25	$(O_{00})_0$	$(O_{32})_1$	1.75	1.25
1.0	$(O_{01})_0$	$(O_{20})_1$	1.75	0.75
0.6	$(O_{10})_0$	$(O_{20})_1$	1.5	0.6
0.35	$(O_{01})_0$	$(O_{21})_1$	1.25	0.35
0.35	$(O_{10})_0$	$(O_{32})_1$	1.25	0.35
0.2	$(O_{10})_0$	$(O_{21})_1$	1.0	0.2
0.1	$(O_{11})_0$	$(O_{33})_1$	0.5	0.1

Table 2: Summary of the selected variations of the two media objects, $(O_{ij})_0$ and $(O_{ij})_1$, under different total maximum data size constraints D_T.

3.5 Server-based selection

Adaptive selection is most appropriate when the InfoPyramid is used to store and manage different variations of the media objects. However, in many cases, the multimedia content is already stored in legacy formats and is served by traditional, non-adaptive content servers. For example, this is usually the case for multimedia documents on the Web. One way to solve this problem is by using active proxies for transcoding the content on-the-fly to adapt it to client devices [8], as we describe next.

4 MULTIMEDIA TRANSCODING

The second form of scalable multimedia delivery to pervasive devices is on-line transcoding. For on-line transcoding, we use the InfoPyramid as a transient structure in transcoding the media objects to the most appropriate modalities and fidelities. In order to best adapt the content, we design a dynamic routing system that exercises the trade-off between delay and distortion in selecting the transcoding paths.

4.1 Input-output signature

We assume that the InfoPyramid needs to manage a potentially large number of different transcoders. In general, it is difficult to hard-code the transcoding paths for each media object for each client device under many different conditions. The difficulty is compounded by the fact that the device constraints fall in a range of continuous values for display, storage, processing and bandwidth. Therefore, we define a process for selecting the transcoding paths dynamically. To do this, we first need to describe each transcoding method to allow automatic selection, as follows:

For each transcoder method L_k in the InfoPyramid, we define an input-output signature $L_k = (L_k^{in}, L_k^{out})$, where

$$\begin{aligned} L_k^{in} &= (M_k^{in}, F_k^{in}, D_k^{in}, S_k^{in}), \\ L_k^{out} &= (M_k^{out}, F_k^{out}, D_k^{out}, S_k^{out}) \end{aligned}$$

and

$$\begin{aligned} M_k^{in,out} &= \text{input-output modality} \\ F_k^{in,out} &= \text{input-output fidelity} \\ D_k^{in,out} &= \text{input-output data size} \\ S_k^{in,out} &= \text{input-output spatial-temporal size.} \end{aligned}$$

For example, a 50% text summarizer defines a signature of

$$\begin{aligned} L_k^{in} &= (M_k^{in}, F_k^{in}, D_k^{in}, S_k^{in}) \\ L_k^{out} &= (M_k^{in}, 0.5F_k^{in}, 0.5D_k^{in}, 0.5S_k^{in}), \end{aligned}$$

which indicates that the transcoder produces an output with the same modality as input, but with a 50% reduction in data size and spatial size.

4.2 Distortion vs. delay

The transcoding input-output signatures allow the dynamic selection of the transcoding paths. For example, given an input with a particular modality, and a desired output data with a particular modality and fidelity, the system can construct a set of valid transcoding paths that carry out the transcoding.

We use a dynamic routing procedure to select the best transcoding path for each of the media objects based on the distortion *vs.* delay tradeoffs as follows:

For each transcoder method L_k we also define a delay rate ρ_k and distortion D_k. The delay rate allows the computation of the total processing delay δ_k required by the transcoder for data of a given input size Z_k^{in}, where

$$\delta_k = \rho_k Z_k^{in}.$$

The distortion D_k gives the ratio of the content value of the transcoded output data compared to the input data,

$$D_k = V_k^{out}/V_k^{in}.$$

4.3 Dynamic routing

We define the following dynamic routing procedure that selects a transcoding path that optimizes the distortion *vs.* delay trade-off, as follows:

4.3.1 Minimum distortion

The dynamic routing process selects a sequence of the transcoders that minimizes the total delay for a total maximum distortion D_T, as follows:

$$\min(\sum_{n=0}^{N-1}(\delta_i)_n) \quad \text{s.t.} \quad \sum_{n=0}^{N-1}(d_i)_n < D_T. \tag{5}$$

4.3.2 Minimum delay

The dynamic routing process selects a sequence of the transcoders that minimizes the total distortion for a total maximum delay $\mathcal{D}_T$, as follows:

$$\min(\sum_{n=0}^{N-1}(d_i)_n) \quad \text{s.t.} \quad \sum_{n=0}^{N-1}(\delta_i)_n < \mathcal{D}_T. \tag{6}$$

As in the previous cases for content selection, we can see that the number of possible transcoding paths is combinatorial, in this case given M different transcoder methods, there are M^N different transcoder paths of length N. We can potentially solve the optimization problems of Eq. 5 and Eq. 6 using the Lagrangian method, as described earlier. In this case, at the optimal solution, the selected transcoders operate at the same constant delay *vs.* distortion trade-off. However, we instead use the input-out signatures of the transcoder functions to construct the allowable transcoding paths. We then need to select the best path in terms of delay and distortion.

We provide a more detailed examination of the dynamic routing procedure for the case of image transcoding.

4.4 IMAGE TRANSCODING

We have many methods for image transcoding, including methods for image size reduction, image color reduction, image compression and format conversion.

4.5 Image transcoding dynamic routing

For each image transcoding method L_k, we define an extended input-output signature $L_k = (L_k^{in}, L_k^{out})$, as follows:

$$\begin{aligned} L_k^{in} &= (M_k^{in}, F_k^{in}, D_k^{in}, S_k^{in}, R_k^{in}, T_k^{in}) \\ L_k^{out} &= (M_k^{out}, F_k^{out}, D_k^{out}, S_k^{out}, R_k^{in}, T_k^{out}) \end{aligned}$$

where $M_k^{in}, F_k^{in}, D_k^{in}, S_k^{in}$ are defined above, and

$$\begin{aligned} R_k^{in,out} &= \text{input-output image format} \\ T_k^{in,out} &= \text{input-out-put image type} \end{aligned}$$

For example, a 50% size reduction method for RGB images defines a signature of

$$\begin{aligned} L_k^{in} &= (L_k^{in}, F_k^{in}, D_k^{in}, S_k^{in}, R_k^{in}, T_k^{in}) \\ L_k^{out} &= (L_k^{in}, 0.5F_k^{in}, 0.25D_k^{in}, 0.5S_k^{in}, R_k^{in}, T_k^{out}) \end{aligned}$$

We define the image transcoding functions that perform image manipulation and image analysis. The objective of image analysis is to obtain information about the images that improves the transcoding.

4.6 Image analysis

Content analysis can be important for developing high-quality efficient transcoders. Image content analysis is motivated by the need to differentially transcode images depending on their characteristics, as illustrated in Figure 5.

For example, for one, it is desirable to transcode graphics and photographs differently with regards to size reduction, color reduction and quality reduction. In order to classify the images, we use image analysis procedures that classify the images into image type and purpose classes, as described in [7]. We define the following image type classes:

$\mathcal{T}$ = {BWG, BWP, GRG, GRP, SCG, CCG, CP},

where BWG = b/w graphics, BWP = b/w photo, GRG = gray graphics, GRP = gray photos, SCG = simple color graphic, CCG = complex color graphics, and CP color photo. We also define the following image purpose classes

$\mathcal{P}$ = {ADV, DEC, BUL, RUL, MAP, INF, NAV, CON},

where ADV = advertisements, DEC = decorations, BUL = bullets, RUL = rules, MAP = maps, INF = informational images, NAV = navigation images and CON = content related images [25].

4.6.1 Image type classification

The image type classification system extracts color features of the images and utilizes a decision tree classifier, as described in [7]. The decision tree classifies the images along the dimensions of color content (color, gray, b/w), and source (photographs, graphics). Distinguishing between b/w, gray and color requires image analysis because of artifacts introduced in the image production and compression often obfuscates the image type.

4.6.2 Image purpose classification

Multi-modal information is often useful in helping to categorize images in multimedia documents [26, 27]. As illustrated in Figure 6, we use the document context of each image, its related text and image type information to classify the images into the image purpose classes $\mathcal{P}$. The system makes use of five contexts for the images in the Web documents: $\mathcal{C}$ = {BAK, INL, ISM, REF, LIN}, which correspond to the HTML tags as follows: BAK = background image, INL = in-line image, ISM = is a map image, REF= referenced image and LIN = linked image.

The image purpose classification system also uses a dictionary of terms extracted from the text related to the images. The terms are extracted from the 'alt' tag text, the image URL address strings, and the text nearby the images in the Web documents. The system makes use of terms such as $\mathcal{D}$ = {"ad", "texture", "bullet", "map", "logo", "icon"}. The system also extracts a number of image attributes, such as image width (w), height (h), and aspect ratio ($r = w/h$).

The system classifies the images into the purpose classes using a rule-based decision tree framework described in [25]. The rules map the values for image type $t \in \mathcal{T}$, context $c \in \mathcal{C}$, terms $d \in \mathcal{D}$, and image attributes $a \in \{w, h, r\}$ into the purpose classes. The following examples illustrate some of the image purpose rules:

$$\begin{aligned} p = \text{ADV} &\leftarrow t = \text{SCG},\ c = \text{REF},\ d = \text{"ad"} \\ p = \text{DEC} &\leftarrow c = \text{BAK},\ d = \text{"texture"} \\ p = \text{MAP} &\leftarrow t = \text{SCG},\ c = \text{ISM},\ w > 256,\ h > 256 \\ p = \text{BUL} &\leftarrow t = \text{SCG},\ r > 0.9,\ r < 1.1,\ w < 12 \\ p = \text{RUL} &\leftarrow t = \text{SCG},\ r > 20, h\ < 12 \end{aligned}$$

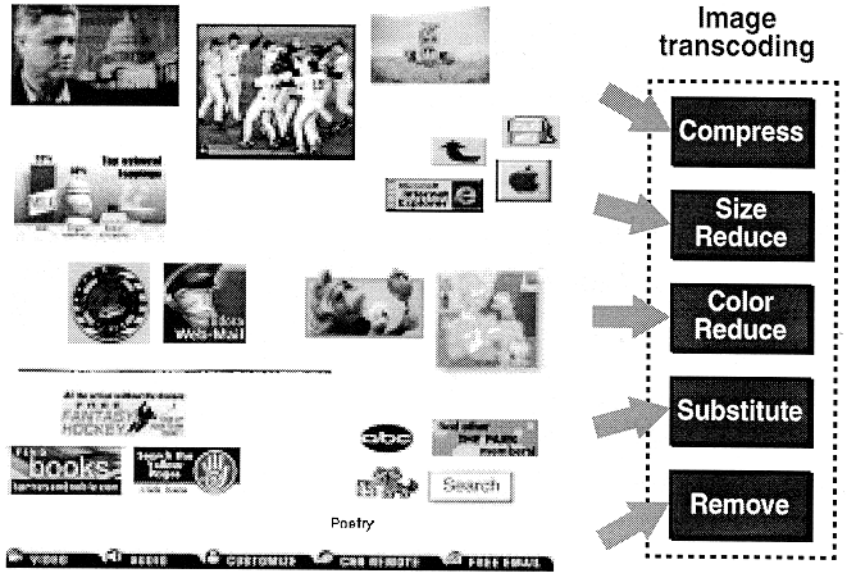

Figure 5: Image content analysis is motivated by the need to differentiate the selection of the transcoding methods based on the input image.

$$p = \text{INF} \quad \leftarrow \quad t = \text{SCG},\ c = \text{INL},\ h < 96,\ w < 96$$

4.7 Image manipulation

The system provides a number of image manipulation functions. Examples include image size reduction using a number of different methods, image color reduction using a number of different methods, image quality reduction and image format conversion.

4.7.1 Spatial size change routing

We consider as an example, the different methods for size reduction as follows:

50% image size reduction

$$\begin{aligned} L_0^{in} &= (M_0^{in}, F_0^{in}, D_0^{in}, S_0^{in}, R_0^{in}, T_0^{in}) \\ L_0^{out} &= (M_0^{in}, 0.5F_0^{in}, 0.25D_0^{in}, 0.25S_0^{in}, R_0^{in}, T_0^{in}), \end{aligned}$$

Graphic 50% image subsampler

$$\begin{aligned} L_1^{in} &= (M_1^{in}, F_1^{in}, D_1^{in}, S_1^{in}, R_1^{in}, T_1^{in}) \\ L_1^{out} &= (M_1^{in}, 0.5F_1^{in}, 0.25D_1^{in}, 0.25S_1^{in}, R_1^{in}, T_1^{in}), \end{aligned}$$

Photograph 50% image subsampler

$$\begin{aligned} L_2^{in} &= (M_2^{in}, F_2^{in}, D_2^{in}, S_2^{in}, R_2^{in}, T_2^{in}) \\ L_2^{out} &= (M_2^{in}, 0.5F_2^{in}, 0.25D_2^{in}, 0.25S_2^{in}, R_2^{in}, T_2^{in}), \end{aligned}$$

Image type analysis

$$\begin{aligned} L_3^{in} &= (M_3^{in}, F_3^{in}, D_3^{in}, S_3^{in}, R_3^{in}, T_3^{in}) \\ L_3^{out} &= (M_3^{in}, F_3^{in}, D_3^{in}, S_3^{in}, R_3^{in}, T_3^{out}), \end{aligned}$$

We also have that these functions have different delays rates ρ_k and distortions d_k, for $k = 0 \ldots 3$.

The size reduction method L_0 generates the size reduction of any input type of image by filtering with a low-quality filter and subsampling. On the other hand, L_1 simply subsamples the any input SCG image, while L_2 performs size reduction on any CP image by filtering with a high quality filter before subsampling. The analysis method, L_3 determines the image type information, i.e., SCG *vs* CP images, which can be used to differentially select the size reduction methods based on image type.

The delay rates and distortions of the size reduction functions differ because functions L_1 and L_2 use specially tuned algorithms for SCG and CP, respectively. But, they require image type information in order to be used. The gathering of image type information however, requires the execution of L_3. This gives the different delay *vs.* distortion options for transcoding the images.

We summarize the possible image size transcoding paths based on the above methods, as follows:

$$\begin{aligned} \text{IMAGE}_{in} &\rightarrow \quad L_0 \quad \rightarrow \text{IMAGE}_{out} \\ \text{IMAGE}_{in} &\rightarrow \quad L_3 L_1 \quad \rightarrow \text{IMAGE}_{out} \\ \text{IMAGE}_{in} &\rightarrow \quad L_3 L_2 \quad \rightarrow \text{IMAGE}_{out} \end{aligned}$$

Given the tradeoff between distortion and delay, the dynamic routing system can select the best transcoding path. For fast, low-quality transcoding, the system can directly execute the first path (L_0). However, for higher-quality, the system can select either of the next two paths ($L_3 L_1$) or ($L_3 L_2$), which require execution of the image analysis function L_3.

4.7.2 Image data size reduction

In general, the dynamic routing can be used to select among the size reduction, color reduction and compression methods in order to reduce the data size of the image. We consider

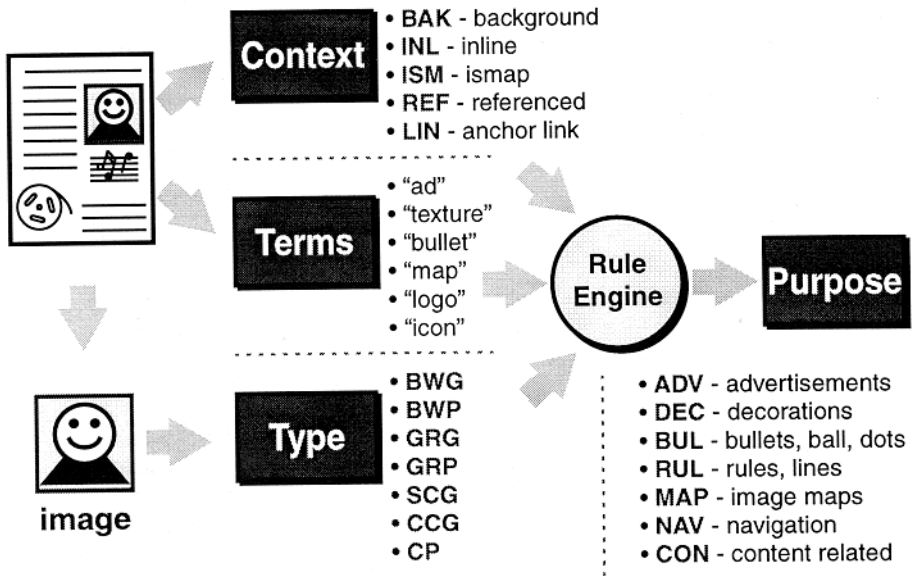

Figure 6: Image purpose detection uses image type information, multimedia document context and related text to classify images into image purpose classes.

different functions for color reduction and compression, in addition to the size reduction methods above, as follows:

Color to Gray

$$\begin{aligned} L_4^{in} &= (M_4^{in}, F_4^{in}, D_4^{in}, S_4^{in}, R_4^{in}, T_4^{in}) \\ L_4^{out} &= (M_4^{in}, 0.5F_4^{in}, 0.33D_4^{in}, S_4^{in}, R_4^{in}, T_4^{in}), \end{aligned}$$

Gray to B/W

$$\begin{aligned} L_5^{in} &= (M_5^{in}, F_5^{in}, D_5^{in}, S_5^{in}, R_5^{in}, T_5^{in}) \\ L_5^{out} &= (M_5^{in}, 0.5F_5^{in}, 0.25D_5^{in}, 0.25S_5^{in}, R_5^{in}, T_5^{in}), \end{aligned}$$

50% compression

$$\begin{aligned} L_6^{in} &= (M_6^{in}, F_6^{in}, D_6^{in}, S_6^{in}, R_6^{in}, T_6^{in}) \\ L_6^{out} &= (M_6^{in}, 0.5F_6^{in}, 0.5D_6^{in}, S_6^{in}, R_6^{in}, T_6^{in}), \end{aligned}$$

We summarize the additional image data size reduction transcoding paths as follows:

$$\begin{aligned} \text{IMAGE}_{in} &\rightarrow L_4 \rightarrow \text{IMAGE}_{out} \\ \text{IMAGE}_{in} &\rightarrow L_4 L_5 \rightarrow \text{IMAGE}_{out} \\ \text{IMAGE}_{in} &\rightarrow L_6 \rightarrow \text{IMAGE}_{out} \end{aligned}$$

Based on the delay *vs.* distortion trade-offs, the dynamic routing procedure can select the transcoding paths.

4.8 Image transcoding policies

We can also use the image type and purpose information more explicitly to define transcoding policies [7]. Consider the following example transcoding policies based upon image type and client device capabilities:

$$\begin{aligned} L_2(X) &\leftarrow \text{type}(X) = \text{CP}, \text{ device} = \text{HHC} \\ L_1(X) &\leftarrow \text{type}(X) = \text{SCG}, \text{ device} = \text{HHC} \\ L_6 L_4 L_5(X) &\leftarrow \text{type}(X) = \text{CP}, \text{ device} = \text{PDA} \\ L_4 L_5(X) &\leftarrow \text{type}(X) = \text{SCG}, \text{ device} = \text{PDA} \\ L_6(X) &\leftarrow \text{type}(X) = \text{GRP}, \\ &\quad \text{bandwidth} \leq 28.8\text{K} \\ L_1(X) &\leftarrow \text{type}(X) = \text{GRG}, \\ &\quad \text{bandwidth} \leq 28.8\text{K} \end{aligned}$$

Here, PDA and HHC indicate two different types of client devices.

4.9 Remove, substitute and null operations

In addition to size reduction, color reduction, and compression, other transcoding operations for images in multimedia documents include image removal and substitution, defined as follows:

Remove

$$\begin{aligned} L_7^{in} &= (M_7^{in}, F_7^{in}, D_7^{in}, S_7^{in}, R_7^{in}, T_7^{in}) \\ L_7^{out} &= (0, 0, 0, 0, 0, 0), \end{aligned}$$

Substitute with text

$$\begin{aligned} L_8^{in} &= (M_8^{in}, F_8^{in}, D_8^{in}, S_8^{in}, R_8^{in}, T_8^{in}) \\ L_8^{out} &= (\text{TEXT}, F_8^{out}, D_8^{out}, S_8^{out}, R_8^{in}, T_8^{in}), \end{aligned}$$

Null operation

$$\begin{aligned} L_9^{in} &= (M_9^{in}, F_9^{in}, D_9^{in}, S_9^{in}, R_9^{in}, T_9^{in}) \\ L_9^{out} &= (M_9^{in}, F_9^{in}, D_9^{in}, S_9^{in}, R_9^{in}, T_9^{in}), \end{aligned}$$

This allows us to design transcoding policies that make use of the image purpose analysis. Consider the following transcoding policies:

$$
\begin{aligned}
L_9(X) &\leftarrow \text{purpose}(X) = \text{MAP} \\
L_8(X) &\leftarrow \text{purpose}(X) = \text{ADV} \\
&\quad \text{bandwidth} \leq 14.4\text{K} \\
L_7(X, \text{"<li>"}) &\leftarrow \text{purpose}(X) = \text{BUL}, \\
&\quad \text{device} = \text{PDA} \\
L_7(X, t) &\leftarrow \text{purpose}(X) = \text{INF}, \\
&\quad \text{display size} = 320 \times 200
\end{aligned}
$$

The first policy makes sure that map images are not reduced in size in order to preserve the click focus translation. The second policy illustrates the removal of advertisement images if the bandwidth is low. The third policy substitutes the bullet images with the HTML code "<li>," which draws a bullet without requiring the image. A similar policy substitutes rule images with "<hr>". The last policy substitutes the information images, i.e., logos, icons, mastheads, with related text if the device screen is small.

5 IMPLEMENTATION

The scalable multimedia delivery system can be deployed at a number of places in the network including at the server, in a proxy, or at the client, as shown in Figure 7. Deployed at the server, the system stores the variations of the media objects and selects the appropriate ones for the client devices. Deployed at a proxy, for each request, the system retrieves, transcodes and delivers the media objects, on-the-fly. Deployed at the client, the scalable multimedia delivery system can be used to customize the content presentation.

5.1 On-line transcoding

As part of the Content Adaption Framework (CAF) project, we are developing an Internet transcoding proxy [1]. The transcoding proxy is based on an Apache web server. The transcoding proxy includes a customized Apache proxy module that transcodes images and text to adapt Web pages to different client devices. At present, information about the client devices is communicated to the proxy by the user. The user is given a form and optionally a slidebar that allows the transcoding levels for text and images to be set. The transcoding proxy receives requests from the client devices, then retrieves the content, transcodes the images and text according to the transcoding preferences and delivers the results to the user.

5.2 Role of standards

In general, standards can be very important in the deployment of scalable multimedia delivery systems. We are currently exploring the use of a client device description vocabulary based on the Composite Capability/Preference Profiles (CC/PP) specification [28]. CC/PP provides an extensible framework for describing user preferences and device capabilities. A proxy or server is able to interpret the CC/PP information to transcode or select content appropriately.

We are also currently exploring the use of content description (MPEG-7) and Web annotation (W3C) standards. One recent proposal to W3C ([29]) describes an annotation system that attaches information to HTML/XML documents to guide their adaptation to the characteristics of diverse information appliances. Our recent proposal to MPEG ([22]) describes MPEG-7 meta-data for annotating multimedia documents and objects to enable adaptive selection and transcoding of multimedia content.

[1] http://www.alphaworks.ibm.com/tech/transcodingproxy

6 SUMMARY

We presented a system for scalable delivery of multimedia documents in order to adapt them to the capabilities of the pervasive client devices such as PDAs, HHCs, smart phones and PCs. The system uses a data model called the InfoPyramid to manage different variations of the media objects and manipulate the media objects by translation and summarization to adapt the multimedia documents to the constraints of the client devices.

References

[1] T. W. Bickmore and B. N. Schlit. Digestor: Device-independent access to the World-Wide Web. *Proc. 6th Int. WWW Conf*, 1997.

[2] G. C. Vanderheiden. Anywhere, anytime (+anyone) access to the next-generation WWW. *Proc. 6th Int. WWW Conf*, 1997.

[3] H. Bharadvaj, A. Joshi, and S. Auephanwiriyakul. An active transcoding proxy to support mobile Web access. *Proc. 17th IEEE Symp. Reliable Distributed Syst.*, October 1998.

[4] A. Fox, S. D. Gribble, Y. Chawathe, and E. A. Brewer. Adapting to network and client variation using active proxies: Lessions and perspectives. *IEEE Personal Commun.*, 40, 1998.

[5] A. Fox, S. D. Gribble, E. A. Brewer, and E. Amir. Adapting to network and client variability via on-demand dynamic distillation. In *ASPLOS-VII*, Cambridge, MA, October 1996.

[6] A. Ortega, F. Carignano, S. Ayer, and M. Vetterli. Soft caching: Web cache management techniques for images. In *Workshop on Multimedia Signal Processing*, pages 475 - 480, Princeton, NJ, June 1997. IEEE.

[7] J. R. Smith, R. Mohan, and C.-S. Li. Content-based transcoding of images in the Internet. In *IEEE Proc. Int. Conf. Image Processing (ICIP)*, Chicago, Il, October 1998.

[8] J. R. Smith, R. Mohan, and C.-S. Li. Transcoding Internet content for heterogenous client devices. In *Proc. IEEE Inter. Symp. on Circuits and Syst. (ISCAS)*, June 1998. Special session on Next Generation Internet.

[9] Intel Quick Web. http://www.intel.com/quickweb.

[10] Spyglass-Prism. http://www.spyglass/products/prism.

[11] Unwired Planet. Handheld device markup language specification. Technical Report Version 2.0, Unwired Planet, Inc., April 1997.

[12] L. Bouthillier. Synchronized multimedia on the Web. *Web Techniques Magazine*, 3(9), September 1998.

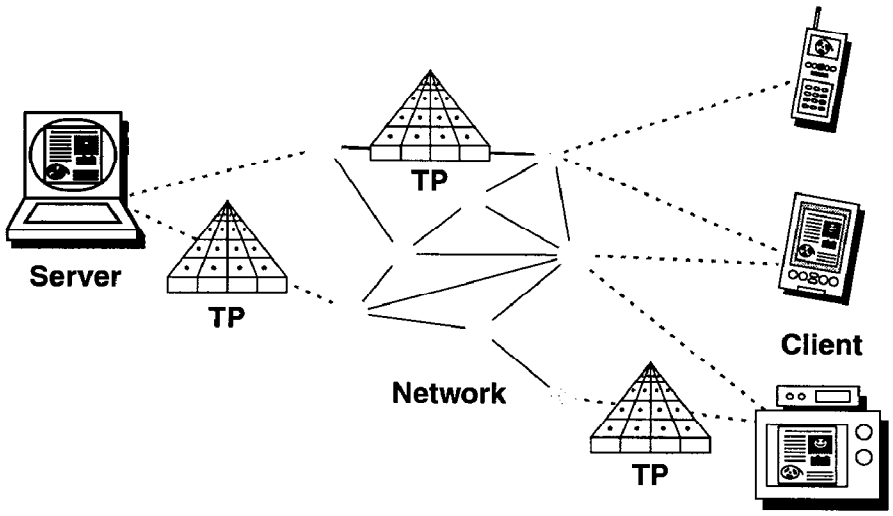

Figure 7: The scalable multimedia delivery system can be deployed in the form of transcoding proxies (TP) at the server, in the network, or at the clients.

[13] O. Avaro, P. Chou, A. Eleftheriadis, C. Herpel, and C. Reader. The MPEG-4 systems and description languages: A way ahead in audio visual information representation. *Signal Processing: Image Communication, Special Issue on MPEG-4*, 4(9):385–431, May 1997.

[14] K. S. Candan, B. Prabhakaran, and V. S. Subrahmanian. CHIMP: A framework for supporting distributed multimedia document authoring and presentation. In *Proc. ACM Intern. Conf. Multimedia (ACMMM)*, pages 329 - 339, Boston, MA, November 1996.

[15] A. Vogel, B. Kerherve, G. von Bockmann, and J. Gecsei. Distributed multimedia and QOS: A survey. *IEEE Multimedia Mag.*, 2(2):10 - 18, Summer 1994.

[16] R. Mohan, J. R. Smith, and C.-S. Li. Adapting multimedia internet content for universal access. *IEEE Trans. Multimedia*, 1(1):104 - 114, March 1999.

[17] L. Weitzman and K. Wittenburg. Automatic presentation of multimedia documents using relational grammars. In *Proc. ACM Intern. Conf. Multimedia (ACMMM)*, pages 443 - 451, San Francisco, CA, November 1994.

[18] J. R. Smith. Digital video libraries and the Internet. *IEEE Communications Mag.*, 37(1):92 - 99, January 1999. Special issue on the Next Generation Internet.

[19] R. Kling and M. Elliott. Digital library design for usability. In *Proc. Conf. Theory and Practice of Digital Libraries*, College Station, TX, June 1994.

[20] J. R. Smith, C.-S. Li, and R. Mohan. Infopyramid description scheme for multimedia content. In *MPEG-7 proposal*, number MPEG99/P473 in ISO/IEC JTC1/SC29/WG11, Lancaster, UK, February 1999.

[21] C. Christopoulos, T. Ebrahimi, V.V. Vinod, J. R. Smith, R. Mohan, and C.-S. Li. Universal access and media conversion. In *MPEG-7 applications proposal*, number MPEG99/M4433 in ISO/IEC JTC1/SC29/WG11, Seoul, Korea, March 1999.

[22] J. R. Smith, C.-S. Li, R. Mohan, A. Puri, C. Christopoulos, A. B. Benitez, P. Bocheck, S.-F. Chang, T. Ebrahimi, and V. V. Vinod. MPEG-7 content description scheme for universal multimedia access. In *MPEG-7 proposal*, number MPEG99/M4949 in ISO/IEC JTC1/SC29/WG11, Vancouver, BC, July 1999.

[23] C.-S. Li, R. Mohan, and J. R. Smith. Multimedia content description in the InfoPyramid. In *IEEE Proc. Int. Conf. Acoust., Speech, Signal Processing (ICASSP)*, Seattle, WA, May 1998. Special session on Signal Processing in Modern Multimedia Standards.

[24] Y. Shoham and A. Gersho. Efficient bit allocation for an arbitrary set of quantizers. *IEEE Trans. Acoust., Speech, Signal Processing*, 36(9), September 1988.

[25] S. Paek and J. R. Smith. Detecting image purpose in World-Wide Web documents. In *IS&T/SPIE Symposium on Electronic Imaging: Science and Technology - Document Recognition*, San Jose, CA, January 1998.

[26] N. C. Rowe and B. Frew. Finding photograph captions multimodally on the World Wide Web. Technical Report Code CS/Rp, Dept. of Computer Science, Naval Postgraduate School, 1997.

[27] J. R. Smith and S.-F. Chang. Visually searching the Web for content. *IEEE Multimedia Mag.*, 4(3):12 - 20, July–September 1997.

[28] F. Reynolds, J. Hjelm, S. Dawkins, and S. Singhal. A user side framework for content negotiation: Composite capability/preference profiles (CC/PP). Technical Report http://www.w3.org/TR/NOTE-CCPP, W3C, November 1998.

[29] M. Hori, R. Mohan, H. Maruyama, and S. Singhal. Annotation of web content for transcoding. Technical Report NOTE-annot-19990524, W3C, July 1999.