Adapting Multimedia Internet Content for Universal Access

Rakesh Mohan, *Member, IEEE*, John R. Smith, *Member, IEEE*, and Chung-Sheng Li, *Senior Member, IEEE*

***Abstract*—Content delivery over the Internet needs to address both the multimedia nature of the content and the capabilities of the diverse client platforms the content is being delivered to. We present a system that adapts multimedia Web documents to optimally match the capabilities of the client device requesting it. This system has two key components. 1) A representation scheme called the *InfoPyramid* that provides a multimodal, multiresolution representation hierarchy for multimedia. 2) A *customizer* that selects the best content representation to meet the client capabilities while delivering the most value.**

We model the selection process as a resource allocation problem in a generalized rate-distortion framework. In this framework, we address the issue of both multiple media types in a Web document and multiple resource types at the client. We extend this framework to allow prioritization on the content items in a Web document. We illustrate our content adaptation technique with a web server that adapts multimedia news stories to clients as diverse as workstations, PDA's and cellular phones.

***Index Terms*—Compression, content adaptation, Internet, multimedia, information appliances, rate-distortion, transcoding, universal access.**

I. Introduction

NETWORK appliances, or information appliances, are computing devices that are network enabled. They typically have fewer resources than personal computers and are geared toward a limited number of applications. Some current examples of network appliances are hand-held computers (HPC's), personal digital assistants (PDA's), set-top boxes, screen telephones, smart cellular phones and network computers. In "ubiquitous" or "pervasive" computing, consumers will use different network appliances to connect to the Internet for different applications, from entertainment to banking, from different settings, from living rooms to cars. Sources, such as *The Economist* [1] and International Data Corporation (IDC) [2], predict that the sales of network appliances will significantly outstrip that of personal computers after the year 2002. Therefore, within a decade, network appliances will replace personal computers as the client device of choice for viewing Web content.

Currently multimedia content is authored with the personal computer as the target client device. Web documents, which have rapidly become the largest deployed form of multimedia, are also authored specifically for personal computers with reasonable wired network connections. However, network appliances are very different from the typical PC on a modem or LAN. The network appliances vary widely in their features such as screen size, resolution, color depth, computing power, storage and software. They also use a variety of network connections ranging from cable to mobile, with different bandwidth, connection characteristics and costs [7]. The diversity of these devices will make it difficult and expensive to author multimedia content separately for each individual type of device. Therefore, technologies that can adapt multimedia content to diverse client devices will become critical in the coming pervasive computing era.

In this paper we present a system that **adapts** multimedia Web content to optimally match the resources and capabilities of diverse client devices. This system employs two key technologies.

1) A progressive data representation scheme called the **InfoPyramid** [25]. Content items on a Web page are transcoded into multiple resolution and modality versions so that they can be rendered on different devices. For example, a video item is transcoded in to a set of images so that it can be rendered on a device not capable of displaying video. The InfoPyramid provides a multimodal, multiresolution representation for the content items and their transcoded versions.
2) A **customizer** that selects the best versions of content items from the InfoPyramids to meet the client resources while delivering the most "value." The customizer allocates resources on the client among the items in the document. This resource allocation results in the selection of the appropriate resolution or modality of the content items. If the client has limited resources (such as a PDA or pager), some of the content items may not get any resources assigned and thus will not be delivered to the client. We propose a novel *value-resource* framework for the customizer. This value-resource framework allows us to design and analyze a number of content adaptation strategies.

We illustrate this content adaptation with a multimedia news delivery system that adapts to clients ranging from workstations to cellular phones.

A. Related Work

Much work (for a small sampling, see [3]–[6]) has been done on adapting video to bandwidth variations by selecting a suitable compression scheme. These systems consider only a single type of media, not composite multimedia documents.

Manuscript received September 9, 1998; revised December 9, 1998. The associate editor coordinating the review of this paper and approving it for publication was Dr. Thomas R. Gardos.

The authors are with the IBM T. J. Watson Research Center, Yorktown Heights, NY 10598 USA.

Publisher Item Identifier S 1520-9210(99)01784-8.

1520–9210/99$10.00 © 1999 IEEE

Akamai Ex. 1014
Akamai Techs. v. Equil IP Holdings
IPR2023-00329
Page 00001

Drastically different clients, such as those that cannot handle video, are not addressed.

Web content adaptation can be performed either at the server, at the client, at an intermediate proxy, or some combination of the three.

Some client devices adapt content at the device. For example, Windows-CE™ devices change color-depth (for example, from 24-bit color to 4-bit gray-level) of images. The drawbacks are that network appliances have low network bandwidth, which results in slow access to pages with rich multimedia, and they are restricted in their computational power, which makes content adaptation at the device slow, or even impossible.

Most content adaptation systems [7]–[16], [18] are http proxy-based. The proxy intercepts client device's requests for Web pages, fetches the requested content, adapts it, and sends the adapted version to the client. This content adaptation is often termed "*transcoding*."

In the TranSend project [7]–[10] a proxy transcodes Web content on the fly. The adaptation, which they term "distillation," is primarily limited to image compression and reduction of image size and color space. Video is also converted into different frame-rates and encodings using a video gateway [6]. Based on this work, a company, Proxinet [16], has been started that provides a proxy which customizes content for a special browser on the 3Com PalmPilot™ [17].

Bickmore and Schilit [11] also propose a proxy based mechanism. They use a number of heuristics and a planner to perform outlining and elision of the content to fit the Web page on the client's screen.

The Spyglass Prism™ [13], a commercial product, is another transcoding proxy. AvantGo [18] offers a solution similar to Proxinet.

Content adaptation upstream of the client results in a faster response time [7], [8]. Based on this observation, Intel launched the QuickWeb™ [12] service that compresses images at a proxy.

These transcoding proxies typically consider a few client devices and employ static, *ad-hoc*, content adaptation strategies. A common policy [7]–[13] is to scale all images by a fixed factor. Thus, these transcoding proxies fail to dynamically address the variation in the resource requirements of different Web documents. The set of client devices will also grow more diverse. Certain resources, such as effective network bandwidth, costs and patience of the users can be different for similar client devices. The static adaptation policies used by these systems do not handle well this variability in Web content and client resources.

None of the existing transcoding systems (with the possible exception of [11] and [14]) consider the requirements of the entire Web page or relationships between its various components in different media. Also, these systems only consider transcoding within the same modality.

In this paper, we propose a content adaptation framework that ***dynamically*** accounts for resource requirements of the complete Web page and its individual components. It selects from a number of different possible transcoded versions of the content, ones that provide the "best value" within the constraints of a client's resources. This system also considers transcoding between modalities. We provide a theoretical framework in which various content adaptation policies can be formulated and analyzed.

One big benefit of the proxy approach is that it is totally transparent to the content providers; they do not have to change the way they author or serve content. However, there are a number of drawbacks to this approach:

1) content providers have no control over how their content will appear to different clients;
2) there may be legal issues arising from copyright that may preclude or severely limit the transcoding by proxies;
3) HTML tags mainly provide formatting information rather than semantic information;
4) on the fly transcoding is difficult to apply to many media types such as video and audio.

These factors limit both the quality and the amount of customization that proxies can provide.

In this paper we present an alternate solution that extends the Web server deployed by a content provider. In this system, the content author can lay the transcoding policies and control the adaptation process. Also, the content author can edit and replace the transcoded versions of content items generated by the system. This control of the customization overcomes problems of publisher control and copyright issues faced by transcoding proxies [7]–[18]. The content is authored in XML [23], allowing the author to provide more information to the transcoding and customization system than can be deduced from an HTML page. The key benefit of this server-based system is that due to the guidance provided by the author, significantly greater level of customization can be performed than is possible in transcoding proxies. The systems generates transcoded versions of the content items prior to any requests; thus, it can handle media items such as video and audio which are difficult to handle in proxies. This off-line transcoding also leads to lower response latencies than proxies. The server shares the benefit of transcoding proxies in speeding content delivery as the customized content is often much smaller than the original content.

B. Outline

We first present the overall architecture of the system. The InfoPyramid, a multimodal, multiresolution representation hierarchy for multimedia, content analysis, transcoding modules, content customization, and cache, is described in Section II.

In Sections III–V, we describe the customization process in detail. In Section VI, we present an implementation of the content adaptation system. We present a summary in Section VII.

II. System Architecture

The content adaptation system is an extension to a Web (http) server. An overview of the system architecture is shown in Fig. 1. The content source contains the multimedia content to be delivered by the Web server. First, content is analyzed to extract meta-data used in guiding subsequent transcoding and selection processes. Based on the capabilities of the typical client devices, different transcoding modules

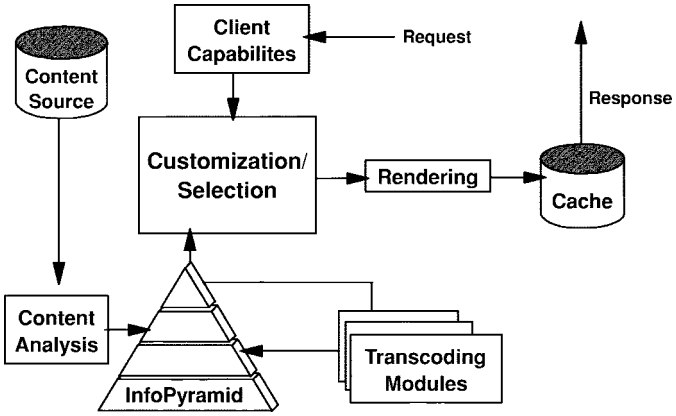


Fig. 1. Internet content adaptation system architecture.

are employed to generate versions of the content in different resolutions and modalities. A novel data representation, the InfoPyramid, is used to store the multiple resolutions and modalities of the transcoded content, along with any associated meta-data. This transcoding is done off-line, during content creation time. When the Web server receives a request, it first determines the capabilities of the requesting client device. A customization module then dynamically selects from the InfoPyramids, the resolutions or modalities that best meet the client capabilities. This selected content is then rendered in a suitable delivery format (for example, HTML) for delivery to the client. A cache that stores these client specific versions of content is used to improve response times. In the following sections, we describe these processes in detail.

A. Client Devices

The types of devices that can access the Internet are rapidly expanding beyond the workstation on LAN that most multimedia Internet content is authored for [1], [2], and [7]. One can now use personal digital assistants (PDA) such as the PalmPilot™ and Sharp Zaurus™, handheld personal computers (HPC) such as the Psion and numerous Windows-CE™ machines, various Internet capable phones such as the AT&T Smartphone™ (cellular) and Screenphone (wired), set-top boxes such as WebTV™ etc. to browse the Web. Even traditional computers such as workstations, laptops and PC may vary widely in their display and specially in their network bandwidth. The browsers designed to meet the special needs of handicapped people can be modeled as client devices with specific capabilities [19]. For example, a speech browser for the blind may be modeled as a device that only supports audio. Thus, we see that to fulfill the promise of universal access to the Internet, devices with very diverse capabilities need to be catered to.

Currently, the system considers the following client device characteristics.

1) *Screen* size i.e., width and height in pixels, color and bits/pixel.
2) Effective *Network bandwidth*.
3) *Payload* defined as the total amounts of bits that can be delivered to the client for the static parts of a Web document. For streaming media this includes only the initial buffer space required before the media starts playing, not the size of the media itself. The payload is defined as the product of the network bandwidth and the time the client is ready to wait (bandwidth* wait-time) before the complete Web page downloads. For storage constrained devices, the payload will be defined as the storage space.
4) *Capabilities* for displaying video/audio/image.

B. Content

We will restrict our discussion to Web pages. The content is authored in XML [21], which is converted to HTML prior to delivery. We are also working on an extension to HTML that allows authors to introduce more information for content customization using XML and also enables our content adaptation system to be deployed at proxies.

A multimedia Web ***document*** W is composed of a number of component ***items*** $A, W = \{A_i\}, i = 1, n$. Each item can be an atomic unit of media, such as an image or a video clip. An item can also be composed of other items, for example a document can have a number of stories as content items, and each story item may be composed of image items, text items, etc. For simplicity, we will first consider only atomic content items, and then, in Section V-B, deal with composite items.

C. Content Analysis

The authored content is analyzed to extract information that will be useful in transcoding and customization. Two types of content analysis are performed.

Each atomic item A of the document is analyzed to determine its resource requirements. The types of resources considered are those that may differentiate different client devices. We determine the following resource requirements.

1) Static content size in bits.
2) Display size such as height, width and area.
3) Streaming bit-rate.
4) Color requirements.
5) Compression formats.
6) Hardware requirements, such as display for images, support for audio and video.

The semantics of the content items are determined in the context of the entire document. We currently analyze images to determine their type and purpose [22], [23]. This analysis allows us to improve image transcoding by selecting policies according to image type and purpose [22].

D. InfoPyramid

The InfoPyramid [25] is a framework for aggregating the individual components of multimedia content with content-descriptions, and methods and rules for handling the content and content descriptions [24]. The InfoPyramid describes content in different modalities, at different resolutions and at multiple abstractions. In addition, it defines methods for manipulating, translating, transcoding, and generating the content. We use InfoPyramids to represent content at multiple

modalities and resolutions so that it can be rendered on a variety of devices. Fig. 1 shows a simplified InfoPyramid for a video.

Multimodal: Multimedia content is usually not in a single media format, or modality. A video clip can contain raw data from video, audio in two or more languages, and closed captions. In the medical arena, MRI, CT, PET, and ultrasound can be captured for the same patient, resulting in multiple three-dimensional (3-D) scans of the same content.

For certain devices, the appropriate content modality may not be available. The required modality may be generated by transforming other modalities. For example, a video clip can be transformed into images showing keyframes [36], while text can be synthesized into speech.

Multiresolution: Each content component can also be described at multiple resolutions. Numerous resolution reduction techniques exist for image and video. Features and semantics at different resolutions can be obtained from raw data or transformed data at different resolutions, thus resulting in a feature or semantic pyramid.

Multiple-Abstraction Levels: The abstraction levels describe features and data in a hierarchical fashion. For example, one hierarchy could be features, semantics and object descriptions, and annotations and meta-data. For content adaption, we store meta-data such as size, color, bandwidth requirements, publisher preferences, etc., for each constituent element. This meta-data may be supplied by content analysis (Section II-C) and/or by the content author.

Methods and Rules: Methods generate content descriptors from the features of the data, or analyze, manipulate, provide modality translation, or process the data in various ways. In addition, the InfoPyramid may have rules to provide flexible application of the methods. Methods and rules provide linkage between different modalities, resolutions and abstractions. For content adaptation, we consider procedures and rules for translating and summarizing (transcoding) between modalities and resolutions.

The InfoPyramid concept can be further generalized by using other axes such as fault/loss tolerance, numerical complexity, interaction modality, etc. Rather than forcing a strong separation between the data and the content description meta-data, the InfoPyramid offers a continuum between the data, various abstractions of the data, and content description data.

Definitions: From each *original* item A_i, in the Web document $W = \{A_i\}$, an ***InfoPyramid*** $M_i = \{M_{ij}\}, j = 0, m_i$, is computed by transcoding A_i into j ***versions*** with different resolutions and modalities.[1] We will denote the original version by $M_{i0} = A_i$. We also introduce a null version, which corresponds to the item being deleted from the delivered content, by $M_{im_i} = \phi$.

E. Transcoding

Content transcoders populate the InfoPyramid structure with multiresolution, multimodal versions of the content. For example, in Fig. 2, the video is transformed to images by extracting a set of key frames [36]. Audio is also extracted from the video. Each of the modalities is then represented at different resolutions, bit-rates, color depth, etc. We have implemented a number of transcoding modules for handling video and images and imported others for text, images, video and audio. The system is designed to allow third-party content transcoders to be plugged in. The capabilities of the typical client devices and content analysis are used to guide the transcoding process. The transcoding is done off-line, unlike in previous proxy-based systems [7]–[18].

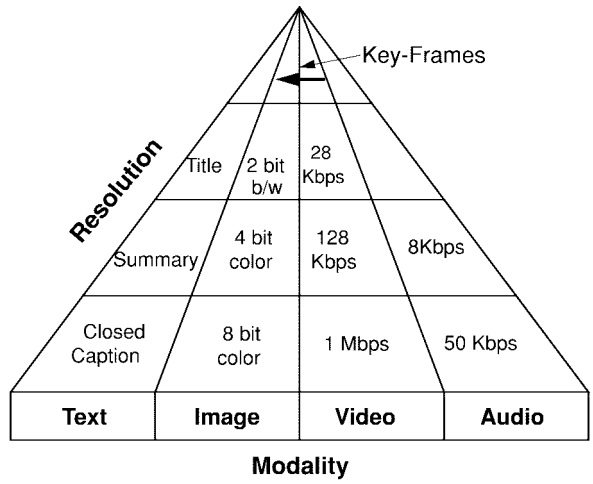


Fig. 2. An InfoPyramid for a video item.

F. Customization

The customization module uses the client device characteristics as constraints to pick the best content representation. The best representation is the one that maximizes content value for that client device. This customization process is detailed in Sections III–V.

The InfoPyramids represent the transcoded resolutions and modalities of the component multimedia items. From the InfoPyramids, the customization module selects the final ensemble such that it optimally satisfies all the client's resource constraints. This content selection is performed dynamically in response to a request. Thus, the customization is able to account for any time varying client resources such as effective bandwidth and storage.

The customization utilizes a value-resource framework, which is generalization of rate-distortion (Section III). We then solve the problem of generating a version of a Web document that provides the most "value" to a client within the client's resource constraints. In Section IV, we model the selection problem as one of optimal allocation of the resources on the client among the different versions of the multimedia items of the Web document. We show that different models for the relationship between the value and the resource requirements lead to different optimal resource allocation strategies. In Section V, we present extensions to the optimization process to 1) account for the importance of each item and 2) to jointly satisfy different class of resources, such as display area and bandwidth.

[1] In the following discussion, we will often use "*item i*" as a shorthand for "*InfoPyramid of the item i.*"

G. Cache

When a customized Web page is delivered to the client, it is also stored in a cache. When the system receives a request for a document, it first checks if a client with the same capabilities made the request previously, and if so, retrieves the corresponding customized. Temporal variations in resources on the client, such as bandwidth, CPU resources, storage, etc., will reduce the cache hit ratio. To effectively handle this, the cost of performing customization versus the variation in the resources will need to be considered. Our system currently performs customization again if the resources for the requesting client differ from the cached versions. Alternatively, one can group clients with very similar capabilities under the same client id. We will also explore the possibility of storing partial InfoPyramids based on customizations performed for clients, and using these to for subsequent customizations, thus reducing the search space for the customization.

III. Content Value

Image or video compression can be viewed as adapting the content to meet bit resource constraints. One framework for compressing to meet bit resource constraints [26], [28] has built on the rate-distortion (R-D) theory due to Shannon [27]. Rate-distortion theory deals with the minimum bit-rate R needed to represent a source with desired distortion D, or alternately, given a bit-rate R determining the distortion D in the compressed version of the source. The rate-distortion framework is employed in many image and video compression systems, for example [26], [28]–[30], [33]. We generalize rate-distortion theory to a ***value-resource*** framework by considering different versions of a content item in an InfoPyramid as analogous to different compressions, and different client resources as analogous to the bit-rate.

Distortion is typically measured as the mean squared error (MSE) between the source and its compressed version. One problem with the MSE based distortion measure is that it may not correspond to the perceived loss of fidelity [31]. However, a bigger drawback is the difficulty of formulating a meaningful distortion measure when the adaptation is drastic. For example, it is difficult to measure the loss of fidelity when a video is transcoded to a set of key frames or transcoded into its textual transcript.

To overcome this problem, we introduce a subjective measure of fidelity which we call ***value***.

Definition: Value

$$V(M_{ij}) = \frac{\text{perceived value of transcoded version } M_{ij}}{\text{perceived value of original } M_{i0}}.$$

$$V \in [0,1], \quad V = \begin{cases} 1 & \text{for original item } M_{i0} \\ 0 & \text{when the item is excluded } M_{im_i}. \end{cases}$$

The benefit of V is that we have a measure for fidelity that is applicable to transcodings of media at multiple resolutions and multiple modalities. This also allows us to compare document items that were in different media types. However, the drawback is that we still do not have a computational mechanism for determining V. The value V can either be assigned by the author for each transcoding, or we can assume some arbitrary functional relation between V and R, the resource utilized. In the special case where we can measure the distortion D of all the versions, and the distortion for the null version is assumed to be infinite, we have $V = (1/1+D)$.

The value/distortion is neither an easily estimated metric, nor is it uniform across different people with diverse interests. In general, it will also be difficult to manually assign values to different transcodings. The content value is a useful construct that helps us analyze various dynamic content adaptation policies in a theoretical rate-distortion based framework and draw parallels with compression.

IV. Resource Allocation

We can then model the content adaptation as the following resource allocation problem:

$$\max \left\{ \sum_i V_i \right\} \text{ such that } \sum_i R_i \leq R_{\text{client}} \tag{1}$$

where $V_i \in \{V(M_{ij})\}$ and $R_i \in \{R(M_{ij})\}$ are the values and resources used by the ith item M_i of the multimedia document. While V_i and R_i are discrete, we will first consider them to be continuous, and then deal with the discrete case. R_{client} is the maximum resource available at the client.

Let the value V_i be some function of the resource, R_i, i.e. $V_i = f_i(R_i)$. We convert the above *constrained optimization* problem to an *unconstrained* optimization problem by considering the Lagrangian [32]:

$$L(R,\lambda) = \left\{ \sum_i V_i + \lambda \left(R_{\text{client}} - \sum_i R_i \right) \right\} \text{ with } \lambda \geq 0.$$

Then if R_0 is an optimal solution, there exists a λ_0 such that $\Delta L(R,\lambda) = 0$. Given that the items, and thus their values, are independent of each other, we get $(\partial L/\partial R_i) = (df_i(R_i)/dR_i) - \lambda$. Therefore, the candidate solutions to (1) are given by

$$\frac{df_1(R_1)}{dR_1} = \frac{df_2(R_2)}{dR_2} = \cdots = \frac{df_i(R_i)}{dR_i} = \cdots = \lambda. \tag{2}$$

A. Analytic Functions

Content value, as an alternative to distortion, makes it possible for authors or users to specify value judgements about various transcoded versions of the content. However, manually assigning the values is not a practical proposition in most scenarios. To mitigate this problem, we introduced functional mappings between content value and resource utilization. This is not to suggest that there actually exist such a simple mechanism for assigning value (or distortion). Computing distortion, even in specific modalities such as images, that is meaningful perceptually over all images and people is not easy [31]. Our framework allows one to design fast adaptation policies for a combinatorial resource allocation problem, by assuming a particular functional mapping that captures the general trend of reduction in value with resource utilization. Fig. 3 shows a table for example values obtained

	Workstation/LAN	PC/Dialup	TV Browser	Gray PDA	BW PDA	Text Browser
Image						"bridge"
Color	24 bit color	24 bit color	256 colors	4 bit gray	B/W	
Size	256x256	192x192	128x128	96x96	64x64	
Bits	34KB	23KB	8KB	4KB	0.6KB	0.01KB
Value = RUF*Bytes	1.0	0.68	0.24	0.12	0.02	0.0003
Value = RUF* ln(Bytes)	1.0	0.96	0.86	0.79	0.61	0.22
Value = RUF* Sqrt(Bytes)	1.0	0.84	0.48	0.34	0.13	0.017

Fig. 3. Table showing value V for different functional relationships between value and resource used in terms of bytes.

using different functional relationships with the resource in bits (payload).

Let us assume a function $f, V_i = f_i(R_i)$. Note that f, and therefore, the solution, is dependent on the choice of units for R. If f is concave, (2) will give us the optimal solution. We will first consider the case when f is not concave and then the case when it is.

Nonconcave: We will limit our discussion to the case when f is either linear or convex. Let us assume that the value of an item is linearly proportional to the resource that it utilizes i.e. $V_i = c_i R_i$. From the definition of V, we have that $f_i(R_i) = 0$ when item i is absent from the delivered document i.e. $R_i = 0$ and $f_i(R_i) = 1$ for the original version of item i i.e. $R_i = R_i^{\max}$. Thus, $c_i = (1/R_i^{\max})$. We term c_i to be the *resource utilization factor* (RUF) because it measures how well the item i utilizes its resources to deliver value. It is easy to see that a greedy algorithm that allocates resources to items in the order of their RUF's gives the optimal resource allocation:

1) store items in order of decreasing RUF, c.
2) starting with the item with the largest RUF, allocate the maximum resources that each item can use until all the resources are depleted.

Similarly, the optimal resource allocation for any convex function f is also the greedy algorithm.

Concave: Let us consider the concave function $V_i = c_i \ \ln(1+R_i)$. We have defined f on $1+R_i$ to avoid negative V_i. For simplicity, we assume that $R \gg 1$ for most versions, and that $R = 1$ is equivalent to the item being deleted, giving $V_i = c_i \ \ln(R_i) = (\ln(R_i)/\ln(R_i^{\max}))$. We now get a RUF of $c_i = (1/\ln(R_i^{\max}))$. Using (2), we see that the resources are distributed among the items in proportion to their RUF's. Since, $V_i = c_i \ \ln(R_i)$ is concave (and the constraint is linear), this solution is optimal. In a similar vein, (2) will give us the optimal solution for all other concave functions.

Discrete Values: Since each item is transcoded into a limited number of versions, we may have no version that uses exactly the same resource as computed in the optimization process above. To account for the discrete values, we use the following algorithm.

1) For each item i, let R_i' be the resource selected by the optimization process. Select version j such that $R_{ij} < R_i'$ and $R_i' - R_{ij}$ is minimum.
2) Order the items in order of decreasing RUF's. Starting from the item with the highest RUF, while there are any resources left, assign to each item the version with the next higher value.

Step 2 needs to be performed only once.

B. Arbitrary Functions

When the values V_i are assigned, say by the author, we adapt a technique by Shoham and Gersho [33]. For each InfoPyramid M_i of each item i, we plot the value V_{ij} versus the resource utilized R_{ij} of each version M_{ij}, as illustrated in Fig. 4. The optimal version M_i' is given by sweeping a line with slope λ, from the top-left to the bottom-right, until it meets the concave hull of these points. As shown by (2), and in [33], the optimal solution is given by the same slope λ for all the different items i. As in [33], we perform a binary search for λ such that $\Sigma_i \ R_i$ is close to, but less than R_{client}. Points outside the concave hull are not in the solution space. For example, a text transcript of video may take more screen space but have less value, so it is out of the solution space. Note that if V is denoted in terms of D, as in (2), this resource allocation strategy becomes equivalent to the one presented in [33].

V. Extensions

Next we consider the extensions of the resource allocation strategies discussed in the previous section to account for

1) priorities on content items;
2) hierarchical or composite items;
3) multiple classes of resources;
4) mutually dependent items.

A. Priorities

In the resource allocation strategies discussed in Section IV, no matter how the value to resource relationship is defined, the items with the least resource requirements for their original versions (i.e., with the highest RUF) get precedence in the allocation of resources. Thus, when considering the bandwidth or computational resources, text items will always be assigned resources ahead of image items, and smaller images will get precedence in resource allocation over larger images.

The author of the Web document may have a mental priority ordering of the items in the document that is different from that given by their RUF's. Consider, for example, a news Web page that has one color photograph of the event covered in the news story. The page also has a large number of small images used for decorative purposes. When the news story is adapted for a client with low bandwidth or small screen size, all the resources may get allocated to the decorative images and the image central to the story may not get delivered.

Thus, we need to extend our content adaptation model to account for priorities on the content items of a document. The priorities may be assigned by the author of the page, as is the case above. Many Internet applications, such as search engines, customized news sites, etc., generate documents dynamically in response to a user request. In these applications, there is often a priority implicitly assigned to the items. For example, in image search engines, the match scores of the returned images serve as priorities. When the result page consisting of the matched images is returned to a client with low bandwidth or screen area, the images should be reduced or removed on the basis of their match scores an not their sizes (in terms of area or bits).

Let P_i be the priority assigned to item i by the author or the application. We then define the *prioritized value* of item i as $V_i^P = P_i V_i$. The goal is now to find $\max_i\{\Sigma_i\ V_i^P\}$ such that $\Sigma_i\ R_i \leq R_{\text{client}}$.

Using this formulation, the following resource allocation strategies mirror those described previously in Section IV, but with prioritized values replacing RUF's.

1) If f is linear or convex, the resources are assigned in a greedy manner in order of the prioritized values of the items.
2) If f is logarithmic, the resources are assigned in proportion to the prioritized values of the items. In general, when f is concave, we can apply the technique outlined in Section IV-A.

When f_i is not analytic, the value versus resource plot (Fig. 4) is replaced with $V_i^P = P_i V_i$ on the Y-axis. The rest of the algorithm is as described previously in Section IV-B.

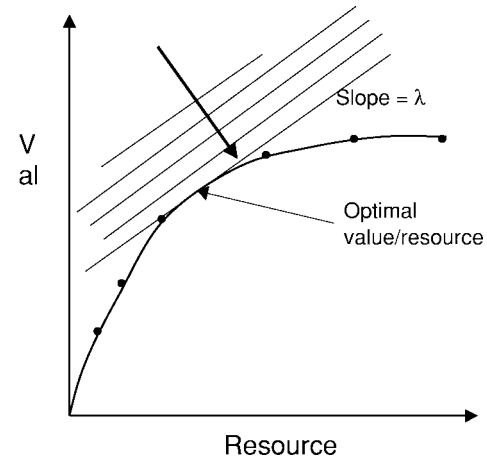


Fig. 4. Optimal value is obtained at the point where a line with slope λ meets the concave hull. The version selected has value closest to this optimal.

One common transcoding practice [7]–[13] is to scale the size of all images by a constant factor (0.75, 0.25, etc.). One can argue that in the original Web document, the larger images were more important as more resources were given to them. We can model these intrinsic priorities as proportional to ln(image size). Then this *ad-hoc* policy of constant scaling is equivalent to allocating the client resource of screen area in proportion to the prioritized values of the images and is optimal with the assumption that image value is a logarithmic function of its size.

B. Composite Items

Each content item i can be a *composite item*, i.e., a hierarchy of other content items. To account for composite items, we allocate resources using $V_i = f_i(R_i)$ where f is a concave analytic function and the items i under consideration may be composite. The optimal resource R_i' thus allocated to each composite item i is in turn used as the resource constraint for its constituent items. We then allocate this resource R_i' among the children of the composite item i. This resource allocation is repeated until the items being considered are atomic. When we have priorities assigned to items, we similarly modify the resource allocation strategy described in Section V-A.

For a composite item, the number of its different versions is combinatorial in the number of its children item. Thus, it is not practical to manually assign values to each version of a composite item.

C. Multiple Resources

A client may have a different number of *capabilities* and *resources*. We term capability to be the ability to handle a particular media type. For example, a hand-held PC (HPC) may not be capable of displaying video, and a PDA may not be capable of displaying color images. Before we start the resource allocation process, we remove from consideration all the versions of items that a client is not capable of handling.

The resources of a client can typically be divided up among several items. Examples of resources are

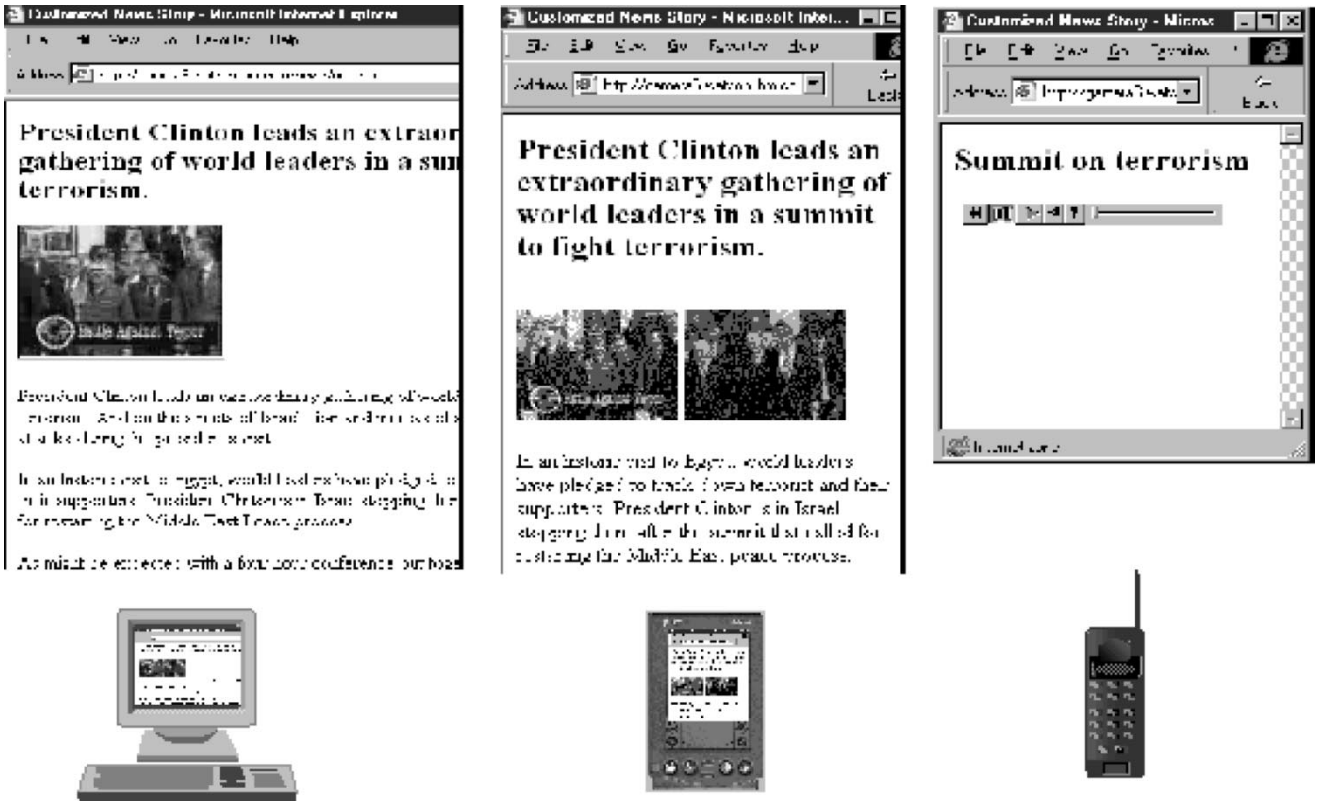


Fig. 5. The workstation gets the full text of the story and streaming video. The PDA gets a text summary and two keyframes in 2-bit grayscale. The cellular phone gets a short title and audio.

1) bandwidth;
2) bits as determined by the product of the bandwidth and the time a client is ready to wait to receive the complete document;
3) bits determined by the clients storage capacity;
4) screen area;
5) money the client is ready to pay for the document, etc.

Let there be r different resources R^k_{client} that we have to consider. Then, the resource allocation problem can be stated as $\max_i\{\Sigma_i V_i\}$ such that $\Sigma_i R^k_i \leq R^k_{\text{client}}$ for all $1 \leq k \leq r$.

We first allocate each resource R^k_{client} separately. Let M^k_i be the version of item M_i that is selected for resource k. We can select only one version of an item to be delivered. We achieve this by the following algorithm.

1) For each item i, find the set of resources $R_i = \{R^k(M^k_i)\}$ used by each of its versions selected for each of the resources.
2) For each item i, find the version M_{ij} that has the highest assigned value among all the versions and such that for all $k, R^k(M_{ij}) \leq R^k(M^k_i) \in R_i$. If no values have been assigned, since we define $V_i = f^k_i(R^k_i)$, the value of a version may be different for different resources. In that case, we choose a resource as the dominant resource, and use the values of the versions for that resource.

The above algorithm is guaranteed to select one version for each item without breaking the constraints of any of the r resources. However, the version so selected may not be optimal. To find the optimal set, a search (possibly combinatorial) may be required.

D. Mutual Dependence

For finding the optimal content adaptation schemes we assumed that the content items on a Web page are independent of each other. This assumption may not hold in general. For example, for a news story, if the text to the story has to be discarded due to space limitations, then delivering the pictures for the story may not be very useful. Our partial solution is to use composite items (Section V-B). We consider dependent items as composite items and allocate resources first to the composite item. This resource can then be allocated among the components of the composite item in an all-or-none manner. This solution is nonoptimal. A better solution would be to extend rate-distortion techniques for handling dependent blocks, such as [34], [35], and [26], to the value-resource framework.

VI. A Multimedia News System

We have implemented a Web server that customized Web pages to the capabilities of the client requesting them, employing the content adaptation process described above as an extension to this server [19].

For content, TV news programs are captured and parsed into stories [36]. The raw content for each news story is the video and the closed captioned text. Currently, we manually add a title to each new story. The content items are the title, the video, and the text of the news story. Based on a template InfoPyramid for news stories, these raw content items are ingested into InfoPyramids. The content is then transcoded to populate the InfoPyramids. The video content is compressed at multiple bit rates (1.14 Mbs, 128 kbs, 56

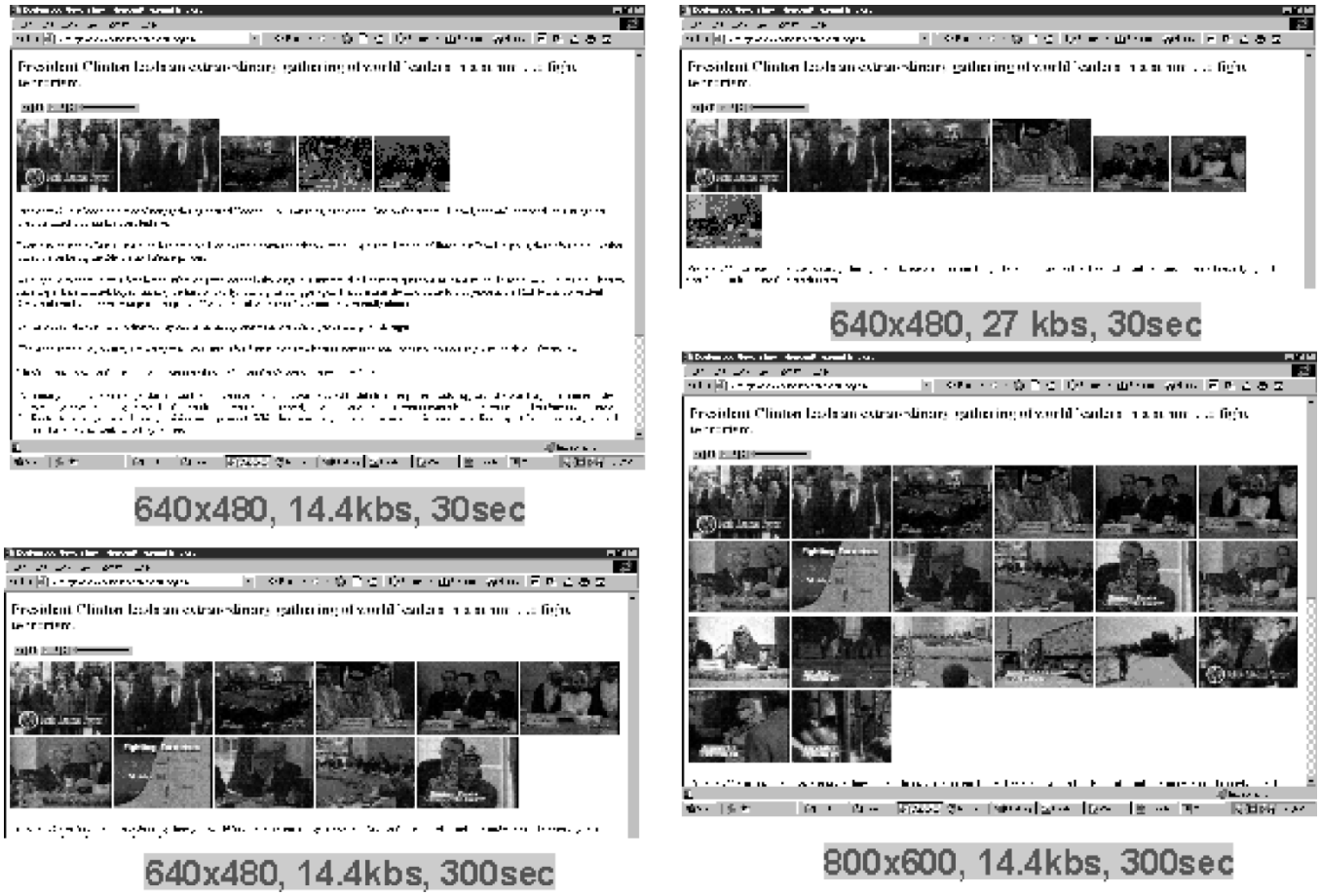


Fig. 6. Dynamic adaptation of same original content to different resources on client. Client resources are specified as *screen width* x *screen height, bandwidth,* and *wait-time*.

kbs, 28 kbs). The audio is extracted from the video and also compressed at multiple bit rates (50 kbs, 8 kbs). The video modality is also converted to images by extracting keyframes. These images are then converted for multiple color depths and resolutions (original size JPEG, 0.75 scale as 16-color GIF, 0.75 scale as gray-level GIF). The textual elements of the story are converted into summary versions. These InfoPyramids are then represented in XML (details on the XML DTD for the InfoPyramid can be found in [25]).

This multimodal, multiresolution version of the multimedia news story has many possible renditions. In our example, we have stories with over 2^{60} different possible renditions. We assigned the same default priorities for the title, video and text InfoPyramids for all the news stories. We also manually added priorities to key-frames based on our subjective judgement of their importance. Since the stories have an average of over a dozen key-frames, this allowed us to test out the effect of adaptation for both small and large differences among client resources.

Currently, the system considers client device resources of screen size, color depth, network bandwidth, maximum download time, and capabilities regarding video/audio/image display and screen color.

The web server dynamically synthesizes a rendition of the news story by selecting and combining the components of each constituent InfoPyramid such that the result both meets the client capability constraints and maximizes the content value for a given set of client resources. Fig. 5 shows example delivery of same story to a workstation on a LAN, a PDA on a CDPD modem, and a smart cellular phone using the priority-based greedy resource allocation strategy described in Section V-A. Some of the content adaptation is based on client device capabilities and some on resource allocation. For example, for the PalmPilotTM, video, audio and color image versions were filtered out prior to the resource allocation. The allocation of screen size and payload resources resulted in a summary of the text being selected and two images with the highest priorities being selected out of the 30 images.

Fig. 6 shows results of adaptation for client devices with the same capabilities but varying resources. The differences in the content delivered to these clients are only due to the resource allocation. In Fig. 6(a), a PC with VGA screen on a 14.4 kb modem gets the title and text of news story along with an audio stream and some images. We can see that images with lower priorities get fewer resources by the fact that some of them are shown at a reduced scale and color depth. If we increase the bandwidth to 27 kbs (at 28 kbs, we would select video instead of audio+keyframes), we see that some of the images from Fig. 6(a) get more resources, and some lower priority images are added. If we increase the wait-time to 300 s, increasing the total number of bits available at the client, we see that in Fig. 6(c) there are more images that get downloaded. In

Fig. 6(d), the client has the same bandwidth and wait time as in Fig. 6(c), but a larger-screen. As this client get more images delivered, we can see that the client in Fig. 6(c) was running out of screen-space before it was running out of bits.

VII. Summary

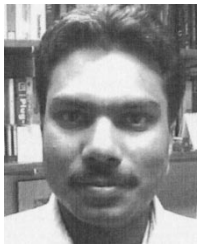
In this paper, we have presented a system for adapting multimedia Internet content. This system adapts Web content to client devices with diverse capabilities. Thus, this system enables *universal access* to the Internet by allowing different types of devices, and people with different abilities, to receive content adapted to a form suitable for them. We used InfoPyramids to represent content transcoded into multiple resolution and modalities. We extended the rate-distortion framework to optimally allocate resources on the client among the different content versions in the InfoPyramid. Finally, we presented an implementation of the content adaptation scheme.

In the value-resource framework, content adaptation is analogous to compressing multimedia documents to meet resource constraints imposed by the client device. However, unlike traditional compression, a composite document is considered and the constraints are not limited to bits or bandwidth, but also include resources such as screen size, color, cost, and hardware and software capabilities. Further, our framework can be used to analyze various content adaptation strategies, for example, the common practice of scaling images [7]–[13] by a constant factor, the greedy policy and the proportional policy. This value-resource framework also clearly shows that it is critical to assign priorities to content items in a multimedia document to get useful content adaptation polices.

The server-based content adaptation approach allows precise control over the process by the publisher. It also allows a higher level of customization, both in terms of the variety of client devices and the variety of media types than is possible by proxy based solutions. We are investigating extensions to HTML that would allow authors to annotate their HTML documents for client based customization and thus allow the deployment of the system, with comparable benefits, on proxies.

References

[1] "The future of computing; After the PC," *The Economist*, Sept. 12, 1998.
[2] Information Appliances: Market Review and Forecast, International Data Corp., Dec. 1997.
[3] S.-F. Chang, A. Eleftheriadis, D. Anastassiou, S. Jacobs, H. Kalva, and J. Zamora, "Columbia's VOD and multimedia research testbed with heterogeneous network support," *Int. J. Multimedia Tools Applicat.*, Sept. 1997.
[4] P. Bocheck and S.-F. Chang, "Content-based modeling for scalable variable bit rate video," in *Proc. IEEE Workshop Network Operating Systems Support Digital Audio Video, (NOSSDAV)*, Japan, Apr. 1996.
[5] T. V. Lakshman, A. Ortega, and A. R. Reibman, "VBR video: Trade-offs and potentials," *Proc. IEEE*, to be published.
[6] E. Amir, S. McCanne, and H. Zang, "An application level video gateway," in *Proc. ACM Multimedia 1995*, San Francisco, CA, Nov. 1995.
[7] A. Fox and E. A. Brewer, "Reducing WWW latency and bandwidth requirements by real-time distillation," in *Proc. 5th Int. WWW Conf.*, Paris, France, 1996.
[8] A. Fox, S. D. Gribble, E. A. Brewer, and E. Amir, "Adapting to network and client variability via on-demand dynamic distillation," *Proc. 7th Int. Conf. Arch. Support Prog. Lang. Op. Sys. (ASPLOS-VII)*, Cambridge, MA, Oct. 1996.
[9] A. Fox, S. D. Gribble, Y. Chawathe, and E. A. Brewer, "Adapting to network and client variation using active proxies: Lessons and perspectives," *IEEE Personal Commun.*, vol. 40, 1998.
[10] A. Fox, I. Goldberg, S. D. Gribble, D. C. Lee, A. Polito, and E. A. Brewer, "Experience with Top Gun Wingman: A proxy-based graphical web browser for the USR PalmPilot," in *Proc. IFIP Int. Conf. Dist. Sys. Plat. Open Dist. Proc. (Middleware '98)*, Lake District, U.K., Sept. 1998.
[11] T. W. Bickmore and B. N. Schilit, "Digestor: Device-independent access to the World Wide Web," in *Proc. 6th Int. WWW Conf.*, Santa Clara, CA, 1997.
[12] Intel: Quick Web. Available: http://www.intel.com/quickweb/index.htm.
[13] Spyglass: Prism 2.0. Available: http://www.spyglass.com/solutions/technologies/prism/.
[14] J. R. Smith, R. Mohan, and C.-S. Li, "Transcoding Internet content for heterogeneous client devices," in *Proc. ISCAS*, Monterey, CA, 1998.
[15] H. Bharadvaj, A. Joshi, and S. Auephanwiriyakul, "An active transcoding proxy to support mobile Web access," in *Proc. 17th IEEE Symp. Reliable Distributed Systems*, Oct. 1998.
[16] Proxinet: http://www.proxinet.com/.
[17] PalmComputing: http://www.palm.com/.
[18] AvatGo: http://www.avantgo.com/.
[19] R. Mohan, J. R. Smith, and C-S. Li, "Multimedia content customization for universal access," in *Multimedia Storage and Archiving Systems*. Boston, MA: SPIE, Nov. 1998.
[20] G. C. Vanderheiden, "Anywhere, anytime (+anyone) access to the next-generation WWW," in *Proc. Sixth Int. World Wide Web Conf.*, Santa Clara, CA, 1997.
[21] W3C XML Recommendation 1998. http://www.w3.org/XML/.
[22] J. R. Smith, R. Mohan, and C.-S. Li, "Content based transcoding of images in the internet," in *Proc. IEEE ICIP'98, Special Session on Network-Based Image Processing*, Chicago, IL, Sept. 1998.
[23] S. Paek and J. R. Smith, "Detecting image purpose in world-wide web documents," in *Proc. IS&T/SPIE Symp. Electronic Imaging: Science and Technology—Document Recognition*, San Jose, CA, Jan. 1998.
[24] ISO/IEC JTC1/SC29/WG11, MPEG-7 Requirements Document, Coding of Moving Pictures and Audio, Tokyo, Mar. 1998.
[25] C-S. Li, R. Mohan, and J. R. Smith, "Multimedia content description in the InfoPyramid," *Proc. ICASSP'98, Special Session on Signal Processing in Modern Multimedia Standards*, Seattle, WA, May 1998.
[26] A. Ortega and K. Ramchandran, "Rate-distortion methods for image and video compression," *IEEE Signal Processing Magazine*, Nov. 1998.
[27] C. E. Shannon. "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, pp. 379–423, 1948.
[28] T. Berger, *Rate-Distortion Theory: A Mathematical Basis for Data Compression*. Englewood Cliffs, NJ: Prentice-Hall, 1971.
[29] A. Ortega, "Optimization techniques for adaptive quantization of image and video under delay constraints," Ph.D. dissertation, Dept. Elect. Eng., Columbia Univ., New York, June 1994.
[30] G. M. Schuster and A. K. Katsaggelos, *Rate-Distortion Based Video Compression*. Norwell, MA: Kluwer, 1996.
[31] N. Jayant *et al.*, "Signal compression based on models of human perception," *Proc. IEEE*, vol. 81, Oct. 1993.
[32] R. Fletcher, *Practical Methods of Optimization*. New York: Wiley, 1987.
[33] Y. Shoham and A. Gersho, "Efficient bit allocation for an arbitrary set of quantizers," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 36, pp. 1445–1453, Sept. 1988.
[34] K. Ramchandran, A. Ortega, and M. Vetterli, "Bit allocation for dependent quantization with applications to multiresolution and MPEG video coders," *IEEE Trans. Image Processing*, vol. 3, Sept. 1994.
[35] T. Wiegand, M. Lightstone, D. Mukherjee, T. Campbell, and S. K. Mitra, "Rate-distortion optimized mode selection for very low bit-rate video coding and the emerging H.263 standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, Apr. 1996.
[36] R. Mohan, "Text-based search of TV news stories," in *Proc. Multimedia Storage and Archiving Systems, SPIE 2916*, Boston, MA, Nov. 1996.



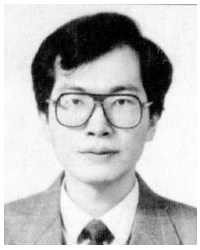
Rakesh Mohan (S'85–M'89) received the B.Tech. degree from the Indian Institute of Technology, Kanpur, India, in 1983 and the M.S. and Ph.D. degrees from the University of Southern California, Los Angeles, in 1989, all in computer science.

Since 1989, he has been a Research Staff Member at the IBM T. J. Watson Research Center, Yorktown Heights, NY. He has worked on perceptual organization, object recognition and video editing, and video indexing. His current research interests include content representation and delivery on the Internet, pervasive computing, and electronic commerce. He has served as an associate editor of the *Journal of Artificial Neural Networks* and edited the book *Progress in Neural Networks*, vol. 4 (Norwell, MA: Kluwer, 1997).

John R. Smith (S'85–M'97) received the M.Phil and Ph.D. degrees in electrical engineering from Columbia University, New York, NY, in 1994 and 1997, respectively.

He joined the IBM T. J. Watson Research Center, Yorktown Heights, NY, in 1997 and is currently a Research Staff Member. He has developed several content-based visual query systems, including VisualSEEk, SaFe, and the WebSEEk image and video search engine. He is currently investigating problems in multimedia and multidimensional data management, compression, access, and retrieval.

Dr. Smith received the Eliahu I. Jury award from Columbia University for outstanding achievement as a graduate student in the areas of systems communication or signal processing in 1997.



Chung-Sheng Li (S'87–M'91–SM'95) received the B.S.E.E. degree from National Taiwan University, Taipei, Taiwan, R.O.C., in 1984, and the M.S. and Ph.D. degrees in electrical engineering and computer science from the University of California, Berkeley, in 1989 and 1991, respectively.

He joined the Computer Science Division of the IBM T. J. Watson Research Center, Yorktown Heights, NY, as a Research Staff Member in September 1991, and has managed the Image Information System Department since 1996. His research interests include broadband applications, which include digital library, knowledge discovery and data mining; broadband network and switching, which includes all-optical networks, storage area networks, and fiber channel; broadband technologies, which include optical chip interconnects, optoelectronics, and high-speed analog/digital VLSI circuit design. He has co-initiated several research activities in IBM on fast tunable receivers for all-optical networks and content-based retrieval in the compressed domain for large image/video databases. He is currently the principle investigator of a satellite image database project funded by NASA.

Dr. Li received a Research Division award from IBM in 1995 for his major contribution to the tunable receiver design for WDMA, and numerous invention and patent application awards. He is serving as the Technical Editor and Feature Editor for the *IEEE Communications Magazine*. He has authored or coauthored more than 100 journal and conference papers and received one of the Best Paper awards from the IEEE International Conference on Computer Design in 1992. He is a senior member of the IEEE Laser Electro-Optic Society, the Communication Society, the Computer Society, and the Circuits and Systems Society.