Client-Centered Multimedia Content Adaptation

YONG WEI
North Georgia College and State University, USA
and
SUCHENDRA M. BHANDARKAR and KANG LI
The University of Georgia, USA

The design and implementation of a client-centered multimedia content adaptation system suitable for a mobile environment comprising of resource-constrained handheld devices or clients is described. The primary contributions of this work are: (1) the overall architecture of the client-centered content adaptation system, (2) a data-driven multi-level Hidden Markov model (HMM)-based approach to perform both video segmentation and video indexing in a single pass, and (3) the formulation and implementation of a Multiple-choice Multidimensional Knapsack Problem (MMKP)-based video personalization strategy. In order to segment and index video data, a video stream is modeled at both the semantic unit level and video program level. These models are learned entirely from training data and no domain-dependent knowledge about the structure of video programs is used. This makes the system capable of handling various kinds of videos without having to manually redefine the program model. The proposed MMKP-based personalization strategy is shown to include more relevant video content in response to the client's request than the existing 0/1 knapsack problem and fractional knapsack problem-based strategies, and is capable of satisfying multiple client-side constraints simultaneously. Experimental results on CNN news videos and Major League Soccer (MLS) videos are presented and analyzed.

Categories and Subject Descriptors: H.3.3 [**Information Storage and Retrieval**]: Information Storage and Retrieval—*Retrieval models*

General Terms: Algorithms

Additional Key Words and Phrases: Video personalization, video indexing, hidden Markov models, multiple choice multidimensional knapsack problem

ACM Reference Format:

Wei, Y., Bhandarkar, S. M., and Li, K. 2009. Client-centered multimedia content adaptation. ACM Trans. Multimedia Comput. Commun. Appl. 5, 3, Article 22 (August 2009), 26 pages.
DOI = 10.1145/1556134.1556139 http://doi.acm.org/10.1145/1556134.1556139

1. INTRODUCTION

The current proliferation of mobile computing devices and network technologies has created enormous opportunities for mobile device users to communicate with multimedia servers, using multimedia streams. As handheld mobile computing and communication devices such as Personal Digital Assistants (PDAs), pocket-PCs, and cellular devices have become increasingly capable of storing, rendering,

Authors' addresses: Y. Wei, Department of Mathematics and Computer Science, North Georgia College and State University, Dahlonega, GA 30597; email: ywei@ngcsu.edu; S. M. Bhandarkar, K. Li, Department of Computer Science, The University of Georgia, Athens, GA 30602-7404.

and display of multimedia data, the user demand for being able to view streaming video on such devices has increased. One of the natural limitations of these handheld devices is that they are constrained by their battery power capacity, rendering and display capability, viewing time limit, and, in many situations, by the available network bandwidth connecting these devices to video data servers. Therefore, the original video content often needs to be personalized in order to fulfill the client's request under various client-side system-level resource constraints (henceforth termed as "client-side resource constraints" in the interest of brevity). Given the client's preference(s) regarding the video content and the various client-side resource constraints, the video personalization system should be able to assemble and disseminate the most relevant video content to the mobile client(s) while simultaneously satisfying multiple client-side resource constraints. In this article, we present a client-centered multimedia adaptation system to attain this goal.

1.1 Salient Features of the Proposed Client-Centered Multimedia Adaptation System

The proposed client-constrained multimedia personalization system has the following salient features.

—*A stochastic modeling approach to automatically segment and index video streams in a single pass.* Inspired by the success of modern continuous speech recognition [Ney et al. 1999], we propose a data-driven multi-level Hidden Markov Model (HMM)-based approach to perform both video segmentation and video indexing in a single pass. Since no domain-dependent knowledge about the structure of video programs is used, the proposed approach is capable of segmenting and indexing a wide variety of videos in a manner that supports semantic content-based video retrieval.

—*A hierarchical scheme to represent multimedia content at multiple levels of abstraction*. Mobile computing devices are typically constrained by the paucity of various system-level resources and by the client's preference(s) with regard to video content. Therefore, the original video content often needs to be transcoded in order to fulfill the client request under these constraints. Transcoded versions of the video preserve the information content of the original video to various degrees. When combined with an appropriate content descriptive ontology, the video content can be represented hierarchically at multiple levels of abstraction. The resulting hierarchical video database can then be used to satisfy various client-constrained queries and requests.

—*A personalization scheme that compiles and delivers optimal video content while satisfying multiple client-side resource constraints*. Given the client's preference(s) regarding the video content and the various client-side resource constraints, the goal of video personalization is to generate, compile, and disseminate the most relevant video content to the mobile client. One of the contributions of the work is the design and implementation of a Multiple-Choice Multidimensional Knapsack Problem (MMKP)-based video personalization strategy which is shown to have significant advantages over the existing 0/1 Knapsack Problem (0/1KP)-based and the Fractional Knapsack Problem (FKP)-based video personalization strategies. The proposed MMKP-based personalization strategy is shown to include more relevant video content in response to the client's request compared to the existing 0/1KP-based and FKP-based personalization strategies. In contrast to the 0/1KP-based and FKP-based personalization strategies which can satisfy only a single client-side resource constraint at a time, the proposed MMKP-based personalization strategy is shown capable of *simultaneously* satisfying multiple client-side resource constraints.

1.2 Related Work

Semantic video indexing is the process of attaching concept terms from a video descriptive ontology to segments of a video. It is regarded as the first step towards automatic retrieval and personalization of video data, since it enables users to access videos based on their interests and preferences regarding

video content. Semantic video indexing typically involves two subprocesses which are usually performed as two separate steps, namely, temporal segmentation of the video stream and semantic labeling of the resulting video segments [Tseng et al. 2004]. In recent years, various applications of HMMs to video segmentation and video annotation have been studied [Eickeler et al. 1999; Huang et al. 2005; Li et al. 2001]. However, in most current HMM-based systems, the overall performance could be compromised due to audio-visual mismatch [Huang et al. 2005] and inaccurate domain-dependent knowledge about the video scenes and the video program structure [Eickeler et al. 1999; Li et al. 2001].

In this article, we propose a data-driven multi-level HMM-based approach to perform both video segmentation and video indexing in a single pass. The proposed approach uses only the visual features in a video stream in order to avoid potential audio-visual mismatches. The proposed approach is purely data-driven, that is, no domain-specific knowledge about the structure of the video program is needed to syntactically or semantically model the underlying video content.

Mobile computing and communication devices such as handheld computers, pocket PCs, PDAs, and smart cellular phones typically have fewer resources than desktop computers. Thus, it is often necessary to adapt or transcode the video content to suit the capabilities of the resource-constrained mobile device. The goal of traditional video transcoding schemes is to reduce the amount of resources consumed in order to receive, render, play, and view the video stream while preserving the desired level of detail. Early video transcoding techniques have typically focused on reducing the video bit rate in order to meet the available channel capacity via computation of the Discrete Cosine Transform (DCT) of the video frames [Eleftheriadis et al 2006; Nakajima et al. 1995; Sun et al. 1996] or by reducing the spatial resolution of the video frames [Zhu et al. 1998]. Temporal transcoding schemes reduce the number of transmitted video frames in order to satisfy the bit rate requirements imposed by a network, maintain a higher quality of the encoded video frames, or satisfy the viewing time limitations imposed by a client [Chen et al. 2002].

Traditional video transcoding techniques do not perform high-level analysis of the underlying video content prior to transcoding. The transcoding is based primarily on statistical analysis of low-level (i.e., pixel-level or feature-level) video content. Consequently, the transcoded output is often divorced from human visual perception of the information conveyed by the video stream. Semantic content-based video transcoding, on the other hand, takes into account the high-level semantic content of the underlying video stream prior to its transcoding. The video is transformed into static images by extracting a set of key frames whereas the accompanying audio is transcoded into text [Li et al. 1998]. Dynamic motion panoramas have also been used to represent both dynamic and static scene elements in a geometrically consistent manner [Bartoli et al. 2004; Irani et al. 1996, 1995; Wei et al. 2006].

In the proposed hierarchical video content representation scheme, videos are indexed and transcoded at multiple levels of abstraction. Multiple transcoded versions of the original video segments are generated for which the bit rate, spatial resolution, and temporal resolution of the original video segments are appropriately adapted. The proposed representation scheme is shown to enable semantic content-based video transcoding. Key shots are first extracted from the underlying video stream. Dynamic motion panoramas are used to represent the static background and dynamic foreground to reduce the data requirements for video shots taken with a panning camera. The transcoded video segments are subsequently labeled using terms selected from a video description ontology.

Various personalization strategies, such as those based on solving the 0/1KP or the FKP, can be used to generate the optimal response to a resource-constrained client's request [Merialdo et al. 1999]. Tseng et al. [2003] propose a personalization strategy termed as *context clustering* based on grouping of successive shots in a video stream that are visually similar. Context clustering is shown to be an enhancement of the FKP-based personalization scheme proposed in Merialdo et al. [1999] in that it considers the temporal smoothness of the generated video summary in order to improve the client's viewing

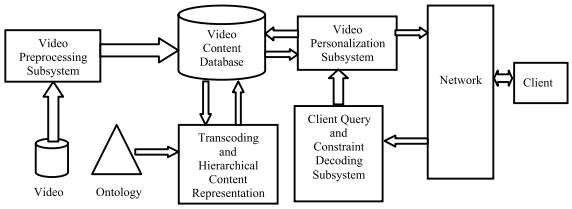


Fig. 1. Overall system architecture.

experience. In this article, we propose, implement, and evaluate a video personalization strategy modeled along solving the MMKP. We present experimental results comparing the proposed MMKP-based video personalization strategy to the existing 0/1KP-based and the FKP-based video personalization strategies.

The remainder of the work is organized as follows. In Section 2, the overall system architecture is described. In Section 3, a stochastic model-based approach to temporal video segmentation and video indexing is presented. In Section 4, a hierarchical video content representation scheme is described and various methods to measure the information content of transcoded videos relative to their original versions are proposed. In Section 5, video personalization strategies based on various versions of the classical Knapsack Problem (KP) are discussed. In Section 6, experimental results of the proposed video temporal segmentation and indexing scheme are presented. In Section 7, results of the experimental evaluation of various KP-based video personalization strategies are presented. Finally, Section 8 concludes the article with an outline for future work.

2. SYSTEM ARCHITECTURE

The client-centered multimedia adaptation system consists of the following four subsystems, as shown in Figure 1: (1) the video preprocessing subsystem, (2) the hierarchical video content representation and multi-level video transcoding subsystem, (3) the client request and client constraint decoding subsystem, and (4) the video personalization subsystem. The relationships amongst the four aforementioned subsystems are described in the following subsections.

2.1 Video Preprocessing Subsystem

The video preprocessing subsystem performs temporal video segmentation and video indexing. In order to provide mobile clients with personalized video content, the original video streams are first segmented and indexed in the temporal domain. A data-driven stochastic algorithm based on a multi-level Hidden Markov Model (HMM) is proposed to perform both video segmentation and video indexing automatically in a single pass.

2.2 Multi-Level Video Transcoding and Hierarchical Video Content Representation Subsystem

Each indexed video segment is transcoded at multiple levels of abstraction. In the proposed scheme, semantic-level transcoding based on key frame selection and motion panorama computation and low-level transcoding based on bit rate reduction, and temporal and spatial resolution reduction are closely integrated. The original video segment and its transcoded versions are deemed to constitute a multi-level *content group*. To facilitate efficient content-based retrieval, a hierarchical ontology-based description of the video content is employed. A multi-level *content group* is associated with a set of appropriate semantic terms derived from the aforementioned ontology.

2.3 Client Query and Constraint Decoding Subsystem

The client query and client constraint decoding subsystem acts as an intermediary between the video personalization subsystem and the mobile client. A client query (request) consists of the client's preference(s) with regard to video content and a list of client-side resource constraints. A client query protocol is established to facilitate the communication of the query between client and subsystem. A client query under the currently implemented protocol is a structure with two fields: *Preferences* and *Constraints*. The *Preferences* field is a list of strings representing semantic terms that encapsulate the client's request for information, whereas the *Constraints* field is a list of numerical parameters representing the client-side resource constraints such as the viewing time limit, bandwidth limit, and the limit on the amount of data the client can receive. Client queries transmitted in the format specified by this protocol are received and subsequently decoded by the subsystem. The decoded query is then forwarded to the video personalization subsystem for further processing.

2.4 Video Personalization Subsystem

The goal of video personalization is to display a video summary that preserves as much of the semantic content desired by the client as possible while simultaneously satisfying the resource constraints imposed by the (potentially) mobile client. In the video personalization subsystem, the client's video content preference(s) is (are) matched with the video segments (and their various transcoded versions) stored in the video database. In order to generate a personalized video summary, the client usage environment and the client-side resource constraints are evaluated. The personalization engine compiles an optimal video summary that is most relevant to the client's content preference(s) subject to the resource constraints imposed by the client.

3. VIDEO SEGMENTATION AND INDEXING—A STOCHASTIC MODEL-BASED APPROACH

In modern speech recognition systems, the continuous speech resulting from a spoken sentence is modeled at both the acoustic-phonetic (subword) level and the language level. The subword units are modeled by Hidden Markov Models (HMMs) [Ney et al. 1999] which have been shown to be powerful stochastic models capable of approximating many time varying random processes [Rabiner 1989]. Inspired by the success of modern HMM-based continuous speech recognition systems and HMM-based video segmentation approaches [Eickeler et al. 1999], we propose a data-driven multi-level HMM-based approach to perform both video segmentation and video indexing in a single pass.

The multi-level HMM-based segmentation and indexing algorithm is essentially a stochastic model-based segmentation algorithm wherein the input video stream is classified frame by frame into *semantic units*. A semantic unit within a video stream is a video segment that can be associated with a clear semantic meaning or concept, and consists of a concatenation of semantically and temporally related video shots. Temporal boundaries in the video stream are then marked at frame locations that represent a transition from one semantic unit to another. An advantage of the proposed multi-level HMM-based segmentation algorithm is that once the set of HMMs for a video stream are defined, future image sequences can be segmented, classified, and indexed in a single pass. Furthermore, semantic units can be added without having to retrain the HMMs corresponding to the other semantic units. Thus, the proposed multi-level HMM makes it possible to process different types of videos in a modular and extensible manner so as to enable video retrieval based on semantic content.

Instead of detecting video shots, it is often much more useful to recognize semantic units within a video stream to be able to support video retrieval based on high-level semantic content. Note that visually similar video shots may be contained within unrelated semantic units. Thus, video retrieval based purely on detection of video shots will not necessarily reflect the semantic content of the video stream.

The semantic units within a video stream can be spliced together to form a logical video sequence that the viewer can understand. In well-organized videos such as TV broadcast news and sports programs, the video can be viewed as a sequence of semantic units that are concatenated based on a predefined video program syntax. Parsing a video file into semantic units enables video retrieval based on high-level semantic content and playback of logically coherent blocks within a video stream. Automatic indexing of semantic components within a video stream can enable a viewer to jump straight to points of interest within the indexed video stream, or even skip advertisement breaks during video playback.

In the proposed scheme, a video stream is modeled at both the semantic unit level and the program model level. For each video semantic unit, an HMM is generated to model the stochastic behavior of the sequence of feature emissions from the image frames. Each image frame in a video stream is characterized by a multidimensional feature vector. A video stream is considered to generate a sequence of these feature vectors based on an underlying stochastic process that is modeled by a multi-level HMM. The advantages of the proposed approach are summarized as follows.

—*Video segmentation and video indexing are performed in a single pass*. This is extremely valuable when dealing with large amounts of video data to populate a video database. Although video segmentation and video indexing are performed offline, they are computationally intensive and often result in a serious bottleneck during the creation of a video database. The ability to perform video segmentation and video indexing in a single pass alleviates this bottleneck to some extent.

—*No domain-dependent knowledge about the structure of video programs is used.* The probabilistic grammar used to define the video program is learned entirely from the training data. This allows the proposed approach to handle various kinds of videos in a modular and extensible manner without having to manually redefine the program model.

—*Semantic unit level HMMs are used to model video units with clear semantic meanings.* The proposed data-driven approach does not need to use HMMs to model video edit effects. This not only simplifies the collection and processing of training data, but also ensures that all video segments in the video database are labeled with concepts with clear semantic meanings in order to facilitate video retrieval based on semantic content. The video edit effects within a semantic unit are considered part of the semantic unit and, as such, are not labeled separately. The HMM representation of a semantic unit can accommodate these video edit effects implicitly.

3.1 Image Features

The success of an HMM-based algorithm for video segmentation and video indexing depends greatly on the image features extracted from each frame in the video stream. These features should contain enough information to characterize each image frame, yet should capture the differences amongst the frames in distinct semantic units in order to be able to distinguish them. In this work, we use two categories of image features. The first category includes a set of simple features. The dynamic characteristics of the video are captured by the differences of successive video frames at both the pixel level and the histogram level. Various motion-based measures describing the movement of the objects in the video, including the motion centroid of the video frame and intensity of motion as well as measures of illumination change at both the pixel level, and the histogram level, are included in the multidimensional feature vector [Eickeler et al. 1999].

In the second category, Tamura features [Tamura et al. 1978] are used to capture the textural characteristics of the image frames at the level of human perception in order to improve the accuracy of temporal video segmentation and video indexing. Tamura contrast, Tamura coarseness, and Tamura directionality have been used successfully in content-based image retrieval [Flickner et al. 1995] and are defined as follows.

Tamura Contrast. Consider

$$k = \frac{1}{N} \sum_{i \in O(x,y)} (c[i] - \mu)^4 \tag{3.1}$$

where μ is the average of the color values in the neighborhood of pixel (x, y) denoted by $O(x, y)$, $c[i]$ is the color or intensity of the ith pixel in the neighborhood $O(x, y)$, and N is the number of pixels in the neighborhood $O(x, y)$. The Tamura contrast at pixel (x, y), denoted by *TCon(x, y)*, is given by

$$TCon(x, y) = \begin{cases} 0 & if\ k < \varepsilon \\ \sigma^2 / \sqrt[4]{k} & otherwise \end{cases}, \tag{3.2}$$

where σ^2 is the color covariance computed in the neighborhood of pixel (x, y) in the image frame and ε is a predefined threshold.

Tamura Coarseness. The Tamura coarseness measures the spatial scale at which the difference in color values between pixels in a local neighborhood of a given pixel (x, y) is a maximum. Given a pixel (x, y), 5 spatial scales are used to measure the horizontal and vertical differences of the mean color value. The horizontal difference and the vertical difference of the mean color values at location (x, y) at scale k are given by

$$E_H(x, y, k) = |A(x - 2^k, y, k) - A(x + 2^k, y, k)|, \tag{3.3}$$

$$E_V(x, y, k) = |A(x, y - 2^k, k) - A(x, y + 2^k, k)|, \tag{3.4}$$

where $k \in [0, 4]$, and $A(x, y, k)$ is the mean color value at pixel (x, y) for window size $(2^k + 1) \times (2^k + 1)$ when $k > 0$. The window size is 1×1 when $k = 0$.

Let us define $E(x, y, k) = \max(E_H(x, y, k), E_V(x, y, k))$, then the Tamura coarseness at pixel (x, y), denoted by *TCoar(x, y)*, is given by

$$TCoar(x, y) = \arg\{\max_k(E(x, y, k))\}. \tag{3.5}$$

The definition of Tamura coarseness given in Eq. (3.5) calls for the computation of the mean color value $A(x, y, k)$ in 20 distinct windows if 5 spatial scales are used. The computation cost is high if we perform the summation directly by enumerating the color values of each pixel in each window. Hence, we need an efficient way to compute $A(x, y, k)$. The *integral image* [Viola et al. 2004] provides an efficient way to compute the summation in a rectangular window. Given an original input image $I(x, y)$, the integral image is given by

$$J(x, y) = \int_0^y \int_0^x I(u, v) du dv. \tag{3.6}$$

For a discrete image $I(x, y)$, the integrals in Eq. (3.6) are replaced by their corresponding summations. The integral image $J(x, y)$ can be computed efficiently using the following recurrence relation.

$$J(x, y) = I(x, y) + J(x - 1, y) + J(x, y - 1) - J(x - 1, y - 1) \tag{3.7}$$

The mean color value within a given rectangular window (x_1, y_1, x_2, y_2), with corner points (x_1, y_1) and (x_2, y_2), can be computed as

$$A(x_1, y_1, x_2, y_2) = \frac{J(x_2, y_2) - J(x_2, y_1) - J(x_1, y_2) + J(x_1, y_1)}{R}, \tag{3.8}$$

(a) semantic concepts for TV broadcast news: from left to right: *News Anchor*, *News*, *Sports News*, *Commercial*, *Weather Forecast*, and *Program Header*

(b) semantic concepts for Major League Soccer (MLS) video: from left to right: *Zoom Out*, *Close Up*, and *Replay*

Fig. 2. Representative image frames of semantic units.

where R is the area size of the rectangular window (x_1, y_1, x_2, y_2). Thus the computation of the mean color value within a given rectangular window can achieved efficiently with three summation operations as shown in Eq. (3.8).

Tamura Directionality. The Tamura directionality is simply the intensity gradient orientation $\theta(x, y)$ at a pixel (x, y) and is given by

$$\theta(x, y) = \tan^{-1} \frac{I_y(x, y)}{I_x(x, y)}, \tag{3.9}$$

where the intensity gradient components $I_x(x, y)$ and $I_y(x, y)$ are computed using the *Sobel* edge operator.

3.2 HMMs for Characterization of Semantic Units in a Video Stream

In the proposed video segmentation and video indexing scheme based on semantic video content, we define six semantic concepts for TV broadcast news video, namely *News Anchor*, *News*, *Sports News*, *Commercial*, *Weather Forecast*, and *Program Header*, and three semantic concepts for Major League Soccer (MLS) video, namely *Zoom Out*, *Close Up*, and *Replay*. Representative images for each of these semantic concepts are shown in Figure 2(a) and 2(b). An HMM is formulated for each individual semantic concept. The optimal HMM parameters for each semantic unit are learned from the feature vector sequences obtained from the training video data. The standard HMM training procedure based on the Baum-Welch algorithm [Baum et al. 1970] is used. The HMMs for individual semantic units are trained separately using the training feature vector sequences. This allows for modularity in the learning procedure and flexibility in terms of being able to accommodate various types of video data. When new video data for a semantic unit are presented, we only need to retrain the corresponding HMM for the relevant semantic unit without having to retrain any of the HMMs corresponding to the other semantic units. Since the states in an HMM are hidden, researchers typically use heuristics to guess the correct HMM topology [Boreczky et al. 1998]. In our work, we adopt a universal left-to-right HMM topology (i.e., one without any backward state transitions) with continuous observations of the feature vector emissions. The distribution of the feature vector emissions in the HMM is approximated by a mixture of three Gaussian distributions in all of our HMM implementations. Since little is known about the underlying physical processes which generate the observable visual features in the video stream, using a universal left-to-right HMM topology with a three Gaussian component mixture as a default choice makes it easy to construct semantic unit HMMs for unknown data without prior detailed investigation into the underlying feature generation process [Eickeler et al. 1999; Shinoda et al.

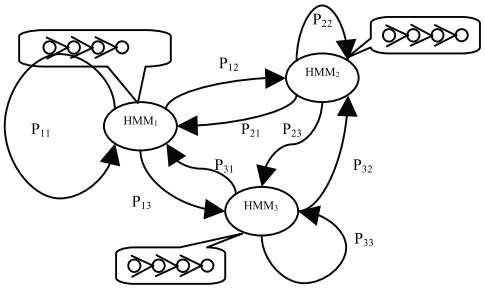


Fig. 3. Concatenation of the individual HMMs.

2005]. Furthermore, the proposed approach to HMM design can be used, without any modification, to recognize new semantic units within a video stream.

3.3 Multi-Level HMM for Single-Pass Video Segmentation and Indexing

The search space for the proposed single-pass video segmentation and video indexing procedure is characterized by the concatenation of the HMMs corresponding to the individual semantic units. The HMM corresponding to an individual semantic unit essentially models the stochastic behavior of the sequence of image features within the scope of that semantic unit. Transitions amongst these semantic unit HMMs are regulated by a prespecified video program model. Figure 3 depicts the concatenation of the individual HMMs corresponding to the three semantic units comprising the video program model. The topologies of the individual HMMs are described in the callouts in Figure 3. The parameter p_{ij}, $1 \leq i, j \leq 3$ is the transition probability from semantic unit i to semantic unit j. The transition probability matrix $P_{3\times3}$, where $P_{ij} = p_{ij}$, $1 \leq i, j \leq 3$, essentially defines the video program model.

A data-driven approach is proposed to estimate the video program model directly from the training data using sequential maximum likelihood estimation; that is, no domain-dependent knowledge about the structure of the video program is used. Most researchers use domain-specific knowledge about the video program in order to determine the video program model [Eickeler et al. 1999; Huang et al. 2005; Li et al. 2001]. This knowledge-driven approach becomes untenable as the size of the semantic unit vocabulary and the complexity of video program increase. The accuracy in the estimation of the video program model directly affects the segmentation and indexing results. Statistical Language Models (SLMs) are typically represented by n-gram models [Brown et al. 1992]. For large values of n, a correspondingly large amount of training data is required to estimate the n-gram model parameters, resulting in a computationally expensive training procedure. Therefore, in this work, the video program is represented by a 2-gram model determined by the conditional probability of the semantic unit sequence given a sequence of image feature vectors as shown in Eq. (3.10) [Brown et al. 1992]. The training data for estimation of the parameters of the video program model are assumed to be manually prelabeled.

The single-pass video segmentation and video indexing procedure is formulated in terms of the following Bayesian decision rule: Given a sequence of image feature vectors $f_1 \ldots f_T$, determine a semantic unit sequence $U_1 \ldots U_N$ such that the conditional probability of the semantic unit sequence given the sequence of image feature vectors is maximized, that is,

$$\max(\Pr(U_1 \ldots U_N \mid f_1 \ldots f_T)) \sim \max(\Pr(U_1 \ldots U_N) \bullet \Pr(f_1 \ldots f_T \mid U_1 \ldots U_N)). \tag{3.10}$$

In Eq. (3.10), $f_1 \ldots f_T$ are the feature vectors extracted from the image frames in the video stream to be segmented and indexed and $\Pr(U_1 \ldots U_N)$ is the video program model. The video program model essentially regulates the transition probability from a predecessor semantic unit to a successor semantic

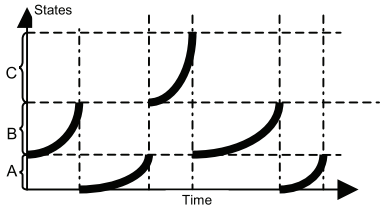


Fig. 4. Single-pass segmentation and indexing of a video stream containing semantic units A, B, and C.

unit. The Viterbi algorithm [Forney et al. 1973; Viterbi et al. 1967] is used to determine the optimal path in the concatenation of the HMMs. Figure 4 depicts the single-pass segmentation and indexing of a video stream containing the semantic units A, B, and C where the y-axis represents the hidden states within the HMM for an individual semantic unit. The bold curves in Figure 4 indicate the change of states within the semantic units A, B, and C. The video stream in this example is segmented into a semantic unit sequence *BACBA*. Bold curves within a semantic unit are monotonically nondecreasing because the HMMs for the individual semantic units have a strict left-to-right topology with no backward-going state transitions. Note that although a two-level HMM is used in the current implementation, the proposed technique for single-pass segmentation and indexing of video can be generalized to a multi-level HMM with more than two levels.

4. HIERARCHICAL CONTENT REPRESENTATION

In the proposed system, each indexed video segment is summarized or transcoded at multiple levels of abstraction using algorithms for content-aware key frame selection and motion panorama generation. The transcoded versions have different requirements in terms of the client-side resources needed to receive, transmit, render, and view the transcoded video. A clustering algorithm is used to parse video segments into shots where the interframe histogram difference measure is used to identify the shot boundaries in a video segment. A temporally localized 2-class (i.e., binary) clustering algorithm is used to detect shots within a video segment. For a temporal window $w[t_1, t_2]$, we perform 2-class clustering to separate frames in the window into classes c_1 and c_2. A rejection threshold R is set such that if the distance between c_1 and c_2 is less than R, then the width of the temporal window is increased to $[t_1, t_3]$ where $t_3 > t_2$ and the clustering repeated.

In the key frame-based transcoding scheme, each shot is represented by a set of key frames. Frames within a shot are clustered into groups where each group is represented by a key frame. The level of abstraction of the video summary is controlled by selecting a threshold value for the group size. The smaller the threshold value, the more detailed the resulting summary and vice versa. Thus, each video summary consists of a set of key frames. If the image frames are displayed at a fixed frame rate, the higher the level of abstraction, the shorter the duration of the video summary. The number of levels of abstraction associated with the transcoded video is set to three and the relative time durations of the transcoded video segments set to 100%, 50%, and 20% of the time duration of the original video segments. The proposed MMKP-based personalization strategy permits additional levels of abstraction and transcoding methods to be incorporated if needed without any modifications to the other parts of the overall system.

In the case of video shots containing dominant panning camera motion (i.e., pan shots), motion panoramas based on image mosaicking are an efficient representation of the video shot [Bartoli et al 2004]. Pan shots can be detected based on the underlying pattern of the Motion Vectors (MVs) [Bhandarkar et al. 1999]. Let θ_{ij} be the direction of the MV associated with the ijth pixel. Let $\theta_{avg} = \frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} \theta_{ij}$

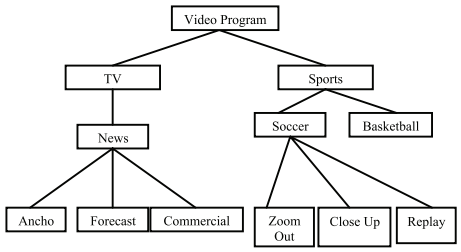


Fig. 5. Three-level hierarchy of semantic concepts.

be the average of the MV directions in the frame. For a frame to qualify as a member of a pan shot, the variance $\sigma_\theta^2 = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}[\theta_{ij} - \theta_{avg}]^2$ should be less than a predefined threshold.

The motion panorama construction algorithm consists of three major phases: static background generation, background-foreground segmentation (i.e., extraction of moving objects), and final panorama composition where the foreground objects or regions are pasted back onto the static background. If the motion panorama is encoded to have the same frame size as the original video shot, then there is a significant saving in terms of the amount of data to be transmitted and the required bandwidth or bit rate. In the case of motion panorama-based transcoding, the frame(s) corresponding to the stationary background need(s) to be transmitted only once or very infrequently. The frames corresponding to the dynamic foreground need to be transmitted in the form of a motion overlay at the required frame rate [Bartoli et al 2004]. Since the dynamic foreground regions are relatively few in number, the bandwidth requirement of the motion overlay is much lower than that of the original video shot [Bartoli et al 2004].

Video segments are labeled using semantic concepts selected from a video description ontology which, in our current implementation, is a three-level hierarchy of semantic terms, depicted in Figure 5. The first level in the ontology represents the video category (*TV Broadcast, Sports, Surveillance*, etc.), the second level represents the video program group (*Broadcast News, Soccer, Basketball, Traffic Surveillance*, etc.), and the third level defines the various semantic concepts (*Anchor*, *Weather Forecast*, *Commercial*, *Replay*, *Closeup*, etc.). The hierarchical nature of the video description ontology provides a structural framework for representation and storage of the video segments and their transcoded versions. The video description ontology enables the generation of an optimal personalized response to a client's request; one that best matches the client's preference(s) with regard to the video content while simultaneously ensuring that the various client-side resource constraints are satisfied.

In order to evaluate the performance of various video personalization strategies, the relationship between the information content of the original video and the amount of information retained in its various transcoded versions needs to be established. This is a complex task due to the inherent difficulty in quantifying the amount of information contained within the original video and due to the diverse nature of the various transcoded versions of the original video. Although Shannon entropy has been used as a measure of pixel-level or feature-level information content of a video [Snoek et al. 2003], the relationship between the low-level feature-based entropy measure of a video stream and its high-level semantic content has not been firmly established.

Although there are many factors that determine the information content of a video, it is reasonable to assume that the amount of information or detail contained within a video summary is related to its duration. For each video segment, its original version is assumed to contain the greatest amount of detail, whereas its summary at the highest level of abstraction is assumed to contain the least amount of detail. Typically, the amount of information contained within a video summary (relative to original version) does not necessarily increase linearly with its relative duration. In this article, we assume that

the relevance value of a video summary (with respect to a client's query) is a function of the relevance value of the original video segment and the relative length of the video summary, that is,

$$v_i = v_{i0} \cdot f(L_i/L_0), \tag{4.1}$$

where v_{i0} is the relevance value of the original segment, and L_0 and L_i are the time durations of the original segment and the summarized (or transcoded) video segment, respectively. The function $f(L_i/L_0)$ represents the relationship between the amount of information contained within a video summary relative to the original segment.

We propose to use empirical laws to quantify the relationship between the amount of information contained in the transcoded videos relative to the original video. The *Zipf* function, *sigmoid* function, and *Rayleigh* distribution are proposed as plausible mapping functions for quantifying the relationship between the amount of information in the transcoded video relative to the original video, and are shown suitable for different kinds of videos. The performance results of the personalization subsystem presented in Section 6 are shown to vary significantly when different empirical mapping functions are used to measure the amount of information contained in the transcoded video relative to the original video.

4.1 The Zipf's Law-Based Mapping Function

For some categories of videos, such as broadcast news, most of the information is revealed or summarized in a video segment spanning the first 20%–30% of the video stream. This observation justifies the use of the *Zipf* function to quantify the relationship between the amount of information contained in the transcoded (or summarized) videos relative to the original video. The mathematical definition of the *Zipf* function [Wheeler 2002] is given by

$$I = H_{k,s}/H_{N,s}, \tag{4.2}$$

where I (expressed as a percentage) is the amount of information contained within a video summary relative to the original video segment, N is the set of all possible discrete durations of the video summary, $k \in N$ is the duration of a video summary, $s > 0, s \in R$ is the characteristic parameter of the *Zipf* function, and $H_{k,s}$ is the kth generalized harmonic number. When $s = 0$, the information content of a video summary increases linearly (i.e., at a constant rate) with its duration.

Eq. (4.2) is a definition of the discrete *Zipf* function, whereas in our application, the relative (i.e., normalized) duration of a video summary is a continuous variable in the range [0, 1]. To use the *Zipf* function defined in Eq. (4.2), the following approximation and linear transform are used. Let $L_{norm} \in \{0.01, 0.02, \ldots, 0.99, 1.00\}$ denote the normalized discrete video duration and let $N = 100$. The linear transform that maps the values of L_{norm} to k is given by

$$k = round(L_{norm} \times N). \tag{4.3}$$

Figure 6 shows a plot of the relative information content of a transcoded video versus its normalized duration. The relative duration and relative information content of the transcoded video are normalized to lie within the range [0, 1] based on the duration and information content of the original video, respectively. The parameter s is set to values 0, 0.5, 1.0, and 1.5, respectively, where s = 0 denotes a special case when the *Zipf* function degenerates to a linear mapping. The derivative of the *Zipf* function in Eq. (4.2) can be considered as the incremental information ΔI introduced by a video segment whose duration is incremented by ΔL and is given by

$$I' = \lim_{\Delta L \to 0} \frac{I(L + \Delta L) - I(L)}{\Delta L} = \frac{1/k^s}{H_{N,s}}. \tag{4.4}$$

Figure 7 shows the plot of the function I' when the linear transform in Eq. (4.3) is performed.

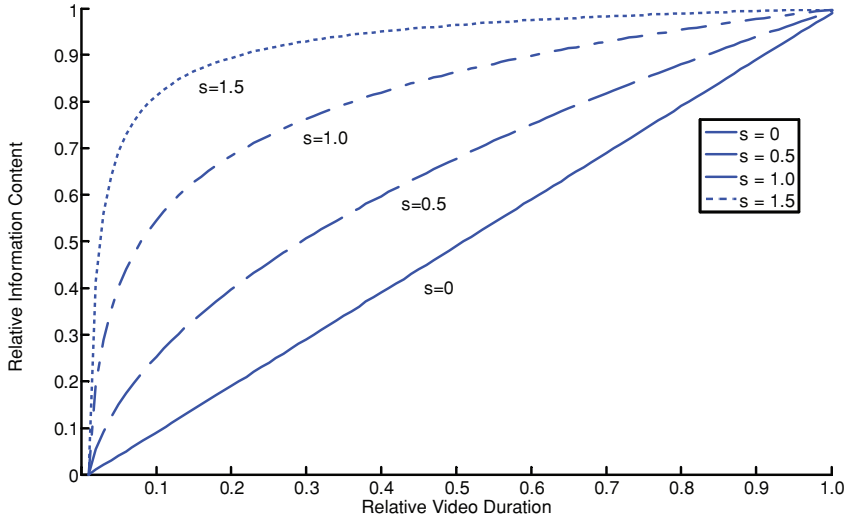


Fig. 6. Relative information content of a transcoded video segment versus normalized video segment duration: *Zipf* function.

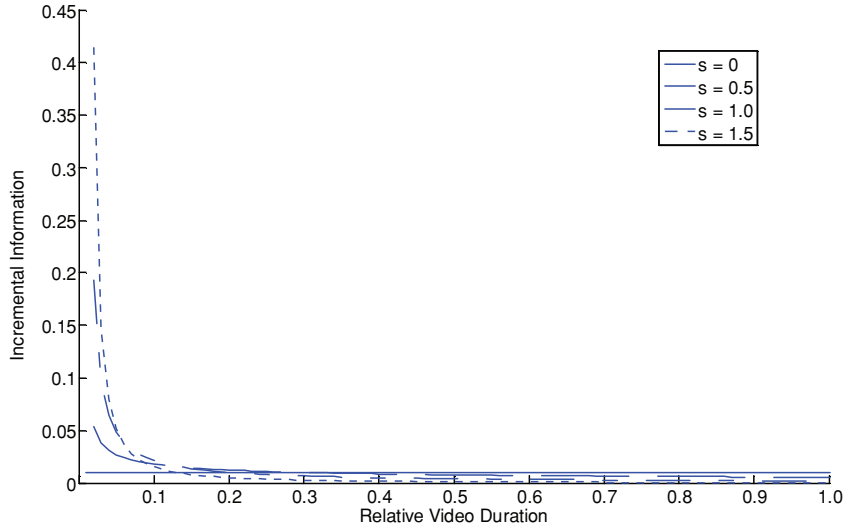


Fig. 7. Incremental information content versus normalized video segment duration: *Zipf* function.

4.2 The Sigmoid Mapping Function

The family of *sigmoid* functions [Uykan et al. 2000] is better suited for categories of videos wherein the middle 20%–30% of the video segment accounts for most of the information content. Eqs. (4.5) and (4.6) describe the sigmoid function family and their corresponding derivatives, respectively, where $\alpha > 0$ is the characteristic shape parameter of the sigmoid function and the parameter L is dependent on the duration of the video segment. Analogous to the parameter s in the *Zipf* function, the parameter σ

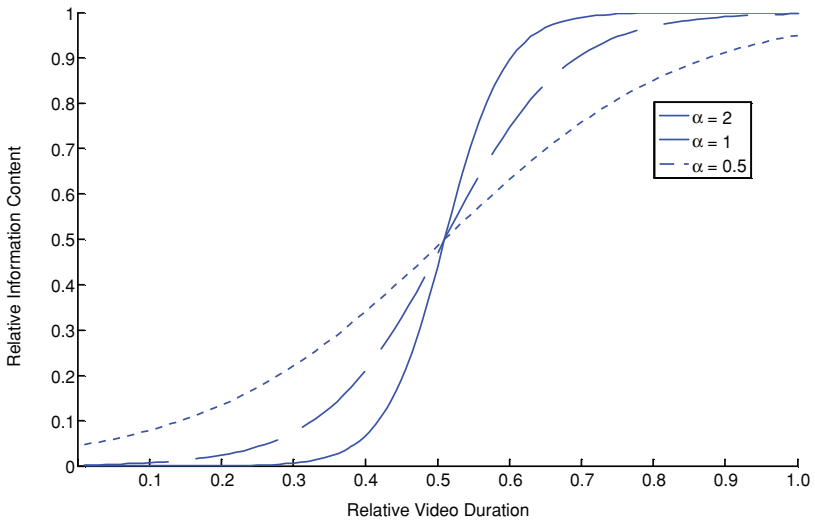


Fig. 8. Relative information content of a transcoded video segment versus normalized video segment duration: *sigmoid* function.

regulates the shape of the *sigmoid* function.

$$I = \frac{1}{1 + e^{-\alpha L}} \tag{4.5}$$

$$I' = \frac{1}{1 + e^{-\alpha L}} \left(1 - \frac{1}{1 + e^{-\alpha L}}\right) \tag{4.6}$$

It suffices in terms of precision to truncate the domains of the functions I and I' such that $L \in [-6, 6]$ in Eqs. (4.5) and (4.6). The normalized video duration $L_{norm} \in [0, 1]$ is mapped to the parameter $L \in [-6, 6]$ in Eqs. (4.5) and (4.6) as follows.

$$L = (12 \times L_{norm}) - 6 \tag{4.7}$$

Figures 8 and 9 respectively show the plots of the *sigmoid* function (Eq. (4.5)) and the corresponding derivatives (Eq. (4.6)) for different values of α. In Figures 8 and 9, the x-axis denotes normalized duration L_{norm} which is related to the value of L in Eqs. (4.5) and (4.6), respectively, via Eq. (4.7).

4.3 The Cumulative Rayleigh Distribution-Based Mapping Function

Unlike the *sigmoid* family of functions, the incremental information governed by the *Rayleigh* distribution [Papoulis 1984] is skewed to the left, thus making the *Rayleigh* distribution suitable for videos in which most of the information is revealed in the earlier portions of the video segment, but not necessarily at the beginning. Eqs. (4.8) and (4.9) define the cumulative *Rayleigh* distribution and its derivative (i.e., the standard *Rayleigh* distribution) respectively where L is video segment duration and $\sigma > 0$ is the characteristic parameter of the *Rayleigh* distribution.

$$I = 1 - e^{(-L^2/2\sigma^2)} \tag{4.8}$$

$$I' = \frac{L \cdot e^{(-L^2/2\sigma^2)}}{\sigma^2} \tag{4.9}$$

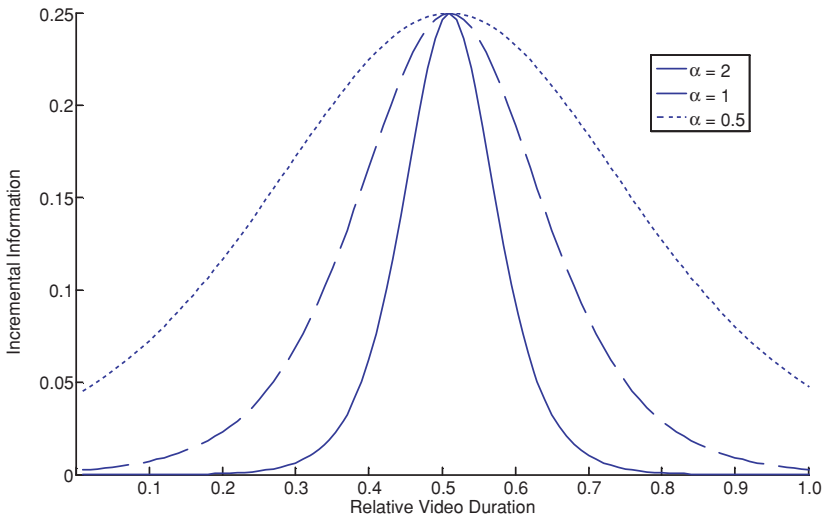


Fig. 9. Incremental information content versus normalized video segment duration: *sigmoid* function.

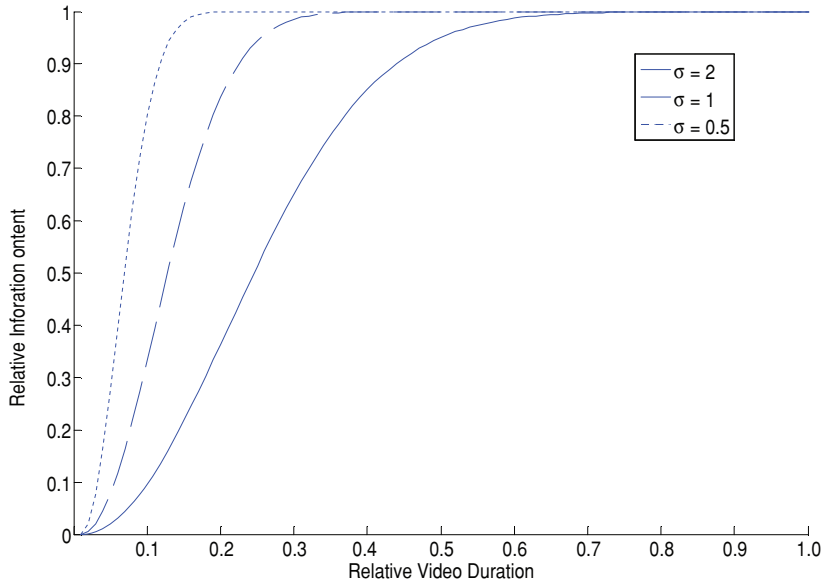


Fig. 10. Relative content of a transcoded video segment vs. normalized video segment duration: cumulative *Rayleigh* distribution.

Figures 10 and 11 plot I (Eq. (4.8)) and I' (Eq. (4.9)), respectively, as functions of the normalized video segment duration L_{norm}. It suffices, in terms of precision, to truncate the domains of functions I and I' such that $L \in [0, 10]$ in Eqs. (4.8) and (4.9). The following linear transform is used to map the normalized video segment duration $L_{norm} \in [0, 1]$ to the parameter $L \in [0, 10]$ in Eqs. (4.8) and (4.9).

$$L = L_{norm} \times 10 \tag{4.10}$$

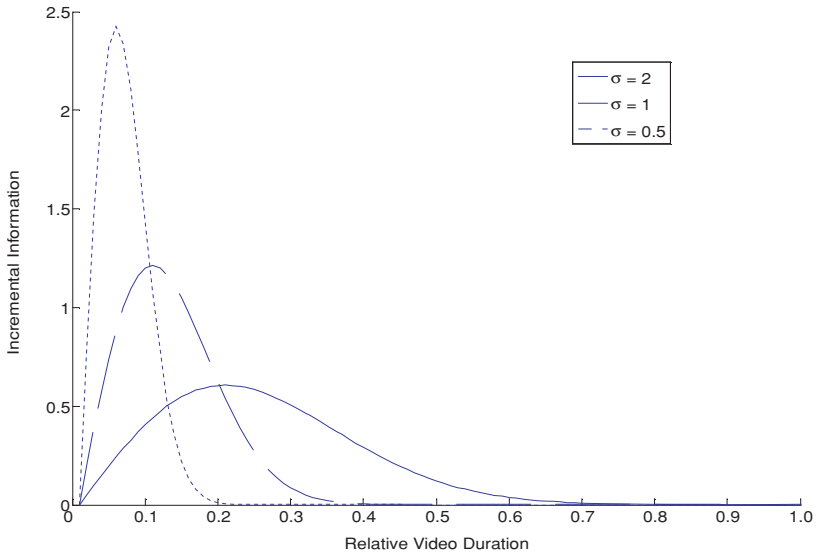


Fig. 11. Incremental information content versus normalized video segment duration: *Rayleigh* distribution.

5. VIDEO PERSONALIZATION: A KNAPSACK PROBLEM APPROACH

The objective of video personalization is to present a customized or personalized video summary that retains as much of the semantic content desired by the client as possible while simultaneously satisfying the multiple resource constraints imposed by the client. This article presents the design and implementation of an MMKP-based video personalization strategy to attain the aforementioned objective. Compared to the 0/1KP-based and the FKP-based video personalization strategies presented in Merialdo et al. [1999], and Tseng et al. [2004, 2003], the proposed MMKP-based video personalization strategy is shown to include more relevant information in its response to the client's request. The MMKP-based personalization strategy is also shown to satisfy *multiple* client-side resource constraints, in contrast to the 0/1KP-based and the FKP-based personalization strategies which can only satisfy a *single* client-side resource constraint at a time.

5.1 Optimization Strategies

The input videos are first segmented and indexed using semantic terms selected from a video description ontology as described earlier. Each video segment is assigned a relevance value based on the client's preference with regard to the video content. Let the set $S = \{S_1, S_2, \ldots S_n\}$ denote the video segments that are stored in the video database, where S_i denotes the ith video segment and n is the total number of candidate video segments to be included in the response to the client's request. Video segment S_i is indexed by a semantic term T_i selected from the video description ontology. In its request for video content, the client specifies a preference for video content using a semantic term P. A relevance value V_i is assigned to the video segment and is given by $V_i = similarity(T_i, P), 0 \leq V_i \leq 1$. The *similarity* function is computed using the *lch* similarity measure algorithm [Leacock and Chodorow, 1998] which measures the length of the shortest path between two semantic concepts, and scales the value by the maximum *is-a* path length in the WordNet lexical database [Fellbaum 1998].

The 0/1KP-based formulation of the video personalization problem [Merialdo et al. 1999] is given by

$$\max_{i\in\{1,2,\ldots,n\}}\left(\sum_i V_i\right),\ \text{subject to}\ \sum_i L_i \le T, \tag{5.1}$$

where L_i is the duration of video segment i and T is the client video viewing time limit. The 0/1KP can be solved using a dynamic programming algorithm that has a time complexity of $O(nT)$. Video segments included in the server's response to the client's request are of the original (i.e., nontranscoded) quality. However, some of the video segments which are excluded from the server's response may still contain some information of potential interest or relevance to the client.

In the FKP-based formulation of video personalization, a fractional portion of a video segment could be included in the set of video segments compiled by the personalization module. The video segment thus included is suitably transcoded to enable it to fit within the limits of the available viewing time. The FKP-based formulation of video personalization is given by

$$\max_{i\in\{1,2,\ldots,n\}}\left(\sum_i x_i V_i\right), \text{subject to}\ \sum_i y_i L_i \le T, \tag{5.2}$$

where T is the client video viewing time limitation, L_i is the temporal length of video segment S_i, and x_i, $y_i \in [0, 1]$. Video segments are sorted in decreasing order of their *Value_Intensity* as computed in Eq. (5.3), where V_i is the relevance value and L_i is the time duration of video segment S_i.

$$Value\_Intensity = V_i/L_i \tag{5.3}$$

The preceding FKP can be solved by using an $O(n)$ greedy algorithm where video segments with high *Value_Intensity* values are selected first followed by fractional portions of video segments. Although the FKP-based optimization scheme can include transcoded video segments, some potentially relevant videos could be excluded from the server's response. This can be attributed to the basic nature of the constrained optimization problem posed by the FKP and the greedy algorithm used to solve it.

Multimedia content can be represented at different semantic levels of abstraction. In this work, the original video segment is considered associated with a discrete set consisting of its various transcoded versions. Each transcoded version is deemed to represent the semantic information content of the original video segment at a certain predefined level of abstraction. Furthermore, client-side resource constraints are typically *multidimensional*. Thus, the amount of relevant information included in the personalized video in response to a client's request needs to be maximized subject to *multiple* resource constraints. This version of the video personalization problem is modeled along the MMKP [Khan 1998; Akbar et al. 2001; Hernandez et al. 2005] as follows.

Each video segment S_i is transcoded into l_i versions, denoted as $S_{ij}, j \in \{1, 2, \ldots, l_i\}$. The original video segment and its transcoded versions constitute a *content group* at multiple levels of abstraction where each *item* within a *content group* is a video segment. Each transcoded version is associated with a relevance value and is deemed to require m resources. The objective of the MMKP-based video personalization strategy is to select exactly one *item* from each *content group* in order to maximize total relevance value of the selected segments, subject to m resource constraints determined by the client. Let v_{ij} be the relevance value of the jth version of the video segment $S_i, \vec{r}_{ij} = (r_{ij1}, r_{ij2}, \ldots, r_{ijm})$ be the required resource vector for the jth version of the video segment S_i,and $\vec{R} = (R_1, R_2, \ldots, R_m)$ be the resource bound of the knapsack representing the m resources. The problem therefore is to determine

$$V = \max\left(\sum_{i=1}^{n}\sum_{j=1}^{l_i} x_{ij}v_{ij}\right), \text{subject to}\sum_{i=1}^{n}\sum_{j=1}^{l_i} x_{ij}r_{ijk} \le R_k, k = 1, 2, \ldots, m \text{ and}\sum_{j=1}^{l_i} x_{ij} = 1, x_{ij} \in \{0, 1\}. \tag{5.4}$$

Based on the previous formulation of the MMKP-based video personalization strategy, in Eq. (5.4) it is obvious that more *items* can be added to a *content group* without requiring any other changes to the video personalization system. The modularity and extensibility of the MMKP-based video personalization strategy is one of its salient features.

The MMKP is known to be NP-hard [Hernandez et al. 2005]. The exact solution to the MMKP can be obtained using a Brand-and-Bound Integer Programming (BBIP) algorithm in Matlab [Vanderbei 1997]. The worst-case time complexity of the BBIP algorithm is exponential in n, m, and l_i, where n is the number of content groups, m is the number of resource constraints, and l_i is the number of items in the ith content group of the MMKP [Hernandez et al. 2005; Khan 1998].

5.2 Performance Metrics

In order to measure and compare the performance of the various video personalization strategies, the sum of relevance values of all video segments included in the response is used as the performance metric, namely $\sum_{i \in response} v_i$. For a transcoded video segment, its relevance value v_i is defined in Eq. (4.1).

6. EXPERIMENTAL RESULTS OF THE MULTI-LEVEL HMM-BASED VIDEO SEGMENTATION AND INDEXING ALGORITHM

We recorded 2 hours of the CNN Headline News program and 1.5 hours of the Major League Soccer (MLS) program, respectively. The video streams were digitized to a frame resolution of 180 $\times$ 120 pixels with a frame rate of 30 frames per second. Sixteen minutes of the CNN video and one hour of the soccer video were reserved for testing. For generation of training data, the remainder of the CNN news video data was manually segmented into six semantic categories: *News Anchor*, *News*, *Commercial*, *Program Header*, *Weather Forecast*, and *Sports News* and denoted by semantic concepts 1 through 6, respectively, and the MLS video data was manually segmented into three semantic categories, *Zoom Out*, *Close Up*, and *Replay* and denoted by semantic concepts 1 through 3, respectively. A multidimensional feature vector was extracted for each image frame in the training video. For each of the semantic units, a left-to-right HMM with continuous emission of observations was trained using feature vector sequences derived from the training video. To estimate the 2-gram video program model, the training video was manually labeled with labels selected from the aforementioned semantic concepts. The maximum likelihood estimation of the video program model was performed using the labeled training sequence.

The performance of the single-pass video segmentation and video indexing scheme was evaluated in terms of accuracy of video segment boundary detection and video segment classification. The performance of the video segment boundary detection algorithm was measured in terms of parameters such as insertion rate, deletion rate, and boundary detection accuracy [Eickeler et al. 2000]. The insertion rate $R_{insertion}$ denotes the fraction of unassigned boundaries in the detected boundaries. The deletion rate $R_{deletion}$ denotes the fraction of missed boundaries in the ground-truth sequence boundaries. The boundary detection accuracy $Accuracy_B$ measures the average shift (in terms of number of frames) between the detected boundary and actual boundary locations. Thus

$$R_{insertion} = \frac{boundaries_{inserted}}{boundaries_{detected}}, \tag{6.1}$$

$$R_{deletion} = \frac{boundaries_{missed}}{boundaries_{actual}}, \tag{6.2}$$

$$Accuracy_B = \frac{\sum \Delta frames}{boundaries_{actual}}, \tag{6.3}$$

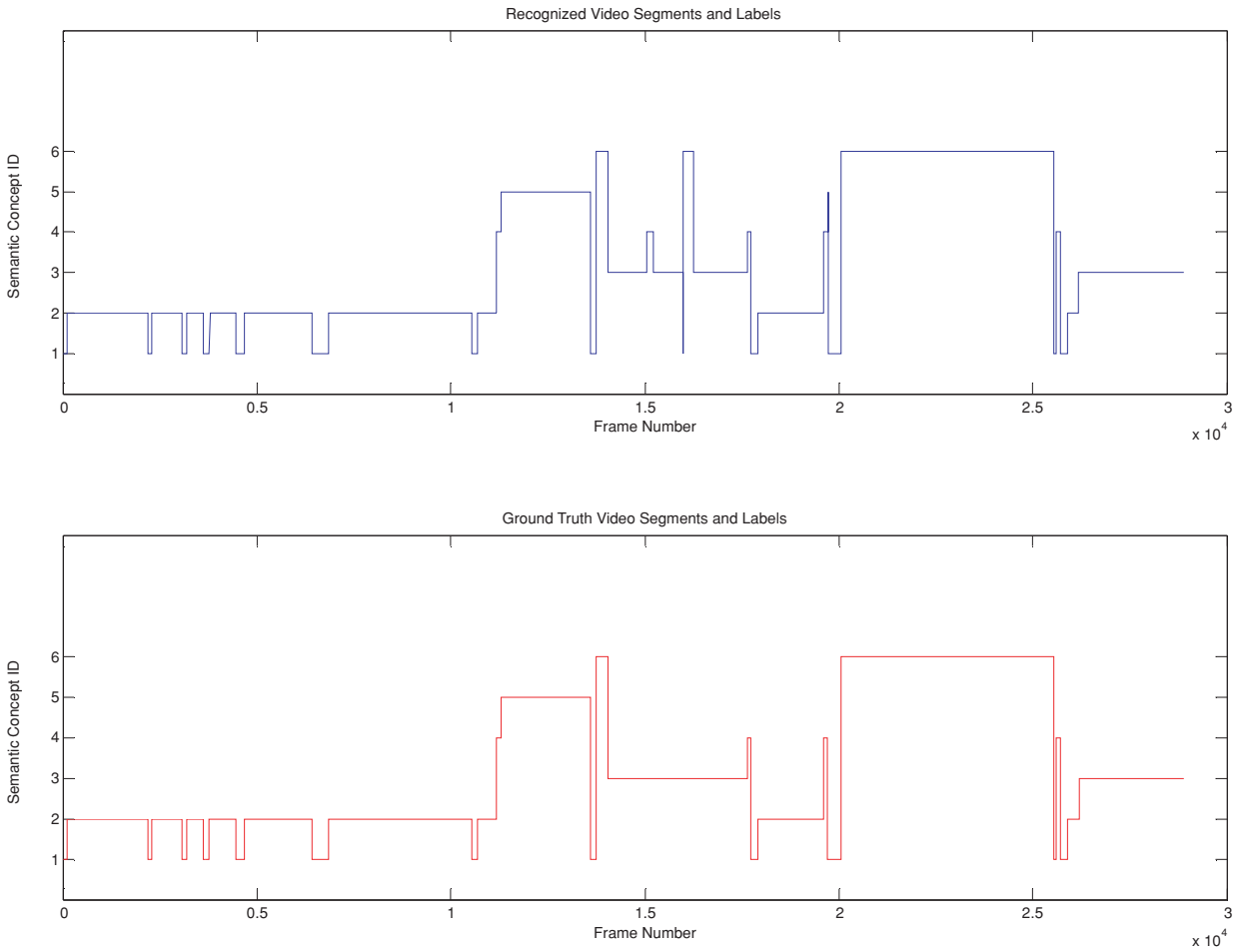


Fig. 12. Frame number vs. recognized/ground-truth video segments and labels: CNN News video.

where $\sum \Delta frames$ is the amount of shift (measured in terms of number of frames) between the detected boundary location and the actual boundary location.

To measure the video segment classification accuracy, we used a frame-based measure to determine the fraction of correctly classified image frames in the total number of frames as follows:

$$Accuracy_F = \frac{frames_{correct}}{frames_{correct} + frames_{false}}, \tag{6.4}$$

where $frames_{correct}$ and $frames_{false}$ are the numbers of correctly classified and incorrectly classified image frames, respectively. $Accuracy_F$ thus measures the temporal classification accuracy, namely the relative duration of correctly recognized segments of a video stream. The algorithm provided by Eickeler et al. [2000] was used to compute performance measures in Eqs. (6.1)–(6.4).

In Figure 12, the recognized semantic label sequence and the ground-truth semantic label sequence of the CNN Headline News video stream are plotted against the frame number for the entire test video segment. For the news video, the single-pass video segmentation and video indexing algorithm was

Table I. Performance Measures for Video Segment Boundary Detection: CNN Headline News Video

Actual Boundaries	Detected Boundaries	Inserted Boundaries	Deleted Boundaries	Insertion Rate (%)	Deletion Rate (%)	$Accuracy_B$ (Frame)
29	36	7	0	7/36 = 19.4	0/29 = 0	55/29 = 1.9

Table II. Performance Measures for Video Segment Classification: CNN News Video

Correctly Classified Segments	Incorrectly Classified Segments	$Accuracy_C$ (%)	Total Number of Frames	Correctly Classified Frames	$Accuracy_F$ (%)
32	5	86.5	28898	514	28384/28898 = 98.2

observed to detect most of the segment boundaries and label them correctly, except for some portion of the *Commercial* segment which was incorrectly classified. In Tables I and II, the numerical measures of performance for the single-pass video segmentation and indexing algorithm are tabulated. Figure 12 shows that most of the inserted boundary detection and false segment classification occurs during the *Commercial* segment (semantic concept ID=3). This is because of the complex nature of the content of TV commercials which could contain large video segments similar to those found in the other semantic units such as *News Anchor* and *Sports*, thus resulting in incorrect classification.

In Figure 13, the recognized semantic label sequence and the ground-truth semantic label sequence are plotted against the frame number for the MLS test video segment. In Tables III and IV, the corresponding numerical measures of performance for the single-pass video segmentation and video indexing algorithm are tabulated in the case of the MLS test video segment. Experimental results show that the proposed multi-level HMM-based video segmentation and video indexing algorithm can segment and index MLS videos quite accurately.

The proposed multi-level HMM-based video segmentation and video indexing algorithm was implemented on a Dell Precision workstation with dual 3.19GHz CPUs and 2.0GB of RAM. In the case of the CNN Headline News video, it took 3076 seconds to extract feature vectors from the training data, 1394 seconds to train the HMMs for the individual video semantic units, and 1400 seconds to segment and index the CNN Headline News test video of duration of 962 seconds as presented in Figure 12. In the case of the MLS video, it took 1078 seconds to extract feature vectors from the training data, 873 seconds to train the HMMs for video semantic units, and 930 seconds to segment and index the MLS test video of duration of 746 seconds as presented in Figure 13. The video files and their transcoded versions were stored at a bit rate of 210kbps.

7. EXPERIMENTAL RESULTS OF VIDEO PERSONALIZATION STRATEGIES

The CNN Headline News video was first segmented and indexed using the stochastic multi-level HMM-based algorithm. Each video segment was labeled with terms selected from a predefined video content description ontology. Video segments were transcoded at multiple levels of abstraction and then stored in a hierarchical video database. Key-frame-based transcoding was performed such that the original video and its transcoded versions have the same spatial resolution, although their time durations are different. The sum of relevance values of the video segments included in the server's response was used to measure the performance of the various video personalization strategies.

In Figures 14(a) and 14(b), the *Zipf* function and linear transform defined in Eqs. (4.2) and (4.3), respectively, were used to measure the relative information content and the relative time duration of the transcoded video segments. The total relevance value of the server's response to the client's request was plotted against the client's viewing time limit. In Figure 14(a), the characteristic parameter s of the *Zipf* function was set to 0 whereas in Figure 14(b), s was set to 1. The MMKP-based personalization

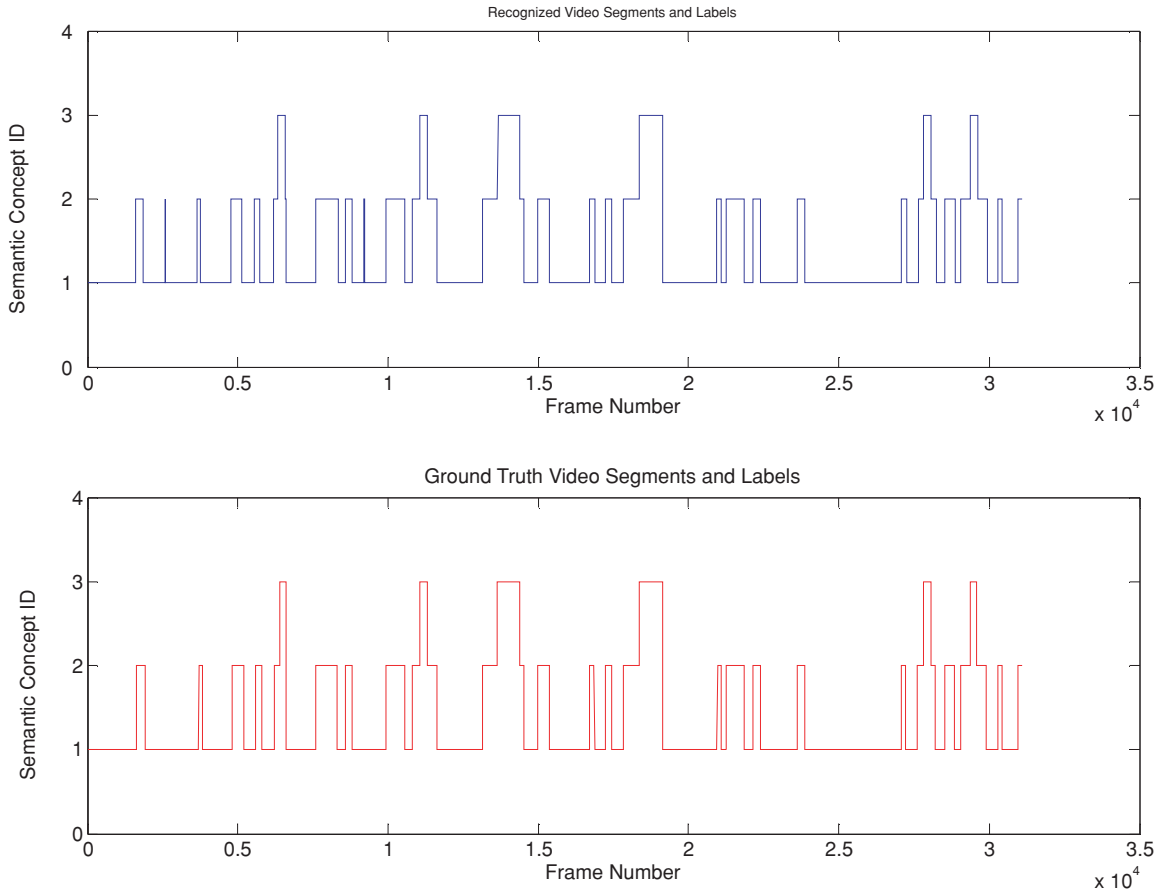


Fig. 13. Frame number vs. recognized/ground-truth video segments and labels: MLS video.

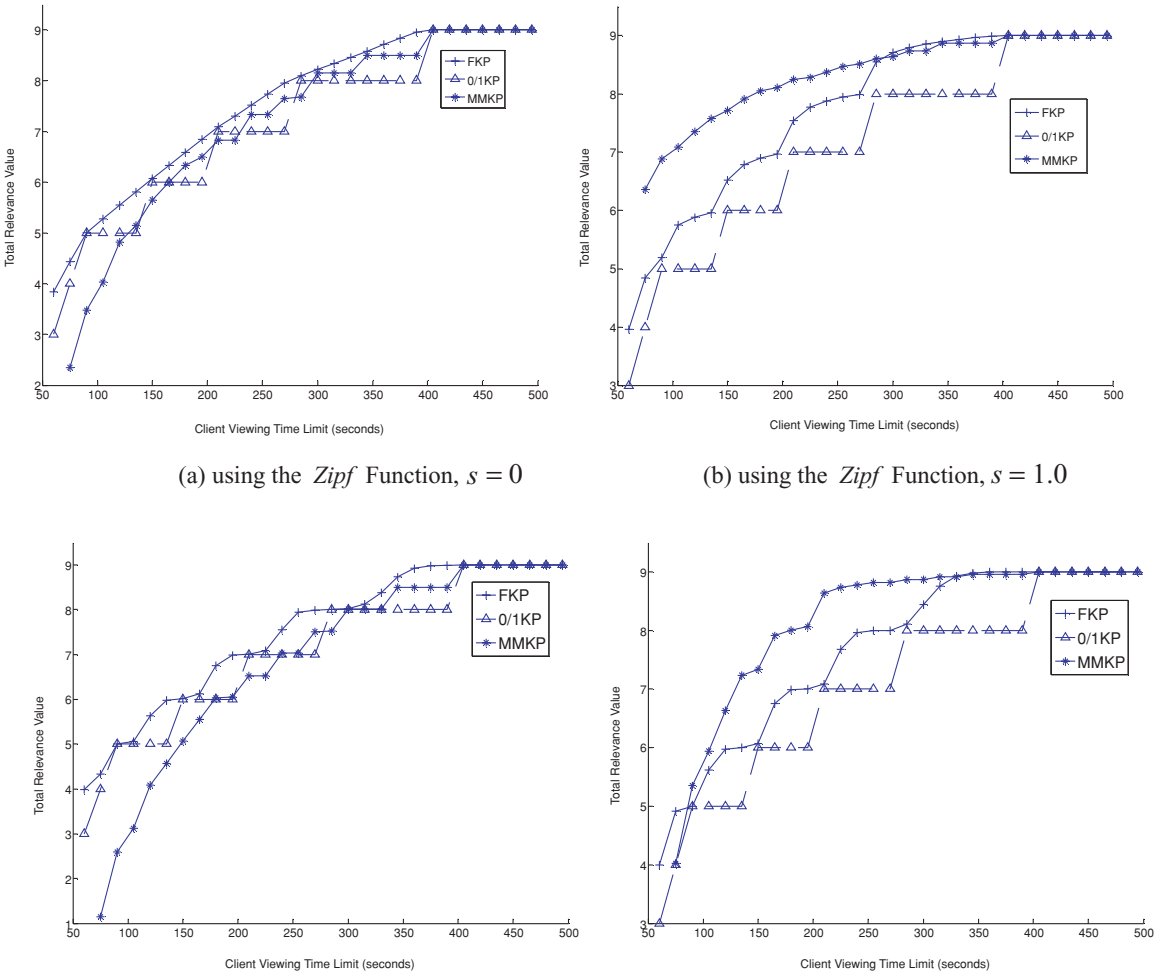
Table III. Performance Measures for Video Segmentation Boundary Detection: MLS Video

Actual Boundaries	Detected Boundaries	Inserted Boundaries	Deleted Boundaries	Insertion Rate (%)	Deletion Rate (%)	$Accuracy_B$ (Frame)
57	60	3	0	3/60 = 5	0/60 = 0	671/57 = 11.8

Table IV. Performance Measures for Video Segment Classification: MLS Video

Correctly Classified Segments	Incorrectly Classified Segments	$Accuracy_C$ (%)	Total Number of Frames	Correctly Classified Frames	$Accuracy_F$ (%)
59	2	96.7	31150	783	30367/31150 = 97.5

strategy was designed to select one *item* (i.e., video segment) from each *content group*. In the case of the MMKP-based personalization strategy, more transcoded video segments were observed to be included in the server's response compared to the FKP-based and 0/1KP-based personalization strategies. When the number of candidate *content groups* was large, the time durations of the included transcoded video segments in the case of the MMKP-based personalization strategy were observed to be short. When $s = 0$ in the *Zipf* function (Figure 6), the total information content of the short video segments included



(a) using the *Zipf* Function, $s = 0$

(b) using the *Zipf* Function, $s = 1.0$

(c) using the *sigmoid* function, $\alpha = 1.0$

(d) using the *Rayleigh* distribution, $\sigma = 2.0$

Fig. 14. Total relevance value of the response versus the client's viewing time limit.

by the MMKP-based personalization strategy in its response to the client's request was observed to be generally less than that of the video segments included in the response generated by the FKP-based personalization strategy as shown in Figure 14(a). However, if we assume that the beginning portion of a video segment contains a major fraction of its overall information content, for example, when $s = 1.0$ in the *Zipf* function (Figure 6), then the short video segments selected by the MMKP-based personalization strategy were observed to contain more relevant information than those contained in the responses generated by the FKP-based and 0/1KP-based personalization strategies, as shown in Figure 14(b). With the *Zipf* function, the 0/1KP-based personalization strategy was observed to underperform both its MMKP-based and FKP-based counterparts.

The proposed system was also designed to provide a testbed for various video personalization strategies. In Figures 14(c) and 14(d), we used the *sigmoid* function and the cumulative *Rayleigh* distribution defined in Eqs. (4.5) and (4.8) respectively to measure the relative information content of the transcoded

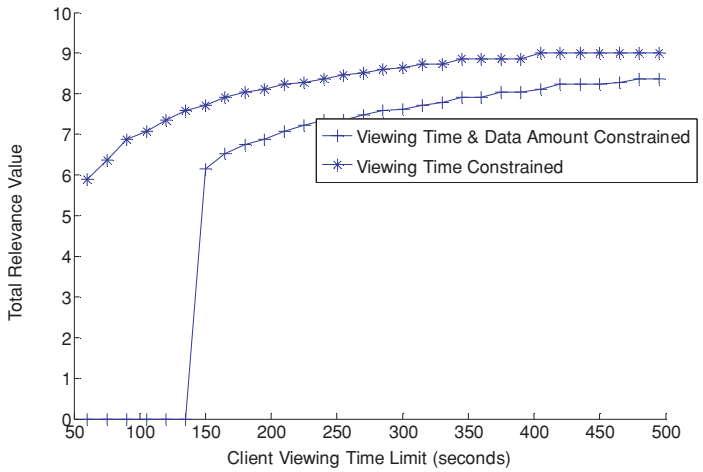


Fig. 15. Performance of the MMKP-based personalization scheme under the viewing time limit constraint and under both the viewing time limit constraint and the data limit constraint ($\leq$ 3Kbytes for each second of received video).

videos in comparison to their original versions. In the case of the *sigmoid* function with $\alpha = 1$, the short transcoded video segments selected by the MMKP-based personalization strategy were observed to contain less information than those selected by the FKP-based and 0/1KP-based personalization strategies as shown in Figure 14(c). In the case of the cumulative *Rayleigh* distribution (Figure 10), more information was assumed to be contained in the beginning portions of a video (Figure 11). It was observed that, when the client's viewing time limit was very short, only very short transcoded videos were included in the server's response, causing the FKP-based personalization strategy to yield a response with higher total relevance value than its MMKP-based counterpart. However, with an increase in the client's viewing time limit, relatively longer transcoded video segments were selected by the MMKP-based personalization strategy, causing it to generate a response with a higher overall relevance value. In summary, it was observed that when the information content of the underlying video segments is concentrated in its earlier portions, the MMKP-based personalization strategy typically outperforms its FKP-based and 0/1KP-based counterparts.

A principal advantage of the proposed MMKP-based video personalization strategy is that it can satisfy multiple client-side resource constraints simultaneously whereas its FKP-based and 0/1KP-based counterparts can only satisfy a single client-side resource constraint at a time. In Figures 14(a)–14(d), the client-side resource constraint under consideration is the client's viewing time limit. Figure 15 shows the experimental results of the MMKP-based personalization strategy when the client has two resource constraints, that is, a viewing time limit and a limit on the total amount of data received. The *Zipf* function with $s = 0$ was used in this case. The received data was limited to at most 3 KBytes for each second of the received video stream. The data limit constraint was held constant (at 3KBytes for each second of the received video stream) whereas the viewing time limit constraint was varied. As seen in Figure 16, when the viewing time was less than 150 seconds, the response to the client's request contained no video segment, resulting in a null response. This was so because in each of the *content groups*, there was no *video item* of size less than 450Kbytes (= 3Kbytes per second $\times$ 150 seconds). When the client's viewing time limit was large enough (greater than 150 seconds in our experiment), it was possible to include video segments or video summaries which satisfied, the data limit constraint in the response to the client's request. It was clear that when both constraints needed to be satisfied the response contained less video information compared to the case wherein the client viewing time was

the only constraint. Thus, the solutions obtained when only a single resource constraint was satisfied at a time and when both resource constraints were simultaneously satisfied were substantially different. The FKP-based and 0/1KP-based personalization strategies cannot handle multiple constraints simultaneously and were forced to satisfy individual constraints one at a time. Since different resource constraints, when employed individually, yield different solutions, determining the optimal combination of these solutions to satisfy multiple resource constraints simultaneously becomes an important (and difficult) issue in the case of the FKP-based and 0/1KP-based personalization strategies. This issue was obviously moot in the case of the MMKP-based personalization strategy since it is inherently equipped to satisfy multiple client-side resource constraints.

The proposed 0/1KP, FKP, and MMKP-based video personalization algorithms were implemented on a Dell Precision workstation with dual 3.19GHz CPUs and 2.0GB of RAM. In the video database, there were 172 video content groups containing three items each. When the client specified viewing time was 90 seconds, it took 31 milliseconds, 16 milliseconds, and 1234 milliseconds for the video personalization algorithm modeled along the 0/1KP, FKP, and MMKP, respectively.

8. CONCLUSIONS

The article proposed a system for client-centered multimedia adaptation with three salient features: a stochastic multi-level HMM-based approach to automatically segment and index video streams, a hierarchical video transcoding and content representation scheme, and a suite of Knapsack Problem (KP)-based video personalization strategies.

Hidden Markov Models (HMMs) were used to model the input video streams at both the semantic unit level and the video program level. For each semantic unit within the input video stream, an HMM was formulated to model the emission of image feature vectors within the scope of that semantic unit as a stochastic process. The 2-gram video program model was used to define the transition probabilities amongst the various semantic units. A data-driven procedure for maximum likelihood estimation of the 2-gram program model from training data was shown to preclude the need for domain-dependent knowledge of the video program model. The individual HMMs for the semantic units were concatenated based on the video program model. Determining the optimal path through the concatenation of the HMMs was shown to result in a data-driven single-pass video segmentation and video indexing algorithm. Experimental results showed that the resulting video boundary detection and video segment classification were highly accurate. The proposed multi-level HMM-based scheme was observed to be scalable and extensible, since the program model could be altered by addition, deletion, and modification (via retraining) of the HMMs corresponding to the relevant semantic units without having to retrain or alter the HMMs corresponding to the other semantic units.

The proposed hierarchical scheme for video transcoding and video content representation at multiple levels of abstraction was used to transform an input video stream into its various transcoded versions. The transcoded videos were observed to differ in terms of the client-side resources required for reception, rendering, and viewing on the client device. The transcoded videos made it possible for the video personalization subsystem to generate an optimal response to a client's query while satisfying various client-side resource constraints, such as battery capacity, bandwidth, and viewing time. Three empirical mapping functions, namely, the *Zipf* function, *sigmoid* function, and cumulative *Rayleigh* distribution, were used to compare the information content of the original video relative to its transcoded versions. These mapping functions shared a common characteristic in conformity with the commonly observed law of diminishing marginal return; that is, the incremental or marginal gain in the information content of a video segment was a diminishing function of its total duration.

The video personalization problem was formulated as one of constrained optimization and modeled along various versions of the classical knapsack problem with the objective of generating an optimal

response, that is, one which maximizes the relevance of the provided information to the client's request, while simultaneously satisfying the client's resource constraints. A Multiple-choice Multidimensional Knapsack Problem (MMKP)-based video personalization strategy was proposed as a means to include as much relevant information as possible in response to a client's request, while satisfying multiple client-side resource constraints. Experimental results showed that when the beginning portions of a video segment contained more information than the rest of the video, the proposed MMKP-based approach yielded a response with higher total relevance value compared to the existing Fractional Knapsack Problem (FKP)-based and 0/1 Knapsack Problem (0/1KP)-based approaches to video personalization.

Future work would include studies to validate the choice of the previously mentioned empirical mapping functions. Human-subject-based evaluation of the generated response to a client's request also needs to be explored in order to further validate this work.

REFERENCES

AKBAR, M. D., MANNING, E. G., SHOJA, G. C., AND KHAN, S. 2001. Heuristic solutions for the multiple-choice multidimension knapsack problem. In *Proceedings of the International Conference on Computational Science*, 659–668.

BARTOLI, A., DALAL, N. AND HORAUD, R. 2004. Motion panoramas. *Comput. Anim. Virtual Worlds*, *15*, 501–517.

BAUM, L. E., PETERIE, T., SOULED, G., AND WEISS, N. 1970. A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *Ann. Math. Statist*, 164–171.

BHANDARKAR, S. M., WARKE, Y. S., KHOMBHADIA, A. A. 1999. Integrated parsing of compressed video. Lecture Notes In Computer Science, vol. 1614, 269–276.

BORECZKY, J. S. AND WILCOX. L. D. 1998. A hidden Markov model framework for video segmentation using audio and image features. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*.

BROWN, P. F., PIETRA, V. J., DESOUZA, P. V., LAI, J. C., AND MERCER, R. L. 1992. Class-based n-gram models of natural language. *Comput. Linguist. 18*, 4, 467–479.

CHEN, M. J., CHU, M. C., AND PAN, C. W. 2002. Efficient motion estimation algorithm for reduced frame-rate video transcoder. *IEEE Trans. Circ. Syst. Video Technol*. *12*, 4, 269–275.

EICKELER, S. AND MÜLLER, S. 1999. Content-based video indexing of TV broadcast news using hidden Markov models. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2997–3000.

EICKELER, S. AND RIGOLL, G. 2000. A novel error measure for the evaluation of video indexing systems. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 4, 1991–1994.

ELEFTHERIADIS, A. AND BATRA, P. 2006. Dynamic rate shaping of compressed digital video. *IEEE Trans. Multimedia 8*, 2, 297–314.

FELLBAUM, C., Ed. 1998. *WordNet—An Electronic Lexical Database*. The MIT Press, Cambridge, MA.

FLICKNER, M., SAWHNEY, H., NIBLACK, W., ASHLEY, J., HUANG, Q., DOM, B., GORKANI, M., HAFNER, J., LEE, D., PETKOVIC, D., STEELE, D., AND YANKER, P. 1995. Query by image and video content: The QBIC system. *IEEE Comput. Mag.* 23–32.

FORNEY, G. D. 1973. The Viterbi algorithm. *Proceedings of the IEEE*, vol. 61, No. 3, 268-278.

HERNANDEZ, R. P. AND NIKITAS, N. J. 2005. A new heuristic for solving the multiple-choice multidimensional knapsack problem. *IEEE Trans. Syst. Man Cybernetics, Part A 35*, 5, 708–717.

HUANG, J., LIU, Z., AND WANG, Y. 2005. Joint scene classification and segmentation based on hidden Markov model. *IEEE Trans. Multimedia* 7, 3, 538–550.

IRANI, M., HSU, S., AND ANANDAN, P. 1995. Mosaic-based video compression. In *Proceedings of the SPIE Conference on Electronic Imaging, Digital Video Compression: Algorithms and Technologies*, vol. 2419, 242–253.

IRANI, M., ANANDAN, P., BERGEN, J., KUMAR, R., AND HSU, S. 1996. Efficient representations of video sequences and their applications. *Signal Process. Image Commun. Special Issue on Image Video Semantics: Processing, Analysis, Appl. 8*, 4, 327–351.

KHAN, S. 1998. Quality adaptation in a multi-session adaptive multimedia system: Model and architecture. Ph.D. thesis, *Department of Electronical and Computer Engineering*, University of Victoria.

LEACOCK, C. AND CHODOROW, M. 1998. Combining local context and wordnet similarity for word sense identification. In *WordNet: An Electronic Lexical Database*, Fellbaum C. (Ed.), MIT Press, Cambridge, MA, 265–283.

LI, B. AND SEZAN, M. I. 2001. Event detection and summarization in sports video. In *Proceedings of the IEEE Workshop on Content-Based Access of Image and Video Libraries* 8, 132–138.

LI, C. S., MOHAN, R., AND SMITH, J. R. 1998. Multimedia content description in the Info-Pyramid. In *Proceedings of the ICASSP'98, Special Session on Signal Processing in Modern Multimedia Standards*, vol.6, 3789–3792.

MERIALDO, B., LEE, K.T., LUPARELLO, D., AND ROUDAIRE, J. 1999. Automatic construction of personalized TV news programs. In *Proceedings of the ACM Conference on Multimedia*, 323–331.

NAKAJIMA, Y., HORI, H., AND KANOH, T. 1995. Rate conversion of MPEG coded video by requantization process. In *Proceedings of the IEEE International Conference on Image Processing*, 408–411.

NEY, H. AND ORTMANNS, S. 1999. Progress on dynamic programming search for continuous speech recognition. *IEEE Signal Proc. Mag. 16*, 5, 64–83.

PAPOULIS, A. 1984. *Probability, Random Variables, and Stochastic Processes*, 2nd Ed. McGraw-Hill, New York, 104, 148.

RABINER, L. R. 1989. A tutorial on hidden Markov models and selected applications in speech recognition. In *Proceedings of the IEEE 77*, 2, 257–286.

SHINODA, K., BACH, N. H., FURUI, S., AND KAWAI, N. 2005. Scene recognition using hidden Markov models for video database. In *Proceedings of the Symposium on Large-Scale Knowledge Resources (LKR'05)*, 107–110.

SNOEK, C. G. M. AND WORRING, M. 2003. Time interval maximum entropy based event indexing in soccer video. In *Proceedings of the IEEE International Conference on Multimedia & Expo*, vol. 3, 481–484.

SUN, H., KWOK, W., AND ZDEPSKI, J. 1996. Architectures for MPEG compressed bitstream scaling. *IEEE Trans. Circ. Syst. Video Technol. 6*, 191–199.

TAMURA, H., MORI, S., AND YAMAWAKI, T. 1978. Textural features corresponding to visual perception. *IEEE Trans. Syst. Man Cybernetics 8*, 460–472.

TSENG, B. L., LIN, C. Y., AND SMITH, J. R. 2004. Using MPEG-7 and MPEG-21 for personalizing video. *IEEE Multimedia*, *11*, 1, 42–52.

TSENG, B. L. AND SMITH, J. R. 2003. Hierarchical video summarization based on context clustering. In *Proceedings of the SPIE*, 5242, 14–25.

TSENG, B. L., LIN, C. Y., AND SMITH, J. R. 2002. Video personalization and summarization system. In *Proceedings of the IEEE Workshop on Multimedia Signal Processing*, 424–427.

UYKAN, Z. AND KOIVO, H. N. 2000. Unsupervised learning of sigmoid perceptron. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 6, 3486–3489.

VANDERBEI, R. J. 1997. *Linear Programming: Foundations and Extensions*. Kluwer Academic, Norwell, MA.

VIOLA, P. AND JONES, M. J. 2004. Robust real-time face detection. *Int. J. Comput. Vision 57*, 2, 137–154.

WEI, Y., WANG, H., BHANDARKAR, S. M., AND LI, K. 2006. Parallel algorithms for motion panorama construction. In *Proceedings of the ICPP Workshop on Parallel and Distributed Multimedia*, 82–92.

WHEELER, E. S. 2002. Zipf's law and why it works everywhere. *Glottometrics*, *4*, 45–48.

ZHU, W., YANG, K., AND BEACKEN, M. 1998. CIF-to-QCIF video bitstream down conversion in the DCT domain. *Bell Labs Tech. J. 3*, 3, 21–29.

Received November 2006; revised May 2007; accepted August 2007