Video Personalization in Resource-Constrained Multimedia Environments

Yong Wei
Department of Computer Science
The University of Georgia
Athens, GA 30602-7404, USA
yong@cs.uga.edu

Suchendra M. Bhandarkar
Department of Computer Science
The University of Georgia
Athens, GA 30602-7404, USA
suchi@cs.uga.edu

Kang Li
Department of Computer Science
The University of Georgia
Athens, GA 30602-7404, USA
kangli@cs.uga.edu

ABSTRACT

Multimedia data, especially video data, is being increasingly transmitted to, transmitted from and viewed on mobile devices such as PDA's, laptop PCs, pocket PCs and cell phones. One of the natural limitations of these multimedia-capable, mobile devices is that they are constrained by their battery power capacity, viewing time limit, amount of data received, and in many situations, by available network bandwidth connecting these devices with video servers. The video server is typically also constrained by its computing power and connection bandwidth. In order to provide a resource-constrained mobile client with its desired video content, it is necessary to adapt or personalize the video content while simultaneously satisfying the aforementioned constraints. Also, in order to limit the client-experienced latency, it is necessary to perform client request aggregation on the server end. To this end, a video personalization strategy is proposed to provide mobile, resource-constrained clients with personalized video content that is most relevant to the client's request while simultaneously satisfying multiple client-side system-level resource constraints. A client request aggregation strategy is also proposed to cluster client requests with similar video content preferences and similar client-side resource constraints such that the number of requests the server needs to process and the client-experienced latency are both reduced.

The primary contributions of the paper are (1) the formulation and implementation of a Multiple-choice Multi-dimensional Knapsack Problem (MMKP)-based video personalization strategy; and (2) the design and implementation of a multi-stage clustering-based client request aggregation strategy. Experimental results comparing the proposed MMKP-based video personalization strategy to existing 0/1 Knapsack Problem (0/1KP)-based and the Fractional Knapsack Problem (FKP)-based video personalization strategies are presented. It is observed that (1) the proposed MMKP-based personalization strategy includes more relevant video content in response to the client's request compared to the existing 0/1KP-based and FKP-based personalization strategies; and (2) in contrast to the 0/1KP-based and FKP-based personalization strategies which can satisfy only a single client-side constraint at a time, the proposed MMKP-based personalization strategy is shown to be capable of satisfying simultaneously multiple client-side resource constraints. Experimental results comparing the client-experienced latency with and without the proposed client request aggregation strategy are also presented. It is shown that the proposed client request aggregation strategy significantly reduces the mean client-experienced latency *without* significant reduction in the average relevance value of the video content delivered in response to the client's request.

Categories & Subject Descriptors: H.5.1 Multimedia Information Systems, Video (e.g., tape, disk, DVI)

General Terms: Algorithms

Keywords

Video personalization, Video summarization, Multiple-choice Multi-dimensional Knapsack Problem, Request aggregation.

1. INTRODUCTION

The current proliferation of mobile computing devices and networking technologies has created enormous opportunities for mobile device users to communicate with multimedia servers. 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 and display of multimedia data, the user demand for being able to view streaming video on such devices has increased several-fold. For example, a mobile handheld client may be interested in viewing a video showing traffic conditions on the road and browsing the weather forecast for his/her travel destination. One of the natural limitations of typical handheld mobile devices is that they are resource constrained, i.e., constrained by their battery power capacity, viewing time limit and in many situations, by the available network bandwidth connecting them with video servers. Thus, the original video content often needs to be personalized in order to fulfill the client's request while satisfying simultaneously various client-side system-level resource constraints. Also, in order to limit the client-experienced latency, it is often necessary to perform client request aggregation on the server end when dealing with multiple client requests.

In light of the above, a definition of video personalization can be given as follows:

Definition 1: Given the client's preferences regarding the video content, and given the client-side resource constraints, video personalization is the process of compiling and disseminating the most relevant video content to the mobile client while satisfying simultaneously the client-side resource constraints.

MM'07, September 23–28, 2007, Augsburg, Bavaria, Germany.

A client 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 the client and the server. 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.

In this paper, we present a client-centered video personalization system which can optimally fulfill the client's requests while simultaneously ensuring optimal utilization of the client-side system-level resources. While there are many challenges to be addressed in the design and implementation of a comprehensive video personalization system, the work presented in this paper focuses on the design and implementation of video personalization strategies. In addition, a client request aggregation strategy is proposed and implemented in order to cluster multiple client requests with similar video content preferences and similar client-side resource constraints. The goal of the proposed client request aggregation strategy is to simultaneously reduce both, the number of client requests the server needs to process and the client-experienced latency.

The video personalization problem is modeled as a constrained optimization problem, i.e., maximization of the *"total relevance value"* of the video summary delivered to the client under multiple constraints that represent the client's content preferences and the available system-level resources. Various personalization strategies based on the classical Knapsack Problem (KP) have been proposed in the literature. However, existing video personalization strategies do not consider a multiple-client scenario. In order to serve multiple clients with acceptable client-experienced latency while ensuring efficient utilization of server resources, it is shown to be necessary for the server to aggregate multiple client requests prior to video content delivery.

In the proposed video personalization scheme, raw videos are automatically segmented and indexed/labeled in a single pass using a stochastic modeling approach [1] and summarized offline at multiple levels of abstraction. Content-aware key frame selection algorithms and dynamic motion panoramas are used to generate video summaries. Videos are labeled using semantic terms selected from an ontology such as WordNet [16]. A client request to the video server consists of the client's video content preferences and a set of client-side resource constraints. The client-side resource constraints can include video viewing time, battery power capacity, transmission bandwidth, amount of received data and expected quality of the received video. The video personalization module matches the client's video content preferences with the indices of the video summaries in the database, and selects from the retrieved video summaries, a subset of video segments or summaries at the appropriate levels of abstraction that best matches the client content preferences while satisfying simultaneously the various client-side resource constraints.

In a typical video personalization system, requests are received from multiple clients. Servicing each request on an individual basis, entails a high degree of consumption of server-side and network resources, such as computing time and network bandwidth and also entails high client-experienced latency on average. The proposed client request aggregation strategy clusters similar client requests together such that the number of effective requests to be processed by the server is reduced. This, in turn, reduces both the server and network load and the average client-experienced latency. Since the client requests are heterogeneous along multiple dimensions, i.e. they differ in terms of their video content preferences and also in terms of the specified client-side resource constraints, a multi-stage clustering strategy is proposed to group similar client requests together.

The remainder of the paper is organized as follows. Section 2 provides a brief review of related work. Section 3 discusses the computation of the relevance values of the video segments and video summaries in response to a client request. In Section 4, various video personalization strategies based on variations of the classical Knapsack Problem (KP) are discussed and the proposed MMKP-based video personalization strategy is detailed. Section 5 provides details of the proposed multi-stage client request aggregation strategy. In Section 6, experimental evaluation results of the proposed MMKP-based video personalization are compared with those of existing 0/1KP-based and FKP-based personalization strategies. Experimental results of the proposed multi-stage client request aggregation strategy are also provided. Section 7 concludes the paper with an outline for future work.

2. BRIEF REVIEW OF RELATED WORK

Video summarization is a field of active research in computer vision and multimedia, and constitutes the first step towards video personalization.

Definition 2: A video summary is defined as a set of still or moving image frames which represents the semantic content of a video segment.

Various innovative key frame selection algorithms have been proposed in the literature in the context of video summarization. Doulamis et al. [2] use a content-sampling algorithm to extract a small set of key frames from a video stream. Kim et al. [3] take advantage of the objects of interest in the video along with their actions and the resulting events to generate a video abstraction. For panning videos with moving object(s) against a static background, dynamic motion panoramas have been used to represent both dynamic and static scene elements in a geometrically consistent manner [4][5].

To facilitate content-based retrieval, video summaries are typically organized in a hierarchical manner. Jaimes et al. [6] propose a visual information indexing framework for systematic representation of image and video data based on syntax and semantics. In the proposed client-centered video personalization system, content-aware key frame selection algorithms and dynamic motion panoramas are used to generate video summaries.

Various personalization strategies have been proposed in the literature to generate the optimal response to the client's request while satisfying various client-side resource constraints. The optimal response to the client's request is defined as a set of video summaries that is most relevant to the client's content preference(s). Merialdo et al. [7] demonstrate that the video personalization problem can be modeled as the classical 0/1 Knapsack Problem (0/1KP). Tseng et al. [8],[9] propose a personalization strategy based on a combination of 0/1KP-based optimization and context clustering to collect successive similar shots. Context clustering is shown to be an enhancement of the scheme proposed in [8] in that it considers the temporal smoothness of the generated video summary in order to improve the client's viewing experience. One of the drawbacks of 0/1KP-based video personalization strategies is that some of the video segments which are excluded in the response to the client's request may still contain information that is potentially relevant or of interest to the client. Another drawback of 0/1KP-based personalization strategies is that they can satisfy only a single client-side resource constraint, such as the viewing time limit, at a time.

MMKP-based optimization has been used in the design of an adaptive multimedia system (AMS) [10]. The admission control in an AMS, where the clients are required to pay a fee based on the desired quality of service, is modeled as an MMKP [10]. A certain quality of service is deemed to consume a predetermined set of server resources. In order to maximize the net revenue generated by providing multimedia services to a client population, the multimedia server admits an optimal set of service requests by solving an MMKP. Since the MMKP is known to be NP-hard, heuristic algorithms are proposed to solve the MMKP for real time applications [10], [11], [12]. It needs to be noted that the AMS admission control problem is quite distinct from the video personalization problem discussed in this paper even though both problems are modeled as an MMKP. First, the objective in the case of the AMS admission control problem is to maximize the net revenue generated by the server; whereas that in the case of video personalization is to maximize the total relevance value of the video content delivered in the response to a client's request. Second, the constraints in the AMS admission control problem are on the server-end system-level resources; whereas those in the case of the video personalization problem are on the client-end system-level resources.

Service request aggregation techniques have been discussed in the context of multimedia streaming systems [13], [14]. Existing video personalization strategies published in the literature [7], [8], [9] address primarily a single-client scenario. Yu et al. [15] investigate user behavior and access patterns in a large video-on-demand (VOD) system. They report that the popularity of videos and user requested session lengths exhibit certain statistical distributions. The user request arrival pattern can be modeled using a modified Poisson distribution. Their findings indicate that multiple clients may request similar video content with similar viewing limits in a given time duration. In this paper, we therefore propose a multi-stage clustering-based client request aggregation strategy with a goal to reduce the server and network load and simultaneously improve the client-experienced latency.

3. RELEVANCE VALUE COMPUTATION

In Section 3.1, we discuss the computation of the relevance value of a video segment in response to the client's content preference(s). In Section 3.2 we discuss the computation of the relevance value of a summarized (or transcoded) version of a video segment based on its relative duration and the relevance value of the original version of the video segment.

3.1 Relevance Value of a Video Segment

Video segments are indexed using semantic terms derived from an ontology such as WordNet [16]. In well organized videos, the video can be viewed as a sequence of semantic units (genres) that are concatenated based on a predefined video program syntax. In the proposed system, we define six semantic terms for TV broadcast news videos, i.e. *News Anchor*, *News*, *Sports News*, *Commercial*, *Weather Forecast* and *Program Header*, and three semantic concepts for Major League Soccer (MLS) video, i.e. *Zoom Out*, *Close Up* and *Replay*. Each video segment is assigned a relevance value based on the client's preference with regard to video content. Assume video segment S_i is indexed by a semantic term T_i. In its request, the client specifies a preference for video content using a descriptive term denoted by P which is also derived from the same ontology. The relevance value V_i assigned to the video segment S_i is then given by:

$$V_i = similarity(T_i, P), 0 \le V_i \le 1 \qquad (3.0)$$

In the current implementation the *similarity* is evaluated using the *lch* semantic similarity measurement algorithm [16]. The *lch* algorithm measures the length of the shortest path between two concepts in the hierarchical WordNet lexical database and scales the value by the maximum *is-a* path length in the hierarchy. For example, *similarity ("anchor", "newsreader)* = 1.56, and *similarity("anchor", "news")* = 0.99.

3.2 Relevance Value of a Video Summary

We now discuss the computation of the relevance value of a video summary given its relative duration and the relevance value of the original video segment computed using equation (3.0). Each indexed video segment is summarized at multiple levels of abstraction using content-aware key frame selection and motion panorama computation algorithms. Each video summary consists of a set of key frames and motion panoramas. 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. This is so because at a higher level of abstraction, fewer image frames are included in the video summary. Since the FKP-based and the MMKP-based video personalization strategies could potentially include both the original video segments and their summaries, the relationship between the relevance value of the original video segment and the relevance value of each of its summaries needs to be first established.

For each video segment, the 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. It is reasonable to assume that the amount of information contained within a video summary (relative to the original version) is related to its relative duration, i.e.

$$v_i = v_{i0} \cdot f(L_i / L_0) \qquad (3.1)$$

where v_{i0} is the relevance value of the original video segment, and L_0 and L_i are the time durations of the original video segment and the video summary respectively.

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 paper, we propose to use the empirical Zipf's law [17] to quantify $f(L_i / L_0)$. For some categories of videos, such as news broadcast, most of the information is revealed in the first 20%-30% of the video segment. In a typical news broadcast video, a news anchor summarizes the news events at the beginning of the video segment which is then followed by detailed field news. This observation justifies the use of the Zipf function. The mathematical definition of the Zipf function is given by:

$$I = H_{k,s} / H_{N,s} \qquad (3.2)$$

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 a video summary's duration, $s > 0, s \in R$ is the characteristic parameter of the Zipf function and $H_{k,s}$ is the k^{th} generalized harmonic number [17]. When $s = 0$, the information content within a video summary increases linearly (i.e., at a constant rate) with its duration.

Equation (3.2) is a definition of the discrete Zipf function. 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 equation (3.2), the following approximation and linear transform are used. Let L_{norm} denote the normalized and discrete video duration where $L_{norm} \in \{0.01, 0.02, ..., 0.99, 1.00\}$ and let $N = 100$. Then the following linear transform maps the values of L_{norm} to k, i.e.,

$$k = round(L_{norm} \times N) \qquad (3.3)$$

Figure 1 depicts the relationship between the relative information content of a video summary versus its normalized discrete duration under the Zipf's law-based mapping function. The relative duration and relative information content of the video summary are normalized to lie within the range [0, 1] based on the duration and information content of the original video segment. In Figure 1, the parameter s takes values 0, 0.5, 1.0 and 1.5.

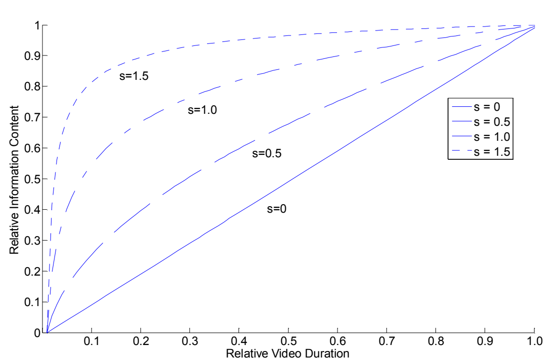


Figure 1. Relative Information Content of a Video Summary versus Segment Duration: Zipf Function.

4. PERSONALIZATION STRATEGIES

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 but within the system-level resource constraints imposed by the client. The client typically wants to retrieve and view only the video content that matches his/her content preferences. In order to generate the personalized video summary, the client preferences, client usage environment and client-side system-level resource constraints need to be considered. The personalization engine needs to select the optimal set of video contents (i.e., the most relevant set of video summaries) for the client within the resource constraints imposed by the client.

This paper presents the design and implementation of an MMKP-based video personalization strategy to generate a customized response to the client's request while satisfying multiple client-side system-level resource constraints. Compared to the 0/1KP-based and the FKP-based video personalization strategies presented in [7], [8] and [9], 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 be capable of simultaneously satisfying 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.

In the video database, each video segment is assigned a relevance value based on the client's content preferences, as computed in equation (3.0). The relevance value of a video summary is computed using equation (3.1). We propose to use the Zipf's law defined in equations (3.2) and (3.3) to quantify $f(L_i / L_0)$.

Merialdo et al. [7] propose that video personalization be modeled along the lines of the classical 0/1 Knapsack Problem (0/1KP) defined by:

$$\max_{i \in \{1,2,...,n\}} (\sum_i V_i),$$

subject to

$$\sum_i L_i \le T \qquad (4.1)$$

where L_i is the time duration of video segment i, T is the client video viewing time limit and n is the number of candidate video segments. In the 0/1KP-based video personalization strategy, a video item is either included in or excluded from the response. However, some of the video segments which are excluded in the response may still contain some useful information that is potentially of interest to the client. The 0/1KP-based video personalization algorithm does not convey this information to the client in its generated response.

In order to include more relevant video content to fill the capacity of the knapsack, i.e., the client viewing time limit in this case, the video personalization problem is formulated along the lines of the following fractional knapsack problem (FKP):

$$\max_{i \in \{1,2,...,n\}} (\sum_i x_i V_i),$$

subject to

$$\sum_i y_i L_i \le T \qquad (4.2)$$

where T is the client video viewing time limit, L_i is the temporal length of *video segment* S_i, and x_i and y_i, where $x_i, y_i \in [0,1]$, are fractional factors pertaining to the *video segment*'s relevance value and its duration respectively. The above FKP can be solved by using a greedy algorithm. Video segments are sorted in decreasing order of their *Value_Intensity* which is computed as $Value\_Intensity_i = V_i / L_i$, where V_i is the relevance value and L_i is the time duration of video segment S_i. Video segments with high *Value_Intensity* values are selected first. Although the FKP-based optimization scheme can include transcoded video segments, some potentially relevant videos could be excluded in 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. In the case of the FKP-based video personalization, a fractional portion of a single video segment could be included in the response generated by the video personalization module. The last video segment in the generated response could be summarized or shortened to enable it to fit within the limit of the client's video viewing time (i.e., the knapsack capacity in our formulation).

In the aforementioned 0/1KP-based and FKP-based video personalization strategies, some video segments and their corresponding summaries could be excluded from the generated response to the client's request. Furthermore, simultaneously satisfying multiple client-side resource constraints, is beyond the capabilities of the 0/1KP-based and FKP-based video personalization strategies.

In many applications, it is desirable to provide the client with as much information as possible. In such cases it may be preferable to include two shorter video summaries in the generated response rather than a single video segment of longer duration that contains more detail. For example, if a client needs to browse the sports news of the day, it might be helpful to provide him/her with multiple, though short, sports news summaries rather than a single long and detailed video segment containing news of a specific sport. We propose a Multiple-Choice Multi-Dimensional Knapsack Problem [10], [11], [12] (MMKP)-based video personalization strategy to address this issue.

<u>Definition 3:</u> A *content group* consists of a video segment and its summaries at multiple levels of abstraction.

Each original video segment S_i is summarized into $l_i - 1$ summaries. The video segment S_i and its $l_i - 1$ summaries $S_{ij}, j \in \{1,2,...,l_i\}$ constitute a multi-level *content group*, as shown in Figure 2.

We denote the original video segment and each of its summaries as an *item*. Each *item* is associated with a relevance value and is deemed to require *m* resources. The computation of the relevance value of a video segment or video summary has already been discussed in Section 3. 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 *items*, subject to the *m* resource constraints of the client.

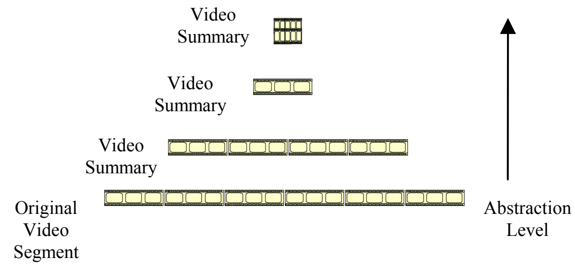


Figure 2. Multiple Abstraction Level *Content Group*

The MMKP-based video personalization strategy is formulated as follows.

Let v_{ij} be the relevance value of the j^{th} summary of the video segment S_i, $\vec{r}_{ij} = (r_{ij1}, r_{ij2}, ..., r_{ijm})$ be the required resource vector for the j^{th} summary of the video segment S_i and $\vec{R} = (R_1, R_2, ..., R_m)$ be the vector that denotes the client-side resource bounds. The problem thus is to determine

$$V = \max\left(\sum_{i=1}^{n}\sum_{j=1}^{l_i} x_{ij} v_{ij}\right),$$

subject to the constraints

$$\sum_{i=1}^{n}\sum_{j=1}^{l_i} x_{ij} r_{ijk} \le R_k, k = 1,2,...,m$$

and

$$\sum_{j=1}^{l_i} x_{ij} = 1, x_{ij} \in \{0,1\} \qquad (4.3)$$

The MMKP-based video personalization strategy is illustrated in Figure 3. Video segments at the bottom of each content group are the original versions. Each original video segment has two associated summaries. The variable v_{ij} denotes the relevance value of the j^{th} *item* in the i^{th} *content group* whereas t_{ij} and b_{ij} denote respectively the duration and the amount of data associated with the j^{th} *item* in the i^{th} *content group*. We assume that the client has two resource constraints, i.e., the viewing time limit T and the total received data limit B. The goal of the MMKP-based video personalization strategy is to select exactly one *item* from each *content group* such that the total relevance value $\sum_{i=1}^{3}\sum_{j=1}^{3} x_{ij} v_{ij}$ is maximized subject to the constraints $\sum_{i=1}^{3}\sum_{j=1}^{3} x_{ij} t_{ij} \le T$ and $\sum_{i=1}^{3}\sum_{j=1}^{3} x_{ij} b_{ij} \le B$. The constraint

$\sum_{j=1}^{l_i} x_{ij} = 1, x_{ij} \in \{0,1\}$ guarantees that one and only one *item* is selected for each *content group*. The selected item for each *content group* is marked with a star in Figure 3. The client content preferences and system-level resource constraints are treated separately and differently by the video personalization strategy. The client content preferences are used to compute the relevance values of video segments and their summaries, as discussed in Section 3. The client-side system-level resource limits are specified as explicit constraints that need to be satisfied by the video personalization strategy.

The MMKP, as formulated above, can be solved using the branch and bound integer programming (BBIP) algorithm described in [18]. It should be noted that the FKP-based and MMKP-based video personalization strategies are fundamentally different. First, the MMKP-based personalization strategy (equation (4.3)) is able to satisfy multiple resource constraints (such as the viewing time limit and total received data limit) simultaneously; whereas the FKP-based personalization strategy (equation (4.2)) can only satisfy a single resource constraint at a time. Second, the MMKP-based personalization strategy selects exactly one *item* from each *content group*; whereas the FKP-based personalization strategy may exclude some video segments in the generated response. Interested readers are referred to [19] for further details on the formulations of the FKP and MMKP.

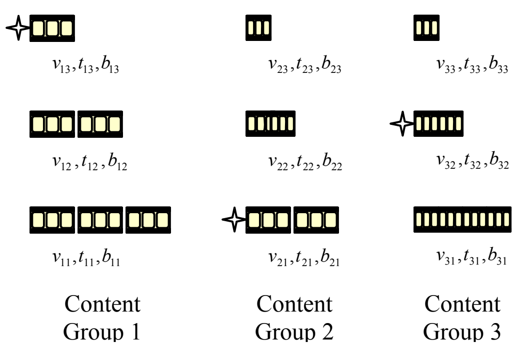


Figure 3. A Depiction of MMKP-based Video Personalization

5. MULTI-CLIENT PERSONALIZATION

5.1 Multi-stage Client Request Aggregation

Client requests for personalized videos in a multiple client scenario are heterogeneous in nature. The heterogeneity can be manifested in the following three ways.

(1) <u>*Arrival time heterogeneity*</u>: Client requests tend to arrive at different time instances.

(2) <u>*Video object heterogeneity*</u>: The client-specified video content preferences vary from client to client.

(3) <u>*Client-side constraint heterogeneity*</u>: Each client request is associated with a set of client-side resource constraints. These constraints can vary significantly from client to client.

The goal of the proposed multi-stage client request aggregation strategy is to reduce the effective number of requests to be processed by the server. It consists of the following steps as shown in Figure 4.

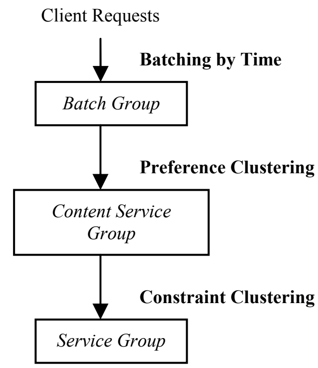


Figure 4. Multi-stage Client Request Aggregation

<u>Step 1:</u> <u>*Batching by time.*</u> Batching by time is performed to group multiple client requests that arrived within a prespecified time window. The resulting group is termed a *batch group*.

<u>Step 2:</u> <u>*Preference clustering.*</u> Client requests within a *batch group* with similar video content preferences are clustered into *content service groups*. The details of *preference clustering* procedure are as follows.

Let q denote the ordered set of semantic terms used to index video segments, and Q denote the ordered set of semantic terms that the clients can use in their requests to specify their video content preferences, such that $q \subseteq Q$. Also assume that a similarity matrix $S_{size(q)\times size(Q)}$ is used to define the similarity of semantic terms, where $0 \le s_{ij} \le 1$ represents the semantic similarity of term $t_i \in q$ and term $t_j \in Q$. The value of s_{ij} is computed using the *lch* algorithm described in Section 3.1. Let T be the semantic term used in the client's video content preference list, and let *k* be the index of term T in the ordered set Q, then the client *content preference vector* P_c is defined as follows:

$$P_c = (s_{1k}, s_{2k}, \ldots s_{size(q)k}) \qquad (5.1)$$

where $s_{ik}, 1 \le i \le size(q)$ is the semantic similarity between term $t_i \in q$ and term $k \in Q$, and can be obtained from the similarity matrix $S_{size(q)\times size(Q)}$. The *preference clustering* algorithm uses the cosine similarity measure between a pair of client query content preference vectors P_{c1} and P_{c2} to represent the distance between them. The *k-means* clustering algorithm is used to cluster client requests with similar content preference values into a group, termed as a *content service group*. The number of the content service groups is less than the number of client requests, and is dynamically adjusted in our experiments. In our experiments, since the number of client requests in a *batch group* is variable, the maximum number of preference clusters generated from a *batch group* is given as follows:

$$N_{pc\_\max} = N_c \times PAF \qquad (5.2)$$

where N_{pc_max} is the maximum number of preference clusters generated from a *batch group*, N_c is the number of client requests in a *batch group*, and PAF denotes the *Preference Aggregation Factor*, where $0 \le PAF \le 1$.

Step 3: *Client-side constraint clustering.* Client requests within a *content service group* are further clustered based on client-side system-level resource constraints to generate a set of *service groups*. In our experiments, client requests with similar values for the video viewing time limit are clustered together using the *k-means* clustering algorithm. In our experiments, the number of *service groups* (viewing time limit clusters) is adjusted as follows.

$$N_{vtc} = N_c^{'} \times CAF \quad (5.3)$$

where N_{vtc} is the number of *service groups* (viewing time limit clusters) generated for a *content service group*, $N_c^{'}$ is the number of client requests in a *content service group*, and *CAF* denotes the *Constraint Aggregation Factor*, where $0 \le CAF \le 1$.

The multimedia server processes the set of client requests in a *service group* as a whole, i.e., only a single video response is generated for all the client requests within a *service group*. In order to represent the set of client requests in *a service group*, the video content preference vector which is closest to the centroid of the *service group* is selected. The representative client-side resource constraint is set to the mean value of the resource constraints within a *service group*. For instance, the representative client viewing time constraint is the mean value of viewing times specified by the set of client requests in a *service group*.

5.2 Performance Metrics

In order to measure the performance of the proposed client request aggregation strategy, the overall client-server relationship is modeled using a single server and a single client request queue as shown in Figure 5.

The following metrics are used to evaluate the performance of the proposed client request aggregation strategy.

1. *Mean client-experienced latency* (in seconds) is defined as the average duration of the interval between the time instant when the client sends a request to the server and the time instant when the client starts to receive video feedback from the server.

2. *Mean client-experienced preference dissimilarity* is defined as the average semantic dissimilarity between the video content preferences specified by the client and the personalized video received. The *lch* algorithm discussed in Section 3.1 is used to compute the dissimilarity. This metric measures how different, on average, the received video contents are from the client-specified content preferences.

3. *Mean client-experienced viewing time difference* (in seconds) is defined as the average absolute value of the difference between the client-specified viewing time and the duration of the video feedback received from the server. In the request sent to the server, there could be multiple client-side resource constraints, of which the viewing time is the most important. This metric measures how close the duration of the received video is to the client-specified duration.

The duration of the batch windows, the number of client video content preference clusters, (i.e., PAF), and the number of client viewing time clusters are shown to influence the overall performance of the proposed client request aggregation strategy.

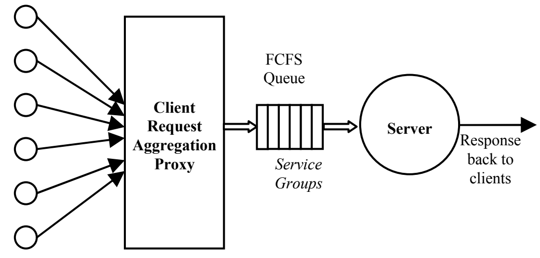


Figure 5. Single Server, Single Client Request Queue Model for Performance Evaluation of Client Request Aggregation

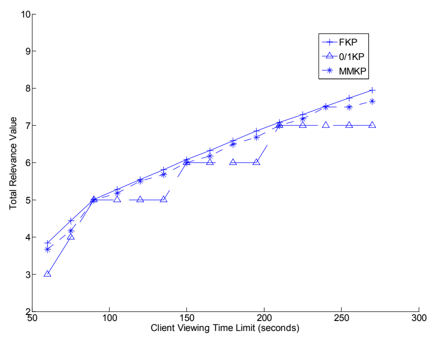
6. EXPERIMENTAL RESULTS

6.1 Video Personalization Results

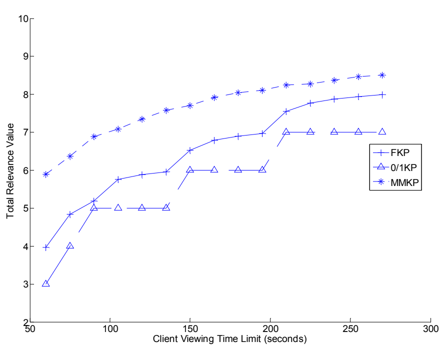
TV broadcast video streams are recorded and digitized. These video streams are automatically segmented, labeled and indexed using the stochastic multi-level HMM-based algorithm described in [1]. Each video segment is labeled with terms selected from a predefined video content description ontology. Video segments are summarized at multiple levels of abstraction using algorithms for content-aware key frame selection and motion panorama computation. The resulting *video items* are stored in a video database.

For the purpose of experimentally comparing the performance of the various aforementioned video personalization strategies, we set the client's content preference to a fixed semantic concept, i.e., *News*. Relevance values of a video segment and its summaries are computed using equations (3.0) and (3.1) respectively. The Zipf's law-based mapping function is used to compute the relevance values of the *video items*. The sum of relevance values of the *video items* included in the response to the client's request is used to quantify and compare the performance of the video personalization strategies. We choose the semantic concept *News* primarily, because our video database contains a large number of fairly diverse *News* video segments which precludes a pre-existing bias for or against the proposed empirical mapping function.

In Figure 6(a), the Zipf function and linear transform defined in equations (3.2) and (3.3) respectively are used to compute the relevance values of the video summaries based on their time durations and the relevance values of the original video segments. The total relevance value of the response to the client's request is plotted against the client's video viewing time limit. In Figure 5(a), the characteristic parameter s of the Zipf function is set to zero. As shown in equation (4.3), the MMKP-based video personalization strategy selects one *item* from each *content group* where an *item* is defined as one of the video segments or video summaries in a *content group*. Hence more *video items* are included in the response generated by the MMKP-based video personalization strategy compared to the 0/1KP-based and FKP-based video personalization strategies. This implies that when the number of candidate *content groups* is large, the time durations of the video segments and/or video summaries included in the response are short. It is observed in Figure 1, that when $s = 0$, the relative information content of a video summary increases linearly with its time duration under the Zipf mapping function. In this case, the total information content of these short video segments or video summaries in the response generated by the MMKP-based video personalization strategy is generally less than that of those included in the response generated by the FKP-based video personalization strategy. However, if we assume that beginning portion of a video segment contains the major portion of its information content, for example when $s = 1.0$, then the short video segments selected by MMKP-based personalization strategy 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 6(b).



(a) Using the Zipf Function, $s = 0$



(b) Using the Zipf Function, $s = 1.0$

Figure 6. Total Relevance Value of the Response versus the Client Viewing Time Limit

A principal advantage of the MMKP-based video personalization strategy is that it can satisfy simultaneously multiple client-side resource constraints whereas both, the FKP-based and 0/1KP-based video personalization strategies can satisfy only a single client-side resource constraint at a time. In Figures 6(a)-(b), the client-side resource constraint is the client viewing time limit. Figure 7 shows the experimental results of the MMKP-based personalization scheme when the client has two system-level resource constraints, i.e., a viewing time limit and a limit on the total amount of data received. The Zipf's law-based mapping function with $s = 1.0$ is used in this case. The received data is limited to at most 3 KBytes for each second of the received video stream. The data limit constraint is held constant (3 Kbytes/second) whereas the viewing time limit constraint is varied. As seen in Figure 6, when the viewing time is less than 150 seconds, the server's response to the client's request contains no video segment (i.e., the server generates a null response). This is because in each of the *content groups*, there is no video item of size less than 450 Kbytes (=3Kbytes/second×150 seconds). When the client's viewing time limit is large enough (>150 seconds in our experiment), it is possible to include video segments or video summaries which satisfy the data limit constraint. It is clear that when both resource constraints need to be satisfied, the response contains less video information compared to the case when the client viewing time is the only resource constraint.

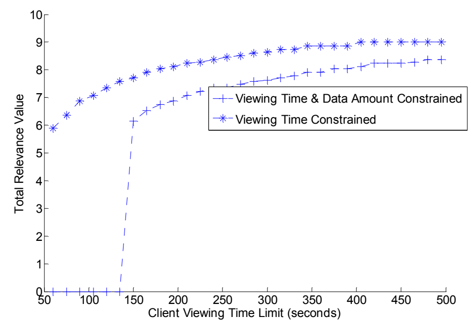


Figure 7. Performance of the MMKP-based Personalization Scheme under both, the Viewing Time Limit Constraint and the Received Data Limit Constraint (≤ 3 Kbytes/Second)

The FKP-based and 0/1KP-based personalization schemes cannot satisfy simultaneously such multiple resource constraints and are forced to handle individual resource constraints one at a time. Since different resource constraints, when employed individually, yield different solutions, determining the optimal combination these solutions to satisfy simultaneously the multiple resource constraints becomes an important (and difficult) issue in the case of the FKP-based and 0/1KP-based personalization schemes. This issue is obviously moot in the case of the MMKP-based personalization scheme since it is inherently equipped to satisfy simultaneously multiple client-side resource constraints.

6.2 Client Request Aggregation Results

The single queue-single server model discussed in Section 5.2 is used to evaluate the performance of the proposed client request aggregation strategy. In our experiments, 100 clients connect to the video personalization server via wireless networks. After a random delay, each client sends a request to the server. The durations of the delays follow a Poisson distribution $P(\lambda)$, where $\lambda = 50$ seconds. In our experiments, the clients specify their viewing time limits as their resource constraints. In order to simulate mobile devices of different battery capacities, the values of viewing time limits are modeled as a mixture of 2 normal distributions $N_1(\mu_1, \sigma_1^2)$ and $N_2(\mu_2, \sigma_2^2)$, where $\mu_1 = 50$ seconds, $\sigma_1 = 70$ seconds, $\mu_2 = 150$ seconds and $\sigma_2 = 70$ seconds respectively. Each of the clients picks its content preference terms from the set Q. Each semantic term in Q has the same probability to be selected. We measure the client-experienced latency, preference dissimilarity and viewing time difference for every client. Each experiment is repeated 10 times. On the server side, the duration of the batch windows is empirically set to 20 seconds. We change the values of *PAF* and *CAF* systematically and measure the resulting performance of the aggregation strategy.

The system parameters, i.e., the values of *PAF* and *CAF*, influence the performance of the client request aggregation strategy. Figures 8-10 show the relationships between each of the performance metrics, i.e., the mean client-experienced viewing time difference, mean client-experienced preference dissimilarity and mean client-experienced latency and the two system parameters *PAF* and *CAF* respectively. Figure 11 shows the relationship amongst the above three performance metrics. For representative combinations of the above three performances metrics, the corresponding system parameters (i.e., *PAF* and *CAF*) are indicated in Figure 11.

It is observed in Figures 8-10 that when no client request aggregation is performed, i.e., when both the values of *PAF* and *CAF* are set to one, each client request is processed individually. In this case, each client get exactly its preferred content within the duration of its viewing time limit, i.e., the mean client-experienced preference dissimilarity and mean viewing time difference are zero. However, the mean client-experienced latency is observed to be very long since every single client request needs to be processed individually by the server.

In Figure 8, it is observed that when the value of *CAF* is relatively small, the mean client-experienced viewing time difference is large. This is because with a small number of viewing time clusters, in a *service group*, the differences amongst the client-specified viewing time limits are large. When the value of *CAF* increases, it is seen in Figure 8 that the mean client-experienced viewing time difference decreases. As regards the client-experienced preference dissimilarity, it is observed in Figure 9 that the client-experienced preference dissimilarity decreases when the value of the *PAF* increases. Figures 8 and 9 show that the client-experienced viewing time difference and the preference difference are primarily decreasing functions of *CAF* and *PAF* respectively. This is because client preferences and constraints are clustered at different stages separately. The multi-stage clustering strategy allows us to control the desired client-experienced preference dissimilarity and constraint difference separately.

Figure 10 shows that either increasing the value of *CAF* or the value of *PAF* causes the mean client latency to increase. It needs to be noted here that when both, the number of client viewing time clusters and the number of client preference clusters are small, the mean client latency is short. However, the price of this short mean client latency is the larger client viewing time difference and larger client preference dissimilarity.

Figure 11 helps to determine the tradeoffs amongst the three performance metrics when considered simultaneously. For example, a system administrator can use Figure 11 to determine the resulting mean client latency when the mean client-experienced viewing time difference and preference dissimilarity are specified.

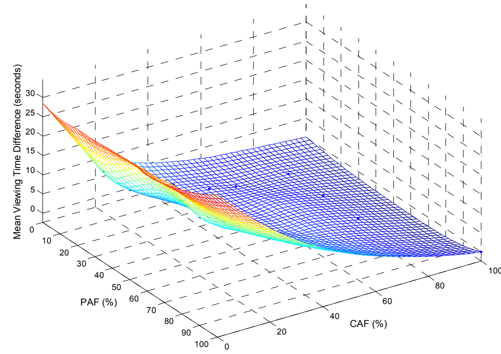


Figure 8. Mean Client-experienced Viewing Time Difference versus *PAF* and *CAF*

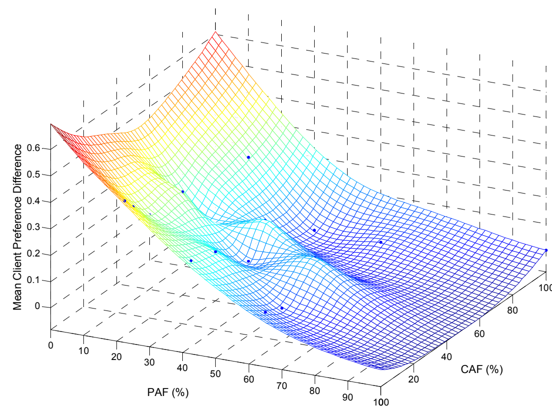


Figure 9. Mean Client-experienced Preference Difference versus *PAF* and *CAF*

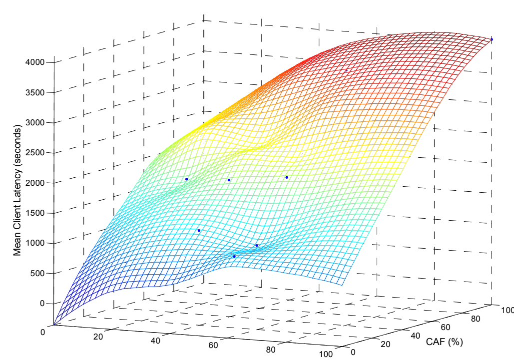


Figure 10. Mean Client-experienced Latency vs. *PAF* and *CAF*

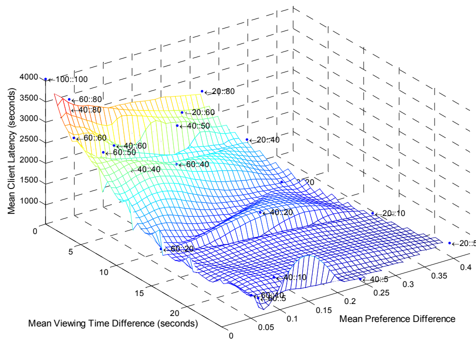


Figure 11. Mean Client-experienced Latency vs. Mean Viewing Time Difference and Mean Preference Difference. Corresponding *PAF* and *CAF* Values are Labeled in the Graph

7. CONCLUSIONS

In order to provide a resource-constrained client with its preferred video content, it is necessary to perform video personalization. In the proposed scheme, video segments are indexed and summarized at multiple levels of abstraction. In order to compare the information content of the video summaries relative to the original video segments, the Zipf's law-based mapping is employed. The Zipf function is used to compute the relevance value of video summaries given their relative time durations and the relevance values of the corresponding original video segments. This is in conformity with the commonly observed law of diminishing marginal return.

Experimental results show that the proposed MMKP-based personalization strategy includes more relevant video content in response to a client's request compared to the 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 capable of satisfying simultaneously multiple client-side resource constraints. Experimental results also show that the proposed multi-stage client request aggregation strategy reduces the effective number of client requests processed by the server thus reducing both, the server and network load and the client-experienced latency.

Future work would include validation of the proposed empirical mapping function and application of the proposed video personalization system to a wide collection of video data to test its robustness and extensibility. Human subject evaluation of the generated responses to the clients' requests needs to be explored in order to further validate this work. Data-driven clustering algorithms also need to be investigated to optimally aggregate client requests and client-side resource constraints.

8. REFERENCES

[1] Wei, Y., Bhandarkar, S.M., and Li, K., Client-Centered Multimedia Content Adaptation. *ACM Trans. Multimedia Computing, Communications and Applications,* Submitted.

[2] Doulamis, N., Doulamis, A., Avrithis, Y., Ntalianis, K., and Kollias, S., Efficient Summarization of Stereoscopic Video Sequences. *IEEE Trans. Circuits and Systems for Video Technology,* Vol. 10, No. 4 (2000), 501–517.

[3] Kim, C., and Hwang, J., An Integrated Scheme for Object-based Video Abstraction. *Proc. ACM Conf. Multimedia,* (Los Angeles, CA, 2000), 303–311.

[4] Wei, Y., Wang, H., Bhandarkar, S.M., and Li, K., Parallel Algorithms for Motion Panorama Construction. *Proc. ICPP Workshop on Parallel and Distributed Multimedia*, (Columbus, Ohio, Aug. 2006), 82-92.

[5] Bartoli, A., Dalal, N., and Horaud, R., Motion Panoramas. *Computer Animation and Virtual Worlds*, Vol. 15 (2004), 501-517.

[6] Jaimes, A., and Chang, S.F., A Conceptual Framework for Indexing Visual Information at Multiple Levels. *IS&T/SPIE Internet Imaging*, Vol. 3964, (San Jose, CA, Jan. 2000), 2-15.

[7] Merialdo, B., Lee, K.T., Luparello, D., and Roudaire, J., Automatic Construction of Personalized TV News Programs. *Proc. ACM Conf. Multimedia*, (Orlando, FL, Sept. 1999), 323—331.

[8] B.L. Tseng and J.R. Smith, Hierarchical Video Summarization Based on Context Clustering, *Proc. SPIE*, Vol. 5242 (Nov. 2003), 14-25.

[9] Tseng, B.L., Lin, C.Y., and Smith, J.R., Video Personalization and Summarization System for Usage Environment. *Jour. Visual Communication and Image Representation*, Vol. 15 (2004), 370–392.

[10] Khan, S., Li, K.F., Manning, E.G., and Akbar, M.M., Solving the Knapsack Problem for Adaptive Multimedia Systems, *Studia Informatica Universalis*, 2(1), pp. 61-82, 2002.

[11] Akbar, M.D., Manning, E.G., Shoja, G.C., and Khan, S., Heuristic Solutions for the Multiple-Choice Multi-dimension Knapsack Problem. *Proc. Intl. Conf. Computational Science*, 2001, 659-668.

[12] Hernandez, R.P., and Nikitas, N.J., A New Heuristic for Solving the Multichoice Multidimensional Knapsack Problem*. IEEE Trans. System, Man and Cybernetics*, Part A, Vol. 35, No. 5 (2005), 708-717.

[13] Bommaiah, E., Guo, K., Hofmann, M. and Paul, S., Design and Implementation of a Caching System for Streaming Media Over the Internet, *IEEE Real Time Technology and Applications Symposium(RATS, 2000)* , 111-121.

[14] Wu, S., Jin, H., Chu, J. and Fan, K., A Novel Cache Scheme for Cluster-based Streaming Proxy Server, *25th IEEE Intl. Conf. Distributed Computing Systems Workshops*, (June 2005), 727 – 733.

[15] Yu, H., Zheng, D., Zhao, B. and Zheng, W., Understanding User Behavior in Large Scale Video-on-Demand Systems, *Proc. of the 1st EuroSys,* Leuven, Belgium, April, 2006.

[16] Leacock, C., and Chodorow, M., Combining Local Context and WordNet Similarity for Word Sense Identification. In *WordNet: An Electronic Lexical Database*, C. Fellbaum (Editor), MIT Press, Cambridge, MA, 1998, 265-283.

[17] Wheeler, E.S., Zipf's Law and Why It Works Everywhere. *Glottometrics*, Vol.4 (2002), 45-48.

[18] Vanderbei, R. J., *Linear Programming: Foundations and Extensions*, Kluwer Academic Publishers, 1997.

[19] Kellerer, H., Pferschy, U. and Pisinger, D., *Knapsack Problem*, Springer-Verlag, Berlin, Germany, 2004.