TR-ITS-2005.010, Swiss Federal Institute of Technology (EPFL), Lausanne, April 2005.

Personalized Content Preparation and Delivery for Universal Multimedia Access

Olivier Steiger, Miguel Schneider Fontan, David Marimon-Sanjuan, Yousri Abdeljaoued, Touradj Ebrahimi, Sergio Dominguez, Jose San Pedro Wandelmer, Nicolas Denis, Fabrizio Granelli and Francesco De Natale

***Abstract*— This paper introduces a personalized content preparation and delivery framework for universal multimedia access. This framework has a wealth of potential applications in broadcasting, the World Wide Web and mobile telecommunication. Content preparation deals with the generation of summaries by automatic video analysis techniques, and with the annotation of content using MPEG–7 for search and retrieval and SMIL for interactive video. The use of open standards favors interoperability with third-party applications and upcoming hardware. Content delivery handles individual user preferences, content selection, real-time content adaptation and streaming in a seamless way. The proposed framework has been implemented in the EC–funded R&D project PERSEO and tested by selected groups of end-users. Personalization, video summaries and the overall usability have been positively evaluated.**

***Index Terms*— Universal Multimedia Access, Content preparation, Content delivery, Video streaming, Personalization, MPEG.**

I. Introduction

NOWADAYS, a wide range of different networks (e.g., PSTN, GSM, LAN, WLAN) and appliances (PC, PDA, mobile phone) are used to access audiovisual content. To cope with the diverse resource capabilities of these networks and devices, and with the individual needs and preferences of end users, personalized content preparation and delivery are essential. This framework, where information is delivered to all types of users under a wide variety of conditions in a transparent form, is referred to as universal multimedia access (UMA) [1], [2]. UMA has a wealth of potential applications in broadcasting, the World Wide Web and mobile telecommunication [3].

Several researchers have been using the MPEG family of standards to target specific challenges of UMA. Van Beek *et al.* [4] use MPEG–7 Variation and Summary tools to provide several alternative content versions to the client. MPEG–7 Transcoding Hints further indicate how multimedia content can be adapted or transcoded. Fossbakk *et al.* [5] perform similar tasks using MPEG–21 Digital Item Declaration. MPEG–21 usage environment description tools are used to provide resource adaptation in a streaming media environment by Sun *et al.* [6], [7]. The usage of MPEG–21 in a UMA context has also been studied by Bormans *et al.* [8]. One concern in the past has been the problem of generating and coding appropriate content for UMA. Some of the basic technology required for media conversion to support mobile users is reviewed by Vetro and Sun in [9]. Lee *et al.* [10] propose a scheme for generating transcoded video sequences to fit the size of the respective display of a variety of client devices. Their scheme uses an image transcoding algorithm based on perceptual hints. MPEG–4 fine-granular scalability coding (FGS) has been adapted in transmission of video over wireless and mobile networks by van der Schaar and Radha [11]. Wang *et al.* [12] combine MPEG–4 FGS and MPEG–21 into a unique testbed for real-time video streaming over heterogeneous networks with time-varying conditions and devices with different capabilities. Content adaptation to different user profiles for digest video delivery and for mobile multimedia services has been reported on by Echigo *et al.* [13] and Chen *et al.* [14], respectively.

This paper introduces a complete personalized content preparation and delivery framework for UMA. The proposed framework has been developed, implemented and tested by the authors in the EC–funded R&D project PERSEO (`http://www.perseoproject.org/`). Content preparation deals with both the creation of different content versions, or *variations*, for various usage environments, and with the annotation of such content. Content delivery refers to personalization, resource adaptation and streaming tools requested to provide UMA. The power and novelty of our solution ensue from the integration of automatic feature extraction, open standards-based content annotation, user preferences handling and usage environment-based content selection and adaptation in a unified framework.

The remainder of the paper is organized as follows. Section II gives an overview of the proposed UMA framework. Content preparation, i.e., the creation of content variations and content annotation, is dealt with in Section III. Clients, i.e., the user and the user's terminal, and the content server for personalized content delivery are presented in Section IV. The PERSEO implementation of the proposed framework and usability validation experiments using focus-group techniques are discussed in Section V. Section VI concludes the paper.

II. Overview

The universal multimedia access framework depicted in Figure 1 is subdivided into two connected parts, namely content preparation (Section III) and content delivery (Section

This work was supported by the European Commission.

O. Steiger, D. Marimon, Y. Abdeljaoued and T. Ebrahimi are with the Swiss Federal Institute of Technology, Lausanne,

M. Schneider is with Ibermatica SA, Madrid,

S. Dominguez, J. San Pedro and N. Denis are with Universidad Politecnica de Madrid,

F. Granelli and F. de Natale are with Universita degli Studi di Trento.

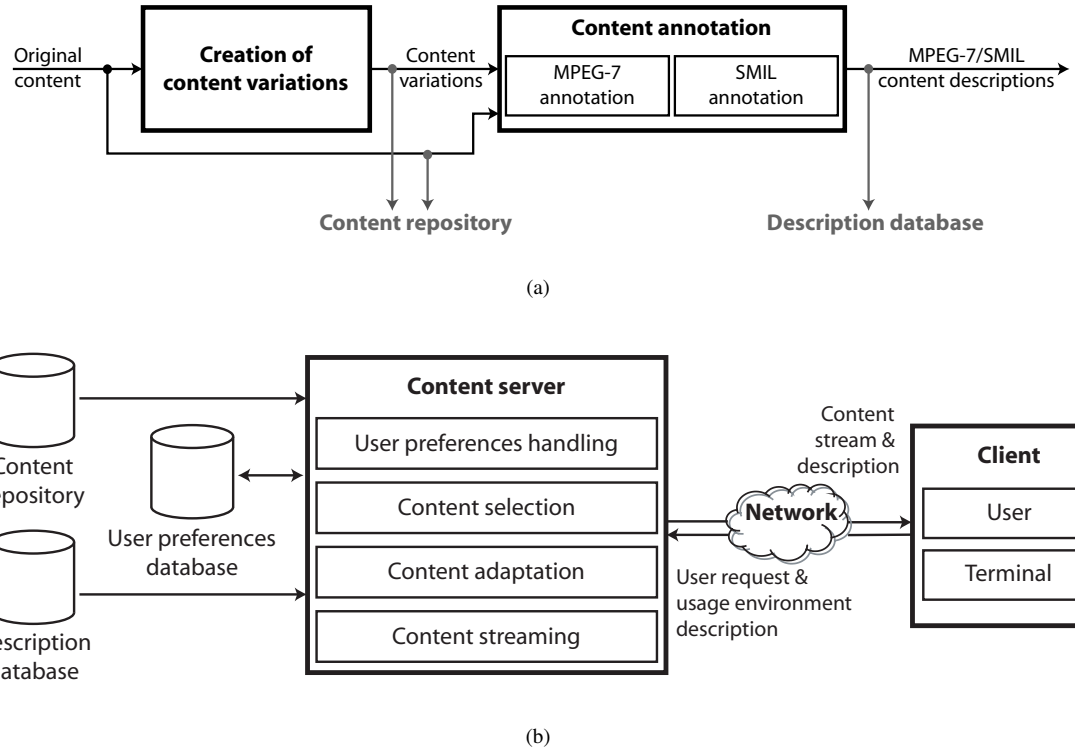


Fig. 1. The proposed universal multimedia access framework is subdivided into two connected parts. (a) Content preparation. (b) Content delivery.

IV). The goal of *content preparation* is to prepare audiovisual material for personalized delivery. This requires the creation of content variations, and the annotation of content. Variations are alternative versions of a particular piece of content, for instance in several coding formats or in several modalities. In UMA applications, variations can be selected and delivered as replacement, if necessary, to adapt to terminal capabilities, network conditions, or user preferences. As the creation of complex variations such as content summaries is a lengthy process, this must often be performed offline and is thus part of content preparation. Content annotation refers to the generation of content descriptions, or metadata. Descriptions of an audiovisual item in terms of various attributes such as author, title, genre, language and keywords, to name a few, enable users to query the system and to search for desired content. Other features like resolution, frame rate and coding format, are matched against the usage environment description to select the content variation that is to be streamed. The user can also define hyperlinks on certain image parts to create interactive video. In the proposed framework, content is annotated using the *multimedia content description interface* MPEG–7. The *synchronized multimedia integration language* SMIL serves for the authoring of interactive video.

Content delivery deals with the selection, adaptation and streaming of audiovisual material to the connected client. Media files are stored in a content repository, with the corresponding metadata stored in a description database. In addition to these, the content server maintains a user preferences database. Preferences are acquired from the user in a form and updated by the server according to the user's usage history. This is referred to as user preferences handling. Content selection takes into account user preferences and the usage environment. Content is filtered and sorted in terms of various attributes such as title, genre, language, etc., according to the preferences of the connected user. The usage environment description determines what content variation is to be streamed. When no adequate variation can be found, content is adapted online in terms of spatial and temporal resolutions, color depth, audio properties and coding format. The selected and possibly adapted content is finally streamed to the client. The streaming process takes into account varying network conditions by adapting the quality of delivered content.

III. Content preparation

The goal of content preparation is to prepare audiovisual material for personalized delivery. This requires the creation of different content variations, and the annotation of content by metadata. In Section III-A, automatic video segmentation algorithms are used to produce summaries. Section III-B presents the MPEG–7 and SMIL annotation of content. MPEG–7 metadata is required for search and retrieval, whereas SMIL annotation is intended for the generation of interactive video.

A. Creation of content variations

To accommodate the requirements of various clients and to support fast video database browsing, different variations of the same content are generated. The creation of content summaries is of particular interest in a UMA environment,

where large content collections are searched by devices with limited capabilities. Summary creation mainly consists in segmenting the video into elementary units and extracting representative frames from these units.

Figure 2 shows the block-diagram of a simple system for video summarization. First, the original video is processed in order to extract low-level visual primitives such as color, motion and texture. Based on the low-level visual primitives, the video is segmented into basic units called shots. A shot is defined as a sequence of frames captured by one camera in a single continuous action in time and space [15]. Temporal segmentation corresponds to the detection of the boundaries between these shots. Once the shot boundaries are detected, the salient content of each shot is represented in terms of a small number of frames, called key frames. Temporal segmentation and key frame extraction make up the video parsing process. Finally, different summary representations are created from the detected shots and from the extracted key frames. The main objective of these representations is to allow a content-based browsing of the video.

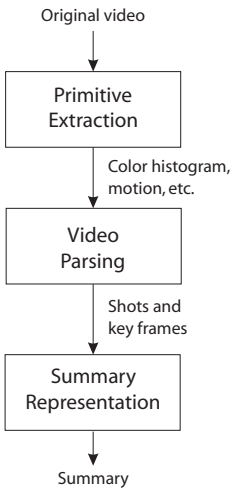


Fig. 2. Block-diagram of a simple system for video summarization.

Most techniques in the literature propose a solution only for one of these fundamental elements. In the following, we give an overview on algorithms for video parsing and summary representation. We do not include methods for visual primitive extraction here.

1) Video parsing:

- **Temporal segmentation** Color is one of the most common visual primitives used for shot boundary detection. Two consecutive frames from different shots are unlikely to have similar colors. Zhang *et al.* [16] used histograms as descriptors for color. If the computed histogram distance between two consecutive frames is higher than a threshold, a scene cut is declared. According to an evaluation of different algorithms for temporal video segmentation in [17], the color-based algorithm is one of the best. However, it has a number of limitations. Specifically, the number of false positives is still high during complex camera operations or object motion. Furthermore, gradual transitions between two shots containing camera operations are not detected.
  Zabih *et al.* [18] proposed to use edges as visual primitives for temporal video segmentation. First, edges are extracted using the Canny detector [19] from two consecutive frames. Then, a Hausdorff distance-based algorithm [20] is applied to the edge-detected images to estimate the dominant motion. This allows to align the two edge-detected images to compensate for camera operations. By computing a dissimilarity measure based on the fraction of edge-pixels which enter and exit between two consecutive frames, it is possible to detect cuts and gradual transitions as local maxima. Moving objects however can cause false positives because the dominant motion estimation algorithm does not compensate for multiple motions.
  Bouthemy *et al.* [21] proposed to use the Iteratively Reweighted Least Squares (IRLS) technique to detect the points which belong to the part of the image undergoing dominant motion (*inliers*). If a cut occurs between two consecutive frames, the number of inliers n_d is close to zero. On the other hand, if the consecutive frames are within the same shot, n_d is nearly constant. In order to normalize the similarity measure, n_d is divided by the number of points n_o which belong to the common part between the two consecutive frames. A cumulative sum test, known as the Hinkley test [22], is used to detect abrupt changes (i.e. cuts) and gradual changes (i.e. gradual transitions).
- **Key frame extraction** Nagasaka and Wang [23] proposed to select the first frame in a shot as a key frame. Although simple, this technique is not able to extract the salient content of a shot, especially for shots with motion and high activity. Instead, the current frame of a shot can be compared to the last selected key frame by using the color histogram-based distance. If this distance is higher than a threshold T_k, the current frame is selected as a new key frame. This process is iterated until the end of the shot. Thus any significant action in the shot is represented by a key frame.
  Wolf [24] proposed a motion-based approach. First the optical flow for each frame is determined [25], and a motion metric based on optical flow is computed. Then, by analyzing the motion metric as a function of time, key frames are selected at the minima of motion. This is explained by the fact that the camera stops on a new position, or that the characters maintain gestures to emphasize their importance. Combining several criteria as proposed by Zhang *et al.* [16] also yields good results for key frame extraction.

2) Summary representation:

- **Hierarchical summary** This representation allows random access. Key frames are grouped and organized in order to obtain a coarse-to-fine hierarchy of summaries, i.e., the content of video is represented at multiple levels of detail, from coarse summaries to detailed summaries. Zhong *et al.* [26] used a fuzzy K-means clustering algo-

rithm to group similar key frames in terms of a low-level visual primitive (e.g., color) into classes at each level of the hierarchy. The key frame closest to the centroid of the cluster is selected to represent each class. Due to the sequential nature of video, the visualization of such hierarchical summaries is difficult, especially when the number of key frames is large.

- **Sequential summary** This representation is simply a concatenation of the key frames which can be shown sequentially in time, for example as an animated slide show. The temporal order the of key frames in the original video is preserved. This allows to understand the relationships between the different events in the video.
- **Mosaic-based summary** Each shot is decomposed into static and dynamic components [27]. The static appearance is represented by a mosaic, which is constructed by aligning and integrating frames. The dynamic behavior of the moving objects is represented by their trajectories and characteristic appearances. One of the drawbacks of this representation is that its use is limited to shots containing camera operations, such as panning and zooming. When the camera is still, key frames are better suited for summarization.
- **Pictorial summary** This summary is generated as follows [28]. First a time-constrained clustering algorithm is used to label the different shots. Then, the label sequence is segmented into subsequences of minimal length that contain similar labels. These subsequences are called *story units*. A video poster is then created for each story unit. The cluster information and the duration of individual shots are used to compute the dominance value. A shot is dominant when its frequency is high or its duration is long within a story unit. A representative frame is then selected to represent a cluster of shots. Based on the dominance value of all clusters, a layout pattern is selected from a set of predesigned patterns. Figure 3 shows an example of such a layout pattern, where the top left sub-image is the most important one. Finally, the representative frames are resized into corresponding subimages in the pattern according to the dominance values of the clusters.

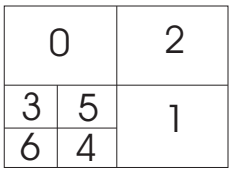


Fig. 3. Layout pattern showing the presence of three important story units (increasing values correspond to decreasing shot dominance).

B. Content annotation

To provide effective access to large multimedia collections, metadata, or data about data, is associated to the content. Interoperability with third-party applications is enabled by the use of MPEG–7 *multimedia content description interface* for content annotation, and the *synchronized multimedia integration language* SMIL for interactive video [3]. MPEG–7 defines a standardized set of descriptors for multimedia content features. The SMIL standard enables simple authoring of interactive audiovisual presentations.

1) MPEG–7 content annotation: Table I lists the MPEG–7 descriptors selected for content annotation. This list is a subset of the MPEG–7 multimedia description schemes (MDS) [29]; the descriptors were selected so as to fit the specific needs of UMA. *Description metadata* gives the version, author and history of the description. This is mainly used to record the annotation history at the content providers' side. *Media information* identifies and describes media-specific information of the multimedia data. In particular, media information provides the unique ID and URL of content, as well as information about media format and quality. This media-specific information is matched against usage environment descriptions to select the best variation. *Creation information* describes information about the creation, production and classification of multimedia content. *Semantic description* is used to describe real-life concepts or narratives which are depicted by or related to multimedia content. Creation information and semantic descriptions enable refined content browsing. Additionally, this information is matched with user preferences to sort content in terms of relevance. *Usage description* holds information about right holders and content usage. This enables the system to restrict delivery to entitled user categories. *Structure description* helps to access content segments rapidly. Video summaries finally are specified by MPEG–7 *access tools*.

Some of the above features are extracted automatically. Automatic feature extraction is usually more effective in terms of extraction time, but can lead to inaccurate results. Thus, some tuning by the user might be required. Content segments (shots) and key frames are both extracted using the algorithms mentioned in Section III-A. All media information with the exception of video quality is read from the content's file header. Perceptual video quality is predicted using models of the human visual system [30], [31]. All remaining features are annotated by hand.

2) SMIL annotation for interactive video: Interactive video contains visual elements that the user can interact with. An interactive element is defined by its shape and color, its starting time, its ending time and its position or trajectory throughout the video. Interactive elements are assigned actions to be performed when clicking on them. Possible actions include opening a web page, sending a message (e.g., email, text message or MMS), and forwarding the player to another multimedia element.

Interactive video is created as SMIL objects referenced from the content description. This means that no new video is created, but the description is used by the terminal to provide interactivity on the fly. As no new media is created, multiple interactive video sequences can be stored with almost no additional cost. Table II lists the SMIL Modules and Elements used to perform the edition of interactive video. This list is a subset of the SMIL 2.0 modules defined by the W3C [32]. The *media object module* describes the media objects used to embed dynamic links. The *layout module* is used for the

MPEG–7 DESCRIPTOR	PURPOSE	EXTRACTION
Description metadata		
`DescriptionMetadata`	Description version, author and history	Manual
Media information		
`MediaIdentification D`	Uniquely identify content	Manual
`MediaFormat D`	Information about storage medium, media file and content coding	Automatic
`MediaQuality D`	Perceptual quality rating	Auto. + Manual
`MediaInstance DS`	URL of content file	Automatic
Creation information		
`Creation DS`	Content title, abstract, creator, creation coordinates and creation tool	Manual
`Classification DS`	Content form, genre, subject, purpose, language, release, audience and review	Manual
Usage description		
`Rights datatype`	Information about right holders and content usage	Manual
Semantic description		
`AgentObject DS`	*Who*: describe objects that are persons or organizations in a narrative world	Manual
`Event DS`	*What*: describe a semantic activity in a narrative world	Manual
`SemanticPlace DS`	*Where*: describe location in a narrative world	Manual
`SemanticTime DS`	*When*: describe time in a narrative world	Manual
Structure description		
`Segment DS`	Specify temporal segments in multimedia content	Auto. + Manual
Access tools		
`SummarySegmentGroup DS`	Define video summaries using key frames	Auto. + Manual

TABLE I
MPEG–7 DESCRIPTORS FOR CONTENT ANNOTATION.

SMIL DESCRIPTOR	PURPOSE	EDITING MODE
Media Object Modules		
`video`	Define interactive video	Automatic
`img`	Define interactive image	Automatic
Layout Modules		
`layout`	Spatial layout of the links	Auto. + Manual
`region`	Spatial coordinates of a link	Auto. + Manual
`z-index`	Links layer hierarchy	Automatic
Linking Modules		
`a`	Object-dependent link information (URL, behavior, ...).	Manual
`area`	Region-dependent link information	Manual
Animation Modules		
`animate`	Temporal behavior of the links	Auto. + Manual
Timing and Synchronization Module		
`par`	Parallel playback	Automatic

TABLE II
SMIL 2.0 MODULES AND ELEMENTS FOR INTERACTIVE VIDEO.

positioning of the interactive areas on the visual rendering surface. The *linking module* defines the attributes and elements of navigational hyperlinks. The *Animation Module* contains elements and attributes for incorporating animations into a timeline. The *timing and synchronization Module* is used for synchronization between video and embedded links.

IV. CONTENT DELIVERY

Personalized content delivery involves two parties: a content server (Section IV-A) and a client (Section IV-B). The client comprises both the user and the user's access terminal (e.g., PC, PDA, mobile phone). According to the individual preferences of the user, and to the capabilities of the terminal and of the transport network, the content server delivers the most suitable content. This process is transparent for the user, leaving him off the tough task of dealing with hardware characteristics.

A. Content server

The server transparently delivers content that suits both the preferences of the connected user and the capabilities of the connected terminal and network. To achieve personalization, the server performs four successive tasks:

1) All content is sorted according to the personal preferences of the connected user. Preferred items are listed first, and undesired content is sorted out. The user then selects the item he wants to retrieve from the sorted content list;
2) When different variations of the selected content exist, the version that best fits the capabilities of the user's terminal is retained for delivery;
3) If no suitable variation is found, content is adapted to terminal capabilities on the fly;
4) The selected and possibly adapted content is streamed to the client.

CONTENT DESCRIPTOR	TERMINAL CAPABILITY	MATCHING CRITERION (DISQUALIFYING)
`FileFormat`	`TransportFormat`	**if** (FileFormat $\notin$ TransportFormat)
`VisualCoding:Format`	`ImageFormat`	**if** (Format $\notin$ ImageFormat)
`AudioCoding:Format`	`AudioFormat`	**if** (Format $\notin$ AudioFormat)
`TargetChannelBitRate`	`BitRate`	**if** (TargetChannelBitRate $>$ (VideoParameters:BitRate + AudioParameters:BitRate))
`Frame:Height`	`Resolution(vertical)`	**if** (Height $>$ Resolution(vertical))
`Frame:Width`	`Resolution(horizontal)`	**if** (Width $>$ Resolution(horizontal))
`Frame:Rate`	`Display:refreshRate`	**if** (Rate $>$ refreshRate)

TABLE III

DISQUALIFYING MATCHING CRITERIA FOR CONTENT SELECTION. WHENEVER ONE OR MORE DISQUALIFYING CRITERION IS RESPECTED, THE VARIATION IS ASSUMED TO BE NON-RENDERABLE AND IS THUS DISQUALIFIED.

CONTENT DESCRIPTOR	TERMINAL CAPABILITY	MATCHING CRITERION (REWARDING)
`VisualCoding:colorDomain`	`Display:colorCapable`	**if** ((colorCapable **and** colorDomain = {color, colorized}) **or** (!colorCapable **and** colorDomain = {binary, graylevel}))
`Pixel:BitsPer`	`Display:bitsPerPixel`	**if** (BitsPer **closest to** BitsPerPixel)
`AudioChannels`	`AudioOut:numChannels`	**if** ((AudioChannels:Front + AudioChannels:Side + AudioChannels:Rear) **closest to** numChannels)
`Pixel:BitsPer`	`Display:bitsPerPixel`	**if** (BitsPer **closest to** BitsPerPixel)
`QualityRating:RatingValue`	–	if different variations result in same priority, send the variation with highest perceived VideoQuality

TABLE IV

NON-DISQUALIFYING MATCHING CRITERIA FOR CONTENT SORTING. EACH VARIATION THAT FULFILLS A MATCHING CRITERION GETS POSITIVELY REWARDED. AFTER THE PROCESS, THE MOST REWARDED VARIATION IS RETAINED FOR DELIVERY.

1) User preferences handling: User preferences are utilized by consumers for accessing multimedia content that fits their personal preferences. By using the user's preferences in conjunction with a history of the actions that he has carried out over a specific period of time, the server dynamically updates the user profile. In our framework, user preference and the usage history are described using MPEG–7 user interaction tools [29]. MPEG–7 user interaction tools contain two types of preferences: *filtering and search preferences*, and *browsing preferences*. The former are used to describe user's filtering or searching preferences for multimedia content in terms of attributes related to the creation, the classification and the source of the content. The latter describe user preferences pertaining to navigation of and access to content. In particular, a user may express preferences of the type and content of video summaries. User preferences and usage history are stored in the user preferences database maintained by the server.

User preferences handling is based on a continuous check of the user's interaction with the system. Depending on the content the user is accessing, the system dynamically updates the MPEG–7 user profile. Updates depend on the category the selected multimedia item belongs to, and on past user's history accessing categorized multimedia material. In order to take both these factors into account, the user profile is update according to a Q-Learning based mechanism [33], were states (categories in the user profile) belonging to branches with similar semantic meaning get rewarded; non selected media are punished (negatively rewarded). As a result of the continuous process of dynamically adjusting and updating the user profile, the system is able to select and to filter all the annotated multimedia content according to the profile. The selection of content from the multimedia database according to the user profile is based the following guidelines:

1) Content in the database is filtered based on the user's preferences. That is, content belonging to undesired categories is sorted out.
2) Multimedia content is sorted according to the user preferences: the more relevant preferences (those that have been more actively accessed in the past) are presented before those that have been accessed less, or not at all.

As a result of the process, a list of classified, sorted and pruned multimedia content regarding the user's preferences is generated.

2) Content selection: Whenever more than one variation of the requested content exists on the server, the version that best fits the capabilities of the user's terminal must be determined. This is the task of content selection. To select content, the media-specific information of all variations (Table I) is matched with the capabilities of the terminal (Table V). A set of matching criteria establishes whether variations can be rendered, and which variation best fits the terminal. *Disqualifying matching criteria* establish whether a variation can be rendered by the terminal. These criteria are summarized in Table III. Whenever one or more disqualifying criterion is respected, the variation is assumed to be non-renderable and is thus disqualified. *Non-disqualifying matching criteria* are shown in Table IV. In order to find the most suitable item among the variations that can be rendered, non-disqualifying matching criteria are applied to all such variations. Each variation that fulfills a matching criterion is positively rewarded.

After the process, the most rewarded variation is retained for delivery.

3) Content adaptation: The diversity of networks and terminals in a realistic UMA environment makes it unattainable to generate a distinct variation for each profile of capabilities. Thus, content adaptation is needed whenever no suitable variation is found. Content adaptation refers to the modification of multimedia content to fit the decoding capabilities of the connected terminal. Adaptation is performed at the time of streaming and has to be achieved in real-time. Two distinct ways to adapt video content have been explored: video transcoding and scalable video coding. Generally speaking, transcoding can be defined as the conversion of one coded signal to another. Work on video transcoding traditionally focuses on reducing the bit rate, the spatial resolution and the temporal resolution [34]. Bit rate reduction is performed to meet an available channel capacity. The most straightforward way to achieve this is to decode the video bit stream and to fully re-encode the reconstructed signal at the new rate. However, significant complexity savings can be achieved by reusing information contained in the original incoming bit streams. For instance, Assuncao and Ghanbari [35] perform dynamic bit-rate reduction of MPEG–2 streams in the frequency domain by using approximate matrices for fast computation of MC–DCT blocks. Transcoding to achieve spatial and/or temporal resolution reduction has been studied to accommodate mobile devices with limited display capabilities and processing power. Similar to bit-rate reduction, subsequent decoding, spatial-domain and/or temporal-domain downsampling, followed by a full re-encoding leads to complexity issues. To achieve compressed domain resolution reduction, new motion vectors are composed out of a set of input motion vectors by motion vector mapping [36].

The objective of scalable coding is to encode the video once and then to obtain lower qualities, spatial resolutions, and/or temporal resolutions by simply truncating certain layers or bits from the original stream [37]. With traditional scalable coding schemes, e.g., as defined by MPEG–2 Video [38], the signal is encoded into a base layer and a few enhancement layers. The enhancement layers add spatial, temporal, and/or SNR quality to the reconstructed base layer. More recently, a new form of scalability, known as fine granular scalability (FGS), has been developed and adopted by the MPEG–4 Visual standard [39]. In contrast to conventional scalable coding schemes, FGS allows for a much finer scaling of bits in the enhancement layer. This is accomplished through a bit-plane coding method of DCT coefficients in the enhancement layer, which allows the enhancement layer bit stream to be truncated at any point. In this way, the quality of the reconstructed frames is proportional to the number of enhancement bits received.

In the context of UMA, video transcoding and scalable coding coexist to meet different needs. Scalable coding does not require any significant processing power at the server side and thus provides low-cost flexibility to meet the target bit rate and resolution. Video transcoding on the other hand is computationally expensive due to the real-time constraints. However, the coding efficiency of a cascaded transcoding architecture that fully decodes and re-encodes the video according to the new requirements is often better than traditional scalable coding [40]. Therefore, transcoding is used either when no scalable video is available, or when a high coding efficiency must be achieved.

4) Content streaming: Content delivery can be carried out in two different ways: progressive download and streaming [41]. During download, e.g., from a web server, data is pushed out to the client as fast as possible. Network conditions are thereby ignored, and the client has no control over the delivery process (e.g, fast forward). Moreover, since the client must first receive the content of the file in its entirety, an unacceptable long delay is introduced if the content is of any significant size. Streaming servers take into account varying network conditions and content file characteristics to provide traffic shaping, bandwidth allocation and quality assessment [42]. Owing to the highly bursty nature of compressed video streams, traffic shaping and traffic smoothing are required for efficient utilization of bandwidth and network resources at various points in a network. Bandwidth allocation distributes available resources among video sessions and layers. Quality assessment maintains the quality of service under varying network conditions. Since the UMA server by definition has to handle a wide variety of networks, clients and media, streaming will be the favorite delivery mode.

B. Client

The client comprises both the user and the user's terminal. The user can be any person or agent, characterized by his own set of user preferences. Initial preferences are acquired from the user by filling a form. Depending on the content the user is accessing, the system dynamically updates the MPEG–7 user profile (Section IV-A).

Possible access terminals range from high-performance appliances like digital TV sets and personal computers (PC), down to mobile devices such as personal digital assistants (PDA), mobile phones and wearable computers. Each terminal is characterized by a set of capabilities that describe the terminal's own technical features. Terminal capabilities are stored in the appliance and accessed by the content server at the time of connection. Alternatively, capabilities of devices with limited memory can be stored in the content server and retrieved according to the terminal's individual ID. Terminal capabilities are described using MPEG–21 DIA *usage environment description tools* [43]. Table V lists the selected descriptors. *Codec capabilities* specify the decoding capabilities of a connected terminal. These includes the transport, video and audio formats that can be decoded, as well as video and audio bit rate specifications. *Input and output capabilities* describe the I/O capabilities of the terminal. For video, resolution, color depth and refresh rate of the display are specified. Audio parameters include the output frequency range and the number of audio channels.

V. Discussion

The proposed content preparation and delivery framework has been implemented in the EC–funded R&D project PERSEO and tested by selected groups of end-users. PERSEO

MPEG–21 DESCRIPTOR	PURPOSE
Codec capabilities	
Decoding capabilities	
`TransportFormat`	Describes the transport formats the terminal is capable of decoding.
`VideoFormat`	Describes the video formats the terminal is capable of decoding.
`AudioFormat`	Describes the audio formats the terminal is capable of decoding.
Video parameters	
`BitRate`	Nominal bit rate in bit/s.
`Maximum`	Max. value for BitRate in case of variable bit rate.
`Average`	Avg. value for BitRate in case of variable bit rate.
Audio parameters	
`BitRate`	Nominal bit rate in bit/s.
`Maximum`	Max. value for BitRate in case of variable bit rate.
`Average`	Avg. value for BitRate in case of variable bit rate.
Input/output capabilities	
Display capabilities	
`Resolution`	Display resolution in pixels.
`BitsPerPixel`	Display color depth in bits.
`ColorCapable`	Describes whether display is color capable.
`RefreshRate`	Display refresh rate in Hz.
Audio output capabilities	
`LowFrequency`	Lower bound of audio frequency range in Hz.
`HighFrequency`	Higher bound of audio frequency range in Hz.
`NumChannels`	Number of supported output audio channels.

TABLE V
MPEG–21 DESCRIPTORS FOR TERMINAL CAPABILITIES.

is a UMA system that covers all the aspects of multimedia production, post-production, annotation, database management and final cross-media and multi-device publication.

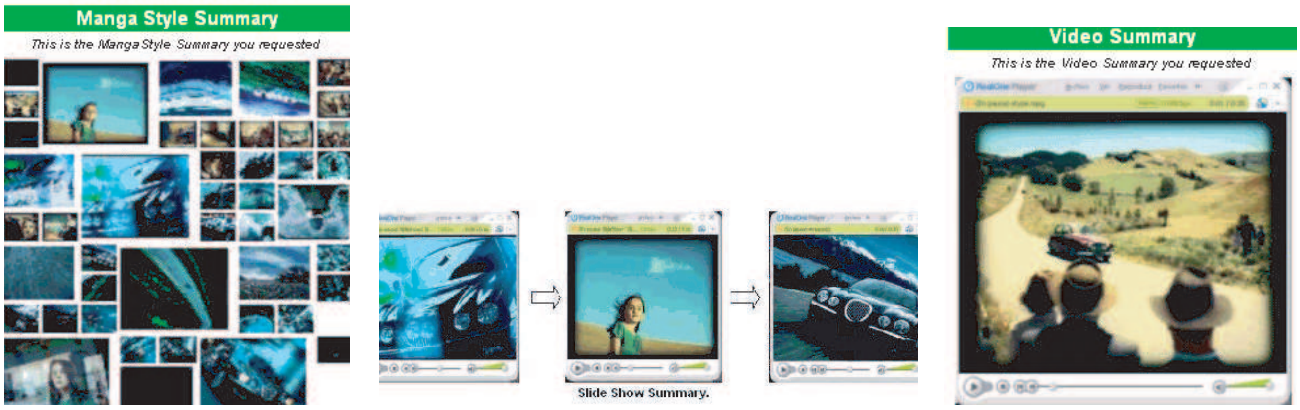
A. PERSEO testbed

The PERSEO testbed comprises two distinct modules: content preparation software, and the UMA server. The subdivision into two modules reflects the organization of the framework in Figure 1. Content preparation software provides facilities for summary generation and for content annotation. Three summary representations are supported by the application: pictorial summaries, sequential summaries using key frames and sequential summaries using key sequences. Pictorial summaries show the most important frames laid out in a web page by chronological and importance criteria. This representation resembles a comics book and has thus been given the name *Manga style summary*. *Slide show summaries* sequentially display the key frames of the video with a latency of a few seconds between them. The *video summary* is a trailer of the video: a set of key sequences in chronological order. These summary representations are depicted in Figure 4. MPEG–7 content annotation is done aid of a graphical user interface (GUI). The customizable GUI in Figure 5(a) is subdivided into input areas corresponding to the different annotation features required by the application at hand. User input is automatically translated to MPEG–7 by the software. Another GUI is provided for interactive video authoring (Figure 5(b)). Interactive elements are drawn onto the video frames, and their temporal extent is specified in the timeline.

The UMA server handles user preferences, content selection, content adaptation and content streaming. Initial preferences are acquired from the user by filling a form and updated dynamically according to a Q-Learning based mechanism. Content selection is achieved by matching MPEG–7 metadata with the MPEG–21 terminal capabilities description. Real-time content adaptation operations performed by the server include bitrate reduction, spatial resolution reduction and temporal resolution reduction. The interaction of the server with a PC, a color PDA and a GPRS-enabled mobile phone is depicted[1] in Figure 6. (i) After the login procedure, possible options are displayed in a menu. The user can access a content catalogue sorted according to his personal preferences or according to his current location (when available), perform a generic search, edit his own profile, or access other services. (ii) The profile-based catalogue shows a sorted list of playable media items, together with available summaries. (iii) Selected content is displayed using the adequate frame rate, resolution and color depth.

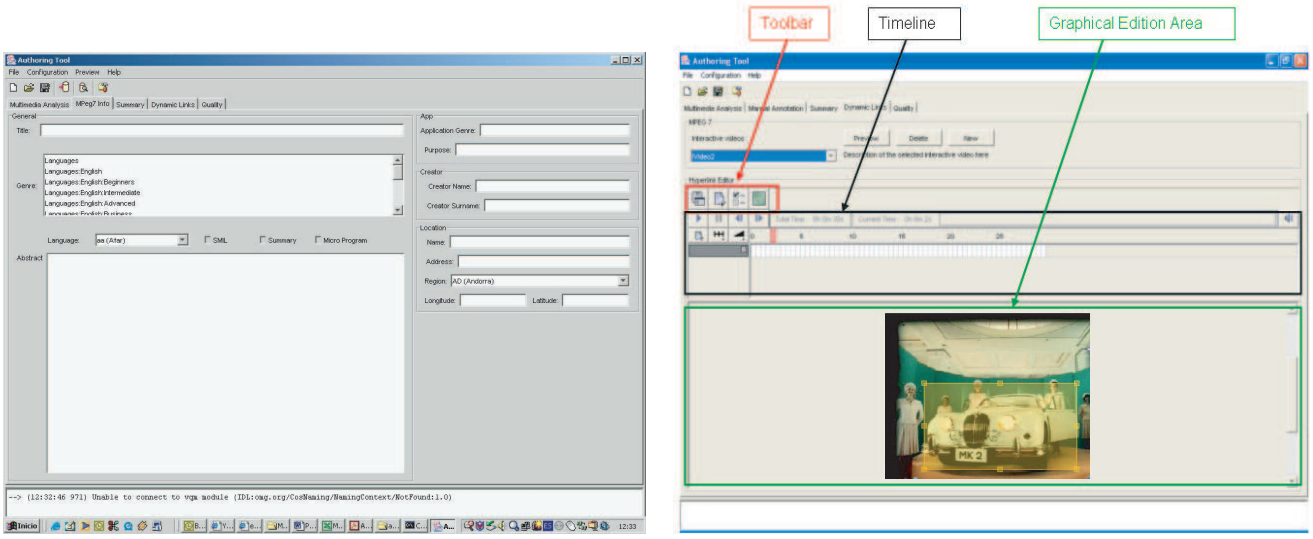
In our implementation of the PERSEO testbed, commercial software has been mixed with custom-implemented parts. A wide variety of media formats is streamed by using four commercial streamers: (i) Real Helix Server version 9 – Basic, (ii) Microsoft Media Services 9 Series, (iii) Apple QuickTime Streaming Server 4.1.1 and (iv) Java Media Framework version 2.1.1. Content descriptions and user preferences are stored in Apache Xindice version 1.1b native XML databases. Video for annotation is displayed by Java Media Framework version 2.1.1. Real-time content adaptation is performed by Real Helix Producer 9.2. The remainder of the software has been custom-implemented in Java and C++ programming languages. Java has mainly been utilized to realize graphical user interfaces, whereas C++ is used for time-critical feature extraction tasks. All individual software modules are linked by CORBA interfaces.

[1]For improved readability, simulations that accurately reflect the actual PERSEO client display are shown.



(a) (b) (c)

Fig. 4. Three summary representations are supported by PERSEO content preparation software. (a) Pictorial summary (*Manga Style*). (b) Sequential summary using key frames (*Slide Show*). (c) Sequential summary using key sequences (*Video Summary*).



(a) (b)

Fig. 5. Graphical user interfaces of PERSEO content preparation software. (a) MPEG–7 content annotation. (b) Interactive video authoring.

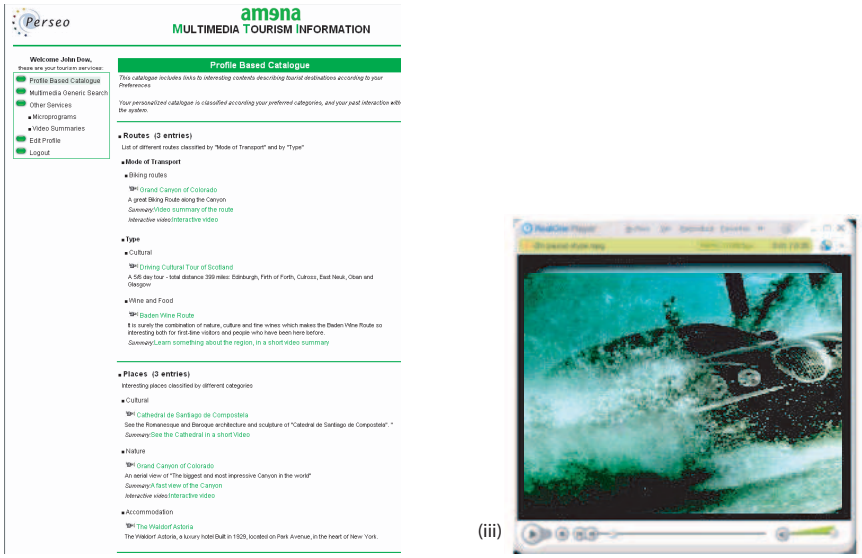
B. Usability validation

The PERSEO testbed has been utilized to provide two different applications to a selected group of end-users: e–learning and tourism. The e–learning application provides multimedia course material to professional and non-professional users, using static and mobile devices. In the tourism application, video material on various European cities is accessed by mobile devices. Usability validation has been achieved by focus group techniques using evaluation forms based on the quality attributes required for the developed services. Focus groups validation considered two main segments to analyze: (i) a residential segment aged 18 to 35. This segment was split in two focus groups, a first group of young people aged 18 to 25, and a second one aged 25 to 35. (ii) Professional users. Here, only one focus group has been formed.

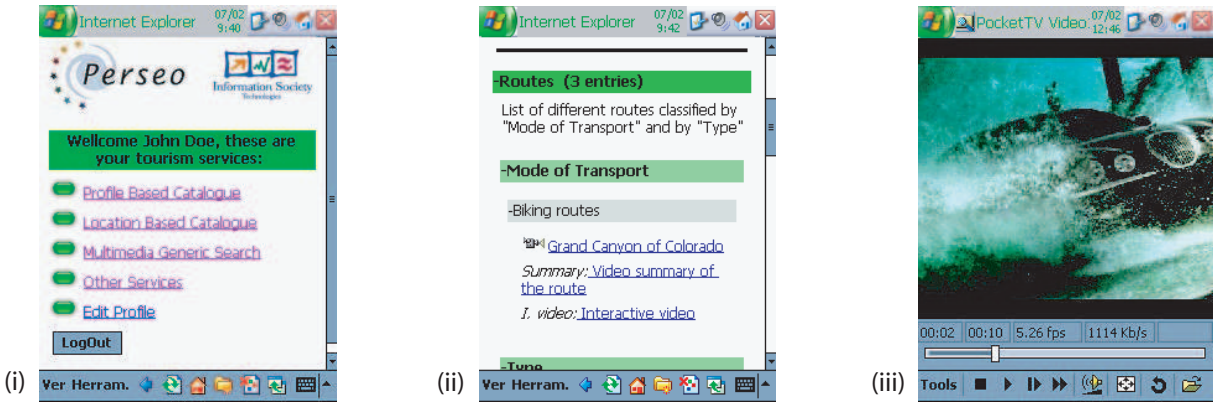
The objectives of the user validation have been grouped into three main objectives: 1) Validate the usefulness or utility of the service provided: are users interested in getting the service? Is the service relevant? 2) Validate the usability of the service provided: is the service good enough to be usable? This covers aspects such as effectiveness and efficiency, but also ergonomics, content quality and product robustness. 3) Contribute to validate the market pertinence and profitability.

The evaluation of the service has been positive in general terms in all the groups, and somewhat more critical in the segment of 25 to 35 years and in professional users. The service is considered to be attractive, and has been very useful in very specific and precise situations. The user validation phase leads to the following conclusions:

- **Personalization** is considered as a positive feature, due



(a)



(b)

(c)

Fig. 6. Interaction of the UMA server with different terminals. (a) PC. (b) Personal digital assistant. (c) GPRS-enabled mobile phone. Three successive steps are depicted: (i) Options menu; (ii) Profile-based content catalogue; (iii) Video streaming.

to comfort, effectiveness and speed that the service gains. Due to the very nature of personalization, these advantages are not distinctly noticed in the direct use of the service.

- **Video Summaries** are very well considered, as they help browsing large content collections efficiently. Summaries are particularly appreciated for generic search.
- **Generic search** on the content list is considered easy to use and positive. The result list should be limited in order to minimize rejection of the service. An extensive list of video can generate deception and rejection of the service. The user's familiarity with Internet search engines contributes to the acceptance of the idea.
- The UMA service has been found **particularly useful in mobile applications**.
- The video must have strong and practical information content. **The service can only be as good as the proposed content.**
- Regarding the **usability** of the developed applications, all the groups considered it as an easy to use service. The basic nature of the groups made that they were used to manage mobile phones and PDAs. The users felt confident that, with a little training, they would learn to efficiently use all the features of the system in a short time period.

VI. Conclusion

This paper introduces a personalized content preparation and delivery framework for universal multimedia access. Content preparation deals with the generation of summaries by automatic video analysis techniques, and with the annotation of content using MPEG–7 for search and retrieval and SMIL for interactive video. Content delivery handles individual user preferences, content selection, real-time content adaptation and streaming in a seamless way. Thus, rich multimedia content is delivered to various users, terminals and networks transparently. The proposed framework has been implemented and tested by the consortium of the EC–funded project PERSEO. Personalization, video summaries and the overall usability of the UMA framework have been positively evaluated by focus groups. The use of open standards for content annotation and user preferences favors interoperability with third-party applications and upcoming hardware.

The present work can be extended in many ways. Additional modalities such as text, speech, stereoscopic video and 3–D will enable novel applications and support delivery to digital radio receivers, text terminals, wearable computers with head-mounted displays, ... Transmoding will be a key technology to the seamless integration of multiple modalities. Additional personalization criteria such as geographical location and time can be taken into account as well to provide selective services.

Acknowledgment

This work has been funded by the European Commission under contract IST-2000-28443 Personalized multichannel services for advanced multimedia stream management (http://www.perseoproject.org).

References

[1] A. Vetro, C. Christopoulos, and T. Ebrahimi, "Universal multimedia access," *IEEE Signal Processing Magazine*, vol. 20, no. 2, p. 16, March 2003.

[2] A. Perkis, Y. Abdeljaoued, C. Christopoulos, T. Ebrahimi, and J. Chicharo, "Universal multimedia access from wired and wireless systems," *Birkhauser Boston transactions on Circuits, Systems and Signal Processing*, vol. 10, no. 3, pp. 387–402.

[3] F. Pereira and I. Burnett, "Universal multimedia experiences for tomorrow," *IEEE Signal Processing Magazine*, vol. 20, no. 2, pp. 63–73, March 2003.

[4] P. van Beek, J. Smith, and T. Ebrahimi, "Metadata-driven multimedia access," *IEEE Signal Processing Magazine*, vol. 20, no. 2, pp. 40–52, March 2003.

[5] E. Fossbakk, P. Manzanares, J. Yago, and A. Perkis, "An MPEG–21 framework for streaming media," in *Proceedings of IEEE Workshop on Multimedia Signal Processing 2001*, October 2001, pp. 147–152.

[6] H. Sun, A. Vetro, and K. Asai, "Resource adaptation based on MPEG–21 usage environment descriptions," in *Proceedings of 2003 IEEE International Symposium on Circuits and Systems, ISCAS'03*, vol. 2, May 2003, pp. II:536–539.

[7] A. Perkis, J. Zhang, T. Holvorsen, J. Kjode, and F. Rivas, "A streaming media engine using digital item adaptation," in *Proceedings of IEEE Workshop on Multimedia Signal Processing 2002*, December 2002, pp. 73–76.

[8] J. Bormans, J. Gelissen, and A. Perkis, "MPEG–21: The 21st century multimedia framework," *IEEE Signal Processing Magazine*, vol. 20, no. 2, pp. 53–62, March 2003.

[9] A. Vetro and H. Sun, "Media conversion to support mobile users," in *Proceedings of Canadian Conference on Electrical and Computer Engineering 2001*, vol. 1, May 2001, pp. 607–612.

[10] K. Lee, H. Chang, S. Chun, H. Choi, and S. Sull, "Perception-based image transcoding for universal multimedia access," in *Proceedings of 2001 IEEE International Conference on Image Processing, ICIP'01*, vol. 2, October 2001, pp. 475–478.

[11] M. van der Schaar and H. Radha, "Adaptive motion-compensation fine-granular-scalability (AMC–FGS) for wireless video," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 6, pp. 360–371, June 2002.

[12] C. Wang, C. Tsai, H. Chuang, Y. Lin, J. Chen, K. Tong, F. Chang, C. Tsai, S. Lee, T. Chiang, and H. Hang, "FGS-based video streaming test-bed for MPEG–21 universal multimedia access with digital item adaptation," in *Proceedings of the 2003 IEEE International Symposium on Circuits and Systems, ISCAS'03*, vol. 2, May 2003, pp. II:364–367.

[13] T. Echigo, K. Masumitsu, M. Teraguchi, M. Etoh, and S. Sekiguchi, "Personalized delivery of digest video managed on MPEG–7," *Proceedings of International Conference on Information Technology: Coding and Computing*, pp. 216–220.

[14] Y. Chen, H. Huang, R. Jana, S. John, S. Jora, A. Reibman, and B. Wei, "Personalized multimedia services using a mobile service platform," *Proceedings of Conference on Wireless Communications and Networking, WCNC2002*, vol. 2, pp. 918–925.

[15] J. Monaco, *How to Read a Movie*. Oxford, 1981.

[16] H. Zhang, J. Wu, D. Zhong, and S. Smoliar, "An integrated system for content-based video retrieval and browsing," *Pattern Recognition*, vol. 30, no. 4, pp. 643–658, April 1997.

[17] J. Boreczky and L. Rowe, "Comparison of video shot boundary detection techniques," in *Proceedings of SPIE International Symposium on Electrical Imaging: Storage and Retrieval for Image and Video Databases*, San Jose, USA, 1996, pp. 170–179.

[18] R. Zabih, J. Miller, and K. Mai, "Feature-based algorithms for detecting and classifying scene breaks," in *Proceedings of ACM Conference on Multimedia*, San Fransisco, USA, November 1993, pp. 189–200.

[19] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679–698, November 1986.

[20] D. Huttenlocher, G. Klanderman, and W. Rucklidge, "Comparing images using the Hausdorff distance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 9, pp. 850–863, September 1993.

[21] P. Bouthemy, M. Gelgon, and F. Ganansia, "A unified approach to shot change detection and camera motion characterization," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, no. 7, pp. 1030–1044, October 1999.

[22] M. Basseville, "Detecting changes in signals and systems – a survey," *Automatica*, vol. 24, no. 3, pp. 309–326, 1988.

[23] A. Nagasaka and Y. Tanaka, "Automatic video indexing and full-video search for object appearances," in *Proceedings of 2nd Working Conference on Visual Database Systems*, September/October 1992, pp. 119–133.

[24] W. Wolf, "Key frame selection by motion analysis," in *Proceedings of International Conference on Acoustics, Speech, and Signal Processing, ICASSP'96*, vol. 2, 1996, pp. 1228–1231.

[25] B. Horn and B. Schunck, "Determining optical flow," *Artificial Intelligence*, vol. 17, pp. 185–203, 1981.

[26] D. Zhong, H. Zhang, and S. Chang, "Clustering methods for video browsing and annotation," in *Proceedings of SPIE International Symposium on Electrical Imaging: Storage and Retrieval for Image and Video Databases*, San Jose, USA, 1996, pp. 239–246.

[27] M. Irani and P. Anandan, "Video indexing based on mosaic representations," *Proceedings of the IEEE*, vol. 86, no. 5, pp. 905–921, May 1998.

[28] M. Yeung and B. Yeo, "Video visualization for compact presentation and fast browsing of pictorial content," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, no. 5, pp. 771–785, October 1997.

[29] ISO/IEC, "Information Technology – Multimedia Content Description Interface – Part 5: Multimedia Description Schemes," ISO/IEC JTC 1/SC29/WG11, Tech. Rep. ISO/IEC FDIS 15938-5:2001, 2001.

[30] A. Hekstra, J. Beerends, D. Ledermann, F. de Caluwe, S. Kohler, R. Koenen, S. Rihs, M. Ehrsam, and D. Schlauss, "PVQM - a perceptual video quality measure," *Signal Processing: Image Communication*, vol. 17, no. 10, pp. 781–798, November 2002.

[31] Z. Yu, H. Wu, S. Winkler, and T. Chen, "Vision-model-based impairment metric to evaluate blocking artifacts in digital video," *Proceedings of the IEEE*, vol. 90, no. 1, pp. 154–169.

[32] W3C, "W3c recommendation of the synchronized multimedia integration language (smil) 2.0," W3C SYMM Working Group, Tech. Rep. W3C REC-smil20, 2001.

[33] J. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, pp. 279–292, 1992.

[34] A. Vetro, C. Christopoulos, and H. Sun, "Video transcoding architectures and techniques: an overview," *IEEE Signal Processing Magazine*, vol. 20, no. 2, pp. 18–29, March 2003.

[35] P. Assuncao and M. Ghanbari, "A frequency-domain video transcoder for dynamic bit-rate reduction of MPEG–2 bit streams," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 8, no. 8, pp. 953–967, December 1998.

[36] T. Shanableh and M. Ghanbari, "Heterogeneous video transcoding to lower spatio-temporal resolutions and different encoding formats," *IEEE Transactions on Multimedia*, vol. 2, no. 2, pp. 101–110, June 2000.

[37] Y. Wang, J. Ostermann, and Y.-Q. Zhang, *Video Processing and Communications*, 1st ed. Prentice Hall, 2001.

[38] ISO/IEC, "Information technology – generic coding of moving pictures and associated audio information: Video, 2nd ed." ISO/IEC JTC1/SC29/WG11, Tech. Rep. ISO/IEC FDIS 13818-2:2000, 2000.

[39] ——, "Information technology – coding of audio-visual objects – part 2 visual – amendment 2: Streaming video profiles," ISO/IEC JTC1/SC29/WG11, Tech. Rep. ISO/IEC FDIS 14496-2:2001/Amd 2:2002, 2002.

[40] N. Bjork and C. Christopoulos, "Transcoder architectures for video coding," *IEEE Transactions on Consumer Electronics*, vol. 44, no. 1, pp. 88–98, February 1998.

[41] F. Halsall, *Multimedia Communications*. Addison-Wesley, 2001, ch. The World Wide Web: Audio and video, pp. 965–971.

[42] S. Gringeri, K. Shuaib, R. Egorov, A. Lewis, B. Khasnabish, and B. Basch, "Traffic shaping, bandwidth allocation, and quality assessment for MPEG video distribution over broadband networks," *IEEE Network*, vol. 12, no. 6, pp. 94–107, November/December 1998.

[43] ISO/IEC, "Information technology – multimedia framework – part 7: Digital item adaptation," ISO/IEC JTC1/SC29/WG11, Tech. Rep. ISO/IEC FDIS 21000-7:2003, 2003.